UNIVERSITÀ
DEGLI STUDI
FIRENZE

PHD PROGRAM IN SMART COMPUTING
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE (DINFO)

Anthropomorphous Visual Recognition: Learning with Weak Supervision, with Scarce Data, and Incrementally over Transient Tasks

Riccardo Del Chiaro

Dissertation presented in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Smart Computing

PhD Program in Smart Computing
University of Florence, University of Pisa, University of Siena

Anthropomorphous Visual Recognition: Learning with Weak Supervision, with Scarce Data, and Incrementally over Transient Tasks

Riccardo Del Chiaro

Advisor:

Prof. Andrew D. Bagdanov
Dr. Lorenzo Seidenari

Head of the PhD Program:

Prof. Paolo Frasconi

Evaluation Committee:
Prof. Tinne Tuytelaars, *Katholieke Universiteit Leuven*
Prof. Lamberto Ballan, *Università degli Studi di Padova*

XXXIII ciclo — October 2021

Ringraziamenti

Ho iniziato il mio percorso nel mondo della ricerca nell'ambito del Machine Learning, così come tutto il mio percorso nell'informatica, seguendo la mia curiosità verso quelle cose talmente complesse da risultare affascinanti e misteriose, come il funzionamento di un computer dall'elettronica fino al sistema operativo o di un algoritmo in grado di riconoscere oggetti all'interno di immagini attraverso neuroni artificiali.

Voglio quindi per prima cosa ringraziare chi ha consentito che potesse nascere e che potessi perseguire tale passione: i miei genitori e mio fratello, che mi hanno supportato durante tutti questi anni e mi hanno sin da piccolissimo avvicinato all'elettronica, ai computers ed ai videogiochi, incuriosendomi a tal punto da trovare l'iniziativa per iniziare a studiare autonomamente i linguaggi di programmazione prima ancora di iniziare le scuole superiori.

Ringrazio tutti i professori che sono riusciti a trasmettermi la loro passione e ad incalanare in modo costruttivo la mia. A partire da Gianluca Braccini, che ha gettato le basi della mia cultura nell'ambito della programmazione durante la mia adolescenza. Ringrazio Alberto Del Bimbo, che è riuscito a mettere in piedi ed a portare avanti un laboratorio di eccellenza nel cuore di Firenze in cui gravitano tantissime persone, passione e conoscenza. Ringrazion Paolo Frasconi, che è riuscito per primo ad appassionarmi all'Artificial Intelligence col suo corso tenuto alla triennale; studiando per il suo esame capii per la prima volta che avrei potuto fare qualcosa di diverso dalla pura ingegneria del software. Ringrazio Marco Bertini per ciò che mi ha insegnato nei suoi corsi, dalla triennale fino ai corsi di dottorato, ed insieme a lui ringrazio anche Stefano Berretti per avermi dato entrambi la bellissima opportunità di aiutare in aula durante i corsi di *Programmazione* e di *Fondamenti* del primo anno: l'esperienza di insegnare in un'aula universitaria è stata qualcosa di impagabile.

Ringrazio tutti i ragazzi del MICC con cui ho condiviso in prima persona questa esperienza: Matteo Bruni, Federico Becattini, Claudio Baecchi, Leonardo Galtieri, Claudio Ferrari, Francesco Turchini, Kiew My, Simone Ricci, Andrea Salvi, João Baptista Cardia Neto, Pietro Bongini, Francesco Marchetti, Federico Pernici, Filippo Principi, Lorenzo Seidenari, Tiberio Uricchio, Giuseppe Becchi, Andrea Ferracani, Pietro Pala, Roberto Caldelli, e tutti gli altri professori e ricercatori con cui ho condiviso questa esperienza, aiutandoci a vicenda e festeggiando con pranzi o aperitivi ad ogni occasione. Molti di voi sono stati uno stimolo e un modello, ed è stato grazie a delle semplici chiacchierate davanti ad un caffé in corridoio che sono venuti fuori consigli ed osservazioni importanti, idee e stimoli per andare avanti con nuova energia nel complesso mondo della ricerca.

Ringrazio Andrew, che si è sin da subito rivelato un professore straordinario e assolutamente fuori dal comune, col quale ho svolto con passione la mia tesi magistrale ad Amsterdam ed a Firenze. È grazie a te che ho scelto di dedicare questi miei anni alla ricerca, e mi ritengo molto fortunato di averlo potuto fare sotto la tua supervisione. Ringrazio tantissimo Joost van de Weijer che mi ha accolto al CVC e con cui ho avuto la fortuna di collaborare a strettissimo contatto. Nonostante le difficoltà del lockdown in Spagna ho un bellissimo ricordo di quel periodo, ed è stata probabilmente l'esperienza di ricerca più intensa e coinvolgente che abbia mai fatto in questi tre anni.

Ringrazio i ragazzi del CVC, per primo Bartłomiej Twardowski, con cui ho avuto il piacere di collaborare, e David Berga con cui ho avuto la fortuna di poter festeggiare a Firenze l'accettazione del mio lavoro svolto a Barcellona, pubblicato su *NeurIPS*. E ringrazio tutti gli altri ragazzi che ho incontrato e che mi hanno accolto al CVC come un compagno. Purtroppo la pandemia ha impedito che potessimo conoscerci meglio. Non c'è molto altro da dire se non che è stato davvero un peccato.

Ringrazio chi mi è stato accanto in questi anni. Giulia Alfani, amica fidata su cui ho sempre potuto contare e che mi ha sempre spronato al miglioramento e alla crescita. Giulia Giannetti, che nonostante tutto ha saputo ascoltare e capirmi. Deborah Grifagni, con cui reciprocamente ho condiviso pensieri e difficoltà riguardanti ricerca scientifica e non, in un periodo estremamente complesso ed incerto per entrambi come può esserlo l'inizio e la conclusione di un percorso di dottorato durante una pandemia globale. Silvio Agresti, che mi ha dato nuove prospettive e nuova linfa vitale grazie alla sua infinita energia e intraprendenza, rivelandosi un vero amico prima ancora che un socio, e col quale ho avuto la fortuna di iniziare uno sfortunatissimo (almeno per ora) percorso imprenditoriale. Sabrina Cortese, che mi è stata accanto con grande pazienza e con la sua dirompente energia positiva, nonostante tutte le difficoltà ed incertezze che ho mostrato alla fine di questo mio percorso. Andrea Pancani, figura di riferimento per ogni mio smarrimento politco e costante opportunità per alte discussioni più o meno surreali davanti a birre sulle sponde del *Mugnone*, nel cuore delle *Cure* a Firenze.

Un ringraziamento speciale va ai miei più cari amici, che durante questi anni sono stati per me come una famiglia: Reza Raissi, Norane Mungly, Lucia Lanzi, Lucrezia Feriti, Pompeo Pedicini, Vlad Sanda, Cosimo Ciofalo, Andrei Pistol, Andrea Sorrentino, Marco Casini, Luca de Roma (Bondì). È difficile riassumere in due righe ciò che avete rappresentato per me in questi anni. Risate, evasione, supporto, confronto, sbronze, ma soprattutto stabilità, sicurezza ed emozioni positive. Senza di voi il mio percorso sarebbe stato molto più triste e difficile.

Acknowledgments

I started my journey in the world of Machine Learning research, as well as my entire computer science journey, following my curiosity towards those things so complex as to be fascinating and mysterious, such as the functioning of a computer from the electronics to the operating system or of an algorithm capable of recognizing objects within images through artificial neurons.

First of all, I want to thank those who allowed this passion to be born: my parents and my brother, who introduced me to electronics, computers and video games from an early age, intriguing me to the point of finding the initiative to start studying programming languages independently before starting high school.

First of all I want to thank those who allowed this passion to be born and that I have been able to carry it on over the years: my parents and my brother, who have supported me during all these years and have approached me from an early age to electronics, computers and video games, intriguing me to the point of finding the initiative to start studying programming languages independently before starting high school.

I thank all the professors who have managed to trasnfer their passion and to constructively channel mine. Starting with Gianluca Braccini, who laid the foundations of my culture in programming during my adolescence. I thank Alberto Del Bimbo, who managed to set up and carry on a laboratory of excellence in the heart of Florence where many people, passion and knowledge gravitate around. I thank Paolo Frasconi, who managed to get me passionate about Artificial Intelligence with his lectures during the bachelor; studying for his exam I realized for the first time that I could do something else besides pure software engineering. I thank Marco Bertini for what he taught me in his lectures, from bachelor to doctoral courses, and together with him I also thank Stefano Berretti for giving me both the wonderful opportunity to lecturing during the coruses of *Programming* and *Fondamentals of Programming* for the first year students: the experience of teaching in a university classroom was something priceless.

I thank all the guys from the MICC with whom I personally shared this experience: Matteo Bruni, Federico Becattini, Claudio Baecchi, Leonardo Galtieri, Claudio Ferrari, Francesco Turchini, Kiew My, Simone Ricci, Andrea Salvi, João Baptista Cardia Neto, Pietro Bongini, Francesco Marchetti, Federico Pernici, Filippo Principi, Lorenzo Seidenari, Tiberio Uricchio, Giuseppe Becchi, Andrea Ferracani, Pietro Pala, Roberto Caldelli, and all the other professors and researchers with whom I have shared this experience, helping each other and celebrating with lunches or drinks at every occasion. Many of you have been a stimulus and a model, and it was thanks to simple chats in front of a coffee in the corridor that important advice and observations, ideas and stimuli came out to move forward with new energy in the complex world of research.

I thank Andrew, who immediately proved to be an extraordinary and absolutely out of the ordinary professor, with whom I passionately carried out my master's thesis in Amsterdam and Florence. It is thanks to you that I have chosen to dedicate these years to research, and I feel very lucky to have been able to do so under your supervision. Thanks a lot to Joost van de Weijer who welcomed me to the CVC and with whom I was lucky enough to collaborate in very close contact. Despite the difficulties given by the lockdown in Spain, I have wonderful memories of that period, and it was probably the most intense and engaging research experience I have ever done in these three years.

I thank the CVC guys, first Bartłomiej Twardowski, with whom I had the pleasure of collaborating, and David Berga with whom I was lucky enough to be able to celebrate in Florence the acceptance of my work done in Barcelona, published on *NeurIPS*. And I thank all the other guys I met and who welcomed me to CVC as a mate. Unfortunately, the pandemic has prevented us from getting to know each other better. There is not much more to say except that it was really a shame.

I thank those who have been close to me over the years. Giulia Alfani, a trusted friend on whom I have always been able to count and who has always encouraged me to improve and grow. Giulia Giannetti, who despite everything was able to listen and understand me. Deborah Grifagni, with whom I mutually shared thoughts and difficulties also concerning scientific research in an extremely complex and uncertain period for both of us, as the beginning and the end of a PhD program during a global pandemic. Silvio Agresti, who gave me new perspectives and new lifeblood thanks to his unlimited energy and resourcefulness, revealing himself to be a true friend even before a business partner, and with whom I was lucky enough to found a very unfortunate (at least for now) start-up. Sabrina Cortese, who stood by me with great patience and with her disruptive positive energy, despite all the difficulties and uncertainties that I showed at the end of my PhD journey. Andrea Pancani, a reference figure for all my political bewilderments and a constant opportunity for more or less surreal discussions in front of beers on the edge of *Mugnone*, in the heart of *Cure* district in Florence.

Special thanks go to my dearest friends that during these years have been like a family for me: Reza Raissi, Norane Mungly, Lucia Lanzi, Lucrezia Feriti, Pompeo Pedicini, Vlad Sanda, Cosimo Ciofalo, Andrei Pistol, Andrea Sorrentino, Marco Casini, Luca de Roma (Bondì). It is difficult to summarize what you have represented for me over these years. Laughter, evasion, support, confrontation, hangovers, but above all: stability, joy and positive emotions. Without you, my path would have been much more difficult and sadder.

Abstract

In the last eight years the computer vision field has experienced dramatic improvements thanks to the widespread availability of data and affordable parallel computing hardware like GPUs. These two factors have contributed to making possible the training of very deep neural network models in reasonable times using millions of labeled examples for supervision. Humans do not learn concepts in this way. We do not need a massive number of labeled examples to learn new concepts; instead we rely on a few (or even zero) examples, infer missing information, and generalize. Moreover, we retain previously learned concepts without the need to re-train. We can easily ride a bicycle after years of not doing so, or recognize an elephant even though we may not have seen one recently.

These characteristics of human learning, in fact, stand in stark contrast to how deep models learn: they require massive amounts of labeled data for training due to overparameterization, they have limited generalization capabilities, and they easily forget previously learned tasks or concepts when trained on new ones. These characteristics limit the applicability of deep learning in some scenarios in which these problems are more evident. In this thesis we study some of these and propose strategies to overcome some of the negative aspect of deep neural network training. We still use the gradient-based learning paradigm, but we adapt it to address some of these differences between human learning and learning in deep networks. Our goal is to achieve better learning characteristics and improve performance in some specific applications.

We first study the artwork instance recognition problem, for which it is very difficult to collect large collections of labeled images. Our proposed approach relies on web search engines to collect examples, which results in the two related problems of domain shift due to biases in search engines and noisy supervision. We propose several strategies to mitigate these problems. To better mimic the ability of humans to learn from compact semantic description of tasks, we then propose a zero-shot learning strategy to recognize never-seen artworks, instead relying solely on textual descriptions of the target artworks.

Then we look at the problem of learning from scarce data for the no-reference image quality assessment (NR-IQA) problem. IQA is an application for which data is notoriously scarce due to the elevated cost for annotation. Humans have an innate ability to inductively generalize from a limited number of examples, and to better mimic this we propose a generative model able to generate controlled perturbations of the input image, with the goal of synthetically increase the number of training instances used to train the network to estimate input image quality.

Finally, we focus on the problem of catastrophic forgetting in recurrent neural networks, using image captioning as problem domain. We propose two strategies for defining continual image captioning experimental protocols and

develop a continual learning framework for image captioning models based on encoder-decoder architectures. A task is defined by a set of object categories that appears in the images that we want the model to be able to describe. We observe that catastrophic forgetting is even more pronounced in this setting and establish several baselines by adapting existing state-of-the-art techniques to our continual image captioning problem.

Then, to mimic the human ability to retain and leverage past knowledge when acquiring new tasks, we propose to use a mask-based technique that allocates specific neurons to each task only during backpropagation. This way, novel tasks do not interfere with the previous ones and forgetting is avoided. At the same time, past knowledge is exploited thanks to the ability of the network to use neurons allocated to previous tasks during the forward pass, which in turn reduces the number of neurons needed to learn each new task.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Visual perception is *the ability to interpret the surrounding environment using the light reflected by objects in the visible spectrum*. Culturally, the visual sense has been historically dominant and the first choice in the range of human sense even for philosophers beginning with Plato and Aristotle. Vision is probably the most important, complex, and developed human sense, and the number of studies about vision published in the psychology literature confirms this interest compared with the other senses. Gallace and Spence (2009) in an early study and Hutmacher (2019) in a recent one confirm this interest in vision in the psychology research community. The histograms in figure 1.1 illustrate the number of results returned for different sensory query in the PsycINFO database. Today, the importance of vision is even more prominent thanks to modern technologies able to record and process images and video even in spectra not visible to human eyes. This has led to significant improvements across all scientific fields, from medicine to astronomy.

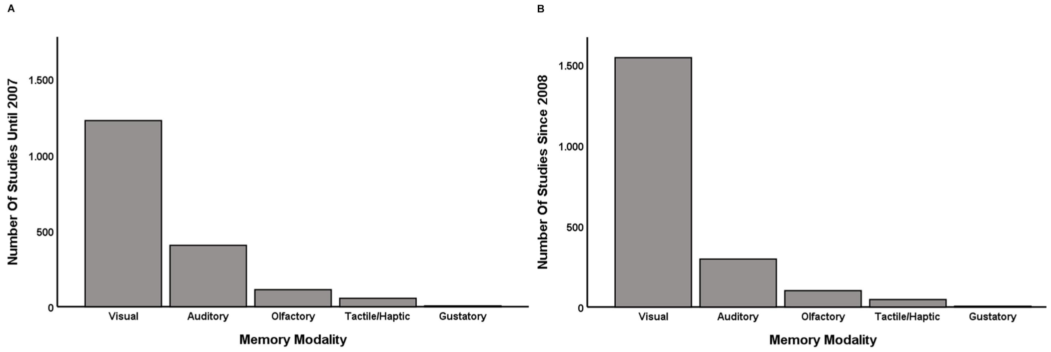


Figure 1.1: Histograms reporting the number of studies of different sensory modalities in the PsycINFO database (Gallace and Spence, 2009; Hutmacher, 2019).

1.1 Visual recognition and Convolutional Neural Networks

Computer Vision is concerned with how computers can achieve visual perception similar to that of humans. The goal is to make machines perform tasks that require visual perceptions and understanding that normally require human intervention, preventing automation. Most computer vision problems are tackled with pattern recognition and machine learning techniques. In particular, Convolutional Neural Networks (CNNs) are revolutionizing the field since the early 2010s, and their sophisticated application has led to solutions to a large range of problems once considered complex (if not impossible) to solve before Krizhevsky et al. (2012).

CNNs are a type of artificial neural network. There were invented and have been studied since the end of the 1980s (Waibel et al., 1989; LeCun et al., 1989, 1998). CNNs take images as input and mostly employ locally connected layers consisting of convolutions on input tensors instead of the fully-connected layers used in the Multi-layer Perceptron. The convolutional filters act as feature extractors and their weights are trained from scratch on a set of labeled images to learn the best settings for the given task. Because of the elevated number of free parameters that need to be learned, CNNs typically require a massive amount of labeled training data and significant computational power for training. These limitations have been mitigated in recent years largely thanks to the availability of data (Deng et al., 2009) and computational power in the form of Graphics Processing Units (GPUs). In fact, the

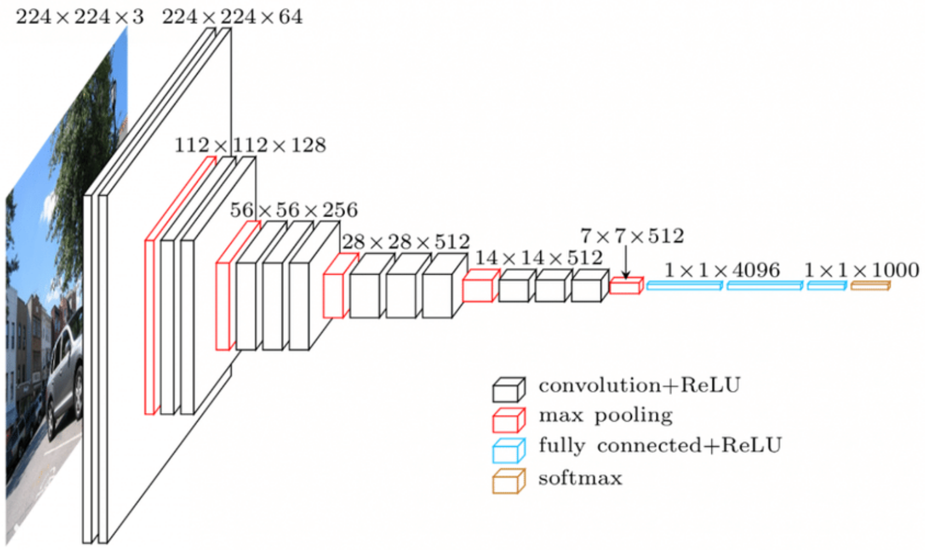


Figure 1.2: The VGG-16 network architecture from Simonyan and Zisserman (2014). Compared to AlexNet, eight additional convolutional layers were added while keeping the same number and size of fully connected layers at the end of the network.

number of image and videos available in the internet began to grow exponentially from the 2000s, and researchers now have access to massive datasets of images and videos. Meanwhile, since the early 2010s affordable parallel computing hardware like GPUs has been available for general purpose tasks like machine learning, partly thanks to frameworks like NVIDIA CUDA (Nickolls et al., 2008). The final piece in the modern deep learning puzzle was the decisive winning of the ILSVRC competition (Deng et al., 2009; Russakovsky et al., 2015a) by AlexNet (Krizhevsky et al., 2012). This showed it was possible to significantly improve on the state-of-the-art by training CNNs on GPUs in reasonable times. In the following years significant improvement in classification performance was obtained by employing deeper and deeper architectures, passing from the eight layers of AlexNet to the sixteen or nineteen layers of the VGG architectures (Simonyan and Zisserman, 2014) illustrated in figure 1.2, and reaching the astonishing number of 152 layers for the ResNet architecture (He et al., 2016).

Although CNNs have been widely employed in many applications, they still suffer from some of the issues that prevented their use at the beginning. First of all, CNNs are notoriously data hungry, requiring massive amounts of data paired with strong human supervision for training. If insufficient data is used for training it is not possible to exploit the entire network capacity and the model will overfit.

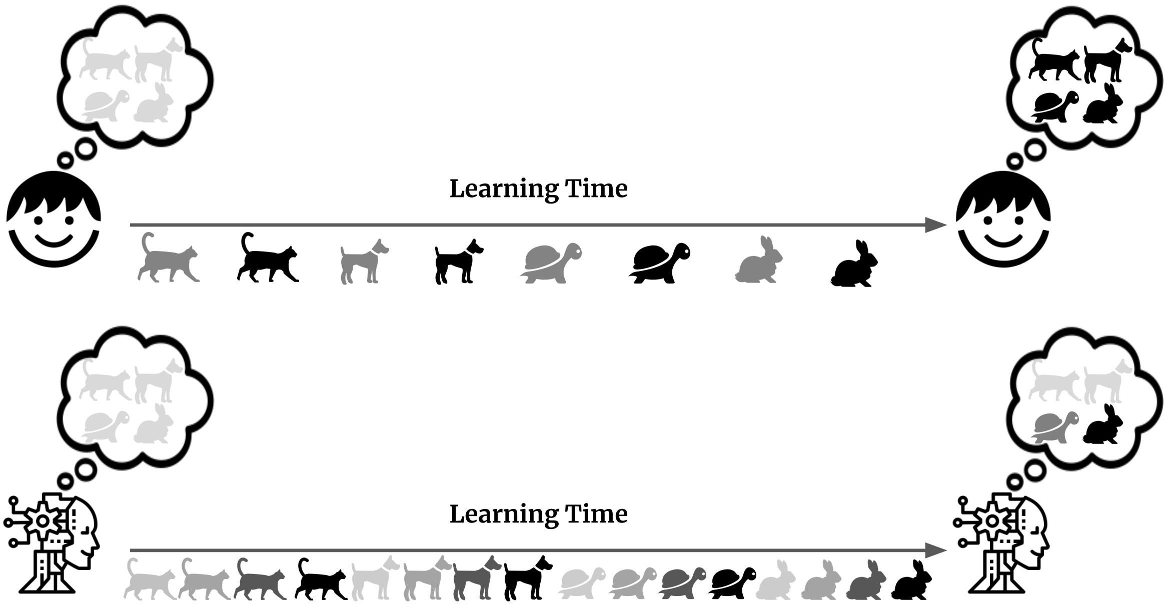


Figure 1.3: Humans can learn from very few examples due to an innate generalization ability. In contrast, machines require much more examples to correctly generalize and understand how to recognize a given category, exhibiting strong forgetting issues when trained sequentially even for tasks that seem trivial for humans.

Moreover, deep networks like CNNs cannot learn multiple tasks sequentially unless careful attention is paid to avoiding the effects of *catastrophic forgetting* in which the network forgets previous tasks when acquiring new ones (see figure 1.3).

These characteristics of learning in CNNs are in stark contrast with how humans learn: we do not need massive amounts of supervised training data to learn new concepts – we can generalize well with few supervised examples without overfitting. We can even acquire new concepts from compact semantic descriptions with *no* labeled training examples. Moreover, we are able to sequentially acquire multiple skills focusing on learning one at a time, without forgetting the previous ones but rather exploiting past knowledge when learning new ones.

Making machine learning more similar to human learning is not our final goal. We instead observe the differences and limitations of machines when compared to humans and try to address some aspects that are currently a limiting factor for certain applications. Some of these aspects can, in practice, be solved with brute-force methods: with more human effort for data collection and supervision or with multiple re-trainings of the network to prevent forgetting we can overcome some of these limitations. And this is often done in practice, directly or indirectly, when CNNs are applied to real word problems. But brute-force solutions are expensive and in certain scenarios could dramatically increase the cost and complexity of training, deployment, and maintenance – not to mention computational power and carbon emissions. Having CNNs that behave more similarly to humans in these aspects would help unlock their full potential and make them more flexible across a range of application scenarios.

In this thesis we focus on specific use cases in which vanilla deep learning paradigms are not effective, and we propose alternatives to minimize the negative effects. Specifically, we consider the following problems:

- **Webly-supervised learning for instance recognition**. Web search engines offer a wealth of multi-modal, contextualized information that is exploitable for training deep neural networks. To do so, however, we must understand how to exploit the *noisy* supervision provided by web search engines. We look at the problem of leveraging image search results to train artwork instance recognition models (chapter 2).

- **Zero-shot webly-supervised instance recognition**. It is relatively easy to automatically retrieve images along with *associated text* from web search engines. We look at the problem of zero-shot artwork instance recognition under noisy supervision (also in chapter 2).

- **Generative data augmentation for image quality assessment**. Image quality assessment is a niche problem for which it is extremely laborious and costly to collect the amounts of labeled data typically needed for training modern

CNN architectures. In order to render such deep models capable of generalizing with fewer labeled examples, we look at the potential of generative models to synthetically expand the number of examples by generating controlled, distorted images on-the-fly at training time (chapter 3).

- **Continual learning for image captioning**. We explore the *catastrophic interference* problem in the case of a recurrent neural network applied to natural language generation for image captioning. We propose a new framework for continual image captioning based on *transient* tasks, as well as dataset splitting procedures that can be used to define continual captioning experimental protocols. We propose an incremental learning framework based on masking in order to mitigate forgetting in recurrent networks (chapter 4).

In the following sections we take a closer look at these issues with CNN training.

1.2 Learning with weak supervision

Convolutional Neural Networks (CNNs), in contrast to humans, must typically be trained on massive supervised datasets. However, in some cases it can be difficult to supervise such large amounts of data due to the high cost of labeling. We might decide to reduce labeling costs at the expense of supervision quality. For example, for a problem that requires expert supervision we could decide to rely on less trained workers, or rely to some extent on software or other AI algorithms. In such cases, we are training with *weak supervision*, and Zhou (2018) identified three types:

- *Incomplete supervision*: we can split the training data in two subsets, the first one with labels and the second without.

- *Inaccurate supervision*: the given labels in the training set are not always ground truth, and we do not have information about which labels we can trust and which ones not.

- *Inexact supervision*: only coarse-grained labels are given, however we are interested in more fine-grained predictions.

Note that these three types of weak supervision can occur simultaneously.

We are interested in the *inaccurate* weak supervision regime, for which a typical scenario is label noise. This problem is usually tackled by the identification of potentially mislabeled examples and removing or relabeling suspicious instances. Outliers and anomaly detection techniques are useful for identifying problematic examples, and clustering can be used to identify the main modes of each class to eventually prune the under-represented or suspicious ones. In chapter 2 we show

how to exploit the supervision offered by web search engines to collect weakly-labeled images and train a classifier for an artwork instance recognition problem. We also describe the NoisyArt dataset which we collected and published specifically to foster research on the artwork instance recognition problem under noisy supervision.

1.2.1 Webly-supervised learning

When supervised data for a specific application is scarce, one could decide to rely on web search engines or social networks to obtain (pseudo) labeled examples and create a so called *webly-supervised* dataset (a pun on *web* and *weakly-supervised*). We consider web supervision as form of *inaccurate supervision*: we collect a dataset using various, inaccurate data sources which provide images paired with labels. The result is a collection of training sets, for each of which we have a different grade of trust in the labels (e.g. one set of results might have clean labels that we trust completely, while others might have lower grades of trust). Considering that we will normally query the web using label-related information, the actual noise is given by the retrieved instances that might include image results we do not expect.

This can be caused by inaccurate queries, by the inaccurate information associated to the data, by the total absence of the data in the source, or by the bias introduced by the web search engine itself. In fact, search engines tend to be high-precision/low-recall by design so that they tend to retrieve very iconic representations of the queried concept at the expense of diversity. This may result in problems reflected in the retrieved instances:

- *Outliers* or *labelflips*: the retrieved instance is something that does not represent the class. If it is an instance of another class in the dataset, we consider it as a valid instance with noisy label (*labelflip* noise), otherwise the example is considered an outlier.

