Real-Time Emotion Detection

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering

in

Computer Engineering

by

AVN Amruta Sai 117A1006

Divya Srikant 117A1016

Anjana Neelayath 117A1053

UNDER THE GUIDANCE OF

Dr. Varsha Patil

DEPARTMENT OF COMPUTER ENGINEERING

SIES GRADUATE SCHOOL OF TECHNOLOGY

2020-2021

CERTIFICATE

This is to certify that the project entitled ***"Real-Time Emotion Detection"*** is a bonafide work of the following students, submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **Bachelor of Engineering** in **Computer Engineering**.

AVN Amruta Sai 117A1006

Divya Srikant 117A1016

Anjana Neelayath 117A1053

Dr. Varsha Patil	Dr. Aparna Bannore	Dr. Atul Kemkar
Internal Guide	Head of Department	Principal

PROJECT REPORT APPROVAL

This project report entitled ***"Real-Time Emotion Detection"*** by following students is approved for the degree of **Bachelor of Engineering** in **Computer Engineering**.

AVN Amruta Sai 117A1006

Divya Srikant 117A1016

Anjana Neelayath 117A1053

Name of External Examiner: _____

Signature with Date: _____

Name of Internal Examiner: _____

Signature with Date: _____

Date:

Place:

DECLARATION

I declare that this written submission represents my ideas in my own words, and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated, or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

AVN Amruta Sai 117A1006 _____

Divya Srikant 117A1016 _____

Anjana Neelayath 117A1053 _____

Date:

ACKNOWLEDGEMENT

We wish to express our deep sense of gratitude and thanks to our Internal Guide, Dr. Varsha Patil, for her guidance, help and useful suggestions, which helped in completing our project work in time. We are also extremely grateful to our Project coordinator Prof. Masooda Modak for her guidance provided whenever required. We also thank our HOD, Dr. Aparna Bannore, for her support in completing the project. We also thank our Principal, Dr. Atul Kemkar, for extending his support to carry out this project.

Also, we would like to thank the entire faculty of the Computer Department for their valuable ideas and timely assistance in this project. Last but not least, we would like to thank the teaching and non-teaching staff members of our college for their support in facilitating the timely completion of this project.

Project Team

AVN Amruta Sai - 117A1006

Divya Srikant - 117A1016

Anjana Neelayath - 117A1053

Abstract

We propose an implementation of a real-time CNN model for emotion classification, which includes 5 different emotions {“angry”, “happy”, “sad”, “surprised”, “neutral”} based on our use case. We validate our model by creating a video conferencing system that collectively accomplishes face detection and emotion classification tasks using our CNN architecture. We achieve an accuracy of 94% in the dataset using the combination of JAFEE, FER-2013, and our own. The datasets which are already available are biased towards western features. Thus, the models trained on such datasets fail to be used on a global basis. Thus we propose to use a self-made data set along with the ones already available. We have made the dataset using the social media (Instagram) hashtags by performing scraping. We convert the dataset collected to a suitable format and normalise the pixel values to reduce the computation cost. We flatten the images into vectors which are later used in prediction based on probability under the one-hot encoding. We infer that the careful implementation of modern CNN architectures, the quality of the dataset, and the processes followed for pre-processing images play significant roles in narrowing down the gaps between the desired and tested accuracies.

Table of Contents

List of Figures

List of Tables

CHAPTER 1
Introduction

Humans are advanced beings and are a class of species known for their ability and range of communication. What sets us apart from the rest of our fellow verbally communicating mammals is our inherent ability to perceive and understand one another without explicit citation. Body language, voice tones, hand gestures and above all, facial expressions play a key role in the same.

Emotion can be defined as a solid or intense feeling brought upon by neurophysiological changes that arise when associated with thoughts, feelings, behavioural responses, and varying degrees of pleasure and displeasure. Emotions are the primary carriers of communication among humans. Different situations give rise to different emotions that get activated, which entirely depends on the environment that stimulates certain behavioural responses. For a machine to understand the complex spectrum of emotions that a human gives out, it needs to get trained on the same range of emotions to empathise with people in various scenarios. Facial expression recognition has been under research for years in Human-Computer Interaction to gain insights into people's varying emotions. It can hence provide distinctive services to people depending on their moods. It is easy for people to detect what another person is going through by their facial expressions. It is equally challenging for a machine to identify the same. Today, when online interviews and online classes are gradually becoming the world's norm, it is paramount that any modes of communication not be obscure. Hence, the ability to perceive a situation and hence decide how to respond largely relies on body language and non-verbal communication. Over time, we have evolved, heavily relying on these non-verbal chunks of information to socialise.

In the modern-day, as humans rely more and more on technology, the number of factors we rely on to understand each other has drastically reduced. Problems like being unable to connect emotionally and not communicating through the virtual medium are increasing. Hence 'Emotion Detection' is an essential and upcoming field that aims to bridge the gap brought by the loss of non-verbal information being transferred across.

1.1 Need of Project

Actions, expressions, speech and behaviours are considered as channels which are used by humans to convey emotions. With the growing reliance on technology, these channels must not be blocked. Emotion recognition has been widely researched for this purpose. The success of meetings decisively depends on a smooth subject to subject interaction. This usually is achieved in a real environment, where one can talk, meet, interact or express the desired subject. Most of the time, the interactions by facial expressions prove to be more convincing. However, considering the virtual or online meetings, one is restricted to the interactions based only on words. Thus, there is a need to extract information just from the face of subjects live in the conference or to be able to connect to words and the emotional state of a being. We see the impact of this in various industries; doctors cannot treat patients properly, teachers and students find it difficult to connect in virtual classrooms, retail and banking sectors have seen a decline in growth. An accessible platform that can help people interact with each other would indeed bridge the gap we are noticing today.

Hence, we propose a platform that can help users communicate effectively without compromising on the information a person subconsciously gives through their expressions using emotion recognition.

1.2 Scope

This system has a wide range of applications that cater to various fields. This software can be used as an umbrella software that can be customised to fit various use case scenarios like :

1. In education, where there is an increasing disconnect between professors and students while teaching on an online platform, our project can detect the engagement levels and gauge the student's affective state.
2. In law enforcement, Emotion Detection can be used along with polygraphs for lie detection among the accused by recognising repeated tics and different expressions on the person's face.

3. It can also be used in psychology for behavioural research and identifying cues of mental health.

4. For human-robot interaction, where robots would be able to start understanding the emotions of humans too.

1.3 Gantt Chart for Project Schedule

The project was implemented in two phases. The first phase primarily consisted of researching the topic and how the existing systems helped cement the foundation of our current system. The second phase consisted of implementing the findings of Phase 1 and executing our approach.

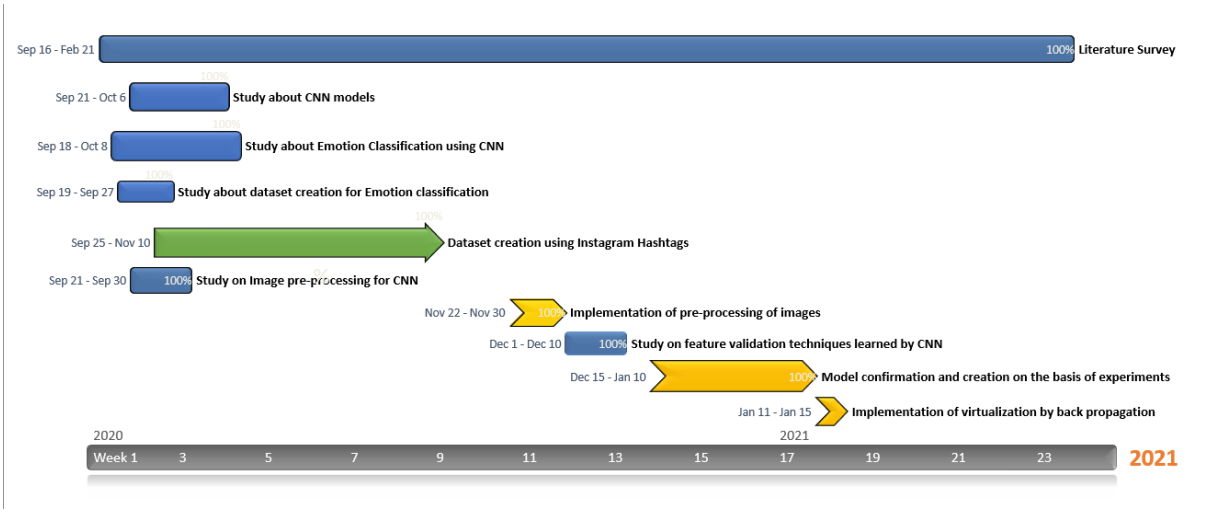


Figure 1.1 Gantt Chart for Phase 1

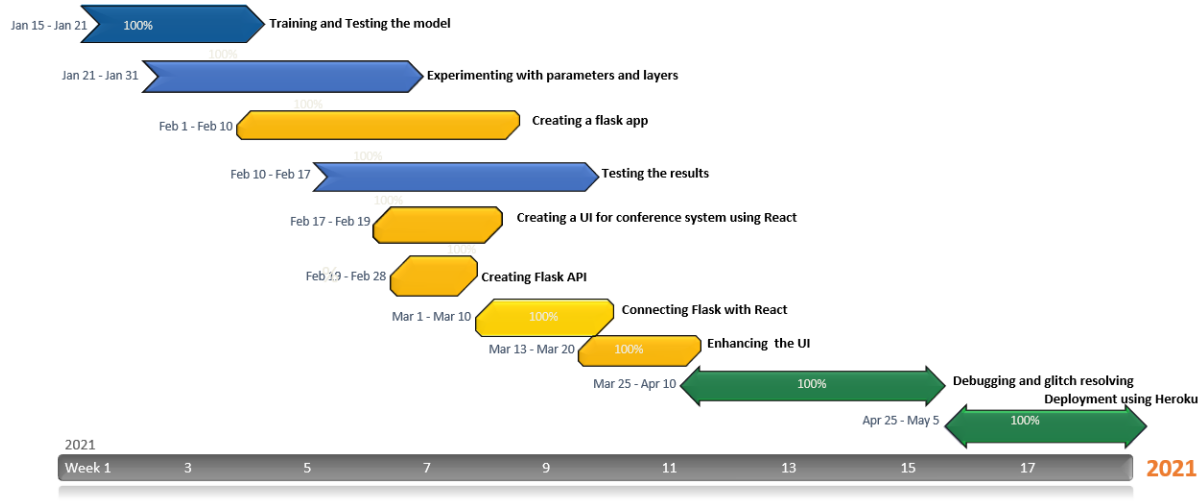


Figure 1.2 Gantt Chart for Phase 2

1.4 Organisation of the Report

This report is categorized into various topics and sub-topics. Each chapter is a chronological order of how our project on “Real-Time Emotion Detection” was implemented.

CHAPTERS

1. Introduction

   - This chapter introduces the topic and highlights our project by covering the need of the project, scope, and main objectives that drove us to implement this system.
   - The introduction elaborates on the abstract and briefly summarises the entire project.

2. Literature Survey

   - This chapter talks about existing systems and related works that were performed in the same field.
   - It discusses the various methods and algorithms employed by the previous literature reviews, the results they managed to obtain, and those systems’ limitations.
   - It also throws light on the problem statement and objectives of our project.

3. Proposed system

    - This chapter describes our proposed system and talks about how we plan to overcome the drawbacks of current systems.
    - It briefly discusses our approach to implementing the project, the emotion classification model, and the user interface that we aim to create.

4. Design and Methodology

    - In this chapter, a detailed description of the methodology that we adopted for this project has been mentioned.
    - It starts with the design details of the system and further explains the proposed procedure and techniques followed in executing the system.

5. Results and Discussions

    - This is the penultimate chapter of the report and gives a thorough evaluation of how the project was implemented.
    - It includes the various test cases that were validated by the proposed methodology that was followed.
    - It also shows the final results that we have obtained on the implementation of the system.

6. Conclusion

    - This is the final chapter of this report which concludes with a brief analysis of our topic.
    - It also discusses the future scope of our project and improvements, if any.

CHAPTER 2
Literature Survey

2.1 Survey Existing System

There are many models out there that have been built specifically for emotion classification.

