

Human Emotion Recognition using Deep Learning with Special Emphasis on Infant's Face

Parismita Sarma^{1*}, Takrim UL Islam Laskar², Dankan Gowda V³ and Ramesha M⁴

¹Assistant Professor, Department of Information Technology, Gauhati University, Guwahati, Assam, India, pari@gauhati.ac.in

²Department of Information Technology, Gauhati University, Guwahati, Assam, India, takrimulislam@gmail.com

³Department of Electronics and Communication Engineering, BMS Institute of Technology and Management, Bangalore, Karnataka, India, dankan.v@bmsit.in

⁴Assistant Professor, Department of Electronics and Communication Engineering, GITAM School of Technology, GITAM (DEEMED TO BE UNIVERSITY), Bengaluru, Karnataka, India, rameshmalur037@gmail.com

*Correspondence: Parismita Sarma; pari@gauhati.ac.in

ABSTRACT- This paper discusses a deep learning-based image processing method to recognize human emotion from their facial expression with special concentration on infant's face between one to five years of age. The work has importance because most of the time it becomes necessary to understand need of a child from their facial expression and behavior. This work is still a challenge in the field of Human Facial Emotion Recognition due to confusing facial expression that sometimes found in some of the samples. We have tried to recognize any facial expression into one of the mostly understood human mood namely Angry, Disgust, Fear, Happy, Sad, Surprise and Neutral. For this purpose, we have trained an image classifier with Convolutional Neural Network with Kaggle's Fer2013 Dataset. After the completion of the project, we achieved good accuracy in most of the prominent emotions by testing with 20 random images for each emotion.

Keywords: Convolution Neural Network, Emotion detection, Deep Neural Network, Haar Cascade Frontal Face Detector.

ARTICLE INFORMATION

Author(s): Parismita Sarma, Takrim UL Islam Laskar, Dankan Gowda V and Ramesha M;

Received: 17/11/2022; **Accepted:** 15/12/2022; **Published:** 20/12/2022;

e-ISSN: 2347-470X;

Paper Id: IJEER 1711-07;

Citation: 10.37391/IJEER.100466

Webpage-link:

www.ijeer.forexjournal.co.in/archive/volume-10/ijeer-100466.html



Publisher's Note: FOREX Publication stays neutral with regard to Jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

Emotion recognition from facial expression is an important area of research now a days. Human are capable of recognizing emotion or mood of a person from his/her speech and facial expression. In recent years sentiment analysis research is progressing with the help of advanced technology like machine learning, deep learning and transfer learning [1]. This research field has many practical implications and basically used in the field of machine learning *i.e.*, Human Computer Interaction, systems relating to Driver Assistance, Cognitive Science *etc.*

Facial emotion recognition can be done using different techniques. A big dataset can help us to use dense network and it is one step above deep neural network that gives more accurate result compared to other machine learning techniques. In general, for all techniques face based model is used where different features of human faces are extracted for emotion detection.

1.1 Deep Neural Network

Neurons are the fundamental units used for construction of neural networks. A number of neurons form a layer are linked to the neurons of subsequent layer.

Together they form the data path which travels throughout the whole network. Every neuron acts upon a mathematical estimation, transmit the result to all the connected neurons [2]. Deeper the layers, the nodes of the last few layers become more learned. One of the special designs of Artificial Neural Network used in present days for training is CNN (Convolutional Neural Network). The layers in CNN can be assumed as a type of feature extractor. This network exhibits extra ordinary strength to any type of classification.

1.2 Emotion Classification

Human being poses different types of emotions. We are emotional by habit, starting from infants to older ones, we express our emotions by shaping different parts on our face like eyes, chick, chin *etc.* In this work we have tried to classify mostly known seven human emotions to seven groups. We are trying to focus on correct emotion recognition for the toddlers as they are more expressive from their facial expression. But there are a few more emotions that are a bit difficult to recognize, for example doubting and confusing state of a mind. Accuracy of these emotions shows better result when using both tune of their speech and facial expression. That is why in this work we are considering only seven mostly understandable emotions. More emphasize is put on to detect child's emotions from their facial expressions, we have filtered our dataset accordingly. The samples of children's facial images are of mainly age group between one to five. The emotion classes which we have considered are 'angry', 'disgust', 'fear', 'happy', 'sad', 'surprise', 'neutral' also shown in the *table 1*.

Table 1: Classification of Emotions

Classes	Emotions
0	Angry
1	Disgust
2	Fear
3	Happy
4	Sad
5	Surprise
6	Neutral

1.3 Motivation

The primary objective of this study is to develop a neural network-based intelligent system capable of identifying children's emotions. Most of the time human expresses feelings spontaneously that causes automatic change in shape and size of different semi movable parts of our face like chick, eyebrow, tongue, lips, eye size *etc.* A precise recognition system greatly helps doctors and parents to understand physical and mental health of the child. According to surveys 7% meaning of a message is understandable through verbal statement, 38% is understood from vocal analysis and most amazingly remaining 55% is attained from facial expression [2].

1.4 Objective

One of the primary goals of this study is to correctly recognize human precisely toddler's emotions from real time videos and photographs. We have used here Deep Convolution Network; input frames are taken from real-time video as input image. This study uses the kaggle fer 2013 labelled dataset. We are using this dataset to classify above mentioned seven emotional classes.