- *Lack of diversity*: instances retrieved for a given class are all very similar. In this case even a high number of retrieved examples might not be sufficiently informative to train a classifier.

When identified, *labelflip* examples can be relabeled or exploited as unsupervised instances, while *outliers* can only be pruned completely. A *Lack of diversity*, however, brings us back to a data scarcity problem: we can only solve this by adding new data sources or making the compromise of manually collecting and supervising examples or, as last resort, decide to completely remove the class from the dataset.

In section 2.3 we applied these concepts to collect a multi-modal dataset of artwork instance images using web search engines. We proposed a soft outlier pruning technique that is able to mitigate the impact of both labelflip noise and outliers during the training of a deep network for instance recognition task. We also proposed a technique to identify problematic classes and prune a small number of them to boost performance of the remaining ones.

1.2.2 Zero-shot learning

In the context of computer vision, the goal of zero-shot learning (ZSL) is to recognize classes whose visual instances are never seen during training. This is only possible when each class is paired with extra information, for example a class attribute vector or a textual description of the class. Zero-shot learning is useful when we do not have information on what exactly we want to classify at test time, but it is also useful to reduce the supervision at class level instead of instance level for a subset of classes. Exploiting information associated at the class level is usually much cheaper. In practice, each class will be paired with a vector describing it (e.g. for the textual description case we can obtain a single vector per each class using a document embedding technique).

To tackle the zero-shot classification problem we should somehow bridge the gap between visual embedding and class descriptor spaces so that at inference time it is possible to exploit the information associated with test classes to recognize and assign to each test image one of them. One option for this is to project the class descriptor vectors into the visual feature space, or use metric learning approaches

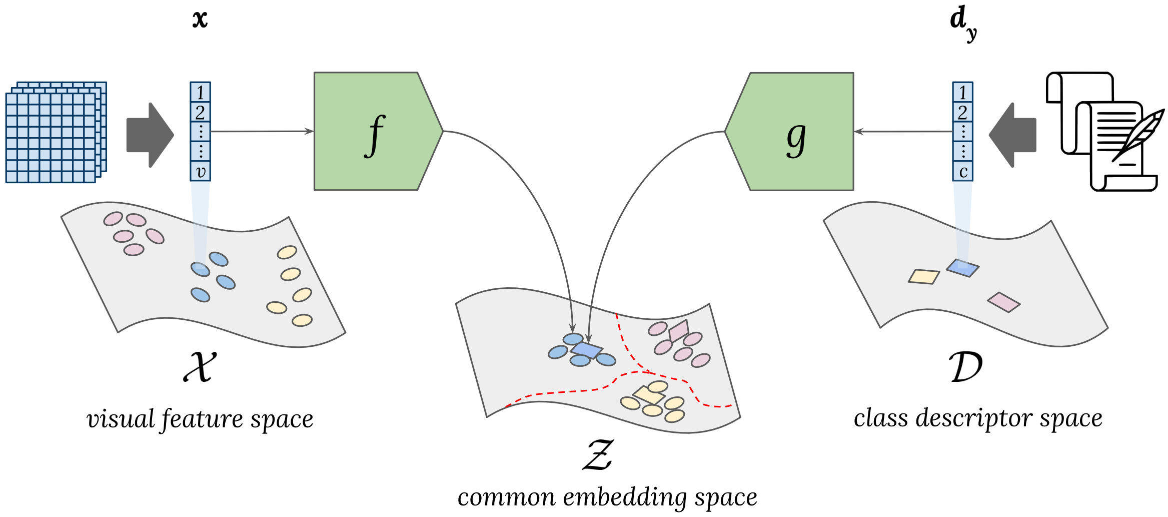


Figure 1.4: A visual representation of a zero-shot learning architecture employing metric learning techniques to bridge the gap between the visual space $\mathcal{X}$ and class descriptor space $\mathcal{D}$. Features coming from these two spaces are projected into a new common embedding space $\mathcal{Z}$.

to learn how to project both visual features and class descriptors into a new common embedding space (see figure 1.4).

In section 2 we applied these techniques to our instance recognition problem for artwork identification under noisy supervision. We use textual descriptions retrieved from the web and the webly-supervised images to train the zero-shot instance recognition model, and we show how the use of a large number of webly-supervised classes is helpful in boosting zero-shot recognition performance on never-seen classes. Note that zero-shot learning, though it is widely used for fine grained classification problems like Welinder et al. (2010), to our knowledge has not yet been applied to instance recognition problems before.

1.3 Learning with scarce data

Convolutional Neural Networks (CNNs) are usually trained on massive, fully-supervised datasets. For certain problems or applications there could be difficulties in collecting or supervising large amounts of data due to high cost of acquisition or labeling. In these cases, several strategies can be used to make the most of small amounts of available data.

1.3.1 Data augmentation and generative models

Data augmentation is a strategy used to artificially increase the number of training examples by applying small and controlled perturbations to the training set examples. Data augmentation is essential to squeeze the best possible generalization performance out of CNNs. Standard augmentations like image flipping, rotation, translation, and scaling have been shown useful for augmenting datasets for general image recognition (Krizhevsky et al., 2012; Chatfield et al., 2014), however such generic augmentations are less useful for more niche problems where relevant augmentations are less evident, or for regression problems where perturbations do not necessarily preserve the target output value.

Recently, generative models like Generative Adversarial Networks (GANs) have been used to artificially increase the number of examples in datasets. These models, when correctly trained, promise to generate images or visual features from the same distribution as the original data on which they are trained. In chapter 3 we consider the possibility of using a GAN to generate new instances paired with a given target for a regression problem (image quality assessment).

GANs consist two main modules: a *generator* and a *discriminator*. In the case of image generation, the generator is responsible for generating new plausible images drawn from the distribution of the training set. The discriminator, on the other hand, is responsible for classifying images as real (coming from the training-set) or

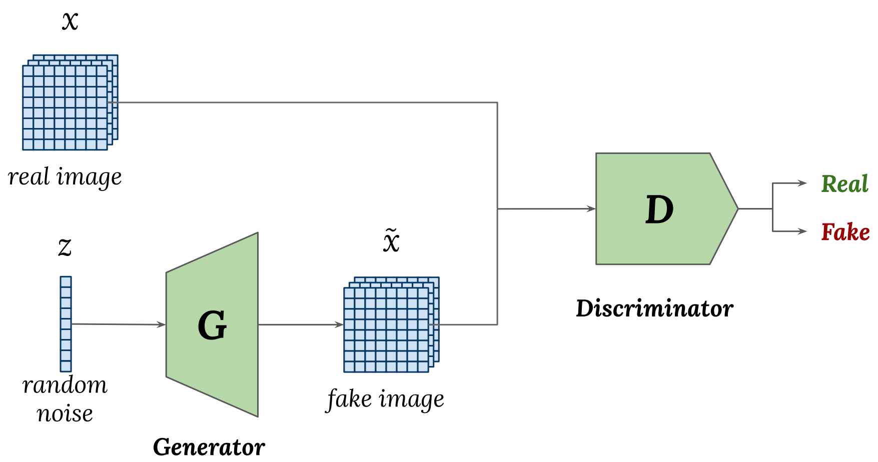


Figure 1.5: Schema of a Generative Adversarial Network (GAN) architecture.

fake (generated). The two modules are trained together in an adversarial game so that improvements of generator comes at cost of discriminator and vice versa. In figure 1.5) is reported a schema of the described architecture.

A limitation of GANs is that generated images are randomly sampled from the training image distribution, which could be problematic for applying them as a data augmentation strategy since we have little control over the *class* of generated images. Conditional Generative Adversarial Networks (cGANs), however, can be used to generate images conditioned on, for example, a class label. A cGAN concatenates the additional information coming from class label to both the input to the generator and the discriminator. Similarly, the Auxiliary Classifier Generative Adversarial Network (AC-GAN) concatenates class label information to the generator only, adding a special classifier branch to the discriminator network that is trained to classify the input image in the original training set categories in addition to the real/fake classification.

In chapter 3 we propose to train an architecture inspired by AC-GAN to synthesize new instances for the Image quality assessment problem. This allows us to expand the training examples of the dataset and boost performance of the evaluator network that predicts the quality factor of the input image.

1.3.2 Image Quality Assessment

Image quality assessment (IQA) refers to the task of estimating absolute image quality as perceived by humans (Wang et al., 2002). IQA has been widely applied to applications like image restoration Katsaggelos (2012), image super-resolution Van Ouwerkerk (2006), and image retrieval Yan et al. (2014).

Because the perceptual quality of images varies from person to person, the la-

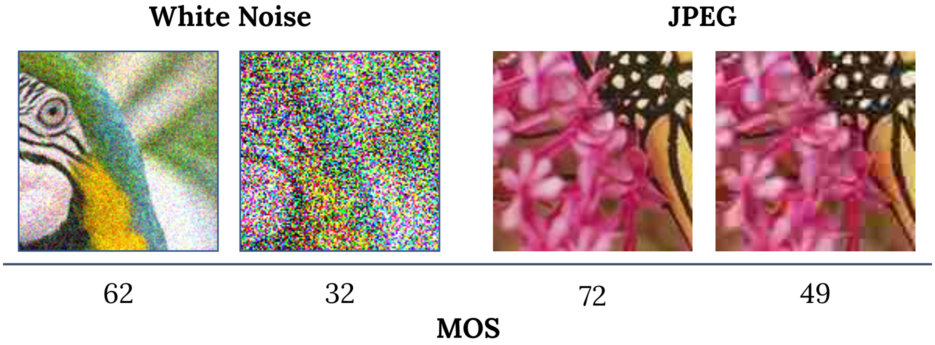


Figure 1.6: Details of distorted images from the LIVE dataset (Sheikh, 2005) with different distortions and quality scores.

beling process of IQA datasets is very expensive. Each distorted image should be annotated by multiple human experts that express a quality score between 0 and 100. These evaluators should be experts so that scores are reliable and consistent with each other. The average of all evaluation for each image is called the Mean Opinion Score (MOS) (Sheikh et al., 2006; Ponomarenko et al., 2013) and is considered the IQA target for the associated image. This is why IQA datasets are usually very small, making it very challenging to train deep networks for this task. In figure 1.6 we show patches coming from perturbed images on which are applied two classes of distortion with different intensities, resulting in different Mean Opinion Scores.

The techniques we propose in chapter 3 are able to virtually expand IQA datasets using an architecture inspired by AC-GAN. Our approach generates new distorted images from a high-quality reference image by conditioning the generation process on the desired distortion class and the desired perceived quality expressed in MOS. This allows us to augment the training set and improve the baseline accuracy of a convolutional network that acts as an image quality evaluator.

1.4 Continual learning and catastrophic forgetting

Continual learning studies the *catastrophic forgetting* (or *catastrophic interference*) phenomenon that affects the sequential training of artificial neural networks (Ratcliff, 1990). Because of catastrophic forgetting, it is very difficult to train neural networks on sequences of tasks in a continual fashion. The only straightforward way to learn multiple tasks is to learn all of them jointly. This severely limits the possibility of networks to adapt to new tasks without forgetting the previous one unless training on both from scratch, which significantly increases the cost of adaptation to a new tasks. This limitation is even more pronounced in contexts in which intelligent agents must to continuously learn new tasks and perform predictions. Continual learning research has until now concentrated primarily on classification problems modeled with deep, feed-forward neural networks. In chapter 4 we consider continual learning for image captioning, where a recurrent neural network (LSTM) is used to produce sentences describing the input image.

1.4.1 Learning and forgetting

In recent years several techniques have been developed to prevent catastrophic forgetting and enable continual learning in feed-forward neural networks. We will briefly describe the main class of strategies that were proposed to prevent forgetting.

Multitask learning. This is a naive way of preventing forgetting that interleaves data from multiple tasks during training. Forgetting does not occur because weights of the network can be jointly optimized for performance on all tasks. This strategy does not actually solve the sequential learning problem, but can be considered an upper bound that the other techniques aspire to reach.

Rehearsal methods. If tasks are presented sequentially we cannot apply the multitask learning paradigm directly. Instead, we can memorize some examples from each task and *replay* these examples during the new task training process. Usually networks will be *fine-tuned* on the new dataset providing examples stored in the memory to mitigare forgetting of old classes.

Pseudo-rehearsal methods. Instead of explicitly memorizing a set of examples for each task, we can train a generative model to generate examples from each task. During training of subsequent tasks, the generative model can be used to generate examples from the previous ones, again mitigating forgetting. A drawback of this strategy is the added complexity since the additional generative model must be trained continually.

Regularization methods. Regularization methods avoid storing exemplars and thus reduce memory requirements. To alleviate forgetting, an extra regularization term is introduced in the loss function which discourages changes in the weights that could harm to previous tasks performances. *Knowledge distillation* methods use the network at the end of previous task as a teacher for the network that we fine-tune on the new task (Li and Hoiem, 2017). Another kind of regularization strategy is based on computing the importance of each network parameter with respect to each task so that when training a new task, changes in weights that are important for previous tasks are penalized (Kirkpatrick et al., 2017).

Parameter and path allocation methods. These approaches explicitly allocate specific parameters or paths in the network to each task. When architecture size can grow, new parameters can be added every time a new task arrives, and old parameters can be frozen. These methods usually require task information during evaluation (i.e. they are *task-aware*), in contrast to the previous strategies that are usually *task-agnostic*. Examples of these approach include (Serra et al., 2018) and (Masana et al., 2020).

In chapter 4 we adapt regularization and parameter allocation methods and propose a novel continual learning framework for image captioning. Our approach is task-aware and is able to completely prevent forgetting in an LSTM network for image captioning.

1.4.2 Image captioning and natural language generation

Automatic image captioning is the generation of textual sentences describing the semantic content of an input image. Most contemporary captioning techniques are inspired by machine translation and employ a CNN as image encoder and an RNN as text decoder. They are trained to "translate" images into sentences. In these cases LSTM cells are normally used for the recurrent part of the architecture. The LSTM is initialized with the visual features extracted by the CNN and the sentence is decoded step-by-step. The LSTM internal state is updated accordingly at each step, and the process ends when a special end-of-sequence character is generated. A wide range of variations exist for this basic approach, like passing the image features at each decoding step together with the previous word embedding, applying a spatial visual attention to outputs of convolutional layers to focus on a specific area of the input image at each deconding step, and so on. During training, the use of *teacher forcing* introduced by Williams and Zipser (1989) is almost ubiquitous: at each step i, instead of passing the embedding of the previously predicted word $\hat{w}_i$ to the LSTM, the embedding of the $i-$th word w_i of a target caption is used. Note that in this case each word w_i is represented by a one-hot vector having the size of the vocabulary. In figure 1.7 we show a figure depicting the general captioning model described, that is similar to the one we used for our experiments in chapter 4, inspired by Neural Image Captioning architecture (Vinyals et al., 2015).

In chapter 4 we propose the novel problem of continual learning for image captioning, and we propose two different algorithms to split existing image captioning datasets into tasks based on visual categories. We also adapt several continual learning techniques to our proposed framework and report on extensive quantitative and qualitative experiments which demonstrate forgetting behavior of captioning networks and how our proposed approach better mitigates forgetting.

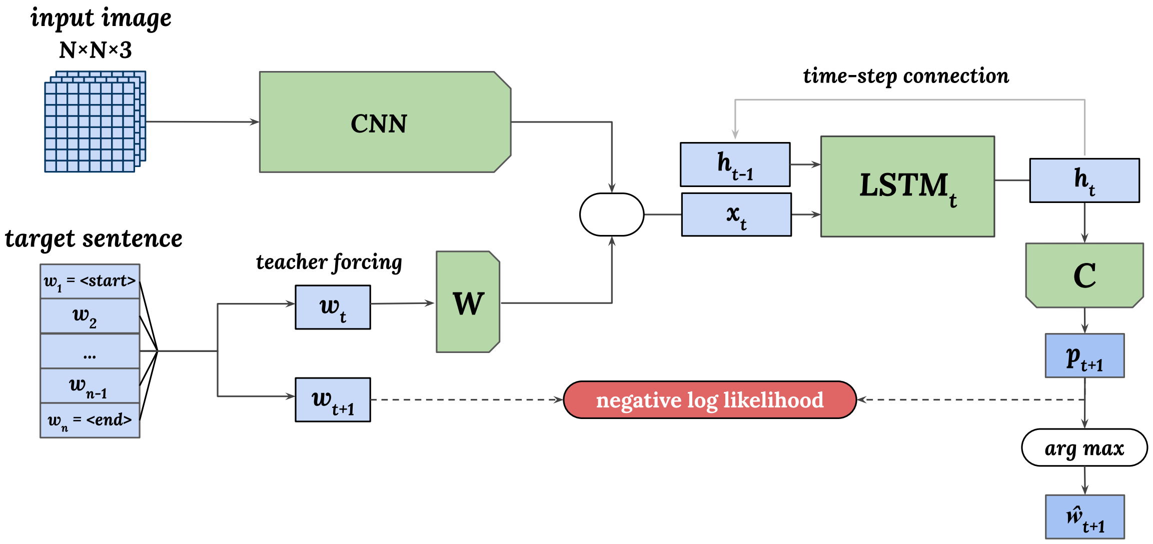


Figure 1.7: Schema of an image captioning architecture with an LSTM decoder inspired by Neural Image Captioning (Vinyals et al., 2015). W represents the word embedding matrix, C the classifier layer, h the LSTM hidden state, x the LSTM input, w the target word represented as one-hot vector and $\hat{w}$ the predicted word.

1.5 Organization of this thesis

The rest of this thesis is organized as follows:

- In chapter 2 we show how to use supervision provided by web search engines to generate weakly-labeled data. Specifically, we apply *webly-supervised learning* to artwork instance recognition. We explore different solutions for minimizing the noisy labels resulting from weak supervision, and moreover show how to exploit textual information associated with each artwork to train a zero-shot model to recognize instances of never-seen artworks.

- In chapter 3 we look at the problem of scarce data and high annotation costs for the image quality assessment. For this problem we use a different strategy: we train a generative model (based on the AC-GAN architecture) to synthetically expand the number of training examples and improve the ability of a CNN to predict the perceived quality of an input image.

- In chapter 4 we introduce the problem of continual learning for image captioning. We propose a new framework for continual image captioning and dataset splitting and show how *catastrophic interference* affects recurrent architectures applied to natural language generation for continual image captioning.

- Finally, in chapter 5 we summarize our contributions and discuss future research directions related to them.

Chapter 2

NoisyArt: Webly-supervised and Zero-shot Artwork Instance Recognition[†]

Cultural patrimony and exploitation of its artifacts is an extremely important economic driver internationally. This is especially true for culturally dense regions like Europe and Asia who rely on cultural tourism for jobs and important industry. For decades now museums have been frantically digitizing their collections in an effort to render their content more available to the general public. Initiatives like EUROPEANA (Valtysson, 2012) and the European Year of Cultural Heritage[*] have advanced the state-of-the-art in cultural heritage metadata exchange and promoted coordinated valorization of cultural history assets, but have had limited impact on diffusion and dissemination of each collection. Meanwhile the state-of-the-art in automatic recognition of objects, actions, and other visual phenomena has advanced by leaps and bounds (Russakovsky et al., 2015a). This visual recognition technology can offer the potential of linking cultural tourists to the (currently inaccessible) collections of museums.

[†] Portions of this chapter were published in:

- R. Del Chiaro, A, D. Bagdanov, and A. Del Bimbo. "Webly-supervised zero-shot learning for artwork instance recognition." *Pattern Recognition Letters*, 2019;
- R. Del Chiaro, A. D. Bagdanov, and A. Del Bimbo, "Noisyart: A dataset for webly-supervised artwork recognition." Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), 2019; and
- R. Del Chiaro, A. D. Bagdanov, and A. Del Bimbo, "NoisyArt: exploiting the noisy web for zero-shot classification and artwork instance recognition." *Data Analytics for Cultural Heritage: Current Trends and Concepts.* Eds: A. Belhi, A. Bouras, A. Al-Ali and A. Hamid Sadka. Springer, 2021.

[*] `https://europa.eu/cultural-heritage/`

Imagine the following scenario:

- A cultural tourist arrives at a destination rich in cultural heritage offerings.
- Our prototypical cultural tourist snaps a photo of an object or landmark of interest with his smartphone.
- After automatic recognition of the artwork or landmark, our tourist receives personalized, curated information about the object of interest and other cultural offerings in the area.

This type of scenario is realistic only if we have some way of easily recognizing a broad range of artworks. The challenges and barriers to this type of recognition technology have been studied in the past in the multimedia information analysis community (Cucchiara et al., 2012).

Recent breakthroughs in visual media recognition offer promise, but also present new challenges. One key challenging factor in the application of state-of-the-art classifiers is the data-hungry nature of modern visual recognition models. Even modestly sized Convolutional Neural Networks (CNNs) can have hundreds of millions of trainable parameters. As a consequence, they can require millions of *annotated* training examples to be effectively trained. The real problem then becomes the *cost* of annotation. Museum budgets are already stretched with *classical* curation requirements, adding to that the additional costs of collecting and annotating example media is not feasible.

Webly-supervised learning can offer solutions to the data annotation problem by exploiting abundantly available media on the web. This approach is appealing as it is potentially able to exploit the millions of images available on the web without requiring any additional human annotation. In our application scenario, for example, there are abundant images, videos, blog posts, and other multimedia assets freely available on the web. If the multimedia corresponding to specific instances of cultural heritage items can be retrieved and verified in some way, this multimedia can in turn be exploited as (noisy) training data. The problem then turns from one of a lack of data, to one of mitigating the effects of various types of noise in the training process that derives from its Webly nature (Temmermans et al., 2011; Sukhbaatar and Fergus, 2014).

The availability of high-quality, curated *textual* descriptions for many works of art opens up the possibility of zero-shot Learning (ZSL) in which visual categories are acquired without *any* training samples. ZSL relies on alignment of semantic and visual information learned on a training set (Xian et al., 2018). Zero-shot recognition is an extremely challenging problem, but it is particularly appealing for artwork recognition because museums normally have at least one curated description for each artwork in their collections. In the example of application scenario that we proposed, is not unrealistic to think of using zero-shot learning to improve the profile of an user for a recommendation system, enabling the possibility of exploiting artworks for which we do not have any information. In the case of artwork recognition we must solve an *instance recognition* problem using zero-shot learning, while all ZSL work to date has been on zero-shot *class recognition*.

In (Del Chiaro et al., 2019a) we presented a dataset for Webly-supervised learning specifically targeting cultural heritage artifacts and their recognition. Starting from an authoritative list of known artworks from DBpedia, we queried Google Images and Flickr in order to identify likely image candidates. The dataset consists of more than 3000 artworks with an average of 30 images per class. A test set of 200 artworks with verified images is also included for validation. We called our dataset *NoisyArt* to emphasize its webly-supervised nature and the presence of label noise in the training and validation sets. *NoisyArt* is designed to support research on multiple types of webly-supervised recognition problems. Included in the database are document embeddings of short, verified text descriptions of each artwork in order to support development of models that mix language and visual features such as zero-shot learning (Xian et al., 2017) and automatic image captioning (Vinyals et al., 2017). We believe that *NoisyArt* represents the first benchmark dataset for webly-supervised learning for cultural heritage collections. [†]

In addition to the *NoisyArt* dataset, we report on baseline experiments designed to probe the effectiveness of pretrained CNN features for webly-supervised learning of artwork instances. We also describe a number of techniques designed to mitigate various sources of noise and domain shift in the training data retrieved from the web, as well as techniques for identifying "clean" classes for which recognition is likely to be robust. Finally, we also report on experiments evaluating zero-shot artwork recognition and we show how fully webly-labeled classes can significantly improve zero-shot recognition performance. As far as we know we are the first to consider the problem of webly-supervised, zero-shot learning for *instance* recognition. These techniques provide several practical tools for building classifiers trained on automatically acquired imagery from the web.

This chapter combines and extends our work on webly-supervision for artwork instance recognition (Del Chiaro et al., 2019a) and zero-shot learning for artwork classification (Del Chiaro et al., 2019b). The chapter is organized as follows. In the next section we review recent work related to our contributions. In section 2.2 we describe the NoisyArt dataset designed specifically for research on Webly-supervised learning in museum contexts, and in section 2.3 we discuss several techniques to cope with noise and domain shift in webly supervised data. In section 2.4 we present a set of zero-shot classification techniques that have been applied to *NoisyArt*. In section 2.5 and 2.6 we present a range of experimental results establishing baselines for

[†] `https://github.com/delchiaro/NoisyArt`

state-of-the-art methods on the NoisyArt dataset, for both artwork instance recognition and zero-shot classification. We conclude with a discussion in section 2.7.

2.1 Related work

In this section we review work from the literature related to the *NoisyArt* dataset, webly-supervised learning and zero-shot learning.

2.1.1 Visual recognition for cultural heritage

Cultural heritage and recognition of artworks enjoys a long tradition in the computer vision and multimedia research communities. The Mobile Museum Guide was an early attempt to build a system to recognize instances from a collection of 17 artworks using photos from mobile phone (Temmermans et al., 2011). More recently, the Rijksmuseum Challenge dataset was published which contains more than 100,000 highly curated photos of artworks from the Rijksmuseum collection (Mensink and Van Gemert, 2014). The PeopleArt dataset, on the other hand, consists of high-quality, curated photos of paintings depicting people in various artistic styles (Westlake et al., 2016). The objectives of these datasets vary, from person detection invariant to artistic style, to artist/artwork recognition. The UNICT-VEDI dataset (Ragusa et al., 2019b) focuses on localization of visitors in a cultural site via wearable devices. A unifying characteristic of these datasets, is the high level of curation and meticulous annotation invested.

Another common application theme in multimedia analysis and computer vision applied to cultural heritage is personalized content delivery. The goal of the MNEMOSYNE project was to analyze visitor interest *in situ* and to then select content to deliver on the basis of similarity to recognized content of interest (Karaman et al., 2016). The authors of (Baraldi et al., 2015), on the other hand, concentrate on closed-collection artwork recognition and gesture recognition using a wearable sensor to enable novel interactions between visitor and museum content.

2.1.2 Webly-supervised category recognition

Early approaches to webly-supervised learning (long before it was called by that name), were the decontamination technique of (Barandela and Gasca, 2000), and the noise filtering approach of (Brodley and Friedl, 1999). Both of these approaches are based on explicit identification and removal of mislabeled training samples. A more recent approach is the noise *adaptation* approach of (Sukhbaatar and Fergus, 2014). This approach looks at two specific types of label noise – labelflip and outliers – and modifies a deep network architecture to absorb and adapt to them. A very recent approach to webly-supervised training of CNNs is the representation adaptation approach of (Chen and Gupta, 2015). The authors, in this work, at first fit a CNN to “easy” images identified by Google, and then adapt this representation to ”harder” images by identifying sub- and similar-category relationships in the noisy data.

The majority of work on webly-supervised learning has concentrated on category learning. However, the *NoisyArt* is an instance-based, webly-supervised learning problem. As we will describe in section 2.2, instance-base learning presents different sources of label noise than category-based.

2.1.3 Landmark recognition

The problem of landmark recognition is similar to our focus of artwork classification, since they are both *instance* recognition problems rather than *category* recognition problems. It is also one of the first problems to which webly-supervised learning was widely applied. The authors of (Raguram et al., 2011) use webly-supervised learning to acquire visual models of landmarks by identifying *iconic views* of each landmark in question. Another early work merged image and contextual text features to build recognition models for large-scale landmark collection (Li et al., 2009). In (Ragusa et al., 2019a) the authors extend the UNICT-VEDI dataset with annotations of points of interests using an object detector.

Artwork recognition differs from landmark recognition, however, in the diversity of viewpoints recoverable from web search alone. As we will show in section 2.2, the NoisyArt dataset suffers from several types of label bias and label noise which are particular to the artwork recognition context.

2.1.4 Zero-shot learning

Techniques for zero-shot learning (ZSL) attempt to learn to classify never-before seen classes for which semantic descriptions (but no images) are available. Recent advances in ZSL use techniques that directly learn mappings from a visual feature space to a semantic space. In some cases a linear mapping is used to learn a compatibility between visual and semantic features (Akata et al., 2015a; Frome et al., 2013; Akata et al., 2015b), in other cases a non-linear mapping is used (Socher et al., 2013), and in others a metric learning approach is used instead of compatibility (Bucher et al., 2016; Hussein et al., 2017).

