Helix (2021) 11 (5): 20-31

https://doi.org/10.29042/2021-11-5-20-31

Optimal Hyperparameter Tuning of Convolutional Neural Networks for Visual Sentiment Analysis

[*1]Sahiti Cheguru, [2]Y Vijayalata

[*1]University of Southern California, Los Angeles, CA 90007, United States

cheguru@usc.edu

[2] Dept of CSE, Gokaraju Rangaraju Institute of Engineering and Technology, Bachupally, Hyderabad, Telanagana, India

vijaya@ieee.org

Received: 23rd August 2021, Accepted: 28th September 2021, Published: 30th October 2021

Abstract

Study of visual sentiment examines complex emotional reaction and reflex behavior of visual expressions, such as pictures and videos. Our research aims to explain the high-level visual information content and obtain recognition results as seven emotional states (neutral, excitement, sorrow, surprise, outrage, fear, disgust), based on facial emotions. The project is divided into three phases: Face identification, which is the ability to recognize facial orientation in any input image or frame inside boundary box coordinates; Facial recognition, which deals with analyzing multiple faces together to recognize the faces belong to the same individual by matching facial embedding vectors and Emotion Detection to define the expression on the face and classify them as happy, neutral, surprise, disgust, fear, outrage or sad.

***Keywords**: Facial detection, Facial Recognition, Emotion detection, OpenCV, Image preprocessing, Feature Extraction, Convolution Neural Networks, Evaluation metrics*

Introduction

Emotion Detection is a cycle of distinguishing and characterizing human feelings. People contrast enormously in their accuracy in understanding others' feelings. A moderately new field of study has up come with utilization of innovation to help individuals with emotion discovery. Regularly, the innovation performs better when it utilizes different relevant modalities. Until this point in time, a significant part of the exploration has been performed on computerizing the identification of outward appearances from pictures, sound articulations, text composed articulations, and wearable-estimated physiology.

Visual Sentiment Analysis is also a method for identifying human feelings from facial expressions. The human mind naturally identifies feelings and now a technology has been built that would identify emotions as well just like our brain. This technology is becoming more effective and accurate progressively, and will one day soon be able to interpret the sentiments as well as our thoughts and impulses in our mind by just reading our facial expressions. AI can interpret emotions by studying thesignificance of our body language, facial expressions and tone of the voice and apply these insights to the new information that is provided. Emotional artificial intelligence is a new technology that has an ability to articulate, replicate, evaluate and react to human face behaviors and emotions.

Recognition of feelings is a very relevant subject matter. The technology has a range of applications. Applications range from different fields such as medical, e-learning, accounting, marketing, entertainment and law. While emotion recognition technology is important one that has been demanding in different fields, it still remains as the unsolvedproblem. Detecting human emotion can be accomplished by the use of facial expression, voice, body form and so on. Among them, the facial image is the most common source for emotion detection. In particular, the facial frontal image is widely used to identify emotions. Recognition of emotions is not a simple but complex process because it requires complex steps to extract proper function and detect emotion. But, with the recent advance in machine learning and computer vision, emotions can be easily identified from images. Within thisproject we use convolutional neural networks (CNN) to introduce a novel technique called facial emotion recognition to perform visual sentiment analysis.

Research Methodology

Data Description

There are 35,888 pictures in this dataset which are characterized into six emotions. The data document contains 3 segments — Class, Image information,and Usage.

The data comprises of 48x48 pixel grayscale pictures of facial expressions images. The appearances have been consequently enlisted with the goal that the faceis pretty much focused and possesses about a similar measure of dimensions and size in each picture. The undertaking is to sort each face dependent on theemotions appeared on the outward appearance into one of seven classifications (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The FER2013 dataset contains pictures that change inperspective, lighting, and scale.

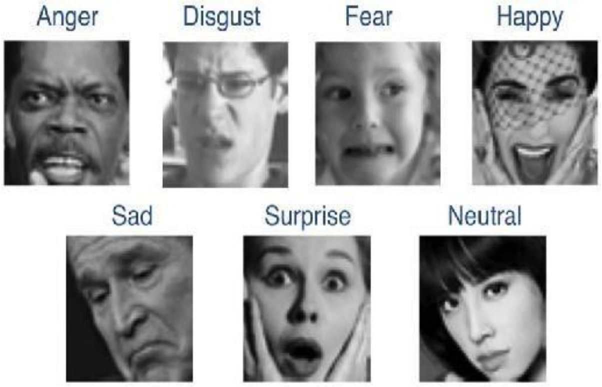
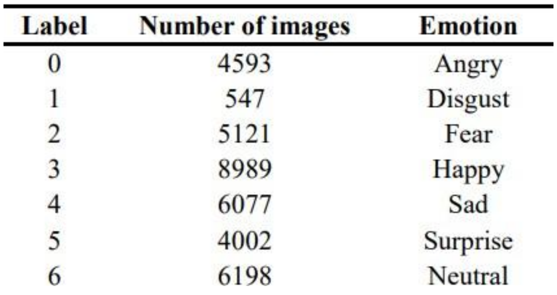