1.5 Contribution of this Work

This system has potentiality in health care sector like computerized scrutiny of mental and physiological state of children or adult by recognizing facial expressions in real-time. Toddler express their emotions mostly by their facial expression. They cry when they are hurt. Though child's facial expression is real to their feelings, sample collection is challenging task to collect these samples. Yet with the limited dataset we were able to get good accuracy for each emotion class but there is scope to improve. Several methods for recognising human emotions have been explored in this paper. For the purpose of creating the FER method, we analysed the FER approaches and other processes. Based on this body of research, it is clear that a combination of cutting-edge methods is required to build an autonomous facial expression detection mechanism that would be both efficient and accurate.

2. LITERATURE SURVEY

To formulate and achieve maximum accuracy we have studied a number of recent journal papers and decided to use deep learning as proposed architecture. A short overall description is mentioned here. Researchers are focusing on new architectures which are mostly based on CNN and Deep learning. Among them a researcher AndreeaPascu under guidance of Prof. Ross King found a high accuracy of 86% using machine learning technique [3]. From source [4] we come to know that that

image structure and quality of the samples influence the recognition rate. They experimented with different deep learning techniques and CNN constructions like VGG-16 and ResNet50, Support Vector Machine (SVM) as the classifier. They achieved only 31.8% accuracy with KDFF datasets. Later they were able to improve it more by combining the output of two neural network layers and accuracy was 67.2% on kaggle dataset and 78.3% for KDFF dataset [4]. In the research paper [5] by H. Qin et al. a new approach is observed where a system is proposed and was trained jointly with R-CNN and RPN. At first stage they trained Risk Priority Number (RPN) then adjust the model. ROIs are generated from RPN training phase and used for training the joint network with R-CNN. A precision of 98.22% is achieved in this model. In Another paper [6] by N. Zeng et. al. a new approach for facial expression recognition is discussed. Three descriptors HOG (Histogram of Oriented Gradients), LBR (Local Binary Pattern) and gray values were used in that approach. By using Deep SparseAuto-encoders on CK+ (Extended Cohn-Kanade) dataset anger, disgust and contempt expression were evaluated with 99-99% accuracy. But for fear and sadness it was 86-87%, happy and surprise mood exhibited most 98-100% accuracy. According to paper [7] by M.H Siddiqi et al. curvelet transform can be used to extract different features from digital images and they have used it in their proposed work. The system is able to extract the curves from different prominent features of human face taken from digital images. This concept is used for image reconstruction purpose also. They used HMM for labeling different human expressions and achieved accuracy of about 99%. A paper by S.K.AKamarol, et al. [8] implemented a method called Spatio Temporal Texture Map (STTM) which was able to construct a three dimensional (3D) texture map. According to them this method is much better than any other look-based feature extraction techniques like CK+, AFEW dataset known for lesser computational cost. By using sensors, 3D facial recognition is able to record the face's contours with greater accuracy. Unlike more conventional approaches, 3D face recognition scans may be performed in low- or no-light conditions without sacrificing accuracy.

3. SYSTEM SPECIFICATION

3.1 Tools

In this work we have used python 2.7 which is an open-source high level language. We are using Open-source computer vision library (open CV) that is free for academic purpose. It takes help of multi core processing and written in C/C++. Open CV has interface of java, python and C++ [9].

Tensor flow is open source software libraries that are basically used through a series of tasks for dataflow purpose [10]. It is the mostly used and perfect math library for deep neural network. It can run on multiple CPUs as well as GPUs. It can be used in mobile operating system also. For recording of real time images our system has in built camera attached with the laptop. Keras and Jupiter Notebook are two tools used in this work.

3.2 Dataset

As already mentioned, we have used here fer2013.csv (Kaggle Facial Expression Recognition Dataset) consisting of 48x48

size 35888 digital images of different faces in gray scale. There are 28,709 photos in the training set, and 3,582 in the testing set [11]. For classification purpose we have used Haar-cascade classifier. It is an object detection algorithm mostly used in machine learning. This strategy involves the training of a cascade function for identification of objects from many constructive and negative images. This algorithm is known for its capability to detect faces and body components from digital images. In the first stage Haar-cascade algorithm needs huge amount of positive images (with faces) and negative images (not faces) for training purpose. The link to the dataset is <https://www.kaggle.com/c/challenges-in-representation-learning-facial-expression-recognition-challenge/data>.

3.3 Environment

Google Colaboratory is a free interactive environment based on the Jupyter notebook, which we are using. It runs on cloud and does not need any set up [12]. It allows to write code and executes in computer languages like python. It has facility to run GPUs for massive computations as training dataset amid Deep learning.

4. METHODOLOGY

Our proposed system will identify seven basic emotions. The prominent stages of the whole system are as follows:

- First stage is image capturing. We are using this tool for real time human emotion detection. So, the frame taken from real-time video is input image.
- Then we have used Haar Cascade frontal face detector and it will detect emotion of the face plotted in the image. In this case Region of Interest (ROI) is the face itself.
- The dataset used here is kaggle fer2013 labelled dataset. It is used to train a data model with CNN that classifies the above-mentioned dataset into seven emotional classes. This model is now usable for detection of emotion of the face in the image.

