VASAVI COLLEGE OF ENGINEERING

Department of Electronics & Communication Engineering

[1]ALEKHYA THOTHA
1602-18-735-063
thothaalekhya1@gmail.com

[2]KEERTHIKA PAKKIR
1602-18-735-074
keerthikapakkir.meet@gmail.com

[3]KYATHI KONDETI
1602-18-735-075
khyathikondeti5266@gmail.com

FACIAL EMOTION DETECTION USING CONVULTION NEURAL NETWORKS

ABSTARCT:

Human emotions are mental states of feelings that come off spontaneously rather than through conscious effort and are accompanied by physiological changes in facial muscles which imply expressions on the face. Non-verbal communication methods such as facial expressions, eye movement, and gestures are used in many applications of human-computer interaction, which among them facial emotion is widely used because it conveys the emotional states and feelings of persons. Technically, the project's goal consists on training a deep neural network with labelled images of static facial emotions. Later, this network could be used as part of a software to detect emotions in real time.

In the machine learning algorithm some important extracted features used for modelling the face, so, it will not achieve a high accuracy rate for recognition of emotion because the features are hand-engineered and depend on prior knowledge. Artificial Intelligence has been witnessing a monumental growth in bridging the gap between the capabilities of humans and machines. Researchers and enthusiasts alike, work on numerous aspects of the field to make amazing things happen. One of many such areas is the domain of Computer Vision.

Convolutional neural networks (CNN) have developed in this work for recognition facial emotion expression. Recognizing emotion from images has become one of the active research themes in image processing and in applications based on human-computer interaction. Our project focuses on live images taken from the webcam. The aim of this project is to develop automatic facial emotion recognition system for stressed individuals thus assigning them music therapy so as to relief stress. The emotions considered for the experiments include happiness, Sadness, Surprise, Fear, Disgust, and Anger that are universally accepted.

Keywords: Convolution, open CV, keras, ReLU, normalization, FER2013, Data preprocessing, Augmentation

I INTRODUCTION:

Facial expression recognition is the process of identifying human emotion. This is both something that humans do automatically but computational methodologies have also been developed. In the facial recognition application faces are matches with the available dataset to find out the human faces. Here in general, identification of various emotion which occur in a human is experimented using facial image even ages ago where several researchers impended. There are seven basic emotions are universal to human beings. Namely neutral, angry, disgust, fear, happy, sad, and surprise, and these basic emotions can be recognized from a human's facial expression. This research proposes an effective way to detect neutral, happy, sad, and surprise these four emotions from frontal facial emotion.

For this particular project many researches have been done and on improvising on and too many algorithms which are developed as well and Computer applications are generally improvised and accurate way to detect the emotion detection as well. There are many ways to approach and traditional methods starting from now whether it is from distinguishing facial front image with user to using kera library, where to produce recognized results. In terms of facial emotion recognition, one major challenge lies in the data collected. Most datasets contain labelled images which are generally posed. This generally involves photos taken in a stable environment such as a laboratory. While it is much easier to accurately predict the emotion in such scenarios, these systems tend to be unreliable in predicting emotions in the "wild" (Uncontrolled environments). Another issue is that most

datasets are from these controlled environments and it is relatively harder to obtain labelled datasets of emotions in the wild. Furthermore, most datasets have relatively lesser training data for emotions such as fear and disgust when compared to emotions such as happiness. Another factor to take into account is a person's pose. It is significantly harder to determine the emotion of a person when only half of their face is visible. In addition, lighting plays a major role in facial emotion recognition. Systems may fail to identify an emotion that it normally would identify if the lighting conditions are poor.

The agenda for this field is to enable machines to view the world as humans do, perceive it in a similar manner and even use the knowledge for a multitude of tasks such as Image & Video recognition, Image Analysis & Classification, Media Recreation, Recommendation Systems, Natural Language Processing, etc. The advancements in Computer Vision with Deep Learning has been constructed and perfected with time, primarily over one particular algorithm — a **Convolutional Neural Network**.

One of the current top applications of artificial intelligence using neural networks is the recognition of faces in photos and videos. Most techniques process visual data and search for general patterns present in human faces. Applications involve automatic blurring of faces on Google Street view footage and automatic recognition of Facebook friends in photos.

II RELATED WORKS

Yu and Zhang [3] used a five layer ensemble CNN to achieve a 0.612 accuracy. They pre-trained their models on the FER-2013 dataset and then finetuned the model on the Static Facial Expressions in the Wild 2.0 (SFEW) [5] dataset. They used an ensemble of three face detectors to detect and extract faces from the labelled movie frames of SFEW. They then proposed a data perturbation and voting method to increase the recognition performance of the CNN. They also chose to use stochastic pooling layers over max pooling layers citing its better performance on their limited data.

