Facial Expression Classification Using Convolutional Neural Network and Real Time Application

Erdem Canbalaban
Hacettepe University
Ankara, Türkiye
n16223654@cs.hacettepe.edu.tr

Mehmet Önder Efe
Hacettepe University,
Ankara, Türkiye
onderefe@hacettepe.edu.tr

***Abstract* - Facial expression is an important feature that gives information about a person's psychological situation. People use these expressions while communicating and socializing and they have lots of information related to inner world of an individual. Therefore it is important to understand the meaning of these facial expressions automatically and use this information. This paper presents a method for facial expression classification with grayscale images from Kaggle Face Dataset with a Convolutional Neural Network and a real-time user interface in order to test the performance online. Data augmentation is used to increase the diversity of the samples. Neural network is created with Matlab and user interface is created via App Designer. Different training and fine-tuning techniques are employed in the design. The overall accuracy 61.8% is achieved across seven different facial expression categories with test dataset supplied in Kaggle domain.**

Keywords – Convolutional Neural Network, Data Augmentation, Real time testing, Training and Fine-Tuning Techniques

I. INTRODUCTION

Human emotions and facial expressions have quite important role for communication in the society and there are lots of research for understanding the relation between the emotions and corresponding facial expressions for decades. With the increase of the usage of social media and photo/video-based applications, the visual data in the Internet grows exponentially. Therefore, the community finds ways to use this data in different kinds of psychological and social experiments. Facial expression recognition is one of the ways that can be used in different kind of research areas such as psychological behavior examination, human-machine interactions and human services applications.

Convolutional neural networks are designed for taking advantage of the two dimensional structure of an input image. Therefore this method can fit better than other learning methods for facial expression recognition problem. Another benefit of CNN is that it is easier to train and have fewer parameters than fully connected networks.
There are two main features of this work. First, in this project, different regularization and data augmentation techniques are used with the convolutional neural network and the study has important role in order to observe that the effect of the configuration of the convolutional neural network layers and used fine-tuning and training techniques. Second, the real time test interface provides the researchers an opportunity to understand the convolutional neural network structure and test the trained network with online data in order to see the effect of the network and how successful the network with these configuration and training techniques.

This paper provides an overview of a designed method for facial expression classification and different techniques for data extraction, augmentation and classification based on CNN architecture. The rest of the paper is organized as follows: Section II provides background information related to facial expression classification with CNN structure from the literature, Section III describes the proposed method for facial expression classification and techniques that are used, Section IV gives an experimental results and discussion, Section V concludes the paper.

II. BACKGROUND

The idea of facial expression recognition starts with the increase of the reachable photo and video data. One of the first studies related to facial expression recognition is Empath from 2002 [1]. In this design, Gabor filtering is used on raw images with different transformations and Principal Component Analysis (PCA) and 3 layer neural net is applied. "Convolutional network" concept is proposed in 1980s by Fukushima [2]. Up to 1999, there were numerous studies related to convolutional networks and at the end, the standard reference for CNNs is proposed by LeCun [3]. Facial Action Coding System (FACS) [4] is one of the first psychological framework that describes the facial movements of a human. With the usage of action units, it classifies human facial movements by their appearance on the face. Convolutional Neural Networks are used for facial expression recognition with different purposes from early 2000s. In [5], CNN is used as a similar concept to the FACS model. Input set decomposed into features and with the usage of layered structure each layer has a more complex representation of the expression. At the end, final representation is used for facial expression classification. The difference between CNNs and FACS model is that if adequately many samples are shown to the network, then the network learns a general representation of a facial expression instead of memorizing a fixed structure.

Also, there are different facial expression recognition techniques that are used in the literature such as Bayesian Networks [6] or Multilevel Hidden Markov Model [7]. There is a drawback related to timing and accuracy of recognition in the mentioned studies. It is a solution to use more than one technique in the same study in order to

978-1-7281-3964-7/19/$31.00 ©2019 IEEE

increase the accuracy. Also, the data extraction and augmentation methods are also critical for accuracy.

