future internet

Article

Tell Me More: Automating Emojis Classification for Better Accessibility and Emotional Context Recognition †

Muhammad Atif [1,*] and Valentina Franzoni [2,*]

[1] Department of Mathematics and Computer Science, University of Florence, Viale Morgagni 67/a, 50134 Florence, Italy
[2] Department of Mathematics and Computer Science, University of Perugia, Via Vanvitelli 1, 06123 Perugia, Italy
* Correspondence: muhammad.atif@unifi.it (M.A.); valentina.franzoni@dmi.unipg.it (V.F.)

† This paper is an extended version of our paper published in the 14th International Conference on Brain Informatics, Virtual Event, 17–19 September 2021; pp. 146–156; Muhammad Atif, Valentina Franzoni, Alfredo Milani: Emojis Pictogram Classification for Semantic Recognition of Emotional.

Abstract: Users of web or chat social networks typically use emojis (e.g., smilies, memes, hearts) to convey in their textual interactions the emotions underlying the context of the communication, aiming for better interpretability, especially for short polysemous phrases. Semantic-based context recognition tools, employed in any chat or social network, can directly comprehend text-based emoticons (i.e., emojis created from a combination of symbols and characters) and translate them into audio information (e.g., text-to-speech readers for individuals with vision impairment). On the other hand, for a comprehensive understanding of the semantic context, image-based emojis require image-recognition algorithms. This study aims to explore and compare different classification methods for pictograms, applied to emojis collected from Internet sources. Each emoji is labeled according to the basic Ekman model of six emotional states. The first step involves extraction of emoji features through convolutional neural networks, which are then used to train conventional supervised machine learning classifiers for purposes of comparison. The second experimental step broadens the comparison to deep learning networks. The results reveal that both the conventional and deep learning classification approaches accomplish the goal effectively, with deep transfer learning exhibiting a highly satisfactory performance, as expected.

Keywords: deep learning; affective computing; emotion recognition; machine learning; context information; artificial intelligence; sentic computing; meme; emoticon; image classification

check for updates

Citation: Atif, M.; Franzoni, V. Tell Me More: Automating Emojis Classification for Better Accessibility and Emotional Context Recognition. *Future Internet* **2022**, *14*, 142. https://doi.org/10.3390/fi14050142

Academic Editors: Haoran Xie, Gary Cheng and Fu-lee Wang

Received: 24 March 2022
Accepted: 29 April 2022
Published: 5 May 2022

1. Introduction

Since its inception, SMS messaging has stimulated the need for additional visual information to help define the context of short messages. While users convey their messages using different facial expressions, in textual communication people relish adding emotional clues through emoji to compensate for limited or unavailable facial expression. Moreover, individuals appreciate adding reactions as feedback to live video communication (e.g., live streaming, video calls). Pictures are among the most straightforward clues for eliciting emotional and empathetic communication through communication media. Thus, when images are not made readable for all users, the resulting accessibility failure introduces a critical bias against individuals with a visual impairment. Originally, the issue was simply resolved using *emoticons*, i.e., pictograms mimicking facial expressions or objects encoded by standard sequences of characters. However, as smartphone technology evolved, software text systems began directly replacing emoticons with related pictures. Such pictograms were visible only to the user but still encoded with characters for the software, which could employ its own set of pictograms to visualize the emoticon. Thus, the same text could be augmented with different pictograms based on the underlying software used to read it. This encoding strategy allowed algorithms to understand the picture and provide its meaning to automated systems (e.g., for sentiment analysis) or aiding software (e.g., vocal readers for people with vision impairment).

Recently, the demand for more complex and descriptive images has introduced in every messaging system and social network the possibility to include images instead of emoticons, usually coded as Graphics Interchange Format (GIF), called *emojis*. These pictograms express emotions, represent objects, or refer to standardized meanings with memes. In most cases, emojis show the same appearance for graphical emoticons, but differences in encoding do not allow for an automated recognition of context analysis, emotion recognition, and readers for people with vision impairment. In fact, besides being encoded as images, they also do not include any alternative text which could enable the image content to be read aloud. This situation sets assistive and accessibility systems back a few decades.

On the other hand, such a situation presents a novel research problem, unprecedented in the existing literature, where sentiment analysis is often performed to identify positive or negative polarity of emojis [1–3], while emotion recognition is neglected.

Therefore, identifying an automated process that can recognize the content of emojis is paramount in restoring accessibility for ethical applications.

2. Related Works

In this work, we expand the conference paper published in Brain Informatics 2021: *Muhammad Atif, Valentina Franzoni, Alfredo Milani: Emojis Pictogram Classification for Semantic Recognition of Emotional Context. BI 2021: 146–156.* Ref. [4] to investigate and benchmark the application of affective classification to emoji pictures using conventional supervised machine learning approaches and deep learning techniques. The previous work is left as a reference for deepening theoretical methods and techniques. The novelty of this work resides in the application of classical machine learning and recent advances in deep learning for image recognition to the new domain of emotions in emojis.

Deep learning has previously been investigated for emoji classification with different aims in Natural Language processing, such as sentiment analysis or translation of offensive sexual meanings but never before, to our knowledge, for emotion recognition of emojis in the context of text accessibility and understandability [5].

Our method does not differ from image classification, but applies and compares the known techniques for image classification to emoji classification from text, where the challenge is to recognise emotions from emojis, in order to provide a valid tool for accessibility.