Kahou et al. [6] used a CNN-RNN architecture to train a model on individual frames of videos as well as static images. They made use of the Acted Facial Expressions in the Wild (AFEW) [7] 5.0 dataset for the video clips and a combination of the FER-2013 and Toronto Face Database for the images. Instead of using long short term memory (LSTM) units, they used IRNNs [8] which are composed of rectified linear units (ReLUs). These IRNNs provided a simple mechanism for dealing with the vanishing and exploding gradient problem. They achieved an overall accuracy of 0.528.

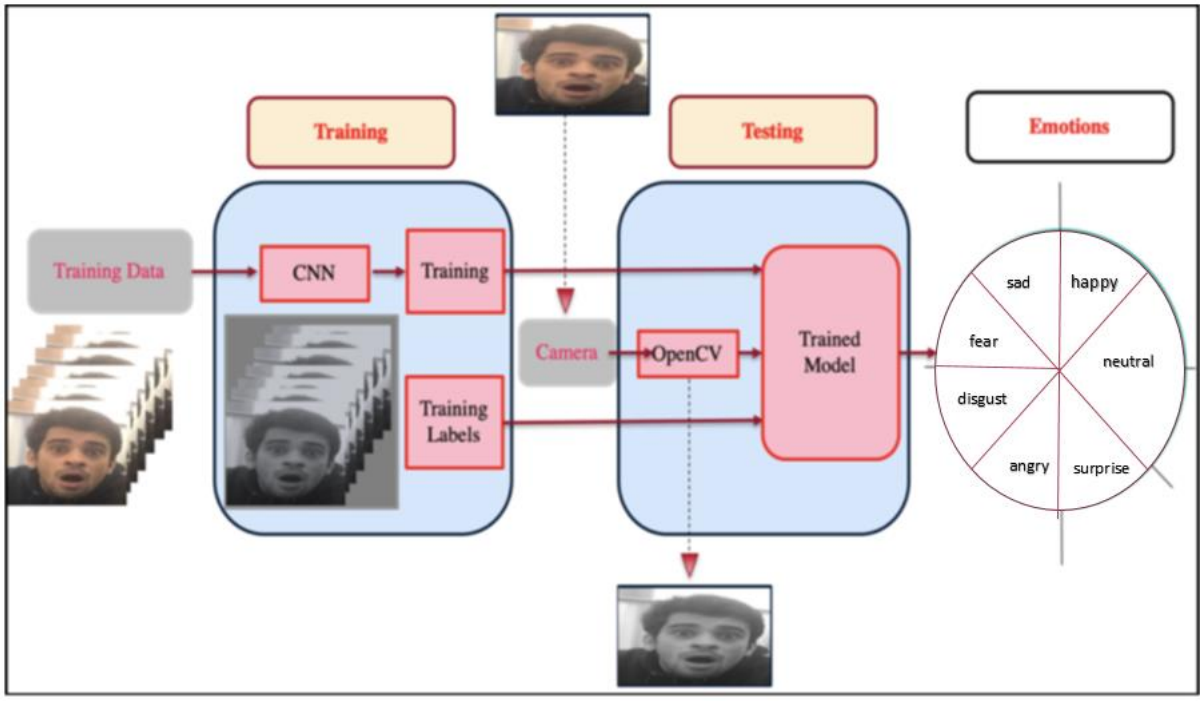
Mollahosseini et al. [9] proposed a network consisting of two convolutional layers each followed by max pooling and then four Inception layers. They used this network on seven different datasets including the FER-2013 dataset. They also compared the accuracies of their proposed network with an AlexNet [12] network trained on the same datasets. They found that their architecture had better performance on the MMI and FER-2013 datasets with comparable performances on the remaining five datasets. The FER-2013 dataset in particular managed to reach an accuracy of 0.664.

Ming Li et al. [10] propose a neural network model to overcome two shortcomings in still image based FERs which are the inter-variability of emotions between subjects and misclassification of emotions. The model consists of two convolutional neural networks - the first is trained with facial expression databases whereas the second is a DeepID network used for learning identity features. These two networks are then concatenated together as a Tandem Facial Expression of TFE Feature which is fed to the fully connected layers to form a new model. The proposed model was evaluated on two datasets, namely the FER+ database and the Extended CohnKanade (CK+) database. The identity features were learned from the CASIA-WebFace database. The model was trained for 200 epochs and achieved an accuracy measure of 71.1% on the FER2013 dataset, 99.31% on

the CK+ database. These experimental results show that the model outperforms many state-of-the-art methods on the CK+ and FER+ databases.

