FACIAL EXPRESSION RECOGNITION

Project report submitted to Indian Institute of Information Technology, Nagpur in partial fulfilment of the requirements for the Award of Degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

by

Saksham Singh , Sneha Roopa Sri , Ritika Sharma

BT20CSE213, BT20CSE161, BT20CSE026

Under the guidance of

Dr. Milind Penurkar

Department of Computer Science and Engineering

***Indian Institute of Information Technology, Nagpur* 440006**

(India)2020-2024

Declaration

We, **Saksham Singh , Sneha Roopa Sri , Ritika Sharma BT20CSE213, BT20CSE161, BT20CSE026** hereby declare that our project work titled "**Facial Expression Recognition**" is carried out by us in the Department of Computer Science and Engineering at Indian Institute of Information Technology, Nagpur. The work is original and has not been submitted earlier whole or in part for the award of any degree/diploma at this or any other Institution / University.

Date: 7/12/2023

Certificate

This is to certify that the project titled "**Facial Expression Recognition**",submitted by **Saksham Singh(BT20CSE213), A Sneha Roopa Sri (BT20CSE161), Ritika Sharma (BT20CSE026)** in the partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering, IIIT Nagpur**. The work is comprehensive, complete and fit for final evaluation.

Date: 7/12/23

Dr. Milind Penurkar

Project Guide, Department of Computer Science and Engineering, IIIT, Nagpur

Dr. Tausif Diwan

Head Department of Computer Science and Engineering,

IIIT, Nagpur

ACKNOWLEDGEMENT

"*Acknowledgement is an art, one can write glib stanzas without meaning a word, and on the other hand one can make a simple expression of gratitude"*

It gives us a great sense of pleasure to present the report of the Project Work undertaken during B. Tech. Final Year. We owe a special debt of gratitude to our Project Mentor Dr Milind Penurkar, Department of computer science engineering, Indian Institute of Information Technology, Nagpur for his constant support and guidance throughout the course of our work. It is only his cognizant efforts that our endeavours have seen the light of the day. No amount of written expression is sufficient to show our deepest scene of gratitude to him.

We are very thankful to the project review committee for their everlasting support and guidance on the ground in which we have acquired a new field of knowledge. We also do not like to miss the opportunity to acknowledge the contribution of all faculty members of the department for their kind assistance and cooperation during the development of our project.

We also acknowledge with a deep sense of reverence, our gratitude towards our parents and members of our family, who have always supported us morally as well as economically.

And last but not least, gratitude goes to all of our friends who directly and indirectly helped us to complete this project and this project report.

Saksham Singh	**Sneha Roopa Sri**	**Ritika Sharma**
BT20CSE213	**BT20CSE161**	**BT20CSE026**

ABSTRACT

Face Expression recognition is of great importance to real-world applications such as video conference, human-machine interaction and security systems. As compared to traditional machine learning approaches, deep learning-based methods have shown better performances in terms of accuracy and speed of processing in image recognition.

This report proposes a Modified Convolutional Neural Network (CNN) architecture by adding normalisation operations to its layers. The normalisation operation which is batch normalisation provided accelerates the network. CNN architecture is employed to extract distinctive face features and a Softmax classifier is used to classify faces in the fully connected layer of CNNs .

Table of Contents

Chapter 1

INTRODUCTION

Facial expressions serve as visible indications of a person's affective state, cognitive activity, intention, and personality, playing a vital role in interpersonal communication. Recognizing these expressions, including seven basic emotions (happy, sad, surprise, fear, anger, disgust, and neutral), holds significant importance in various domains like human-machine interfaces, behavioural science, and clinical practice.

Automatic recognition of facial expressions presents challenges due to the complexity and diversity of these expressions. The use of Convolutional Neural Networks (CNNs) in machine learning provides a promising avenue for analysing facial expressions. CNNs, inspired by the animal visual cortex, mimic the receptive fields of cortical neurons and have shown effectiveness in pattern recognition tasks.

Efforts in this domain aim to represent and categorise static or dynamic facial characteristics, addressing challenges related to the recognition of emotion-specific facial deformations. Traditional methods, such as Geometric-Based Parameterization, focused on tracking facial features' motions but faced limitations in robustness, especially in cases of pose and illumination changes.

The primary objective of this project is to develop a system capable of accurately recognizing various facial emotions. Leveraging advancements in CNNs and other methodologies, the goal is to address the complexities of facial expressions and improve accuracy in identifying emotions associated with specific facial features and expressions.

1.1 Motivation for the work

Facilitating Real-Time Analysis:

Enable real-time facial expression recognition through webcam feeds.

Provide quick analysis of facial expressions for various applications.

Ease of Use and Accessibility:

Offer a simple interface for users to engage with the facial expression recognition system.

Allow easy access to the webcam-based recognition system without elaborate setups.

Automation and Recognition Accuracy:

Automate the recognition process to analyse and classify facial expressions accurately.

Minimise manual intervention for recognizing various facial emotions.

Efficiency and Performance:

Ensure the recognition process is efficient and operates with reasonable performance, even with real-time webcam input.

Application in Human-Machine Interfaces:

Apply recognition technology to improve human-computer interaction or other related domains.

1.2 Objective of the work

The objective of the facial expression recognition project is to develop a robust system capable of accurately identifying and categorising facial expressions from images or video frames. This entails employing advanced techniques like convolutional neural networks (CNNs) for automatic feature extraction and learning patterns associated with different emotional states such as happiness, sadness, anger, surprise, and more. The system aims to achieve a good accuracy and generalizability across various individuals, lighting conditions, and facial poses. Ultimately, this recognition system has versatile applications, spanning human-computer interaction, healthcare, entertainment, and other domains where understanding and responding to human emotions are crucial.

1.3 Organization of Thesis

The rest of the thesis is organised as follows:

Chapter 2

This chapter focuses on the Literature Survey of the project

Chapter 3

This chapter gives a detailed report of the work completed.

Chapter 4

This chapter provides the overall summary of the project.

Chapter 5

This chapter includes references mentioned throughout the report.1

LITERATURE REVIEW

Two different approaches are used for facial expression recognition, both of which include two different methodologies . Dividing the face into separate action units or keeping it as a whole for further processing appears to be the first and the primary distinction between the main approaches. In both of these approaches, two different methodologies, namely the 'Geometric based' and the 'Appearance-based' parameterizations, can be used. Making use of the whole frontal face image and processing it in order to end up with the classifications of 7 universal facial expression prototypes: disgust, fear,happy, surprise, sadness,neutral and anger; outlines the first approach.