2.2 The NoisyArt dataset

NoisyArt (Del Chiaro et al., 2019a) is a collection of artwork images collected using articulated queries to metadata repositories and image search engines on the web. The goal of *NoisyArt* is to support research on webly-supervised artwork recognition for cultural heritage applications. Webly-supervision is an important feature, since in the cultural applications data can be acutely scarce. Thus, the ability to exploit abundantly available imagery to acquire visual recognition models would be a tremendous advantage.

We feel that *NoisyArt* can be well-suited for experimentation on a wide variety of recognition problems. The dataset is particularly well-suited to webly-supervised instance recognition as a weakly-supervised extension of fully-supervised learning. To support this, we provide a subset of 200 classes with manually verified test images (i.e. with *no label noise*).

In the next section we describe the data sources used for collecting images and metadata. Then in section 2.2.2 we describe the data collection process and detail the statistics of the *NoisyArt* dataset.

Here we report the textual descriptions obtained from DBPedia for the artworks shown in figure 2.1. These are used to create document embeddings:

- **Self-Portrait (Raffaello):** *"The Self-portrait is commonly dated between 1504 and 1506. It measures 47.5 cm by 33 cm. The portrait was noted in an inventory of the private collection of Duke Leopoldo de' Medici, completed in 1675, and later listed in the 1890 Uffizi inventory."*

- **Alien (David Breuer-Weil):** *"Alien is a 2012 sculpture by the British artist David Breuer-Weil. It depicts a giant humanoid figure five times as large as a person, embedded head-first in grass. The sculpture was first installed in Grosvenor Gardens in the City of Westminster in April 2013, as part of the City of Sculpture initiative. In September 2015 it was moved to the National Trust property of Mottisfont in Hampshire."*

- **St. Jerome in His Study (Antonello da Messina):** *"St. Jerome in His Study is a painting by the Italian Renaissance master Antonello da Messina, thought to have been completed around 1460–1475. It is in the collection of the National Gallery, London.The picture was painted by Antonello during his Venetian sojourn, and was the property of Antonio Pasqualino."*

- **Anxiety (Munch):** *"Anxiety (Norwegian: Angst) is an oil-on-canvas painting created by the expressionist artist Edvard Munch in 1894. It's currently housed in Munch Museum in Oslo, Norway. Many art critics feel that Anxiety is closely related to Munch's more famous piece, The Scream.[who?] The faces show despair and the dark*

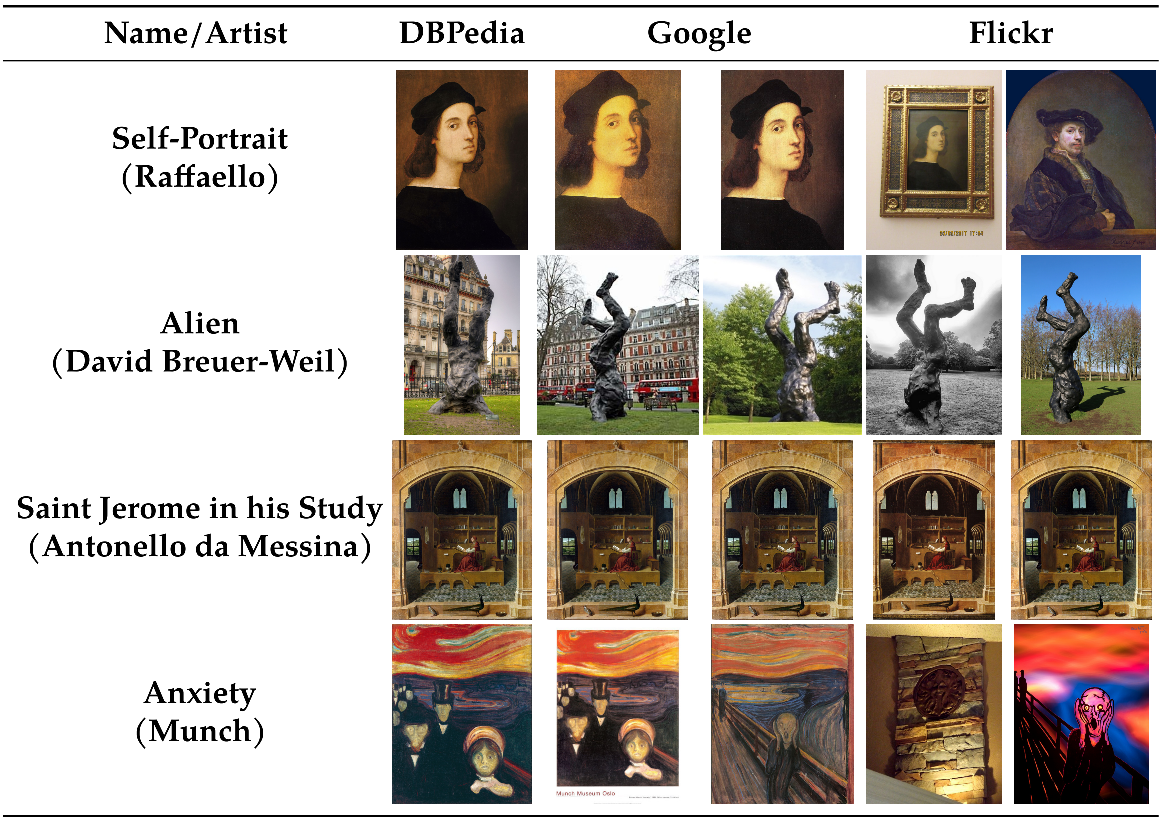

Name/Artist
DBPedia
Google
Flickr
Self-Portrait (Raffaello)
Alien (David Breuer-Weil)
Saint Jerome in his Study (Antonello da Messina)
Anxiety (Munch)

Figure 2.1: Sample classes and training images from the *NoisyArt* dataset. For each artwork/artist pair we show the seed image obtained from DBpedia, the first two Google Image search results, and the first two Flickr search results.

> *colors show a depressed state. Many critics also believe it's meant to show heartbreak and sorrow, which are common emotions all people feel."*

2.2.1 Data sources

To collect the *NoisyArt* dataset we exploited a range of publicly available data sources on the web.

Structured knowledge bases. As a starting point, we used public knowledge bases like DBpedia (Bizer et al., 2009; Mendes et al., 2011) and Europeana (Valtysson, 2012) to query, select, and filter the entities to be used as basis for *NoisyArt*. The result is a set of 3,120 artworks with Wikipedia entries and ancillary information for each one.

DBpedia. DBpedia is the same source from which we retrieved metadata. For some artworks it also contains one or more images. We call this kind of images a *seed image* because it is unequivocally associated with the metadata of the artwork. Note, however, that though the association is reliable, some times the seed image is an image of the *artist* and not of the *artwork*. We also retrieved descriptions and metadata for each artwork from this source of information. With these we created textual documents associated to each artwork in the dataset, and we produced compact vector space embedding for each artwork using doc2vec (Le and Mikolov, 2014). Both the additional information and the document embedding vectors are included in the dataset to support zero-shot learning and other multi-modal approaches to learning over weakly supervised data.

Google Images. We queried Google Images using the title of each artwork and the artist name. For each query we downloaded the first 20 retrieved images. These images tend to be very clean, in particular for paintings, most of which do not have a background and tend to be very similar to scans or posters. For this reason the variability of examples can be poor: we can retrieve images that are almost identical, maybe with just different resolutions or with some differences in color calibration. Another issue with Google Image search results is the label flip phenomenon: searching for minor artworks by a famous artist can result in retrieving images of other artworks from the same artist. Outliers are also present in a small part for less famous artworks by less famous artists.

Flickr. Finally, we used the Flickr API to retrieve a small set of images more similar to real-world pictures taken by users. Due to its nature, the images retrieved from Flickr tend to be more noisy: the only supervision is by the end-users, and a lot of images (specially for famous and iconic artworks) do not contain the expected subject. For least famous artworks, the number of retrieved images is almost zero and can be full of outliers. For these reasons we only retrieve the first 12 images from each Flickr query in order to filter some of the outlier noise.

Discussion. In the end, Flickr images are the most informative due to variety and similarity to real-world pictures. However, a lot of them are incorrect (outliers). DBpedia seed images are the most reliable but are at most one per artwork. Google images are usually more consistent with the searched concept when compared to the Flicker ones, but normally present low variability.

2.2.2 Data collection

From these sources we managed to collect 89,395 images for the 3120 classes, that became 89,095 after we pruned unreadable images and some error banners received from websites. Before filtering, each class contained a minimum of 20 images (from google) and a maximum of 33 when we could retrieve a full set of 12 images from Flickr and the DBpedia seed.

We could have used the seed images as a single-shot test set (pruning all the classes without the seed) but the importance of these images in the training phase joined to the inconsistency of seed in some classes led us to create a supervised

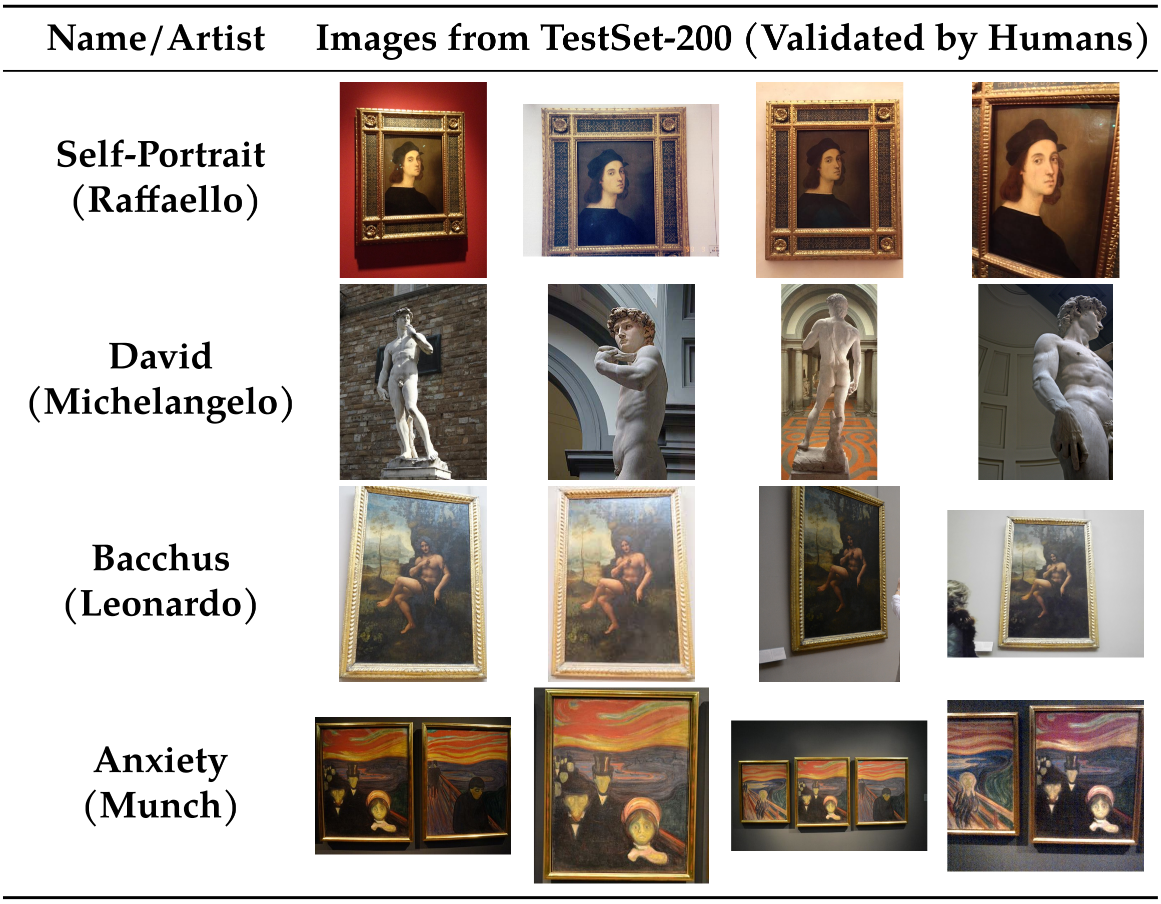

Name/Artist
Images from TestSet-200 (Validated by Humans)
Self-Portrait (Raffaello)
David (Michelangelo)
Bacchus (Leonardo)
Anxiety (Munch)

Figure 2.2: Sample verified test images from the *NoisyArt* test set. For a random sample of 200 classes we collected an additional set of images that we manually verified. Note the significant domain shift on these images with respect to those in figure 2.1

test set using a small subset of the original classes: 200 classes containing more than 1,300 images taken from the web or from our personal photos. We have been careful not to use images from the training set. This test set is not balanced: for some classes we have few images, and some others have up to 12. Figure 2.2 illustrates some sample classes and images from our verified test set. Note the strong domain shift in these images, in particular for paintings, with respect to those in the training set shown in figure 2.1.

Finally, each artwork has a description and metadata retrieved from DBpedia, from which a single textual document was created for each class. These short descriptions were then embedded using doc2vec (Le and Mikolov, 2014) in order to provide a compact, vector space embedding for each artwork description. These embeddings are included to support research on zero-shot learning and other multimodal approaches to learning over weakly supervised data.

Table 2.1: Characteristics of the *NoisyArt* dataset for artwork recognition.

Split Type	**split name**	**classes**	**webly images**		**verified images**
			training	**validation**	**test**
Classification	fully-webly	2,920	65,759	17,368	0
	verified-webly	200	4,715	1,253	1,379
	totals:	3,120	70,474	18,621	1,379
Zero Shot	unseen	50	0		355
	3-fold-seen	150	4,459		1,024
	webly-seen	2,920	83,127		0
	seen totals:	3,070	87,586		1,024

In the end, *NoisyArt* is a multi-modal, weakly-supervised dataset of artworks with 3,120 classes and more than 90,000 images, 1,300 of which are human validated. Table 2.1 details the breakdown of the splits defined in *NoisyArt* and summarizes the provided data.

2.2.3 Discussion

In figure 2.1 we give a variety of examples from the *NoisyArt* dataset. For each artwork we show: the seed image from DBpedia, the first two Google Image search results, and the first two results from Flickr. These examples show typical scenarios of this artwork instance recognition problem:

- **Best case**. The second row of figure 2.1 contains pictures of a statue. For these kinds of objects it is usually much easier to retrieve images with a good level of diversity, both from Google and Flickr. This is due to the 360° access and thus the relative variety of viewpoints from which such artworks are photographed.

- **Lack of diversity**. The first row of figure 2.1 is an example of an artwork for which Google retrieves images with extremely low variety, although in this case Flickr returns images with some diversity, but also outliers. In the third row we can observe an example for which both Google and Flickr failed to have diversity.

- **Labelflip**. In the fourth row of figure 2.1 we see a pathology particular to our instance recognition problem: we are looking for images of a *not-so-famous artwork* (Anxiety) by a *famous* artist (Munch) who also made much more iconic artworks (like The Scream). In these cases the risk of labelflip is high, and

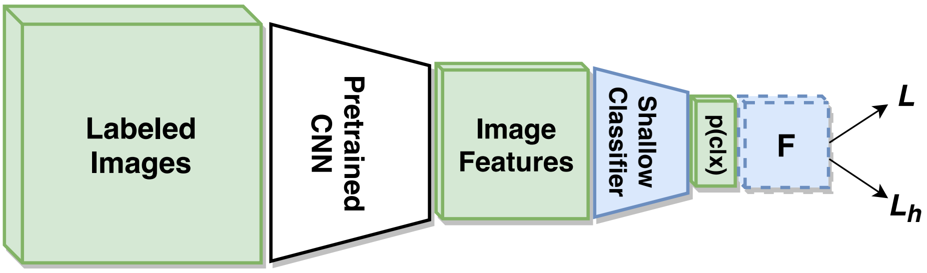


Figure 2.3: Classifier models used for webly-supervised experiments on *NoisyArt*. Green blocks represent data flowing through the network, blue ones components with trainable parameters. A CNN (pretrained on ImageNet) is used to extract features from training images, and then a shallow network with a single hidden layer and an output layer is trained to predict class probabilities. The F matrix (see section 2.3.2) is used to model and absorb labelflip noise in the training set, and the loss function is either the cross entropy loss L or the weighted cross entropy loss L_h described in section 2.3.3.

in fact we retrieved from both Google and Flickr also images of *The Scream* (together with some correct and some outlier images).

These types of label noise in the *NoisyArt* dataset render it difficult to acquire robust visual models using webly supervision. In the next section we discuss techniques to mitigate or identify noise during training.

2.3 Webly-supervised artwork recognition

In this section we describe several techniques we used to implement artwork recognition on *NoisyArt* dataset, with a focus on techniques for mitigating and/or identifying label noise during training. We also report on a simple technique designed to reduce the effect of domain shift intrinsic in the data, which led to significant improvements in artwork recognition performances on *NoisyArt* test set. First we describe the baseline classifier model used in all experiments, then we introduce three different techniques used to mitigate label noise, and finally we describe a technique that focuses on mitigating domain shift that manifests most noticeably in painting images.

2.3.1 Baseline classifier model

For all our experiments we use a shallow classifier based on image features extracted from CNNs pretrained on ImageNet. Figure 2.3 shows the architecture of our networks. Given an input image $\mathbf{x}$, we extract a feature vector using the pretrained CNN and then we pass it through a shallow classifier, consisting of a single hidden layer (with the same size as the extracted features) and an output layer that estimates class probabilities $p(c|\mathbf{x})$ for each of the 200 test classes.

The shallow classifier is then optionally followed by a multiplication with 200 × 200 *labelflip* matrix F (see section 2.3.2). For the baseline experiments F is set to the identity matrix. Finally, the loss function used to train the shallow network weights is the cross entropy loss:

$$L(\mathbf{x}, y; \theta) = -\sum_{c} \mathbb{1}_y(c) p(c \mid \mathbf{x}),$$

where $\mathbb{1}_y(c)$ is the indicator function:

$$\mathbb{1}_y(c) = \begin{cases} 1 & \text{if } c = y \\ 0 & \text{otherwise.} \end{cases}$$

2.3.2 Labelflip noise

Labelflip noise refers to images in the training set which are mislabeled as belonging to the *incorrect* class. This problem can be acute in instance recognition, for example when artists have works which are significantly more famous than their others and these famous works are often returned on queries. We experimented with the technique for labelflip absorption proposed in (Sukhbaatar and Fergus, 2014).

The main idea of labelflip absorption is to introduce a new fully connected layer without bias after the final softmax output (see the component F in figure 2.3). The weights of this layer, which we call F, are an $N \times N$ stochastic matrix, where N is the number of classes. Each row of F models the likelihood of confusing one class for any of the other classes. This matrix is initialized to the identity matrix and, at the start of training, the weights are locked (not trainable). After a number of training epochs (500 in our experiments), the weights are unlocked, allowing F to model class confusion probabilities and spread out the probability mass from each class to common confusions for that class, thanks also to a trace regularization. At each training iteration the rows of F are re-projected onto the N-simplex to keep F stochastic. The result is that labelflip noise is absorbed into the F matrix, leaving the network free to learn on “clean” labels.

2.3.3 Entropy scaling for outlier mitigation

The labelflip matrix described in the previous section attempts to compensate for class-level confusions during training. In this section we describe an alternate technique that performs soft outlier detection in order to weight training samples during training. Our hypothesis is that the class-normalized entropy of a training sample is an indicator of how confident the model is about a particular input sample. The normalized entropy of a training sample $\mathbf{x}_i$ is defined as:

$$\hat{H}(\mathbf{x}_i) = -\frac{1}{C}\sum_c p(c \mid \mathbf{x}_i) \ln p(c \mid \mathbf{x}_i),$$

where C is a normalizing constant equal to the maximum entropy attainable for the given number of classes. When $\hat{H}(\mathbf{x}_i)$ is zero, the classifier is absolutely certain about $\mathbf{x}_i$; when it is one, the classifier has maximal uncertainty. The entropy weighted loss is defined as:

$$L_h(\mathbf{x}, y; \theta) = -\sigma(\hat{H}(\mathbf{x})) \sum_c \mathbb{1}_y(c) p(c \mid \mathbf{x}),$$

where the normalized entropy is passed through a modified sigmoid σ function of the types illustrated in figure 2.4. This function is defined as:

$$\sigma(x; m, b) = \frac{1}{1 + e^{m(x-b)}}$$

so that the loss for training sample $\mathbf{x}$ is weighted inversely proportionally to the normalized entropy $\hat{H}(\mathbf{x})$.

2.3.4 Gradual bootstrapping

The entropy scaling technique described in the previous section applies soft weights to the loss contributed by specific training samples. These weights are based on an estimate of the class uncertainty. However, CNNs are known to produce highly-confident predictions even on outliers. Instead, here we propose a method for gradually bootstrapping during training by starting from highly reliable training examples, and sequentially introducing less reliable training data.

For *NoisyArt* we have the *seed images* acquired from DBpedia metadata records that can be used as a reliable image for each class. If there is no seed image for a specific class, we use the first result returned by Google Image Search as the initial bootstrap image for that class. Training is performed for 80 epochs using only seed images, then the rest of the examples are added and training proceeds using entropy scaling as described in section 2.3.3. We expect that after acquiring a reliable model on seed images, entropy scaling will be more robust as the classifiers should be more conservative as they have been initially trained on a very reduced training set.

2.3.5 Domain shift mitigation and L_2 normalization

Because of domain shift and differences observed between test and validation performance, we investigated the use of an L_2 normalization layer (Ranjan et al., 2017)

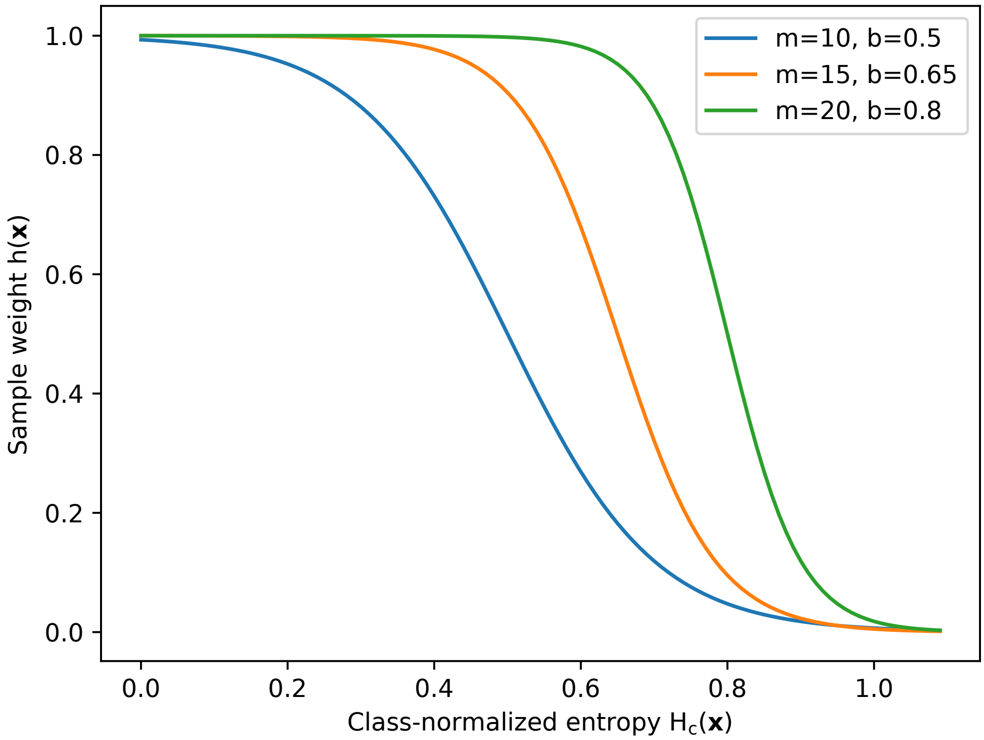


Figure 2.4: Modified sigmoid function used to calculate per-sample weights based on normalized entropy. The m parameter controls the steepness of the transition from 1.0 to 0.0, and the b parameter the point at which is begins its transition.

inserted before the output layer in our shallow recognition network. The authors of (Ranjan et al., 2017) proposed this strategy for face recognition problems, observing that normalization helps create similar representation for images with different visual characteristics (e.g. picture quality) because the magnitude of features is ignored by the final classification layer.

When using this technique, we simply modify our baseline model and replace the ReLU activation after the hidden layer with an L_2 normalization layer. This layer simply normalizes the features so that they have unit norm:

$$f(\mathrm{x}) = \frac{\alpha \mathrm{x}}{||\mathrm{x}||_2}, \tag{2.1}$$

where x is the output of the last hidden layer and α is a parameter used to rescale the radius of the unit hypersphere. Using $\alpha = 1$ we project each feature x in the hypersphere with unit radius, increasing α we are increasing the radius, and with that the surface area of the hypersphere.

Our intuition is that this should help mitigate domain shift and in general reduce the distance of the features given by images of different quality. Moreover, it should force features from the same class to be closer, while keeping features from different

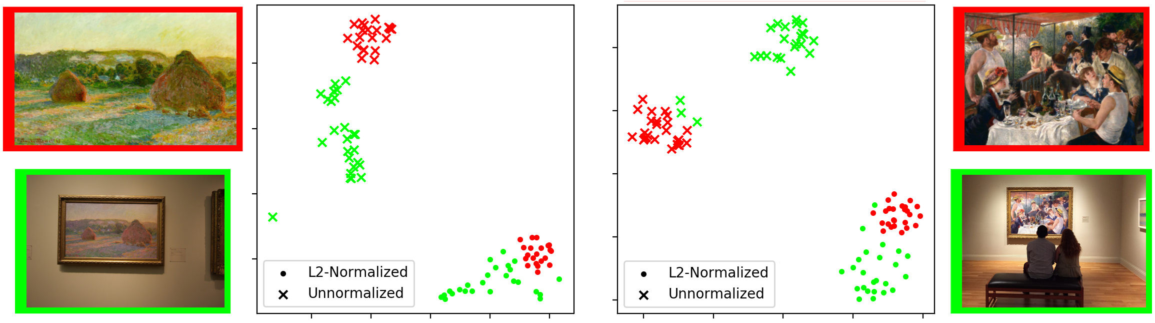


Figure 2.5: t-SNE plots of features from examples of single artworks extracted using networks trained on *NoisyArt*. Dots come from a L_2 normalized network and crosses from the baseline network. Green indicates verified test images and red webly-labeled training images. Note how, when L_2 normalization is used, the training and test image clusters approach one another.

classes far from each other in the normalized space. In figure 2.5 we illustrate the difference in features extracted from images of paintings from the test set (i.e. real world-photos) compared to those from the training set (webly-supervised, lacking in variety, and similar to scans). In the first case features cluster together and it is easy to confuse the two different kinds of images from the same class. In contrast, without L_2 normalization, scans and photographs of paintings tend to cluster in different regions of the space, rendering the classification task much harder. We found this simple technique to be much more helpful to final network performance than all the other techniques implemented with the end of reducing label noise.

2.4 Zero-shot artwork recognition

Thanks to the textual description provided for each artwork in the *NoisyArt* dataset, we also performed several zero-shot learning (ZSL) experiments. We embed text descriptions with the doc2vec (Le and Mikolov, 2014) model pretrained on Wikipedia. These 300-dimensional vectors become the semantic descriptions classes for zero-shot learning. In the following subsections we describe several baseline ZSL techniques that we implemented and tested on *NoisyArt*, as well as an extension to a known ZSL technique that we proposed in (Del Chiaro et al., 2019b).

2.4.1 Compatibility models

Compatibility models learn mappings from a visual embedding space (e.g. CNN features) to the semantic space (e.g. doc2vec embeddings). Training usually consists of pair or triple sampling and a loss function that balances distances between positive and negative image examples to their semantic class embeddings.

Linear compatibility

Linear models rely on mapping visual features into the semantic space through a linear mapping trained to maximize a compatibility function for pairs of visual/semantic features coming from the same class.

For EsZSL (Romera-Paredes and Torr, 2015) we used an open source implementation available online[‡], while for the other comparisons we adapted linear compatibility approaches to our task.

The authors of DEVISE (Frome et al., 2013) used a dedicated language model trained together with a linear embedding of visual features into semantic space. Instead, we use fixed doc2vec (Le and Mikolov, 2014) semantic features. The loss used in the original paper is the hinge loss, defined as:

$$\mathcal{L}(\mathbf{x}, \mathbf{y}) = \sum_{\mathbf{y}' \neq \mathbf{y}} max[0, m - \mathbf{x}^T \mathbf{W} \mathbf{y}' + \mathbf{x}^T \mathbf{W} \mathbf{y}], \tag{2.2}$$

where m is a strictly positive margin, $\mathbf{x}$ is a visual embedding, $\mathbf{y}$ is the corresponding semantic embedding and $\mathbf{y}'$ are semantic embedding not related with $\mathbf{x}$. Differently from Frome et al. (2013), we used a margin of 0.5 instead of 0.1.

