# Behaviour Based Data Dispatcher

**Mohan Sai S, Mona Teja K, Akhil Amudala, Sai Karthik M, Swarnalatha P**

***Abstract: Human life is a complex social structure. It is not possible for the humans to navigate without reading the other persons. They do it by identifying the faces. The state of response can be decided based on the mood of the opposite person. Whereas a person's mood can be figured out by observing his emotion (Facial Gesture). The aim of the project is to construct a "Facial emotion Recognition" model using DCNN (Deep convolutional neural network) in real time. The model is constructed using DCNN as it is proven that DCNN work with greater accuracy than CNN (convolutional neural network). The facial expression of humans is very dynamic in nature it changes in split seconds whether it may be Happy, Sad, Angry, Fear, Surprise, Disgust and Neutral etc. This project is to predict the emotion of the person in real time. Our brains have neural networks which are responsible for all kinds of thinking (decision making, understanding). This model tries to develop these decisions making and classification skills by training the machine. It can classify and predict the multiple faces and different emotions at the very same time. In order to obtain higher accuracy, we take the models which are trained over thousands of datasets.***

***Keywords: Deep Convolution Neural Networks, Tensor Flow, Vgg16, Mobile Net, CV2, Haar Cascades.***

## I. INTRODUCTION

This paper proposes a method for recognition of the emotions on the face using Convolutional Neural networks using a live camera. Face emotions are one of the most informative gestures. So, we make use of these gestures on the face for the emotion recognition of the person. The recent trends show that the deep learning applications have got noticeable performance in the computer vision applications. And the convolutional neural networks have outperformed in the field of image classification. In this paper we have used VGG-16 Model and MobileNet Architecture. These architectures are the state of art architecture for the image classification. Face Recognition and the Facial emotion recognition are the similar problems. In the face recognition system, the model recognizes using the facial features of the person. And in the facial emotion recognition the model identifies the emotion of the face expressed by person using facial features and the nodal points. In this implementation we have used fine-tuning of the model. Which will support the model for learning in improving the feature extraction.

This paper analysis the performance of the model on the different CNN architectures.

In this paper Section II discuss about the Methodology carried out and also discuss about the dataset and pre-processing of the data and the architectures that are used in the model. Section III discuss about the Proposed System. Section IV discuss about the Results and Analysis. Section V discuss about the conclusion part. And the Section VI lists all the references that have been used in this paper.

## II. OBJECTIVES

### PRIMARY OBJECTIVE

Facial Emotion Recognition of multiple faces and different expressions in real time and Dispatch data when needed (*).

### SECONDARY OBJECTIVES

Importing and Integration of Modules and Accessing the system components (Webcam) etc.

Initiate the session and allot the memory fragments, Collection of continuous frames from the video capturing device to recognize faces, Classification and prediction of emotions in realtime.

Merging the data with several simple modules and forming a customized hybrid module. Here we have tried parsing, dispatching the data and alerting the persons when needed.

## III. METHODOLOGY

### Dataset

For Training the version. I actually have used FER 2013(Dataset) from Kaggle, we additionally have attempted with numerous different datasets. The statistics includes 48x48 pixel grayscale pictures of faces. The faces were robotically registered in order that the face is extra or much less centered and occupies approximately the equal quantity of area in every photograph. The project is to categorize every face primarily based on the emotion that is classified into seven categories (zero=Angry, 1=Disgust, 2=Fear, three=Happy, 4=Sad, five=Surprise, 6=Neutral). Train.Csv includes two columns, "emotion" and "pixels". The "emotion" column contains a numeric code starting from zero to six, inclusive, for the emotion that is present in the photograph. The "pixels" column includes a string surrounded in prices for every photo. The contents of this string a space-separated pixel values in row essential order. Test.Csv contains simplest the "pixels" column and your venture is to expect the emotion column. The schooling set includes 28,709 examples. The public take a look at set used for the chief board consists of 3,589 examples. The very last take a look at set, which became used to decide the winner of the competition, consists of every other three,589 examples. The people who organized this data set have been 0graciously supplied the workshop organizers with a preliminary version in their dataset to apply for this contest.