After classification is completed probability of detection for individual emotion classes is plotted. The class with highest percentage is tagged as resultant emotion. In *figure 1* we can see a block schematic of the system we propose.

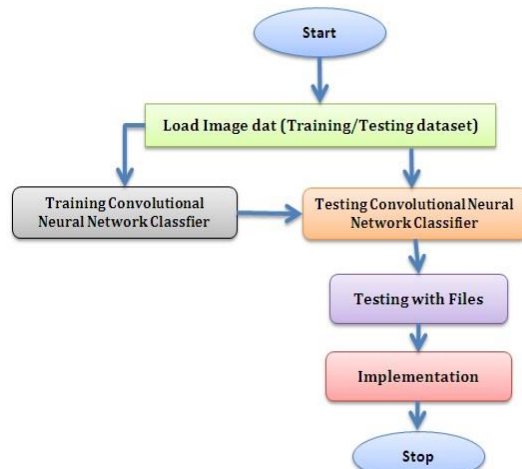
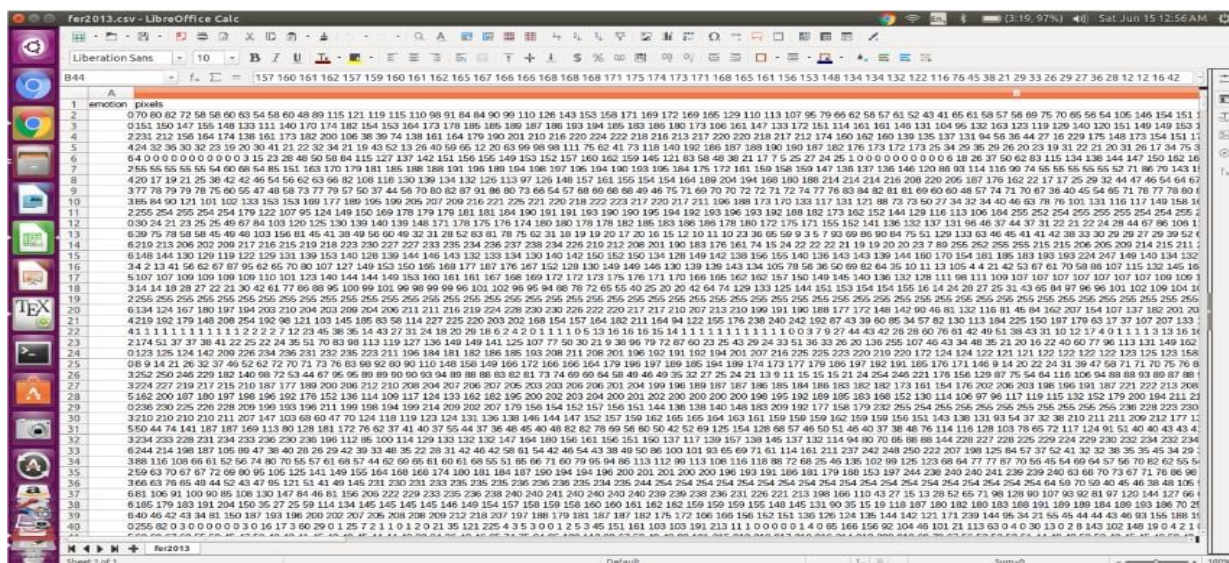


Figure 1: Schematic Representation of the Proposed System's Main Components

4.1 Loading Image Dataset

According to our work plan, the first phase consists of training and testing that performs on our repository. The image dataset is uploaded to the google drive. A jupyter notebook has been created in the Google colab with Python 3. Google colab can access these data from google drive as it is authorized for the same. The dataset used here is in form of three columns: Emotion, Pixel and Usage. The Emotion column has values from 0 to 6, it represents seven emotion classes. Pixel column correspond to pixel values of the 48 X 48-pixel images. Last the usage column gives the purpose of every data's individual instances. The rows have pixel values of one instance for one image. The *figure 2* shows a snapshot of emotion column and pixel values.

**Figure 2: Snapshot of Emotion and Pixel of the Dataset**

4.2 Storing Data in .csv File

All the data stored in fer2013.csv file are read and stored in an array 'content'. Next it is converted to numpy array and stored into a new array 'lines'. We have calculated the number of occurrences. Next the length of occurrence is calculated. The strategies we have used are

Arrays x_train, y_train, x_test, y_test and all are initialized. For each instance at a time the line is split into emotion class, image data and usage as shown below.

```
emotion, img, usage = lines[i].split(",")
```

Then the pixel values are split with reference to an empty space " " and stored in an array "val" and converted to numpy array "pixels" that is shown below.

```
val = img.split(" ")
pixels = np.array(val, 'float32')
```

The emotion is then categorised to its class with keras. Similarly, for testing dataset emotion is appended to y test and pixels to x test. y test.append(emotion)

```
x test.append(pixels)
```

The above assigned values are then converted to numpy arrays with 'float32' data type x_train. x_test pixel values are divided by 255 and reassigned the respective values to normalize input between [0,1]. For Training dataset, the emotion class is appended to the y_train and respective pixels are appended to x_train.

```
y_train.append(emotion)
```

```
x_train.append(pixels)
```

4.3 EPOCHS and BATCHES

We have taken 25 epochs to train the dataset. One Epoch means a complete process of passing the dataset through the neural network. We have assigned 256 batch size to each epoch. That means each epoch will divide the whole dataset into batches with each batch processed 256 instances.

