Emojis Pictogram Classification for Semantic Recognition of Emotional Context

Muhammad Atif[1], Valentina Franzoni[2(✉)], and Alfredo Milani[2]

[1] University of Florence, Florence, Italy
muhammad.atif@unifi.it
[2] University of Perugia, Perugia, Italy
{valentina.franzoni,milani}@dmi.unipg.it

Abstract. In online interactions, users frequently add emojis (e.g., smileys, hearts, angry faces) to text for expressing the emotions behind the communication context, aiming at a better interpretation to text especially of polysemous short expressions. Emotion recognition refers to the automated process of identifying and classifying human emotions. If text-based emoticons (i.e., emojis created by textual symbols and characters) can be directly understood by semantic-based context recognition tools used in the Web and Artificial Intelligence and robotics, image-based emojis need instead image recognition for a complete semantic context interpretation. This study aims to explore and compare systematically different classification models of emoticon pictograms collected from the Internet, with different labels according to the Ekman model of six basic emotions. A first comparison involves supervised machine learning classifiers trained on features extracted through neural networks. In the second phase, the comparison is extended to different deep learning models. Results indicate that deep learning models performed excellent, and traditional supervised algorithms also achieve very promising outcomes.

Keywords: Machine learning · Deep learning · Emotion recognition · Transfer learning · Emoticons

1 Introduction

The need to express emotional context to text message or to give emotional feedback, lead to the spread of *emojis* (i.e., image-based emoticons) and *memes* in web-based social interactions. While emoticons were initially codified by standard sequences of characters, the large variety of pictograms available on different platforms and devices, allow denoting a wide range of emotional nuances. Emotion recognition of image-based emoticons for context interpretation became,

This work is partially supported by the Italian Ministry of Research under PRIN Project "PHRAME" Grant n. 20178 XXKFY.

M. Mahmud et al. (Eds.): BI 2021, LNAI 12960, pp. 146–156, 2021.
https://doi.org/10.1007/978-3-030-86993-9_14

thus, a novel task for web-based semantics. For instance, in social networks, social media websites and applications, as well as in video-calling tools (e.g., Zoom, Webex, Teams, Meet) image-based emotional pictograms (e.g., emojis, GIFS, memes) are integrated to be used to communicate messages in an emotional context or to send emotional feedback, usually called *reaction*. If humans convey their messages by using different facial expressions, in textual communication over different social media platforms, people like to add emotional clues through emojis to convey the emotional meaning when the face visual is not available or limited, and to add reactions as feedback to live video communication (e.g., live streaming, video calls). Images are among the most immediate clues to arouse emotional communication and empathy through media.

Most studies conducted regarding emotion recognition, based on real facial expressions, and speech, use a limited set of emotions. The most used and simple for a universal emotion recognition without cultural or geographical biases is the Ekman model of emotions with its six basic emotions categories (i.e., fear, anger, joy, sadness, disgust, and surprise) [8]. Classification of real facial images [9] has been done with high precision into the six basic emotion of the Ekman categorization, based on micro-expressions [10–12]. Also, the semantic analysis of textual messages in social networks is well studied, using term semantics or textual emoticons. To the best of our knowledge, there is no automatic system focused on the recognition of emotions from emojis pictograms, which hype of use is still recent.

We study and compare the application to pictogram emojis of emotional classification by traditional supervised machine learning techniques and deep learning approaches. From traditional machine learning techniques, we leverage k-nearest neighbors (K-NN) [1], Support Vector Machine (SVM) [2], Decision Tree [3] and Linear Discriminant Analysis (LDA) [4] classifiers. In deep learning approaches, to solve the problem of the extremely high number of samples required for training we have used transfer learning techniques based on pre-trained classification models for AlexNet [5,18], GoogleNet [14] and InceptionV3 [6]. Experimental results indicate that deep-learning classifiers with transfer learning perform better compared to traditional machine learning classifiers on a limited number of samples and balanced classes. The rest of the paper is organized as follows. Section 2 reports the research methodology and data set used for conducting this study. Section 3 describes deep learning models, traditional supervised classifiers, and feature descriptors used in this study. Section 4 presents the experimental results followed by the discussion. Finally, Sect. 5 concludes this study and outlines some future directions.