Here, it is assumed that each of the above-mentioned emotions has characteristic expressions on the face and that's why recognition of them is necessary and sufficient. Instead of using the face images as a whole, dividing them into some sub-sections for further processing forms the main idea of the second approach for facial expression analysis. As the expression is more related to subtle changes of some discrete features such as eyes, eyebrows and lip corners; these fine grained changes are used for analysing automated recognition. There are two main methods that are used in both of the above-explained approaches. Geometric Based Parameterization is an old way which consists of tracking and processing the motions of some spots on image sequences, firstly presented by Suwa et al to recognize facial expressions . Cohn and Kanade, later on, tried geometrical modelling and tracking of facial features by claiming that each AU is presented with a specific set of facial muscles . The disadvantages of this method are the contours of these features and components have to be adjusted manually in this frame, the problems of robustness and difficulties come out in cases of pose and illumination changes while the tracking is applied on images, as actions & expressions tend to change both in morphological and in dynamical senses, it becomes hard to estimate general parameters for movement and displacement. Therefore, ending up with robust decisions for facial actions under these varying

conditions becomes difficult. Rather than tracking spatial points and using positioning and movement parameters that vary within time, colour (pixel) information of related regions of the face is processed in Appearance Based Parameterizations; in order to obtain the parameters that are going to form the feature vectors. For classification problems, algorithms like Machine learning, Neural networks, Support Vector Machine, Deep learning, and Naive Bayes are used. Research into automatic recognition of facial expressions addresses the problems surrounding the representation and categorization of static or dynamic characteristics of these deformations of face pigmentation . Our facial emotions are expressed through the activation of specific sets of facial muscles. These sometimes subtle, yet complex, signals in an expression often contain an abundant amount of information about our state of mind. Automatic recognition of facial expressions can be an important component of natural human-machine interfaces; it may also be used in behavioural science and in clinical practice. It has been studied for a long period of time and obtained progress in recent decades. Though much progress has been made, recognizing facial expressions with a high accuracy remains to be difficult due to the complexity and variety of facial expressions .

Individual cortical neurons respond to stimuli in a restricted region of space known as the receptive field. The receptive fields of different neurons partially overlap such that they tile the visual field. The response of an individual neuron to stimuli within its receptive field can be approximated mathematically by a convolution operation. Convolutional networks were inspired by biological processes and are variations of multilayer perceptrons designed to use minimal amounts of preprocessing.

Chapter 3

WORK DONE

In this chapter we present the work done including datasets selection, system design, model architecture and UML Diagrams. We have shown the design of our system and the various tools and frameworks adopted in the process.

Selection of Dataset :

FER-2013

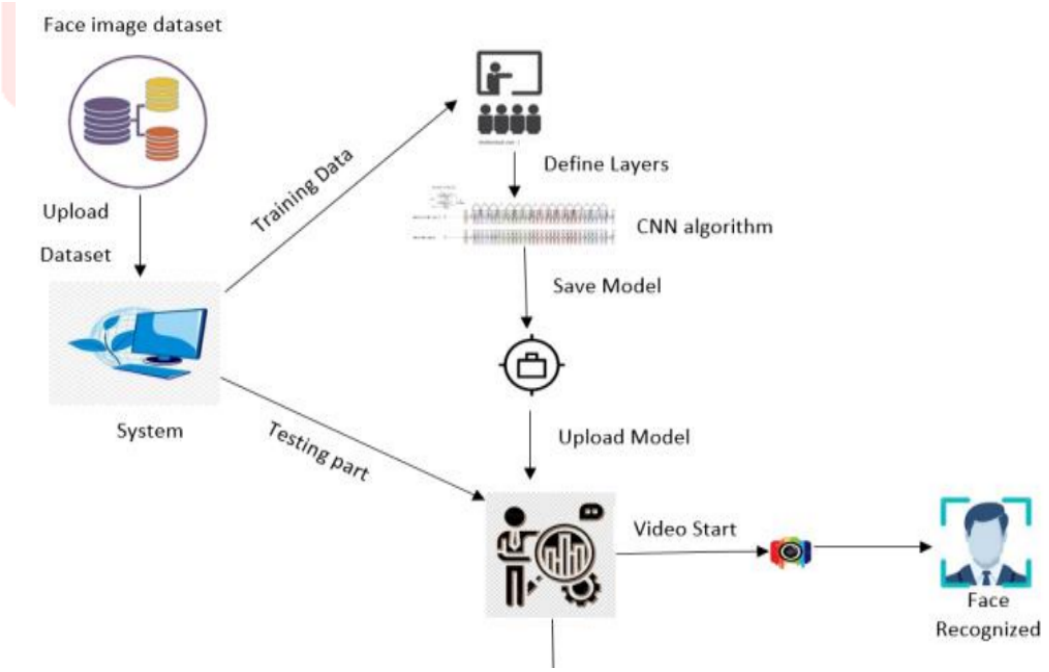
Learn facial expressions from an image

About the dataset :

The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centred and occupies about the same amount of space in each image.

The task is to categorise each face based on the emotion shown in the facial expression into one of seven categories (0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral). The training set consists of 28,709 examples and the public test set consists of 3,589 examples.

System Design :



1) Image Data Preprocessing

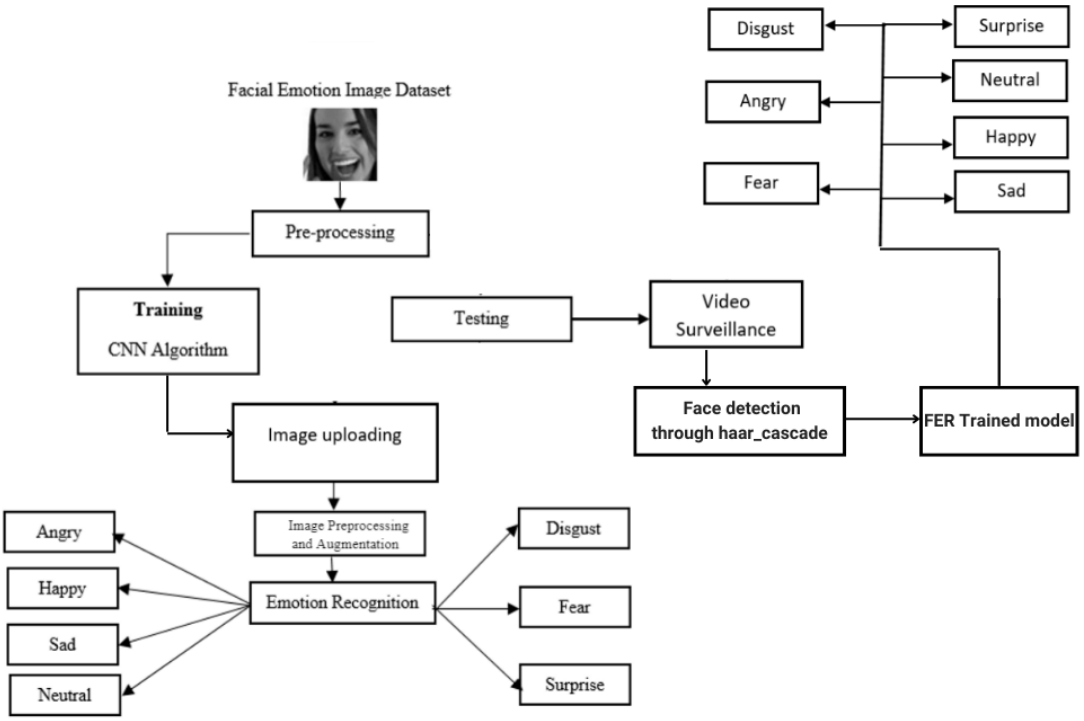
2) Image Data Augmentation

3) Feature Extraction and Training

4) Validation and Output.

