University of Florence, University of Perugia, INdAM, CIAFM consortium

DOCTORAL PROGRAMME
IN MATHEMATICS, COMPUTER SCIENCE AND STATISTICS

CURRICULUM IN COMPUTER SCIENCE, MATHEMATICS E STATISTICS
XXXV CYCLE

Administrative Headquarters: University of Florence
Coordinator: PhD. Prof. Matteo Focardi

Innovative approaches to some modern complex problems

Academic Discipline (SSD) INF/01

Doctoral candidate
Damiano Perri

Supervisors:
PhD. Prof. Osvaldo Gervasi
PhD. Prof. Sumi Helal

Coordinator
PhD. Prof. Matteo Focardi

Years 2019/2022

Contents

List of Figures

Introduction

Over the course of the three-year doctoral period, various topics ranging from machine learning, deep neural networks, data augmentation, virtual and augmented reality, cloud computing and web programming, and finally, edge computing has been deepened and explored. The thesis is organised into four chapters, all of which are closely interlinked but with a different topic underpinning them.

Chapter 1 deals with the research strands based on machine learning and describes the studies for the optimisation of convolution operators in the field of GPGPU computing, the techniques tackled to solve emotion recognition (carried out by focusing only on the mouth of the subjects), the identification of proteins from their amino acid chain, and the analysis of images for the recognition of certain skin diseases.

Chapter 2 of the thesis focuses on the research we did on the topics of virtual reality and augmented reality, which was a cornerstone of my PhD. We tried to improve the quality of teaching certain subjects, such as mathematics, through the use of virtual reality and augmented reality. For example, we conducted interesting research with schools in Umbria, meeting students and professors, to whom we showed use cases we had realised for the representation of mathematical functions and three-dimensional objects useful for teaching. We then focused on the possibility of recreating three-dimensional objects from real objects, real Digital Twins. This allowed us to start a collaboration with a school of higher education in Umbria, the ITS Umbria Academy, for the realisation of some digital twins of industrial machinery useful for Industry 4.0. The knowledge gained in this field was used in another strand, which focused on the tele-rehabilitation of people with neurological diseases. We then applied the knowledge obtained through the use of virtual reality software to machine learning, with which an alternative technique was developed to the classic data augmentation that is often performed during the learning phase of neural networks.

Chapter 3 deals with cloud computing, the main strands of research being the creation of a scalable cloud architecture that exploits the potential of open-source software. Web Apps, web programming and the study of Cloud infrastructures have been fundamental. Applications such as LibreEOL were used to tell and publicise our work, and the fundamental insights into Cloud computing led us to receive an important invitation to the AWS summit, where Amazon Italy itself invited us on stage in Milan to talk about how Open Source technologies, Machine Learning and Virtual Reality have contributed positively to the open source community. The results obtained from the study of Cloud infrastructures have been published in scientific journals and represent a milestone in my doctorate.

Chapter 4 of the thesis consists of the results obtained from the profound and efficient collaboration between our research group and Professor Sumi Helal's[1] research group in the field of IoT. Our experience with cloud computing was indeed crucial and a prerequisite for understanding the complex topics he suggested. The aim of the collaboration was the use of artificial intelligence techniques, machine learning and artificial neural networks for the optimisation of smart city architectures, which require the following three layers: cloud, edge, and sensors. In fact, it is well known that the cloud has costs that scale according to how it is used. This means that if it is used in a very important way, as in the case of a smart city, then the costs could be unsustainable. In particular, it is possible to optimise the push and pull phases of information between the various layers, for example, by aggregating it or sending it only when strictly necessary based on statistical inference techniques. We are also collaborating to model a dataset, which will be made Open Source via the Kaggle platform, to train neural networks to perform the best actions regarding the management of data traffic coming from homes with IoT sensors inside them that monitor parameters of various kinds (such as temperature, light level humidity, etc.) so as to minimise the amount of data exchanged over the network, minimise the use of the cloud, and maximise the use of edge computing, while still guaranteeing the optimal functioning of the infrastructure in the event that it needs to scale up to a level where it can be operated over a city with tens of thousands of connected homes. This three-year period of work, which has seen the development of four distinct research strands, has also been very fruitful in terms of scientific publications. In fact, this paper will describe and list most of the publications produced by the research group and, in particular, those of which the candidate is the author. The high number of 15 publications as conference proceedings, seven journal publications and two book chapters have been achieved. At the time of writing this paper, two other articles have been submitted to scientific journals, and others are planned for the

[1] https://www.cise.ufl.edu/helal-abdelsalam-sumi/

coming months, a clear sign of how hot and extremely topical these issues have been for the scientific community in these years. The results obtained from the study of cloud infrastructures have been published in scientific journals and represent a milestone in my doctorate.

Chapter 1

Deep Learning approaches for solving complex problems

This chapter describes the first of four research strands that have been addressed during the three-year PhD programme. Machine learning is one of the most studied topics by researchers today.

1.1 Introduction

The idea behind machine learning is to teach the machine to perform certain tasks without programming or directly writing the algorithm that solves the problem. For example, we could imagine that we want to teach a machine to recognise different types of animals, different geometric shapes, or different car models. Operations such as the one just described were extremely complicated, if not a type of science fiction, until a few decades ago, for which we could estimate onerous hardware requirements. Today, thanks to the advancement of the computational capacity of modern computers, it is possible to perform complex mathematical operations, the basis of learning neural networks, using not only environments that facilitate the writing of code through the use of high-level frameworks such as Keras[1], but there are even services that provide cloud environments that students can use to perform these tasks, such as Google colab[2] or Kaggle[3]. This chapter contains some of the most important work we have produced following this line of research.

Section 1.2 describes a work path whose aim is to optimise the execution

[1] https://keras.io/about/

[2] https://research.google.com/colaboratory/faq.html

[3] https://www.kaggle.com/

speed of neural networks by selecting the best possible convolution operator according to the network to be used and the input chosen. This work took several years and the results have been published in various articles. The first paper was published in a conference [1], the second paper was published in a journal [2]. In both papers, the candidate is one of the authors.

Scientific HPC applications are built on top of monolithic parallel routines that are often customised for a specific target architecture. With the advent of data-driven applications such as deep learning and graph analytics, the traditional library design loses performance portability mainly due to the unpredictable size and structure of the data on the wide range of hardware available. The convolution operator, the core of a Convolutional Neural Network (CNN), is a notable example where it is hard to determine the optimal method and implementation for a given input or layer [3]. Furthermore, it can be implemented in different ways by using direct convolution [4], Generic Matrix Multiplication (GEMM) [5], Winograd minimal filtering algorithm [6] or, FFT [7]. Regardless of the method adopted, the performance of each implementation can vary greatly even among the layers of a single CNN. For example, by using the Winograd algorithm, the number of multiplications passes from 36 to 16 for (4×4) and (3×3) filters. Winograd-based convolution can reduce the computational time by up to 4x [8]. However, the improvement obtained shows a discrepancy from the theory due to several implementation variables [9]. Another example would be in the context of Automated Machine Learning (AutoML) [10]. Here, the architecture of the DNN is not known a priori, and as a consequence, from an engineering perspective, the performance is not predictable. Also, Generic Matrix Multiplication (GEMM), which is a widely used building block in convolution implementations, requires specific optimizations in order to scale-up over different input dimensions [11]. Several BLAS implementations for GEMM provide fast performance on a target architecture by assuming a fixed data size, layouts and structure (e.g. square matrices) [12, 13, 14]. However, such user-transparent implementations are selected by hand-written heuristics based on simple decision rules that are not able to generalize the wide range of data used in practice. For example, Nvidia heuristics selects the best algorithm (relative speed-up $0.97 - 1.0$) in the 33% of the instances from DeepBench [15]. On the contrary, by changing the dataset, the performance decreases. The heuristics correctly chooses the best GEMM implementation in the 10% of the matrices only.

Section 1.3, important work is described that was carried out with the collaboration of the chemistry department of the University of Perugia, for the classification of proteins, based on the sequence of amino acids that make them up. The aim is to be able to discriminate whether amino acid sequences characterise true proteins or false proteins that do not exist in nature. The research path being presented has been published in a conference paper, and the candidate is one of the authors[16]. The paper describes research that represents the evolution of that described in the paper published in 2020[17]. The classification of the geometric structures of proteins and the individuation of possible simple criteria to base their discrimination are complex tasks and longstanding issues in chemical sciences. To investigate the relationships between structure and activity and for a satisfactory theoretical understanding of the protein folding process, the ability to assess the "correctness" and similarity of possible spatial arrangements of such macromolecules is a prerequisite. The recently increased practicability of computational approaches based on Deep Learning and Neural Networks further motivates renewed efforts in such direction since it permits one to resort to approaches based on the search for hidden patterns and regularities across large set of experimentally resolved protein structures. In this work, a Convolutional Neural Network is again aimed at classifying as "true" or "false" a given structure, but the CNN has been developed after significant new improvements to the original approach for the recognition of the geometric structures. The idea was not to lose valuable spatial information regarding the shape of the protein to be examined, so we moved beyond the molecule model as a simple sequence of amino acids to get to a more effective and realistic description preserving spatial information. To this purpose, we exploited the well-known suitability of Convolutional Neural Networks for image analysis, where they are particularly appreciated in recognition of images and their characteristics, both on two-dimensional or three-dimensional objects, through the extraction of particular features from images so that different kinds of objects, like people or things, can be correctly classified.

In the section 1.4 are discussed the techniques used for image recognition and classification using convolution neural networks that were trained using transfer learning and continuous learning. In particular, in the Subsection 1.4.1, a number of skin diseases are analysed using artificial neural networks in order to instruct a machine to correctly classify the type of disease among a number of possible classes. The research path being presented has been published in a conference, and the candidate is one of the authors[18]. Training neural networks for automated diagnosis of pigmented skin lesions can be difficult due to the small size and lack of diversity of available datasets of dermatoscopic images. The HAM10000 ("Human Against Machine with 10000 training images") dataset is a collection of dermatoscopic images from different populations acquired and stored by different modalities. We used the benchmark dataset, with a small number of images and a strong imbalance among the 7 different types of lesions, to prove the validity of our approach, which is characterized by good results and light usage of resources. Exploiting a highly engineered convolutional neural network with transfer learning, customized data augmentation and a non-adaptive optimization algorithm, we show the possibility of obtaining a final model able to precisely recognize multiple categories, although scarcely represented in the dataset. The whole training process has a limited impact on computational resources, requiring no more than 20 GB of RAM space. Dermatoscopy is often used to get better diagnoses of pigmented skin lesions, either benign or malignant. With dermatoscopic images is also possible to train artificial neural networks to recognize pigmented skin lesions automatically. Nevertheless, training requires the usage of a large number of samples, although the number of high-quality images with reliable labels is either limited or restricted to only a few classes of diseases, often unbalanced.

The Subsection 1.4.2 illustrates a research pathway that aims to recognise people's emotions by analysing only a small portion of their face, characterised by the mouth. The research path being presented has been published in a journal, and the candidate is one of the authors[19] and it's a follow-up of another paper published at a conference where the candidate is also one of the authors [20]. Our analysis focuses on emotion classification from the mouth only, to study how much the mouth is expressive in emotion expression and can obtain accuracy compared to a full-face recognition. We analysed the position and curve of lips through CNNs. Recently, some of our preliminary works [21, 22, 20] obtained favourable results regarding emotion analysis using the mouth, and this is our first attempt to recap and complete our full study on mouth-based emotion recognition on extensive data sets. All our previous works used Convolutional Neural Networks to reach their goals on the topic, using a self-collected data set. In-deep work has been primarily made on three emotions (i.e., *Joy*, *Disgust*, *Neutral*) [20], where *Disgust* is the less studied in the literature.

Section 1.5 describes one of the most important pieces of work that was carried out during this three-year period. It demonstrates how virtual reality can be used to generate synthetic datasets with which to train artificial neural networks and thus speed up all those steps that researchers generally have to go through for the preliminary study phase of a machine learning problem. The research path being presented has been published in a journal, and the candidate is one of the authors[23]. This paper will also report on use cases, the first is aimed at generating images through a technique that can be seen as an evolution of data augmentation[24], and the second instead generate synthetic images by randomly arranging objects along the scenario and randomly varying the lighting and shading conditions of the scene. This section is also the point of connection with the second chapter of this paper. To train the neural networks correctly, it is necessary to have available datasets of examples that the network can use to learn and understand how to solve the problem that has been assigned to it. The datasets for the training of the neural networks are generally very large and require considerable effort to be constructed correctly. Let us think, for example, of the convolutional neural networks: these are used today to extract the features that compose the images and classify them according to the labels that the programmer has predefined. As an example, let us imagine a dataset for binary classification of animals, such as dogs or cats. Unless we consider a dataset that is already available on the web, it will be necessary to create one specific to the problem to be addressed. Such a dataset would be very complex and expensive to create in the real world and is a general case of representing three-dimensional environments with completely random lighting conditions and object arrangements, so it may be appropriate and advantageous to create it virtually, thanks to the enormous developments that have taken place in 3D modelling software. This solution allows us to recreate virtual scenarios, generating a high number of images with specific techniques of scene illumination and a random arrangement of objects to enrich the amount of information to be fed to the neural network for its training.

Section 1.6 presents the conclusions of the research conducted in these areas.

1.2 Predictive Models for Accelerating GPU Convolution Kernels

With the variety of parallel architectures available on the market ranging from traditional parallel processors to accelerators (e.g., GPUs) and system on chips (SoCs), several standards have been established to enable portability for heterogeneous architectures (i.e., OpenCL [25] and OpenACC [26]). However, developing generic and performance-portable code has become extremely challenging, especially from an algorithmic point of view. Here, auto-tuning techniques have partially mitigated the performance portability problem by adapting, for example, the underlying memory hierarchies and loop unrolling to a specific architecture. Within this context, a plethora of hardware-oblivious solutions have been developed [27, 28, 29, 30, 31]. Auto-tuners seek the best configuration by assuming a specific instance as input. Thus, they can hardly achieve the best performance across all possible inputs. For example, GEMM routines are often *per-default* tuned for squared matrices [27, 30]. Furthermore, GEMM implementations that are usually tuned for scaling over large matrices drops in performance on small matrices [27].

Recently, auto-code generation techniques made it possible to write high-performance code on specific architecture automatically [32, 33, 34, 9, 35]. Several aspects might limit such approaches. First, they often require that the micro-architecture is exposed. Second, they generate highly-optimised code for a specific algorithm, so that in the presence of multiple algorithms the problem of selecting the best solution still remains. Finally, architectural changes require to re-encode the problem [34].

In the last years, several studies focused on the use of machine learning techniques to model performance [36, 37, 9] or to prune the search space of auto-tuners [38]. For example, hand-written decision rules can easily be replaced by learn-based methods like Decision Tree [39]. Although a lot of studies emphasise the performance achieved by such methods as black-box models, several questions are still open. Specifically, in light of the success of deep learning, many aspects are worth to be investigated. From a methodology perspective, there is no consensus in the community regarding how to model the problem by using supervised methods (e.g., classification and regression are two valid options) or other machine learning approaches (e.g., semi-supervised techniques). For example, the criteria for generating high-quality dataset have not been investigated yet. Such aspect is fundamental for SoC where the generation of large dataset for the training of the model may be expensive. Another aspect concerns the assessment of the model. In literature, well-established measures like accuracy usually reflect the quality of a model. However, the relation between accuracy and practical performance has yet to be investigated in depth. Finally, from an application perspective, sophisticated deep learning compilers already adopt machine learning techniques for generating optimised code [40, 33], however, they assume that the same implementation of convolution is used for all the convolution layers. Such an assumption does not guarantee the best performance across all the existing networks, or for those generated by AutoML.

Motivated by these limitations, this paper provides a deep analysis of predictive models to design efficient adaptive and input-aware libraries. The paper makes two main contributions:

- the study of several supervised classification techniques shows that models learned by using Gradient Tree Boosting and Multi Layer Perceptron outperform other models in different settings. Our evaluation considers various aspect like accuracy-based metrics, learning rate, generalisation ability, performance metrics in relation to dataset generation strategies;
- the implementation of learning-based heuristics to accelerate three libraries and two popular routines, convolution and GEMM, over three different GPU architectures, two high-end Nvidia GPUs (Pascal and Volta) and an embedded low-power ARM GPUs.

Specifically, we accelerate the convolution operator on the ARM Compute Library. Unlike well-established deep learning compilers, which generate optimised code for a specific convolution method (e.g. direct convolution), our framework is able to use a different convolution method for each layer at runtime. Secondly, by applying the same methodology on Nvidia cuBLAS, our model-based heuristics is 2 times better than Nvidia heuristics in selecting the best GEMM implementation by achieving an improvement of up to 2.2x over random sizes and up to 2.8x on matrices collected by popular CNNs. Finally, instead of using predictive models to infer parameters from the search space of traditional auto-tuners, we integrate them to select the best algorithm and related parameters when the input changes. We implement this solution by integrating the CLBlast library and several models that have been trained from the information provided by auto-tuners. We show that the model-based version of CLBlast outperforms the traditional auto-tuned version with up to 3x and 2.5x speed-ups on a high-end Nvidia GPU architecture and on an embedded ARM Mali GPU.

Performance modelling focuses on the maximisation of the objective function over the input domain. One formulation consists of defining the maximisation problem as a classification task or rather selecting the best classes among multiple possibilities such that the objective function is maximised. Another formulation would consist of predicting the performance of each implementation and selecting the best one (ordinal rankings) or alternatively selecting the best value of the tunable parameters. Such an approach requires the use of regression techniques as well as the need for further features for architecture characterisation (e.g., CPU and memory characteristics like frequency, cache organisation and other aspects of the microarchitecture). While regression seems to be more flexible than classification, the latter shows fewer potential issues within this task. First, classification can handle arbitrarily structured solution spaces A, as it treats each solution as a black box; for example, a certain algorithm may admit more than one implementation, each with a different number of configuration parameters, of which every combination can be considered as a single class. Secondly, as for the prediction error, regression may infer invalid parameters choice, or wrong values if the features selected for modelling the architecture are not relevant. Although we believe that regression is worth investigating, the present work focuses on classification techniques. To this aim, the optimisation problem is defined as a classification task as follows: let a be a solution to a particular problem and A a collection of such solutions. Let $f_a : I \to \mathbb{R}$ be an objective function (e.g. FLOPS), where I is the multidimensional input domain for a. For example, I can represent the set of all triples (M, N, K) which describe matrix dimension in a matrix-matrix multiplication, and A a collection of algorithmic choices for such multiplication. The solution set A may contain algorithms, collections of parameters, or any other implementation description. A classification task is then defined as follows: for each $i \in I$, we identify the class label $\arg\max_{a \in A} f_a(i)$, the best solution for that input according to the objective function; a classifier will then consist of a mapping from input descriptions i to solutions a. Supervised techniques require datasets that pairs training points in I to optimal solutions in A. To build a dataset, entries can be generated by fixing an input i and benchmarking $f_a(i)$ for every solution a on the target architecture. In this process, the size of the solution space (set of possible classes) plays a fundamental role. In the simplest scenario, the solution space includes few algorithm implementations without the need for searching optimal parameters. In this case, the classes are known in advance, and the only challenge is to find enough training points for each class. This will be the case of the convolution use-case with its simple, ternary classification problem. In other cases, the solution space A is composed by several algorithms, each one having its own configuration parameters. In this situation, for each algorithm, every possible combination of parameters can potentially be a class on its own. For example, in the GEMM case in the presence of auto-tuners, the solution space has more than 12.000 elements. A classifier will only be aware of classes that appear at least once as the optimal solution of some instance in the training set. Unfortunately, this means that for complex problems the class space strongly depend on the choice of training points. Furthermore, classes that appear in the test set but not in the training set will be impossible to predict. A related issue is that classes lose all their internal structure and thus are treated as black boxes. Specific policies for dataset generation must be identified to address this issue. In the GEMM use-case, we present and analyze the impact on the performance of different strategies based on both synthetic and real-world input instances. Furthermore, since the dataset generation is the most expensive part, we analyse the *learning curves* by showing the *validation* and *training score* of the model over the training samples. By analysing the learning curves, we understand how much the model benefit from adding more training data and whether it suffers more from a variance error or a bias error. The proposed solution is to train standard, cost-insensitive classifiers and then try to discriminate among them in the model selection phase. To assess the quality of the models, we evaluate them in terms of the ability to adapt properly to new, previously unseen data (generalisation), the model performance (w.r.t. the objective function), and the inference cost (which includes feature extraction and class selection). We determine the ability to model unseen data (a generalisation of the model) by studying the differences between training and test scores. Specifically, if both are low, the model will be underfitting. Overfitted models present a high training score and a low validation score. Otherwise, the model is working well[4]. As for the performance, the choice of an evaluation measure depends on the complexity of the task. For simple classification tasks, where the accuracy of some classifiers is close to 100%, accuracy-like metrics are effective in the prediction of the best model as high accuracy corresponds to high performance. In more complex tasks, where high accuracy is hard to achieve, near-optimal classes in terms of performance are more important regardless of the accuracy score. In these cases, accuracy is less effective as a performance estimator, and studying the impact of misclassification becomes necessary to evaluate the models. We introduce two metrics to assess the quality of the heuristics based on the predictive model. The first one considers the possible best performance of an ideal estimator (oracle) against the predictive model. The second one considers the baseline performance of the original heuristics/baseline implementation instead. We denote such metrics as **SoO** (‘speed-up over the oracle’) and **SoB** (‘speed-up over the baseline’). They measure how far the performance is from the possible best (**SoO**) and how much it improves the baseline (**SoB**). Such metrics can also be used as a reference for generating the dataset to the number of points during the benchmark. Finally, we also considered the strictly related problem of *cost-effectiveness*: the cost of selecting the new routine must be lower than the performance improvement. In other words, we require $f_{\overline{a}}(i) + c(i) < f_a(i)$ where $c(i)$ is the cost to select a new implementation $\overline{a}$ for the input i by using a predictive model. Additionally, we could take into account the cost of extracting more complex features: $f_{\overline{a}}(i) + c(i) + \sum_1^t (c(t))$ where $c(t)$ is the cost for extracting the t-th feature. In our case, this cost is zero as we use

[4] A low training score and a high validation score is usually not possible in practical cases

the parameters of input as features only. Such a cost may become relevant when dealing with sparse matrix multiplication. For example, extracting the sparsity pattern might be costly. This aspect is crucial for predicting the best storage format and algorithm [41, 42] and it still remains an open challenge. In our use-cases, we show examples of how the inference cost may weight on the selection of a specific model.

We select convolution and GEMM as use-cases due to their importance in deep learning and scientific computing. Specifically, we formalize three different classification tasks, one for convolution and two for GEMM. The first one aims to model the performance of convolution in the ARM Compute Library [43]. Regarding GEMM, we focus on two different libraries (cuBLAS [13] and CLBlast [27]), which differ from each other in their design. The convolution and GEMM tasks cover different scenarios. The first one requires a prediction among up to three classes using a greater number of features. The convolution operator is described by eight features that define the shape of the input tensor and the number of convolution kernels; three convolution methods, *conv*, *directconv*, and *winogradconv* represents the three admissible classes. The GEMM tasks presents the opposite situation and provides two larger, structured class spaces as reported in Table 1.1. GEMM operations are described by the three dimensions (M, N, K) that represent the matrices being multiplied. The number of classes depends on the BLAS implementation. In the case of Nvidia cuBLAS, the number of possible algorithms is fixed and represents the number of distinct GEMM implementations. Other BLAS libraries may expose a larger number of possible classes which may depend on either implementation or architecture. In CLBlast, for example, the number of possible classes depends on their configuration parameters that, in turn, depend on the architecture. Potentially, the number of classes in a dataset may correspond to the size of the search space of the tunable parameters of both implementations (*xgemm* and *xgemm-direct*). However, in practice, the number of distinct classes is much small. In the present work, the problem of extracting more complex features is not addressed; even though it can improve the classification process by improving the accuracy of the model (especially for the GEMM task), it can introduce significant overhead in the inference.

For the training phase we use the 10-fold cross-validation with balanced accuracy scoring to tune their hyper-parameters to avoid biasing and overfitting. It is important to prevent the selection of invalid classes if they are inferred by the predictive model. In order to solve this issue we defined a policy for the trained models which uses the standard convolution method *conv* whenever an inapplicable class is inferred.

Task/Application	Problem	Features	Classes
Convolution	Algorithm selection	8	2 - 3
cuBLAS GEMM	Algorithm selection	3	24
CLBlast GEMM	Algorithm and parameters selection	3	28 - 82

Table 1.1: Description of the classification tasks. The number of classes for CLBlast GEMM depends on the benchmark and dataset strategy generation on a specific GPU architectures.

We carry out an exhaustive experimental analysis to investigate three main aspects. First, the quality of the models learned from different machine learning techniques in terms of generalisation, accuracy, performance and inference cost. Second, the impact of misclassification on performance. Third, the performance on real use-cases. To this aim, we use three different GPU architectures: a Nvidia Tesla P100, an NIVIDA Titan V, an embedded ARM Mali-T860 based on the Midgard architecture.We used the Collective Knowledge framework [44] for collecting and automating the benchmarks. We compared the performance of several models against the oracle, which always selects the fastest convolution method. We also report the performance against the fixed, manually-selected method *conv*. We tested the learned models on several popular CNNs by collecting 238 convolutional layers from Alexnet, FCN-16s, ResNet-50, VGG-16s, GoogLeNet and InceptionV3. We named such data collection *aggregate* test-set. All the models converge by providing both good generalisation and good balanced accuracy with the exception of KNN which suffers from overfitting and SVM which would require more training points to converge. Decision tree-based estimators showed superior performance overall. One aspect to stress concerns the number of training points required to achieve satisfactory accuracy. For example, RF required just around 500 points to get a very good test accuracy. Such information is particularly useful for the dataset generation on SoCs. Table 1.2 shows several classification metrics for the models, determined through cross-validation on the the training dataset.

SVM also showed a good performance improvement against the fixed methods, using only 4% more time than the oracle, against the +11.5% of the fixed *conv* method. **LoR**, **NBC** and **polynomial-kernel SVM** (not shown) achieved poor performance and accuracy, showing that in the feature space no simple cut exists that adequately separates the classes. **KNN**, although it obtained better performance than static methods, achieved a low *balanced accuracy* score due to its inability to identify *winogradconv* instances. Thus, such a model is not suitable to tackle tasks where this class is more populated. The RF model outperforms the others by correctly recognizing the *directconv*

	accuracy	balanced accuracy	conv		directconv		winogradconv	
			precision	recall	precision	recall	precision	recall
RF	0.93 ± 0.005	0.94 ± 0.005	0.98 ± 0.005	0.93 ± 0.01	0.85 ± 0.02	0.92 ± 0.01	0.83 ± 0.03	0.99 ± 0.01
MLP	0.89 ± 0.01	0.85 ± 0.05	0.92 ± 0.02	0.93 ± 0.01	0.82 ± 0.03	0.80 ± 0.05	0.82 ± 0.085	0.82 ± 0.1
DT	0.91 ± 0.01	0.93 ± 0.01	0.98 ± 0.01	0.92 ± 0.01	0.83 ± 0.02	0.89 ± 0.025	0.81 ± 0.07	0.98 ± 0.015
LoR	0.57 ± 0.01	0.70 ± 0.01	0.90 ± 0.015	0.50 ± 0.015	0.53 ± 0.025	0.64 ± 0.035	0.22 ± 0.015	0.98 ± 0.02
KNN	0.72 ± 0.2	0.50 ± 0.01	0.80 ± 0.01	0.84 ± 0.03	$0,65 \pm 0.045$	0.62 ± 0.025	0.05 ± 0.025	0.05 ± 0.025
NBC	0.71 ± 0.15	0.74 ± 0.01	0.85 ± 0.015	0.80 ± 0.015	0.59 ± 0.02	0.41 ± 0.025	0.37 ± 0.03	0.99 ± 0.05
SVM	0.72 ± 0.15	0.69 ± 0.02	0.90 ± 0.01	0.72 ± 0.02	0.64 ± 0.03	0.76 ± 0.02	0.27 ± 0.025	0.59 ± 0.055
GTB	0.94 ± 0.05	0.94 ± 0.01	0.96 ± 0.005	0.97 ± 0.005	0.90 ± 0.02	0.89 ± 0.01	0.93 ± 0.03	0.96 ± 0.025

Table 1.2: Classification metrics for different models, from 10-fold cross-validation on the training dataset.

CNN	SPEED-UP AGAINST 'CONV'	SLOW-DOWN AGAINST ORACLE	BEST MODEL
Alexnet	1.19x	+0.00%	RF
FCN-16s	1.55x	+5.77%	RF
GoogLeNet	1.01x	+0.05%	MLP
InceptionV3	1.25x	+0.00%	DT
Mobilenet	1.00x	+0.00%	DT
NiN	1.00x	+0.00%	DT
ResNet-50	1.04x	+0.00%	DT
VGG-16	1.44x	+0.00%	RF

Table 1.3: Speed-up over the fixed 'conv' method and slow down over the Oracle for the best model in each CNN. Ties between models are broken by inference time.

instances. The motivation is behind the learning curve: RF performs better than others in data generalisation. This demonstrates that in more balanced test sets the model-driven approach can achieve even higher speed-ups. This is also observed by analysing the results on other CNNs, as reported in Table 1.3. Since the relative proportion of classes varies considerably among CNNs, the *balanced accuracy* score reported in Table 1.2 is a better performance indicator than simple accuracy.

We compared the inference time of the models w.r.t DTs, which have the lowest inference cost overall. All the models show up to a 7.5x (KNN) of overhead with the exception of RF, for which the overhead cost depends on the number of trees. They provided the best performance (balanced accuracy, generalisation, SoO), when using around 300 trees, which makes their inference cost significant (261x). This cost can be reduced by cutting the number of trees, eventually down to the still good performance of a single DT. This allows for an easily adjustable trade-off between accuracy on hand, and training and inference costs on the other. The optimal trade-off between

FEATURE	K	C-IN	C-OUT	W + H	NP	P	S
IMPORTANCE	37%	20%	17%	13%	11%	2%	0%

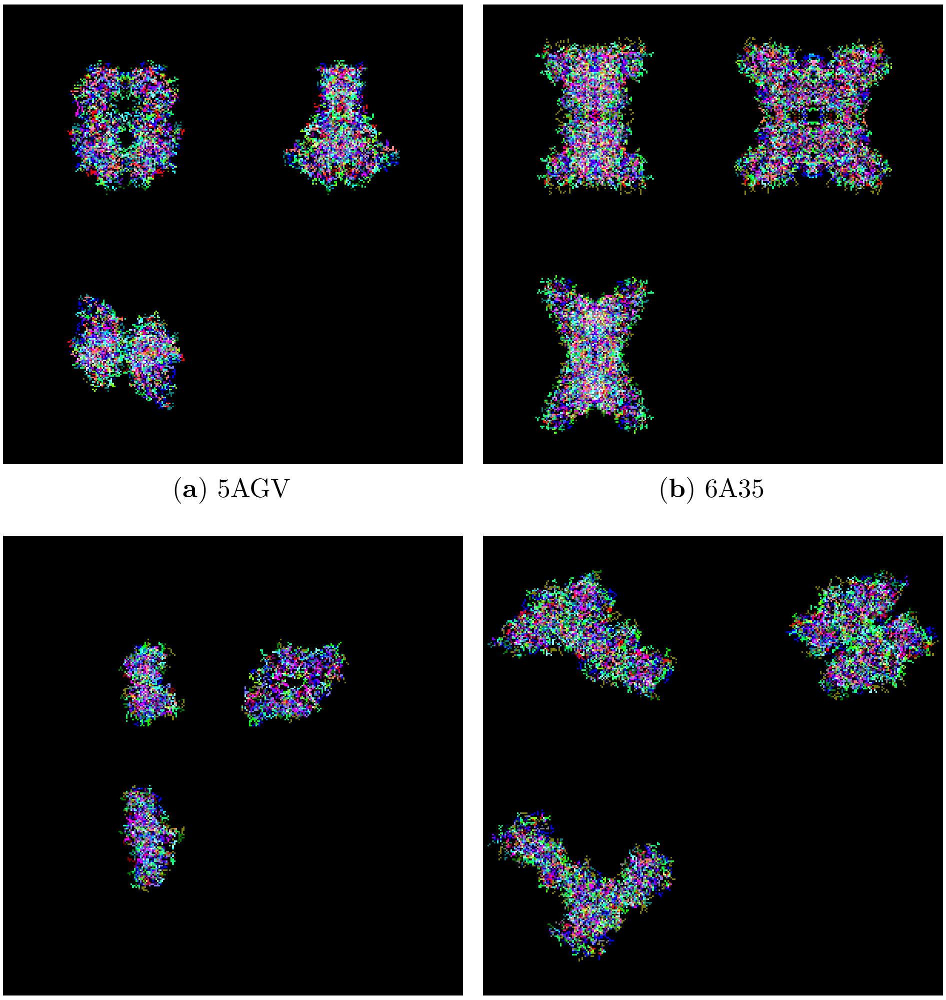
Table 1.4: Relative importance of features in the convolution classification task, extracted from a trained Random Forest.

precision and inference overhead depends on the circumstances. Our study on inference time for GEMM revealed that the overhead of an efficiently implemented DT amounts to less than 1% of the computation time of the average GEMM multiplication. Considering training cost, inference cost and performance, DTs, RF and GTB are the best models for this task. Moreover, these models allow for an easy-to-interpret feature importance analysis. In a DT, the *importance* of a feature is the normalised total reduction of the splitting criterion (in our case, the entropy) due to that feature. Ranking feature by their average *importance* in the RF model revealed that kernel size has the greatest impact on classification, followed by the number of input and output channels, as summarised in Table 1.4. Such knowledge is relevant when generating or expanding a training dataset, as it can guide the choice and density of training points throughout the feature space.

1.3 Machine learning applied to chemistry

In this Section, we illustrate our approach to the problem using a two-dimensional representation based on 2D Convolutional Neural Networks, where any given protein is mapped into a two-dimensional grid of coloured pixels and then processed by the CNN in order to extract the relevant features and the characteristic properties to carry out the protein classification. In order to train the neural network, similarly to the previous work, the set of protein structures was obtained from the Protein Data Bank (PDB)[45] an open access repository containing data about proteins and nucleic acids' structures. The system is a classic binary classifier whose task is to correctly subdivide the biomolecules given in input as "protein" or "non-protein". The data relating to the molecules to be examined are passed to the neural network as two-dimensional image processing, which faithfully reconstructs the geometric structure of the molecule itself. In order to validate our methodology, we have extracted from the PDB a sufficient number of records useful to effectively train the network; after several tests, it was found that good results were obtained with a number of samples around 3,000 units. This allowed us to select a group of proteins (equal to 2,911 molecules, with 16,924,350 amino acids) with the best images of their structure and focus our attention on the results. The whole process, from data extraction to image evaluation by the neural network, was performed in Python3, with the help of well-known libraries useful for scientific computing and data processing, such as Numpy and Pandas.

The various steps required for the generation and management of the dataset are listed below:



(**a**) 5AGV

(**b**) 6A35

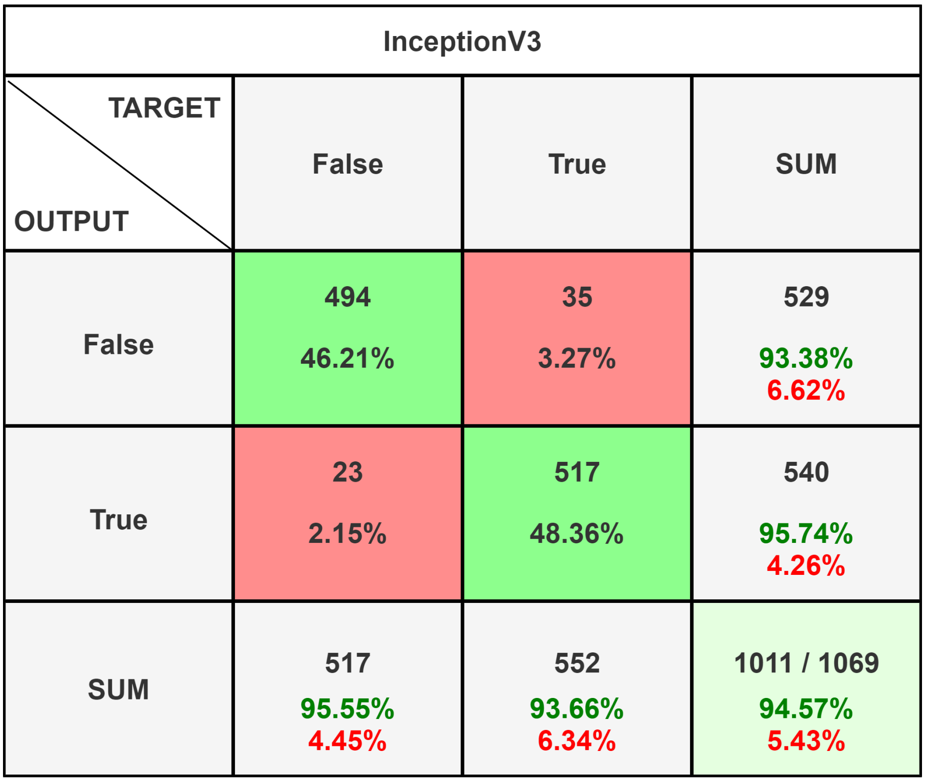
(**c**) 6A51

(**d**) 6AF4

Figure 1.1: (**a**) 5AGV, the sliding clamp of Mycobacterium tuberculosis in complex with a natural product, (**b**) 6A35, crystal structure of 5-methylthioribose 1-phosphate isomerase from Pyrococcus horikoshii OT3 - Form II, (**c**) 6A51, novel Regulators CheP and CheQ Specifically Control Chemotaxis Core Gene cheVAW Transcription in Bacterial Pathogen Campylobacter jejuni, (**d**) 6AF4, toxin-Antitoxin module from Streptococcus pneumoniae.

1) Data extraction in XML format from the PDB database, identification of the necessary information we wish to keep and its subsequent transformation into a single object managed by the Pandas library
2) Data cleaning, with the elimination of any duplicate records, possibly

generated by the two different methods of measuring the crystal lattice structure for the molecule
3) Association of a unique RGB color code to each amino acid present in proteins (e.g. Alanine 128, Glycine 65280, Lysine 8421376), in order to visualize the structure of the molecule as an image, on which every single amino acid is coloured in a different way

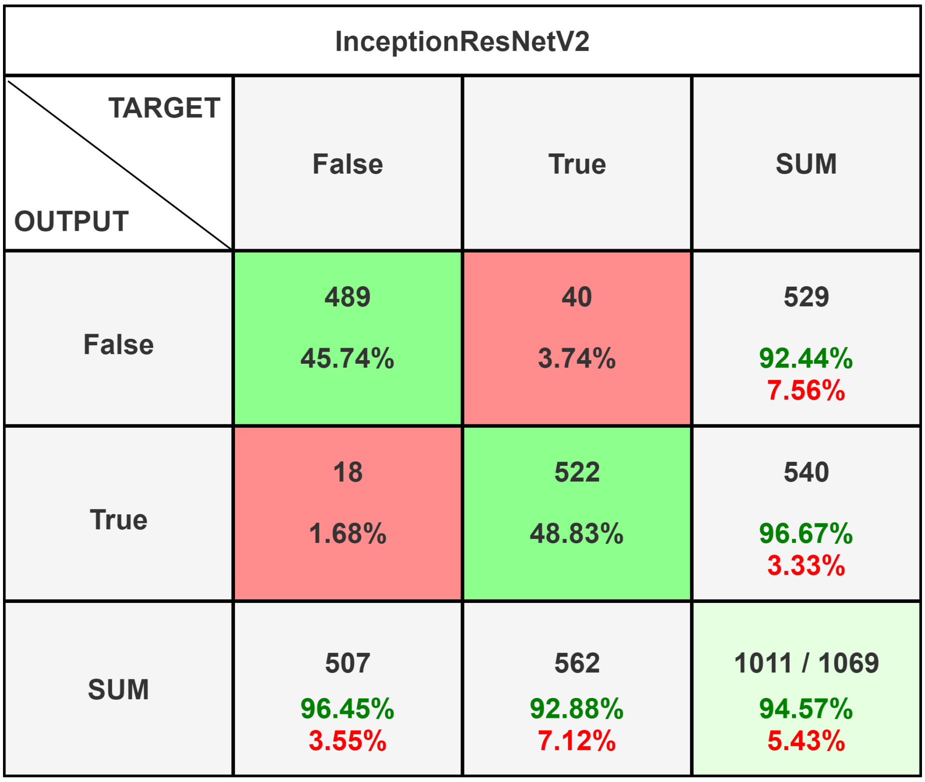


InceptionV3

TARGET OUTPUT	False	True	SUM
False	494 46.21%	35 3.27%	529 93.38% 6.62%
True	23 2.15%	517 48.36%	540 95.74% 4.26%
SUM	517 95.55% 4.45%	552 93.66% 6.34%	1011 / 1069 94.57% 5.43%

Figure 1.2: Confusion matrix on the Validation Set with the network Inception V3.

- Visualization of the molecule according to a multiview orthographic axonometric map with orthogonal projections on three planes (horizontal plane, vertical plane and lateral/profile plane, in order to split and project the whole 3D image on a flat surface, without losing the isometricity and symmetries on the x, y and z components for the individual amino acids. From the analysis of the coordinates of all the amino acids present in the dataset, it was possible through an appropriate translation for the origin of the Cartesian reference system and an integer mapping for the numerical values of the coordinates themselves to represent each single protein in the domain $D = [0, 3200]^3 \subset \mathbb{N}^3$;



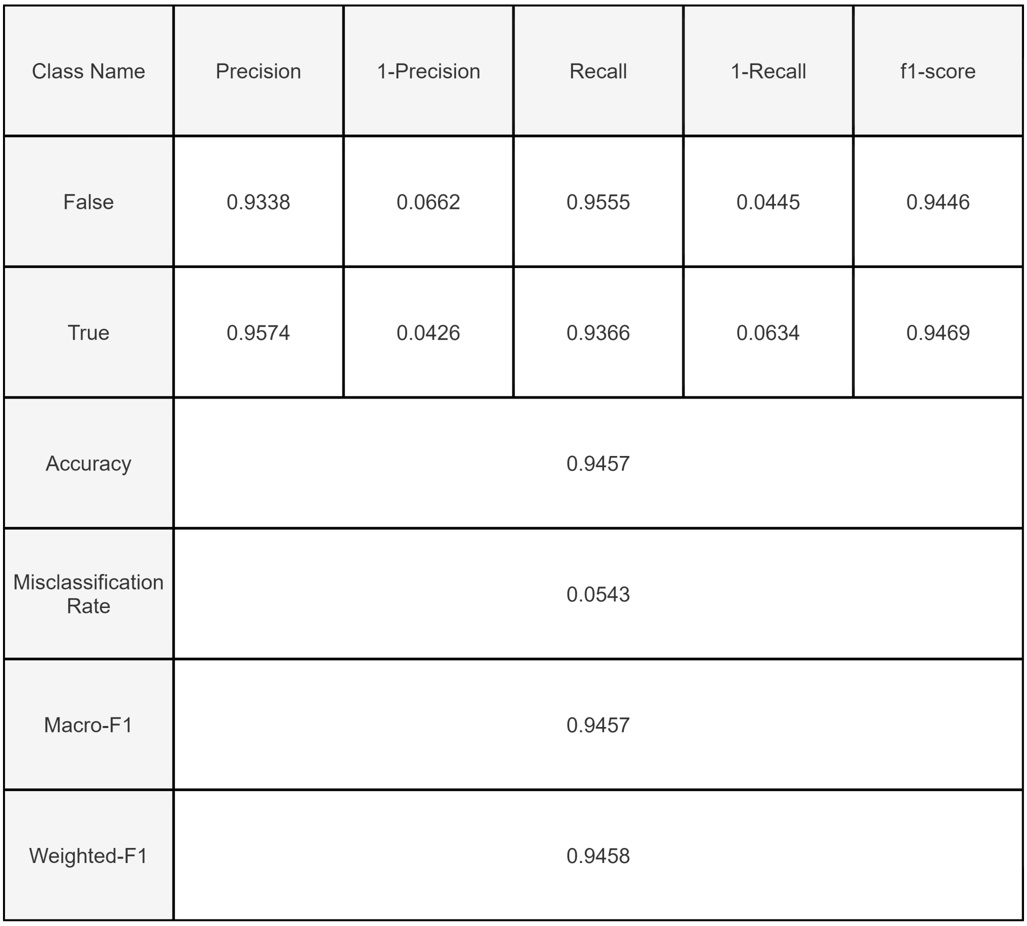
InceptionResNetV2			
TARGET OUTPUT	False	True	SUM
False	489 45.74%	40 3.74%	529 92.44% 7.56%
True	18 1.68%	522 48.83%	540 96.67% 3.33%
SUM	507 96.45% 3.55%	562 92.88% 7.12%	1011 / 1069 94.57% 5.43%

Figure 1.3: Confusion matrix on the Validation Set with the network InceptionRes-Net-V2.

this allowed us to refer to each point (x, y, z) belonging to the cube D as a 3-indices tuple for the tensor with dimensions 3200x3200x3200, capable of containing the entire biomolecule

- Each image has been processed to fit within the 299x299 pixel dimensions, necessary as input dimensions for a 2D convolutional neural network; in order to avoid distorting the original axonometric proportions, the figures have been carefully cut out at the edges and in the central areas, to reduce unnecessary black padding.

Therefore, for each single protein, the various amino acids were projected, as colored dots, on the three main planes: in Figure 1.1 four images obtained with our method for four different proteins are shown.
4) Generation of false samples, necessary for the learning of the neural network. It was decided to proceed starting from the original images; for the single amino acids belonging to each protein we applied a mutation probability, established at the beginning of the process (in our case experimental



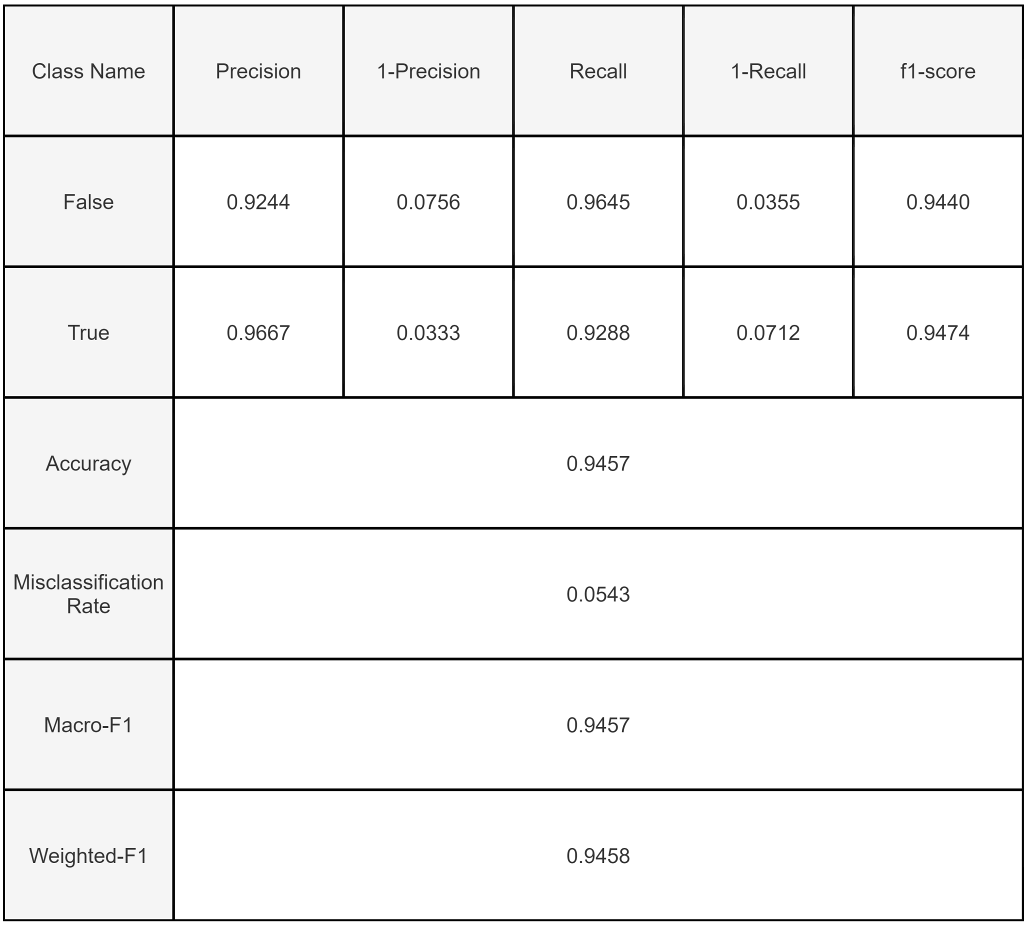
Class Name	Precision	1-Precision	Recall	1-Recall	f1-score
False	0.9338	0.0662	0.9555	0.0445	0.9446
True	0.9574	0.0426	0.9366	0.0634	0.9469
Accuracy	0.9457				
Misclassification Rate	0.0543				
Macro-F1	0.9457				
Weighted-F1	0.9458				

Figure 1.4: Statistics and performance of the model generated using InceptionV3.

tests made us lean towards a fixed value of 5%), which induced a colour change on the coloured points representing the amino acid: 2911 images of false proteins were thus produced. The performances of two neural networks were analyzed: InceptionV3 and InceptionResNetV2.

These networks were trained through the **transfer learning** technique which allowed us to obtain high accuracy values with a reduced number of learning epochs.

Transfer learning is the process by which a neural network learn a new task through knowledge's transfer to a related task previously learned. Using transfer learning, the weight values of the convolutional layers are imported (they are generally used for feature extraction and already present as initial parameters of the network itself), and only the final layers of the neural network that are generally used for classification are trained[19, 18]. Both



Class Name	Precision	1-Precision	Recall	1-Recall	f1-score
False	0.9244	0.0756	0.9645	0.0355	0.9440
True	0.9667	0.0333	0.9288	0.0712	0.9474
Accuracy	0.9457				
Misclassification Rate	0.0543				
Macro-F1	0.9457				
Weighted-F1	0.9458				

Figure 1.5: Statistics and performance of the model generated using InceptionResNet-V2

networks are pre-trained on ImageNet, an image dataset used for object recognition consisting of 14 million photographs.

The results of the training are summarised in the Table 1.5, where it can be seen that the InceptionV3 network has a higher percentage of accuracy in recognising false proteins from true ones compared to the results obtained by InceptionResNetV2.

The Confusion Matrices[46] of the two models indicate that they are both able to classify proteins with good results, also we believe that a correct parameters' modulation, through the realisation of ad-hoc models, can further increase the degree of accuracy. The matrices are shown in Figure 1.2 and Figure 1.3. The statistics related to the output of the training phase are shown in Figure 1.4 and 1.5.

	InceptionV3	InceptionResNet-V2
Accurac Training Set	99.06%	98.36%
Accuracy Validation Set	94.57%	92.14%
Model Size	461MB	465MB

Table 1.5: Final Results of the training phase

1.4 Analysis and extraction of information from images

1.4.1 Skin Cancer Classification

Due to these limitations, some previous research activities focused on melanocytic lesions (in order to differentiate between a benign and malignant sample) and disregarded non-melanocytic pigmented lesions, even if they are very common. In order to boost research on the automated diagnosis of dermatoscopic images, HAM10000 has been providing participants of the ISIC 2018 classification challenge, hosted by the annual MICCAI conference in Granada, Spain [47], specific images.

The set of 10015 8-bit RGB colour images were collected in 20 years from populations from two different sites, specifically the Department of Dermatology of the Medical University of Vienna and the skin cancer practice of Cliff Rosendahl in Queensland. Relevant cases include a representative collection of all important diagnostic categories of pigmented lesions[47]:

- **akiec:** Actinic Keratoses and Intraepithelial Carcinoma, common noninvasive variants of squamous cell carcinoma that can be treated locally without surgery. [**327 images**]

- **bcc:** Basal cell carcinoma, a cancer that rarely metastasizes but grows destructively if untreated. [**514 images**]

- **bkl:** Generic label that includes seborrheic keratoses, solar lentigo and lichen-planus like keratoses (LPLK), which corresponds to a seborrheic keratosis or a solar lentigo with inflammation and regression, often mistaken for melanoma. [**1099 images**]

- **df:** Dermatofibroma, a benign skin lesion. [**115 images**]

- **nv:** Melanocytic nevi are benign neoplasms of melanocytes. [**6705 images**]

- **mel:** Melanoma, if diagnosed in an early stage, it can be cured by simple surgical excision. [**1113 images**]

- **vasc:** Vascular skin lesions in the dataset range from cherry angiomas to angiokeratomas and pyogenic granulomas. [**142 images**]

More than 50% of lesions are confirmed through histopathology (histo), the ground truth for the rest of the cases is either follow-up examination (followup), expert consensus (consensus), or confirmation by in-vivo confocal microscopy (confocal). Other features in the individual dataset include age, gender and body-site of lesion (localization)[47].

Since their first appearance in Le Cun *et al.* publication [48], Convolutional Neural Networks (CNN) have been widely applied to data that have a known, grid-like structure. Possible set of interests are time-series data, which can be modeled as a 1D grid taking samples at regular time intervals, and image data, which can be thought of as a 2D grid of pixels. The foundational layer of a convolutional network consists of three stages. In the first stage, the layer performs several parallel convolutions to produce a set of linear activations. In the second stage, each convolution output is run through a nonlinear activation function, such as the rectified linear activation function (ReLU). In the third stage, a pooling function is used to modify the output of the layer further [49]. The max pooling operation, used in this work, reports the maximum output within a rectangular neighborhood. Since the objective images of skin lesions present great variation in size, we decided to use a network made by inception modules, which make use of filters of different size operating at the same level. This usage of wide modules with multiple cheap convolutional operations entails a reduced computational complexity with respect to a deep network with large convolutional layers. In the specific model we used, based on Inception-ResNet-v2, another point of speed improvement is the introduction of residual connections, which replace pooling operations within the main inception modules. However, the previously mentioned max pooling operations are still present in the reduction blocks.

The original Inception-ResNet-v2 architecture [50] has a stem block consisting of the concatenation of multiple convolutional and pooling layers, while Inception-ResNet blocks (A, B and C) contain a set of convolutional filters with an average pooling layer. As prevously mentioned, reduction blocks (A, B) replace the average pooling operation with a max pooling one. This structure has been extended with a final module consisting of a flattening step, two fully-connected layers of 64 units each, and the softmax classifier. The overall module is trainable on a single GPU with reduced memory consumption. This work consists of two training rounds, after a step of data processing in order to deal with the strong imbalance of the dataset:

- A first classification training process using class weights.
- Rollback of previously obtained best model to improve classification performance with a second training phase.

In the first stage of data processing, after the creation of a new column with a more readable definition of labels, each class was translated into a numerical code using *pandas.Categorical.codes*. Afterwards, missing values in "age" column was filled with column mean value.

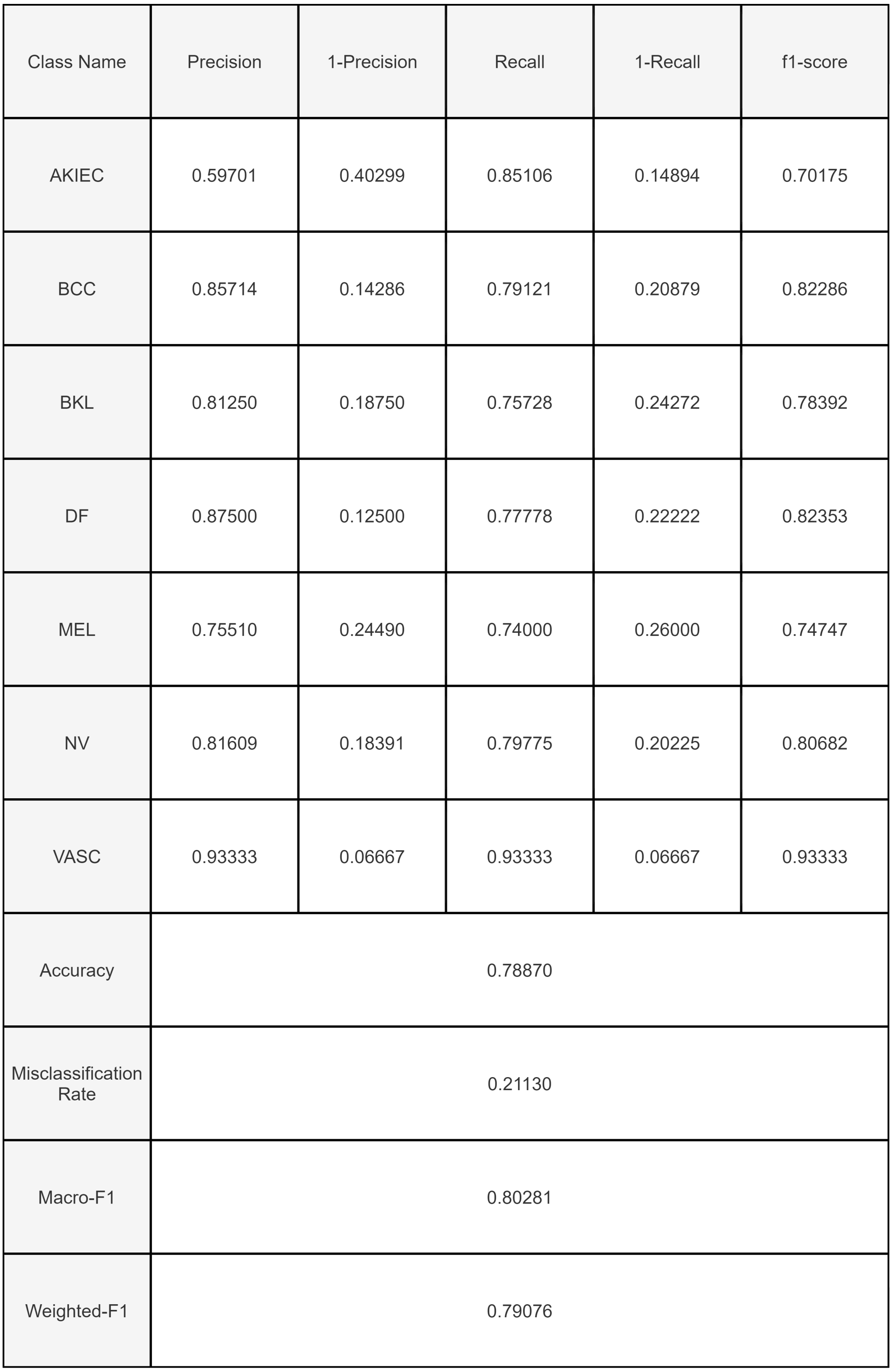
Finally, images are loaded and resized from 450×600 to 299×299 in order to be correctly processed by the network. After a normalisation step on RGB arrays, we split the dataset into a training and validation set with 80:20 ratio.

In order to re-balance the dataset, we chose to shrink the amount of images for each class to an equal maximum dimension of 450 samples. This significant decrease of available images is then mitigated by applying a step of data augmentation. Training set expansion is made by altering images with small transformations to reproduce some variations, such as horizontal flips, vertical flips, translations, rotations and shearing transformations. Due to the limited number of samples for the training process, we decided to take advantage of transfer learning, utilising Inception-ResNet-v2 pre-trained on ImageNet[51] and Tensorflow, a deep learning framework developed by Google, for fine-tuning of the last 40 layers. Keras library offers a wide range of optimises: Adaptive optimisation methods such as AdaGrad, RMSProp, and Adam are widely used for deep neural networks training due to their fast convergence times. However, as described in [52], when the number of parameters exceeds the number of data points these optimises often determine a worse generalisation capability compared with non-adaptive methods. In this work we used a stochastic gradient descent optimiser (SGD), with learning rate set to 0.0006 and usage of momentum and Nesterov Accelerated Gradient in order to adapt updates to the slope of the loss function (categorical crossentropy) and speed up the training process. The total number of epochs was set to 100, using a small batch size of 10. A set of class weight was introduced in the training process to get more emphasis on minority class recognition. A maximum patience of 15 epochs was set to the early stopping callback in order to avoid the problem of the overfitting. Finally, the model achieves an accuracy of 73.4% on the validation set, using weights from the best epoch. Dermatofibroma (df), are not properly recognised. Melanoma (mel) is often mistaken for generic keratoses (bkl). In order to improve classification performance, especially in minority classes, we loaded the best model obtained in the first round to extend the training phase and explore other potential local minimum points of the loss function, by using an additional amount of 20 epochs. This second step led to an enhancement in overall predictions, reaching the maximum accuracy value of 78.9%.

Figure 1.6 shows the normalised confusion matrix on the validation set for the final fine-tuned model. Figure 1.7 shows the stats of the model. In this case, 6 out of 7 categories are classified with a total ratio of True Positives higher than 75%, even in presence of an extremely limited sample set, as vascular lesions (vasc), 30 samples, and dermatofibroma (df), 16 samples. The whole process of training has required less than four hours on Google Colab cloud's GPU, for an overall RAM utilization below 20 GB. In conclusion, in this paper we investigate the possibility of obtaining improved performances in the classification of 7 significantly unbalanced different types of skin diseases, with a small amount of available images. With use of a fine-tuned deep inception network, data augmentation and class weights, the model can achieve a good final diagnostic accuracy. The described training process has a light resource usage, requiring less than 20 GB of RAM space, and it can be executed in a Google Colab notebook. For future improvements, larger datasets of dermatoscopic images are needed. The model shown in this paper can be regarded as a starting point to implement a lightweight diagnostic support system for dermatologists, for example in the Web as well as through a mobile application.

