A

PROJECT REPORT

ON

DETECTION OF HUMAN BIOLOGICAL EMOTIONS USING OPENCV AND TENSORFLOW

Submitted by

Sagar Ramrao Bharad

Prashish Gautam Borkar

Shreya Bhojraj Gawali

Janhavi Jayant Karande

AnilKumar Ramji Sharma

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GOVERNMENT COLLEGE OF ENGINEERING,

CHANDRAPUR- 442403

GONDWANA UNIVERSITY, GADCHIROLI (M.S.), INDIA

[2020-2021]

A

PROJECT REPORT

ON

DETECTION OF HUMAN BIOLOGICAL EMOTIONS USING OPENCV AND TENSORFLOW

Submitted by

Sagar Ramrao Bharad

Prashish Gautam Borkar

Shreya Bhojraj Gawali

Janhavi Jayant Karande

AnilKumar Ramji Sharma

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

GOVERNMENT COLLEGE OF ENGINEERING,

CHANDRAPUR- 442403

GONDWANA UNIVERSITY, GADCHIROLI (M.S.),

INDIA[2020-2021]

CERTIFICATE

This is to certify that the project entitled "DETECTION OF HUMAN BIOLOGICAL EMOTIONS USING OPENCV AND TENSORFLOW" has been carried out by the team under my guidance in partial fulfillment of the degree of Bachelor of Engineering in Computer Science and Engineering of "Gondwana Digital University", Gadchiroli during the academic year 2020-2021.

Team:

1. **Sagar Ramrao Bharad**
2. **Prashish Gautam Borkar**
3. **Shreya Bhojraj Gawali**
4. **Janhavi Jayant Karande**
5. **AnilKumar Ramji Sharma**

Date:

Place: Chandrapur

Guide
(Prof. D. J. Chaudhari)

Head of the Department
(Prof. D. J. Chaudhari)

Principal
(Dr. P. S. Adwani)

GOVERNMENT COLLEGE OF ENGINEERING, CHANDRAPUR- 442 403

PROJECT APPROVAL SHEET

Following team has done the appropriate work related to the "**DETECTION OF HUMAN BIOLOGICAL EMOTIONS USING OPENCV AND TENSORFLOW**" in partial fulfillment for the award of Bachelor of Engineering in Computer Science & Engineering of "Gondwana Digital University",Chandrapur and is being submitted to Government College of Engineering, Chandrapur.

Team:

1. **Sagar Ramrao Bharad**
2. **Prashish Gautam Borkar (Team Leader)**
3. **Shreya Bhojraj Gawali**
4. **Janhavi Jayant Karande**
5. **AnilKumar Ramji Sharma**

Project Guide:
(Prof. D. J. Chaudhari)

External Examiner:

Date:
Place: Government College of Engineering, Chandrapur.

ACKNOWLEDGEMENT

With immense pleasure and sense of satisfaction we present the project as part of the curriculum of the Degree in Computer Science & Engineering. We wish to thank all the people who gave us an unending support right from the idea was conceived. We are very fortunate to be guided by the vast and resourceful experience in their respective field of work.

We express our sincere gratitude to our guide Prof. D. J. Chaudhari for their timely guidance, extensive support and suggestion. We are also thankful for their sincere help in providing facilities to the department.

Our sincere thanks to our HOD Prof. D. J. Chaudhari the teaching and non-teaching staff for the help extended by them for making this project a success.

Thank you..!

Team:

1. **Sagar Ramrao Bharad**
2. **Prashish Gautam Borkar (Team Leader)**
3. **Shreya Bhojraj Gawali**
4. **Janhavi Jayant Karande**
5. **AnilKumar Ramji Sharma**

PAGE INDEX

FIGURE INDEX

ABSTRACT

Human emotion are natural expressions that people tend to make naturally, instead of any conscious effort that is accompanied by the reflexing of facial muscles. Humans can quickly and even subconsciously assess a multitude of indicators such as word choice, voice inflection, and body language to discern the sentiment of others. OpenCV and TensorFlow are used to make the system which contain face recognition, emotion detection, drowsiness detection and ID card detection . This system can be used as a second layer of security where along with the face emotion is also detection. This can also be useful to verify that the person in front of the camera is not just a 2-dimensional representation. The system will detect the face from the given image, after processing it and on the basis of the expression, the emotion will be recognized.

1.INTRODUCTION

1.1 Overview:

The user will initiate the process by executing a self executable file. Then the user will interact with the main user interface where a video stream from the integrated webcam will be used to predict emotion.

The user can then navigate to local storage to change the input stream to picture or recorded media using the ‘Browse files’ button. The user can return back to the webcam source by clicking ‘Switch to camera’ or can close the process by using the red cross arrow (close) button or minimize the window.

1.2 Problem Statement:

In mass processions individuals’ can show repulsive and violent behavior, such individuals show pre-emotions such as anger, or despair. Which when identified using emotion recognition can avert disastrous events. Also using such emotions detection public places can be equipped with solutions to suppress these behaviors.

Nowadays, more and more intelligent systems are using emotion recognition models to improve their interaction with humans. This is important, as the system can adopt their responses and behavioral patterns according to the human emotion and make it more natural.

1.3 Objective:

The objective of the project is to be an affordable and efficient product that can detect emotions based on input images sources.

The objective of this project is to develop a system which can analyze the image and predict the expression of the person.

An important domain where we see the importance of emotion detection is for business promotions. Most of the businesses thrive on customer responses to their product and offers.

The main purpose of this system is to classify human emotion based on their facial appearance. Here are the different types of emotion: anger, happiness, surpries, sadness, fear, disgust, peace, sorrow, courage.

1.4 Methodology

The facial expression recognition system is implemented using convolutional neural networks. The block diagram of the system is shown in following figures:

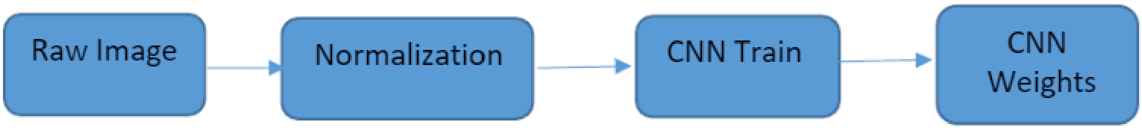


Fig 1.1 Training Phase Figure

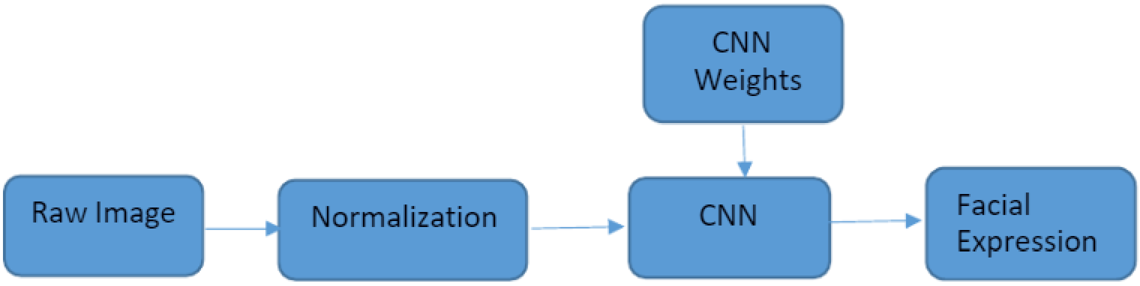


Fig 1.2 Testing Phase

During training, the system receives training data comprising grayscale images of faces with their respective expression label and learns a set of weights for the network. The training step took as input an image with a face. Thereafter, an intensity normalization is applied to the image. The normalized 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, validation dataset is used to choose the final best set of weights out of a set of trainings 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.

Dataset

The dataset from a Kaggle Facial Expression Recognition Challenge (FER2013) is used for the training and testing. It comprises pre-cropped, 48-by-48-pixel grayscale images of faces each labeled with one of the 7 emotion classes: anger, disgust, fear, happiness, sadness, surprise, and neutral. Dataset has a training set of 35887 facial images with facial expression labels.. The dataset has class imbalance issues, since some classes have a large number of examples while some have few. The dataset is balanced using oversampling, by increasing numbers in minority classes. The balanced dataset contains 40263 images, from which 29263 images are used for training, 6000 images are used for testing, and 5000 images are used for validation.

