Human Emotion Analysis using Convolutional Neural Network

Punit Prajapati
Department of Information Technology
Vidyavardhini's College of Engg. and Tech.,
University of Mumbai Mumbai, India

Sumil Jain
Department of Information Technology
Vidyavardhini's College of Engg. and Tech.,
University of Mumbai Mumbai, India

Parmar Tarun
Department of Information Technology
Vidyavardhini's College of Engg. and Tech.,
University of Mumbai Mumbai, India

***Abstract*—Human Expressions play a major role in providing subtle details in any face to face conversation. These details are easily understood by humans. If such emotions were also to be recognized by machines too, then Human Machine Interaction (HMI) would be robust. This paper proposes a method to fulfill such purpose using Convolutional Neural Network. The proposed method in this paper comprises of two channels of Convolutional Neural Network- first channel takes input of face image where second channel is provided with facial parts. The result of both the channels is then combined at fully connected layers to generate a better prediction. This method generates a better accurate result by combining local features instead of all global features simultaneously.**

***Index Terms*—Facial expressions, Facial Emotions, Non- Verbal Communication, Face Detection, Convolutional Neural Network (CNN), Facial Parts, Human Emotion Recognition.**

I. Introduction

Facial emotion recognition (FER) is an significant topic when considered fields like computer vision and artificial intelligence. Expressions analyzed while having an interpersonal communication can be proved as a source of information. Expressions such as fear, happiness, anger, sadness, disgust and surprise are termed as basic expressions. Recognizing such expressions is challenging task. Just by analyzing someone's expression, one can get a good information regarding the kind of mood the person is in. Humans convey information through many sources. Information can be conveyed verbally or non-verbally . Facial expression provides a great source of information in non-verbal mode of communication. Some sentences may not be interpreted correctly if not provided with facial expression information. Hence a combination of both -verbal information as well as non-verbal information helps in better evaluation of the information. One of the most impressive forms of ANN architecture is that of the Convolutional Neural Network (CNN). CNNs are primarily used to solve difficult image-driven pattern recognition tasks and with their precise yet simple architecture. We use two channels of Convolutional Neural Network in this paper to extract the face landmarks for predicting better results. For traning the model and testing, we will be using FERC2013 [1] dataset which has found to be having accuracy of 75.2, according to the paper ”Facial Expression Recognition using Convolutional Neural Networks: State of the Art” [2], to make our model more accurate.

II. Related Work

A. A New Approach for Automatic Face Emotion Recognition and Classification Based on Deep Networks

[3] This paper helps in building an artificial intelligence which focuses on recognizing the human facial emotions through their expression. Three convolution layers are used in the network mentioned in this paper and these layers are followed by max pooling and Relu layer. Training dataset for the network was FER2013 dataset and for testing, RaFD dataset was used to give a good range of images for training, so that it can overcome the basic problem of recognition of unknown faces.

B. Facial Emotion Analysis using Deep Convolution Neural Network

[4] In this paper, author has focused on the face intensity from low to high level of emotion to provide a better approach in predicting facial expressions by using Convolutional Neural network (CNN). Same as above paper, the FERC2013 dataset is used for training. The accuracy ovided by the system gives a good inspiration for future work in this field.

C. Deep Convolutional Neural Network for Facial Expression Recognition using Facial Parts

[5] In this work, author focused on three main things - expression recognition, feature extraction and the emotion classification and use combination of algorithms for the purpose. A two-channel convolutional neural network is used in this paper.Information from both channels are merged in fully connected layer for increasing the accuracy. Experiments are carried out on the Japanese Female Facial Expression (JAFFE) and the Extended Cohn-Kanada (CK+) datasets to determine the recognition accuracy for the proposed FER system.