From the conventional machine learning techniques, we exploit k-nearest neighbors (K-NN) classifiers [6], Support Vector Machine (SVM) [7], TreeBagger [8], Decision Tree [9], Boosting algorithms [10], Random Forest [8], and Linear Discriminant Analysis (LDA) [11]. In deep learning designs, a significant volume of training samples are required. To address this challenge, we employed transfer learning methods that rely on previous general training on images, able to identify the main elements (e.g., lines, edges, color distribution, shapes) [12,13]. As deep learning shows improved results when a sufficient amount of training items is provided, this study also exploits deep learning pre-trained classification models for AlexNet [12,14], GoogleNet [15], SqueezeNet [16], MobileNetV2 [17] and InceptionV3 [18]. We selected these networks due to the availability of a pre-trained version on which to exploit transfer learning. Our observational findings indicate that deep-learning classifiers with transfer learning perform satisfactorily compared to conventional machine learning classifiers on a restricted number of samples, and balanced classes.

The majority of research conducted on emotion recognition, based on genuine facial expressions and speech, use a restricted set of emotions [19–22]. The most widely used and simplest for universal emotion recognition is the Ekman model of six basic emotion classes (i.e., fear, anger, joy, sadness, disgust, and surprise) [23]. One of the main values offered by this model, and the reason it is the most widely used, is that it has been studied worldwide, and proven to be cross-cultural since its facial features are recognized with the same expression without any geographic bias. Furthermore, including a relatively small set of classes, it can guarantee sufficient inter-class variability for classification. The main drawbacks of the Ekman model are two-fold: on one hand it guarantees sufficient inter-class variability, on the other hand, it is prone to some error biases versus some specific classes. For instance, a neutral expression can easily be misclassified as sad, but these types of errors also occur in human-based recognition and are thus intrinsic in any dataset. More complex models such as Plutchick's [20] may better represent facial emotion expressions since it accounts also for the valence of each emotion, which results in weighted and additional information for data analysis. In addition to these considerations, the Ekman model of six basic emotions has provided the inspiration for most of the features behind emotional tagging of text (e.g., view Facebook reactions or emojis sets from any chat-based software), and is well-known for the movie "Inside out". Moreover, the diffusion of this model in popular knowledge supports a worldwide knowledge base for consistent labeling of textual data with emojis, despite any anagraphical data, e.g., gender, age, and culture. Concerning facial recognition, the classification of realistic facial imagery [24] was also performed with consistently high accuracy in Ekman's six basic emotions for micro-expression-based categorization [25–27]. Regarding text classification, the semantic breakdown of text posts in social media networks has been well-investigated [19,20,28], using semantic terminology. However, there is no such automated system that focuses on the emotion recognition of emoji pictograms, the use of which is still recent. Our work aims at filling this gap (see also the previous work [4]).

3. Materials and Methods for Emojis Classification

This section briefly describes dataset acquisition and preprocessing, deep feature extraction, brief introduction of transfer learning, conventional supervised machine learning classifiers and deep models with their own different parameters. The dataset will be available on IEEE Data Port under the name "Emojis Classification for Better Accessibility and Emotional Context Recognition".

3.1. Dataset Collection and Preprocessing

The data have been collected over six emotional classes representing the basic model of emotions by Ekman [23]. Our dataset has been collected over the web, focusing on the sets of emojis used by the most famous chat apps and augmenting them. The emojis have been labeled using Google Search labels, searching for the term "emoji" and the term defining each emotion of the emotional mode, e.g., "emoji AND happy". With this research strategy, we obtained several different sets of emojis for each emotional expression. The collection of emojis has been therefore balanced with data augmentation techniques of image transformation (i.e., rotation, translation, shear, and reflection). The final dataset is split for training and testing at an 80%–20% rate.

The dataset includes a total of:

Emojis: 4680 images over six classes, i.e., 780 images per class;

Training: 624 training images for each class;

Test: 156 test images for each class.

Please see Figure 1 for a visual sample of images from the dataset.

Initially, the pictograms are preprocessed to filter out text or noisy items. Then, data augmentation is applied to balance the classes by incrementing the number of items in the dataset for the summoned classes using transformation techniques. The images are then scaled as per the models' input, i.e., [227 227 3], [224 224 3], and [299 299 3] pixels for AlexNet, GoogleNet, and InceptionV3 Convolutional Neural Networks (CNN), respectively. The dataset is split into training and test sets. Attributes are retrieved through AlexNet and ResNet-18 pre-trained CNNs, provided as input to conventional machine learners for training and testing. Each conventional classifier is taught individually on deep features using its own parameter setting (see Section 3.3). Employing transfer learning, the last fully-connected layers of the pre-trained deep model are fine-tuned on the emotion classes.

3.2. Experiments Workflow

Figure 1 shows the framework for image-based emoji classification. Initially, the dataset of emoji images collected from the Internet is preprocessed, and resized to fit the input of the deep models. Then, deep learning is employed for both of our classification approaches: classical algorithms, where deep learning (DL) is used to extract the features, and transfer learning. In the first approach, the Alexnet and ResNet-18 deep models are used to extract features from the dataset, chosen among others because of their wide use in the literature for this purpose [29–32]. Such DL-extracted features from the two models are then independently provided as input features to train and test each traditional supervised classifier, as shown in the upper part of Figure 1. The lower part of Figure 1 displays the re-training for knowledge transfer of the pre-trained deep classifiers using transfer learning, fine-tuning emojis in the last three fully connected layers of the models pre-trained with DL. Different parameters are applied to train the traditional and deep classifiers according to their parameter settings (see Section 3.4). These two approaches are compared to classify emojis pictograms into the Ekman model of six basic emotion classes.

