Facial Expressions Recognition using Computer Vision

Ahmed Elgammal
Rutgers University
Piscataway, NJ, USA
Email: elgammal@cs.rutgers.edu

Fatima AlSaadeh
Rutgers University
Piscataway, NJ, USA
Email: fatima.alsaadeh@rutgers.edu

Naveen Narayanan Meyyappan
Rutgers University
Piscataway, NJ, USA
Email: nm941@scarletmail.rutgers.edu

***Abstract*— In this era of remote and digital communication, understanding the speaker's emotional status is becoming a crucial task; it can help the listener evaluate the situation and make a better decision, whether this listener is a doctor, professor, or manager. This paper is trying to extract human facial expressions using computer vision techniques and compare its performance to this field's previous work.**

I. Introduction

Facial expressions are one or more motions of the muscles beneath the face's skin in response to the person's internal emotional state [1]. Analyzing these emotions is becoming more critical with the rising use of digital communication tools to conduct educational, business, and medical meetings. Still, this research topic had been active since Darwin's The Expression of the Emotions in Man and Animals in 1872 [2]. After much research, the affective computing term appeared in 1995 when Rosalind Picard wrote that humans' emotions could be explored and analyzed. With the popularity of machine learning and artificial intelligence, computer vision research started to focus on this topic. Classification models were applied to solve this problem, support vector machines (SVM), neural networks (NN) provided successful results. Convolutional neural networks (CNN), on the other hand, appeared to overcome various limitations providing an end-to-end process by feeding the raw facial images as an input to the model. In this paper, we are trying to take a more in-depth look into these approaches and develop a model that can help digital tools analyze facial expressions and help the listener evaluate the situation.

II. Prior Work

To get a better understanding of the recent developments and advancements in computer vision regarding Facial Expression Recognition (FER), we reviewed papers published in International conferences during the last decade. The first paper that we reviewed was "Facial Expression Recognition Using Computer Vision: A Systematic Review" by Daniel Canedo and Antonio J. R. Neves [3]. The authors conducted a systematic review on more than 500 International papers published in computer vision to solve Facial Emotion Recognition (FER). This paper reviews the popularly used methods and techniques for Facial Expression Recognition and various computer vision and machine algorithms used in FER, and the most commonly used FER databases. The paper concludes by discussing the results achieved from prior work. We also examined four more articles related to this topic. "Development of deep learning-based facial expression recognition system" [4] provides a comparison of FER experiments implemented using Deep Neural Network and Convolution Neural Network. In "Real time face detection and facial expression recognition: Development and applications to human computer interaction" [5] introduces a novel method for real-time FER using Adaboost and SVM. Development of deep learning-based facial expression recognition system [6] applies a deep learning method to perform FER.

III. Datasets

In this project we are experimenting on two main datasets Karolinska Directed Emotional Faces Database and The Extended Cohn–Kanade Database:

- **Karolinska Directed Emotional Faces Database (KDEF)**: is a set of totally 4900 pictures of human facial expressions. The set of pictures contains 70 individuals (35 males and 25 females) displaying 7 different emotional expressions (afraid, angry, disgusted, happy, neutral, sad, surprised). Each expression is viewed from 5 different angles and in this experiment, we are using the straight angle ending up with 840 pictures. [7]
- **The Extended Cohn–Kanade Database (CK+)**: contains 981 pictures of human facial expressions (males and females) displaying 7 different emotional expressions (happy, contempt, fear, surprise, sadness, anger, disgust). [8]

IV. Approach, Challenges, and Improvements

A. Approach

1) ***The Appearance Feature-Based Network Model****:* [9] This approach started by applying a smoothing data preprocessing technique, a filtering technique to reduce noise while capturing relevant patterns; one smoothing technique is a bilateral filter we use in this experiment. Bilateral filtering is a nonlinear filtering technique that smooths the image while preserving its edge, where a weighted average of its neighbors replaces each pixel. It is important in FER systems as it helps to preserve the eyes, mouth, nose edges from blurring, which are important parts to detect emotions.[10]

After that, we extracted the features using Local binary pattern which is a texture processing method that describes the local texture patterns of an image. This method works in a block size of 3x3. The center pixel is used as a threshold for the neighboring pixel. The LBP code of a center pixel is generated by encoding the computed threshold value into a decimal value. [11]

$$LBP = \sum_{i=0}^{P-1} s(p_i - G_c)2^i$$

$$s(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Where P is the number of neighborhood pixels, p_i represents the ith neighboring pixel, and c represent the center pixel, G_c is the value of the center pixel. Figure 1 is an example of LBP operation with G_c is equal to 12.