Fig 1: Sample Images from FER2013 dataset

Table 1. Description of the FER2013 dataset



Label	Number of images	Emotion
0	4593	Angry
1	547	Disgust
2	5121	Fear
3	8989	Happy
4	6077	Sad
5	4002	Surprise
6	6198	Neutral

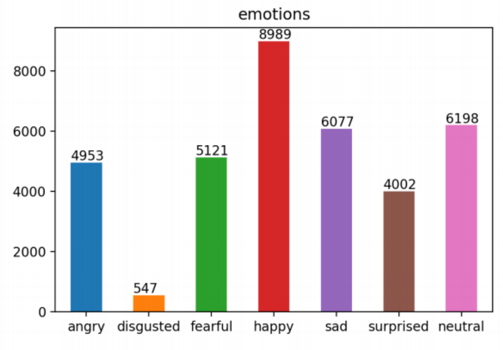


Fig 2: FER-2013 Expression Distribution

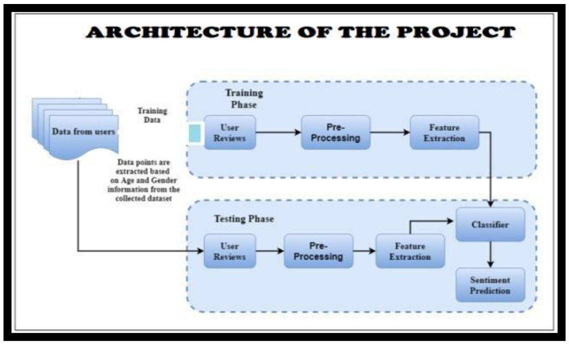


Fig 3: Architecture of the project

Data Preprocessing: The input picture from the FER2013 dataset may contain noise and vary predominantly in brightening, size, shape and shading. To get exact and quicker outcomes on the calculation, some preprocessing tasks were done on the input image. The preprocessing methodologies utilized are transformation of picture to grayscale, standardization, normalization and resizing of given input picture dimensions.

a) The FER2013.csv consists ofthree columns namelyemotion, pixels and purpose.

b) The column in pixel first of allis stored in a list format. Since computational complexity ishigh for computing pixel values in the range of （0- 255), the data in pixel field is normalized to values between [0-1].

c) The face objects stored are reshaped and resize the input image into an appropriate standardized format of 48 X
48. The respective emotionlabel's and their respectivepixel values are stored in objects.

d) We use scikit-learn's train test split()function to partition the dataset into training and testing data, The test size being 02 meaning, 20% of data is for validation while 80%of the data will be trained.

1. **Facial Detection:** Here the face is detected in a image with bounding box coordinates. Face detection using Haas Cascades is a machine-based approach to learning where a cascade function is trained with a set of input data. OpenCV already includes many pre- trained classifiers for face, eyes, smiles,etc.
2. **Emotion Detection:** Here the face isstudied and the emotion of the human face is given of the result.

Proposed Algorithm: Convolutional Neural Networks (CNN)

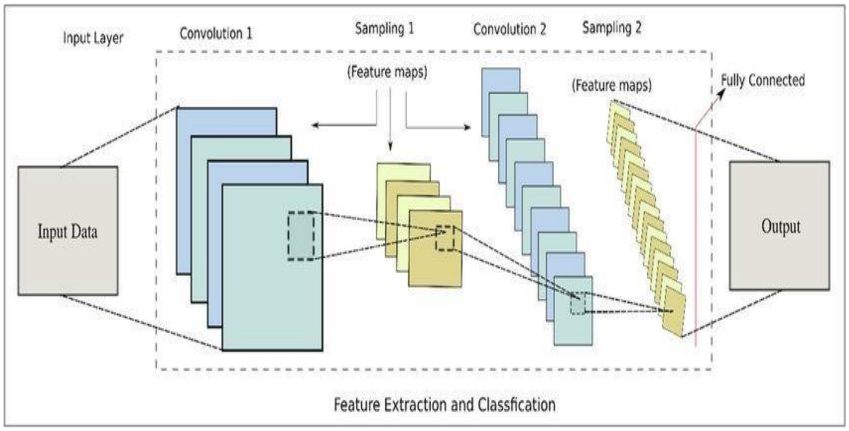


Fig 4: General Architecture of the Convolutional Neural Network

A Convolutional Neural Network is an artificial neural network that is so far been the most popularly used for analyzing images, although image analysis has been the most widespread use of CNN, they can also be used for other data analysis or classification problems as well. Most substantially we can think about a CNN as something which has a sort of specialization for having the option to choose or identify patterns and figure out them. This pattern recognition is the thing that makes CNN so helpful for image analysis. This is what separates a CNN structure from a standard multi-layer perceptron or MLP, a CNN has shrouded layers called convolutional layers and these layers are unequivocally what make a CNN novel.
CNNs also have other non- convolutional layers, however the premise of a CNN is the convolutional layers. Like some other layers, a convolutional layer gets an input image. This received input image is transformed and then the output yielded from the transformation is contributed to the next following layer. With each convolutional layer we have to determine the quantity of filters the layer ought to have (these filters are really what recognize the pattern in the input image).

