Facial Emotion Recognition using Convolutional Neural Networks

Zeynab Rzayeva
School of Information Technologies and Engineering
ADA University
Baku, Azerbaijan
zrzayeva2018@ada.edu.az

Emin Alasgarov
School of Information Technologies and Engineering
ADA University
Baku, Azerbaijan
ealasgarov@ada.edu.az

***Abstract*—Facial emotion recognition is one of the promptly developing branches within the machine learning domain. In this paper, we are presenting our model based on Convolutional Neural Networks, which is trained on Cohn-Kanade and RAVDESS datasets. The proposed model gets satisfactory results in detecting macro facial emotions on the aforementioned datasets.**

Keywords—Emotion Recognition, Convolutional Neural Networks, Face Expressions

I. INTRODUCTION

In face to face communication, body language accounts for 55% of overall message, while only 7% of it are words. The way we convey the message is crucial to understand the overall situation. While it is easy to understand the facial expressions for human beings, a challenging problem emerges when we intend to teach a machine how to analyze the data and comprehend the emotions of humans in real-time applications. Machines can understand some basic verbal communication with the help of NLP and recognizing facial expression will help machines to understand human beings better. The broad area of Human Computer Interaction (HCI) focuses on getting the max effective interfaces for applications, such as the ones used in psychological consultations, patient care or healthcare, monitoring, rehabilitations, marketing, advertisement, video games, movies, music, and education.

In this paper, we propose a different CNN model that recognizes 5 facial emotions, which are sadness, happiness, anger, surprise and neutral in a real time. The proposed model is trained with Cohn-Kanade and Ravdess datasets. In comparison to already existing approaches, in our work we provide a whole face as an input to the algorithm, instead of using geometry-based or appearance-based methods. Every single pixel in a face is treated like a feature instead of connecting different parts of a face to facial action units.

II. RELATED WORK

Different approaches and methodologies are used in the literature to solve facial emotion recognition tasks. [1] discuss two crucial stages in facial expression recognition system in their paper; facial feature representation and classifier design. Facial feature representation is the process to extract required features from original images to describe the image and currently, there are two known approaches to extract those features: geometry-based and appearance-based. In this work, researchers preferred to use geometry-based approach for feature extraction. Above mentioned work introduced valuable knowledge about geometry-based facial feature representation and provided profound information about current algorithms for the geometry-based approach. Illustrating working principle, advantages, limitation, and accuracy of each algorithm used for feature extraction, this work is quite informative for the researchers. Appearance-based approach used to acquire required features for facial expression recognition is another way to achieve this task. While geometry-based approach mainly focuses on shape, texture and location information of salient components like the eyes, eyebrows, nose, mouth, and chin, the appearance-based method mainly focuses on fiducial points, which are located on the corners of the eyes, outer middle and the corner of the mouse, corner of the eyebrows, tip of the chin, and the tip of the nose. Researchers used fiducial points to extract intransient facial features like eyes, mouth, nose, which are always existing on the face, while ignoring transient components like wrinkles or bulges. This approach is highly preferable for its computational efficiency.

Developing a model for measuring customer loyalty and value with RFM technique and clustering algorithms for facial expression classification, authors used multilayer perceptron (MLP) algorithm. [2] In their work, the number of input neurons corresponds to the size of the feature vector, while this number equals to six for six different emotions classified in this research (happiness, sadness, anger, fear, surprise, and disgust). To start with, PCA algorithm is applied for dimensionality reduction, which allows to decrease the computational cost and prevent overfitting. Researchers used previously mentioned Facial Action Coding System (FACS) technique to map the effects of Action Units (e.g., inner brow raiser, nose-wrinkle, lip corner depressor) to the changes in geometric features. Multiple emotions may be dependent on the same geometric features, however, the magnitude of the movement of those features is deterministic for displaying a different emotion. The researchers tested MLP, which is a neural network-based algorithm. While more hidden layers are preferable when modeling neural network, finding the optimal number of hidden layers is still an actual problem for researchers. In this work,