Tan et al. [11] propose a neural network model to classify a group image into a particular emotion - positive, neutral or negative. The model consists of two convolutional neural networks - the first is based on group images and the second is based on individual facial emotions. The facial emotion CNN comprises of two CNNs - one for aligned faces which is trained using the ResNet64 model using the Webface dataset and the other for non-aligned faces which is trained using the ResNet34 model on the FER+ dataset. The group images are trained using VGG19 model on the Places and ImageNet datasets. Fine-tuning is done with batch normalisation and average pooling of the ResNet101 and BN-Inception models, with a dropout of 0.5. The validation set consists of 2068 images - combined from all of the datasets used for training and the model achieved an accuracy measure of 80.9.

Most other works in the same field attempted to solve the facial emotion recognition problem by the use of a combination of different datasets. In this paper, a single dataset, FER-2013 was chosen over such a combination of different datasets and then experiments were conducted with different models to find the highest accuracy that each model could reach.



[a]architecture of CNN for facial expression recognition

III THEORITICAL BACKGROUND

Convolutional layers are the major building blocks used in convolutional neural networks. A convolution is the simple application of a filter to an input that results in an activation. Repeated application of the same filter to an input results in a map of activations called a feature map, indicating the locations and strength of a detected feature in an input, such as an image.

1] Data Pre-processing: The data pre-processing is to standardize the data. The typical way is to set the mean of the data to 0 and to also divide the data by the standard deviation.

2] Feature Extraction: The typical conventional method is to detect the face and extract the Action Units (AU) from the face, and certain emotion contain the combination of AUs code as feature.

3] Model Construction: The conventional classifier can be either supervised or unsupervised algorithm. A typical example of supervised algorithm is Support Vector Machine, and the examples of unsupervised IV algorithm include Principle Component Analysis (PCA) and Linear Discriminant Analysis (LDA)

4] Label or Result Generation The typical way to generate label or result is to find which decision boundary has the minimum Euclidean distance from the data.

The issues of Conventional method are:

A. Variance of Lights: Since each image is taken in the completely different background and lighting conditions, the intra-class noise of lights will distort the model to classify the emotion. As results, the same type of emotions may be classified differently because of the effect of lighting noise.

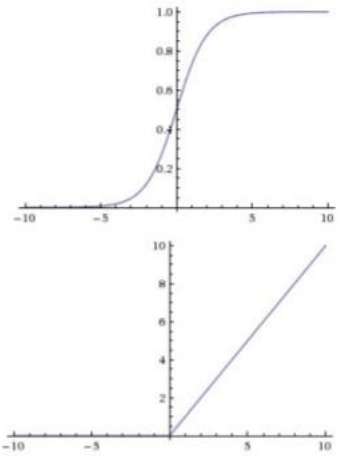
B. Variance of Location: Since the feature is typically extracted by filters such application of Local Binary Pattern in, the location of the feature, therefore, may affect the functionality of the feature extraction. As results, the AU may be extracted incorrectly if the face has is rotated or is in different part of the image. These two issues are major problems faced by conventional algorithms.

Solution to the Issues of Conventional Method

Summary of Solution: In order to solve the issues of the conventional methods, the solution is to apply the algorithm of Convolutional Neural Network (CNN) to perform classification of emotion. In contrast to conventional method, the key differences of algorithm are:

1. Automatic Generation of Feature Extractor: The feature of images can be captured automatically without users' built feature extractor because the feature extractors are generated in the process of training based on the given ground truth.

2. Differences of Mathematical Model: The conventional method is typically performing classification through linear transformation, so they are typically referred as Linear Classifier. In comparison, CNN as well as other Deep Learning algorithm typically combines the linear transformation with nonlinear function such as sigmoid (Logistic Function) and Rectified Linear Unit (ReLU) to distinguish differences in process of classification.



The sigmoid function(upper) and ReLU function (lower)

3. The Deeper Structure: The conventional method typically performs only one layers of operation: for example SVM only has one set of weights.(shown in equation 1) However, CNN as well as other deep learning algorithm does multiple layers of operation in the process of classification.

$$S = W * x_i + b \text{[1]}$$

Where, S is the classification score, W is weights matrix, and b is bias

III METHODLOGY:

In general, CNN contains 2 blocks of layers: convolutional layers and fully-connected Layers (also known as Dense Layers). The convolutional layers are used to perform the task of feature extraction as it detects and produces signals of feature by doing dot product between convolutional kernels and the feature map or images. On the other hand, the fully-connected layers consists of units as neuron cells that mathematically represents the linear operation and the operation of activation function .The last layer, a softmax classifier (equation 2) will generate the output in the format of the probability of each class in multinomial distribution, and the correct label will be the one with the maximum probability.(equation 2 and 3 show the softmax classifier model)

$$P(y = j \mid \theta^i) = \frac{e^{\theta^i}}{\sum_{j=0}^{k} e^{\theta_k^i}} \quad [2]$$