Validation Set								
TARGET OUTPUT	AKIEC	BCC	BKL	DF	MEL	NV	VASC	SUM
AKIEC	40 8.37%	10 2.09%	6 1.26%	2 0.42%	8 1.67%	1 0.21%	0 0.00%	67 59.70% 40.30%
BCC	4 0.84%	72 15.06%	5 1.05%	0 0.00%	3 0.63%	0 0.00%	0 0.00%	84 85.71% 14.29%
BKL	1 0.21%	5 1.05%	78 16.32%	2 0.42%	6 1.26%	4 0.84%	0 0.00%	96 81.25% 18.75%
DF	0 0.00%	0 0.00%	1 0.21%	14 2.93%	0 0.00%	1 0.21%	0 0.00%	16 87.50% 12.50%
MEL	1 0.21%	2 0.42%	8 1.67%	0 0.00%	74 15.48%	11 2.30%	2 0.42%	98 75.51% 24.49%
NV	1 0.21%	2 0.42%	5 1.05%	0 0.00%	8 1.67%	71 14.85%	0 0.00%	87 81.61% 18.39%
VASC	0 0.00%	0 0.00%	0 0.00%	0 0.00%	1 0.21%	1 0.21%	28 5.86%	30 93.33% 6.67%
SUM	47 85.11% 14.89%	91 79.12% 20.88%	103 75.73% 24.27%	18 77.78% 22.22%	100 74.00% 26.00%	89 79.78% 20.22%	30 93.33% 6.67%	377 / 478 78.87% 21.13%

Figure 1.6: Confusion matrix of the validation set.



Class Name	Precision	1-Precision	Recall	1-Recall	f1-score
AKIEC	0.59701	0.40299	0.85106	0.14894	0.70175
BCC	0.85714	0.14286	0.79121	0.20879	0.82286
BKL	0.81250	0.18750	0.75728	0.24272	0.78392
DF	0.87500	0.12500	0.77778	0.22222	0.82353
MEL	0.75510	0.24490	0.74000	0.26000	0.74747
NV	0.81609	0.18391	0.79775	0.20225	0.80682
VASC	0.93333	0.06667	0.93333	0.06667	0.93333
Accuracy	0.78870				
Misclassification Rate	0.21130				
Macro-F1	0.80281				
Weighted-F1	0.79076				

Figure 1.7: Statistics and performance of the model on the dataset.

1.4.2 Enhancing mouth-based emotion recognition

Visual *Emotion Recognition* (ER) has been widely studied, as one of the first *Affective Computing* techniques, based on visual features of the face expression combining features about eyes, mouth and various facial elements at the same time. Several different approaches at visual recognition obtained different grades of classifications for different visual recognition techniques [22, 53, 54]. Recently, studies using only the mouth for facial emotion recognition obtained promising results, still not gaining the proper attention in the state-of-the-art. Such works used *Convolutional Neural Networks* (CNNs) to detect basic emotions from innovative and ubiquitous devices, e.g., smartphone or computer cameras, to produce a textual, audio or visual feedback for humans, or digital output to support other services, mainly for healthcare systems [21].

After proving that the mouth can be itself a promising element for emotion recognition, we focus on the mouth as a unique element for facial-expression-based emotion recognition using deep learning advanced techniques. The goal of this approach is to enhance the mouth-based approach to emotion recognition in order to generalise the previous experimentation on a multiple-users data set. To this aim, advanced deep learning techniques have been implemented, i.e. *Transfer Learning* with elements of *Continuous Learning*. Such techniques are particularly suitable to be applied in the healthcare environment.

For instance, connecting this architecture to appropriate services can help users to convey emotions in an automated way effectively, e.g. providing augmented emotional stimuli to users affected from autism or other conditions involving social relationship abilities, where the user experiments difficulties in recognising emotions expressed by other people. Another example may be a system able to recognise severe conditions and call a human assistant for intervention, e.g., for hospitalised patients feeling intense pain or needing psychological support. Such applications may provide feedback from healthcare personnel to exploit Continuous Learning.

Transfer Learning allowed our neural networks to be pre-trained on low-level features, e.g., edges, corners and colour distribution.

In the last layers of the CNN, we carried out an ad-hoc training dedicated to human face recognition of emotional classes. This work concludes the experiments testing and comparing several CNNs on a widely-used data set [55] of human faces labelled with emotions from the Ekman model. As in the preliminary work, also for the final data set, we provide a filtered version, where photos showing simulated emotions, i.e. non-spontaneous expressions, have been removed.

Problem description and proposed solution

The study exploits the high precision of CNNs processing mouth images to recognise emotional states applying for the most recent advances in affecting computing. The main emotions and facial expressions we want to classify are *Happiness*, *Surprise*, *Anger* and *Neutral*.

Our implementation of mouth-based ER exploits Transfer Learning (i.e., knowledge transfer), which uses general-purpose neural networks pre-trained on extensive data sets including different shapes, later fine-tuned on the classification domain (i.e., mouth-based emotion recognition). Our application can be easily adapted for Continuous Learning, given a domain where such method is useful, and appropriate data are available as feedback, to track the weights of the neural network and prosecute the training. Such a method should allow enhancing the precision of emotion recognition in real-time situations, where the final weights of a previous training can be used in a subsequent time. The collateral effect of such a technique is the requirement of more exceptional computational capabilities and a semi-supervised system in order to stop and rollback in the case of glaring errors or overfitting on a specific environment, e.g., a particular camera resolution or light situation in a specific place or timing. Continuous Learning can continuously adapt and enhance the network accuracy, leading on the go to a more reliable system, when used for a prolonged time.

The Framework Implementation

The proposed implementation has been developed using the *Python* programming language and the *Keras* framework [56]. The tests were carried out using the Google *Colab* platform, in which the Python code was developed and executed, exploiting the powerfulness of the *Keras* libraries. Data analysis and processing have been developed with a chain of operations. The implemented procedure can be described with the following steps: *Raw data set import*, *Raw data set cleaning*, *Data set generation*, *Training with data augmentation*, *Model generation*, *Results Evaluation*

The mouth extraction from the images of the raw data set is carried out using as a pre-trained neural network the shape predictor 68 face landmarks[5] CNN [53, 55, 57], which produces in output 68 landmarks, detected for each image. The shape predictor is pre-trained on the *ibug 300-W* data set. The landmarks are expressed as a series of coordinates identifying specific

[5] https://github.com/davisking/dlib-models

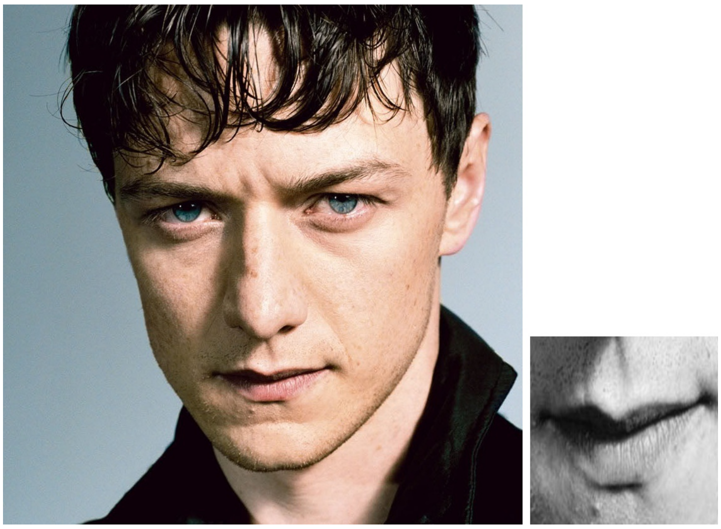


Figure 1.8: Mouth detection, cropping and resizing

elements of a face, e.g., the position of mouth, eyes, cheekbones. Once we obtained the landmarks of a face, we use those to identify the area of the mouth, cropping the image (i.e., cutting out the original image to obtain only the part related to the mouth). All images of the mouth are transformed in size of 299x299 pixels.

In Figure 1.8, an example of mouth detection, cropping, and resizing is shown. On the left side, an image taken from the data set is visible; on the right, the corresponding extracted mouth.[6] At the end of the process, the mouth images are stored in a directory structure associated with the labeled emotion; a final inspection of the mouth images data set shows that the mouth is always correctly cut, whether it is open or not. We used data augmentation techniques [58] to increase the robustness of the neural network, randomly transforming the images, i.e., with a flip, and a random rotation within the [-4,+4] degrees range.

CNN settings

The experiments have been performed on four Convolutional Neural Networks: *VGG16* [59], *InceptionResNetV2* [50], *InceptionV3* [60] and *Xception* [61]. Among CNNs, these networks outperform *AlexNet* [62] on the widely used *ImageNet* [63] data set, which is one of the largest image data set used for the Transfer Learning pre-training phase.

All neural networks, have been tested using Transfer Learning (see Figure 1.9 showing the process for a general CNN).

[6] source image taken from AffectNet database

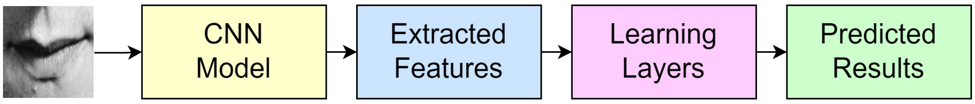


Figure 1.9: General scheme of our CNN

Adopting Transfer Learning, we used the pre-trained neural networks, which weights have been computed on the *ImageNet* [63] data set. The first layers of the networks have been frozen for the training phase, i.e. the weights of the convolutional layers have been fixed and have not been altered during the fine-tuning phase on the mouth emotion data set. This choice is due to the high capability of the CNNs to recognise low-level features, e.g., points, lines, edges, corners, colour distribution. We replaced the final layers to fine-tune the network on mouth emotion recognition, using two dense layers with 64 neurons each and a *SoftMax layer*. This final layer ranks the likelihood of the most appropriate class, thus returning the emotion classification. Only the final layers of the CNN networks, i.e. the *fully-connected* and the final *Softmax* level therefore change and can be re-trained in the fine-tuning phase. The model is set up saving the weights only when they improve the emotion classification accuracy concerning the previous epoch, thus resulting in a final best neural network training configuration.

We tested two optimisers: Adam and SGD. Adam performed better with *InceptionV3* and *VGG16* with a learning rate equal to 0.001, with *Xception* with a learning rate of 0.01. SGD performed better with *InceptionResNetV2* with a learning rate equal to 0.001, `momentum`=0.9 and `nesterov` equal True. The remaining parameters used were the following: batch size equal to 25, the maximum number of epochs equal to 100; however, we used the early stopping technique: if results do not improve, the training is stopped.

Data set and filtering phase

In the actual state of the art, one of the most relevant issues for our goal is the total lack of a sufficient number of images of real emotions. Most of the available data sets, in fact, include a mixed set of real and fake emotions, performed by actors or pretended, which cannot be considered usable for a technique like Deep Learning, able to base the classification on very detailed features, which are different between a real and a fake emotion even at a micro-expression level, and may vary in different cultures. People who are taking a pose for a photograph (e.g., smiling voluntarily in front of the camera) often assume an artificial, distorted expression that is not generated autonomously by the human brain, even if it can be appropriately identified and recognised by the human brain as a smile, thanks to mirror neurons. We must report that most of the available data sets provide images depicting actors, thus a cleaning phase of the chosen data set has been mandatory. It should be considered that there are emotions such as *Anger* and *Fear* that are not easily replicated in a laboratory, while emotions such as *Joy* or the *Neutral* state, on the other hand, can be easily triggered.

Another relevant limitation is that many data sets include only images of a particular light state, perspective, image resolution, which will lead the system overfitting that particular environment and fail on others. Until a proper extensive data set will be ready for the test, any result has to be trivially considered preliminary.

For the emotional training (i.e., fine-tuning of the CNNs) and classification test of our work, the data set *AffectNet* provided by the University of Denver [64], USA, has been used. The raw data set is composed of the images of a large number of faces, and a spreadsheet file used to assign to each image an emotion label. We analyse a subset of the Ekman emotions, i.e., *Neutral*, *Happy*, *Surprise*, or *Anger*. A sample image for each class of emotion is shown in Figure 1.10.

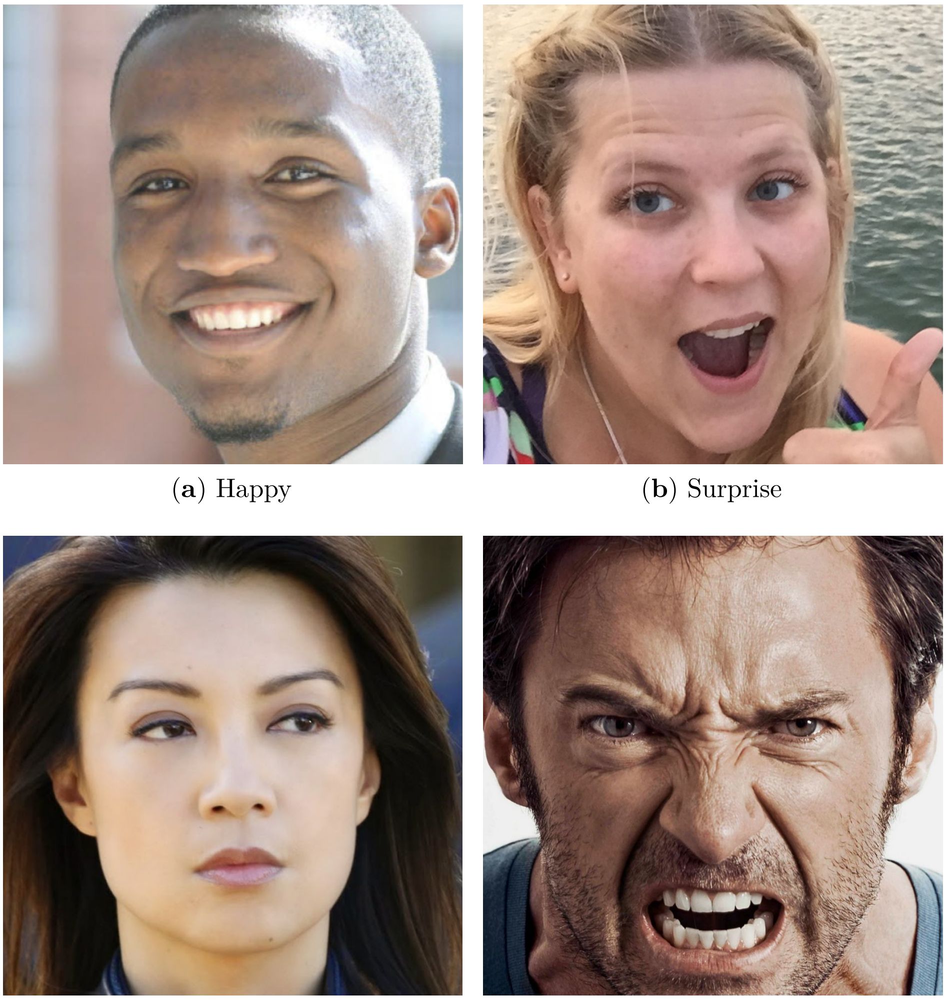
The data set has been cleaned analysing images and removing duplicates. The automated mouth recognition removed also all the items where the mouth was not visible. If grainy or unsatisfactory resolution photographs can be removed, it is also essential to maintain a significant difference in resolution, point of view, and light conditions. This variety helps the network to train on general images and not only on a particular light condition, resolution, and so on.

The primary filtering improvement of the data set has been exploited on the content of images, avoiding all the photographs showing fake emotions, i.e. facial expressions apparently simulated.

In Table 1.6, the number of images per type of emotion in the cleaned data set is shown.

	Neutral	*Happy*	*Surprise*	*Anger*
# of images	1239	562	463	478

Table 1.6: Number of images per type of emotion considered in our study.



(**a**) Happy

(**b**) Surprise

(**c**) Neutral

(**d**) Anger

Figure 1.10: Sample images from the filter data set

Results and discussion

The experimental work has been divided into two phases. A preliminary phase included experiments carried out freezing the pre-trained convolutional layers and fine-tuning the fully-connected layers, as explained in section 1.4.2. The results suggest to discard the CNN networks with the lowest accuracy percentages, i.e., *VGG16*, *Inception V3*, and *Xception*, keeping *InceptionResnetV2*, which achieved the best performance. In the second phase, the freezing on *InceptionResnetV2* has been removed, to further fine-tune all the layers of the network. For each network, the best model obtained in the training phase has been saved, rolling back the weights to keep the highest accuracy. Iterating this process, we obtained a final accuracy of 79.5% on the Validation Set using the best network, as shown in Table 1.7. As shown, the network that performed worse is *VGG16*, while the best network is *InceptionResnetV2*; in Table 1.7, we included the final results for all the networks. Note that only *InceptionResnetV2* has been fine-tuned in all its layers.

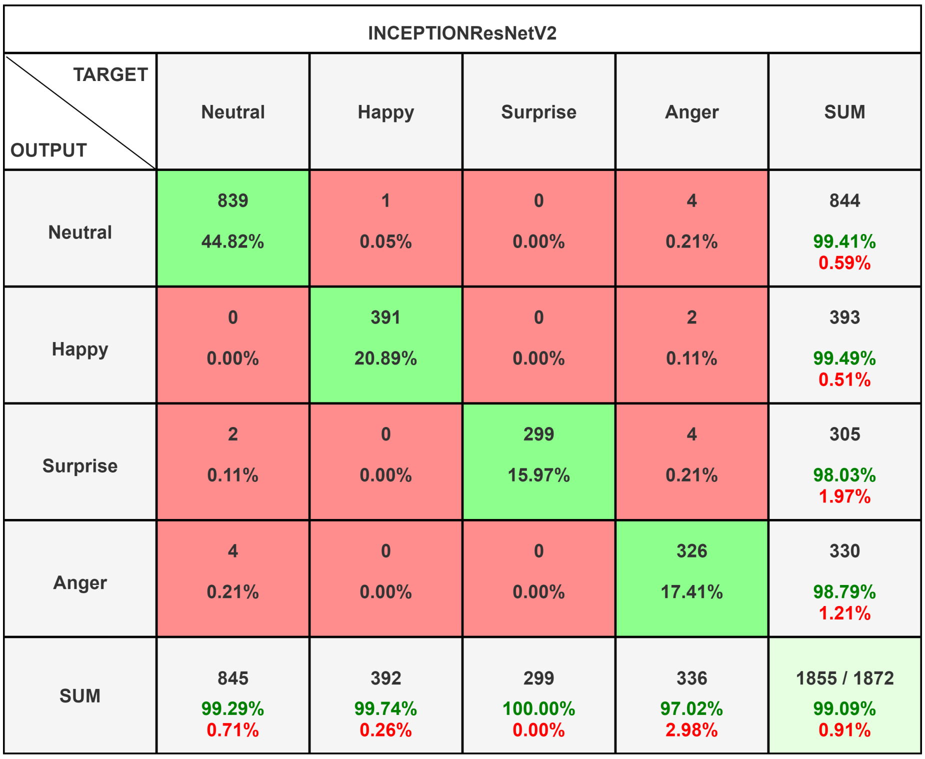
Network	Accuracy
InceptionResNetV2	79.5 %
Inception V3	77.0 %
Vgg-16	71.8 %
Xception	75.5 %

Table 1.7: Final results related to the considered CNNs

The number of epochs is different for each CNN, due to the use of the early stopping criterion. Early stopping is essential to eliminate or mitigate the overfitting effect that, as the epochs grow, would lead the accuracy on the training and validation sets, represented by the two curves (blue and orange) to diverge due to a loss of generalisation capability of the trained model. In agreement with the accuracy, *InceptionResNetV2* shows the best trend for the loss function.

The confusion matrix of the training set for the *InceptionResNetV2* network is reported in Figure 1.11.

The cells on the diagonal of the matrix shown in Figure 1.11, representing correctly classified images, show good performances, close to the perfect classification. Errors appear in classifying *Happiness*, sometimes interpreted as *Anger*; the same issues occur for *Surprise* interpreted as *Neutral* and *Anger*. Finally, *Anger* is sometimes misinterpreted as *Neutral*. As introduced, the misclassification of the *Neutral* emotion presents a different issue, thus requiring a separate discussion. Studies in cultural anthropology, including the first work of Paul Ekman on facial Emotion Recognition [65], show that humans sometimes tend to misclassify *Neutral* as *Anger* or *Sadness*, with different results in different cultures. Such bias is clearly reflected in the *AffectNet* data set because it is present also in our automated recognition. We can realistically suppose that our CNN is correctly classifying the mouths basing on biased labels. In both the classification directions, *Neutral* is sometimes misclassified, and the other classes are misclassified as *Neutral*. In Table 1.8 the absolute values of misclassified images is reported for each



INCEPTIONResNetV2

TARGET OUTPUT	Neutral	Happy	Surprise	Anger	SUM
Neutral	839 44.82%	1 0.05%	0 0.00%	4 0.21%	844 99.41% 0.59%
Happy	0 0.00%	391 20.89%	0 0.00%	2 0.11%	393 99.49% 0.51%
Surprise	2 0.11%	0 0.00%	299 15.97%	4 0.21%	305 98.03% 1.97%
Anger	4 0.21%	0 0.00%	0 0.00%	326 17.41%	330 98.79% 1.21%
SUM	845 99.29% 0.71%	392 99.74% 0.26%	299 100.00% 0.00%	336 97.02% 2.98%	1855 / 1872 99.09% 0.91%

Figure 1.11: Confusion matrix on the Training Set with the network InceptionRes-Net-V2.

class on the *Neutral* evaluation. We can say that our evidence suggests that our networks behave similarly to the human brain, possibly learning the features associated with the human bias present in the considered data set.

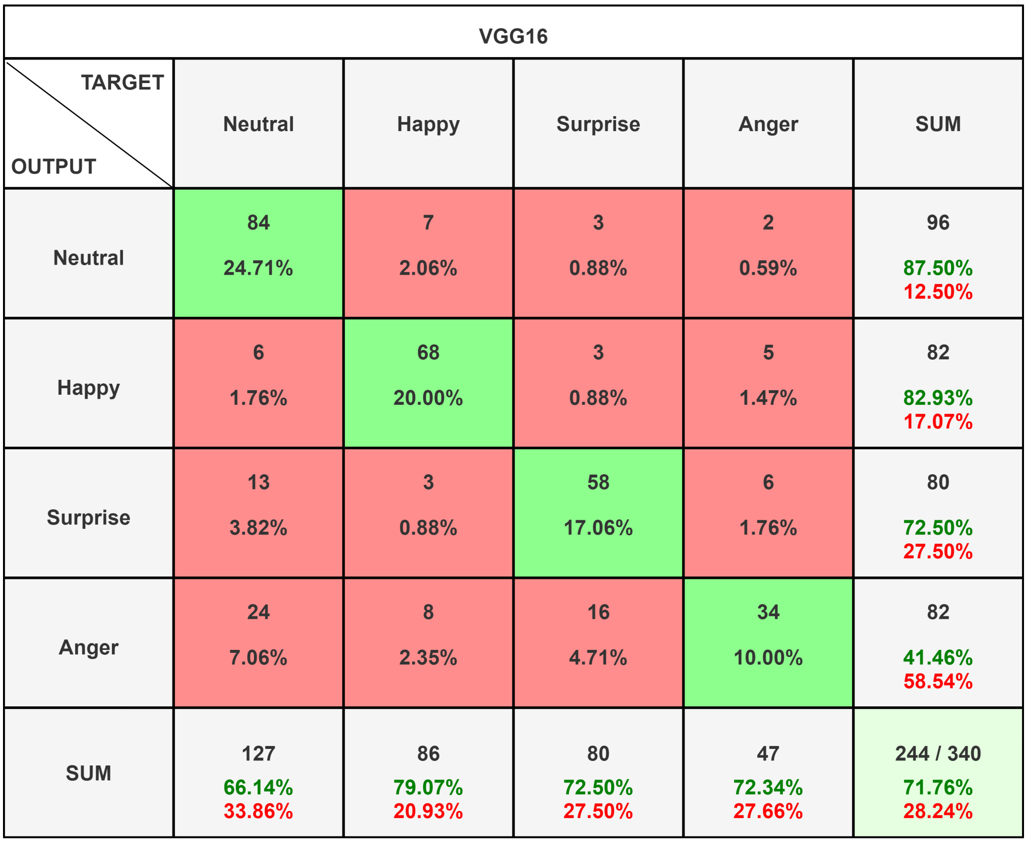
The confusion matrix of the validation set relative to each tested network is shown in Figure 1.12(VGG16), 1.13(Xception), 1.14(Inception V3), 1.15(InceptionResNet-v2).

	Happy	*Surprise*	*Anger*
# of images	11	7	4

Table 1.8: Number of misclassified images of *Neutral* faces in the three wrong categories over a total of 221 images.

The experiments have been replicated on images of the whole face, to study the contribution of the mouth; the resulting confusion matrix is shown in Figure 1.16.

Results can be summarised in the following points:



VGG16					
TARGET OUTPUT	Neutral	Happy	Surprise	Anger	SUM
Neutral	84 24.71%	7 2.06%	3 0.88%	2 0.59%	96 87.50% 12.50%
Happy	6 1.76%	68 20.00%	3 0.88%	5 1.47%	82 82.93% 17.07%
Surprise	13 3.82%	3 0.88%	58 17.06%	6 1.76%	80 72.50% 27.50%
Anger	24 7.06%	8 2.35%	16 4.71%	34 10.00%	82 41.46% 58.54%
SUM	127 66.14% 33.86%	86 79.07% 20.93%	80 72.50% 27.50%	47 72.34% 27.66%	244 / 340 71.76% 28.24%

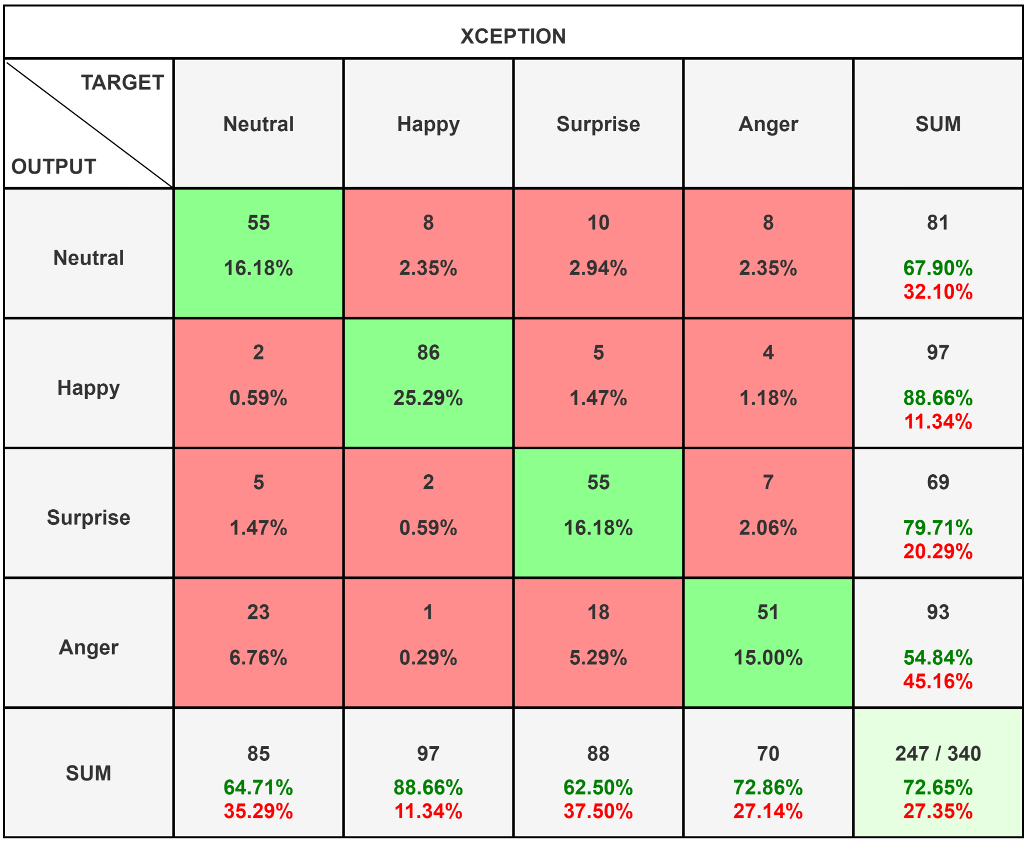
Figure 1.12: Confusion matrix on the Validation Set with the network VGG16.

- The network that provides the best performance is the same as mouth-based ER, i.e., *InceptionResNetV2*.
- *Anger* is again the most difficult emotion to classify.
- The performance improvement is only 5%, with an accuracy rate of 84.4%, thus proving the consistent contribution of the mouth for ER.

1.5 Object detection improvement with artificial data augmentation

In this section, we present a technique for generating synthetic datasets using the popular three-dimensional environment modelling software Unity[7]

[7] https://unity.com/



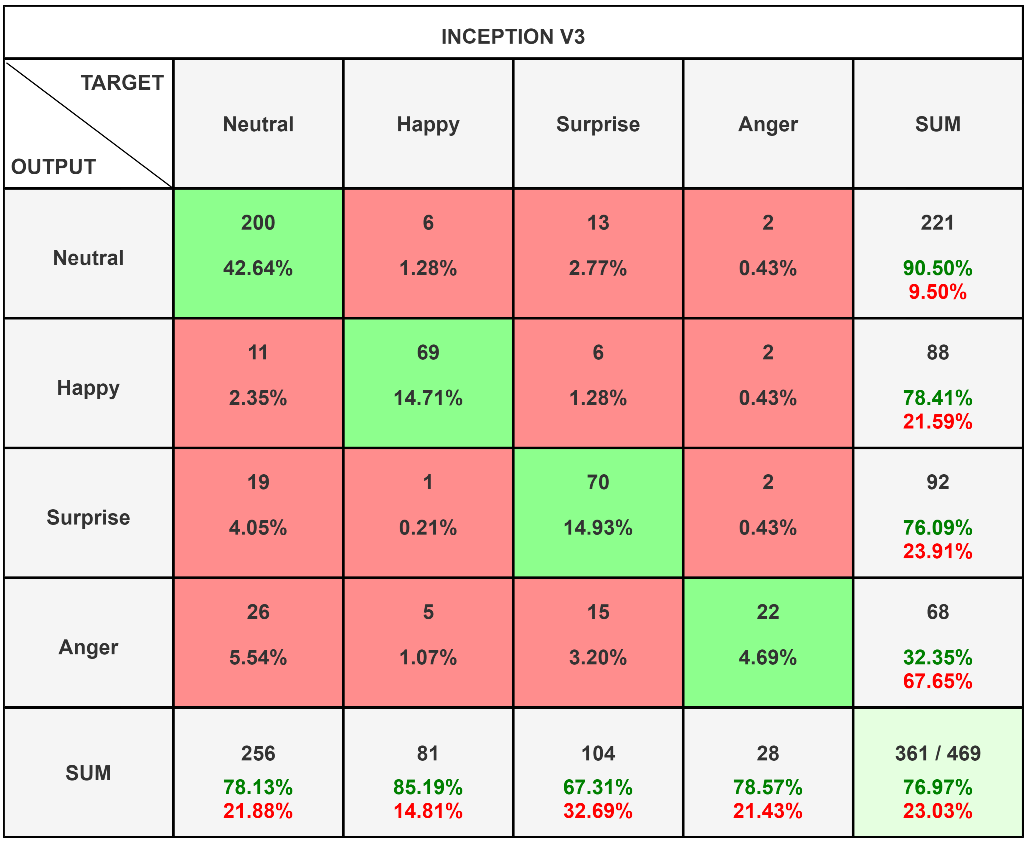
XCEPTION

TARGET OUTPUT	Neutral	Happy	Surprise	Anger	SUM
Neutral	55 16.18%	8 2.35%	10 2.94%	8 2.35%	81 67.90% 32.10%
Happy	2 0.59%	86 25.29%	5 1.47%	4 1.18%	97 88.66% 11.34%
Surprise	5 1.47%	2 0.59%	55 16.18%	7 2.06%	69 79.71% 20.29%
Anger	23 6.76%	1 0.29%	18 5.29%	51 15.00%	93 54.84% 45.16%
SUM	85 64.71% 35.29%	97 88.66% 11.34%	88 62.50% 37.50%	70 72.86% 27.14%	247 / 340 72.65% 27.35%

Figure 1.13: Confusion matrix on the Validation Set with the network Xception.

and Blender[8]. We illustrate two different *use cases*, which we think can be of valuable help to researchers, as they are general cases that can help to solve specific problems. The first *use case* involves generating a dataset for the binary classification of paintings of the Baroque and Impressionist styles. With this example, we want to show and describe how it is possible to generate images of objects (in our case paintings) and how they can be analysed using neural networks. This use case, therefore, involves framing a three-dimensional model from various angles and under various lighting conditions. The second use case involves generating a dataset for recognising men at sea, specifically migrants. This example is completely different from the previous one because instead of having an object, we have a scenario, and through graphical modelling environments we can easily recreate different weather and light conditions: for example, we can have a scenario with the

[8] https://blender.org/



INCEPTION V3					
TARGET OUTPUT	Neutral	Happy	Surprise	Anger	SUM
Neutral	200 42.64%	6 1.28%	13 2.77%	2 0.43%	221 90.50% 9.50%
Happy	11 2.35%	69 14.71%	6 1.28%	2 0.43%	88 78.41% 21.59%
Surprise	19 4.05%	1 0.21%	70 14.93%	2 0.43%	92 76.09% 23.91%
Anger	26 5.54%	5 1.07%	15 3.20%	22 4.69%	68 32.35% 67.65%
SUM	256 78.13% 21.88%	81 85.19% 14.81%	104 67.31% 32.69%	28 78.57% 21.43%	361 / 469 76.97% 23.03%

Figure 1.14: Confusion matrix on the Validation Set with the network Inception V3.

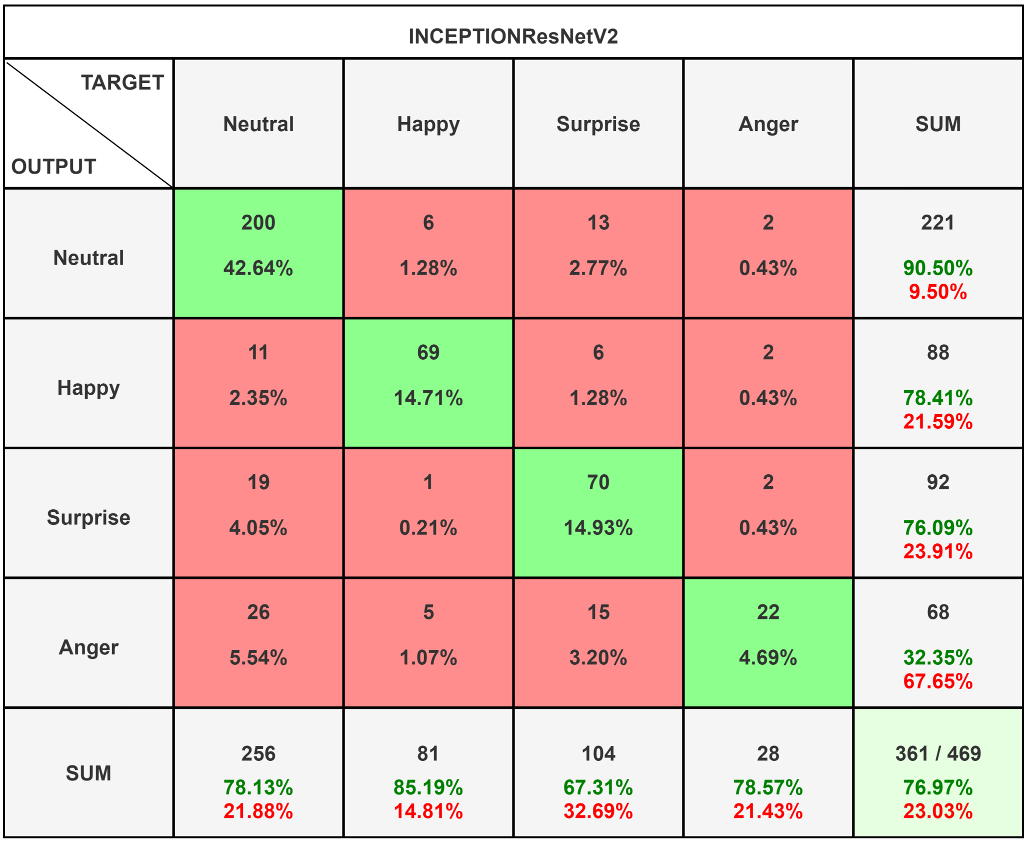
typical sunlight of the morning, afternoon, evening or night. We can also simulate what men at sea would look like in different weather conditions, for example, clear skies or cloudy skies. A further qualifying aspect of our work is that we have automated the process of data augmentation on virtually generated images by manipulating each image in various aspects.

The datasets that are generated for this article and the codes we developed to implement them have been made public and are freely accessible via the GitHub page[9]. All the code on the repository is open source, licensed under the GNU General Public License v3.0 and freely usable by anyone.

Research methodology

When approaching a machine learning challenge, the first step is to identify a dataset that accurately defines the problem and train a classifier using

[9] https://github.com/DamianoP/DatasetGenerator



INCEPTIONResNetV2

TARGET OUTPUT	Neutral	Happy	Surprise	Anger	SUM
Neutral	200 42.64%	6 1.28%	13 2.77%	2 0.43%	221 90.50% 9.50%
Happy	11 2.35%	69 14.71%	6 1.28%	2 0.43%	88 78.41% 21.59%
Surprise	19 4.05%	1 0.21%	70 14.93%	2 0.43%	92 76.09% 23.91%
Anger	26 5.54%	5 1.07%	15 3.20%	22 4.69%	68 32.35% 67.65%
SUM	256 78.13% 21.88%	81 85.19% 14.81%	104 67.31% 32.69%	28 78.57% 21.43%	361 / 469 76.97% 23.03%

Figure 1.15: Confusion matrix on the Validation Set with the network InceptionResNet-v2.

it, such as a decision tree, a neural network, or a Support Vector Machine. Manually constructing a dataset is a time-consuming, sometimes expensive, and even risky task. Consider the following scenarios: you want to train a neural network that recognises men overboard, or you want to train a neural network that recognises animals that pass along a road at night: these are specific scenarios that could present many challenges for a researcher, as well as a significant cost and time benefit from using graphic modelling software. The pipeline that we suggest for approaching these difficulties is depicted in Figure 1.17, which begins with the development of a synthetic dataset before moving on to the actual one.

Experimental Protocol

The construction of our datasets (each related to the case study dealt with) was carried out by acquiring images once the virtual scenario was created and a script was developed that generates random situations in both cases. In each photograph, the lighting, the shadows, the orientation of the object

INCEPTIONResNetV2					
TARGET OUTPUT	Neutral	Happy	Surprise	Anger	SUM
Neutral	188 40.60%	10 2.16%	5 1.08%	13 2.81%	216 87.04% 12.96%
Happy	8 1.73%	73 15.77%	0 0.00%	2 0.43%	83 87.95% 12.05%
Surprise	7 1.51%	1 0.22%	71 15.33%	6 1.30%	85 83.53% 16.47%
Anger	29 6.26%	0 0.00%	3 0.65%	47 10.15%	79 59.49% 40.51%
SUM	232 81.03% 18.97%	84 86.90% 13.10%	79 89.87% 10.13%	68 69.12% 30.88%	379 / 463 81.86% 18.14%

Figure 1.16: Confusion matrix of the Validation Set obtained running *InceptionResNetV2* analysing the whole face images instead of the mouth portion of the images.

concerning the camera, and the appearance of random objects (in the case of the men at sea) were varied. In this way, it was possible to generate a very high number of images necessary for learning the neural networks. About the use case of men at sea, a second dataset was generated, the validation set, with which we tested the ability of neural networks to recognise the presence of men overboard in new images synthetically generated. Regarding the use case of generating a dataset for the recognition of paintings, we used photographs of real paintings. In the construction of the training set and the validation set, the images were multiplied starting from photographs of the paintings and processing them in the Unity environment, varying the lighting, the orientation for the camera and the shadows. The images used for the construction of the testset, with which we measured the accuracy of the neural network in the recognition of paintings through images, were not subjected to any processing. The results of the experimental analyses are

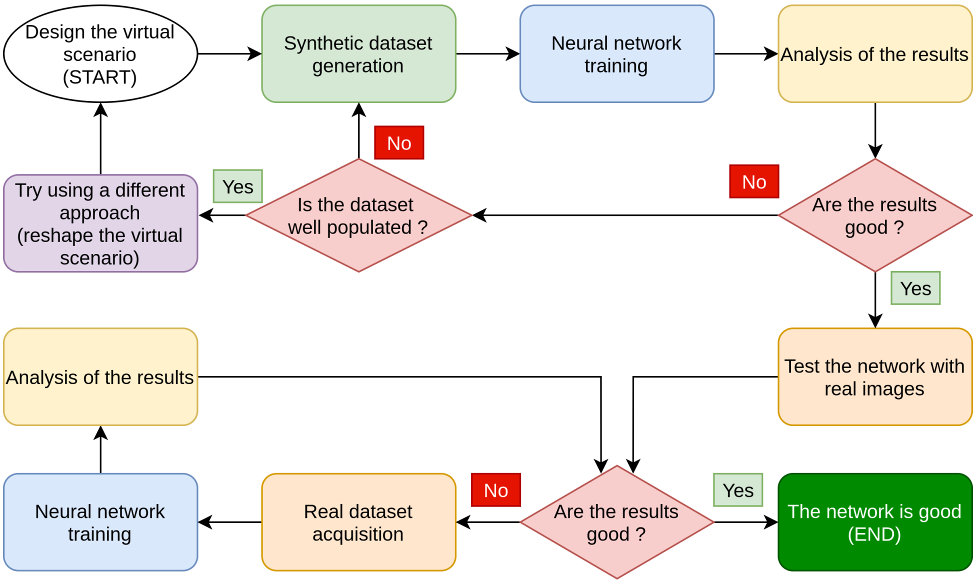
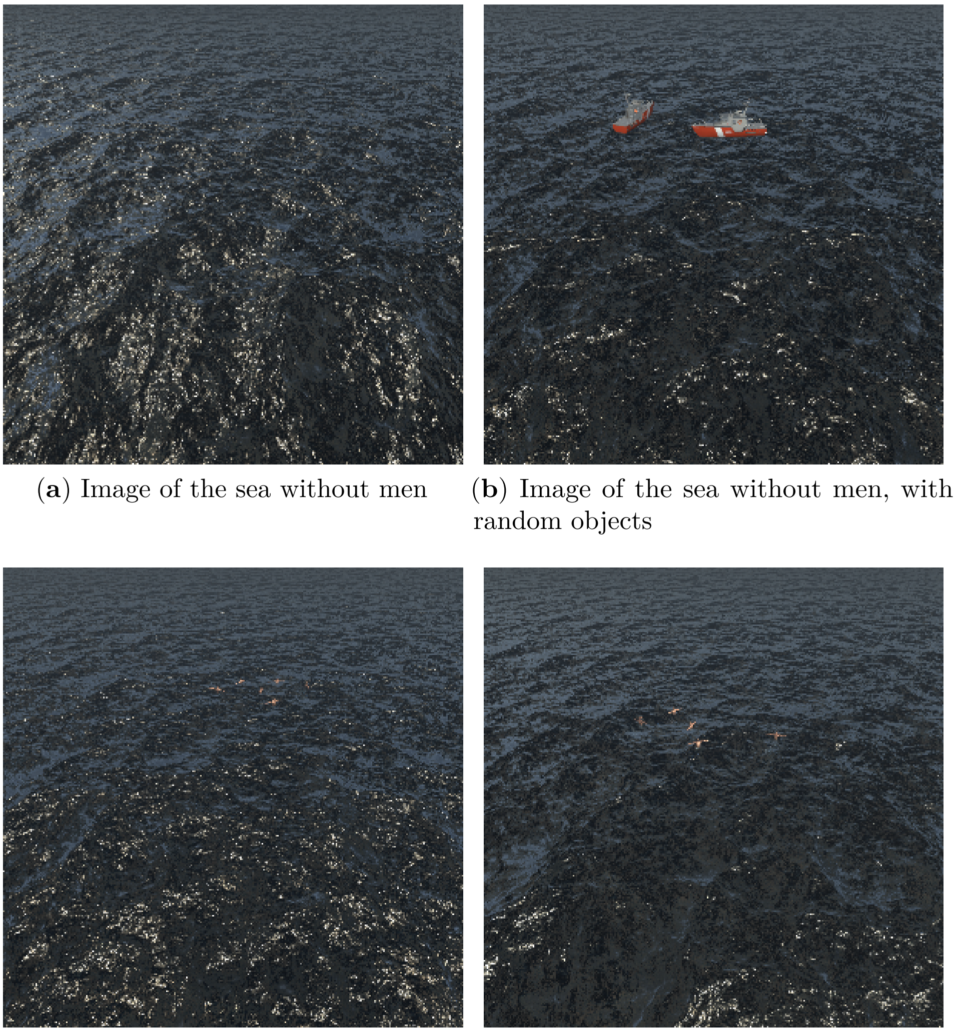

Design the virtual scenario (START)
Synthetic dataset generation
Neural network training
Analysis of the results
No
Are the results good ?
No
Is the dataset well populated ?
Yes
Try using a different approach (reshape the virtual scenario)
Yes
Test the network with real images
Analysis of the results
Neural network training
Real dataset acquisition
No
Are the results good ?
Yes
The network is good (END)

Figure 1.17: Flow chart summarising the various phases of the training of a neural network using virtual scenarios.

reported in the graphs and confusion matrices. The graphs will show the trend of the neural networks in the training phases, in the abscissas reported time instants (expressed as the number of epochs), while in the ordinates the percentage of accuracy in image recognition is reported.

Our proposed pipeline

The pipeline that we proposed is now being evaluated. First and foremost, the virtual setting must be designed and built using software such as Unity and Blender. This is a crucial stage, and having some example images and a clear sense of how the setting will be constructed might help. After you have finished creating the scenario, you may start creating the synthetic dataset. Photographs of the virtual world must be taken in this step, using combinations of light, shadows, and items that we deem appropriate for the research. After training the neural network, the results must be analysed: if the results are poor, the generated dataset must be double-checked, and the scenario modelling may need to be adjusted. If the results are satisfactory, we may assert that our problem can be solved using neural networks, and we can spend time and money looking for and photographing objects in the real world. If financial resources allow, it may be able to replace the synthetic images with real ones and evaluate the dataset generated by the neural network using the continuous learning approach. If the results are positive, the task is completed. Otherwise, you will have to go back and examine the dataset of real photographs, train the network again, and see if the goals have been met.



(**a**) Image of the sea without men

(**b**) Image of the sea without men, with random objects

(**c**) Image of the sea with men

(**d**) Image of the sea with men

Figure 1.18: Samples of the men at sea scenarios used in our work.

Instead of immediately beginning with the capture of real-world images, we advocate starting with synthetic photographs created using 3D modelling software utilising our pipeline. If the three-dimensional environment is created with care, realism, and detail fidelity, we will have a clear idea of the performances that we will be able to obtain in the real world quickly and with low initial costs, and only then begin the construction phase of a dataset with photographs taken in the real world.

We have discovered several advantages using this method. The first benefit is the speed with which we can generate photographs to train the networks because once we have set up the working environment, we will be able to generate an almost infinite number of images by simply running our algorithm and letting it generate images with random combinations of lights, shadows, and objects. Another advantage is the cost: developing a synthetic dataset is far less expensive than creating a genuine dataset, which in our example would include the hiring of a helicopter, actors, and at least one ship. It should also be remembered that by utilising a synthetic dataset, we will be able to determine much more quickly if the technique we are applying is appropriate or not, and, if required, entirely alter strategy and attack the problem from different perspectives. We believe that a synthetic dataset should not be used in place of a dataset made up of real images because it is currently impossible to faithfully reconstruct all of the graphic facets and decals that make up the natural world, but we do believe that it is a useful tool for researchers working on machine learning problems, particularly those involving image classification.

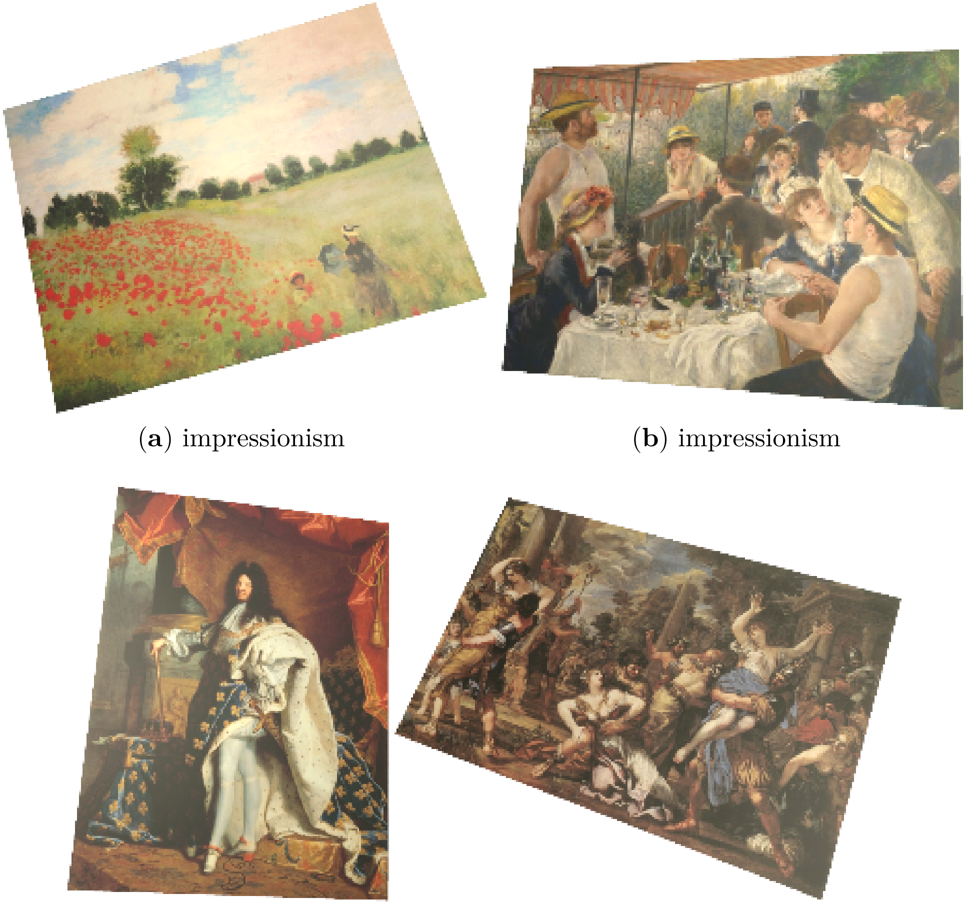
In Figure 1.18 it is possible to see some photographs of the scenario we created. In Figures 1.18A and 1.18B we can see two examples of photos in which there are no men overboard. In Figures 1.18C and 1.18D there are men overboard, generated randomly and at a random point of the scene. In Figure 1.19 we find 4 examples of paintings, which make up the dataset of objects we have created. In Figures 1.19A and 1.19B are represented paintings of the Impressionist period, while in Figures 1.19C and 1.19D are represented paintings of the Baroque period.

Discussion of results

We evaluated the synthetic datasets with neural networks after synthesising them with virtual reality software using the methodology outlined in the previous sections. The goal of the test is to determine whether or not a neural network can operate with pictures that are not from the real world and whether or not the results it generates are adequate. The analysis was carried out using two convolutional neural networks. The first network is Alexnet[62] while the second neural network is InceptionResnet-V2[66]. We trained the first neural network via Matlab[10] software while the second network was trained via Python code running on the Google Colab[11] cloud environment. Both networks were trained using the transfer learning technique[67]. This approach assumed that the networks would be partially trained before

[10] https://it.mathworks.com/products/matlab.html

[11] https://colab.research.google.com/



(**a**) impressionism

(**b**) impressionism

(**c**) baroque

(**d**) baroque

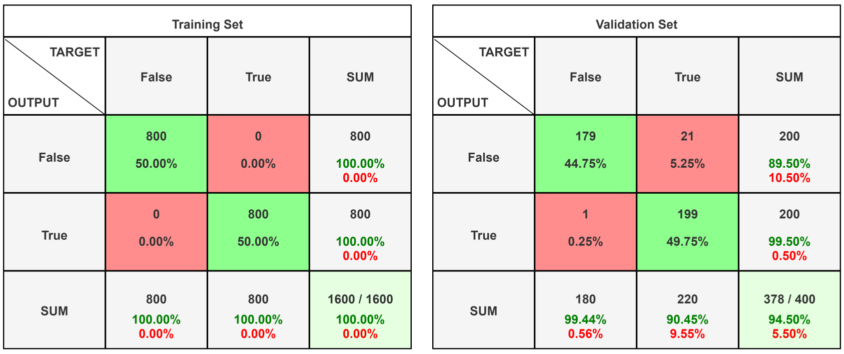
Figure 1.19: Sample paintings used in our work.

the training began. Before the actual training, the weights of the neural connections are preloaded. The values are derived from the training that these networks performed on the public dataset ImageNet[63]. We then deleted the network's head, as well as the initial 1000-class prediction layer, and added a layer to conduct a binary classification of our photos, for example, by utilising the sigmoid activation function[68]. Finally, after evaluating the general-purpose networks mentioned above, we constructed a customised neural network by modelling the internal structure of the layers to match our proposed problem. This network comprises 12 million neurons, which is a small amount in comparison to general-purpose networks, yet it has produced a good performance in the prediction phase on the validation set.

Alexnet

Alexnet was the first neural network we looked at, and we used Matlab software to analyse it. Using the transfer learning method, the neural network was imported and analysed. The network's last three layers were eliminated after it was pre-trained on the ImageNet dataset. These layers acted as learning layers for the network, allowing it to recognise the 1000 ImageNet classes. We have replaced these layers with a dense, fully connected layer with a WeightLearnRateFactor of 20 and a BiasLearnRateFactor of 20. A Softmax layer and, lastly, a Classification Layer were attached to this layer. We examined both the dataset depicting paintings and the dataset representing men at sea with the network setup in this way.

Figure 1.20 shows the confusion matrices obtained by analysing the dataset of men at sea.



Training Set

TARGET OUTPUT	False	True	SUM
False	800 50.00%	0 0.00%	800 100.00% 0.00%
True	0 0.00%	800 50.00%	800 100.00% 0.00%
SUM	800 100.00% 0.00%	800 100.00% 0.00%	1600 / 1600 100.00% 0.00%

(**a**) Training set Confusion Matrix

Validation Set

TARGET OUTPUT	False	True	SUM
False	179 44.75%	21 5.25%	200 89.50% 10.50%
True	1 0.25%	199 49.75%	200 99.50% 0.50%
SUM	180 99.44% 0.56%	220 90.45% 9.55%	378 / 400 94.50% 5.50%

(**b**) Validation set Confusion Matrix

Figure 1.20: Confusion matrices of the Sea dataset analysed on Alexnet with Matlab

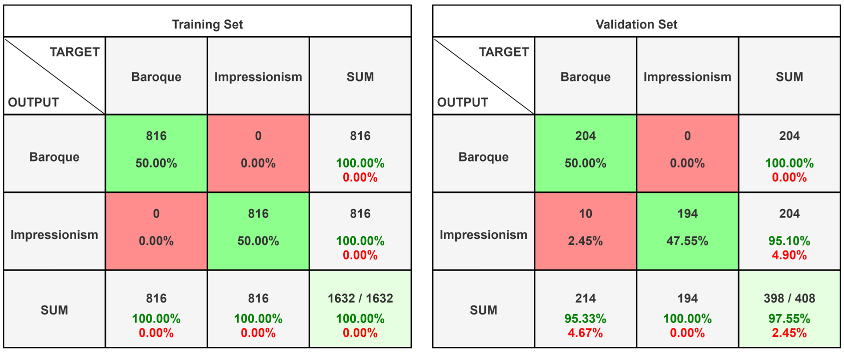
Figure 1.21 shows the confusion matrices obtained by analysing the dataset of the paintings.

Figure 1.22 shows the result of Alexnet's training on the two datasets.

The graph showing the network's percentage of accuracy as a function of time can be seen in the upper section of the photos. In the lower section of the photos, the Loss function as a function of time is presented.

The learning curve of the network in the recognition of the training set is shown in blue in Figure 1.22A, while the dataset of men at sea is studied. As you can see, the network begins to settle after a few early oscillations, becoming increasingly precise.

Figure 1.22B, on the other hand, depicts the network's learning curve as it analyses the collection of paintings. Blue highlights the various training iterations. We have added periodic checks in the recognition of the validation set as an extra analysis in this example, and the interpolation of the



Training Set			
TARGET OUTPUT	Baroque	Impressionism	SUM
Baroque	816 50.00%	0 0.00%	816 100.00% 0.00%
Impressionism	0 0.00%	816 50.00%	816 100.00% 0.00%
SUM	816 100.00% 0.00%	816 100.00% 0.00%	1632 / 1632 100.00% 0.00%

(**a**) Training set Confusion Matrix

Validation Set			
TARGET OUTPUT	Baroque	Impressionism	SUM
Baroque	204 50.00%	0 0.00%	204 100.00% 0.00%
Impressionism	10 2.45%	194 47.55%	204 95.10% 4.90%
SUM	214 95.33% 4.67%	194 100.00% 0.00%	398 / 408 97.55% 2.45%

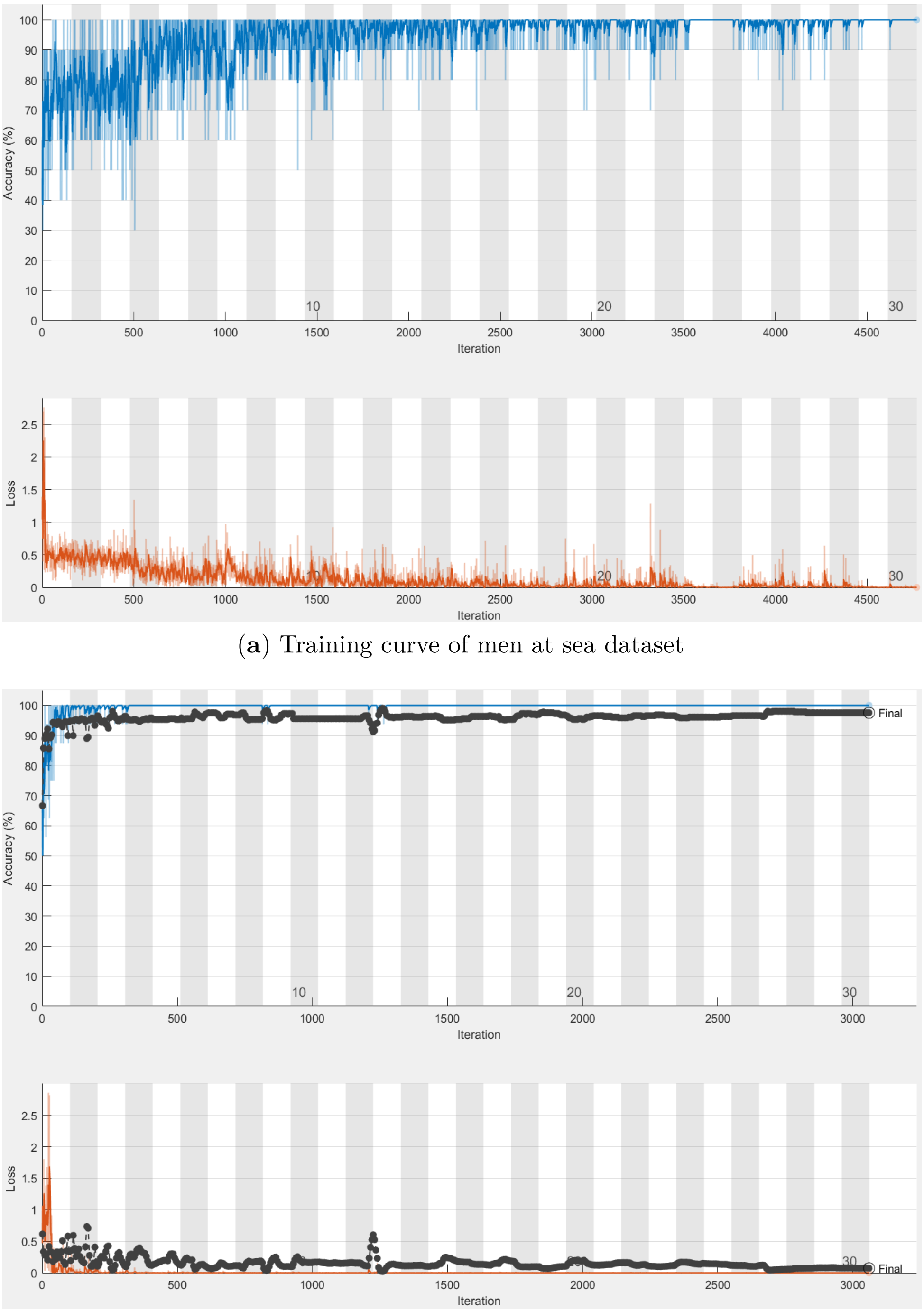
(**b**) Validation set Confusion Matrix

Figure 1.21: Confusion matrices of the Painting dataset analysed on Alexnet with Matlab.

recognition percentages achieved is shown by the black curve. Figure 1.20B shows the final result in the recognition of the validation set of the man overboard dataset. The findings acquired from the first tested network are, in our perspective, extremely good; in particular, we can see how the validation set paintings are identified with an accuracy of 97.5%. The accuracy of the dataset of men at sea, on the other hand, is lower, with an overall accuracy of 94.5%. When in confusion matrix we consider the cell corresponding to the column "true" and the row "true", that is, when there are men overboard, we can see that 199 of the 200 photos in the sample were properly identified. The Alexnet neural network described above takes up a total of 201.9MB on disk.