one hidden layer is chosen for the neural network. The novelty of this work is that researchers tried to prune the search space by unifying geometric features and FACS. According to the researchers, this method ends up with excellent results in time efficiency with retaining the accuracy of the predictions. During the implementation stage, researchers used Cohn-Kanade and FER-13 databases and algorithms were implemented using MATLAB. After the dimensionality reduction and edge extraction (in most cases, there are no valuable pixels for emotion recognition on the edges of images), the neural network is trained with 224 images for each emotion category and results are compared with application of SVM algorithm to the same database. Results showed that SVM algorithm provides slightly better accuracy in predicting each emotion of samples from FER-13 database. However, the accuracy of SVM with respect to MLP in Cohn-Kanade database is considerably higher (on average 85%, and 78%, respectively). MLP method performs well when it comes to time efficiency: while on average, it took 7.2 ms for SVM to execute on Cohn-Kanade database, the time consumed for execution of MLP is drastically lower, with 1.3 ms. And with the new method mentioned previously, this number is even lower with 0.5 ms for MLP and 2.4 ms for SVM. Although, we can observe that new method offered by researchers retains accuracy for MLP, this is not the case for SVM classifier: there is a considerable decrease in prediction accuracy of SVM (from 85% to 78%). Nevertheless, 3 times decrease in time consumption for the calculation of the results is worth mentioning.

The difference of our approach is that instead of using geometry-based or appearance-based approach, we gave the whole face as an input to the algorithm. Every single pixel in a face is treated like a feature instead of connecting different parts of a face to Facial Action Units. Moreover, compared to other papers that use SVM and MLP to detect emotions, in Cohn-Kanade dataset, we used Convolutional Neural Network.

III. METHODOLOGY

To solve facial emotion recognition task, we used Cohn Kanade [3] and Ryerson Audio-Visual Database [4] of Emotional Speech and Song (RAVDESS) datasets to train the model. Cohn Kanade contains black and white pictures which were taken in a sequential form from neutral state to emotion. The dataset has 6 emotions (anger, happiness, sadness, fear, disgust, surprise) and each image is provided with facial landmark points. Since dataset doesn't contain neutral images, we divided each image sequence into two parts and the first part was assigned as neutral while the second part was corresponding to its facial expression label. As a result of manual labeling, we had 3000 images. RAVDESS dataset contains videos of 24 subjects (12 male, 12 female) vocalizing 2 texts with normal and strong emotional intensity. The dataset consists of calm, neutral, happy, sad, angry, surprise, disgust and fearful emotions. Images from RAVDESS dataset were extracted with obtaining frames every 0.5 seconds. As a result, we got 11000 images. Providing raw data to the model is not a preferred option, since it contains a lot of unnecessary information and noise, which can result in low performance during the learning process. Choosing right data preprocessing techniques is as important as building the right model itself. In order to better extract features from data, first we converted all image to grayscale since most parts of our data consisted of grayscale images, and while the number of channels in rgb images is 3, this number is 1 in grayscale images. Secondly, to remove the noise we detected faces in images and cropped them out. By doing so we removed unnecessary background data. As a final step, we zoomed image and pivot points in faces became clearer. We used OpenCV [5] library to perform data preprocessing step.

Final step was to choose the right machine learning algorithm. In our problem domain, little differences in face matters and could change the result. We decided to use Convolutional Neural Networks for this task. CNN is a type of artificial neural network and one of the strongest and most used algorithms in computer vision problems. The CNN consists of a sequence of convolutional layers, the output of which is connected only to local regions in the input. This is achieved by sliding a filter, or weight matrix, over the input and at each point computing the dot product between the two (i.e. a convolution between the input and filter) [6]. This structure allows the model to learn filters that can recognize specific patterns in the input data.

Choosing layers and build model is one of the challenging tasks. That's why we build various models by changing variables. First, we experimented with the size of parameters. We chose two different dimensions of images that we used as an input to the algorithm, which are 32x32x1 and 128x128x1. Cropping images in smaller sizes decreases the number of parameters, which in its turn decreases the time to train the model and at the same time use less memory. This approach makes us light when it comes to prototyping and model efficiency check. The other variable that we chose to change is pooling method. Pooling is a filter that extracts features from some parts of an image, which reduces computation. We experimented with max-pooling and average pooling techniques. The other parameters that we thought will affect the efficiency of the model are kernel size and number of filters change.

IV. IMPLEMENTATION

A. Experiments

We decided to train experimental models with the Cohn-Kanade dataset and after getting enough results, apply the same model to the RAVDESS dataset. The experimental models are built based on two logics. The first logic was to keep the number of filters fixed and decrease the kernel size layer by layer. Each convolutional layer is followed by pooling layer and as a final step all input is flattened and added to the dense layer with units 256 and 128 and the last/output layer was with 5 units with SoftMax activation function because we needed categorical output. In all other layers we used ReLU activation function since deep convolutional neural networks with ReLU are trained several times faster than their equivalents with tanh

units. After building our model we started to train it, and from the Table I we can see that the accuracy is over 75%. In this model, kernel size which has a role in extracting main features is relatively big. In facial emotion recognition domain even subtle and small changes matter so we decided to experiment with small kernel sizes to increase accuracy.