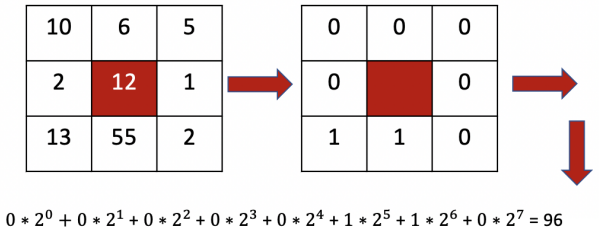

Fig. 1. LBP Operation

In this experiment we are using radius of 3 and 24 number of points to achieve the LBP operation on the images.

In "Efficient Facial Expression Recognition Algorithm Based on Hierarchical Deep Neural Network Structure" paper, the authors are implementing what is called the appearance feature-based CNN network, which is a process of extracting the holistic features of the face. [9] The State-of-the-Art Convolutional Neural Networks are mainly used in computer vision and help detect underlying patterns from the images during the learning process. Convolutional neural networks have neurons with adjustable weights and biases. It applies dot products between the weights and the input generating a map of features and having an activation function to get the convolution layers' results. The CNN network takes the processed images as input and is built of different convolutional, pooling, and fully connected layers; this layer will extract the features from the input generating features maps by multiplying the filters by the input features. In the formula below, the feature map x at layer l, which is generated from j filters w and bias matrix b.

$$x_i^l = \sum_{n=1}^{j} w_{i,j} * x_i^{l-1} + b_i^l$$

On the other hand, the pooling layer performs a downsampling operation to reduce each feature map's dimensions keeping the critical information. The fully connected layer applies softmax to compute the classification scores. The appearance feature-based network model implemented by the authors and replicated by us has a 128x128 size image input; the input will go through convolutional and pooling layers three times, followed by two fully connected layers, and finally pass through softmax generating the output. The convolutional layers are of 5x5 kernel size, the max-pooling layers selecting a pixel from pixels in 2x2 blocks. The authors set the kernels' size and layers parameters, reducing the input size by half. After these layers, the process is followed by flattening the results and inserting them into the two fully connected layers, one with 1024 nodes and the second with 500 nodes. They also used a dropout between the fully connected layers so the network turns off the neurons randomly to avoid overfitting. The rectified linear unit (ReLU) was used as the activation function between the convolutional and the fully connected layers and softmax to get the final classification scores. The steepest gradient descent (SGD) was used as an optimizer with a 0.01 learning rate and the categorical cross-entropy loss. The model architecture is shown in Figure 2, the figure is taken from the original paper we are trying to replicate here. [9]

*2) **Hybrid Deep Learning Neural Approach**:* [12]

In this appraoch, we used the Viola-Jones method to detect the faces on the CK+ and KDEF dataset. The faces were detected from all images, and the images were cropped and resized to the size of 120 x 110.

The paper discuss two method of feature extraction: Gabor Filter and Deep Convolutional Neural Network (CNN). The extracted features were classified using two methods: Support Vector Machines (SVM) and Multi Layer Perceptron (MLP). All the models were implemented and tested on both the datasets using a train-test split of 0.8 and 0.2 respectively. Gabor filters are a special type of filter for feature extraction. The input images were convoluted with the filter response of the Gabor filter bank kernel. We used the same specifications as mentioned on the paper to model the Gabor filter bank. The Gabor filter bank response is given by:

$$g_{\lambda,\theta}(x,y) = \exp\left[-\frac{1}{2}\left\{\frac{x_{\theta n}^2}{\frac{2}{x}} + \frac{y_{\theta n}^2}{\frac{2}{y}}\right\}\right]\cos(2\pi * \theta n * \lambda)$$

where

$$x_{\theta n} = x(\sin\theta n) + y(\cos\theta n)$$

$$y_{\theta n} = x(\cos\theta n) + y(\sin\theta n)$$

$$\|g_{\lambda,\theta}(x,y)\| = \sqrt{R^2\{g_{\lambda,\theta}(x,y)\} + I^2\{g_{\lambda,\theta}(x,y)\}}$$

The parameter values mentioned on the paper for high performance is: $\theta = \frac{2\pi}{3}$, $\lambda = 6$, $\gamma = 0.5$, $\sigma = 0.4$

where R represents the real component and I represents the imaginary component.

