Facial Emotion Classification on Convolutional Neural Network Based Method with Distinctiveness Pruning

Xutong Wei

Research School of Computer Science,
Australian National University
u7034958@anu.edu.au

Abstract. Facial expression recognition and its derived applications are currently receiving a lot of attention from society, and some applications of it have begun to be used in real life. The aim of this paper is to build Neural Network based classifiers to classify 7 different emotions under the Static Facial Expressions in the Wild (SFEW) dataset, where the data is closer to real life than other datasets that are collected under laboratory control. A pre-trained convolutional neural network based model, ResNet-34, has been used in this paper. This model outperforms the baseline of a two-layer feedforward neural networks and another baseline that provided by the dataset paper. In addition, distinctiveness pruning is also used in this article to reduce the size of the network, thereby increasing the speed of training. The best performance of our ResNet-34 model on SFEW reached 53.01%.

Keywords: Convolutional Neural Networks; Neural Network ; Network Pruning ; Distinctiveness ; Facial Emotion Recognition ; SFEW database

1 Introduction

The research to recognise human emotions through facial expressions is currently a hot topic in the field of computer vision. Emotions are an essential part of everyone's life. Social interaction between people requires emotions, and the expression of emotions is also a way for people to share their mental status. By observing the facial expressions of other people, we can understand their current emotions. Therefore, facial expressions are a non verbal way of communication, and it is universal in the world. In psychological research, by studying the expression of human facial emotions, we can better study the human internal world and provide proper help if it is needed. In the field of computer science, facial emotions recognition can also be used in computer human interaction applications, virtual reality applications, cybersecurity and so on [10]. This paper aims to propose a classification method by using a Residual Network, compared with a two-layer Neural Network to recognize 7 different facial emotions on Static Facial Expressions in the Wild (SFEW) dataset. The distinctiveness technique will then be applied to the model to reduce the size of the network.

Distinctiveness pruning is one of the simple network pruning method. Neural Network pruning techniques started by inspecting and removing the hidden neurons, because it is easy to decide the number of neurons in the input layer and the output layer, but hard to decide the exact number of neurons in the hidden layer, even following the rules of thumb [5]. However, as [4] mentioned, pruning the network by human inspection is hard, so, we need automatable pruning methods. There are several properties to eliminates the hidden neurons, which include: relevance, contributions, sensitivity, badness and distinctiveness [5]. In this paper, by following distinctiveness pruning technique [4, 5], with observing the neuron behaviour and the weights matrix, less important hidden neurons and redundant hidden neurons from the network will be pruned. As [4] mentioned, too many hidden neurons will slow the speed of running a network. Therefore, eliminating the neurons, with duplicating functionality or no functionality, helps to speed up the trained network with equivalent performance without extra training.

SFEW is a static database that is collected from a dynamic database called Acted Facial Expressions in the Wild (AFEW) [2]. All data is selected from several movies with different background scenes and a relatively large age range of the characters, allowing to simulate the conditions as closest to real life as possible. As [2] mentioned, the data in the SFEW database is close to real life, where most of other databases such as JAFFE [9] and Multi-PIE [6] are created under lab-controlled conditions. Since facial emotion recognition applications will be used for society and real life, a dataset close to real life like SFEW is more worthy of being selected. There are 7 classes of facial emotions in SFEW, which includes Angry, Disgust, Fear, Sad, Surprise, and Neutral. Our Neural Network based classifier will be trained to recognize these 7 types of emotions.

Due to the popularity of facial emotion recognition topics, many methods have been proposed already and achieved good results. In the original paper of SFEW [2], a non-linear support vector machine has been selected to perform facial recognition. Similarly, [3] also reported a method by using support vector machine for classification, and selecting features with random forest recursive feature elimination. In addition, other machine learning and deep learning methods such as convolutional neural networks are also widely used. As [8] proposed, applying GoogleNet as an emotion recognizer provides another feasible solution to contribute on facial emotion recognition.

In this paper, a pre-trained Residual Network (ResNet) [7] will be used to classify emotions based on images. Transfer learning technique will be applied to the pre-trained ResNet, so to make the model adapt to our task. Different numbers of layers ResNets have been tried, and we decided to use ResNet-34 as our model, then apply distinctiveness pruning on it to reduce the size of the network. Besides that, we will also compare our ResNet model with a 2-layer feed-forward neural network that is feeding 2 sets of top-5 Principal Components (PCA) features from the Local Phase Quantization (LPQ) descriptor and pyramid of histogram of oriented gradients (PHOG) descriptor as the input of the network.