InceptionResNet-V2

We used the Google Colab cloud environment to test our second system, the InceptionResNet-V2 neural network. This network was also preloaded using the transfer learning approach, thus the weights of the neurons were already capable of classifying the 1000 ImageNet classes. We imported the network, removed the network's head, which was made up of dense learning layers, and replaced it with the following layers. The first is a Flatten type layer, whose job was to convert the network's data structure into a float vector. After that, we added two dense layers of 64 neurons with Rectified Linear Unit (ReLU) activation functions to learn the features collected from the network's convolutional layers. Finally, we have added a layer that just contains one neuron and is responsible for binary categorization. Binary



(**a**) Training curve of men at sea dataset

(**b**) Training curve of paintings dataset

Figure 1.22: Training curve of the datasets analysed on Alexnet with Matlab

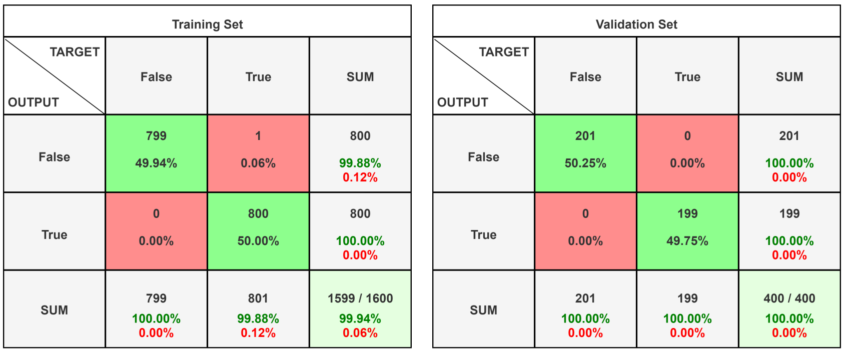
crossentropy[12] was employed as the loss function, while Adam[13][69] was chosen as the optimizer. In a cloud environment like Google Colab, the network described above has a total of 60,632,481 parameters, which is a big quantity to maintain and handle. As a consequence, we devised a strategy that enabled us to operate effectively with such a large network. The early layers of the InceptionResNet-V2 network were frozen, and only the final layers were permitted to be trained. The layers that extract picture characteristics and are pre-trained on ImageNet are not impacted in this way. Because of this approach, 54,336,736 parameters are immutable and will not change during the network's training, whereas the remaining 6,295,745 parameters will need to be trained to learn how to accurately categorise our datasets. We have additionally set up two more aspects. The first is in the training phase: we assess the network accuracy percentage on the validation set every time an epoch finishes. If the accuracy rate has increased, the network model is saved and this step is considered a checkpoint. In this method, even if the network overfits during training and its accuracy on the validation set deteriorates, we may still trace the optimum combination of parameters gained during training. The second point to consider is when to halt (*early stopping*). We can track the development of the network training phase and verify its accuracy in identifying the dataset using this method. After each period, a check is performed. If the network does not increase recognition accuracy for a predefined number of epochs (generally three), we may say that we have found a local minimum in the range of possible solutions to the problem. This abnormality will be detected, and network training will be halted, saving time that would otherwise be squandered.

Figure 1.23 shows the confusion matrices obtained by analysing the dataset of men at sea with the InceptionResNet-V2 neural network. In this case, the network recognises each image provided through the validation set, reaching an accuracy of 100%. The confusion matrices derived by examining the dataset of the paintings shown in Figure 1.24; the validation set has a very high accuracy percentage of 98.03%.

Furthermore, we report in Figure 1.25 the graphs obtained with the training of the neural network. The graphs show how the proportion of accuracy has increased over time, peaking at epoch n.10 and then declining. The network then entered an overfitting phase, during which the accuracy of the validation set decreased significantly. We were able to save the condition in which the network was at its highest level of accuracy, which was obtained around epoch n.10, thanks to early stopping and checkpoints. The neural

[12] https://keras.io/api/losses/probabilistic_losses/

[13] https://keras.io/api/optimizers/adam/



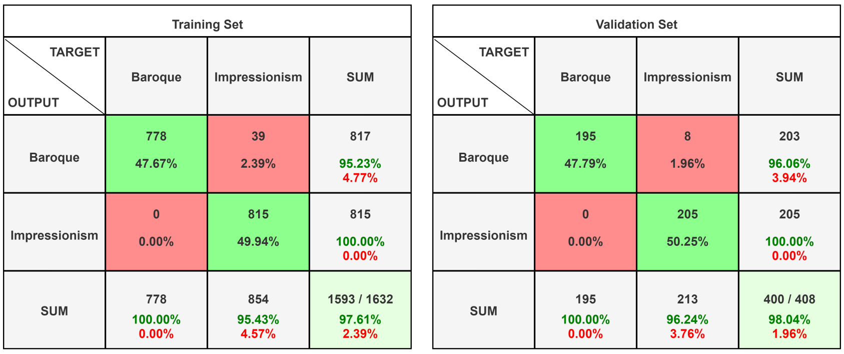
Training Set			
TARGET OUTPUT	False	True	SUM
False	799 49.94%	1 0.06%	800 99.88% 0.12%
True	0 0.00%	800 50.00%	800 100.00% 0.00%
SUM	799 100.00% 0.00%	801 99.88% 0.12%	1599 / 1600 99.94% 0.06%

(**a**) Training set Confusion Matrix

Validation Set			
TARGET OUTPUT	False	True	SUM
False	201 50.25%	0 0.00%	201 100.00% 0.00%
True	0 0.00%	199 49.75%	199 100.00% 0.00%
SUM	201 100.00% 0.00%	199 100.00% 0.00%	400 / 400 100.00% 0.00%

(**b**) Validation set Confusion Matrix

Figure 1.23: Confusion matrices of the men at sea dataset analysed on InceptionResNet V2 with Google Colab



Training Set			
TARGET OUTPUT	Baroque	Impressionism	SUM
Baroque	778 47.67%	39 2.39%	817 95.23% 4.77%
Impressionism	0 0.00%	815 49.94%	815 100.00% 0.00%
SUM	778 100.00% 0.00%	854 95.43% 4.57%	1593 / 1632 97.61% 2.39%

(**a**) Training set Confusion Matrix

Validation Set			
TARGET OUTPUT	Baroque	Impressionism	SUM
Baroque	195 47.79%	8 1.96%	203 96.06% 3.94%
Impressionism	0 0.00%	205 50.25%	205 100.00% 0.00%
SUM	195 100.00% 0.00%	213 96.24% 3.76%	400 / 408 98.04% 1.96%

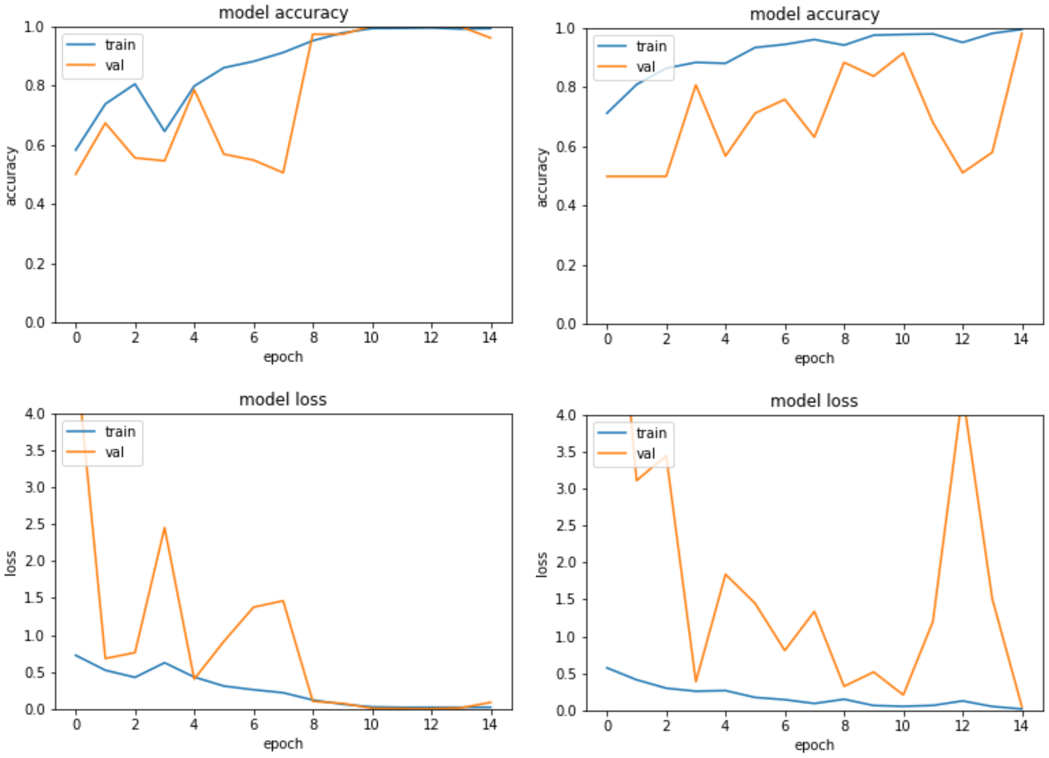
(**b**) Validation set Confusion Matrix

Figure 1.24: Confusion matrices of the Painting dataset analysed on InceptionResNet V2 with Google Colab

network as described takes up 54.5MB of storage space. The convolutional layers have been frozen, thus no changes in the weight values of the neural interconnections have been made compared to the network trained on ImageNet, allowing for such a small amount of space to be filled.

Custom Convolutional Neural Network

To finish our study, we constructed a custom neural network for the collection of paintings. The goal of this network's architecture is to achieve a more linear training phase than InceptionResNet-V2. Indeed, we want to avoid the frequent spikes and decays on accuracy that we saw with the general-purpose network, and we want the loss function to be smoother. In this approach, we want to develop a neural network that is theoretically more stable when



(**a**) Sea dataset (**b**) Paintings dataset

Figure 1.25: Training curve of datasets analysed on InceptionResNet-V2 with Google Colab

applied to pictures that are not synthetically created. The Figure 1.26 depicts the network we are presenting.

There are three blocks to its construction. The identification of the highest level characteristics is the initial block. We employ a convolutional layer sequence and max-pooling to do this. The three first convolutional layers have 64, 128 and 256 filters, respectively, as we go deeper into the network. The second block is made up of three convolutional layers, each with 128 filters. These layers are responsible for learning the finer details of our datasets. A series of dense layers with ReLU activation functions compose the third block. These layers are responsible for learning and categorising the characteristics retrieved by convolutional layers. The network's last layer is composed of a single neuron with a sigmoid activation function. Adam was the optimizer for this neural network, and there were a total of 12,209,553 parameters to train. The setup of the neuronal weights of this neural network requires 139.8MB of storage space. Figure 1.27 shows the confusion matrices

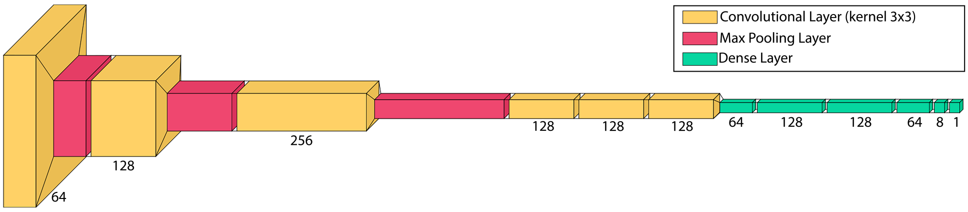
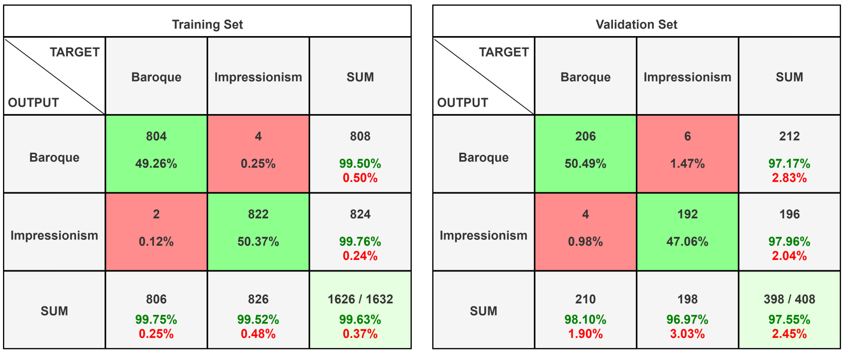


Figure 1.26: Structure of the custom Convolutional Neural Network

created by testing the neural network on the datasets. As it can be seen, the recognition percentages are quite high, with a validation set accuracy of 97.54%. Figures 1.28A and 1.28B show the statistics gained during the training phase, whereas Figure 1.28D shows the Roc curve. As we may see, the learning curve is highly steady, and the validation set's identification percentage is very near to the training set's recognition accuracy. Checkpoints and early stopping were also implemented in this situation, allowing us to save the optimal configuration of neural weights that the training phase could create. The best configuration was found at the end of epoch n.11, however, the early stopping of the training at epoch n.14, as the network was approaching overfitting, caused the training to be terminated. Finally, we conducted a deeper investigation. We created a test set of paintings that were not included in our synthetic dataset and utilised it to see if the neural network we trained could distinguish the painting style. The analysis gave very good results with an accuracy percentage of 94%. The confusion matrix constructed on the test set as presented is shown in Figure 1.28C.



Training Set

TARGET OUTPUT	Baroque	Impressionism	SUM
Baroque	804 49.26%	4 0.25%	808 99.50% 0.50%
Impressionism	2 0.12%	822 50.37%	824 99.76% 0.24%
SUM	806 99.75% 0.25%	826 99.52% 0.48%	1626 / 1632 99.63% 0.37%

(**a**) Training set Confusion Matrix

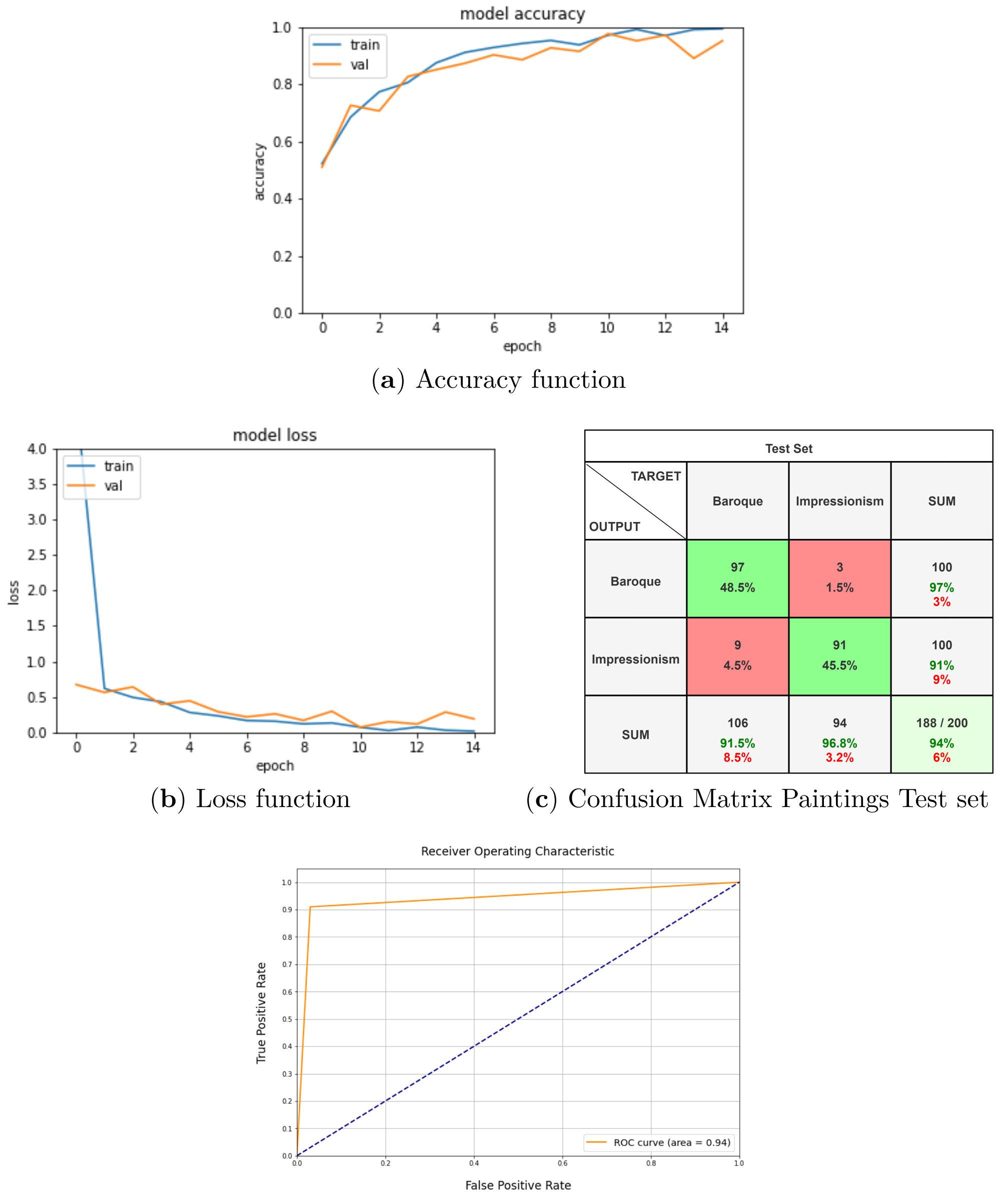
Validation Set

TARGET OUTPUT	Baroque	Impressionism	SUM
Baroque	206 50.49%	6 1.47%	212 97.17% 2.83%
Impressionism	4 0.98%	192 47.06%	196 97.96% 2.04%
SUM	210 98.10% 1.90%	198 96.97% 3.03%	398 / 408 97.55% 2.45%

(**b**) Validation set Confusion Matrix

Figure 1.27: Confusion matrices of the Paintings dataset analysed on Custom-CNN with Google Colab

Figure 1.29 shows the table comparing the results on the recognition



(**a**) Accuracy function

(**b**) Loss function

Test Set

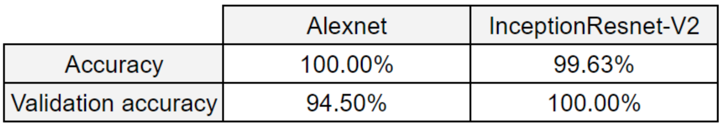
TARGET OUTPUT	Baroque	Impressionism	SUM
Baroque	97 48.5%	3 1.5%	100 97% 3%
Impressionism	9 4.5%	91 45.5%	100 91% 9%
SUM	106 91.5% 8.5%	94 96.8% 3.2%	188 / 200 94% 6%

(**c**) Confusion Matrix Paintings Test set

(**d**) Receiver Operating Characteristic Curve

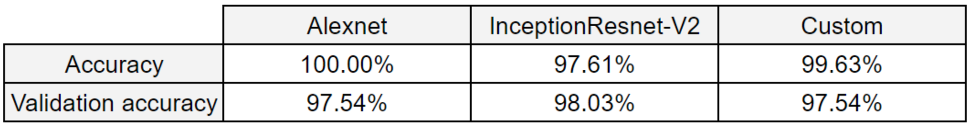
Figure 1.28: Confusion matrix and statistical analysis of the Paintings dataset analysed on Custom-CNN with Google Colab

of the men at sea dataset, according to the chosen metrics of the two neural networks tested, Alexnet and InceptionResNet-V2. It can be seen that for validation accuracy, the InceptionResNet-V2 network has a better validation accuracy. Figure 1.30 shows the table comparing the results on the recognition of the Paintings dataset, according to the chosen metrics, of the three neural networks tested, Alexnet, InceptionResNet-V2 and the Custom network. It is noted that for the validation accuracy, the InceptionResNet-V2 network shows a better validation accuracy, but the Custom network has several trainable parameters equal to 1/5 compared to Inception.



	Alexnet	InceptionResnet-V2
Accuracy	100.00%	99.63%
Validation accuracy	94.50%	100.00%

Figure 1.29: Comparing between Alexnet and InceptionResNetV2 CNNs for the use case of the men at sea



	Alexnet	InceptionResnet-V2	Custom
Accuracy	100.00%	97.61%	99.63%
Validation accuracy	97.54%	98.03%	97.54%

Figure 1.30: Comparing among Alexnet, InceptionResNetV2 and the Custom CNNs for the use case of the paintings

Best Practices generating a synthetic dataset in virtual environments

This section describes the techniques we recommend for generating synthetic datasets. Firstly we describe the techniques we recommend for generating synthetic scenarios using Unity. Next we describe how to generate synthetic datasets depicting objects. The techniques described can be applied generically to many other graphic modelling software we used Unity only as a use case.

Representation of the synthetic scenario

The first step is to use graphic modelling software to properly represent the scenario. One of the most well-known and widely used programmes to do this is Blender [14]. This software product also has a significant community that has created a lot of tutorials that may be helpful to individuals who are just getting started in the realm of graphic modelling. After you have received the model, it will be necessary to import it into Unity. It will be sufficient to place the model we produced into the gameobject after constructing an empty gameobject to use as a container. As a result, a camera must be

[14] https://www.blender.org/

placed in the scene and positioned so that it frames precisely the scene, as illustrated in Figure 1.31.

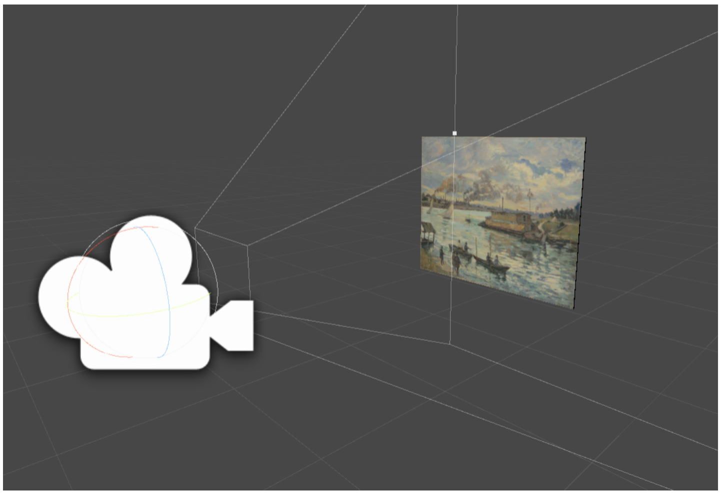
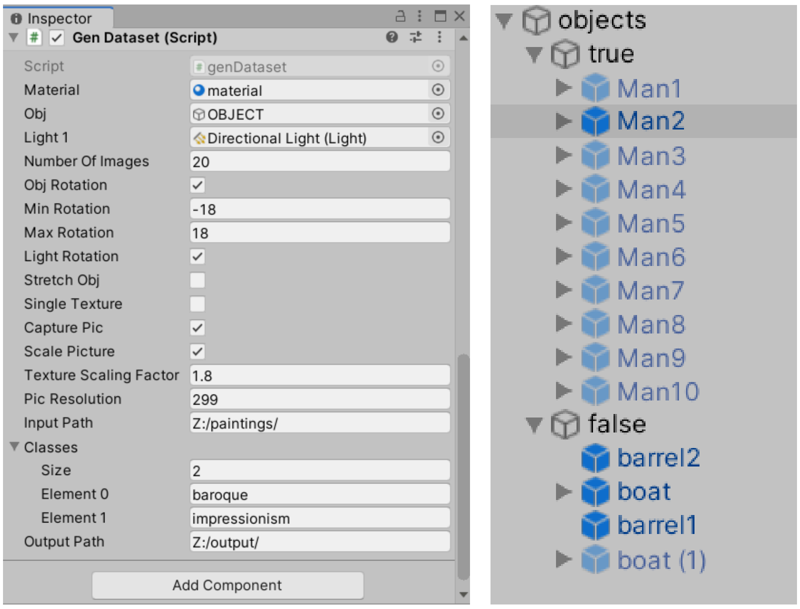


Figure 1.31: Camera that precisely frames the scene.

Depending on the lighting conditions we want to reproduce, we will also need a directional light and maybe one or more pointlights or spotlight types of lights. After you have completed these preparatory procedures, you will need to write a script that will change the lighting conditions at random, rotate the item, and take photos with the appropriate resolution. We propose including some public variables in the script so that you can define the parameters using the Unity graphical interface (the Inspector) rather than having to change the code all the time. Some examples of public variables are the path to the output image folder, the number of photographs to be taken, the object's and light's minimum and maximum rotations, the light's minimum and maximum intensities, and so on. Figure 1.32a shows an example. The operations flow that we suggest is as follows. Execute the following procedures within a loop that iterates a number of times equal to the number of photos we wish to generate:

1. Random rotation of the object: we produce three random integers that are contained in the minimum and maximum rotation ranges that we previously established and assign them as coefficients of the object's X, Y, and Z rotations.

2. Changing the scene's global lighting: we produce three random integers that represent the potential rotations of the light in the scene; it is critical to establish the ranges of the three variables accurately to



Inspector

☑ Gen Dataset (Script)

Script	genDataset
Material	material
Obj	OBJECT
Light 1	Directional Light (Light)
Number Of Images	20
Obj Rotation	☑
Min Rotation	-18
Max Rotation	18
Light Rotation	☑
Stretch Obj	□
Single Texture	□
Capture Pic	☑
Scale Picture	☑
Texture Scaling Factor	1.8
Pic Resolution	299
Input Path	Z:/paintings/
Classes	
Size	2
Element 0	baroque
Element 1	impressionism
Output Path	Z:/output/

- objects
  - true
    - Man1
    - Man2
    - Man3
    - Man4
    - Man5
    - Man6
    - Man7
    - Man8
    - Man9
    - Man10
  - false
    - barrel2
    - boat
    - barrel1
    - boat (1)

(**a**) Inspector, the script and the parameters (**b**) Objects in the sea

Figure 1.32: Unity interface

guarantee that the representation obtained is believable. A picture with illumination from the bottom up, or even directly into the viewer's eyes, for example, would be unusual. As a result, we recommend trying until you get the appropriate outcome. These numbers are then allocated as light rotation coefficients once they have been formed. It is also possible to produce a random value that alters the light's intensity and hue.

3. Acquisition of image: to store a photograph of the scene, generate a rectangle that overlaps the user interface starting at the x_0, y_0 coordinates of (0,0) and finishing at the x, y coordinates of (Screen.width, Screen.height), then extract the RGB values of the pixels included inside the rectangle.

4. Scaling the image: after we have gotten the RGB values, we will need to scale the image to fit the size requirements of the neural network we are going to utilise. For example, if we are creating a dataset to train the InceptionResNetV2 network, we may scale the photos directly in Unity to 299x299 resolution.

5. Saving the image: once the pixels have been captured and resized, the

image must be saved to a file system, for example, in the PNG format. During the saving step, we propose distinguishing the objects using an identifying name and an incremental integer that will be used to create the saved file's name, such as `"impressionism_0000X.png"`.

Dataset for the classification of images representing objects The followings are best practices for generating virtual scenario datasets using Unity. Now let us assume we want to do a true-false binary classification. First and foremost, the models of the things that will make up the scene must be created. The models must be realistic, and we suggest Blender in this situation as well. It is also important to consider how to construct the examples for the *true* and *false* classes. We must also ensure that the items are not always arranged in the same area of the image; to do so, we need to designate locations where they can be formed and then displayed to the camera. We created a scenario suitable for both photographs associated with the *true* category and photographs associated with the *false* category, in which we represented men at sea, shipwrecked people, and created a scenario suitable for both photographs associated with the *true* category and photographs associated with the *false* category. The images in the *true* category account for half of the whole dataset. The remaining 50% are in the *false* category, which is split into two sub-categories, with half of them representing simply the scenario with the sea and random light conditions. The other half is a representation of the sea, with ships and barrels floating on the surface. First of all, we recommend generating and putting a container object, in our case *"objects"*, into the scene. We have added two more containers to it, one for objects of the *true* class and the other for objects of the *false* class. Three-dimensional models of people, ships, and barrels have been placed within the two containers. Figure 1.32b provides an example.

As a result, the flow of the operations that we suggest is as follows. The following procedures should be executed within a loop that iterates a number of times equal to the number of images we wish to generate:

1. Restore the scene to its original state by hiding all items on the scene except the sea, the camera, and the sunlight at the start of each new iteration.

2. Make a random number of the objects active (and so display them within the scene) while producing pictures of the class *true/false*.

3. Alter the location of the objects: for each object in the scene, we generate three random numbers (X, Y, and Z) from the range of

coordinates that the camera can frame and update the position of this object to the produced coordinates.

4. Change the rotation of objects: for each object in the scene, we produce three random integers (X, Y, and Z) that are within the desired rotation interval and realistically match the class to be formed. Then, using the provided values, rotate the item under examination.

5. Modify the lighting of the scene as described in point 2 in the previous list.

6. Acquire the scene image as specified in the preceding list's point 3.

7. Resize the image to the size acceptable by the neural network we wish to test, as explained in the preceding list's point 4.

8. Save the picture to the file system as stated in point 5 from the preceding list, be sure to give each image a name that allows you to identify the class to which it belongs.

1.6 Conclusion

With the research path discussed in 1.2 we presented and analyzed several models learned by different machine learning techniques for designing highly-optimized adaptive input-aware libraries. The analysis shows that models learned from GTB, and in general DT-based techniques, and MLP are more effective than other models by enabling efficient heuristics even in the presence of few features.

With the research path discussed in 1.3 we presented an improved Convolutional Neural Network for the classification of protein structure. Preliminary results appear encouraging, showing that the methodology used for the representation of the protein geometry, based on a 2D projected image associated with each molecule, retains most of the spatial information and is suitable for recognition. Also, the computational performances seems quite good, being calculations extremely fast as we provide the neural network with an image for each molecule.

However, considerations on the general nature of the CNNs used lead us to think that specifically designed neural networks could significantly improve the results, or even outperform them. A further research path worth being followed is to train the neural network using a greater number of samples, to better analyze the link between the samples' structural complexity and the classification capacity of the neural network itself. We believe that even better performance might be achieved if we developed a neural network customised for the graphical representation we proposed. Our representation is in fact very particular and within our images the areas with a high information content are located in very specific sectors of the images. Furthermore, a personalised neural network could also reduce the size in MegaBytes of the model obtained in output.

The research path discussed in 1.4.1 and in 1.4.2 demonstrates the performance of the neural networks in the image classifications. Using the *InceptionResNetV2* neural network we obtain an accuracy of 79.5% on the classification of the emotion by analyzing the images of the validation set and this result is obtained by analyzing only the mouth of the subjects. With respect to the full face, results show a loss of accuracy of only 5%, which in our opinion, is offset by the advantages of our method. The mouth is, in fact, a critical element of human face recognition, almost symmetric and usually visible from any perspective, thus the ideal element to focus in all those cases where the user can be shot from any point of view. Moreover, focusing on a smaller area of the image requires fewer computational capabilities.

A straightforward application of our approach is emotion/pain recognition for healthcare management and automated supervision of critical patients, e.g., bedridden in hospitals. The system can support patients using an ER system when direct human assistance is not available, e.g., during the night or in wards where assistance is not allowed, for advanced detection of an initial pain or discomfort state, raising a signal and letting the sanitary staff be informed to react promptly, avoiding the patient suffering. Supportive systems can be planned using Emotion Recognition, e.g., to assist patients after car accidents, still in the emergency phase or post-surgery, in order to understand their pain level. If the system recognises a critical situation, it can report the case to nurses or send an intervention/checkout request for the patient's room. Our approach could be easily extended to the scenario previously described, upon availability of proper data sets on pain images. An additional application can be the early recognition of depression states or the support of people with difficulties to see or interpret emotions, e.g., blind users, or people with autism spectrum disorders. Our Emotion Recognition system is planned to give feedback to the user (e.g., text, emoticon, sound) or set a channel for information transfer to software.

With the research path discussed in 1.5 we have demonstrated that synthetic datasets may be a valuable resource for researchers utilising machine learning algorithms to identify objects or scenarios. When dealing with challenges of this nature, you often have two options: either use datasets that other scientists have made accessible on the internet or invest time and money in constructing an ad-hoc dataset specific to the topic at hand. It takes time and money to collect materials and images to create a dataset, which generally comprises tens of thousands of shots. Taking the photos required to train a neural network might be risky in some circumstances. We can naturally think, for example, of the training of the network of men at sea, the recognition of animals or pedestrians along the motorways for the automatic braking systems of cars, and others. The synthetic datasets created by the pipeline and the methodologies presented in this article allow us to accelerate this process and predict which kinds of images will perform better for the task at hand. We also believe that, as a result of the high degrees of realism achieved by computer graphics, the image quality is quite good and will continue to improve, allowing the construction of increasingly realistic datasets. We are going to expand our study in the future and concentrate on particular elements, such as the union of synthetic and realistic datasets, by examining how neural networks trained on synthetic datasets react while adding instances to the original dataset and utilising continuous learning approaches.

Chapter 2

Virtual and Augmented Reality impact on real-life scenarios

This chapter describes the second of four research strands that have been addressed during the three-year PhD programme. Various works and activities will be described using, among other technologies, virtual reality and augmented reality.

2.1 Introduction

Virtual reality and augmented reality techniques are constantly evolving. Thanks to the development of increasingly high-performance hardware, they make it possible to achieve results and levels of realism unthinkable just a few years ago. The increase in computational power of the GPU makes it possible to create highly complex three-dimensional models rich in vertices, polygons and very high-resolution textures. As a result, the virtual environments that can be realised allow the user to be immersed in photo-realistic worlds that can even trigger the suspension of disbelief. This sentiment is a psychological state where the user stops considering the virtual world as a fake and detached environment but rather as an alternative reality that engages him or her psychologically. Section 2.2, we describe work that aims to reconstruct buildings and environments with computer graphics allowing people to view arts and culture from home during the pandemic period. This work was published in a conference, and the candidate is one of the authors [70].

The reproduction of images through computer graphics and synthetic reconstruction has always been searching for a compromise between the quality of the result and processing performance [71, 72]. A scene taken from a video, or an image, to be considered credible from the point of view of realism, needs a particularly advanced level of detail, which inevitably requires an underlying complex numerical processing [73]. Indeed, every image has got to be divided into numerous small adjacent polygons, the number of which is proportional to the quality and fidelity of the reproduction itself [74, 75, 76]. Certainly, the advent of more performing hardware such as GPUs has allowed a significant leap forward in the main sectors of computer graphics and virtual reality [77, 78, 79]. More recently, artificial intelligence, and more specifically convolutive networks, has made it possible to quantitatively raise the level of realism of images and video streams [80, 81, 82].

The goal becomes particularly complex when one wants to create 3D models of historical and archaeological items and monuments in their existing condition. So, a more sophisticated approach capable of capturing and digitally modelling these places' precise geometry and morphological features is required [83]. For years, close-range photogrammetry has been proposed for cultural heritage documentation and has worked very well. This conventional approach has been supplanted by digital close-range photogrammetry due to recent computer and information technology advances. This innovative technology opens up new possibilities like landscape laser scanning, automatic orientation and measurement operations, 3D vector data production, digital ortho-image generation, and digital surface model development [84]. Whereas many methodologies and sensors are now available, the correct strategy for obtaining a better and more realistic 3D model with the necessary level of detail is still a blend of multiple techniques and models. That is why a single approach is not finally capable of providing consistent results in all scenarios [85]. However, it is precisely this mix that enables a high-quality graphic reproduction and the possibility of manipulating the image to highlight and better understand its constituent details. The virtual environments present in the images can then be rotated and resized; one can enter them and explore them in their spatial depth, appreciating the aesthetic elements [86, 87]. Furthermore, each environment can be enriched with a series of important information and data of different nature (*data and metadata*), such as texts and audio, useful for better describing the object under examination [88, 89].

This outstanding potential to create a diverse cognitive experience in humans is precisely what makes these modern technologies an effective way of communicating ideas and culture; they already pervade numerous domains ranging from education [90] to healthcare [91, 92], from recreation and entertainment [93] to tourism [94].

The growing interest in these technologies in tourism and cultural promotion is evident; by their nature, they allow the enhancement, knowledge and accessibility to the vast public of humankind's immense cultural heritage, even remotely.

We use two important software: The first is Blender[1], open-source software that allows the three-dimensional modelling of objects and the faithful representation of environments. The modelling tools made available to the user are of particular importance, rich in functions performed by the software that greatly facilitate the user's work. Blender is a cross-platform 3D modelling programme, i.e. it can run on various operating systems such as Linux, Windows and MacOS, offering developers the possibility of working independently of the computer being used. Blender is released under the Gnu Public Licence (GPL) and allows numerous creative functions. These include the possibility of modelling three-dimensional objects, creating animations, rendering complex scenes and exporting the result by producing PNG or JPEG images. The latest versions of this programme have considerably improved the user interface, lowering the learning curve and allowing it to be used even by people who do not have the aptitude for advanced computer software. Polygon models made with Blender can also be exported as OBJ or FBX files. These files are highly compatible and can be used by many other complementary programmes, such as Unity.

The second software used is Unity[2], which is frequently used for the realisation of three-dimensional environments and interactive synthetic scenarios, in particular videogames and serious games. The creation of immersive environments is facilitated by the presence on the market of software with a large community. Thanks to constant updates, it keeps pace with the needs of developers. Unity makes it possible to realise immersive environments by inserting models and assets created with Blender into the scenes. The scenes realised in Unity can then be exported for the platforms on the market today, such as desktop computers with Windows operating system or consoles or Android or iOS smartphones.

Section 2.3, we describe work that aim help patients with cognitive impairments with the option of practising from home using a tablet, smartphone, or desktop computer. This work was published in a journal, and the candidate is one of the authors [95].

In 1970 Abt coined the phrase "serious game" [96], which refers to games with a stated and thoroughly thought-out educational aim and are not intended to be played merely for entertainment. Nowadays, "serious game" is a widespread and specific phrase that refers to any video game-based learning and instruction (e.g. business, military, medical, marketing). Serious

[1] https://www.blender.org

[2] https://unity.com

games can place in both professional and casual contexts, and their target audiences range in age [97, 98]. An increasing number of researchers and educators have worked to integrate serious games into schools [99, 100]. For years, attempts have been made to establish the potential effects of this type of activity [101]: the question is how far games may be utilised for educational reasons [102, 103]. Serious games provide a virtually immersive environment in which students can experiment and repeat tasks that are rare in their ordinary lives [73]. In fact, it is argued that their application in education can increase the likelihood of providing students with a real and deep form of learning [104, 105]. The use of virtual reality and augmented reality seems to have a profound positive influence on activating specific areas of the brain; this gave rise to the idea of stimulating specific nerve centres to recover functionality that had gradually been lost due to illnesses or traumatic events [106]. So, in recent years, there has been an increase in interest in bringing healthcare services that can be operated remotely. The increased demand for healthcare services drove the development of easy and quick telerehabilitation solutions in various sectors. These services were designed to be reliable and routine (rather than just pilot experiments), and they are now even more vital in providing continuous and efficient health care [107, 108]. Telerehabilitation was initially used in motor rehabilitation, but it was later discovered to be beneficial in cognitive rehabilitation [109, 110]. Specific cognitive capacities can be severely impaired due to various conditions and events, the majority of which are age-related issues like dementia, circulatory difficulties, head traumas, persistent mental diseases, or brain pathology [111, 112, 113]. Following a thorough neuro-psychological examination that reveals the presence of cognitive deficiencies, neurological rehabilitation cycles can be carried out to enhance cognitive functioning, stabilise deficits, boost residual cognitive capacities, and reduce the path of decay.

Virtual reality-based telerehabilitation systems use three-dimensional virtual environments to stimulate and obtain specific movements from the patient. The patient can view the virtual environment on the computer screen or utilise fully immersive devices, like Head Mounted Display, 3D movement sensors, and haptic devices [114, 115, 116]. The effectiveness of the exercise will be directly proportional to the degree of immersiveness the user experiences. Currently, there are three theoretical-practical schools of thought underlying the rehabilitation of neurological deficits. The first one proposes to address the rehabilitation of cognitive processes through non-specific stimulation. The second, on the contrary, suggests that recovery must necessarily pass through specific stimulation of the disorder; this type of intervention can be differentiated into a restorative approach (recovery) or a substitution approach (compensation). The third does not deal with the specific disorder but proposes stimulating the residual abilities to overcome the difficulties and inhibitory effects, guaranteeing the most significant possible autonomy.

Our methodology implements the third approach by creating exercises that, by stimulating the patient's neurological activity, allow him to recover functionality and find alternative neurological pathways to the traditional ones damaged by the trauma suffered. The creation of virtual environments designed ad-hoc, enriched with visual stimuli and evocative contents that arouse the patient's interest, have the function of accelerating and stimulating this rehabilitation process.

Section 2.4, we describe work for the rapid prototyping of cognitive telerehabilitation exercises. This work was published in a journal, and the candidate is one of the authors [106].

Section 2.6, we describe work for the help people affected by Visual Snow. This work was published in a conference, and the candidate is one of the authors [117]. Visual Snow Syndrome is a chronic condition that has only been described and studied in recent years. Its sufferers have a visual impairment in which tiny dots of light are superimposed on the perceived image, which is difficult to describe and explain. Generally, the image perceived by these patients is described as that obtained with an incorrectly tuned television setting. In addition to the ether's information content, we get a partial snow effect. Visual snow is a neurological issue portrayed by a constant visual unsettling influence that involves the whole visual field and is depicted as tiny gleaming flecks similar to old detuned TV [118]. Notwithstanding static, or 'snow', impacted people might encounter extra visual side effects like visual pictures that continue or return after the image has disappeared, aversion to light, unique visualisations from inside the eye and hindered night vision.

The causes of visual snow in patients are still relatively obscure. The average age when the visual snow appears for the first time in the subjects seems, by all accounts, to be more premature than numerous other neurological problems [119]. This initial phase is almost always accompanied by a general lack of recognition of the pathology by specialists; this means that it is still an uncommon question.

Research suggests that visual snow is a mental problem; a preliminary examination of functional cerebrum imaging [120] and electroencephalographic tests propose this interpretation [121].

Visual snow is a physical condition, most often exceptionally disabling, that emerges suddenly and is highly complex to diagnose and treat [122]. That is due to the fact that it is still an open field of study: there is little much-targeted research on the phenomenon and those that do exist need to be reviewed and synthesised [123].

Section 2.7, we describe a research path that aim to improve the quality of teaching in schools. We introduce a system based on VR (Virtual Reality) for examining analytical-geometric structures that occur in the study of mathematics and physics concepts in the last high school classes. This work was published in a conference, and the candidate is one of the authors [73]. In our opinion, an immersive study environment has several advantages over traditional two-dimensional environments (such as a book or the simple screen of a PC or tablet), such as the spatial understanding of the concepts exposed, more peripheral awareness and moreover an evident decreasing in the information dispersion phenomenon. This does not mean that our pedagogical approach is a substitute for traditional pedagogical approaches, but is simply meant to be a robust support. The system which provides for the integration of two machine levels, hardware and software, was subsequently tested by a representative sample of students who returned various food for thought through a questionnaire.

2.2 Digitalization of cultural heritage

We aims to identify a series of methodologies that enable the creation of virtual worlds through which art treasures can be accessible to a vast population in a simple and relatively inexpensive manner in various ways. To this end, use cases are presented which were realised using two of the most successful approaches, based on the most modern virtual and augmented reality techniques. The first use case was realised through photogrammetry. The "Fontana Maggiore" in Perugia (PG, Italy), one of the main monuments in the city's historic centre, was reconstructed. Photogrammetry has made it possible, through appropriately taken photographs, to reconstruct the fountain in a highly reliable manner without manually modelling its polygonal shapes. We preferred relying solely on the results obtained by neural networks and algorithms that analyse the images produced by the three-dimensional model. Photogrammetry is a technique that allows very high-quality results that could hardly be achieved by manually producing the models and textures. As a limitation, this technology has a high number of vertices required for faithful reconstruction, limiting the virtual visualisation of the artefact with low performing hardware.

The second use case presented follows the manual modelling approach and concerns the reconstruction of the Republic square in Foligno (PG, Italy) and the surrounding areas where the buildings and architectural works were reproduced.

Starting modeling from Google maps or other images
Reproducing natural objects can be challenging. In particular, it is not trivial to reproduce accurately scaled shapes and dimensions within computer graphics software. Specific techniques can help the developer maintain a high fidelity of the shapes realised. The technique we recommend is the use of reference images. First of all, a photograph must be taken of the object to be modelled, acquiring the image perpendicular to the object to be reproduced. Let us suppose that we want to acquire the frontal image of a building. It is sometimes difficult to obtain images of this type when one wants to reproduce buildings built in an urban context, as there may be no suitable points for taking the images. For this reason, we recommend using a drone, on which a remote-controlled camera must be installed. After having had the drone positioned perpendicularly to the façade of a building, the shot can then be taken. Particular attention must be paid to the focal length of the camera used, as focal lengths too short, e.g. less than 50 mm, may cause perspective distortion that is difficult to correct. In this case, it could be helpful to have the drone move within an imaginary grid positioned along the façade of the building, taking photographs after moving the drone by a predefined amount of metres, an example of which is shown in Figure 2.1. This operation will produce a collage of photographs, which, when joined together, will provide a faithful reproduction of the building façade without the problem of perspective distortion.

Procedural textures
The realisation of realistic three-dimensional objects involves, among other things, the creation of textures, i.e., images that are applied to the object making it easily identifiable with real-world objects. A properly realised texture can simulate the materials that make up the objects in the three-dimensional scene. There are various techniques for creating realistic textures. The first involves manual creation using photo editing software (such as Gimp[3]), while the second involves the mathematical definition of the structure that the resulting texture should have. Textures created using the latter technique are called Procedural Textures. They are generated mathematically to simulate complex materials such as marble, wood and granite. Blender allows procedural textures to be created via a block graphics programming environment, an example of which is shown in Figure 2.2.

The various blocks we can see in Figure 2.2 are abstractions of mathe-

[3] https://www.gimp.org/

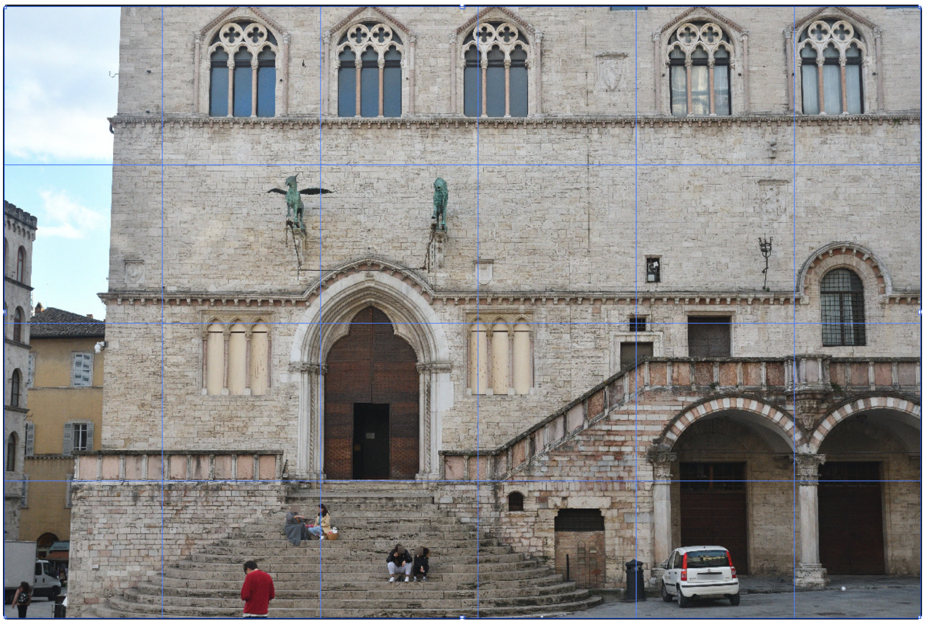


Figure 2.1: Sample virtual grid applied to a building.

matical algorithms that perform operations on an initially empty texture. When these blocks are interconnected, the output of the previous block is given as input to the next block. Thanks to the concatenation of these blocks, extremely complex textures can be obtained, such as the one shown in Figure 2.3, which is the result of the procedural texture used to create the façade of a building in Perugia's central square next to the "Fontana Maggiore".

Another advantage of procedural textures is that they are created and managed by the software itself, eliminating the requirement for file system references to images. These textures likewise adjust to the object they're applied to and don't have any visible discontinuities. It should also be noted that if one of these textures is applied to an object, such as a building, a change in the size of the building's facade will not have an adverse effect on the texture's aesthetics because the software will handle this fully automatically and the texture will be correctly placed on the object's facade, generating the missing parts. A procedural texture has no fixed resolution and adapts to the objects to which it is applied dynamically. Finally, it should be noted that creating a procedural texture is a time-consuming process that will use the CPU for several minutes.

Creating relevant details

The manual modelling of complex polygonal details may not be easy, and

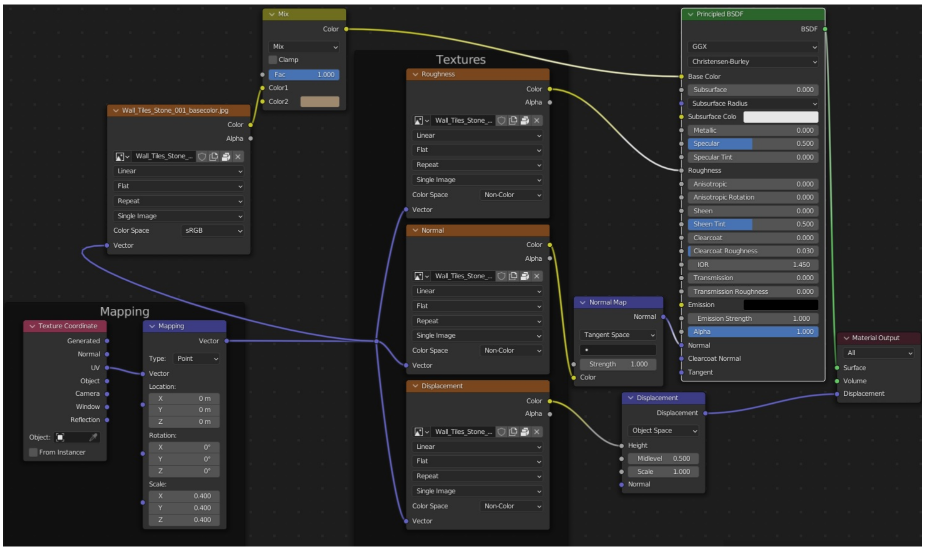


Figure 2.2: Visual program related to a procedural texture of the wall.

the final result may not be satisfactory. In this case, we recommend using the technique of photogrammetry, which allows a very high level of realism to be obtained by delegating the realisation of the most complex parts of 3D models to artificial intelligence techniques. To set up the photogrammetry software, one has to create an image dataset of the object to be reproduced. The dataset must consist of a sufficiently large amount of images, as all parts and facets of the object must be photographed. Most of today's cameras have an automatic mode, capable of setting photographic parameters that are most often adequate.

However, knowing how to set the correct parameters and juggle the use of manual mode will make a significant difference in terms of the quality and efficiency of the photogrammetric survey. It would be good to ensure that all photographs in the dataset are taken at the same focal length and with as similar an illumination level as possible between the various images.

In photogrammetry, similarly to human vision, if an object is captured in at least two images taken from different viewpoints, the different positions of the object in the images allow stereoscopic views to be obtained and three-dimensional information to be derived from the overlapping areas in the images. To obtain a very detailed three-dimensional model, a large amount of overlap between the different photographs is necessary; in general, it would be optimal that between one photograph and the next, approximately 70 per cent of the details captured in the previous photograph also appear in the following photograph. Furthermore, the number of photographs required

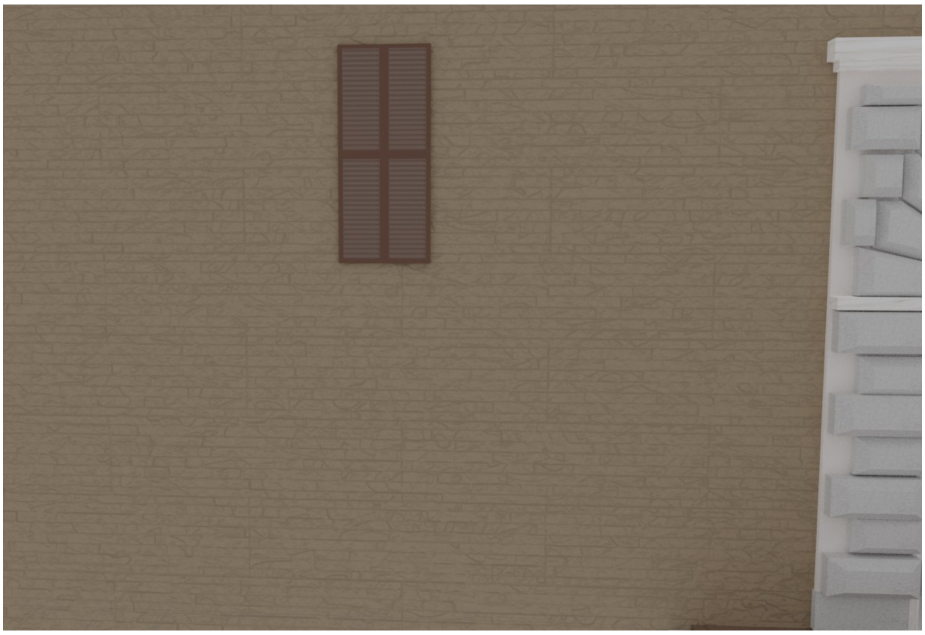


Figure 2.3: The procedural texture output simulating the real wall.

for the three-dimensional reconstruction of an object is directly proportional to its size; the larger the size of the object, the greater the number of photographs required for its reconstruction.

Digitisation of cultural heritage artifacts

In this section, we describe the methodologies that can be adopted to realise virtual worlds of monuments and works of art for the remote enjoyment of the objects of interest. To this end, we will first analyse the salient phases of the work that led to the virtual representation of the areas of the two Umbrian towns considered.[4]

The methodologies indicated can be reused for virtual representation and the creation of virtual journeys in other scenarios and contexts.

The Piazza IV Novembre

The "Piazza IV Novembre" is located in the centre of the city of Perugia and features several historical buildings such as the "Palazzo dei Priori" and the "Fontana Maggiore".

[4] Umbria is a region in central Italy untouched by the sea, rich in green areas and monuments that bear witness to the wealth and splendour experienced under the papal empire in the twelfth and thirteenth centuries.

The "Fontana Maggiore"

The "Fontana Maggiore" in Perugia has three levels, is polygonal in shape and was built in the thirteenth century using an alternating pink and white marble. The lowest level is set on a flight of steps from which rises a twenty-five-sided basin. Each face of the twenty-five sides forms a diptych, i.e. two images joined by a central link, adorned with sculpted reliefs with small, slender columns. The twenty-five sides describe the 12 months of the year, each accompanied by the zodiac symbol. Each month is associated with moments of daily life and agricultural work. Thus, each bas-relief on each side is different from the others. The photogrammetric survey of the fountain required photographs taken at different heights rotating at each iteration around the fountain to view the fountain's surroundings completely. An example is shown in Figure 2.4.

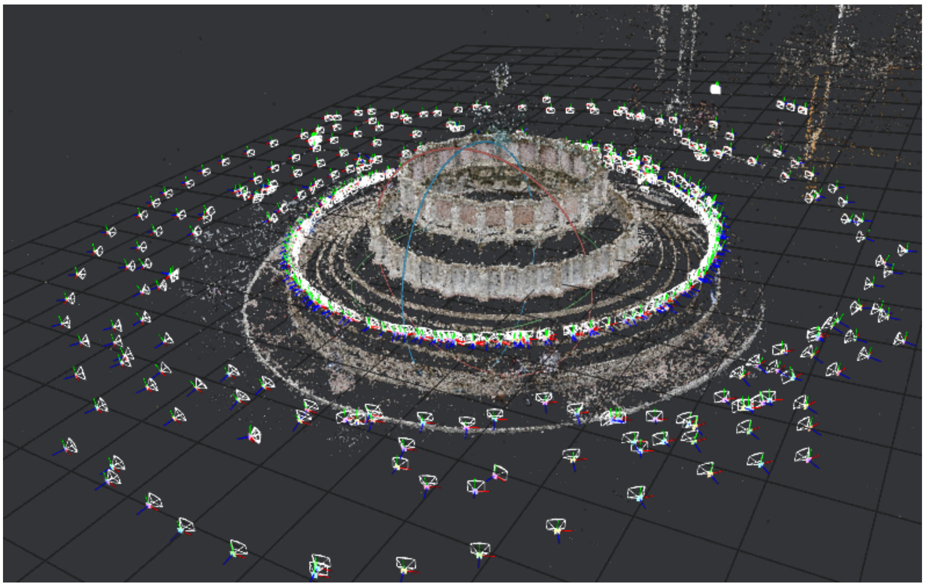


Figure 2.4: Images taken around the "Fontana Maggiore".

A total of 564 photographs were collected. The photographs are all taken with the same parameters: ISO sensitivity is set to 500, the focal length is 24mm, the exposure time is 1/320 sec, and the aperture is f/9. Since the scenery in which the photographs are taken is open to the public, care must be taken to avoid people appearing in the shots. Shadows of animals such as birds, dogs or cats should also be avoided as much as possible.

Palazzo dei Priori

Next, photographs were taken at the "Palazzo dei Priori", located opposite the fountain. The palace is characterised by a staircase that connects the main door with the floor of the square. There are several arches supported by columns in the lower part of the palace, while in the upper part, one can observe several windows arranged in the highest part of the façade. There are also aesthetic embellishments along the walls, signs of the wear and tear of time, and statues. The dataset, in this case, consists of 626 photographs and the parameters used are the same as those used for the fountain dataset. The point cloud of the building is shown in Figure 2.5.

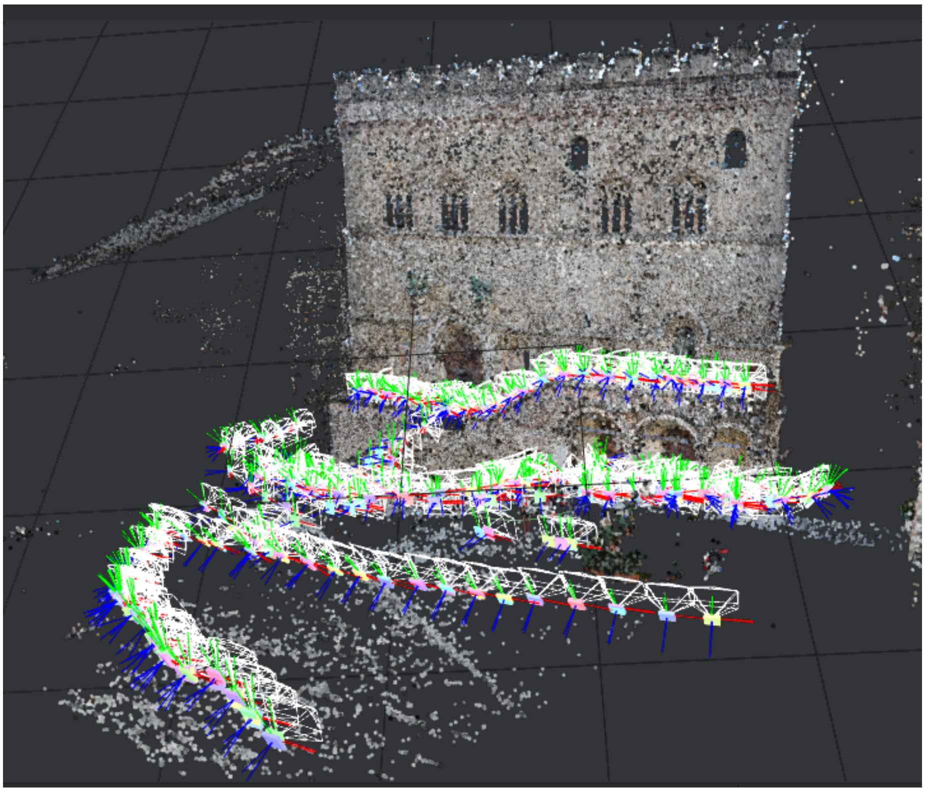


Figure 2.5: Images taken for reproducing the "Palazzo dei Priori".