1. **Facial Expression Recognition Using SVM Classifier[3]:**

   Methodology:

   In this model, the algorithm initially detects the eye and mouth. Features of eye and mouth are extracted using Gabor filter, LBP (Local Binary Pattern) and PCA (Principal Component Analysis) to reduce the dimensions of the features. Finally, SVM is used for the classification of expression and facial action units.

   Advantages:

   The accuracy rate achieved is quite decent at about 72%.

   Shortcomings:

   The system only promises to detect fewer emotions. The emotions detected are happy, sad and neutral.

2. **Hybrid-Deep Learning Model for Emotion Recognition Using Facial Expressions[6]:**

   Methodology:

   The model used in this research consists of two Convolutional Neural Networks. The first CNN was used to analyse a primary emotion of the face and classifies it as happy or sad and the second CNN detects the secondary emotion of the face. The model which was proposed in this research was trained on JAFFE (Japanese Female Facial Expression) and FER2013 datasets which gave much better results than the existing state of the art approaches of accurately classifying emotions from facial expressions.

   Advantages:

   The training accuracy of the model is 97.07% with a loss of 0.094 when tested with FER2013 dataset and gave a 94.12% training accuracy with JAFFE.

   Shortcomings:

   As the model has been trained on Japanese faces, the high accuracy rate it promises is limited to the facial features of east Asian women—this narrows down the model's scope.

3. **Emotion Recognition Using a Cauchy Naive Bayes Classifier[4]:**

   Methodology:

   The common assumption here is that the model distribution is Gaussian. However, the model successfully uses the Cauchy distribution assumption and provides an algorithm to test whether the Cauchy assumption is better than the Gaussian assumption. The person-dependent and person-independent experiments have been performed, and the conclusion shows that the Cauchy distribution assumption provides better results than the Gaussian distribution assumption. The dataset is divided into two non-overlapping sets: the training set and the test set. The parameters are estimated using only the training set. The classification is performed using only the test set.

   Advantages:

   The accuracy rates here are significantly high.

   The person-dependent and person-independent rates are calculated.

   The Gauss and Cauchy approaches are compared.

   Shortcomings:

   The model does not accurately recognise people of all ethnicities.

   Thus, the model can be stated biased towards the western features

   .

4. **Facial Emotion Recognition using Deep CNN Based Features[10]:**

   Methodology:

   The Convoluted Neural Network model VGG16 is used for the feature extraction, and further, the image is classified into six different emotions using the SVM classifier. The pre-processing of the dataset used consists of cropping the image using the viola jones face detection algorithm. The model is compared with the breakthrough model in CNN architecture ResNet50 and performs well comparatively. The comparative study of the model's accuracy before and after face detection is seen. The comparison with other similar models shows the better performance of the proposed model.

   Advantages:

   A detailed comparative study can be seen.

   The in-depth features are used to extract face features rather than using the handcrafted features.

A decent accuracy of 86% is achieved.

Shortcomings:

The dataset used is biased towards western facial characteristics thus has reduced scope on a global basis.

5. **Real Time Convolution Neural Networks for Emotion and Gender Classification[13]:**

   Methodology:

   Provides a CNN framework for real-time emotion classification based on various experiments performed to test accuracy. It provides two frameworks for the Xception model by reducing the number of parameters used and altering the presence and absence of fully connected layers, alleviating the model from slow performances in hardware constrained systems. It also makes the model more generalised. Implementation of real-time virtualisation of guided-gradient backpropagation is seen in order to validate the features of CNN.

   Advantages:

   Virtualisation of previously hidden features.

   Reduction in the number of parameters considered.

   The use of an ADAM optimiser enhances the performance.

   Shortcomings:

   Again, the dataset used cannot be considered diverse, which narrows down the model's scope.

   The accuracy achieved is 66% hence has areas of improvement.

2.2 Limitation of Existing System

Problem Studied	Technology used	Dataset used	Limitations
Real-Time Convolution Neural Networks for Emotion and Gender Classification	Xception models with changes in parameters	FER-2013	Low accuracy and biased dataset
Facial expression using SVM Classifier	Gabor features, Local Binary Pattern features, Principal Component Analysis, Support Vector Machine	Not Mentioned	It recognises only three emotions (happy, sad, neutral)
Facial Emotion Recognition using Deep CNN Based Features	VGG16, Support Vector Machine, ResNet50	Extended Cohn-Kanade	Dataset limitations
Emotion Recognition using a Cauchy Naive Bayes Classifier	Cauchy Naive Bayes classifier, Gaussian assumption	Self Made (Accuracy: 92.04%)	Biased towards western features
Hybrid Deep Learning Model for Emotion Recognition using Facial Expressions	CNN	FER-2013, JAFFE	Dataset limitations

Table 2.1 Limitation of Previous Works

2.3 Problem Statement

Communication channels are getting blocked tremendously as the current pandemic situation demands the interactions to be excessively virtualised, which otherwise too had its graph on the descent. It is very important to overcome the compromise of communication's efficiency. This can be achieved if there is access to a person's subconscious model of emotion delivery; their facial expressions, postures.

2.4 Objectives

The main objective of our system is to ensure that the users understand and identify the emotional states of the people they are communicating with.

- To create a dataset of our own to overcome the present bias towards the western facial features.
- To create a CNN model for emotion classification with the best performance.
- To validate the created model by creating a web-based conferencing system that will help a user understand and identify the emotional state of the person he/she is communicating with.

CHAPTER 3
Proposed System

3.1 Drawbacks of Current Systems

During the literature survey, we found that the current systems have many drawbacks and cannot be deployed for actual use. From the literature survey, we have concluded that there are three major recurring issues in the numerous implementations of current systems.

1. They have a bias towards western features. The models designed gravitate towards facial features of one ethnicity. Hence they are inefficient in prediction and analysis when dealing with a multitude of ethnicities.

2. The database does not have enough data. For proper sentiment analysis, the minimum number of sentiments/emotions a model must recognise is at least 5. However, many datasets condense the features into positive, negative, and neutral, giving us vague conclusions instead of the precise and insightful ones we expect from them.

3. The database is not balanced. The majority of the databases that exist are biased towards either one set of features or favour one particular expression. These systems are not capable of accurately detecting the emotion of the person either because of generalisation of one emotion (for example, one model leans towards showing output as happy; it has generalised the “happy”, “excited”, “surprised” sentiment) or they do not recognise the expression itself because of the features of the person.

3.2 Our Approach

We propose an implementation of a real-time CNN model for emotion classification,which includes 5 different emotions {“angry”, “happy”, “sad”, “surprised”, “neutral”} based on our use case. We validate our model by creating a video conferencing system which accomplishes the tasks of face detection and emotion classification collectively using our CNN architecture. The video conferencing system is a web-based platform that can help a user understand and identify the emotional state of the person he/she is communicating with.

We propose the following system:

1. Dataset

   We propose to generate our dataset from social media networks. Instagram is a social networking platform whose main method of communication is through image posting. The platform has an API that developers can use to mine for photos segregated by the user into various hashtags like #happy, #excited.

2. Pre-Processing

   The images collected have to be filtered for replications. They have to be cropped and greyscaled, and labelled to use in the recognition process.

3. Model

   The Xception model combines depth-wise separable convolutions and residual modules, which are responsible for its improved performance compared to other CNN models for the classification of datasets having high class. Moreover, it has the same amount of model parameters as Inception, implying that the computation cost is taken care of.

4. Deploying

   Our trained and tested model deployed through various platforms to communicate with the backend.

5. Usage

   ReactJS front end can use the API to send video footage loaded through the camera to the backend for processing.

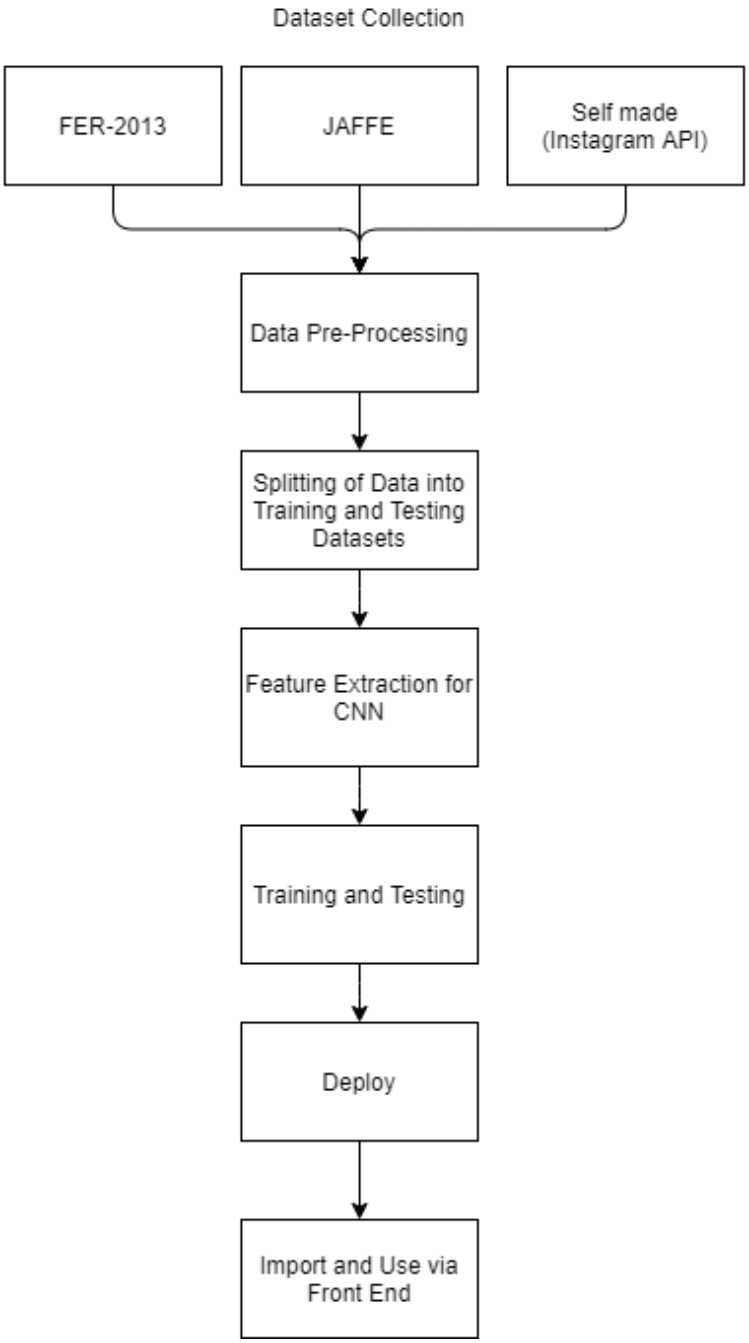


Figure 3.1 Flow of the System

3.3 Emotion Classification

The dataset collection is divided into training and testing sets. After which, the images are greyscaled, cropped and normalised. These are done with the motive to reduce the computation cost. The labels of images are converted to one-hot vectors, which aid our probabilistic model to predict and classify the images to their corresponding label based on probability. In order to

validate the features learned by CNN, we perform visualisation of guided-gradient backpropagation.