4.4 Constructing Convolutional Neural Network Structure

In the sequential CNN model structure, there are four layers [13]. It is called sequential because the model has to know the proper input shape it is supposed to expect. The specification of each layer is as follows:

1st Layer -The first layer added is 2D CNN layer with 64 filters/kernels, kernel size (5, 5), activation function 'relu' and input shape 48 X 48 in grayscale. Then we added an average pooling layer of pool size (5,5) and stride size (2,2).

2nd Layer - The second layer is added with 64 filters/kernels, kernel size (3,3) and activation function 'relu'. Then we added an average pooling layer of pool size (3,3) and stride size (2,2).

3rd Layer - We have increased the filters/kernels from 64 to 128 in the third layer. Then by using flatten operation, which have converted the array to have 1-D shape.

4th Layer - Next the final dense layer is added with 1024 inputs from all the previous neurons. We have added Dropout value 0.2 and activation function 'relu'. There are two such neurons. Final dense layer is added with the total number of classes and activation function 'softmax'.

Rectified Linear Unit (ReLU): If the activation function is given a negative input, it will return 0, but it will return the same result if the input is positive [14]. That is,

$$f(n) = \max(0, x)$$

Softmax Activation Function: It produces multiple outputs for 1 input array [15]. This helps to build a model which can classify more than 2 classes.

4.4.1 Epochs, Batch Processing and Model Generation

ImageDataGenerator() Function has been initialized and data flow has been defined with x_train, y_train and batch size are assigned to variable 'train_generator'.

The model is then compiled to calculate the categorical_crossentropy (loss) and accuracy.

The model generation is then started with train generator as **model.fit_generator(train generator, steps per epoch=batch size, epochs=epochs)** with batch size and epochs as parameter.

The system has been run to obtain the trained model. The image of the training process has been shown in the figure 3.

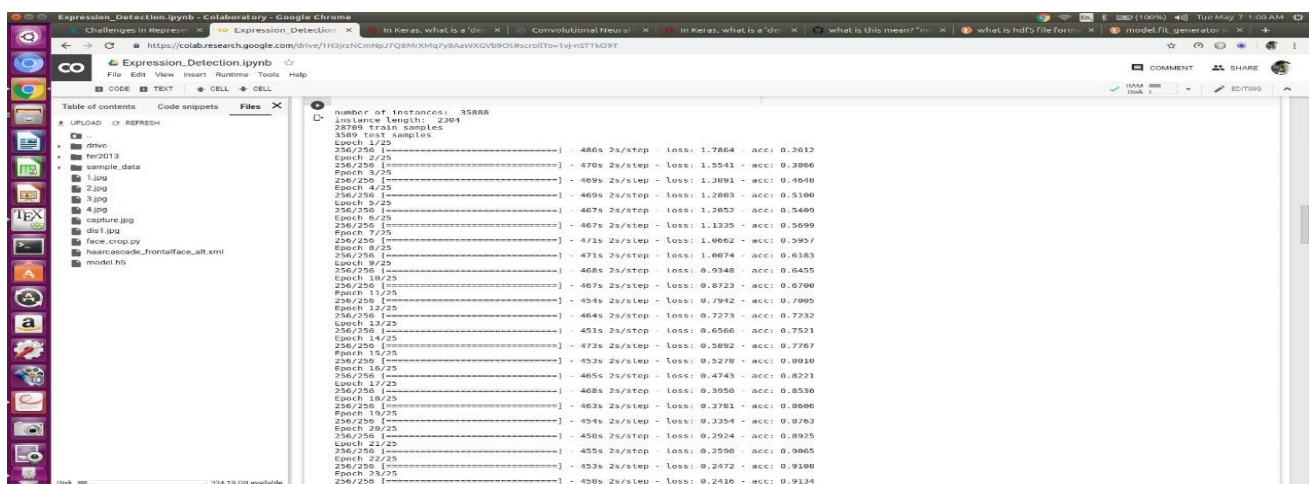


Figure 3: Training Data Set to Obtain a Model

The trained model is then saved in a Hierarchical Data Format file with extension “h5” as “model.h5”. HDF5 / H5 (Hierarchical Data Format) is a file format designed to organize and store huge amount of data. It contains two major objects dataset and group.

- Datasets of multidimensional array with homogeneous type.
- Groups are the container structures which are capable of storing dataset and other groups.

4.5 Training Time on GPU and CPU

When we trained the image classifier, we tried different configurations. Firstly, we used only 56 CPUs and found that the time required was 3 hours and 15 minutes. Then we tried adding a GPU and the time consumption was only 4 minutes. And then the result of testing with 1 GPU and 1 CPU was shocking, as the time was 3 minutes, which was less by 1 (approx.) minute than that of the with 1 GPU and 56 CPUs. Table 2 clearly describes this scenario. In presence of GPU if more CPUs are assigned process will consume more time. It is because more processes assigned to CPUs for parallel processing and as a result increases time taken by CPUs.