As can be seen, the photographs were taken at different heights, moving from left to right and trying to capture as much detail as possible of the building.

Photogrammetry

The dataset is then processed with the Meshroom photogrammetry software [5]. Meshroom is a software released under an open source licence, which allows the reconstruction of three-dimensional models from photographic datasets. Once the dataset has been imported into Meshroom, it is necessary to define

[5] https://github.com/alicevision/meshroom

the pipeline, i.e. the set of operations that the programme must perform to generate the object's three-dimensional model. The pipeline used is as follows: *Camera Init*, *Feature Extraction*, *Image Matching*, *Feature Matching*, *Structure from Motion*, *Prepare Dense Scene*, *Depth Map*, *Depth Map Filter*, *Meshing*, *Meshing filter*, *Texturing*. The fountain model that was produced consists of approximately 2.9 million vertices and approximately 5.9 million faces. The model of the "Palazzo dei Priori" consists of approximately 2.1 million vertices and approximately 4.3 million faces. The resulting models are then imported into Blender and placed within a scenario. The square floor is then created using a procedural texture, and several adjacent buildings are modelled to give a sense of immersiveness to the scene. Figure 2.6 shows a view of the "Piazza IV Novembre" rendered within the virtual scenario, and the beauty of the final result can be appreciated.

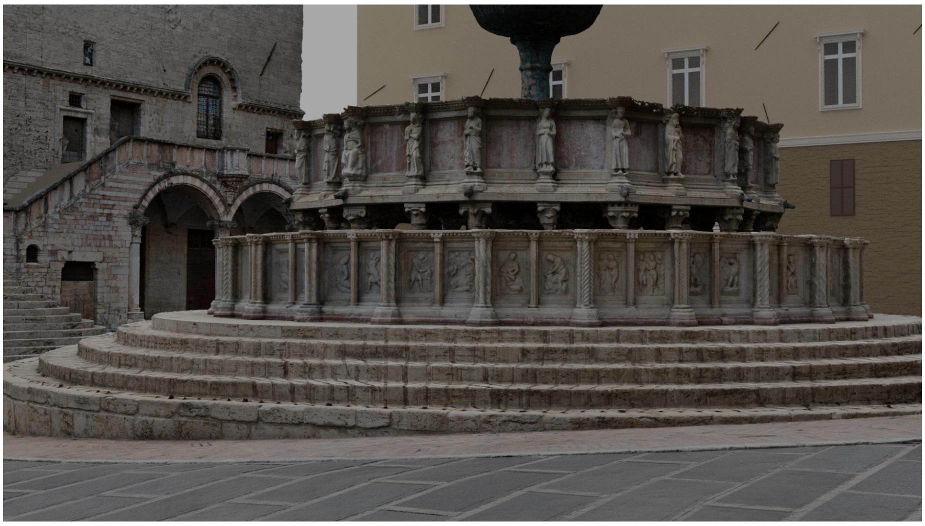


Figure 2.6: Final result of the virtual representation of "Piazza IV Novembre", Perugia (PG, Italy).

The Republic square, Foligno

The Republic square in Foligno was reconstructed using the technique of manual 3D modelling with Blender. There are various buildings within the square, among them the "Palazzo Trinci", the Town Hall, the Cathedral of San Feliciano and the "Palazzo delle Canoniche". Similarly to what was done to reconstruct "Piazza IV Novembre" in Perugia, preliminary work aimed at collecting images was also crucial in this case. Images are necessary for reconstructing rooms and buildings because they provide details and proportions and are used as a trace to be followed during modelling.

The "Palazzo Trinci"

On the eastern side of the square is the "Palazzo Trinci", formerly the residence of the Foligno seigniory and now home to the City Museum. Characteristic elements of "Palazzo Trinci" are the six large Corinthian-style columns resting on the central loggia of the palace façade, the windows are characterised by two distinct architectural forms, the 'curbs' under each row of windows, the row of small parallelepipeds positioned below the roof and the bricks decorate the lower part of the building. Figure 2.7 shows the virtually reconstructed "Palazzo Trinci". The picture was taken with a focal length of 35mm.

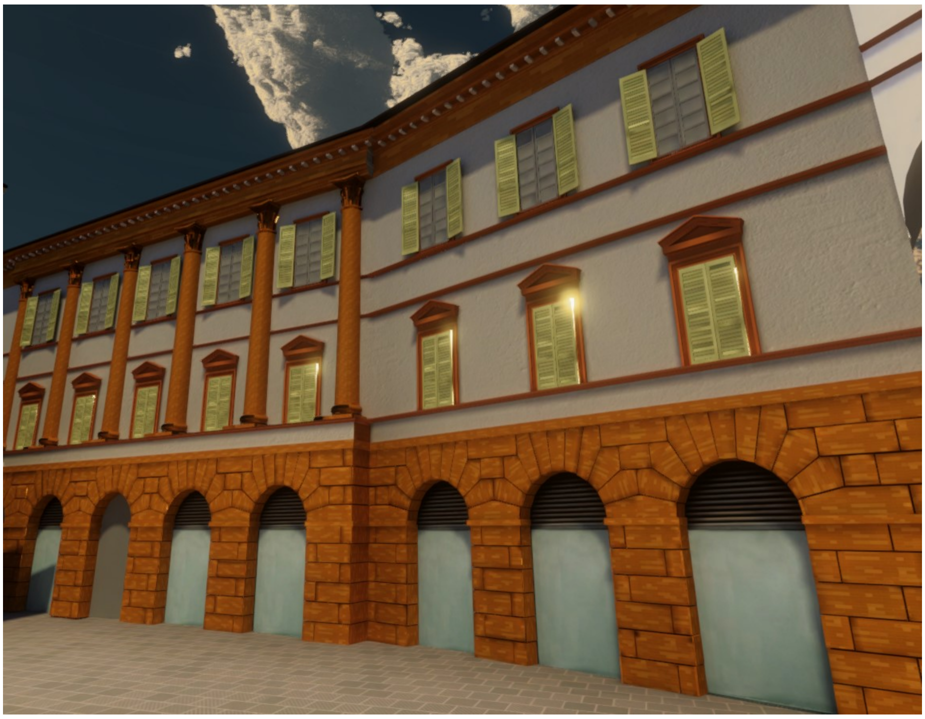


Figure 2.7: Virtual representation of "Palazzo Trinci" located in Republic square, Foligno (PG, Italy).

The Town Hall

The Town Hall is set on a medieval tower called the "Pucciarotto", which has the function of a bell tower. The elements that characterize this building are described. The façade is punctuated by six Ionic-style columns resting on a ledge from which five round arches open. The windows of four different shapes have a frame very similar to that of the "Palazzo Trinci", which is why the same model has been reused. A characteristic element of the building is undoubtedly the tower, built in the thirteenth century, which collapsed following the 1997 earthquake, and then restored in 2007. The structure of the turret ends with an umbrella-shaped dome. The virtually reconstructed Town Hall is shown in Figure 2.8. The perspective distortion seen in the image is due to the focal length that was used to capture the photograph. The photo was taken with a focal length of 18mm and this was necessary to fit the whole building into one photographic shot.

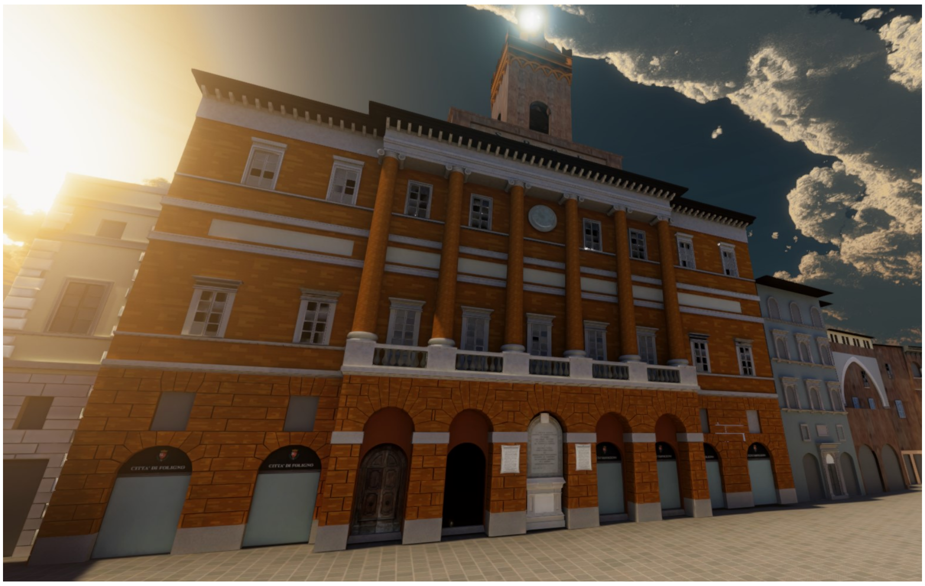


Figure 2.8: Virtual representation of the Town Hall located in Piazza della Repubblica, Foligno (PG, Italy).

The Cathedral of San Feliciano

The Cathedral of San Feliciano, also called the Duomo, is the most influential building in Republic square, both from an artistic and cultural point of view. Two facades face the square, the side one and the main one, which more precisely overlooks Largo Carducci. The main facade of the cathedral, restored in 1904, has three doors in the lower part, which were completely rebuilt during the restoration work. On the axis of the side portals, there are two mullioned windows surmounted by a rhombus and a circle and a loggia articulated into eight small arches. In the second order, the facade has a central rose window. At the same time, the third level, introduced with the restoration of the sixteenth century, contains a mosaic depicting *the Redeemer enthroned, the saints Feliciano and Messalina and the Pope Leo XIII genuflected.* The minor façade of the Duomo presents, in the lower part, the multiple ring gate adorned with a series of decorative motifs. There is a large loggia in the upper order with six openings above which the two small side rosettes and the larger central one stand out. The main facade has a single rose window consisting of two rows of columns arranged radially starting from the central core and connected by small arches. The side façade, on the other hand, is characterized by three rose windows, one larger and placed centrally, and two smaller lateral ones, formed by a single row of columns. Three mullioned windows characterize the minor facade of the cathedral, created starting from the outer frame and then moving on to the modelling of the internal details. On both sides of the church, numerous bas-reliefs and sculptures present a significant level of detail, many of which were not made through modelling but simulated through procedural textures. The Cathedral also features a bell tower and a large dome. The virtually reconstructed Cathedral of San Feliciano is shown in the Figure 2.9. The picture was taken with a focal length of 35mm.

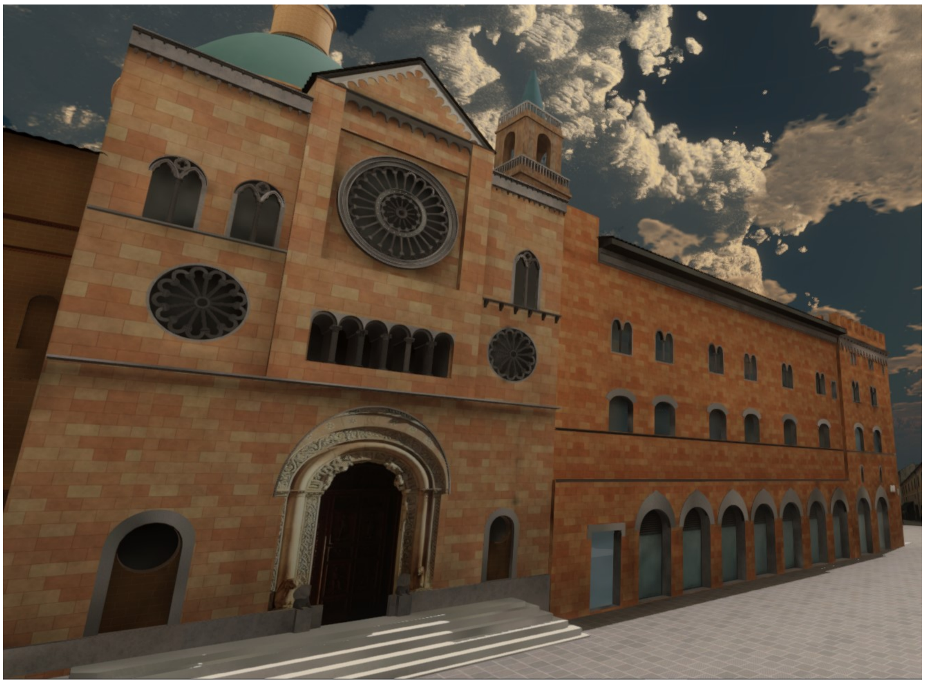


Figure 2.9: Virtual representation of the Cathedral of San Feliciano (left side) and "Palazzo delle Canoniche" (right side) located in Piazza della Repubblica, Foligno (PG, Italy).

The "Palazzo delle Canoniche"

The "Palazzo delle Canoniche" is built between the nave and the left arm of the main facade of the cathedral. Inside we can find the Capitular and Diocesan Museum of Foligno. The aesthetic aspects that characterize this building are the arches that run along the base, the windows with a very elongated shape, and the roof, with various parallelepiped-shaped elements that run along its perimeter. The "Palazzo delle Canoniche", virtually reconstructed, is shown in the Figure 2.10. The picture was taken with a focal length of 30mm.

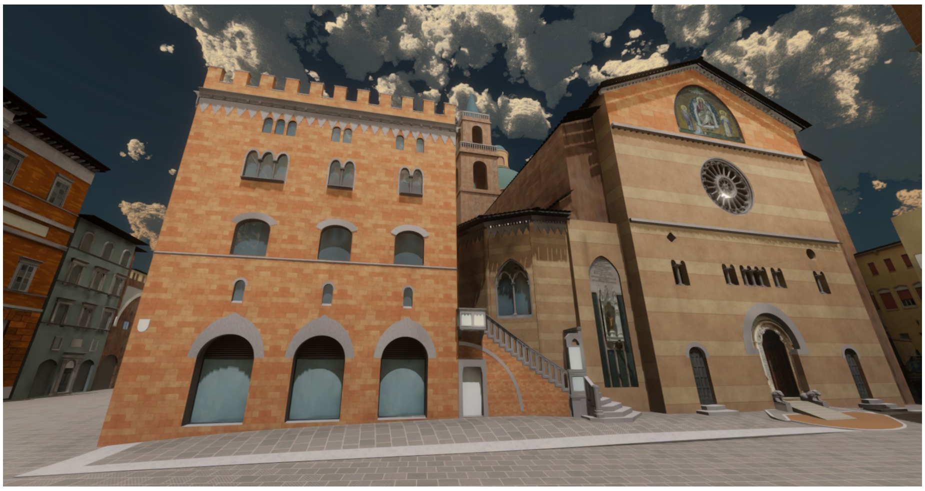


Figure 2.10: Virtual representation of "Palazzo delle Canoniche" (left side) and the Cathedral of San Feliciano (right side), Piazza della Repubblica, Foligno (PG, Italy).

2.3 Serious games to improve tele-rehabilitation

Virtual reality is a powerful tool for motor telerehabilitation due to neurological lesions [124, 125]; therefore, games for non-recreational purposes, serious games, are powerful rehabilitation tools that have given relevant results [126, 127]. Cognitive rehabilitation is a subset of rehabilitative treatments that is dedicated to all patients who require ongoing cognitive function training due to injury (Traumatic Brain Injury, Stroke, Cerebral Palsy, etc.) or pathology (Alzheimer's disease, Multiple Sclerosis, etc.) affecting the Central Nervous System. Stroke is the second leading cause of death and disability in the world. It is a disease whose timely access to treatment is crucial to allow important improvements to recover motor and cognitive skills [128]. With the advent of the pandemic that has made access to rehabilitation centres more problematic for patients with cognitive rehabilitation needs (such as those who have suffered a stroke), the importance of being able to exercise these patients safely in their own homes has emerged strongly [129]. There is no doubt that at such a problematic time, poor digital organisation and not being used to organising processes with principles of digital sustainability has been a major detriment to the quality of life of citizens in many countries around the world.

Many studies show that immediate action and appropriate, specific rehabilitation can guarantee satisfactory results [130]. Appropriate therapy is based on key factors to be taken into account such as frequency, intensity and specificity of the exercises.

Our ultimate objective is to create a modality for developing open-source digital goods that may provide access to a virtual environment at any time. That allows patients who have experienced cognitive limitations to exercise and recover all or part of these skills in the least amount of time.

In view of the spread of IoT devices capable of easily monitoring various vital parameters, we propose with our system a low-cost and very efficient solution that can provide the doctor not only with quantitative data on the exercises performed (number and type of exercises, time spent, results obtained) but also an overview of vital parameters, so as to observe any states of agitation or excessive effort in completing the exercise. This information is provided to the doctor through a CSV file. That contains the measured values of the patient's parameters collected from the start to the end of the exercise.

The system architecture

The system we developed[6] requires the patient to use a mobile or fixed device to access a Web App containing a set of exercises planned by the doctor according to the type and severity of the patient's cognitive deficit. The outstanding feature of the proposed system is to identify a set of open-access technologies[7] using which developers can produce serious games with a relatively low investment of time and energy.

In fact, the exercises are characterised by being placed in a very poor virtual context from the realistic rendering of the virtual world. The setting of the exercises does not require a high level of realism and graphic detail, as the user must be focused on the exercise and not risk distracting his or her attention from the assigned task. Instead, it is vital that whoever designs the virtual world does so while leaving the doctor free to include as many

[6] The system is available at the URL `https://ogervasi.unipg.it/seriousGames/index.html`, including the source code.

[7] We used Blender, which is Open Source, and recommend it for creating objects and Unity, which can be used freely even if the app induces revenues up to 100,000 USD.

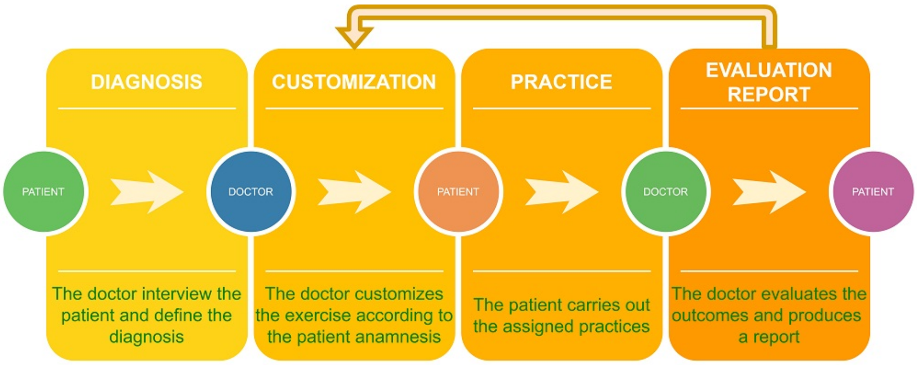

DIAGNOSIS
CUSTOMIZATION
PRACTICE
EVALUATION REPORT
PATIENT
DOCTOR
PATIENT
DOCTOR
PATIENT
The doctor interview the patient and define the diagnosis
The doctor customizes the exercise according to the patient anamnesis
The patient carries out the assigned practices
The doctor evaluates the outcomes and produces a report

Figure 2.11: Diagram of the required steps that a doctor can follow

elements as possible relating to the patient's life and history (images and objects related to his or her daily life before the trauma occurred).

The use of immersive devices per se is not indispensable, but their use allows the patient to have a much more intense experience during rehabilitation practice. Moreover, the availability of high-end hardware allows the patient to perform much more natural movements, facilitating the success of the rehabilitation practice. When designing the virtual environment surrounding the rehabilitation exercise, the developer must take great care to set up areas of the scene where the doctor can dynamically insert either images or virtual models of objects that can stimulate the patient's memory and attention. The importance of contextualising the exercise in terms of images or objects familiar to the patient is well documented in the literature[131]. It is vital to enable the doctor to easily set up the exercise, including pictures of people or landscapes, objects or other things that evoke memories in the patient to make the exercise even more effective and engaging. In Figure 2.11 the diagram of the steps the doctor can follow is shown. The last three blocks can be iterated until the patient rehabilitation is completed.

Use of IoT devices and Edge Computing

The use of IoT enables the doctor to monitor several vital parameters of the patient in order to be able to track the state of anxiety and performance stress and provide further helpful information. The large quantity of data collected may be analysed by computational resources located close to the patient to minimise the data transmitted to the Web App server, following the paradigms of Edge Computing [132]. We used the Fitbit Sense smartwatch device to keep some of the patient's vital parameters under surveillance, like heart rates, tissue oxygenation SpO2, respiratory rate, skin temperature, and aFib (atrial fibrillation) signals. The monitoring we carry out takes place in

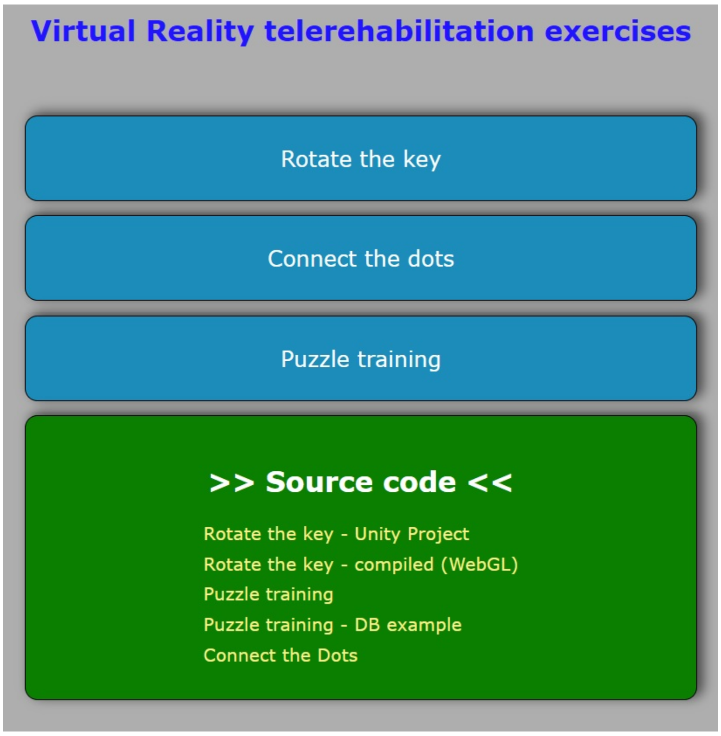

Virtual Reality telerehabilitation exercises
Rotate the key
Connect the dots
Puzzle training
>> Source code <<
Rotate the key - Unity Project
Rotate the key - compiled (WebGL)
Puzzle training
Puzzle training - DB example
Connect the Dots

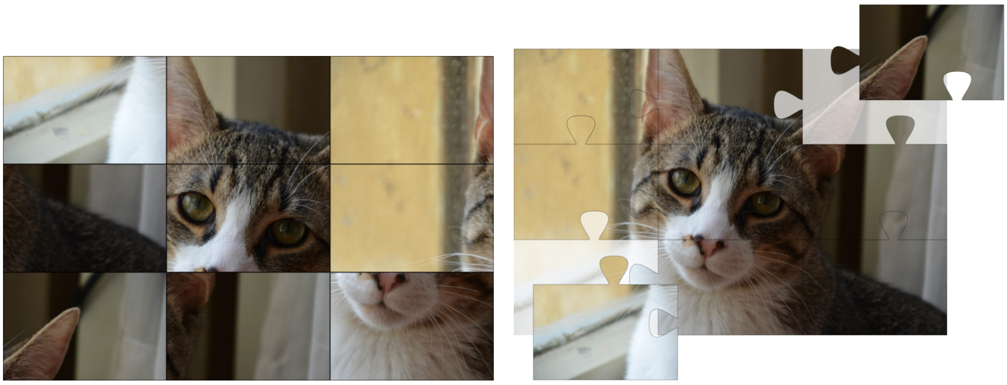
Figure 2.12: Doctor's Graphical User Interface for the Main Menu.

the few minutes before exercise (value set by the doctor), during exercise, and a few minutes after exercise. In this way, we can obtain data to compare the trend of the parameters in a state of calm and concentration state.

The Web App

The Web App consists of the front-end environment accessed by the patient, which contains the daily exercise list, and the environment reserved for doctors, where the exercise program for the specific patient is set up. Figure 2.12 shows the doctor's graphical user interface to choose and set one of the several types of exercises. The Linux server that delivers the Web App is based on a Debian distribution within which a Docker container orchestration service runs[8]. One of the features we focused on most was the compatibility of our applications with the most significant number of devices on the market. Indeed we tried to create a software environment that did not require exceptionally high computational power.

[8] https://www.docker.com/



(**a**) This type of puzzle has the tiles all with the same shape

(**b**) This type of puzzle has the tiles that must fit together with each other

Figure 2.13: The figure represents two types of puzzles, with an increasing complexity from (a) to (b).

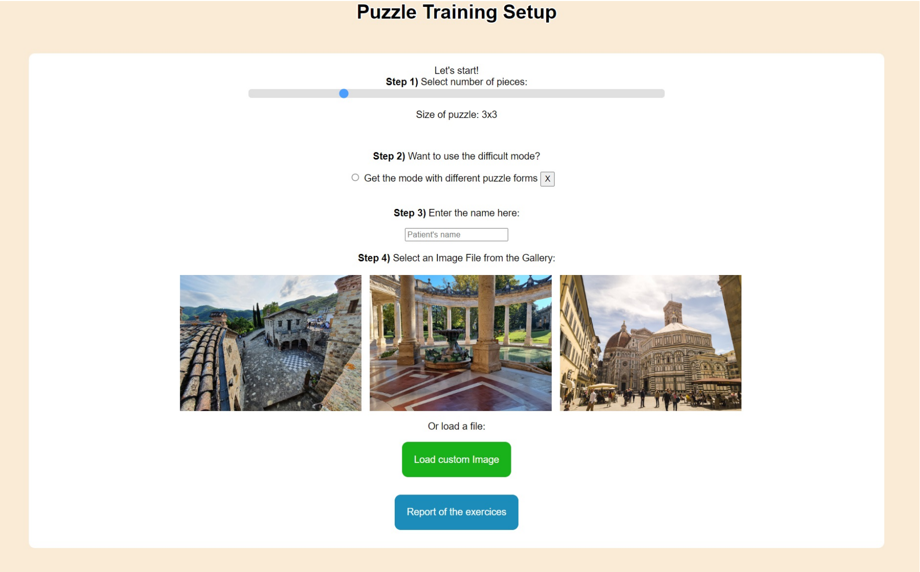
Implementing the case studies

The innovative aspects of this work are related to the definition of a methodology that allows the production release of tele-rehabilitation exercises in a relatively simple and low-cost manner. In order to clarify the implementation aspects and the various problems, we have developed three use cases with different characteristics. The first concerns the solution of a puzzle, for which the doctor will determine the level of difficulty and the type of image according to the patient's characteristics. The second one consists of joining numbered dots to draw a figure, so the doctor can produce the exercise by customising it according to the patient's needs. The third is related to inserting the key into the lock and opening it: the doctor can personalise the virtual environment with objects or images that the patient loved.

Thanks to these practices, the patient can recover necessary skills and relieve himself from an inability to perform essential daily activity routinely functions.

Solving a puzzle

In this type of exercise, the patient ought to be able to determine the right order and position of tiles obtained dividing into pieces a photo. This practice is important for stimulating patient's visual and motor coordination. The patient must also coordinate hand and finger movements to select the puzzle tiles and position them correctly. In order to make the exercise more enjoyable, the doctor selects images that have an emotional connection to



Puzzle Training Setup

Let's start!
Step 1) Select number of pieces:

Size of puzzle: 3x3

Step 2) Want to use the difficult mode?

○ Get the mode with different puzzle forms X

Step 3) Enter the name here:

Patient's name

Step 4) Select an Image File from the Gallery:

Or load a file:

Figure 2.14: Graphical Window for the puzzle page

the patient's life history, e.g. photos of the family, pets with which there is a strong emotional bond or pictures related to the childhood.

The application has two separate interfaces: one for the doctor and the other for the patient. The doctor has the task of setting up the exercise, selecting a suitable image and the number of tiles that will make up the puzzle. After that, the software breaks down the image into the proper numbers of tiles and then randomly shuffles them on the screen. Next, the server generates a public URL, which can be sent to the patient. When the patient opens the URL, he or she will immediately find the exercise ready to be performed, with a very short loading time, as the software is entirely realised in HTML5, JavaScript and CSS. The additional components that the user's browser will have to download are a JavaScript script and the JPEG image.

The doctor can also choose the shape of the puzzle tiles. The choice can be made between rectangular tiles that are all the same shape, as shown in Figure 2.13a, or interlocking tiles, which must be fitted together, as shown in Figure 2.13b. During the exercise, the time taken is timed and displayed.

Figure 2.14 shows the doctor's graphical user interface to manage the various parameters for the puzzle training exercise.

The timer turns on automatically as soon as the puzzle is started. After a congratulatory message, the time taken will be saved in the database and displayed when the puzzle is finished. In the doctor's interface, a report will show the exercise time for each image used, providing the opportunity to analyse the patient's progress and monitor the progress over time.

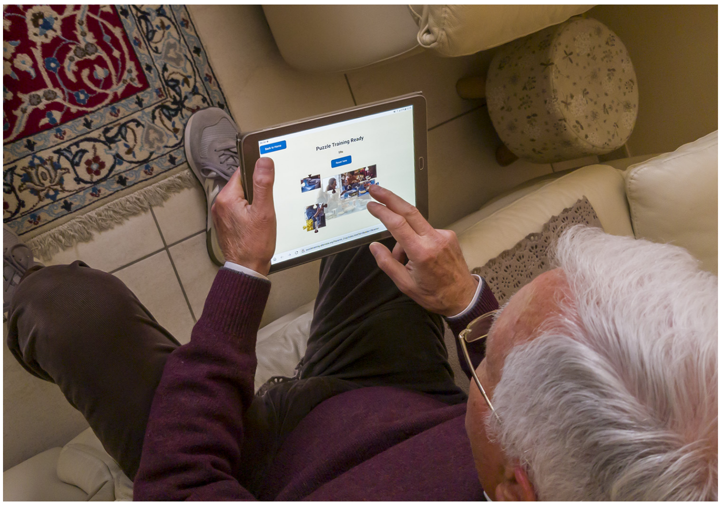


Figure 2.15: Patient during the puzzle resolution

Figure 2.15 shows a patient performing the puzzle exercise.

Connect the dots

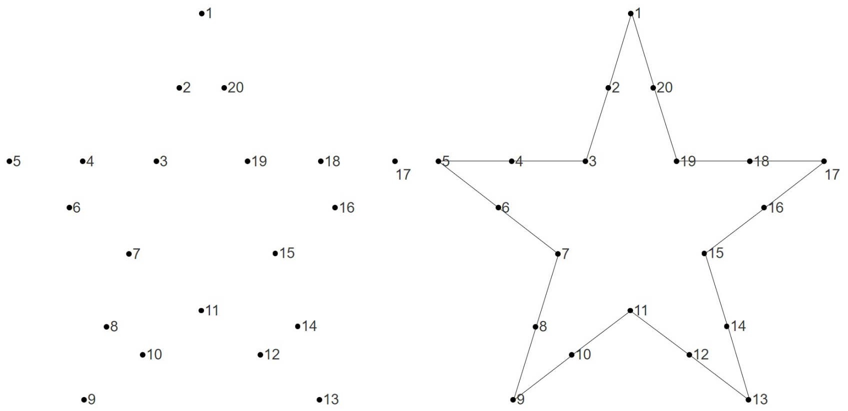
The second exercise is called *Connect the dots* and has a slightly higher degree of difficulty than the previous one. To complete the exercise, one must connect the dots identified by integer numbers, starting with the number 1 and continuing in ascending order up to the highest number shown. This exercise aims to stimulate the patient's neurological skills. In fact, he/she must identify the numbers and the grapheme, try to deduce the final figure, and then use visual and motor coordination to move the finger across the screen until the exercise is completed. There are also two graphical interfaces in this exercise; the doctor's interface and the patient's interface relating to the execution of the exercise.

The doctor's interface allows selecting an image on which to create the series of dots to be joined to reproduce the main lines.

Images suitable for this type of exercise are predominantly black and white, with a low level of detail and graphic complexity. The shapes we recommend to be simple and easy understood, such as regular polygons, animal drawings or children's colouring figures.

The Figure 2.16a shows an image with dots to be joined. The Figure 2.16b shows the image with the dots connected.

Once the doctor has selected one of the predefined images or has uploaded an image of his preference, a JavaScript script will be executed. After analysing the image, it will extract features using the OpenCV graphics



(**a**) The starting phase of the exercise with the dots to be connected

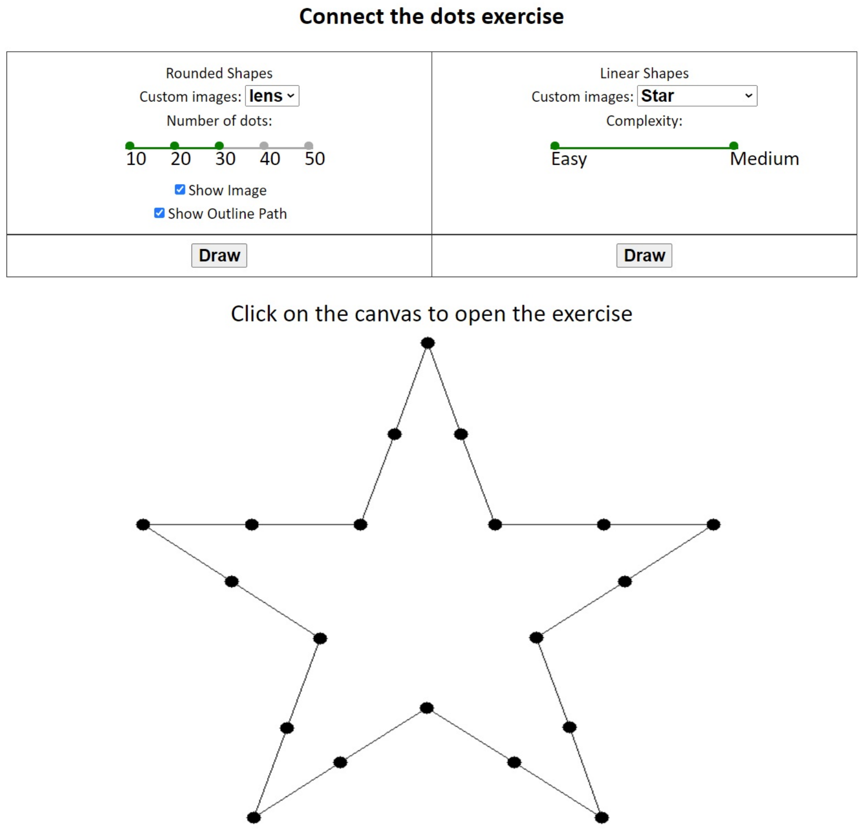
(**b**) The exercise has been completed, with the dots connected

Figure 2.16: Sample figure representing the "Connect the dots" exercise.

library. In fact, comma filters are applied to bring out the contours and the most defined parts of the image. Then the result obtained is processed in such a way as to eliminate the segments and replace them with a sequence of dots to which labels characterised by integer numbers are associated. The doctor will then see the result obtained and decide whether the complexity level is acceptable. Suppose the image obtained is evaluated as too simple or too complicated. In that case, the doctor can adjust the number of dots shown on the exercise using a stepped slider. These adjustments allow calibrating the exercise finely. Following each adjustment, a function associated with the `onChange` event of the slider is executed. The function executes the script again, which analyses the image and determines its representation in points and segments, adjusting the complexity according to the doctor's preference. Once the doctor has finished setting up the exercise, the system will provide a URL that can be given to the patient.

Figure 2.17 shows the menu to choose shapes and their parameters in the "Connect the Dots" exercise.

Once the patient has received the URL and clicked on it, the exercise page, which has already been set up correctly, will open. The exercise page has a timer, a picture with the dots to be joined, and a button to stop the exercise if the patient does not want to finish it. The timer starts automatically as soon as the patient connects the first two dots. The web page has a very short loading time as it consists of a JavaScript code, which manages the timer and the detection of finger touch input, and a canvas on which there is



Connect the dots exercise

Rounded Shapes	Linear Shapes
Custom images: lens Number of dots: 10 20 30 40 50 ☑ Show Image ☑ Show Outline Path	Custom images: Star Complexity: Easy Medium

Click on the canvas to open the exercise

Figure 2.17: Doctor's Graphical User Interface for "Connect the dots" page.

a monochrome JPEG image. The exercise can be stopped in two cases: firstly, the patient completes the joining of all the dots present in the image; next case, the patient prematurely interrupted the exercise by pressing the exit button presented below the image. The interruption of the exercise involves sending the result achieved by the patient encoded via a JSON list to the central server, which is stored in the database. The doctor can then see the level reached by the patient, how many dots have been connected correctly, and the time taken. If the exercise has been performed several times, the records of previous attempts are also reported to show any progress.

Figure 2.18 shows a patient performing the exercise of connecting the dots on his tablet.

Key turning in the lock

The exercise that is now presented was realised by exploiting immersive virtual reality techniques. This exercise involves the patient wearing a VR visor and using hand controllers capable of transforming the wrist's movement into actions that are performed in the simulated environment. The patient

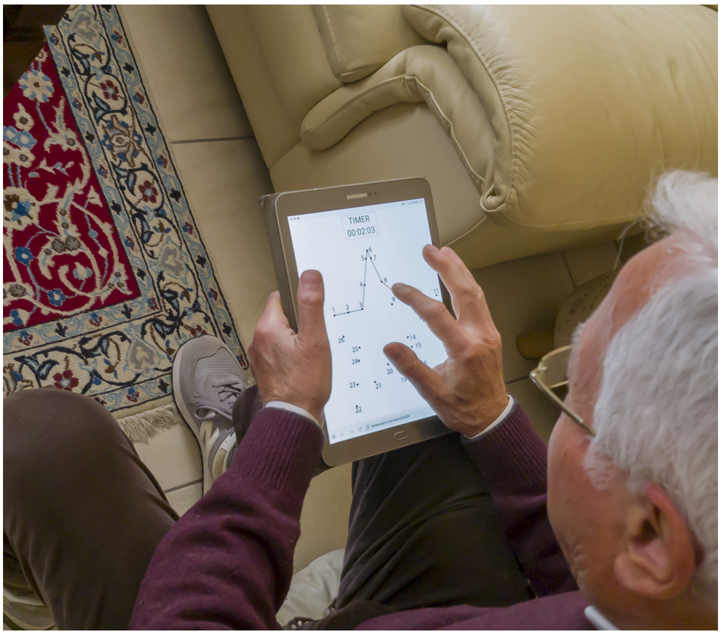
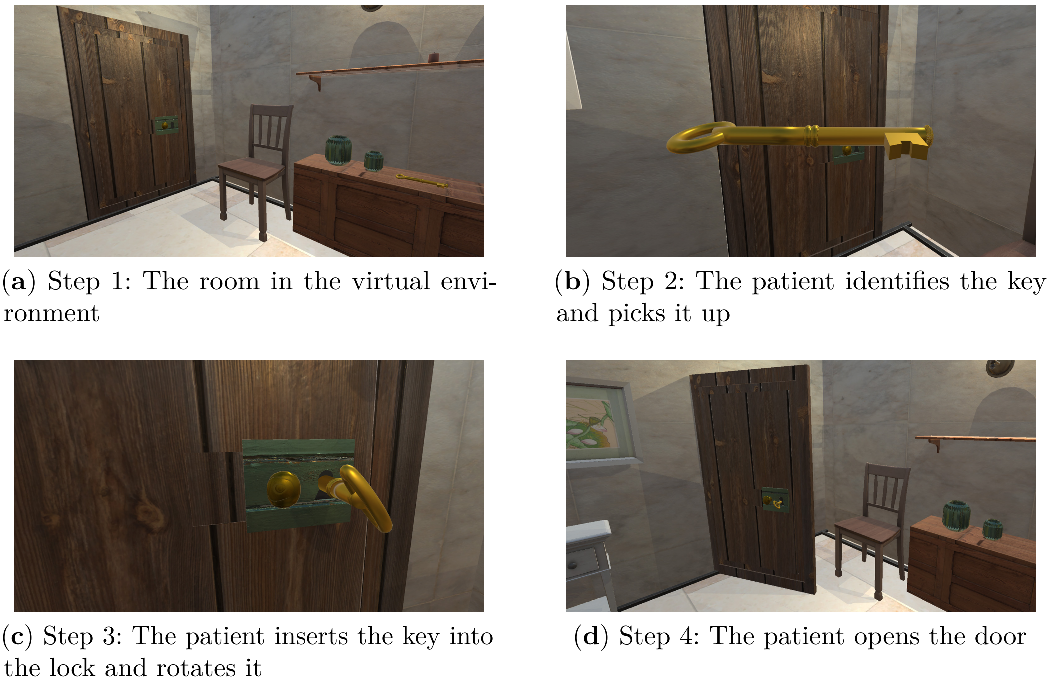


Figure 2.18: Patient solving the "*connect the dots*" exercise



(**a**) Step 1: The room in the virtual environment

(**b**) Step 2: The patient identifies the key and picks it up

(**c**) Step 3: The patient inserts the key into the lock and rotates it

(**d**) Step 4: The patient opens the door

Figure 2.19: Main steps for performing the exercise "*key turning in the lock*".

will be in a room, have to pick up a key, locate the door and its lock, and finally, after approaching it, insert the key into the lock and rotate it. The door will open at the end of the rotation, and the exercise will be finished. Using Unity[9] and the Open Source software Blender[10], a domestic scenario was created, with a room and various objects in it. All objects in the scenario were created using Blender.

The objects were then arranged in an interactive virtual environment using Unity software. The software has been configured to exploit the potential offered by immersive visors, such as the HTC Vive. This exercise is designed for patients who have neuro-muscular problems and need rehabilitation to recover motor coordination of the hand. In Figure 2.19 can be observed the main step executed by a patient performing this exercise.

2.4 Rapid prototyping of telerehabilitation exercises

Prototyping is a tool that lets anyone to be able to create a so-called prototype, i.e. the basis used to make new products. Rapid prototyping is a faster prototyping technique, that allows to reduce the production costs, and it is best known as Rapid Application Development (RAD). This technique involves the use of CASE (Computer-Aided Software Engineering) tools and implies software development through graphical and visual interfaces and libraries.

In general, RAD approaches to software development privileges the adaptive process rather than planning. It is characterised by a better flexibility against vague specifications or not already closed-loop development, a general risk reduction against the rigidity of plan-driven waterfall design. On the other hand, rapid prototyping idea is often linked to agile production concepts[133, 134]; these gain importance in recent years as an immediate and easy answer to more and more sophisticated customer expectations.

The attention to bring services operated remotely has grown in the last years. The pressure on health related services prompted for simple and fast solutions for tele-rehabilitation in several fields. Those services were conceived as reliable and routinely services (not only pilot tests)[135], and are nowadays even more important to provide continuous and efficient health services.

Cognitive rehabilitation is a branch of the whole rehabilitative treatments,

[9] https://unity.com

[10] https://www.blender.org

and is devoted to all patients who need a continuous training of cognitive functions because of injury (Traumatic Brain Injury, Stroke, Cerebral Palsy, etc) or pathology (Alzheimer's disease, Multiple Sclerosis, etc), which are affecting the Central Nervous System (CNS).

In general terms, to tune the subsequent therapy phases, results from cognitive routinely exercises must be logged and tracked by the therapist. She/he can monitor the execution of all activities, which can be either based on paper and pencil or software tools.

Pilot cognitive tele-rehabilitation practices are reported in literature[136]: they report relevant advances for patients and their families, either objective results or perceived comfort, even if some barriers because of remote treatment modality have been identified as well.

Advanced Unity techniques

Unity gives us the opportunity to save our time especially in the development phase, because it features a RAD logic. It allows applications to be developed for a wide range of devices, introducing dynamic properties thanks to Assets and scripts. The term Asset refers to "objects of various kinds", i.e. animations, textures, models or even entire projects created by third parties or ourselves.

At the program start, a subset of objects gets randomly picked up from those earlier defined; for example: the book, the puppet, the camera, a wall clock, a lamp, a bottle, and so on. These objects can belong to two distinct classes: contextual and not contextual related with the scenario. Next, we have defined a series of points within the scenario, called spawn points. At the start the software let these objects appear in their own specific spawn point, extracted randomly among the available ones. An example of spawn points' arrangement is shown in Figure2.20.

In order to make the exercise easier, it is available a simplified software version that helps the patient to look for an object in a list, simply calling by its name. In order to make the exercise easier, it is available a simplified software version that helps the patient to look for an object in a list, simply calling by its name. To avoid patient can learn the solutions by heart, different patterns are presented every single time the exercise is started, with different objects in different positions.

During the development of the software we took great care of all performance aspects. Our goal was to obtain the shortest application loading time to give to the users the possibility to immediately access the exercise that the therapist wants to manage. To do this we balanced various aspects, in particular we focused on how to obtain the best compromise among graphics quality, loading time and size in megabytes when the project is compiled

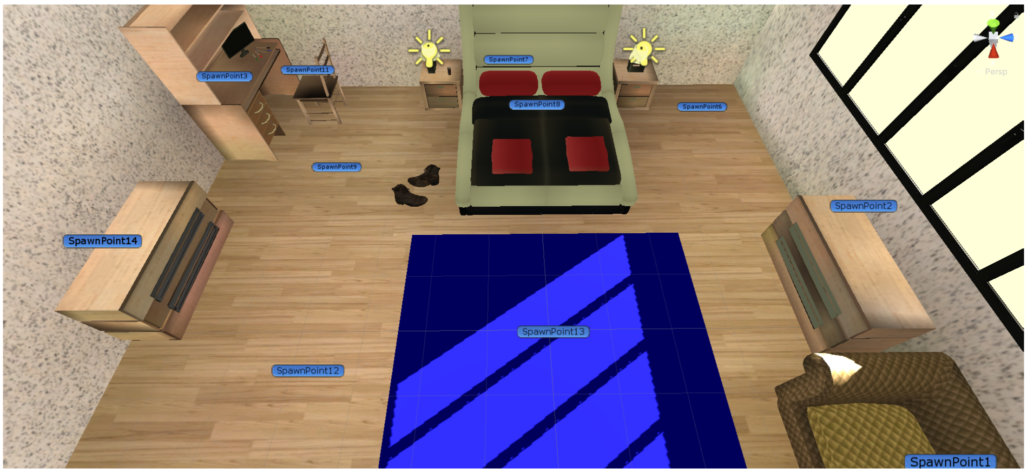


Figure 2.20: The spawn points

in WebGL. Patients who will use the software may have outdated devices and this aspect actually requires to generate software which will require a minimum amount of RAM and VRAM while running.

Generally, outdated devices are rarely updated by smartphone manufacturers for major releases of the operating system, so we need to take advantage of APIs that allow us to obtain the highest percentage of market share even in the smartphone sector.

We have also considered that users may have slow and unstable connections, so it is important that the program should require them to download as little data as possible over the Internet.

Graphics Settings

To obtain the best possible performance we implemented as much as possible technical optimisations on the Unity project. First of all we analysed the rendering techniques. The graphic settings related to the image quality that will be shown in the rendering phase on the device was manually calibrated by creating a custom profile in the Quality section of the Project Settings. The resolution of the textures has been set to "Full Res" thanks to the optimisations implemented with the techniques that will now be described, and there is no reason to further compress them. Anisotropic filtering and antialiasing filtering was disabled as well as the Soft Particles calculation engine, since they will not be used. We've also turned off the real-time reflections. The Texture Streaming feature has been enabled. This allows us to reduce the memory quality required for textures so that only small portions of mipmaps are loaded into memory. For this functionality, we have set a maximum amount of VRAM memory that Unity can use equal to

512MB and a maximum of 1024 simultaneous input/output requests. As for the shadows, we used the Distance Shadowmask calculation method and render only the "hard" shadows on the scene, thus ignoring the calculation of soft shadows.

This calculation method allows us to obtain a realistic and fairly faithful graphic environment. To get a little bit lighter project, we could have set the rendering in shadowmask mode, but quality loss would not have been justified by the gained performance. Indeed, the project is contained and the rendering of the room is efficiently calculated anyway. The shadow resolution was set to a medium quality level.

The Forward rendering path was set on the Graphics card, which guarantees compatibility with the greatest number of devices compared to deferred mode. We describe now how the lighting system was configured, since it is well known that this is one of the aspects that most affects the rendering performance of a three-dimensional environment.

To try to contain the required computational capabilities as much as possible, we have disabled the real-time rendering of the lights, preferring instead to calculate them at compile time. To do this we deselected the RGI (Realtime Global Illumination) and we activated the "Baked Global Illumination". This operation requires that the objects should be defined with the "Static" property. All the objects that are part of the setting was therefore set as "Static" and the light bulbs that illuminate the scene was set with "Baked" type of light. This lighting calculation methodology pre-calculates how the light will affect the objects and elements in the scene by saving the calculated information in the LightMap. In total we defined 4 lights in the scene: the sun, the light in the room, the lights coming from the left bedside table and the right bedside table. The lights in the room are all Point Light set with the Baked type lighting mode. The sun is mimicked via a directional light in Mixed calculation mode. This calculation mode allows to take advantage of the pre-calculation (baking) of the lighting that interacts with static objects but leaves the rendering engine the ability to correctly calculate the lighting of any moving objects.

The patient will find in the room several moving objects that once clicked will perform a short animation that provides the user with feedback on the correctness of the action performed. The number of Direct samples was set to 32 and the number of indirect samples to 128. Our tests showed these values are an excellent balanced set that allows to obtain a good graphic quality while consuming few resources. Furthermore, environmental occlusion was activated to further improve the visual rendering. The Lightmaps obtained therefore has a size of 9x32x32px and occupies a space of 144 KB of memory.

To minimise the space it occupies, we compressed it to obtain a size of 48.5KB. Also on the Player tab of the Project Settings panel we needed to set the Graphics API in order to use WebGL 1.0 instead of WebGL 2.0. This operation let us further reduce the size occupied by the compiled project while maintaining a comparable graphic quality.

Polygonal complexity and texture

Then we worked and thoroughly analysed how to optimise the objects that make up the scene. Each single object has been processed and analysed one by one, in order to minimise the number of polygons, finding the right balance between polygonal complexity and graphic quality. This is an operation that cannot in any way be defined a priori and strongly depends on the type of object. For example, the cushions that are placed above the bed can have a very high compression, while the polygonal model of a shoe due to its shape and structure requires more definition to be appreciated on the screen.

Object compression was done using two technologies: the first is through the Blender software, the second is provided directly by Unity which allows by the Mesh Inspector to obtain a copy of the object model that gets compressed and directly saved in the project file system. To do this we did set the Mesh Compression and the Optimise Mesh for each object by calibrating the parameters one by one. Furthermore, following all these intermediate steps, we recalculated the Normals for each object by generating the individual UV LightMaps that tell the graphic engine how the light should interact with the shape in the scene.

We also paid great attention to the textures we wanted to examine and study. First of all, textures were chosen so that they could create a contrast with the background and did not overlap the colors with each other in order to improve accessibility for users.

Compiler Optimisations

The last optimisation phase was dedicated to the final compression of the project and to the study of the techniques that allow to reduce as much as possible the waiting time for the loading and start-up phase of the project. Unity provides different ways to export a project to WebGL:

- Without any kind of compression, the project has a large memory occupation and requires a considerable amount of time and bandwidth to be downloaded. Because of this difficulty we decided not to choose this option.

- Gzip, the project is compressed using the famous algorithm designed by Jean-loup Gailly, widely used in Unix systems and published for the first time in 1992

- Brotli, a compression algorithm born in 2013 and developed by Google. This algorithm is particularly efficient when it comes to text compression.

It is also possible to choose between two ways to export the code that makes up the project:

- ASM, this is the "legacy" mode. The JavaScript code is optimised and exported in bytecode through the JavaScript interpreter.
- Web Assembly streaming, permits to export the code in binary format and does not require any parsing since it is ready to run. The code is also compressed to save additional space. Its execution speed is almost comparable to what we would have achieved natively using machine code.

From the preliminary tests we observed that the best combination was WebAssembly Streaming with Brotli compression. To validate this preliminary result we analyzed and studied in depth how these technologies affect the size of a WebGL project and its start-up time, taking into account a wide variety of hardware. With this analysis we could precisely determine the best possible combination for our purposes.

The obtained results

In total we tested 17 different devices among smartphones and tablets. The devices which the software has been tested with belong to different brands, have different CPU and GPU architectures and have different operating systems, including Android, Windows, Linux, and MAC Os. This wide variety of hardware has made it possible to systematically compare several kinds of the most common used hardware so that we can have a clear idea of how the program behaves with the various compilation technologies and what type of experience the end user will get. Eight different projects, which used several combinations of technologies, was tested for every single device. These are:

- WebGL 2.0 with Gzip and Web Assembly Streaming
- WebGL 2.0 with Gzip without Web Assembly Streaming
- WebGL 2.0 with Brotli and Web Assembly Streaming
- WebGL 2.0 with Brotli without Web Assembly Streaming
- WebGL 1.0 with Gzip and Web Assembly Streaming
- WebGL 1.0 with Gzip without Web Assembly Streaming

- WebGL 1.0 with Brotli and Web Assembly Streaming
- WebGL 1.0 with Brotli without Web Assembly Streaming

We have summarized the results obtained in Tables to make them easier to read. In Table 2.1 we carry out the analysis for WebGL 1.0 platform and we tested the combinations of Gzip or Brotli, with and without Web Assembly Streaming.

		Operative System	***WebGL 2.0***	***WebGL 2.0***	***WebGL 2.0***	***WebGL 2.0***
Device	Year	/	***Gzip***	***Gzip***	***Brotli***	***Brotli***
		Browser web	***no stream***	***WAS***	***no stream***	***WAS***
Asus Zenfone 5	2018	Android 9 Chrome 85	*00:20.21*	*00:17.05*	*00:14.27*	*00:16.10*
Amazon Fire HD 8	2017	Fire OS 6 Chrome 84	*00:20.00**	*00:20.43**	*00:21.59**	*00:12.63**
Samsung Galaxy Tab E	2015	Android 4.4.4 Chrome 81	*No webGL2.0*	*No webGL2.0*	*No webGL2.0*	*No webGL2.0*
Huawei MediaPad M5 lite	2019	Android 8 Chrome 84	*00:45.36*	*00:45.67*	*00:42.54*	*00:28.87*
Samsung Galaxy S7	2016	Android 8 Chrome 84	*00:08.992*	*00:06.522*	*00:09.262*	*00:06.075*
Samsung Galaxy S4 Active	2013	Android 5.01 Chrome 85.0.4	*No webGL2.0*	*No webGL2.0*	*No webGL2.0*	*No webGL2.0*
DESKTOP CUSTOM	2014	Windows 10 Chrome 84	*00:02:440*	*00:01:980*	*00:02:830*	*00:01:650*
Asus GL502VM	2018	Windows 10 Chrome 84	00:02.900	00:02.138	00:02.776	00:01.818
Hp prodesk 400 g1	2013	Ubuntu 18.04 Firefox 79	00:03:040	00:02:640	00:03:480	00:02:230
DESKTOP CUSTOM	2012	Windows 10 Firefox 79	00:08:100	00:08:714	00:08:139	00:07:312
Redmi Note 8 Pro	2019	Android 10 Chrome 84	00:22:21	00:14:34	00:23:07	00:16:54
Acer Swift SF314-52	2018	Windows 10 Firefox 79	00:15:13	00:14:37	00:14:46	00:12:30
Honor 8	2016	Android 10 Chrome	00:19:20	00:19:09	00:21:26	00:15:17
HP-PC ProBook 450 G6	2018	Manjaro Chromium	00:11:78	00:10:16	00:09:45	00:09:01
ASUS H81M-D R2.0	2015	Mint 20 Mozilla Firefox	00:11:94	00:10:29	00:09:71	00:10:13
MacBookPro 14.2	2017	macOS 10.15 Safari	No webGL2.0	No webGL2.0	No webGL2.0	No webGL2.0

Table 2.1: WebGL1.0 comparison

In Table 2.2 we carry out a similar study for the WebGL 2.0 platform. In the Tables we reported the project's loading times.

Each device was tested by resetting and clearing the browser cache and using the anonymous navigation mode in order to have the most accurate test as possible. The loading times are reported according to the notation: *minutes:seconds:hundredths of a second*. Since most of the times the project compressed when Brotli and Web Assembly Streaming were active was faster,

Device	Year	Operative System / Browser web	***WebGL 1.0*** ***Gzip*** ***no stream***	***WebGL 1.0*** ***Gzip*** ***WAS***	***WebGL 1.0*** ***Brotli*** ***no stream***	***WebGL 1.0*** ***Brotli*** ***WAS***
Asus Zenfone 5	2018	Android 9 Chrome 85	*00:14.72*	*00:17.92*	*00:13.57*	*00:09.51*
Amazon Fire HD 8	2017	Fire OS 6 Chrome 84	*00:18.75*	*00:12.51*	*00:20.07*	*00:12..62*
Samsung Galaxy Tab E	2015	Android 4.4.4 Chrome 81	*01:31.88**	*01:05.34**	*01:35.25**	*01:01.57**
Huawei MediaPad M5 lite	2019	Android 8 Chrome 84	*00:26.71*	*00:33.68*	*00:31.73*	*00:26.54*
Samsung Galaxy S7	2016	Android 8 Chrome 84	*00:09.473*	*00:06.699*	*00:08.421*	*00:05.708*
Samsung Galaxy S4 Active	2013	Android 5.01 Chrome 85.0.4	*00:18:570*	*00:15:860*	*00:24:320*	*00:15:650*
DESKTOP CUSTOM	2014	Windows 10 Chrome 84	*00:02:550*	*00:01:810*	*00:02:550*	*00:01:250*
Asus GL502VM	2018	Windows 10 Chrome 84	00:02.962	00:01.922	00:02.914	00:01.816
Hp prodesk 400 g1	2013	Ubuntu 18.04 Firefox 79	00:03:150	00:02:480	00:03:550	00:01:980
DESKTOP CUSTOM	2012	Windows 10 Firefox 79	00:08:371	00:07:560	00:06:921	00:05:620
Redmi Note 8 Pro	2019	Android 10 Chrome 84	00:18:37	00:12:02	00:11:30	00:19:08
Acer Swift SF314-52	2018	Windows 10 Firefox 79	00:11:57	00:10:36	00:11:50	00:13:32
Honor 8	2016	Android 10 Chrome	00:14:50	00:27:15	00:18:14	00:22:58
HP-PC ProBook 450 G6	2018	Manjaro Chromium	00:11:59	00:09:79	00:10:09	00:09:45
ASUS H81M-D R2.0	2015	Mint 20 Mozilla Firefox	00:10:28	00:11:57	00:08:92	00:08:54
MacBookPro 14.2	2017	macOS 10.15 Safari	No webGL1.0	No webGL1.0	No webGL1.0	No webGL1.0

Table 2.2: WebGL2.0 comparison

we produced a summary table 2.3 that shows the final results obtained comparing this type of compilation when using WebGL1.0 and WebGL2.0.

As it can be seen some very old and outdated devices such as Samsung Galaxy S4 are not compatible with WebGL2.0 while they are able to run the project with WebGL1.0.

The results obtained show that between the two versions WebGL1.0 is faster than WebGL2.0. Finally, in Table 2.4 the size of the projects in MB is reported. As it can be seen the exported project using WebGL1.0 technology, Brotli and WebAssembly Streaming occupies 8,189MB, proving to be the lightest and most efficient combination for WebGL projects.

The Demo exercises

The goal of the paper is to present the creation of a new environment, whose features have been previously described, that lets the therapist take advantage of the paradigm of rapid prototyping, to generate therapeutic exercises in a very fast and optimised way. To get this point, we show here a couple of Demo exercises which helps understand the huge potential and effectiveness of the method and the complexity of the technical structure behind. The actual maturity of Virtual Reality has allowed to implement a new and rich environment where patient can find a set of tools to extend the available possible practices, exploiting a virtual world.

Demo A involves the patient, asking her/him to identify non-contextual objects within a room. The objects that have been chosen are simple and unequivocal, so as to be sure that the error is due to a difficulty inherent in the subject rather than due to the type of stimulus to be recognised or individual variables. The objective of this activity is to train the executive functions of the subject and more particularly his ability to categorise. In fact, the objects placed inside the room must firstly be recognised by the subject and secondly defined as belonging to the category "typical objects present in the bedroom" or not. Demo B, instead, requires the patient to remember a list, made up of a variable number of objects positioned in an unconventional way in a room. The two demos have several aspects in common, in fact both follow a very precise structure:

- explanation of the task that the user has to perform, both in a textual and auditory way.
- objects randomly appear in the room during the execution of the exercises, as shown in Figure 2.21; if the user correctly clicks on one of them, there is a positive audio reinforcement and the object rotates and disappears. Whereas only a negative audio reinforcement is returned if the object is in the right context, which means a user failure.
- feedback on the results are expressed with some stars and based on the ratio between the total number of clicks and the number of correct objects found. The score is expressed in such a way so that the patient does not feel frustrated even if she/he obtained a bad score. Moreover, phrases have been added to encourage the patient to keep on doing exercises in order to improve.