TABLE I. ACCURACY OF THE MODELS

Epoch Size	Accuracy				
	Model I	Model II	Model III	Model IV	Model V
20	0.765	0.7833	0.765	0.705	0.774
30	0.783	0.7666	0.7933	0.773	0.856
50	0.814	0.7866	0.726	0.828	0.832

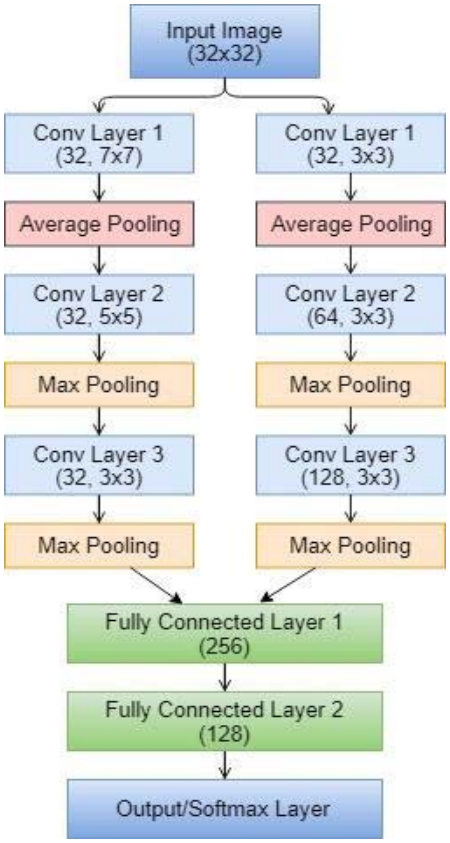

Fig. 1. CNN Architecture of Model I (on the right) and Model II (on the left)

The second idea is based on the logic of VGG16 Visual Geometry Group model [7]. VGG16 model is 16-layer CNN model that is trained on ImageNet dataset. The main feature of this model is using very small (3x3) kernel size in each layer which improved accuracy [7]. In our previous model, our kernel size is relatively big. The small kernel is better at detecting small unique features and in our problem domain even small change in facial lines. So, in this model, instead of changing the kernel size, we changed the number of filters for each layer and the size of kernel stayed stable which is 3x3. Having small kernel size means that more features will be extracted from each layer compared to the previous model. The architecture of both models is shown in Fig. 1. Our first layer is Convolutional layer with 32 filters followed by average pooling. Then second Conv. layer with 64 filters and third Conv. layer has 128 filter and each followed by max-pooling. As you can be seen from the figure, the other layers are the same as in the previous model.

Even though our algorithm performed well in few layers, we wondered whether the number of layers will dramatically change the result. So, we build a new neural network architecture with images size of 128x128. The reason we switched to 128x128 images is that because of the size of 32x32 dimensional images, we can only add a limited number of layers to our architecture. Adding more layers makes the network deeper which is better at detecting features.

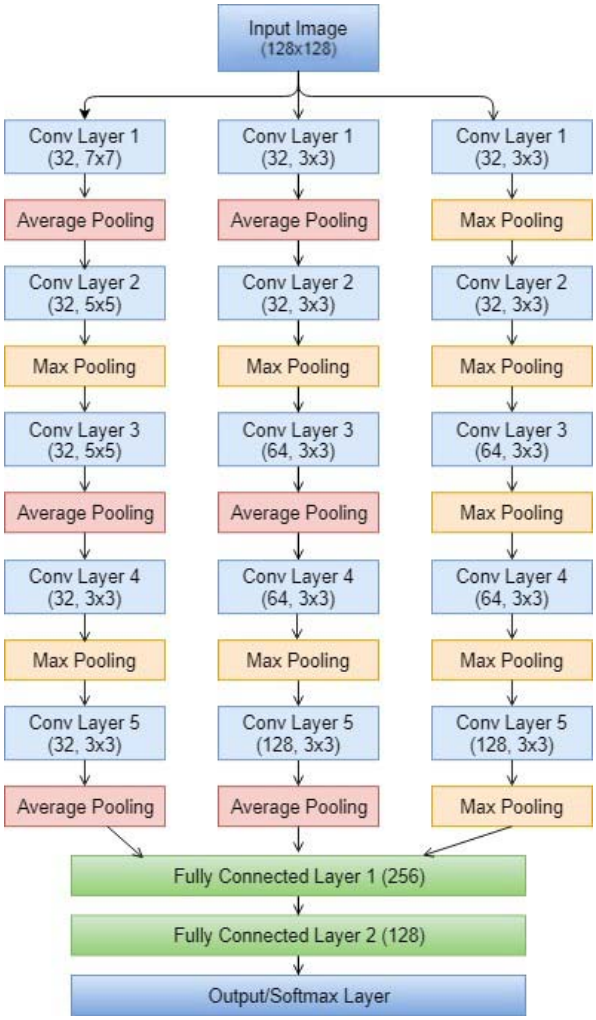

Fig. 2. CNN Architecture of Model III, Model IV and Model V (left to right)