Revised Manuscript Received on November 15, 2019

**Mohan Sai S,** UG Student, School of Computer and Engineering, VIT Vellore, India.

**Mona Teja K,** UG Student, School of Computer and Engineering, VIT Vellore, India.

**Akhil Amudala,** UG Student, School of Computer and Engineering, VIT Vellore, India.

**Sai Karthik M,** UG Student, School of Computer and Engineering, VIT Vellore, India.

**Swarnalatha P,** Associate professor, School of Computer and Engineering, VIT Vellore, India.

For schooling the version Cohn-Kanade dataset, a unfastened and publicly available dataset on the internet, for training and trying out the face reputation machine is used.

### PRE-PROCESSING

Mostly, the images available in face databases contain some irrelevant details from classification point of view (e.g. background, hair etc). The region comprising of only face information is termed as the relevant area of interest and it should be cropped before feature extraction. To find relevant area, center of the chin right and left corners of the eyes, three-point corners are selected on the face.

To make triangle these three points are linked with each other. Then, the computational process is being performed on centroid of the triangle (say, C) and distance of a corner from centroid (say, P). From centroid C, a square of length (say, 2P) is drawn. Within this square, captured area of the image is the desired face image. Now, Gray scale images are generated from the cropped face images. The relevant cropped area i.e. face image are of different size, and hence images are further resized to 230X230 pixels by using bi-cubic interpolation technique.

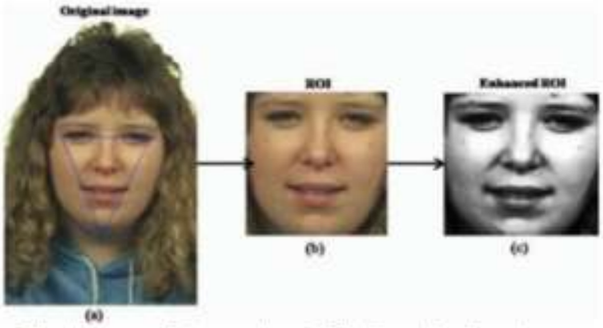


**Fig-1 Extracted face region (ROI) from database image**

## IV. ACHITECTURES USED VGG16 MODEL

For the order of Dataset, the model structure that we have utilized is a Convolutional Neural System which comprises of two Convolutional layers. It is an irregular weight appointed in the start of the preparation procedure. In the wake of finding the highlights the pooling is done which used to lessen the dimensionality of the picture. This procedure is additionally conveyed till we acquire the enhanced outcomes. These highlights acquired are given as contribution to the completely associated neural system where the arbitrary loads are doled out first and foremost. As the preparation progress the loads get balanced or refreshed utilizing the backpropagation calculation. For more precision VGG16 Model can be conveyed which has additionally won the ImageNet rivalry in 2012 by yielding high exactness.

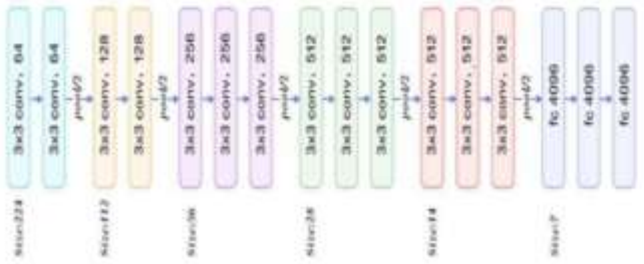


**Fig-2 VGG16 Model (CNN Architecture)**

## V. MOBILENET ARCHITECTURE

We present a class of effective models called Versatile Nets for portable and inserted vision applications. Versatile Nets depend on a streamlined engineering that utilizations profundity savvy divisible convolutions to fabricate light weight profound neural systems. We present two basic worldwide hyper-parameters that proficiently exchange off among inertness and precision. These hyper-parameters enable the model developer to pick the privilege estimated model for their application dependent on the imperatives of the issue. We present broad investigations on asset and exactness exchange offs and show solid execution contrasted with other famous models on ImageNet arrangement. We at that point exhibit the adequacy of Portable Nets over a wide scope of utilizations and use cases including object discovery, fine grain order, face traits and enormous scale.