Multiple edges, shapes, textures, object etc are types of pattern a filter could detect and it could be edges or corners or circles other squares. These simple and kind of geometric filters are what we would see at the start of our network. The deeper our network goes the more sophisticated these filters become so in later layers rather than edges and simple shapes our filters maybe able to detect specific objects like eyes, ears, hair, nose etc.In this step, the system classifies the picture as one of the seven universal expressions – Happy, Sadness, Anger,Surprise, Disgust, Fear, and Neutral – labeled in the FER2013 dataset. Training was carried out using CNN, which is a type of neural networks that has been shown to be effective in image processing. The dataset was first split into training and test datasets, and then trained in the testing collection. Feature extraction process was not completed on the data before it was fed to CNN. The approach followed was toexperiment with different architectures on CNN, to achieve better accuracy with the validation set, with minimal over-fitting.

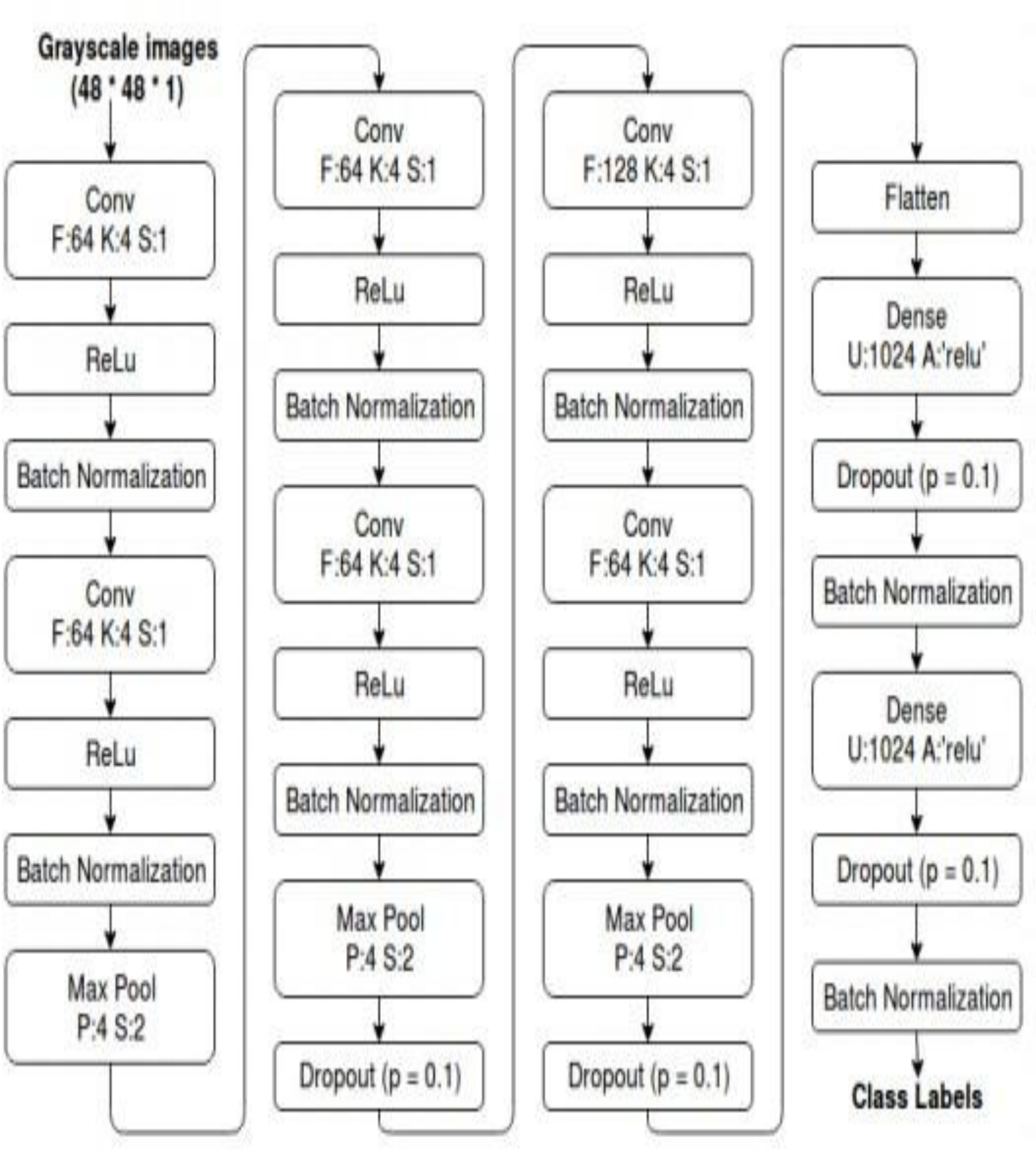
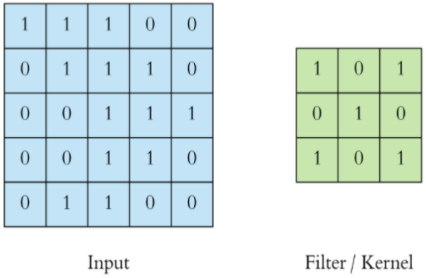