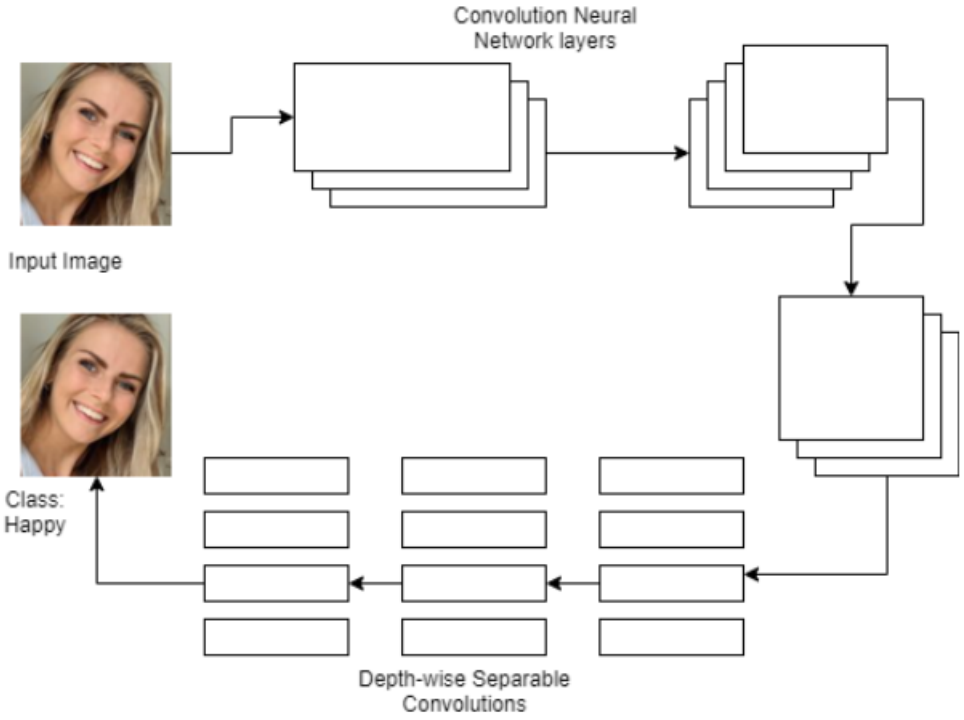


Figure 3.2 Skeleton of Working

CHAPTER 4

Design and Methodology

4.1 Design Details

4.1.1 Black Box Design

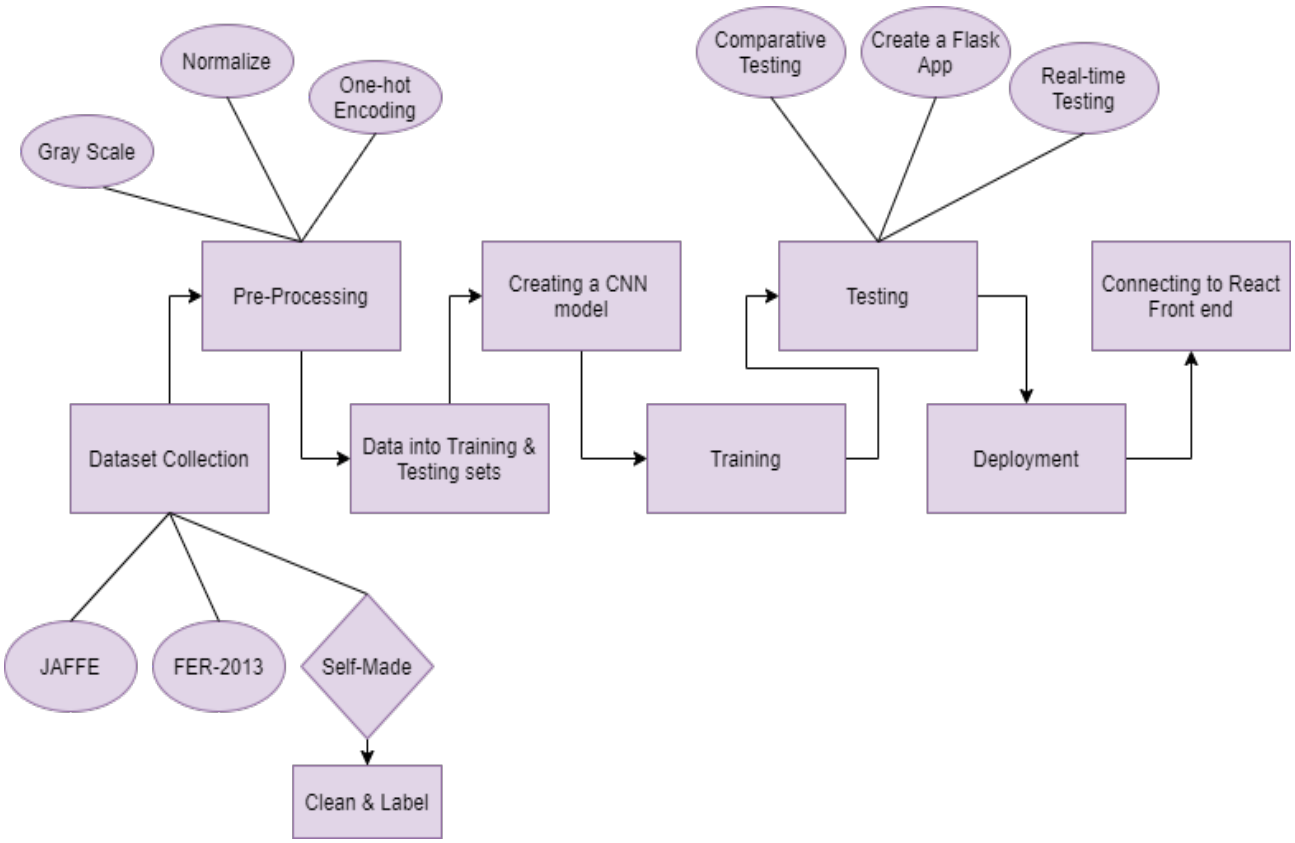


Figure 4.1 Black Box Model

4.1.2 Use Case Diagram

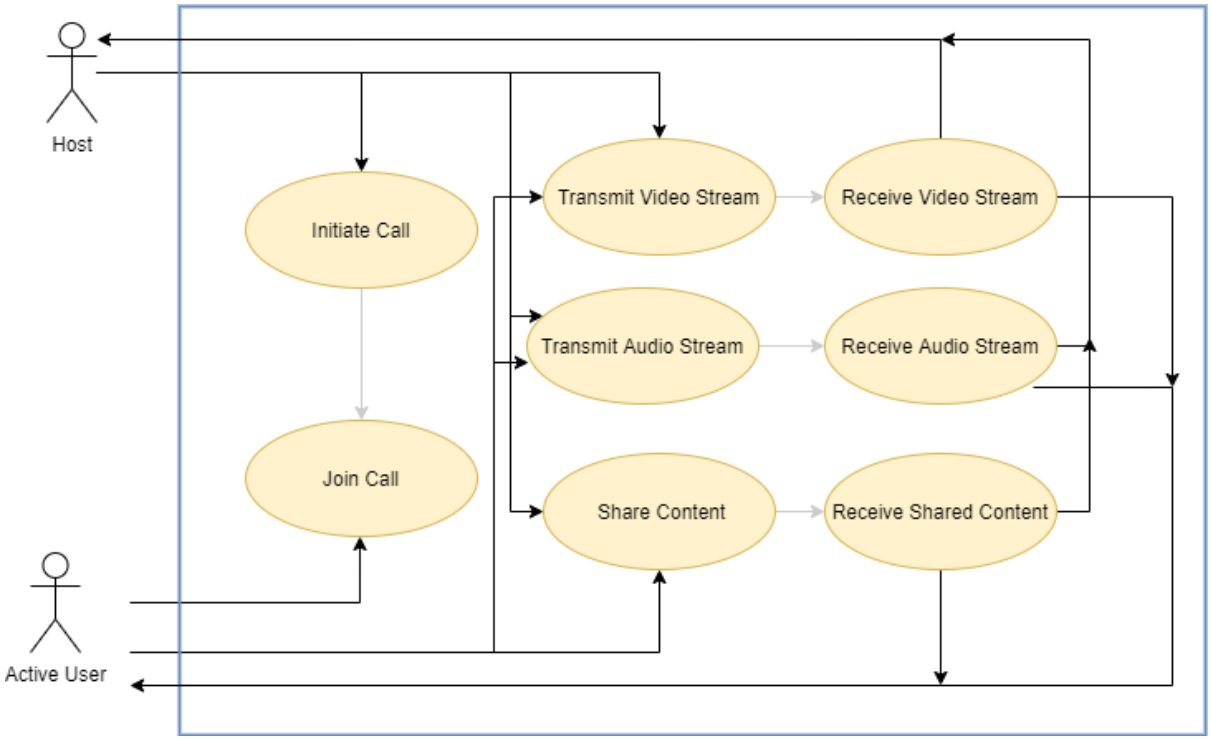


Figure 4.2 Use Case Diagram

4.1.3 Activity Diagram

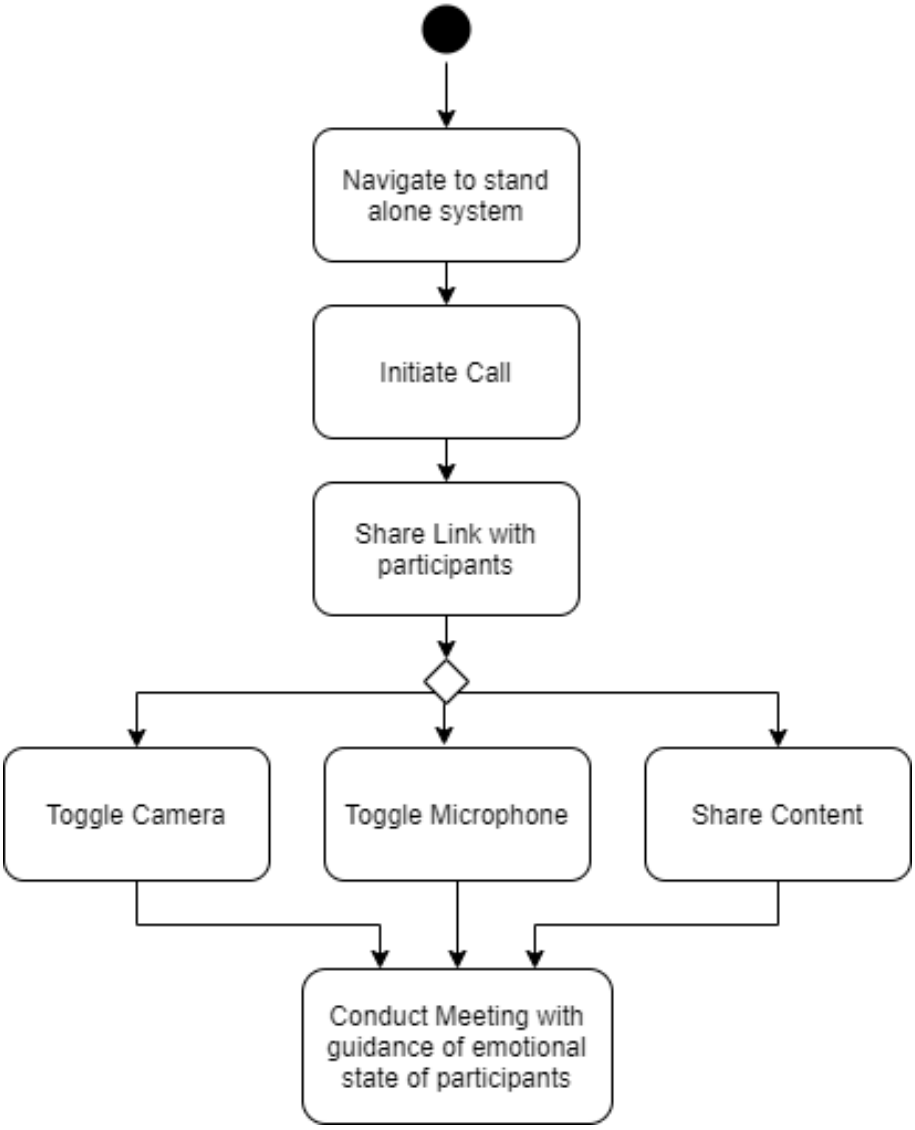


Figure 4.3 Host Activity Diagram

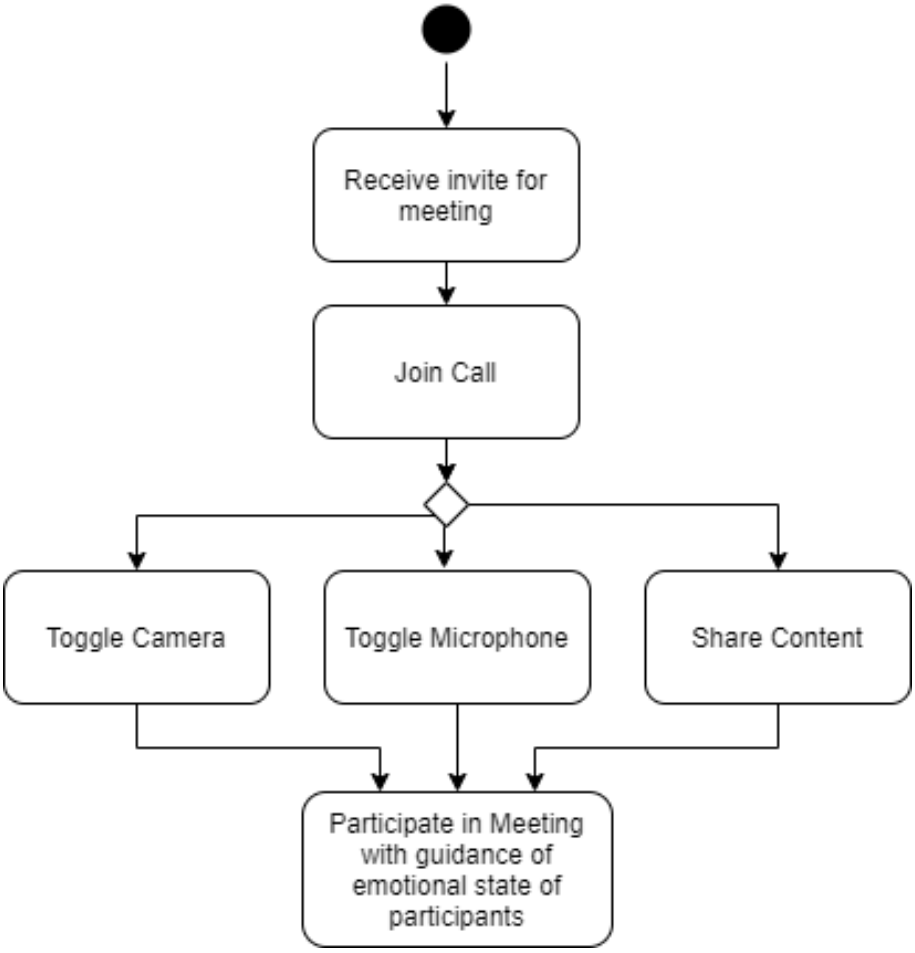


Figure 4.4 Participant Activity Diagram

4.2 Methodology

4.2.1 Dataset

Using machine learning techniques, the correct interpretation of each class of emotion has proved to be a complex task due to the high variability of samples within each class. The accuracy rate of humans themselves to classify an image of a face in one of the different emotions is 65% approximately. This itself is capable of enlightening us about the complexity of the process to be achieved. One can experience the mentioned difficulty in classifying the images from JAFFE or FER-2013 dataset into different emotions.

Figure 4.5 FER2013 Sample Images

Figure 4.6 JAFFE Sample Images

Moreover, the datasets which are already available are biased towards western features. The models trained on such datasets fail to be used on a global basis. Thus, we proposed to generate our dataset from social media networks along with the ones already available. Instagram is a social networking platform whose primary method of communication is through image posting. The platform has an API that developers can use to mine for photos segregated by the user into various hashtags like #happy, #excited. This would randomise the dataset and remove the bias towards one ethnicity. We would also yield a balanced dataset in this method.

Web scraping is a common terminology in the field of artificial intelligence and machine learning. The extraction of data from websites and converting that data into a valuable form for the users to employ in their programs is known as web scraping. In order to create a dataset by extracting images of hashtags from Instagram[1], the process of web scraping needs to be implemented. When any public Instagram page containing images from a particular hashtag is opened, an HTML page is returned by Instagram. Some pictures from the first few posts are already loaded with the help of server-side rendering. More pictures will continue to load as we scroll down the page using a request that goes to Instagram's GraphQL API[1] endpoint. Using this API, the content of that hashtag can be extracted from the Instagram web page.

Modules like *Beautiful soup* and *Selenium* have been used for this purpose.

Figure 4.7 Dataset Flow Diagram

Turicreate[2] is a module that is usually used for creating custom machine learning models. With the help of turicreate, we can build systems of image classification, object detection, recommendations, image similarity quickly. Since turicreate is easy to use and flexible, we have opted to use this library to classify the images we mined from Instagram. This library uses scalable data frames of SFrames to handle data. It is mutable and is used to scale data that is too big. After extracting all possible images from Instagram, we loaded the images from the dataset into an image sframe. Following which we created a sframe for future use using the

image sframe that we created as data. The sframe images were then converted into an array and pre-processed by resizing all the images into equal size.

Figure 4.8 Mined Images

4.2.2 Data Pre-processing

The data extracted manually usually has images that are inconsistent and contain discrepancies due to which they cannot be directly sent to the model for training. In order to procure the images in an understandable format as opposed to the raw form that it is in actuality, the collected data needs to be pre-processed before it is passed to the model. This ensures that the training model would get a complete and error-free dataset, thereby improving the accuracy and quality of data. The main objective of pre-processing the data is to validate that the dataset is ready to be trained and analysed.

The first step includes the removal of irrelevant or duplicated images. Pre-processing would not be required for the images obtained from a readily available dataset as those datasets make sure that the images are fit for training. The images acquired via Instagram's API[1] need to be manually cleaned as the images collected are randomised. The data so created should also be correctly labelled for its corresponding emotion. We then split the whole data into training