Table 2: Time required by CPUs and GPU

GPU	CPU	TIME (hh:mm)
0	56	03:15
1	56	00:04
1	1	00:03

4.6 Evaluation

Figure 4 shows the evaluation of train, test loss and accuracy of our system.

```

Train loss: 0.08908920260003199
Train accuracy: 97.40151172106309
Test loss: 2.668699996389696
Test accuracy: 58.26135413847317

```

Figure 4: Evaluation of Testing, Training loss and Accuracy

4.7 Monitoring Test Set Results

Table 3 below shows the mapping of different emotion class to the Emotion Index used in our work.

Table 3: Mapping of Emotion Class to Emotion Index in the Array

Class number	Emotion	Emotion index in array
0	Angry	0
1	Disgust	1
2	Fear	2
3	Happy	3
4	Sad	4
5	Surprise	5
6	Neutral	6

For visualizing and monitoring the test set results, x_{test} (pixels) has been passed through predict function.

predictions = model.predict(x_{test})

The pixel values from 20 to 30 instances have been retrieved and image have been reconstructed. Figure 5 shows reconstruction of Twentieth test dataset and plotted along with the percentage of accuracy graph. The bar graph in the figure 5 detects an angry face.

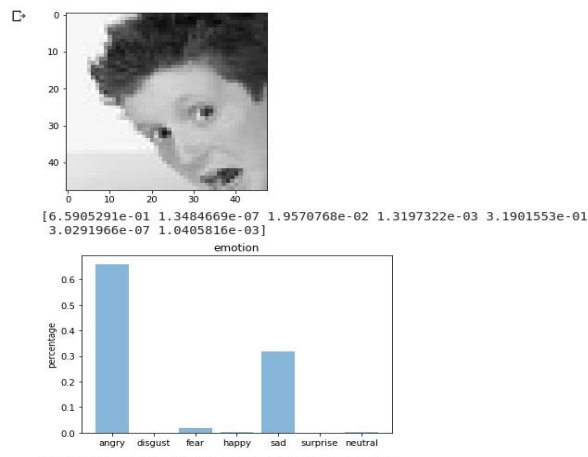


Figure 5: Images with Reconstruction of Twentieth Test Dataset

We have used Haar-Cascade frontal face detector for source image face detection [16]. More focus is given on Haar-like features compared to pixel intensity. This method considers adjacent rectangular region and search for total differences to classify the feature region (subsections) of the image [16].

Cascade classifier is initialized by loading the pre-trained cascade file ‘haarcascade_frontalface_alt.xml’ then passing through openCV cascade classifier. Next the image is passed through multi scale detector and hence all the faces in the image are detected. Faces are next bounded by rectangular boxes, cropped and stored. These cropped images are then converted to gray scale for emotion detection. Figure 6 is the screen shot of a detected face with Haar_Cascade classifier.

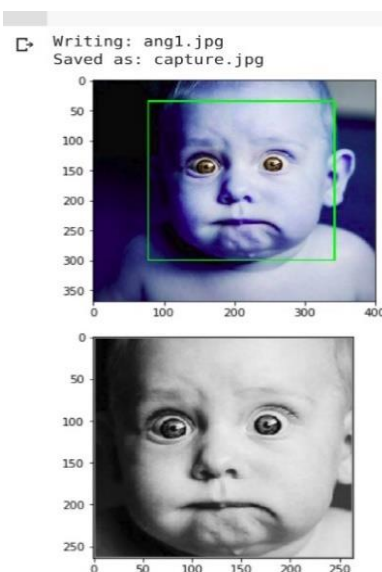


Figure 6: Face detection with Haar_Cascade classifier

4.8 Defining the Emotion Analysis Function

An array of emotions is created, for each emotion there is its index number that corresponds to emotion class number. That is:

`objects = ('angry', 'disgust', 'fear', 'happy', 'sad', 'surprise', 'neutral')`

4.9 Prediction of Emotions

The stored or real-time images are loaded in gray scale with 48 X 48 resolution and finally converted to a linear array containing all the pixel values. It is also made sure that no pixel value exceeds 255 bits.

This array is then passed through the prediction function and stored in an array 'custom'. This array 'custom' contains seven classes with the probability of every emotion. This is then passed to the emotion analysis function defined earlier for plotting using matplotlib.

5. RESULTS AND CONCLUSION

We have shown below a number of figures (screenshots) each one recognizing the appropriate emotion from the facial expression predicted from our system. From the analysis of the report, we have seen that Disgust and Fear produced accuracy less than 70% as these emotions resemble with each other most often. Disgust resembles with anger and sadness; fear resembles with surprise.

Model: "sequential_12"

Layer (type)	Output Shape
conv2d_60 (Conv2D)	(None, 44, 44, 64)
max_pooling2d_12 (MaxPooling)	(None, 20, 20, 64)
conv2d_61 (Conv2D)	(None, 18, 18, 64)
conv2d_62 (Conv2D)	(None, 16, 16, 64)
average_pooling2d_24 (Average)	(None, 7, 7, 64)
conv2d_63 (Conv2D)	(None, 5, 5, 128)
conv2d_64 (Conv2D)	(None, 3, 3, 128)
average_pooling2d_25 (Average)	(None, 1, 1, 128)
flatten_12 (Flatten)	(None, 128)
dense_36 (Dense)	(None, 1024)
dropout_24 (Dropout)	(None, 1024)
dense_37 (Dense)	(None, 1024)
dropout_25 (Dropout)	(None, 1024)
dense_38 (Dense)	(None, 7)

Figure 7: Screenshot of classification model with output parameter

Some of our output samples with snapshots are shown in figure 7, to figure 18. The bar graph shown in every emotion is predicted emotion class, the highest valued emotion class is the resultant one. From those bar graphs we can conclude that figure... shows the angry emotion, figure... for fear, figure ... for happy, figure ... for sad, figure ... for surprise and figure.... for neutral emotion. All the screenshots are shown below.