Fig 5: Flow Process of the proposed CNN

The Emotional Classification Step consists of the following phases:

(a) Convolution:

The center structure block of CNN is the convolutional layer. Convolution is a numerical activity to join two arrangements of data. For our situation, the convolution is applied to enter information utilizing a convolution filter to produce a function map.

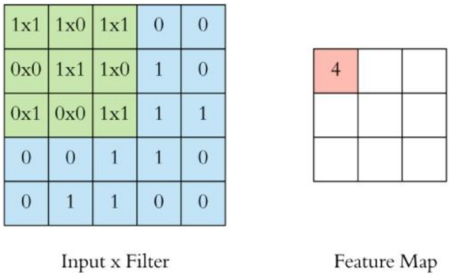


1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

1	0	1
0	1	0
1	0	1

Fig 6: On the left side is the input to the convolution layer, for example the input image. To the other side, the convolution filter, otherwise called the kernel, will be utilized reciprocally.

We perform the convolution cycle by sliding the channel over the given inputimage. At each position, we do a multiplication of the corresponding element-wise matrix and sum the generatedresult. This absolute value is applied to the feature map. The green zone where the convolution cycle is occurring is known as the receptive field. On account of the filter scale, the receptive field is also additionally 3x3.



1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

4		

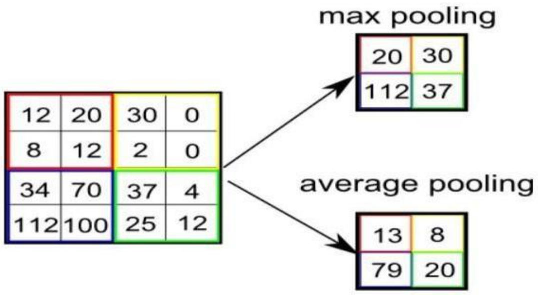
Fig 7: The output yielded by the convolution operation "4" is appearedin the subsequent feature map where thefilter is at the upper left.

Further, at this point we move the filter to right side and perform a similar activity, adding the outcome to the feature map subsequently. We proceed with like this throughout the elements of the kernel and finally total the convolution brings about the feature map.

(b) Pooling

Typically we do pooling after a convolution layer to decrease thedimensionality. It encourages us to decrease the measure of parameters that both abbreviate training time and over-fit the data. Pooling layers down-samples every sample of the feature map exclusively, limiting height and width, and keeping the depth unaltered.

The most famous technique for pooling is max pooling, which just takes themaximum incentive value in the pooling window. In contrast with the convolution process, there are no parameters for pooling. It slides the window over its input, and it essentially takes the full value in the window. Like the convolution, we indicate the size of the window and the step.



12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

max pooling

20	30
112	37

average pooling

13	8
79	20

Figure 8: Output of max pooling and average pooling using a 2x2 window and stride 2

(c) Fully connected layer:

In the fully connected layer, every neuron from the past layer is associated with the yielded output neurons. The size of final output layer is equivalent to the quantity of classes in which the input image is to be categorized.

(d) Activation function:

Activation functions are used in to reduce the over fitting. In the CNN architecture, the ReLu activation function has been used. The advantage of the ReLu activationfunction is that its gradient is always equal to 1, which means that most of the error is passed back during back-propagation.

(e) Softmax:

The Softmax function takes a vector of N real numbers and normalizes that vectorinto a range of values between (0, 1).

(f) Dropout:

Dropout is used to avoid over-fitting, and the principle is quite basic. During the training cycle, at each progression, the neuron is voluntarily –dropped" or incapacitated with p likelihood. It implies that at the current iteration, all inputs and outputs of this neuron will be incapacitated. The drop-out neurons are re-sampled with the probability p at each training stage, so the drop-out neuron at one point can be can be active at the next iteration. Hyper parameter p is called the drop-out rate and is usually around 0.5, equivalent to 50 percent of the neurons being lost.