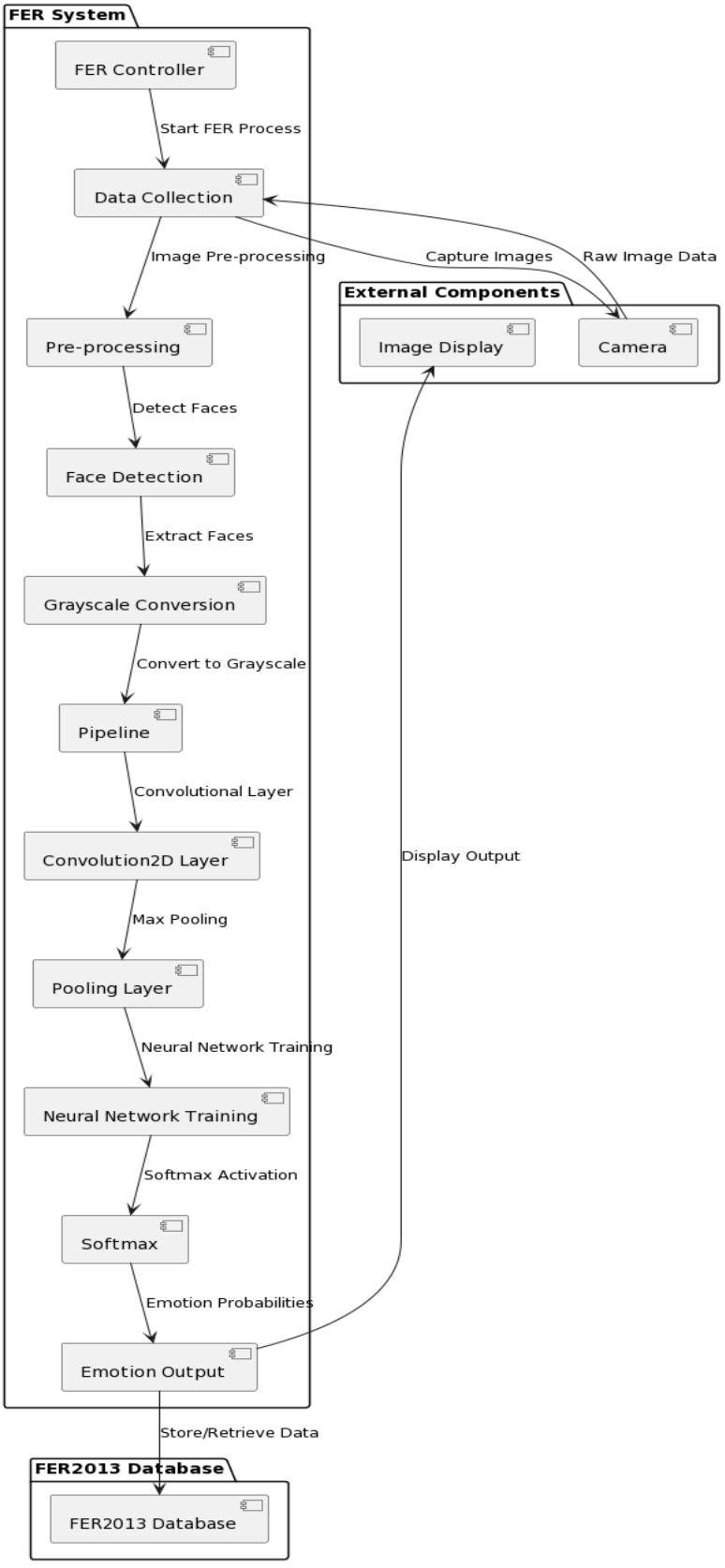
If not satisfied with accuracy do hyper parameters, changing learning rate, number of epochs, batch size, etc.

1. Block Diagram:

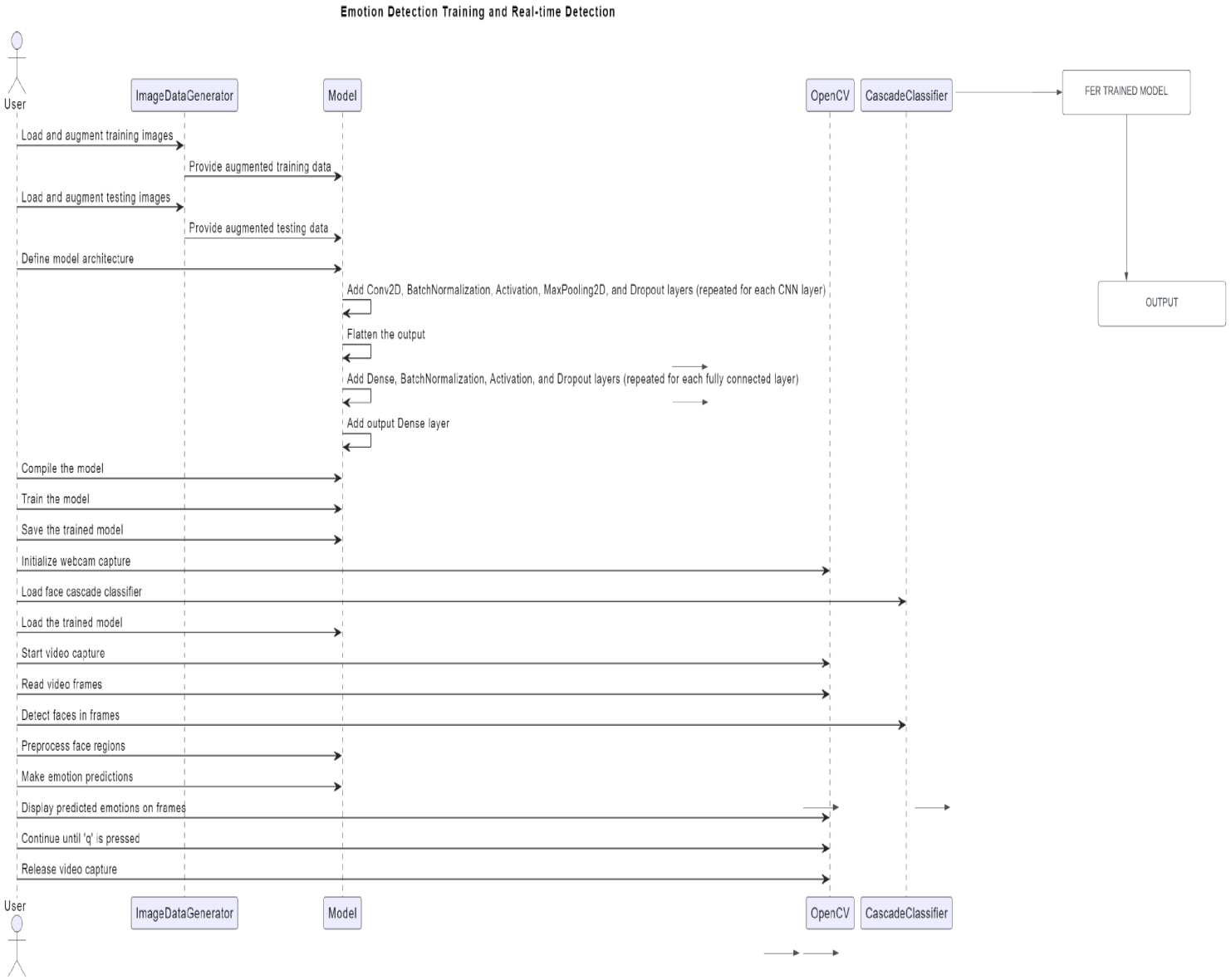


2. Use Case Diagram:

3. Data Flow Diagram



4. Sequence Diagram



Model Architecture Code:

```
#1st CNN Layer
model.add(Conv2D(64,(3,3),padding = 'same',input_shape = (48,48,1)))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout(0.25))

#2nd CNN Layer
model.add(Conv2D(128,(5,5),padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout (0.25))

#3rd CNN Layer
model.add(Conv2D(128,(3,3),padding = 'same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Dropout (0.25))

#4th CNN Layer
model.add(Conv2D(512,(3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

#5th CNN Layer
model.add(Conv2D(512,(3,3), padding='same'))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))


model.add(Flatten())

#Fully connected 1st Layer
model.add(Dense(256))
model.add(BatchNormalization())
model.add(Activation('relu'))
model.add(Dropout(0.25))


# Fully connected Layer 2nd Layer
```

Description of Layers In Architecture

First Layer

- Conv2D(64, (3,3), padding='same', input_shape=(48,48,1)): This is the first convolutional layer with 64 filters of size 3x3, using 'same' padding, and taking input images of size 48x48 with a single channel (grayscale).
- BatchNormalization(): Normalises the activations of the previous layer, which helps with faster convergence and improved generalisation.