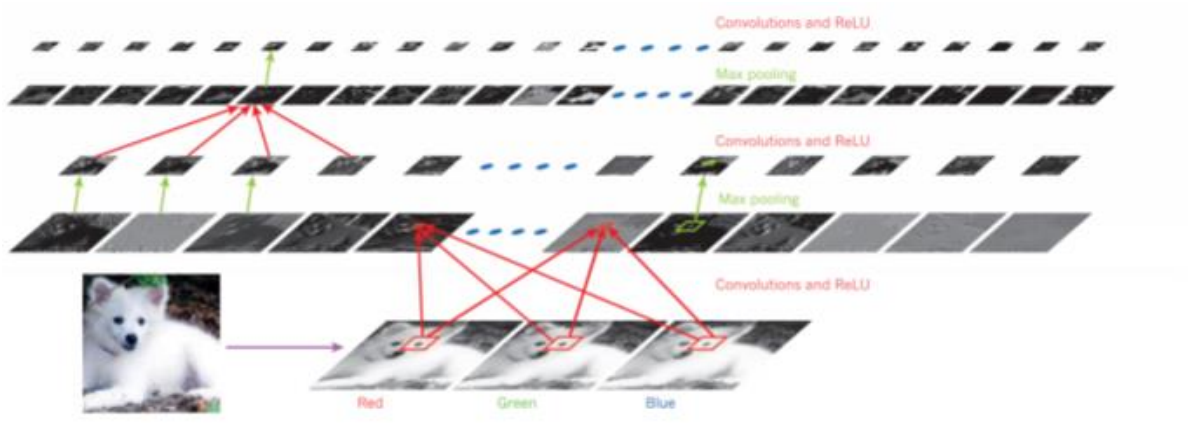
where :

$$\theta = w_0 x_0 + w_1 x_1 + \ldots + w_k x_k = \sum_{i=0}^{k} w_i x_i = w^T x$$

[3]

For the Convolution layers, the structure includes convolution filters, activation function, and Maxpooling filters. The convolution filters are kernels with certain size that perform dot product between the images or feature maps and filters. In such process, one filter represents one specific feature the model intends to detect, and this feature can be located anywhere on the image or feature-map since the kernel is connected with every region with size of kernel on the image or feature maps.

In addition, the kernel is sensitive enough to detect the feature since the feature is the region with high cross-correlation. Besides convolutional layers, the activation function will serve as a threshold that only allows the acceptable signal from previous convolutional layers to pass through, thereby making the system more complex so that the feature can be easily distinguished. The last type of convolutional block is the Maxpooling layer. The Maxpooling layer performs the task to downsample the image or feature map, thereby reducing the computation and directing the next layers to focus on more detailed features. In conclusion, these 3 types of layers construct the uniqueness of convolution block.



In addition to convolutional block, the fully connected layers will serve as classifier. Each unit of the layer contains the weight matrix, and through the linear transformation and activation function the output becomes the input of next layers of units. In contrast to traditional linear transformation, the activation function ReLU will act in the same way as the convolutional layers to make the system more easily distinguish the feature.

Output = max (input, 0) …[4]

In addition, the last layer is normally a softmax classifier, and the result is typical the one with maximum probability in multinomial distribution.

classi ~ Multinomial(φ_i) ….[5]

label = arg max i (φ_i) …..[6]

where

$$\sum_{i \in < \text{dass} >} \phi_i = 1$$

The Optimization and Training of CNN

In order to train the model to correctly predict the result, the model must minimize the difference between the predicted result and the actual result. Mathematically, we construct the loss function to calculate the difference between the true label and the predicted one. In the case of the softmax classifier as mentioned in 5.2.1, the loss function is cross-entropy shown as equation 7,8,and 9

$$L_i = -\log\left(\frac{e^{f_{yi}}}{\sum_j e^{f_{y_j}}}\right) \quad \text{.....[7]}$$

$$L_i = -f_{yi} + \log\sum_j e^{f_{y_j}} \quad \text{.....[8]}$$

$$L_{\text{total}} = \sum_i L_i \quad \text{.....[9]}$$

Here equation 7 is equivalent equation 8 and the loss function of entire dataset is described in equation 9.

In order to optimize the loss function, we need to find the derivative that leads us to the minima.

In the process to find derivative of the loss function, we apply back propagation to trace back each layers and use chain rule to deduct the derivative in respect to weights of whole system.

