FACIAL EMOTION DETECTION USING CONVOLUTIONAL NEURAL NETWORKS

This project report is submitted for the partial fulfillment of the degree of

Bachelor of Science in Computer Science

Submitted by

Tanmoy Das **Reg.No. 1131911400411**

Utsav Saha **Reg.No.1131911400403**

Soumik Choudhury **Reg.No. 1131911400408**

Under the guidance of

Dr. Rajib Sarkar

Department of Computer Science

WEST BENGAL STATE UNIVERSITY

Kolkata - 700126

Derozio Memorial College

July 2022

DECLARATION

we hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is our own work and contains nothing which is the outcome of work done in collaboration with others, exceptas specified in the text and Acknowledgements.

——————————

Tanmoy Das
Reg. No. 1131911400411
Computer Science DepartmentWest Bengal State University

——————————

Utsav Saha
Reg. No. 1131911400403
Computer Science Department
West Bengal State University

——————————

Soumik Choudhury
Reg. No. 1131911400408
Computer Science Department
West Bengal State University

West Bengal State University

Department of Computer ScienceBarasat,
Kolkata – 7000126

CERTIFICATE

This is to certify that the project entitled **"Facial Expression Detection "** has been prepared according to the regulation of the Degree of Bachelor of Science in Computer Science and the candidates have partially fulfilled the academic session 2019-2022.

———————————

Dr. Rajib Sarkar
Project Supervisor
Computer Science Dept.
Derozio Memorial College
West Bengal State University

———————————

Dr. Rajib Sarkar
Head of Department Computer Science Dept.
Derozio Memorial College West Bengal State University

Candidates:

———————————

Tanmoy Das
Reg. No 1131911400411
Computer Science Dept. West Bengal State University

———————————

Soumik Choudhury
Reg. No. 1131911400408
Computer Science Dept. West Bengal State University

———————————

Utsav Saha
Reg. No. 1131911400403
Computer Science Dept. West Bengal State University

West Bengal State University

Department of Computer ScienceBarasat,
Kolkata – 7000126

CERTIFICATE OF APPROVAL

The foregoing project report entitled "**Facial Expression Detection** " is hereby approved as a creditable study of B.Sc.(Hons.) subject carried out and presented in a manner satisfactory to warrant its acceptance as a prerequisite to the degree for which it has been submitted. It is understood by this approval the undersigned do not necessarily endorse or accept every statement made, opinion expressed or conclusion drawn therein but approve the report only for the purpose for which it has been submitted.

———————— ————————

Internal Examiner ***External Examiner***

Candidates:

————————	————————	————————
Tanmoy Das	**Utsav Saha**	**Soumik Choudhury**
Reg. No. 1131911400411	Reg. No. 1131911400403	Reg. No. 1131911400408

Acknowledgements

It is a matter of pleasure for us to be assigned with this project work. We have put our knowledge and effort in the best possible manner. First and foremost,we want to express our sincere thanks and gratitude to Dr. Rajib Sarkar for his persistent interest, constructive criticism and encouragement throughout the project. And we greatfully acknowledge our deepest gratitude to Dr. Rajib Sarkar for his guidance and input which made this project successful. We want to thank our teacher Mr. Laxmi Kant Rana as under his guidance we first came to know about the capabilities of python progamming language which we areusing in this project. We also want to thank Dr. Papri Saha and Mr. Debashish Chatterjee for providing us with the necessary components needed to build this project. It was due to their consistent support that enabled us to make the theFacial Expression what it is today. Finally, We want to thank our parents for providing us with unfailing support and continuous encouragement throughout our years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

——————————

Tanmoy Das
Reg. No. 1131911400411

——————————

Utsav Saha
Reg. No. 1131911400403

——————————

Soumik Choudhury
Reg. No. 1131911400408

CONTENTS

Table of Contents

ABSTRACT

Human emotions are the mental state of feelings and are spontaneous. There is no clear connection between emotions and facial expressions and there is significant variability making facial recognition a challenging research area. Features like Histogram of Oriented Gradient (HOG) and Scale Invariant Feature Transform (SIFT) have been considered for pattern recognition.

These features are extracted from images according to manual predefined algorithms. In recent years, Machine Learning (ML) and Neural Networks (NNs) have been used for emotion recognition. In this report, a Convolutional Neural Network (CNN) is used to extract features from images to detect emotions.

The Python Dlib toolkit is used to identify and extract 64 important landmarks on a face. A CNN model is trained with grayscale images from the FER 2013 dataset to classify expressions into five emotions, namely happy, sad, neutral, fear and angry. To improve the accuracy and avoid overfitting of the model, batch normalization and dropout are used. The best model parameters are determined considering the training results.

The test results obtained show that CNN Model is **80% accurate for five emotions** (happy, sad, angry, fear, Neutral) **and 72% accurate for Seven emotions** (happy, sad, angry, neutral, fear, Surprise, Disgust).

LIST OF FIGURES

LIST OF TABLES

ABBREVATIONS

HOG	Histogram of Oriented Gradient
SIFT	Scale Invariant Feature Transform
ML	Machine Learning
NN	Neural Network
CNN	Convolutional Neural Network
FER 2013	Facial Emotion Recognition 2013 Dataset
FER	Facial Emotion Recognition
AI	Artificial Intelligence
DL	Deep Learning
EEG	Electroencephalograph
HCI	Human Computer Interaction
FE	Feature Extraction
CV	Computer Vision
RNN	Recurrent Neural Network
MMOD	Maximum Margin Object Detection
NumPy	Numerical Python
ELU	Exponential Linear Unit
API	Application Programming Interface
TP	True Positive
FP	False Positive
FN	False Negative
TN	True Negative
ANN	Artificial Neural Network
LR	Learning Rate
RNN	Recurrent Neural Network

CHAPTER 1: INTRODUCTION

Facial emotions are important factors in human communication that help to understand the intentions of others. In general, people infer the emotional state of other people, such as joy, sadness and anger, using facial expressions and vocal tones. Facial expressions are one of the main information channels in interpersonal communication. Therefore, it is natural that facial emotion research has gained a lot of attention over the past decade with applications in perceptual and cognitive sciences . Interest in automatic Facial Emotion Recognition (FER) has also been increasing recently with the rapid development of Artificial Intelligent (AI) techniques. They are now used in many applications and their exposure to humans is increasing. To improve Human Computer Interaction (HCI) and make it more natural, machines must be provided with the capability to understand the surrounding environment, especially the intentions of humans. Machines can capture their environment state through cameras and sensors. In recent years, Deep Learning (DL) algorithms have proven to be very successful in capturing environment states . Emotion detection is necessary for machines to better serve their purpose since they deliver information about the inner state of humans. A machine can use a sequence of facial images with DL techniques to determine human emotions .

1.1 Motivation

AI and Machine Learning (ML) are widely employed in many domains. In data mining, they have been used to detect insurance fraud . In clustering based data mining was used to identify patterns in stock market data. ML algorithms have played a significant role in pattern recognition and pattern classification problems such as FER, Electroencephalography (EEG) and spam detection . ML can be used to provide cost-effective, reliable and low computation time FER solutions .

1.2 Facial Emotion Recognition [1][2][3][4][5]

FER typically has four steps. The first is to detect a face in an image and draw a rectangle around it and the next step is to detect landmarks in this face region. The third step is extracting spatial and temporal features from the facial components. The final step is to use a Feature Extraction (FE) classifier and produce the recognition results using the extracted features. Figure 1.1 shows the FER procedure for an input image where a face region and facial landmarks are detected. Facial landmarks are visually salient points such as the end of a nose, and the ends of eyebrows and the mouth as shown in Figure 1.2. The pairwise positions of two landmark points or the local texture of a landmark are used as features. Table 1.1 gives the definitions of 64 primary and secondary landmarks . The spatial and temporal features are extracted from the face and the expression is determined based on one of the facial categories using pattern classifiers.