III. PROPOSED METHOD

The proposed facial expression recognition method consists of three main parts. These parts can be seen in Figure 1. First part is preparing the dataset for training and data augmentation techniques for increasing the variety of the dataset. When the dataset is ready for training, CNN handles the feature extraction and feature classification. At the end, with the help of fully connected hidden layers and classification layer, classification can be performed according to the maximum probability.

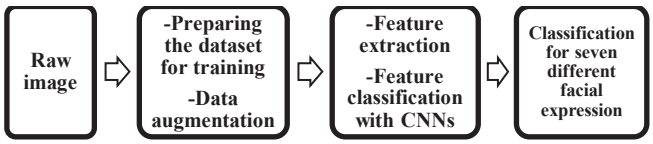

Fig. 1. Process Diagrams of the Proposed System

A. Preparing the Dataset and Data Augmentation

The first part of the proposed system is to prepare the data for training purposes and having a dataset which can represent the real world as much as possible. In this project, Kaggle Facial Expression Recognition [8] dataset is used for training and test purposes. The dataset consists of one training set and two test sets. Figure 2 shows sample images from the training dataset. Training set has 28709 different 48×48-grayscale images in 7 facial expression categories and each test set has 3589 images. The images in the dataset are located nearly center of the pixels and each image has same amount of face to image ratio which means the number of empty pixels are nearly same for the whole dataset. The number of images in the training set and corresponding labels are given in Table 1.

The data stored in a '.csv' file and each pixel is represented by a grayscale pixel value. Since the convolutional neural network needs an image itself, the conversion mechanism is implemented in order to construct each image separately and they are stored in seven different category according to corresponding facial expression label.

TABLE I. NUMBER OF FACIAL EXPRESSION SAMPLES

Facial Expression	Number of Sample in Training Set	Number of Sample in Test Set
Angry	3995	467
Disgust	436	56
Fear	4097	496
Happy	7215	895
Neutral	4965	607
Sad	4830	653
Surprise	3171	415
Total	28709	3589

Data augmentation is another important technique in order to have higher accuracies while training a network. The effect of low quality or unbalanced data in different classes can be resolved using various augmentation techniques. Rescaling, resizing, rotating, shearing and translation are mostly used techniques which can be employed for this purpose. Since the dataset size is quite large in this case, online augmentation is preferred for data augmentation. Instead of creating and storing the new image samples, transformations are performed at the beginning of the each mini-batch supplying to the network. In this method, rotation, shearing and translation techniques on two axes are used with random values from predefined intervals. Table 2 shows the details of the augmentation techniques.

Fig. 2. Sample Images from Training Dataset

TABLE II. DATA AUGMENTATION OPTIONS

Data Augmentation Technique	Augmentation Interval and Unit
Rotation	[-30, +30] degree
Shearing	[-20, +20] degree
Translation	[-5, +5] pixel

B. Feature Extraction and Feature Classification with CNNs

Convolutional Neural Networks have their own receptive fields and shared weights which can be used for extracting features. With the usage of these concepts, CNN starts to find patterns in an image. It tries to use every possible position and match similarities by getting across the whole image using filters. The propagation of information through the network is calculated by convolution.

The proposed CNN architecture can be seen in Figure 3. CNN consists of three convolutional layers with batch normalization, Rectified Linear Unit Layer (Relu) and max pooling layers. It continues with three fully connected hidden layers, Softmax error calculation and concludes with the classification layer.

Convolutional layers apply sliding convolutional filters with the size of 5×5 to the input. Each layer convolves the input with the filter and computes the product of the weights and the input. The number of filters increases while moving through the network.

Batch normalization layer normalizes input channel with respect to the mini-batch. The layer normalizes the activation of each channel by using mini-batch mean and standard deviation. Relu performs a simple threshold operation and each element, which has a value less than zero, become equal to zero. Max pooling layer is responsible for down-sampling with specified size and calculate the maximum of each region separately.