Our third model followed the logic of the first model, so the first layer is convolutional layer with kernel size 7x7 followed by average pooling. Third and fifth layers are the Conv. layers with kernel size of 5x5 followed by max-pooling and average pooling. The last kernel size of convolutional layers is 3x3, followed again by max-pooling and average pooling. The number of filters is fixed, which is 32. The architecture of the model is shown in Fig 2. As we can see from Table I, the accuracy does not improve a lot compared to the Model 1. So,

this logic didn't work well with the input size 128x128x1, as it did with 32x32x1.

Our fourth model is inspired again by VGG16 model, which means kernel size stays fixed and the number of filters increase layer by layer and unlike VGG16, Conv. layers are followed by average pooling and max-pooling in a row. In this model, we got the highest accuracy so far, which is 82.8%.

However, then we decided to change all the average pooling layers with max-pooling layers as VGG16 model. The reason why we perform pooling is to reduce the computation complexity and extract features better. Max-pooling may perform better than average pooling with exponential features sampled from mixture distributions, with one of the components of the mixture being shared between classes [8]. So, in our fifth model, we took the 4th model and changed all the average pooling layers to the max-pooling layers. As we can see from Table I, loss decreased, and accuracy got better. So, after these experiments we conclude that small kernel size and increasing filter size line by line and using max-pooling worked better for our problem with images of size 128x128x1.

B. Proposed model

Even though 5th model performed better than all other models we have built, as you can see from Fig 3. there is an obvious overfitting in our 5th model since there is a big gap between training (blue line) and validation (orange line) lines. In order to solve this problem, we decided to add dropout layers.

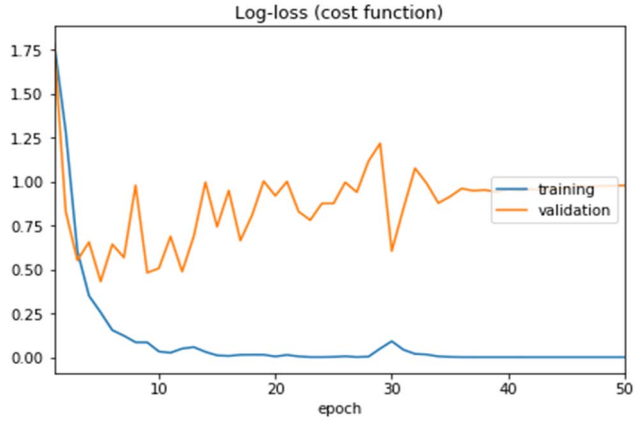

Fig. 3. Accuracy and Log-loss plot of Model V

The final proposed architecture, which consists of 12 layers is inspired by famous VGG16 model. Following the logic of the above model, kernel size is fixed to 3x3, while the number of filters increased by two convolutional layers from 32 to 128 and each followed by max-pooling. Then, two fully connected layers follow convolutional layers. To prevent overfitting, 3 dropout layers with the rate of 0.5 are added to the model. The model predicts the distribution of 5 emotions on an image, which is converted into grayscale and cropped in 128x128x1 dimension around human face.

V. DISCUSSION OF RESULTS

Emotion recognition is a hard problem to solve, even small changes indeed play a big role for further experiments. From the experiments with 5 models, we came to some conclusions about building CNN architecture. The first conclusion is that cropping images in a big dimension significantly increases the accuracy since features become more concrete. Second, is that max-pooling performs better than average pooling with exponential features sampled from mixture distributions, with one of the components of the mixture being shared between classes. Based on the characteristics the final model was built. Three experiments were conducted with the proposed model. Firstly, we trained our model with 2400 images using Cohn-Kanade dataset. Evolution of our model was made based on 600 images from the same dataset, which wasn't used during the training. Accuracy for emotion detection that we got is 88.17±3.08%. Secondly, we trained this model with RAVNESS dataset. Thirdly, we trained our model with both datasets. Accuracies of each training session are in the Table II.