Algorithm 1: the Algorithm of Gradient Descent

Initialize the weights of a CNN and run forward pass of the network:
Obtain the loss function:

$$L_{\text{total}} = \sum_i -\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_{y_j}}}\right)$$

while epochs< max steps **do**
Find the gradient: $\nabla_\theta L_{\text{total}}$ in respect to weights θ
Update the weight: $\theta_{\text{new}} = \theta - \alpha * \nabla_\theta L_{\text{total}}$
Where, α is the learning rate re-run the forward pass and obtain the new Loss Function
end while

Algorithm 2: the Mini-Batch Algorithm

while The data is not completely run through **do**
Select a Mini-Batch of size n and do Algorithm
Do not return these data points back to the entire dataset
end while

A]DATA PREPROCESSING AND AUGUMENTATION

Algorithm 3: The Algorithm of Data Augmentation

Function of Random noise()
if mode = one type of noise then
Add noise into image
Else
Keep the original state
end if
Function of Flip()

```
if mode = Add flip then
  Perform flip
else
Keep original state
end if
 Function of Rotation()
if the degree = certain degrees then
  Perform rotation in such degree
end if

Main function to combine these 3 effects
set up array with mode of random noise
set up array with mode of whether to flip the image
set up array with degrees of rotation
import image file
for each image in entire image dataset do
 for one degree in degree array do
  for one mode in noise array do
   for one mode in flip array do
        augmented data = flip function(rotation function(random noise function(image)))
        Save the data in corresponding directory of labels
     end for
    end for
  end for
end for
```

1. Random noise: We added three kinds of noises for augmentation (Gaussian, Salt, and Poisson)

–Gaussian noise: The noise is in a Gaussian distribution and randomly added into images

– Salt: The noise is to add 1 randomly into images

– Poisson: The noise is in a Poisson distribution and randomly added into images.

2. Horizontal flip: Adding an effect of flipping the images horizontally.

3. Rotation: We rotate the image from 0° to 50°by increment of 10°.

B] NORMALIZATION OF DATA:

Before we start the training, we also have to normalize the data. Though we may not need to normalize data by manually writing program on that, it is still important that these procedures should be included before the model can do either training or prediction. The normalization of our solution includes:

• Rescale each pixels to 0 to 1

• center the mean to 0

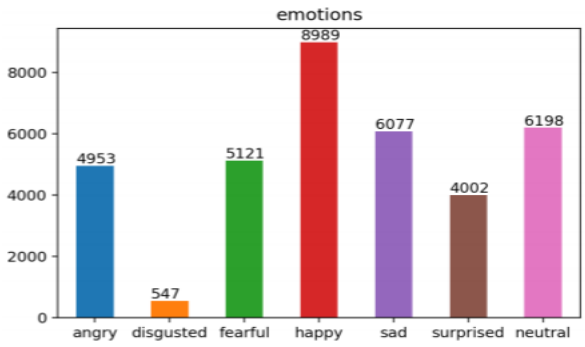
• divide each point by the standard derivation.

Some advanced methods include PCA which removes the correlation between the data and projects the data onto the Eigen space of its co-variance matrix and whitening which normalize the data point on eigenspace. However, we do not consider these two methods in this project.

C] DATA SET

In general, neural networks, especially deep neural networks, tend to perform better when larger amounts of training data set is present. With this in mind, the more popular FER-2013 Expressions Extended Cohn-Kanade (CK+) [13] and the Static Facial Expressions in the Wild (SFEW) datasets were overlooked. Instead the Facial Expression dataset (FER-2013) was chosen. The FER-2013 dataset was introduced in the ICML 2013 Challenges in Representation Learning .It contains 35,887 images with the following basic expressions: angry, disgusted, fearful, happy, sad, surprised and neutral. Figure 1 shows the distribution of each expression. Each image is a frontal view of a subject, taken from the wild and annotated to one of the seven expressions. A sample of these expressions are shown in Figure 2. It is to be noted that the number of disgusted expressions (547) is much lower in comparison to the other expressions. There was also an obvious bias towards happy expressions due to

the sheer number of sample data present for the expression.



Fer 2013 dataset

D] CHOOSING MODEL:

4 different models were used in order to select the best foundation to work on - 3 neural networks and a decision tree. The neural networks were implemented using Keras with a TensorFlow backend running in Python. The decision tree was implemented with the help of Sci-Kit Learn .All the implementations are written in Python 3.6 and are wholly reproducible with freely available software. A comparison of the final accuracy of all the models can be seen in Table I.

Since there are 4 different models used, the individual training algorithm for each model is detailed with the model description itself. The testing algorithm for the same is described in subsection D.

1)Decision Tree: A decision tree was chosen at first chosen due to a combination of it requiring little to no effort in preparation of data as well as its reputation for working well in almost all scenarios. The first attempt was to implement a decision tree using Sci-Kit Learn. A standard decision tree classifier was used and the parameters were tuned. However the only parameter that showed a noticeable difference while tuning was the emphmin_samples split parameter. With the parameter set to 40, the overall accuracy was only a mere 0.309. As state of the art models reached accuracies greater than 0.6, it was decided to try out a neural network instead.