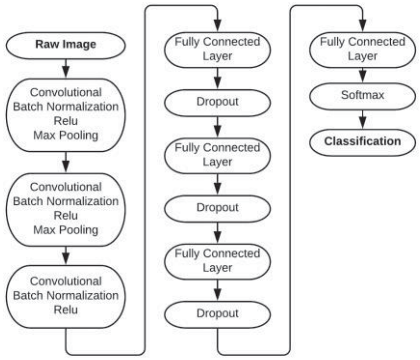

Fig. 3. The Proposed CNN Architecture

Fully connected hidden layer is specified with the number of neurons in the layer. There are three different layers in the design and we have Dropout layer between each fully connected layer with the probability of 0.5. This means that some neurons are dropped out randomly with the given probability. Their contribution to the activation is removed on the forward pass and the weights of these neurons are not updated during the backward pass as well. This technique introduces a better generalization and decrease the risk of overfitting the training data.

C. Classification of Facial Expression

Softmax Layer is applied to the last fully connected hidden layer output and use softmax function, which can be considered the multi-class generalization of the logistic sigmoid function. For classification problems, softmax layer and classification layer are used together at the end of the CNNs. A classification layer computes the cross entropy loss for mutually exclusive classes; therefore it is used in this method. The loss formula is given in (1).

$$\sum_{i=1}^{N}\sum_{j=1}^{K} t_{ij}\, ln\, y_{ij} \qquad (1)$$

where J is loss, N is the number of samples, K is the number of classes, t_{ij} is the indicator that i^{th} sample belongs to the j^{th} class and y_{ij} is the output for sample I for class j that is softmax layer output. It is also the probability of associating i^{th} input to the class j.

IV. Experimental Results

The proposed method runs on a computer which has Intel i7-4700 HQ CPU, 8GB RAM and GeForce GTX 765. MATLAB 2018b is used for development environment. The design is trained with the dataset that contains 28709 images validated and tested with a dataset with 3589 images. The options that are used while training the network are given in Table 3.

TABLE III. TRAINING OPTIONS

Training Properties	Training Option
Solver Name	ADAM
Initial Learning Rate	0.0003
Data Shuffling	Every Epoch
Hardware Resource	GPU
Learning Rate Dropping Factor	0.9
Learning Rate Dropping Period	5 Epochs
Maximum Number of Epochs	100
Size of Mini-batch	80
Validation Frequency	0.44 Epoch

In this problem, ADAM [9] optimizer is used instead of classic stochastic gradient descent algorithm in order to update network weights in training data. Instead of adapting the parameter learning rates averaging at first moment, ADAM uses the average of the second moments of the gradients. The algorithm calculates Exponential Moving Average (EMA) of the gradient and squared gradient and control the decay rates of these EMAs.

The number of total epoch is chosen 300 in order to observe the training and validation accuracy behaviors. The accuracy and loss relations with respect to the number of epoch can be seen in Figure 4. In Figure 4, top subplot shows the training and test accuracy for 300 epoch. The lines which have larger deviation represent the training accuracy whereas the dots show the test accuracy. The bottom subplot represents the loss of both training and test data. The lines, which display larger deviation, show the loss of each mini-batch and dots are for test data.

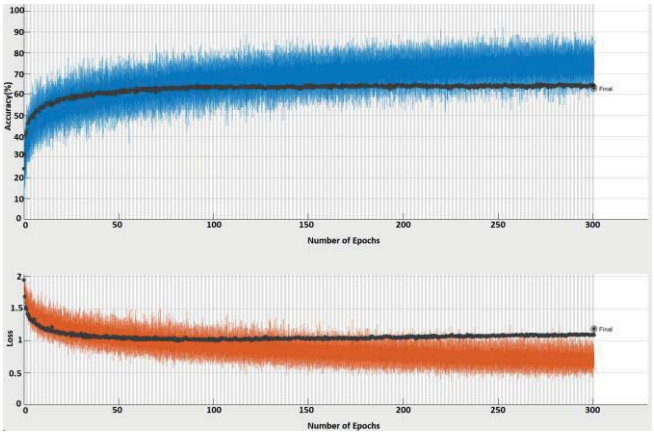

Fig. 4. Training the Network for 300 Epochs

It is clear from the training result that validation accuracy becomes saturated around 50th epoch and after that point, loss of the test data starts to increase and the training accuracy also increases. In order not to face with the overfitting of the data, the training is repeated with the same options and 50 epochs.