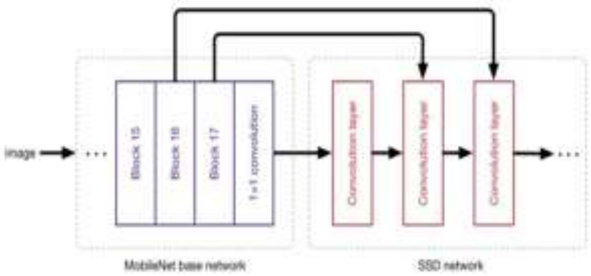


**Fig-3 Mobile Net Model (CNN Architecture)**

A lot of appearances with comparing marks are bolstered into a classifier, which after preparing, it learns and predicts the feeling class for another face.

### KERAS MODEL:

Keras is an open source neural network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, or Theano. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. It was developed as part of the research effort of project ONEIROS (Open-ended Neuro-Electronic Intelligent Robot Operating System), and its primary author and maintainer is François Chollet, a Google engineer.

In 2017, Google's TensorFlow team decided to support Keras in TensorFlow's core library. Chollet explained that Keras was conceived to be an interface rather than a standalone machine-learning framework. It offers a higher-level, more intuitive set of abstractions that make it easy to develop deep learning models regardless of the computational backend used. Microsoft added a CNTK backend to Keras as well, available as of CNTK. Keras contains numerous implementations of commonly used neural network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier. Keras allows users to productize deep models on smartphones (iOS and Android), on the web, or on the Java Virtual Machine.

It also allows use of distributed training of deep learning models on clusters of Graphics Processing Units (GPU) and Tensor processing units (TPU).

A tensor processing unit (TPU) is an AI accelerator application- specific integrated circuit (ASIC) developed by Google specifically for neural network machine learning. A graphics processing unit (GPU) is a specialized electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. Modern GPUs are very efficient at manipulating computer graphics and image processing, and their highly parallel structure makes them more efficient than general-purpose CPUs for algorithms where the processing of large blocks of data is done in parallel.

## VI. PROPOSED SYSTEM

This project aims to predict the emotions of the person by his/her facial expression and swap the appropriate emoticon in place of the face.

There are seven types of human emotions shown, that are universally recognised across different cultures namely anger, happiness, disgust, fear, sadness, surprise and contempt.

**Preparing the dataset**

It involves the preparation of dataset upon which the learning algorithm will work. The Model will be using different datasets of faces available on internet example CK+ dataset and some images followed by applying it to convolution neural networks. The model will be trained to recognize different emotions.

**Detecting faces**

This is finished by HAAR course work in OpenCV In the wake of recognizing the appearances the picture is changed over to greyscale and is resized to a similar size as the pictures in dataset. A Haar Course is fundamentally a classifier which is utilized to recognize specific objects from the source. The haarcascade_frontalface_default.xml is a Haar course planned by OpenCV to recognize the frontal face. This Haar course is accessible on GitHub. A Haar Course works via preparing the course on a huge number of negative pictures with the positive picture superimposed on it. The Haar course is fit for distinguishing highlights from the source. Learning to Recognize Emotions.

This step involves training the program to recognise and differentiate between the emotions. Training is done by Convolution Neural Network, a deep learning algorithm and will be on the layers on dataset of faces.

**Learning to Recognize Emotions**

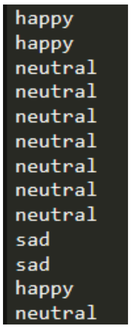
This is the major step and it involves placing of the appropriate emoticon over the face of the person according to his/her emotion. The HAAR cascade function returns the coordinates of the faces detected and these coordinates can be used to place the emoticon at the exact place.

**Swapping emoticons in place of faces**

This is the major step and it involves placing of the appropriate emoticon over the face of the person according to his/her emotion. The HAAR cascade function returns the coordinates of the faces detected and these coordinates can be used to place the emoticon at the exact place.

**Parsing the Data and emotions frame by frame**

Now we will parse the data frame by frame and the output in the form of emotion. Now we will set a threshold for certain emotion for which it dispatches the data if the threshold is breached.