D. Facial Expression Recognition using Convolutional Neural Networks: State of the Art

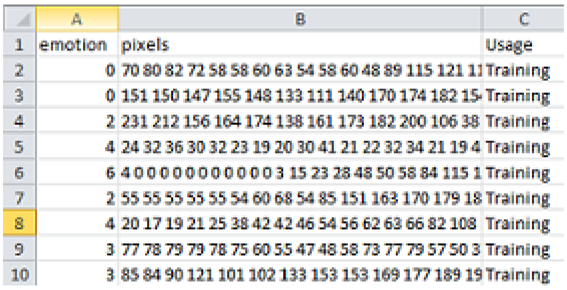
[2] In this paper, author focused on recognising emotions based on images using convolution neural network. The author performed deep convolution neural network and obtained accuracy of 75.2 percentage, outperforming previous works without requiring auxiliary training data or face registration.

E. Real-time Algorithms for Facial Emotion Recognition: A Comparison of Different Approaches

[6] In this paper, author present a comparison of five different approaches for real-time emotion recognition of four basic emotions (happiness, sadness, anger and fear) from facial images. They compared three deep-learning approaches based on convolutional neural networks (CNN) and two conventional approaches for classification of Histogram of Oriented Gradients (HOG) features: 1) AlexNet CNN, 2) commercial Affdex CNN solution, 3) custom made FER-CNN, 4) Support Vector Machine (SVM) of HOG features, 5) Multilayer Perceptron (MLP) artificial neural network of HOG features. The result of real-time testing of five different algorithms on the group of eight volunteers is presented.

III. Dataset Description

The choice of images used for training is responsible for a big part of the performance of the eventual model. This implies the need for a both qualitative and quantitative dataset. For emotion analysis there are several datasets available for training and testing the model which contain more than thousand of images captured in low or high resolution, at different brightness, etc. As mentioned in many papers (references of some papers) we have also used FERC2013 dataset for our training and evaluating purpose. Fer2013 is a challenging dataset. There are 32000 low resolution face pictures containing dissimilar age groups.



	A	B	C
1	emotion	pixels	Usage
2	0	70 80 82 72 58 58 60 63 54 58 60 48 89 115 121 11	Training
3	0	151 150 147 155 148 133 111 140 170 174 182 15	Training
4	2	231 212 156 164 174 138 161 173 182 200 106 38	Training
5	4	24 32 36 30 32 23 19 20 30 41 21 22 32 34 21 19 4	Training
6	6	4 0 0 0 0 0 0 0 0 0 0 0 0 3 15 23 28 48 50 58 84 115 1	Training
7	2	55 55 55 55 55 54 60 68 54 85 151 163 170 179 18	Training
8	4	20 17 19 21 25 38 42 42 46 54 56 62 63 66 82 108	Training
9	3	77 78 79 79 78 75 60 55 47 48 58 73 77 79 57 50 3	Training
10	3	85 84 90 121 101 102 133 153 153 169 177 189 19	Training

Fig. 1. FERC2013 Raw Dataset

IV. Implementation

There are two phases of the proposed alorithm - testing and training. The training is done so that network is able to classify the emotions properly of the provided face images. The first step of proposed algorithm is to verify if the trained data/model is already present or not. If it isn't, then we need to train the system first, to execute the next step which is testing of emotion classification else we would have gone directly for testing. Another part is the image pre-processing, in this we make the suitable input to train the model by using certain python library. Last but not the least we use deep convolution network to classify the image according to the trained model and predict the seven basic emotion. The following figure shows such a flow.