- Activation('relu'): Applies the Rectified Linear Unit (ReLU) activation function to introduce non-linearity.
- MaxPooling2D(pool_size=(2,2)): Performs max pooling with a pool size of 2x2 to downsample the spatial dimensions.
- Dropout(0.25): Applies dropout, randomly setting a fraction of input units to 0 at each update during training, which helps prevent overfitting.

Second Layer

- Similar structure as the first convolutional layer, but now with 128 filters of size 5x5.

Third Layer

- Another convolutional layer, similar to the second, but with 128 filters of size 3x3.

Fourth Layer

- A deeper convolutional layer with 512 filters of size 3x3.

Fifth Layer

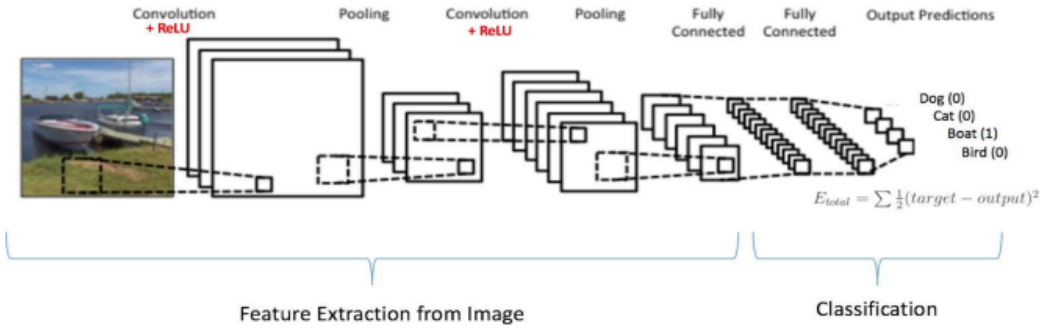
- Another convolutional layer, similar to the fourth, again with 512 filters of size 3x3.

Other Functions

- `Flatten()`: Flattens the output of the previous layer into a 1D array, preparing it for input into a fully connected (dense) layer.
- Dense(256): A fully connected layer with 256 neurons.
- BatchNormalization(): Normalises the activations.
- Activation('relu'): Applies ReLU activation.
- Dropout(0.25): Dropout for regularisation.

Methodology

The facial expression recognition system is implemented using a convolutional neural network. During training, the system received training data comprising grayscale images of faces with their respective expression labels and learns a set of weights for the network. The training step took as input an image with a face. Thereafter, an intensity normalisation is applied to the image. The normalised images are used to train the Convolutional Network. To ensure that the training performance is not affected by the order of presentation of the examples, the validation dataset is used to choose the final best set of weights out of a set of training performed with samples presented in different orders. The output of the training step is a set of weights that achieve the best result with the training data. During the test, the system received a grayscale image of a face from the test dataset, and output the predicted expression by using the final network weights learned during training. Its output is a single number that represents one of the seven basic expressions.



There are four main operations in the Convolution Neural Network shown in Figure above:

1. Convolution:

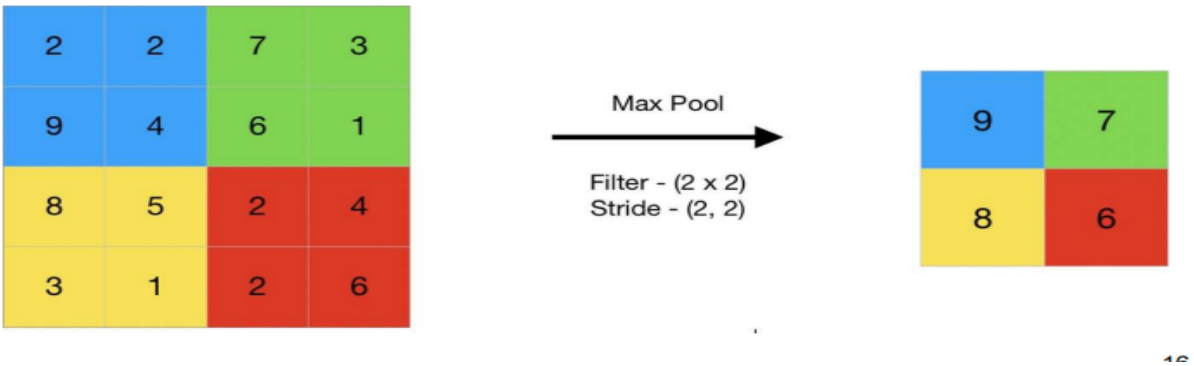
The primary purpose of Convolution in the case of a CNN is to extract features from the input image. Convolution preserves the spatial relationship between pixels by learning image features using small squares of input data. The convolution layer's parameters consist of a set of learnable filters. Every filter is small spatially (along width and height), but extends through the full depth of the input volume. **For example**, a typical filter on the first

layer of a CNN might have a size of 3x5x5 (i.e. images have depth 3 i.e. the colour channels, 5 pixels width and height). During the forward pass, each filter is convolved across the width and height of the input volume and computes dot products between the entries of the filter and the input at any position. As the filter convolves over the width and height of the input volume it produces a 2-dimensional activation map that gives the responses of that filter at every spatial position. Intuitively, the network will learn filters that activate when they see some type of visual feature such as an edge of some orientation or a blotch of some colour on the first layer, or eventually entire honeycomb or wheel-like patterns on higher layers of the network. Now, there will be an entire set of filters in each convolution layer (e.g. 20 filters), and each of them will produce a separate 2-dimensional activation map.

2. Rectified Linear Unit:

An additional operation called ReLU has been used after every Convolution operation. A Rectified Linear Unit (ReLU) is a cell of a neural network which uses the following activation function to calculate its output given x: R(x) = Max(0,x) (2.2) Using these cells is more efficient than sigmoid and still forwards more information compared to binary units. When initialising the weights uniformly, half of the weights are negative. This helps create a sparse feature representation. Another positive aspect is the relatively cheap computation. No exponential function has to be calculated. This function also prevents the vanishing gradient error, since the gradients are linear functions or zero but in no case nonlinear functions.