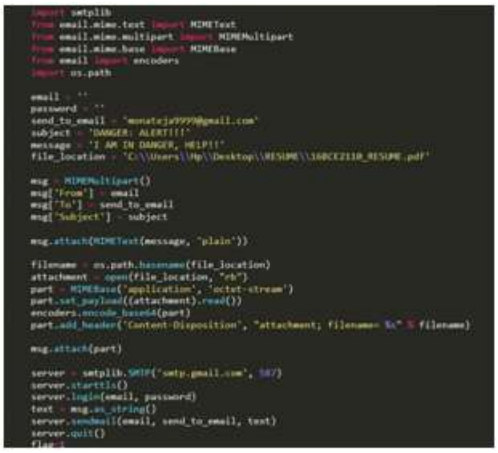


```
happy
happy
neutral
neutral
neutral
neutral
neutral
neutral
neutral
sad
sad
happy
neutral
```

**Fig-4 Parsing Emotions**

**Dispatching the data**

If Certain threshold is breached then we will dispatch the data of the emotions frame by frame and any other important data that needs to be shared with specific people using automated python script.



```python
import smtplib
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email import encoders
import os.path

email = ''
password = ''
send_to_email = 'monateja9999@gmail.com'
subject = 'DANGER: ALERT!!!'
message = 'I AM IN DANGER, HELP!!'
file_location = 'C:\\Users\\Hp\\Desktop\\RESUME\\16BCE2110_RESUME.pdf'

msg = MIMEMultipart()
msg['From'] = email
msg['To'] = send_to_email
msg['Subject'] = subject

msg.attach(MIMEText(message, 'plain'))

filename = os.path.basename(file_location)
attachment = open(file_location, "rb")
part = MIMEBase('application', 'octet-stream')
part.set_payload((attachment).read())
encoders.encode_base64(part)
part.add_header('Content-Disposition', "attachment; filename= %s" % filename)

msg.attach(part)

server = smtplib.SMTP('smtp.gmail.com', 587)
server.starttls()
server.login(email, password)
text = msg.as_string()
server.sendmail(email, send_to_email, text)
server.quit()
flag=1
```

**Fig-5 Python Script for Dispatching**

## VII. RESULTS AND DISCUSSION

| MODEL | IMAGE NET ACCURACY | MILLION MULT ADDS | OTHER PARAMETERS (in millions) |
|---|---|---|---|
| VGG16 | 71.5% | 15300 | 138 |
| GOOGLENET | 69.8% | 1550 | 6.8 |
| MOBILENET | 70.6% | 569 | 4.2 |

**Table-1: Comparison between the model**

| NETWORK | CLASSIFICATION TIME (TENSOR FLOW 1.5) | CLASSIFICATION TIME (TENSOR FLOW 1.8) |
|---|---|---|
| DenseNet201 | 658 | 628 |
| InceptionV3 | 175 | 184 |
| ResNet | 301 | 299 |
| MobileNet | 131 | 135 |

**Table-2: Performance comparison of TensorFlow**

The outcomes of the table clearly show the difference between various architectures and their accuracy in the prediction.

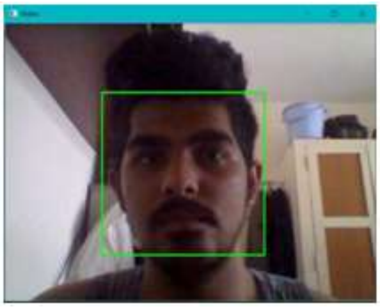


Fig-6 Single Face Detected

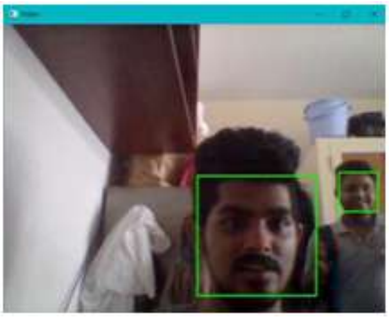
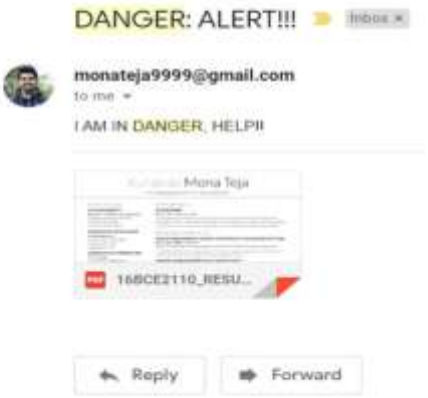


Fig-7 Multiple Faces Detected

As we are using supervised Learning the most relevant outcome in the class is given as output It may be ['fear', 'neutral', 'angry', 'surprise', 'happy', 'sad'].