2 Classification Methodology

In this study, two approaches have been compared for classifying emojis pictograms into the six basic emotional categories of the Ekman model. Pretrained deep models, i.e., *AlexNet*, *InceptionV3*, and *GoogleNet*, have been used, then applying a fine-tuning (i.e., a re-training) phase using transfer learning, which specializes the training on the six categories of emotions. In addition,

traditional machine learning classifiers, i.e., *k-Nearest Neighbors (k-NN)*, *Support Vector Machines (SVM)*, *Decision Tree*, and *Linear Discriminant Analysis (LDA)* are also trained on the deep features extracted through the AlexNet and Resnet18 [13,18] neural networks (NN) [7].

2.1 Feature Extraction for Supervised Machine Learning

Supervised machine learning requires features and labels for classification. In our experiments using traditional supervised algorithms, the label is provided by the image emotion class, while the classification features are extracted from the image training set using deep neural networks. In particular, we used the AlexNet and ResNet18 Convolutional Neural Networks (CNNs) for feature extraction [7]. CNNs are a specialized type of Deep Neural Network, able to reduce the information explosion: in the convolutional layers, the image information is filtered, generating a feature map. The number of final feature maps will be equal to the amount of filters used in the convolutional layers. In the fully connected layers following the pooling layer which samples the size of each feature map to reduce the computation, the filtered information is converted to a feature vector which can be given as output after a weighting phase. Such a final weighted vector is extracted and fed to machine learning algorithms. In particular, AlexNet features are extracted from the *fc7* fully-connected layer and Resnet18 features from the *pool5* global pooling layer.

2.2 Knowledge Transfer for Deep Learning

Deep learning models can be trained from scratch, requiring high computational power and a large number of training samples. On the other hand, using Knowledge Transfer (i.e., Transfer Learning) a neural network pre-trained on large data sets of general images is used because already capable of recognizing the low-level features of images, e.g. color distribution, shapes, edges, and corners [17]. The neural network is then fine-tuned with additional fully connected layers according to our data set of emojis, to recognize the emotional categories. This method is proved efficient on image and emotion classification, using the knowledge acquired by the NN on images, i.e., the abilities to recognize low-level features, as the foundation to create a new model for a new problem.

2.3 Emojis Pictogram Classification Framework

Figure 1 shows the structure of our emojis pictogram recognition framework. The figure shows two blocks representing the flow of the training (left) and testing (right) phases. The top part shows the training and testing process of traditional machine learning classifiers; the bottom part shows how deep learning models are trained and tested. After feature extraction, the images go through the preprocessing phase (i.e., cleaning from text and frames), then data augmentation techniques are used to have more samples and to recognize emojis that are not perfectly even. Machine learning or deep learning using knowledge transfer are used for classification.

3 Experimental Setup

3.1 Data Set Collection and Balancing

The data set used in this work has been built by authors collecting emojis pictograms from different social media and devices. Emotion terms related to the Ekman model were used to search the Web for the images, with a focus on selecting visualizations from different software and devices, which may show with different facets the images related to the same emoticon. Image-based emoticons have been added to images related to different visualizations of text-based emoticons. The labels have been assigned to images based on the Ekman model, using the same emotional words used in the web-based search phase. Data augmentation techniques i.e., rotation, translation, shear, and reflection, allowed to diversify the samples and balance classes. The balanced number of samples for each class in the data set is 624 training images and 156 test images for each class, for a total of 4680 images, split for training and testing at an 80%–20% rate.

3.2 Preprocessing of Input Images and Experimental Setup

Initially, images are preprocessed to filter out textual or noise elements. Then, data augmentation techniques are applied to balance categories increasing the number of samples in the data set for the required categories, as explained in Sect. 2.1. Then images are resized according to the input of the models i.e., [227 227 3], [224 224 3], and [299 299 3] pixels for AlexNet, GoogleNet, and InceptionV3 deep neural networks respectively. The data set is divided into training and testing sets, i.e., 80% samples in training, while 20% images in the test set used for validation. Features are extracted through AlexNet and ResNet-18 pre-trained deep models, that are fed as input to traditional machine learning classifiers for the training and testing phase. Each traditional classifier is trained independently on both deep features using their own parameters setup. Using