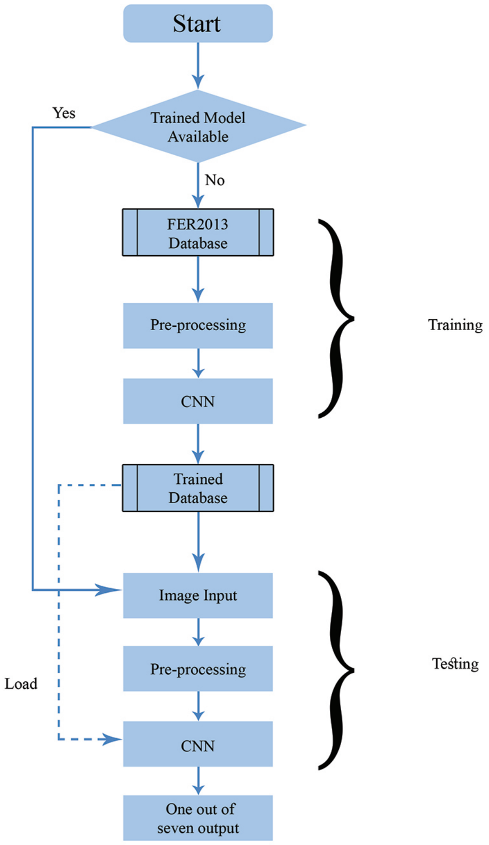


Fig. 2. Flow chart of proposed system

Following are the four major steps followed.

A. Pre-processing

Usually, the purpose of using preprocessing steps in face detection system is to speed up the detection process and reducing false positives. The dataset is basically rows of individual image features such as label, actual pixel values and the emotions. Hence, this info need to be converted to serve as an input to our next step, that is, training. Training step will take images as an input.

The CNN will take image array input of size 48x48. This objective is completed in this pre-processing step.
Following image is an example of output of this step. The pixel values from dataset is taken as an input and converted to jpegs of size 48x48. Also arrays are created to serve as an input to our convolutional neural network.

Fig. 3. Pre-processing Step Output

B. Training

The network is programmed with the use of the TFLearn [10] library on top of TensorFlow, running on Python. This environment reduces the complexity of the code as neuron layers are created instead of single neurons. The advantage of this set up is that we can get real-time feedback on the training progress and accuracy, which increases the reusability of the trained mode.

Before training, images from FERC2013 dataset are preprocessed, in the pre-processing, 28709 samples are used, after validating we got 11,246 valid samples for training.

1) Network: In recent times, convolutional neural networks (CNN) have confirmed inspiring performance in plentiful computer vision tasks. A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. According to our classification requirement, CNN architecture would serve a great benefit and this will help us to achieve better accuracy. Following is the layer by layer explanation of the architecture.

2) Layers of the Network:

- **Layer 0** Input layer contains the input image with dimensions 48x48x1
- **Layer 1** The Convolutional layer then is used to calculate the a dot product
- **Layer 2** RELU layer will be applied then as an activation function, such as the max (0, x) zero. The size remains unchanged ([44x44x64]),and batch normalization is done.
- **Layer 3** To perform down sampling we use Max pool layer that results in reduction of dimension to [22x22x64]. Size of filter is [3x3] and stride is 2,hence (44-3)/2+1=22 is output size, depth remains constant (64)
- **Layer 4** Again performing convolution using 64 filters, size 55, stride 1, size becomes [18x18x64],i.e. (22-5)/1+1=18; is size of output 64 depths because of 64 filters.
- **Layer 5** Max Poling Layer with 64 filters, size 55, stride 1,now size is [18x18x64],i.e. (18+2*1-3)+1=18 original size is restored.