Fig-8 Emotions that are Trained

The data is dispatched finally after the threshold limit is breached. Now we are clearly depicting the dispatching of the data.



DANGER: ALERT!!!

monateja9999@gmail.com

to me

I AM IN DANGER, HELP!!

Fig-9 Dispatching of data

## GRAPHICAL ANALYSIS

### Graph-1

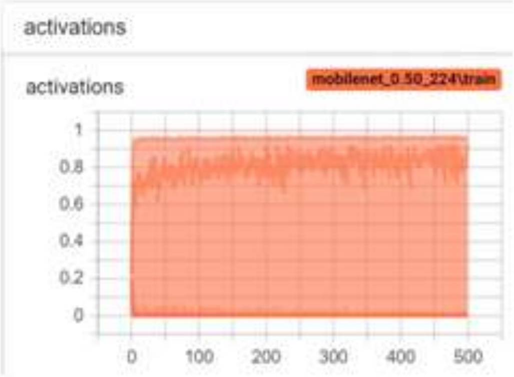


Fig-10 Activations while training the model

### Graph-2

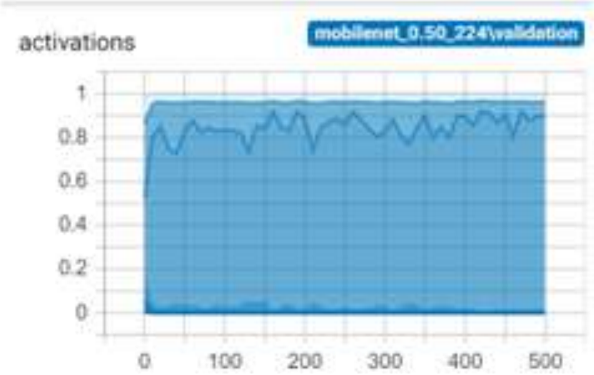


Fig-11 Activations while validating the model

### Graph-3

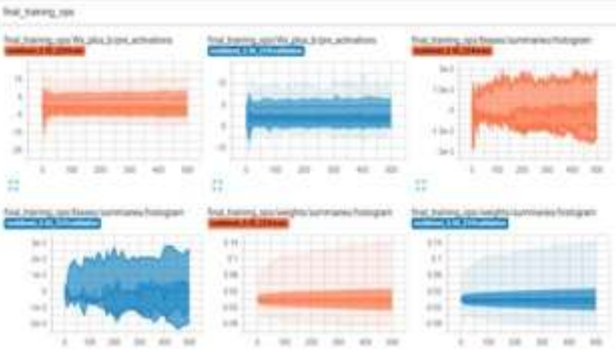


Fig-12 Final Training Ops Histograms

### Graph-4

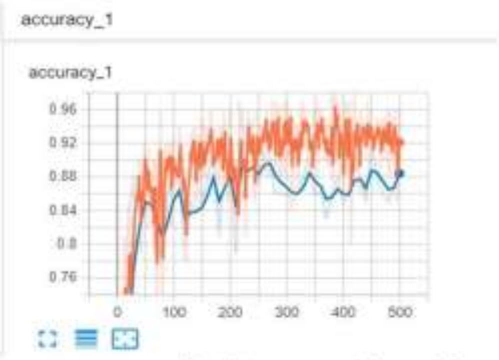


Fig-13 Accuracy of the model

**Graph-5**

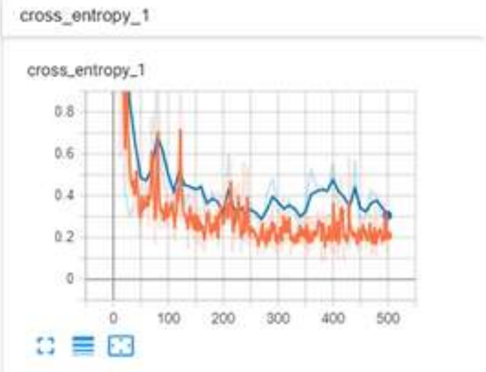
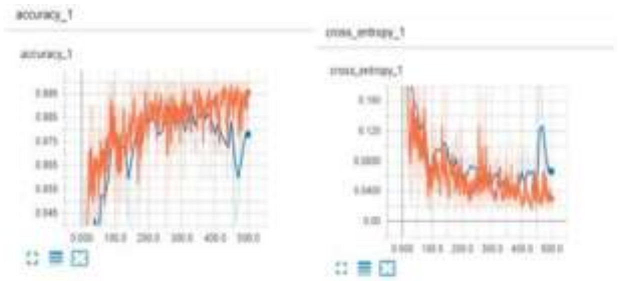


**Fig-14 Cross Entropy of the model**

While training, the performance of the model can be viewed and analysed using TensorBoard in the local host which is shown in the below figure. The accuracy and other detailed statistics can also be analysed and visualized by using the TensorBoard.



The two graphs represent the accuracy and cross entropy. The accuracy graph depicts how well the model is classifying in the training phase. we can see that the accuracy gradually increases (orange line represents the accuracy for the training data and the blue line indicates the accuracy of the validation data).

Whereas the second half of the system shows the cross entropy which represents the learning of the model the network weights gets updated as the training gets processed. Here the cost function is decreased. Cost function represents the difference between the actual and the predicted outcome. The Following graphs shown above are tracked down during the training process and in order to visualize these graphs we have used tensor board. The procedure to obtain this graphical analysis is clearly demonstrated below.