2 Methodology

SFEW dataset will be discussed at first. In addition, both the residual convolutional neural network (ResNet) for image classification and the 2-layer feed-forward neural network (NN) for classification on features will be introduced in detail in this section. For each model, data-inspection, data pre-processing, and how the neural network is defined will be illustrate. For the ResNet model part, transfer learning will also be discussed, so to give reasons of why the original ResNet model can be used to our dataset successfully. After that, distinctiveness pruning will be concluded for both models. And at the end, we will mention the training and evaluating of both networks.

2.1 SFEW Dataset

As mentioned above, SFEW is a facial emotions dataset that is close to real world conditions, which means that data are not collected under the lab environment [2]. All data are collected from movies by following the person independent training and testing protocol. There are 7 classes of facial emotions in this dataset, which includes angry, disgust, fear, happy, neutral, sad and surprise. Based on different models, we used two different versions of dataset, where the first one is the facial images with related labels, and each image has size 720 pixels in width and 576 pixels in height. This version of the dataset will be used in the ResNet. The second set of datasets is the extracted features based on the images, which will be the dataset for the 2-layer NN model. For each data instance, there are 10 features which is combined by top-5 Principal Components of Local Phase Quantization (LPQ) features and top-5 Principal Components of Pyramid of Histogram of Gradients (PHOG) features.

2.2 Two-layer Feedforward Neural Network Model

Data Inspection and Pre-processing In this case, the dataset with 10 features of Local Phase Quantization (LPQ) features and Pyramid of Histogram of Gradients (PHOG) features were used to classify facial expressions. After inspecting the data, we deleted the first column of data since they are the names of each original instance. Then we cleaned the data, which removes all missing data. Therefore, there were 674 data instances left, where there were 675 data instances originally. The types of the dataset are formed by strings and numbers.

From the data, 80% is randomly selected as training data, and the remaining 20% are testing data. To pre-process the data, normalization has been applied to both training dataset and testing dataset separately. Besides that, target classes' values are also transformed from value 1 to 7 to value 0 to 6, which is necessary to calculate the loss function between predicted targets with the ground truth.

Define a 2-layer Neural Network A two layers feed-forward neural network is implemented for facial emotion recognition. Since each instance has 10 features, the input size of our neural network is 10, and the output size of the network is the same as the number of emotion classes, which is 7.

As for the loss function, cross entropy is applied, because the network is a multi-class classification model. However, from the output is difficult for us to distinguish which emotion class it belongs to directly. So, we apply the Softmax function over the output layer result, so that each output result becomes a percentage, where the sum of the percentages of all output results is equal to 1. After that, we can easily identify that the class with the largest proportion is the predicted class by the network for the input instance.

Hyper-parameters Selection Hyper-parameters selection is another significant element when building a neural network. It is hard to decide all hyper-parameters with the best performance at the beginning. Therefore, in this paper, hyper-parameters are decided by running multiple experiments, and we decide to use 3000 epochs as a standard and 150 hidden neurons. Sigmoid function is used as the activation function of the hidden layer. Moreover, after comparing with test accuracy by using SGD and Adam optimizer, Adam had been chosen as the optimizer for this network, with the learning rate 0.02.

2.3 Residual Convolutional Neural Network Model (ResNet)

Data Inspection and Pre-processing There are 675 images in the dataset, where for class disgust, there are only 75 images, and there are 100 images for each of the other classes. By observing the data, we found that there was an image that belonged to both classes of fear and surprise. To avoid any unnecessary troubles, we set the image to 2 different names under 2 class labels. The training data were randomly selected 80% from the whole dataset, and the remaining of 20% were testing data. Since the input size of ResNet was 224 x 224, before putting data into the network, we resized them to 224 x 224, and then normalized them into the same range, which helped the network learn with less bias. Besides that, since the limited amount of data, we also provided an option to apply data augmentation for training data. Data augmentation enables experimenters to gain more data immediately from the existing data to train the model without collecting more real data. Our data augmentation methods included randomly rotating images by 15 degrees and then flipping them horizontally.