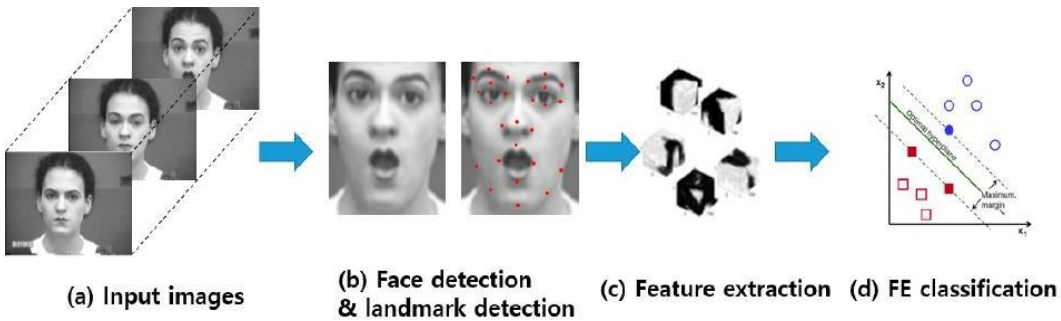


Figure 1: FER procedures for an image .

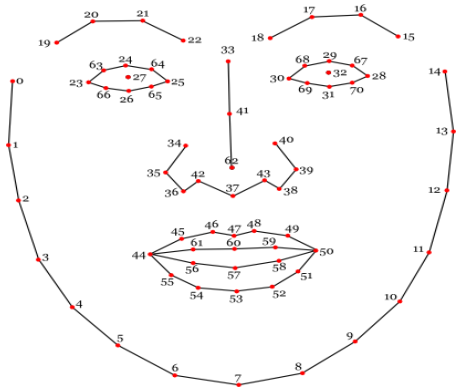


Figure 2: Facial landmarks to be extracted from a face.

Table 1.1 Definitions of 64 primary and secondary landmarks.

Primary landmarks		Secondary landmarks	
Number	Definition	Number	Definition
16	Left eyebrow outer corner	1	Left temple
19	Left eyebrow inner corner	8	Chin tip
22	Right eyebrow inner corner	2-7,9-14	Cheek contours
25	Right eyebrow outer corner	15	Right temple
28	Left eye outer corner	16-19	Left eyebrow contours
30	Left eye inner corner	22-25	Right eyebrow corners
32	Right eye inner corner	29,33	Upper eyelid centers
34	Right eye outer corner	31,35	Lower eyelid centers
41	Nose tip	36,37	Nose saddles
46	Left mouth corner	40,42	Nose peaks (nostrils)
52	Right mouth corner	38-40,42-45	Nose contours
63,64	Eye centers	47-51,53-62	Mouth contours

DL based FER approaches greatly reduce the dependence on face-physics based models and other preprocessing techniques by enabling end to end learning directly from the input images . Among DL models, Convolutional Neural Networks (CNNs) are the most popular. With a CNN, an input image is filtered through convolution layers to produce a feature map. This map is then input to fully connected layers, and the facial expression is recognized as belonging to a class based on the output of the FE classifier.

The dataset used for this model is the Facial Emotion Recognition 2013 (FER 2013) **[1]** dataset. This is an open source dataset that was created for a project then shared publicly for a Kaggle competition. It consists of 35,000 grayscale size 48 × 48 face images with various emotion labels. For this project, five emotions are used, namely happy, angry, neutral, sad and fear.

1.3 Review Literature

Facial expressions are used by humans to convey mood. Automatic facial expression analysis tools have applications in robotics, medicine, driving assist systems, and lie detection . Recent advances in FER have led to improvements in neuroscience and cognitive science . Further developments in Computer Vision (CV) , and ML have made emotion identification more accurate and accessible. Table 1.2 gives a summary of FER systems based on DL methods.

Table 1.2 A summary of FER systems based on DL

Reference	Emotions analyzed	Recognition algorithm	Database
Hybrid CNN-RNN **[2]**	Seven emotions (angry, disgust, fear, happy, sad, surprise, neutral)	1. Hybrid Recurrent Neural Network (RNN)-CNN framework for propagating information over a sequence 2. Temporal averaging is used for aggregation	EmotiW
Spatio temporal feature representation	Six emotions (angry, disgust, fear, happy, sad, surprise)	1. Spatial image characteristics of the representative expression state frames are learned using a CNN 2. Temporal characteristics of the spatial feature representation in the first part are learned using a long short term memory model	MMI CASME II
Joint fine using	Seven emotions (angry, disgust, fear, happy, sad, surprise, neutral)	Two different models 1. CNN for temporal appearance features 2. CNN for temporal geometry features from temporal facial landmark points	CK+ MMI
Candide-3	Six emotions (angry, disgust, fear, happy, sad, surprise)	1. Candide-3 model in conjunction with a learned objective function for face model fitting 2. RNN for temporal dependencies present in the image sequences during classification	CK+
Multi angle FER	Six emotions (angry, disgust, fear, happy, sad, neutral)	1. Extraction of texture patterns and the relevant key features of the facial points 2. CNN to predict labels for the facial expressions	CK+ MMI

CHAPTER 2: DATASET PREPARATION [6][7]

The FER 2013 dataset is well known and was used in the Kaggle competition. The data must be prepared for input to the CNN because there are some issues with this dataset as discussed below. The

input to the model should be an array of numbers, so images must be converted into arrays.
Some dataset challenges are given below.

i) Imbalance: Imbalance is when one class has many more images than another class. This results in the model being biased towards one class. For example, if there are 2000 images for the happy expression and 500 images for the fear expression, then the model will be biased towards the happy expression. Data augmentation is done to avoid this problem. Data augmentation increases the amount of data using techniques like cropping, padding, and horizontal flipping.

ii) Contrast variation: Some images in the dataset can be too dark and some can be too light. Since images contain visual information, higher contrast images have more information than lower contrast images. A CNN takes images as input, automatically learns image features and classifies the images into output classes. Thus, variations in image contrast affect CNN performance. This problem can be solved by changing the images to focus on the faces.

iii) Intra-class variation: Some images in the dataset are not human faces as there are drawings and animated faces. The features in real and animated faces differ and this creates confusion when the model is extracting landmark features. Model performance will be better if all images in the dataset are human faces so other images should be removed.

iv) Occlusion: Occlusion is when part of the image is covered. This can occur when a hand covers a part of the face such as the right eye or nose. A person wearing sunglasses or a mask also creates occlusion. Table 1.1 indicates that eyes and noses have primary features which are important to extract and recognize emotions.
Thus, occluded images should be removed from the dataset as the model cannot recognize emotions from these images.

The images used for training should be free from the above issues. Thus, manual filtering of the 35,000 images in the FER 2013 dataset was done and 7,074 images from five classes were selected, 966 for angry, 859 for fear, 2477 for happy, 1466 for neutral and 1326 for sad.

2.1 Python Libraries Used []

NumPy: Numerical Python (NumPy) is an open source Python library used for working with arrays and matrices. An array object in NumPy is called nd.array. CNN inputs are arrays of numbers and NumPy can be used to convert images into NumPy arrays to easily perform matrix multiplications and other CNN operations.

OpenCV: OpenCV is an open source library for CV, ML and image processing. Images and videos can be processed by OpenCV to identify objects, faces and handwriting. When it is integrated with a library such as Numpy, OpenCV can process array structures for analysis. Mathematical operations are performed on these array structures for pattern recognition.