and testing sets. This is done using the OpenCV module. The images hence obtained will all be of free sizes. For a model to accurately classify images, it should be paramount that the images not consist of any noise or distracting elements. For example, each image will have some background or lighting effects that would result in the model not identifying the features correctly. For this purpose, the images need to be clipped such that only the face is visible and is in focus. The cropped images are then converted to a float32 format, after which we normalise its pixel values. These are performed to create uniformity on the images and to reduce the computation cost significantly. The images are then greyscaled, which is an essential step for pre-processing[8] as coloured images require more training parameters, thereby leading to overfitting of the data. On learning that the probabilistic machine learning model cannot recognise the labels, the labels were converted into one-hot vectors and used by the model to predict the probability of an image corresponding to the specific label.

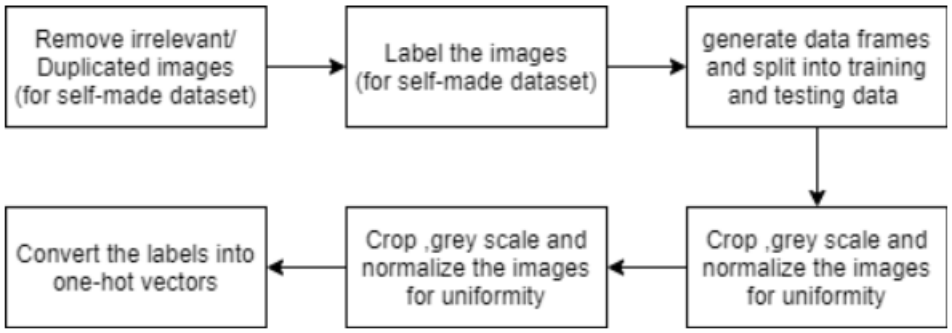


Figure 4.9 Pre-Processing of Images

4.2.3 Model

We proposed to use Convoluted Neural Network on our now diverse dataset for optimal performance. Modules like *Keras* and *scikit-learn* are used to make such a model.

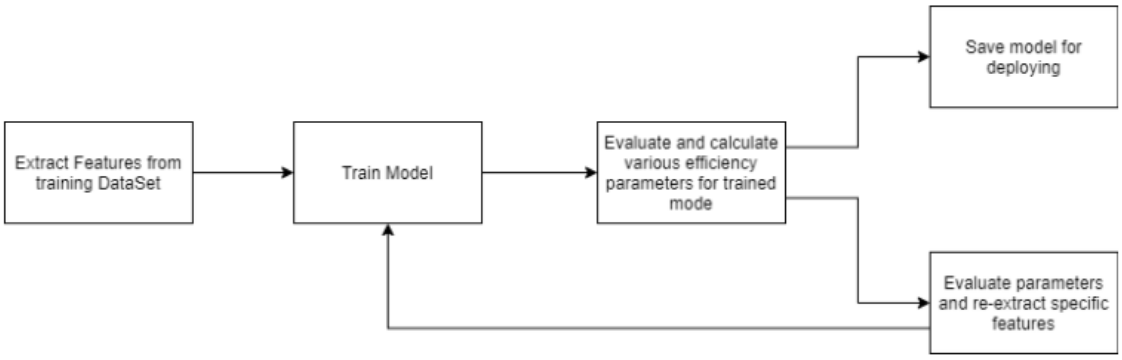


Figure 4.10 FLow of Model

4.2.3.1 Comparison of CNN Architectures

On a global survey of all the Deep Learning approaches where various learning techniques, models and recently proposed training approaches were acquainted, we could infer that the progress in Deep Learning could be boiled down to just a handful of CNN architectures. Thus, the SVM classifier[3] with the maximum accuracy of 72% or the Cauchy's Naïve Bayes Classifier[4] with the maximum accuracy of about 80% were not our concerned approaches. Moreover, these models are trained on the JAFFE dataset alone. Thus, when the images of people's faces exhibiting western characteristics are fed, the accuracy rate drops to less than 53%.

The now archetypal layout of CNNs used commonly in feature extraction include a series of convolutional, max-pooling and activation layers before some fully connected classification layers at the end, and most of the parameters are contained by these fully connected layers. For example, the VGG16[5] contains 90% of its parameters in the aforementioned fully connected layers and manages to achieve a top-1 accuracy of 71% on the ImageNet validation dataset. The ResNet50[11] model came up with a crisp single hypothesis that direct mappings are hard to learn for the performance degradation of deep networks undergoing multiple additions of layers. This model is much lighter than VGG16 yet manages to score an accuracy of 74% on the same dataset, along with a relatively considerable reduction in the parameters. In comparison, the InceptionV3 model focuses more on the computational efficiency of training larger nets. It is lighter than the RestNet50 and manages to score an accuracy of 78%. However, the Xception models replace the Inception modules with depth-wise separable convolutions. They have the

same number of model parameters as Inception, which implies that the computational efficiency is not compromised. The Xception[7] model slightly outperforms InceptionV3 on the ImageNet dataset and largely outperforms it when the classes are very high on a larger image classification dataset. Thus, it truly is an eXtreme to Inception as per its name.

4.2.3.2 Xception Model

A convolutional neural network based entirely on depthwise separable convolution layers, compared with InceptionV3, seemed to be a better version of the underlying Inception architecture. Having been trained on the ImageNet and JFT dataset, an image classification and object detection-based dataset, the Xception model already promised to deliver higher accuracy than other models such as ResNet, VGG and Inception. An extreme version of the Inception network, Xception[7], differs from it in that, it heavily uses depth-wise separable convolution along with residual connections in its architecture.

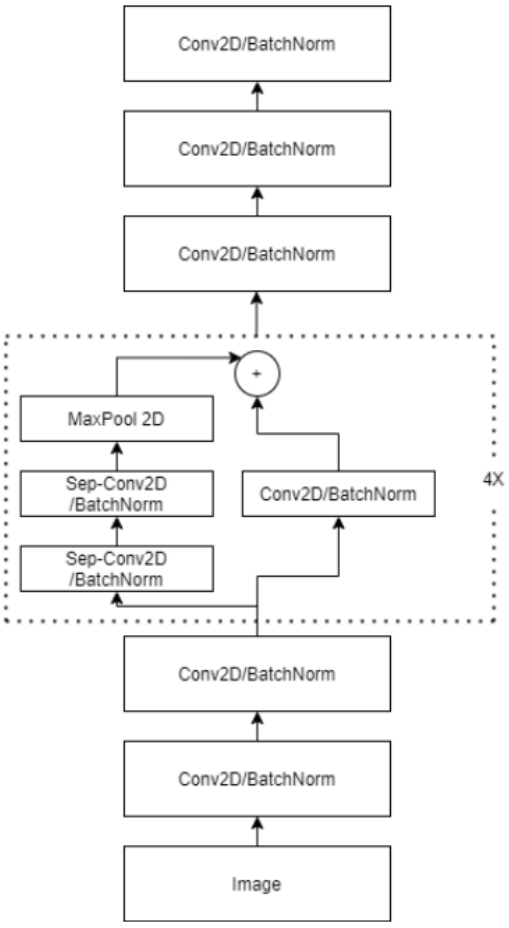


Figure 4.11 System Architechture

As seen in the figure, the model architecture[13] contains four residual depth-wise separable convolutions. Each of the convolution layers is followed by a batch-normalisation operation. This results in the standardisation of inputs to a layer for each mini-batch. A ReLU activation function also follows each convolution layer. The ReLU activation function has an important association with backpropagation[5] during the model's training. ReLU helps prevent exponential growth in the computation needed for the neural network operation. Hence, the computational cost of adding a ReLU increases linearly if the neural network scales in size. Following this, the last layer generates the prediction by applying a global average pooling and a soft-max activation function. The architecture is trained without and with Adam optimiser[12], and enhanced performance was deduced in the latter case.