TABLE II. ACCURACY OF THE PROPOSED MODEL

Dataset	Accuracy	Precision	Baseline error
Cohn_kanade	0.88	0.90	11.83%
RAVNESS	0.92	0.96	6.63%
Cohn-Kanade&Ravness	0.92	0.94	7.49%

As we can see from Table III, happy and surprise is easily detected emotions. The reason is that during these emotions eyebrows and a mouth takes the special position as can be seen from Figure 4. During a surprise the mouth gets a rounded shape and the eyebrows get lifted. When a person is smiling, which also means a person is happy, the corner of the lips lift eyebrows stays stable.

TABLE III. CONFUSION MATRIX FOR COHN-KANADE AND RAVDESS

Actual Emotion	Predicted Emotion					
	Happy	Sad	Angry	Surprise	Disgust	Neutral
Happy	98	0	0	0	0	2
Sad	1	60	4	2	0	33
Angry	0	0	95	0	0	5
Surprise	0	0	0	100	0	0
Disgust	0	0	13	0	84	3
Neutral	0	2	3	0	3	92

Fig. 4. Surprise and Happy Faces from Cohn-Kanade dataset

Moreover, disgust is usually confused with anger and it is related to mouth and forehead. In both angry and disgust images represented in Fig. 5, frown lines appear on a forehead and the corners of a mouth get down. The only noticeable difference is the position of a nose.

The most confused emotion is sadness, and it is mostly mistaken with neutral emotion. The most noticeable difference between two emotions is the shape of a mouth as we can see from Fig. 6. However, the algorithm is not able to detect this feature explicitly.

Fig. 5. Disgust and Angry Faces from Cohn-Kanade dataset

Fig. 6. Neutral and Sad Faces from Cohn-Kanade dataset

From the above discussion, we can conclude that the model we propose detect emotions more easily when there are at least two distinguished features and the topmost shape of a face plays an important role in emotion recognition compared to the lower part of a face.

VI. CONCLUSION

In this paper, we propose a CNN model that is trained on Cohn-Kanade and RAVDESS datasets to find 5 major facial emotions. The uniqueness of our model is that it's not pretrained and performed well for both datasets with few neural layers.

The proposed, final model has outperformed the models that we've previously built, and it consists of 8 convolutional layers with the addition of pooling and dropout layers. The model gave good results on both datasets and it predicted surprise and happy emotions better in comparison to other emotions. Having our results obtained, several future research and development directions have been identified by the end of our work.

First of all, we are planning to add more life images to our training dataset since datasets we used in training process only contain images taken in a laboratory environment. We will add more such pictures since it will allow us to apply this application in real life scenarios.

Moreover, our datasets only consist of images where only the major version of facial emotions is depicted. To expand the scope of our application, we will train our model with minor facial expression since in daily life we use them more.

In addition, we are planning to transfer our model on a distributed platform in order to handle streams of images in a real (or a near-real) time that will allow us to find an industry application of our approach. Distributed platform will allow to train our model with a higher number of images and do it efficiently and effectively. One of the interesting application directions is to apply our model within the shopping mall environments in order to collect customer experience feedback based on their facial expressions.

REFERENCES

[1] X. Zhao, S. Zhang and B. Lei, "Facial expression recognition based on local binary patterns and local fisher discriminant analysis," WSEAS Transactions on Signal Processing, vol. 8, no. 1. pp.21-31, 2012.

[2] M. Pourebadi and M. Pourebadi, "MLP neural network-based approach for facial expression analysis," 2016.

[3] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., & Matthews, I. "The Extended Cohn-Kanade Dataset (CK+): A complete expression dataset for action unit and emotion-specified expression," Proceedings of the Third International Workshop on CVPR for Human Communicative Behavior Analysis (CVPR4HB 2010), San Francisco, USA, 94-101, 2010.

[4] Livingstone SR, Russo FA, "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English," PLoS ONE, 13(5): e0196391, 2018.

[5] Bradski G., "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2010.

[6] Borovykh A, Bohte S, and Oosterlee CW, "Conditional time series forecasting with convolutional neural networks," preprint arXiv:170304691, 2017.

[7] Simonyan K, Zisserman A, "Very deep convolutional networks for large-scale image recognition," ICLR 2015, San Diego, CA, 2015.

[8] Boureau, Y-Lan & Ponce, J & Lecun, Yann, "A Theoretical Analysis of Feature Pooling in Visual Recognition," ICML 2010 - Proceedings, 27th International Conference on Machine Learning, 111-118, 2010.