TensorFlow: Tensorflow is an end-to-end open source platform for machine learning. It has a comprehensive , flexible, ecosystem of tools and libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML Powered applications

Keras: keras is an API designed for human beings, not machines. Keras follows best practices for reducing cognitive load; it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases .

Code Editors Used: Jupyter Notebook and VS-Code

2.2 Image To Arrays [8] [9] [10]

An image is represented by values (numbers) that correspond to the pixel intensities. The array module in NumPy (nd.array) is used to convert an image into an array and obtain the image attributes. Figure 2.2 shows an image in the sad class from the FER 2013 dataset [1] converted into a NumPy array. Figure 2.3 shows the attributes of this image which are 2304 pixels, 2 dimensions and size 48 × 48 pixels.

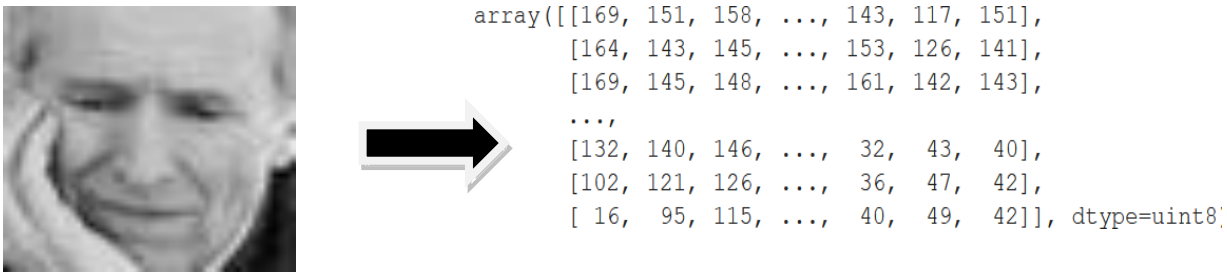
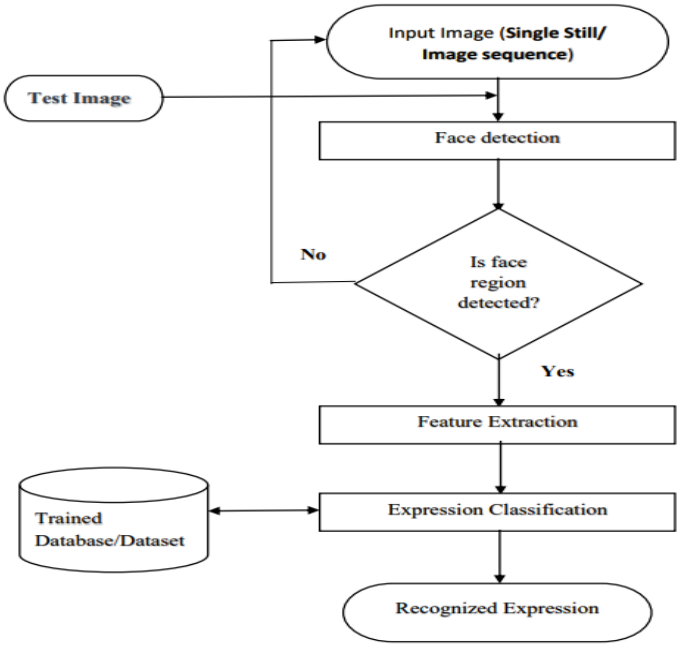


Figure 3: Sad image from the FER 2013 dataset converted into an array.

2.3 Image To Landmarks

The library is used to detect facial landmarks. This process consists of two steps, localize the face in an image and detect the facial landmarks. The frontal face detector from is used to detect the face in an image. A rectangle on the face is obtained which is defined by the top left corner and the bottom right corner coordinates. The shape predictor is used to extract the key facial features from an input image. An object called landmarks which has two arguments is passed. The first argument is an image in which faces will be detected and the second specifies the area where the facial landmarks will be obtained. This area is represented by the coordinates of the rectangle. Figure 4 shows the 64 landmarks detected in an image.

Figure 4: Landmarks detected on a face.



2.4 Flow Chart

Figure 5: Facial Emotion Recognition (FER) Process Using Flowchart.

CHAPTER 3: PREPARATION AND APPLICATION

3.1 Training Model [11]

3.2 Testing Model [12]

3.3 Pre-processing of Images

Image pre-processing are the steps taken to format images before they are used by model training and inference. This includes resizing, orienting, and color corrections etc. Here, we used 'ImageDataGenerator' class from 'Keras' deep learning library for these actions- rotation, shear, rescale, zoom, height_shift, width_shift, horizontal_flip that are applied to all the input sample images.

3.4 Building the Model for Training Basically

Training model is used to run the input data through the algorithm to correlate the processed output against the sample output. Specifically for our purpose of work we chose to build and work with a 'CNN' model. Model type is Sequential and the configurations are pretty close to the standard 'VGG16' which is a widely famous CNN model.

Within Deep Learning, a Convolutional Neural Network or CNN is a type of artificial neural network(An artificial neural network is a computational model that mimics the way nerve cells work in the human brain), which is widely used for image/object recognition and classification. Deep Learning thus recognizes objects in an image by using a CNN. Our model has 7 blocks. In creating first four blocks we have use two convolutional layer(conv2D) and a pooling(MaxPooling2D) layer with

kernel_initializer='he_normal' and activation function ‘elu’ and 0.2 dropout percentage for each. Then the 5th and 6th blocks that are basically used for feature extraction has been created with a dense (or fully connected layer) layer and kernel_initializer='he_normal' and Activation function ‘elu’ and 0.5 dropout percentage for each.

We also use Flatten() in block 5 to flatten the multi-dimensional input tensors into a single dimension. At last in the last block there is a dense layer with the activation fuction ‘softmax’, this layer is for the classification and output data. NOTE: In our case we didn’t use any special feature extraction or face detection method. Actually both things are done by the CNN model automatically. CNN model for facial expression recognition. After creating or building the model we have to compile it. We use ‘adam’ optimizer for our work but there are many other optimizers available.

3.5 Applying the Model on the Dataset

We need to fit the model and apply it on the data set in order to get the trained model. In our case, we fit the model with the batch_size of 100 and no of epochs are 50 and also use of three callbacks (earlystop, checkpoint, reduce_lr) with patience of 5 Our model runs till 43 epoch then it early stopped because after the best epoch that is 38 epoch (where the val_loss is improved from 1.06779 to 1.04583) the validation loss or val_loss doesn’t improve.

3.6 System Specification

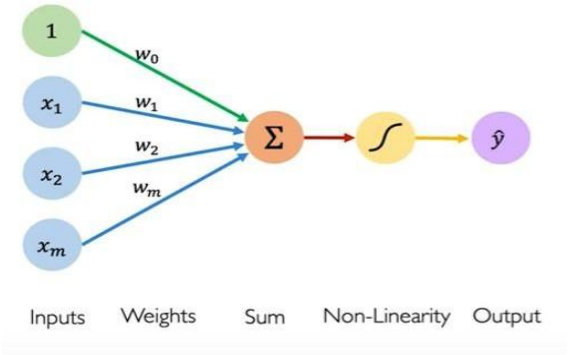
Device name	AlphaTanmoy
Processor	Intel(R) Core(TM) i5-9400F CPU @ 2.90GHz 2.90 GHz
Installed RAM	8.00 GB
Device ID	1916DA07-63F2-4E21-9133-1786240B7832
Product ID	00331-10000-00000-AA094
System type	64-bit operating system, x64-based processor
Pen and touch	No pen or touch input is available for this display
Edition	Windows 10 Pro

CHAPTER 4: CONVOLUTIONAL NEURAL NETWORKS

The fundamental building block of a NN is a neuron. Figure 3.1 shows the structure of a neuron. Forward propagation of information through a neuron happens when inputs $x1$ to xm are multiplied by their corresponding weights and then added together. This result is passed through a nonlinear activation function along with a bias term which shifts the output. The bias is shown as $w0$ in Figure 3.1. For an input vector x= $x1$, $x2$,…, xm and weight vector w= $w1$, $w2$,…, wm, the neuron output is

$$\hat{y} = g(w_0 + \sum_{i=1}^{m} x_i\, w_i)$$

The output is between 0 and 1 which makes it suitable for problems with probabilities. The purpose of the activation function is to introduce nonlinearities in the network since most real world data is nonlinear. The use of a nonlinear function also allows NNs to approximate complex functions.