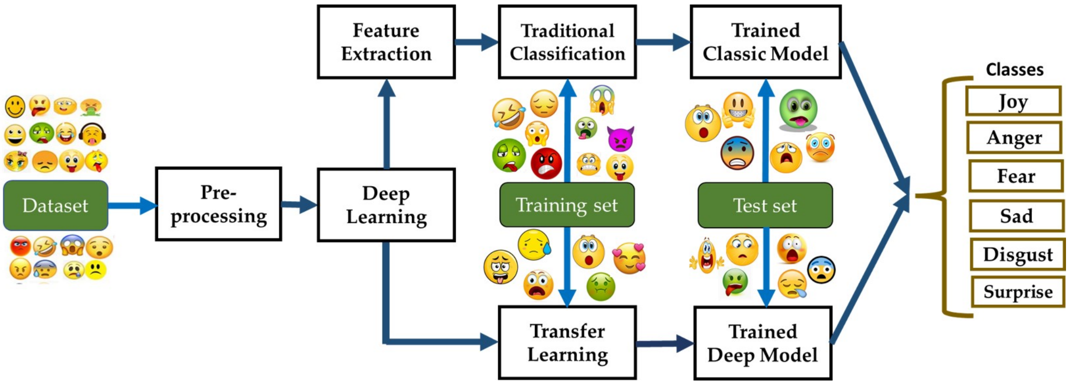


Figure 1. A framework based on traditional and deep-learning based classifiers for the automatic recognition of image-based emoticons through Convolutional Neural Networks (CNN).

3.3. Transfer Learning

Deep-model training from scratch requires high computational power and an excessive number of training examples, while knowledge transfer (i.e., transfer learning) uses a neural network that has been pre-trained with large datasets of general images, and fine-tunes the last fully connected layers on the work-specific dataset of images. Being already capable of recognizing the low-level features of images, e.g., color distribution, shapes, edges, and corners [33], the last fully connected layers of CNN is fine-tuned to our dataset of emojis for emotional class recognition. This method is efficient in classifying images and emotions since it can apply the basic knowledge acquired by CNN on images, i.e., the ability to recognize low-level features, to build a new model for a new problem.

3.4. Deep Features Extraction

Training and testing deep classifiers require adequate computational power and a large number of training samples. To optimize this phase, we can employ the deep features extracted through pre-trained deep models. Higher layers of deep models provide low-level features, with a long length of feature descriptors, while deeper layers provide higher-level features with smaller feature descriptors that can be easily processed. Extracted deep features are provided as input to train and test traditional machine learning classifiers. Fixed-size feature descriptors are extracted [29], with the two pre-trained deep models, AlexNet [14] and ResNet-18 [34]. The length of the feature descriptors, i.e., vectors extracted from the deep models of AlexNet and ResNet-18, measure 1 × 4096 and 1 × 512 for each emoji pictogram, respectively. These features are extracted to train and test traditional machine learning classifiers.

3.5. Experiments Settings

This section provides a brief overview of the conventional and deep classifiers and their parameter setup.

3.5.1. Conventional Supervised Classifiers

We briefly discuss traditional supervised machine learning (ML) classifiers, i.e., conventional ML algorithms used before the current deep-learning hype, that will be trained using Alexnet and ResNet-18 with their parameter setup.

K Nearest Neighbors (K-NN) *[6]* finds the k data points that are closest to a given pattern data point. The number of k neighbors is set experimentally by the developer. For each sample of test data, the algorithm assigns membership to each emotion class depending on the value of k, i.e., how many nearest neighbors vote for a given class. We exploited the experiments of K-NN with different values of k, i.e., odd values between 1 and 15, to reduce the probability of a tie.

Support Vector Machine (SVM) *[7]* is designed for binary classification, but can be adopted for multi-class classification. In classifying the data points (i.e., our emoji examples), the goal is to identify a partition of the input space using hyperplanes as decision boundaries. Support vectors are a subset of the training samples that calculate the location of the separation hyperplane. In multi-class classification, the problem is partitioned into a set of binary classification problems. In this study, we used different variants of SVM such as linear SVM, Radial Basis Function (RBF) kernel, and polynomial kernel.

Decision Tree (DS) *[9]* is a method for approximating discrete-valued functions that recursively divides the data into subgroups. DS learns a heuristic, non-backtracking search through the space of all possible decision trees. Such a tree-like structure supports the prediction of the decisions in the tree from the root node down to a leaf node, where the leaf node contains the response. A pruning algorithm is used to avoid overfitting.

Linear Discriminant Analysis Classifier (LDA) *[11]* effectively isolates categories based on the linear combination of features. In multi-class classification, the Fisher discriminant is used to find a subspace that constrains class inconsistency. In this study, we employed different discriminant types, i.e., 'linear', 'diagLinear', 'diagQuadratic', 'pseudoLinear' and 'pseudoQuadratic'. Linear estimates a covariance matrix for all classes, while quadratic estimates a covariance matrix for each class.

Boosting is an ensemble learning technique that combines several weak learners into one strong learner to overcome training errors. Similarly, ensemble learning refers to a group of base learners working together to achieve a better prediction. In this study, we used different variants of boosting algorithms, i.e., AdaboostM2 [10], TotalBoost [35], and LPBoost [35]. For each boosting algorithm, we used a forest of 100 classification trees.

Random Forest (RF) *[8]* is a classification algorithm comprised of many decision trees. It uses bagging and random features in the creation of each tree to create an uncorrelated forest of trees. Forest prediction by committee is more accurate than a single tree. In our experiments, we used a forest of 100 classification trees and combined the results using bagging techniques (see the *Tree Bagger*).

Tree Bagger *[8]* grows ensemble decision trees using bootstrap samples of the data. Tree Bagger selects a random subset of predictors to use in each decision split similar to a random forest. Bootstrap-aggregated decision trees combine the results of many decision trees, reducing the effects of overfitting and improving generalization. We used 100 trees while training the tree bagger.

3.5.2. Deep Classifiers

We trained three deep classifiers using transfer learning to emoji classification. To such an extent, this section provides an overview of the deep classifiers and the training setup information with different parameters.