In addition, input variables and output variables have been associated to both of them: the input variables are parameters that can be set by the specialist, in order to customise the activity according to the skills and needs of their patients.

The output variables instead represent those parameters that are traced during the execution of the activity and that allow the therapist to monitor the rehabilitation path carried out by each patient.

Although they both share a common activity structure, the two demos present significant differences from a technical and clinical point of view.

Demo A is not so structured from the point of view of programming, as the scenario remains fixed, whereas in Demo B the spatial exploration has been implemented and made possible by the use of directional arrows (the exploration is only permitted on the horizontal plane and not on the vertical one).

However, this technical difference also has a resonance on the clinical level because in the second case the user has to apply particular skills, such as visual-spatial orientation and eye-manual coordination. On the clinical level, finally, we choose to train different functions simultaneously. While in Demo A the patient primarily exercises categorisation, in the other case we aim to train both the subject's mnemonic skills (working memory) and cognitive functions, with particular attention to cognitive flexibility.

The point is not only to remember what objects can be found inside a room, but also to be able to recognise them in an unusual perspective and position. We have chosen to insert a "HELP" key in the training screen: so the subject will be able to listen again to what objects she/he has to find inside the room. She/he can press the button and access the Help menu every time it is thought to be necessary.

These kinds of actions are recorded and taken into account when the final score is assigned. The target objects found, the time to complete the activity and the errors made are all traced and stored. These last output variables are traced in the Demo A as well. In this way, during the monitoring of the activities, the operator can distinguish if the subject has mainly got a working memory deficit or if he presents difficulties in both areas.

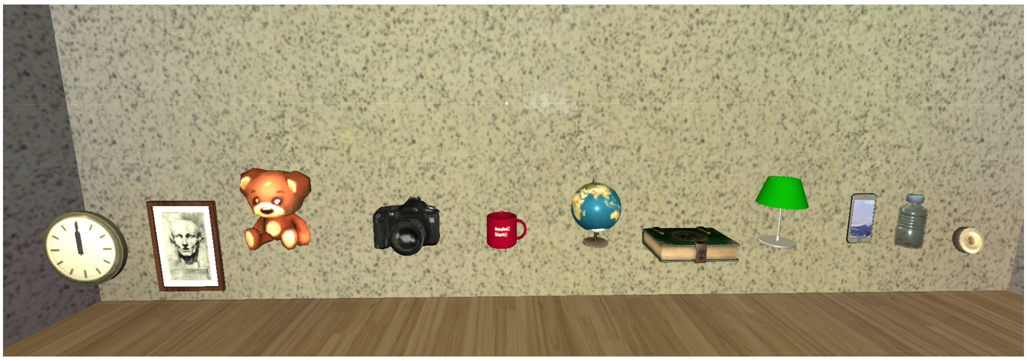


Figure 2.21: A screenshot taken form the Demo example representing the objects that will be randomly arranged in the room

The exercises have a maximum duration, after that if the user did not find all the non-contextual objects, she/he will receive a negative score.

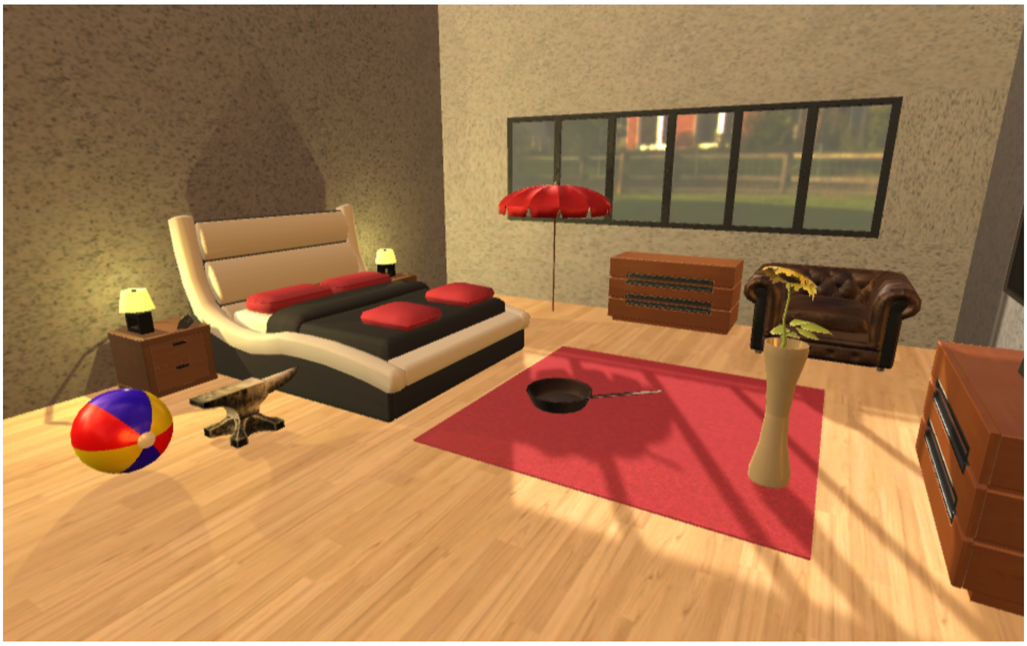


Figure 2.22: A screenshot taken form the Demo example

It is possible to add additional sentences linked to audio tracks. The text part will be displayed on screen while the corresponding audio will be played and, at the end of the audio, the program will automatically pass to the next sentence, loading its audio track. As soon as the explanation of the exercise is finished, another program is launched for randomly visualise the objects in the room. The program selects a series of coordinates inside the room and positions the objects one by one in the selected coordinates, taking them from the set of objects previously chosen by the doctor and checking that there are no overlapping objects or empty areas of the scene. This part has been developed to prevent the patient remembering a certain sequence of areas where to click, trying to stimulate his ability to promptly recognise which objects do not belong to the shown room context.

Each non-contextual object is linked to the scripts that allow the rotation, its disappearance and the expression of a positive reinforcement, while in the case of non contextual object it has been assigned a negative reinforcement.

The rotation script, as soon as the mouse click on the object is detected, rotates the object of 180 degrees, emitting a sound that express the correct execution of the exercise. Then, if the object is not contextual, the script that makes the positive reinforcement and remove the object from the scene will be activated. In this case we simplify as much as possible the understanding of the exercise by the patient, since the script removes from the scene the elements already discovered and facilitates the identification of the remaining ones.

2.5 Creation of immersive environment with open source software

The project consists in a virtual reality experience in which the player can move and interact with the objects available in the environment.

The project was designed thinking about of the real available space. For this reason the experience consists in a circular path: the player will start the game at one edge of the real room, he has to reach a footplate (that is placed both in the virtual and in the real worlds), shown in Figure 2.23, get on it and press a button in the virtual world that will activate its movement. When the footplate is activated, in real world it starts to vibrate while, in virtual world, the player is moved to the upper floor. Upstairs there is a path where the player can move and walk until it reaches the bridge. The bridge is able to detect the collision with the player so when he starts to walk over it some forces are applied to the bridge to simulate the behaviour of a suspended bridge in the vacuum. The path to be followed by the player is circular in the real world and developed upwards in the virtual world. This technique is designed to solve room spacing's issue and to give the player the feeling of walk a long way.

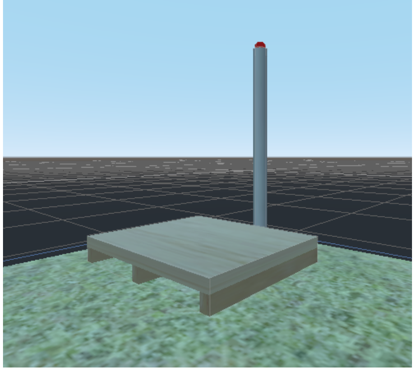


Figure 2.23: The footplate and the button in the initial scene of the virtual environment

The aim that our project would reach is to mislead the player's feelings. For this reason we made a path developed upwards in which there is a suspended Tibetan bridge, shown in Figure 2.24. Furthermore, we add a wooden footplate that is present both in virtual and real world and its position is mapped in the same point. In this way when the player reaches the footplate in virtual world he will reach it in real world too.

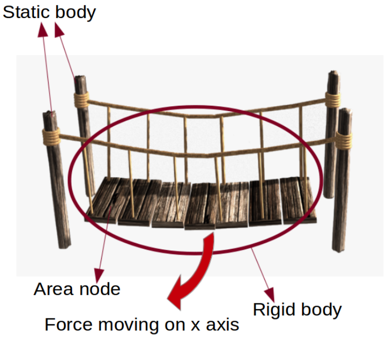


Figure 2.24: The Tibetan bridge

Under the footplate, we added some vibration motors, shown in Figure 2.25, used to mislead the perception of the player. When the player goes on it and presses the button in the virtual world, the motors will be activated. In this way in the real world the player is affected by some vibrations that give him some feelings of instability while in the virtual world he is moved upstairs, ready to start exploring the virtual world.

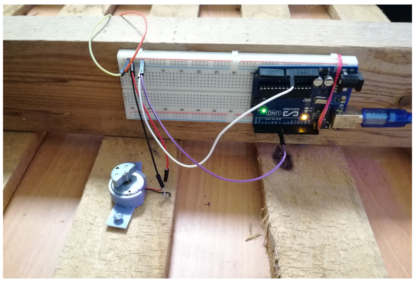


Figure 2.25: The Arduino board installed on the real footplate

First of all, a headset is needed. Our project was developed to work with **HTC Vive**. This is a virtual reality headset developed by **HTC** and **Valve**. The headset uses *room scale tracking technology*, allowing the user to move in 3D space and use motion-tracked handheld controllers to interact with the environment. The kit that HTC Vive offers is composed by different components. We use the following:

- *Headset:* is a device that uses two OLED panels, one per eye, each having a display resolution of 1080x1200 (2160x1200 combined pixels).

The software can also use the integrated camera to identify any static or moving objects in the room; this functionality can be used as part of *Chaperone* safety system which will automatically display a virtual wall or a feed from the camera to safely guide users from obstacles or real-world walls.

- *Controllers:* represent the hands of the player. They have multiple input methods including a track pad, grip buttons, and a dual-stage trigger and a use per change of about 6 hours.
- *Base stations:* Also known as the Lighthouse tracking system are two black boxes that create a 360 degree virtual space. The base stations emit timed infrared pulses at 60 pulses per second that are then picked up by the headset and controllers with sub-millimeter precision.
- *Trackers:* Are some motion tracking accessory. They are designed to be attached to physical accessories and controllers, so that they can be tracked via the Lighthouse system. In our project we attach trackers to the player's feet. In this way we do not have to implement the teleport system.

Godot is a free open source project developed by a community of volunteers. It provides a huge set of common tools and a fully integrated game development environment.

To interact with headset components and setup a virtual environment a new architecture was introduced in Godot called *ARVRServer*. Once ARVRServer architecture is added in game project, every time Godot starts, each available interface will make itself known to the server. *ARVRServer* plays an important role because through it we can bind code to interact with *SteamVR*. This is a virtual reality hardware and software that makes possible execute a virtual reality experience and communicate with the headset components. So ARVRServer represent a middle-layer between SteamVR and Godot. All available headset components that SteamVR detects are made available in Godot thanks to ARVRServer interfaces.

To interact with Arduino we use a plugin based on GDNative called *GDSercomm*. It is a GDNative module that allows a serial port communication between Arduino and Godot. Arduino is an open source hardware and software company, project and user community that designs and produces single-board micro-controllers kits for building digital devices. We used it to implement the vibration system under the wooden footplate.

To create the game environment like footplate, bridge and other objects with which the player can interact we use Blender. It is a free and open

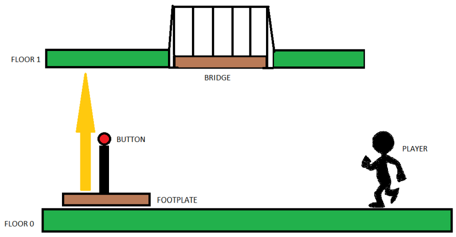


Figure 2.26: Diagram with all visible elements

source 3D creation suite and supports the entire 3D pipeline – modeling, rigging, animation, simulation, rendering, compositing and motion tracking, even video editing and game creation.

The Virtual World made on Godot and Blender

To create a game with Godot the recommended approach is to think about the scenes that compose the project in a naturally way. So we started to imagine all the visible elements with which the player should interact, write it down in a diagram and create a Scene for each player.

Our final diagram is shown in Figure 2.26.

The 4 basic elements that composed our game are:

- **Environment**: composed by the first floor and second floor;
- **Player**: is composed by hands, feet, camera and an arrow that indicates the teleport end point;
- **Footplate** and **Button**;
- **Tibetan bridge**.

Each of these elements has one or more Mesh assign to it. A Mesh is a collection of vertices, edges and faces that defines the shape of a polyhedral object.

The first mesh that we create was the footplate, able to move the observer in the upper floor. This consists of a rectangular table to which we apply a wood texture. Blender provides options to improve some effects related to opacity and color, once the object is hit by light. A similar method has been used for the construction of the Tibetan bridge because also in this case we started modeling from simple cubes and cylinders.

For the Player's hands and feet we download some free assets while, for the first and second floor, we use the basic mesh that Godot makes available.

Once all the meshes were exported in Godot we started to organized them in different scenes to better handle each game's component behaviour. A **Scene** is a group of nodes organized hierarchically while a **Node** represents the core component of Godot and it can perform a variety of specialized functions. Another important aspect that we have to take in mind in game development is the ability to detect some collisions through the objects; for this reason Godot provides four basic types of physical bodies able to detect and respond to collisions:

- **Area**: provides detection and influence. It can detect when objects overlap and can emit signals when bodies enter or exit.
- **StaticBody**: is not moved by the physics engine, it participates to collision detection but does not move in response to the collision.
- **RigidBody**: implements simulated physics, we do not control it directly, but we apply forces to it (such as gravity, impulse, etc.) and the physical motor calculates the resulting movement.
- **KinematicBody**: It is not influenced by physics. This type of physic body is useful for moving objects that do not require advanced physics. It must be controlled by the user.

To better understand these information we have to take in mind that each element that we mention before performs a specific role in the game. For this reason we have to use specific nodes in order to obtain the desired behaviour.

For example, because the Environment is not intended to move, the elements that compose this scene are StaticBody. Each StaticBody wraps the mesh instance and the collision shape.

The Footplate is a KinematicBody because it has to move the player in the upper floor. This behaviour is obtained using the `move_and_slide` function that translates the footplate along the y axis and it stops once it reaches the upper floor. The Bridge is a RigidBody and we apply a rotation impulse to it when the Player walks over it.

The most complex Scene is represented by the Player. Basically, it is a RigidBody but, because it has to work in virtual reality environment, it has to implement some specifics nodes like:

- **ARVROrigin**: is a special node within the AR/VR system that maps the physical location of the center of our tracking space to the virtual location within our game world.

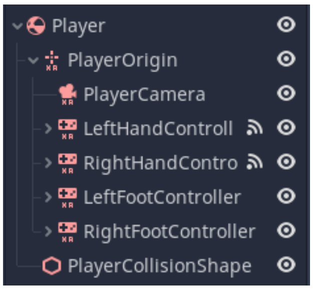


Figure 2.27: Player structure tree.

- **ARVRCamera**: is a camera node with a few overrules for AR/VR applied, such as location tracking.
- **ARVRController one for each hand and foot**: is a spatial node representing a spatially-tracked controller. In our project it wraps the hands' and feet's mesh istances and their collision shapes.

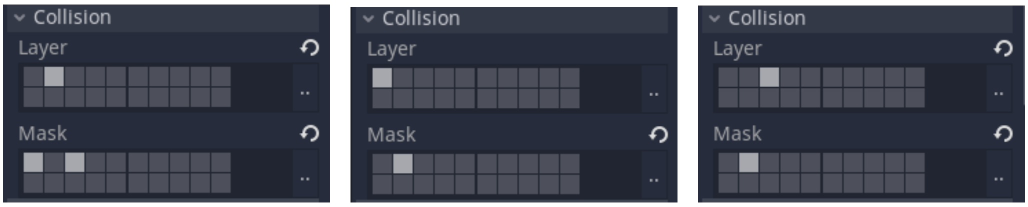
The structure of the Player is shown in the above image.

Most of the elements that compose our scenes contains a **CollisionShape** node. It defines the object's collision bounds that is used to detect contact with other objects. In base of the shape of the object that it has to wrap we could use some basic types or create new complex collision shape in Godot. It is important to know that it is possible to create some complex collision shape inside other external software – like Blender – and import them in Godot.

One of the most powerful collision features that Godot provides is the collision layer system. It allows to build up complex interactions between a variety of objects. The key concepts are **layers** and **masks**. The **collision_layer** describes the layers that the object appears in. The **collision_mask** describes what layers the body will scan for collisions. Keeping track of what layer you are using could be difficult so Godot provides the possibility to rename the layer. In our project we have the collision situation shown in Figure 2.28.

The platform and the environment's floors appear in layer 1 and checks for collisions with player (layer 2). The player appears in layer 2 and checks for collisions with environment (layer 1) and bridge (layer 3). The bridge appears in layer 3 and checks for collisions with player (layer 2).

In order to obtain the desired behaviour from Footplate, Bridge and Player we attach to them a **GDScript**. It is a high-level, dynamically typed programming language used to create content. It uses a syntax similar to Python (blocks are indent-based and many keywords are similar). Its goal



(**a**) Player collision (**b**) Footplate collision (**c**) Bridge collision

Figure 2.28: Different types of collisions

is to be optimized for and tightly integrated with Godot Engine, allowing great flexibility for content creation and integration.

First of all we combined all the scenes created in the Main scene. We chose it as the main scene that Godot runs when it starts. Also the Main scene has a script attached to it because, when we start the game we have to check the availability of the ARVRServer and its devices. The following code shows how we check the state of ARVRServer and its interfaces. From the code we can see that the Main scene is a Spatial node. Most of 3D game objects inherit from Spatial because it allows to move, scale, rotate and show/hide its children in a 3D project.

The function `_ready()` is called only once when the node is created. Here it is possible to initialise variables, load nodes, materials and all the things that we need to start a scene.

In our case we try to find *OpenVR* interface because, if it is available we initialise the virtual environment and sets the frame per second to 90 to avoid the user sickness.

Once all the interfaces are ready the game starts and all the scenes that we create are visible in the virtual world.

First of all we initialise the controllers with which the Player will interact with the virtual world components. The hands can triggers – thank to the collision shape attached to them – if there are some objects that the Player can press or hold. The feet controllers are useful to handle the movements of the Player. In fact, thank to these we can update the position of the Player in the virtual world every time he walks.

The `_process()` is a special function that updates object data every delta frames. We use it to update the position of the player every time he moved in the space.

The other functions that are in the script are used to handle specific signals emitted by some Area nodes.

To handle the Player teleport we attach to the LeftHandController node another script that create an arrow that became visible each time the trigger button of controller is clicked. This arrow show the end point to which the Player will be moved.

Very interesting is how we connect virtual Footplate with the real one. The connection between the Godot footplate node and Arduino was setup by a special GDNative module called **GDSercomm**. It is a module that allows a serial communication between Godot and Arduino, in other words it provides an API from them. It presents some methods like:

- **list_ports()**: to get all the available ports;
- **open()**: to open a communication, flush() - to update the buffer;
- **get_available()**: return the available reading bytes;
- **write()**: write a string in the buffer.

All these function are contained in Sercomm, a C library. In the `_ready()` function – in the last rows of code – we establish a connection with Arduino port.

The `_move_platform_with_button()` function is called when the Player press the button near the Footplate in virtual world. When it is pressed the player is moved upstairs and a string is sent to Arduino. In this way we activate the vibration motors that are under the real footplate. To stop it we check if the virtual footplate reaches the upper floor and eventually we stop the translation in the y axis and send a string to Arduino to stop the vibration of the motors. The `_physic_process()` is used when one needs a framerate-independent deltatime between frames.

In the end, to handle the rotation of the Bridge when the Player walk over it, we use this script that checks when the **LeftFootArea** or **RightFootArea** collide with one bridge's board and when it occurs we apply an `apply_torque_impulse()` to each peace of Bridge the Player touches. The final result that we reach is shown as the image in Figure 2.29.

The vibration footplate and Arduino

To make the user to feel the footplate moving under his feet we placed in the real environment a wooden footplate, hacked using an `Arduino Uno` board and two `vibration motors`.

In the setup function we initialise the serial communication at 9600 bps and wait until the serial port is ready to communicate, then we configure the pin 8 to behave as output. The loop function does what the name suggests, loops consecutively and reads from the serial, when it receives the "h" character Arduino will set a 3.3 voltage to the pin 8 starting the

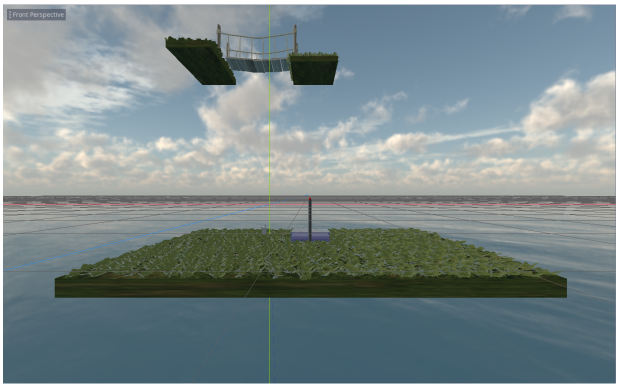


Figure 2.29: Final result of the environment created with Godot

vibration motors, in contrary when it receives the "1" character Arduino set a 0 voltage to the pin 8 which stops the vibration motors.

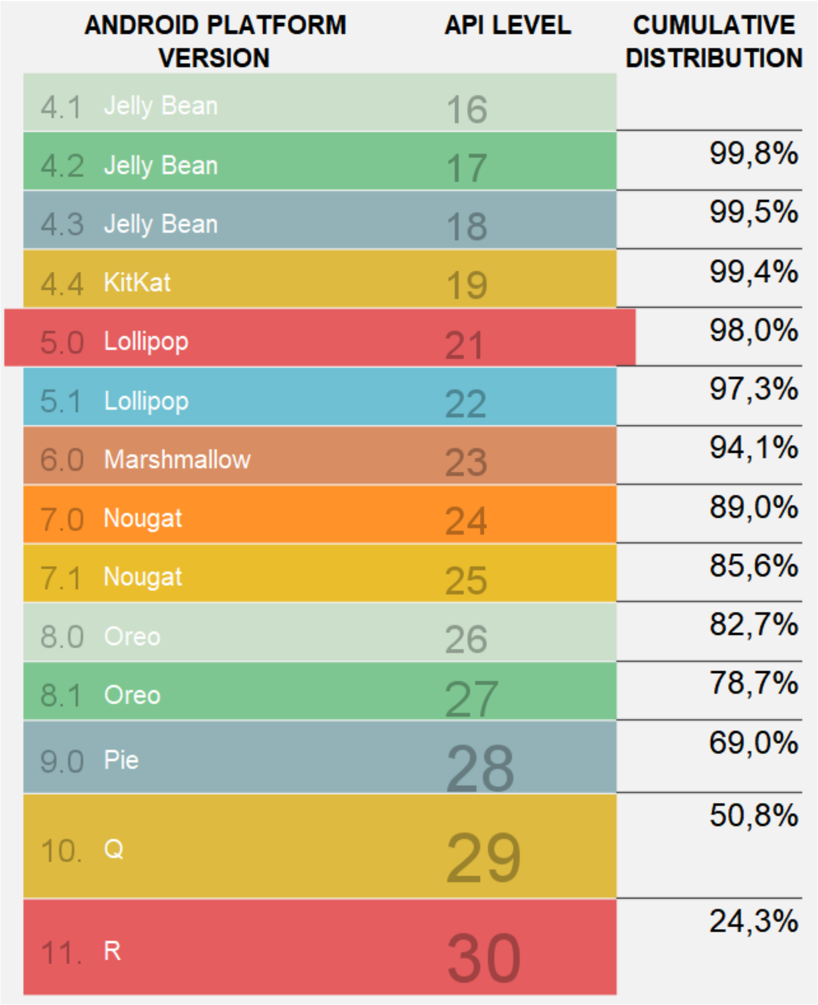
2.6 Helping people affected by Visual Snow

In this section, we propose an application realised through modern Virtual Reality and Augmented Reality technologies that allow simulating the vision of people affected by Visual Snow. The advantages of this application are twofold. The first is from the patients' point of view: it gives people the possibility to show doctors or family members what their eye sees, overcoming the language barrier that makes it difficult to explain the problem. Consequently, misunderstandings can be avoided. The second is from the doctors' point of view: thanks to a mobile app, they can ask the patient to confirm if the image they perceive is similar or the same as the one shown by the software.

To try to provide help to people who have this type of condition, we have outlined the following research methodology. First of all we will make an Android application so that we can reach a large percentage of users in a very short time. After the release of the application we want to collect as much feedback as possible from people. After a subsequent phase of the improvement of the application, which will come as a result of the feedback received in the previous phase, we want to proceed with the creation of a series of anonymous questionnaires with which to collect opinions and specific and detailed feedback from users. What we want to outline is a development path that will not end with this article, but will have to proceed along a period of a few months in order to improve the application as much as possible.

The Visual Snow simulator

This section describes the proposed application to simulate what people with Visual Snow see. The mobile app is built using the Unity software. Unity is a software for the creation of multi-platform interactive environments. It is often used to create video games, virtual reality scenarios or augmented reality scenarios [137, 138, 139, 140]. We have set Android 11.0 (API 30) as the target environment and Android 5.0 (API 21) as the minimum supported version. That allows the application to be installed on 98.0 % of the Android devices currently in circulation, as shown in Figure 2.30; the data shown in the Figure are released by Google annually. The software has been developed



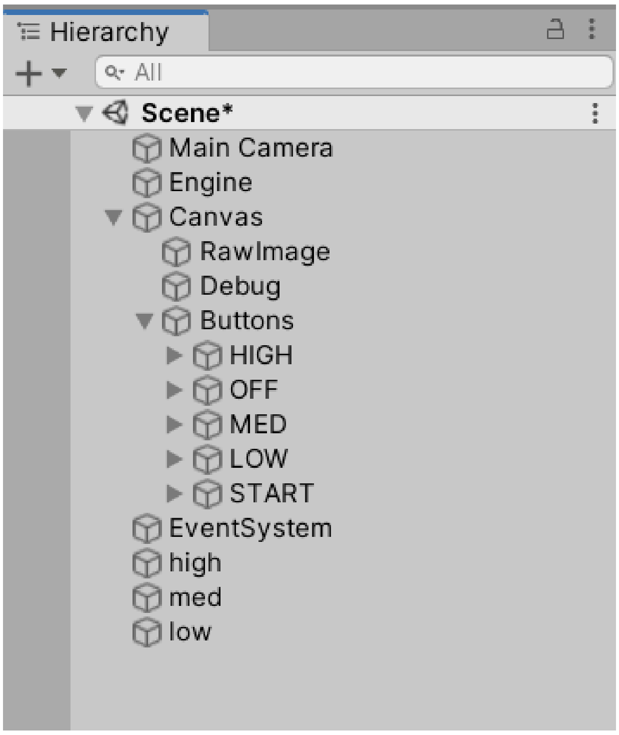
ANDROID PLATFORM VERSION		API LEVEL	CUMULATIVE DISTRIBUTION
4.1	Jelly Bean	16	
4.2	Jelly Bean	17	99,8%
4.3	Jelly Bean	18	99,5%
4.4	KitKat	19	99,4%
5.0	Lollipop	21	98,0%
5.1	Lollipop	22	97,3%
6.0	Marshmallow	23	94,1%
7.0	Nougat	24	89,0%
7.1	Nougat	25	85,6%
8.0	Oreo	26	82,7%
8.1	Oreo	27	78,7%
9.0	Pie	28	69,0%
10.	Q	29	50,8%
11.	R	30	24,3%

Figure 2.30: Android platform distribution - November 2021

to be compatible with the ARM64 and ARMv7 architectures, 64-bit and

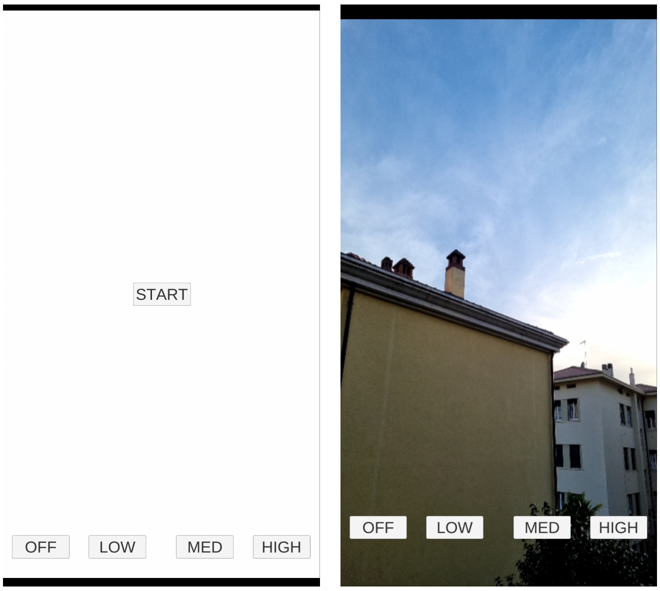
32-bit, respectively.

Figure 2.31 shows the list of Game Objects that compose the software. The first element is the *Main Camera*, its task is to capture images and show them on the user's screen. This should not be understood with the smartphone camera, but instead as a virtual camera inserted inside a Unity scene. The second Game Object is called *Engine*. Some scripts are connected to it, such as those that are executed when buttons are clicked, and it is used to manage the user interface. The third object is the *Canvas*, which represents a graphical drawing environment, on which the buttons and the whole user interface are placed. Inside the *Canvas* we can see that there is an object called *RawImage*. This is used to apply a background to the canvas. In our case the background applied to the *RawImage* (and consequently to the canvas) is the image captured by the smartphone camera. The video stream is managed by a script that periodically updates the image shown on the screen, giving the idea of a smooth view of the world captured by the camera. The buttons that make up the graphical interface are collected within a container that allows a simplified management from a programming point of view. Inside the scene we find the *EventSystem*. This is used for the recognition of user input, such as clicks on the screen. Finally, there are 3



- Scene*
  - Main Camera
  - Engine
  - Canvas
    - RawImage
    - Debug
    - Buttons
      - HIGH
      - OFF
      - MED
      - LOW
      - START
  - EventSystem
  - high
  - med
  - low

Figure 2.31: Scene composition of the software



(**a**) (**b**)

Figure 2.32: Starting screen of the mobile app (**a**), Camera without Visual Snow effect (**b**)

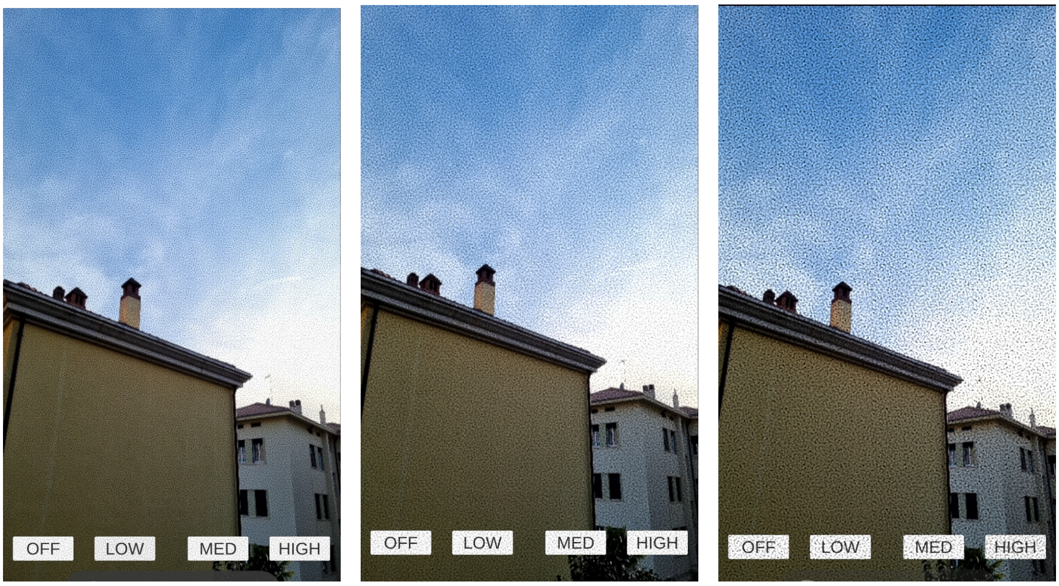
Game Objects called *high*, *med*, and *low* that are the basis for the activation of filters that simulate the Visual Snow. These Game Objects are passed by reference to the Game Object *Engine* which will use them to activate the on-screen effects based on user input.

The initial screen, which appears on the screen, is shown in Figure 2.32a.

The *START* button is in the centre of the scene, which, once pressed, will activate the user's smartphone camera. The application is programmed to ask for permission to access the API that controls the camera if necessary. If the user responds affirmatively, the captured image will be shown on the screen, as shown in Figure 2.32b.

The buttons at the bottom of the user interface are needed to activate or deactivate the effect simulating Visual Snow. After careful consideration, we have programmed three different effects. The button *LOW* presents a barely perceptible Visual Snow effect and tries to simulate what a person sees when his pathology is not particularly serious. The *MED* button, when pressed, activates a much more intense effect than the previous one. The *HIGH* button sets the Visual Snow effect on screen at an extremely high intensity. The *OFF* button deactivates the Visual Snow effect.

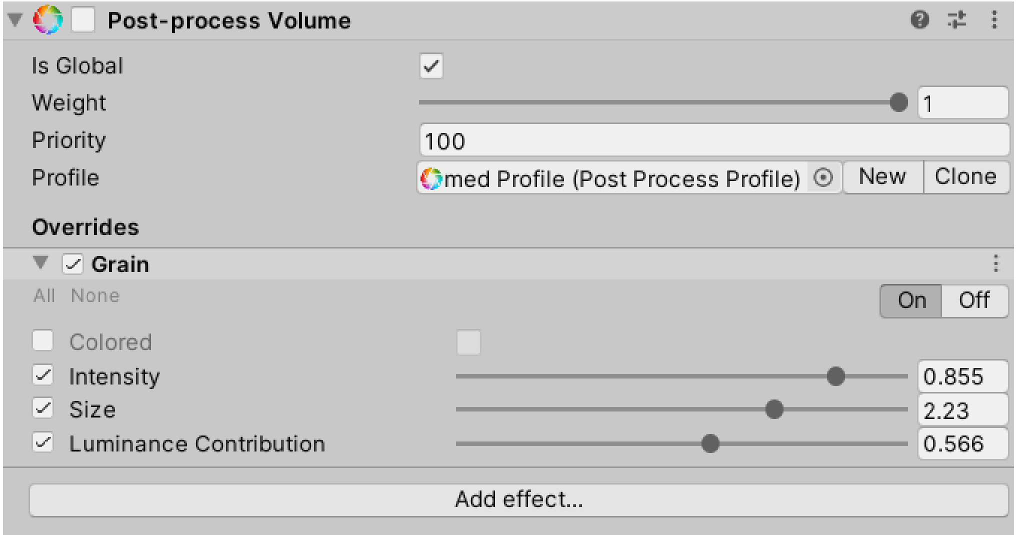
The technical realisation of these effects takes place thanks to the use of



(**a**) (**b**) (**c**)

Figure 2.33: Different Grain filter levels set on screen (**a**) low mode, (**b**) medium mode, (**c**) high mode

the *Post Process Volume*, which allow the insertion of graphic effects that are calculated at the end of the pipeline that manages the rendering of the scene. For each button used to activate an effect, we have programmed a specific Post Process Volume. Inside each Post Process Volume, we inserted a graphic filter of the Grain type. We found it particularly effective to simulate Visual Snow to use a filter of this type. Initially, the Grain filter is programmed to mimic what is captured by old cameras that use chemical photographic film to capture images on film and add video noise to the scene by their nature. The Grain filter can be configured thanks to four different parameters that the developers can adjust. The first parameter is a Boolean variable called *Colored* which allows us to define whether or not the grain effect should be coloured. The second parameter is called *Intensity* and allows us to define the number of particles to be shown on the screen. The third parameter is *Size* and allows you to define the size of the particles shown on the screen. The fourth and last parameter is the *Luminance contribution* which is used by the graphics engine to modify the effect according to the brightness of the scene and to reduce its intensity in poorly lit areas. An example of how the filter has been configured for the medium intensity configuration is shown in Figure 2.34. The application was developed taking into account



Post-process Volume

Is Global ☑

Weight 1

Priority 100

Profile med Profile (Post Process Profile)

Overrides

☑ Grain

□ Colored □

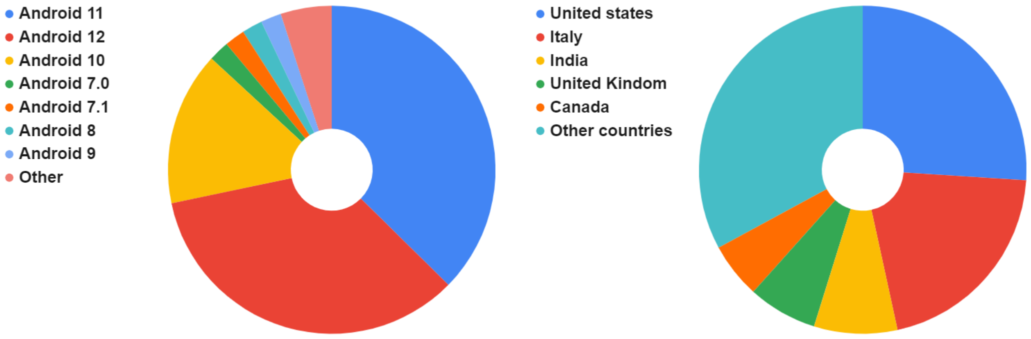
☑ Intensity 0.855

☑ Size 2.23

☑ Luminance Contribution 0.566

Figure 2.34: Post-process Volume with Grain effect on medium settings

Android's best practices. In fact, the latest versions of the operating system require to inform the user before accessing the camera or microphone. In the absence of informed consent from the user, mobile applications will not be able to use these devices. The first time the program is launched, the user is



(**a**)

(**b**)

Figure 2.35: Distribution of the installations across Android versions (**a**), Number of users using the application (**b**)

asked to authorise or not the use of the camera, and only after the user's authorisation can the program run correctly. Regarding the compilation and export phase of the application, we have used the App Bundle format instead of the old APK format as required by Google from August 2021. The final size of the app is 12MB, which makes it possible to install the program even on devices that have minimal amounts of ROM memory. Finally, we made several measurements to estimate the minimum RAM requirements that the smartphone must have. From our tests, we found that the average RAM usage is 32.0MB.

First impressions

In this section, we report the first evaluations and opinions expressed by the users of the application. The application was made available on the Play Store on February 12, 2022. During this period of time, we detected a large flow of downloads. Many users downloaded the application from the Store.

Figure 2.35a shows the distribution of Android versions used by users who installed the application. As we can see, the most recent versions of Android are the most popular ones, and this parameter is in agreement with what is indicated by Google and with which we produced the graph shown in Figure 2.30. Figure 2.35b shows the distribution of installations by geographic area, and the countries that have installed the application are shown in a pie chart. As can be seen from the Figure, the greatest diffusion took place in the United States and Italy. This is probably due to the fact that the message with which we have informed people about the presence of this application in the Store has been inserted in a group composed mainly of people living in the United States.

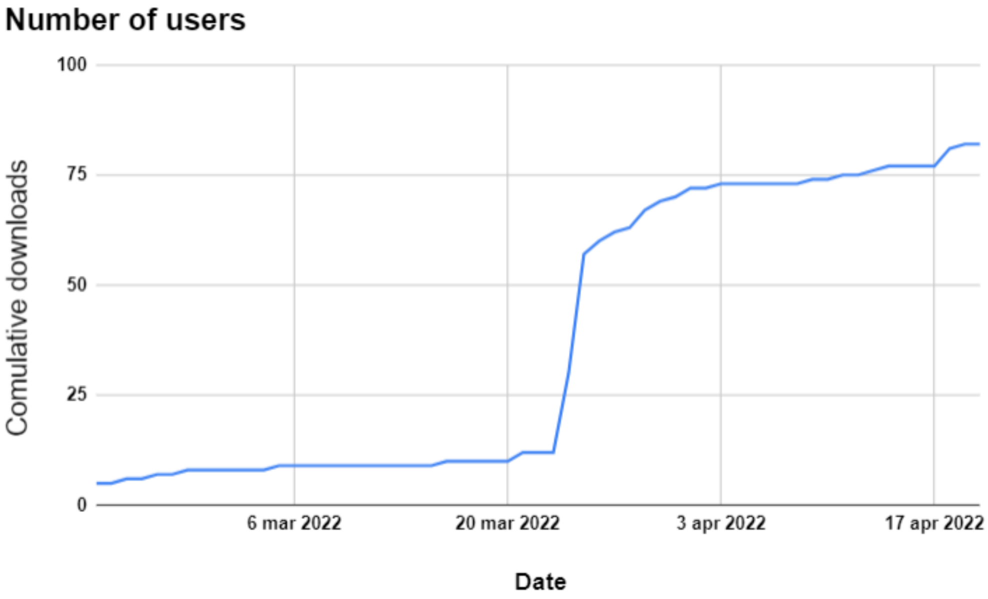
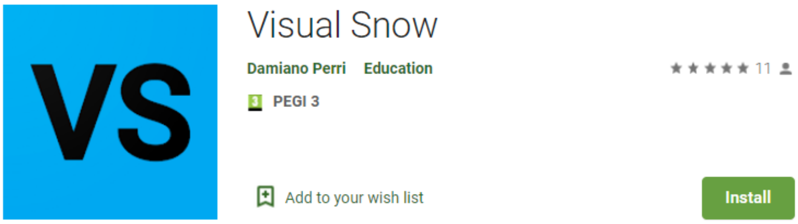


Figure 2.36: Number of users using the application

In Figure 2.36 is shown the trend of downloads from the date of publication of the application. As we can see, the growth is steady, moreover there is a peak of downloads in correspondence on March 25, the day on which we mentioned the application with a post on Facebook.

During this time period, we received many ratings that people spontaneously made within the Google Play Store and also we received several email messages from people suffering from this condition. Regarding the ratings, we got 11 reviews and all the reviews were 5 stars as shown in Figure 2.37. The comments made to us are mainly of appreciation towards



Visual Snow

Damiano Perri Education

★★★★★ 11

PEGI 3

Add to your wish list

Install

Figure 2.37: The mobile app in the Store

the work we have done. As for the suggestions, we got some very interesting ones. For example, a user asked us to implement a feature that allows us to manually set the level of Visual Snow simulated by the software, because his case did not find correspondence in the three levels we preset. The problem of this patient was characterised by an intensity lower than what we have set as "low level". This is surely a very important aspect and we will try to satisfy this request with further updates. Other suggestions concerned the possibility of making the pixels representing the Visual Snow coloured or black and white. Finally, some users have asked us to create the same application for the iOS platform as soon as possible because they are owners of iPhones and, therefore, can not install the program that, by design, is programmed to work only on Android.

2.7 Improving the teaching with Virtual Reality techniques

We have created two different user experiences. One uses the VR, while the other uses the AR. In both cases the graphic engine used is the same, Unity. This software allows the composition of virtual environments starting from basic elements called Assets which the scene is composed with. It also takes care of rendering, real-time lighting calculation and user interaction management. The fundamental tools that have been used are the following:

- the game objects, i.e. the basic elements that make up the scene you want the user to view.
- the scripts, code files written in C# language which you can execute predefined tasks by, such as managing the appearance of objects on the displayed scene, or the camera movement as a key on the keyboard gets pressed.
- the colliders, that prevent intersection or collision between the character user is controlling and the objects in the scene

The generation of the shapes is realised with two C# scripts that allow to model any mathematical function in two or three dimensions. The first script generates the vertices. The second script receives in input a list of vertices, and generates a three-dimensional figure.
So, let us suppose we want to render a three-dimensional function, for example: $f(z) = x * y$. As we all know a standard mathematical function like this is defined in the continuum space, so if we wanted it to be represented we should need an infinite number of points: the aim is therefore to make the

scenario as plausible as possible in a discreet environment, by appropriately choosing the points to be drawn. So to do that, it must be chosen a well-defined length along the X, Y and Z axis, and the number of points (i.e. the number of vertices) that compose the graph must be predetermined. In other words, we need to define a grid of points that will define the level of maximum detail we want to achieve. In addition, it must be kept in mind that the greater the level of detail, the more calculations the user's device will have to perform to display the object on the screen. Defining a grid of points is equivalent to defining a sampling rate. This is the same as when you are processing an electronic signal (e.g. an audio signal) and want to convert from a continuous signal to its discrete representation. If the number of samples is too low we can in fact obtain Aliasing, obtaining an inaccurate representation of the mathematical function we want to show. It is possible to run the program already with the code just described. However, this requires to recalculate all forms at runtime each time. To improve performance we have therefore saved the forms generated with Unity inside the filesystem so that they can directly be reloaded at program start, with no need to recalculate all objects from the beginning. We then processed the shapes with Blender in order to reduce polygonal complexity without changing their information content. In other words, the complexity of the figures in terms of vertices has been reduced but an user who observed them would not notice any difference. This is possible using Blender, a software made to process models and three-dimensional objects. Finally, we have included them again in the Unity project.

The VR environment has generated and compiled by WebGL technology. This means that the application is compatible with all devices (computers or smartphones) on the market since the environment is usable through a web browser. The graphic quality of the scene adapts according to the computational power of the device, while remaining undemanding in terms of hardware requirements. The scene can be observed through a virtual reality viewer, such as HTC Vive, or through a normal computer monitor. The user has the possibility to move around the virtual environment using mouse and keyboard. Inside the environment are visible three-dimensional geometric shapes that allow to understand some mathematical functions otherwise difficult to draw. The AR environment uses the Vuforia framework. The program created is an apk, installable on Android smartphones with 7.0+ operating system. Vuforia is an SDK that allows you to analyze the video stream recorded in real time by the phone camera. Vuforia allows you to create a database of markers (called Vumark). These have manually been associated to the game objects of the scene. When one of the markers

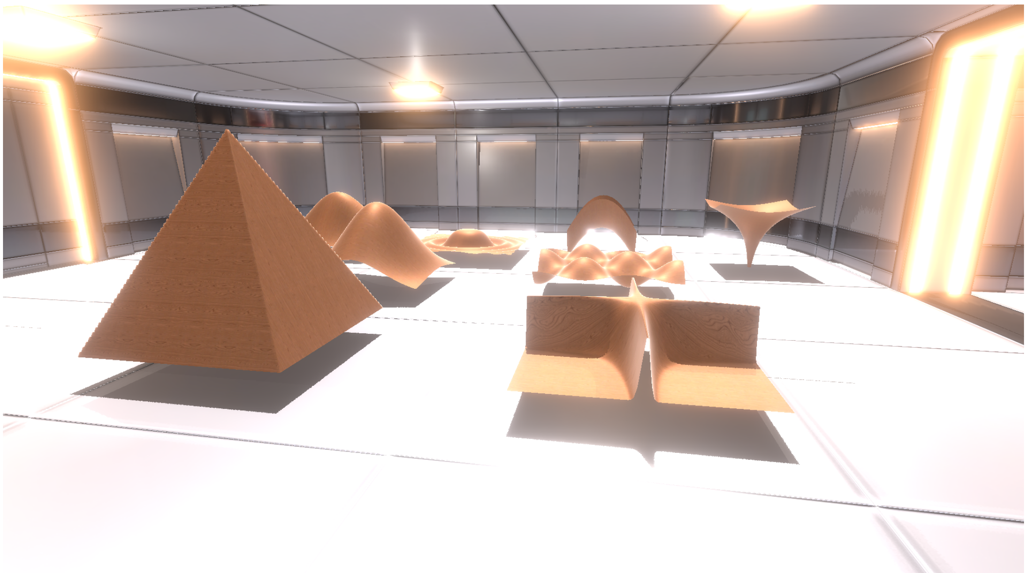


Figure 2.38: VR Example

present in the database is framed by the camera of the user's device, Vuforia tells Unity to show on the scene (and then on the user's screen) the game object associated with the framed Vumark. Moreover, this SDK manages the spatial orientation of the object according to the user's position with respect to the Vumark. If we frame a Vumark and move around it, the object associated with it will rotate as well, allowing us to appreciate it in a realistic way.

The Virtual World made on Unity, Blender and Vuforia

Suppose we want to represent a three-dimensional function within a virtual world created with Unity. The creation of the figures is done through two scripts. First of all it is defined the resolution, that is the level of detail, that the figure must have. If for example we set a resolution equal to 100, then we will have a matrix of 100x100 points. In this way we will scroll the variable X and the variable Y along the grid. Then the step is defined, i.e. how much space must elapse between one point and the next. The step variable has been fixed at 0.1f. By means of a nested double "for cycle", where the first takes care of the X variable and the second the Z variable, we can calculate the Y value in the grid.

The code we made allows to obtain the list of points that makes up the figure. The next step is to calculate the list of triangles. Triangles are a fundamental element in computer graphics. They allow you to specify how the points are interconnected to each other and how they should be rendered on screen. Each "game object" and its figure have associated a script of this type. The difference among the various scripts therefore remains in ability to calculate the getY function. In fact every time it was necessary to change the graph to be represented, it would be sufficient to specify how to calculate the Y to obtain the change of the shown figure; in all cases the unique thing to be done would be to scan the grid (in our example a 100x100 matrix) and recalculate the correct values. The fundamental difference is in the case of two-dimensional functions, like the periodic sine function which is mentioned in chapter 2. In that case only one "for cycle" was sufficient and a depth fixed at 0.05f. In this way we can represent a two-dimensional function as if it were a tube, which allows us to observe it better when we move around it. The next step is the generation of the three-dimensional mesh from the vertices and triangles calculated in the previous step. To do this, a generic script has been created, which can be recalled from all the codes present in the Unity project. Since a three-dimensional object has been generated inside the program, it is necessary to calculate how the light should behave in order to make it visible to the user's camera. The hard work would therefore seem how to illuminate the object in the right way: that means that it is necessary to calculate a light intensity value for each polygon which makes up the entire object structure.

Actually, this calculation, which seems very complex, is carried out very quickly in Unity. The calculation is carried out in two phases. At the beginning, a call to a function integrated in the Unity libraries doubles up triangles in the figure, so each of them has got a "specular twin", with the normal straight line to the surface with opposite direction to its homologous, so that flow of incident light can correctly be obtained.

Finally, a mesh object is created and vertices, normals and triangles are assigned to it. At the end of this operation the figure (also called mesh) is ready to be shown on screen. We have then divided the following work into two different ramifications, involving Virtual Reality and Augmented Reality respectively. In first case a room has been created, and inside the room the three-dimensional figures have been positioned, as shown in Figure 2.38. The project has then been compiled in WebGL in order to be easily used by a web browser and not have any dependence on a specific operating system (Windows, Linux, Android, iOS, etc). As far as the use with augmented reality is concerned, Vuforia software has been used instead. Vuforia is a framework that integrates within Unity. Vuforia allows you to create projects that use augmented reality by providing all the functions essential for operation on mobile phones. In particular, we focused on smartphones with Android operating system.

2.8 Conclusions

Section 2.2 presented some guidelines to digitally reconstruct monuments and artefacts, which enable public administrations, museums, and organisations in charge of promoting art treasures to realise virtual exhibitions and the digitisation of such important cultural heritage. The theme is highly relevant in Europe, particularly Italy, where the topic is part of the National Recovery and Resilience Plan (NRRP).

Two virtual environments were created as case studies, using two different work paths in this work. In the first case, photogrammetry was used, thanks to which the "Palazzo dei Priori" and the "Fontana Maggiore" in Perugia were reconstructed. The objects obtained with photogrammetry were then arranged in an interactive Unity virtual environment. In the second case, manual three-dimensional modelling of the buildings present in the "Piazza della Repubblica" of Foligno was carried out. In this second case, the virtual scenario can be freely explored through virtual reality viewers or the most common desktop computers that interact with the environment via mouse and keyboard input.

The photogrammetry technology to reconstruct three-dimensional elements is able to produce very detailed models that are able to immerse a user within the scenario.

As possible future developments, we will try to investigate how to use these techniques in Metaverse launched by Meta, and how to make virtual visits to museums and cultural places that can be explored through virtual reality and augmented reality techniques more and more engaging and effective. The aim is to allow even very distant people to view the works of art, the architecture of historic buildings, and appreciate the beauty of our cities from a distance.

Virtual reality also has enormous potential for telerehabilitation techniques, and our research has shown how it can be used effectively to help patients with neurological problems. That is accomplished by providing the doctor with an environment easily extendable to new exercises and customizable with media content that may stimulate the patient during practices. The environment we created and described in Section 2.3 and 2.4 are good examples of the sustainability of digital approaches, in line with the EU 2030 agenda. The patient is able to exercise consistently and the doctor can easily monitor the progress. Moreover, the doctor ought to control patients' vital parameters to be able to quantify the emotional stress and any difficulties arising during the exercises. With the actual development of technology, cost-effective devices connected to smartphones and computers, which can monitor an increasing number of the patient's parameters, the doctor has got an increasingly detailed picture of the person's physical and emotional condition while performing the exercises. However, it must be emphasised that although this software is easily accessible as it is open source, the implementation of exercises in accordance with our methodology still requires personnel experienced in virtual reality and this is undoubtedly a limitation of our approach. An advantage of the proposed methodology is its versatility to adapt to the devices possessed by the user. Immersive hardware will undoubtedly provide a more immersive experience, but it is not indispensable for rehabilitation purposes. The proposed approach is of enormous benefit in difficult times, such as we have experienced with the COVID-19 pandemic. There is no doubt that it would be essential to use everything we have learned over the past two years while living in absolutely precarious conditions and often unable to reach the designated places of care.

The project's purpose described in section 2.5 is to create an immersive VR experience from the design to the implementation. Through this project, we tested the Godot's efficiency for VR games. We obtained good performances both in physics simulations and graphics quality. The work won the important title of Best Paper Award at the ICCSA 2021 conference.

In Section 2.6 we created an Android software application capable of representing what people affected by Visual Snow Syndrome perceive in a simple and effective way. The application has been published on the Play Store and can be freely downloaded from any device with at least Android 7.0. The code produced during the development of the application has been made Open Source and made available to the scientific community through GitHub page[11]. A short video showing the application running has been uploaded to YouTube and is publicly available[12]. We are very interested in continuing this project and analysing further developments for the application. The first goal is to make the application cross-platform, i.e., expand its compatibility with the iOS operating system and allow iPhone and iPad users to use it. A second objective is to analyse the opinions of doctors and patients suffering from Visual Snow. For this reason, we are planning to carry out a series of anonymous questionnaires to collect data that will allow us to improve the application. Moreover, hopefully, it will help people suffering from this pathology explain their problems better and better and help doctors get

[11] https://github.com/DamianoP/VisualSnow

[12] https://youtube.com/shorts/cl_SAjyGY64/

adequate information from patients quickly. Following the publication of the article, we were contacted by the Visual Snow Initiative[13]. VSI is a foundation dedicated to helping Visual Snow patients worldwide. They were very impressed by the effectiveness of the Android application and asked us to start a collaboration for future improvements and development of even more effective applications that will hopefully also be able to alleviate the symptoms of this disease. The collaboration has already been agreed upon and will start after March 2023.

In Section 2.7 we then created 3D environments to help improve learning in high schools through virtual and augmented reality techniques. We asked the students to fill in various questionnaires to understand the quality level of the 3D environments created and their efficiency in improving learning. The first results were published in the journal [141]. Subsequently, the research was further expanded and we involved 3 schools in Umbria. The results of this research are still being studied and will be published in the course of 2023.

[13] https://www.visualsnowinitiative.org/

Device	**Year**	**Operative System / Browser web**	***WebGL 2.0 Brotli WAS***	***WebGL 1.0 Brotli WAS***
Asus Zenfone 5	2018	Android 9 Chrome 85	*00:16.10*	*00:09.51*
Amazon Fire HD 8	2017	Fire OS 6 Chrome 84	*00:12.63**	*00:12..62*
Samsung Galaxy Tab E	2015	Android 4.4.4 Chrome 81	*No webGL2.0*	*01:01.57**
Huawei MediaPad M5 lite	2019	Android 8 Chrome 84	*00:28.87*	*00:26.54*
Samsung Galaxy S7	2016	Android 8 Chrome 84	*00:06.075*	*00:05.708*
Samsung Galaxy S4 Active	2013	Android 5.01 Chrome 85.0.4	*No webGL2.0*	*00:15:650*
DESKTOP CUSTOM	2014	Windows 10 Chrome 84	*00:01:650*	*00:01:250*
Asus GL502VM	2018	Windows 10 Chrome 84	00:01.818	00:01.816
Hp prodesk 400 g1	2013	Ubuntu 18.04 Firefox 79	00:02:230	00:01:980
DESKTOP CUSTOM	2012	Windows 10 Firefox 79	00:07:312	00:05:620
Redmi Note 8 Pro	2019	Android 10 Chrome 84	00:16:54	00:19:08
Acer Swift SF314-52	2018	Windows 10 Firefox 79	00:12:30	00:13:32
Honor 8	2016	Android 10 Chrome	00:15:17	00:22:58
HP-PC ProBook 450 G6	2018	Manjaro Chromium	00:09:01	00:09:45
ASUS H81M-D R2.0	2015	Mint 20 Mozilla Firefox	00:10:13	00:08:54
MacBookPro 14.2	2017	macOS 10.15 Safari	No webGL2.0	No webGL1.0

Table 2.3: Synthetic comparison of the results

Project	**Size in MB**
WebGL 1.0 Gzip	9,965
WebGL 1.0 Gzip WAS	9,967
WebGL 1.0 Brotli	8,189
WebGL 1.0 Brotli WAS	8,189
WebGL 2.0 Gzip	10,454
WebGL 2.0 Gzip WAS	10,455
WebGL 2.0 Brotli	8,560
WebGL 2.0 Brotli WAS	8,564

Table 2.4: Size of the exported project

Chapter 3

Strategies to make the most of the advantages of Cloud Computing

3.1 Introduction

This chapter describes the research path that was tackled during the three-year PhD studies.. The chapter describes the study undertaken on the world of Cloud Computing and the delivery of IT services via highly reliable, efficient and high-performance environments. The achieved results were then the subject of two scientific journal publications. The first one focuses on best practices for the implementation of a cloud infrastructure [142] and it represents a follow-up of another paper published at a conference [143]. The second one describes the successful implementation of a platform for university examinations [144] and, also in this case, it extends the work presented in another conference paper, where the candidate is also one of the authors [145].

The platform, called LibreEOL[1], which is used by the University of Perugia, has been instrumental during the COVID-19 pandemic and has enabled more than 100,000 certified exams to be held. LibreEOL has a history of dozens of years of development behind it[2], but it is since 2015 that the fine-tuning of the platform has begun, including the work of optimising and rewriting the code carried out by the candidate for the three-year degree thesis [146]. The COVID-19 pandemic, however, required major infrastructural and functional improvements. The system must be able to

[1] https://www.libreeol.org/info

[2] The candidate was designated as the project's technical manager

ensure proper functioning for the conduct of legally valid examinations. From a technical point of view, it is important to note that examinations, by their nature, generate an unpredictable and impulsive load, with hundreds of simultaneous accesses and sudden load peaks.

First of all, the best infrastructural architectures for implementing applications of this type will be described, and then the operation of the platform will be explained in detail.

The infrastructures and techniques that enable the deployment of IT services are constantly evolving. Today, it seems pretty natural to use online storage spaces to store heterogeneous documents, such as photographs, music files or text documents. Web applications, which do not require additional software installed on clients and perform calculations via powerful remote servers, are also typical. There are also chargeable services that guarantee the ability to perform scientific calculations or exploit the computational capacity of hardware with specific power and cooling requirements. This is possible thanks to modern technologies and software development models such as cloud computing. Users delegate the responsibility of creating available and reliable infrastructures to third-party companies that take care of the management and maintenance of the hardware necessary to complete the required tasks. This Section will analyse the best practices for creating a modern and reliable cloud architecture, which exploits the potential of the most advanced software technologies, such as Docker containers, and guarantees high availability and scalability.

The infrastructures and techniques that enable the deployment of IT services are constantly evolving. Today, it seems pretty natural to use online storage spaces to store heterogeneous documents, such as photographs, music files or text documents. Web applications, which do not require additional software installed on clients and perform calculations via powerful remote servers, are also typical. There are also chargeable services that guarantee the ability to perform scientific calculations or exploit the computational capacity of hardware with specific power and cooling requirements. All this is possible thanks to modern technologies and software development models such as cloud computing. Users delegate the responsibility of creating available and reliable infrastructures to third-party companies that take care of the management and maintenance of the hardware necessary to complete the required tasks. Section 3.2 will analyse the best practices for creating a modern and reliable cloud architecture, which exploits the potential of the most advanced software technologies, such as Docker containers, and guarantees high availability and scalability. Section 3.3 will describe a real-life use case of the best practices we have defined with extensive use of cloud capabilities.