ALE (Akata et al., 2015a) is a linear compatibility approach that introduce a decreasing γ_k function used to weight examples in a ranking loss. The compatibility of the current example with all the classes is computed to creating a ranking, the rank index k is then used to weight the contribution of that class when computing the loss. We used a modified sigmoid function defined as:

$$\gamma_k = 1 + \frac{\alpha}{1 + e^{\beta k}} - \frac{\alpha}{2}, \tag{2.3}$$

with α and β fixed to 1.7 and 0.02, respectively.

Non-linear compatibility

These models learn a non-linear mapping of image features into the semantic space. For our investigation we implemented variants of a non-linear compatibility model from the literature (Socher et al., 2013). These models use a shallow MLP network that embeds visual features in a semantic space. During training we randomly picked a negative label $y' \neq y_n$ for each visual feature x_n with label y_n, and we computed the following loss to train the embedding network using stochastic gradient descent:

$$\mathcal{L}(\mathbf{x}_n, y_n, y') = [m + F(\mathbf{x}_n, y'; W) - F(\mathbf{x}_n, y_n; W)]_+ \tag{2.4}$$

where $m > 0$ is a margin, W are the network weights and $F(\mathbf{x}, y; W)$ is the cosine distance between the embedding $\mathbf{x}$ and the corresponding semantic embedding of class label y.

[‡] `https://github.com/chichilicious/embarrassingly-simple-zero-shot-learning`

CNN	Reference	Feature	Size
ResNet-50	(He et al., 2016)	Global pool	2048
ResNet-101	(He et al., 2016)	Global pool	2048
ResNet-152	(He et al., 2016)	Global pool	2048
VGG16	(Simonyan and Zisserman, 2014)	FC7	4096
VGG19	(Simonyan and Zisserman, 2014)	FC7	4096

Table 2.2: Networks used for image feature extraction in our experiments.

To experiment with non-linear compatibility, we implemented a simplified version of CMT (Socher et al., 2013) that follows our framework. We used the same network architecture described by Socher et al. (2013), but with the use of pre-computed doc2vec text embeddings. We refer to this model as CMT* in what follows.

2.4.2 Zero-shot learning with webly-labeled data

In (Del Chiaro et al., 2019b) we proposed three extensions of ZSL techniques using non-linear compatibility for our instance recognition problem.

COS: the COS model is a modification of CMT* using three hidden layers with 2048, 1024 and 512 units instead of the single hidden layer in CMT. Moreover, ReLU activations is used instead of tanh.

COS+NLL: this model is inspired by Hussein et al. (2017). We want visual features embedded in the semantic space by the COS model to be good for classification, and to encourage this we add a new linear layer acting as a classifier connected to the output of the last layer of the COS model. Then we added an additional negative log-likelihood loss (NLL) weighted with a factor 0.1 before adding it to the original margin loss described in equation 2.4.

COS+NLL+L2: in this model we added an L_2 normalization layer as we explained in section 2.3.5 before the classifier in the COS+NLL model. This forces all visual features embedded in the doc2vec space onto a hypersphere, simplifying the work of the classifier as shown in section 2.3.5.

2.5 Experimental results: artwork instance recognition

In this section we report experimental results for a number of feature extraction and label noise compensation methods. All experiments were conducted using features extracted from CNNs pretrained on ImageNet, which are then fed as input to a shallow classifier (see figure 2.3). More specifically, we extracted features using the networks shown in table 2.2.

The shallow networks were trained with the Adam optimizer (Kingma and Ba, 2014) for 1500 epochs on the 200-class training set. We used a learning rate of 1e-4 and L_2 weight decay with a coefficient of 1e-7. For experiments using entropy scaling, we used parameters $m = 20$ and $b = 0.8$ for the modified sigmoid function. After 1500 epochs, the model corresponding to the best classification accuracy on the webly-supervised validation set was evaluated on the verified test set.

For the domain shift mitigation experiment described in section 2.3, we trained the shallow networks for only 500 epochs with the same optimizer and learning rate. In this case the L_2 normalization layer is used instead of the hidden layer activation.

2.5.1 Datasets

We used two datasets for our experiments on webly-supervised artwork recognition.

NoisyArt. Most experiments were performed on the *NoisyArt* dataset described in section 2.2. This dataset was designed specifically to experiment with webly-labeled data for both supervised instance recognition and zero-shot recognition scenarios.

CMU-Oxford Sculptures. For the domain shift mitigation experiment, in addition to *NoisyArt*, we also experimented on CMU-Oxford Sculptures (Fouhey et al., 2016). It contains about 143K images of 2,197 different sculptures. We chose this dataset because it is another artwork instance recognition problem, although in this case without label noise. We used this dataset only for supervised instance recognition experiments and we generated a different split from the original: training, validation, and test sets now contain all the classes, but different images. Our new split for CMU-Oxford Sculptures has about 74K, 33K and 37K images for the training, validation and test, respectively.

2.5.2 Webly-supervised classification

Table 2.3 gives results for all extracted features, reporting in bold the best result on each column. For each extracted feature type we report results for:

- **Baseline (BL)**: the shallow network trained with no noise mitigation.
- **LabelFlip (LF)**: the shallow network trained with labelflip absorption as described in section 2.3.2.
- **Entropy Scaling (ES)**: the shallow network trained with entropy scaling as described in section 2.3.3.

Table 2.3: Recognition accuracy (acc) and mean average precision (mAP) on *NoisyArt* and *CMU-Oxford-Sculptures*. BL refers to the baseline network, LF to labelflip, ES to entropy scaling, BS to gradual bootstraping and L_2 to L_2 normalization network. The reported approaches are described in section 2.3

	NoisyArt				CMU-Oxford-Sculptures			
	test		validation		test		validation	
	acc	mAP	acc	mAP	acc	mAP	acc	mAP
ResNet-50 BL	64.80	51.69	76.14	63.08	83.32	66.78	83.39	66.91
ResNet-50 LF	67.90	55.83	76.54	63.54				
ResNet-50 ES	68.71	57.42	76.46	63.74				
ResNet-50 BS	68.27	57.44	75.98	62.83				
ResNet-50 L_2	74.89	**62.86**	77.14	63.71	86.02	71.78	86.01	71.05
ResNet-101 BL	64.96	52.21	75.37	62.10	83.76	66.87	83.86	67.90
ResNet1-101 LF	67.08	55.58	77.09	64.17				
ResNet-101 ES	67.16	56.60	76.38	63.56				
ResNet-101 BS	68.27	57.41	76.78	63.46				
ResNet-101 L_2	74.53	62.55	77.05	63.56	86.34	71.81	86.66	72.80
ResNet-152 BL	64.28	52.05	75.31	62.37	84.12	68.11	84.25	68.53
ResNet-152 LF	66.72	54.66	76.46	63.02				
ResNet-152 ES	67.16	56.06	76.70	64.16				
ResNet-152 BS	67.38	55.81	76.22	62.90				
ResNet-152 L_2	**75.04**	62.75	**79.03**	**66.55**	**86.85**	**73.66**	**86.90**	**73.36**
VGG16 BL	64.37	50.71	74.25	60.10	78.19	58.31	78.25	58.23
VGG16 LF	64.65	50.62	73.74	59.23				
VGG16 ES	64.80	51.17	75.42	61.65				
VGG16 BS	66.27	52.52	74.38	60.07				
VGG16 L_2	68.47	55.32	74.94	61.34	82.59	66.15	82.47	65.93
VGG19 BL	62.07	48.14	73.73	59.62	78.51	59.72	78.53	58.98
VGG19 LF	61.33	46.53	73.07	57.84				
VGG19 ES	61.92	48.43	72.87	58.34				
VGG19 BS	63.99	51.14	72.63	58.21				
VGG19 L_2	66.25	53.05	74.49	60.42	82.29	64.91	82.50	65.48

- **BootStrapping (BS)**: the shallow network trained with gradual bootstrapping as described in section 2.3.4.

- **L_2 Normalization (L_2)**: the shallow network trained with L_2 feature normalization as described in section 2.3.5.

Looking at the *NoisyArt* results in table 2.3 we can draw a few conclusions. First of all, despite the high degree of noise in the training labels, even the baseline classifiers perform surprisingly well on the webly-supervised learning problem. All of the ResNet models achieve nearly 65% classification accuracy on the verified test set. The shallow classifier seems to be able to construct models robust to noise in the majority of classes.

All three of the noise mitigation techniques improve over the baseline shallow classifier. The gradual bootstrapping technique described in section 2.3.4 generally yields the most consistent and significant improvement. But in the end the L_2 normalization is the technique that really made the difference in recognition performances for NoisyArt, giving a boost of about 10% in the *NoisyArt* test set performance when compared to the baseline, passing from 64.80% to 74.89% accuracy for ResNet-50. performance over the baseline for all the networks.

Results on *NoisyArt* validation set are an unreliable fine-grained predictor of classifier performance on validated test data. Though the performance on the validation set between ResNet and VGG models is a reliable indicator, performance on the different ResNet models is generally too close to call.

The performance gap between *NoisyArt* test and validation when using the baseline is evidence of domain shift, while for CMU-Oxford Sculptures – which has no domain shift – they achieve similar performance. Moreover, for *NoisyArt* using L_2 normalization yields huge improvement in the clean test set, while in CMU-Oxford Sculptures the improvement is much lower (but still significant). Finally, note how the performance gap for the best performing model on *NoisyArt* and CMU-Oxford Sculptures is high (about 10%). We think this gap is due to the small number of examples per class in *NoisyArt* compared to CMU-Oxford Sculptures along with the intrinsic noise in webly-labeled images.

2.5.3 Identifying problem classes

In figure 2.6 we show the improvement that can be gained by filtering classes with high average entropy. The figure plots classifier accuracy for all models with bootstrapping as a function of progressively filtered test sets (i.e. removing unreliable classes). Observe that the average class entropy is a reasonable measure of classifier reliability. After filtering only about 20% of the problem classes we can obtain an overall accuracy greater than 80% on the remaining ones for the ResNet models, reaching performance values similar to the ones obtained on CMU-Oxford-Sculpture dataset.

2.6 Experimental results: zero-shot recognition

In this section we report on a range of experiments we performed in (Del Chiaro et al., 2019b) to evaluate the effectiveness of webly-labeled data for both supervised

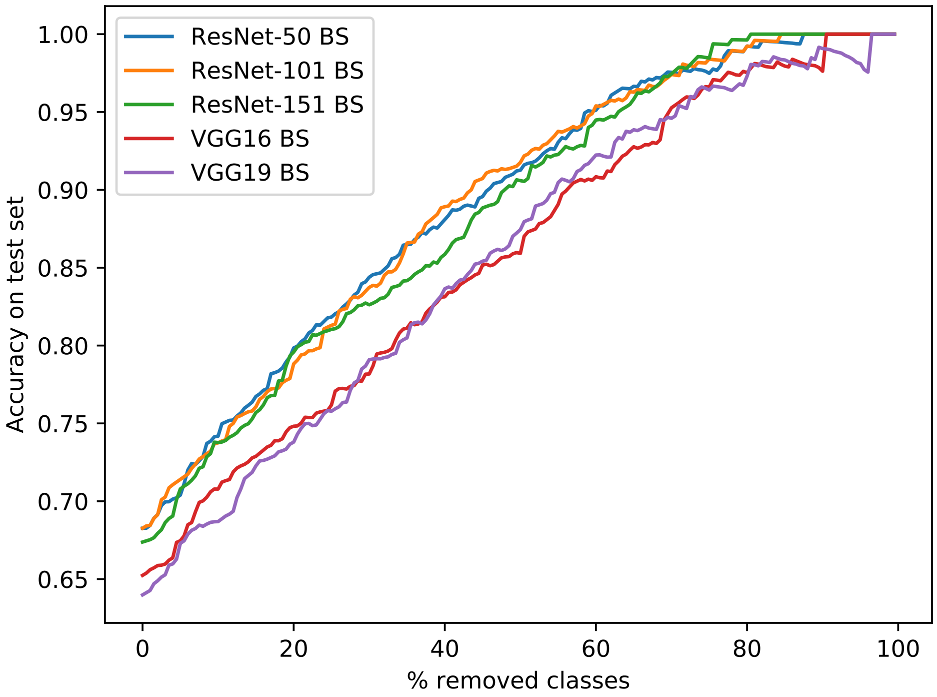


Figure 2.6: Filtering problem classes. We progressively remove classes with high entropy from the test set. Accuracy is plotted as a function of the number of remaining classes.

and zero-shot recognition of artwork instances.

2.6.1 Zero-shot recognition with webly-labeled data

We trained the models from section 2.4 on *NoisyArt* using three-fold cross validation: we split the 200 verified classes into 150 for training/validation and 50 for zero-shot test classes. Test and validation sets only contain human-verified images, while training set can exploit webly-labeled images.

For testing, we again trained each network from scratch on the combined training and validation sets (150 classes) using the early stopping epoch computed during cross validation. Each experiment is repeated four times with different training data:

- **V**: verified images from the training classes;
- **W**: webly-labeled images from the training classes;
- **VW**: both verified and webly-labeled images; and
- **VWC**: all the images of **VW** together with all the images from the 2,920 webly-labeled classes.

Table 2.4: Zero-shot recognition accuracy for NoisyArt.

	Accuracy				Mean Average Precision			
Images	**V**	**W**	**VW**	**VWC**	**V**	**W**	**VW**	**VWC**
ResNet50								
SJE (Akata et al., 2015b)	10.70	15.49	7.04	13.52	17.55	14.13	16.06	14.72
EsZSL (Romera-Paredes and Torr, 2015)	12.11	15.21	14.37	25.63	20.48	18.68	22.29	29.89
ALE (Akata et al., 2015a)	14.08	14.08	15.49	22.54	21.43	16.90	18.28	34.99
DEVISE (Frome et al., 2013)	16.62	14.93	16.90	24.79	22.63	19.18	20.95	31.90
CMT* (Socher et al., 2013)	19.44	13.24	15.21	21.13	21.53	19.09	24.02	43.72
COS	20.56	18.03	16.62	26.48	26.02	17.84	26.05	43.94
COS+NLL	14.65	15.77	16.06	26.20	27.10	23.53	25.36	44.70
COS+NLL+L2	18.31	8.45	18.03	34.93	24.81	21.26	25.36	45.53
ResNet152								
SJE (Akata et al., 2015b)	10.70	15.49	7.04	13.52	17.55	14.13	16.06	14.72
EsZSL (Romera-Paredes and Torr, 2015)	20.28	14.08	17.75	26.48	24.19	19.52	23.94	29.36
ALE (Akata et al., 2015a)	17.18	13.52	14.37	21.69	24.54	19.65	20.79	33.93
DEVISE (Frome et al., 2013)	14.37	15.77	17.18	22.54	23.02	19.32	22.44	32.49
CMT* (Socher et al., 2013)	21.13	12.39	15.77	22.82	25.23	19.63	20.41	37.02
COS	17.75	11.55	16.34	27.04	26.51	17.9	23.04	40.18
COS+NLL	20.56	14.93	18.59	27.32	24.32	25.48	25.67	41.82
COS+NLL+L2	18.31	12.96	17.75	29.58	27.80	20.61	29.14	48.17

The results for zero-shot recognition are shown in table 2.4. Note how adding webly-labeled images to the fully-verified classes does not always improve recognition performance. However, adding new classes containing only webly-labeled images (together with a single semantic vector for each class) greatly improves results, especially for non-linear techniques.

One of our goals was to understand if the additional webly-labeled images and classes containing only webly-labeled images can help zero-shot recognition performance. We trained the COS+NLL+L2 and CMT* networks several times, gradually increasing the number of webly-labeled classes in each run. For this experiment we used the test set as validation, computing the performance for the best-performing epoch. Results are shown in figure 2.7. Note the rapid increase in mAP values for both models in the first half of the runs (until about 1460 additional classes) passing from 0.24 to 0.37 for CMT* and from 0.30 to 0.44 for COS+NLL+L2. The next 1,460 additional classes increase performance, but the growth is slower.

2.7 Conclusions

In this chapter we described all the experiments we made on *NoisyArt* to exploit web data for artwork instance recognition and zero-shot learning. The results on artwork recognition show that shallow classifiers trained on features extracted with pretrained CNNs over webly-labeled images can be effective at artwork instance

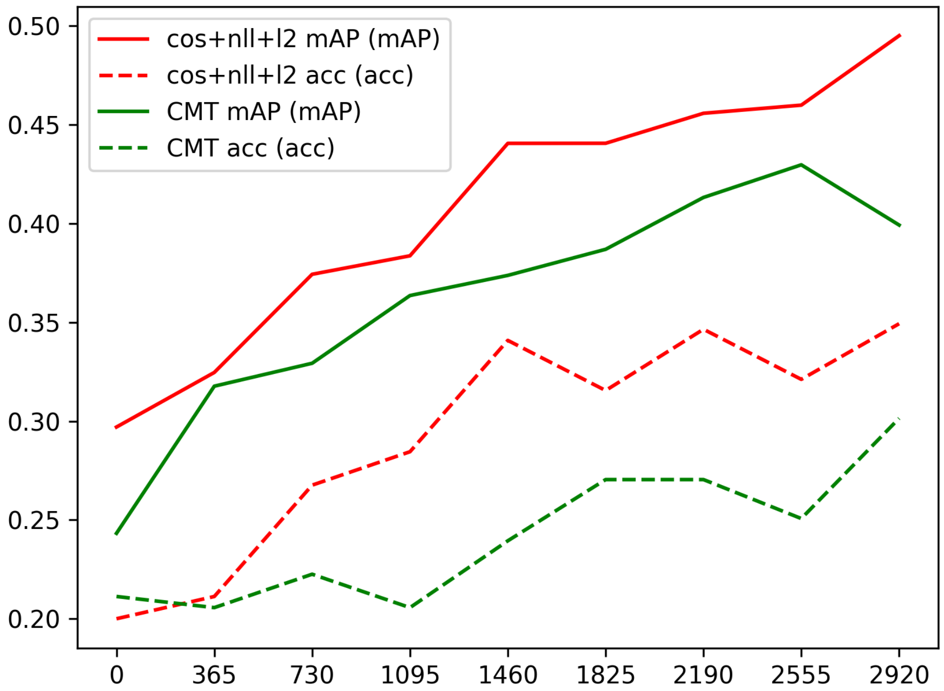


Figure 2.7: Performance of COS+NLL+L2 and CMT* with increasing numbers of fully webly-labeled classes. Performance is an upper bound since we report the best performing epoch on the test set.

recognition. Using relatively simple networks and compact image features, classifiers achieve nearly 80% classification accuracy. Key to achieving this performance is treating webly-supervised artwork recognition as an *instance* recognition problem and using L_2 normalization layer before classification. This simple technique, in the case of both *NoisyArt* and CMU-Oxford Sculptures, leads to significant improvement even over more complicated noise mitigation techniques.

Cultural heritage applications involving artwork recognition have the advantage that semantically rich, textual descriptions are abundantly available. These can be exploited with a minimal effort using webly-labeled data and a zero-shot learning approaches. Experiments show how, despite the noisy supervision, a large set of additional classes can improve zero-shot recognition for this kind of problem – especially when using L_2 normalization to compensate for domain shift introduced by the different data source biases.

Museums and cities of art seem to struggle in the creation and sharing of well organized knowledge bases containing the information required to recognize cultural heritage objects, and this limit cultural heritage users from the possibility to benefit from the latest computer vision technologies. In the current scenario, we propose an alternative viable road that exploits web search engines, social media and potentially the active interaction of users to enable artwork instance recognition, giving the cue for the development of a variety of different applications that take advantage of computer vision techniques to involve visitors and citizens in the enjoyment of the cultural heritage of our cities.

Chapter 3

GADA: Generative Adversarial Data Augmentation for Image Quality Assessment[†]

In the last few decades images are increasingly a part of everyday life and are used for many purposes. However, images are often not of the best possible quality. This can be caused by many factors, such as the device used for acquisition, the lossy compression algorithm used to store the information (e.g. JPEG), and the entire image acquisition, storage, and transmission process.

Image quality assessment (IQA) (Wang et al., 2002) refers to a range of techniques developed to automatically estimate the perceptual quality of images. IQA estimates should be highly correlated with quality assessments made by multiple human evaluators (commonly referred to as the Mean Opinion Score (MOS) (Sheikh et al., 2006; Ponomarenko et al., 2013)). IQA has been widely applied by the computer vision community for applications like image restoration (Katsaggelos, 2012), image super-resolution (Van Ouwerkerk, 2006), and image retrieval (Yan et al., 2014).

IQA techniques can be divided into three different categories based on the available information on the image to be evaluated: full-reference IQA (FR-IQA), reduced-reference IQA (RR-IQA), and no-reference IQA (NR-IQA). Although FR-IQA and RR-IQA methods have obtained impressive results, the fact that they must have knowledge of the undistorted version of the image (called the *reference image*) for quality evaluation, makes these approaches hard to use in real scenarios. On the contrary, NR-IQA only requires the knowledge of the image whose quality is to be estimated, and for this reason is more realistic (and also more challenging).

In the last few years Convolutional Neural Networks (CNNs) have obtained

[†] Portions of this chapter were published in: P. Bongini, R. Del Chiaro, A. D. Bagdanov, and A. Del Bimbo, "GADA: Generative Adversarial Data Augmentation for Image Quality Assessment." Proceedings of the *International Conference on Image Analysis and Processing* (*ICIAP*), 2019.

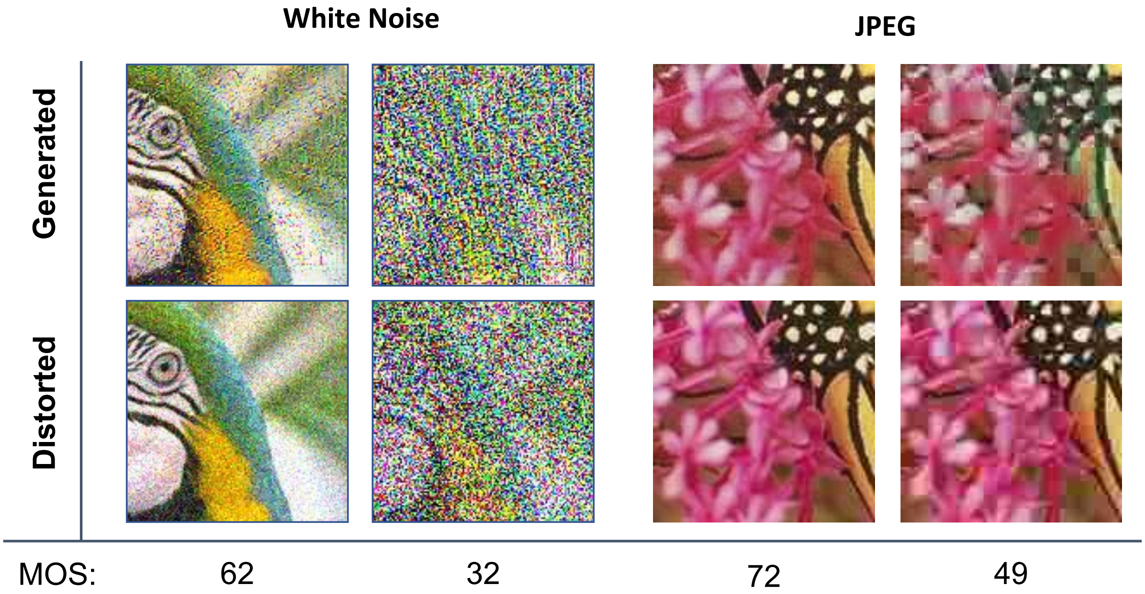


Figure 3.1: Patches extracted from images generated by the proposed method compared with the same patches from true distorted images having the same image quality and distortion type.

great results on many computer vision tasks, and their success is partially due to the possibility of creating very deep architectures with millions of parameters, thanks to the computational capabilities of modern GPUs. Massive amounts of data are needed for training such models, and this is a big problem for IQA since the annotation process is expensive and time consuming. In fact, each image must be annotated by multiple human experts, and consequently most available IQA datasets are too small to effectively train CNNs from scratch.

In this chapter, we propose an approach to address this lack of large, labeled datasets for IQA. Since obtaining annotated data to train the network is difficult, we propose a technique to generate new images with a specific image quality and distortion type. We learn how to generate distorted images using Auxiliary Classifier Generative Adversarial Networks (AC-GANs), and then use these generated images in order to improve the accuracy of a simple CNN regressor trained for IQA. In figure 3.1 we show patches of images generated with our approach alongside their corresponding patches with real distortions.

3.1 Related work

In this section we briefly review the literature related to no-reference image quality assessment (NR-IQA) and Generative Adversarial Networks (GANs).

No-Reference Image Quality Assessment. Most traditional NR-IQA can be classified into Natural Scene Statistics (NSS) methods and learning-based methods. In NSS methods, the assumption is that images of different quality vary in the statistics of responses to specific filters. Wavelets, DCT and Curvelets are commonly used to extract the features in different sub-bands. These feature distributions are parametrized, for example with the Generalized Gaussian Distribution. The aim of these methods is to estimate the distributional parameters, from which a quality assessment can be inferred. Mittal et al. (2012) propose to extract NSS features in the spatial domain to obtain significant speed-ups. In learning-based methods, local features are extracted and mapped to the MOS using, for example, Support Machine Regression or Neural Networks (Chetouani et al., 2010). Codebook Methods combines different features instead of using local features directly. Datasets without MOS can be exploited to construct the codebook (Ye and Doermann, 2012; Ye et al., 2012) by means of unsupervised learning, which is particularly important due to of the small size of existing datasets. Saliency maps can be used to model human vision system and improve precision in these methods.

Deep Learning for NR-IQA. In recent years several works have used deep learning for NR-IQA. These techniques requires large amounts of data for training and IQA datasets are especially lacking in this regard. Therefore, to address this problem different approaches have been proposed. Kang et al. (2014) use small patches of the original images to train a shallow network and thus enlarging the initial dataset. A similar approach was presented in (Kang et al., 2015) where the authors use a multi-task CNN to learn the type of distortion and the image quality at the same time. Bianco et al. (2016) used a pre-trained DCNN fine tuned with an IQA dataset to extract features, and then train a Support Vector Regression model that maps extracted features to quality scores. Liu et al. (2017) use a learning from rankings approach. They train a Siamese Network to rank images in term of image quality and subsequently the information represented in the Siamese network is transferred, trough fine-tuning, to a CNN that predicts the quality score. Another interesting work is from Lin and Wang (2018) who use a GAN to generate a hallucinated reference image corresponding to a distorted version and then give both the hallucinated reference and the distorted image as input to a regressor that predicts the image quality.

In our work we present a novel approach to address the scarcity of training data: we train an Auxiliary Classifier Generative Adversarial Network (AC-GAN) (Odena et al., 2017) to produce distorted images given a reference image together with a specific quality score and a category of distortion. In this way we can produce new labeled examples that we can use to train a regressor.

Auxiliary Classifier GANs. In the last few years GANs have been widely used in different areas of computer vision. The Auxiliary Classifier GAN (AC-GAN) (Odena et al., 2017) is a variant of the Generative Adversarial Network (GAN) (Goodfellow et al., 2014) which uses label conditioning. This kind of network produces convincing results. Our aim is to use this architecture to generate distorted images conditioned to a distortion category and image quality value. Since the main objective of the work is NR-IQA and the performance of the quality regressor is highly related to the generated image, it is crucial that the generator produce convincing distortions.

3.2 Generative adversarial data augmentation for NR-IQA

In this section we describe our approach to perform data augmentation for NR-IQA datasets. We first show the general steps that characterize our technique, and then describe the use of AC-GAN in this context.

3.2.1 Overview of proposed approach

The main idea of this work is to generate new distorted images with a specific image quality level and distortion type to partially solve the problem of the poverty of annotated data for IQA. We use an AC-GAN to generate new distorted images. Once the generator has learned to produce distorted images convincingly we use it to generate new examples to augment the training set as we train a deep convolutional regressor to estimate IQA. The pipeline of our technique is as follows:

1. **Training the AC-GAN**. Using patches of the training images we train an AC-GAN. The generator learns to generate distorted images with a given distortion class and quality level starting from reference images. The regressor, which aims is to predict the image quality, is trained with both generated and real distorted images using the adversarial GAN loss.