***AlexNet** [14]* is a pre-trained Convolutional Neural Network (CNN) trained with the ImageNet [36] dataset of 1000 object categories. This CNN consists of eight layers, of which the first five are convolutional layers and the last three are fully connected layers. AlexNet requires an input image of [227 227 3]. The output of the last layer, i.e., the softmax layer, produces a distribution over the given categories. AlexNet uses Rectified Linear Units (ReLU).
***Inceptionv3** [18]* is a 48-layer deep CNN architecture that assists in image analysis and object recognition. It is pre-trained on the ImageNet [36] dataset requiring an input image of size [299 299 3]. It employs factorized convolutions, as it reduces the number of parameters involved in network training. A 3 × 3 convolutional layer is replaced by a 1 × 3 convolution followed by a 3 × 1 convolution. It is structured in three main blocks: the basic convolution block, the inception, and the classification block.
***Googlenet** [15]* is a pre-trained 22-layer deep CNN architecture requiring an input image of size [224 224 3]. It is comprised of nine inception modules and contains two max-pooling layers between some inception modules to downsample the input. A dropout layer before the linear layer reduces eventual overfitting.
***MobileNetV2** [17]* is a CNN architecture based on inverted residual structure. It is a pre-trained model on the ImageNet [36] dataset requiring an input image of size [224 224 3]. It uses inverted residual blocks with bottlenecking features to bear a lower parameter count and optimize performance on mobile devices.
***SqueezeNet** [16]* architecture consists of *squeeze* and *expand* layers. A squeeze convolutional layer has only a 1 × 1 filter. Data are fed into an expand layer, which contains a mixture of 1 × 1 and 3 × 3 convolutional filters. Between all the squeeze and expand layers, the ReLU activation function is applied, and dropout layers are added to reduce overfitting. The network is already trained on the ImageNet [36] dataset requiring an input image of size [227 227 3].

Deep Classifiers adopted for emoji recognition using transfer learning can be found in research by [37]. To fine-tune deep neural networks for emoji recognition, we used three different training optimizers, i.e., Adaptive Moment Estimation (adam)[38], Stochastic Gradient Descent with Momentum (sgdm) [33], and Root Mean Square Propagation (rmsprop) [39]. Adam is an extension of stochastic gradient descent that has a small memory footprint and requires only first-order gradients, while sgdm uses stochastic gradient descent with momentum, i.e., a moving average of the gradients is utilized to update the weights. Finally, rmsprop uses an adaptive learning rate rather than specifying it as a hyperparameter. The learning rate is an important hyper-parameter that controls how quickly weights are updated in response to estimated errors, therefore controlling both the time and resources required to train a neural network. Finding an optimal learning rate is usually a tricky and time-consuming task: excessively large learning rates can lead to fast but unstable training and a small value usually results in a long training period and can even become stuck before completing correctly. We trained each deep model by varying the learning rate {0.01, 0.001, 0.0001, 0.00001} with a batch size of *32*.

4. Results and Discussion

This section compares and discusses the performance of deep and traditional classifiers based on accuracy as commonly used performance metrics. Figure 2 shows the comparison of the top three best performing deep learning and conventional supervised classifiers trained on features extracted with AlexNet and ResNet-18, separately. Deep learning classification outperforms traditional machine learning, with InceptionV3 achieving the highest performance. GoogleNet and MobileNetV2 achieve approximately the same accuracy, i.e., 98.40% and 98.61%, respectively. Among the conventional supervised classifiers, K-NN and SVM share the same best performance of about 95%, similar to LDA in second place with 92%.

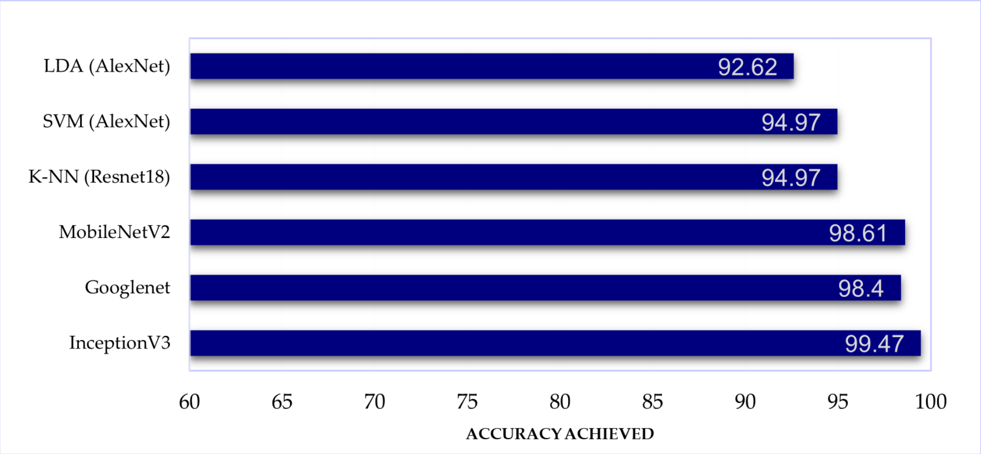
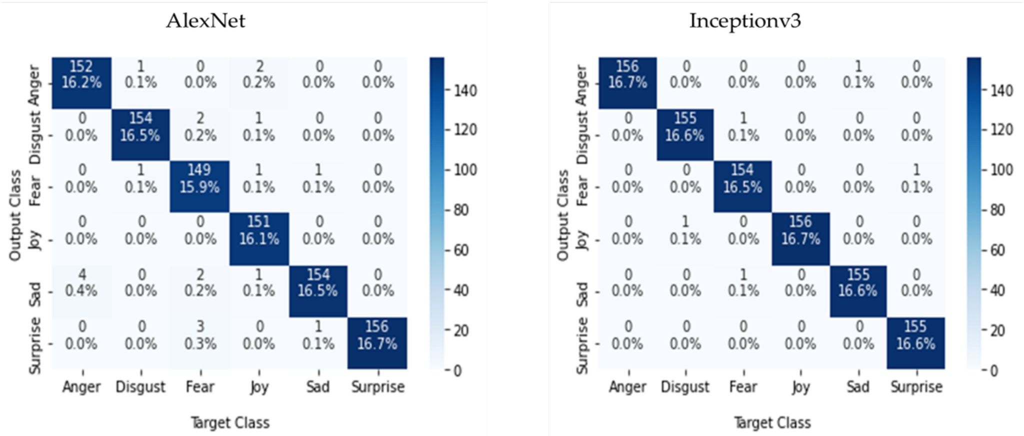


Figure 2. Comparison of three best performing deep and conventional supervised classifiers (in parentheses the networks used for feature extraction).