The Xception model combines the residual modules and depth-wise separable convolutions.

- Residual Module:

  The residual module is a fix for the hypothesis: *direct mappings are hard to learn.*

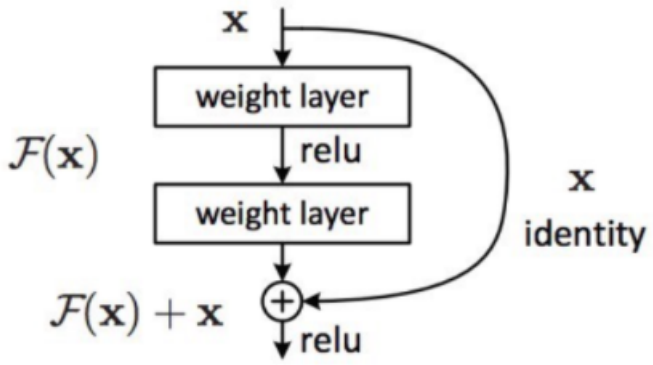


Figure 4.12 Residual Module

The module implies to directly learn the difference between x and H(x), or the 'residual' instead of trying to learn the underlying mapping between the two. That is, if :

$$F(x) = H(x) - x \text{ is a residual}$$

Then rather than trying to learn the H(x) directly, learn

$$H(x) = F(x) + x$$

Thus, the desired features H(x) are modified to an easier learning problem. Each block or module in Xception is followed by a shortcut connection which adds the input of the block to its output and the add operation is performed element-wise.

- Depth-wise separable convolutions:
  Depth-wise separable convolutions[7] reduce the computation cost considerably in comparison to the standard convolutions. This is achieved by the separation of spatial cross-correlations from the channel cross-correlations by the two different layers present in depth-wise separable convolutions:

  1. Depth-wise convolutions
  2. Point-wise convolutions

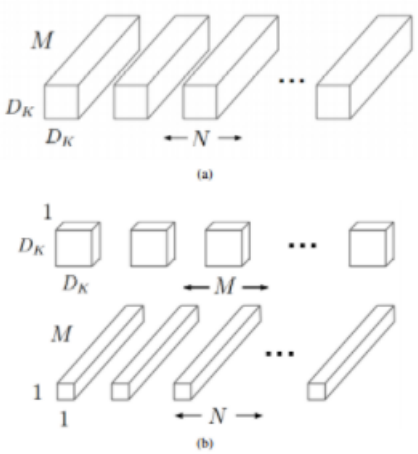


Figure 4.13 Depicting difference between (a) Standard convolutions and (b) Depth-wise separable convolutions

The depth and spatial dimensions of a filter are separated in Depth-wise Separable Convolution. Here the depth-wise convolution is followed by a 1*1 filter to cover the depth dimension. The major advantage of these convolutions compared to depth-wise convolutions or standard convolutions[13] is the reduced number of parameters to output the same number of channels.

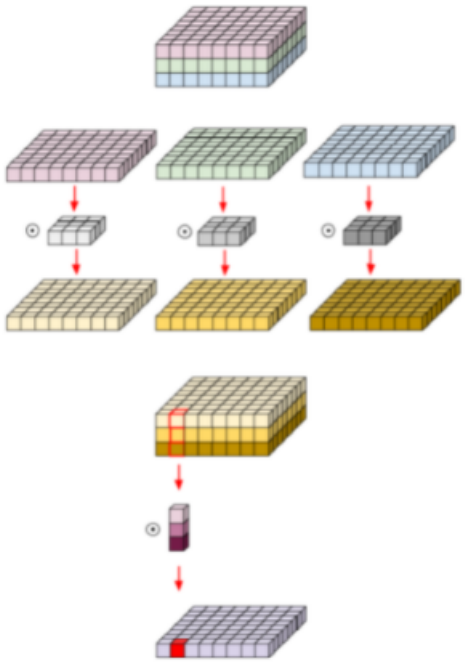


Figure 4.14 Depth-wise separable convolutions

Mathematically, we require 3*3*3 parameters and 1*3 parameters in-depth dimension to produce one channel in depth-wise convolutions. However, in depth-wise separable convolutions to produce three output channels, the only requirement is of three 1*3 filters, which gives us a total of 27 + 9 = 36 parameters.

While in the case of standard convolutions, to produce the same number of output channels, we need three 3*3*3 filters, giving us 81 parameters in total. The problem of having too many parameters is that it forces the function to memorise lather and thus over-fits.

Depth-wise, separable convolutions are the real saviour in that case.

4.2.3.3 Visualisation

Often, in CNN, the learned features remain hidden, which disrupts the balance between classification accuracy and unnecessary parameters. In order to overcome this, a real-time visualisation technique by Springenberg is implemented. The images are reconstructed by real-time backpropagation[6] to observe the pixels responsible for activating an element of a higher-level feature map. These are performed by the ReLU activation functions added in intermediate layers.

A reconstructed image can be given by,

$$R_{i,j}^{l} = (R_{i,j}^{l+1} > 0) * R_{i,j}^{l+1}$$

4.2.4 Deploying

Our trained and tested model for emotion classification is first deployed locally using *Flask*. The front end uses the API to send video footage loaded through the camera to the backend for processing. After trying, testing and improving the results obtained, we validate our model by creating a conferencing system using *ReactJS*. The UI of the system is kept simple yet made sure to meet all our purpose. The flask API made is then connected to React frontend; thus, the system is now an emotion classifying conference system. The system is then deployed using *Heroku*.

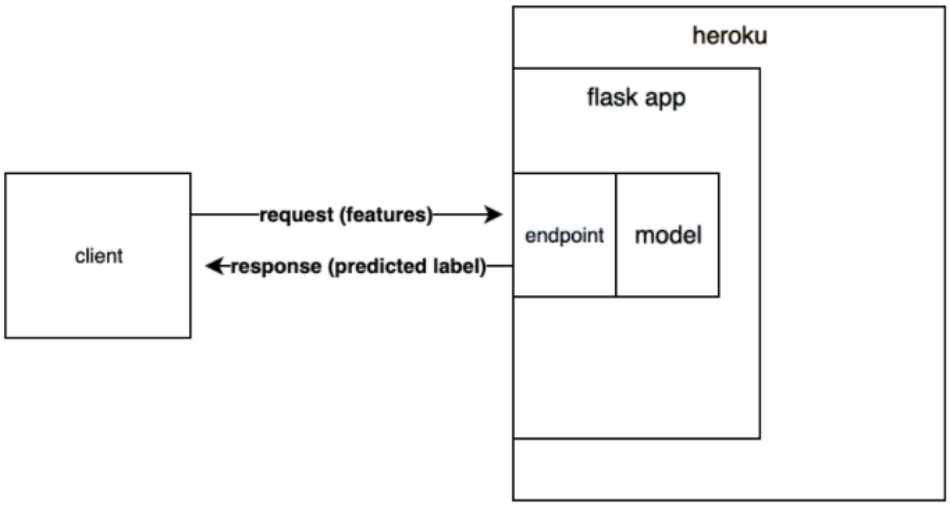


Figure 4.15 Deploying Model FLow Diagram

4.3 Details about Hardware and Software

Our experiments have been conducted on a PC with Intel Core i5 CPU of 8th Generation and Intel UHD Graphics 620. We have trained our model on Google Colaboratory with a 16GB NVIDIA Tesla T4 GPU hardware accelerator. We use Google Colaborratory to train and test our model with the collected dataset.

4.3.1 Hardware Requirements

1. RAM: 4+ GB DDR4, 1TB HDD (5400 rpm)
2. GPU: 16 GB NVIDIA Tesla T4
3. CPU: Intel i5 8th Generation 3.9 GHz

4.3.2 Software Requirements

1. Python 3: The construction of the model, analysis of knowledge and displaying of output on the online page is all done using the Python language. It is a high-level, interpreted, interactive and object-oriented scripting language.
2. Flask: Flask is a web framework that provides tools, technologies and libraries to create a web application. It is a third-party app that uses HTML to send the contents of the Python codes to the web page.
3. HTML: Stands for HyperText Markup Language. It is a popular language used for creating web pages and describing the structure of a web page. It is combined with languages such as Cascading Style Sheets and Javascript to create a complete web application.
4. CSS: Short for Cascading Style Sheets, it decides how the HTML elements should look on a web page. It is mainly used for making a web page attractive and interactive. It is used to control the layout of many web pages at the same time.
5. ReactJS: It is an open-source Javascript library used for making user interfaces. It is a predefined framework that is efficient and flexible and can be used to make complex user interfaces.

6. Sklearn: It is a free and popular machine learning library in the python programming language, commonly used for regression models, classification and clustering algorithms such as support vector machines.

7. OpenCV: It is an image and video processing machine learning library primarily used in real-time computer vision models.

8. Pandas: It is a free software library that is usually used to analyse and manipulate data. It is fast and powerful to use and is built upon the Numpy library.

9. Keras: It is a well-known library built on top of TensorFlow's machine learning module. It provides an interface in python for artificial neural networks.

10. Tensorflow: This module is exclusively used for machine learning and artificial intelligence practices. It has a wide range of applications but is primarily used for deep learning. It is a math-based library that is based on the flow of data and differential programming.

11. Matplotlib: It is a plotting library in python used for plotting and embedding various types of plots into different applications. It is commonly used for data visualisation and plotting graphs.

12. Numpy: NumPy is a Python programming library used for working with arrays and matrices. It also has functions for working in the domain of linear algebra, Fourier series transform.

13. Pillow: It is an image library used to open, use, save, and manipulate various types of image formats. It provides a strong foundation for image processing as well.

CHAPTER 5
Results and Discussions

5.1 Implementation

5.1.1 Dataset

The initial step of training any model involves pre-processing of data so that the model will get an improved quality of data instead of the raw form it was before. Sending the data in a raw format would lead to considerably lower accuracy in training and testing as the model would not correctly classify and identify the features required to predict the class. Our module has pre-processed our data, which included filtering, cropping, removing replicated images, normalising and greyscaling[8]. The data was further split into two – the training and testing dataset in a ratio of 90:10, which will be used for evaluating the precision of the model.

Figure 5.1 Examples of 48x48 sized and greyscaled images

5.1.2 Process

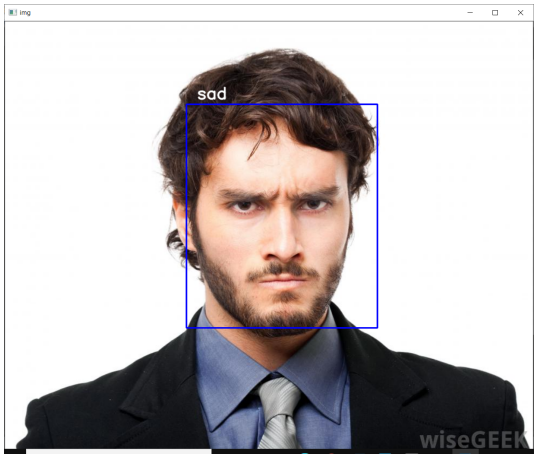
Our project primarily encompasses three modules, the first one being the training and testing of the model, the second one is the emotion recognition from faces in real-time and the final one where the output from the emotion recognition is passed to the frontend where the emotions are classified for multiple users simultaneously in a video conferencing setting.

We use the OpenCV module to open the webcam and capture the live video feed. OpenCV is a widely known library that can be used to perform image and video processing. Using this library, we capture the video object and continuously send individual frames to the calling object. The frames are then sent to OpenCV's already present haar cascade classifier, which is used to identify faces. It is an algorithm of object detection used to detect faces in an image or a real-time video.