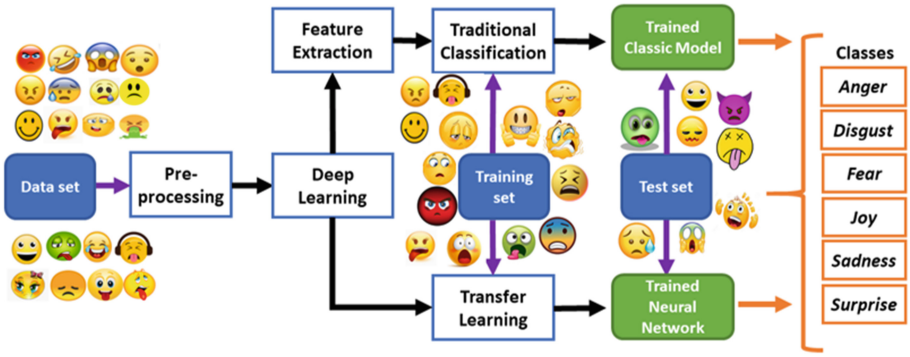

Fig. 1. Framework of the image-based emojis classification

transfer learning, the last fully connected layers of pre-trained deep models are fine-tuned on the emotion categories.

3.3 Feature Descriptors

To train and test traditional machine learning classifiers, fixed-size feature descriptors have been extracted [7] using the two pre-trained deep models of AlexNet [5] and Resnet18 [13]. The length of the feature descriptors, i.e. vectors, extracted through AlexNet and ResNet18 deep models is 1×4096 and 1×512 respectively for each emoji pictogram. Higher-level layers give low-level features and the feature descriptors length will be long, while deeper layers give us higher-level features with reduced size feature descriptors that can be easily processed. These features are extracted to train and test traditional machine learning classifiers.

3.4 Supervised Machine Learning Classifiers

The following supervised machine learning classifiers trained on the feature extracted through AlexNet and ResNet18 deep models.

K Nearest Neighbors (K-NN) [1] finds the k data points that are nearest to a given sample data point. The number of k neighbors is tuned by the user. For each sample of the testing data, the algorithm output associates membership to each emotion class, which depends on the value of k i.e., how many nearest neighbors are voting to a specific class. Experiments for K-NN have been performed with different k values i.e., odd values between 1 and 15. The value of K i.e., the number of neighbors which will contribute to the final decision, is tuned by the user and odd values are recommended to have fewer chances of a tie.

Support Vector Machine (SVM) [2], is designed for binary classification. To classify the data points (i.e., our emoji samples), the objective is to find the partition of the input space through hyper-planes as decision boundaries. For multi-class classification, the problem is divided into multiple binary classification problems. In this study we have used *Linear SVM*, *Radial Basis Function (RBF) Kernel*, and *Polynomial Kernel*.

Decision Tree (DS) divides the data into sub-groups recursively. It is a method for the approximation of discrete-valued functions. DS learns a heuristic, non-backtracking search, through the space of all possible decision trees. A pruning algorithm is given to avoid over-fitting [3].

Linear Discriminant Analysis Classifier (LDA) [4] is used to discover the linear combination of features that effectively isolates categories. For multi-class classification, *Fisher discriminant* is used to discover a subspace that restrains class inconsistency.

3.5 Deep Learning Classifiers

Three deep learning classifiers, i.e., AlexNet, GoogleNet, and InceptionV3 are pre-trained on the ImageNet database including 14,197,122 images of over 1000 different general object categories. Then, transfer learning is applied.

AlexNet [5] contains 8 layers, i.e., the first 5 layers are convolutional layers and the last 3 are fully connected layers. The input image is fed to the pre-trained network with a size of $[227 \times 227 \times 3]$ pixels. The softmax function receives the output of the last fully connected layer, which produces a distribution over the given categories.

InceptionV3 [6] is a deep convolutional neural network with 48 layers. The InceptionV3 pre-trained neural network has 3 main blocks: the basic convolutional block, Inception module, and classification block. The training process of InceptionV3 is accelerated by using a 1×1 convolutional kernel by decreasing the number of feature channels.

GoogleNet [14] is a 22 layers deep convolutional neural network. The input of the GoogleNet RGB images is of size $[224 \times 224 \times 3]$ pixels.