.

Neurons can be combined to create a multi output NN. If every input has a connection to every neuron it is called dense or fully connected. Figure 3.2 shows a dense multi output NN with two neurons. A deep NN has multiple hidden layers stacked on top of each other and every neuron in each hidden layer is connected to a neuron in the previous layer. Figure 3.3 shows a fully connected NN with 5 layers.

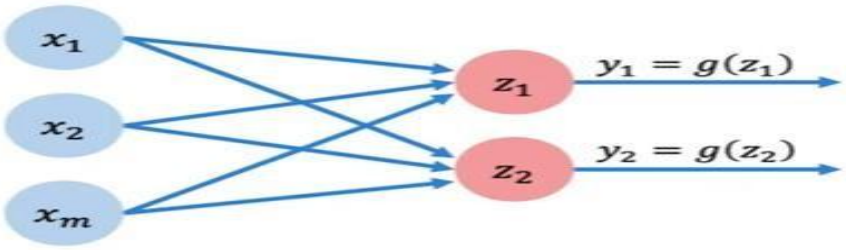


Figure 7: A multi output NN with two neurons.

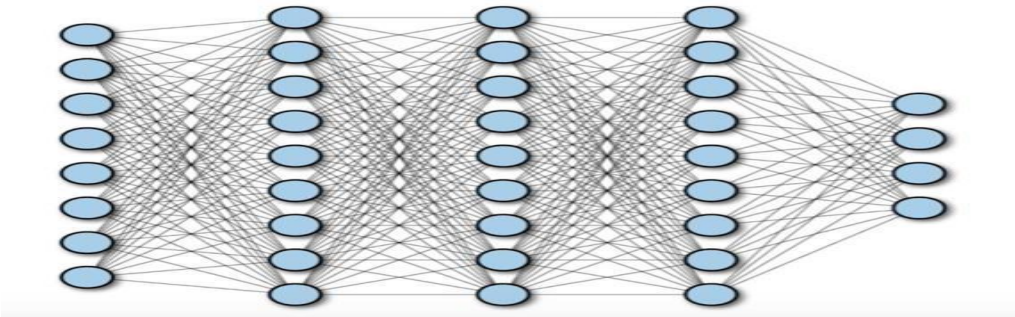


Figure 8: A fully connected NN.

4.1 The CNN Concept

A CNN is a DL algorithm which takes an input image, assigns importance (learnable weights and biases) to various aspects/objects in the image and is able to differentiate between images. The preprocessing required in a CNN is much lower than other classification algorithms. Figure 3.4 shows the CNN operations. The architecture of a CNN is analogous to that of the connectivity pattern of neurons in the

human brain and was inspired by the organization of the visual cortex . One role of a CNN is to reduce images into a form which is easier to process without losing features that are critical for good prediction. This is important when designing an architecture which is not only good at learning features but also is scalable to massive datasets. The main

CNN operations are convolution, pooling, batch normalization and dropout which are described below.

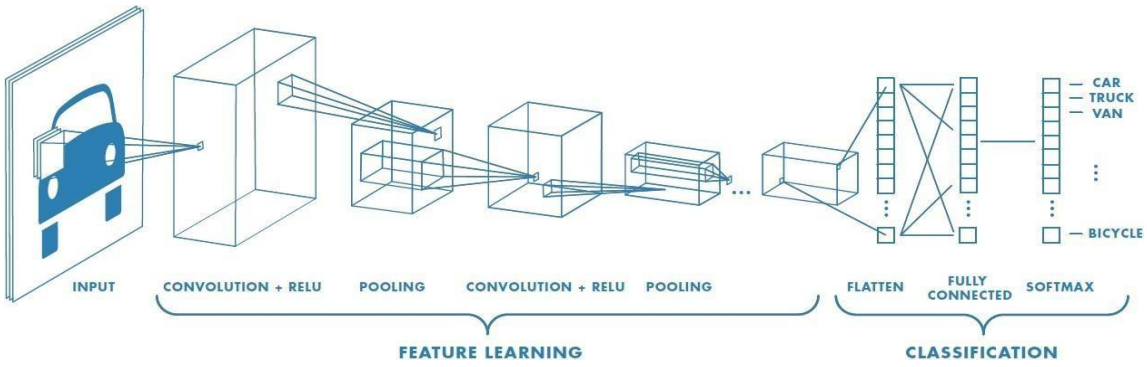


Figure 9: The CNN operations.

4.1.1 Convolution Operations

The objective of the convolution operation is to extract high level features such as edges from an input image. The convolution layer functions are as follows.

•The first convolutional layer(s) learns features such as edges, color, gradient orientation and simple textures.

•The next convolutional layer(s) learns features that are more complex textures and patterns.

•The last convolutional layer(s) learns features such as objects or parts of objects.

The element involved in carrying out the convolution operation is called the kernel. A kernel filters everything that is not important for the feature map, only focusing on specific information. The filter moves to the right with a certain stride length till it parses the complete width. Then, it goes back to the left of the image with the same stride length and repeats the process until the entire image is traversed.

Figure 3.5 presents an image with dimensions 5 × 5 (shown in green) and the following 3 × 3 kernel filter

1	**0**	**1**
0	**1**	**0**
1	**0**	**1**

The stride length is chosen as one so the kernel shifts nine times, each time performing a matrix multiplication of the kernel and the portion of the image under it.

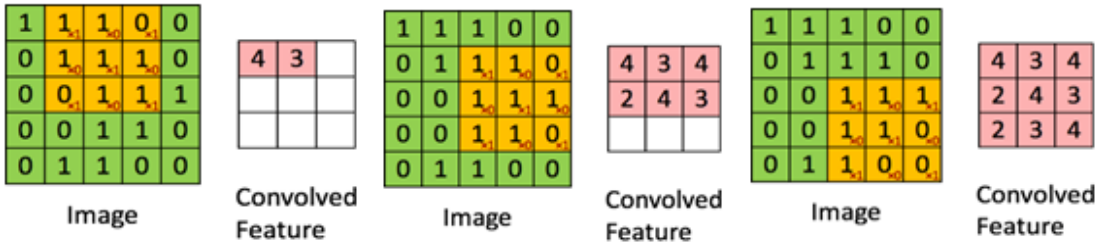


Figure 10: Convolving a 5 × 5 image with a 3 × 3 kernel to get a 3 × 3 convolved feature.

The convolved feature can have the same dimensions as the input or the kernel. This is done by same or valid padding. Same padding is when the convolved feature has the dimensions of the input image and valid padding is when this feature has the dimensions of the kernel.

4.1.2 Pooling Operation

The pooling layer reduces the spatial size of a convolved feature. This is done to decrease the computations required to process the data and extract dominant features which are rotation and position invariant. There are two types of pooling, namely max pooling and average pooling. Max pooling returns the maximum value from the portion of the image covered by the kernel, while average pooling returns the average of the corresponding values. Figure 3.6 shows the outputs obtained by performing max and average pooling on an image.