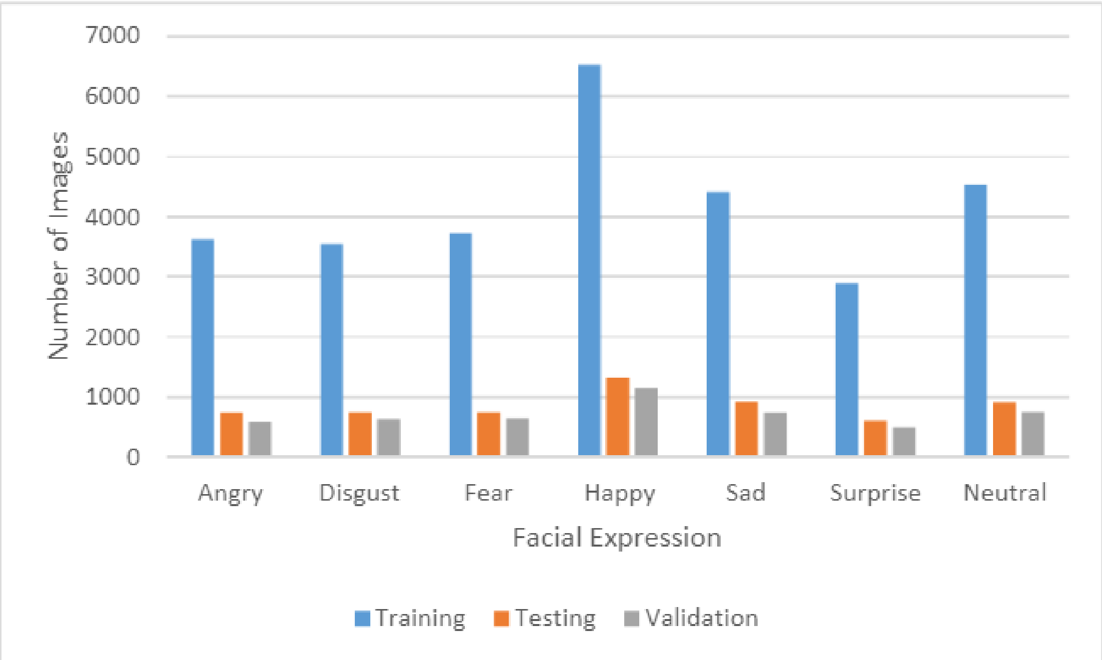


Fig 1.3 Training, Testing and Validation Data distribution

Architecture of CNN

A typical architecture of a convolutional neural network contains an input layer, some convolutional layers, some fully-connected layers, and an output layer. CNN is designed with some modification on LeNet Architecture. It has 6 layers without considering input and output. The architecture of the Convolution Neural Network used in the project is shown in the following figure.

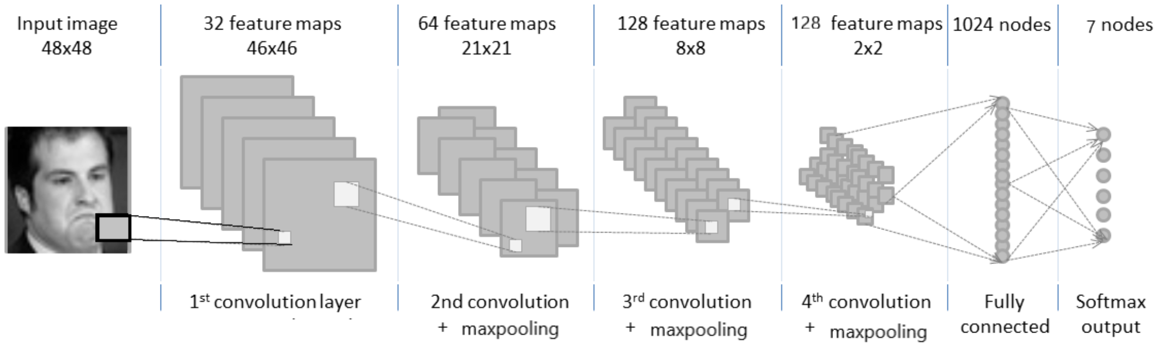


Fig 1.4 Convolution Layer

1. Input Layer**:** The input layer has predetermined, fixed dimensions, so the image must be pre-processed before it can be fed into the layer. We used OpenCV, a computer vision library, for face detection in the image. The haarcascade_frontalface_default.xml in OpenCV contains pre-trained filters and uses Adaboost to quickly find and crop the face. The cropped face is then converted into grayscale using cv2.cvtColor and resized to 48-by-48 pixels with cv2.resize. This step greatly reduces the dimensions compared to the original RGB format with three color dimensions (3, 48, 48). The pipeline ensures every image can be fed into the input layer as a (1, 48, 48) numpy array.

2. Convolution and pooling: The numpy array gets passed into the Convolution2D layer where we specify the number of filters as one of the hyperparameters. The set of filters(aka. kernel) are unique with randomly generated weights. Each filter, (3, 3) receptive field, slides across the original image with shared weights to create a feature map.Convolution generates

feature maps that represent how pixel values are enhanced, for example, edge and pattern detection. A feature map is created by applying filter 1 across the entire image. Other filters are applied one after another creating a set of feature maps.

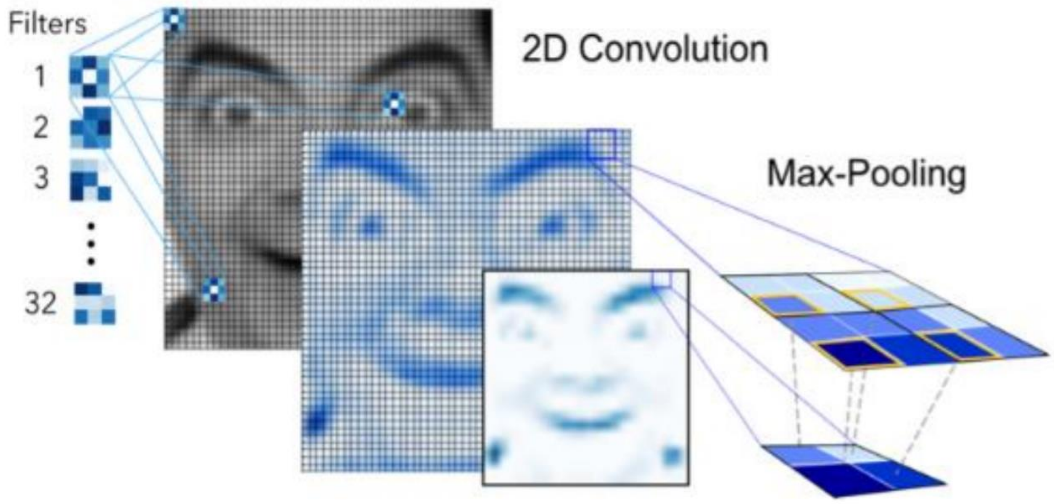


Fig 1.5 Max Pooling

Pooling is a dimension reduction technique usually applied after one or several convolutional layers. It is an important step when building CNNs as adding more convolutional layers can greatly affect computational time. We used a popular pooling method called MaxPooling2D that uses (2, 2) windows across the feature map only keeping the maximum pixel value. The pooled pixels form an image with dimensions reduced by 4.

3. Fully Connected Layer:This layer is inspired by the way neurons transmit signals through the brain. It takes a large number of input features and transforms features through layers connected with trainable weights. Two hidden layers of size 1024 and 7 units are used in a fully-connected layer. The weights of these layers are trained by forward propagation of training data then backward propagation of its errors.

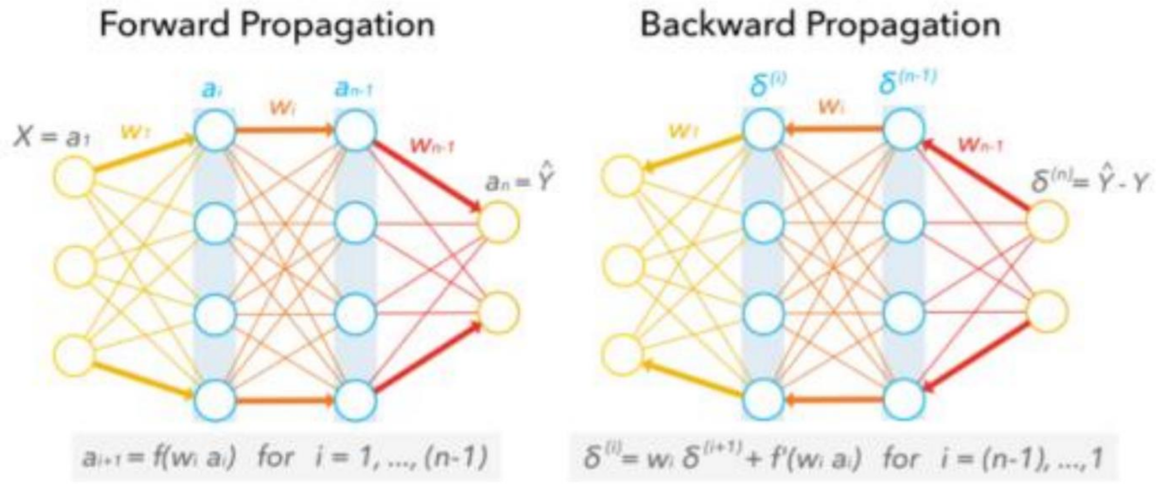


Fig 1.6 Type of Propagation

Back propagation starts from evaluating the difference between prediction and true value, and back calculates the weight adjustment needed to every layer before. We can control the training speed and the complexity of the architecture by tuning the hyper-parameters, such as learning rate and network density. Hyper-parameters for this layer include learning rate, momentum, regularization parameter, and decay. The output from the second pooling layer is of size Nx20x9x9 and input of the first hidden layer of fully-connected layer is of size Nx1024. Output from the first hidden layer is fed to the second hidden layer. Second hidden layer is of size Nx7 and its output is fed to an output layer of size equal to the number of facial expression classes.