Define a ResNet We implemented a ResNet model with 34 layers to classify images, which was proposed by [7] in 2015. Even though a two-layer NN is able to address the problem, since it may make the layer to be massive, the network may end with overfitting the data. So, deep network has been proposed to address the problem. However, as the number of layers going up, deep network may hard to be trained properly because of the vanishing gradient problem, which makes the result of worse performance. ResNet can help with the problem of performance degrading, as its core idea is to use residual learning framework in the neural network [7]. A residual learning block is shown in 1, and the general idea of it is to use the shortcut connection (i.e., the directly connection that from the upper layer to the lower layer with skipping some layers) to perform identity mapping x and stack it to the layers output F(x) that are in between the connection, so to get the formulation of F(x)+x. This solves the problem of degradation because "if the added layers can be constructed as identity mappings, a deeper model should have training error no greater than its shallower counterpart" [7].

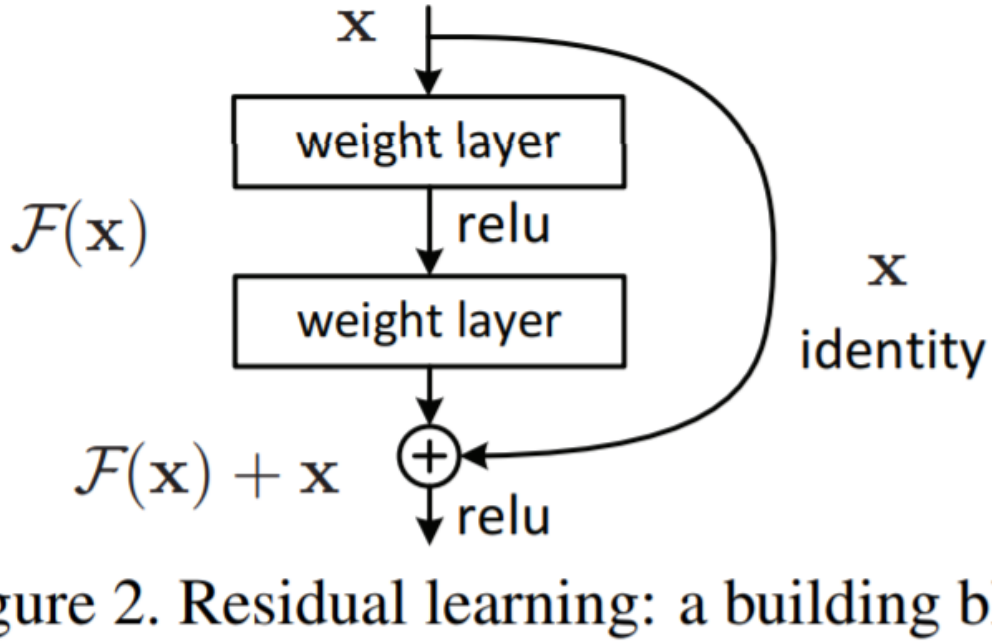

Figure 2. Residual learning: a building block.

Fig. 1. Residual learning [7]

Residual Network is implemented based on the plain network, which is convert the plain network into the residual version, where the only difference is to insert the shortcut connections, and each connection skips two hidden layers. This paper used ResNet-34 as the classification model, and this model was selected by comparing the results and hidden layers in ResNet-18, ResNet-34, and ResNet-50. The evaluation result will be shown in Section 3.1. The structure of ResNet34 is show in 3, where each layer has kernel size size 3x3 with different number of filters.

Since ResNet-34 is a huge convolutional based deep neural network, it will take a long time to train from the scratch, also lots of training data will be needed. Hence, we used a pre-trained ResNet-34 by ImageNet dataset, and keep training it in our dataset. Transfer learning technique had been implemented by our code, so that we can reuse the pre-trained model to our classification task. There are two scenarios for transfer learning, where the first one is

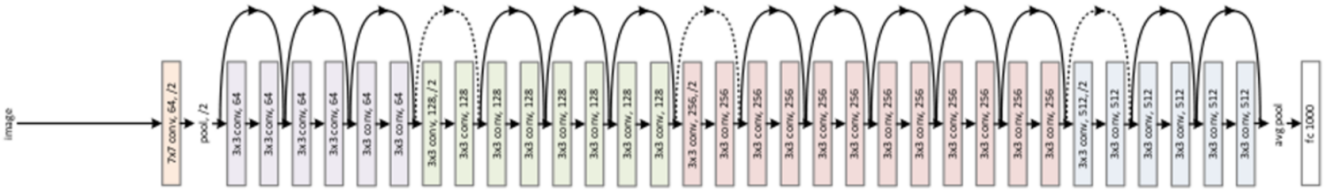

Fig. 2. ResNet-34 Structure [1]

fine tuning, and the second one is fixed feature extraction. As for fine tuning, the pre-trained network is treated as an initialization, and we will keep training the model with our dataset. As for fixed feature extraction, several layers of weights will be frozen, which is depending on how many layers the network needs to train. The frozen layers are not able to be trained, and the rest of unfrozen layers will be trained as normal. By comparing with the test data accuracy of each scenario, we decided to transfer learning as fine tuning. The evaluation result will be provided in section 3.1.