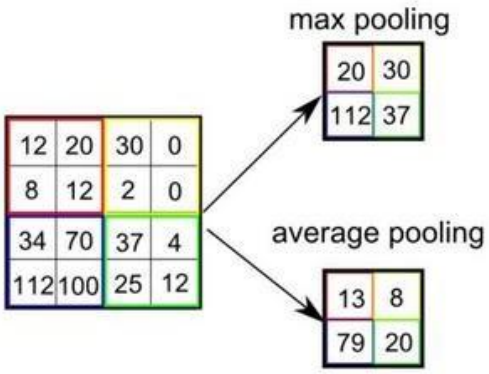


Figure 11: Max and average pooling outputs for an image.

4.1.3 Fully Connected Layer

Neurons in a fully connected layer have connections to all neurons in the previous layer. This layer is found towards the end of a CNN. In this layer, the input from the previous layer is flattened into a one-dimensional vector and an activation function is applied to obtain the output.

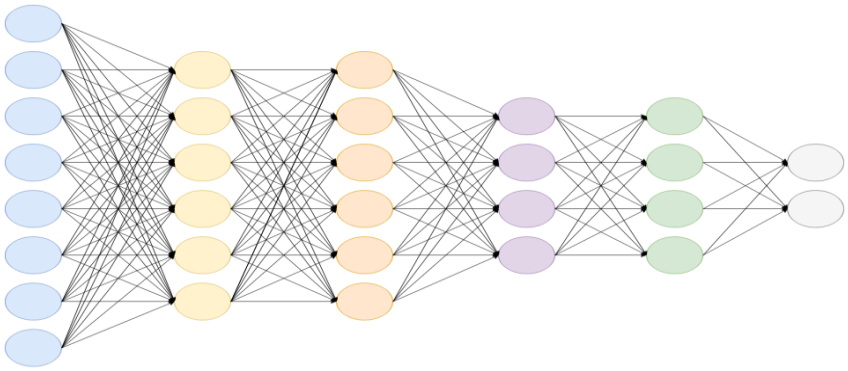


Figure 12: Fully Connected Layer.

4.1.4 Dropout

Dropout is used to avoid overfitting. Overfitting in an ML model happens when the training accuracy is much greater than the testing accuracy. Dropout refers to ignoring neurons during training so they are not considered during a particular forward or backward pass leaving a reduced network. These neurons are chosen randomly and an example is shown in Figure 3.7. The dropout rate is the probability of training a given node in a layer, where 1.0 means no dropout and 0.0 means all outputs from the layer are ignored.

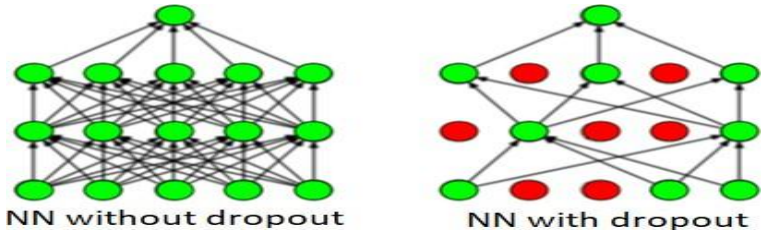


Figure 13: Dropout in a NN.

4.1.5 Batch Normalization

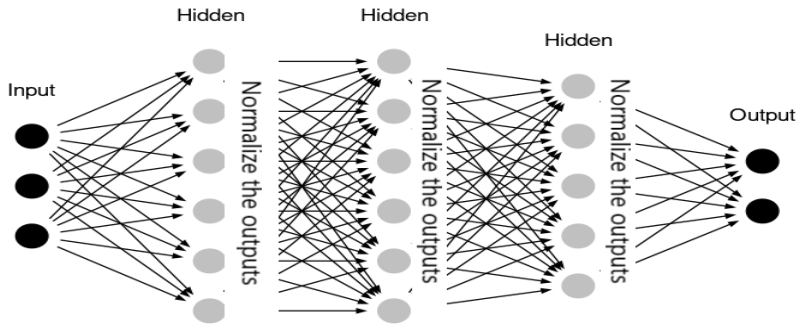


Figure 14: Batch Normalization.

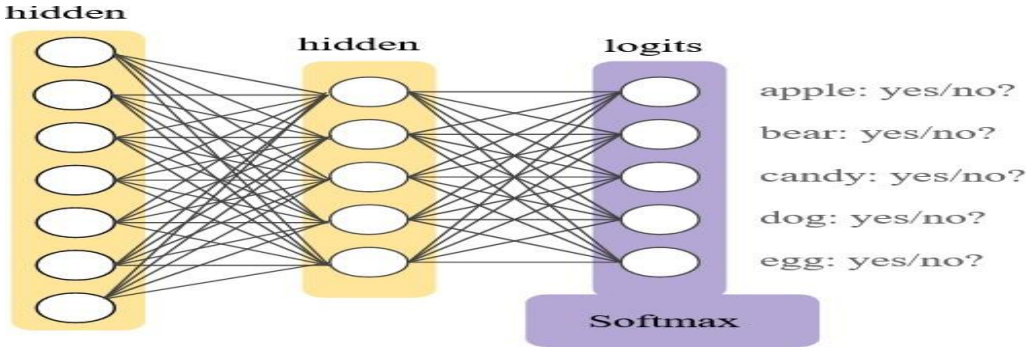
Training a network is more efficient when the distributions of the layer inputs are the same. Variations in these distributions can make a model biased. Batch normalization is used to normalize the inputs to the layers.

4.1.6 Activation Functions

Softmax and Exponential Linear Unit (ELU) are activation functions commonly used in CNNs and are described below. The softmax function is given by

$$\frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}}$$

where the z_i are the input values and K is the number of input values. This function converts real numbers into probabilities as it ensures the output values sum to 1 and are in the range 0 to 1. Softmax is used in the fully connected layer of the proposed models so the results can be interpreted as a probability distribution for the five emotions. Figure 3.8 shows the location of the softmax function.



The ELU function is

$$\begin{cases} x, if\ x > 0 \\ a(e^x - 1), if\ x \leq 0 \end{cases}$$

where x is the input value and α is the slope. This function saturates to a negative value when x is negative and α controls the saturation. This decreases the information passed to the next layer [38].

4.2 CNN ARCHITECTURE

ML models can be built and trained easily using a high level Application Programming Interface (API) like Keras. In this report, a sequential CNN model is developed using Tensorflow with the Keras API since it allows a model to be built layer by layer. Tensorflow is an end to end open source platform for ML. It has a flexible collection of tools, libraries and community resources to build and deploy ML applications. Figure 3.9 shows the structure of a CNN where conv. denotes convolution.

Figure 15: Structure of a CNN.

CHAPTER 5: RESULTS AND DISCUSSION

In this chapter, the metrics used to evaluate model performance are defined. Then the best parameter values for each model are determined from the training results. These values are used to evaluate the accuracy and loss for CNN models 1 and 2. The results for these models are then compared and discussed.

5.1 Evaluation metrics

Accuracy, loss, precision, recall and F-score are the metrics used to measure model performance. These metrics are defined below.

Accuracy: Accuracy is given by

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions}$$

Loss: Categorical cross-entropy is used as the loss function and is given by

$$Loss = -\sum_{c=1}^{m} (y_0 \log(p_{0,c}))$$

where y is a binary indicator (0 or 1), p is the predicted probability and m is the number of classes (happy, sad, neutral, fear, angry)

Confusion matrix: The confusion matrix provides values for the four combinations of true and predicted values, True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN). Precision, recall and F-score are calculated using TP, FP, TN, FN. TP is the correct prediction of an emotion, FP is the incorrect prediction of an emotion, TN is the correct prediction of an incorrect emotion and FN is the incorrect prediction of an incorrect emotion. Consider an image from the happy class. The confusion matrix for this example is shown in Figure 5.1. The red section has the TP value as the happy image is predicted to be happy. The blue section has FP values as the image is predicted to be sad, angry, neutral or fear. The yellow section has TN values as the image is not sad, angry, neutral or fear but the model predicted this. The green section has FN values as the image is not happy but was predicted to be happy.