The performance analysis of InceptionV3 and AlexNet can be committed using the confusion matrices shown in Figure 3a,b to show overall performance and performance by class. InceptionV3 achieved an overall highest accuracy of 99.47% among five tested deep classifiers, while AlexNet achieved 97.86%. InceptionV3 perfectly classified all images in the Joy and Anger classes, while AlexNet misclassified several images. In addition to the confusion matrix of the best-performing network, it is interesting to observe the confusion matrix of the deep model with low performance.



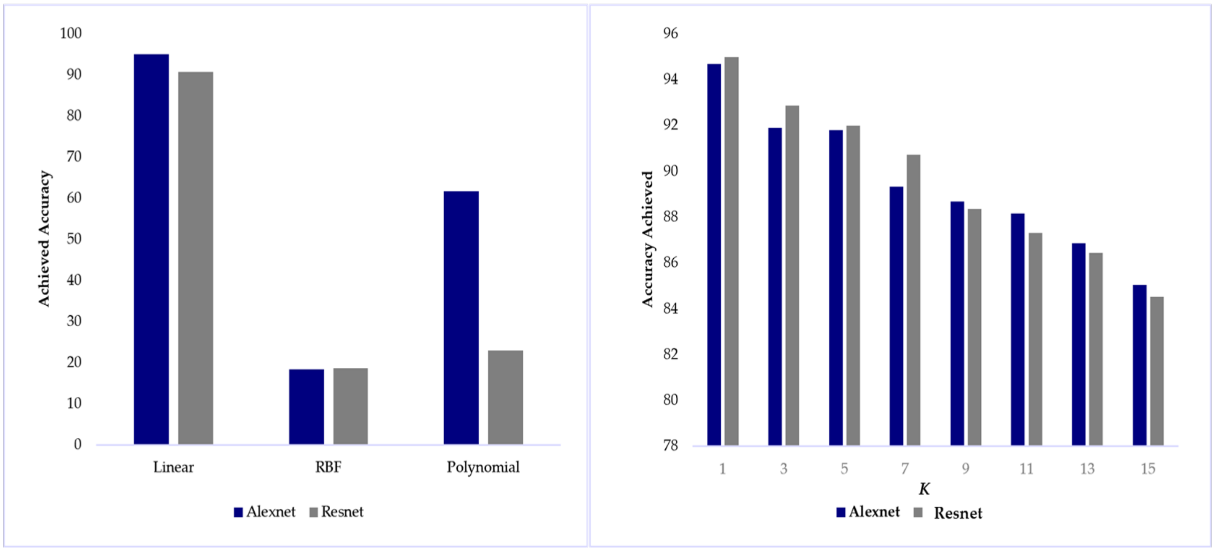
(**a**) (**b**)

Figure 3. Confusion Matrices of AlexNet and InceptionV3 deep models (**a**) on the left, confusion matrix of AlexNet achieved an accuracy 97.86% (**b**) on the right, confusion matrix of best performing deep model (InceptionV3) achieved 99.47% accuracy for emoji classification.

Indeed, from the confusion matrix, we can see which classes were misclassified into other classes, and whether these classes share a common element that can motivate the errors, or if the error is obvious when training the network. In this case, we can observe that AlexNet cannot recognize the emoji pictogram features as easily as other networks. If some errors depend on the examples, such as the emotion Angry being mistaken for Sadness or Disgust, where the associated expressions share a downward direction of the lips, or in other cases, such as Joy being mistaken for Sadness, Fear and Angry, then an error has occurred in training the network. The final result is highly accurate, but the individual errors are more critical than those of InceptionV3. In the latter, Joy, the emotional class that is also easier to detect in facial recognition, has no error. The errors are evident in the Fear class, which are mistaken for Sadness or Disgust since they share similar graphical features in emojis. Sadness is recognized once as anger, surprise once as fear, with the large open mouth being a common element. Only in the case of disgust, which is mistaken for joy, is a training issue evident. The results obtained with the InceptionV3 model are superior to the other tested classifiers in this study.

4.1. Performance of K-NN, SVM and LDA with Different Parameter Values

This subsection displays the experimental results obtained with the conventional supervised machine learning classifiers, K-NN and Support Vector Machines (SVM), for the different parameters described in Section 3.5.1. To train and test conventional classifiers, we used deep features extracted through AlexNet and ResNet-18 that were provided as input to the classifiers. The SVM classifier was trained using linear, Radial Basis Functions (RBF), and polynomial kernels. Figure 4a shows that the linear SVM performs better than the RBF and polynomial kernels, i.e., with the AlexNet extracted features bearing an accuracy of 94.97%, while the SVM with RBF kernels achieves a low accuracy of less than 25%. As linear SVM achieves better accuracy, this implies the data are linearly separable. On the other hand, Figure 4b shows the accuracy obtained with K-NN classifiers for different odd values of k between 1 and 15. To train and test K-NN, we used features extracted with AlexNet and ResNet-18. The results show that for k = 1, we achieved the highest accuracy of 94.66% and 94.97% with features extracted from AlexNet and Resnet-18 deep models, respectively. With a higher k value, the classification performance deteriorates further. A possible reason behind the lower accuracy for a greater value of k, i.e., number of neighbors may represent the inter-similarity among different emotion classes emojis that ultimately misguide the classifier. K-NN achieves higher performance with features extracted by the ResNet-18 model. K-NN achieves the highest accuracy of 94.97% with k = 1 on the ResNet-18 feature descriptor, while linear SVM achieves the same high accuracy of 94.97% on the AlexNet feature descriptor. The results show that K-NN and SVM achieve the same highest accuracy (i.e., 94.97%) on ResNet-18 and AlexNet features, respectively.