2. **Generative data augmentation**. Once the training of the AC-GAN converges, the generator is able to produce convincing distortions and we can stop its training. We continue training the discriminator branch, augmenting the training data via the trained generator. The regressor is trained with both real distorted images from the training set and images artificially distorted using the generator.

3. **Fine-Tuning of the regressor**. Once convergence is reached in step 2 we perform a final phase of fine-tuning: the regressor is trained with only real distorted images from the IQA training set.

3.2.2 Auxiliary classifier GANs for NR-IQA

An Auxiliary Classifier Generative Adversarial Network is a GAN variant in which it is possible to condition the output on some input information. In the AC-GAN every generated sample has a corresponding class label, $c \sim p_c$, in addition to the noise z. This information is given in input to the generator which produces fake images $X_{\text{fake}} = G(c, z)$. The discriminator not only distinguishes between real and generated examples but predicts also the class label of the examples. The sub-network that classifies the input is called the *classifier*. The objective function is characterized by two components: a log-likelihood on the correct discrimination L_S and a log-likelihood on the correct class L_C:

$$\begin{aligned} L_S &= E[\log P(S = \text{real} \mid X_{\text{real}})] + E[\log P(S = fake \mid X_{\text{fake}})] \qquad (3.1) \\ L_C &= E[\log P(C = c \mid X_{\text{real}})] + E[\log P(C = c \mid X_{\text{fake}})] \qquad (3.2) \end{aligned}$$

The discriminator is trained to maximize $L_S + L_C$ and the generator is trained to minimize $L_C - L_S$.

Our approach is slightly different from a standard AC-GAN: the latter expects only noise and class label as input, but in our case we want to generate an output image that is a *distorted* version of a reference one, so we also need to feed the reference image and force a reconstruction with an $L1$ loss. Moreover, we want to distort the reference image so that the output matches a target *image quality*, so we feed also this value as input. Because we would like to reconstruct a distorted version of the reference image given in input, we can write the additional $L1$ loss as it follows:

$$\mathcal{L}_{L1} = E[||y - G(z, x, c, q)||_1]$$

where y is the distorted ground truth image, z is a random Gaussian noise vector, x is the reference image, c is the distortion class and v is the image quality.

The goal of this work is to predict the quality score of images, so we introduce a regressor network whose aim is to predict the quality score of input images. The loss used to train this component is a mean squared error (MSE) between the predicted quality score and the ground truth:

$$L_E = E[(q - \hat{q})^2] \qquad (3.3)$$

where q and $\hat{q}$ are the ground truth and the prediction of the image quality score, respectively.

The expectations for all losses defined here are taken over minibatches of either generated or labeled training samples.

3.2.3 The GADA architecture

In Figure 3.2 we give a schematic representation of the proposed model. The components of the GADA network are as follows.

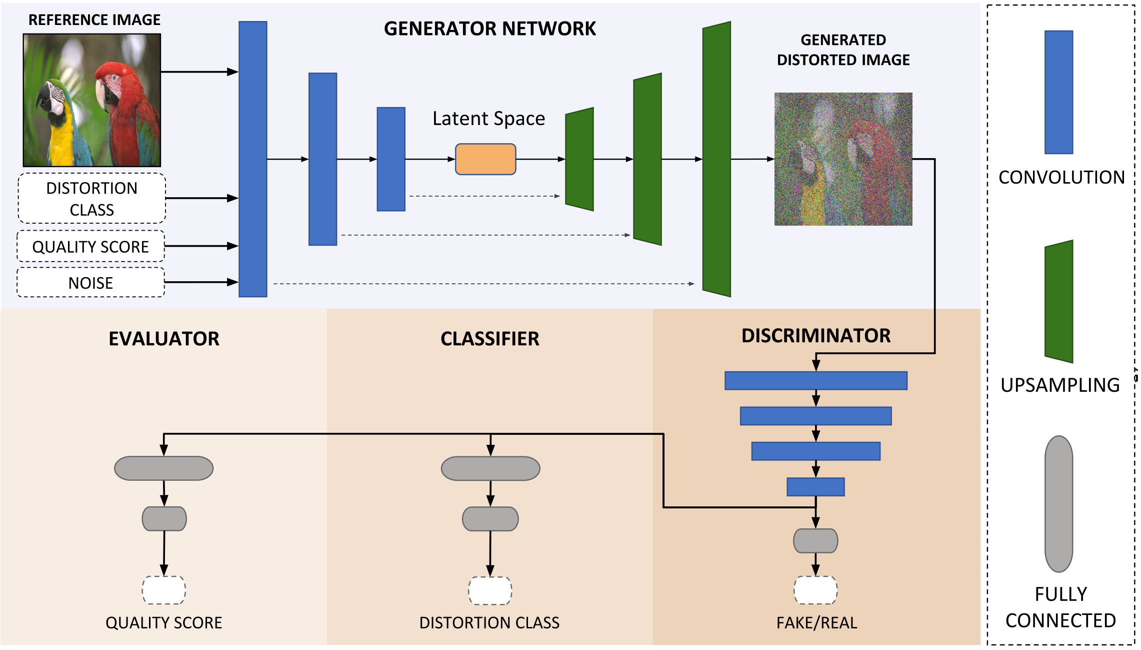


Figure 3.2: A schematic representation of the proposed network.

Generator. The Generator follows the general auto-encoder architecture. It takes as input a high quality reference image, a distortion class, and a target image quality. The input information is encoded through three convolutional layers (one with 64 feature maps and two with 128). Before up-sampling we concatenate a noise vector z to the latent representation, together with an embedding of the distortion category and image quality. We use skip connections (Ronneberger et al., 2015; Isola et al., 2017) in the generator, which allows the network to generate qualitatively better results.

Discriminator. The Discriminator takes as input a distorted image and through three convolutional layers (one with 64 feature maps, and two with 128 to mimic the encoder) followed by a 1×1 convolution extracts 1024 feature maps (that are also fed to the classifier and the regressor). A single fully-connected layer reduces these feature maps to a single value and a sigmoid activation outputs the prediction of the provenance of the input image (i.e. real or fake). This output is used to compute the loss defined in equation 3.1.

Classifier. The Classifier takes as input the feature maps described for the Discriminator. This network consists of two fully-connected layers. The first layer has 128 units and the second has a number of units equal to the number of distortion categories and is followed by a softmax activation function. The output of this module is used in the classifier loss for the AC-GAN as defined in equation 3.2.

Evaluator. The Evaluator takes as input the feature maps described for the Discriminator and should accurately estimate the image quality of the input image. This module consists of two fully-connected layers, the first with 128 and the second with a single unit. The MSE loss defined in equation 3.3 is computed using the output of this module.

3.3 Experimental results

In this section we describe experiments conducted to evaluate the performance of our approach. We first introduce the datasets used for training and testing our network, then we describe the protocols adopted for the experiments.

3.3.1 Datasets

For our experiments we used the standard LIVE (Sheikh et al., 2020) and TID2013 (Ponomarenko et al., 2013) datasets for IQA. LIVE contains 982 distorted versions of 29 reference images. Original images are distorted with five different types of distortion: JPEG compression (JPEG), JP2000 compression (JP2K), white noise (WN), gaussian blur (GB) and fastfading (FF). The ground truth quality score for each image is the Difference Mean Opinion Score (DMOS) whose value is in the range $[0, 100]$. TID2013 consist of 3000 distorted images versions of 25 reference images. The original images are distorted with 24 different types of distortions. The Mean Opinion Score of distorted images varies from 0 to 9.

3.3.2 Experimental protocols

We analyze the performance of our model using the standard IQA metrics. For each dataset we randomly split the reference images (and their corresponding distorted versions) in 80% used for training and 20% used for testing, as described from Kang et al. (2014) and Zhang et al. (2015). This process is repeated ten times. For each split we train from scratch and compute the final scores on the test set.

Training strategy. At each training epoch, we randomly crop each image in the training-set using patches of 128×128 pixels and feed it to the model. For all the three phases we train using these crops with a batch size of 64. During the first one we use Adam optimizer with a learning rate of $1e^{-4}$ for the discriminator and $5e^{-4}$ for the generator, classifier and evaluator. During the second and third phases we divide the learning rate by 10.

Testing protocol. At test time We randomly crop 30 patches from each test image as suggested from Bianco et al. (2016). We then pass all 30 crops through the discriminator network (with only the evaluator branch) to estimate IQA. The average of the predictions for the 30 crops gives the final estimated quality score.

Evaluation metrics. We use two evaluation metrics commonly used in IQA context: the Linear Correlation Coefficient (LCC) and Spearman Correlation Coefficient (SROCC). LCC is a measure of the linear correlation between the ground truth and the predicted quality scores. Given N distorted images, the ground truth of i-th image is denoted by y_i, and the predicted score from the network is $\hat{y}_i$. The LCC is computed as:

$$LCC = \frac{\sum_{i=1}^{N}(y_i - \overline{y})(\hat{y}_i - \overline{\hat{y}})}{\sqrt{\sum_{i}^{N}(y_i - \overline{y})^2}\sqrt{\sum_{i}^{N}(\hat{y}_i - \overline{\hat{y}})^2}} \tag{3.4}$$

where $\overline{y}$ and $\overline{\hat{y}}$ are the means of the ground truth and predicted quality scores, respectively.

Given N distorted images, the SROCC is:

$$SROCC = 1 - \frac{6\sum_{i=1}^{N}(v_i - p_i)^2}{N(N^2-1)}, \tag{3.5}$$

where v_i is the *rank* of the ground-truth IQA score y_i in the ground-truth scores, and p_i is the *rank* of $\hat{y}_i$ in the output scores for all N images. The SROCC measures the monotonic relationship between ground-truth and estimated IQA.

3.3.3 Generative data augmentation with AC-GAN

As described in section 3.2.1 our approach consists of three phases: a first one where we train the generator, a second phase where we perform data augmentation, and the final fine-tuning phase of the evaluator over the original training-set. As a first experiment, we calculated the performance obtained after each of the three different phases and compared with the performance of a direct method which consists of training *only* the evaluator and classifier branches of the discriminator directly on labeled training data (e.g. no adversarial data augmentation). We trained and tested the proposed method and the direct baseline on the LIVE dataset as described in section 3.3.2, but for this preliminary experiment we used crops of 64×64 pixels and a shallower regression network.

In table 3.1 we give the LCC and SROCC values computed for the baseline and after each of the three phase of our approach. We note first that each phase of our training procedure results in improved LCC and SROCC, which indicates that generative data augmentation and fine-tuning both add to performance. At the end of phase 3 the LCC and SROCC results surpass the direct approach by $\sim 2\%$, confirming the effectiveness of GADA with respect to direct training.

3.3.4 Comparison with the state-of-the-art

Here we compare GADA with state-of-the-art results from the literature.

		JP2K	**JPEG**	**WN**	**GBLUR**	**FF**	**ALL**
Baseline	**LCC**	0.950	0.964	0.973	0.938	0.933	0.943
	SROCC	0.938	0.931	0.977	0.939	0.898	0.935
Phase 1	**LCC**	0.944	0.952	0.967	0.920	0.912	0.933
	SROCC	0.933	0.930	0.980	0.926	0.889	0.930
Phase 2	**LCC**	0.958	0.958	0.974	0.939	0.924	0.942
	SROCC	0.941	0.933	0.988	0.945	0.891	0.939
Phase 3	**LCC**	0.959	0.973	0.993	0.953	0.935	0.962
	SROCC	0.955	0.941	0.990	0.953	0.912	0.955

Table 3.1: Comparison of baseline and each phase of the GADA approach in LCC and SROCC. In the first block results for the direct baseline method (directly training the evaluator with only labeled IQA data) are shown. In the second block results for our method are shown after each of the three phases: training of the AC-GAN (Phase 1), generator data augmentation (Phase 2), and evaluator fine-tuning (Phase 3).

Results on LIVE. We trained on LIVE dataset following the protocol described in 3.3.2. The results are shown in table 3.2. Each column of the table represents the partial scores for a specific distortion category of LIVE dataset. Our method seems to be very effective on this dataset despite the fact that many other approaches process larger patches (e.g. 224×224, the input size of the VGG16 network) and capture more context information. We observe from the table that our model performs very well on Gaussian noise (GN) and JPEG2000 (JP2K). We obtain worse results for Fast Fading (FF), which is probably due to the fact that FF is a local distortion and we process patches of small dimension, so for each crop the probability of picking a distorted region is not 1.

Results on TID2013 We follow the same test procedure for TID2013 and report our SROCC results in table 3.3. We see that for 11 of the 24 types of distortion we obtain the best results. For local and challenging distortions like #14, #15 and #16 the performance of our model is low, and again we hypothesize that the small size and uniform sampling of patches could be a limitation especially for extremely local distortions.

	LCC						SROCC					
	JP2K	**JPEG**	**GN**	**GB**	**FF**	**ALL**	**JP2K**	**JPEG**	**GN**	**GB**	**FF**	**ALL**
DIVINE (Moorthy and Bovik, 2011)	.922	.921	.988	.923	.888	.917	.913	.91	.984	.921	.863	.916
BLIINDS-II (Saad et al., 2012)	.935	.968	.980	.938	.896	.930	.929	.942	.969	.923	.889	.931
BRISQUE (Mittal et al., 2012)	.923	.973	.985	.951	.903	.942	.914	.965	.979	.951	.887	.940
CORNIA (Ye et al., 2012)	.951	.965	.987	.968	.917	.935	.943	.955	.976	.969	.906	.942
CNN (Kang et al., 2014)	.953	.981	.984	.953	.933	.953	.952	.977	.978	.962	.908	.956
SOM (Zhang et al., 2015)	.952	.961	.991	.974	.954	.962	.947	.952	.984	.976	.937	.964
BIECON (Kim and Lee, 2017)	.965	.987	.970	.945	.931	.962	.952	.974	.980	.956	.923	.961
PQR (Zeng et al., 2017)	–	–	–	–	–	.971	–	–	–	–	–	.965
DNN (Bosse et al., 2016)	–	–	–	–	–	.972	–	–	–	–	–	.960
RankIQA+FT (Liu et al., 2017)	.975	.986	.994	.988	.960	.982	.970	.978	.991	.988	.954	.981
Hall.-IQA (Lin and Wang, 2018)	.977	.984	.993	.990	.960	.982	.983	.961	.984	.983	.989	.982
NSSADNN (Yan et al., 2019)	–	–	–	–	–	**.984**	–	–	–	–	–	**.986**
GADA (ours)	**.977**	.978	**.994**	.968	.943	.973	.963	.948	**.991**	.958	.917	.964

Table 3.2: Comparison between GADA and the state-of-the-art on LIVE.

Method	**#01**	**#02**	**#03**	**#04**	**#05**	**#06**	**#07**	**#08**	**#09**	**#10**	**#11**	**#12**	**#13**
BLIINDS-II (Saad et al., 2012)	0.714	0.728	0.825	0.358	0.852	0.664	0.780	0.852	0.754	0.808	0.862	0.251	0.755
BRISQUE (Mittal et al., 2012)	0.630	0.424	0.727	0.321	0.775	0.669	0.592	0.845	0.553	0.742	0.799	0.301	0.672
CORNIA-10K (Ye et al., 2012)	0.341	-0.196	0.689	0.184	0.607	-0.014	0.673	**0.896**	0.787	0.875	0.911	0.310	0.625
HOSA (Xu et al., 2016)	0.853	0.625	0.782	0.368	0.905	0.775	0.810	0.892	0.870	0.893	**0.932**	**0.747**	0.701
RankIQA+FT (Liu et al., 2017)	0.667	0.620	0.821	0.365	0.760	0.736	0.783	0.809	0.767	0.866	0.878	0.704	**0.810**
NSSADNN (Yan et al., 2019)	–	–	–	–	–	–	–	–	–	–	–	–	–
HALLUCINATED IQA (Lin and Wang, 2018)	0.923	0.880	**0.945**	0.673	**0.955**	0.810	0.855	0.832	**0.957**	**0.914**	0.624	0.460	0.782
GADA (ours)	**0.932**	**0.897**	0.943	**0.825**	0.949	**0.920**	**0.919**	0.790	0.881	0.775	0.886	0.435	0.702

Method	**#14**	**#15**	**#16**	**#17**	**#18**	**#19**	**#20**	**#21**	**#22**	**#23**	**#24**	**ALL**
BLIINDS-II (Saad et al., 2012)	0.081	0.371	0.159	-0.082	0.109	0.699	0.222	0.451	0.815	0.568	0.856	0.550
BRISQUE (Mittal et al., 2012)	0.175	0.184	0.155	0.125	0.032	0.560	0.282	0.680	0.804	0.715	0.800	0.562
CORNIA-10K (Ye et al., 2012)	0.161	0.096	0.008	0.423	-0.055	0.259	0.606	0.555	0.592	0.759	0.903	0.651
HOSA (Xu et al., 2016)	0.199	0.327	0.233	0.294	0.119	0.782	0.532	0.835	0.855	0.801	**0.905**	0.728
RankIQA+FT (Liu et al., 2017)	0.512	**0.622**	**0.268**	0.613	0.662	0.619	0.644	0.800	0.779	0.629	0.859	0.780
NSSADNN (Yan et al., 2019)	–	–	–	–	–	–	–	–	–	–	–	0.844
HALLUCINATED IQA (Lin and Wang, 2018)	**0.664**	0.122	0.182	0.376	0.156	0.850	0.614	0.852	**0.911**	0.381	0.616	**0.879**
GADA (ours)	0.206	0.200	0.196	**0.739**	**0.688**	**0.950**	**0.679**	**0.937**	0.895	**0.843**	0.889	0.790

Table 3.3: Comparison between GADA and the state-of-the-art on TID2013 (SROCC).

3.4 Conclusions

With this work we proposed a new approach called GADA to resolve the problem of lack of training data for no-reference image quality assessment. Our approach uses a modified Auxiliary Classifier GAN. This technique allows us to use the generator to generate new training examples and to train a regressor which estimates the image quality score. The results obtained on LIVE and TID2013 datasets show that our performance is comparable with the best methods of the state-of-the-art. Moreover, the very shallow network used for the regressor can process images with an high frame rate (about 120 image per second). This is in stark contrast to state-of-the-art approaches which typically use very deep models like VGG16 pre-trained on ImageNet.

We feel that the GADA approach offers a promising alternative to laboriously annotating images for IQA. Significant improvements can likely be made, especially for highly local distortions, through saliency-based sampling of image patches during training.

Chapter 4

Recurrent Attention to Transient Tasks for Continual Image Captioning[†]

Classical supervised learning systems acquire knowledge by providing them with a set of annotated training samples from a task, which for classifiers is a single set of classes to learn. This view of supervised learning stands in stark contrast with how humans acquire knowledge, which is instead *continual* in the sense that mastering new tasks builds upon previous knowledge acquired when learning previous ones. This type of learning is referred to as *continual* learning (sometimes *incremental* or *lifelong* learning), and continual learning systems instead consume a sequence of tasks, each containing its own set of classes to be learned. Through a sequence of *learning sessions*, in which the learner has access only to labeled examples from the current task, the learning system should integrate knowledge from past and current tasks in order to accurately master them all in the end. A principal shortcoming of state-of-the-art learning systems in the continual learning regime is the phenomenon of *catastrophic forgetting* (Goodfellow et al., 2013; Kirkpatrick et al., 2017): in the absence of training samples from previous tasks, the learner is likely to *forget* them in the process of acquiring new ones.

Continual learning research has until now concentrated primarily on classification problems modeled with deep, feed-forward neural networks (De Lange et al., 2019; Parisi et al., 2019). Given the importance of recurrent networks for many learning problems, it is surprising that continual learning of recurrent networks has received so little attention (Coop and Arel, 2013; Sodhani et al., 2019). A recent study on catastrophic forgetting in deep LSTM networks (Schak and Gepperth, 2019) observes that forgetting is more pronounced than in feed-forward networks. This is caused by the recurrent connections which amplify each small change in

[†] Portions of this chapter were published in: R. Del Chiaro, B. Twardowski, A. D. Bagdanov, and J. van de Weijer, "RATT: Recurrent Attention to Transient Tasks for continual image captioning." *Advances in Neural Information Processing Systems* (*NeurIPS*), 2020.

the weights. In this chapter, we consider continual learning for captioning, where a recurrent network (LSTM) is used to produce the output sentence describing an image. Rather than having access to all captions jointly during training, we consider different captioning tasks which are learned in a sequential manner (examples of tasks could be captioning of sports, weddings, news, etc).

Most continual learning settings consider tasks that each contain a set of classes, and these sets are disjoint (Pfülb and Gepperth, 2019; Rebuffi et al., 2017; Serra et al., 2018). A key aspect of continual learning for image captioning is the fact that tasks are naturally split into overlapping vocabularies. Task vocabularies might contain nouns and some verbs which are specific to a task, however many of the words (adjectives, adverbs, and articles) are *shared* among tasks. Moreover, the presence of homonyms in different tasks might directly lead to forgetting of previously acquired concepts. This *transient* nature of words in task vocabularies makes continual learning in image captioning networks different from traditional continual learning.

In this chapter we take a systematic look at continual learning for image captioning problems using recurrent, LSTM networks. We consider three of the principal classes of approaches to exemplar-free continual learning: weight-regularization approaches, exemplified by Elastic Weight Consolidation (EWC) (Kirkpatrick et al., 2017); knowledge distillation approaches, exemplified by Learning without Forgetting (LwF) (Li and Hoiem, 2017); and attention-based approached like Hard Attention to the Task (HAT) (Serra et al., 2018). For each we propose modifications specific to their application to recurrent LSTM networks, in general, and more specifically to image captioning in the presence of transient task vocabularies.

The contributions of this work are threefold: (1) we propose a new framework and splitting methodologies for modeling continual learning of sequential generation problems like image captioning; (2) we propose an approach to continual learning in recurrent networks based on transient attention masks that reflect the transient nature of the vocabularies underlying continual image captioning; and (3) we support our conclusions with extensive experimental evaluation on our new continual image captioning benchmarks and compare our proposed approach to continual learning baselines based on weight regularization and knowledge distillation. To the best of our knowledge we are the first to consider continual learning of sequential models in the presence of *transient tasks vocabularies* whose classes may appear in some learning sessions, then disappear, only to reappear in later ones.

4.1 Related work

Catastrophic forgetting. Early works demonstrating the inability of networks to retain knowledge from previously task when learning new ones were conducted by McCloskey and Cohen (1989) and Goodfellow et al. (2013). Approaches include methods that mitigate catastrophic forgetting via replay of exemplars like iCarl from Rebuffi et al. (2017), EEIL from Castro et al. (2018) and GEM from Lopez-Paz and Ranzato (2017) or by performing pseudo-replay with GAN-generated data (Liu et al., 2020; Shin et al., 2017; Wu et al., 2018). Weight regularization has also been investigated by Aljundi et al. (2018), Kirkpatrick et al. (2017) and Zenke et al. (2017). Output regularization via knowledge distillation was investigated in LwF (Li and Hoiem, 2017), as well as architectures based on network growing (Rusu et al., 2016; Schwarz et al., 2018) and attention masking (Mallya et al., 2018; Masana et al., 2020; Serra et al., 2018). For more details we refer to recent surveys on continual learning (Parisi et al., 2019; De Lange et al., 2019).

Image captioning. Modern captioning techniques are inspired by machine translation and usually employ a CNN image encoder and RNN text decoder to "translate" images into sentences. NIC (Vinyals et al., 2015) uses a pre-trained CNN to encode the image and initialize an LSTM decoder. Differently Mao et al. (2015) use image features at each time step, while Donahue et al. (2015) employed a two-layer LSTM. Recurrent latent variable was introduced by Chen and Lawrence Zitnick (2015), encoding the visual interpretation of previously-generated words and acting as a long-term visual memory during next words generation. Xu et al. (2015) introduced a spatial attention mechanism: the model is able to focus on specific regions of the image according to the previously generated words. *ReviewerNet* (Yang et al., 2016) also selects in advance which part of the image will be attended, so that the decoder is aware of it from the beginning. *Areas of Attention* (Pedersoli et al., 2017) models the dependencies between image regions and generated words given the RNN state. A visual sentinel is introduced by Lu et al. (2017) to determine, at each decoding step, if it is important to attend the visual features. Anderson et al. (2018) mixed bottom-up attention (implemented with an object detection network in the encoder) and a top-down attention mechanism in the LSTM decoder that attend to the visual features of the salient image regions selected by the encoder. Recently, transformer-based methods (Vaswani et al., 2017) have been applied to image captioning (Huang et al., 2019; Herdade et al., 2019; Cornia et al., 2020), eliminating the LSTM in the decoder.

The focus of this chapter is to study how RNN-based captioning architectures are affected by catastrophic forgetting. For more details on image captioning we refer to recent surveys (Hossain et al., 2019; Li et al., 2019).

Continual learning of recurrent networks. A fixed expansion layer technique was proposed by Coop and Arel (2013) to mitigate forgetting in RNNs. A dedicated network layer that exploits sparse coding of RNN hidden state is used to reduce the overlap of pattern representations. In this method the network grows with each new task. Sodhani et al. (2019) used a Net2Net technique for expanding the RNN. The method uses GEM (Lopez-Paz and Ranzato, 2017) for training on a new task, but has several shortcomings: model weights continue to grow and it must retain previous task data in the memory.

Experiments on four synthetic datasets were conducted by Schak and Gepperth (2019) to investigate forgetting in LSTM networks. The authors concluded that the LSTM topology has no influence on forgetting. This observations motivated us to take a close look to continual image captioning where the network architecture is more complex and an LSTM is used as a output decoder.

4.2 Continual LSTMs for transient tasks

We first describe our image captioning model and some details of LSTM networks. Then we describe how to apply classical continual learning approaches to LSTM networks.

4.2.1 Image captioning model

We use a captioning model similar to Neural Image Captioning (NIC) originally proposed by Vinyals et al. (2015). It is an encoder-decoder network that "translates" an image into a natural language description. It is trained end-to-end, directly maximizing the probability of correct sequential generation:

$$\hat{\theta} \;=\; \arg\max_{\theta} \sum_{(I,\,\bar{s})} \log p(s_N \mid I,\, s_1,\, \dots,\, s_{N-1};\, \theta). \tag{4.1}$$

where $\bar{s} = [s_1,\, \dots,\, s_N]$ is the target sentence for image I, θ are the model parameters.

The decoder is an LSTM network in which words $s_1, \dots, s_{n-1}$ are encoded in the hidden state h_n and a linear classifier is used to predict the next word at time step n:

$$x_0 \;=\; V\,\mathrm{CNN}(I) \tag{4.2}$$

$$x_n \;=\; S\,s_n \tag{4.3}$$

$$h_n \;=\; \mathrm{LSTM}(x_n,\, h_{n-1}) \tag{4.4}$$

$$p_{n+1} \;=\; C\,h_n \tag{4.5}$$

where S is a word embedding matrix, s_n is the n-th word of the ground-truth sentence for image I, C is a linear classifier, and V is the visual projection matrix that projects image features from the CNN encoder into the embedding space at time $n = 0$.

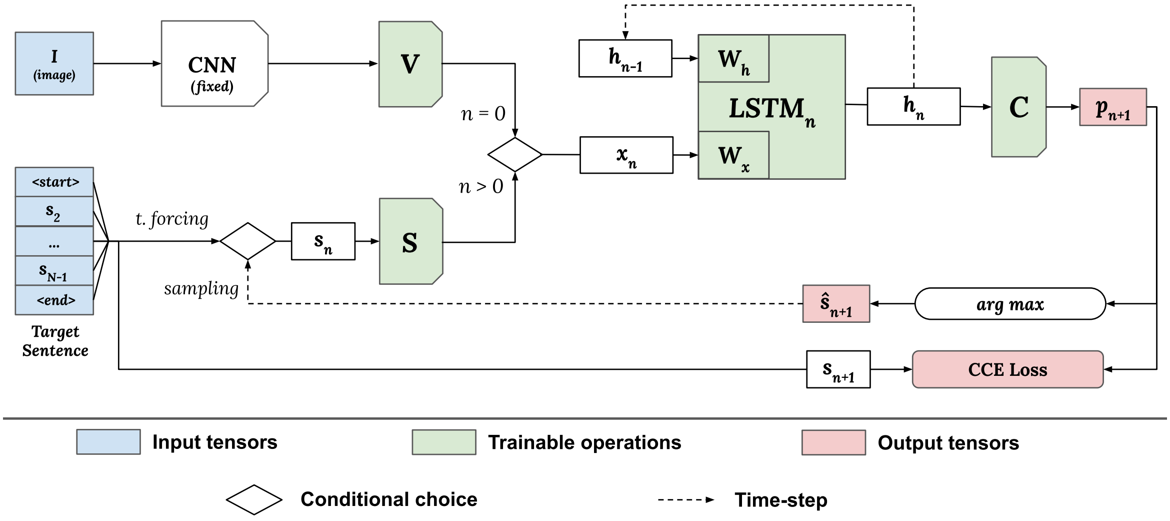


Figure 4.1: The Neural Image Captioning architecture.