3.2 Best practice for the realisation of a robust cloud infrastructure

Nowadays there are several aspects that have emerged with the spread of cloud environments, social networks and multimedia platforms that tend to capture personal information of users. This is despite the fact that many organisations (such as the European Union) have issued (e.g. through the General Data Protection Regulation, GDPR[3]) a series of recommendations aimed at protecting personal data. There is no doubt that today this information constitutes an absolutely attractive asset for the big players in Information Technology because through it they are able to convey, in a much more incisive manner, messages and other services that suit the increasingly frequent uses and customs of the population. Our work aims to provide organisations and those sensitive to the importance of respecting the protection of personal data with a set of best practices that will enable them to implement a private or hybrid cloud environment, i.e. one that is able to interface with commercial providers to access computational resources without compromising the security of sensitive data. In order to facilitate the adoption of this best practices and the dissemination of these open solutions, thus also oriented to the mitigation of the lock-in[4] phenomenon, we have provided a use case that makes this adoption of open technologies easier and more transparent.

The use case that we present involves the implementation of a website based on a PHP backend assisted by an SQL database for data storage, the creation of a container for the management of IT services (e-mail, DNS), and we will make the necessary considerations for the management of the horizontal scalability of the web server, under established metrics and the number of user requests. The design of an IT infrastructure must also consider the aspects concerning data security, protecting them from attacks that can occur from outside and from any catastrophic events that can lead to severe consequences, such as the complete loss of information. This Section is divided as follows: In Subsection 3.2.1 we summarise the characteristics and aspects of the various types of IT infrastructures concerning disaster recovery techniques, highlighting their main strengths and weaknesses.

[3] https://gdpr.eu/

[4] Lock-in refers to the tendency of proprietary solutions to anchor the user within the area of action of the services provided by the company.

In Subsection 3.2.2 we present an architecture capable of satisfying the requirements imposed by us and capable of exploiting the most current and modern technologies, defining best practices for the creation of scalable architectures using Docker containers. The proposed architecture is also adopted in an actual use case that uses the technologies and methods proposed.

In Subsection 3.2.3, we presented the infrastructure in the light of indicators showing the efficiency and cost savings obtained from adopting a solution capable of scaling down and up according to the load induced by users.

Subsection 3.2.4 describes the statistics and experimental results that have been collected over the period considered. The various graphs are explained in light of the operations carried out, taking into account the trends observed in them for both CPU and memory by the various components of the architecture.

3.2.1 Disaster recovery

To design an IT architecture that is robust and resistant to cyber-attacks and natural events is necessary to perform a preliminary requisites analysis that establishes three fundamental parameters [147]. The first parameter is the Service Level Agreement (SLA). It represents the percentage of time our IT system will work correctly during a calendar year. As the SLA value increases, the costs necessary for implementing an infrastructure capable of satisfying it will also increase. For example, if we wanted an SLA value equal to 90%, this would be equivalent to saying that our IT system could be offline (during a whole year of operation) for 36d 12h 34m 55s. An SLA value of 99.99%, on the other hand, is equivalent to saying that we can tolerate downtime of 52m 35s over a year.

The second parameter that must be defined is the Recovery Point Objective (RPO) and represents the maximum time interval that we are willing to lose. This parameter is related to data backup, increasing cost and complexity as the required RPO value decreases.

The third parameter that must be defined is the Recovery Time Objective (RTO) and represents the time that we are willing to accept between the interruption of the operation of our IT infrastructure and its recovery. In general, we can say that older architectures have an RTO time of more than 48 hours. For example, in the case of hardware problems, which require turning off the machine and replacing the non-functioning part, it is also necessary to consider the time required to find the components to be replaced and the time required to restore the system [148, 149]. In addition, it is

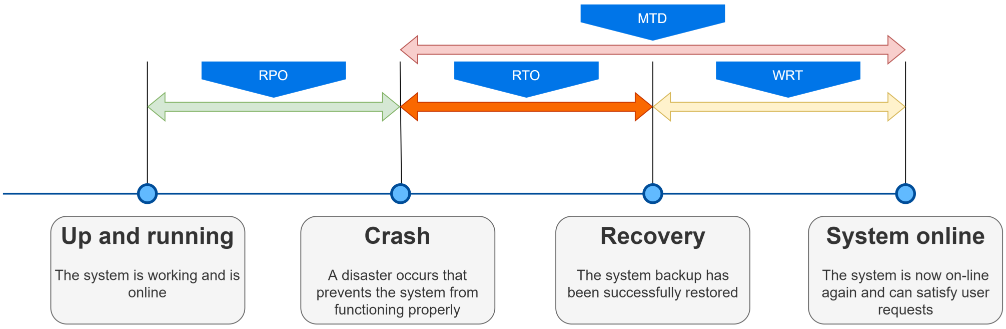

MTD
RPO
RTO
WRT
Up and running
The system is working and is online
Crash
A disaster occurs that prevents the system from functioning properly
Recovery
The system backup has been successfully restored
System online
The system is now on-line again and can satisfy user requests

Figure 3.1: Disaster recovery timeline

essential to define the Work Recovery Time (WRT), which indicates the maximum tolerable time frame required to verify the integrity of the data recovered. The sum of RTO and the WRT gives the Maximum Tolerable Downtime (MTD) parameter shown in Figure 3.1 along with the other parameters.

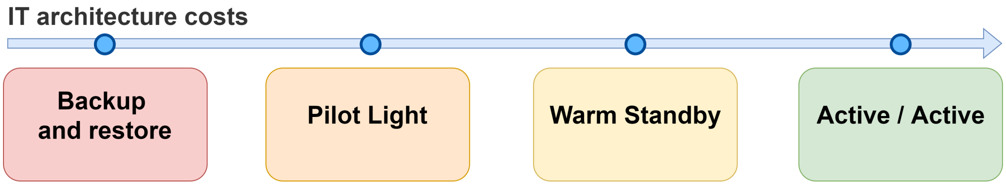


Figure 3.2: Disaster recovery plans

Classification of disaster recovery

Considering that different levels of disaster recovery imply different costs and organisational complexities, four different plans have been defined in the literature[150].

The first type is called "Backup and Restore" and involves the creation of manual or periodic data backups. If a failure or disaster occurs, the system will completely stop working, and it will be necessary to proceed with the restoration of the infrastructure manually; in the most severe cases, it will be necessary to shut down the system and restore the backup. It was frequently adopted in legacy architectures and is still used today for data systems and delivering non-critical services. This solution presents the lowest costs but exposes to the highest risks. The backup and restore type has an RPO parameter that can be measured in hours and an RTO that varies between 24 and 72 hours.

A second type is the "Pilot light", which provides the creation of data backups on an hourly or daily basis. The architectures that use this type are built using virtual machines or Docker containers and do not install the software that must provide the services directly on the machine's main operating system since the data can be quickly restored in case of a disaster even remotely. The RPO value can be measured in minutes, while the RTO value can be measured in hours.

A third type is the "Warm Standby". The systems that adopt it constantly replicate the data of the primary storage devices in auxiliary backup systems installed in remote locations. Thanks to the synchronisation of data that occurs almost in real time, it is possible to obtain an RPO in seconds and an RTO parameter in minutes.

The fourth type is called "Active/Active": it provides the highest level of reliability, implementing the application's load balancing. Such infrastructure is built using at least two availability zones, far enough to guarantee service continuity even in natural disasters. Creating a second availability zone implies the complete replica of hardware and software resources. A load balancer service will allocate the user requests on the web server instances available on the two availability zones, balancing them and constantly monitoring their health. The allocation of resources in the availability zones may be asymmetric. This type of architecture allows obtaining an RPO in the order of milliseconds, which can even be 0 in some cases. The RTO value is instead potentially equal to 0. The types just described are shown in Figure 3.2. We should note that the higher the requirements of the infrastructure, the greater the costs necessary for its implementation: by observing the Figure, we can see that moving to the right increases both the general security of the system and the overall cost that must be faced to create the required IT architecture.

3.2.2 The proposed system architecture

This section describes the architecture proposed to create an IT infrastructure that guarantees high reliability and high resistance to failures. The infrastructure is built using the Active/Active disaster recovery type within the virtualisation environment provided by a cloud provider. The use case we intend to focus on is the management of a website based on PHP and bash for backend management; HTML5, JavaScript and CSS3 for frontend management; SQL for the relational database. The site can manage a very high number of simultaneously connected users, dynamically scaling the number of PHP and databases nodes based on the number of users connected. We have two possible ways of increasing the resources available to an application: horizontal scaling and vertical scaling. By horizontal scaling, we mean the possibility of increasing the number of server instances based on the number of connected users and sentinel parameters that enable the mechanism to be activated. With the expression vertical scaling we mean the increase in terms of RAM and CPU cores of the same server. Prudent use of resources, aimed at obtaining maximum performance at the best cost, requires careful planning of these two mechanisms to deal with situations of massive users with more powerful hardware. In this way, we obtain the advantage of low operating costs at night or idle scenarios with a low number of nodes, with costs that will increase when we have to afford high workloads. In the case of planned intensive workloads, it is recommended to perform a vertical scaling to afford the extraordinary workload better. The adoption of vertical scaling involves shutting down the virtual machines increasing the number of cores and the amount of RAM, redefining the related metrics for the horizontal scaling, and increasing the related costs.

The scheme of the architecture discussed in this Section is shown in Figure 3.3, and in the following subsections we will describe the main components.

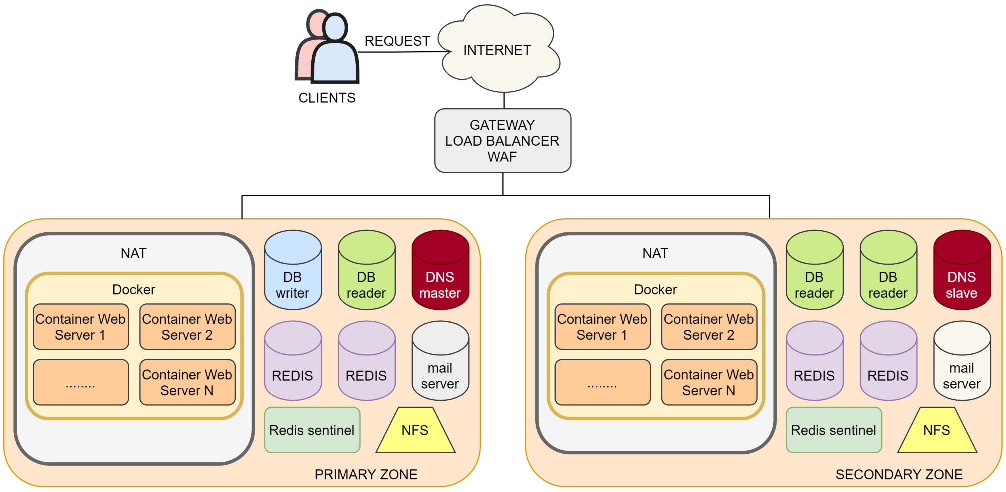


Figure 3.3: The proposed architecture

The load balancer

The first element is the load balancer, which allocates the incoming requests on the resources available in the two availability zones [151, 152, 153]. Several algorithms can be implemented for configuring a load balancer. The simplest algorithm is the "randomised" one; one of the target web servers is selected randomly for each new request. A disadvantage of this algorithm is that balancing requests between the available resources is not guaranteed.

Another algorithm is the "round-robin". In this case, the requests will be sent in a perfectly balanced way to the two available resources. This algorithm has a disadvantage: it does not consider the number of pending requests on computing resources.

We recommend using the third available algorithm: the Least Outstanding Requests (LOR). A new request will be allocated to the resource with the lowest number of pending requests. In this case, if horizontal scaling is active, new requests will be allocated to the newly instantiated servers until a new equilibrium is reached between the requests allocated to each computational resource.

The Redis database

Redis is a very high-performance non-relational database, whose characteristics make it extremely suitable for managing session variables in a distributed environment [154]. In fact, in a distributed environment, it is impossible to maintain the consistency of session variables since the user can switch from one server to another in the course of navigation. We have adopted a configuration based on two Redis servers, one configured as primary and the other as secondary, to guarantee this fundamental service's high reliability. The official image for building a Redis server is distributed directly by the Docker repository[5]. To manage operation between the two Redis servers, one can use the Redis Sentinel daemon, which can monitor the health of the servers and manage the switching of roles if necessary[6].

The SQL database

Considering our proposed use case, an essential component of the architecture is the relational Database which manages the storage and high-performance search of information. We adopted the MariaDB relational database, available under the open-source license GPL v2 [155, 156]. At night, or when the workload is low, there are only two database instances, one in the primary Availability Zone and one in the secondary Availability Zone. The first instance can be identified as master, with write and read permissions. The second instance is configured as slave, with only read permissions. Relational databases must guarantee the consistency of the data structure within them; if we used two databases and both had write permissions, both might write data, and that once inserted, it could violate the uniqueness conditions of a primary key inside of the tables. It will be possible to implement horizontal scaling mechanisms during intensive system load, i.e., to instantiate slave replicas. These replicas will enable high-performance data reading. It should be noted that each replica of the database will have a latency time for the

[5] https://hub.Docker.com/r/bitnami/redis/

[6] https://hub.Docker.com/r/bitnami/redis-sentinel

master, generally in the order of milliseconds or tenths of a millisecond. A latency of this type is utterly irrelevant to the use case we are presenting since, if users see new content with only a one-millisecond delay, there is no problem. Vertical scaling can be adopted in cases where latency is essential, for example, because an application is used in a mission-critical environment.

Database auto-scaling

Too many simultaneous connections to the database may cause slowdowns and delays in delivering data to users. In order to efficiently scale the database, it is necessary to identify appropriate parameters to be used as metrics [157], so that appropriate scaling policies can be activated when a peak in user requests is observed that generates the overcoming of limit thresholds. We have identified two fundamental parameters: CPU utilisation and the number of concurrently active connections. When defining CPU threshold, we have to consider the time a database takes to be instantiated; our tests show that for a medium-sized database are necessary approximately 3 minutes. Our estimate for the CPU utilisation percentage threshold is 70%. A higher value could prevent the system from handling a peak of users, as the time required to start the database could be too long, and therefore users could encounter requests denied when using the service. A lower value would instead lead to a waste of money as we will run replicas of the database when there is no real need.

As per the number of concurrently active connections, our estimate for a medium-sized database is 30 connections. However, this parameter is also influenced by the machine's computational power on the database. Each database is managed by two vCPUs and 4 GiB of RAM in our case.

Storage Configuration

We consider the use case based on containers, which have no persistent memory. Therefore, it is necessary to configure a permanent memory that can be shared between the various active containers. Many Cloud providers offer proprietary technologies for the creation of highly available disks, such as Amazon EFS (Elastic File System)[7], Microsoft Azure File Storage[8] or Oracle Direct NFS[9]. An open source solution is represented by GlusterFS[10], configured in the two virtual machines which are in charge to deploy the containers in the availability zones in use. To this end, each virtual machine

[7] https://aws.amazon.com/en/efs/

[8] https://azure.microsoft.com/it-it/services/storage/files/

[9] https://docs.oracle.com/en/database/oracle/oracle-database/12.2/ssdbi/about-direct-nfs-client-mounts-to-nfs-storage-devices.html

[10] https://www.gluster.org/

must configure an XFS partitioned disk[11] on which the data will be stored. GlusterFS will perform real-time data replication between the various nodes that make up the cluster it manages.

The containers in backend

The configuration of web containers requires a step-by-step approach. The first step must be done in a local environment by writing a Dockerfile [158, 159]. The Dockerfile is a text file that contains the definitions of the software packages to be installed inside the Docker image we need to run. The second phase requires an in-depth test of the Docker image we have created. For example, in our case, we tested the web application by running the Docker container inside a dedicated Virtual Machine to which a domain name was assigned via DNS. Next, we need to configure a NAT to assign private IP addresses to the Docker containers we will instantiate. We have two ways to proceed to the next stage, which are now described in detail.

The first way is to manually configure a cluster that orchestrates the web containers and does not use the preconfigured services made available by the Cloud providers. In this case, one must first prepare a virtual machine with Docker installed. Since this machine will have to manage another number of containers, it should be instantiated with a high number of CPUs, and a suitable amount of RAM. Inside the machine, we can install the HAproxy[12] daemon, which will sort requests between the Web containers that are part of the cluster we have created. The downside of this first route is that we will not be able to exploit auto-scaling effectively: it will not be possible to perform horizontal auto-scaling, but we can only perform vertical scaling since the maximum number of CPU cores in the cluster is determined at the time of creating the virtual machine.

The second way is to rely on the services offered by the Cloud provider. Many providers make it possible to orchestrate Docker containers automatically, managing horizontal scaling. Amazon provides the Elastic Container Service (ECS)[13], Microsoft uses Azure Kubernetes Service (AKS)[14], Google provides Google Kubernetes Engine (GKE)[15]. If one wants to use services offered by the Cloud provider, one needs to upload the Docker image generated to the private repository of the user's account. Then one can start the image within the container orchestration service, taking care to have the NAT previously configured manage the IP addresses of the nodes.

[11] https://docs.oracle.com/cd/E37670_01/E37355/html/ol_about_xfs.html

[12] http://www.haproxy.org/

[13] https://aws.amazon.com/ecs/

[14] https://azure.microsoft.com/en-us/services/kubernetes-service/

[15] https://cloud.google.com/kubernetes-engine

Auto-scaling

After configuring the Docker containers that manage the backend, we can proceed to configure horizontal scaling, i.e. adding equivalent nodes that can handle the users' requests [160]. Since the auto-scaling feature ensures that the cluster delivers the computational power needed to manage users and minimises costs during idle states, we recommend assigning a modest amount of CPU and RAM to individual web nodes. For example, in our case, we allocated 1 vCPU and 2 GiB of RAM per node. These values depend very strongly on the use case and can only be estimated through system load tests and local tests that monitor RAM usage. However, it is advisable to keep these values low to take advantage of the benefits of auto-scaling.

There are several parameters for configuring an auto-scaling service. The first parameter is the minimum number of nodes we wish to have operational within an availability zone.

The second parameter indicates the maximum number of nodes that auto-scaling can instantiate. This parameter varies according to the use case; it is generally advisable to insert a high value capable of satisfying load peaks. Then we have to define additional parameters concerning the Policies that activate auto-scaling and which we consider guidelines to be followed when configuring infrastructures of this type. The percentage of CPU utilisation represents the first parameter. If the average processor usage value within the cluster exceeds a certain threshold, the orchestrator will activate new nodes. It is necessary to remind that there is a start-up time for activating containers, so it is advisable to enter a value that is not too high; otherwise, we could not manage load peaks. In our use case, we consider the optimal parameter for adding a new node is the 60% average CPU usage. The second parameter concerns the average RAM usage; in this case, we recommend a value of about 80% of the maximum RAM allocated to a node. According to our tests, sometimes auto-scaling based on CPU and RAM usage may not be sufficient. An essential parameter is the number of requests a node has handled in a given period. For example, it is possible to define a policy that monitors the requests received by a node over 5 minutes. In our use case, we have found that in order to have effective and efficient auto-scaling, it is recommended a value below 26,000. It is interesting to note that a high number of requests within 5 minutes is not related to high processor utilisation, as requests can be of different kinds, from file system data requests to database data requests to computational calculations. However, it is essential to measure this parameter and calibrate it according to the particular use case by carrying out load tests before the system goes into production. The last configuration parameter concerns downscaling used to remove nodes when they are no longer required. For this policy, we recommend monitoring the cluster's CPU utilisation. When the cluster reaches a low CPU utilisation, for example, around 15%, it is possible to remove one node at a time, keeping the system performance constant and reducing costs.

DNS Server

The architecture we propose involves configuring a DNS server inside a Docker container[161] which enables the autonomous management of this service. An independently managed DNS allows for a refined configuration of zone parameters and makes it possible to propagate updates since the parameters can be managed internally quickly. A further advantage is the ability to backup DNS settings and configuration parameters since they are included in the persistent storage attached to the Docker container according to the presented use case. To install this service, one can use a docker image containing a server DNS, e.g. "bind" [16], then open and expose in the Docker and firewall configuration files, the port 53 TCP (used for zone transfer) and port 53 UDP (used to answer client queries) and finally associate a storage space with persisting the DNS data. In order to achieve a configuration that respects the canons of high reliability and high availability, it is necessary to configure a master DNS in the main availability zone and a slave DNS in the secondary availability zone.

The Mail Server

Similarly to the DNS server, a Docker-based mail server should also be configured to allow complete management of the e-mail transit in our infrastructure. In order to reduce false positives of anti-spam software and enable reliable message delivery, mail servers must be authenticated through Domain Keys Identified Mail (DKIM), Sender Policy Framework (SPF) and Domain-based Message Authentication (DMARC). The DKIM key is added to the DNS file zone as a TXT record. It ensures that no messages going from server to server are tampered with and that messages can be identified. SPF authentication works by specifying the number of allowed IPs that can send e-mails from a specific domain. The domain manager can add a file or record on the server that tells the receiving server which domains are allowed to send e-mails. DMARC builds on SPF and DKIM to further validate e-mails by matching the validity of SPF and DKIM records. This

[16] A DNS server with a web interface and preconfigured for Docker: https://hub.Docker.com/r/sameersbn/bind/

allows to set policies and get alerts generated in case DMARC validation fails. There are various Docker images that allow you to build a mail server, for example Poste.io[17], Mailu[18], Postfix[19] and many others. Once chosen the image one wants to use, it is needed to create a Docker container that uses it, and then one has to expose all the ports necessary for the service to work. Finally, setting up a relay mail server in the secondary availability zone is advisable to avoid losing mail if the main availability zone goes offline.

WAF

Modern cyber threats exploit vulnerabilities and technologies that are increasingly effective and complex, so we need to protect software applications and equipment within our infrastructure to the best of our ability. There are attack patterns used by hackers and bots that can be identified by software designed to defend IT infrastructures. A WAF is a Web Application Firewall, i.e. a firewall that works at level 7 of the ISO/OSI model [162]. This application helps defend our software against SQL injection or cross-site scripting attacks, as it can also analyse the HTTP requests that clients make to our services. For example, it filters specific IP addresses by creating blacklists. These devices have undergone an evolution in time and might be classified into three types:

The first one can detect malicious pattern matches and use whitelists and blacklists to monitor traffic and cyber attacks.

The second one can automatically generate whitelists of acceptable request patterns, i.e. not considered dangerous for computer applications; the inconvenience of this generation is that human intervention is still required to verify the correctness of the lists automatically generated by the system.

The third one, where threat detection is based on logical rules, represents the most modern generation and combines the technologies of the second type WAFs to which they add real-time packet analysis with a categorisation of attacks based on boolean logic. The most interesting feature of this type of WAFs is the proactive defence of applications, capable of locating and identifying vulnerabilities in the services we provide before an attack occurs.

Backup

A final aspect that needs to be configured concerns backup policies. The backups must guarantee that they can be restored quickly, minimising data

[17] https://hub.docker.com/r/analogic/poste.io/

[18] https://hub.docker.com/u/mailu/

[19] https://hub.docker.com/_/postfixadmin

loss in the event of disasters and protecting as far as possible against cyber-attacks [163]. Various parameters must be defined for data backup in the architecture described, which differ according to the service analysed. The first configuration parameter is the time interval between one backup and the following. The second parameter to configure is the retention time for a backup before its removal from the system.

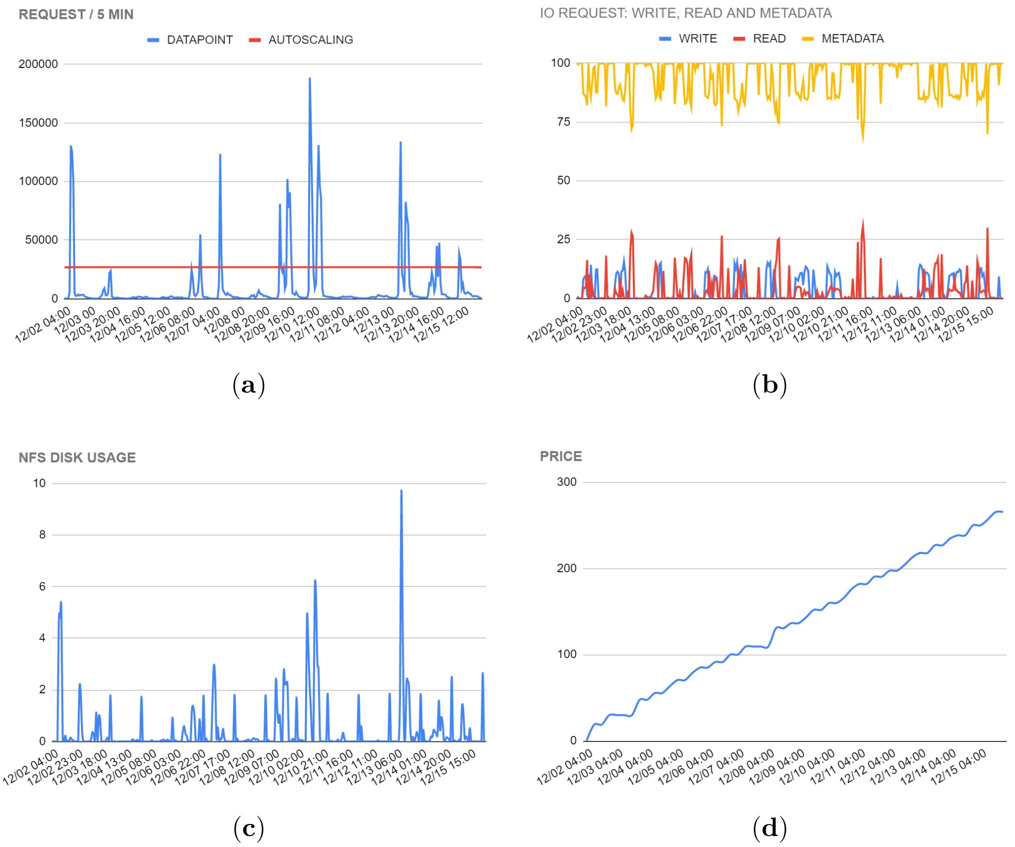
Periodic and incremental backups are necessary: that means storing only the data that has really changed since the previous backup on disk. It is advisable to backup the database, the shared network file system disk and the system disk used by the virtual machines. A proper value could be at least one backup per day and a retention time of at least three weeks to prevent threats that can damage data, such as ransomware attacks that aim to corrupt even system backups.

For the Redis database that manages the session, we did not set any backups because in our use case, the most serious situation that could happen is that the user is asked again to enter the username and password to access the system. We also set up a permanent backup made when the system was fully functional and kept for possible future needs. All the backups described above are encrypted, password-protected and cannot be accessed outside the infrastructure.

3.2.3 The case study

The case study infrastructure we propose has been operational for about nine months. During this time, we have obtained statistics and metrics that have allowed us to refine the model until it reached its current state. The red line shown in Figure 3.4 indicates the metric that induces the auto-scaling up (increasing in the number of server instances, fast process) and down (decreasing the number of server instances, slow process).

Every day about 1000 users use the web application delivered through the infrastructure. The users are divided into different roles, characterised by different activities, depending on the role. The average duration of the sessions is two hours, however the type of activity differs according to the role assumed: in some cases they are sporadic sessions, linked to information maintenance activities, of unpredictable duration, which can take place at any time of the day or night. In the other case, sessions take place at the same time, even in very large groups of users, and are characterised by intense activity for a period varying between half an hour and two hours. During this time, generally, a user performs operations that require a high number of disk input and output operations and a high number of database operations. We have extracted some graphs that we believe are significant on



(**a**)

(**b**)

(**c**)

(**d**)

Figure 3.4: Input output request and price chart: **(a)** Average number of requests received in the considered range of two weeks and a sampling time of 5 minutes, **(b)** Type of Input Output requests to the network drive, **(c)** Percentage of shared network file system usage, **(d)** Graph showing costs trend over 2 weeks of usage

the functioning of the infrastructure; in particular, to make the data easily readable and interpretable, we focused on a time range of 2 weeks. A more extended period of time would have made it difficult to highlight the peaks and trends in the use and load levels of the system.

The data collected shows that designing a horizontally scalable architecture is the best solution that allows the expected level of performance to be borrowed with operating costs. Figure 3.4 illustrates the data collected on the number of requests per time unit (see Figure 3.4a), the number of requests for read and write operations (see Figure 3.4b), the use of the shared network file system (see Figure 3.4c) and operating costs (see Figure

3.4d). Analysing the graphs, it can be seen that when there are peaks in the number of requests, there are corresponding peaks in the other quantities measured. This proves that resources are only used when really needed, guaranteeing adequate performance and optimising costs. The use of the infrastructure undoubtedly influences the cost, but the costs curve shows an almost flat behaviour in correspondence of an idle system. However, thanks to the use of auto-scaling technologies, the infrastructure can be scaled down at low activity levels. Keeping an IT infrastructure active in an "always-on" state, with a high number of servers capable of satisfying the requests of thousands of users, would involve an unnecessary waste of resources. An infrastructure such as the one described in this Section, which is limited to low-end resources, may involve a monthly expense of approximately 550 USD.

3.2.4 Statistics obtained while using the infrastructure

This section describes the experimental results we have collected. Figure 3.4 shows also various statistics regarding the load balancer and the shared network file system disk. Figure 3.4a shows the total number of requests received by the load balancer calculated in 5-minute intervals. As previously discussed, this is a significant parameter because it identifies the appropriate moments to activate horizontal auto-scaling policies. A too high number of requests could mean high latency input and output operations and do not burden the CPU, but which cause delays in the delivery of services to users, increasing the platform's response time. Especially in the first weeks, we monitored the data obtained from this metric very carefully, and on it, we based our auto-scaling policy. If the red line in Figure 3.4a, which represents

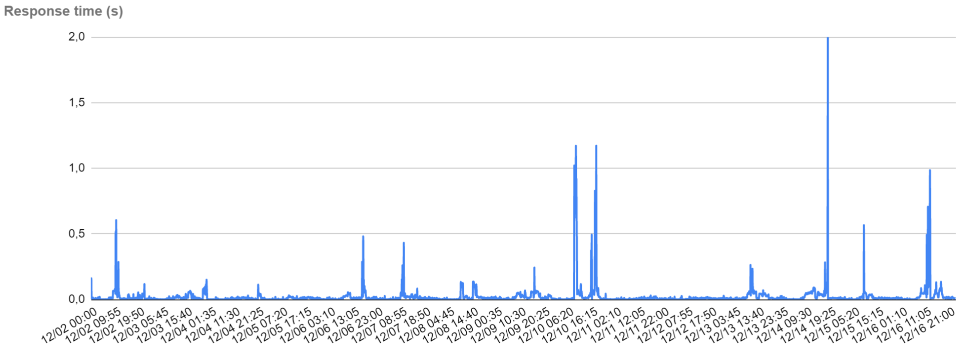


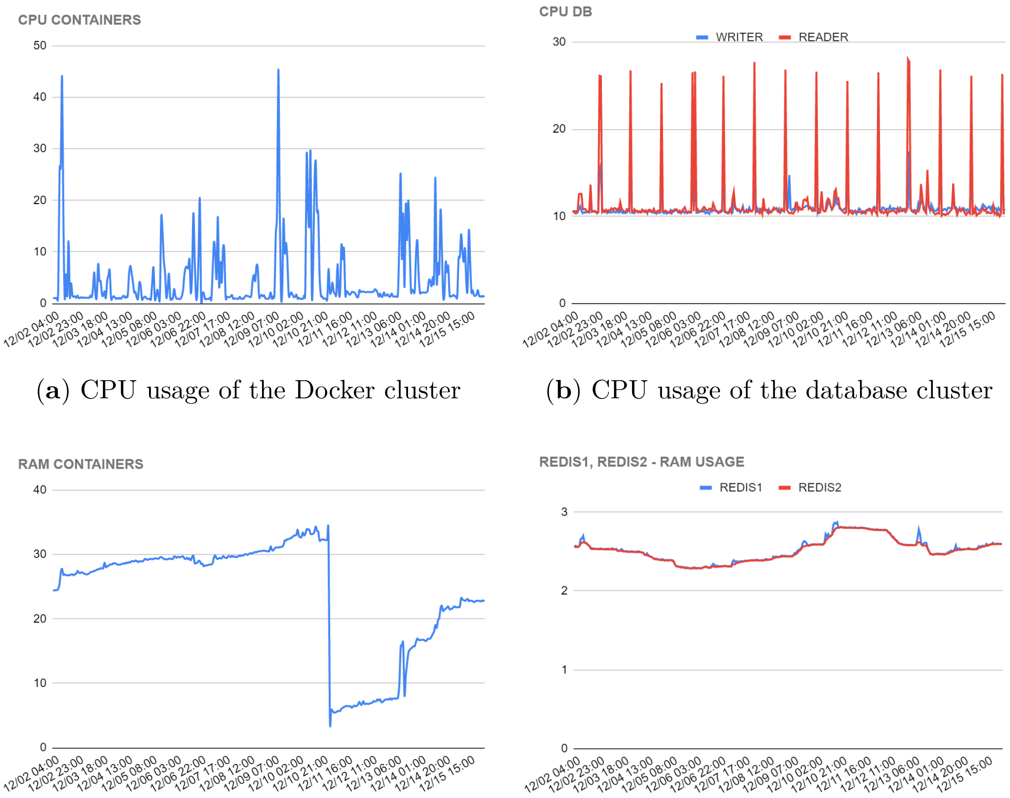
Figure 3.5: Average response time of the nodes (5 minutes interval).

the number 26,000, is crossed, new nodes for PHP backend management will be created. The addition of new nodes occurs directly proportional to the load encountered by the system; when the load level returns to a state of rest, they will be progressively removed the excess nodes until returning to the initial state where there are only two nodes: one on the primary Availability Zone and one on the secondary Availability Zone. Figure 3.4b shows the three different types of data access on disk: writings, readings and metadata access. In our case of use, the operations carried out more markedly are those of reading than those of writing. Figure 3.4c illustrates the utilisation of the shared network file system disk by the infrastructure in terms of percentage throughput. The shared network file system disk configured by us has high performance in reading and writing, and that is why, even in peak moments, 10% of the maximum allowed throughput has not been exceeded.

Figure 3.5 shows the graph of the response time in the considered range of two weeks and a sampling time of 5 minutes. As we can see, the response time is excellent, except for a few peaks where it reaches an acceptable value between 1 and 2 seconds. These peaks coincide with extreme stress periods for the infrastructure because many users are connected at the same time, carrying out intense activities, or a maintenance system procedure involving intense access to the persistent storage has been carried out. In fact, the peak observed in Figure 3.5 on 12/14 19:25 is also observed in Figure 3.4c at the same time interval. Meanwhile, the peaks on 12/10 after the 06:20 and 16:15 marks are also related to the peaks shown in Figures 3.6a (CPU load) and 3.4a (number of requests/5 minutes). The average response time is 21.7 milliseconds and the standard deviation is 76.3 milliseconds.

Figure 3.6 shows various statistics regarding the containers, the SQL database and the Redis servers. In particular, Figure 3.6a shows the average CPU usage levels of the containers that manage the PHP Web Server. As one can see, the average CPU load is not constant, but there are load peaks; these are due to user activity that occurs intensely only in certain time bands. Figure 3.6b shows the average CPU usage of the SQL database. Generally, the CPU usage of the database with our application is particularly low, the peaks that are visible in the Figure are due to a script, which runs every night at 01:00 AM and cleans the system from the activity that users carried out during the previous day by securely archiving data. In Figure 3.6c, you can see the average RAM trend of the containers that manage the PHP Web Server. Our application needs a modest amount of RAM: the average occupation is generally less than 40% of the 2048 MiB available on each node.

It is interesting to note that there are areas in the graph where the peaks



(**a**) CPU usage of the Docker cluster

(**b**) CPU usage of the database cluster

(**c**) RAM usage of the Docker cluster

(**d**) RAM usage of the Redis cluster

Figure 3.6: Average CPU and RAM usage of the clusters

are very close to zero: this is due to a system update that we have carried out and which involved the restart of the containers. The technique we use to carry out an update, or the implementation of a security patch in the code used by the nodes, is the following: we run two or more nodes with the updated code, which runs in parallel to the nodes that use the previous code version. After the new nodes are started up and operational, in about 5 minutes, the old nodes will gradually stop working and will be shut down and decommissioned. In this way, users will not notice any disruptions and the continuity of the service is guaranteed. Upon restart, the new nodes are restarted with a RAM usage of less than 10%.

In Figure 3.6d, you can see the average RAM trend of the Redis servers. Our application uses Redis servers only for user session management. The session is defined by a simple hexadecimal string, which occupies a minimal amount of memory, which is why even with more than 1000 daily users, we observe an average use of RAM of less than 3% of the 512 MiB made available to the server.

3.3 System implementation strategies for e-assessment

From the very beginning, the dissemination of knowledge and information has been one of the needs most keenly felt by humankind. Over time, the diffusion of ideas has adapted to technological development, taking on characteristics of structural maturity, speed of dissemination and social sharing of knowledge. Over the centuries, we have passed on knowledge orally to writing to arrive at modern times where digital information has become one of the predominant components. Nowadays, switching from traditional assessment methods to digital systems is an extraordinary leap forward in terms of modernity, respect for the student, the diversity with which different people express their skills and the enhancement of the teacher's abilities.

The COVID-19 epidemic has hastened the digitisation process in many industries, revealing several significant challenges that have hindered the development of digital technology in many domains. One of these is, without a doubt, skills testing, which on a digital platform is a more streamlined, reliable, and speedier procedure, allowing certain persons from disadvantaged groups to take the exam more readily. We discuss various ideas for facilitating online distant secure written examinations and how they may be implemented using the open-source system LibreEOL. Our experience in this direction started more than 20 years ago with our participation in a European Union initiative called *Leonardo 2* with a project called DASP[164, 165, 166]. From that embryonic system, we moved in the following years to a full-fledged system capable of conducting computer-based written examinations for various university courses. In 2015, the system was adopted by the *European Chemistry Thematic Network* (ECTN) to carry out the Echemtest® sessions for the certification of competencies in the various fields of Chemistry[145].

Starting from January 2020 with the advent of the Pandemic of COVID-19, the system has been completely re-engineered to afford the crucial and delicate task of carrying out written exams in a secure way from remote. The reorganisation of the platform was carried out under the following guiding principles: user-friendliness and optimal user experience, full adherence to open standards, respect for user data protection by always managing the minimum amount of data possible, maximum protection from all possible attacks, scalability and elasticity of the service to ensure adequate response times. We are confident that we have achieved our objectives in light of the data collected through the questionnaires submitted to students and thanks to the volume of examinations taken.

The Section is structured as follows: Subsection 3.3.1 outlines the architecture of the services forming the heart of the system, especially concerning its execution in a Cloud environment. We point out the open-source solutions available to achieve the expected efficiency of the system, currently implemented in a commercial Cloud environment. Moreover, it describes the logical organisation of the various user roles, the content relating to the organisation of courses and questions, the various modality to deliver exams and the available statistics that may help the teacher to monitor the students' performances. Subsection 3.3.2 describes the algorithms implemented to validate the user's activity while conducting the exam. This part is the most important that has been developed recently, which has required an enormous effort to try to reduce to a minimum the forcing and evident controls on the user, adopting technologies that are non-invasive for the user but at the same time warrant that the teacher can control how the exam is taken by the user, guaranteeing its veracity. Subsection 3.3.3 carefully describes the usability of the system, underlying the strategies adopted, inspired to the minimisation of operation and the transparency of the operations necessary to achieve a given task successfully. Subsection 3.3.4 presents in detail the results obtained, both in terms of the quantity of exams carried out and of the satisfaction measured from the users analysing the responses to the questionnaires. Subsection 3.3.5 presents the results of the Data Protection Impact Analysis (DPIA) applied to the LibreEOL[20] platform, which is highly relevant given the very invasive strategies adopted by many commercial e-assessment solutions.

3.3.1 The System Architecture

The system architecture is distributed on a cloud platform through a complex subdivision of scalable and redundant microservices that guarantee very high reliability to service availability offered to users. The overall architecture is divided into different availability areas, and in particular, we have configured two availability zone, zone A and zone B, two regions far enough away to guarantee disaster recovery but still in the surroundings of the city of Milan. This means each microservice, like the file system or the database, is replicated across multiple areas within the Milan area. This architectural model assures us that LibreEOL functions can continue to be provided to

[20] The project URL is: https://libreeol.org.

users even in case of faults. A synthetic schema of the system architecture is shown in Figure 3.7.

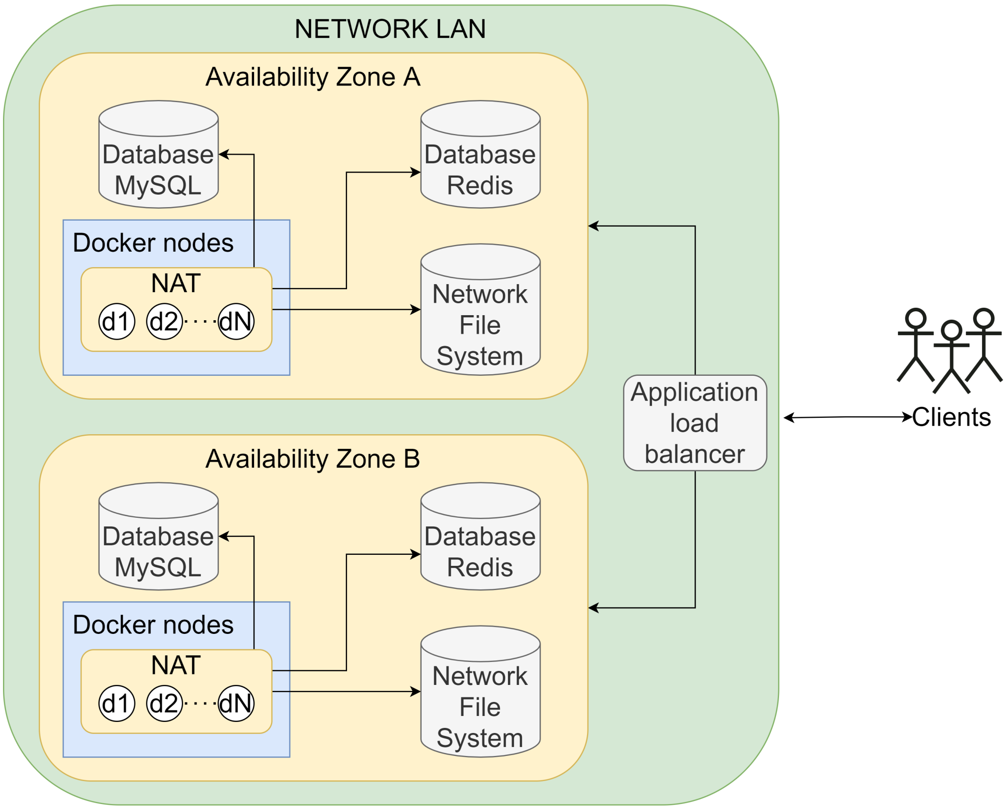


Figure 3.7: Schema of LibreEOL system architecture

The client requests are managed by an Application Load Balancer[143, 141], which is a significant application operating at level 7 on the ISO/OSI stack. Its task is to sort the number of active connections among the various Apache nodes, distributing the load over them in the most homogeneous way possible. The algorithm used for load sharing is Least Outstanding Requests (LOR) which assigns new HTTPS connections to nodes with fewer outstanding requests. Nodes are Docker containers running within a cluster managed by an orchestrator, which ensures that these are always properly functional and efficient. If a container no longer responds to requests or has saturated the RAM because of a memory leak, it is immediately restarted. The containers are all identical, running the Ubuntu Linux OS and an Apache webserver. HTTPS requests reach one of the apache servers and are then handled by returning, for example, the LibreEOL homepage. The container cluster uses horizontal autoscaling. Autoscaling increases or decreases the number of active containers in the Docker cluster based on CPU utilisation percentage and makes user requests get handled correctly even during peak hours (i.e. when there are many exams simultaneously). If the CPU usage reaches a global average value of 60% on all the containers, the newly incoming requests will be routed to a new container created to empower those already present. As soon as the exams finish, the CPU goes back to lower usage levels and therefore, the autoscaling brings the number of active containers progressively back to a lower quantity, reducing platform costs and keeping the system efficiency at the highest levels all the time. In order to preserve the consistency of the data saved on the storage, the system uses a Network File System distributed throughout mount points individually hooked by each container. In this way, data written on container 1 located in availability zone A can also be correctly processed by container N in the availability zone B. Correct data redirection can only be done in real time by tracking the PHP session. A user could start his interaction with the system on node 1 of Availability Zone A and continue to use the contents through another node thanks to the load balancing performed by the load balancer. A Redis database is used to manage the user session, to assure consistency and reliability. That allows making computational processes' movements transparent to the user between one container and another. Further strength for the architecture created is the possibility of distributing the computation cost due to the SQL requests on the database. The database is a cluster of MySQL servers created through the use of a horizontally scalable architecture that includes a Writer (Master) node and N Reader (Slaves) nodes. The cluster is replicated on both Availability Zones to guarantee fault-tolerance and High Availability (HA).

LibreEOL is a Web App structured with a PHP-based backend, a JavaScript-based FrontEnd and MySQL is used as RDBMS. We have always tried to adopt standard solutions in developing the code, preferring HTML5 native approaches and respecting good practices in Human Computer Interaction and User Experience. So, the languages used to create and develop LibreEOL have been JavaScript, PHP, SQL and Bash. A student's test execution is not needed to download additional components or install other software to the web browser, except technical cookies; this guarantees immediacy and ease in using the contents without introducing further overhead to the computer.

LibreEOL is a web application based on the Model-View-Controller (MVC) architectural pattern [167] that allows the management of electronic examinations for individual courses and enables the collection of statistical data that can be used to improve the quality of teaching.

User organisation

Users and content are divided into groups (i.e. Universities) and subgroups (i.e. Departments). Users are divided into the following roles: students, examiners, teachers and administrators. The examiner role is designed to support the teacher in conducting the examination sessions of a course. This role does not have any authority over the information content of questions relating to the course itself. The operations that a user can carry out within the system are determined by the role to which they belong. Each profile is associated with different interfaces with functionalities consistent with the operations allowed within the specific role. It is possible to connect it to an Identity Management service to assign roles automatically, to enhance the platform's usability.

Content organisation

Each teacher can create one or more courses made up of a set of questions classified according to the level of difficulty and topic.

Several teachers and examiners may be associated with each course. The teacher is allowed to express his questions using a set of 11 different types: Multiple choice, Multiple choice with a score, Multiple responses, True/False, Yes/No, Essay, Hotspot, Numerical, Text match, Text match case sensitive and "QR code". While Multiple Choice supports one answer as true and the others as false, in Multiple Choice with a score it is possible to assign a positive/negative floating point value to all answers. The Hotspot type allows selecting a portion of the image to indicate the answer. The Essay type allows answering by entering free text. A Numerical question allows providing a numerical result (an answer with a tolerance of 3% of the value given by the teacher is accepted). The "QR code" typology allows the upload of the images acquired with a smartphone of a free-written paper, as what happens in in-person examinations.

Exam organisation

Each teacher can create examination sessions according to the settings they have configured. In this way, it is possible to create complete examination sessions or mid-term evaluation tests covering a subset of the topics. Creating an exam call requires the teacher to indicate which exam settings to use. These include the possibility of activating or not activating the algorithms for conducting examinations remotely. Once the exam session has been created, students can register and eventually take some demo tests provided by the teacher. On the day of the exam at the scheduled time, the teacher will provide the password to allow students to access the exam and take the test. If the exam session is held remotely, the teacher can monitor all the students through the "Live view" tool, which allows him/her to observe all the students taking the exam.

At the end of the examination, the teacher will access a panel summarising the tests taken and listing the candidates. This panel allows the teacher to view the individual tests and set the grade, either by accepting the value proposed by the system or by defining it, as is the case with "QR code" or open-ended questions (Essay).

Once the exam has been completed, the teacher archives the exam. The results could be sent by email to the students if the teacher enabled the option while creating the exam.

Statistical analysis
The system provides various statistics and reports, and it is in this ability, one of the central added values of conducting examinations on an electronic platform is hidden. The teacher can monitor the frequency of correct answers among the various students within a course and between courses of different years. Careful evaluation of these indicators allows the teacher to calibrate the content delivered and the mode of delivery.

Administrators obtain a distribution of the tests by day and month, enabling them to proactively calibrate the resources made available to the system to carry out horizontal scaling (increasing active resources initially in anticipation of high usage peaks). It is also possible to plan vertical scaling of the infrastructure (switching to more powerful hardware in anticipation of high usage peaks).

3.3.2 Algorithms implemented to validate students's activity

The central aspect that deserves serious consideration when one decides to take exams remotely is to ensure that the test measures the actual skills, quality and quantity of a student's knowledge. Furthermore, it is also essential to ensure that the test has legal validity, i.e. it is necessary to prevent the student from receiving suggestions or information from the outside. The technology developed by us is divided into various components that analyze the ambient audio, the video stream captured by the webcam, the mouse movements and the keyboard. In the design phase, we took particular care of efficiency, and we intended to focus on a code capable of running on inexpensive devices with low computational power. Students often own old laptops and smartphones. Therefore, our technique is completely based on geometry and basic mathematical operations that make our algorithm fast and executable on most of the hardware used by students. Neither a dedicated graphics accelerator is required. All the algorithms presented have properly been run on dual-core CPUs produced since 2010.

Algorithm for video stream analysis

Our objective is for the student to take the test without being distracted or receiving illicit assistance. The system requires a camera and microphone to verify how the test is being carried out remotely; thus, the test may be conducted from remote locations. The images captured by the camera represent sensitive data and are rich in information. We built our algorithm to keep the number of photographs and sounds saved on the server to a bare minimum and keep the server's information for as little time as possible. Each student had to provide permission for this sensitive data to be processed before taking the exam.

Identification of the face movements

A very significant aspect for identifying a student misbehaviour concerns the identification of video distractions, intended as too wide and sudden movements of the head. In Figure 3.8 the 68 landmarks produced by the neural network that we use for this function are shown. Neural networks perform the task of automatically classifying classes of objects in the most varied fields of application [17, 16]; in our specific case, the network has been developed in such a way that it can recognise the landmarks of the user's face.

Starting from analysing the subject's face and its movements in space, we can assume with high reliability whether his/her conduct has been inappropriate. This is obtained using a neural network that provides landmarks extraction and detects 68 points on the subject's face representing them by X, Y coordinates in a Cartesian plane.

In Figure 3.9(a), we can graphically analyze the algorithm computation. The photograph shows explicitly a student who is staring at the monitor. It is possible to notice that the face is contained within two red-bordered rectangles, an inner and an outer one. The blue rectangle represents the polygon tangent to the outermost points of the student's face and represents our centre of gravity. These rectangles represent the tolerance of the system. The algorithm computes as follows: in the beginning, the neural network computes the landmarks, and the algorithm evaluates the relative tolerance threshold so that the student can move, breath, yawn. These two rectangles are periodically calibrated over the student's face in order to ensure that they are always consistent with his/her position. It is, in fact, taken into

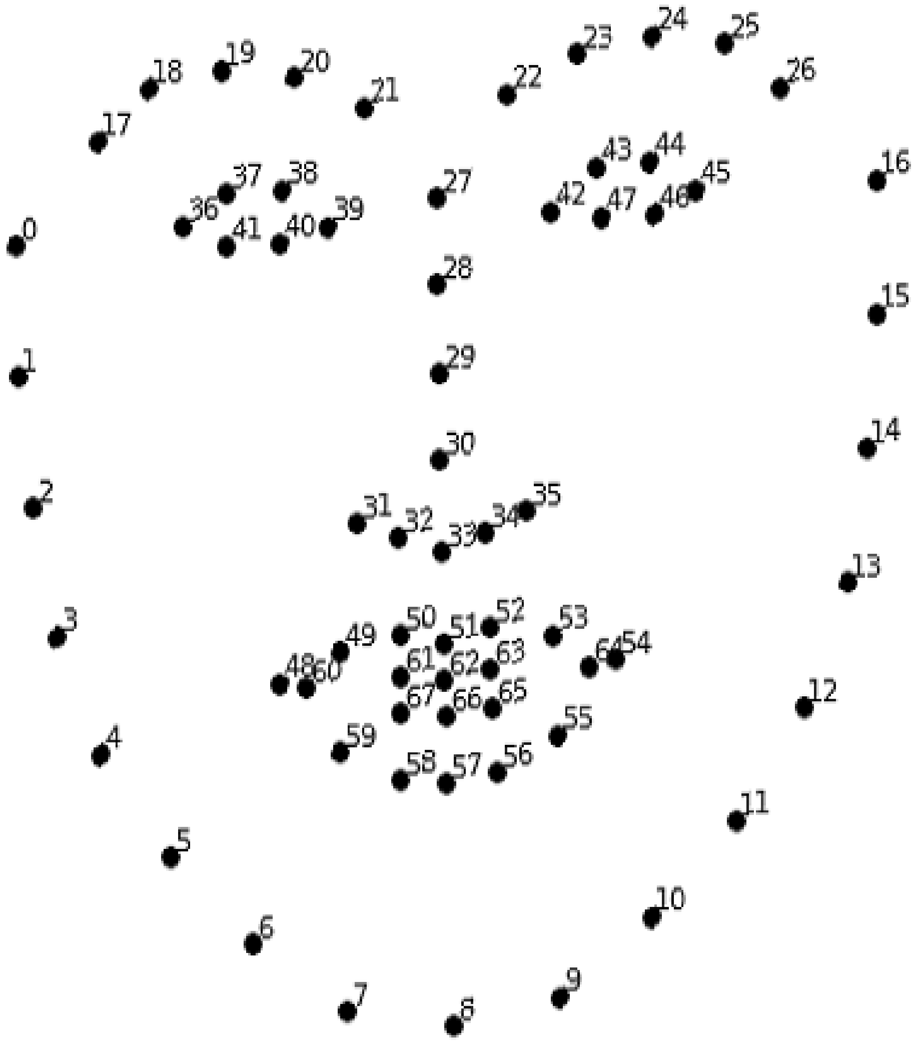
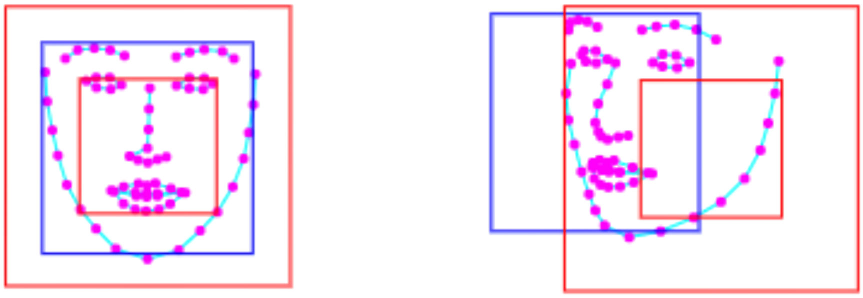


Figure 3.8: The identification of the landmarks computed by the neural network from the collected webcam flow of images.

account that during an exam that can even last 2 hours, the student can move slightly to get more comfortable. If the learner turns away from the screen and does not look at it, the blue square will touch or even exceed the boundaries set by the red rectangles' corners. In Figure 3.9 (b) an example of this behaviour has shown. In that case, a photograph of the event is taken, and a distraction is counted.



(**a**) Good positioning (**b**) Excessive rotation of the face

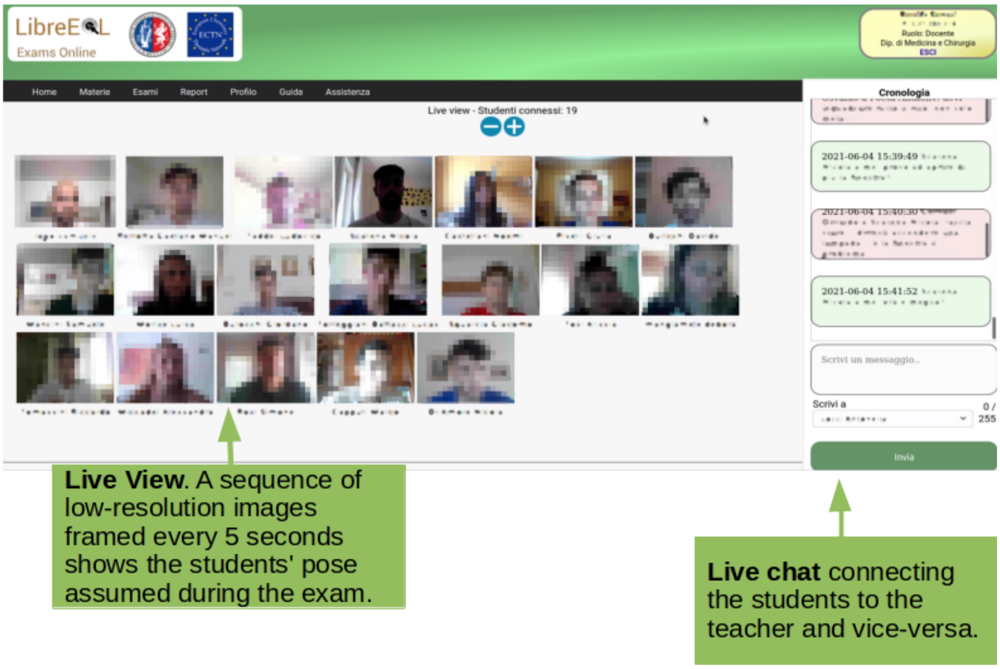
Figure 3.9: Areas computed by our algorithm when the user is properly positioned in front of the webcam (a) and when she/he turns the head, causing a distraction to be detected by the system(b).

Monitoring of all students taking the exam

Low-resolution images are extracted from the webcam images at 2-second intervals and displayed on the exam's associated *Live View* page so that the teacher can monitor the progress of the exam as a whole. Based on the images shown, the teacher can send Chat messages to individual students or all the participants. This tool has an enormous impact and power on the teacher and allows them to establish important contact with students during the exam. The images shown in the "Live View" panel are also displayed on the page dedicated to each student, showing the various video distractions, any audio recorded, the times the client was disconnected from the server and for how long, and the number of times the exam page was reloaded. This page is essential for assessing the correctness of the test taken by the student. Figure 3.10 shows the Live View environment described above while the exam is in full swing. Figure 3.11 shows a student's summary page with indications of the events that characterised the test and allow the teacher to make a final decision on the correctness of the test.

Algorithm for analysing environmental audio

Through JavaScript code, we obtain the audio stream from the microphone. The audio is analysed with a sampling rate of 48K times per second and a bit rate of 64 KBps. The audio we capture is stereophonic. We first calculate the average value of the ambient sound pressure. If we detect a peak, i.e. a sound with an intensity more remarkable than the average of the last period, then, for the duration of the event, we record an audio file that in this first phase will be stored in the RAM memory of the computer. The audio file is recorded in OGG format. This is an open format, open-source and released



Live View. A sequence of low-resolution images framed every 5 seconds shows the students' pose assumed during the exam.

Live chat connecting the students to the teacher and vice-versa.

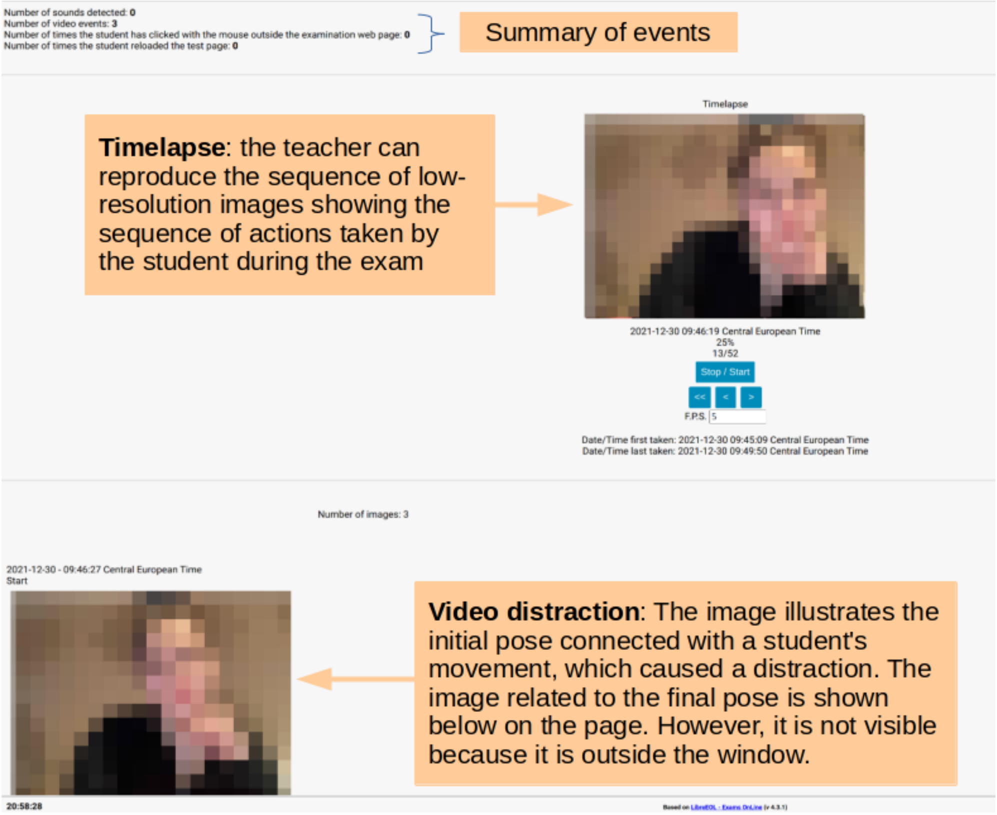
Figure 3.10: Annotated panel showing the Live View of a running exam (the image has been altered to protect the personal data of the persons in the picture

under a BSD license.