5.1 Classification Report

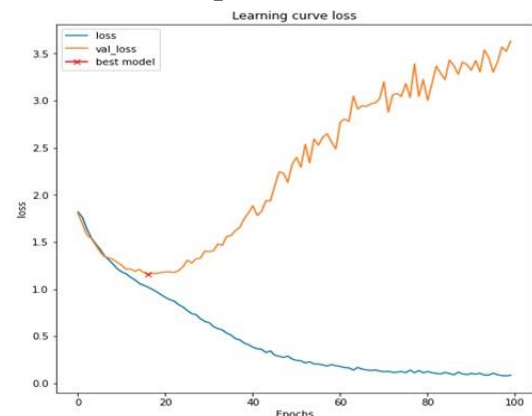


Figure 8: Learning curve showing the best model

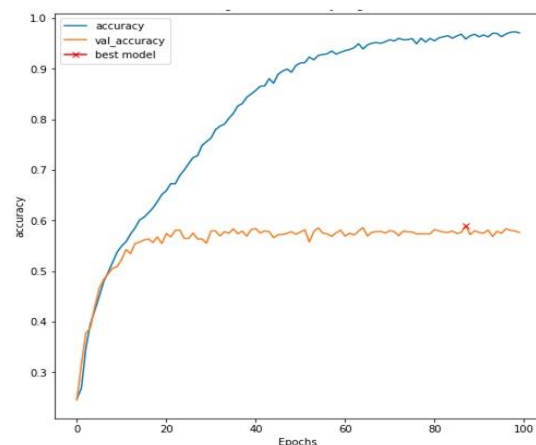


Figure 9: Accuracy curve of the best model

Confusion Matrix

	0	1	2	3	4	5	6
0	262	11	72	41	69	14	61
1	9	26	4	3	3	3	1
2	44	6	221	36	76	45	47
3	26	1	37	665	43	40	63
4	74	6	124	58	294	14	105
5	14	1	40	15	9	278	11
6	69	1	47	63	94	20	323

Figure 10: Confusion Matrix

	Precision	recall	f1-score	support
0	0.49	0.53	0.51	498
1	0.53	0.50	0.51	52
2	0.47	0.41	0.43	545
3	0.76	0.75	0.76	881
4	0.44	0.50	0.47	588
5	0.76	0.67	0.71	414
6	0.52	0.53	0.53	611
accuracy			0.58	3589
macro avg	0.57	0.56	0.56	3589
Weighted avg	0.58	0.58	0.58	3589