For fine-tuning deep neural networks, 3 different independent training functions/optimizers are used, i.e., Adaptive Moment Estimation (adam) [15], stochastic gradient descent with momentum (sgdm) [16], and Root Mean Square Propagation (rmsprop). *Adam* is an extension of stochastic gradient descent that has a small memory requirement and requires only first-order gradients, while *sgdm* uses stochastic gradient descent with momentum, i.e. a moving average of gradients, used to update the weights. *rmsprop* uses an adaptive learning rate instead of setting it as a hyper-parameter.

4 Experimental Results

This section describes and compares the experimental results achieved through traditional and deep learning classifiers. Accuracy is used as a performance metric to grade different classification algorithms, chosen as the most commonly used metric both for supervised and deep learning.

4.1 Traditional Machine Learning Classifiers Performance

This section expands on the experimental results achieved using traditional machine learning classifiers described in Sect. 3.2. Figure 3(a) highlights an accuracy achieved through k-NN classifiers using different odd values of k ranging from 1 to 15. We used features that are extracted through AlexNet and ResNet18. Experimental results indicate that for k = 1, we achieved the highest accuracy 94.66% and 94.97% using features extracted through AlexNet and Resnet18deep models, respectively. With a higher k, the classification performance keeps on degrading. K-NN Achieves higher performance on features extracted through Resnet18 model. SVM classifier trained using Linear, Radial Basis Function

(RBF), and Polynomial Kernels. Results Fig. 3(b) indicates that Linear SVM gives better performance compared to RBF and polynomial kernels i.e., 94.97% accuracy is achieved using features extracted through AlexNet, while SVM with RBF kernel gives less than 25% accuracy, which is very low. Unlike K-NN, SVM performs better on AlexNet extracted features. When data is linearly separable, Linear SVM performs better. K-NN achieved the highest accuracy 94.97% with k = 1 on Resnet18 feature descriptor, while linear SVM achieve the same highest accuracy of 94.97% through AlexNet feature descriptor. Results show that K-NN and SVM achieve the same highest accuracy (94.97%), while LDA and Decision tree performance is low compared to SVM and K-NN. Decision Tree achieves less than (66%) accuracy, which is very low compared to the other three traditional supervised classifiers.

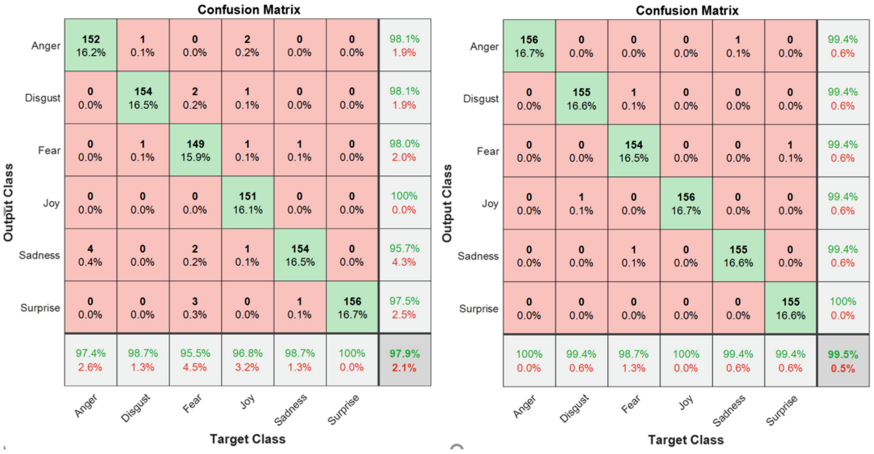

Fig. 2. (a) and **(b)**: on left confusion matrix of AlexNet and right confusion matrix of the best performing InceptionV3 NN (accuracy achieved: 99.47%)

4.2 Deep Classifiers Performance

This paragraph shows the results of the experimented deep neural networks.

GoogleNet and InceptionV3 perform better compared to AlexNet. We achieved the highest accuracy of 97.86%, 98.40%, and 99.47% through AlexNet, GoogleNet, and InceptionV3 model respectively. For training of these neural networks, we have used three different training functions (optimizer) i.e. *adam*, *sgdm*, and *rmsprop*. Table 1 shows the details of the experiments performed using different training functions and Learning Rates. The loss for InceptionV3 is lower compared to GoogleNet. The highest accuracy (97.47%) is achieved using InceptionV3 model with a learning rate 0.0001 and training function *adam*, while GoogleNet achieve highest performance 98.40% using training function *rmsprop* and learning rate 0.0001. The possible reason for the highest accuracy achieved