4. Output Layer: Output from the second hidden layer is connected to the output layer having seven distinct classes. Using Softmax activation function, output is obtained using the probabilities for each of the seven classes. The class with the highest probability is the predicted class.

2. LITERATURE SURVEY

This chapter provides an introduction to the areas of research. The scope has been clearly explained and the technology used to obtain this has been mentioned in this chapter.

2.1 Existing System

Automation has always been a centre of focus in this 21st century. Though it cuts jobs but at the same time it provides a better and standard way of living in the society. In the modern era, traditional businesses are on the verge of almost leaving the race since new market leaders have come up with ideas of intelligent businesses. A business that consists of highly advanced systems for providing and taking decisions for the organizations. What would be the case if we would be able to implement a real-world application that can collect feedback from customers directly through their expression. What if we would be able to develop a system that can detect any antisocial activities before it happens through the mood of a mob. Human facial Expression recognition has applications in various fields. From customer feedback to criminal confession and sometimes finding the anti-social elements in the crowd.

2.2 Proposed System

This proposed system is capable of performing automatic facial expression or emotion recognition of seven universal emotions, considered to be universal across cultures: disgust, anger, fear, happiness, sadness, neutral and surprise. Such a system would analyze the image of a face and produce a calculated prediction of the expression. The approach integrates a module for automatic face detection, by generating a neural network using a training data set. Methods to get more accuracy : Deep Convolutional neural network , where CNN and deep neural network architectures are customized, trained and subjected to various classification tasks. Input image data will be provided to the network and Which returns values of the output layer the performance matrix of the final model will generate and max value from matrix is calculated; this value represents the current emotion of the provided input.

3. SYSTEM ANALYSIS REQUIREMENTS

3.1 System Analysis: Emotion recognition based on the observation of contours, namely facial features displayed in still pictures. Facial features used are obtained by edge detection and focusing on specific facial regions of eyes and the mouth. Therefore, classification and emotion recognition is performed exclusively through those facial regions.

3.1.1 Relevance of Programming Language:

Python will be used for development of systems because of its many advantages over languages. Python offers concise and readable code. While complex algorithms and versatile workflows stand behind machine learning and AI.Python’s simplicity allows developers to write reliable systems. Developers get to put all their effort into solving an ML problem instead of focusing on the technical nuances of the language.

Also many prototyping framework and gpu acceleration libraries are implemented in python.

Requirement analysis:

The system to develop will require a prototyping framework to construct the model and train the model. It will also require Gpu based acceleration for accurate and fast training.

The system uses Tensorflow for rapid prototyping and Cuda tensor cores of Nvidia Ampere architecture.

Scope and Boundary:

The scope of the system includes development of an accurate model to detect emotions and display them on the interface of the system.

It aims to be compatible with any systems further integrated into it and work flawlessly

The boundary of the system arsise where it can only detect 7 emotions trained into the system. It can not detect micro emotions displayed by subject
Also the system can only detect faces which are completely in frame and not cropped frames. It also detects only emotion from face data and not other physiological / physical parameters.

User Objective:

The user expects the system to accurately predict emotions and also display them accurately. The user also expects that the system works well within minimal resources and out of the box.

Inputs and Outputs:

Inputs: The system would use integrated webcam and/or recorded media on the host system .
Output: The system will give predicted emotion on the user interface with a bounding box.

3.3 Functional Requirements:

The functional requirement of the system requires a well sophisticated emotion detection algorithm that will detect emotions accurately and within low system resources.

3.3.1 Software specification:

Front-End:	Python, C++
Development Environment:	JetBrains Pycharm 2020.2.1
Back-End:	OpenCV 2.4.13.7, TensorFlow Core 2.3.0
UI Interfacing:	PyQt, Tk
VCS (Version Control System):	Local – Git 2.27 Cloud- Microsoft Github

3.3.2 Hardware specification:

Processor:	Intel i5 1.2 GHz or higher AMD Ryzen 3 2.0 GHz or higher
Ram:	8 GB DDR4 2400 MHz or higher
Host OS:	Windows8 SP1 or higher with webcam support

3.4 Non-Functional requirements:

The system will require to be well adaptable to other technologies it may be integrated with. It also requires that it will be reliable and work flawlessly with them.

It is non functional because it is an expected property of a good software but not a mandatory one.

4. SOFTWARE APPROACH

Software approach is the sequence of steps involved in the development process along with the software used in each step.

4.1 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

4.2 Computer Vision:

Computer vision is a process by which we can understand the images and videos, how they are stored and how we can manipulate and retrieve data from them. Computer Vision is the base or mostly used for Artificial Intelligence. Computer-Vision is playing a major role in self-driving cars, robotics as well as in photo correction apps.

4.2.1 OpenCV

OpenCV is the huge open-source library for computer vision, machine learning, and image processing and now it plays a major role in real-time operation which is very important in today’s systems. By using it, one can process images and videos to identify objects, faces, or even the handwriting of a human. When integrated with various libraries, such as Numpuy, python is capable of processing the OpenCV array structure for analysis. To Identify image patterns and its various features we use vector space and perform mathematical operations on these features.

4.3 TensorFlow

TensorFlow is an open source library for fast numerical computing.
It was created and is maintained by Google and released under the Apache 2.0 open source license. The API is nominally for the Python programming language, although there is access to the underlying C++ API. Unlike other numerical libraries intended for use in Deep Learning like Theano, TensorFlow was designed for use both in research and development and in production systems It can run on single CPU systems, GPUs as well as mobile devices and large scale distributed systems of hundreds of machines.

4.4 Matplotlib

Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open source alternative to MATLAB. Developers can also use matplotlib's APIs (Application Programming Interfaces) to embed plots in GUI applications.

4.5 Keras

Keras is a powerful and easy-to-use free open source Python library for developing and evaluating *deep learning models*.

It wraps the efficient numerical computation libraries Theano and TensorFlow and allows you to define and train neural network models in just a few lines of code.

5. SYSTEM DESIGN

Systems design is the process of defining the architecture, product design, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

5.1 General Design Architecture

The general design architecture defines a basic outline of the system.It consists of 3 stages where middle stages consist of three substages and the last parent stage consists of a combination of two stages.

The system first captures a frame which is subject source for identification, Then it is fed for feature extraction.The subject data is fed to the face classifier which identifies probabilistic location of a face , face identification.

The region is then splitted into face data and non face data.This data frame is given to next stage where a trained model classify the face to give approximate Values .

These values are then compared to known accurate values and classified into an emotional state.