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

Figure 4. SVM and K-NN Performance: (**a**) Shown on the left, the performance of SVM for different kernels while, (**b**) the right side shows the performance of K-NN for different values of *K* on ResNet-18 and Alexnet deep features.

LDA was trained using five different discriminants, i.e., 'linear', 'diagLinear', 'diagQuadratic', 'pseudoLinear' and 'pseudoQuadratic'. Experimental results in Figure 5 show achievement of the highest accuracy 92.62% by LDA with discriminant of type pseudoLinear, and Alexnet features. Through ResNet-18 features, we achieve the highest accuracy 91.55% with the pseudoQuadratic discriminant. On the other hand, discriminant types 'diagQuadratic' and 'diagLinear' performed poorest, both on Alexnet and ResNet-18 features.

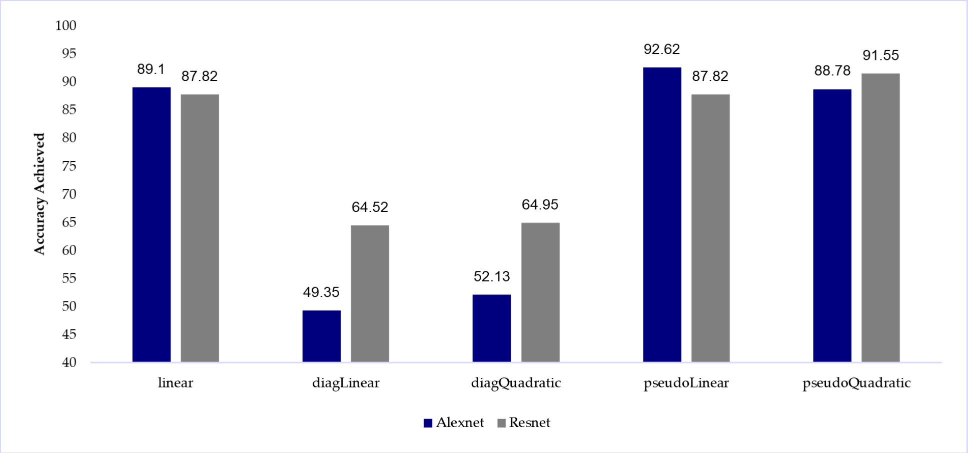


Figure 5. Performance comparison of LDA trained with different discriminants on AlexNet and ResNet-18 Deep features.

4.2. Overall Comparison of Conventional Supervised Machine Learning Classifiers on Deep Features

We conducted experiments with nine different conventional supervised classifiers trained on deep features, such as the features extracted by Alexnet and ResNet-18 models. Figure 6 shows that SVM and K-NN achieve the highest overall accuracy of 94.97% when using Alexnet and ResNet-18 features, respectively. An important observation is that 'LPBoost' and DS achieve an accuracy of less than 65.5%, while all other classifiers achieve an accuracy of over 76.80%. Among the conventional supervised classifiers, K-NN and SVM perform the best, while the second-highest accuracy is achieved by LDA. Additionally, it is worth noting that K-NN achieves the highest accuracy of 94.97% with ResNet-18 features, where the feature vector size is 1 × 512; on the other hand, SVM achieves the same accuracy with Alexnet features, where the feature vector size is 1 × 4096. K-NN processes a smaller feature vector and therefore requires less memory and computing power. Overall, Alexnet features provide the highest accuracy compared to ResNet-18, except K-NN, where ResNet-18 features achieve the highest accuracy as shown in Figure 6.

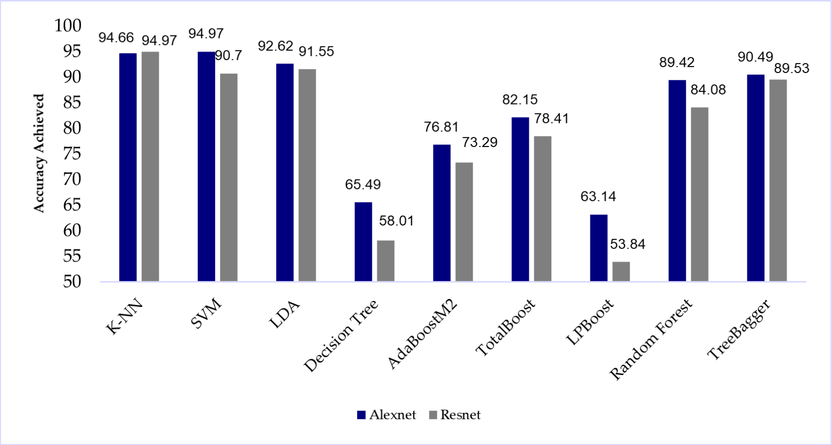


Figure 6. Performance comparison of conventional supervised machine learning classifiers on AlexNet and ResNet-18 Deep features.