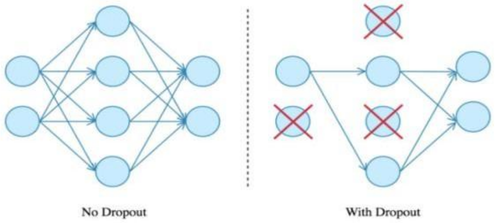


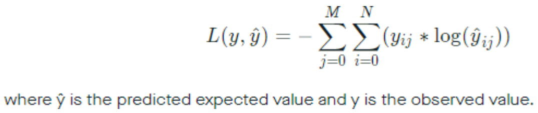
Fig 10: Visualization representation of Dropout method

(g) Batch Normalization:

The batch normalizer fosters up the training process and applies a transformation that maintains the mean activation close to 0 and the activation standard deviation close to 1.

Training the model: After the input image is preprocessed andthe necessary libraries are imported, make place holders for all the significant features removed by the CNN model and for the target variable. At that point characterize a batch size and initialize the number of epochs in the backend. Initialize X_train and Y_train datasets as validation datasets. Consolidate all the functions above into a solitary CNN organization and utilize mini- batch gradient descent for training purpose. It is an iterative optimization algorithm used to locate the best outcomes (minimum of a curve) in data science. Gradient suggest a slope tendency rate or decay rate. Descent means the descending case. The iterative approach implies we have to get the outcomes a few times so as to get the most effective outcome. The gradient descent's iterative quality helps a graphical diagram that is under fit to make the graph fit ideally to the input data.

Cross validation: Here, we use categorical entropy as the loss function. Categorical cross entropy is a loss function that is used for single label classification. Categorical cross entropy will equate the prediction distribution (activations in the output layer, one for each class) with the true distribution, where the true class likelihood is set at 1 and 0 for the other classes.



$$L(y,\hat{y}) = -\sum_{j=0}^{M}\sum_{i=0}^{N}(y_{ij} * \log(\hat{y}_{ij}))$$

where ŷ is the predicted expected value and y is the observed value.

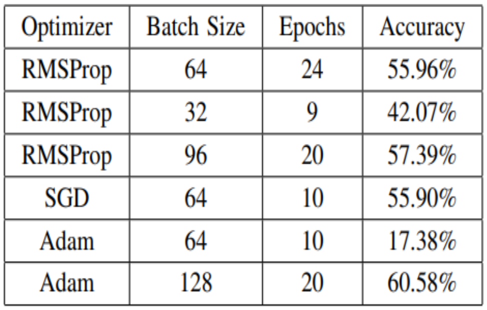
.Fig 11: Algorithmic representation ofcategorical entropy (loss function)

Hyperparameter tuning: After tuning the hyper parameters, the most elevated precision was accomplished for each optimizer. Utilizing the RMSProp optimizer, an exactness of 0.57 was reached over 20 epochs and a batch size of 96. The Stochastic Gradient Descent optimizer gave an exactness of 0.55 and it couldn't be expanded altogether by additional tuning of the hyper parameters. Utilizing the Adam optimizer with the default values, a batch size of 64 and 10 epochs leadto an astoundingly low exactness or accuracy of 0.17. Further setting the learning rate to 0.0001 and the decay rate to 10e − 6, the most noteworthy precision of 0.60 was accomplished. A comparison of the varioushyper parameters that were tuned can be seen in table beneath. Results were achieved by experimenting with the CNN algorithm. The loss over training and test set decreased with each epoch. The batch size was 128, which was kept constant in all experiments. In order to produce successful performance, the following improvements have been made to the neural network architecture:

1) Number of epochs: it has been found that the accuracy of the model has improved with an increasing number of epochs. A high number of epochs, however, resulted in over- fitting. It was concluded that eight epochs resulted in minimal over-fitting and high accuracy.

2) Number of layers: the neural network architecture consists of three hidden layers and one fully connected layer. A total of six convolution layers were built using 'relu' as the activation function.

3) Filters: The accuracy of the neural network on the data set varied with the number of filters applied to the image. The number of filters for the first two layers of the network was 64, and 128 for the third layers of the network was maintained.