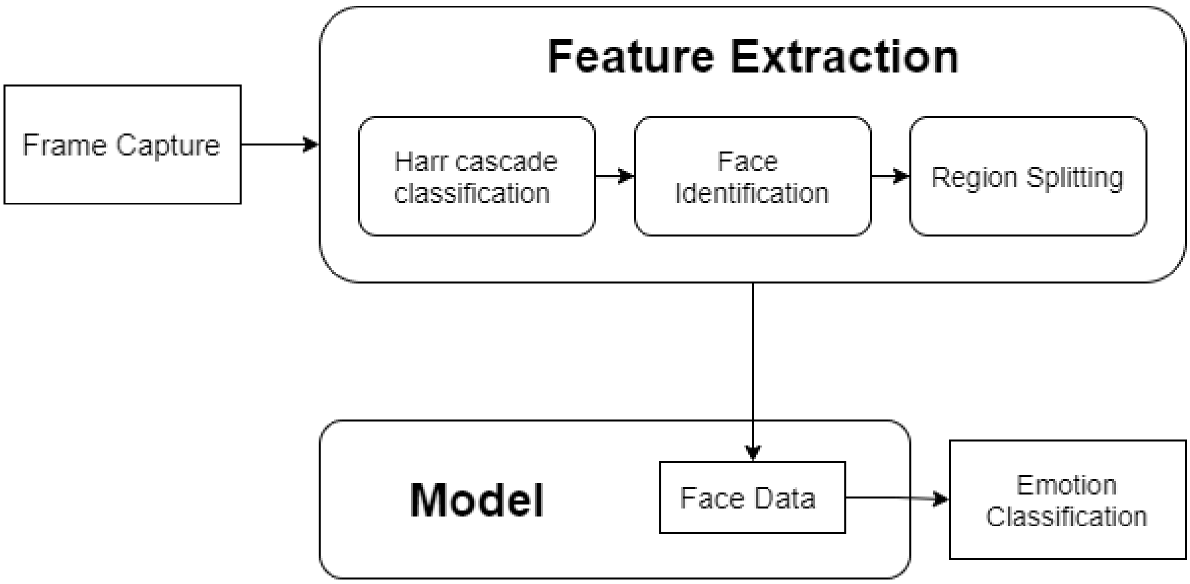


Fig 5.1 General Design Architecture

5.1.1 USE Case:

A use case diagram is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships. It models the tasks, services, and functions required by a system/subsystem of an application. It depicts the high-level functionality of a system and also tells how the user handles a system.

There is one actor in the system i.e. User. The user can interact with the system through the main system interface. He will execute the main program which will show emotions of the subject within the webcam's frame. The user can import an external video source using 'Browse files' and switch back to camera using 'Switch back to camera' buttons. Users could close the system using the main interface interaction buttons.

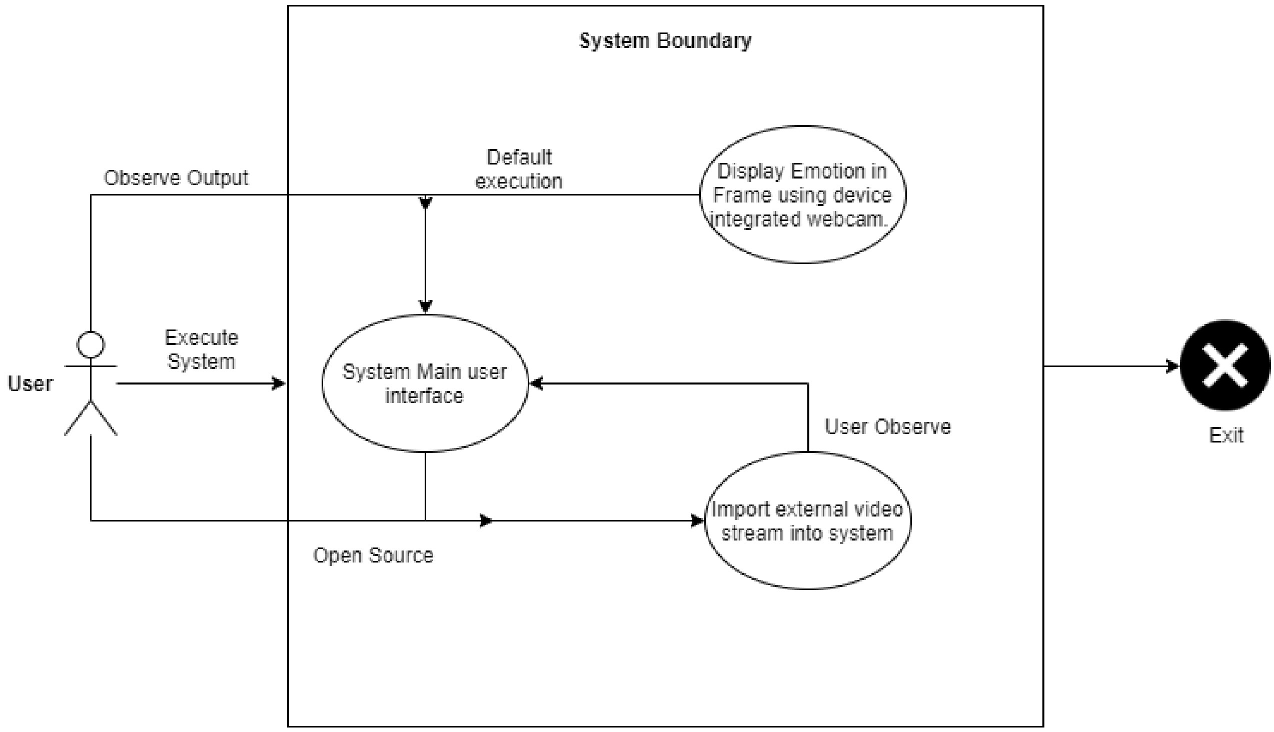


Fig 5.2 Use- Case Diagram

5.2.2 State Machine Diagram:

The state machine diagram is also called the Statechart or State Transition diagram, which shows the order of states undergone by an object within the system.

On Initialization the system will load Runtime Libraries and System support libraries. The system then will acquire a video stream from the integrated webcam. Then it will load the main user interface and will be in default State. Now the system will be in the loop of predicting emotion and displaying on the interface. The system can jump to another state where the video input stream is changed to images or recorded video on user interaction. The user will navigate to a source file using the 'Browse File' button. Now from this state the system can jump to a webcam using 'Switch to Camera' on user interaction.

On the main interface the system can go to an exciting state , by interacting with Close, Minimize window buttons.