The LSTM network is defined by the following equations (for which we omit the bias terms):

$$
\begin{aligned}
i_n &= \sigma(W_{ix}\, x_n + W_{ih}\, h_{n-1}) && (4.6)\\
o_n &= \sigma(W_{ox}\, x_n + W_{oh}\, h_{n-1}) && (4.7)\\
f_n &= \sigma(W_{fx}\, x_n + W_{fh}\, h_{n-1}) && (4.8)\\
g_n &= \tanh(W_{gx}\, x_n + W_{gh}\, h_{n-1}) && (4.9)\\
h_n &= o_n \odot c_n && (4.10)\\
c_n &= f_n \odot c_{n-1} + i_n \odot g_n && (4.11)
\end{aligned}
$$

where $\odot$ is the Hadamard (element-wise) product, σ the logistic function, c the LSTM cell state. The W matrices are the trainable LSTM parameters related to input x and hidden state h, for each gate i, f, o, g. The loss used to train the network is the sum of the negative log likelihood of the correct word at each step:

$$\mathcal{L}(x,\, \bar{s}) = -\sum_{n=1}^{N} \log p_n(s_n). \tag{4.12}$$

In figure 4.1 is shown a schematization of the described captioning model.

Inference. During training we perform teacher forcing using the n-th word of the target sentence as input to predict word $n+1$. At inference time, since we have no target caption, we use the word predicted by the model at the previous step $\arg\max\ p_n$ as input to the word embedding matrix S.

4.2.2 Continual learning of recurrent models

Normally catastrophic forgetting is highlighted in continual learning benchmarks by defining tasks that are mutually disjoint in the classes they contain (i.e. no class belongs to more than one task). For sequential problems like image captioning, however, this is not so easy: sequential learners must classify *words* at each decoding step, and a large vocabulary of *common* words are needed for any practical captioning task.

Incremental model. Our models are trained on sequences of captioning tasks, each having different vocabularies. For this reason any captioning model must be able to enlarge its vocabulary. When a new task arrives we add a new column for each new word in the classifier and word embedding matrices. The recurrent network remains untouched because the embedding projects inputs into the same space. The basic approach to adapt to the new task is to fine-tune the network over the new training set. To manage the different classes (words) of each task we have two possibilities: (1) Use different classifier and word embedding matrices for each task; or (2) Use a common, growing classifier and a common, growing word embedding matrix.

The first option has the advantage that each task can benefit from ad hoc weights for the task, potentially initializing from the previous task for the common words. However, it also increases decoder network size consistently with each new task. The second option has the opposite advantage of keeping the dimension of the network bounded, sharing weights for all common words. Because of the nature of the captioning problem, many words will be shared and duplicating both word embedding matrix and classifier for all the common words seems wasteful. Thus we adopt the second alternative. With this approach, the key trick is to *deactivate* classifier weights for words not present in the current task vocabulary.

We use $\hat{\theta}^t$ to denote optimal weights learned for task t on dataset D_t. After training on task t, we create a new model for task $t+1$ with expanded weights for classifier and word embedding matrices. We use weights from $\hat{\theta}^t$ to initialize the shared weights of the new model.

4.2.3 Recurrent continual learning baselines

We describe how to adapt two common continual learning approaches, one based on weight regularization and the other on knowledge distillation. We will use these as baselines in our comparison.

Weight regularization. A common method to prevent catastrophic forgetting is to apply regularization to important model weights before proceeding to learn a new task (Aljundi et al., 2018; Chaudhry et al., 2018; Kirkpatrick et al., 2017; Zenke et al., 2017). Such methods can be directly applied to recurrent models with little effort. We choose Elastic Weight Consolidation (EWC) (Serra et al., 2018) as a regularization-based baseline. The key idea of EWC is to limit change to model parameters vital to previously-learned tasks by applying a quadratic penalty to them depending on their importance. Parameter importance is estimated using a diagonal approximation of the Fisher Information Matrix. The additional loss function we minimize when learning task t is:

$$\mathcal{L}^t_{\text{EWC}}(x,\ \bar{s};\ \theta^t) = \mathcal{L}(x,\ \bar{s}) + \lambda \sum_i \frac{1}{2}\, F_i^{t-1}(\theta_i^t - \hat{\theta}_i^{t-1})^2, \tag{4.13}$$

where $\hat{\theta}^{t-1}$ are the estimated model parameters for the previous task, θ^t are the model parameters at the current task t, $\mathcal{L}(x,\ \bar{s})$ is the standard loss used for fine-tuning the network on task t, i indexes the model parameters shared between tasks t and $t-1$, F_i^{t-1} is the i-th element of a diagonal approximation of the Fisher Information Matrix for model after training on task $t-1$, and λ weights the importance of the previous task. We apply Eq. 4.13 to all trainable weights. Due to the transient nature of words across tasks, we do not expect weight regularization to be optimal since some words are shared and regularization limits the plasticity needed to adjust to a new task.

Optimizing $\mathcal{L}^t_{EWC}(\theta)$ (equation 4.13) we obtain $\hat{\theta}^t$ and when proceeding to the task $t+1$ we again use $\mathcal{L}^t_{EWC}$ but supplying instead $\hat{\theta}^t_i$ as its argument. $\mathcal{L}^t_{EWC}$ is applied to every trainable weight of our network, but we have special cases for the weights of the word embedding and final classifier: these are only *partially* shared between tasks. At each new task some of the weights will be completely new since they are related to new words, we do not want to force these weights to stay where they are since they have never been trained before, and so we do not regularize them. Note this problem is not present in the standard continual learning for classification because each new task has a disjoint set of classes and a dedicated classifier is used per task.

Recurrent Learning without Forgetting. We also apply a knowledge distillation (Hinton et al., 2015) approach inspired by Learning without Forgetting (LwF) (Li and Hoiem, 2017) on the LSTM decoder network to prevent catastrophic forgetting. The model after training task $t-1$ is used as a teacher network when fine-tuning on task t. The aim is to let the new network freely learn how to classify new words appearing in task t while keeping stable the predicted probabilities for words from previous tasks.

To do this, at each step n of the decoder network the previous decoder is also fed with the data coming from the new task t. Note that the input to the LSTM at each step n is the embedding of the $n-$th word in the target caption, and the same embedding is given as input to both teacher and student networks – i.e. the student network's embedding of word n is also used as input for the teacher, while each

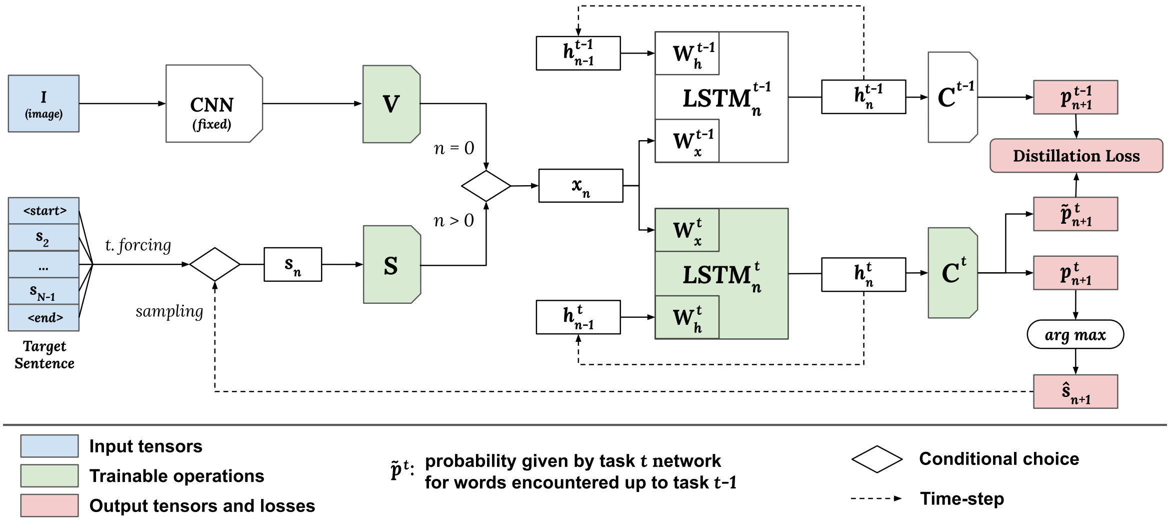


Figure 4.2: Transient Learning without Forgetting: adaptation of LwF to the Neural Image Captioning architecture. Predictions from the model at $t-1$ are used during training of task t to compute the distillation loss. W'

network uses its own hidden state h_{n-1} and cell state c_{n-1} to decode the next word. At each decoding step we define the output probabilities from the student LSTM network corresponding only to words encountered up to the previous task as $\tilde{p}^t_{n+1}$. These are compared with the output probabilities p^{t-1}_{n+1} predicted by the teacher network. A distillation loss ensures that the student network does not deviate from the teacher:

$$\mathcal{L}^t_{\text{LwF}}(\tilde{p}^t,\ p^{t-1}) \quad = \quad -\sum_n H(\gamma(\tilde{p}^t_n),\ \gamma(p^{t-1}_n)) \tag{4.14}$$

where $\gamma(\cdot)$ rescales a probability vector p -with temperature parameter T, $\tilde{p}^t_n$ and p^{t-1}_n are respectively the predicted probability for all the words in tasks $1,\ldots,t-1$ at LSTM step n for the current model we are training and the best model trained for the previous tasks. This loss is combined with the LSTM training loss (see Eq. 4.12). The final loss for training the decoder network with LwF is:

$$\begin{aligned}\mathcal{L}^t(x,S) &= \mathcal{L}(x,S) + \mathcal{L}^t_{\text{LwF}}(\tilde{p}^t, p^{t-1})\\ &= -\sum_{n=1}^{N}\Big[\log p^t_n\ (s_n) - \lambda H(\gamma(\tilde{p}^t_n), \gamma(p^{t-1}_n))\Big]\end{aligned} \tag{4.15}$$

where λ is the hyperparameter weighting the importance of the previous task. Note that differently from (Li and Hoiem, 2017), we do not fine-tune the classifier of the old network because we use a single, incremental word classifier. An overview of *recurrent learning without forgetting* is provided in figure 4.2.

4.3 Attention for continual learning of transient tasks

Inspired by the Hard Attention to the Task (HAT) method (Serra et al., 2018), we developed an attention-based technique applicable to recurrent networks. We name it *Recurrent Attention to Transient Tasks* (*RATT*), since it is specifically designed for recurrent networks with task transience. The key idea is to use an attention mechanism to allocate a portion of the activations of each layer to a *specific* task t. An overview of RATT is provided in figure 4.3.

Attention masks. The number of neurons used for a task is limited by two task-conditioned attention masks: embedding attention $a_x^t \in [0,1]$ and hidden state attention $a_h^t \in [0,1]$. These are computed with a sigmoid activation σ and a positive scaling factor s according to:

$$a_x^t = \sigma(sA_x t^T) \ , \ a_h^t = \sigma(sA_h t^T), \tag{4.16}$$

where t is a one-hot task vector, and A_x and A_h are embedding matrices. Next to the two attention mask, we have a vocabulary mask a_s^t which is a binary mask identifying the words of the vocabulary used in task t: $a_{s,i}^t = 1$ if word i is part of the vocabulary of task t and is zero otherwise. The forward pass (see Eqs.4.2 and 4.5) of the network is modulated with the attention masks according to:

$$\begin{array}{rclr@{\qquad}rclr} \bar{x}_0 &=& x_0 \odot a_x^t & (4.17) & \bar{h}_n &=& h_n \odot a_h^t & (4.19) \\ \bar{x}_n &=& x_n \odot a_x^t & (4.18) & \bar{p}_{n+1} &=& p_{n+1} \odot a_s^t & (4.20) \end{array}$$

Attention masks act as an inhibitor when their value is near 0. The main idea is to learn attention masks during training, and as such learn a limited set of neurons for each task. Neurons used in previous tasks can still be used in subsequent ones, however the weights which were important for previous tasks have reduced plasticity (depending on the amount of attention to for previous tasks).

Training. For training we define the cumulative forward mask as:

$$a_x^{<t} \quad = \quad \max(a_x^{t-1}, a_x^{<t-1}), \tag{4.21}$$

$a_h^{<t}$ and $a_s^{<t}$ are similarly defined. We now define the following backward masks which have the dimensionality of the weight matrices of the network and are used to selectively backpropagate the gradient to the LSTM layers:

$$B_{h,ij}^t = 1 - \min(a_{h,i}^{<t}, a_{h,j}^{<t}) \ , \ B_{x,ij}^t = 1 - \min(a_{h,i}^{<t}, a_{x,j}^{<t}) \tag{4.22}$$

Note that we use $a_{h,i}$ refer to the i-th element of vector a_h, etc. The backpropagation with learning rate λ is then done according to

$$W_h \leftarrow W_h - \lambda B_h^t \odot \frac{\partial \mathcal{L}^t}{\partial W_h} \ , \ W_x \leftarrow W_x - \lambda B_x^t \odot \frac{\partial \mathcal{L}^t}{\partial W_x}. \tag{4.23}$$

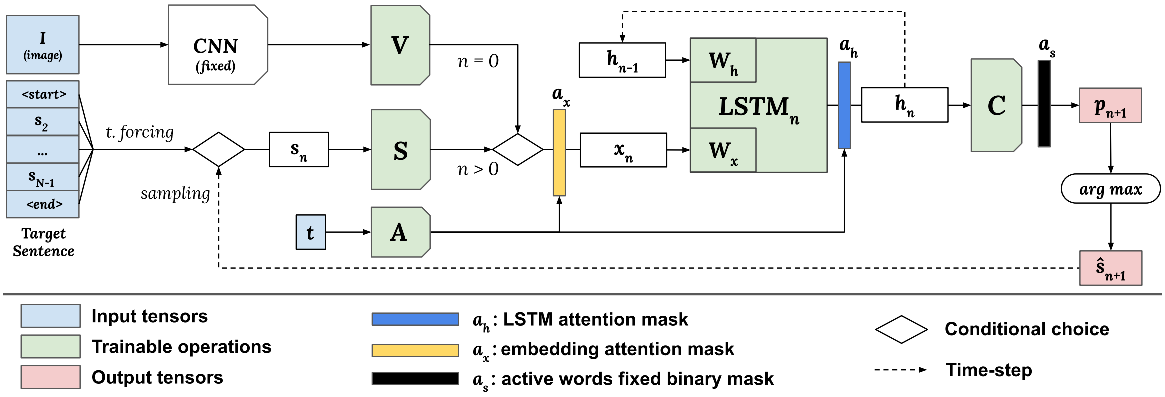


Figure 4.3: Recurrent Attention to Transient Tasks (RATT). See section 4.3 for a detailed description of each component of the continual captioning network.

The only difference from standard backpropagation are the backward matrices B which prevents the gradient from changing those weights that were highly attended in previous tasks. The backpropagation updates to the other matrices in Eqs. 4.6-4.9 are similar.

Differently than Serra et al. (2018) we also define backward masks for the word embedding matrix S, the linear classifier C, and the image-projection matrix V:

$$B^t_{S,ij} = 1 - \min(a^{<t}_{x,i}, a^{<t}_{s,j}) \ , \ B^t_{C,ij} = 1 - \min(a^{<t}_{s,i}, a^{<t}_{h,j}) \ , \ B^t_{V,ij} = 1 - a^{<t}_{x,i}, \qquad (4.24)$$

and the corresponding backpropagation updates:

$$S \leftarrow S - \lambda B^t_S \odot \frac{\partial \mathcal{L}^t}{\partial S} \ , \ C \leftarrow C - \lambda B^t_C \odot \frac{\partial \mathcal{L}^t}{\partial C} \ , \ V \leftarrow V - \lambda B^t_V \odot \frac{\partial \mathcal{L}^t}{\partial V}. \qquad (4.25)$$

The backward mask B^t_V modulates the backpropagation to the image features. Since we do not define a mask on the output of the fixed image encoder, this is only defined by $a^{<t}_x$.

Linearly annealing the scaling parameter s, used in Eq. 4.16, during training (like (Serra et al., 2018)) was found to be beneficial. We apply:

$$s = \frac{1}{s_{max}} + \left(s_{max} - \frac{1}{s_{max}}\right) \frac{b-1}{B-1} \qquad (4.26)$$

where b is the batch index and B is the total number of batches for the epoch. We used $s_{max} = 2000$ and $s_{max} = 400$ for experiments on Flickr30k and MS-COCO, respectively.

The loss used to promote low network usage and to keep some neurons available for future tasks is:

$$\mathcal{L}^t_a = \frac{\sum_i a^t_{x,i}(1 - a^{<t}_{x,i})}{\sum_i (1 - a^{<t}_{x,i})} + \frac{\sum_i a^t_{h,i}(1 - a^{<t}_{h,i})}{\sum_i (1 - a^{<t}_{h,i})}. \qquad (4.27)$$

This loss is combined with Eq. 4.12 for training. The loss encourages attention to only a few new neurons. However, tasks can attend to previously attended neurons without any penalty. This encourages forward transfer during training. If the attention masks are binary, the system would not suffer from any forgetting, however it would lose its backward transfer ability.

Differently than Serra et al. (2018), when computing B_S^t we take into account the recurrency of the network, considering the classifier C to be the previous layer of S. In addition, our output masks a_s allow for overlap to model the transient nature of the output vocabularies, whereas Serra et al. (2018) only considers non-overlapping classes for the various tasks.

The final loss for training the decoder network with RATT is:

$$\mathcal{L}^t(x,\bar{s}) = \mathcal{L}(x,\bar{s}) + \mathcal{L}_a^t = -\sum_{n=1}^{N} \log p_n(s_n) + \lambda \frac{\sum_i a_{x,i}^t(1-a_{x,i}^{<t})}{\sum_i(1-a_{x,i}^{<t})} + \lambda \frac{\sum_i a_{h,i}^t(1-a_{h,i}^{<t})}{\sum_i(1-a_{h,i}^{<t})}$$

where λ is the hyperparameter weighting the importance of future tasks: for larger λ, fewer neurons will be allocated to the current task (and more neurons will be available for the future tasks).

The backpropagation updates for each LSTM gate matrix are:

$$\begin{aligned}
W_{ih} &\leftarrow W_{ih} - \lambda B_{ih}^t \odot \frac{\partial \mathcal{L}^t}{\partial W_{ih}} & \qquad W_{fh} &\leftarrow W_{fh} - \lambda B_{fh}^t \odot \frac{\partial \mathcal{L}^t}{\partial W_{fh}} \\
W_{ix} &\leftarrow W_{ix} - \lambda B_{ix}^t \odot \frac{\partial \mathcal{L}^t}{\partial W_{ix}} & W_{fx} &\leftarrow W_{fx} - \lambda B_{fx}^t \odot \frac{\partial \mathcal{L}^t}{\partial W_{fx}} \\
W_{oh} &\leftarrow W_{oh} - \lambda B_{oh}^t \odot \frac{\partial \mathcal{L}^t}{\partial W_{oh}} & W_{gh} &\leftarrow W_{gh} - \lambda B_{gh}^t \odot \frac{\partial \mathcal{L}^t}{\partial W_{gh}} \\
W_{ox} &\leftarrow W_{ox} - \lambda B_{ox}^t \odot \frac{\partial \mathcal{L}^t}{\partial W_{ox}} & W_{gx} &\leftarrow W_{gx} - \lambda B_{gx}^t \odot \frac{\partial \mathcal{L}^t}{\partial W_{gx}}
\end{aligned}$$

During training, we applied the gradient compensation procedure described from Serra et al. (2018) to help training the task-embedding matrices A_x and A_h:

$$\begin{aligned}
A_{x,i} &\leftarrow \frac{s_{max}[\cosh(sA_{x,i}t^T)+1]}{s[\cosh(A_{x,i}t^T)+1]} \frac{\partial \mathcal{L}^t}{\partial A_{x,i}} \\
A_{h,i} &\leftarrow \frac{s_{max}[\cosh(sA_{h,i}t^T)+1]}{s[\cosh(A_{h,i}t^T)+1]} \frac{\partial \mathcal{L}^t}{\partial A_{h,i}}
\end{aligned}$$

Moreover, for numerical stability, we clamp $|s\ A_{x,i}t^T| \leq 50$ and $|s\ A_{h,i}t^T| \leq 50$.

4.4 Task splits for incremental captioning

Here we first describe the two splitting procedures we propose that are applicable to captioning datasets with categorical annotations. Then we describe how we apply them to the MS-COCO (Lin et al., 2014) and Flick30k (Plummer et al., 2017) datasets.

4.4.1 Disjoint visual categories

We exploit categorical image annotations available in many captioning datasets. If each image in the dataset belongs to a single category, we can simply define each task as a set of categories that does not overlap with any other task. If an image can belong to multiple categories we instead use the following procedure:

1. **Define K tasks**. Tasks are sets $\mathcal{C}_t$ of categories such that: $\mathcal{C}_i \cap \mathcal{C}_j = \emptyset \; \forall i \neq j$.

2. **Identify candidate example sets**. For each task t select all the examples in the original dataset having at least one of the labels in common with task t categories:
$$\mathrm{P}_t = \{i \mid \exists \, c \in \mathcal{C}_t \text{ s.t. } y^i_c = 1\} \tag{4.28}$$
where i is the index of example in the original dataset and $y^i \in \{0,1\}^{|\mathcal{C}_t|}$ is a multi-label vector such that $y^i_c = 1 \Leftrightarrow$ the i-th example belongs to category c.

3. **Identify common examples sets**. Find common examples in candidate sets: $\mathrm{Q}_{i,j} = \mathrm{P}_i \cap \mathrm{P}_j$

4. **Define final task examples**. Define example sets of each task t as: $\mathrm{E}_t = \mathrm{P}_t \setminus \cup_{i \neq t}(\mathrm{Q}_{t,i})$

This guarantees that if an image belongs to multiple tasks due to its labels, it will be completely pruned from the dataset instead of added to both or added to only one.

4.4.2 Incremental visual categories

As an alternative to visually-disjoint task splits, we also evaluate continual image captioning in a more real-life setting, where a first task contains a set of visual concepts that can reappear in following tasks. Subsequent tasks contain new or more specific concepts, without the guarantee of having no overlap with the already seen data. The idea is to train the network over general concepts and then progressively train it on more specific ones. The network should continue to perform well on old tasks without overfitting to the more recently seen. The procedure is as follows (note that two first steps are the same as before):

1. **Define K tasks**. Tasks are sets $\mathcal{C}_t$ of categories.
2. **Identify candidate example sets**. As in point (2) of the previous procedure:

$$\mathrm{P}_t = \{i \mid \exists\, c \in \mathcal{C}_t \text{ s.t. } y^i_c = 1\} \tag{4.29}$$

   where i is the index of an example in the original dataset and $y^i \in \{0,1\}^{|\mathcal{C}_t|}$ is a multi-label vector such that $y^i_c = 1 \Leftrightarrow$ the i-th example belongs to category c.

3. **Define final task examples**. Define example sets of each task t as: $\mathrm{E}_t = \mathrm{P}_t \setminus \cup_{i=t}^{K} (\mathrm{P}_t \cap \mathrm{P}_i)$

Given the sets E_t we define the training set for the task t as:

$$\mathcal{D}_t = \{x^i, \bar{s}^{\,i,1}, \bar{s}^{\,i,2}, ...\bar{s}^{\,i,\Sigma} \mid i \in \mathrm{E}_t\} \tag{4.30}$$

where $\bar{s}^{\,i,j}$ is a sentence describing image x^i and Σ is the number of sentences describing each image.

4.4.3 An MSCOCO task split

We applied the *disjoint visual categories* splitting procedure to arrive at the following task split for MS-COCO (Lin et al., 2014):

- **transport**: bicycle, car, motorcycle, airplane, bus, train, truck, boat.
- **animals**: bird, horse, sheep, cow, elephant, bear, zebra, giraffe.
- **sports**: snowboard, sports ball, kite, baseball bat, baseball glove, skateboard, surfboard, tennis racket.
- **food**: banana, apple, sandwich, orange, broccoli, carrot, hot dog, pizza, donut, cake.
- **interior**: chair, couch, potted plant, bed, toilet, tv, laptop, mouse, remote, keyboard, cell phone, microwave, oven, toaster, sink, refrigerator.

We removed categories *dog* and *cat* from *animals* because objects of these classes are very likely to appear also in images of *interiors* and *sports* tasks. For the same reason we removed *dinning table* from *interior* because of the likely overlap with the *food* task. In table 4.1 we report word overlaps between tasks for our MS-COCO splits. From this breakdown we see that the task vocabularies are approximately the same size (between around 2,000 and 3,000 words), and there is significant overlap between all tasks. MS-COCO does not provide a test set, so we randomly selected half of the validation set images and used them for testing only. Since images have at least five captions, we used the first five captions for each image as the target.

	T	A	S	F	I
T	3,116 (100.0%)	1,499 (48.11%)	1,400 (44.93%)	1,222 (39.22%)	1,957 (62.80%)
A	1,499 (48.11%)	2,178 (100.0%)	1,175 (53.95%)	1,025 (47.06%)	1,492 (68.50%)
S	1,400 (44.93%)	1,175 (53.95%)	1,967 (100.0%)	933 (47.43%)	1,355 (68.89%)
F	1,222 (39.22%)	1,025 (47.06%)	933 (47.43%)	2,235 (100.0%)	1,530 (68.46%)
I	1,957 (62.80%)	1,492 (68.50%)	1,355 (68.89%)	1,530 (68.46%)	3,741 (100.0%)

Table 4.1: Word overlaps between tasks for our MS-COCO splits.

4.4.4 A Flickr30k task split

In the Flickr30k Entities dataset (Plummer et al., 2017) we have five captions per image and each caption is labeled with a set of *phrase types* that refers to parts of the sentence. We use the union of all phrase types associated to each example as the set of categories for that example. A subset of these categories is used to split the dataset using the *incremental visual categories* procedure. For this dataset we use a single category per task, so tasks are named after assigned categories. The list of categories (tasks) is: **scene**, **animals**, **vehicles**, and **instruments**. If a category is over-represented, random sub-sampling is done to get maximum of 7,500 examples (like in the case of **scene**). Moreover, the most common phrase type is **people** and we omit it in purpose because almost all photos contain people. In figure 4.4 we give the co-occurrence matrix between Flickr30k images and categories based on phrases types from Plummer et al. (2017). The influence of the **people** category is clearly visible.

4.5 Experimental results

All experiments use the same architecture: for the encoder network we used ResNet-152 (He et al., 2016) pre-trained on ImageNet (Russakovsky et al., 2015b). Note that the image encoder is frozen and is not trained during continual learning, as is common in many image captioning systems.The decoder consists of the word embedding matrix S that projects the input words into a 256-dimensional space, an LSTM cell with hidden size 512 that takes the word embedding (or image feature embedding for the first step) as input, and a final fully connected layer C that takes as input the hidden state h_n at each LSTM step n and outputs a probability distribution p_{n+1} over the $|V^t|$ words in the vocabulary for current task t.