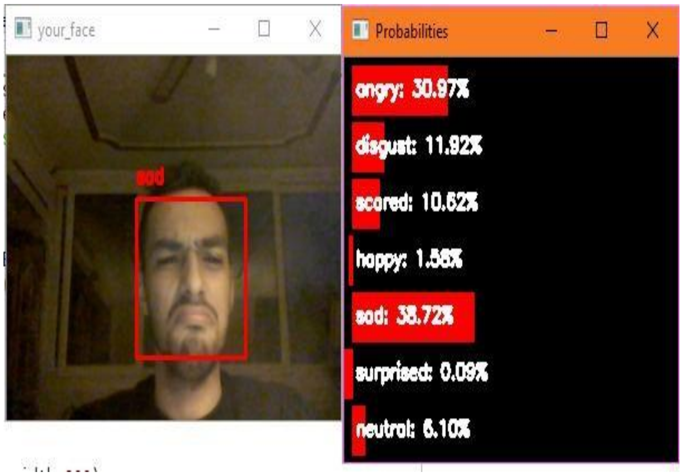


Optimizer	Batch Size	Epochs	Accuracy
RMSProp	64	24	55.96%
RMSProp	32	9	42.07%
RMSProp	96	20	57.39%
SGD	64	10	55.90%
Adam	64	10	17.38%
Adam	128	20	60.58%

Fig 12: Comparison of Hyper parameters

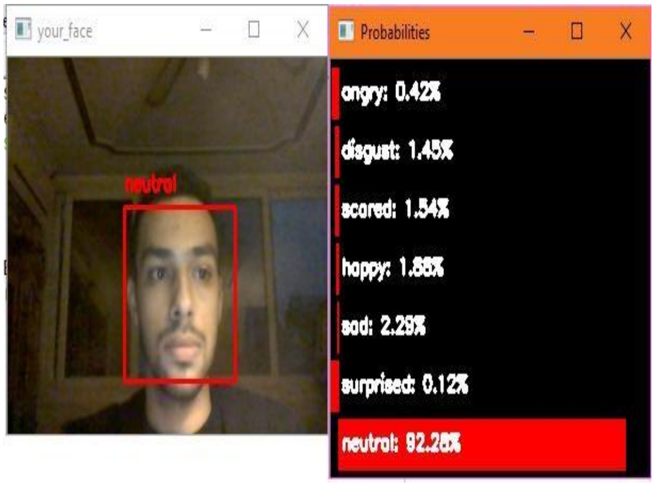
Results and Discussion

The input of the program will be an image of ahuman taken with the help of the webcam. This human face will have to display one of the following emotions happy, sad, fear, disgust, anger, surprise. The program will thenanalyze the given image with the help of the CNN and compare the result with that of the dataset. The most similar emotion from the dataset is then delivered as the final result.This can be understood by viewing the given below images. The user can display a number of emotions as follows:



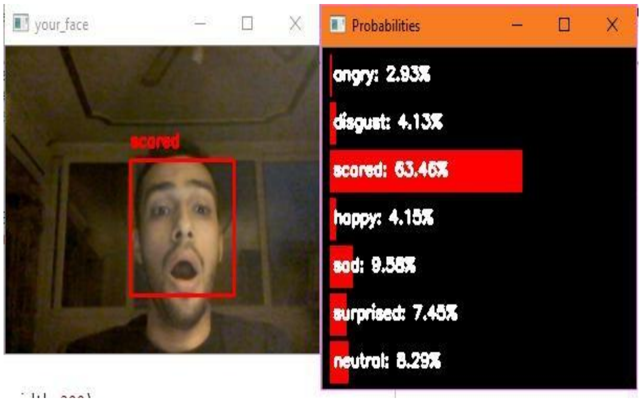
```
angry: 30.97%
disgust: 11.92%
scared: 10.62%
happy: 1.58%
sad: 38.72%
surprised: 0.09%
neutral: 6.10%
```

Fig 13: Sentiment analysis of sad face



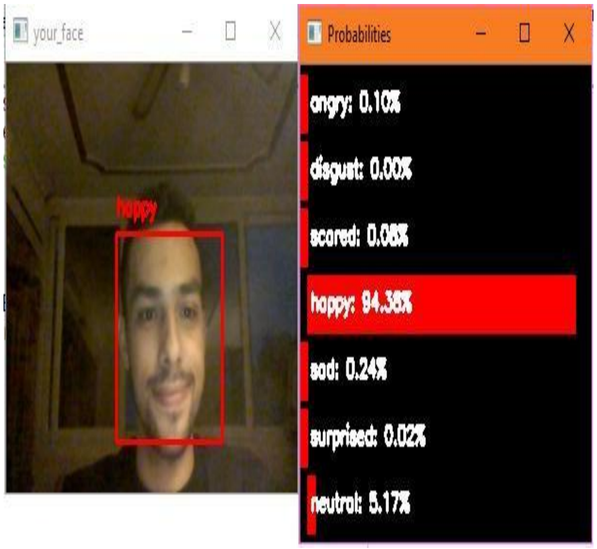
```
angry: 0.42%
disgust: 1.45%
scared: 1.54%
happy: 1.86%
sad: 2.29%
surprised: 0.12%
neutral: 92.28%
```