## VIII. CONCLUSION

This paper proposes a new method for finding the facial emotions of the persons and its application in different situations. Which is a kind of behavior based emotional dispatcher. We believe that the proposed implementation has given promising results for finding the emotion with different convolutional layers. We have also deployed a visualization for the emotional learning process. This can be further extended by using different data sources and many architectures and attention models.

## REFERENCES

1. Alpaydin, E. (2014). *Introduction to machine learning*. MIT press.
2. Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.
3. Newmark, C. (2013). Charles Darwin: The Expression of the Emotions in Man and Animals. In *Hauptwerke der Emotionssoziologie* (pp. 85-88). Springer VS, Wiesbaden.
4. Ekman, P., Friesen, W. V., & Hager, J. C. (2002). Facial action coding system: The manual on CD ROM. *A Human Face, Salt Lake City*, 77-254.
5. Adelson, E. H. (1986). The extraction of spatio-temporal energy in human an machine vision. In *Proceedings from the workshop on motion representation and analysis, Charleston, SC, May 7-9^< th>, 1986*.
6. Baron, R. J. (1981). Mechanisms of human facial recognition. *International Journal of Man-Machine Studies*, *15*(2), 137-178.
7. Bichsel, M. (1991). *Strategies of robust object recognition for the automatic identification of human faces* (Doctoral dissertation, ETH Zurich).
8. Brunelli, R., & Poggio, T. (1993). Face recognition: Features versus templates. *IEEE transactions on pattern analysis and machine intelligence*, *15*(10), 1042-1052.
9. Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., & Taylor, J. G. (2001). Emotion recognition in human-computer interaction. *IEEE Signal processing magazine*, *18*(1), 32-80.
10. Fan, Y., Lu, X., Li, D., & Liu, Y. (2016, October). Video-based emotion recognition using CNN-RNN and C3D hybrid networks. In *Proceedings of the 18th ACM International Conference on Multimodal Interaction* (pp. 445-450). ACM.
11. Ng, H. W., Nguyen, V. D., Vonikakis, V., & Winkler, S. (2015, November). Deep learning for emotion recognition on small datasets using transfer learning. In *Proceedings of the 2015 ACM on international conference on multimodal interaction* (pp. 443-449). ACM