3. **Pooling (sub-sampling)** Spatial Pooling (also called subsampling or downsampling) reduces the dimensionality of each feature map but retains the most important information. Spatial Pooling can be of different types: Max, Average, Sum etc. In the case of Max Pooling, a spatial neighbourhood (for example, a 2×2 window) is defined and the largest element is taken from the rectified feature map within that window. In the case of average pooling, the average or sum of all elements in that window is taken. In practice, Max Pooling has been shown to work better. Max Pooling reduces the input by applying the maximum function over the input xi. Let m be the size of the filter, then the output calculated as follows:



The function of Pooling is to progressively reduce the spatial size of the input representation. In particular, pooling

- Makes the input representations (feature dimension) smaller and more manageable.
- Reduces the number of parameters and computations in the network, therefore, controlling over-fitting.
- Makes the network invariant to small transformations, distortions and translations in the input image (a small distortion in the input will not change the output of Pooling.
- Helps us arrive at an almost scale invariant representation. This is very powerful since objects can be detected in an image no matter where they are located.

4. Classification (Multilayer Perceptron):

The Fully Connected layer is a traditional Multi-Layer Perceptron that uses a softmax activation function in the output layer. The term “Fully Connected” implies that every neuron in the previous layer is connected to every neuron in the next layer. The output from the convolutional and pooling layers represents high level features of the input image. The purpose of the Fully Connected layer is to use these features for classifying the input image into various classes based on the training dataset.

Softmax is used for the activation function. It treats the outputs as scores for each class. In the Softmax, the function mapping stays unchanged and these scores are interpreted as the unnormalized log probabilities for each class.

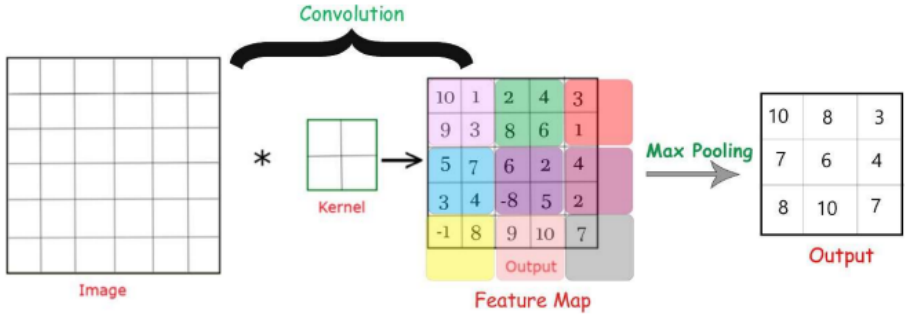
Apart from classification, adding a fully-connected layer is also a (usually) cheap way of learning nonlinear combinations of these features. Most of the features from convolutional

and pooling layers may be good for the classification task, but combinations of those features might be even better.

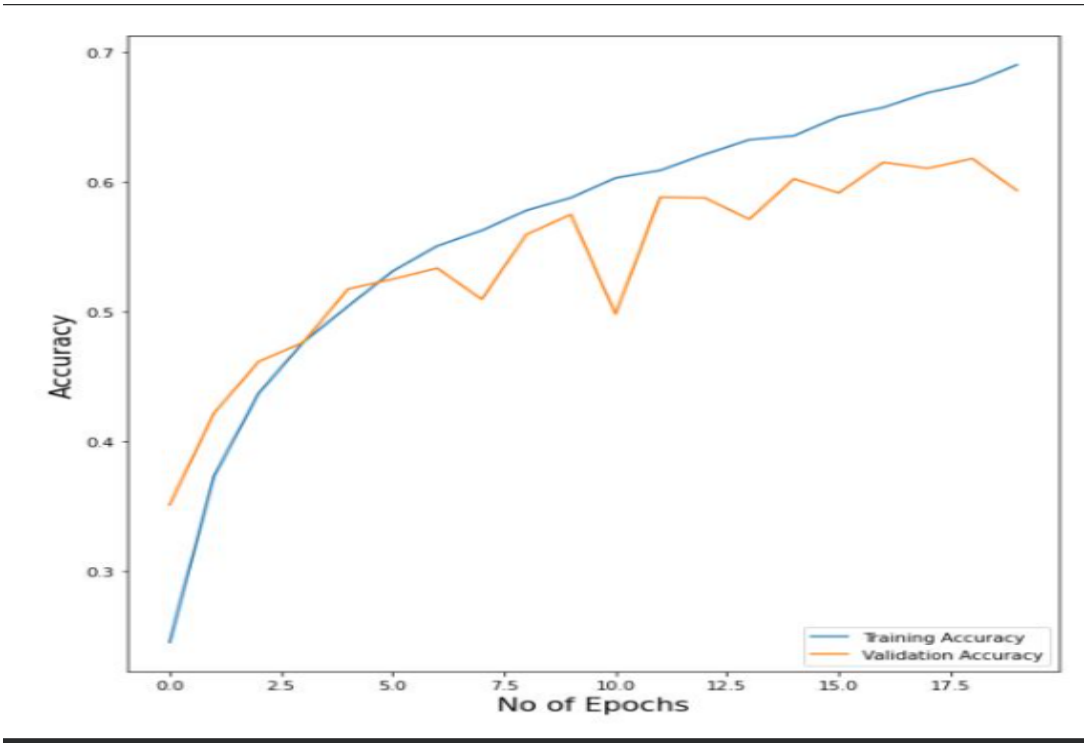
The sum of output probabilities from the Fully Connected Layer is

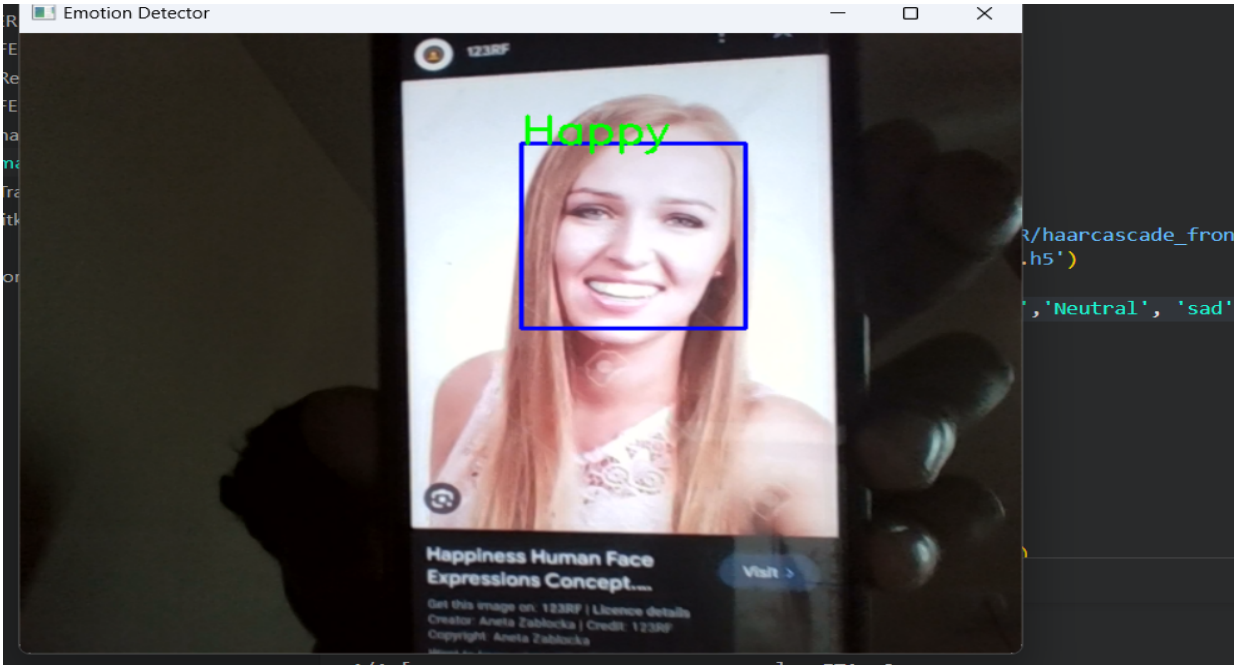
1. This is ensured by using the activation function in the output layer of the Fully Connected Layer. The Softmax function takes a vector of arbitrary real-valued scores and squashes it to a vector of values between zero and one.