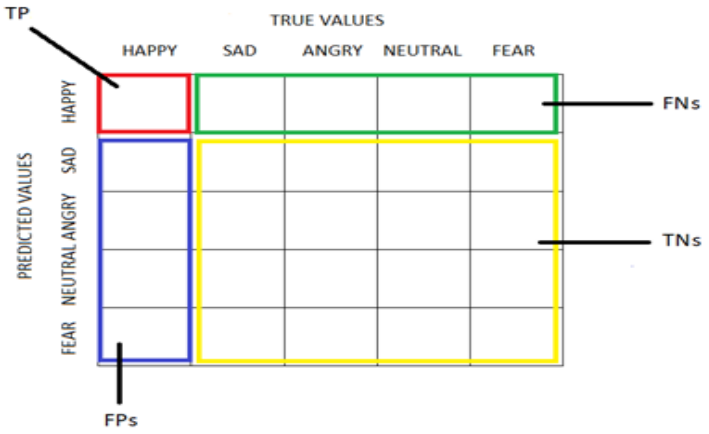


Figure 16: Confusion Matrix For Five Emotions.

Recall: Recall is given by

$$Recall = \frac{TP}{TP + FN}$$

Precision: Precision is given by

$$Precision = \frac{TP}{TP + FP}$$

F-Score: F-Score is the harmonic mean of recall and precision and is given by

$$F - Score = \frac{2 * Recall * Precision}{Recall + Precision}$$

5.2 Result Analysis

The Terminal Output

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 48, 48, 32)	320
activation (Activation)	(None, 48, 48, 32)	0
batch_normalization	(BatchNo (None, 48, 48, 32)	128
conv2d_1 (Conv2D)	(None, 48, 48, 32)	9248
activation_1 (Activation)	(None, 48, 48, 32)	0
batch_normalization_1	(Batch (None, 48, 48, 32)	128

max_pooling2d	(MaxPooling2D) (None, 24, 24, 32)	0
dropout (Dropout)	(None, 24, 24, 32)	0
conv2d_2 (Conv2D)	(None, 24, 24, 64)	18496
activation_2 (Activation)	(None, 24, 24, 64)	0
batch_normalization_2	(Batch (None, 24, 24, 64)	256
conv2d_3 (Conv2D)	(None, 24, 24, 64)	36928
activation_3 (Activation)	(None, 24, 24, 64)	0
batch_normalization_3	(Batch (None, 24, 24, 64)	256
max_pooling2d_1	(MaxPooling2 (None, 12, 12, 64)	0
dropout_1 (Dropout)	(None, 12, 12, 64)	0
conv2d_4 (Conv2D)	(None, 12, 12, 128)	73856
activation_4 (Activation)	(None, 12, 12, 128)	0
batch_normalization_4	(Batch (None, 12, 12, 128)	512
conv2d_5 (Conv2D)	(None, 12, 12, 128)	147584
activation_5 (Activation)	(None, 12, 12, 128)	0
batch_normalization_5	(Batch (None, 12, 12, 128)	512
max_pooling2d_2	(MaxPooling2 (None, 6, 6, 128)	0
dropout_2 (Dropout)	(None, 6, 6, 128)	0
conv2d_6 (Conv2D)	(None, 6, 6, 256)	295168
activation_6 (Activation)	(None, 6, 6, 256)	0
batch_normalization_6	(Batch (None, 6, 6, 256)	1024
conv2d_7 (Conv2D)	(None, 6, 6, 256)	590080
activation_7 (Activation)	(None, 6, 6, 256)	0
batch_normalization_7	(Batch (None, 6, 6, 256)	1024
max_pooling2d_3	(MaxPooling2 (None, 3, 3, 256)	0
dropout_3 (Dropout)	(None, 3, 3, 256)	0

flatten (Flatten)	(None, 2304)	0
dense (Dense)	(None, 64)	147520
activation_8 (Activation)	(None, 64)	0
batch_normalization_8	(Batch (None, 64)	256
dropout_4 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 64)	4160
activation_9 (Activation)	(None, 64)	0
batch_normalization_9	(Batch (None, 64)	256
dropout_5 (Dropout)	(None, 64)	0
dense_2 (Dense)	(None, 5)	325
activation_10 (Activation)	(None, 5)	0

Total params: 1,328,037
Trainable params: 1,325,861
Non-trainable params: 2,176

Training (Model):
Model only trained till epoch 83. Then it early stopped according to the callbacks mentioned in methodology because the validation loss (val_loss) didn't improve after epoch 78 (best epoch), rather the loss kept increasing in the following epochs.

Hence, the model early stopped the training process to avoid over fitting.
Maximum accuracy: 0.6642 (82nd epoch)
Minimum loss: 1.1945 (82nd epoch)
Validation accuracy: 0.7925 (81st epoch)
Minimum validation loss: 1.0458 (78th epoch)

Two Random Plot Graphs Are Shown Bellow:

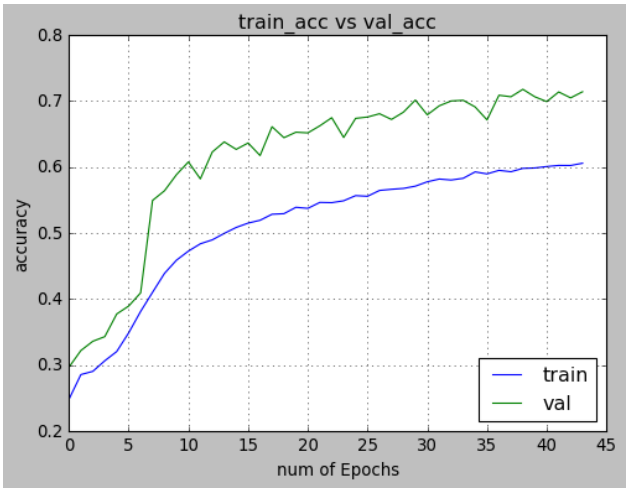


Figure 17: Random Train accuracy VS Validation Accuracy.

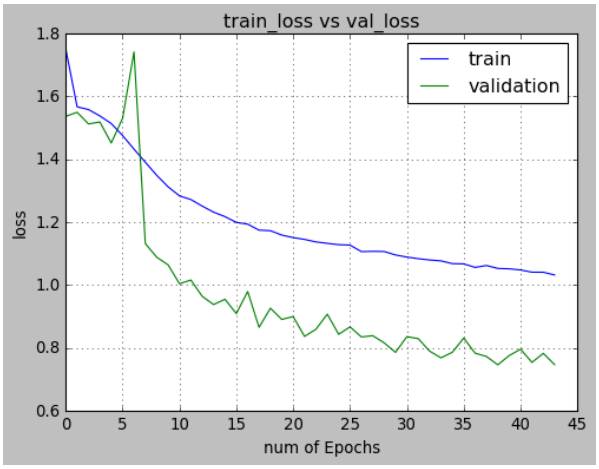


Figure 18: Random Train Loss VS Validation Loss.

