**Emotion Detection Project using Machine learning**

A Project Report Submitted in Partial Fulfilment of The Requirements of Meghnad Saha Institute of technology For the Award Of the degree

**Master of Computer Application**



**Submitted By**

1. Ripam Kundu (14271021001)
2. Arijit Mandel 14271021025)
3. Mritanka Dey (14271021003)

***Under Guidance Of***

Mr. Subhojit Santra

Subject Matter Expert (Angular JS)

**Project Carried Out At**

****

**Ardent Computech Pvt Ltd(An ISO 9001:2008 Certified)**

CF-137, Sector - 1, Salt Lake City, Kolkata - 700 0

***(Note:* *All entries of the proforma of approval should be filled up with appropriate and complete information. Incomplete proforma of approval in any respect will be summarily rejected.)***

1.**Name of the Student With Group :** 1. Ripam Kundu 2. Arijti Mandel 3. Mritanka Dey

**2. Title of the Project :** Emotion Detection Project using Python Machine learning

**3. Name and Address of the Guide** : Mr. Subhojit Santra

Subject Matter Expert & Technical Head(Angular JS)

Ardent Computech Pvt Ltd(An ISO 9001:2008 Certified)

CF-137, Sector - 1, Salt Lake City, Kolkata - 700 064

Ph.D**\***  M.Tech.**\*** B.E\***/**B.Tech.**\***  MCA M.Sc.**\***

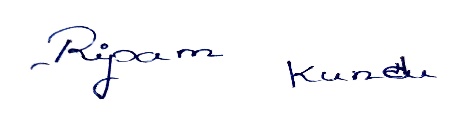
**Y**

**4. Educational Qualification of the Guide**

**5. Working / Teaching experience of the Guide : 2** Year

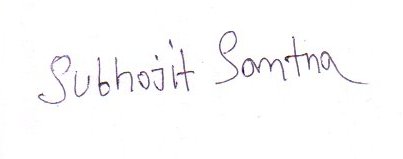
**6. Software used in the Project**

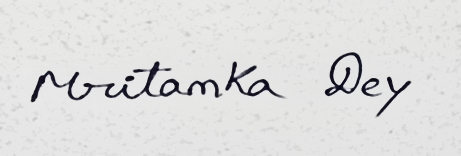
* Python
* Machine Learning
* supervised machine learning algorithm
* Support Vector Machine(SVM)
* MS WORD
* Smart Draw

**Submitted By :**

1.



2.



3.

**Signature of the Student**  **Signature of the Guide**

**Date:** 04/11/2022 **Date :** 04/11/2022

**Name:** Mr. Subhojit Santra

**For Office Use Only**

Not Approved

**Subject Matter Expert**.

Approved

Signature, Designation, Stamp of the Project Proposal Evaluator

**Department of Master of computer Application**

**Date :**

I hereby forward the documentation prepared under my supervision by **Mr. Subhojit Santra**

entitled **Meghnad saha Institute Of Technology** to be accepted as fulfilment of the requirement for the Degree of Master of computer application, **Meghnad saha Institute Of Technology** affiliated to **Maulana Abul Kalam Azad University of Technology** (**MAKAUT**).

|  |  |
| --- | --- |
| **Mr. Subhojit Santra**  **Project Guide**  **Ardent Computech Pvt Ltd** | **Head of the Department**  **Department Of MCA (MSIT)** |

# **Project Responsibility Form**

# **Emotion Detection Project using Python Machine learning**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | GROUP NO. | SL.NO. | | NAME OF MEMBER | RESPONSIBILITY | | 1 | 1 | Ripam Kundu | | Project Leader & PPT, Coding | | | 2 | Arijti Mandel | | Coding & Designing | | | 3 | Mritanka Dey | | System Analysis | |   *Each group member must participate in project development and developing the ideas for the required elements. Individual group members will be responsible for completing tasks which help to finalize the project and the performance. All group members must be assigned a task.* | |  | | --- | |  | |

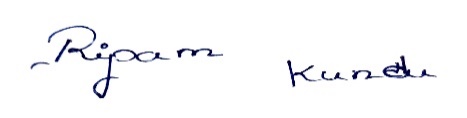
# **Self-Declaration Certificate**

This is to certify that the dissertation/project proposal entitled “**Emotion Detection Project using Python Machine learning**” is done by us, is an authentic work carried out for the partial fulfilment of the requirements for the award of the certificate of **Master of Computer Application** under the guidance of **Mr. Subhojit Santra**. The matter embodied in this project work has not been submitted earlier for award of any certificate to the Best of our knowledge and belief.