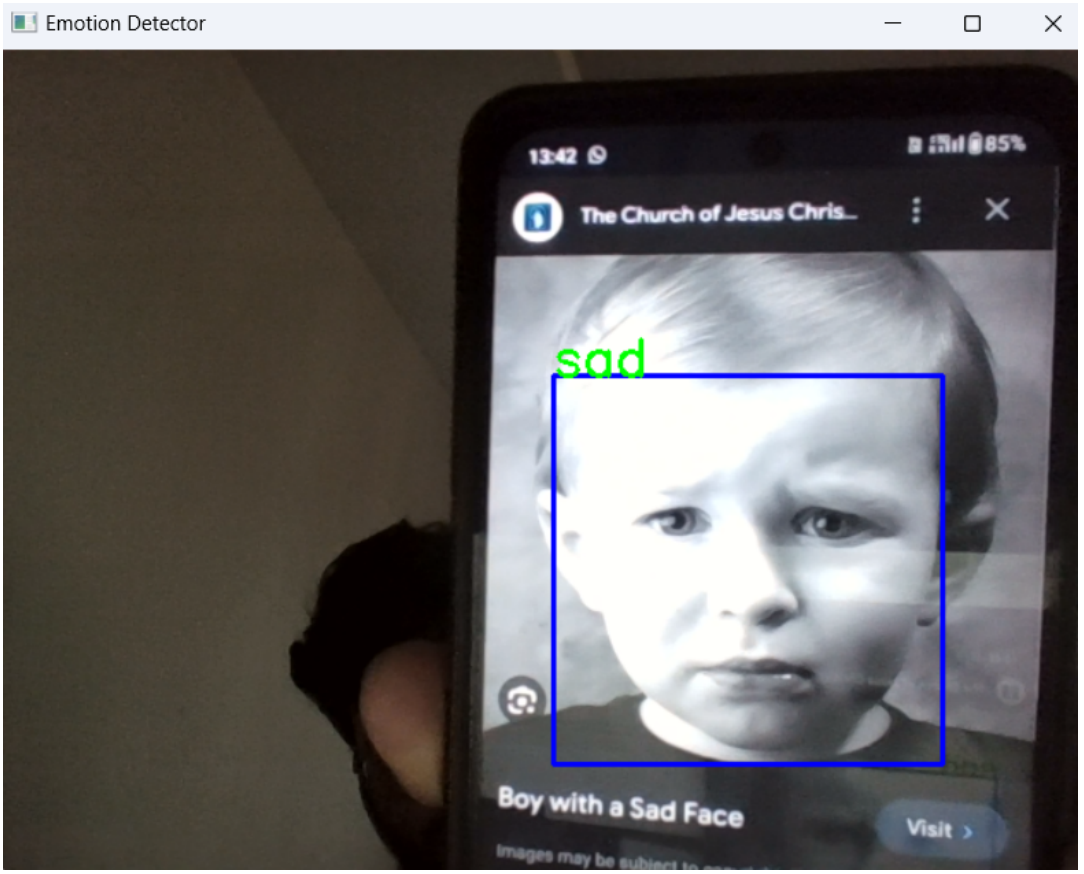
The purpose of the Fully Connected layer is to use these features for classifying the input image into various classes based on the training dataset.