Fig 14: Sentiment analysis of Neutral face



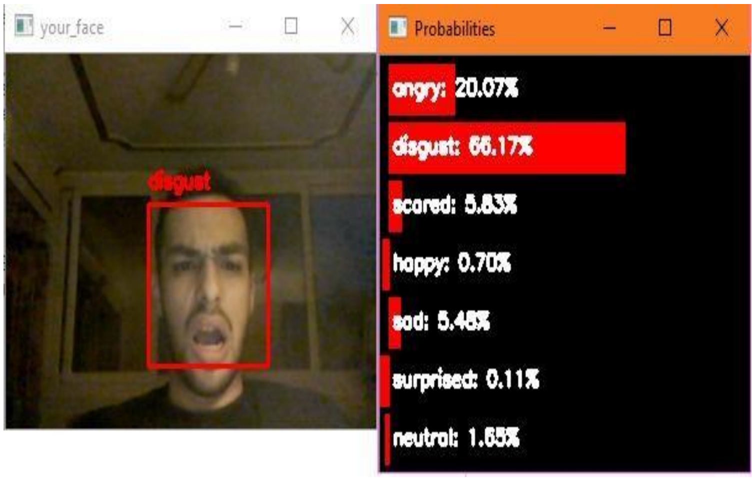
```
angry: 2.93%
disgust: 4.13%
scared: 63.46%
happy: 4.15%
sad: 9.58%
surprised: 7.45%
neutral: 8.29%
```

Fig 15: Sentiment analysis of scared face



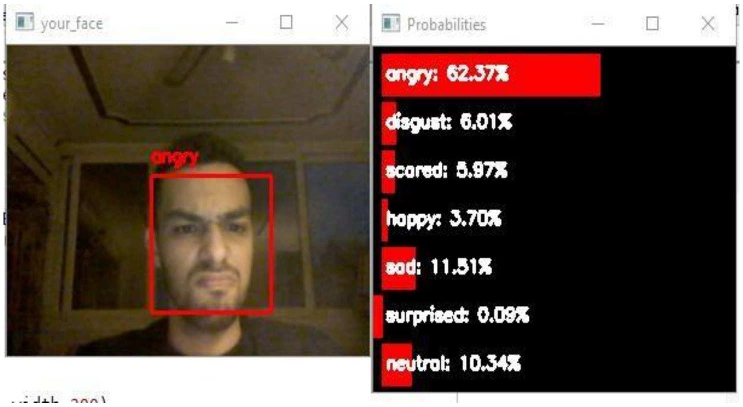
```
angry: 0.10%
disgust: 0.00%
scared: 0.08%
happy: 94.38%
sad: 0.24%
surprised: 0.02%
neutral: 5.17%
```

Fig 16: Sentiment analysis of Happy face



```
angry: 20.07%
disgust: 66.17%
scared: 5.83%
happy: 0.70%
sad: 5.48%
surprised: 0.11%
neutral: 1.65%
```

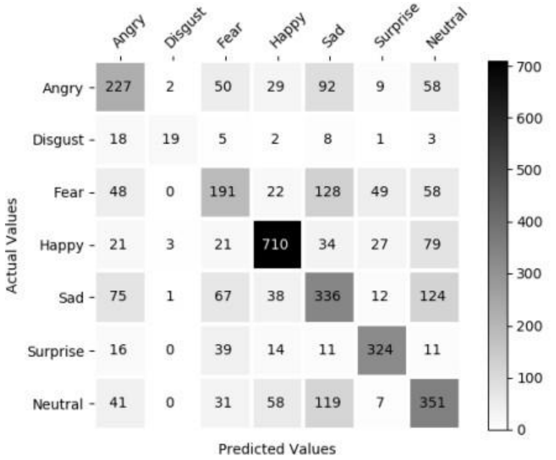
Fig 17: Sentiment analysis of Disgust face



```
angry: 62.37%
disgust: 6.01%
scared: 5.97%
happy: 3.70%
sad: 11.51%
surprised: 0.09%
neutral: 10.34%
```

Fig 18: Sentiment Analysis of Angry face

The confusion matrix generated over the test data is shown in figure 19. The dark blocks along the diagonal show that the test data has been classified well. It can be observed that the number of correct classifications is low for disgust, followed by fear. The numbers on either side of the diagonal represent the number of wrongly classified images. As these numbers are lower compared to the numbers on the diagonal, it can be concluded that the algorithm has worked correctly and achieved state of the art results.