Our dataset has 7 classes, so we can just modify the fully connected layer of the ResNet, to make the model adapt to our data. After several experiments, we decided to assign a 'nn.Sequence()' as the fully connected layer, which is constructed by a linear function with input as the features of the original fully connected layer and the output neurons is 128, a sigmoid function, another linear function with 128 neurons of input and 7 outputs which indicates 7 classes of our data, and then apply a softmax function to get the result of classification

Hyper-parameters Selection Hyper-parameters are decided by running multiple experiments, and we decide to use 30 epochs and batch size is 4. SGD with momentum equals to 0.9 is chosen as the optimizer for this network. The learning rate starts with 0.002, and for every 10 epochs, the learning rate will reduce to the 1/10 of the previous learning rate.

2.4 Pruning the Network by Distinctiveness

The distinctiveness of pruning is based on hidden neurons. The hidden neurons are determined by the representation of unit output activation vector from the input pattern representation [4]. After that, find the angle between them, and then use the angle to determine whether the hidden neuron should be removed from the neural network or not. The result of the angle calculation should be between 0 and 180 degrees. If the angle is less than 15 degrees, it means that the two hidden neurons are similar, which represents that both of neurons have similar functionality. We should remove one of those neurons from the network and add its weight to the other neuron that is not removed. On the other hand, if the calculated angle is greater than 165 degrees, it means that those two hidden neurons are complementary, because they are opposite to each other. This also means that this pair of neurons has no function, so both of them should be removed from the network. After removing neurons in any situation, by updating the network, we not only update the new size of the current hidden layer, but also update the size of the following layer.

In this paper, we only remove either one (similar neurons) or both compared neurons (complementary neurons) in each pruning time.

For 2-layer NN, to follow the distinctiveness property, we compute hidden unit output activation vectors, and then to reduce the size of the neurons by comparing the similarities of each pair of vectors. Neurons that are too similar to each other or complementary to each other will be removed.

In ResNet-34, we proposed that the similarities of the weight matrix also can help with pruning the network. Since the size of ResNet is huge, performing the pruning technique on the whole network is time consuming, hence, we decided to perform pruning on the fully connected layer only.

2.5 Training and Evaluating the Network

Before training the network, the SFEW database will be randomly divided into two parts. The first part is the training set, which contains 80% of the data. The other part is the testing set, which has the remaining 20% of the data. We will train the network base on the training set, and then use the testing set to evaluate the accuracy of the network. In order to make the evaluation results less biased and have a more generalized result, we will randomly select the training set and the detection set 3 times and train them separately. The final accuracy rate is an average of the testing accuracy gathered from the previous 3 tests.

3 Results and Discussion

3.1 Fine Turning or Feature Extraction for ResNet

We experimented on 3 different layers of ResNet on both fine tuning and feature extraction models to observe the testing accuracy of which model gives the best result. All results are shown in 1, where FT meaning fine tuning scenario, FE meaning feature extraction, HTA meaning highest test accuracy and ATA meaning average testing accuracy. We can see that fine tuning scenario has better testing accuracy overall than feature extraction. It happened because for feature extraction, the whole pre-trained model was frozen and the only trained layer was the last layer that we added to predict classification, which did not have enough capacity. Hence, we decided to fine tuning the model.

Even though ResNet-50 had the best performance, we decided to choose ResNet-34 as our pre-trained model. One reason is that, the higher number of layers, the more features can be extracted. Since our dataset has limited size, too much layers may overfit the model. Therefore, we decided to use ResNet-34.