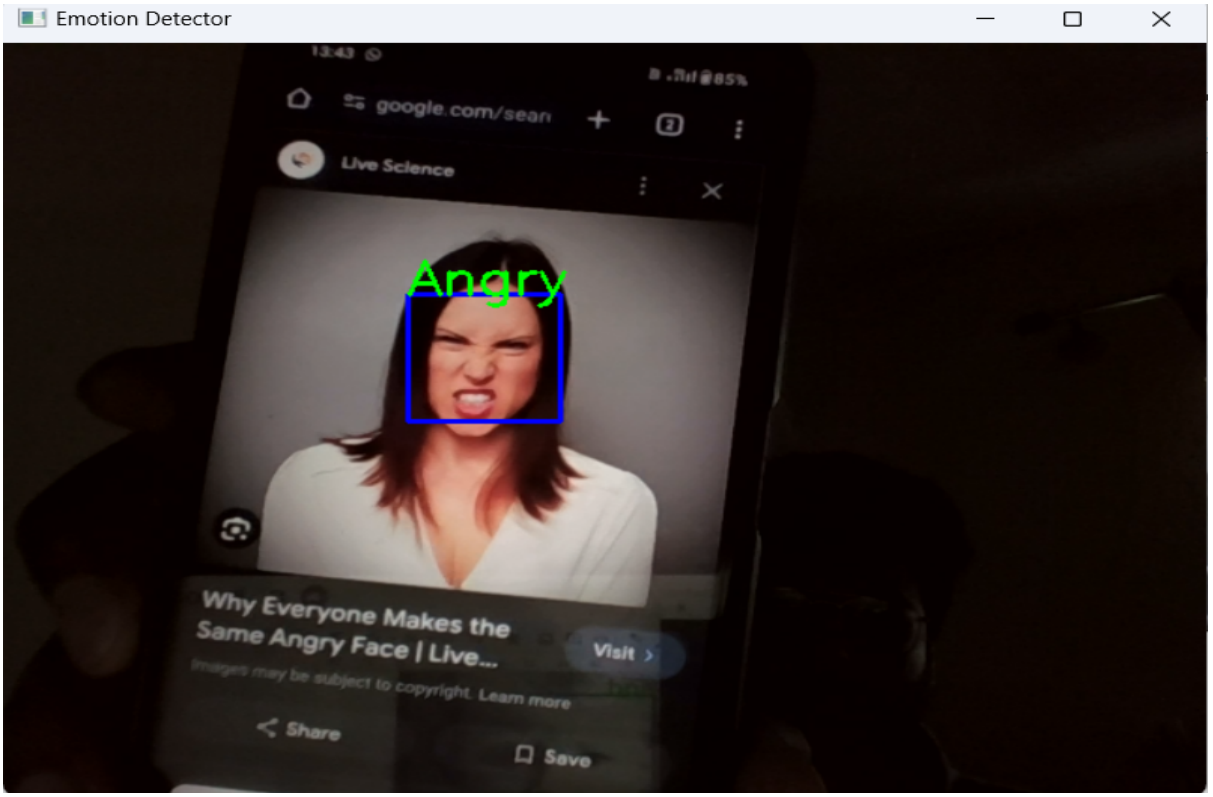


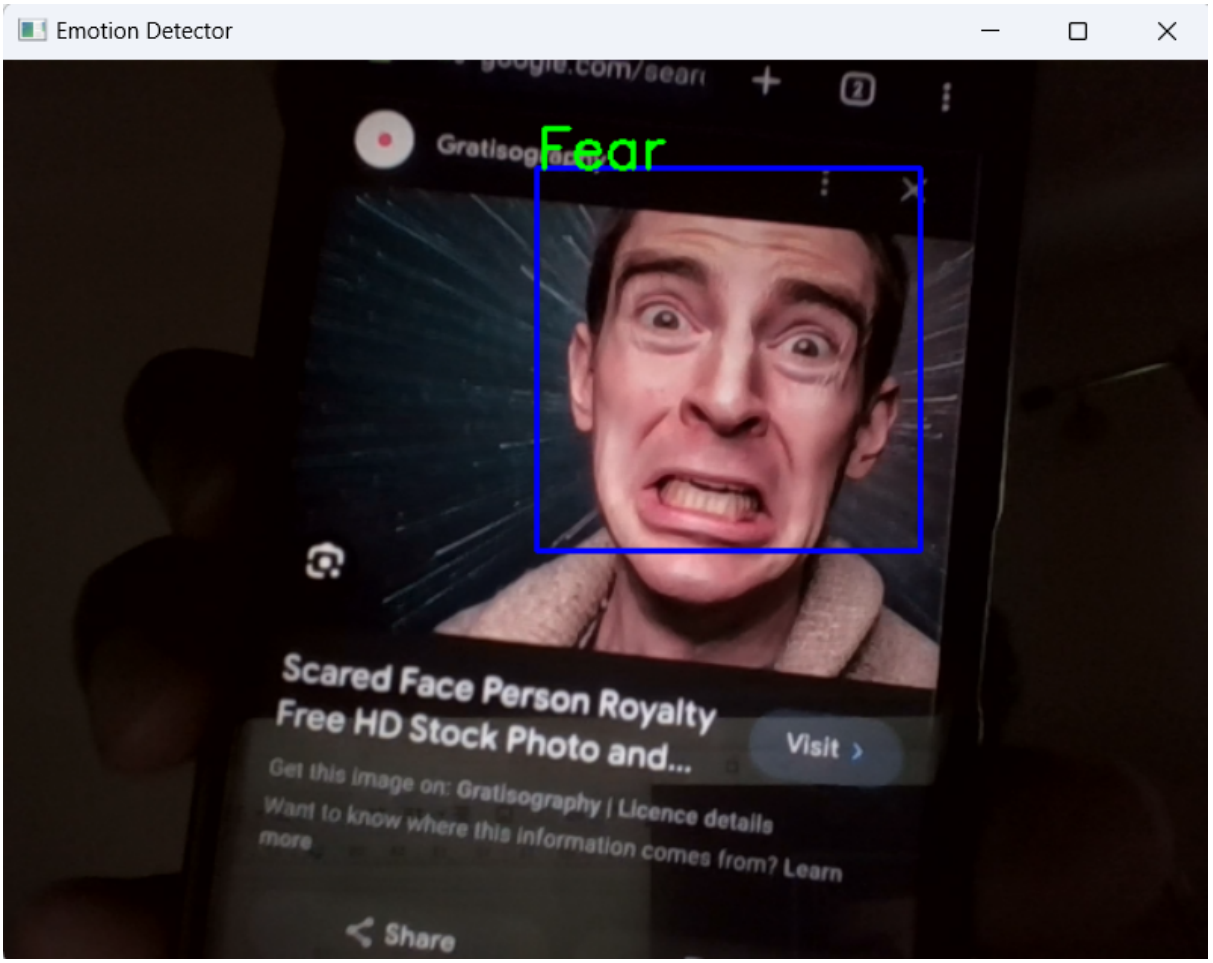
Results :

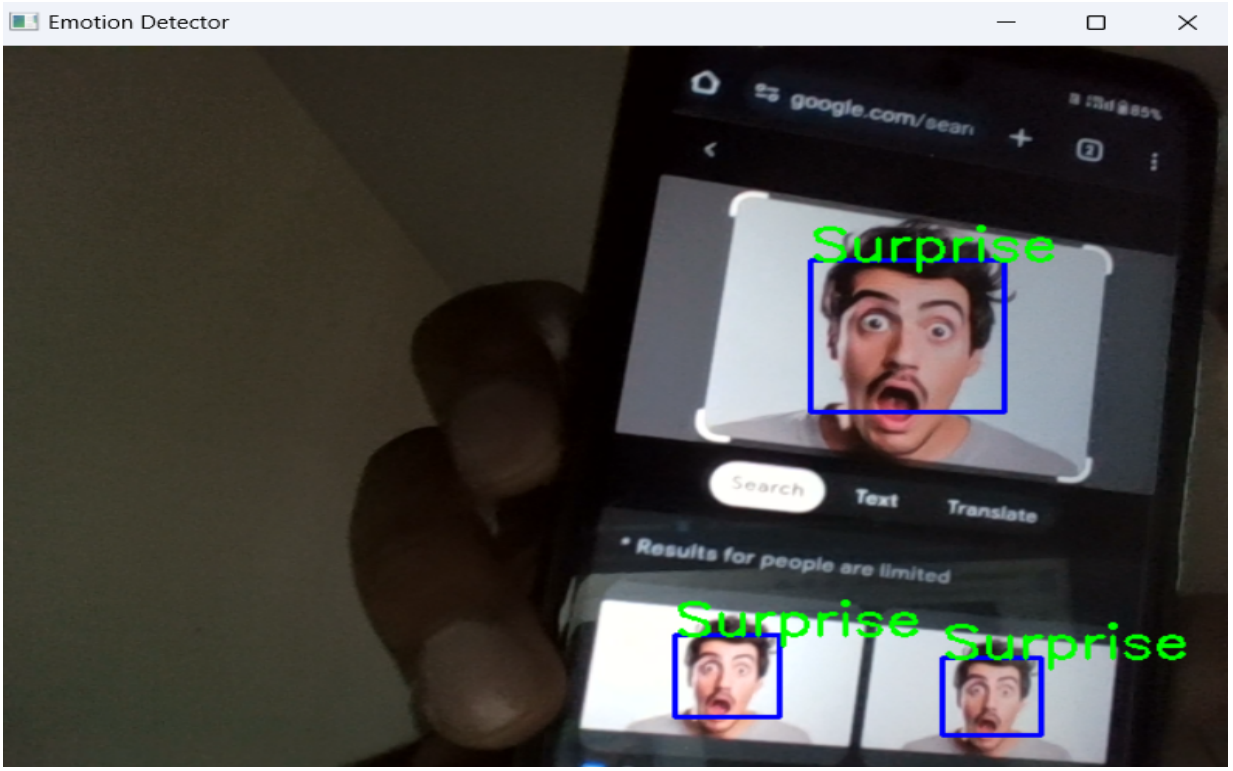

0.7
0.6
0.5
0.4
0.3
Accuracy
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
No of Epochs
Training Accuracy
Validation Accuracy