4.3. Performance of Deep Classifiers Trained Using Transfer Learning

This section shows the results of the deep neural networks tested. GoogleNet, MobileNetV2, and InceptionV3 perform better than AlexNet and SqueezeNet. We achieved the highest accuracies 97.86%, 96.04%, 98.40%, 98.61% and 99.47% with AlexNet, SqueezeNet, MobileNetV2 GoogleNet and InceptionV3, respectively. The training of these neural networks is performed with three different training optimizers, namely adam, sgdm, and rmsprop. Table 1 shows the details of the experiments performed by varying training functions and learning rates to fine-tune the deep models. The loss for InceptionV3 is lower compared to other tested deep models. The highest accuracy (99.47%) is achieved by the InceptionV3 model with Adam optimizer and a learning rate of 0.0001, while GoogleNet with the training function rmsprop and a learning rate 0.0001 achieves the highest performance of 98.40%. On the other hand, MobileNetV2 perform slightly better compared to GoogleNet and achieved an accuracy of 98.61% with a learning rate of 0.001, and the sgdm optimizer. The possible reason for the highest accuracy achieved with InceptionV3 could be the number of layers in the model. InceptionV3 contains more layers than GoogleNet and AlexNet. AlexNet achieves the highest accuracy of 97.86% with the training function adam and a learning rate 0.00001. Another important observation is that for both AlexNet and InceptionV3, the highest accuracy is achieved with Adam, while GoogleNet and SqueezeNet achieve the highest accuracy with rmsprop. Only MobileNetV2 that achieved the second-highest accuracy with the sgdm optimizer. The learning rate is an important hyperparameter: In particular, AlexNet and SqueezeNet achieve the lowest accuracy 16.67% at a learning rate of 0.01, which is due to an unstable training process. If we decrease the value assigned to the learning rate, we achieve better performance as shown in Table 1. Overall, InceptionV3 performs the best, and SqueezeNet performs the worst among all five tested deep classifiers.

Table 1. Performance of deep classifier by varying the learning rate and optimizer. The bold numbers are to highlight the best results.

Classifier	Optimizer	Learning Rate			
		0.01	0.001	0.0001	0.00001
		Achieved Accuracy			
AlexNet	adam	16.67	16.67	94.55	**97.86**
	sgdm	16.67	16.67	96.69	95.51
	rmsprop	16.67	16.67	92.95	97.65
GoogleNet	adam	16.67	0.8472	98.29	96.69
	sgdm	16.67	0.9786	97.33	82.26
	rmsprop	16.67	16.67	**98.40**	97.54
InceptionV3	adam	82.37	95.51	**99.47**	93.91
	sgdm	98.4	98.61	93.91	75.53
	rmsprop	79.81	96.37	98.18	94.76
MobileNetV2	adam	54.8	97.43	97	94.76
	sgdm	97.54	**98.61**	95.72	70.72
	rmsprop	35.04	95.61	98.29	96.68
SqueezeNet	adam	16.67	30.66	90.81	83.76
	sgdm	16.67	90.17	89.2	65.38
	rmsprop	16.67	16.67	**96.04**	86.11

5. Conclusions

In this study, we performed systematic experiments to classify emojis into six classes of emotions based on the Ekman model. We ran the experiments using traditional supervised classifiers trained on deep features extracted using AlexNet and ResNet-18 pre-trained networks, and five pre-trained deep CNNs trained using transfer learning. The traditional classifiers K-NN and SVM achieved 94.97% accuracy using ResNet-18 and AlexNet features, respectively, while the decision tree and LPBoost achieved the lowest accuracies of 65.49% and 63.14%, respectively, using AlexNet features. The highest accuracy of 99.47% was achieved by the InceptionV3 model, while MobilNetV2 and GoogleNet performed better compared to the other two deep classifiers AlexNet, SqueezeNet, and traditional supervised classifiers.

Future works can operate multimodal approaches, e.g., merging our work with natural language processing techniques for deeper context analysis. Recurrent neural networks can be adopted for GIF-based emojis and memes over different platforms. However, there is currrently a scarcity of available datasets labeled with emotions. A critical point should thus be the creation of a sound dataset including emotions expressed by memes providing a sufficient variability of data to correctly classify the emotions of related pictograms.

Author Contributions: Idea and conceptualization, V.F.; methodology, V.F., M.A.; software, M.A.; validation, V.F., M.A.; formal analysis, V.F., M.A.; investigation, V.F, M.A.; resources, V.F., M.A.; data curation, V.F., M.A.; writing—original draft preparation, V.F., M.A.; writing—review and editing, V.F.; visualization, V.F., M.A.; supervision, project administration, and funding acquisition, V.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by EmoRe.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are not availablle to the public.

Acknowledgments: The authors thank Alfredo Milani from University of Perugia for their kind support and contribution to this project.

Conflicts of Interest: The authors declare no conflict of interest. This work added more than 50% of new content to the previous, and includes no copy-and-paste from any previous version. The content, contextualization, and discussion are changed. The images are changed. Only experimental results overlap the conference paper of which this paper is an extension.

Abbreviations

The following abbreviations are used in this manuscript:

SMS	Short Message Service
GIF	Graphic Interchange Format
K-NN	k-nearest neighbors
SVM	Support Vector Machine
LDA	Linear Discriminant Analysis
CNN	Convolutional Neural Networks
RBF	Radial Basis Functions
ReLU	Rectified Linear Units

References

1. Read, J. Using Emoticons to Reduce Dependency in Machine Learning Techniques for Sentiment Classification. In *Proceedings of the ACL Student Research Workshop*; Association for Computational Linguistics: Stroudsburg, PA, USA, 2005; pp. 43–48
2. Li, M.; Ch'ng, E.; Chong, A.Y.L.; See, S. Multi-class Twitter sentiment classification with emojis. *Ind. Manag. Data Syst.* **2018**, *118*, 1804–1820. [CrossRef]
3. Grover, V. Exploiting Emojis in Sentiment Analysis: A Survey. *J. Inst. Eng. India Ser. B* **2022**, *103*, 259–272. [CrossRef]
4. Atif, M.; Franzoni, V.; Milani, A. Emojis Pictogram Classification for Semantic Recognition of Emotional Context. In Proceedings of the 14th International Conference on Brain Informatics, Virtual Event, 17–19 September 2021; pp. 146–156.
5. Chandra, P.; Prasad, U. Classification of Emojis using Artificial Neural Network and Natural Language Processing. In Proceedings of the 2021 8th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 17–19 March 2021; pp. 205–212.
6. Guo, G.; Wang, H.; Bell, D.; Bi, Y.; Greer, K. KNN model-based approach in classification. In Proceedings of the OTM Confederated International Conferences CoopIS, DOA, and ODBASE 2003 Catania, Sicily, Italy, 3–7 November 2003; pp. 986–996.