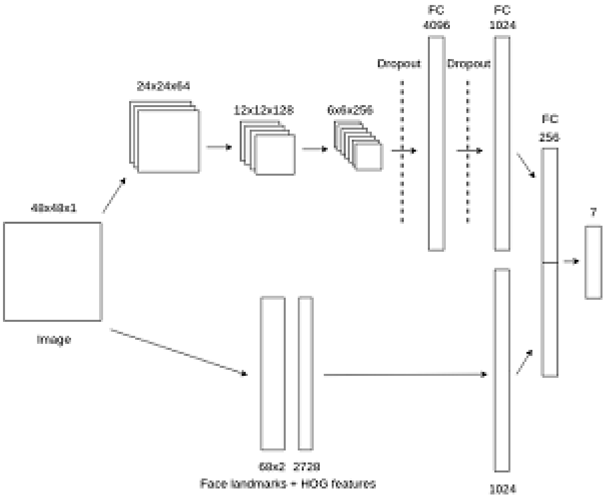


Fig. 4. System Architecture

- **Layer 6** Convolution with 128 filters of size 4x4 and stride 1 and we used padding 0,therefore now size is given as [15x15x128], i.e. (18-4)/1+1=15, is size of output 64 and depths of 128 filters.
- **Layer 7** Fully cconnected with 3072 neurons. In this layer, each of the 15x15x128=28800 pixels is fed into each of the 3072 neurons and weights determined by back-propagation.
- **Layer 8** Fully-connected layer calculates the class scores, resultant volume of size [1x1x7], where each of the seven numbers correspond to a class score, such as among the seven classes of emotions. As with normal neural networks and as the name implies, each neuron in this layer will be linked to all the numbers in the previous volume and soft max layer with 3072 neurons.
- **Layer 9** Soft max layer with 7 neurons to predict 7 classes output. Since the outputs of a softmax function can be interpreted as a probability (i.e.they must sum to 1), a softmax layer is typically the final layer used in neural network functions. It is important to note that a softmax layer must have the same number of nodes as the output later.

C. Optimizing

A kind of configuration which is provided to the model, which is external to data and one cannot determined its value from data are hyperparameters.They are often used in processes to help estimate model parameters. Hyperparameters are basically tuned when experimenting with machine learning models. This parameters are tuned such that the prediction accuracy increases.

1) keep probability: During the training of the model, the dropout rate needs to be controlled, hence keep probability is used. Keep probability helps in avoiding overfitting by means of keeping the connection between each layer with probability of, say 0.5.

TABLE I
HYPER PARAMETERS

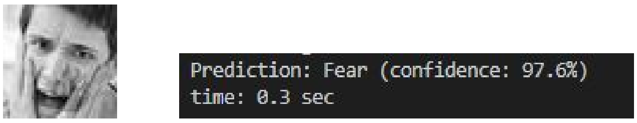
Parameters	Values
Keep Probabilitys	0.956
Learning Rate	0.061
Learning rate Decay	0.864
Decay Step	50
Optimizer	momentum
Optimizer parameter	0.95

2) Learning Rate: Learning rate decides with how much pace are we travelling downwards over the slope. It is a hyper-parameter which is used to control the adjusting of weights of our network. Using learning rate may make sure that we do not miss any local minima, however this might result into taking long time to converge when on plateau region.

3) Optimizer: To increase the the models's accuracy, we use optimization algorithms that helps in minimizing the objective function, i.e. the Error function E(x). It's a mathematical function dependent on model's internal learnable parameters.These parameters are used in computing to target values (Y). For example we call the Weights(W) and the Bias(b) values of the neural network as its internal learnable parameters which are used in computing the output values and are learned and updated in the direction of optimal solution i.e minimizing theLossby the networks training process and also play a major role in the training process of the Neural Network Model.

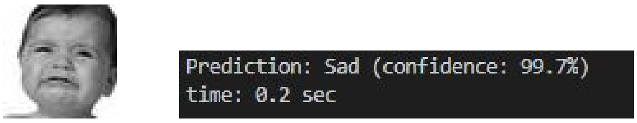
D. Prediction

After the optimizing step, the prediction is done. Following are some outputs of this step.



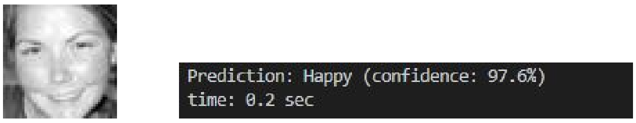
```
Prediction: Fear (confidence: 97.6%)
time: 0.3 sec
```

Fig. 5. Sample Result 1



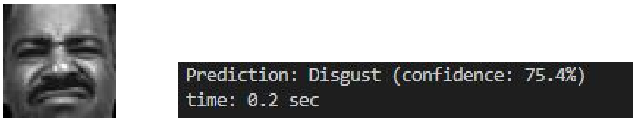
```
Prediction: Sad (confidence: 99.7%)
time: 0.2 sec
```