The OGG format can store audio encoded using the lossy compression algorithm 'Vorbis', also released under a BSD licence. The recording phase, monitoring of ambient sound pressure and coding of the audio flow, takes place entirely on the client, i.e. it is carried out by the student's computer, with a low expenditure of computational resources. An example that graphically shows the variation of the ambient sound pressure during a speech can be seen in Figure 3.12. The system then checks the length of the recorded audio file. If it is less than 1 second long, it is not uploaded to the server. If, on the other hand, its duration is greater than or equal to one second, then it is securely stored in the system, and the audio event associated with the student is recorded in the database. Monitoring takes place throughout the examination.

Mouse and keyboard monitoring algorithm

To prevent students from obtaining content and information on the Internet, we built algorithms that ensure that they only sit the test with the knowledge



Number of sounds detected: **0**
Number of video events: **3**
Number of times the student has clicked with the mouse outside the examination web page: **0**
Number of times the student reloaded the test page: **0**

Summary of events

Timelapse

Timelapse: the teacher can reproduce the sequence of low-resolution images showing the sequence of actions taken by the student during the exam

2021-12-30 09:46:19 Central European Time
25%
13/52

F.P.S. 5

Date/Time first taken: 2021-12-30 09:45:09 Central European Time
Date/Time last taken: 2021-12-30 09:49:50 Central European Time

Number of images: 3

2021-12-30 - 09:46:27 Central European Time
Start

Video distraction: The image illustrates the initial pose connected with a student's movement, which caused a distraction. The image related to the final pose is shown below on the page. However, it is not visible because it is outside the window.

20:58:28

Based on LibreEOL - Exams OnLine (v 4.3.1)

Figure 3.11: Annotated panel showing the summary information of a remote exam (the image has been altered to protect the personal data of the person in the picture

they have gained throughout their studies. The examination screen is enlarged and displayed in full-screen mode when the test is started. Any windows that may have been open are then covered by the application screen, which will occupy the entire visible area of the student's screen. At the same time, mouse monitoring is started; in particular, we want to prevent the mouse from leaving the examination screen and clicking on external areas, such as on a second monitor. To do this, we first install an event listener on the web page that will monitor the property "focus". This property tells us whether or not the web page is in the foreground, or whether the user is clicking or acting on elements in the background or on other monitors. The loss of focus, in fact, guarantees that the student has clicked outside the examination window.

If a student clicks outside the examination window, a picture of the event is taken, and the timestamp of the event is stored in a server-side counter.

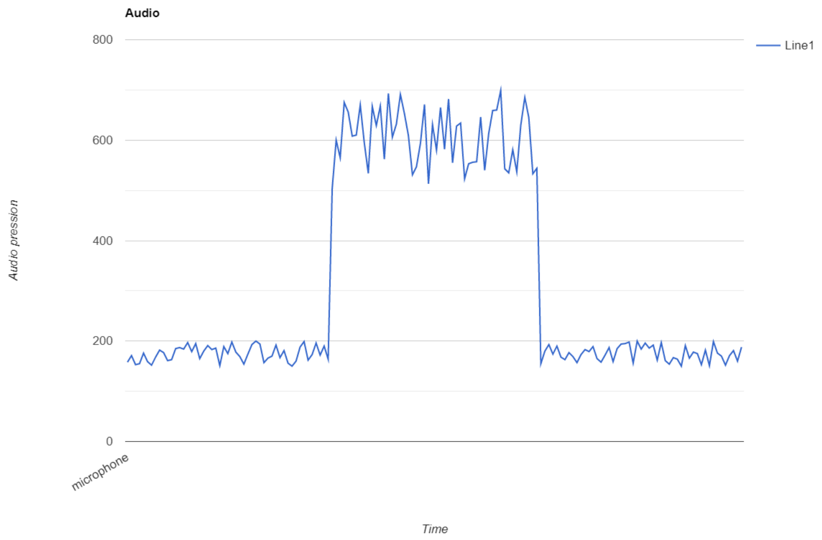


Figure 3.12: A sample audio monitoring graph

Before starting an exam, the teacher can define the maximum number of times a student can click outside the exam window. It is essential to provide a certain margin from our tests, and not be too strict on this aspect. We think it is correct to set the 'focus lost' parameter to 3. It is, in fact, possible that during the exam, the student has notification from the antivirus program requesting an interaction, or he has audio problems and needs to click on the headset control panel, or it is possible that he accidentally clicks out of the exam. In any case, once the occurrences the student loses focus on the page exceed the parameter set by the teacher, the exam is closed and the student is ejected from the session. In addition to this, a series of checks are carried out on the keyboard and on the mouse's interaction with the page. These controls prevent the copying/pasting of text to and from the examination page.

Listing 3.1: Mouse control code

```
$(document).mouseleave(function () {
  console.log("FocusLost!");
});
$(document).mouseenter(function () {
  console.log('FocusOk');
});
$(window).blur(function () {
  if(!document.activeElement instanceof HTMLIFrameElement ){
    console.log("FocusLost!");
  }
});
```

Furthermore, the page printing functionality, the screen recording functionality and the print screen functionality are inhibited.

3.3.3 System's usability

The system's usability is one of the essential aspects of designing a software application. A modern web application must be easy to use, must not require lengthy explanations of how the graphical interface works, be respectful of minorities and consider users' disabilities. Furthermore, applications of this type should not require any special skills on the part of the users. We have also considered the different types of users, and the disabilities that people using the system may have.

For example, some students have difficulty using the mouse or carrying out drag and drop operations, so we have designed an interface that does not require the mouse, drag and drop movements or operations requiring complex motor coordination. A low percentage of students also have Specific Learning Disorders (SLD) to varying degrees. An e-assessment system allows the teacher to offer these students a test tailored to their needs, including more time for the test and questions formulated to facilitate comprehension or mathematical calculations.

When developing the user interface, we took the importance of colours seriously, so we chose a colour palette that would not cause confusion or problems for people with colour blindness. According to the most recent estimates, 4.2 per cent of the world's population is colour-blind, and in Italy itself, there are 2.5 million people affected by this condition[168, 169, 170]. In the light of these considerations, all the messages that the graphical interface proposes to users have been calibrated on an appropriate shade of colour compatible with Protanopia (insensitivity to red), Protanomalia (insufficient sensitivity to red), Deuteranopia (insensitivity to green), Deuteranomalia/Teranomalia (insufficient sensitivity to green), Tritanopia (insensitivity to blue, violet and yellow) and Tritanomalia (insufficient sensitivity to these colours). In addition, we have taken particular care to ensure the compatibility of the system with the main web browsers (Google Chrome, Microsft Edge, Mozilla Firefox and Opera) and also compatibility with old and resource-poor computers; this has involved taking care to minimise the computational impact of the application on the client's hardware resources.

In many disciplines such as mathematics, chemistry, physics or the humanities, it is crucial to write by hand on sheets of paper, carry out calculations better, prove theorems, and make graphs or drawings. To this end, a particular type of question has been created that allows students to securely upload photographs of the paper they have produced, through a QR Code using a smartphone. The smartphone is necessary to take high-resolution photos of the paper produced by the student. Once the camera has framed the QR Code, this is decoded and the web page dedicated to uploading the images appears. Each QR Code uniquely associates the paper produced by the student with his examination.

We submitted a questionnaire to the users, allowing them to voluntarily express a series of considerations on the qualitative assessment and problems encountered during the use of the system. We obtained a satisfaction index of 81.4% of the sample, indicating an important general appreciation level. Thanks to an agile programming approach and constant interaction between developers and end-users, the system's usability is continuously improved.

3.3.4 The results obtained

From 21 February 2021 to 21 September 2021, we kept a direct communication channel open between the platform and the students by delivering an online questionnaire. That allowed us to collect valuable information directly from the primary users of the platform. The students voluntarily participating in the anonymous questionnaire were 1423, expressing their opinions and views on the system we have developed. The questionnaire consisted of eleven questions divided into eight sections. The preliminary questions were technical and aimed at understanding the tools used by the students to access the examination platform. The first question asked which browser the student was using, while the second asked about the operating system. The results show that 79.4% of the students use Google Chrome, 8% use Firefox, and 5.8% use Safari. The remaining 6% of the students use less popular browsers such as Brave, Chromium, Microsoft Edge, Internet Explorer, Opera and Samsung's Browser Internet. The second question got the following responses: 62.5% of the students use Windows 10, 11.9% of the students use macOS, in the third position we find Windows 7, used by 8.2% of the students. The results obtained from these first two questions are highly significant as they show us that the development environment on which to program and develop the software code must first be tested on the Google Chrome browser. As far as the functionality of the application we have created is concerned, the operating system is of less importance since we do not use any particular APIs that would make Windows rather than other systems decisive. In the preliminary software's development, we have tried to devise a code that runs on as many devices as possible.

The next question asked the students whether they experienced particular events during their examination, such as unexpected notifications, irrelevant error reports or unexpected behaviour. 84% of the students answered that they had not experienced any particular events, while 16% answered that they thought the system had behaved unexpectedly. These students, in particular, were asked, employing an open-ended question, to describe the event that had occurred during the examination session. We collected a total of 228 answers. Some of them explained that the cause of the problems they experienced was slow internet line at home, which delayed the loading of images and texts. Others described that notifications from the operating system, not caused by LibreEOL, had appeared in the foreground and distracted them, such as notifications of antivirus software, new emails or system updates. The most interesting answers are related to the algorithm we programmed for face tracking. In fact, some students reported that they had received warnings following small movements of the face that are part of the natural behaviour in concentration situations. These latter responses were precious to us, as we immediately took them into account and made adjustments, corrections and calibrations to the system to address these issues. In particular, we calibrated the face at the start of the test and at regular intervals to ensure that the face tracking also took into account the natural posture adjustments that people make when they sit in front of the computer for one or more hours. In addition, we have made sure that the distraction notifications we give to students are even better calibrated. These notifications are important because they allow the student to be warned if the system detects incorrect conduct. Nevertheless, it is essential to consider that the number of notifications and warnings that an automated system can give to a person while he or she is taking an exam must be wisely limited. There may be situations that are not caused by a person's failure to comply with the most basic behavioural rules but are caused by other problems, such as a webcam with poor resolution, or poor room lighting, or an incorrect framing of the face that could generate a high number of alerts. In this case, we stop sending notifications to students when a certain threshold is exceeded. According to our tests, the optimal maximum number of alerts per person is 6. More warnings could only be counterproductive and lead to the person becoming impatient. It should also be borne in mind that the teacher is responsible for monitoring the students' video streams throughout the exams. So, the teacher can also view the counters for the number of facial distractions, and the number of audios detected and analyse them in real-time via a secondary browser window. Thanks to these measures that we have implemented in the platform, we have not received any further reports of inappropriate warnings given to students.

The next question asked the students whether they experienced any additional discomfort during the examination. This question was answered by 100% of the students. Among them, 75.9% answered that they had not experienced any discomfort, while 24.1% (343 students) stated that they had experienced further problems. These students were then asked to express what kind of discomfort they had experienced while using the system through an open-ended text field.

The students answered that they had experienced anxiety were 76, while 12 students stated they had experienced stress during the examination. Students who experienced anxiety represented 5% of the total number of students interviewed, while those who experienced stress represented 0.8% of the total. Their valuable evaluations and the relevant advice provided to us by this small percentage of students contributed to an highly significant improvement in the usability and characteristics of the system. We realized that to reduce the students' stress and anxiety levels, we needed to implement a series of algorithms to help them feel less alone. The algorithms we introduced provide visual and textual feedback to the student, reassuring them that they are progressing well.

The first algorithm relates to the auto-saving function of the exam, which occurs automatically at periodic intervals. This function allows the student's answers to be stored in the system's database before the exam is finally handed in. Auto-saving allows the exam to be retrieved if the student's computer crashes, the power goes out, a network disconnection occurs, or other problems prevent the exam from continuing. Each time the exam is auto-saved, a small green text appears to the student informing him that the test has been correctly saved in the database.

The second algorithm implemented carries out a periodic check on the photographs that are loaded into the system relating to the QR codes displayed on the student's exam page. When the system detects that the student has uploaded a photograph via QR code, a photograph relating to an exercise that he had to carry out on paper, the system will inform him that the photograph has been saved correctly on the exam page and not only on the smartphone screen.

The third algorithm we implemented allows direct communication via chat messages between the student and the teacher. One of the reasons that caused the most anxiety was the lack of communication between the teacher and student during the exam. This feature allows students to feel part of a community and not just an individual mechanically assessed by a software programme.

The next question asked for an overall assessment of the system's quality: 81.4%, i.e. 1158 students, said that the system has a high level of quality and praised its features; 10.3% of the students considered the software invasive from the point of view of privacy; 8.3% of the students felt that the system was not appreciated and that it should not be used.

Thanks to the data collected, we received a confirmation of the goodness of the solutions adopted to manage better and protect personal data. In fact, through application's design, only a minimal amount of sensitive data (photos, sounds, text of chats) are collected. They are saved on the file system only for the time needed by the teacher to carry out the examination procedures.[21]

We have met with student representatives several times, collected suggestions and objections, and publicised the guides we have produced and all the documentation available on the website. We have explained to them the techniques adopted to protect personal data, receive their active collaboration and calm the students' spirits.

All facial detection and tracking algorithms only work on the student's computer, and no biometric data is processed or sent from the clients to the server.

The last question was an open-ended one. We asked the students to express any criticism or constructive proposals for improving the software. A total of 248 people responded, most of the criticisms and suggestions related to topics we discussed earlier, such as more effective face calibration and the request to have visual feedback on events that occur during the exam (such as self-saving) and the request to have the possibility to speak even only verbally with the teacher. These were all valuable suggestions that motivated us to refine and release changes that had already been planned and developed in the test environment, such as the chat environment, real-time calibration to adapt the control system to the natural movements of the person, refining the online documentation, enhancing support via social networks and email.

[21] An exception to this procedure is represented by the information collected for students who have been excluded from the exam, either by the teacher or the system. This information is saved encrypted on a file system outside the web domain for 130 days so that, in the event of legal proceedings, the organization will be able to exercise its defence by producing this evidence.

3.3.5 Data protection issues

The European Union adopted in 2016 the General Data Protection Regulation (GDPR)[22]. A processing operation is likely to pose a high risk to the rights and freedoms of data subjects because of the systematic monitoring of their behaviour, the large number of data subjects whose sensitive data are perhaps processed, or even because of a combination of these and other factors. In such cases, GDPR obliges data controllers to conduct an impact assessment before commencing the processing, consulting the supervisory authority if the technical and organizational measures they have identified to mitigate the impact of the processing are not deemed sufficient, i.e., when the residual risk to the rights and freedoms of data subjects remains high.

We have drafted the Data Protection Impact Analysis (DPIA) for LibreEOL using the program made available by the French Data Protection Authorithy, CNIL[23], a multilingual open-source distribution made available for all Linux distros and Windows 10 under the Windows Subsystem for Linux (WSL) as appimage[24].

Following the document's structure proposed by the CNIL, the operations involving the management of personal data (images of the student, of the room in which the examination is conducted and environmental audio) have been detailed, specifying the technical solutions adopted and the associated risk. This phase of analysis allowed us to adopt appropriate changes to the code of the proposed application, aimed at minimizing the amount of data collected, the length of the data retention period before deletion and the methods of secure and encrypted storage of data that must be retained for more extended periods.

In some situations, choices had to be made, in the sense that it was necessary to give up handling data that would have allowed better support to users but could give rise to ambiguous interpretations on the appropriateness of such processing.

The outcome of the impact assessment was that the risk was negligible.

[22] The General Data Protection Regulation is a European Union regulation on the processing of personal data and privacy, adopted on April 27, 2016, published in the Official Journal of the European Union on May 4, 2016, and entered into force on May 24 of the same year and operational as of May 25, 2018. It is known as Regulation 2016/679.

[23] www.cnil.fr

[24] https://appimage.org

3.4 Conclusions

In this Chapter, we discuss techniques for efficiently delivering web applications that provide an adequate level of service while optimising operating costs. We believe that what is presented in our use case can help organisations efficiently release their applications, guaranteeing the respect of personal data and the durability of the stored information. The solution presented lends itself to be implemented with different Cloud platforms and is based on open solutions, limiting the lock-in phenomenon by various Cloud providers. This paper also presents a possible implementation of a service that guarantees high fault tolerance and the ability to scale its computational power according to the number of incoming requests from users. We have discussed an efficient and modern model on which applications running in the cloud can be based. The architecture is made up of two availability zones in an Active/Active (50/50) configuration that are meant to support each other. In fact, if there is a technical problem in the main availability zone, the secondary availability zone will be perfectly capable of operating and ensuring the continuity of the service provided. The comparison between the various graphs that have been widely discussed in Sections 3.2.3 and 3.2.4 makes clear the goodness of the choices made; in fact, the data relative to the response times measured for the application in the same temporal range shows excellent values with peaks that are associated with events of exceptional load on the server, and even though they are considerable, they do not exceed two seconds. That leads us to conclude that the infrastructure associated with the presented case study has the optimal characteristics to allow access to the application, which is fluid and constantly able to serve users with optimal service levels.

We have created a platform for secure remote online exams that respects personal data and the users' primary needs. Security by design and privacy by design were the foundations of our work. During the COVID-19 pandemic, the platform proved helpful, with the University of Perugia adopting it as the official tool for conducting exams remotely. The high level of participation in the questionnaire we administered, together with the high level of acceptance and the impressive number of examinations taken in two years (over 50.399 in 2021 and 54926 in 2022), are, in our opinion, the best proof of the validity of our work.

Thanks to the difficult period that we have experienced at world level, we believe that, at least in our university, an extraordinary cultural revolution has taken place, driven by students and led to a profound transformation among the teaching staff. They have experimented mainly with an efficient and effective way of evaluation, made possible thanks to a re-thinking of how evaluation is carried out. We are confident that this acquired patrimony will remain over time and facilitate the teachers' evaluation work and allow the students to expound their competencies and knowledge more completely.

Our future work on the platform will be inspired by a desire to increase support for teachers and students by identifying new forms of questions and answers that make the assessment process as natural, intuitive, and immediate as possible. We also wish to increase support for those who experience difficulties in their daily lives, for example, by implementing screen-reading assistants or voice assistants for entering answers to questions via voice feedback provided by the student.

Chapter 4

Optimising the smart city-scale Cloud-Edge-IoT infrastructures

4.1 Introduction

The "Internet of Things" (IoT) devices refer to a collection of objects that gain intelligence through the internet, exchanging data about themselves and gaining access to information from other sources [171, 172].

Generally, these devices have a number of well-defined characteristics [173]. First and foremost among these is *heterogeneity*, as these devices can be used in different network architectures, and can be based on different hardware architectures. To date, no standard has been defined for the realisation of IoT devices. A second characteristic is *interconnectivity*, these devices must in fact be able to communicate and transmit information through a network infrastructure. As far as *energy consumption* is concerned, there are strong limits regarding the resources that can be used, the same can be said for the computational power that can be delivered by the chips. These devices are also *versatile*. Their main areas of use are often related to things or the monitoring of certain parameters in the environment, such as privacy protection or control the temperature of a room. Finally they are *scalable*, we can in fact create applications using IoT devices with tens or hundreds of simultaneously connected devices. These devices do not limit developers on the amount of interconnections that can be realised. The number of devices that can be used, and their status, can vary and is generally not predetermined by the developers.

Today's strategy is to make machines increasingly autonomous and intelligent[18, 17, 1], rather than highly reliant on human supervision[174, 175]. This has required interaction with a complex world, which necessitates the development of numerous external inputs on which to take decisions[176]. As a result, the Internet of Things has become a significant paradigm in a variety of fields, including manufacturing[177, 178], remote control[179, 180] and monitoring[181, 182], home automation[183, 184], gaming[137] and Virtual and Augmented Reality[185, 73].

Generally, a system that uses IoT devices employs a three-layer model, the lowest layer is called the *perception layer* and is in charge of acquiring data from sensors, the main objective being to collect information. Subsequently, this information transits to the *network layer* whose task is to efficiently transmit the information content across heterogeneous networks without losing data. Finally, we reach the *application layer*, which represents the highest level, within which we can generally find the software that utilises the data and presents it to users through graphical interfaces [186].

During the three-year period, the candidate dealt with a number of topics related to IoT devices, in particular a conference paper was published demonstrating how these devices are crucial for monitoring the flow of people inside buildings and shops [187]. This work was carried out at the peak of the COVID-19 pandemic in order to help the community control access at various time slots when they are open to the public.

IoT systems and devices are now widely used in various sectors. For some years now, telemedicine has started to use tools for remote diagnosis of patients' illnesses, or devices capable of monitoring people's vital parameters without the doctor having to physically reach the place where the patient lives. Some examples of such devices may be bracelets to be worn on the wrist, or devices to be placed on the patient's chest. There are cameras that can scan the face using artificial intelligence to recognise facial micro-expressions that can identify pain or agitation, or analyse a patient's sleeping behaviour. A second possible application of IoT devices can be environmental monitoring, this can be done inside a home, where we want to monitor parameters such as pressure, temperature, humidity or lighting level, or outdoors as in the case of sensors and detectors connected to control units and exposed to the elements [188].

4.2 IoT application areas

The interconnection of sensors and IoT devices creates a network capable of exchanging large amounts of data. This data must then be processed and analysed, and thus adequate computational power is required to process the data produced by the sensors in real-time. This problem can be approached with different strategies, for example, trying to optimise the transfer and thus the use of bandwidth that the IoT devices make towards the higher levels of the network hierarchy, and also optimising the use of the cache memories that these devices possess in order to send the data only when it is needed. All of this becomes even more evident when we think to the four major macro areas of application of IoT devices. These are the *Smart Home*, the *Smart City*, the *Smart Health* and the *Smart Agriculture*.

When we talk about Smart Home, we mean homes that optimise energy consumption according to the habits of the people living in them, and according to the quality and structural conditions of the home itself.

We can, for example, imagine devices that, after completing a phase of monitoring and learning about the lifestyles of the occupants, make decisions based on artificial intelligence techniques capable of regulating the internal temperature to keep it constant within an acceptable range when people are inside the house, and allow it to fall when the house is not inhabited because, for example, people are at work. Software applications can also be implemented to remotely monitor and control heating parameters, or perform actions to change temperature conditions such as turning on radiators or closing shutters. A first problem that one encounters is that of the interoperability of these devices, as they are created and distributed by different manufacturers. For this reason, the need to use levels of abstraction is evident, for example by channelling the data into a cloud environment in order to then analyse it effectively. Smart Homes are dedicated to the consumer audience, peoples that will not use them in a professional or working environment.

If, on the other hand, we want to interface with a professional audience, then in this case we are talking about *Smart Building*[189]. These buildings are for example offices and company buildings that are designed or renovated to use smart devices. These are characterised by a wide variety of sensors, ranging from access control, so that we always know how many people are inside the facility, temperature control and monitoring systems, smoke detection systems to effectively suppress fires, lighting management and monitoring systems or energy management and consumption control systems.

These buildings are in any case created with a purpose to reduce their impact and polluting footprint within the cityscape. There is also a subcategory of these buildings called nZEB (near Zero Energy Building). These are characterised by having very low energy consumption, tending towards zero. In order to realise buildings of this type, it is evident that it is necessary to rely on the use of technology, and in particular, the internet of things takes on a decisive role in making all the systems necessary for monitoring and regulating energy consumption interconnected so as to limit or even avoid any kind of waste.

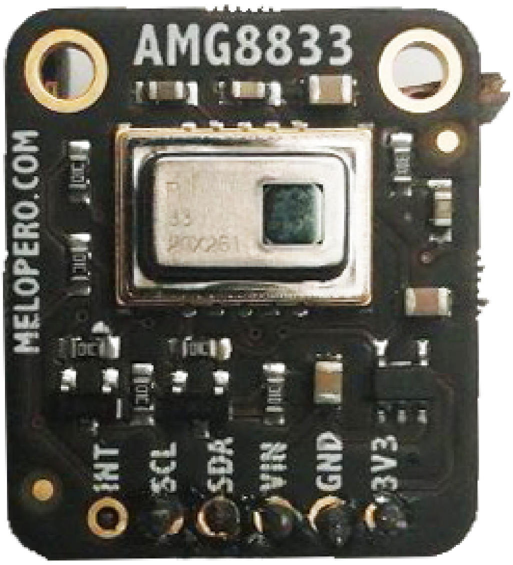


Figure 4.1: Thermal Camera

Smart Health[190], encapsulates all those processes of digitisation of healthcare that, among many other things, include the use of IoT devices. In fact, using these devices capable of, for example, detecting and monitoring certain vital parameters in a fully automatic manner can be a valuable aid to reducing costs, managing bureaucracy, and reducing the time needed to obtain the values required for examinations. The advent of smart Health has then contributed to the development of some of its derivations such as the *Smart Hospital* and the *Smart Patient*. The first category includes all those healthcare facilities that aim to optimise the organisation of information collected from patients, the management of both technical and administrative personnel and a consequent improvement in the efficiency of the service provided to citizens [191]. In the second category are those patients who make use of devices capable of monitoring their health, of sending this information to the doctor or to the competent structure so as to facilitate tele-monitoring, tele-consultation, tele-health cooperation. Devices of this kind today come in various sizes, they can be for example smart watches, smart wrist bands, smart rings, smart scales, and so on [192]. Figure 4.1 shows a thermal camera that use only 4.5 mA and with an operating voltage equals to 3.3V.

Another sector where technological development is particularly important is agriculture. Smart Agriculture aims to improve and increase the productivity of agricultural systems through the use of IoT devices, which can analyse and predict the water resources needed for spraying the land, and can also regulate the amount of water to be supplied daily. The same can be said of fertilisers and manures, which are fully governable through artificial intelligence systems that receive input from these small, interconnected devices [193]. The technology can also be used to improve and guarantee the traceability of products after they have been produced by the companies, small tags can be applied to the bags or packages of products in order to monitor the chain of movement from producer to consumer [194].

4.3 Description of the problem

We have seen how the fields of application of IoT devices are very variegated. Their versatility allows researchers to find applications in so many areas and the number of these devices is expected to grow exponentially in the near future [195]. Using these devices to solve problems, however, triggers new ones. First of all, there is the problem of the use of the *Internet bandwith*. If we imagine that we have millions and millions of devices continuously exchanging data, then we have to consider a considerable increase in costs.

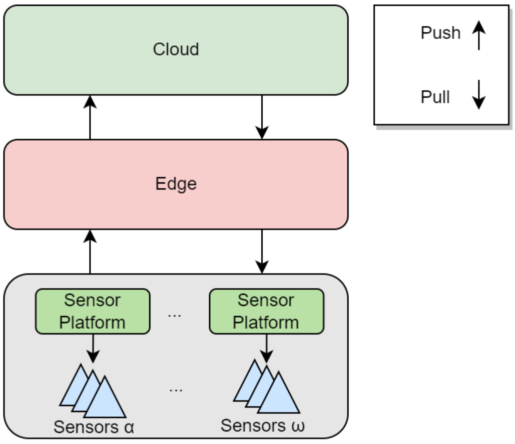


Figure 4.2: Data transfer from the IoT devices to the Cloud

In addition to this, the sent data must be processed, and doing so requires considerable *computing power* on the side of the environments that are located at higher levels of the infrastructure hierarchy, such as the cloud environment. There are many optimisations that can be made when working with IoT devices, for instance, one can efficiently manage data push and pull operations between the various layers of the infrastructure [196]. It must also be remembered that the computational power offered by the cloud has significant costs when used over long periods of time. In recent times, the term *Edge Computing* has started to be used, and represents an intermediate layer between IoT sensors and the cloud. Figure 4.2 shows an example of the path that data take during its transfer from IoT sensors to the cloud.

Optimisations that can be realised concern various aspects of the infrastructure. A first example is the optimisation of the use of the *network bandwidth* required for data transfer, because we only send information when and if it is needed, then we will achieve significant cost savings. Modern cloud infrastructure providers charge high fees in relation to the amount of bandwidth used by users. A second advantage of optimising data transfer concerns the battery consumed by IoT devices. In order to transmit information over the Internet, these devices must connect to the server, authenticate themselves, send the data, and then, depending on the network protocol used, wait or not for confirmation that the server has received the information. This process, leads to considerable battery consumption, because it is important to remember that these devices tend to be powered by small rechargeable batteries. In other cases, however, these devices are connected to the mains, but even in this case, if we imagine that we have dozens and dozens of devices running 24 hours a day, there is still a significant energy saving in optimising the execution of this process [196]. A further aspect to consider is *latency*, which represents the response time of a device. The definition of latency could also be dependent on the use case addressed in fact it could be understood as the time from when the cloud requests a new measurement from a sensor, to when this data is detected, and delivered to its destination. If the goal is to realise an application that performs calculations, measurements, and decisions in real time, then a high latency could be a big problem, which could render new data old and unusable because it arrived with too high a latency time. In general, the optimisation areas that need to be addressed when it comes to large-scale applications using IoT systems are the following:

- Communication bandwidth
- Cloud processing cost
- Energy consumption rate of the IoT device
- Latency

Table 4.1: The rooms and the sensors in the dataset

ROOM \ SENSOR	Brightness	Humidity	History	T	Thermostat T	Virtual outdoor T	Outdoor T	L Thermostat T	R Thermostat T
Bathroom	x	x	x	x	x	x			
Kitchen	x	x	x	x	x	x			
Room1	x	x	x	x	x	x			
Room2	x	x	x	x	x		x		
Room3	x	x	x	x	x	x		x	x
Toilet	x	x	x	x	x	x			

4.4 Reinforcement Learning approach

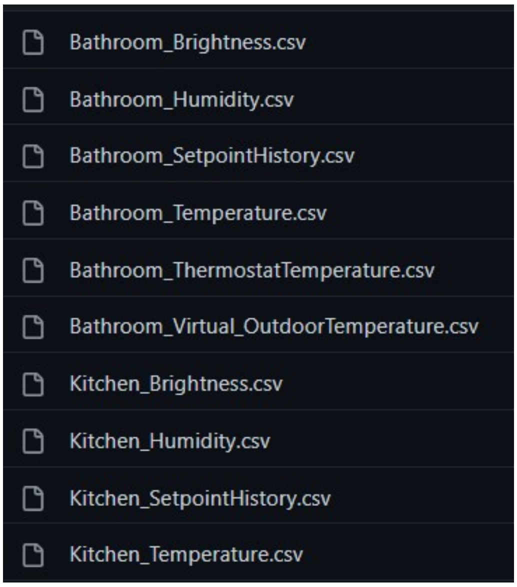
To tackle this problem, we first tried to create a dataset with which to experiment and test. A dataset is essential to properly train an artificial neural network, with which we can then test cases in a real system. To keep the results valid and true, it is also important to use a dataset obtained from real IoT devices and not a synthetic dataset generated e.g. by a random mathematical function. After a careful search within the most popular repositories of scientific datasets, we found a dataset on Kaggle[1] that fulfilled our requirements, namely a high number of examples, a high sampling level, a high number of sensors and a licence to use it. The dataset is called Open Smart Home IoT/IEQ/Energy[2] sensor dataset [197]. It was firstly developed by a group of researchers, who collected information about the parameters of a dwelling. In particular, internal and external temperatures, humidity levels, and the brightness of the various rooms that make up the dwelling, such as the bathroom the kitchen and the various bedrooms, were monitored. In addition to this information, the desired temperature, i.e. the temperature set on the house thermostat in each room, is also reported for each time stamp. The researchers carried out the monitoring for a period from the beginning of March 2017 to June 2017.
The data collected by the researchers are divided within numerous files, an example shown in Figure 4.3 point each file contains the readings that were made by that specific sensor, which characterises the name of the file. An example of the contents of the various files is shown in Figure 4.4. Each file has two columns, where the first column contains time stamps of the measurements taken, while the second column contains the measured values.

The list of the parameters included in the dataset is reported in the Table 4.1, the letter "T" means Temperature, and the letter "L" and "R" means Left and Right, respectively.

The dataset also includes actuation actions named set-points, which are the commands given to the internal thermostats to change and set

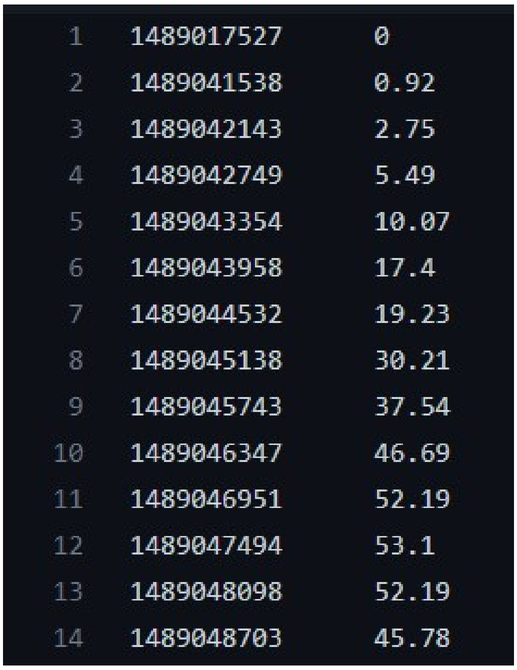
[1] Kaggle

[2] https://www.kaggle.com/datasets/claytonmiller/open-smart-home-iotieqenergy-data



Bathroom_Brightness.csv
Bathroom_Humidity.csv
Bathroom_SetpointHistory.csv
Bathroom_Temperature.csv
Bathroom_ThermostatTemperature.csv
Bathroom_Virtual_OutdoorTemperature.csv
Kitchen_Brightness.csv
Kitchen_Humidity.csv
Kitchen_SetpointHistory.csv
Kitchen_Temperature.csv

Figure 4.3: The repository containing the dataset



```
1489017527    0
1489041538    0.92
1489042143    2.75
1489042749    5.49
1489043354    10.07
1489043958    17.4
1489044532    19.23
1489045138    30.21
1489045743    37.54
1489046347    46.69
1489046951    52.19
1489047494    53.1
1489048098    52.19
1489048703    45.78
```

Figure 4.4: An example of the content of a single file

temperature at specific points in time. The logic behind the actions is unknown but relates to energy savings. Based on the published paper by the authors of the dataset[198], such actions are based on the values of the sensors as well as other external knowledge including for instance presence and absence from the apartment by the resident, which the resident knows and decides. The Open Smart Home IoT/IEQ/Energy dataset is therefore a combined sensor dataset and actuator dataset. Generally speaking, modern artificial neural networks need a data structure with a different form than the one proposed in the original dataset, as the classical input layers of networks are usually structured to receive a sequence of numbers in the form of a vector or tensor. The first phase was therefore to carry out pre-processing work in order to obtain an effective and efficient data structure. This task was carried out in two steps. The first step was to assemble all the data containing the measurements produced by the sensors in a single file. The structure chosen is that of a csv-type file where the first column contains the timestamp and the subsequent columns contain the sensor measurements that were recorded at that particular timestamp. The result of this first processing produced 102,000 rows. Each of these rows consists of 32 columns, representing measurements or data provided by the sensors. A data structure of this kind is very similar, metaphorically speaking, to a photograph of the scenario, it is as if at each timestamp we were able to provide the neural network with a photograph of the house composed of all the measurements taken by the sensors. The next step was to process the null values. If we take a close look at the data provided by the original dataset, we realise that the timestamps are not synchronised between the various sensors, in some cases there are deviations in the order of seconds or even minutes. Notice that in the original dataset, every timestamp is associated with only one sensor change (or one set-point command).

This, which originally appeared to be a defect, can instead be seen as a virtue because if properly exploited, it can very easily lead to a considerable increase in the number of examples within the dataset thanks to the use of the data augmentation technique. This phase was realised with a script in Python that scans all the rows of the dataset and performs a series of operations. The first operation is the pruning of the initial rows. The row index encountered is kept in memory, and it is scrolled starting from the first row until all columns show at least one element, i.e. at least one detection by the sensor to which they belong. In our case, we eliminated the first 1178 rows, so as to eliminate the initial phase in which sensors were added or switched on within the house and thus arrive at the condition in which all sensors are up and running and capable of recording. Subsequently, the null values were filled with a mathematical linear function. Let us assume for example that for column 'n', the last recorded temperature value is 15°C, and that for the next 5 rows there are null values and that on the 6th row there is a value of 18°C. With a simple mathematical calculation, the null values in our example are filled with increments of half a degree centigrade from the first value to the next non-zero value.

Proliferating the sensors and application datasets

To investigate the scalability of our RL technique, we proliferate the original yet compressed sensor dataset into a number of datasets reflecting the number of houses/apartments (nh). We plan to use the Bayesian Proliferation method in [199].

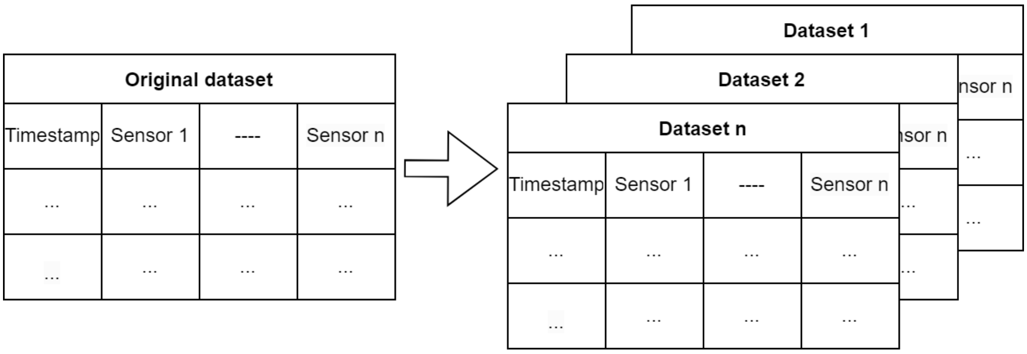


Figure 4.5: Synthesis of the Application Set and Proliferation of the Data and Application Sets

In any proliferated dataset, the set points will remain constant. This is shown in Figure 4.5. The application dataset will be proliferated similarly (based on a parameter nh) also with the actions (set_points) remaining invariant. We should be able to construct our RL algorithm progressively and incrementally after we have the sensor and application datasets available and proliferated. We begin with the most basic reward function design feasible then gradually and carefully add more advanced features. The algorithm should execute at the edge (at every edge) and learn on its own. It should help with application execution by doing the following:

Decision Dimension 1: Deciding whether and where to move or store sensor data in light of new information and circumstances. From the sensor device to the sensor platform cache, from the sensor platform cache to the edge cache, or from the edge cache to the cloud cache (i.e., requesting the sensor device to sample a new value). Determine if a prior caching choice has to be revoked by invalidating the cache. The sensor value location is changed to "the sensor device" by an invalidated cache, suggesting that there is no cache for the sensor data and that it must be created again through sensor sampling.

Decision Dimension 2: Deciding where to cache an application fragment and whether it needs to be cached in light of fresh information and circumstances. Applications live on the cloud at all times. The application's smaller pieces might be cached always on the edge. On the sensor platform, more fragmented pieces of the edge-cached fragments may be cached. Make a decision on pieces of the application cache as well. Within the application hosted in the cloud, an invalidated fragment returns to its original position.

It is also necessary to define some costs that are intrinsic to the problem. For example there is a cost for the communication between the Edge and the Cloud, either direction, we call this parameter C_{EC}. Then we should take into account the cost of the communication from between the Edge and the Sensors, we call this parameter C_{ES}. Moreover there is a cost for the processing operation carried out in the Cloud, P_{C}. Finally there are the costs of the energy consumption of the sensor platform over cached sensor data, E_{S} and the cost of the sensor sampling E_{SS}. Additionally, we must comprehend and design the constraint-based environment in which the RL algorithms (RL agents) will work. Given the limitations of the edge computer, in particular the limited memory and computing capability, this is undoubtedly important. The design of one of the RL agents that partially implements Decision Dimension 2 as stated above is illustrated in the part that follows. An intriguing "complexity" question is how many agents are required. Too many would be complicated in and of themselves, but attempting to have one agent handle everything may also be extremely difficult and error-prone.

Preliminary agent for the Decision Dimension 2

The agent now being described has the specific task of moving fragments of information from the cloud to the edge. This agent specialises in decision to move fragments from the cloud to the edge. The overall result is comparable to, if not the same as, the AFCA-1 algorithm [200]. The preliminary design is depicted in Figure 4.6. The Environment for this agent is described by the remaining memory available and CPU processing at runtime (Edge_Mem_cap and Edge_Proc_cap, respectively), in addition to a status bit (Edge_cap_full) that indicates whether the edge has used up all of its memory, processing, or both. Status 't' embodies knowledge and conditions relevant to this agent, mainly access intensity to parts (fragments) of the applications served by the corresponding edge1. Status 't' embodies conditions and knowledge that are important and relevant to this agent,

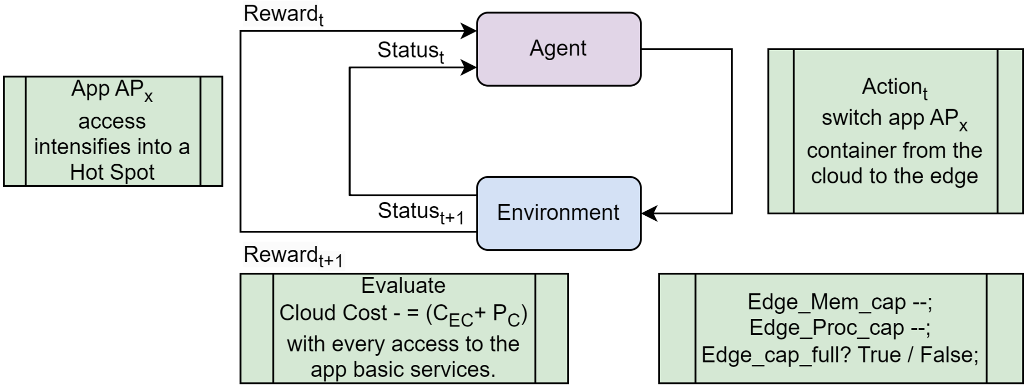

Reward$_t$
Status$_t$
Agent
App AP$_x$ access intensifies into a Hot Spot
Action$_t$ switch app AP$_x$ container from the cloud to the edge
Status$_{t+1}$
Environment
Reward$_{t+1}$
Evaluate Cloud Cost - = (C$_{EC}$+ P$_C$) with every access to the app basic services.
Edge_Mem_cap --;
Edge_Proc_cap --;
Edge_cap_full? True / False;

Figure 4.6: Agent Design for Decision Dimension 2 in terms of Actions, Status and Rewards within a constrained Environment

$$CloudCost = CloudCost - (C_{\text{EC}} + P_{\text{C}}) \tag{4.1}$$

4.1: Evaluated with every access to a basic event

particularly the access intensity to portions (fragments) of the applications served by the corresponding edge.

Instead, it is a long-lived, dynamic function whose initial evaluation starts a monitor that is updated with each access related to this reward function, at least until a certain set of conditions is no longer true. An example of the reward function is:

As long as the application fragment cache is not invalidated, such basic events are incorporated within it.

By simply switching to the container replica of the same virtual event, Action 't' is the process of caching a specific application fragment from the cloud to the edge. The RL algorithm's primary task is to decide whether to take this action or not. It is based on understanding of potential gain or loss as represented by the Reward$_{t+1}$ function.

Figure 2. Agent Design for Decision Dimension 2 in terms of Actions, Status and Rewards within a constrained Environment. What has been described in this section has aspects that still need to be refined and improved, and generally speaking, the work is still in development. For this reason, a collaboration between our research group and the University of Florida is planned in order to further our goals and improve our results.

4.5 Literature Review

We analysed the scientific literature published in recent years that focused on the study of the improvement of techniques for the use of IoT devices with particular emphasis on the topics mentioned in Section 4.3. The search was restricted to IEEE journals and conferences published from 2010 to the present. The results of the research were initially summarised in Table 4.2 and will be described in detail in this Section.

In the table, several columns are shown. The first column identifies the name of the work that was analysed and that is considered to be particularly important for future developments in IoT systems utilising edge and cloud computing. Columns numbering 2 to 5 indicate whether the work examined addressed the specific topic identified by that column. The last column of the table contains references that can be used by the reader to effectively trace the work discussed.

I Optimizing the energy consumption with an appropriate scheduling

The authors of this paper[201] address the problem of optimising energy consumption. Using an algorithm, the IDLE times of devices are analysed. By means of a mathematical model, the time history makes it possible to optimise the scheduling of operations while reducing the overall energy consumption of the system. In fact, it is assumed that the tasks to be performed by the sensors in a network are repetitive and generally always of the same nature, e.g. a temperature sensor will measure the values in a room on a regular basis. For each task completed by a device, the authors store the time taken to complete the current task the time taken by the previous task. An objective function is then used, which among other parameters contains the time taken to complete the task, the energy consumption incurred by the device and the number of connected devices within the network. As a result of the optimisations carried out, a considerable increase in energy efficiency and a reduction in power consumption of approximately 84.37% is declared. The value refers to the use case they tested and their specific sensor configuration.

II Data compression to improve energy efficiency

The authors of this paper[202] address the problem of energy consumption by analysing two aspects in particular, which are the consumption of the radio component used to transfer information to the cloud and then the

Table 4.2: Summary of the article analyzed

Title of the study	Energy power	Cloud cost	Bandwidth	Latency	Reference
Study on Energy Consumption Optimization Scheduling for Internet of Things	✓	✗	✗	✗	Ref.[201] Par. I
Energy Efficient Data Compression in Cloud Based IoT	✓	✗	✓	✓	Ref.[202] Par. II
Energy Optimization for Green Communication in IoT Using Harris Hawks Optimization	✓	✗	✗	✓	Ref.[203] Par. III
CE-IoT: Cost-Effective Cloud-Edge Resource Provisioning for Heterogeneous IoT Applications	✗	✗	✓	✓	Ref.[204] Par. IV
Hierarchical Fog-Cloud Computing for IoT Systems: A Computation Offloading Game	✗	✓	✗	✓	Ref.[205] Par. V
Cost-Driven Off-Loading for DNN-Based Applications Over Cloud, Edge, and End Devices	✗	✓	✗	✗	Ref.[206] Par. VI
Twenty-One Key Factors to Choose an IoT Platform: Theoretical Framework and Its Applications	✓	✓	✓	✓	Ref.[207] Par. VII
Energy-Efficient Resource Allocation Strategy in Massive IoT for Industrial 6G Applications	✗	✗	✓	✓	Ref.[208] Par. VIII
6G-Enabled IoT Home Environment Control Using Fuzzy Rules	✗	✗	✓	✗	Ref.[209] Par. IX
Delay-Aware and Energy-Efficient Computation Offloading in Mobile-Edge Computing Using Deep Reinforcement Learning	✓	✓	✗	✗	Ref.[210] Par. X

consumption of the main electronic circuit of IoT devices. The mathematical model used to improve the current situation is mixed-integer linear programming (MILP). MILP is a mathematical programming system that is used to find optimal solutions from the objective function provided as input. In order to improve the efficiency of the radio component, they have worked on both path optimisation, i.e. on reducing the number of hops that have to be dealt with in the transfer of an information content from source to destination, in other words they have tried to reduce the distance that the information has to travel.

In addition, the selection of the transmission channel is also important; if information is sent using a channel that overlaps with that used by other devices, then energy consumption may suffer, and ideally the freest channel should be selected. With regard to optimisation techniques that affect the energy consumption of the main circuit, first of all data compression technologies are used that take into account the battery charge level and the computational power of the chip, trying to perform the compression steps on the devices that have the best energy efficiency ratio.

III Select a master node to aggregate the data to be sent

The authors of this paper [203] propose a new methodology for sending information about a sensor network. During the life cycle of an iteration between the cloud and an IoT sensor, the phases have different power consumptions. The phase that is most energy-consuming is the transfer of data from the sensors to the cloud. This step requires initial negotiation, possible authentication, sending data, and if there is a lot of data, also maintaining an active connection for a few seconds. The authors have developed an algorithm to select the best cluster head (CH) within the network. This node will have the task of receiving data from all the sensors in the cluster, and once it has received it, it will aggregate it into a packet that will then be sent to the cloud. The idea behind the work is that sending aggregated data, in other words compacting the information content of the measurements made by the network, within a single message will reduce overall consumption. The selected node will be the one with the best energy efficiency ratio, the lowest computational load and also the one with the best latency. A further advantage of the proposed solution is the overall reduction of the system's temperature. The lower computational load borne by the various sensors, thanks to the use of a central node that aggregates the data and sends it on behalf of the network, allows for easier cooling of the network's sensors. It is also important to note that if sensors are operated for a longer period of time at lower temperatures, their life expectancy can be prolonged and the possibility of failure delayed [211].

IV A framework for the resource provisioning

The authors of this paper [204] focus on the study of IoT sensors interconnected via 5G networks. In particular, their case studies focus on sensors for self-driving cars or, for example, in smart agriculture. These types of applications use the concept called mobile-edge computing (MEC). A framework for cloud edge architectures is proposed that is based on the cost-efficiency of devices with the aim of reducing the expense to be incurred when performing long-term operations. The authors also want to keep the latency of the system low, which is also possible through the use of a buffer memory that has the task of maintaining data for a small period of time.

V A Computation Offloading Game

The authors of this paper [205] study the best techniques for resource allocation in a fog cloud paradigm. The problem was modelled using a mathematical algorithm that simulates competition between IoT devices for resources. In their experimentation they used several cloud servers configured as virtual machines with equal computational power, and several fog servers each with limited computational power, much less than that deliverable by the cloud servers. They then modelled the competition between the devices in the form of a game, where each device tries to achieve the best quality of experience (QoE) characterised by low latency and faster resource utilisation. Among their main results, they demonstrated that using fog environments as middleware between the cloud and IoT devices significantly reduces latency time. In addition, their proposed algorithm improves the efficiency of computational computation by approximately 20%.

VI Distribute computational task between Edge and Cloud

The authors of this paper [206] study some innovative new possibilities to reduce cloud costs. It is well known that companies that make virtual machines available in the cloud have policies that provide not only for costs related to server rental, but also for costs related to the use of the machine's computational capacity. Other companies instead use a credit system, where users have a certain amount of credits to spend on CPU utilisation, which are then recharged at times when there is low machine utilisation. The researchers, in particular, addressed the problem of neural networks, which are characterised by a large number of layers and a high demand for computing power to be followed. Let us assume that these networks are to be used by devices within an MEC architecture. Their idea is to exploit the computational capacity of Edge nodes to execute the calculations of certain layers of the neural networks and thus balance the load in the execution of a single network between different architectures. In their structure, the small layers are carried out on Edge servers while the larger layers are carried out on cloud servers. Of course, running a neural network on different servers increases latency, but their purpose is only to reduce the required computational power and consequently CPU costs.

VII How to choose an IoT Platform

The authors of this article [207] carried out a very particular analysis, different from the one presented in the Paragraphs VI. A total of 21 questions are posed, which attempt to describe the key characteristics of cloud platforms that provide virtualised environments within which to develop software applications that utilise the potential of IoT devices. The vendors that are analysed are IBM Watson, AWS, Google Cloud, Oracle IoT and Azure. The result of the analysis is a ranking of the various cloud providers that can help researchers in choosing the environment to be used for future applications, trying to select the most advantageous one based on the desired criteria, which may be the costs of the cloud, the latency of applications the energy consumed, the security required by the environment, the possibility or not of using a hybrid cloud, the possibility of managing bandwidth and so on. To determine the 21 questions needed to rank the cloud providers, the authors scoured the web by analysing over 200 articles, which thanks to filtering steps were reduced to 46. The end result was that Amazon AWS was identified as the best provider, followed by Microsoft Azure, Google cloud IBM Watson and finally Oracle IoT.

VIII 6G for the industries

The authors of this article [208], after analysing the situation of current networks and connection technologies, they look to the future, and focus on the connection technologies that will be used from 2028/2030, such as 6G. In fact, it will be crucial to make proper use of this new connection technology, which in a landscape where IoT devices will be widely deployed, will make it possible to provide the necessary bandwidth for the large amount of data that will be produced. The authors propose a model for the allocation of resources in an environment where IoT devices are massively deployed, with a particular focus on the industrial sector. The multitude of sensors is then subdivided into subsets, or clusters in order to achieve a more intelligent allocation of resources and less computational complexity in managing them.

IX 6G for smart home

The authors of this article [209] carry out an analysis that in some respects is similar to that carried out by the authors of the article described in the previous paragraph. Again, 6G is the main subject of the research. However, instead of focusing on the industrial sphere, they focused on the domestic sphere. Their idea of a smart home involves various sensors that are able to monitor, for example, external environmental conditions and activate mechanisms to protect the home in the event that there are adverse weather events and no tenant is present. The monitoring of a tenant's presence is done by sensing the user's smartphone. The home's water system is also monitored, and sensors can be used to reduce or eliminate dangerous situations that could occur if a pipe breaks. The use case they presented was realised through the use of WiFi wireless technology. The authors describe that the use of 6G can provide significant improvements, such as the possibility of deploying cameras and sensors in places not covered by the home router signal is able to transfer data with a high bandwidth to the main servers. The use of 6G could in fact ensure that high-definition cameras can also be deployed outside the home.

X Optimising task distribution through Reinforcement Learning

The authors of this article [210] deal with how to distribute tasks and resources between the various elements that make up a network consisting of IoT devices and edge servers. To achieve this, they developed an approach based on machine learning, specifically Deep Reinforcement Learning (DRL). The goal of the neural network is to complete as many tasks as possible before the deadline of each task, and at the same time minimise the energy consumed by the system. The choice of DRL was due to the fact that it is not necessary to provide the neural network with input examples with associated labels, as this type of network is able to learn by interacting directly with the environment. The main problem they encountered was the exponential complexity of trying to find the best combination to allocate tasks in a MEC system. The solution was to use a series of approximations that can also be used in the real world. Finally, to validate their thesis and results, a number of simulations are presented to demonstrate the effectiveness of the proposed method.

4.6 Conclusion and future work

In this chapter, a research and study path is presented that addresses the issues of the Internet of Things (IoT). These devices will become increasingly common and widespread, and their numbers are expected to grow exponentially, creating new technological needs in users and new challenges for researchers and scientists. Among the many complications that energy-efficient devices present today are the size of the batteries and the capacity they can hold, and the need to update and study new network communication technologies that will look towards an interconnected, free and easily accessible future. During his PhD, the candidate tackled this problem first hand by developing a dataset from which to model the smart homes of the near future and which can be, by means of data augmentation techniques, expanded in a coherent and consistent manner to represent a synthetic Smart City, in other words a city that collects and delivers information from IoT devices.

The modelling of this dataset took into account an extensive research and review of the scientific literature. The most influential articles from the last five years published in journals and conferences dealing specifically with IoT were searched. From these articles, it could be seen that the main problems faced by researchers are four, energy consumption, the cost of modern cloud platforms, the data bandwidth available for sending and receiving information, and last but not least, latency in measurements, i.e. the time that elapses between the reading of a data item being sent and its reception by the recipient. These four problems proved very difficult to tackle all together. None of the papers encountered were able to solve all four at the same time, but it is clear that for a large-scale deployment of these devices it will be necessary to address them clearly and definitively.

The researchers of the 10 papers that were selected probably took their cue from Alan Turing's theories and broke down this major problem into sub-problems. It is very interesting to note that many researchers are trying to figure out how to optimise the distribution of calculations between nodes, a problem that in some ways appears similar to that which had to be faced decades ago in the realisation of a Linux process scheduler capable of distributing tasks among various processors. Compared to then, the problem faced today is much more complicated because there are different latencies between the various sensors and servers in the Edge. Furthermore, even the sensors themselves can have heterogeneous characteristics and it is not even guaranteed that the servers all have the same computational capacity and storage memory. For these reasons, the approaches that seem to be preferable are those based on artificial neural networks with reinforcement learning. This technique does not require the network to be given examples with labels, but allows the system to learn simply by interacting with the MEC (Fog, Edge, Cloud Computing) environment.

Considerable research and studies are carried out with a view to reducing the costs that Cloud providers require to deliver services. Generally, these costs are calculated in a different manner depending on the Cloud provider, but there tends to be a basic cost necessary for the rental of servers, to which must be added a cost directly proportional to the percentage of use of the processor of the rented machine. Let's imagine for a moment that we are setting up a network with hundreds of IoT sensors inside it that can monitor and detect parameters in an entire district of a city. Processing this quantity of data requires an enormous amount of computing power and consequently increases costs.

There are various strategies that have been identified for reducing the expense of supporting this type of infrastructure. First of all, it is necessary to take advantage of the servers that are located in the Edge, which have sufficient computing power to perform many tasks, but it is still important to remember that they are generally less powerful than servers that we can rent in the Cloud. A second trick is to distribute the calculations between these two layers. The closer the calculations are performed to the user, the lower the latency we will measure between reading the data and getting the response from the servers. We have seen that some researchers have also been able to distribute the calculations necessary for the functioning of artificial neural networks in such a way that some layers of the network are executed in Edge computing and other (larger) layers on the Cloud. This solution in particular appears to be extremely interesting, as it drastically reduces costs; of course, one must keep in mind that a large bandwidth is required and there may be an increase in latency. Latency and bandwidth are in fact two other extremely relevant topics that need to be addressed in order to use IoT devices profitably. The scientific community seems to agree that as far as devices that are located outside the home are concerned, it is imperative to have the most efficient connectivity, i.e. one that can transfer information quickly and without consuming too much battery power. Today, the prevailing technology is 5G LTE, which is now widely deployed in many countries around the world and opens the door to the use of a large number of interconnected devices. From the years 2028/2030 onwards, it is expected that there will be a switch to 6G technology capable of guaranteeing the proper functioning of millions and millions of devices with higher bandwidth, for example, able to operate street cameras with an HD video stream without incurring increased latency. With regard to future work the candidate believes it would be appropriate to continue this path by expanding the dataset created and keeping in mind the information learned from this work of study and review of scientific literature so as to contribute as much as possible to the development and improvement of new technologies and techniques that can be applied in modern society for the improvement of wellbeing, which we remember is one of the main aspects of life. Smart cities should not only be understood as intelligent cities capable of monitoring and detecting parameters with the mere aim of increasing productivity, but should instead represent environments capable of enacting positive influence and improving people's quality of life, environments where neural networks are used to solve problems and satisfy citizens' needs, thus contributing to the improvement of society.

Conclusions

In this thesis, we have presented several themes and approaches aimed at solving modern and complex problems. In the course of the first chapter, we were able to see how machine learning is extremely useful today for solving classification problems, for recognising objects or patterns within images, or even for supporting algorithms to be used for the identification of amino acid chains. To achieve these results, the various frameworks that are now available to the scientific community, such as Keras, have been analysed and their qualities exploited and identified. Numerous artificial neural networks consisting of tens of millions of neurons were modified from general-purpose networks or created from scratch and then trained to solve the problems faced by the research team. The mastery of these technologies and the possibility of running them on affordable hardware is invaluable.

We then saw how virtual reality, augmented reality, and in recent times the technologies behind virtual worlds, such as meta-verses, are becoming more and more concrete and capable of providing realistic results for users. In particular, the studies conducted with local high schools have enabled us to develop software that can support and, in some cases, improve the quality of teaching, especially for the study of scientific subjects where the representation of functions or mathematical objects is complicated to achieve on a classic chalkboard. Instead, showing animations of trigonometric functions, three-dimensional solids, or animations of physical phenomena, which, thanks to augmented reality, are projected above a sheet of paper placed on the desk with the sole help of the student's smartphone, aroused amazement and enthusiasm on the part of the students and great joy on the part of the entire research group to which the candidate belongs, the other members being Professor Osvaldo Gervasi and Dr Marco Simonetti, with which the applications were developed. These technologies have also been applied in the world of telerehabilitation with the development of applications for patients with neurological diseases who need exercises that can be programmed by the doctor to re-train those areas of the brain that have lost the ability to coordinate certain aspects of motor functions.

Moreover, the response obtained with the development of the Android application for patients suffering from Visual Snow, a condition that causes the person to see an image comparable to that obtained when the television is not correctly tuned, with the presence of a classic and characteristic snow effect, was also extraordinary. After the publication of an initial article on this subject, and the publication of a free Android application that can be freely downloaded by users, we were, in fact, contacted by the Visual Snow Initiative, an association whose aim is to help people suffering from this pathology, and with which a fruitful research collaboration has been initiated that will continue in the coming months,

All of the technologies and projects presented and described have made constant and effective use of cloud computing, which has been a milestone in the journey over the past three years, and which has enabled content to be disseminated and enjoyed simply and effectively by users using simple web browsers. Working in tandem on these technologies, which at first sight appear heterogeneous, has instead made it possible to demonstrate that they are perfectly amalgamable and interoperable and that one cannot do without the other. Combining the potential of the Cloud with the potential of virtual reality has, in fact, made it possible to simplify the workflow, for instance, by exploiting WebGL to create three-dimensional graphics applications, which are then made available to users via web servers managed and created by the candidate. Accessible to users regardless of the operating system used because they can be natively executed by smartphones, tablets or computers equipped with an Internet connection and an up-to-date browser.