Model	Accuracy
Decision tree	30.84%
Feed forward NN	17.56%
Simple CNN	23.54%
Proposed CNN	63.19%

Table I: Comparison of models

TABLE II
ARCHITECTURE OF THE FEEDFORWARD NEURAL NETWORK

Feed-Forward Neural Network
FULLY CONNECTED RELU DROPOUT
FULLY CONNECTED RELU DROPOUT
FULLY CONNECTED SOFTMAX

2) Feedforward Neural Network: The next attempt was to try out the most basic form of an artificial neural network, a Feedforward Neural Network. Three layers were used - an input layer, a single hidden layer and an output layer. Each of these layers was a fully connected (dense) layer. The architecture of this model can be seen in Table II.

A dropout of rate 0.2 was applied for the input and hidden layer in an attempt to prevent over fitting. The output layer uses a softmax activation function while the remaining layers use ReLU (Rectified Linear Units). However this model ended up predicting the same expression for every input - angry. Further tuning of the hyper parameters lead to no difference and so it was decided that a convolutional neural network might work better.

TABLE III
ARCHITECTURE OF THE SIMPLE CNN

Simple Convolution Neural Network
CONV2D-32 RELU
CONV2D-32 RELU MAXPOOL2D DROPOUT
FLATTEN FULLY CONNECTED RELU DROPOUT
FULLYCONNECTED SOFTMAX

3) Simple Convolutional Network: Next, a basic convolutional network was tried out. This model's architecture can be seen in Table III. This model consisted of two two-dimensional convolutional layers followed by a two-dimensional max pooling layer which was followed by two fully-connected (Dense)

layers. The output was flattened before entering the fully connected layers. Dropout was applied to the max pooling and to the first fully connected layer to reduce overfitting. This model while more complicated, ended up having the same issue as the feedforward neural network except instead of predicting angry, it predicted happy for all inputs. This makes sense as a quarter of the inputs are for the happy expression. In order to allow the model to actually learn, attempts were made to make the model deeper.

4) Final Model: The final model is depicted in Table IV. The network consists of six two-dimensional convolutional layers, two max pooling layers and two fully connected layers. Max pooling uses the maximum value from each of a cluster of neurons at the prior layer. This reduces the dimensionality of the output array. The input to the network is a pre-processed face of 48 x 48 pixels. The model was developed based on the observation of the performance of the previous models. It was decided to go with a deeper network over a wide one. The advantage of using more layers is that it prevents memorization. A wide but shallow network memorizes well but does not generalize well. Multi-layer networks learn features at levels of abstractions allowing them to generalize well. The number of layers were selected so as to maintain a high level of accuracy while still being fast enough for real-time purposes. The proposed CNN differs from a simple CNN in that it uses 4 more convolutional layers and each of its convolutional layers differ in filter size. In addition, it utilized max pooling and dropout more effectively in order to minimize overfitting

TABLE IV ARCHITECTURE OF THE PROPOSED CNN

Proposed Convolutional Neural Network
CONV2D-64 RELU MAXPOOL 2D
CONV2D-64 RELU
CONV2D-64 RELU AVERAGE POOL2D
CONV2D-128 RELU
CONV2D-128 RELU AVERAGEPOOL2D
FLATTEN
FULLY CONNECTED RELU DROPOUT
RELU DROPOUT
FULLY CONNECTED SOFTMAX

The network consists of three convolutional layers with a filter size of 64 each. This is then followed by a max pooling layer. A dropout of rate 0.25 is applied to reduce overfitting. This is followed by a sequence of four convolutional layers. The first two have a filter size of 64 each and the latter three have a filter size of 128 each. A single max pooling layer follows these four layers with a dropout of rate 0.25. In order to convert the output into a single dimensional vector, the output of the previous layers was flattened. Finally, a fully connected layer with a softmax activation function serves as the output layer. The kernel size, that is, the width and height of the 2D convolutional window is set to 3 x 3 for all convolutional layers. Each max pooling layer is two dimensional and uses a pool size of 2 x 2. This halves the size of the output after each pooling layer. All the layers bar the output layer used a ReLU activation function. The ReLU activation function is used here due to benefits such as sparsity and a reduced likelihood of vanishing gradient. The softmax activation function was used in the final output layer to receive the predicted probability of each emotion. This model provided a base accuracy of 0.5519 on the testing set. The hyperparamters were then tuned, namely the batch size, the optimizer and the number of epochs. Each model was set to run for 30 epochs. This saved both time and computational power, especially in cases where there was no change in the accuracy within the earlier epochs themselves. The decision turned out to be a good one as none of the models exceeded 20 epochs.