Table 1. Comparison Between Fine tuning and Feature Extraction for Different layers of ResNet

Model	FT-HTA	FT-ATA	FE-HTA	FE-ATA
ResNet-18	%	%	%	%
ResNet-34	53.33%	49.1%	39.26%	35.19%
ResNet-50	56.29%	50.6%	39.25%	34.36%

3.2 ResNet-34 with data augmentation and without augmentation

The same model is trained differently by one with data augmentation and other one without data augmentation. Since the size of our dataset for ResNet-34 is relatively small, which has only 675 images, if we train the model direactly on the dataset, the model will be easy to overfit. As shown in the 3, we can see that for the data that was not trained by augmented data, after 30 epochs, the training model was already overfitted, which had more than 90% training accuracy, where the testing accuracy was only around 52%. On the other hand, for the model that was trained with data augmentation, after 30 epochs, the training accuracy was around 64% with 46% testing accuracy. Even though the model with data augmentation had relative worse testing accuracy, if we keep training on the model, it has more potential than the other one, since the other model is already overfitted.

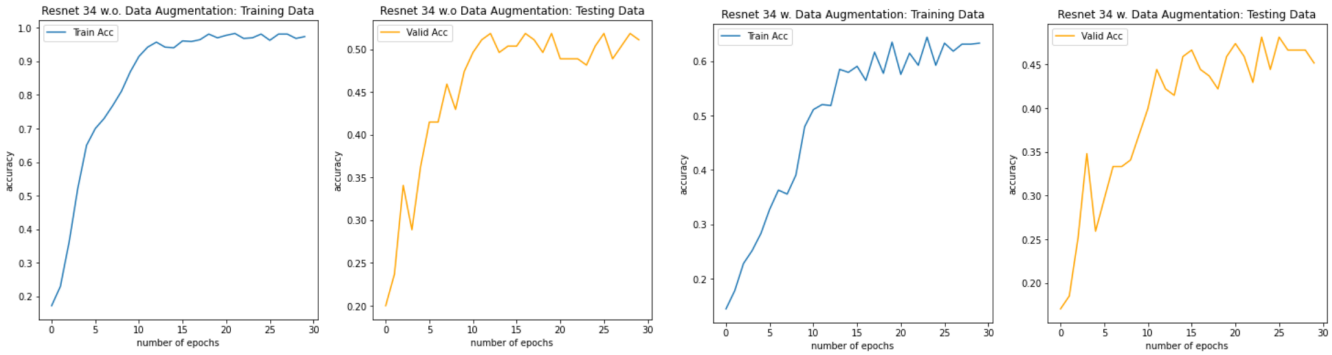

Fig. 3. ResNet-34 Model training without (on the left) and with (on the right) data augmentation

3.3 ResNet-34 with pruning and without pruning technique

The purpose of pruning the neural network is to reduce the size of the neural network, which can speed up running the trained network. Also, we try to keep the performance of the pruned neural network to have an equivalent accuracy as before pruning. In the Table 2, we exemplified the comparison of the results of the ResNet without pruning and the network after pruning. The results show that under the same settings, and run for 35 epochs, the training accuracy of the network without pruning is around 6.1% higher than the network after pruning, and the test accuracy of pruned network is slightly higher than the one without pruning by 2%. We thought that this may be caused due to a bit overfitting happening in the network with no pruning, which leaded to a higher training accuracy and lower testing accuracy. We found that after each pruning, the weight and size of the updated network were changed, so the training accuracy was still affected to a certain extent.

Table 2. Training and Testing Accuracy of NN and Pruned NN

Model	Training accuracy	Testing accuracy
ResNet-34	66.28%	50.93%
ResNet-34 + Distinctiveness Pruning	60.17%	53.01%

3.4 Comparing pruned neural networks with SFEW paper's method

In this part, we compare our methods with the method provided in the SFEW database [2], and the results are shown in table 3. As mentioned in [2], they combine the PHOG and LPQ descriptors, and used PCA to extract features to simplify the complexity. The extracted features keep 98% of the variance. Then they put the features into the non-linear SVM to get the final classification result. Similarly, our 2-layer NN model also uses those extracted principal components as the input data instances of our neural network, and achieves an accuracy of 27.39%. However, the baseline classification accuracy of [2] is 19%. By comparison, our accuracy is higher than them by greater than 8%.

However, although our results surpass the baseline, since our accuracy of NN model is only around 27%, the possibility of misclassification on testing data is still very high. There are some possible reasons to cause this problem. First of all, the features we use are extracted by PCA from the combination of two descriptors. Although 98% of the variance is kept, some information will still be lost during the extraction process. And the lost information may cause the neural network to be biased in the training process, which results in a lower accuracy on the testing data. On the other hand, due to the limited amount of data contained in the data set, the network model is trained with a small amount of data, which is resulting in overfitting, and has an insufficient generalization ability of the model. Due to the overfitting, it affects the testing accuracy.