Figure 11: Classification Report

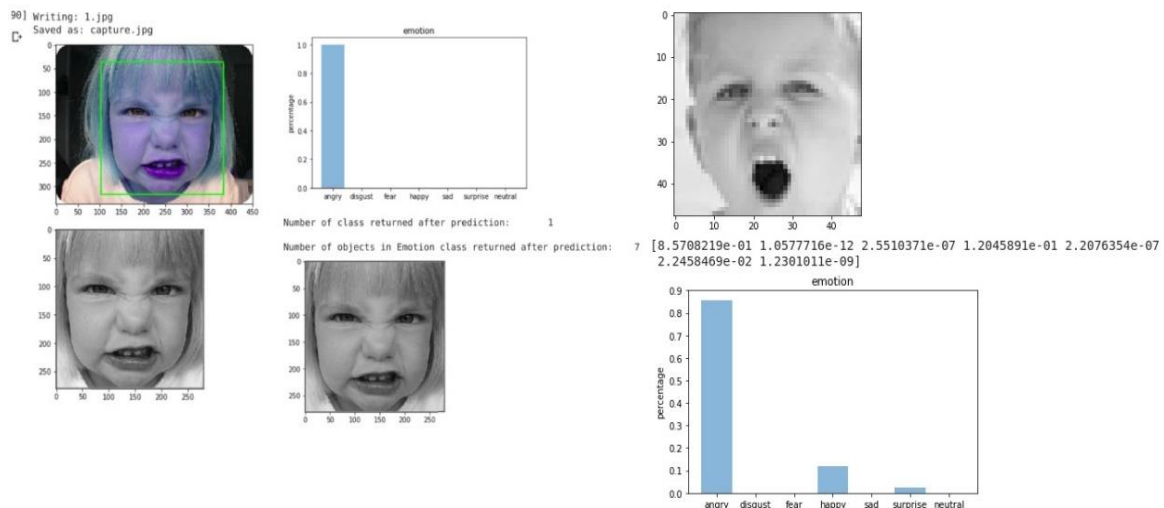


Figure 12: Output of random test for angry emotion

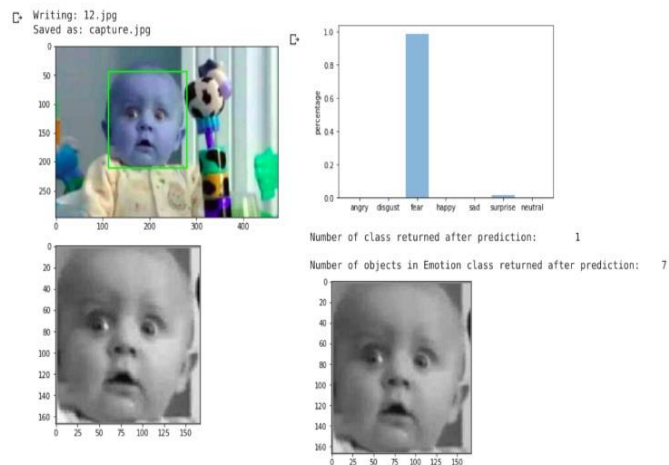


Figure 13: Output of random test for Fear emotion

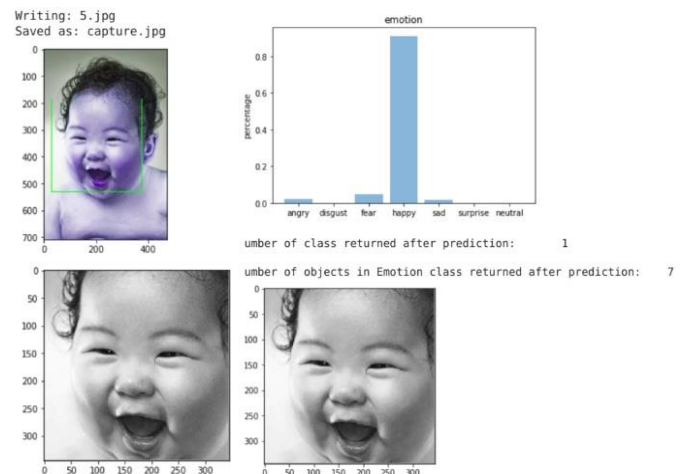


Figure 15: Screenshot of Happy emotion

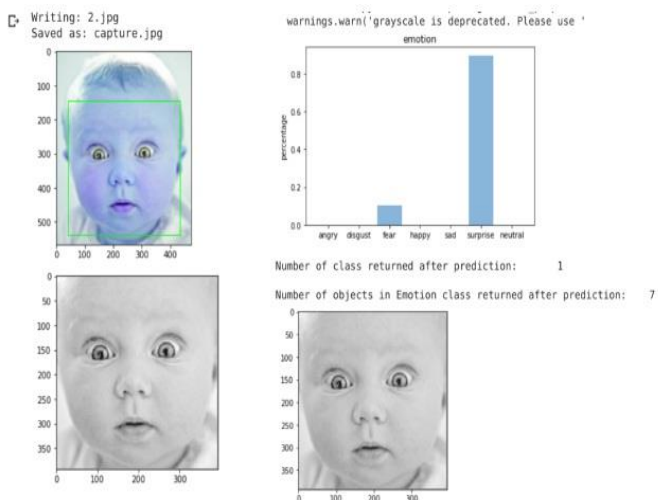


Figure 14: Output of random test for Surprise emotion

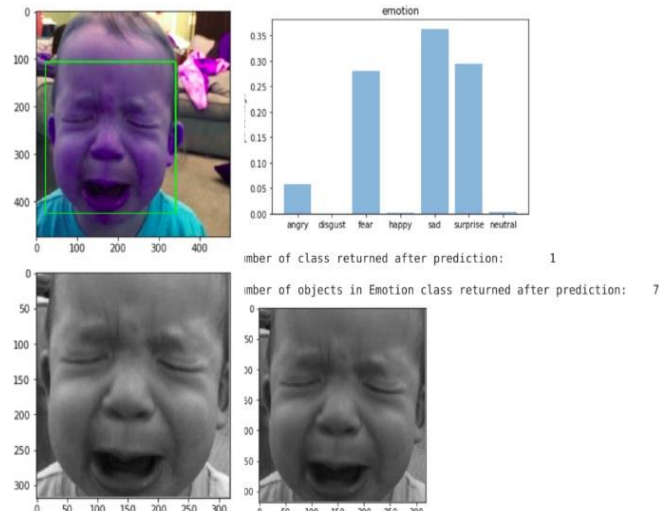


Figure 16: Screenshot of sad emotion

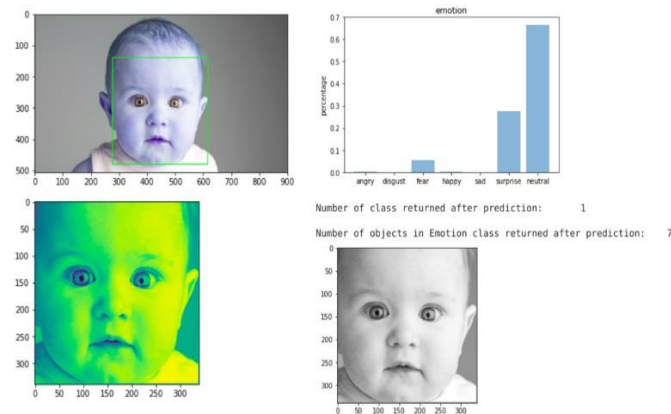


Figure 17: Screenshot of Neutral Emotion

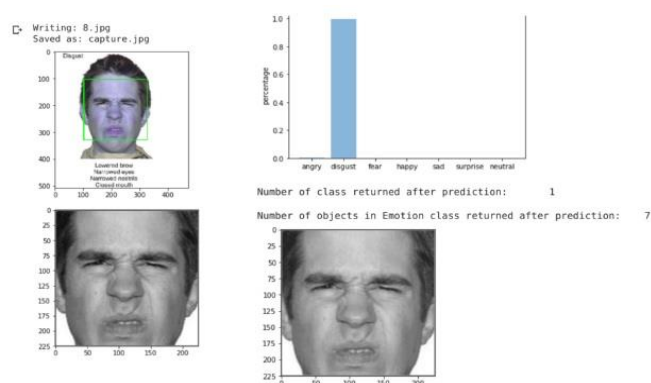


Figure 18: Screenshot of Disgust emotion

5. CONCLUSION

In this research work, more emphasis is given to emotion classification in the context of facial images, speech, and brain signals. Several deep learning models are evaluated, including Convolutional and Recurrent neural networks. An evaluation clearly shows that deep learning models are superior to traditional models for image and signal classification. This work focuses on developing a system that improves the accuracy of Deep Learning models for emotion recognition. Experimental results for each technique are analysed to highlight the most significant aspects of each approach.

REFERENCES

- [1] Avinash Sharma, and A. Azhagu Jaisudhan Pazhani, A novel method of data compression using ROI for biomedical 2D images, Measurement: Sensors, Volume 24, 2022, 100439, ISSN 2665-9174.
- [2] Mayur Rahul, Namita Tiwari, Rati Shukla, Devvrat Tyagi and Vikash Yadav (2022), A New Hybrid Approach for Efficient Emotion Recognition using Deep Learning. IJEER 10(1), 18-22. DOI: 10.37391/IJEER.100103.
- [3] Shashank M Gowda and H N Suresh (2022), Facial Expression Analysis and Estimation Based on Facial Salient Points and Action Unit (AUs). IJEER 10(1), 7-17. DOI: 10.37391/IJEER.100102.
- [4] I. Goodfellow, D. Erhan, P.-L. Carrier, A. Courville, M. Mirza, B. Hamner, W. Cukierski, Y. Tang, D. Thaler, D.-H. Lee, Y. Zhou, C. Ramaiyah, F. Feng, R. Li, X. Wang, D. Athanasakis, J. Shawe-Taylor, M. Milakov, J. Park, R. Ionescu, M. Popescu, C. Grozea, J. Bergstra, J. Xie, L. Romaszko, B. Xu, Z. Chuang, and Y. Bengio, "Challenges in representation learning: A report on three machine learning contests,"

Available at: <http://arxiv.org/abs/1307.0414>, Accessed on 11 March, 2021.