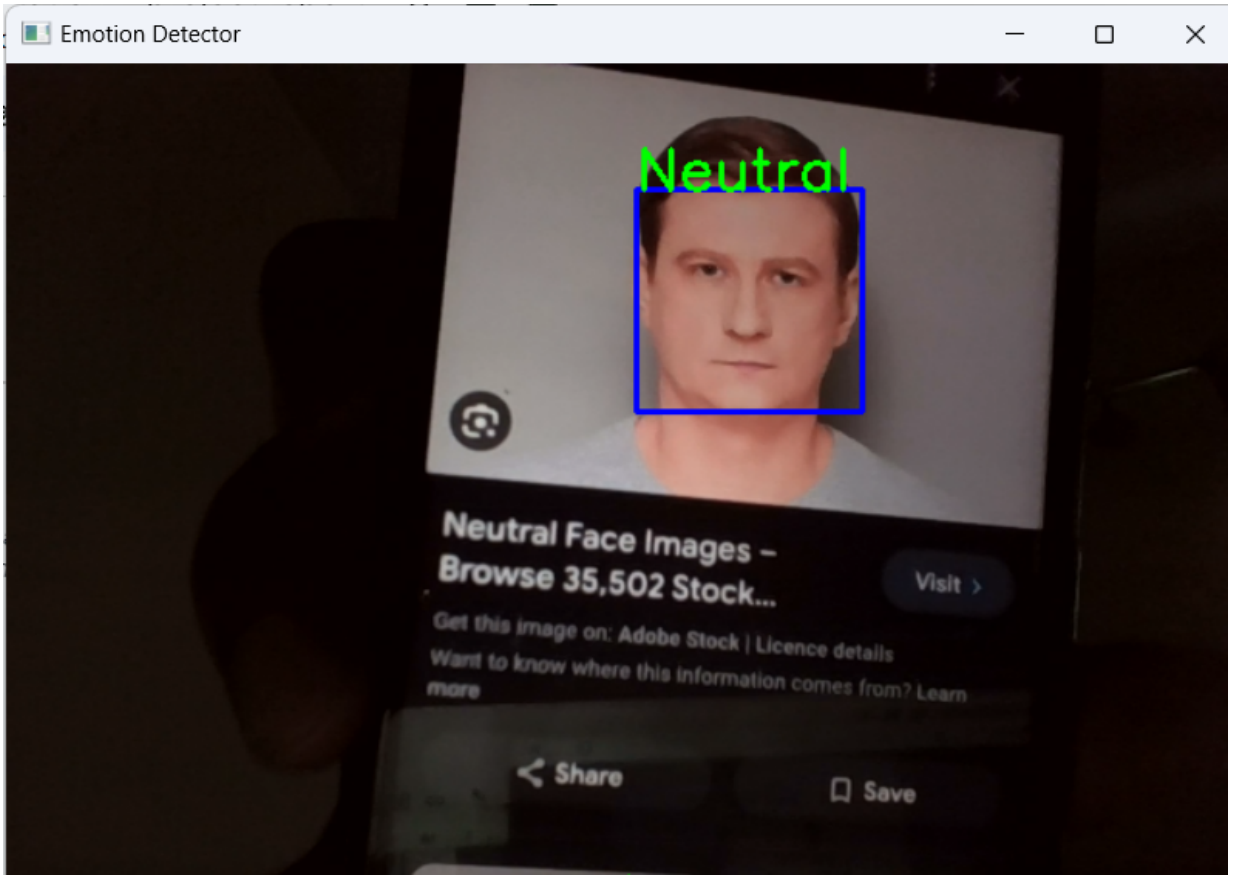

Emotion Detector
Happy
Happiness Human Face Expressions Concept...
Visit
R/haarcascade_fron
.h5')
','Neutral', 'sad'


Emotion Detector
13:42
85%
The Church of Jesus Chris...
sad
Boy with a Sad Face
Visit


Emotion Detector
13:43
google.com/sear
Live Science
Angry
Why Everyone Makes the Same Angry Face | Live...
Visit
Images may be subject to copyright. Learn more
Share
Save


Emotion Detector
Fear
Scared Face Person Royalty Free HD Stock Photo and...
Visit
Get this image on: Gratisography | Licence details
Want to know where this information comes from? Learn more
Share


Emotion Detector
Surprise
Search
Text
Translate
* Results for people are limited
Surprise
Surprise


Emotion Detector
Neutral
Neutral Face Images –
Browse 35,502 Stock...
Visit
Get this image on: Adobe Stock | Licence details
Want to know where this information comes from? Learn more
Share
Save

Chapter 4

SUMMARY

The following observations were made:

Human facial expressions i.e. happy, sad, surprise, fear, anger, disgust, and neutral. The system has been evaluated using Accuracy, Precision, Recall and F1-score. The classifier achieved an accuracy of 62.77 %, on a challenging Dataset (FER 2013)the precision of 0.57, recall of 0.57 and F1-score of 0.57.The overall precision and recall are 0.57 and 0.57 respectively. **The model performs really well on classifying positive emotions resulting in relatively high precision scores for happy and surprised**. Disgust has highest precision and recall as 0.95 and 0.99 as images in this class were oversampled to address class imbalance. Happy has a precision of 0.68 and recall of 0.69 which could be explained by having the most examples (6500) in the training set. Interestingly, surprise has a precision of 0.69 and recall of 0.65 having the least examples in the training set.**Model performance seems weaker across negative emotions on average with disgust being lowest**. In particular, the emotion sad has a low precision of only 0.44 and a recall 0.38. Model seemed confused when predicting sad and neutral faces because these two emotions are probably the least expressive (excluding crying faces).The overall F1 score is also 0.57. F1-score is highest for disgust due to oversampling of images. Happy and surprise have higher F1-score of 0.69 and 0.67 respectively.

Chapter 5

REFERENCES

[1] Shan, C., Gong, S., & McOwan, P. W. (2005, September). Robust facial expression recognition using local binary patterns. In Image Processing, 2005. ICIP 2005. IEEE International Conference on (Vol. 2, pp. II-370). IEEE.

[2] Chibelushi, C. C., & Bourel, F. (2003). Facial expression recognition: A brief tutorial overview. CVonline: On-Line Compendium of Computer Vision, 9.

[3] "Convolutional Neural Networks (LeNet) – DeepLearning 0.1 documentation". DeepLearning 0.1. LISA Lab.

[4] Matusugu, Masakazu; Katsuhiko Mori; Yusuke Mitari; Yuji Kaneda (2003). "Subject independent facial expression recognition with robust face detection using a convolutional neural network" (PDF). Neural Networks. 16 (5): 555–559. doi:10.1016/S0893-6080(03)00115-1.

[5] LeCun, Yann. "LeNet-5, convolutional neural networks".

[6] C. Zor, “Facial expression recognition,” Master’s thesis, University of Surrey, Guildford, 2008.

[7] Suwa, M.; Sugie N. and Fujimora K. A Preliminary Note on Pattern Recognition of Human Emotional Expression, Proc. International Joint Conf, Pattern Recognition, pages 408-410, 1978

[8] Recognizing action units for facial expression analysis YI Tian, T Kanade, JF Cohn IEEE Transactions on pattern analysis and machine intelligence 23 (2), 97- 115

[9] Raghuvanshi, Arushi, and Vivek Choksi. "Facial Expression Recognition with Convolutional Neural Networks." Stanford University, 2016

[10] Alizadeh, Shima, and Azar Fazel. "Convolutional Neural Networks for Facial Expression Recognition." Stanford University, 2016

[11] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. ”Imagenet classification with deep convolutional neural networks.” Advances in neural information processing systems. 2012.

[12] Simonyan, Karen, and Andrew Zisserman. ”Very deep convolutional networks for large-scale image recognition.” arXiv preprint arXiv:1409.1556 (2014)