**Name of the Student : -**

1. Ripam Kundu
2. Mritanka Dey
3. Arijti Mandel

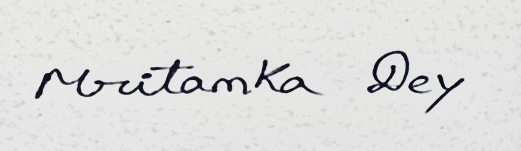
**Signature of the students : -**

****

a.



b.



c.

# Certificate by Guide

This is to certify that this project entitled “**Emotion Detection Project using Python Machine learning**”submitted in partial fulfilment of the certificate of Master of Computer Application through **Ardent Computech Pvt Ltd**, done by the

**Group Members**

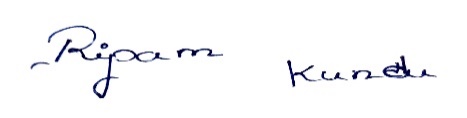
1. Ripam Kundu
2. Mritanka Dey
3. Arijit Mandel

Is an authentic work carried out under my guidance & best of our knowledge and belief,

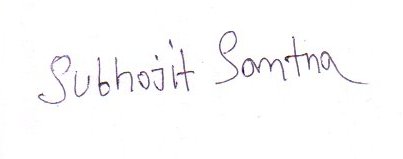
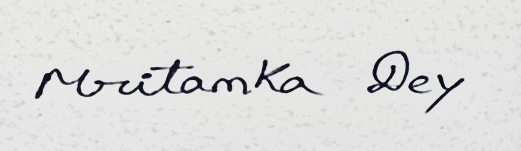
a.

b.

c.

****

1.

2.

3.

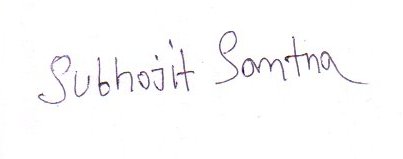
**Signature of the students Signature of the Guide**

**Date:** 04/11/2022  **Date:** 04/11/2022

# **Certificate of Approval**

This is to certify that this proposal of Minor project, entitled **“Emotion Detection Project using Python Machine learning”** is a record of bona-fide work, carried out by 1. Ripam Kundu 2. Mritanka Dey 3. Arijit Mandel under my supervision and guidance through the Ardent Computech Pvt Ltd.In my opinion, the report in its present form is in partial fulfilment of all the requirements, as specified by the ***Ardent Computech Pvt Ltd.*** and as per regulations of the ***Ardent®***. In fact, it has attained the standard, necessary for submission. To the best of my knowledge, the results embodied in this report, are original in nature and worthy of incorporation in the present version of the report for Master of Computer Application.

**Guide / Supervisor**



**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Mr. Subhojit Santra**

Subject Matter Expert & Technical Head (Angular JS)

Ardent Computech Pvt Ltd (An ISO 9001:2008 Certified)

CF-137, Sector - 1, Salt Lake City, Kolkata - 700 064

# **ABSTRACT**

# 

The project has been done under the guidance of **Mr. Subhojit Santra**, from **Ardent Computech Pvt. Ltd, Durgapur**. The project is based on **Machine Learning Support Vector Machine** (SVM) is a supervised machine learning algorithm used for both classification and regression **for Emotion Detection Project using Python Machine learning**.

Emotion detectors are used in many industries, one being the media industry where it is important for the companies to determine the public reaction to their products.

Most smartphones these days have a feature that automatically takes a selfie when we smile. It is amazing how accurately it detects smiles for not only one but multiple faces and captures a selfie immediately. If you have wondered how this is possible, it is actually quite simple. Using some of the libraries like OpenCV, it is possible to build a selfie-capturing application with just a few lines of code.

When we investigate the last one years from a technological point of view, one thing becomes very apparent; technology has infiltrated all parts of human lives in every way imaginable to us, the number of devices in a person's home exceed the number of people itself, the time has come we can no longer ignore the presence of such devices since they hold immense computational power. In this project we want to bring about a very frequently discussed topic of automatically being able to detect human emotions. The project is developed as an android application. The proposed system is being developed using latest technologies such as OpenCV, Haar features, Android Studio, the project aims at face detection as well as emotion detection efficiently, quickly and in minimum amount of steps. The project can detect a wide range of emotions and the unique feature would be that it can detect the person’s level and percentage of different emotions he/she is experiencing.