7. Hsu, C.-W.; Lin, C.-J. A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Netw.* **2002**, *13*, 415–425. [PubMed]
8. Breiman, L. Bagging Predictors. *Mach. Learn.* **1996**, *26*, 123–140. [CrossRef]
9. Mitchell, T.M. *Machine Learning*; McGraw Hill: New York, NY, USA, 1997.
10. Freund, Y.; Schapire, R.E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.* **1997**, *55*, 119–139. [CrossRef]
11. Şener, B.; Çokluk-Bökeoğlu, Ö. Discriminant Function Analysis: Concept and Application. *Eurasian J. Educ. Res. (EJER)* **2008**, *33*, 73–92.
12. Gervasi, O.; Franzoni, V.; Riganelli, M.; Tasso, S. Automating facial emotion recognition. *Web Intell.* **2019**, *17*, 17–27. [CrossRef]
13. Carlini, N.; Wagner, D. Towards Evaluating the Robustness of Neural Networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy, San Jose, CA, USA, 22–24 May 2017; pp. 39–57. [CrossRef]
14. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2012; Volume 25.
15. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1–9.
16. Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5 MB model size. *arXiv* **2016**, arXiv:1602.07360.
17. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. Available online: https://openaccess.thecvf.com/content_cvpr_2018/papers/Sandler_MobileNetV2_Inverted_Residuals_CVPR_2018_paper.pdf (accessed on 15 March 2022).
18. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
19. Biondi, G.; Franzoni, V.; Poggioni, V. A deep learning semantic approach to emotion recognition using the IBM watson bluemix alchemy language. In *Computational Science and Its Applications—ICCSA 2017*; Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics); Springer International Publishing: Cham, Switzerland, 2017; pp. 719–729._51. [CrossRef]
20. Franzoni, V.; Milani, A.; Biondi, G. SEMO: A semantic model for emotion recognition in web objects In Proceedings of the 2017 IEEE/WIC/ACM International Conference on Web Intelligence, Leipzig, Germany, 23–26 August 2017; pp. 953–958. [CrossRef]
21. Nandwani, P.; Verma, R. A review on sentiment analysis and emotion detection from text. *Soc. Netw. Anal. Min.* **2021**, *11*, 81. [CrossRef] [PubMed]
22. Ferrara, E.; Yang, Z. Quantifying the effect of sentiment on information diffusion in social media. *PeerJ Comput. Sci.* **2015**, *1*, e26. [CrossRef]
23. Ekman, P. An argument for basic emotions. *Cogn. Emot.* **1992**, *6*, 169–200. [CrossRef]
24. Jain, D.K.; Shamsolmoali, P.; Sehdev, P. Extended deep neural network for facial emotion recognition. *Pattern Recognit. Lett.* **2019**, *120*, 69–74. [CrossRef]
25. Yan, J.; Zheng, W.; Xu, Q.; Lu, G.; Li, H.; Wang, B. Sparse kernel reduced-rank regression for bimodal emotion recognition from facial expression and speech. *IEEE Trans. Multimed.* **2016**, *18*, 1319–1329. [CrossRef]
26. Wollmer, M.; Metallinou, A.; Eyben, F.; Schuller, B.; Narayanan, S. Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional LSTM modeling. In Proceedings of the 11th Annual Conference of the International Speech Communication Association (INTERSPEECH 2010), Chiba, Japan, 26–30 September 2010; pp. 2362–2365.
27. He, G.; Liu, X.; Fan, F.; You, J. Image2audio: Facilitating semi-supervised audio emotion recognition with facial expression image. In Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 912–913.
28. Franzoni, V.; Milani, A. Semantic context extraction from collaborative networks. In Proceedings of the 2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2015, Calabria, Italy, 6–8 May 2015; pp. 131–136; doi: 10.1109/CSCWD.2015.7230946. [CrossRef]
29. Sahoo, J.; Prakash, S.A.; Patra, S.K. Hand gesture recognition using PCA based deep CNN reduced features and SVM classifier. In Proceedings of the 2019 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS), Rourkela, India, 16–18 December 2019; pp. 221–224.
30. Tarawneh, A.S.; Hassanat, A.B.; Chetverikov, D.; Lendak, I.; Verma, C. Invoice classification using deep features and machine learning techniques. In Proceedings of the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 9–11 April 2019; pp. 855–859.
31. Behera, S.K.; Rath, A.; Sethy, P.K. Fruit Recognition using Support Vector Machine based on Deep Features. *Karbala Int. J. Mod. Sci.* **2020**, *6*, 16. [CrossRef]
32. Mahbod, A.; Schaefer, G.; Wang, C.; Ecker, R.; Ellinge, I. Skin lesion classification using hybrid deep neural networks. In Proceedings of the ICASSP 2019—2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12-17 May 2019; pp. 1229–1233.

33. Liu, Y.; Yuan, G.; Wotao, Y. An improved analysis of stochastic gradient descent with momentum. In Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, BC, Canada, 6–12 December 2020.
34. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
35. Warmuth, M.; Liao, J.; Ratsch, G. Totally corrective boosting algorithms that maximize the margin. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 1001–1008.
36. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
37. Franzoni, V.; Biondi, G.; Perri, D.; Gervasi, O. Enhancing Mouth-Based Emotion Recognition Using Transfer Learning. *Sensors* **2020**, *20*, 5222. [CrossRef] [PubMed]
38. Lei Ba, J.; Kingma, D.P. Adam: A method for stochastic gradient descent. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1–15.
39. Dogo, E.M.; Afolabi, O.J.; Nwulu, N.I.; Twala, B.; Aigbavboa, C.O. A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks. In Proceedings of the 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS), Belgaum, India, 21–22 December 2018; pp. 92–99.