We applied all techniques on the Flickr30K (Plummer et al., 2017) and MS-COCO (Lin et al., 2014) captioning datasets (see next section for task splits). All experiments were conducted using PyTorch, networks were trained using the Adam opti-

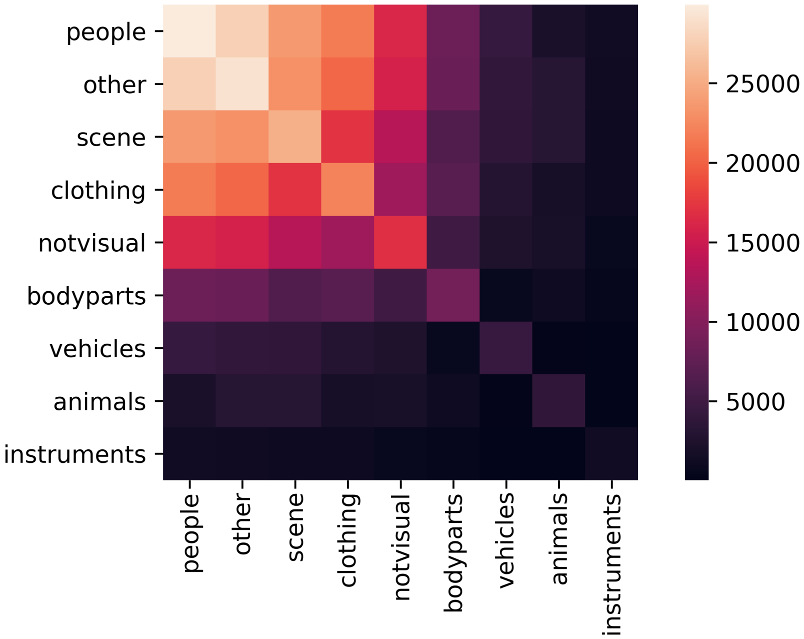


Figure 4.4: Flicker30r co-occurrence matrix for assigned categories.

mizer (Kingma and Ba, 2014) and all hyperparameters were tuned over validation sets. Batch size, learning rate and max-decode length for evaluation were set respectively to 128, 4e-4, and 26 for MS-COCO, and 32, 1e-4 and 40 for Flickr30k. These differences are due to the size of the training set and by the average caption lengths in the two datasets.

Inference at test time is *task-aware* for all methods. For EWC and LwF this means that we consider only the word classifier outputs corresponding to the correct task, and for RATT that we use the fixed output masks for the correct task. All metrics where computed using the nlg-eval toolkit (Sharma et al., 2017). Models where trained for a fixed number of epochs and the best model according to BLEU-4 performance on the validation set were chosen for each task. When proceeding to the next task, the best model from the previous task were used as a starting point.

4.5.1 Ablation study

We conducted a preliminary study on our split of MS-COCO to evaluate the impact of our proposed Recurrent Attention to Transient Tasks (RATT) approach. In this experiment we progressively introduce the attention masks described in section 4.3. We start with the basic captioning model with no forgetting mitigation, and so is equivalent to *fine-tuning*. Then we introduce the mask on hidden state h_n of the LSTM (along with the corresponding backward mask), and then the constant binary mask on the classifier that depends on the words of the current task, then the visual and word embedding masks, and finally the combination of all masks.

Task	Train	Valid	Test	Vocab (words)
transport	14,266	3,431	3,431	3,116
animals	9,314	2,273	2,273	2,178
sports	10,077	2,384	2,384	1,967
food	7,814	1,890	1,890	2,235
interior	17,541	4,340	4,340	3,741
total	59,012	14,318	14,318	6,344

(a) MS-COCO task statistics

Task	Train	Valid	Test	Vocab (words)
scene	5,000	170	170	2,714
animals	3,312	107	113	1,631
vehicles	4,084	123	149	2,169
instruments	1,290	42	42	848
total	18,283	607	636	4,123

(b) Flickr30k task statistics

Table 4.2: Number of images and words per task for our MS-COCO and Flickr30K splits.

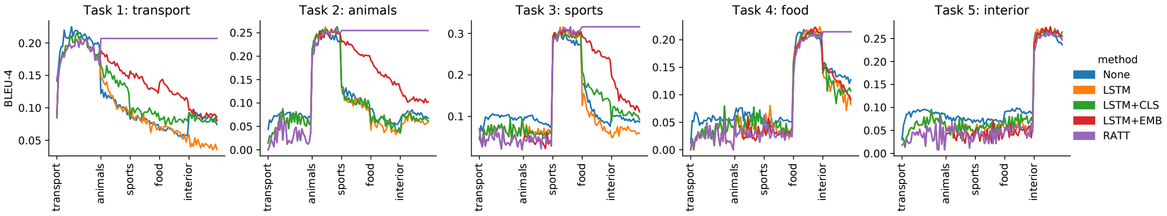


Figure 4.5: BLEU-4 performance for several ablations at each epoch over the whole sequence of MS-COCO tasks.

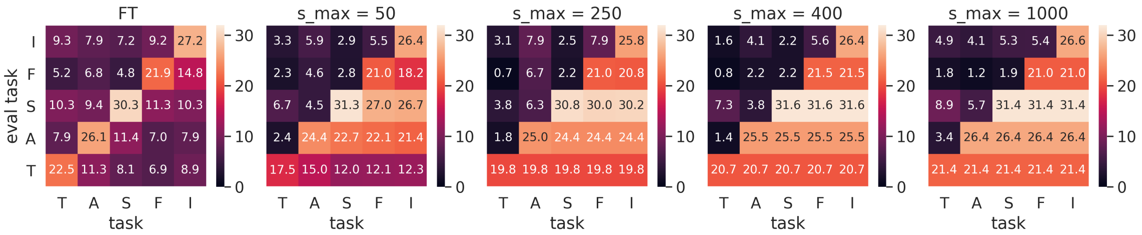


Figure 4.6: RATT ablation on the MS-COCO validation set using different s_{max} values and fine-tuning baseline. Each heatmap reports BLEU-4 performance for one of the ablated models evaluated on different tasks at the end of the training of each task.

Target a train riding on a track near a platform			
After training task 1 (Transport):		**After training task 5 (Interior):**	
FT	a train on a track near a platform	**FT**	a couple of rows of benches sitting next to a sidewalk
EWC	a train on a track near a platform	**EWC**	a train is pulling into a station with people standing on the platform
LwF	a train on a track near a platform	**LwF**	a train with a backpack and a backpack on the side of the road
RATT	a train is on the tracks at a station	**RATT**	a train is on the tracks at a station

Figure 4.7: Qualitative results for image captioning on MS-COCO. Forgetting for baseline methods can be clearly observed.

In figure 4.5 we plot the BLEU-4 performance of these configurations for each training epoch and each of the five MS-COCO tasks. Note that for later tasks the performance on early epochs (i.e. *before* encountering the task) is noisy as expected – we are evaluating performance on *future* tasks. These results clearly show that applying the mask to LSTM decreases forgetting in the early epochs when learning a new task. However, performance continues to decrease and in some tasks the result is similar to fine-tuning. Even if the LSTM is forced to remember how to manage hidden states for previous tasks, the other parts of the network suffer from catastrophic forgetting. Adding the classifier mask improves the situation, but the main contribution comes from applying the mask to the embedding. Applying all masks we obtain zero or nearly-zero forgetting. This depends on the s_{max} value used during training: in these experiments we use $s_{max} = 400$, which results in zero forgetting of previous tasks. We also conducted an ablation study on the s_{max} parameter. From the results in figure 4.6 we can see that higher s_{max} values improve old task performance, and sufficiently high values completely avoid forgetting. Using moderate values, however, can be helpful to increase performance in later tasks.

4.5.2 Results on MS-COCO

In table 4.3 we report the performance of a fine-tuning baseline with no forgetting mitigation (FT) and of EWC, LwF, and RATT on our splits for the MS-COCO captioning dataset. The forgetting percentage is computed by taking the BLEU-4 score for each model after training on the last task and dividing it by the BLEU-4 score at the end of the training of each individual task. From the results we see that all techniques consistently improve performance on previous tasks when compared to the FT baseline. Despite the simplicity of EWC, the improvement over fine-tuning is clear, but it struggles to learn a good model for the last task. LwF instead shows the opposite behavior: it is more capable of learning the last task, but forgetting is more noticeable. RATT achieves *zero* forgetting on MS-COCO, although at the cost of some performance on the final task. This is to be expected, though, as our approach deliberately and progressively limits network capacity to prevent forgetting of old tasks. Qualitative results on MS-COCO are provided in figure 4.7.

	Transport				Animals				Sports				Food				Interior			
	FT	EWC	LwF	RATT	FT	EWC	LwF	RATT	FT	EWC	LwF	RATT	FT	EWC	LwF	RATT	FT	EWC	LwF	RATT
BLEU-4	.0928	.1559	.1277	**.2126**	.0816	.1545	.1050	**.2468**	.0980	.2182	.1491	**.3161**	.1510	.1416	.1623	**.2169**	.2712	.2107	.2537	**.2727**
METEOR	.1472	.1919	.1708	**.2169**	.1396	.1779	.1577	**.2349**	.1639	.2209	.1918	**.2707**	.1768	.1597	.1962	**.2110**	**.2351**	.1967	.2286	.2257
CIDEr	.2067	.4273	.3187	**.6349**	.1480	.4043	.2158	**.7249**	.1680	.5146	.3277	**.8085**	.2668	.2523	.3816	**.5195**	**.6979**	.4878	.6554	.6536
% forgetting	59.1	31.2	43.7	0.0	67.5	33.8	45.0	0.0	68.9	23.6	45.0	0.0	32.8	14.6	16.5	0.0	N/A	N/A	N/A	N/A

Table 4.3: Performance on MS-COCO. Numbers are the per-task performance after training on the *last* task. Per-task forgetting in the last row is the BLEU-4 performance after the last task divided by the BLEU-4 performance measured immediately after learning each task.

	Scene				Animals				Vehicles				Instruments			
	FT	EWC	LwF	RATT	FT	EWC	LwF	RATT	FT	EWC	LwF	RATT	FT	EWC	LwF	RATT
BLEU-4	.1074	.1370	.1504	**.1548**	.1255	.1381	.1384	**.1921**	.1083	.1332	.1450	**.1724**	.1909	.2313	.1862	**.2386**
METEOR	.1570	.1722	**.1851**	.1710	.2046	.1833	.1954	**.2107**	.1625	.1770	**.1847**	.1750	**.1933**	.1714	.1876	.1782
CIDEr	.1222	.1688	.2402	**.2766**	.2460	.2755	.2756	**.4708**	.1586	.1315	.1748	**.2988**	.2525	.2611	**.2822**	.2329
% forgetting	31.1	11.3	2.7	-2.5	38.7	19.2	-15.1	0.0	35.6	4.9	-1.5	0.0	N/A	N/A	N/A	N/A

Table 4.4: Performance on Flickr30K. Evaluation is the same as for MS-COCO.

4.5.3 Results on Flickr30k

In table 4.4 we report performance of a fine-tuning baseline with no forgetting mitigation (FT), EWC, LwF, and RATT on our Flickr30k task splits. Because these splits are based on *incremental visual categories*, it does not reflect a classical continual-learning setup that enforce disjoint categories to maximize catastrophic forgetting: not only there are common words that share the same meaning between different tasks, but some of the visual categories in early tasks are also present in future ones. For this reason, learning how to describe task $t = 1$ also implies learning at least how to partially describe future tasks, so forward and backward transfer is significant.

Despite this, we see that all approaches increase performance on old tasks (when compared to FT) while retaining good performance on the last one. Note that both RATT and LwF result in *negative forgetting*: in these cases the training of a new task results in backward transfer that increases performance on an old one. EWC improvement is marginal, and LwF behaves a bit better and seems more capable of exploiting backward transfer. RATT backward transfer is instead limited by the choice of a high s_{max}, which however guarantees nearly zero forgetting.

4.5.4 Human evaluation experiments

We performed an evaluation based on human quality judgments using 200 images (40 from each task) from the MS-COCO test splits. We generated captions with

	MS-COCO					Flickr30k			
	T	**A**	**S**	**F**	**I**	**S**	**A**	**V**	**I**
RATT vs EWC	75.0%	77.5%	72.5%	85.0%	57.5%	61.8%	76.4%	67.3%	59.5%
RATT vs LwF	77.5%	82.5%	75.0%	62.5%	47.5%	45.5%	69.1%	63.6%	59.5%

Table 4.5: Human captioning evaluation on both MS-COCO and Flickr30k. For each task, we report the percentage of examples for which users preferred the caption generated by RATT.

RATT, EWC, and LwF after training on the last task and then presented ten users with an image and RATT and baseline captions in random order. Users were asked (using forced choice) to select which caption best represents the image content. A similar evaluation was performed for the Flickr30k dataset with twelve users. The percentage of users who chose RATT over the baseline are given in the table 4.5. These results on MS-COCO dataset confirm that RATT is superior on all tasks, while on Flickr30k there is some uncertainty on the first task, especially when comparing RATT with LwF. Note that for the last task of each dataset there is no forgetting, so it is expected that baselines and RATT perform similarly.

4.6 Additional ablations

In figure 4.8 we provide a different visualization of the RATT ablation reported above. Here we apply attention masking in different layers of the decoder architecture. In figure 4.8 we observe an increase of performance on old tasks when the classifier mask is used, and even more clearly when the embedding mask is used. Even further improvement in the performance is made when all the attention masks (the RATT approach) are used and there is no forgetting.

We also conducted an ablation study on the s_{max} parameter on Flickr30k, and results are reported in figure 4.9. Different visualizations for this ablation are shown in figure 4.10 (for MS-COCO) and 4.11 (for Flickr30k). From the MS-COCO experiment backward transfer for RATT is not noticeable, while for the Flickr30k case we observe in figure 4.11 that lower s_{max} values result in a small boost in performance for previous tasks when the training is started on each new one. However at the end of each training session the forgetting is always greater than the backwards transfer. Moreover, the model with highest s_{max} (purple line in figure 4.11) still shows a small amount of backward transfer, and in this case the performance gain is retained until the end of training. This is also noticeable in the last heatmap of figure 4.9 for the first task (Sport) (bottom row).

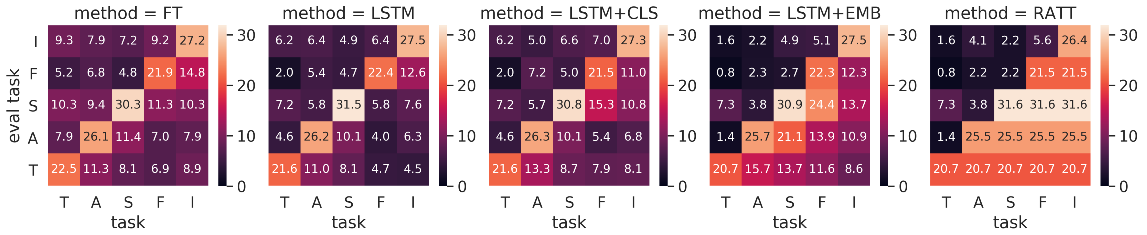


Figure 4.8: RATT ablation on the MS-COCO validation set using different attention masks. Each heatmap report BLEU-4 performance for one of the ablated models evaluated on different tasks at the end of the training of each task.

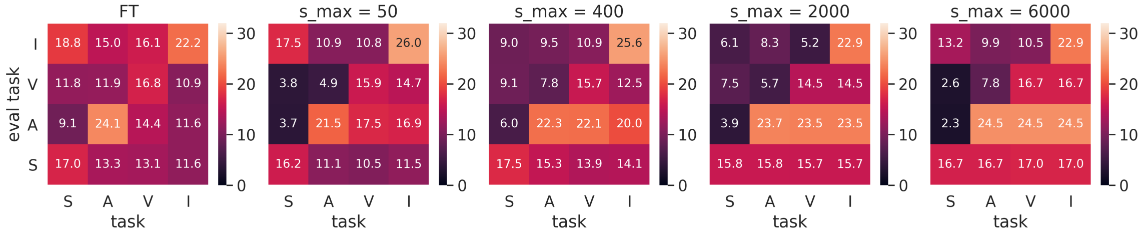


Figure 4.9: RATT ablation on Flickr30k validation set using different s_{max} values and finetuning baseline. Each heatmap reports BLEU-4 performance for one of the ablated models evaluated on different tasks at the end of the training of each task.

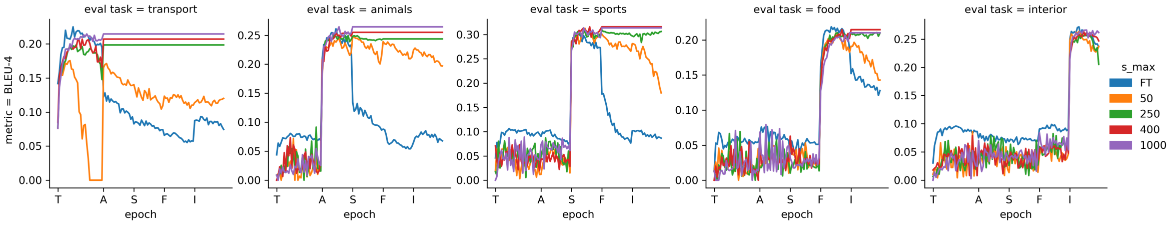


Figure 4.10: RATT ablation on the MS-COCO validation set using different s_{max} values and finetuning baseline. Each plot reports BLEU-4 performance evaluated on one of the tasks at different training epochs and different training tasks for each of the ablated models.

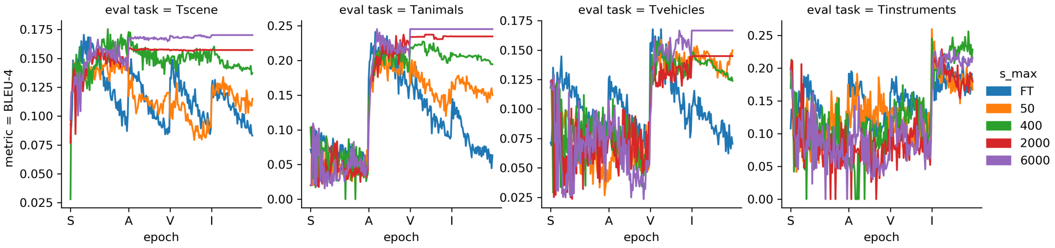


Figure 4.11: RATT ablation on Flickr30k validation set using different s_{max} values and finetuning baseline. Evaluation is the same as MS-COCO (figure 4.10).

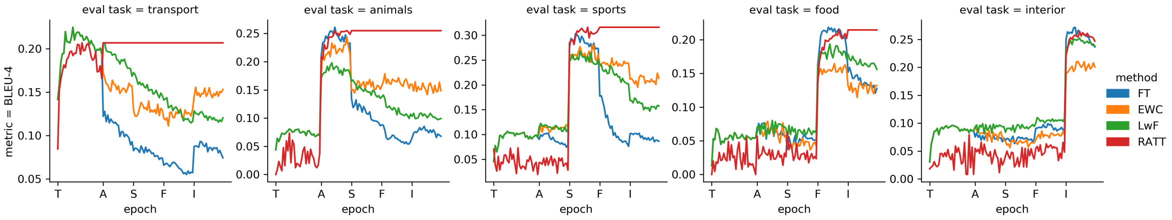


Figure 4.12: Comparison for all approaches on MS-COCO validation set. Each plot reports BLEU-4 performance evaluated on one of the tasks at different training epochs and different training tasks for each of the ablated models.

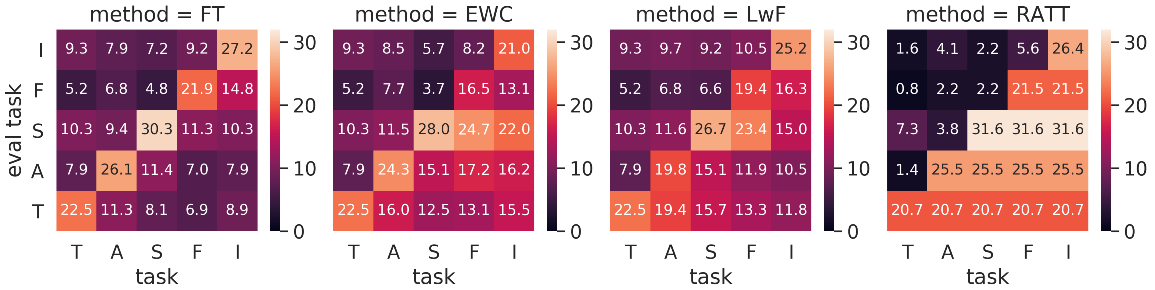


Figure 4.13: Comparison for all approaches on MS-COCO validation set. Each heatmap reports BLEU-4 performance for one of the models evaluated on different tasks at the end of the training of each task.

4.7 Additional experimental analysis

In this section we give additional comparative performance analysis for RATT, EWC, and LwF on both datasets.

4.7.1 Learning and forgetting on MS-COCO

In figures 4.12 and 4.13, we give a comparison of performance for all considered approaches on the MS-COCO validation set. These learning curves and heatmaps allow us to appreciate the ability of RATT to remember old tasks. The forgetting rate of EWC seems to be higher than the one of LwF, but EWC shows an ability to recover performance after noticeable forgetting – probably due to increased backward transfer. This is clear looking at figure 4.13 in which both LwF and EWC seems to suffer noticeable forgetting on the first two tasks (transport and animal) after training on the third one (Sport). EWC seems able to recover when trained on the next task, while LwF continues to forget more.

4.7.2 Learning and forgetting on Flickr30k

In figure 4.14 and 4.15 we give a comparison of performance for all approaches on the Flickr30k validation set. The first figure depicts the training process over all

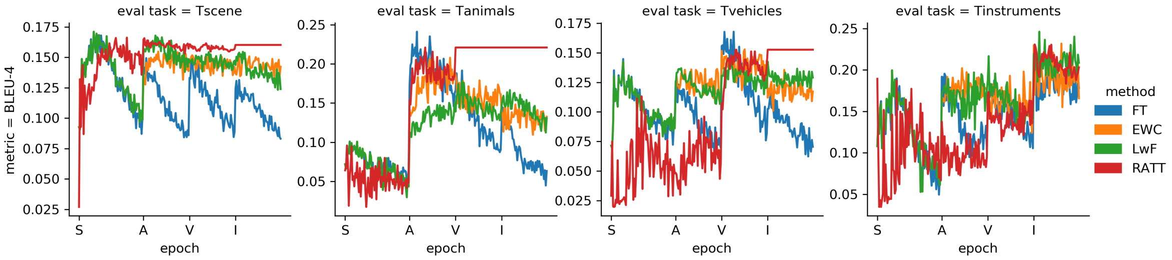


Figure 4.14: Comparison for all approaches on Flickr30k validation set. Each plot reports BLEU-4 performance evaluated on one of the tasks at different training epochs and different training tasks for each of the ablated models.

tasks, where the model is evaluated on each task while progressing through training. The results for Flickr30k show more variance than MS-COCO, as this setting is more challenging and the validation dataset is much smaller.

RATT exhibits almost no forgetting in comparison to other methods – an almost straight line after learning each task. Degradation of the FT model is visible, but for Flickr30k we notice that subsequent, more specific tasks keep previously learned and more generic concepts rather than completely forgetting (i.e. the first task category *scene*). The BLEU-4 score for LwF remains almost at the same level after learning the task, and EWC shows similar performance but with a bigger drop when going from task A to V. In figure 4.15 an evaluation summary is provided in form of BLEU-4 heatmaps. Going from the left (FT) to right (RATT), less forgetting can be observed by each of evaluated method, with RATT showing almost no loss in performance when reaching the final task.

It is useful to compare and contrast the results on Flickr30k and MS-COCO. In Flickr30k there is much more information shared between tasks and this is shown by the significant forward transfer that we see: after training on the first task (*scene*), the performance on the last task (*instrument*) is significant for all methods. Forward transfer is much less evident for RATT, and this is due to the fact that it uses the task embedding of future tasks for which it has no information (they all are randomly initialized). The backward transfer on Flickr30k is also evident looking at the relatively high performance of the FT baseline in figures 4.14 and 4.15 (and comparing with the MS-COCO equivalents in figures 4.12 and 4.13).

Although the overall performance on Flickr30k is much lower than on MS-COCO (evident when looking at the anti-diagonal of FT in figures 4.13 and 4.15), given the difficulty of the dataset itself and given the small number of examples (especially in validation/test sets) is difficult to draw firm conclusions about backward transfer for LwF and EWC.

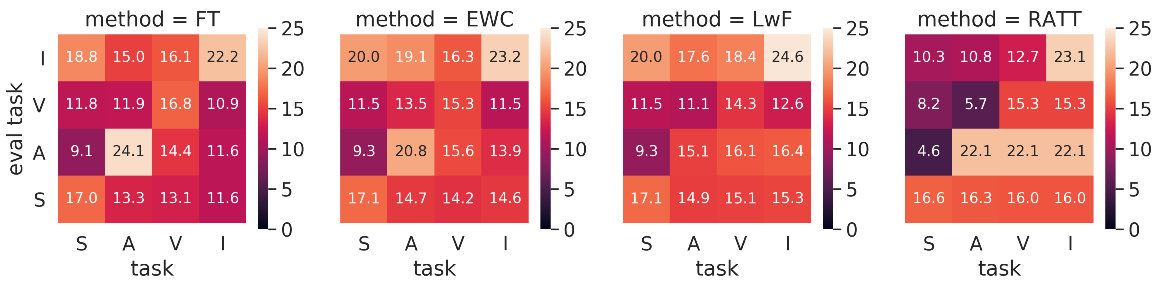


Figure 4.15: Comparison for all approaches on Flickr30k validation set. Each heatmap reports BLEU-4 performance for one of the models evaluated on different tasks at the end of the training of each task.

4.8 Additional captioning results

In figure 4.16 we give an example image from each of the first four MS-COCO tasks with the prediction made by the models after training on the correct task (on the left) and the one made after training on the complete sequence of tasks (on the right). Both EwC and LwF retain some correct words and "insight", but they are clearly confused by the last task on which they are trained. In the second image EWC predicts *zebras in a living room* because the last task contain house interiors. In a similar way, in the last picture EWC predicts the words *refirgerator* and *bed*, while LwF predicts *table*. In figure 4.17 we can see a similar analysis conducted on Flickr30k dataset. Again the quality of RATT captions is retained after training on the last task. In figure 4.18 we give two qualitative examples taken from the last task from the MS-COCO dataset for which fine-tuning provides better descriptions than RATT. In this case the baseline does not suffer from catastrophic forgetting because we evaluate the last trained task. RATT could be limited by the fact that neurons allocated to previous tasks are not trainable.

4.9 Conclusions

In this chapter we proposed a technique for continual learning of image captioning networks based on Recurrent Attention to Transient Tasks (RATT). Our approach is motivated by a feature of image captioning not shared with other continual learning problems: tasks are composed of *transient* classes (words) that can be shared across tasks. We also showed how to adapt Elastic Weight Consolidation and Learning without Forgetting, two representative approaches of continual learning, to the recurrent image captioning networks. We proposed task splits for the MS-COCO and Flickr30k image captioning datasets, and our experimental evaluation confirms the need of recurrent task attention in order to mitigate forgetting in continual learning with sequential, transient tasks. RATT is capable of zero forgetting at the expense of

Target a passenger bus that is driving down the street

	After training task 1 (Transport):		After training task 5 (Interior):
FT	a bus is stopped at a bus stop	FT	a street scene with focus on the wall
EWC	a bus is stopped at a bus stop	EWC	a double decker bus is on the street
LwF	a bus is stopped at a bus stop	LwF	a group of people standing next to each other
RATT	a bus is parked in front of a building	RATT	a bus is parked in front of a building

Target a number of zebras standing in the dirt near a wall

	After training task 2 (Animal):		After training task 5 (Interior):
FT	a group of zebras are standing in a field	FT	a woman in a black shirt is walking by a beach
EWC	a group of zebras standing in a dirt field	EWC	a group of zebras are standing in a living room
LwF	a group of zebras are standing in a field	LwF	a black and white photo of a group of people
RATT	a group of zebras are standing in the dirt	RATT	a group of zebras are standing in the dirt

Target a man is holding a surfboard and staring out into the ocean

	After training task 3 (Sport):		After training task 5 (Interior):
FT	a man carrying a surfboard on top of a beach	FT	a woman in a black shirt is walking by a beach
EWC	a man carrying a surfboard on top of a beach	EWC	a man riding a surfboard on a beach
LwF	a man carrying a surfboard on top of a beach	LwF	a woman walking down a beach with a umbrella
RATT	a man holding a surfboard in the ocean	RATT	a man holding a surfboard in the ocean

Target a woman sells cupcakes with fancy decorations on them

	After training task 4 (Food):		After training task 5 (Interior):
FT	a woman is standing in front of a table full of food	FT	a woman is holding a glass of wine at a restaurant
EWC	a man is holding a banana in a kitchen	EWC	a woman is holding a white refrigerator in a bed
LwF	a woman standing in front of a store with a large crowd of people	LwF	a woman standing in front of a table with a large pot of food
RATT	a woman standing in front of a store filled with cakes	RATT	a woman standing in front of a store filled with cakes

Figure 4.16: Captioning results for all methods on MS-COCO. Images and target captions belong to a specific task and captions are generated by all techniques after training the correct task (left) and a later task (right). Approaches except RATT contextualize to some degree generated captions with respect to the most recently learned task.