So, the main goal of the project is to analyze how it’s working and try to make it in a real-life application by using OpenCV-Python

1.

2.

3.

Signatures of Students Signature of Faculty

Date: Date

# **ACKNOWLEDGEMENT**

I take this opportunity to express my deep gratitude and sincerest thank to my project mentor **Mr. Subhojit Santra** for giving most valuable suggestion, helpful guidance and encouragement in the execution of this project work.

It is a great pleasure for me to acknowledge the assistance and participation of a large number of individuals in this attempt. Our project report has been structured under the valued suggestion, support, and guidance of **Mr. Subhojit Santra**. Under his guidance, we have accomplished the challenging task in a very short time.

Finally, we express our sincere thankfulness to our family members for inspiring me all throughout and always encouraging us. I would like to give a special mention to my colleagues. Last but not the least, I am grateful to all the faculty members of **Ardent Computech Pvt. Ltd, Durgapur** for their support.

# **TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO.** | **Name of the Topic** | **Page No.** |
| 1. | Introduction | 11 |
| 2 | Project Category | 12 |
| 3. | Objectives, Scope & Background | 13 |
| 4. | Problem & System | 14 |
| 5. | System Analysis | 15 |
| 6. | Technology Used | 16 |
| 7. | Methodology | 17 - 18 |
| 8. | Image Processing Feature | 19 |
| 9. | Functionality | 20 |
| 10. | Flow Chart | 21 |
| 11. | Block Diagram | 22 |
| 12. | CNN Model Diagram | 23 |
| 13. | Requirement | 24-25 |
| 14. | Feasibility Study | 26 - 27 |
| 15. | Project Planning & Scheduling, Gantt Chart | 28 - 31 |
| 16. | Snapshot | 32 - 33 |
| 17. | Code | 34-37 |
| 18. | Testing | 38 - 43 |
| 19. | Process Overview | 44 |
| 20. | Future Scope | 45 |
| 21. | Conclusion | 46 |
| 22. | Reference | 47 |

**INTRODUCTION**

One of the singularities of human beings that have contributed enormously to the development and growth of mankind is the ability to communicate precisely with rich and powerful spoken (and later in history, also written) languages. Having said that, a significant percentage of what is communicated does not circulate through those languages, but through nonverbal cues. These cues can be in the form of gestures performed, for example, with the hands; or also facial expressions that convey information about what is inside but not necessarily spoken. Given how relevant facial expressions have been to human interactions, it is not surprising that they have been researched for centuries. In it is described how studies on facial expressions were already performed in the Aristotelian era —4th century BC. It can also be read how the foundations of current research were written in the 17th century. Then, in the 19th century Charles Darwin developed one of the important works on facial expression analysis that is more helpful for the science of automatic recognition performed by machines. His work classifies different expressions in groups by similarities, as well as the facial deformations associated to each of these groups. Then, in the 1970s the psychologist Paul Ekman and his associates developed essential work that has been referenced and used by the majority of posterior researchers on the topic of expression recognition.

In the project we have implemented emotion detection through the android  
platform, through the flow of this paper we shall be discussing the methodology and techniques that have been employed to develop this project.

Everyone loves a smiling picture, so we have developed a project which will capture images every time you smile, a **Python Project to automatically detect and capture selfies.**

# **PROJECT CATEGORY:**

Emotion Detection Project using Python Machine learning

# **HARDWARE AND SOFTWARE REQUIREMENT SPECIFICATION :**

#### **Hardware Requirements-**

* **Processor :** I3 8th Gen
* **RAM :** 512 MB and above
* **Printer :** any
* **Monitor :** TFT/VGA/LCD
* **Hard Disk :** 80 GB and above
* **Mouse :** PS2/USB
* **Keyboard :** PS2/USB

#### **Software Requirements-**

* **Operating System :** Windows 8
* **Front End :** Python
* **Back End :** Python Open CV
* MS WORD

# **Platform -**

# Microsoft Windows 8, windows 10, windows 11

1. **Web Browser –**

* Internet Explorer 7
* Google Chrome
* Mozilla Firefox

# **OBJECTIVES**

The main objective of this system is to reduce the consumption of time during maintaining the records of picture in a particular folder. Separate divisions are provided to maintain the picture. In other words -

* Simple database is maintained.
* Easy operations for the operator of the system.
* User interfaces are user friendly and attractive, it takes very less time for the operator to get used to with the system.