The other method to extract features from images is to use

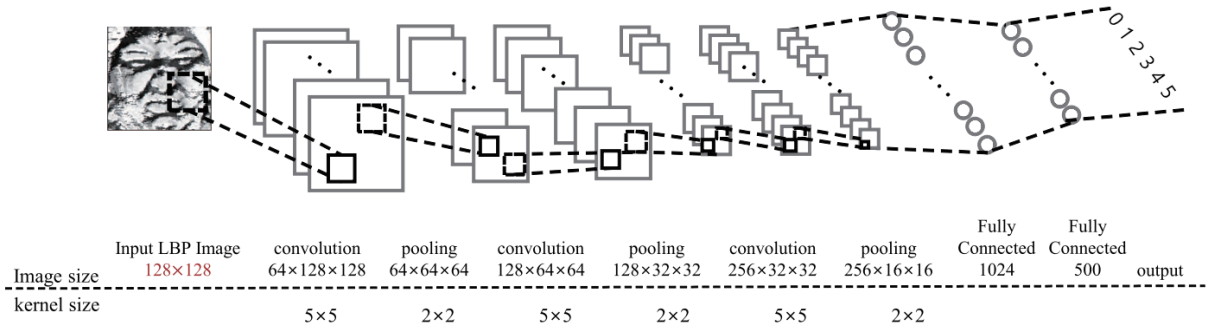

Fig. 2. The proposed appearance feature-based CNN structure from the Efficient facial expression recognition algorithm based on hierarchical deep neural network structure. [9]

convolution neural networks. CNN are neural networks that extract features using the spatial information from the images. We replaced Gabor filters in the previous section for features extraction with a Convolution Neural Network to extract features from each image. Four convolution layers were built, each with 20, 40, 60, and 30 filters. The filter sizes were 5 x 5 for the first two layers and 3 x 3 for the last two layers. All the layers used the same ReLU activation function and Batch Normalization. The output at each layer was max pooled with strides of 2 x2 and kernel size of 2 x 2 with zero paddings set to 1. The initial learning rate was set to 0.1 and decay of 0.01. The number of iterations or epochs was set to 100. Once the features were extracted from each image, we divided the data set into train and test splits in the ration of 80%-20% respectively.

The models used by Garcia, Elshaw, Altahhan, and Palade for classification was SVM and MLP. Firstly, The extracted feature vectors were classified into into 6 categories using Support Vector Machines (SVM). As per the paper's specifications, the SVM used linear kernels and a one vs. one approach to classify the images. The value of c was set to 1000 for the SVM. The results of this approach on the CK+ and KDEF data set is presented on section V.2. The MLP was composed of 100 neurons with 1 hidden layer and 1 output layer. The output layer used 7 neurons and a softmax function to classify the output as 1 of the 7 images. The results are discussed in section V.2.

B. Challenges and Improvements

The two models we implemented in this project; the Appearance Feature-Based Network Model, Hybrid Deep Learning Neural Approach, gave high accuracy (above 90% on both train and test splits) on the explained datasets (KDEF and CK+, this case only if both the train and test datasets are taken from the same database and give a lower accuracy 20%-50% if the model is trained on one dataset and tested on another. To address this issue, we reviewed our pre-processing stages. We experimented with cross-validation instead of the holdout validation technique to obtain higher accuracy on different train-test set combinations. We found that the KDEF dataset images include the actor background in the image, unlike the CK+. We implemented the face detection preprocessing technique to solve this problem using Haar feature-based cascade classifiers. This is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features." [13] Fig 3 shows one of the KDEF images before and after the face detection preprocessing technique. After