Figure 5 shows the results of the second training. In Figure 5, top subplot shows the training and test accuracy for 50 epoch. The lines which have larger deviation represent the training accuracy whereas dots show the test accuracy. The bottom subplot represents the loss of both training and test data. The lines, which have larger deviation, show the loss of each mini-batch and dots are for test data.

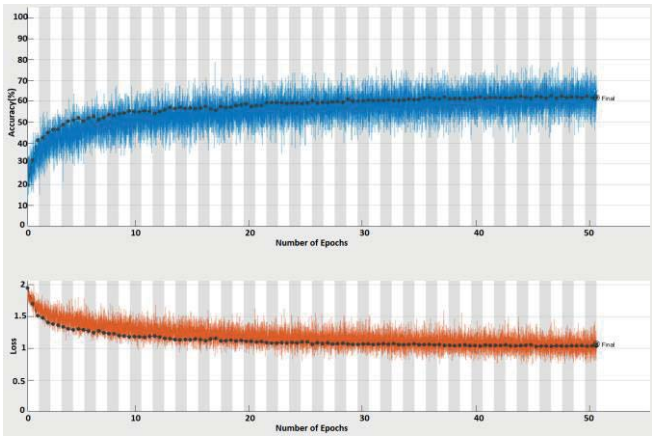

Fig. 5. Training the Network for 50 Epochs

The test results and detailed information related to training can be seen in Figure 6.

Results	
Validation accuracy:	61.80%
Training finished:	Reached final iteration
Training Time	
Start time:	19-Jan-2019 16:08:25
Elapsed time:	72 min 15 sec
Training Cycle	
Epoch:	50 of 50
Iteration:	17900 of 17900
Iterations per epoch:	358
Maximum iterations:	17900
Validation	
Frequency:	160 iterations
Patience:	Inf
Other Information	
Hardware resource:	Single GPU
Learning rate schedule:	Piecewise
Learning rate:	0.00011623

Figure 6. Results of the Training for 50 Epochs

As it can be seen in Figures 5 and 6, test accuracy is around 61.8% with this configuration of the network. The dataset consists of 28709 images which are classified into seven different labels. When the images are investigated closely, it is observed that each class has some images that can be labeled for other groups of facial expressions also. In other words, in the dataset there are some images which can be multiple-labeled. Due to this fact, it should be understood that the training accuracy cannot reach 100%.

Data augmentation is an important technique which brings higher percentage of accuracy while training the network. Without data augmentation trained network perform 57% test accuracy. Data augmentation techniques that are applied in the final design increase the accuracy up to 61.8% since they diversify the training dataset by adding extra features to the samples.

Once a stable accurate CNN is obtained with this configuration, it is tested also online with real time data from webcam data. Since the resolution of the webcam is much higher than the network predefined input size, the conversion of the captured RGB image to the 48×48 grayscale image is performed first. The problem here is that, once the webcam image is converted to the small size grayscale image, the ratio of face size to image size is not same with the dataset sample images. Because of this difference, real-time tests fail at the first trial. In order to overcome this issue, before the conversion of the webcam image to the small size, face extraction method is developed and applied to the webcam output. Once the face is cropped from the original image and the ratio of face size to image size is adjusted, the conversion is made and the sample image is achieved. With this configuration, the proposed CNN structure and trained network is tested also with real-time data. Sample images and corresponding facial expression labels can be seen in Figure 7.