# **SCOPE**

This is generic type of web application, suitable for everyone to taken the picture. This system capture more than one type of emotion and store the date in a particular folder.

# **THEORETICAL BACKGROUND OF THE PROJECT:**

#### Front End and Back End-

* **Front End :-** Python
* **Back End :-** Python Open CV
* **Keywords :** Emotion Detection, Face Detection, Haar Features, Human emotion, OpenCV

# **DEFINITION OF PROBLEM**

Today all the management works are done manually by ink and paper, which is very slow and takes more time and efforts. Since the number of students is growing, and management has to handle records of everyone to who taken the picture, that leads to maintaining the records. It is required to Design of a Computerized Emotion Detection System, to speed up and make it easy to use system.

# **EXISTING SYSTEM**

Whenever we implement new system it is developed to remove the shortcomings of an existing system. The computerized has more flexibility over the manual system. Since we are doing a project on “COLLEGE MANAGEMENT”, so firstly we will introduce the existing system, the existing system is based on manual system, which takes lot of time to get performance of the test. It has the following disadvantages:

# **PROPOSED SYSTEM**

In our proposed system we have the provision for adding the details of who taken the picture. Another advantage of the system is that it is very easy to edit and delete when it found unnecessary. By developing the system, we can attain the following facilities-

* Easy to handle and feasible.
* Easy to operate.
* Cost reduction.
* Fast and convenient.

# **SYSTEM ANALYSIS**

# **INTRODUCTION**

After analyzing the requirements of the task to be performed, the next step is to analyze the problem and understand its context. The first activity in the phase is studying the existing system and other is to understand the requirements and domain of the new system. Both the activities are equally important, but the first activity serves as a basis of giving the functional specifications and then successful design of the proposed system.

Understanding the properties and requirements of a new system is more difficult and requires creative thinking and understanding of existing running system is also difficult, improper understanding of present system can lead diversion from solution.

# **WORK FLOW**

This Document plays a vital role in the development life cycle (SDLC) as it describes the complete requirements of the system. It is meant for use by the developers and will be the basic during testing phase. Any changes made to the requirements in the future will have to go through formal change approval process.

Classification and Regration was being chosen because all requirements were known beforehand and the objective of our software development is the computerization/automation of an already existing manual working system.

**Technology Used**

**Python :** Python is a computer programming language often used to build websites and software, automate tasks, and conduct data analysis. Python is a general-purpose language, meaning it can be used to create a variety of different programs and isn't specialized for any specific problems.

**NumPy :** NumPy is an open-source Python library that facilitates efficient numerical operations on large quantities of data. There are a few functions that exist in NumPy that we use on pandas Data Frames. For us, the most important part about NumPy is that pandas are built on top of it. So, NumPy is a dependency of Pandas.

**Pandas :** Pandas is a very popular library for working with data (its goal is to be the most powerful and flexible open-source tool, and in our opinion, it has reached that goal). Data Frames are at the center of pandas. A Data Frame is structured like a table or spreadsheet. The rows and the columns both have indexes, and you can perform operations on rows or columns separately.

A pandas Data Frame can be easily changed and manipulated. Pandas has helpful functions for handling missing data, performing operations on columns and rows, and transforming data. If that wasn’t enough, a lot of SQL functions have counterparts in pandas, such as join, merge, filter by, and group by. With all of these powerful tools, it should come as no surprise that pandas are very popular among data scientists.

**Open CV :** **OpenCV** is a huge open-source library for computer vision, machine learning, and image processing. OpenCV supports a wide variety of programming languages like Python, C++, Java, etc. It can process images and videos to identify objects, faces, or even the handwriting of a human. When it is integrated with various libraries, such as Numpy which is a highly optimized library for numerical operations, then the number of weapons increases in your Arsenal i.e whatever operations one can do in NumPy can be combined with OpenCV.

OpenCV is a cross-platform library using which we can develop real-time **computer vision applications**. It mainly focuses on image processing, video capture, and analysis including features like face detection and object detection. In this tutorial, we explain how you can use OpenCV in your applications.This OpenCV tutorial will help you learn the Image-processing from Basics to Advance, like operations on Images, Videos using a huge set of OpenCV-programs and projects.

**CSV :** A CSV file (Comma Separated Values file) is a type of plain text file that uses specific structuring to arrange tabular data. Because it's a plain text file, it can contain only actual text data—in other words, printable ASCII or Unicode characters. The structure of a CSV file is given away by its name.