The haar cascade classifier[12] is an already pre-trained model and is highly accurate in identifying faces. It is capable of detecting up to a thousand faces in one picture itself. Apart from face detection, haar cascade models are further used explicitly for identifying eyes and lips, upper and lower body detection, license plate detection. To summarise what the model does, the haar feature incessantly traverses from top to bottom of an image to search for a particular feature, a face in our case. There are four types of haar cascade classifiers for detecting faces – alt, alt2, alt_tree and default. The default classifier is the best amongst those four and gives the most accurately detected face, which we used in our model.

After identifying the face, the image is cropped such that only the face is in focus and re-sized to 48x48 because the model trains on images which are of 48 by 48 size. The image then predicts, compared with the trained model that we have obtained and classified the corresponding emotion. That emotion is returned and displayed to the user with a bounding box around the face and the labelled emotion on top of it. Similarly, by continuously sending such image frames to get processed and receiving the emotion labelled output, we can see the result in a real-time format where the emotions change as fast as our faces move.

5.1.3 Comparison between SVM and Xception



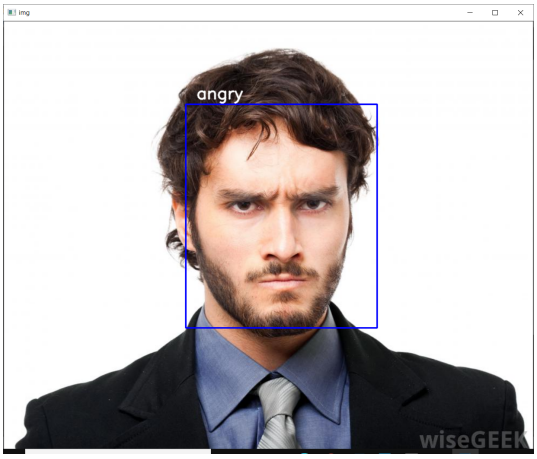


Figure 5.2 Example result of SVM Classifer and Figure 5.3 Example result of Xception model

As seen in the above figures, the actual image is of an angry person. On initially training the dataset with the SVM classifier, we obtained an accuracy of about 67%. Having achieved a relatively low accuracy, it did not promise to identify the emotions correctly. When the camera loads the live video feed, it represents a few emotions that resemble some other emotions. For

example, a surprised person is sometimes misclassified as happy and a neutral person as sad.

Similarly, on training the dataset with the Xception model we had proposed to use, we obtained a considerably high accuracy of about 94%. The emotions were correctly classified most of the time. As seen above, we have two images of the same emotion, which shows the difference in using two different classifiers.

5.1.4 The Model

With the dataset that we have collected, all the images that were split into training and testing datasets were converted into a NumPy array, an integer array. Training happens faster when the input consists of integer values. These were further normalised and reshaped to be passed to the Convolution2D layer, where the hyperparameters need to be mentioned. Each Convolution2D layer is followed by a Batch Normalisation layer and the ReLU activation function.

Batch Normalisation is a type of training optimisation method. It is a standard method used in CNNs, which results in each layer of the network learning more independently. As the name suggests, it is the normalisation of outputs of the previous layers.

We have noted that the convolutional neural networks work significantly better with the ReLU activation function than other functions. We tried using functions like sigmoid and tanh to compare what gives a better output, even if it is only a marginal difference in the result. However, on using ReLU, we found out that this function worked better in our situation for the emotion recognition training. Alternatively, adding a linear activation function would have meant that an ordinary multiplication of matrices is occurring. In that case, the ability to approximate the network would have had a limited scope. Using a non-linear activation function[5] such as ReLU, the expression ability of the convolution neural network becomes more powerful.

Figure 5.4 Visualisation of backpropagation

The ReLU activation function's equation is defined by,

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

It is clear from the equation that ReLU is a function that will give the maximum value.

From the Xception architecture, it can be seen that residual has been used. It has been experimentally tested and validated that an Xception network with residual connections performs much better than a network without residual connections[7]. That is, the accuracy is relatively high by using residual connections. The separable convolution layers that follow are part of the main concept of the xception architecture. When compared with normal convolutions, it is realised that the separable convolutions perform a lot fewer computations. Hence, the network will be able to process more in a short period. In a normal convolution layer, each image has to be transformed multiple times, which lead to many multiplications. In contrast, inseparable convolution is transformed only once and is elongated to the different channels. Due to this remarkable difference between the two, the network achieves more in less computation power.

The pooling layers like Max Pooling and Global Average Pooling are used as dimension reduction techniques that reduce the dimensionality of each feature while simultaneously summarising the features generated by the convolution layer. A fully connected layer is usually the last step of a network. Here, the Global Average Pooling is used instead to reduce each channel in the feature map to a single value sent to softmax for generating probabilities.

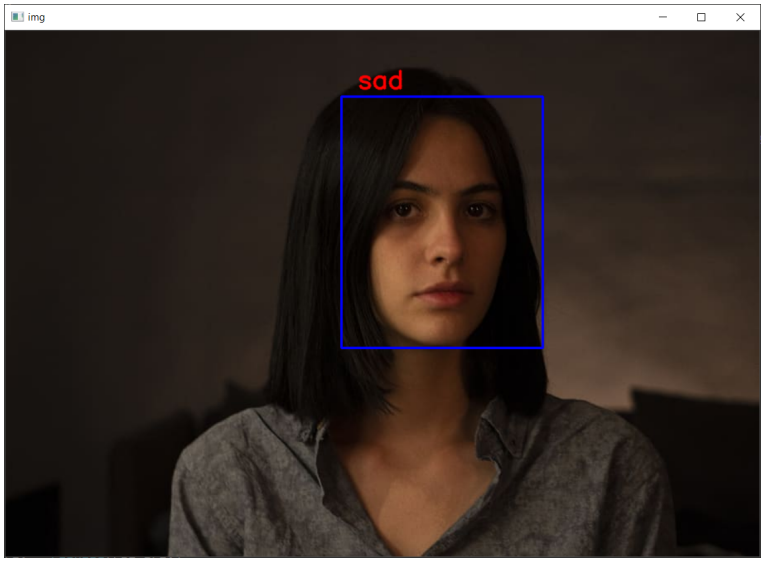


Figure 5.5 Before using Adam Optimizer

As seen in the image above, we were not getting accurate results of the 'Neutral' emotion. While we were getting happy, sad, surprise and anger correctly, the neutral face was majorly shown as sad and frequently fluctuated between neutral and sad.

Initially, we had executed the model without using any optimiser. On getting dissatisfaction slightly, we proposed using an optimiser to enhance the training and testing accuracy of the model. We chose to go with Adam optimiser on learning that it is greatly used to reduce the prediction cost of the model during backpropagation as it calculates individual adaptive learning rates for different parameters. The hyperparameters required for the algorithm are alpha, also known as learning rate, two beta values and epsilon. After trying and testing with a few values, we settled on a learning rate of 0.001, beta1 as 0.9, beta2 as 0.999 and the epsilon value as 1e-08[12]. The epsilon value of 1e-08 showed better performance with validation loss as compared to other values. We kept the size of the mini-batch as 32 and the number of epochs as 50. On executing the model with these parameters, we found the model to perform better than earlier iterations. Additionally, the model could classify the neutral emotion on the face for a stable period.

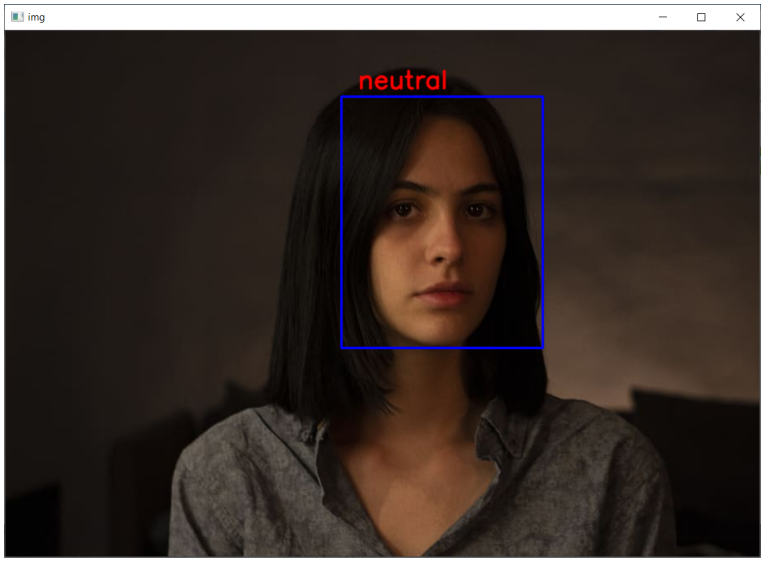


Figure 5.6 After using Adam Optimizer

From the above figure, we can see that after using the Adam optimiser, we could get marginally better results that ultimately contribute to more correct classifications.

	Precision	**Recall**	**F1-Score**	**Support**
Angry	0.92	0.92	0.92	501
Happy	0.94	0.98	0.96	879
Sad	0.95	0.89	0.92	608
Surprise	0.98	0.95	0.96	420
Neutral	0.91	0.94	0.92	614

Table 5.1 Precision, Recall, F1 Score

	Precision	**Recall**	**F1-Score**	**Support**
Accuracy	-	-	0.94	3022
Macro avg	0.94	0.93	0.94	3022
Weighted avg	0.94	0.94	0.94	3022

Table 5.2 Accuracy Metrics

5.1.5 Deploying to Heroku

We had initially connected the model's results to the web page via the Flask module. The real-time emotion classification data obtained on the camera were sent to render the HTML page as a Flask response object. After connecting it to Flask, its API was used to send the same data to a video conferencing user interface. Thus, multiple users in a video call setting can view their and others' emotions on their respective screens.