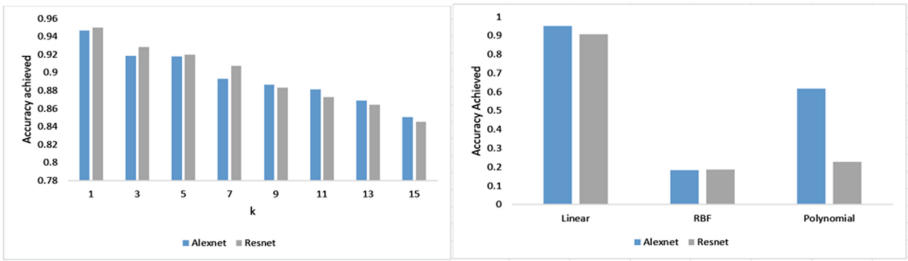

Fig. 3. (a) and **(b)**: on the left, performance of K-NN for different k values; on the right, performance of SVM with different kernels on deep features

through InceptionV3 model may be the number of layers of the model, InceptionV3 has more layers compared to GoogleNet and AlexNet. AlexNet achieves highest accuracy 97.86% using training function *adam* and learning rate 0.00001. Another important observation is that for both AlexNet and InceptionV3, the highest accuracy is achieved through *adam*, while GoogleNet achieves the highest accuracy through *rmsprop*.

4.3 Discussion

Figure 4 shows the comparison of the deep learning (DL) and traditional supervised classifiers trained on the features extracted through AlexNet and Resnet18. InceptionV3 achieves the highest performance, while deep learning outperforms traditional machine learning. Among traditional classifiers, K-NN and SVM have the same best performance around 95%, similar to the LDA in the second place around 92%. Decision Tree has the worst performance, i.e. around 65%.

Further analysis of the performance of InceptionV3 (IV3) and AlexNet (AN), i.e., the best and the worst DL classifiers, can be achieved with the help of confusion matrices shown in Fig. 2(a) and (b), to show the overall and class-wise performance. IV3 achieved an overall accuracy of 99.47%, while AN achieved 97.86%. IV3 perfectly classified all the images of class Joy and Anger, while AN

Table 1. Validation accuracy (%) achieved using different learning rates (LR) and training functions.

Classifier	AlexNet			GoogleNet			InceptionV3		
Optimizer	*adam*	*sgdm*	*rmsprop*	*adam*	sgdm	rmsprop	*adam*	*sgdm*	*rmsprop*
Learning rate	Accuracy achieved								
0.01	16.67	16.67	16.67	16.67	16.67	16.67	82.37	98.4	79.81
0.001	16.67	16.67	16.67	84.72	97.86	16.67	95.51	98.61	96.37
0.0001	94.55	96.69	92.95	98.29	97.33	**98.40**	**99.47**	93.91	98.18
0.00001	**97.86**	95.51	97.65	96.69	82.26	97.54	93.91	75.53	94.76

misclassified several images. Besides the confusion matrix of the best-performing network, it is interesting to see also the confusion matrix of the worse deep model. In fact, from the confusion matrix, we can see which classes are wrongly classified in which classes, and see if such classes have any common element which can motivate the errors or if instead, the mistake on the network training is evident. In this case, we can see that AlexNet cannot detect the emoji pictogram features as easily as other networks. If some errors can depend on the samples, such as the Angry emotions mistaken into Sad or Disgust, where we have the same downward direction of lips, in other cases such as joy mistaken as Sad, Fear, and Angry, something went wrong in the network training. The final result is a high accuracy, but the single mistakes are heavier than the ones made by InceptionV3. In the latter, Joy, which is the emotional class that also in face detection is easier to recognize, does not present any mistake. The errors are apparent in the Fear class mistaken as Sad or Disgust, sharing similar features, Sad is one time mistaken as Angry, Surprise as Fear one time, having the big open mouth as a shared element. Only in the case of disgust mistaken as Joy, the training issue is more evident. Results achieved through InceptionV3 model are better compared to the other tested classifiers of this study.