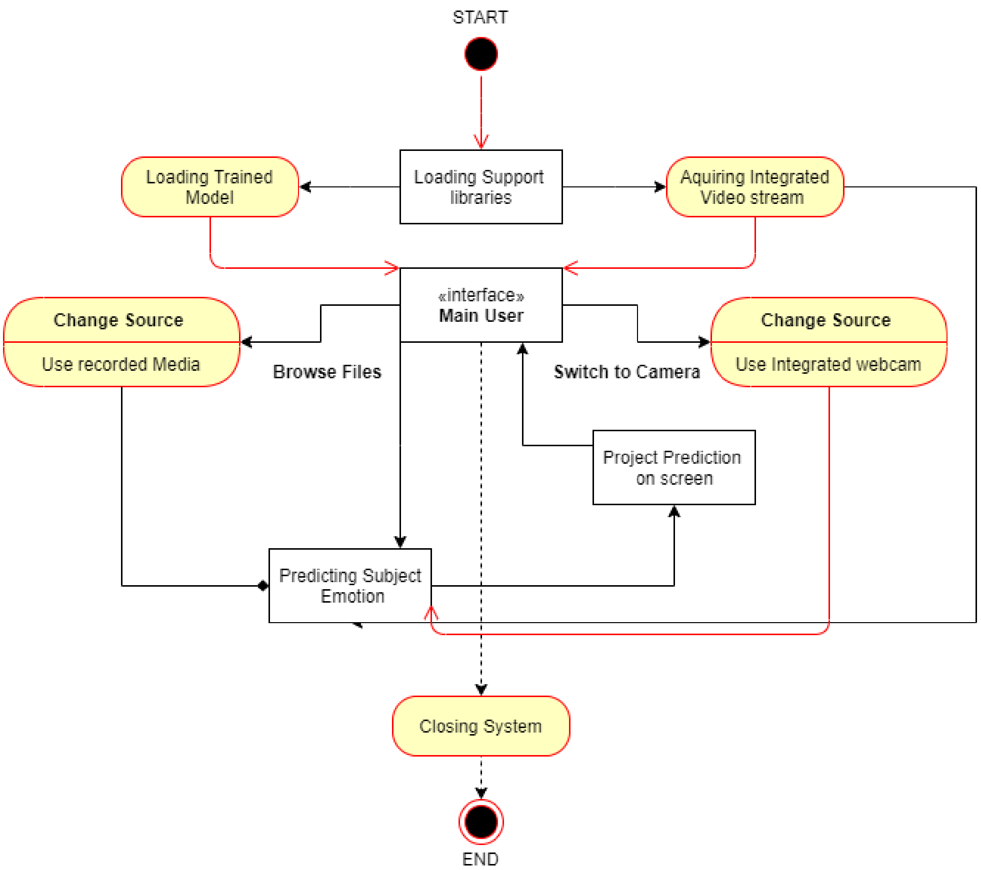


Fig 5.3 State Machine Diagram

5.3.3 Data Flow Diagram:

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

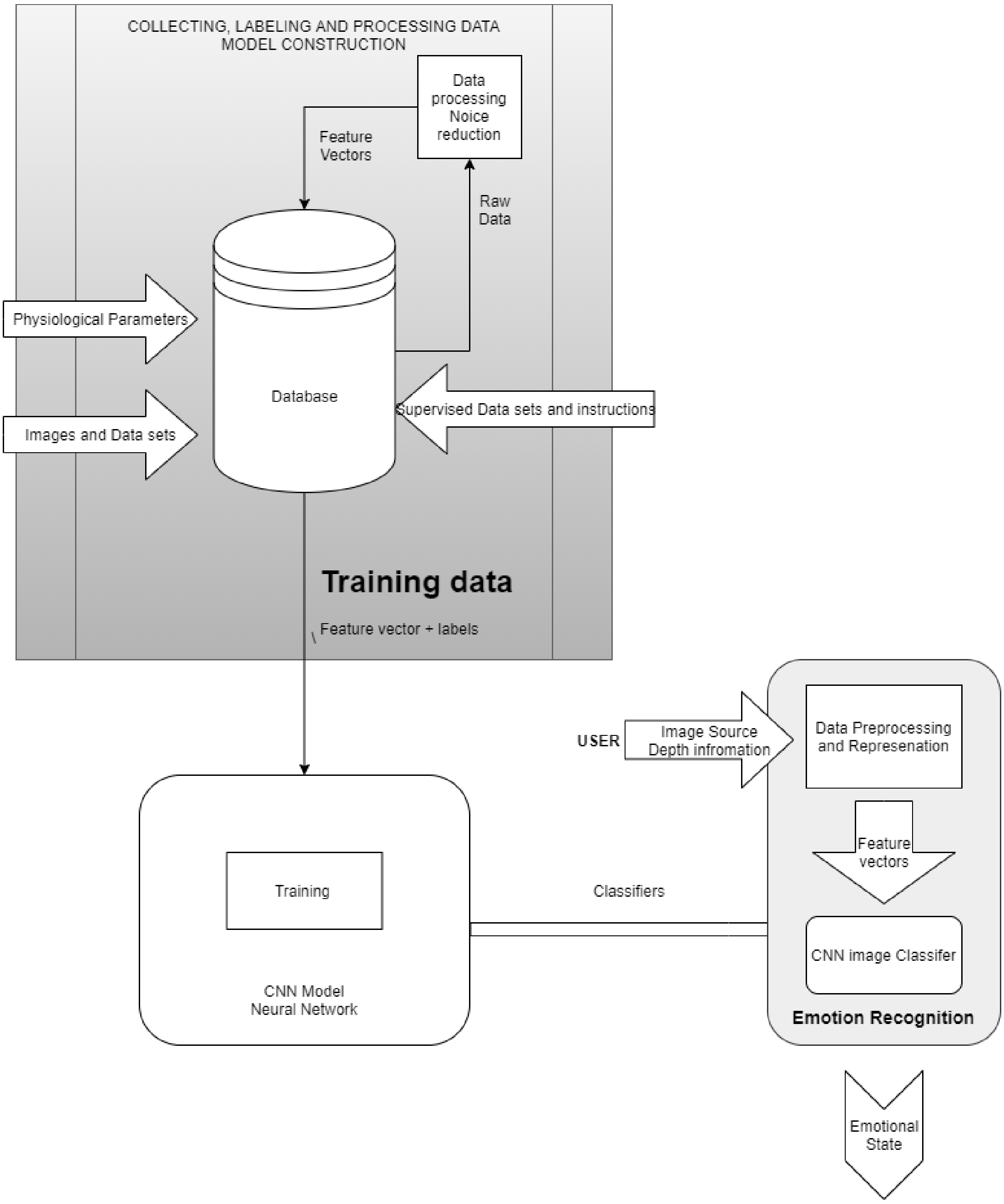


Fig 5.4 Data Flow Diagram

6. SYSTEM IMPLEMENTATION

The basic goal in system implementation is to specify the logic for the different modules that have been specified during system design. Specifying the logic will require developing an algorithm that will implement the given specification. This chapter gives the implementation details of the entire system.

6.1 Development Phase

6.1.1 Preprocessing

In preprocessing the data set acquired to train the model is extracted cleaned and converted into 48*48 pixel data images.

6.1.2 Model Generation and Training:

The model is created using Tensorflow API which will be trained on Preprocessing Data and then used to predict emotion.

6.1.3 Emotion Prediction:

The trained model (neural network) is provided image data from video source and known patterns are inferred from it.

6.2 Deployment Phase

6.1.1 Display

The apps can reliably recognise the emotion of images uploaded from the local storage (of a device) or taken immediately from the camera of a Laptop/Webcam. The app provides a simple and intuitive interface.

First by default emotion detection by camera is used.The user is asked to either upload an image from their device. Once the user selects/uploads a picture, they are taken to the processing section where they are shown the image they selected and asked to press the "Open" button to process that image and get the emotion predictions for it. Then User can switch back to the camera by pressing the “Switch to Camera” button.

6.1.2 Emotions

This is the main module the system starts from here. It load the model weights into memory and call the display module to predict emotion.

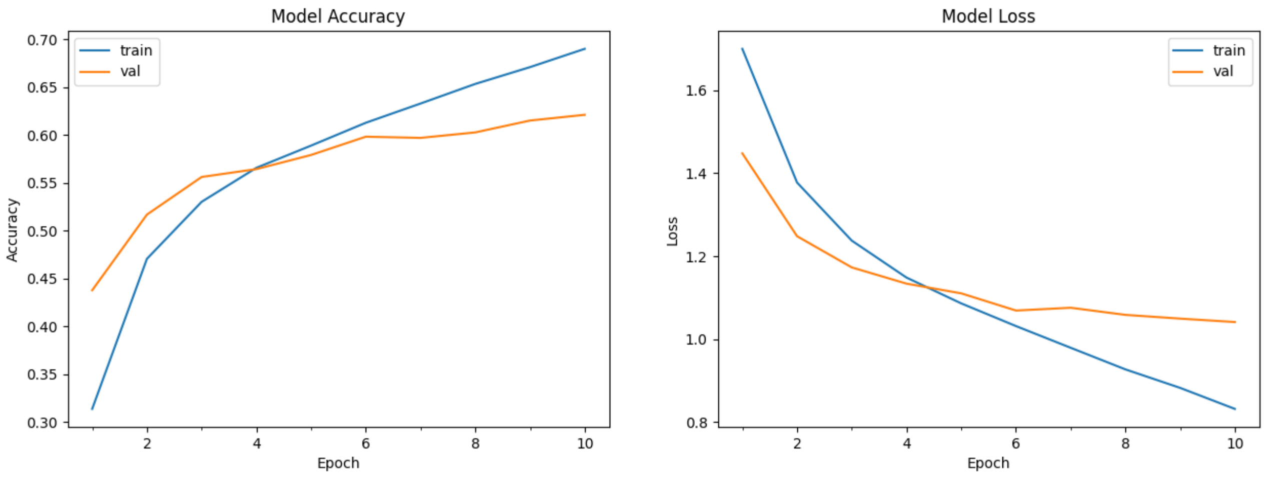


Fig 6.1 Model Accuracy and Loss Curves

7. SYSTEM TESTING

System Testing is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements.

7.1 Introduction

In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested.

7.1.2 Unit Testing

The system was subjected to various procedures to perform unit testing. There are the modules in the system each were separately tested for quality check and to be bug/glitch free.

In the training module, the model was verified using multiple input parameters and changing subject data to be detected. The training module was verified using tensorflow model summary. It was done to validate the correspondence of proposed and validated model

In the second module of ‘Display’ or emotion prediction. ‘nvidia-smi’ were used to determine if the process is using gpu acceleration during runtime. It was also tested whether the module detects face data in low resolution sources.

The system consisted of another module called preprocessing where the dataset was to be converted into training data. This module was just for the development phase and will not be included in the final deployment build.

In Display another function such as User Interface was tested for quality and performance.

7.1.2 Integration Testing

Integration testing (sometimes called integration and testing, abbreviated I&T) is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements.