Actual Values \ Predicted Values	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	227	2	50	29	92	9	58
Disgust	18	19	5	2	8	1	3
Fear	48	0	191	22	128	49	58
Happy	21	3	21	710	34	27	79
Sad	75	1	67	38	336	12	124
Surprise	16	0	39	14	11	324	11
Neutral	41	0	31	58	119	7	351

Fig 19: Confusion matrix represented as a heatmap

Future Enhancements

This project has a lot of scope for enhancement and further development. The main concept of this project can be integrated in an artificial intelligence based chatbot which will, based on the emotional state of the user, speak/interact with the user. This will help the user in case of an emotional distress or a medical emergency. This project can be further improved to detect the emotions of multiple faces in a single frame/photo. Detecting the emotions of multiple faces in a large crowd will help with security measures and also in events it provides a scope of improvement based on the audience reactions. The current computational capacity of the system can be enhanced by using a faster processor and a system with bigger RAM capacity. This will help reduce the time taken to detect the emotion. The number of emotions that this project can detect is limited to only seven namely, sadness, happy, surprise, disgust, neutral, anger and fear. But the number of emotions can be increased by making changes in the algorithm making it efficient enough to detect other important emotions like confusion, boredom, pain and satisfaction. It can be further enhanced to accurately detect the emotions in fast moving vehicles, despite the moving objects.

Conclusion

In this project, a face is detected with the location of face in any input image or frame within bounding box coordinates using the module OpenCv2. This human face will have to display one of the following emotions happy, sad, fear, disgust, anger, surprise. Hence, this input is then given to the model for feature extraction and classification. We designed an efficient CNN model for facial feature extraction and performed softmax classification technique for face emotion detection. Adam optimizer is used for adaptivetraining of deep neural networks. The experimental results on FER2013 dataset was demonstrated 0.6012 and a validation accuracy of 0.8978, indicating it has out- performed other basic classification models used for visual sentiment analysis. Here,categorical cross entropy is chosen as the loss function with evaluation metrics as ‘accuracy’.

References

[1] B, K.S., Rameshan, R., 2017. Dictionary Based Approach for Facial Expression Recognition from Static Images. Int. Conf. Computer Vision, Graph. Image Process. pp. 39–49.

[2] Clawson, K., Delicato, L.S., Bowerman, C., 2018. Human Centric Facial Expression Recognition. Proc. Br. HCI pp. 1–12.

[3] Cui, R., Liu, M., Liu, M., 2016. Facial expression recognition based on ensemble of multiple CNNs. Chinese Conf. Biometric Recognition. 511–518. https://doi.org/10.1007/978-3-319- 46654-5

[4] Ding, H.; Zhou, S.K.; Chellappa, R. Facenet2expnet: Regularizing a deep face recognition net for expression recognition. In Proceedings of the 12th IEEE International Conference on Automatic Face & Gesture Recognition, Washington, DC, USA,30 May–3 June 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 118– 126.

[5] Ernst H (1934) Evolution of facial musculature and facial expression. J Nerv Ment Dis 79(1):109

[6] Faria, D.R.; Vieira, M.; Faria, F.C. Towards the development of affective facial expression recognition for human-robot interaction. In Proceedings of the 10th International Conference on PErvasive Technologies Related to Assistive Environments, Island of Rhodes, Greece, 21–23 June 2017; pp. 300–304.

[7] Levi, G.; Hassner, T. Emotion recognition in the wild via convolutional neural networks and mapped binary patterns. In Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, Seattle, WA, USA, 9–13 November 2015; pp. 503–510

[8] Lopez-Rincon, A. Emotion recognition using facial expressions in children using the NAO Robot. In Proceedings of the International Conference on Electronics, Communications and Computers (CONIELECOMP), Cholula, Mexico, 27 February–1 March 2019; IEEE: Piscataway, NJ, USA; pp. 146–153.

[9] Lu, G.; He, J.; Yan, J.; Li, H. Convolutional neural network for facial expression recognition. J. Nanjing Univ. Posts Telecommun. 2016, 36, 16–22. 8. Zeng, J.; Shan, S.; Chen, X. Facial expression recognition with inconsistently annotated datasets. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 222–237

[10] Lyons, M.J.; Akamatsu, S.; Kamachi, M.; Gyoba, J. Coding facial expressions with Gabor wave. In Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 14–16 April 1998; pp. 200–205.

[11] Masi, I.; Wu, Y.; Hassner, T.; Natarajan, P. Deep face recognition: A survey. In Proceedings of the 31st SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Paraná, Brazil, 29 October–1 November 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 471–478.

[12] Ng, H.W.; Nguyen, V.D.; Vonikakis, V.; Winkler, S. Deep learning for emotion recognition on small datasets using transfer learning. In Proceedings of the 2015 ACM on International Conference on Multimodal Interaction, Seattle, WA, USA, 9–13 November 2015; pp. 443–449.

[13] Zhang, Z.; Luo, P.; Loy, C.C.; Tang, X. From facial expression recognition to interpersonal relation prediction. Int. J. Comput. Vis. 2018, 126, 550–569.[CrossRef]

[14] Zhao, X.; Liang, X.; Liu, L.; Li, T.; Han, Y.; Vasconcelos, N.; Yan, S. Peak-piloted deep network for facial expression recognition. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 425–442. Sensors 2020, 20, 2393 19 of 21