**XML :** XML stands for extensible Markup Language. It was designed to store and transport small to medium amounts of data and is widely used for sharing structured information. Python enables you to parse and modify XML documents. In order to parse XML document, you need to have the entire XML document in memory.

**Methodology Used**

* ***Face Detection***

Haar cascade mainly works with face detection. The algorithm requires a lot of training datasets the image which contain faces as positive dataset and images without faces as negative datasets. The next process will be extracting features from these images. For this, haar features shown in Fig.1 are used. Each rectangle represents bright and dark spots of the image. These rectangles represent a single value obtained by subtracting dark spot with the bright spots.

Consider the first-row images shows nearly perfect images of classifier model. The region of eyes is darker than the bridge of nose and eye region is considered as black rectangle and vice-versa.

Each input image will undergo instance classification of determining if the input image contains a face or not. But for sure there will be miscalculations or misclassifications. Furthermore, the images with comparatively low error rate are considered and weighted sum of these week classifiers is computed. Week classifier because they cannot classify the image independently. The classifier dataset is well classified into face and non-face images.

Classifiers are more efficient and time saving while applying then applying each frame one–by-one. This way the classification is more accurate.



Haar classifier for face Detection

* **Feature Extraction**

Feature extraction, in pattern recognition and in image processing, starts from measured dataset and building corresponding values which are intended to be informative and facilitating the learningand carry out steps to better human interpretations. Feature extraction involves reducing thenumber of assets used to represent huge amount of data. While performing complex calculations, large number of variables is likely to cause errors in the end result. Henceforth to overcome this problem we use a general method of combining variables and computing the problem.

One of the important sectors of this application is images processing, in which the application algorithm detects and segregates shapes and points containing features of image or video on grey scale. It plays a vital role in optical object determination. Features are extracted following a step of calculating the pixel code value of a pixel in the metrics of an image. Based on that, the features are  
extracted particularly from those Grayscale image and then the output is displayed. In software packages, Predefined interfaces provide for feature extraction and dimension reduction. The algorithms are available publicly.

* **Emotion Detection**

Emotions and expressions are calculated using valance points which will be placed around the face. The distance and angle between these points calculates the category of emotion and gives the percentage (ranges from 0 to 100) of facial expression on the user’s face. For instance, if the input is joying the total percentage is determined by the sum of all the expression and the difference of weighted sum of individual expressions. The respective emotions are displayed if the total exceeds the threshold value.

In addition to valance metric, the points also compute degree of confidence which calculates the probability of true population parameters. Higher the degree of confidence higher wills the accuracy of emotion detection.

As demonstrated in the valance point metric gives the comprehensive feedback of the face expression. The valance points on the user’s face corresponds to the array of locations on the  
face. The value ranges from 0 to 100 specifying a neutral to positive face expression and the values ranging from 0 to -100 specify neutral to negative face expression.

**Image Processing Feature**

* **Scanning the image**

A raw image is acquired which has to be processed. It can be expressed in form of pixels as stated above. The aim of this step is to extract information which is suitable for computing.

* **Processing and Enhancing**

The image is converted into digital form by using a digitizer which samples and quantizes the input signals. The rate of sampling should be high for good resolution and high quantization level for human perception of different shades using different using Gray-scale.The obtained result describes the property of the image and further classifies the image.

* **Conversion of Colour Image to Gray Scale**

There are basically two methods to convert a color image to a gray scale image ---

1. **Average Method :** In Average method, the mean is taken of the three colors i.e. Red, Blue & Green present in a color image. Thus, we get Grayscale= (R+G+B)/3

But what happens sometimes is instead of the grayscale image we get the black image. This is because we in the converted image we get 33% each of Red, Blue & Green. Therefore, to solve this problem we use the second method called Weighted Method or Luminosity Method.

1. **Weighted or Luminosity Method :** To solve the problem in Average Method, we use Luminosity method. In this method, we decrement the presence of Red Color and increment the color of Green Color and the blue color has the percentage in between these two colors. Thus, by theequation, Grayscale=((0.3 \* R) + (0.59 \* G) + (0.11 \* B)).

We use this because of the wavelength patterns of these colors. Blue has the least wavelength while Red has the maximum wavelength.