➢ **Testing (Model):**

Testing samples: 7,178

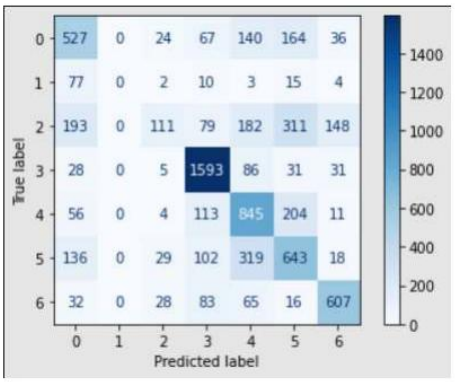


Figure 19: Random Confusion Matrix.

Output:

Single test_sample picture-

Figure 20: Output Sample Image.

Prediction Score: [[0.10943649 0.00761079 0.16303203 0.00313706 0.12231368 0.59257793 0.00189197]] 'sad

Figure 21: Multiple sample images with their predictions.

In this above set of predicted output pictures, there are some pictures that are not correctly predicted. Specially four pictures that present in [1. first column-second row, 2. fourth column-third row, 3. first column-fourth row, 4. Fifth column-forth row] are predicted wrong since our model accuracy is somewhat low.

1 st image is belongs to angry class but predicted as happy.
2 nd image is belongs to disgust class but predicted as sad.
3 rd image is belongs to surprise class but predicted as angry.
4 th image is belongs to angry class but predicted as surprise.

All other images are predicted correctly. In the case, the model predicts incorrectly, the correct label is often the second most likely emotion. (Based on the predicted scores)

5.3 Internal Layers Representation

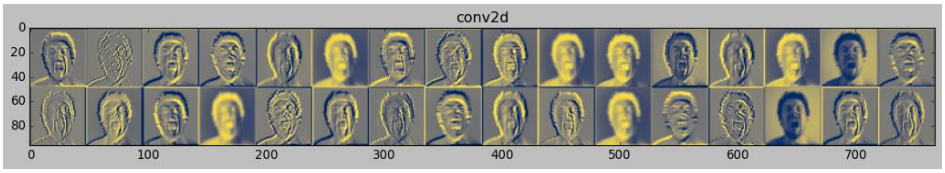


Figure 22: con2d Internal Layer.

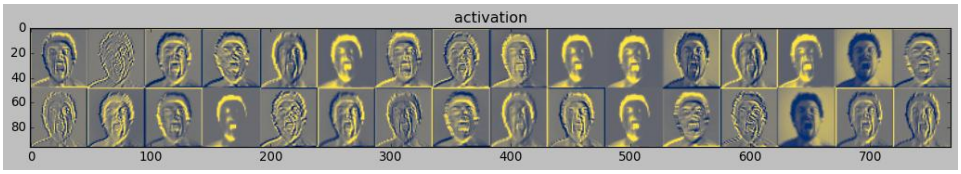


Figure 23: activation Internal Layer.

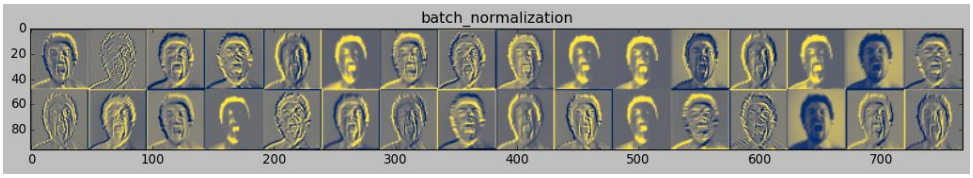


Figure 24: batch_normalization Internal Layer.

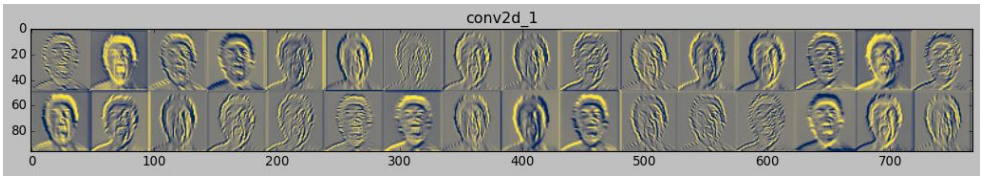


Figure 25: conv2d_1 Internal Layer.

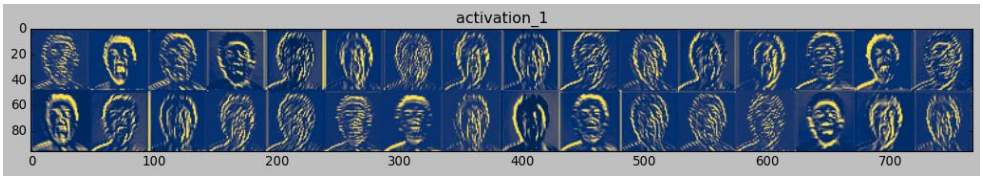


Figure 26: activation_1 Internal Layer.

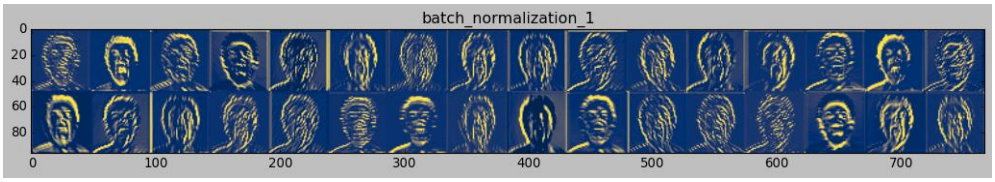


Figure 27: batch_normalization_1 Internal Layer.

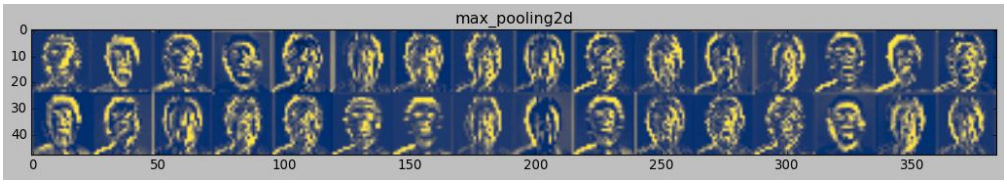


Figure 28: max_pooling2d Internal Layer.

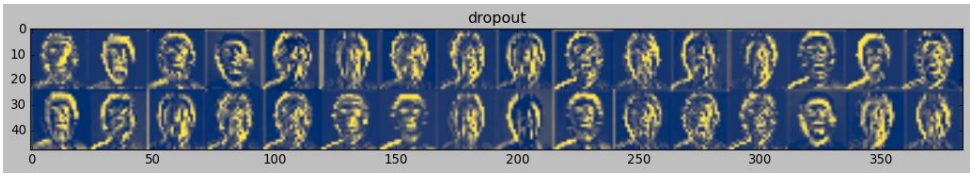


Figure 29: dropout Internal Layer.

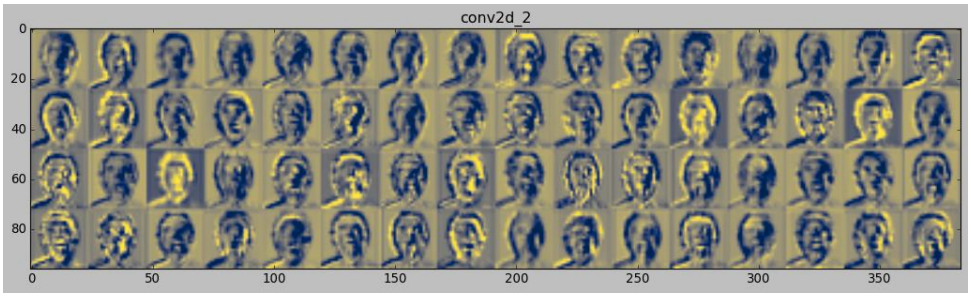


Figure 30: conv2d_2 batch_normalization Internal Layer.