5.2 Testing

5.2.1 Testing with Happy

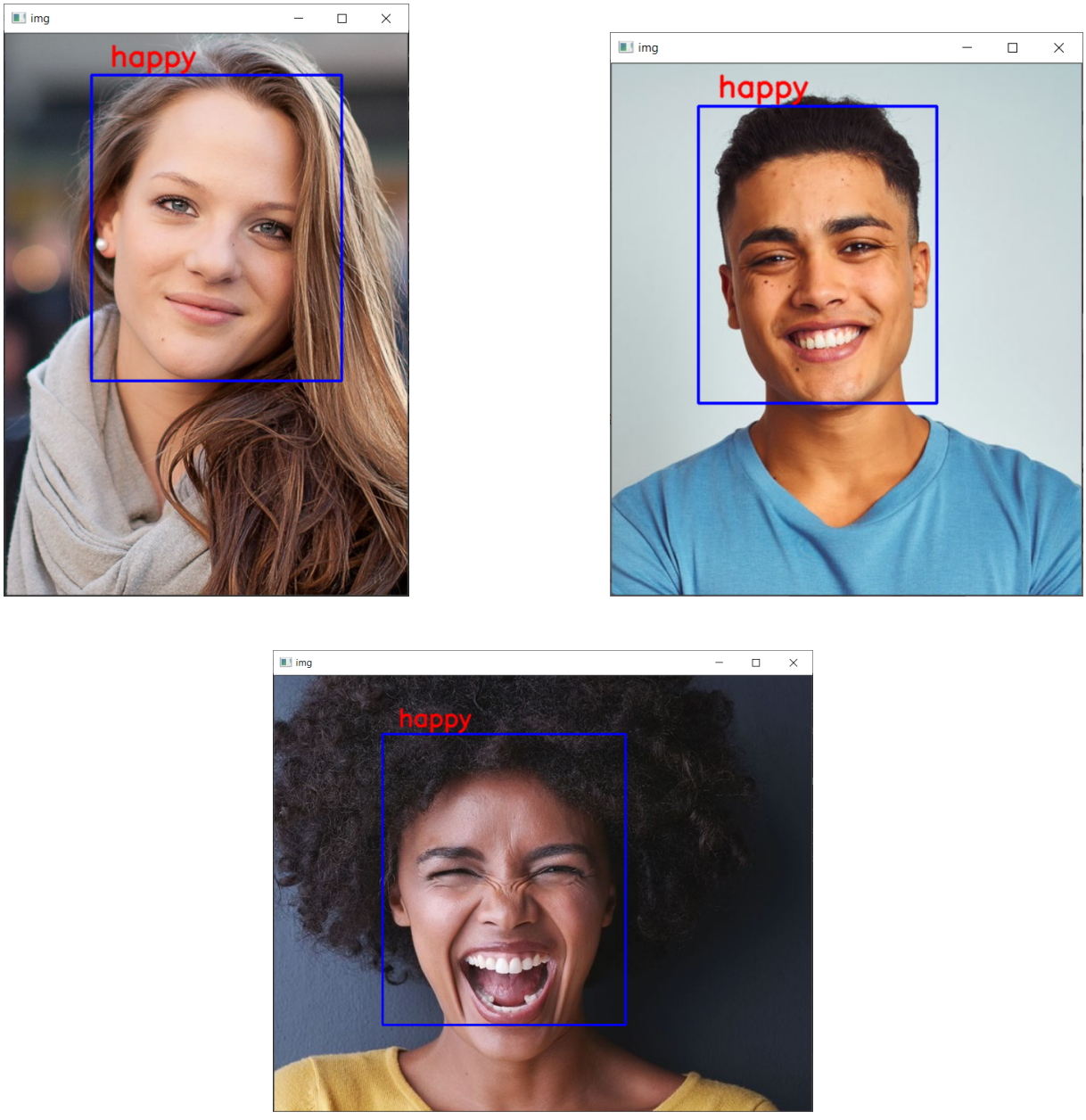


Figure 5.7 Testing with Happy emotion

5.2.2 Testing with Sad

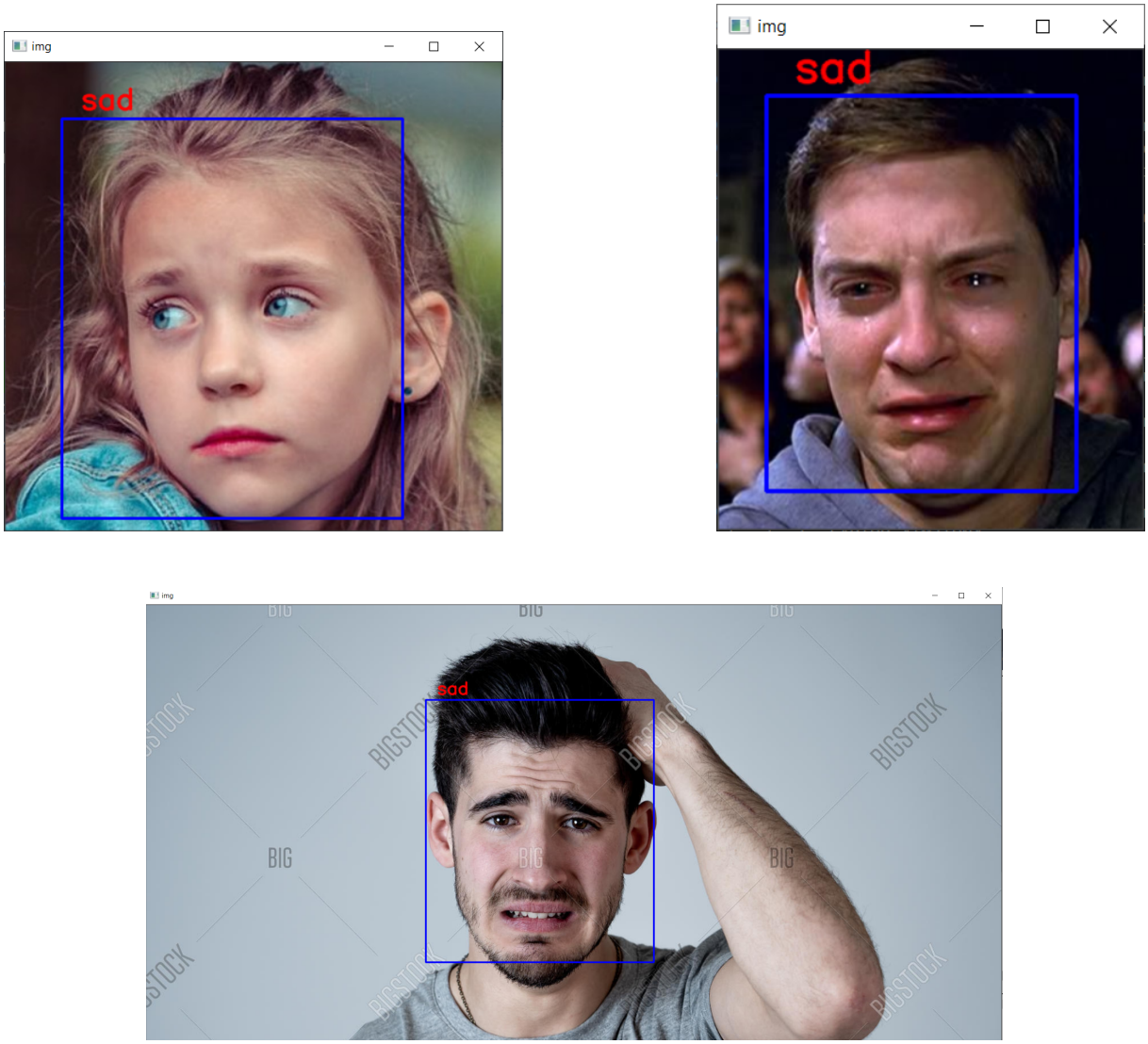
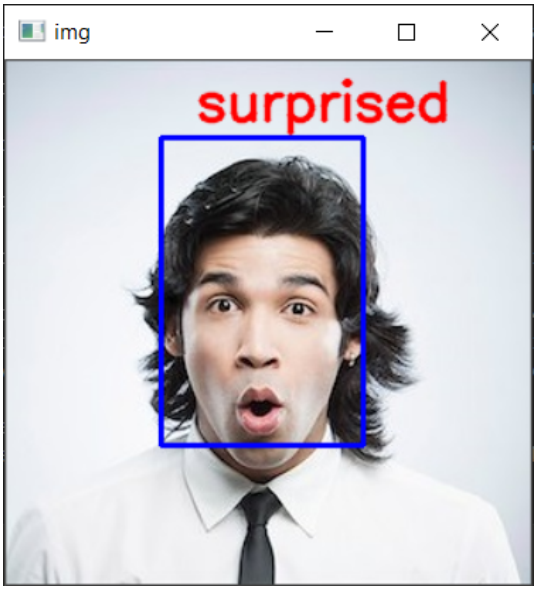