Fig. 6. Sample Result 2



```
Prediction: Happy (confidence: 97.6%)
time: 0.2 sec
```

Fig. 7. Sample Result 3

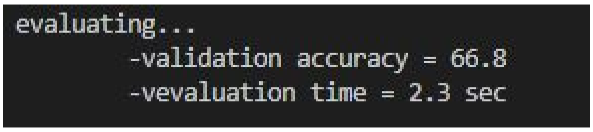


```
Prediction: Disgust (confidence: 75.4%)
time: 0.2 sec
```

Fig. 8. Sample Result 4

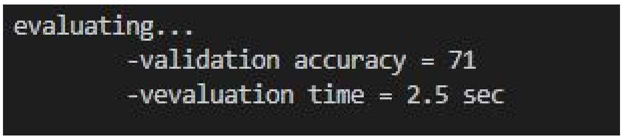
V. RESULTS AND DISCUSSION

1) Hyperopt: Hyperopt is a way to search through an hyperparameter space. which explore intelligently the search space while narrowing down to the estimated best parameters. It is hence a good method for meta-optimizing a neural network which is itself an optimisation problem: tuning a neural network uses gradient descent methods, and tuning the hyperparameters needs to be done differently since gradient descent cant apply. Therefore, Hyperopt can be useful not only for tuning hyperparameters such as the learning rate, but also to tune more fancy parameters in a flexible way, such as changing the number of layers of certain types, or the number of neurons in a layer, or even the type of layer to use at a certain place in the network given an array of choices, each with nested tunable hyperparameters. This is an oriented random search, in contrast with a Grid Search where hyperparameters are pre-established with fixed steps increase. Random Search for Hyper-Parameter Optimization (such as what Hyperopt do) has proven to be an effective search technique. The paper about this technique sits among the most cited deep learning papers. To sum up, it is more efficient to search randomly through values and to intelligently narrow the search space rather than looping on fixed sets of values for the hyperparameters.



```
evaluating...
        -validation accuracy = 66.8
        -vevaluation time = 2.3 sec
```

Fig. 9. Accuracy after training



```
evaluating...
        -validation accuracy = 71
        -vevaluation time = 2.5 sec
```

Fig. 10. Accuracy after using Hyper Opt

2) HOG: HOG (Histogram of Oriented Gradients) is used for extracting features from image data. It is widely used in computer vision tasks for object detection.It's a feature descriptor that focuses on structure or the shape of an object. Edge direction can also be provided by HOG. This is done by extracting the gradient and orientation (or you can say magnitude and direction) of the edges.

TABLE II
COMPARATIVE VIEW

Features	7 Emotions	5 Emotions
Hog Features	29.0%	34.4%
Face Landmarks	39.2%	46.9%
Face Landmarks + HOG	48.2%	55.0%
Face Landmarks + HOG on sliding window	50.5%	59.4%
Face Landmarks + HOG (With hyperOpt)	66%	71%

VI. CONCLUSION

This work presents a convolution neural network architecture that uses feature information from facial parts (Eyes and Mouth) as input into two separate CNN channels. The output from the two channels converges into a fully connected layer and the result used for classification. This method is aimed to have an advantage over using the whole face as an input by having an increased recognition accuracy and reduced cost. Experimental results based on the FER2013 confirms the effectiveness and robustness of this method. It is shown that our proposed method can achieve the average expression recognition accuracy of 71 percentage. Another interesting aspect of this work that can be explored in the future would be to test this approach on more databases.

REFERENCES

[1] Kaggle. Challenges in representation learning: Facial expression recognition challenge, 2013
[2] Facial Expression Recognition using Convolutional Neural Networks: State of the Art, Pramerdorfer and al. 2016
[3] A New Approach for Automatic Face Emotion Recognition and Classification Based on Deep Networks
[4] Facial Emotion Analysis using Deep Convolution Neural Network
[5] Deep Convolutional Neural Network for Facial Expression Recognition using Facial Parts
[6] Real-time Algorithms for Facial Emotion Recognition: A Comparison of Different Approaches