- [5] T. Nath, A. Mathis, A. C. Chen, A. Patel, M. Bethge and M. W. Mathis "Using DeepLabCut for 3D markerless pose estimation across species and behaviors", Nature protocols, Vol. 14, No. 7, pp. 2152-2176, 2019.
- [6] H. Sadr, M. M. Pedram and M. Teshnehlab "A robust sentiment analysis method based on sequential combination of convolutional and recursive neural networks", Neural Processing Letters, Vol. 50, No.3, pp.2745-2761, 2019.
- [7] M. Khan, S. Chakraborty, R. Astya & S. Khepra "Face Detection and Recognition Using OpenCV", proceedings in 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS), IEEE, pp. 116-119, 2019.
- [8] Z. Qin, H. Ye, G. Y. Li and B. H. F. Juang "Deep learning in physical layer communications", IEEE Wireless Communications, Vol. 26, No. 2, pp. 93-99, 2019.
- [9] X. Jiang, Y. Pang, X. Li, J. Pan and Y. Xie "Deep neural networks with elastic rectified linear units for object recognition", Neurocomputing, Vol. 275, pp. 1132-1139, 2018.
- [10] Agarap, A. F. "Deep Learning using Rectified Linear Units (ReLU)", Cornell University, arXiv preprint arXiv: 1803.08375, 2018.
- [11] N. Zeng, H. Zhang, B. Song, W. Liu, Y. Li, and A. M. Dobaie, "Facial expression recognition via learning deep sparse autoencoders", Neurocomputing, Vol. 273, pp. 643-649, 2018.
- [12] A. Savoiu and J. Wong, "Recognizing facial expressions using deep learning," Stanford University, 2017.
- [13] S. K. A. Kamarol, M. H. Jaward, J. Parkkinen, and R. Parthiban, "Spatiotemporal feature extraction for facial expression recognition", IET Image Processing, Vol. 10, No. 7, pp. 534-541, 2016.
- [14] M. Rabeie and A. Gasparetto, "System and method for recognizing human emotion state based on analysis of speech and facial feature extraction; applications to human-robot interaction," in 2016 4th International Conference on Robotics and Mechatronics (ICROM). IEEE, pp. 266-271, 2016.
- [15] Pascu, A., Intelligence, B. A., & King, R. "Facial Expression Recognition System", University of Manchester, 2015.
- [16] D. Lapakko "Communication is 93% nonverbal: An urban legend proliferates", Communication and Theater Association of Minnesota Journal, Vol. 34, No. 1, 2007.



© 2022 by the Parismita Sarma, Takrim UL Islam Laskar, Dankan Gowda V and Ramesha M. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).