Figure 5.8 Testing with Sad emotion

5.2.3 Testing with Surprise



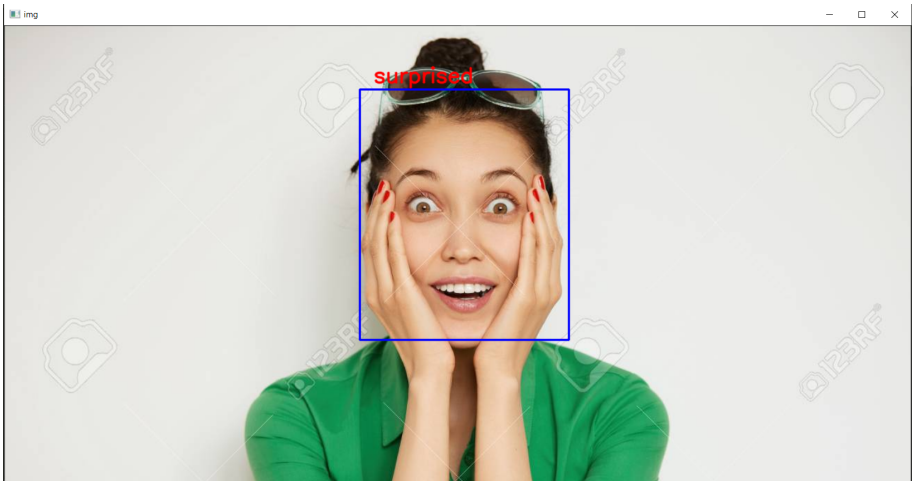


Figure 5.9 Testing with Surprise emotion

5.2.4 Testing with Anger

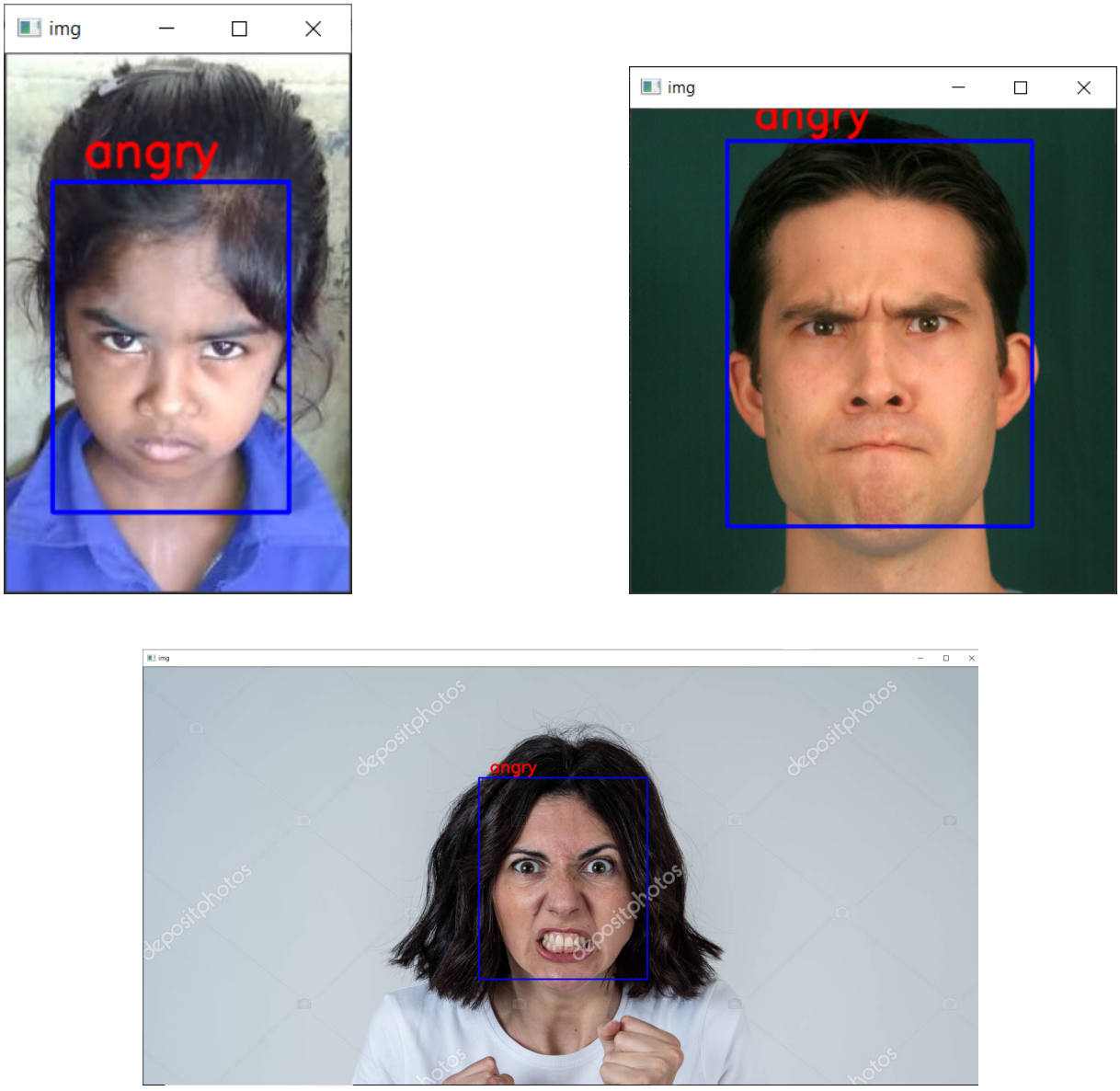


Figure 5.10 Testing with Angry emotion

5.2.5 Testing with Neutral

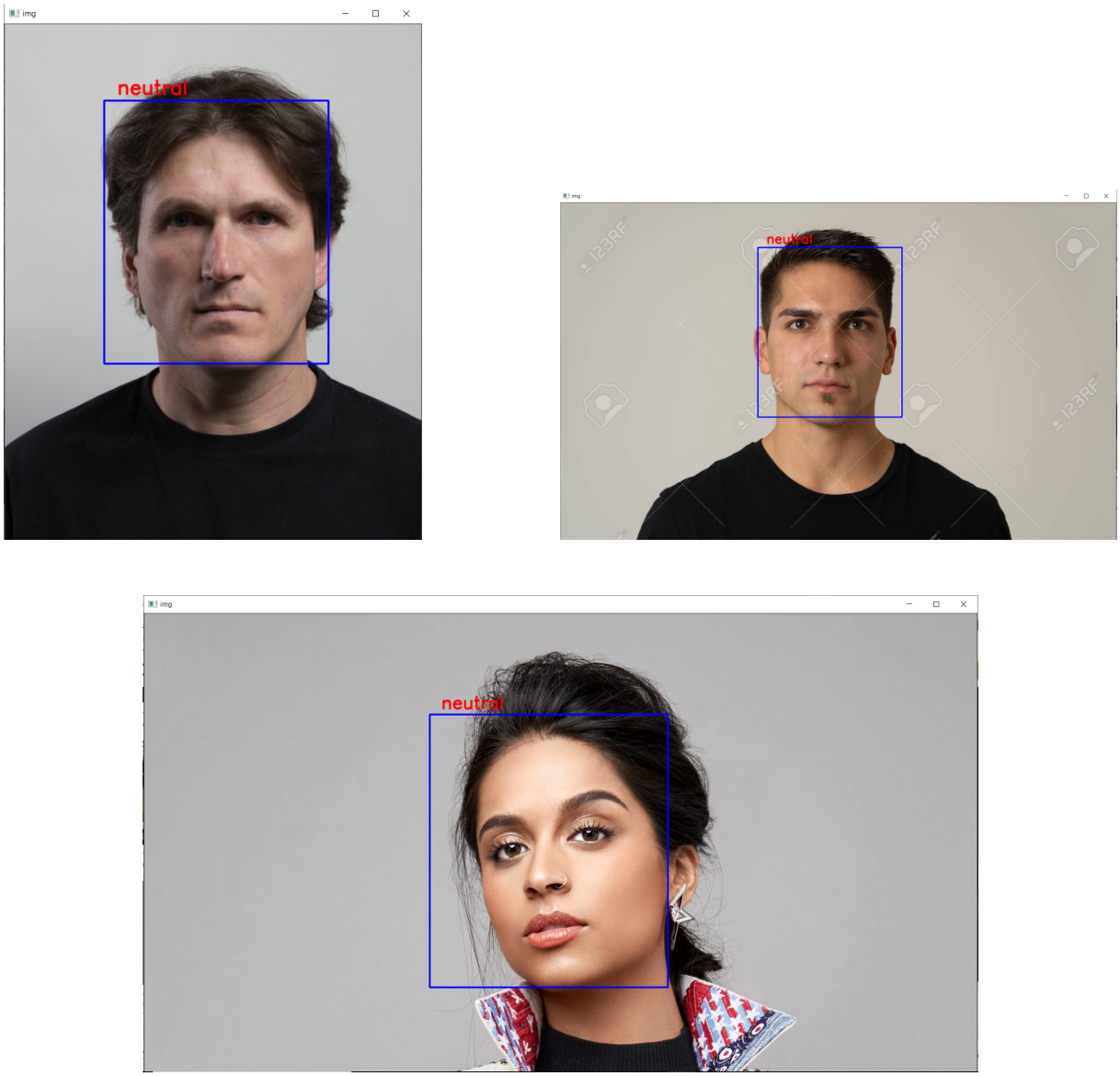


Figure 5.11 Testing with Neutral emotion

5.3 Results

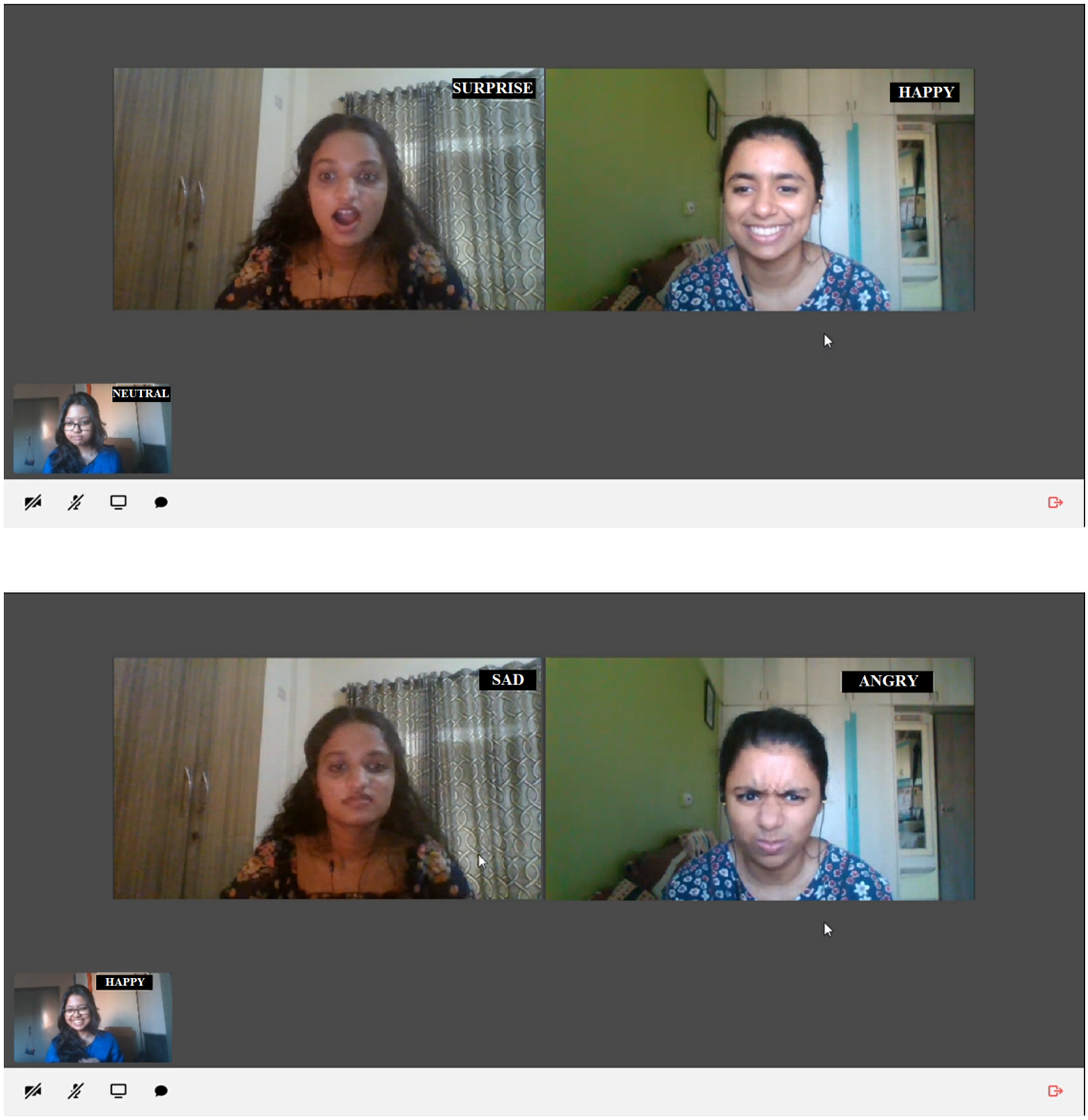


Figure 5.12 Video Conferencing User Interface with Emotion Detection

CHAPTER 6
Conclusion and Future Scope

6.1 Conclusion

In light of the pandemic and the reliance on virtual medium for communication, our project could make the tempering into virtual medium easier for all entities involved. Teachers, students, doctors, patients, lawyers, analysts, and so many more people who are now forced to rely on a new mode of interaction will benefit from our initiative.

After critically analysing current implementations and weighing the pros and cons of each of them, we have proposed making a working model for sentiment analysis using the CNN approach using our self-made dataset. The shortcomings from presently available systems are taken into consideration and have been avoided. Our system has reassuring statistics pointing towards an unbiased and inclusive analysis.

6.2 Future Scope

As we know, the human expressions one can come across is almost endless. So, the scope in this area goes on forever. Our model, as of now, is capable of classifying five emotions that we have already discussed. The further emotions which can be added are contempt, disgust, smirk, overwhelmed. The improvements in the area of accuracies can also be considered. The model currently works with an accuracy of about 94%, which can be improved on further analysis. We could observe a setback in classification when people with wrinkled faces were encountered. The model classifies particular emotions based on how the faces exhibit the change in emotions, i.e. by considering changes of shape of one's eye or the curve of lips. For older people, these changes hide considerably under their loose skin model tends to perform low. Additionally, speech inflections can be integrated with facial expressions in order to get an accurate prediction based on not just the faces but also the way we speak. It will be able to give precise classification of emotions as compared to other models.

In our project, we have created a system for a video conference where people can interact with each other and gauge their emotions. In the future, we could create our system to truly act as an umbrella software and cater to other applications.

REFERENCES

[1] Instagram API for mining images

[2] Turi Create Documentation for dataset generation and processing

[3] PC, Vasanth & KR, Nataraj. (2015). Facial Expression Recognition Using SVM Classifier. Indonesian Journal of Electrical Engineering and Informatics (IJEEI).
3. 10.11591/ijeei.v3i1.126

[4] N. Sebe, M. S. Lew, I. Cohen, A. Garg and T. S. Huang, "Emotion recognition using a Cauchy Naive Bayes classifier," Object recognition supported by user interaction for service robots, Quebec City, Quebec, Canada, 2002, pp. 17-20 vol.1, doi: 10.1109/ICPR.2002.1044578.

[5] Quanming Liu, Jing Zhang, and Yangyang Xin. 2019. Face expression recognition based on improved convolutional neural network. In Proceedings of the 2nd International Conference on Artificial Intelligence and Pattern Recognition (AIPR' 19). Association for Computing Machinery, New York, NY, USA, 61–65. DOI: https://doi.org/10.1145/3357254.3357275

[6] Verma, G., Verma, H. Hybrid-Deep Learning Model for Emotion Recognition Using Facial Expressions. Rev Socionetwork Strat 14, 171–180 (2020). https://doi.org/10.1007/s12626-020-00061-6

[7] F. Chollet, "Xception: Deep Learning with Depthwise Separable Convolutions" 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 1800-1807, doi: 10.1109/CVPR.2017.195.

[8] Fatima, Syed & Kumar, Ashwani & Raoof, Syed. (2021). Real Time Emotion Detection of Humans Using Mini-Xception Algorithm. IOP Conference Series: Materials Science and Engineering. 1042. 012027. 10.1088/1757-899X/1042/1/012027

[9] M. A. Ozdemir, B. Elagoz, A. Alaybeyoglu, R. Sadighzadeh and A. Akan, "Real Time Emotion Recognition from Facial Expressions Using CNN Architecture," 2019 Medical Technologies Congress (TIPTEKNO), 2019, pp. 1-4, doi: 10.1109/TIPTEKNO.2019.8895215.

[10] Jyostna Devi Bodapati, N. Veeranjaneyulu "Facial Emotion Recognition Using Deep Cnn

Based Features", International Journal of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-8Issue-7, May 2019.

[11] Franzoni V, Biondi G, Perri D, Gervasi O. Enhancing Mouth-Based Emotion Recognition Using Transfer Learning. Sensors (Basel). 2020;20(18):5222. Published 2020 Sep 13. doi:10.3390/s20185222

[12] Poulose, Alwin & Kim, Jung Hwan & Han, Dong. (2021). The Extensive Usage of the Facial Image Threshing Machine for Facial Emotion Recognition Performance. Sensors. 21. 2026. 10.3390/s21062026.

[13] Arriaga, Octavio & Valdenegro, Matias & Plöger, Paul. (2017). Real-time Convolutional Neural Networks for Emotion and Gender Classification.