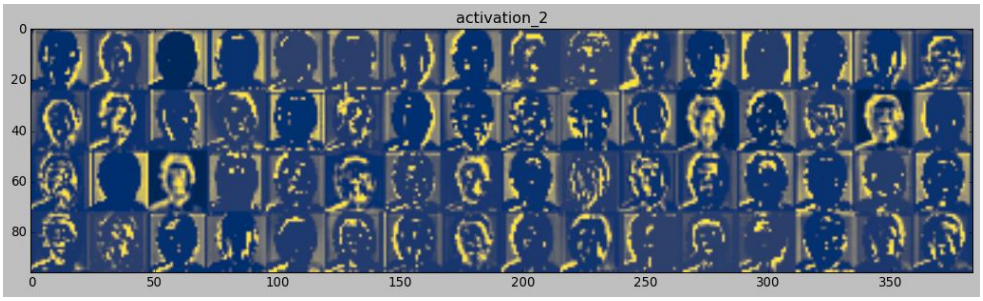


Figure 31: activation_2 Internal Layer.

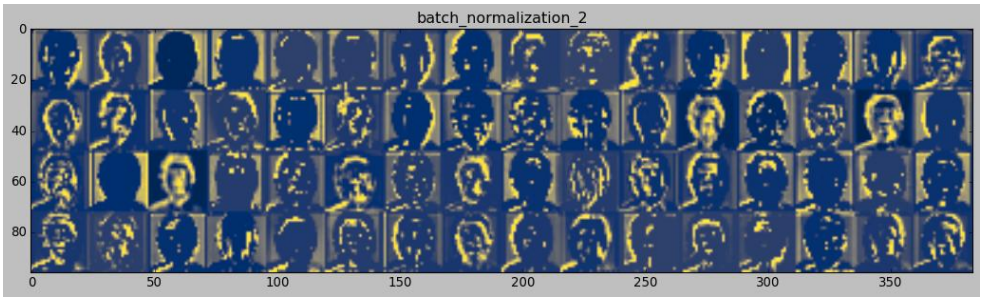


Figure 32: batch_normalization_2 Internal Layer.

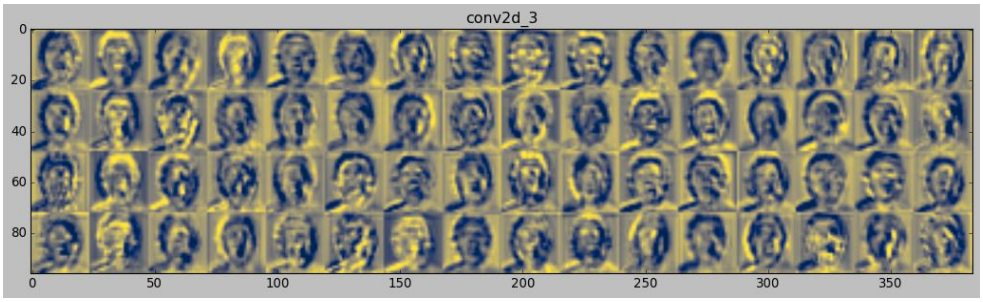


Figure 33: conv2d_3 Internal Layer.

CHAPTER 6: ERROR ANALYSIS

The accuracy of our project is somewhat low because of some problems which we face during our work process like not enough resources during the project work we didn't have enough time to

improve our code, limitation of data in our project our data (pictures) were limited, we tried to detect seven emotion but there was not enough pictures for seven emotion and the pictures were not evenly spread among the emotion so some emotion did not had enough pictures to train & the quality of data, we used FER 2013 data set but in that set the quality of the data was too low which affected our training of the model, and also inconsistency of FER 2013 dataset was a big problem, like in the dataset there were a few pictures which are corrupted some pictures misplaced during sorting. Also, this is our first time working on a project based on Artificial Neural Network and hence we lacked experience on this work. If we were able to overcome these problems we may have had better accuracy.

CHAPTER 7: CONCLUSION

The facial expression recognition system presented in this project work contributes a resilient face recognition model based on the mapping of behavioral characteristics with the physiological biometric characteristics. The physiological characteristics of the human face with relevance to various expressions such as happiness, sadness, fear, anger, surprise and disgust are associated with geometrical structures which restored as base matching template for the recognition system. The behavioral aspect of this system relates the attitude behind different expressions as property base. This project work promises a new direction of research in the field of asymmetric biometric cryptosystems which is highly desirable in order to get rid of passwords and smart cards completely.

CHAPTER 8: FUTURE SCOPE

It is important to remember that there is no specific formula to build a neural network that can guarantee to work properly fine. Different problems would require different network architectures and a lot of trail and errors to produce desirable validation accuracy. This is the reason why neural nets are often perceived as "Black box algorithms."

We need to improve in specific areas like-

- Number (weights etc.) and configuration of convolution layers.
- Number and configuration of dense layers.
- Dropout percentage in both dense and convolution layers.
- Better optimization in compiling the model.
- Using of some special feature extraction and face detection methods.

But due to lack of highly configured system we could not go deeper into dense neural network as the system gets very slow and we will try to improve in these areas in future. We would also like to train more databases into the system to make the model more and more accurate but again resources become a hindrance in the path.

Having examined techniques to cope with expression variation, in future it may be investigated in more depth about the face classification problem and optimal fusion of colour and depth information. Further study can be laid down in the direction of allele of gene matching to the geometric factors of the facial expressions.

Bibliography

[1] H.-W. Ng, V. D. Nguyen, V. Vonikakis, and S. Winkler, “Deep Learning for Emotion Recognition on Small Datasets using Transfer Learning,” Proc. the 2015 ACM on International Conference on Multimodal Interaction (ICMI '15), New York, NY,USA, pp. 443-449, 2015.

[2] G. Muhammad and M. F. Alhamid, “User Emotion Recognition from a Larger Pool of Social Network Data Using Active Learning,” Multimedia Tools and Applications, vol. 76, no. 8, pp. 10881-10892, April 2017.

[3] N. Zeng, H. Zhang, B. Song, W. Liu, Y. Li, A. M. Dobaie, “Facial expression recognition via learning deep sparse autoencoders,” Neurocomputing, Volume 273, 2018, Pages 643-649.

[4] M. S. Hossain and G. Muhammad, “An emotion recognition system for mobile applications,” IEEE Access, vol. 5, pp. 2281-2287, 2017.

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

[6] H. Ding, S. K. Zhou and R. Chellappa, "FaceNet2ExpNet: Regularizing a Deep Face Recognition Net for Expression Recognition," 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), Washington, DC, 2017, pp. 118-126.

[7] Y. Guo, D. Tao, J. Yu, H. Xiong, Y. Li and D. Tao, "Deep Neural Networks with Relativity Learning for facial expression recognition," 2016 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), Seattle, WA, 2016, pp. 1-6

[8] “Facial Signs of Emotional Experience” Paul Ekman and Wallace V. Friesen University of California, San Francisco, Sonia Ancoli University of California, San Diego, 1980 Link: https://www.paulekman.com/wp-content/uploads/2013/07/Facial-SignOf-Emotional-Experience.pdf

[9] FER Dataset 2013: Link: https://www.kaggle.com/datasets/msambare/fer2013

[10] MMA FACIAL EXPRESSION Dataset: https://www.kaggle.com/datasets/mahmoudima/mma-facial-expression

[11] Trainer Model Link: https://github.com/AlphaTanmoy/Facial-Expression-Detection/blob/main/Trainer_Model.py

[12] Testing Model Link: https://github.com/AlphaTanmoy/Facial-Expression-Detection/blob/main/Testing_Model.py

[13] Project Source Code Link: https://github.com/AlphaTanmoy/Facial-Expression-Detection