The candidate also worked to support teaching at the University of Perugia, for which he acted as Technical Manager of the LibreEOL web platform. This platform was used during the pandemic period by COVID-19 for conducting written examinations remotely and is now used for conducting in-person examinations. The platform was originally the subject of the candidate's three-year thesis, and over the years, he has continued to improve and support it as the sole developer. Since its adoption by the University of Perugia, 109,377 examinations have been conducted, a number that shows just how much study and work has gone into ensuring the effectiveness and efficiency of the Javascript and PHP code created and the precise programming of the hardware infrastructure that runs the system from behind the scenes.

Thanks to further collaboration between our research group and the University of Florida, it was possible to work closely with Professor Sumi Helal. The main topic that has been addressed is that of Internet of Things devices in an attempt to solve some of the problems encountered when using devices of this type, such as the low battery capacity and the cost generated by the use of Cloud services, which to date is directly proportional to the number of devices used. The research path that has been started has yet to be completely finished and will be continued in the months and years to come. For the time being, a thorough search has been conducted of the modern scientific literature of the last few years that addresses these problems, and a summary has been drawn up in such a way as to summarise the most effective solutions that researchers have managed to discover. In addition to this, the work that has been undertaken is aimed at the realisation by means of machine learning techniques of a system capable of reducing the costs that users face when using the Cloud. In order to simulate the aspects and characteristics necessary to realise this path, we started with the analysis of a dataset created by third-party researchers, which we cleaned up and remodelled, and which we will expand in the future through proliferation techniques. The aim of expanding the dataset is to be able to simulate not just one household but a multitude of households using IoT devices, This large number of examples will therefore be crucial for the training of neural networks whose objective function will be to reduce system costs while maintaining high efficiency.

Bibliography

[1] Damiano Perri, Paolo Sylos Labini, Osvaldo Gervasi, Sergio Tasso, and Flavio Vella. "Towards a Learning-Based Performance Modeling for Accelerating Deep Neural Networks". In: *Computational Science and Its Applications - ICCSA 2019 - 19th International Conference, Saint Petersburg, Russia, July 1-4, 2019, Proceedings, Part I*. Vol. 11619. Lecture Notes in Computer Science. Springer, 2019, pp. 665–676. DOI: 10.1007/978-3-030-24289-3_49. URL: https://doi.org/10.1007/978-3-030-24289-3%5C_49.

[2] Paolo Sylos Labini, Marco Cianfriglia, Damiano Perri, Osvaldo Gervasi, Grigori Fursin, Anton Lokhmotov, Cedric Nugteren, Bruno Carpentieri, Fabiana Zollo, and Flavio Vella. "On the Anatomy of Predictive Models for Accelerating GPU Convolution Kernels and Beyond". In: *ACM Trans. Archit. Code Optim.* 18.1 (Jan. 2021). ISSN: 1544-3566. DOI: 10.1145/3434402. URL: https://doi.org/10.1145/3434402.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Curran Associates, Inc., 2012, pp. 1097–1105. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[4] Yulin Zhao, Donghui Wang, and Leiou Wang. "Convolution Accelerator Designs Using Fast Algorithms". In: *Algorithms* 12.5 (2019), p. 112.

[5] Aravind Vasudevan, Andrew Anderson, and David Gregg. "Parallel multi channel convolution using general matrix multiplication". In: *2017 IEEE 28th International Conference on Application-specific Systems, Architectures and Processors (ASAP)*. IEEE. 2017, pp. 19–24.

[6] Andrew Lavin and Scott Gray. "Fast algorithms for convolutional neural networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. June 2016, pp. 4013–4021.

[7] Victor Podlozhnyuk. "FFT-based 2D convolution". In: *NVIDIA white paper* 32 (2007).

[8] Partha Maji, Andrew Mundy, Ganesh Dasika, Jesse Beu, Matthew Mattina, and Robert Mullins. "Efficient Winograd or Cook-Toom

Convolution Kernel Implementation on Widely Used Mobile CPUs". In: *arXiv preprint arXiv:1903.01521* (2019).

[9] Sanket Tavarageri, Alexander Heinecke, Sasikanth Avancha, Gagandeep Goyal, Ramakrishna Upadrasta, and Bharat Kaul. "PolyDL: Polyhedral Optimizations for Creation of High Performance DL primitives". In: *arXiv preprint arXiv:2006.02230* (2020).

[10] Catherine Wong, Neil Houlsby, Yifeng Lu, and Andrea Gesmundo. "Transfer Learning with Neural AutoML". In: *Proceedings of the 32nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., 2018, pp. 8366–8375.

[11] Kazushige Goto and Robert A. van de Geijn. "Anatomy of High-Performance Matrix Multiplication". In: *ACM Trans. Math. Softw.* 34.3 (May 2008). ISSN: 0098-3500. DOI: 10.1145/1356052.1356053. URL: https://doi.org/10.1145/1356052.1356053.

[12] R. Clint Whaley, Antoine Petitet, and Jack J. Dongarra. "Automated Empirical Optimization of Software and the ATLAS Project". In: *PARALLEL COMPUTING* 27 (2000), p. 2001.

[13] Nvidia. "Cublas library". In: *NVIDIA Corporation, Santa Clara, California* 15.27 (2008), p. 31.

[14] Intel. *Intel Math Kernel Library. Reference Manual*. Santa Clara, USA. ISBN 630813-054US. 2018. URL: https://software.intel.com/sites/default/files/managed/83/0a/mkl-2018-developer-reference-c_0.pdf.

[15] Sharan Narang. *DeepBench*. last access 20 October 2017. URL: https://github.com/baidu-research/DeepBench.

[16] Damiano Perri, Marco Simonetti, Andrea Lombardi, Noelia Faginas-Lago, and Osvaldo Gervasi. "A New Method for Binary Classification of Proteins with Machine Learning". In: *Computational Science and Its Applications – ICCSA 2021*. Ed. by Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Chiara Garau, Ivan Blečić, David Taniar, Bernady O. Apduhan, Ana Maria A. C. Rocha, Eufemia Tarantino, and Carmelo Maria Torre. Cham: Springer International Publishing, 2021, pp. 388–397. ISBN: 978-3-030-87016-4.

[17] Damiano Perri, Marco Simonetti, Andrea Lombardi, Noelia Faginas-Lago, and Osvaldo Gervasi. "Binary Classification of Proteins by a Machine Learning Approach". In: *Computational Science and Its Applications – ICCSA 2020*. Ed. by Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Chiara Garau, Ivan Blečić, David Taniar, Bernady O. Apduhan, Ana Maria A. C. Rocha, Eufemia Tarantino, Carmelo Maria Torre, and Yeliz Karaca. Springer. Cham: Springer International Publishing, 2020, pp. 549–558. ISBN: 978-3-030-58820-5.

[18] Priscilla Benedetti, Damiano Perri, Marco Simonetti, Osvaldo Gervasi, Gianluca Reali, and Mauro Femminella. "Skin Cancer Classification Using Inception Network and Transfer Learning". In: *Computational*

Science and Its Applications – ICCSA 2020. Ed. by Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Chiara Garau, Ivan Blečić, David Taniar, Bernady O. Apduhan, Ana Maria A.C. Rocha, Eufemia Tarantino, Carmelo Maria Torre, and Yeliz Karaca. Cham: Springer International Publishing, 2020, pp. 536–545. ISBN: 978-3-030-58799-4.

[19] Valentina Franzoni, Giulio Biondi, Damiano Perri, and Osvaldo Gervasi. "Enhancing Mouth-Based Emotion Recognition Using Transfer Learning". In: *Sensors* 20.18 (2020). ISSN: 1424-8220. DOI: 10.3390/s20185222. URL: https://www.mdpi.com/1424-8220/20/18/5222.

[20] Giulio Biondi, Valentina Franzoni, Osvaldo Gervasi, and Damiano Perri. "An Approach for Improving Automatic Mouth Emotion Recognition". In: *Computational Science and Its Applications – ICCSA 2019*. Ed. by Sanjay Misra, Osvaldo Gervasi, Beniamino Murgante, Elena Stankova, Vladimir Korkhov, Carmelo Torre, Ana Maria A.C. Rocha, David Taniar, Bernady O. Apduhan, and Eufemia Tarantino. Vol. 11619. LNCS. Cham: Springer International Publishing, 2019, pp. 649–664. ISBN: 978-3-030-24289-3.

[21] Matteo Riganelli, Valentina Franzoni, Osvaldo Gervasi, and Sergio Tasso. *EmEx, a Tool for Automated Emotive Face Recognition Using Convolutional Neural Networks*. Vol. 10406. LNCS. Cham: Springer International Publishing, 2017, pp. 692–704. ISBN: 978-3-319-62398-6.

[22] Osvaldo Gervasi, Valentina Franzoni, Matteo Riganelli, and Sergio Tasso. "Automating facial emotion recognition". In: *Web Intelligence* (2019). ISSN: 24056464. DOI: 10.3233/WEB-190397.

[23] Damiano Perri, Marco Simonetti, and Osvaldo Gervasi. "Synthetic Data Generation to Speed-Up the Object Recognition Pipeline". In: *Electronics* 11.1 (2022). ISSN: 2079-9292. DOI: 10.3390/electronics11010002. URL: https://www.mdpi.com/2079-9292/11/1/2.

[24] David A van Dyk and Xiao-Li Meng. "The Art of Data Augmentation". In: *Journal of Computational and Graphical Statistics* 10.1 (2001), pp. 1–50. DOI: 10.1198/10618600152418584. eprint: https://doi.org/10.1198/10618600152418584. URL: https://doi.org/10.1198/10618600152418584.

[25] John E Stone, David Gohara, and Guochun Shi. "OpenCL: A parallel programming standard for heterogeneous computing systems". In: *Computing in science & engineering* 12.3 (2010), pp. 66–73.

[26] Sandra Wienke, Paul Springer, Christian Terboven, and Dieter an Mey. "OpenACC—first experiences with real-world applications". In: *European Conference on Parallel Processing*. Springer. 2012, pp. 859–870.

[27] Cedric Nugteren. "CLBlast: A Tuned OpenCL BLAS Library". In: *CoRR* abs/1705.05249 (2017). URL: http://arxiv.org/abs/1705.05249.

[28] Max Heimel, Michael Saecker, Holger Pirk, Stefan Manegold, and Volker Markl. "Hardware-oblivious parallelism for in-memory column-stores". In: *Proceedings of the VLDB Endowment* 6.9 (2013), pp. 709–720.

[29] Jason Ansel, Shoaib Kamil, Kalyan Veeramachaneni, Jonathan Ragan-Kelley, Jeffrey Bosboom, Una-May O'Reilly, and Saman Amarasinghe. "Opentuner: An extensible framework for program autotuning". In: *Proceedings of the 23rd international conference on Parallel architectures and compilation*. PACT '14. Edmonton, AB, Canada: ACM, 2014, pp. 303–316. ISBN: 978-1-4503-2809-8. DOI: 10.1145/2628071.2628092. URL: http://doi.acm.org/10.1145/2628071.2628092.

[30] Ari Rasch and Sergei Gorlatch. "ATF: A generic directive-based auto-tuning framework". In: *Concurrency and Computation: Practice and Experience* 31.5 (2019), e4423.

[31] Johannes Sailer, Christian Frey, and Christian Kühnert. "GPU GEMM-Kernel Autotuning for scalable machine learners". In: *Machine Learning for Cyber Physical Systems*. Springer, 2019, pp. 66–76.

[32] Nicolas Vasilache, Oleksandr Zinenko, Theodoros Theodoridis, Priya Goyal, Zachary DeVito, William S Moses, Sven Verdoolaege, Andrew Adams, and Albert Cohen. "Tensor comprehensions: Framework-agnostic high-performance machine learning abstractions". In: *arXiv preprint arXiv:1802.04730* (2018).

[33] Philippe Tillet and David Cox. "Input-aware Auto-tuning of Compute-bound HPC Kernels". In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. SC '17. Denver, Colorado: ACM, 2017, 43:1–43:12. ISBN: 978-1-4503-5114-0. DOI: 10.1145/3126908.3126939. URL: http://doi.acm.org/10.1145/3126908.3126939.

[34] Simon Garcia De Gonzalo, Sitao Huang, Juan Gómez-Luna, Simon Hammond, Onur Mutlu, and Wen-mei Hwu. "Automatic generation of warp-level primitives and atomic instructions for fast and portable parallel reduction on GPUs". In: *Proceedings of the 2019 IEEE/ACM International Symposium on Code Generation and Optimization*. IEEE Press. 2019, pp. 73–84.

[35] Somashekaracharya G Bhaskaracharya, Julien Demouth, and Vinod Grover. "Automatic Kernel Generation for Volta Tensor Cores". In: *arXiv preprint arXiv:2006.12645* (2020).

[36] Bryan Singer and Manuela Veloso. "Learning to Predict Performance from Formula Modeling and Training Data". In: *In Proceedings of the Seventeenth International Conference on Machine Learning*. Morgan, 2000, pp. 887–894.

[37] Amir H. Ashouri, William Killian, John Cavazos, Gianluca Palermo, and Cristina Silvano. "A Survey on Compiler Autotuning Using Machine Learning". In: *ACM Comput. Surv.* 51.5 (Sept. 2018). ISSN:

0360-0300. DOI: 10.1145/3197978. URL: https://doi.org/10.1145/3197978.

[38] Thomas L. Falch and Anne C. Elster. "Machine learning-based autotuning for enhanced performance portability of OpenCL applications". In: *Concurrency and Computation: Practice and Experience* 29.8 (2017). e4029 cpe.4029, e4029–n/a. ISSN: 1532-0634. DOI: 10.1002/cpe.4029. URL: http://dx.doi.org/10.1002/cpe.4029.

[39] Frédéric de Mesmay, Yevgen Voronenko, and Markus Püschel. "Offline Library Adaptation Using Automatically Generated Heuristics". In: *IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. 2010, pp. 1–10.

[40] Tianqi Chen, Thierry Moreau, Ziheng Jiang, Lianmin Zheng, Eddie Q. Yan, Haichen Shen, Meghan Cowan, Leyuan Wang, Yuwei Hu, Luis Ceze, Carlos Guestrin, and Arvind Krishnamurthy. "TVM: An Automated End-to-End Optimizing Compiler for Deep Learning". In: *OSDI*. 2018.

[41] Zhen Xie, Guangming Tan, Weifeng Liu, and Ninghui Sun. "IA-SpGEMM: An Input-aware Auto-tuning Framework for Parallel Sparse Matrix-matrix Multiplication". In: *Proceedings of the ACM International Conference on Supercomputing*. ICS '19. Phoenix, Arizona: ACM, 2019, pp. 94–105. ISBN: 978-1-4503-6079-1. DOI: 10.1145/3330345.3330354. URL: http://doi.acm.org/10.1145/3330345.3330354.

[42] Yue Zhao, Jiajia Li, Chunhua Liao, and Xipeng Shen. "Bridging the Gap Between Deep Learning and Sparse Matrix Format Selection". In: *Proceedings of the 23rd ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*. PPoPP '18. Vienna, Austria: ACM, 2018, pp. 94–108. ISBN: 978-1-4503-4982-6. DOI: 10.1145/3178487.3178495. URL: http://doi.acm.org/10.1145/3178487.3178495.

[43] ARM. *A Software Library for Computer Vision and Machine Learning*. https://www.arm.com/why-arm/technologies/compute-library. May 2018.

[44] "Collective Knowledge: Towards R&D sustainability". In.

[45] H. M. Berman, Feng Westbrook, G. Gilliland, T. N. Bhat, H. Weissig, I. N Shindyalov, and P. E. Bourne. "The Protein Data Bank". In: *Nucleic Acids Research* 28 (2000), pp. 235–242. URL: http://www.rcsb.org/.

[46] Sofia Visa, Brian Ramsay, Anca L Ralescu, and Esther Van Der Knaap. "Confusion Matrix-based Feature Selection." In: *MAICS* 710 (2011), pp. 120–127.

[47] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions". In: *Scientific Data* 5.1 (Aug. 2018), p. 180161. ISSN: 2052-4463. DOI: 10.1038/sdata.2018.161. URL: https://doi.org/10.1038/sdata.2018.161.

[48] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.

[49] Ian J. Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. http://www.deeplearningbook.org. Cambridge, MA, USA: MIT Press, 2016.

[50] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*. Feb. 2016. DOI: 10.1609/aaai.v31i1.11231. arXiv: 1602.07261. URL: http://arxiv.org/abs/1602.07261.

[51] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[52] Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nati Srebro, and Benjamin Recht. “The Marginal Value of Adaptive Gradient Methods in Machine Learning”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper/2017/file/81b3833e2504647f9d794f7d7b9bf341-Paper.pdf.

[53] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. “A Semi-automatic Methodology for Facial Landmark Annotation”. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. 2013, pp. 896–903.

[54] V. Kazemi and J. Sullivan. “One millisecond face alignment with an ensemble of regression trees”. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 1867–1874.

[55] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. “300 Faces In-The-Wild Challenge: database and results”. In: *Image and Vision Computing* 47 (2016), pp. 3–18. ISSN: 0262-8856. DOI: https://doi.org/10.1016/j.imavis.2016.01.002. URL: http://www.sciencedirect.com/science/article/pii/S0262885616000147.

[56] François Chollet et al. *Keras*. https://github.com/fchollet/keras. 2015.

[57] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic. “300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge”. In: *2013 IEEE International Conference on Computer Vision Workshops*. 2013, pp. 397–403.

[58] Antreas Antoniou, Amos Storkey, and Harrison Edwards. *Data Augmentation Generative Adversarial Networks*. Nov. 2017. arXiv: 1711.04340. URL: http://arxiv.org/abs/1711.04340.

[59] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Sept. 2014. arXiv: 1409.1556. URL: https://arxiv.org/abs/1409.1556.

[60] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. *Rethinking the Inception Architecture for Computer Vision*. Dec. 2015. arXiv: 1512.00567. URL: http://arxiv.org/abs/1512.00567.

[61] F. Chollet. "Xception: Deep Learning with Depthwise Separable Convolutions". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1800–1807.

[62] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Commun. ACM* 60.6 (May 2017), pp. 84–90. ISSN: 0001-0782. DOI: 10.1145/3065386. URL: https://doi.org/10.1145/3065386.

[63] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255.

[64] Ali Mollahosseini, Behzad Hasani, and Mohammad H. Mahoor. "AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild". In: *IEEE Trans. Affect. Comput.* 10.1 (Jan. 2019), pp. 18–31. ISSN: 1949-3045. DOI: 10.1109/TAFFC.2017.2740923. URL: https://doi.org/10.1109/TAFFC.2017.2740923.

[65] Paul Ekman. "An Argument for Basic Emotions". In: *Cognition and Emotion* (1992). ISSN: 14640600. DOI: 10.1080/02699939208411068.

[66] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". In: *CoRR* abs/1602.07261 (2016). arXiv: 1602.07261. URL: http://arxiv.org/abs/1602.07261.

[67] Lisa Torrey and Jude Shavlik. "Transfer learning". In: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010, pp. 242–264.

[68] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. "Activation functions in neural networks". In: *towards data science* 6.12 (2017), pp. 310–316.

[69] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[70] Osvaldo Gervasi, Damiano Perri, Marco Simonetti, and Sergio Tasso. "Strategies for the Digitalization of Cultural Heritage". In: *Computational Science and Its Applications – ICCSA 2022 Workshops*. Ed. by Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Ana Maria A. C. Rocha, and Chiara Garau. Cham: Springer International Publishing, 2022, pp. 486–502. ISBN: 978-3-031-10592-0.

[71] David Luebke, Martin Reddy, Jonathan D Cohen, Amitabh Varshney, Benjamin Watson, and Robert Huebner. *Level of detail for 3D graphics*. Morgan Kaufmann, 2003.

[72] Tomas Akenine-Moller, Eric Haines, and Naty Hoffman. *Real-time rendering.* AK Peters/crc Press, 2019.

[73] Marco Simonetti, Damiano Perri, Natale Amato, and Osvaldo Gervasi. “Teaching Math with the Help of Virtual Reality”. In: *Computational Science and Its Applications – ICCSA 2020.* Ed. by Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Chiara Garau, Ivan Blečić, David Taniar, Bernady O. Apduhan, Ana Maria A. C. Rocha, Eufemia Tarantino, Carmelo Maria Torre, and Yeliz Karaca. Cham: Springer International Publishing, 2020, pp. 799–809. ISBN: 978-3-030-58820-5.

[74] Klaus Engel, Markus Hadwiger, Joe M Kniss, Aaron E Lefohn, Christof Rezk Salama, and Daniel Weiskopf. “Real-time volume graphics”. In: *ACM Siggraph 2004 Course Notes.* 2004, 29–es.

[75] Richard Szeliski. *Computer vision: algorithms and applications.* Springer Science & Business Media, 2010.

[76] Philip Schneider and David H Eberly. *Geometric tools for computer graphics.* Elsevier, 2002.

[77] Mirza Beig, Bill Kapralos, Karen Collins, and Pejman Mirza-Babaei. “G-SpAR: GPU-based voxel graph pathfinding for spatial audio rendering in games and VR”. In: *2019 IEEE Conference on Games (CoG).* IEEE. 2019, pp. 1–8.

[78] Won-Jong Lee, Seok Joong Hwang, Youngsam Shin, Jeong-Joon Yoo, and Soojung Ryu. “Fast stereoscopic rendering on mobile ray tracing GPU for virtual reality applications”. In: *2017 IEEE International Conference on Consumer Electronics (ICCE).* IEEE. 2017, pp. 355–357.

[79] Chenhao Xie, Fu Xin, Mingsong Chen, and Shuaiwen Leon Song. “OO-VR: NUMA friendly object-oriented VR rendering framework for future NUMA-based multi-GPU systems”. In: *2019 ACM/IEEE 46th Annual International Symposium on Computer Architecture (ISCA).* IEEE. 2019, pp. 53–65.

[80] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. “Fastnerf: High-fidelity neural rendering at 200fps”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 2021, pp. 14346–14355.

[81] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. “Neural geometric level of detail: Real-time rendering with implicit 3D shapes”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2021, pp. 11358–11367.

[82] Francesco Milano, Antonio Loquercio, Antoni Rosinol, Davide Scaramuzza, and Luca Carlone. “Primal-dual mesh convolutional neural networks”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 952–963.

[83] Fabio Remondino. “Heritage recording and 3D modeling with photogrammetry and 3D scanning”. In: *Remote sensing* 3.6 (2011), pp. 1104–1138.

[84] Naci Yastikli. “Documentation of cultural heritage using digital photogrammetry and laser scanning”. In: *Journal of Cultural heritage* 8.4 (2007), pp. 423–427.

[85] Gabriele Guidi, Michele Russo, Sebastiano Ercoli, Fabio Remondino, Alessandro Rizzi, and Fabio Menna. “A multi-resolution methodology for the 3D modeling of large and complex archeological areas”. In: *International Journal of Architectural Computing* 7.1 (2009), pp. 39–55.

[86] Elmedin Selmanović, Selma Rizvic, Carlo Harvey, Dusanka Boskovic, Vedad Hulusic, Malek Chahin, and Sanda Sljivo. “Improving accessibility to intangible cultural heritage preservation using virtual reality”. In: *Journal on Computing and Cultural Heritage (JOCCH)* 13.2 (2020), pp. 1–19.

[87] Meltem Altinay Ozdemir. “Virtual Reality (VR) and Augmented Reality (AR) technologies for accessibility and marketing in the tourism industry”. In: *ICT Tools and Applications for Accessible Tourism.* IGI Global, 2021, pp. 277–301.

[88] Franca Garzotto, Vito Matarazzo, Nicolò Messina, Mirko Gelsomini, and Carlo Riva. “Improving museum accessibility through storytelling in wearable immersive virtual reality”. In: *2018 3rd Digital Heritage International Congress (DigitalHERITAGE) held jointly with 2018 24th International Conference on Virtual Systems & Multimedia (VSMM 2018).* IEEE. 2018, pp. 1–8.

[89] Leandro Soares Guedes, Luiz André Marques, and Gabriellen Vitório. “Enhancing interaction and accessibility in museums and exhibitions with Augmented Reality and Screen Readers”. In: *International Conference on Computers Helping People with Special Needs.* Springer. 2020, pp. 157–163.

[90] Bernadette Perry. “Gamifying French language learning: A case study examining a quest-based, augmented reality mobile learning-tool”. In: *Procedia-Social and Behavioral Sciences* 174 (2015), pp. 2308–2315.

[91] Carl Laverdière, Jason Corban, Jason Khoury, Susan Mengxiao Ge, Justin Schupbach, Edward J Harvey, Rudy Reindl, and Paul A Martineau. “Augmented reality in orthopaedics: a systematic review and a window on future possibilities”. In: *The bone & joint journal* 101.12 (2019), pp. 1479–1488.

[92] Roque Marin, Pedro J Sanz, and J Salvador Sánchez. “A very high level interface to teleoperate a robot via Web including augmented reality”. In: *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No. 02CH37292).* Vol. 3. IEEE. 2002, pp. 2725–2730.

[93] Prithwijit Das, Meng'ou Zhu, Laura McLaughlin, Zaid Bilgrami, Ruth L Milanaik, et al. "Augmented reality video games: new possibilities and implications for children and adolescents". In: *Multimodal Technologies and Interaction* 1.2 (2017), p. 8.

[94] TH Jung and Dai-In Han. "Augmented Reality (AR) in Urban Heritage Tourism." In: *e-Review of Tourism Research* 5 (2014).

[95] Damiano Perri, Marco Simonetti, and Osvaldo Gervasi. "Deploying Serious Games for Cognitive Rehabilitation". In: *Computers* 11.7 (2022). ISSN: 2073-431X. DOI: 10.3390/computers11070103. URL: https://www.mdpi.com/2073-431X/11/7/103.

[96] Clark C Abt. *Serious games*. University press of America, 1987.

[97] Francesco Bellotti, Bill Kapralos, Kiju Lee, Pablo Moreno-Ger, and Riccardo Berta. "Assessment in and of serious games: an overview". In: *Advances in human-computer interaction* 2013 (2013).

[98] Alessandro De Gloria, Francesco Bellotti, and Riccardo Berta. "Serious Games for education and training". In: *International Journal of Serious Games* 1.1 (2014).

[99] Francesco Bellotti, Michela Ott, Sylvester Arnab, Riccardo Berta, Sara de Freitas, Kristian Kiili, and Alessandro De Gloria. "Designing serious games for education: from pedagogical principles to game mechanisms". In: *Proceedings of the 5th European Conference on Games Based Learning*. University of Athens Greece. 2011, pp. 26–34.

[100] Ute Ritterfeld, Cuihua Shen, Hua Wang, Luciano Nocera, and Wee Ling Wong. "Multimodality and interactivity: Connecting properties of serious games with educational outcomes". In: *Cyberpsychology & Behavior* 12.6 (2009), pp. 691–697.

[101] Michael Meehan, Brent Insko, Mary Whitton, and Frederick P Brooks Jr. "Physiological measures of presence in stressful virtual environments". In: *Acm transactions on graphics (tog)* 21.3 (2002), pp. 645–652.

[102] Hans W Giessen. "Serious games effects: an overview". In: *Procedia-Social and Behavioral Sciences* 174 (2015), pp. 2240–2244.

[103] Yu Zhonggen. "A meta-analysis of use of serious games in education over a decade". In: *International Journal of Computer Games Technology* 2019 (2019).

[104] Craig Savage, Dominic McGrath, Tim McIntyre, Margaret Wegener, and Michael Williamson. "Teaching physics using virtual reality". In: *AIP Conference Proceedings*. Vol. 1263. 1. American Institute of Physics. 2010, pp. 126–129.

[105] Daniel Weiskopf, Marc Borchers, Thomas Ertl, Martin Falk, Oliver Fechtig, Regine Frank, Frank Grave, Andreas King, Ute Kraus, Thomas Muller, et al. "Visualization in the einstein year 2005: a case study on explanatory and illustrative visualization of relativity and astrophysics". In: *VIS 05. IEEE Visualization, 2005*. IEEE. 2005, pp. 583–590.

[106] Damiano Perri, Martina Fortunelli, Marco Simonetti, Riccardo Magni, Jessica Carloni, and Osvaldo Gervasi. “Rapid Prototyping of Virtual Reality Cognitive Exercises in a Tele-Rehabilitation Context”. In: *Electronics* 10.4 (2021). ISSN: 2079-9292. DOI: 10.3390/electronics10040457. URL: https://www.mdpi.com/2079-9292/10/4/457.

[107] Michael McCue, Andrea Fairman, and Michael Pramuka. “Enhancing quality of life through telerehabilitation”. In: *Physical Medicine and Rehabilitation Clinics* 21.1 (2010), pp. 195–205.

[108] Marco Rogante, Mauro Grigioni, Daniele Cordella, and Claudia Giacomozzi. “Ten years of telerehabilitation: A literature overview of technologies and clinical applications”. In: *NeuroRehabilitation* 27.4 (2010), pp. 287–304.

[109] Keith D Cicerone, Yelena Goldin, Keith Ganci, Amy Rosenbaum, Jennifer V Wethe, Donna M Langenbahn, James F Malec, Thomas F Bergquist, Kristine Kingsley, Drew Nagele, et al. “Evidence-based cognitive rehabilitation: systematic review of the literature from 2009 through 2014”. In: *Archives of physical medicine and rehabilitation* 100.8 (2019), pp. 1515–1533.

[110] Maria Grazia Maggio, Giuseppa Maresca, Rosaria De Luca, Maria Chiara Stagnitti, Bruno Porcari, Maria Cristina Ferrera, Franco Galletti, Carmela Casella, Alfredo Manuli, and Rocco Salvatore Calabrò. “The growing use of virtual reality in cognitive rehabilitation: fact, fake or vision? A scoping review”. In: *Journal of the National Medical Association* 111.4 (2019), pp. 457–463.

[111] Michelle H Chen, Nancy D Chiaravalloti, and John DeLuca. “Neurological update: Cognitive rehabilitation in multiple sclerosis”. In: *Journal of neurology* 268.12 (2021), pp. 4908–4914.

[112] Fatemeh Farokhi-Sisakht, Mehdi Farhoudi, Saeed Sadigh-Eteghad, Javad Mahmoudi, and Gisou Mohaddes. “Cognitive rehabilitation improves ischemic stroke-induced cognitive impairment: Role of growth factors”. In: *Journal of Stroke and Cerebrovascular Diseases* 28.10 (2019), p. 104299.

[113] Maria Grazia Maggio, Rosaria De Luca, Francesco Molonia, Bruno Porcari, Massimo Destro, Carmela Casella, Ramona Salvati, Placido Bramanti, and Rocco Salvatore Calabro. “Cognitive rehabilitation in patients with traumatic brain injury: A narrative review on the emerging use of virtual reality”. In: *Journal of Clinical Neuroscience* 61 (2019), pp. 1–4.

[114] Ladislav Batalik, Filip Dosbaba, Martin Hartman, Katerina Batalikova, and Jindrich Spinar. “Benefits and effectiveness of using a wrist heart rate monitor as a telerehabilitation device in cardiac patients: A randomized controlled trial”. In: *Medicine* 99.11 (2020).

[115] Piotr Walter, Bartłomiej Podsiadły, Marcin Zych, Michał Kamiński, Andrzej Skalski, Tomasz Raczyński, Daniel Janczak, and Małgorzata

Jakubowska. “CNT/Graphite/SBS Conductive Fibers for Strain Sensing in Wearable Telerehabilitation Devices”. In: *Sensors* 22.3 (2022), p. 800.

[116] Samantha G Rozevink, Corry K van der Sluis, Ainara Garzo, Thierry Keller, and Juha M Hijmans. “HoMEcare aRm rehabiLItatioN (MERLIN): telerehabilitation using an unactuated device based on serious games improves the upper limb function in chronic stroke”. In: *Journal of NeuroEngineering and Rehabilitation* 18.1 (2021), pp. 1–12.

[117] Damiano Perri, Marco Simonetti, Osvaldo Gervasi, and Natale Amato. “A Mobile App to Help People Affected by Visual Snow”. In: *Computational Science and Its Applications – ICCSA 2022 Workshops*. Ed. by Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Ana Maria A. C. Rocha, and Chiara Garau. Cham: Springer International Publishing, 2022, pp. 473–485. ISBN: 978-3-031-10592-0.

[118] Francesca Puledda, Christoph Schankin, and Peter J. Goadsby. “Visual snow syndrome”. In: *Neurology* 94.6 (2020), e564–e574. ISSN: 0028-3878. DOI: 10.1212/WNL.0000000000008909. eprint: https://n.neurology.org/content/94/6/e564.full.pdf. URL: https://n.neurology.org/content/94/6/e564.

[119] D. Kondziella, M. H. Olsen, and J. P. Dreier. “Prevalence of visual snow syndrome in the UK”. In: *European Journal of Neurology* 27.5 (2020), pp. 764–772. DOI: https://doi.org/10.1111/ene.14150. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/ene.14150. URL: https://onlinelibrary.wiley.com/doi/abs/10.1111/ene.14150.

[120] Isin Unal-Cevik and F Gokcem Yildiz. “Visual snow in migraine with aura: further characterization by brain imaging, electrophysiology, and treatment–case report”. In: *Headache: The Journal of Head and Face Pain* 55.10 (2015), pp. 1436–1441.

[121] Jenny L. Lauschke, Gordon T. Plant, and Clare L. Fraser. “Visual snow: A thalamocortical dysrhythmia of the visual pathway?” In: *Journal of Clinical Neuroscience* 28 (2016), pp. 123–127. ISSN: 0967-5868. DOI: https://doi.org/10.1016/j.jocn.2015.12.001. URL: https://www.sciencedirect.com/science/article/pii/S0967586815006530.

[122] Owen B White, Meaghan Clough, Allison M McKendrick, and Joanne Fielding. “Visual snow: visual misperception”. In: *Journal of Neuro - Ophthalmology* 38.4 (2018), pp. 514–521.

[123] Francesca Puledda, Christoph Schankin, Kathleen Digre, and Peter J Goadsby. “Visual snow syndrome: what we know so far”. In: *Current opinion in neurology* 31.1 (2018), pp. 52–58.

[124] Giacomo Rizzolatti and Laila Craighero. “The mirror-neuron system”. In: *Annu. Rev. Neurosci.* 27 (2004), pp. 169–192.

[125] Rocco Salvatore Calabrò, Antonino Naro, Margherita Russo, Antonino Leo, Rosaria De Luca, Tina Balletta, Antonio Buda, Gianluca La Rosa,

Alessia Bramanti, and Placido Bramanti. "The role of virtual reality in improving motor performance as revealed by EEG: a randomized clinical trial". In: *Journal of neuroengineering and rehabilitation* 14.1 (2017), pp. 1–16.

[126] Ute Ritterfeld, Michael Cody, and Peter Vorderer. *Serious games: Mechanisms and effects*. Routledge, 2009.

[127] Johannes Breuer and Gary Bente. "Why so serious? On the relation of serious games and learning". In: *Journal for Computer Game Culture* 4 (2010), pp. 7–24.

[128] Maria Grazia Maggio, Desirèe Latella, Giuseppa Maresca, Francesca Sciarrone, Alfredo Manuli, Antonino Naro, Rosaria De Luca, and Rocco Salvatore Calabrò. "Virtual reality and cognitive rehabilitation in people with stroke: an overview". In: *Journal of Neuroscience Nursing* 51.2 (2019), pp. 101–105.

[129] Elisa Mantovani, Chiara Zucchella, Sara Bottiroli, Angela Federico, Rosalba Giugno, Giorgio Sandrini, Cristiano Chiamulera, and Stefano Tamburin. "Telemedicine and virtual reality for cognitive rehabilitation: a roadmap for the COVID-19 pandemic". In: *Frontiers in neurology* 11 (2020), p. 926.

[130] Alessandro Peretti, Francesco Amenta, Seyed Khosrow Tayebati, Giulio Nittari, and Syed Sarosh Mahdi. "Telerehabilitation: review of the state-of-the-art and areas of application". In: *JMIR rehabilitation and assistive technologies* 4.2 (2017), e7.

[131] McKay Moore Sohlberg and Catherine A Mateer. *Cognitive rehabilitation: An integrative neuropsychological approach*. Guilford Press, 2001.

[132] Keyan Cao, Yefan Liu, Gongjie Meng, and Qimeng Sun. "An Overview on Edge Computing Research". In: *IEEE Access* 8 (2020), pp. 85714–85728. DOI: 10.1109/ACCESS.2020.2991734.

[133] Mitchel Resnick. "Learn to code, code to learn". In: *EdSurge, May* 54 (2013).

[134] K Muita, Mika Westerlund, and R Rajala. "The evolution of rapid production: How to adopt novel manufacturing technology". In: *IFAC-PapersOnLine* 48.3 (2015), pp. 32–37.

[135] Mauro Zampolini, Elisabetta Todeschini, Montserrat Guitart, Hermie Hermens, Stephan Ilsbroukx, Velio Macellari, Riccardo Magni, Marco Rogante, Sandro Marchese, Miriam Vollenbroek - Hutten, and Claudia Giacomozzi. "Tele-rehabilitation: Present and future". In: *Annali dell'Istituto superiore di sanità* 44 (Jan. 2008), pp. 125–34.

[136] M. Zampolini E. Todeschini R. Magni. "Sistema di tele riabilitazione per la riabilitazione cognitiva". In: *EUR MED PHYS 2009*. Vol. 45. Suppl. 1 to No. 3. Congresso 37 SIMFER. Campobasso, 2009.

[137] Francesca Santucci, Federico Frenguelli, Alessandro De Angelis, Ilaria Cuccaro, Damiano Perri, and Marco Simonetti. "An Immersive Open Source Environment Using Godot". In: *Computational Science and*

Its Applications – ICCSA 2020. Ed. by Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Chiara Garau, Ivan Blečić, David Taniar, Bernady O. Apduhan, Ana Maria A. C. Rocha, Eufemia Tarantino, Carmelo Maria Torre, and Yeliz Karaca. Cham: Springer International Publishing, 2020, pp. 784–798. ISBN: 978-3-030-58820-5.

[138] Jonathan Linowes. *Unity virtual reality projects*. Packt Publishing Ltd, 2015.

[139] Damiano Perri, Marco Simonetti, Sergio Tasso, and Osvaldo Gervasi. "Learning Mathematics in an Immersive Way". In: *Software Usability*. IntechOpen, 2021.

[140] Jason Jerald, Peter Giokaris, Danny Woodall, Arno Hartbolt, Anish Chandak, and Sebastien Kuntz. "Developing virtual reality applications with Unity". In: *2014 IEEE Virtual Reality (VR)*. IEEE. 2014, pp. 1–3.

[141] Damiano Perri, Marco Simonetti, Osvaldo Gervasi, and Sergio Tasso. *Chapter 4 - High-performance computing and computational intelligence applications with a multi-chaos perspective*. Ed. by Yeliz Karaca, Dumitru Baleanu, Yu-Dong Zhang, Osvaldo Gervasi, and Majaz Moonis. Academic Press, 2022, pp. 55–76. ISBN: 978-0-323-90032-4. DOI: https://doi.org/10.1016/B978-0-323-90032-4.00010-9. URL: https://www.sciencedirect.com/science/article/pii/B9780323900324000109.

[142] Damiano Perri, Marco Simonetti, and Osvaldo Gervasi. "Deploying Efficiently Modern Applications on Cloud". In: *Electronics* 11.3 (2022). ISSN: 2079-9292. DOI: 10.3390/electronics11030450. URL: https://www.mdpi.com/2079-9292/11/3/450.

[143] Damiano Perri, Marco Simonetti, Sergio Tasso, Federico Ragni, and Osvaldo Gervasi. "Implementing a Scalable and Elastic Computing Environment Based on Cloud Containers". In: *Computational Science and Its Applications – ICCSA 2021*. Ed. by Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Chiara Garau, Ivan Blečić, David Taniar, Bernady O. Apduhan, Ana Maria A. C. Rocha, Eufemia Tarantino, and Carmelo Maria Torre. Cham: Springer International Publishing, 2021, pp. 676–689. ISBN: 978-3-030-86653-2.

[144] Osvaldo Gervasi, Damiano Perri, and Marco Simonetti. "Strategies and System Implementations for Secure Electronic Written Exams". In: *IEEE Access* 10 (2022), pp. 20559–20570. DOI: 10.1109/ACCESS.2022.3150860. URL: https://doi.org/10.1109/ACCESS.2022.3150860.

[145] Antonio Laganà, Osvaldo Gervasi, Sergio Tasso, Damiano Perri, and Francesco Franciosa. "The ECTN Virtual Education Community Prosumer Model for Promoting and Assessing Chemical Knowledge". In: *Computational Science and Its Applications – ICCSA 2018*. Ed. by Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Elena Stankova, Carmelo M. Torre, Ana Maria A.C. Rocha, David Taniar, Bernady O.

Apduhan, Eufemia Tarantino, and Yeonseung Ryu. Cham: Springer International Publishing, 2018, pp. 533–548. ISBN: 978-3-319-95174-4.

[146] Osvaldo Gervasi Damiano Perri. “Predisposizione del software EOL ai test di certificazione delle abilità linguistiche”. In: *University of Perugia, Italy* Bachelor degree thesis (2016).

[147] Mohammed A Al-Masni, Mugahed A Al-Antari, Jeong-Min Park, Geon Gi, Tae-Yeon Kim, Patricio Rivera, Edwin Valarezo, Mun-Taek Choi, Seung-Moo Han, and Tae-Seong Kim. “Simultaneous detection and classification of breast masses in digital mammograms via a deep learning YOLO-based CAD system”. In: *Computer methods and programs in biomedicine* 157 (2018), pp. 85–94.

[148] Victor Chang. “Towards a big data system disaster recovery in a private cloud”. In: *Ad Hoc Networks* 35 (2015), pp. 65–82.

[149] Mohammad Ali Khoshkholghi, Azizol Abdullah, Rohaya Latip, Shamala Subramaniam, and Mohamed Othman. “Disaster Recovery in Cloud Computing: A Survey”. In: *Computer and Information Science* 7 (Sept. 2014). DOI: 10.5539/cis.v7n4p39.

[150] Siham Hamadah. “Cloud-based disaster recovery and planning models: An overview”. In: *ICIC Express Letters* 13.7 (2019), pp. 593–599.

[151] Klaithem Al Nuaimi, Nader Mohamed, Mariam Al Nuaimi, and Jameela Al-Jaroodi. “A survey of load balancing in cloud computing: Challenges and algorithms”. In: *2012 second symposium on network cloud computing and applications*. IEEE. 2012, pp. 137–142.

[152] Sambit Kumar Mishra, Bibhudatta Sahoo, and Priti Paramita Parida. “Load balancing in cloud computing: a big picture”. In: *Journal of King Saud University-Computer and Information Sciences* 32.2 (2020), pp. 149–158.

[153] Mohammadreza Mesbahi and Amir Masoud Rahmani. “Load balancing in cloud computing: a state of the art survey”. In: *International Journal of Modern Education and Computer Science* 8.3 (2016), p. 64.

[154] Houcine Matallah, Ghalem Belalem, and Karim Bouamrane. “Evaluation of NoSQL databases: MongoDB, Cassandra, HBase, Redis, Couchbase, OrientDB”. In: *International Journal of Software Science and Computational Intelligence (IJSSCI)* 12.4 (2020), pp. 71–91.

[155] Sasalak Tongkaw and Aumnat Tongkaw. “A comparison of database performance of MariaDB and MySQL with OLTP workload”. In: *2016 IEEE conference on open systems (ICOS)*. IEEE. 2016, pp. 117–119.

[156] Mark Zaslavskiy, Alexander Kaluzhniy, Tatyana Berlenko, Ilfat Kinyaev, Kirill Krinkin, and Timofey Turenko. “Full automated continuous integration and testing infrastructure for MaxScale and MariaDB”. In: *2016 19th Conference of Open Innovations Association (FRUCT)*. IEEE. 2016, pp. 273–278.

[157] George Feuerlicht and Jaroslav Pokornỳ. “Can relational DBMS scale up to the cloud?” In: *Information Systems Development*. Springer, 2013, pp. 317–328.

[158] Carl Boettiger. “An introduction to Docker for reproducible research”. In: *ACM SIGOPS Operating Systems Review* 49.1 (2015), pp. 71–79.

[159] James Turnbull. *The Docker Book: Containerization is the new virtualization*. James Turnbull, 2014.

[160] Luis M Vaquero, Luis Rodero-Merino, and Rajkumar Buyya. “Dynamically scaling applications in the cloud”. In: *ACM SIGCOMM Computer Communication Review* 41.1 (2011), pp. 45–52.

[161] Pavlos Papadopoulos, Nikolaos Pitropakis, William J Buchanan, Owen Lo, and Sokratis Katsikas. “Privacy-Preserving Passive DNS”. In: *Computers* 9.3 (2020), p. 64.

[162] Gustavo Betarte, Eduardo Giménez, Rodrigo Martínez, and Álvaro Pardo. *Machine learning-assisted virtual patching of web applications*. 2018. arXiv: 1803.05529 [cs.CR].

[163] Yong Jin, Masahiko Tomoishi, Satoshi Matsuura, and Yoshiaki Kitaguchi. “A secure container-based backup mechanism to survive destructive ransomware attacks”. In: *2018 International Conference on Computing, Networking and Communications (ICNC)*. IEEE. 2018, pp. 1–6.

[164] Federico Giorgetti Osvaldo Gervasi and Antonio Laganà. *Distance Assessment System for Accreditation of Competencies and Skills Acquired Through In-Company Placements (DASP)*. INET ’99, The Internet Global Summit, 9th INET International Conference, 22-25 June 1999 McEnery Convention Center San José, CA, USA. 1999. URL: https://web.archive.org/web/20100311045420/http://www.isoc.org/isoc/conferences/inet/99/proceedings/posters.htm.

[165] Osvaldo Gervasi and Antonio Laganà. “EoL: A Web-Based Distance Assessment System”. In: *Computational Science and Its Applications – ICCSA 2004*. Ed. by Antonio Laganá, Marina L. Gavrilova, Vipin Kumar, Youngsong Mun, C. J. Kenneth Tan, and Osvaldo Gervasi. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 854–862. ISBN: 978-3-540-24709-8.

[166] Osvaldo Gervasi, Riccardo Catanzani, Antonio Riganelli, and Antonio Laganà. “Integrating Learning and Assessment Using the Semantic Web”. In: *Computational Science and Its Applications – ICCSA 2005*. Ed. by Osvaldo Gervasi, Marina L. Gavrilova, Vipin Kumar, Antonio Laganà, Heow Pueh Lee, Youngsong Mun, David Taniar, and Chih Jeng Kenneth Tan. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 921–927. ISBN: 978-3-540-32043-2.

[167] Avraham Leff and James T Rayfield. “Web-application development using the model/view/controller design pattern”. In: *Proceedings fifth ieee international enterprise distributed object computing conference*. IEEE. 2001, pp. 118–127.

[168] Luke Jefferson and Richard Harvey. “An Interface to Support Color Blind Computer Users”. In: *Proceedings of the SIGCHI Conference on*

Human Factors in Computing Systems. CHI '07. San Jose, California, USA: Association for Computing Machinery, 2007, pp. 1535–1538. ISBN: 9781595935939. DOI: 10.1145/1240624.1240855. URL: https://doi.org/10.1145/1240624.1240855.

[169] N Gordon. "Colour blindness". In: *Public Health* 112.2 (1998), pp. 81–84. ISSN: 0033-3506. DOI: https://doi.org/10.1038/sj.ph.1900446. URL: https://www.sciencedirect.com/science/article/pii/S0033350698005903.

[170] M Marmor. "Vision, eye disease, and art: 2015 Keeler Lecture". In: *Eye (London, England)* 30 (Nov. 2016), pp. 287–303. DOI: 10.1038/eye.2015.197.

[171] Luigi Atzori, Antonio Iera, and Giacomo Morabito. "The Internet of Things: A survey". In: *Computer Networks* 54.15 (2010), pp. 2787–2805. ISSN: 1389-1286. DOI: https://doi.org/10.1016/j.comnet.2010.05.010. URL: https://www.sciencedirect.com/science/article/pii/S1389128610001568.

[172] Luca Mainetti, Luigi Patrono, and Antonio Vilei. "Evolution of wireless sensor networks towards the internet of things: A survey". In: *SoftCOM 2011, 19th international conference on software, telecommunications and computer networks*. IEEE. 2011, pp. 1–6.

[173] Wazir Zada Khan, Mohammed Y. Aalsalem, Muhammad Khurram Khan, and Quratulain Arshad. "Enabling Consumer Trust Upon Acceptance of IoT Technologies Through Security and Privacy Model". In: *Advanced Multimedia and Ubiquitous Engineering*. Ed. by James J. (Jong Hyuk) Park, Hai Jin, Young-Sik Jeong, and Muhammad Khurram Khan. Singapore: Springer Singapore, 2016, pp. 111–117. ISBN: 978-981-10-1536-6.

[174] Duncan McFarlane, Sanjay Sarma, Jin Lung Chirn, ChienYaw Wong, and Kevin Ashton. "Auto ID systems and intelligent manufacturing control". In: *Engineering Applications of Artificial Intelligence* 16.4 (2003), pp. 365–376.

[175] Ray Y Zhong, Xun Xu, Eberhard Klotz, and Stephen T Newman. "Intelligent manufacturing in the context of industry 4.0: a review". In: *Engineering* 3.5 (2017), pp. 616–630.

[176] Valentina Franzoni, Sergio Tasso, Simonetta Pallottelli, and Damiano Perri. "Sharing Linkable Learning Objects with the Use of Metadata and a Taxonomy Assistant for Categorization". In: *Computational Science and Its Applications – ICCSA 2019*. Ed. by Sanjay Misra, Osvaldo Gervasi, Beniamino Murgante, Elena Stankova, Vladimir Korkhov, Carmelo Torre, Ana Maria A.C. Rocha, David Taniar, Bernady O. Apduhan, and Eufemia Tarantino. Cham: Springer International Publishing, 2019, pp. 336–348. ISBN: 978-3-030-24296-1.

[177] Xifan Yao, Jiajun Zhou, Jiangming Zhang, and Claudio R Boër. "From intelligent manufacturing to smart manufacturing for industry 4.0 driven by next generation artificial intelligence and further on". In:

2017 5th international conference on enterprise systems (ES). IEEE. 2017, pp. 311–318.

[178] Yueting Wang. “Industrial structure technology upgrade based on 5G network service and IoT intelligent manufacturing”. In: *Microprocessors and Microsystems* 81 (2021), p. 103696.

[179] Soham Adhya, Dipak Saha, Abhijit Das, Joydip Jana, and Hiranmay Saha. “An IoT based smart solar photovoltaic remote monitoring and control unit”. In: *2016 2nd international conference on control, instrumentation, energy & communication (CIEC)*. IEEE. 2016, pp. 432–436.

[180] S Pallavi, Jayashree D Mallapur, and Kirankumar Y Bendigeri. “Remote sensing and controlling of greenhouse agriculture parameters based on IoT”. In: *2017 International Conference on Big Data, IoT and Data Science (BID)*. IEEE. 2017, pp. 44–48.

[181] Ananda Mohon Ghosh, Debashish Halder, and SK Alamgir Hossain. “Remote health monitoring system through IoT”. In: *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*. IEEE. 2016, pp. 921–926.

[182] Abdullah Na, William Isaac, Shashank Varshney, and Ekram Khan. “An IoT based system for remote monitoring of soil characteristics”. In: *2016 International Conference on Information Technology (InCITe)-The Next Generation IT Summit on the Theme-Internet of Things: Connect your Worlds*. IEEE. 2016, pp. 316–320.

[183] Ravi Kishore Kodali, Vishal Jain, Suvadeep Bose, and Lakshmi Boppana. “IoT based smart security and home automation system”. In: *2016 international conference on computing, communication and automation (ICCCA)*. IEEE. 2016, pp. 1286–1289.

[184] D Pavithra and Ranjith Balakrishnan. “IoT based monitoring and control system for home automation”. In: *2015 global conference on communication technologies (GCCT)*. IEEE. 2015, pp. 169–173.

[185] Miao Hu, Xianzhuo Luo, Jiawen Chen, Young Choon Lee, Yipeng Zhou, and Di Wu. “Virtual reality: A survey of enabling technologies and its applications in IoT”. In: *Journal of Network and Computer Applications* (2021), p. 102970.

[186] Michele De Donno, Koen Tange, and Nicola Dragoni. “Foundations and Evolution of Modern Computing Paradigms: Cloud, IoT, Edge, and Fog”. In: *IEEE Access* 7 (2019), pp. 150936–150948. DOI: 10.1109/ACCESS.2019.2947652.

[187] Damiano Perri, Marco Simonetti, Alex Bordini, Simone Cimarelli, and Osvaldo Gervasi. “IoT to Monitor People Flow in Areas of Public Interest”. In: *Computational Science and Its Applications – ICCSA 2021*. Ed. by Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Chiara Garau, Ivan Blečić, David Taniar, Bernady O. Apduhan, Ana Maria A. C. Rocha, Eufemia Tarantino, and Carmelo Maria Torre.

Cham: Springer International Publishing, 2021, pp. 658–672. ISBN: 978-3-030-87016-4.

[188] Xuan Hung Le, Sungyoung Lee, Phan Tran Ho Truc, La The Vinh, Asad Masood Khattak, Manhyung Han, Dang Viet Hung, Mohammad M. Hassan, Miso Kim, Kyo-Ho Koo, Young-Koo Lee, and Eui-Nam Huh. "Secured WSN-integrated cloud computing for u-Life Care". In: *2010 7th IEEE Consumer Communications and Networking Conference*. 2010, pp. 1–2. DOI: 10.1109/CCNC.2010.5421618.

[189] Mengda Jia, Ali Komeily, Yueren Wang, and Ravi S Srinivasan. "Adopting Internet of Things for the development of smart buildings: A review of enabling technologies and applications". In: *Automation in Construction* 101 (2019), pp. 111–126.

[190] Shuo Tian, Wenbo Yang, Jehane Michael Le Grange, Peng Wang, Wei Huang, and Zhewei Ye. "Smart healthcare: making medical care more intelligent". In: *Global Health Journal* 3.3 (2019), pp. 62–65.

[191] Banu Çalış Uslu, Ertuğ Okay, and Erkan Dursun. "Analysis of factors affecting IoT-based smart hospital design". In: *Journal of Cloud Computing* 9.1 (2020), pp. 1–23.

[192] C Senthamilarasi, J Jansi Rani, B Vidhya, and H Aritha. "A smart patient health monitoring system using IoT". In: *International Journal of Pure and Applied Mathematics* 119.16 (2018), pp. 59–70.

[193] Othmane Friha, Mohamed Amine Ferrag, Lei Shu, Leandros Maglaras, and Xiaochan Wang. "Internet of things for the future of smart agriculture: A comprehensive survey of emerging technologies". In: *IEEE/CAA Journal of Automatica Sinica* 8.4 (2021), pp. 718–752.

[194] Xing Yang, Lei Shu, Jianing Chen, Mohamed Amine Ferrag, Jun Wu, Edmond Nurellari, and Kai Huang. "A survey on smart agriculture: Development modes, technologies, and security and privacy challenges". In: *IEEE/CAA Journal of Automatica Sinica* 8.2 (2021), pp. 273–302.

[195] Fengxian Guo, F Richard Yu, Heli Zhang, Xi Li, Hong Ji, and Victor CM Leung. "Enabling massive IoT toward 6G: A comprehensive survey". In: *IEEE Internet of Things Journal* 8.15 (2021), pp. 11891–11915.

[196] Yi Xu, Sumi Helal, Choonhwa Lee, and Ahmed Khaled. "Energy savings in very large cloud-iot systems". In: (2019).

[197] Georg Ferdinand Schneider and Mads Holten Rasmussen. *TechnicalBuildingSystems/OpenSmartHomeData: First release of Open Smart Home Data Set*. Version v1.0.0. May 2018. DOI: 10.5281/zenodo.1244602. URL: https://doi.org/10.5281/zenodo.1244602.

[198] Schneider, Georg Ferdinand and Rasmussen, Mads Holten and Bonsma, Peter and Oraskari, Jyrki and Pauwels, Pieter. "Linked building data for modular building information modelling of a smart home". eng. In: *EWORK AND EBUSINESS IN ARCHITECTURE, ENGINEERING AND CONSTRUCTION*. Ed. by Karlshoj, Jan and

Scherer, Raimar. Copenhagen, Denmark: CRC Press, 2018, 407–414. ISBN: 9780429506215.

[199] Yunsick Sung, Abdelsalam Helal, Jae Woong Lee, and Kyungeun Cho. "Bayesian-Based Scenario Generation Method for Human Activities". In: *Proceedings of the 1st ACM SIGSIM Conference on Principles of Advanced Discrete Simulation*. SIGSIM PADS '13. Montral, Quebec, Canada: Association for Computing Machinery, 2013, pp. 147–158. ISBN: 9781450319201. DOI: 10.1145/2486092.2486111. URL: https://doi.org/10.1145/2486092.2486111.

[200] Yi Xu and Sumi Helal. "Application Caching for Cloud-Sensor Systems". In: *Proceedings of the 17th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*. MSWiM '14. Montreal, QC, Canada: Association for Computing Machinery, 2014, pp. 303–306. ISBN: 9781450330305. DOI: 10.1145/2641798.2641814. URL: https://doi.org/10.1145/2641798.2641814.

[201] Xuefeng Ding and Jiang Wu. "Study on Energy Consumption Optimization Scheduling for Internet of Things". In: *IEEE Access* 7 (2019), pp. 70574–70583. DOI: 10.1109/ACCESS.2019.2919769.

[202] Halah Mohammed Al-Kadhim and Hamed S. Al-Raweshidy. "Energy Efficient Data Compression in Cloud Based IoT". In: *IEEE Sensors Journal* 21.10 (2021), pp. 12212–12219. DOI: 10.1109/JSEN.2021.3064611.

[203] Kapal Dev, Praveen Kumar Reddy Maddikunta, Thippa Reddy Gadekallu, Sweta Bhattacharya, Pawan Hegde, and Saurabh Singh. "Energy Optimization for Green Communication in IoT Using Harris Hawks Optimization". In: *IEEE Transactions on Green Communications and Networking* 6.2 (2022), pp. 685–694. DOI: 10.1109/TGCN.2022.3143991.

[204] Zhi Zhou, Shuai Yu, Wuhui Chen, and Xu Chen. "CE-IoT: Cost-Effective Cloud-Edge Resource Provisioning for Heterogeneous IoT Applications". In: *IEEE Internet of Things Journal* 7.9 (2020), pp. 8600–8614. DOI: 10.1109/JIOT.2020.2994308.

[205] Hamed Shah-Mansouri and Vincent W. S. Wong. "Hierarchical Fog-Cloud Computing for IoT Systems: A Computation Offloading Game". In: *IEEE Internet of Things Journal* 5.4 (2018), pp. 3246–3257. DOI: 10.1109/JIOT.2018.2838022.

[206] Bing Lin, Yinhao Huang, Jianshan Zhang, Junqin Hu, Xing Chen, and Jun Li. "Cost-Driven Off-Loading for DNN-Based Applications Over Cloud, Edge, and End Devices". In: *IEEE Transactions on Industrial Informatics* 16.8 (2020), pp. 5456–5466. DOI: 10.1109/TII.2019.2961237.

[207] Mehar Ullah, Pedro H. J. Nardelli, Annika Wolff, and Kari Smolander. "Twenty-One Key Factors to Choose an IoT Platform: Theoretical Framework and Its Applications". In: *IEEE Internet of Things*

Journal 7.10 (2020), pp. 10111–10119. DOI: 10.1109/JIOT.2020.3000056.

[208] Amrit Mukherjee, Pratik Goswami, Mohammad Ayoub Khan, Li Manman, Lixia Yang, and Prashant Pillai. "Energy-Efficient Resource Allocation Strategy in Massive IoT for Industrial 6G Applications". In: *IEEE Internet of Things Journal* 8.7 (2021), pp. 5194–5201. DOI: 10.1109/JIOT.2020.3035608.

[209] Marcin Woźniak, Adam Zielonka, Andrzej Sikora, Md. Jalil Piran, and Atif Alamri. "6G-Enabled IoT Home Environment Control Using Fuzzy Rules". In: *IEEE Internet of Things Journal* 8.7 (2021), pp. 5442–5452. DOI: 10.1109/JIOT.2020.3044940.

[210] Laha Ale, Ning Zhang, Xiaojie Fang, Xianfu Chen, Shaohua Wu, and Longzhuang Li. "Delay-Aware and Energy-Efficient Computation Offloading in Mobile-Edge Computing Using Deep Reinforcement Learning". In: *IEEE Transactions on Cognitive Communications and Networking* 7.3 (2021), pp. 881–892. DOI: 10.1109/TCCN.2021.3066619.

[211] Michael Pecht. "Prognostics and Health Management of Electronics". In: *Encyclopedia of Structural Health Monitoring*. John Wiley and Sons, Ltd, 2009. ISBN: 9780470061626. DOI: https://doi.org/10.1002/9780470061626.shm118. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/9780470061626.shm118. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470061626.shm118.