Targets "the brown dog is running on the grass" - "brown dog is running in a field"

	After training task 1 (Scene):		After training task 4 (Instruments):
FT	a group of people are walking through a snowy mountain	FT	a man in a black shirt and a man in a <unk> <unk> ...
EWC	a group of people are walking through a snowy mountain	EWC	a group of people are standing on a <unk> in the snow
LwF	a group of people are walking through a snowy mountain	LwF	a group of people are standing in front of a <unk>
RATT	a group of people are walking down a dirt road	RATT	a group of people are walking down a dirt road

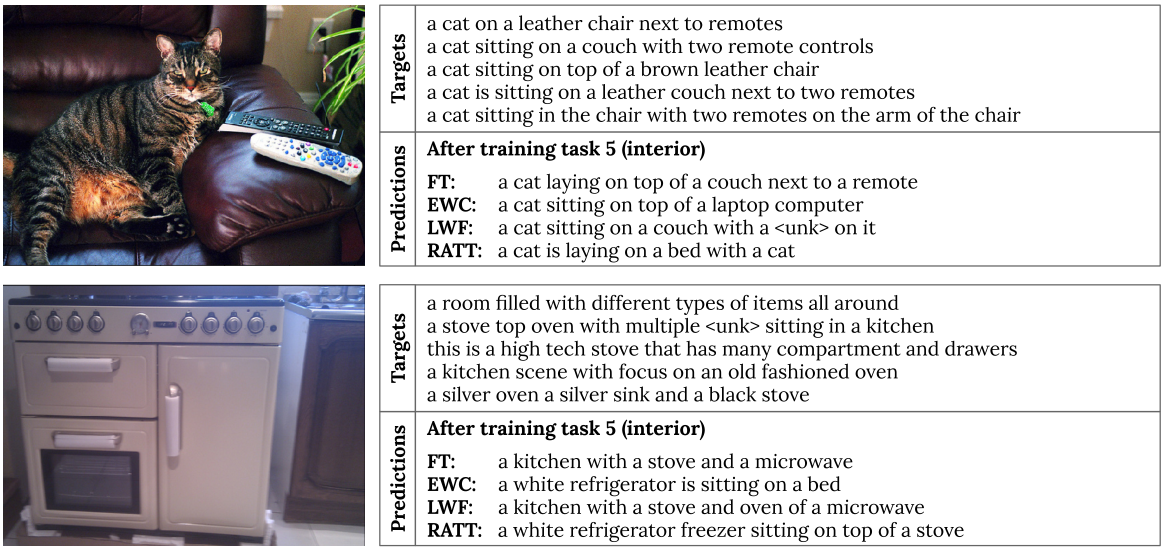
Targets "a group of people walk through the desert" - "a group of 5 people are walking toward the mountains"

	After training task 2 (Animals):		After training task 4 (Instruments):
FT	a brown dog is running through a field	FT	a dog is playing a frisbee in the grass
EWC	a brown dog is running through the grass	EWC	a black dog is playing in the grass
LwF	a man in a blue shirt is running in the grass	LwF	a man playing with a dog
RATT	a brown dog is running through a grassy field	RATT	a brown dog runs through the grass

Targets "a race car speeding on the track" - "red and white car rounds a corner on a racetrack"

	After training task 3 (Vehicles):		After training task 4 (Instruments):
FT	a race car is driving down a road	FT	a <unk> <unk> <unk> <unk> <unk> <unk> <unk> ...
EWC	a man in a red shirt is riding a bike	EWC	a man in a red shirt is riding a motorcycle
LwF	a man in a red shirt is riding a motorcycle on a street	LwF	a man in a red shirt is playing a <unk>
RATT	a car is racing on a track	RATT	a car is racing on a track

Figure 4.17: Captioning results on Flickr30k. Images and target captions belong to a specific task and captions are generated by all techniques after training the correct task (left) and a later task (right).



Targets		a cat on a leather chair next to remotes a cat sitting on a couch with two remote controls a cat sitting on top of a brown leather chair a cat is sitting on a leather couch next to two remotes a cat sitting in the chair with two remotes on the arm of the chair
Predictions	**After training task 5 (interior)**	
	FT:	a cat laying on top of a couch next to a remote
	EWC:	a cat sitting on top of a laptop computer
	LWF:	a cat sitting on a couch with a <unk> on it
	RATT:	a cat is laying on a bed with a cat

Targets		a room filled with different types of items all around a stove top oven with multiple <unk> sitting in a kitchen this is a high tech stove that has many compartment and drawers a kitchen scene with focus on an old fashioned oven a silver oven a silver sink and a black stove
Predictions	**After training task 5 (interior)**	
	FT:	a kitchen with a stove and a microwave
	EWC:	a white refrigerator is sitting on a bed
	LWF:	a kitchen with a stove and oven of a microwave
	RATT:	a white refrigerator freezer sitting on top of a stove

Figure 4.18: Examples of images from MS-COCO dataset for which fine-tuning achieves better results than the proposed method. These images are taken from the last task, so there is no catastrophic interference.

plasticity and backward transfer: the ability to adapt to new tasks is limited by the number of free neurons and it is difficult to exploit knowledge from future tasks to better predict older ones. The focus of this work is on how a simple encoder-decoder image captioning model forget, which limits the quality of captions when comparing with current state-of-the-art. As future work, we are interested in applying the developed method in more complex captioning systems.

Chapter 5

Conclusions and Future Work

The goal of this thesis was to improve some aspects of deep network learning in order to be more similar to human learning. This was not done to merely mimic the nature of human learning, but rather to improve some aspects in which humans learning exhibits specific advantages over traditional techniques for training deep neural networks. In this chapter we summarize the approaches proposed in this thesis and we propose future research directions.

5.1 Conclusions

In this thesis we investigated problematic aspects of deep network learning: weak supervision and forgetting. Despite the fact that correctly trained models can today achieve performance surpassing humans in some tasks, they are still far from the human learning ability and flexibility: they require many more examples and supervision to be correctly trained, they must be trained to perform precise and specific tasks, and when trained incrementally on sequential tasks they suffer from *catastrophic forgetting*, which limits their use to predetermined tasks that must be chosen in advance during the training phase and cannot be easily expanded after.

In chapter 2 we explored the supervision problem, applying deep networks to the challenging task of instance recognition in a domain for which is very difficult to collect large amounts of annotated data. We showed how the absence of data and supervision can be mitigated by the use of web search engines to collect multimodal datasets composed of images, text and metadata. The drawback of this is the introduction of noise in image labels, which can hurt the performance of learned classifiers. We then proposed in section 2.3 several techniques to mitigate the impact of noisy labels in webly-supervised data:

- **Entropy scaling**, which performs soft outlier detection and decreases the impact of both labelflip noise and outliers during training.

- **Gradual bootstraping**, which explicitly makes use of the most reliable data sources to create a solid knowledge foundation for the model so that soft outlier detection via entropy scaling is more reliable.

We then proposed to use L_2 normalization instead of ReLU non linearity after the last embedding layer, which forces all the features input to the classifier to lie on a hypersphere with the aim of mitigating domain shift between training and test sets. Domain shift is very noticeable in this type of dataset since training images comes from search engines which will reflect their biases into the retrieved examples.

Finally, in section 2.4 we studied how to exploit textual information associated with classes in order to overcome the total absence of visual examples for some classes. To this end, we applied zero-shot techniques to the artwork instance recognition problem. We proposed a non-linear compatibility technique that maps visual instances into the document embedding space and uses the cosine similarity to compute compatibility between points. The proposed model uses a multi-task learning paradigm, having an extra classifier trained to discriminate points in the document embedding space into the original categories of the dataset. We showed how the web can be exploited to retrieve multi-modal data to train a zero-shot recognition model, and we explicitly showed that the use of a large number of webly-supervised training classes can dramatically boost zero-shot recognition performance on a small number of test classes, almost doubling the zero-shot classification performance in our case.

In chapter 3 we again studied the issue of weak supervision of deep network training, but in this case for the problem of image quality assessment with limited training data. We chose a different strategy to overcome the absence of data and the high cost of labeling process: we trained a generative model inspired by AC-GAN to artificially increase the size of the training-set, generating new synthetically distorted examples given the original high quality images, the desired class of distortions, and the desired quality factors. The discriminator of the GAN itself is used to evaluate the quality factor of the input images. We showed that the use of generative data augmentation directly increased the performance of the model. The performance obtained are comparable with the state-of-the-art despite the fact that the proposed model is relatively small and thus also very efficient.

In chapter 4 we looked at catastrophic forgetting in recurrent network architectures on incremental, transient tasks using image captioning context as our application. We proposed two different splitting procedures which adapt existing captioning datasets (having at least object detection or multi-class labels) to the continual learning framework. We proposed two baselines for continual image captioning based on Elastic Weight Consolidation and Learning without Forgetting. Both baseline approaches are able to partially overcome the catastrophic forgetting problem when compared to a fine-tuning baseline, but continue to suffer from forgetting whenever new tasks arrive. We proposed a novel technique that is able to perfectly retain captioning performance when trained on sequential tasks. This is possible thanks to how specific weights are selectively learned during each task training phase so that the network still exploits most of the network weights during the forward pass.

5.2 Future work

Continual learning still requires research and improvements. While there is an increasing interest in the field, we are still far from being able to apply continual learning in real-world scenarios, and much work is needed to realize solutions to catastrophic interference that are practical, flexible, and without compromises. Even if regularization techniques try to model a sort of biologically-inspired neural plasticity so that neural networks approach human brain behavior, such solutions do not seem to help so much in recurrent networks. Weight allocation techniques seems to be the most practical tool to prevent forgetting if we allow slight increases in the number of weights when network capacity is exhausted and new tasks arrive, but then we are still bound in the task-aware context of continual learning, which limits applicability in some scenarios.

To extend the technique proposed in chapter 4 to *task-agnostic* continual image captioning, we must eliminate the need to provide the activation mask during inference. A possibility would be to force neurons allocated to the current task to not interfere with the forward pass of previous task examples. Weights should thus be learned so that they minimize the activation for inputs of the previous tasks. Another possibility would be to predict the task at inference time given the test image itself, but we would then have to incrementally learn a classifier to do so. Moreover a prediction error from the task classifier would catastrophically propagate to the captioning model.

We are also interested in extending the data augmentation work presented in chapter 3, that achieved very promising results on no-reference Image Quality Assessment. While an optimization of the architecture and an increasing depth could be a strategy to increase performance, we are more interested in modifications that are better able to model the nature of the problem and fix intrinsic problems of the proposed method. One example comes from the following observation: while most of the distortions are very sparse in the whole image and more noticeable at a small scale (like jpeg compression, white noise, or gaussian blur), others are well-localized but very evident even at coarse scale (like non eccentricity pattern noise or local block-wise distortion from the TID2013 dataset). Moreover, these localized distortion classes seem to be the ones for which the proposed model have the biggest gap in performance compared to the state-of-the-art. For these localized distortions, the use of small random patches taken from the image to evaluate the distortion level could be misleading since most of the patches would not be not distorted, while the whole image clearly is when observed globally. To address this issue one could process two different versions of the same image at each step: the resized version and a randomly cropped version, potentially using dedicated processing paths for the first layers of the network, and providing a mechanism at inference time to decide which version we should rely for the image quality prediction.

Appendix A

Publications

Peer-reviewed Journal Papers

1. **R. Del Chiaro**, A. D. Bagdanov, and A. Del Bimbo, "Webly-supervised zero-shot learning for artwork instance recognition." *Pattern Recognition Letters*, 2019.

Peer-reviewed Conference Papers

1. **R. Del Chiaro**, B. Twardowski, A. D. Bagdanov, and J. van de Weijer, "RATT: Recurrent Attention to Transient Tasks for continual image captioning." Advances in Neural Information Processing Systems (NeurIPS), 2020.

2. **R. Del Chiaro**, A. D. Bagdanov, and A. Del Bimbo, "Noisyart: A dataset for webly-supervised artwork recognition." In Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISAPP), 2019.

3. P. Bongini, **R. Del Chiaro**, A. D. Bagdanov, and A. Del Bimbo, "GADA: Generative Adversarial Data Augmentation for Image Quality Assessment." Proceedings of the International Conference on Image Analysis and Processing (ICIAP), 2019.

Book Chapters

1. **R. Del Chiaro**, A. D. Bagdanov, and A. Del Bimbo, "NoisyArt: exploiting the noisy web for zero-shot classification and artwork instance recognition." In: *Data Analytics for Cultural Heritage: Current Trends and Concepts*. Eds: A. Belhi, A. Bouras, A. Al-Ali and A. Hamid Sadka. Springer, 2021.

Bibliography

Akata, Z., Perronnin, F., Harchaoui, Z., and Schmid, C. (2015a). Label-embedding for image classification. *IEEE transactions on pattern analysis and machine intelligence*, 38(7):1425–1438.

Akata, Z., Reed, S., Walter, D., Lee, H., and Schiele, B. (2015b). Evaluation of output embeddings for fine-grained image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2927–2936.

Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., and Tuytelaars, T. (2018). Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154.

Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., and Zhang, L. (2018). Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086.

Baraldi, L., Paci, F., Serra, G., Benini, L., and Cucchiara, R. (2015). Gesture recognition using wearable vision sensors to enhance visitors' museum experiences. *IEEE Sens. J*, 15(5):2705–2714.

Barandela, R. and Gasca, E. (2000). Decontamination of training samples for supervised pattern recognition methods. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 621–630. Springer.

Bianco, S., Celona, L., Napoletano, P., and Schettini, R. (2016). On the use of deep learning for blind image quality assessment. *arXiv preprint arXiv:1602.05531*.

Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., and Hellmann, S. (2009). Dbpedia-a crystallization point for the web of data. *Web Semantics: science, services and agents on the world wide web*, 7(3):154–165.

Bosse, S., Maniry, D., Wiegand, T., and Samek, W. (2016). A deep neural network for image quality assessment. In *Proceedings of ICIP*, pages 3773–3777. IEEE.

Brodley, C. E. and Friedl, M. A. (1999). Identifying mislabeled training data. *Journal of artificial intelligence research*, 11:131–167.

Bucher, M., Herbin, S., and Jurie, F. (2016). Improving semantic embedding consistency by metric learning for zero-shot classiffication. In *European Conference on Computer Vision*, pages 730–746. Springer.

Castro, F. M., Marín-Jiménez, M. J., Guil, N., Schmid, C., and Alahari, K. (2018). End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 233–248.

Chatfield, K., Simonyan, K., Vedaldi, A., and Zisserman, A. (2014). Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*.

Chaudhry, A., Dokania, P. K., Ajanthan, T., and Torr, P. H. (2018). Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547.

Chen, X. and Gupta, A. (2015). Webly supervised learning of convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1431–1439.

Chen, X. and Lawrence Zitnick, C. (2015). Mind's eye: A recurrent visual representation for image caption generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2422–2431.

Chetouani, A., Beghdadi, A., Chen, S., and Mostafaoui, G. (2010). A novel free reference image quality metric using neural network approach. In *Proc. Int. Workshop Video Process. Qual. Metrics Cons. Electrn*, pages 1–4.

Coop, R. and Arel, I. (2013). Mitigation of catastrophic forgetting in recurrent neural networks using a fixed expansion layer. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7, Dallas, TX, USA. IEEE.

Cornia, M., Stefanini, M., Baraldi, L., and Cucchiara, R. (2020). Meshed-memory transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10578–10587.

Cucchiara, R., Grana, C., Borghesani, D., Agosti, M., and Bagdanov, A. D. (2012). Multimedia for cultural heritage: Key issues. In *Multimedia for Cultural Heritage*, pages 206–216. Springer.

De Lange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. (2019). Continual learning: A comparative study on how to defy forgetting in classification tasks. *arXiv preprint arXiv:1909.08383*.

Del Chiaro, R., Bagdanov, A. D., and Del Bimbo, A. (2019a). Noisyart: A dataset for webly-supervised artwork recognition. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 4: VISAPP,*, pages 467–475. INSTICC, SciTePress.

Del Chiaro, R., Bagdanov, A. D., and Del Bimbo, A. (2019b). Webly-supervised zero-shot learning for artwork instance recognition. *Pattern Recognition Letters*, 128:420–426.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634.

Fouhey, D. F., Gupta, A., and Zisserman, A. (2016). 3d shape attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1516–1524.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Mikolov, T., et al. (2013). Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129.

Gallace, A. and Spence, C. (2009). The cognitive and neural correlates of tactile memory. *Psychological bulletin*, 135(3):380.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., and Bengio, Y. (2013). An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Herdade, S., Kappeler, A., Boakye, K., and Soares, J. (2019). Image captioning: Transforming objects into words. In *Advances in Neural Information Processing Systems*, pages 11137–11147.

Hinton, G., Vinyals, O., and Dean, J. (2015). Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hossain, M. Z., Sohel, F., Shiratuddin, M. F., and Laga, H. (2019). A comprehensive survey of deep learning for image captioning. *ACM Comput. Surv.*, 51(6).

Huang, L., Wang, W., Chen, J., and Wei, X.-Y. (2019). Attention on attention for image captioning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4634–4643.

Hussein, N., Gavves, E., and Smeulders, A. W. (2017). Unified embedding and metric learning for zero-exemplar event detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1096–1105.

Hutmacher, F. (2019). Why is there so much more research on vision than on any other sensory modality? *Frontiers in psychology*, 10:2246.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kang, L., Ye, P., Li, Y., and Doermann, D. (2014). Convolutional neural networks for no-reference image quality assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1733–1740.

Kang, L., Ye, P., Li, Y., and Doermann, D. (2015). Simultaneous estimation of image quality and distortion via multi-task convolutional neural networks. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 2791–2795. IEEE.

Karaman, S., Bagdanov, A. D., Landucci, L., D'Amico, G., Ferracani, A., Pezzatini, D., and Del Bimbo, A. (2016). Personalized multimedia content delivery on an interactive table by passive observation of museum visitors. *Multimedia Tools and Applications*, 75(7):3787–3811.

Katsaggelos, A. K. (2012). *Digital image restoration*. Springer Publishing Company, Incorporated.

Kim, J. and Lee, S. (2017). Fully deep blind image quality predictor. *IEEE Journal of selected topics in signal processing*, 11(1):206–220.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, page 1097–1105, Red Hook, NY, USA. Curran Associates Inc.

Le, Q. and Mikolov, T. (2014). Distributed representations of sentences and documents. In *International Conference on Machine Learning*, pages 1188–1196.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Li, S., Tao, Z., Li, K., and Fu, Y. (2019). Visual to text: Survey of image and video captioning. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 3(4):297–312.

Li, Y., Crandall, D. J., and Huttenlocher, D. P. (2009). Landmark classification in large-scale image collections. In *Computer vision, 2009 IEEE 12th international conference on*, pages 1957–1964. IEEE.

Li, Z. and Hoiem, D. (2017). Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947.

Lin, K.-Y. and Wang, G. (2018). Hallucinated-iqa: No-reference image quality assessment via adversarial learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 732–741.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer.

Liu, X., van de Weijer, J., and Bagdanov, A. D. (2017). Rankiqa: Learning from rankings for no-reference image quality assessment. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1040–1049.

Liu, X., Wu, C., Menta, M., Herranz, L., Raducanu, B., Bagdanov, A. D., Jui, S., and van de Weijer, J. (2020). Generative feature replay for class-incremental learning.

In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 226–227.

Lopez-Paz, D. and Ranzato, M. (2017). Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476.

Lu, J., Xiong, C., Parikh, D., and Socher, R. (2017). Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 375–383.

Mallya, A., Davis, D., and Lazebnik, S. (2018). Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Mao, J., Xu, W., Yang, Y., Wang, J., and Yuille, A. L. (2015). Deep captioning with multimodal recurrent neural networks (m-rnn). In Bengio, Y. and LeCun, Y., editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Masana, M., Tuytelaars, T., and van de Weijer, J. (2020). Ternary feature masks: continual learning without any forgetting. *arXiv preprint arXiv:2001.08714*.

McCloskey, M. and Cohen, N. J. (1989). Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier.

Mendes, P. N., Jakob, M., García-Silva, A., and Bizer, C. (2011). Dbpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th international conference on semantic systems*, pages 1–8. ACM.

Mensink, T. and Van Gemert, J. (2014). The rijksmuseum challenge: Museum-centered visual recognition. In *Proceedings of International Conference on Multimedia Retrieval*, page 451. ACM.

Mittal, A., Moorthy, A. K., and Bovik, A. C. (2012). No-reference image quality assessment in the spatial domain. *Image Processing, IEEE Transactions on*, 21(12):4695–4708.

Moorthy, A. K. and Bovik, A. C. (2011). Blind image quality assessment: From natural scene statistics to perceptual quality. *IEEE Transactions on Image Processing*, 20(12):3350–3364.

Nickolls, J., Buck, I., Garland, M., and Skadron, K. (2008). Scalable parallel programming with cuda. *Queue*, 6(2):40–53.

Odena, A., Olah, C., and Shlens, J. (2017). Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 2642–2651. JMLR. org.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. (2019). Continual lifelong learning with neural networks: A review. *Neural Networks*.

Pedersoli, M., Lucas, T., Schmid, C., and Verbeek, J. (2017). Areas of attention for image captioning. In *Proceedings of the IEEE international conference on computer vision*, pages 1242–1250.

Pfülb, B. and Gepperth, A. (2019). A comprehensive, application-oriented study of catastrophic forgetting in dnns. In *ICLR*.

Plummer, B. A., Wang, L., Cervantes, C. M., Caicedo, J. C., Hockenmaier, J., and Lazebnik, S. (2017). Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. *IJCV*, 123(1):74–93.

Ponomarenko, N., Ieremeiev, O., Lukin, V., Egiazarian, K., Jin, L., Astola, J., Vozel, B., Chehdi, K., Carli, M., Battisti, F., et al. (2013). Color image database tid2013: Peculiarities and preliminary results. In *Visual Information Processing (EUVIP), 2013 4th European Workshop on*, pages 106–111. IEEE.

Raguram, R., Wu, C., Frahm, J.-M., and Lazebnik, S. (2011). Modeling and recognition of landmark image collections using iconic scene graphs. *International journal of computer vision*, 95(3):213–239.

Ragusa, F., Furnari, A., Battiato, S., Signorello, G., and Farinella, G. (2019a). Egocentric Point of Interest Recognition in Cultural Sites. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, pages 381–392.

Ragusa, F., Furnari, A., Battiato, S., Signorello, G., and Farinella, G. M. (2019b). Egocentric visitors localization in cultural sites. *Journal on Computing and Cultural Heritage (JOCCH)*, 12(2):11.

Ranjan, R., Castillo, C. D., and Chellappa, R. (2017). L2-constrained softmax loss for discriminative face verification. *arXiv preprint arXiv:1703.09507*.

Ratcliff, R. (1990). Connectionist models of recognition memory: constraints imposed by learning and forgetting functions. *Psychological review*, 97(2):285.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. (2017). icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010.

Romera-Paredes, B. and Torr, P. (2015). An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, pages 2152–2161.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015a). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015b). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. (2016). Progressive neural networks. *arXiv preprint arXiv:1606.04671*.

Saad, M. A., Bovik, A. C., and Charrier, C. (2012). Blind image quality assessment: A natural scene statistics approach in the dct domain. *Image Processing, IEEE Transactions on*, 21(8):3339–3352.

Schak, M. and Gepperth, A. (2019). A study on catastrophic forgetting in deep lstm networks. In Tetko, I. V., Kůrková, V., Karpov, P., and Theis, F., editors, *Artificial Neural Networks and Machine Learning – ICANN 2019: Deep Learning*, Lecture Notes in Computer Science, pages 714–728, Cham. Springer International Publishing.

Schwarz, J., Luketina, J., Czarnecki, W. M., Grabska-Barwinska, A., Teh, Y. W., Pascanu, R., and Hadsell, R. (2018). Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning (ICML)*.

Serra, J., Suris, D., Miron, M., and Karatzoglou, A. (2018). Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning (ICML)*.

Sharma, S., El Asri, L., Schulz, H., and Zumer, J. (2017). Relevance of unsupervised metrics in task-oriented dialogue for evaluating natural language generation. *CoRR*, abs/1706.09799.

Sheikh, H. (2005). Live image quality assessment database release 2. *http://live. ece. utexas. edu/research/quality*.

Sheikh, H. R., Sabir, M. F., and Bovik, A. C. (2006). A statistical evaluation of recent full reference image quality assessment algorithms. *Image Processing, IEEE Transactions on*, 15(11):3440–3451.

Sheikh, H. R., Wang, Z., Cormack, L., and Bovik, A. C. (2020). Live image quality assessment database. `http://live.ece.utexas.edu/research/quality`.

Shin, H., Lee, J. K., Kim, J., and Kim, J. (2017). Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2990–2999.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Socher, R., Ganjoo, M., Manning, C. D., and Ng, A. (2013). Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943.

Sodhani, S., Chandar, S., and Bengio, Y. (2019). Toward training recurrent neural networks for lifelong learning. *Neural Computation*, 32:1–34.

Sukhbaatar, S. and Fergus, R. (2014). Learning from noisy labels with deep neural networks. *CoRR*, abs/1406.2080.

Temmermans, F., Jansen, B., Deklerck, R., Schelkens, P., and Cornelis, J. (2011). The mobile museum guide: artwork recognition with eigenpaintings and surf. In *Proceedings of the 12th International Workshop on Image Analysis for Multimedia Interactive Services*.

Valtysson, B. (2012). Europeana: The digital construction of europe's collective memory. *Information, Communication & Society*, 15(2):151–170.

Van Ouwerkerk, J. (2006). Image super-resolution survey. *Image and Vision Computing*, 24(10):1039–1052.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2015). Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Vinyals, O., Toshev, A., Bengio, S., and Erhan, D. (2017). Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):652–663.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. J. (1989). Phoneme recognition using time-delay neural networks. *IEEE transactions on acoustics, speech, and signal processing*, 37(3):328–339.

Wang, Z., Bovik, A. C., and Lu, L. (2002). Why is image quality assessment so difficult? In *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*, volume 4, pages IV–3313. IEEE.

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., and Perona, P. (2010). Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology.

Westlake, N., Cai, H., and Hall, P. (2016). Detecting people in artwork with cnns. In *Computer Vision – ECCV 2016 Workshops*, pages 825–841.

Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Wu, C., Herranz, L., Liu, X., Wang, Y., van de Weijer, J., and Raducanu, B. (2018). Memory replay GANs: learning to generate images from new categories without forgetting. In *Advances in Neural Information Processing Systems*.

Xian, Y., Lampert, C. H., Schiele, B., and Akata, Z. (2018). Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly. *IEEE transactions on pattern analysis and machine intelligence*.

Xian, Y., Schiele, B., and Akata, Z. (2017). Zero-shot learning - the good, the bad and the ugly. In *IEEE Computer Vision and Pattern Recognition (CVPR)*.

Xu, J., Ye, P., Li, Q., Du, H., Liu, Y., and Doermann, D. (2016). Blind image quality assessment based on high order statistics aggregation. *IEEE Transactions on Image Processing*, 25(9):4444–4457.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057.

Yan, B., Bare, B., and Tan, W. (2019). Naturalness-aware deep no-reference image quality assessment. *IEEE Transactions on Multimedia*, PP:1–1.

Yan, J., Lin, S., Kang, S. B., and Tang, X. (2014). A learning-to-rank approach for image color enhancement. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2987–2994. IEEE.

Yang, Z., Yuan, Y., Wu, Y., Cohen, W. W., and Salakhutdinov, R. R. (2016). Review networks for caption generation. In Lee, D. D., Sugiyama, M., Luxburg, U. V., Guyon, I., and Garnett, R., editors, *Advances in Neural Information Processing Systems 29*, pages 2361–2369. Curran Associates, Inc.

Ye, P. and Doermann, D. (2012). No-reference image quality assessment using visual codebooks. *Image Processing, IEEE Transactions on*, 21(7):3129–3138.

Ye, P., Kumar, J., Kang, L., and Doermann, D. (2012). Unsupervised feature learning framework for no-reference image quality assessment. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1098–1105. IEEE.

Zeng, H., Zhang, L., and Bovik, A. C. (2017). A probabilistic quality representation approach to deep blind image quality prediction.

Zenke, F., Poole, B., and Ganguli, S. (2017). Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3987–3995. JMLR. org.

Zhang, P., Zhou, W., Wu, L., and Li, H. (2015). Som: Semantic obviousness metric for image quality assessment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2394–2402.

Zhou, Z.-H. (2018). A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53.