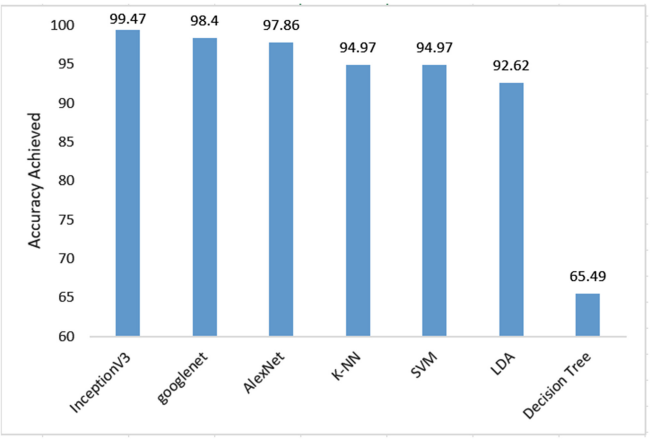

Fig. 4. Highest performance achieved by deep learning models & traditional classifiers

5 Conclusion

In this study, systematic experiments are performed to classify emojis pictograms into six basic classes of Ekman emotions. We run the experiments on traditional supervised classifiers trained on deep features extracted through AlexNet and Resnet18 pre-trained networks, and three deep learning pre-trained NNs, trained

using transfer learning. Traditional classifiers K-NN and SVM achieved 94.97% accuracy using Resnet and AlexNet features respectively, while Decision Tree achieved the lowest accuracy i.e., 65.49% and 58.01% using AlexNet and ResNet features respectively. The highest 99.47% accuracy is achieved by InceptionV3 model, while AlexNet and GoogleNet performances are better compared to traditional supervised classifiers.

A fruitful extension of this work is to use multi-modal approaches e.g., merging our work with Natural Language Processing techniques for a deeper context analysis.

References

1. Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K.: KNN model-based approach in classification. In: Meersman, R., Tari, Z., Schmidt, D.C. (eds.) OTM 2003. LNCS, vol. 2888, pp. 986–996. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-39964-3_62
2. Chih-Wei, H., Chih-Jen, L.: A comparison of methods for multiclass support vector machines. IEEE Trans. Neural Netw. **13**(2), 415–425 (2002)
3. Mitchell, T.M.: Machine learning (1997)
4. Şener, B., Çokluk-Bökeoğlu, Ö.: Discriminant function analysis: concept and application. Eurasian J. Educ. Res. (EJER) **33**, 73–92 (2008)
5. Li, W., Li, D., Zeng, S.: Traffic Sign Recognition with a small convolutional neural network. In: IOP, vol. 688, no. 4 (2019)
6. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 25, pp. 1097–1105 (2012)
7. Sahoo, J., Prakash, S.A., Patra, S.K.: Hand gesture recognition using PCA based deep CNN reduced features and SVM classifier. In: IEEE International Symposium on Smart Electronic Systems (iSES), pp. 221–224 (2019)
8. Ekman, P.: An argument for basic emotions. Cogn. Emot. **6**(3–4), 169–200 (1992)
9. Jain, D.K., Shamsolmoali, P., Sehdev, P.: Extended deep neural network for facial emotion recognition. Pattern Recogn. Lett. **120**, 69–74 (2019)
10. Yan, J., Wenming, Z., et al.: Sparse kernel reduced-rank regression for bimodal emotion recognition from facial expression and speech. IEEE Trans. Multimed. **18**(7), 1319–1329 (2016)
11. Martin, W., Metallinou, A., et al.: Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional LSTM modeling. In: Proceedings of INTERSPEECH, pp. 2362–2365 (2010)
12. Liu, X., Fan, F., et al.: Image2Audio: facilitating semi-supervised audio emotion recognition with facial expression image. In: Proceedings of the IEEE/CVF, pp. 912–913 (2020)
13. He, K., Zhang, X., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
14. Szegedy, C., Liu, W., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9 (2015)
15. Ba, J.L., et al.: Adam: a method for stochastic gradient descent. In: ICLR, pp. 1–15 (2015)

16. Liu, Y., Gao, Y., Yin, W.: An improved analysis of stochastic gradient descent with momentum. arXiv preprint arXiv:2007.07989 (2020)
17. Franzoni, V., Biondi, G., Perri, D., Gervasi, O.: Enhancing mouth-based emotion recognition using transfer learning. Sensors **20**(18), 5222 (2020)
18. Gervasi, O., Franzoni, V., Riganelli, M., Tasso, S.: Automating facial emotion recognition. Web Intell. **17**(1), 17–27 (2019)