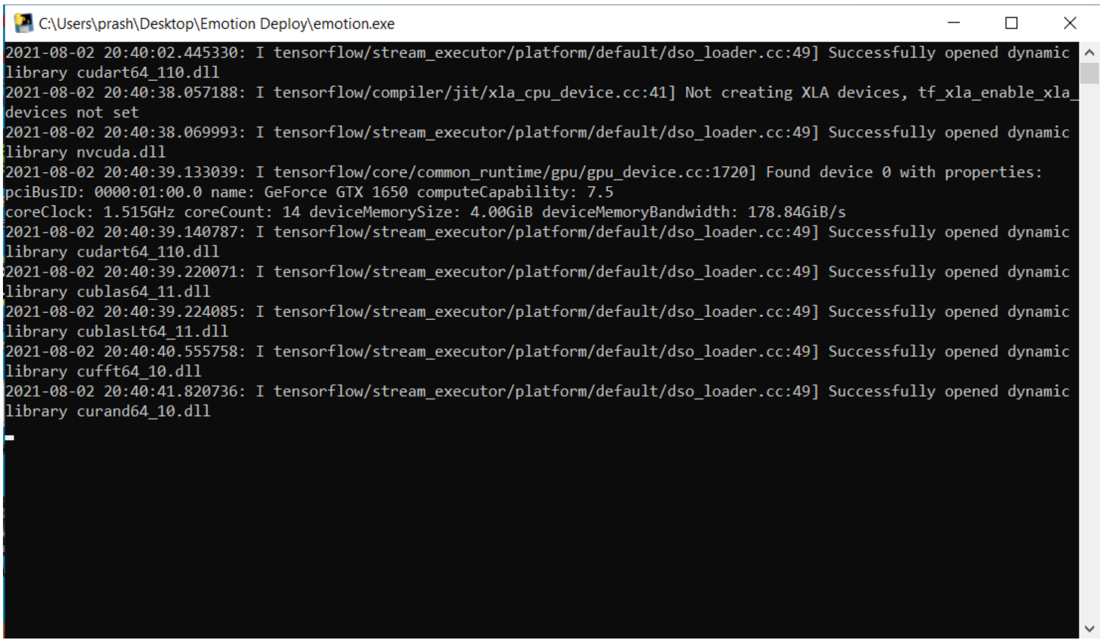
In integration testing all the modules were tested for inter compliance and intercompatibility. The system was validated for, that no module conflicts with the execution of another and no one overwrites the operation of another. The system tested whether the changing of source data does not compromise the process of emotion detection. It was also checked that the system can be properly executed and also killed. The system can properly jump between integrated webcam and other sources.

8 DEPLOYMENT

The complete system is now converted into a self-sufficient package. The working/ result of the system is as:

8.1 Result /user Interface:

On execution of the file the system loads runtime libraries and models, which can be seen on the shell command window(cmd).



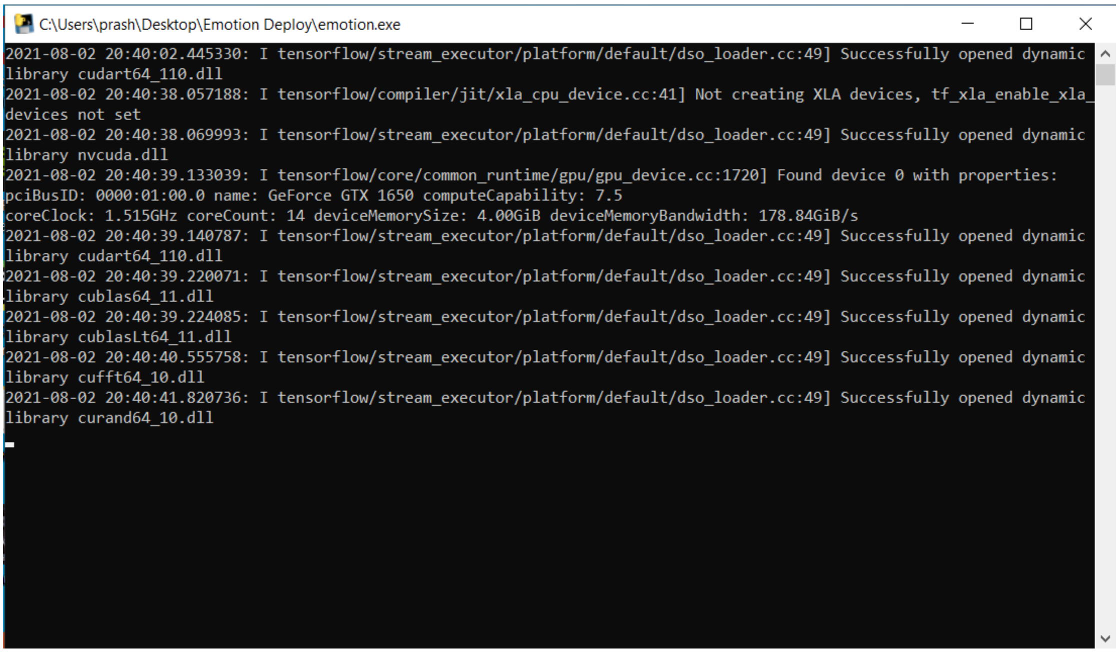


Fig 8.1 Loading Library

A. Main User Interface

The user can observe the screen for emotions detected by the system through the integrated webcam.

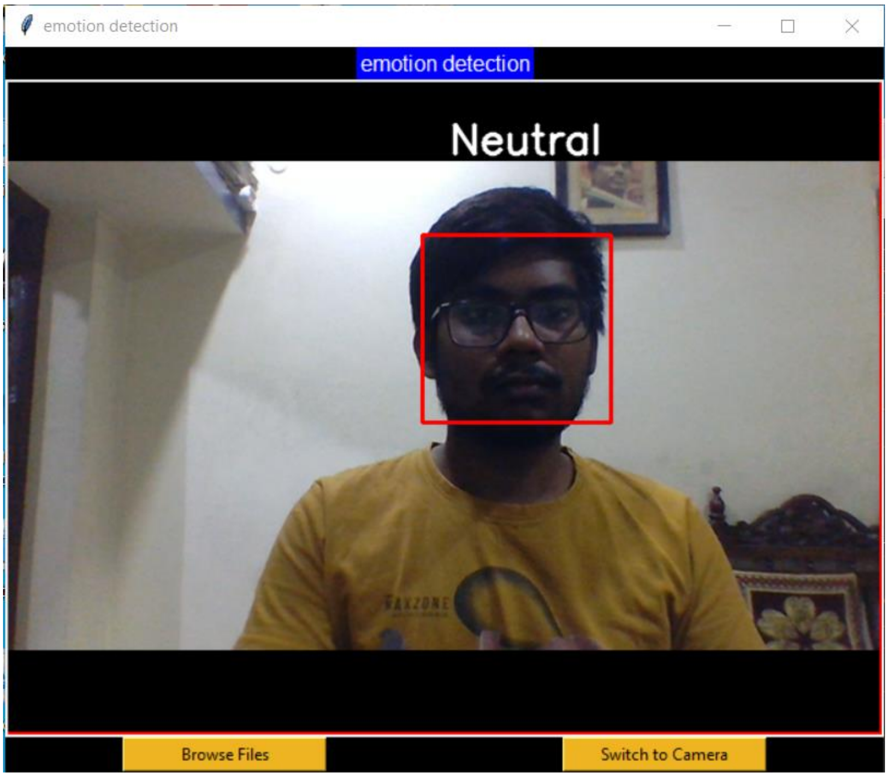


Fig 8.2 Main user Interface

B. Browse Files

The user can change the input stream from integrated webcam to another source using the Browse files button.

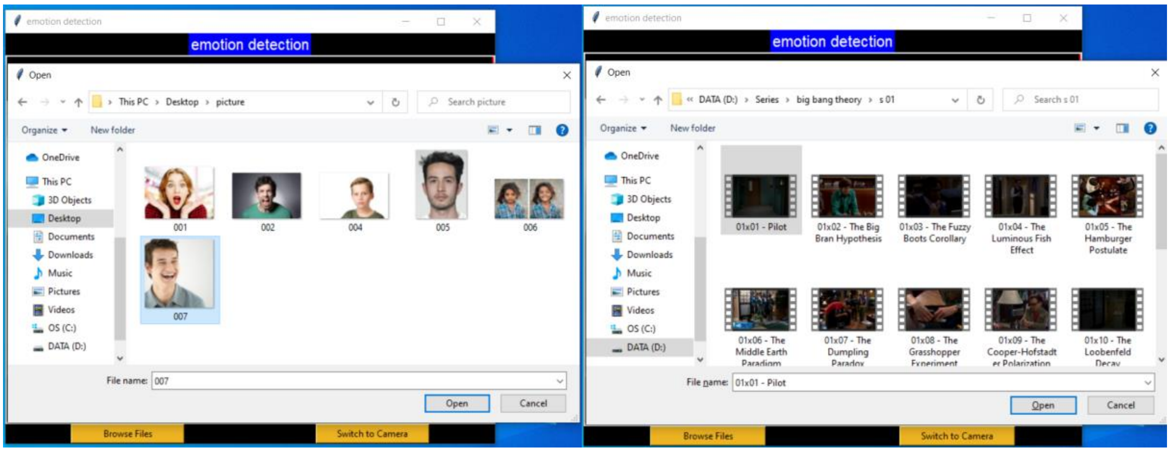


Fig 8.3 Browse Files

a. Photo/Image:

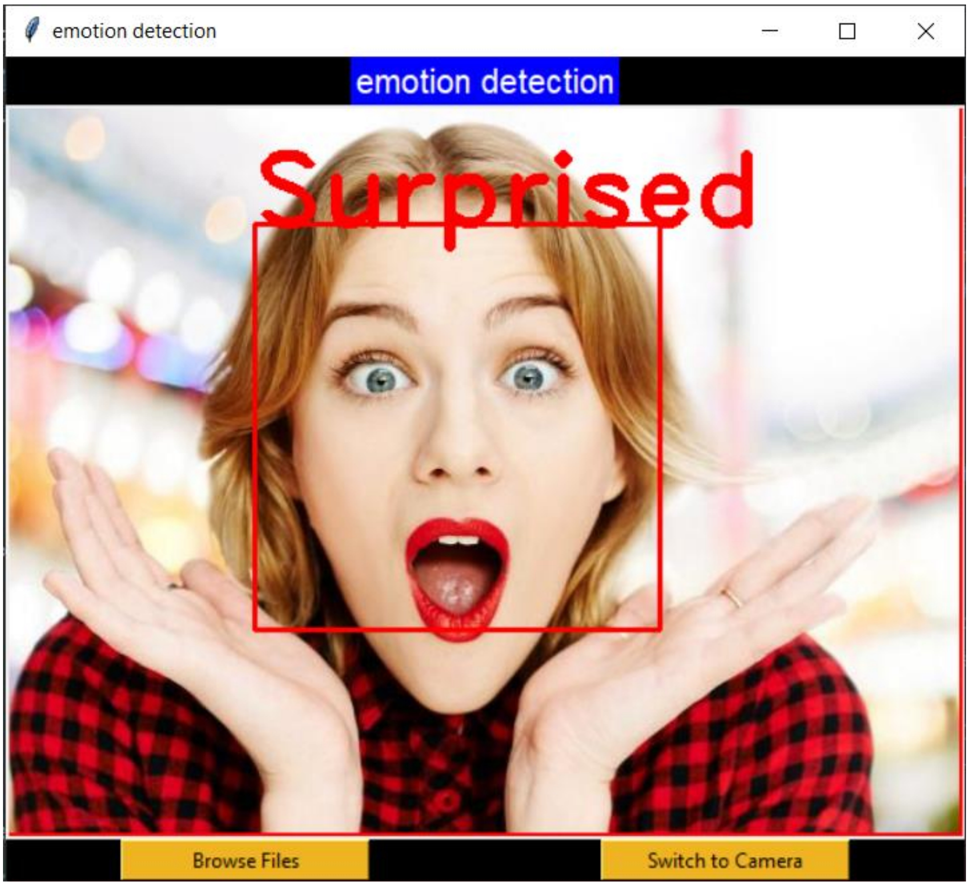


Fig 8.4 Image Output

b. Recorded video

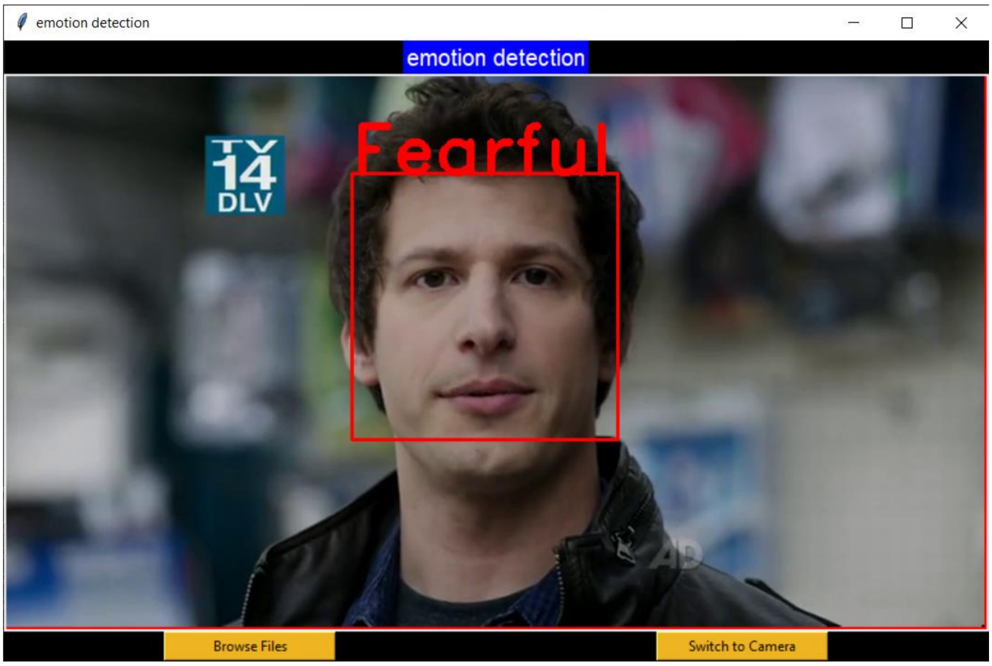


Fig 8.5 Video Output

C. Switch to Camera

The user can switch back to camera using the switch to camera button.

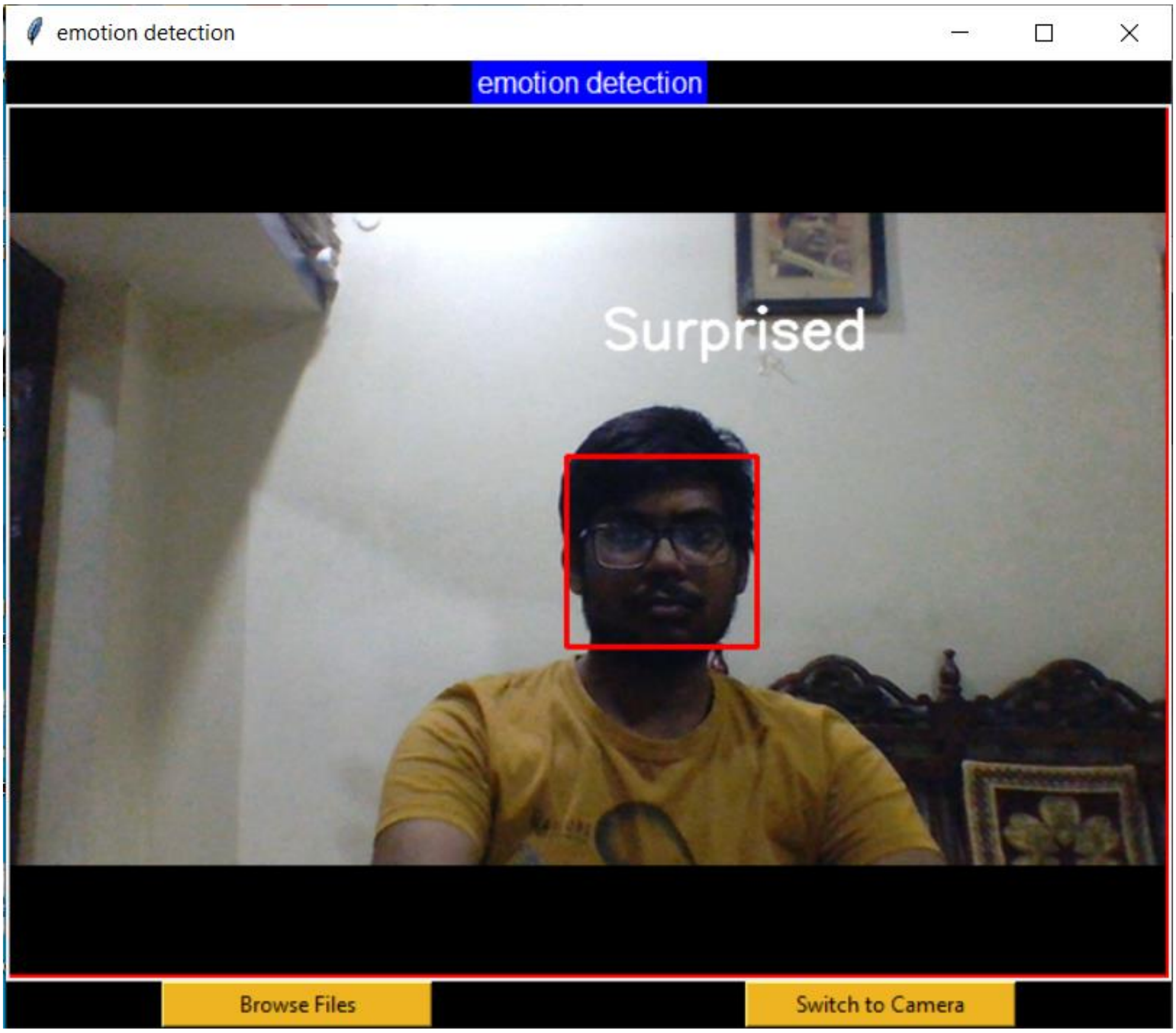


Fig 8.5 Switch to camera

9 CONCLUSION AND FUTURE WORK

9.1 Conclusion

The system developed effectively detects emotion exhibited by subjects present in the system frame. Hence we have successfully developed a Neural Network based system which can be used in User-Interaction system, Surveillance in crowded places, and much more business processes. This software meets the objective and goals proposed earlier in the report. The user will find it efficient to use with lesser complexities and better user interfaces.