E] TESTING

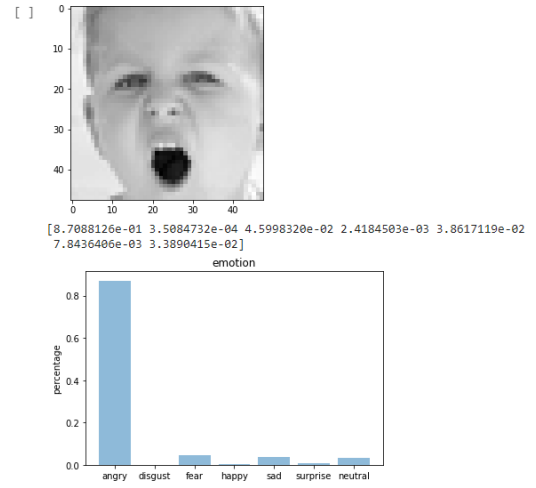
The dataset was initially split into an 80%-training set and a 20%-testing set. During the testing phase, each of the trained networks was loaded and fed the entire testing set one image at a time. This image was a new one which the model had never seen before. The image fed to the model was preprocessed in the same way as detailed. Thus the model did not know already what the correct output was and had to accurately predict it based on its own training.

It attempted to classify the emotion shown on the image simply based on what it had already learned along with the characteristics of the image itself. Thus in the end, it gave a list of classified emotion probabilities for each image. The highest probability emotion for each image was then compared with the actual emotions associated with the images to count the number of accurate predictions. The accuracy formula is detailed below. It simply counts the number of samples where the model correctly predicted the emotion and divides it by the total number of samples in the testing set. Here, the testing set consists of about 7,178 images.

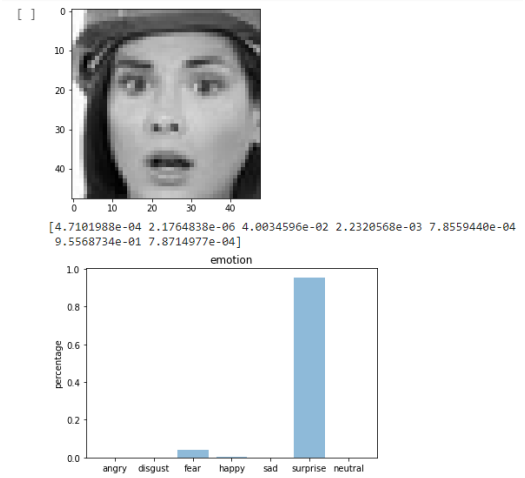
$$Accuracy = \frac{Num.CorrectlyPredictedEmotions}{TotalNum.Samples}$$

F] RESULTS

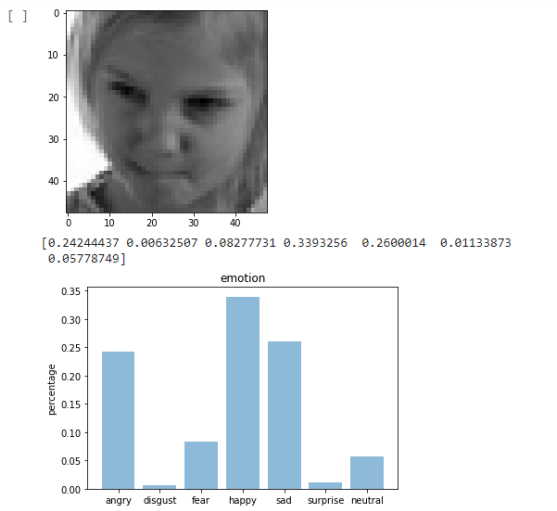
Upon batch size of 128 and using 30 epoch we got the training accuracy as 63.1957% and testing accuracy as 55.19%



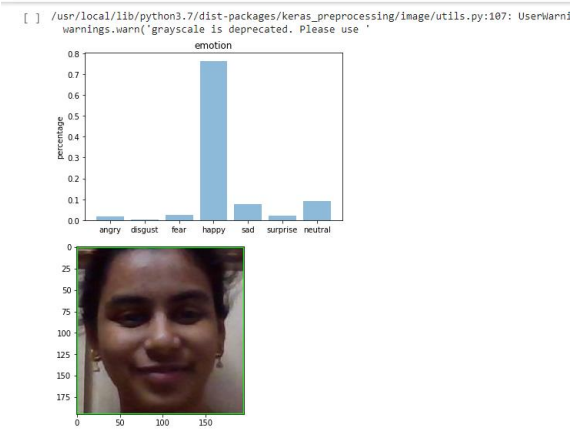
[a]



[b]



[c] Here is picture c the predicted output is flucuating so in here were here cropping the image ie were avoiding the background disturbances over predicted output



Live testing module

IV FUTURE SCOPE

In the future, the model will be trained and tested on a boarder sets of data to not only fit the training dataset or lab condition images but also to achieve better accuracy on wild emotion recognition. In addition to this, the model is also expected to deliver the real life application in education to improve the learning and interactions in class, deep face detection which can be used to identify impersonators. To be used for detecting emotions of autism patients. It can also be used in automotive driver detection.