Fig. 7. Webcam Images from Real Time Test with Corresponding Results

According to the results seen in Figure 7, for the image on top left corner, the algorithm evaluates a probability of 0.5 for label ‘surprise’, which is the highest probability compared to the other labels.

During the tests with real time data, it is observed that some of the facial expression types can be identified easily whereas some of them can be mixed to each other. Actually, this is because of the nature of these facial expression types. Although they represent different emotions, the characteristics of them are similar on the face.

With these results, proposed solution outperforms other methods which use CNN for training. In 2016, Raghuvanshi [10] proposed a solution based on Kaggle facial expression dataset and achieves 48% test accuracy. In 2017, Lekhak [11] proposed a solution with same dataset and achieves 56.77% test accuracy. Usage of well-designed Convolutional Neural Network and different data augmentation techniques increases the accuracy of the proposed method.

V. Conclusion and Future Work

In this study, CNN based facial expression recognition method is designed with Kaggle Facial Expression Recognition Dataset and an accuracy of 61.8% is achieved in the classification of seven different categories. In order to have a proper dataset, preprocessing and data augmentation techniques are applied to the raw grayscale image data. Convolutional neural network with 20 layers are used and different types of facial expressions are classified.
In design process, we have faced different difficult problems and overcome each of them by applying required solution steps. Although the resolution of the sample images are quite low, from only 48×48 pixels, we managed to design a convolutional neural network which is capable of classifying seven different group effectively both offline testing and with real time data.

To further improve the performance, we will apply different feature extraction methods with more powerful machine and increase the size of the images; we will try to achieve higher accuracies at the end.

VI. References

[1] M. N. Dailey, G. W. Cottrell, C. Padgett, and R. Adolphs, "Empath: A neural network that categorizes facial expressions," *J. Cogn. Neurosci.*, vol. 14, no. 8, pp. 1158–1173, 2002.

[2] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biol. Cybern.*, vol. 36, no. 4, pp. 193–202, 1980.

[3] Y. LeCun, "Object Recongnition with Gradient-based Learning," *Shape, Contour Group. Comput. Vis.*, vol. 91, no. 0, pp. 399–404, 2017.

[4] P. Ekman and W. V Friesen, "Facial action coding system: A technique for the measurement of facial movement.," *CA Consult. Psychol. Press. Ellsworth, PC, Smith, CA (1988). From Apprais. to Emot. Differ. among unpleasant Feel. Motiv. Emot.*, vol. 12, pp. 271–302, 1978.

[5] H. Da Cunha Santiago, T. I. Ren, and G. D. C. Cavalcanti, "Facial expression Recognition based on Motion Estimation," in *Proceedings of the International Joint Conference on Neural Networks*, 2016, vol. 2016-Octob, pp. 1617–1624.

[6] M. Singh, A. Majumder, and L. Behera, "Facial expressions recognition system using Bayesian inference," in *Proceedings of the International Joint Conference on Neural Networks*, 2014.

[7] M. Schmidt, M. Schels, and F. Schwenker, "A Hidden Markov Model Based Approach for Facial Expression Recognition in Image Sequences," in *Artificial Neural Networks in Pattern Recognition*, 2010, pp. 149–160.

[8] I. J. Goodfellow *et al.*, "Challenges in representation learning: A report on three machine learning contests," *Neural Networks*, vol. 64, pp. 59–63, 2015.

[9] R. Shindjalova, K. Prodanova, and V. Svechtarov, "Modeling data for tilted implants in grafted with bio-oss maxillary sinuses using logistic regression," *AIP Conf. Proc.*, vol. 1631, pp. 58–62, 2014

[10] A. Raghuvanshi and V. Choksi, "Facial Expression Recognition with Convolutional Neural Networks," *CS231n Course Proj.*, 2016.

[11] D. Lekhak, "A Facial Expression Recognition System Using Convolutional Neural Network," *Tribhuwan University, Institute of Engineering.,*