Hence, ResNet-34 has been built, which uses images directly in the CNN and makes classification prediction. We fine tuned a pre-trained model, and keep training on the augmented training dataset. The accuracy under this model increases to 53.01%, which is the best result compared with the other two methods. This model shows Three advantages compared with the NN model. At first, this ResNet-34 model feed the images directly into the neural network, which avoids missing important features before putting them into the network. Secondly, fine tuning a pre-trained model can reduce the training time, and can improve the performance since the model was trained on a very large dataset. Besides that, data augmentation increases the number of data significantly from the original data, which helps to avoid overfitting problem.

However, even though the result of ResNet-34 significantly improved the classification performance, the accuarcy is still not good enough to solve real-world problems. The reason to cause this accuracy may because this dataset aims to closely simulate the real world, rather than being completed under lab-controlled environment, the noise of our database may also be higher than those of the database established under the control of the laboratory. So, it may affect the result of accuracy.

Table 3. Methodology Evaluation Compare with SFEW paper

Model	Testing accuracy
ResNet-34 + Pruning	53.01%
NN + Pruning	27.39%
SFEW method (Support Vector Machine)	19%

4 Conclusion and Future Work

In conclusion, this paper aims to build a facial emotion recognition classifier by using a feed-forward neural network. SFEW has been chosen as the dataset since it is close to the real life, and two versions of dataset are provided. Two models are implemented in this paper, where the first one is a ResNet-34 model, and the second one is a two-layer NN. To reduce the network size and make the network run faster, distinctiveness pruning technique is applied to the neural network, which removes hidden neurons if two of them are similar or two of them are complementary. Finally, our ResNet-34 model reaches 53.01% accuracy, where the accuracy of our NN model was 27.39% and the accuracy classification baseline on SFEW was 19%, .

Even though our ResNet model has the higher accuracy than the baseline and the NN's accuracy, the model is still not ideal to be used as a classifier in real world. For the future study, we can try more data augmentation techniques can be applied, so to increase the number of data instances, and make the model more robust. Also, to improve the accuracy, we can try to combine the attention mechanism with convolutional neutral network, because the attention machanism can identify the pertinent part of a task. Also, Vision Transformer model is another method we can try, which is a Transformer that can be applied in computer vision field, and also uses the attention mechanism.

References

1. Kaggle: Deep residual learning for image recognition (2018), https://www.kaggle.com/pytorch/resnet34
2. Dhall, A., Goecke, R., Lucey, S., Gedeon, T.: Static facial expression analysis in tough conditions: Data, evaluation protocol and benchmark. In: 1st IEEE International Workshop on Benchmarking Facial Image Analysis Technologies BeFIT, ICCV. pp. 2106–2112 (2011)
3. Domínguez-Jiménez, J., Campo-Landines, K., Martínez-Santos, J., Delahoz, E., Contreras-Ortiz, S.: A machine learning model for emotion recognition from physiological signals. Biomedical Signal Processing and Control **55**, 101646 (2020). https://doi.org/https://doi.org/10.1016/j.bspc.2019.101646
4. Gedeon, T.: Indicators of hidden neuron functionality: the weight matrix versus neuron behaviour. Proceedings 1995 Second New Zealand International Two-Stream Conference on Artificial Neural Networks and Expert Systems pp. 26–29 (1995)
5. Gedeon, T., Harris, D.: Network reduction techniques. Proc. Int. Con. on Neural Networks Methodologies and Applications **1**, 119–126 (1991), aMSE, San Diego
6. Gross, R., Matthews, I., Cohn, J., Kanade, T., Baker, S.: Multi-pie (12 2013). https://doi.org/10.1109/AFGR.2008.4813399
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015)
8. Ivanova, E., Borzunov, G.: Optimization of machine learning algorithm of emotion recognition in terms of human facial expressions. Procedia Computer Science **169**, 244–248 (2020). https://doi.org/https://doi.org/10.1016/j.procs.2020.02.143, postproceedings of the 10th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2019 (Tenth Annual Meeting of the BICA Society), held August 15-19, 2019 in Seattle, Washington, USA
9. Kamachi, M., Lyons, M., Gyoba, J.: The japanese female facial expression (jaffe) database. Availble: http://www. kasrl. org/jaffe. html (01 1997)
10. Mehta, D., Siddiqui, M.F.H., Javaid, A.: Facial emotion recognition: A survey and real-world user experiences in mixed reality. Sensors (Basel, Switzerland) **18** (02 2018). https://doi.org/10.3390/s18020416