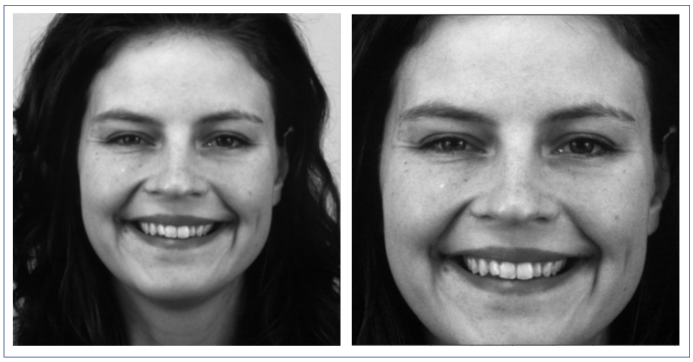

Fig. 3. KDEF image before and after face detection technique

that, we used k-fold Cross-Validation on both datasets with k = 5. These modifications gave us an accuracy of 96.38% when training on both datasets and testing on one of them. On the other hand, training the model on one dataset and testing on another still gave us a low accuracy due to the low number of images we had in the datasets.

V. Results

*1) **The Appearance Feature-Based Network Model** :* [9] We described in IV-A1 the model architecture - CNN section the authors model. We replicated this model, then trained and tested it on two datasets: KDEF [7] and CK+ [8], Section III describe these two datasets in details. After processing the images using bilateral filtering, extracting the features using local binary patterns, we standardized the features

$$standardized = \frac{features - \mu}{\sigma}$$

Next, we encoded the labels using the one-hot encoder; because the two datasets have six emotions in common, we used only these common emotions. Giving an index to each emotion happy: 0, fear: 1, surprise: 3, sadness: 4, anger: 5, disgust: 6, the one-hot encoding gave the label [1, 0, 0, 0, 0, 0] for the image with happy emotion. For the evaluation method, we used the holdout approach where we split the data into three sets 80% training, 20% of the training set is for validation and 20% of the data for testing. Figure 4 shows the training and validation accuracy and loss for CK+ dataset. Figure 5 is showing the confusion matrix for the model predictions on the same dataset with 96.77% accuracy.