9.2 Future Work

The System is effective in identification of emotion.In view of future enhancement , the system can be made more accurate by using different classification algorithms.The following efforts will be made,

- Improving system accuracy.
- Integrating systems with business processes and/or humanoid robotic systems.
- Integration of Body language analysis for better emotion recognition.

It is important to note that there is no specific formula to build a neural network that would guarantee to work well. Different problems would require different network architecture and a lot of trial and error to produce desirable validation accuracy. In this project we got an accuracy of almost 70% which is not bad at all compared to all the previous models. But we need to improve in specific areas like- number and configuration of convolutional layers number and configuration of dense layers dropout percentage in dense layers 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.

BIBLIOGRAPHY

[1] Wan, S., & Aggarwal, J. K. (2013, April). A scalable metric learning-based voting method for expression recognition . In Automatic Face and Gesture Recognition (FG), 2013 10TH IEEE International Conference and Workshops on (pp. 1-8). IEEE,

[2] [OpenCV Dev Team] “Welcome to opencv documentation!” Dec 31, 2019 , Access Online: September 9, 2020 , Available : https://docs.opencv.org/2.4/index.html

[3] M. Xiaoxi, L. Weisi, H. Dongyan, D. Minghui and H. Li, "Facial emotion recognition," 2017 IEEE 2nd International Conference on Signal and Image Processing (ICSIP), Singapore, 2017, pp. 77-81, doi: 10.1109/SIPROCESS.2017.8124509.

[4] Turabzadeh, S,; Meng, H.; Swash, R.M.; Pleva, M.; Juhar, J. Real-time emotional state detection from facial expression on embedded devices. In Proceedings of the 2017 Seventh International Conference on Innovative Computing Technology (INTECH), Luton, UK, 16-18 August 2017; pp.46-51.

[5] “TensorFlow Core”[Online] Accessed: September 9, 2020 ,Available : https://www.tensorflow.org/guide

[6] C. Jain, K. Sawant, M. Rehman and R. Kumar, "Emotion Detection and Characterization using Facial Features," 2018 3rd International Conference and Workshops on Recent Advances and Innovations in Engineering (ICRAIE), Jaipur, India, 2018, pp. 1-6, doi: 10.1109/ICRAIE.2018.8710406.

APPENDIX - A

USER MANUAL

An Emotion Detection Application

User Manual

Final year project
Computer Science and Engineering

EULA

The MIT License (MIT)
Copyright (c) 2021 Group 6 , Computer Science and Engineering Dept.

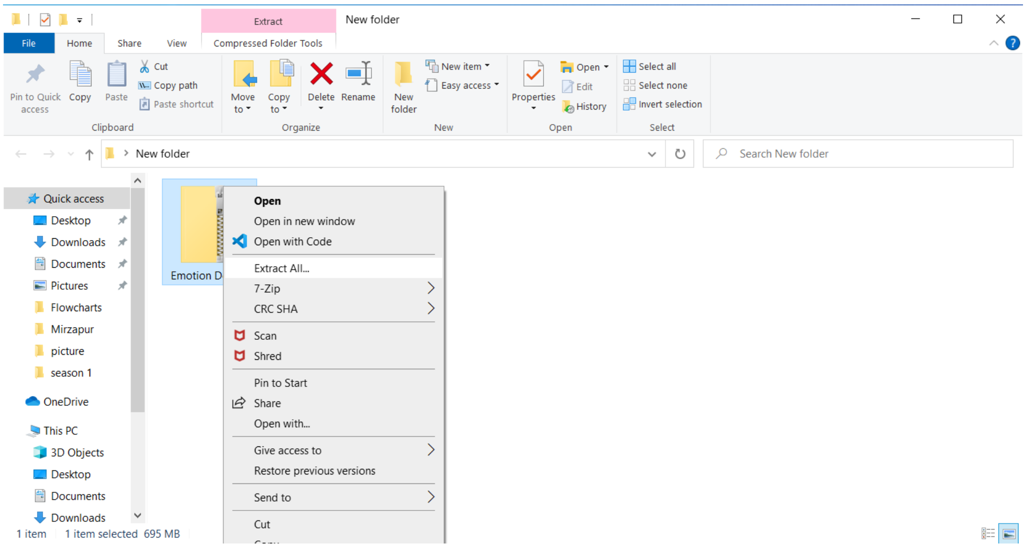
Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

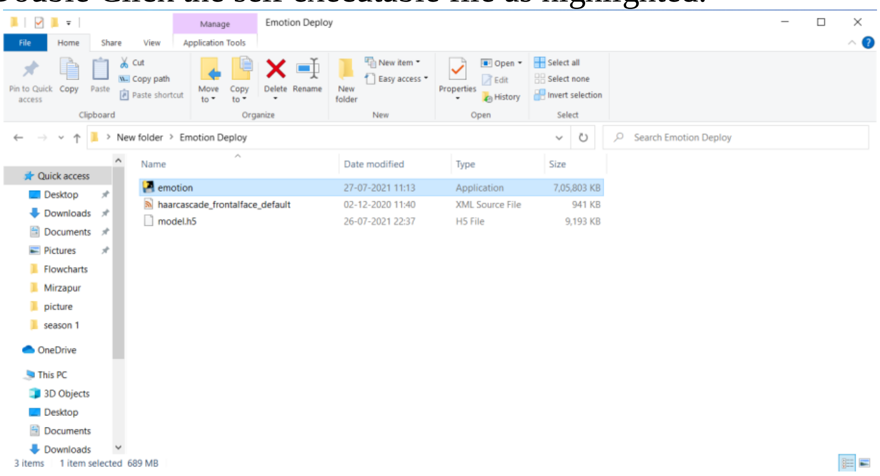
1 SETUP

This software is distributed as a self sufficient zip file. To Install the software extract the zip file to the location where you want to install using any archive software or program.



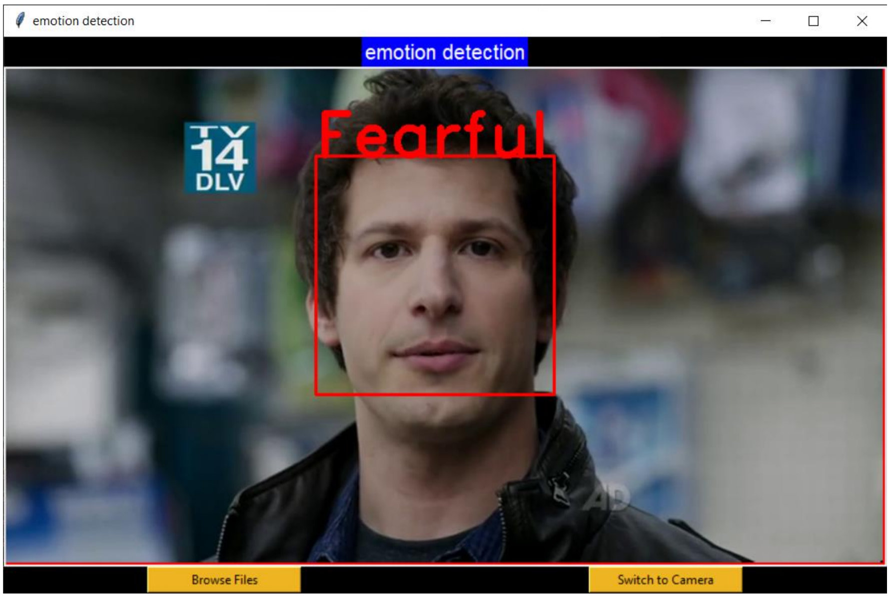
2 USE-AGE

The software can be used with an integrated webcam and/or external source. To Run the system Double Click the self executable file as highlighted.



You will be greeted with the main user interface, Where a webcam feed will be automatically selected.

Now to use External source or switch back to camera uses the buttons labelled on screen.



To close the system click on the close button of the system.