V CONCLUSION

In this paper, the aim was to classify facial expressions into one of seven emotions by using various models on the FER2013 dataset. Models that were experimented with include decision trees, feed forward neural networks

and smaller convolutional networks before arriving at the proposed model. The effects of different hyperparameters on the final model was then investigated. The final accuracy of 0.63 was achieved using the Adam optimizer with modified hyperparameters. It should also be noted that a nearly state-of-the-art accuracy was achieved with the use of a single dataset as opposed to a combination of many datasets.

The ability of the model to make predictions in effectively real-time, indicates that real world uses of facial emotion recognition is barred only by the relative inaccuracies of the model itself. In the future, an indepth analysis of the top2 predicted emotions may lead to a much more accurate and reliable system. Further training samples for the more difficult to predict emotion of disgust will definitely be required in order to perfect such a system. The real-time capacity of the model in addition to its quick training time and near-state-of-the-art accuracy allows the model to be adapted and used in nearly any use-case. This also implies that with some work, the model could very well be deployed into real-life applications for effective utilization in domains such as in healthcare, marketing and the video game industry.

VI REFERNCES

[1] P. Ekman, & W. V. Friesen. Constants across cultures in the face and emotion. Journal of Personality and Social Psychology, 17(2), 124-129. (1971)

[2] Challenges in representation learning: Facial expression recognition challenge. (2013) http://www.kaggle.com/c/challenges-in-representationlearning-facial-expression-recognition-challenge

[3] Zhiding Yu & Cha Zhang. (2015). Image based Static Facial Expression Recognition with Multiple Deep Network Learning. Proceedings of the 2015 ACM on International Conference on Multimodal Interaction . 435- 442.

[4] Raghuvanshi, A., & Choksi, V. (2016). Facial Expression Recognition with Convolutional Neural Networks. CS231n Course Projects.

[5] Abhinav Dhall, Roland Goecke, Simon Lucey, and Tom Gedeon. (2011). Static Facial Expressions in Tough Conditions: Data, Evaluation Protocol And Benchmark, First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies BeFIT, IEEE International Conference on Computer Vision ICCV2011, Barcelona, Spain, 6-13 November 2011

[6] Ebrahimi Kahou, S., Michalski, V., Konda, K., Memisevic, R., and Pal, C. (2015). Recurrent neural networks for emotion recognition in video. Proceedings of the 2015 ACM on International Conference on Multimodal Interaction. 467-474. ACM.

[7] Abhinav Dhall, Roland Goecke, Simon Lucey, Tom Gedeon. (2012). Collecting Large, Richly Annotated Facial-Expression Databases from Movies, IEEE Multimedia, 19(3):3441, July 2012

[8] Q. V. Le, N. Jaitly, and G. E. Hinton. (2015). A simple way to initialize recurrent networks of rectified linear units. arXiv preprint arXiv:1504.00941.

[9] A. Mollahosseini, D. Chan and M. H. Mahoor. (2016). Going deeper in facial expression recognition using deep neural networks. 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Placid, NY, 1-10.

[10] Li M., Xu H., Huang X., Song Z., Liu X. and Li X. (2018). Facial Expression Recognition with Identity and Emotion Joint Learning. IEEE Transactions on Affective Computing. 1-1.

[11] Tan L., Zhang K., Wang K., Zeng X., Peng X. and Qiao Y. (2017) Group emotion recognition with individual facial emotion CNNs and global image based CNNs. Proceedings of the 19th ACM International Conference on Multimodal Interaction - ICMI 2017, 549-552. ACM.

[12]M. Saaidia et. al., “Facial Expression Recognition Using Neural Network Trained with Zernike Moments”, IEEE International Conference on Artificial Intelligence with Applications in Engineering and Technology, 2014.

[13]Zhiding Yu et. al., “Image based Static Facial Expression Recognition with Multiple Deep Network Learning”, ACM, 2015.

[14]V. D. Bharate et. al., “Human Emotions Recognition using Adaptive Sublayer Compensation and various Feature Extraction Mechanism”, IEEE WiSPNET, 2016

[15]Parkhi, Omkar M., Andrea Vedaldi, and Andrew Zisserman. "Deep face recognition." In bmvc, vol. 1, no. 3, p. 6. 2015.