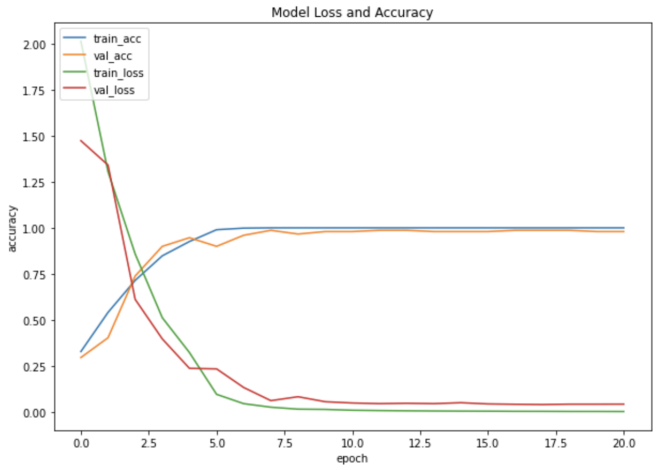

Fig. 4. The appearance feature-based CNN loss and accuracy on CK+

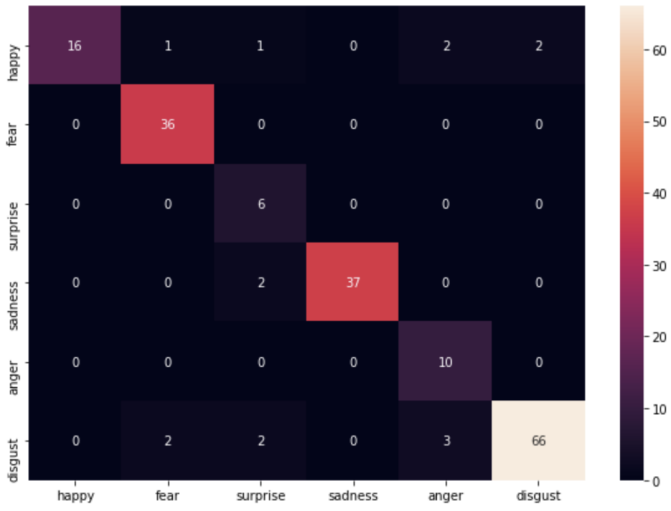

Fig. 5. The appearance feature-based CNN confusion matrix on CK+

The table below shows a comparison between the two datasets' accuracies which is similar to what the original paper achieved as the authors achieved 96.46% accuracy on testing. We experimented with training the model on a dataset and testing it on another. It gave us a lower accuracy of around 50% when we used the model to train on KDEF and test in CK+.

TABLE I
CK+ AND KDEF ACCURACY RESULTS

Dataset	Training	Validation	Testing
CK+	100.0%	97.9	96.77%
KDEF	100.0%	94.07	94.05%

*2) **Hybrid Deep Learning Neural Approach** :* [12] We implemented the different models on the JupyterHub installed in Rutgers ilab clusters. The model was developed using the opencv2 library in python 3 and the following results were achieved on the CK+ and KDEF datasets.

TABLE II
CK+ AND KDEF ACCURACY RESULTS

Dataset/Model	CK+	KDEF
Gabor-SVM	92.41%	95.58%
Gabor-MLP	94.46%	93.5%
CNN-SVM	96.34%	96.26%
CNN-MLP	92.26%	91.16%

VI. DISCUSSION AND CONCLUSIONS

In this project, we implemented two different novel models for Facial Expression Recognition (FER); the Appearance Feature-Based Network Model, Hybrid Deep Learning Neural Approach. All the discussed models gave high accuracy (above 90% on both train and test splits) on the explained datasets (KDEF and CK+). Using both datasets in training and testing the model at the same time gave us a high accuracy (96%) after applying face detection and k-fold cross-validation. Future work will involve introducing more images from different datasets and the internet to predict the correct emotion for new images the model wasn't trained on before.

REFERENCES

[1] A. J. F. (1994), "Human facial expression (1 ed.)."

[2] (2004) Chapter 11. facial expression analysis. [Online]. Available: "http://www.cs.cmu.edu/~cga/behavior/FEA-Bookchapter.pdf"

[3] D. Canedo and A. Neves, "Facial expression recognition using computer vision: A systematic review," 2019.

[4] H. J. et al., "Development of deep learning-based facial expression recognition system," 2015.

[5] I. F. M. S. Bartlett, G. Littlewort and J. R. Movellan, "Real time face detection and facial expression recognition: Development and applications to human computer interaction." 2003.

[6] H. Jung, S. Lee, S. Park, B. Kim, J. Kim, I. Lee, and C. Ahn, "Development of deep learning-based facial expression recognition system," pp. 1–4, 2015.

[7] Kdef and akdef. [Online]. Available: "https://kdef.se/download-2/index.html"

[8] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 2010, pp. 94–101.

[9] J.-H. Kim, B.-G. Kim, P. P. Roy, and D.-M. Jeong, "Efficient facial expression recognition algorithm based on hierarchical deep neural network structure," *IEEE Access*, vol. 7, pp. 41 273–41 285, 2019.

[10] Bilateral filtering. [Online]. Available: "https://people.csail.mit.edu/sparis/publi/2009/fntcgv/Paris_09_Bilateral_filtering.pdf"

[11] N. Sairamya, L. Susmitha, S. Thomas George, and M. Subathra, "Chapter 12 - hybrid approach for classification of electroencephalographic signals using timeâfrequency images with wavelets and texture features," in *Intelligent Data Analysis for Biomedical Applications*, ser. Intelligent Data-Centric Systems, D. J. Hemanth, D. Gupta, and V. Emilia Balas, Eds. Academic Press, 2019, pp. 253–273. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780128155530000136

[12] A. Ruiz-Garcia, M. Elshaw, A. Altahhan, and V. Palade, "A hybrid deep learning neural approach for emotion recognition from facial expressions for socially assistive robots," *Neural Computing and Applications*, vol. 29, no. 7, pp. 359–373, 2018.

[13] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, vol. 1, 2001, pp. I–I.