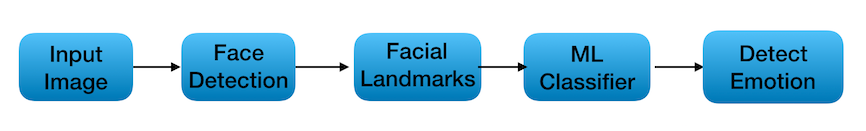
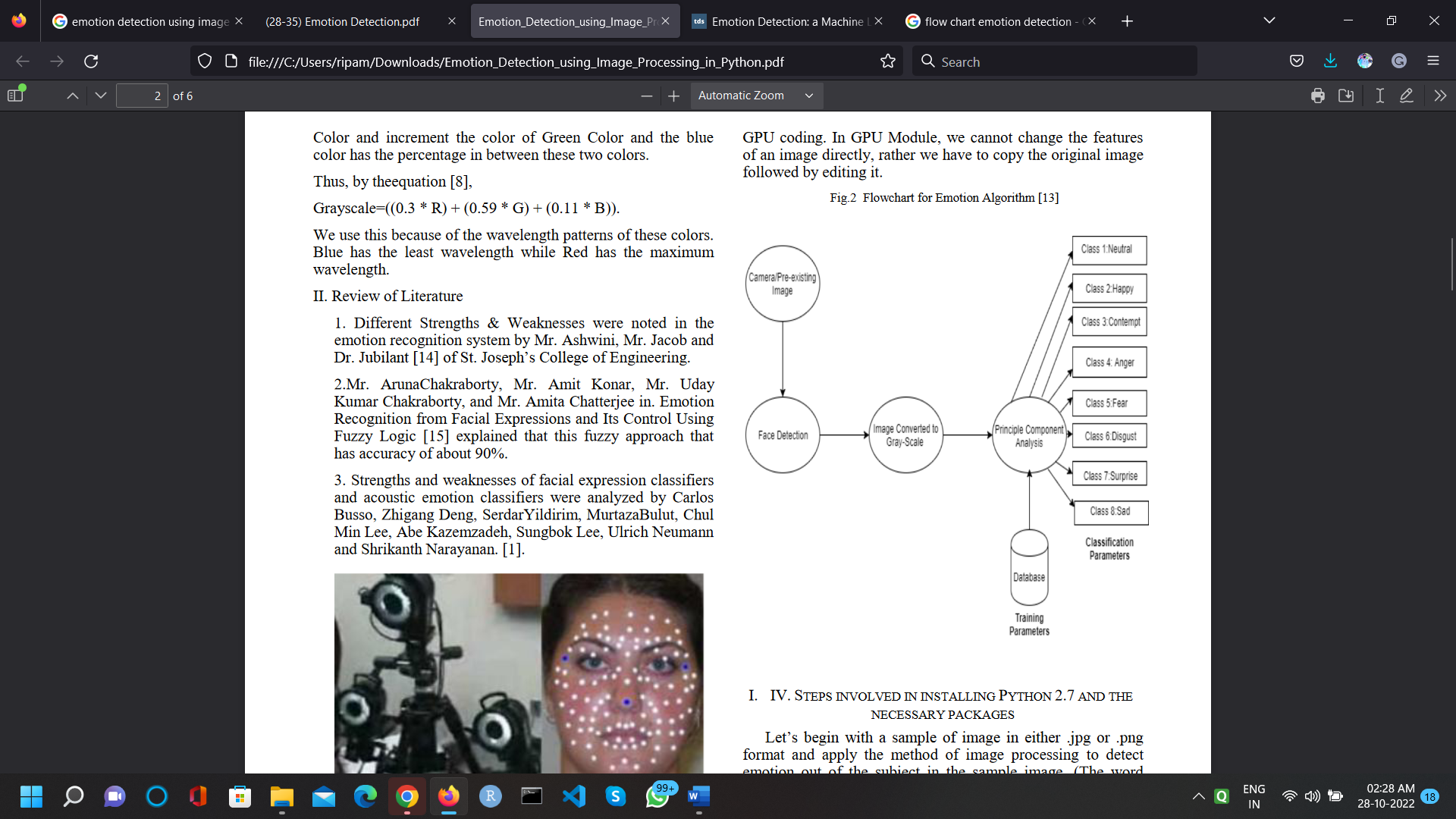
1. **DIFFERENT EMOTIONS THAT CAN BE DETECTED OUT OF AN IMAGE:**  
   A. Neutral  
   B. Happy  
   C. Anger  
   D. Surprise  
   E. Fear  
   F. Sad

**Functionality**

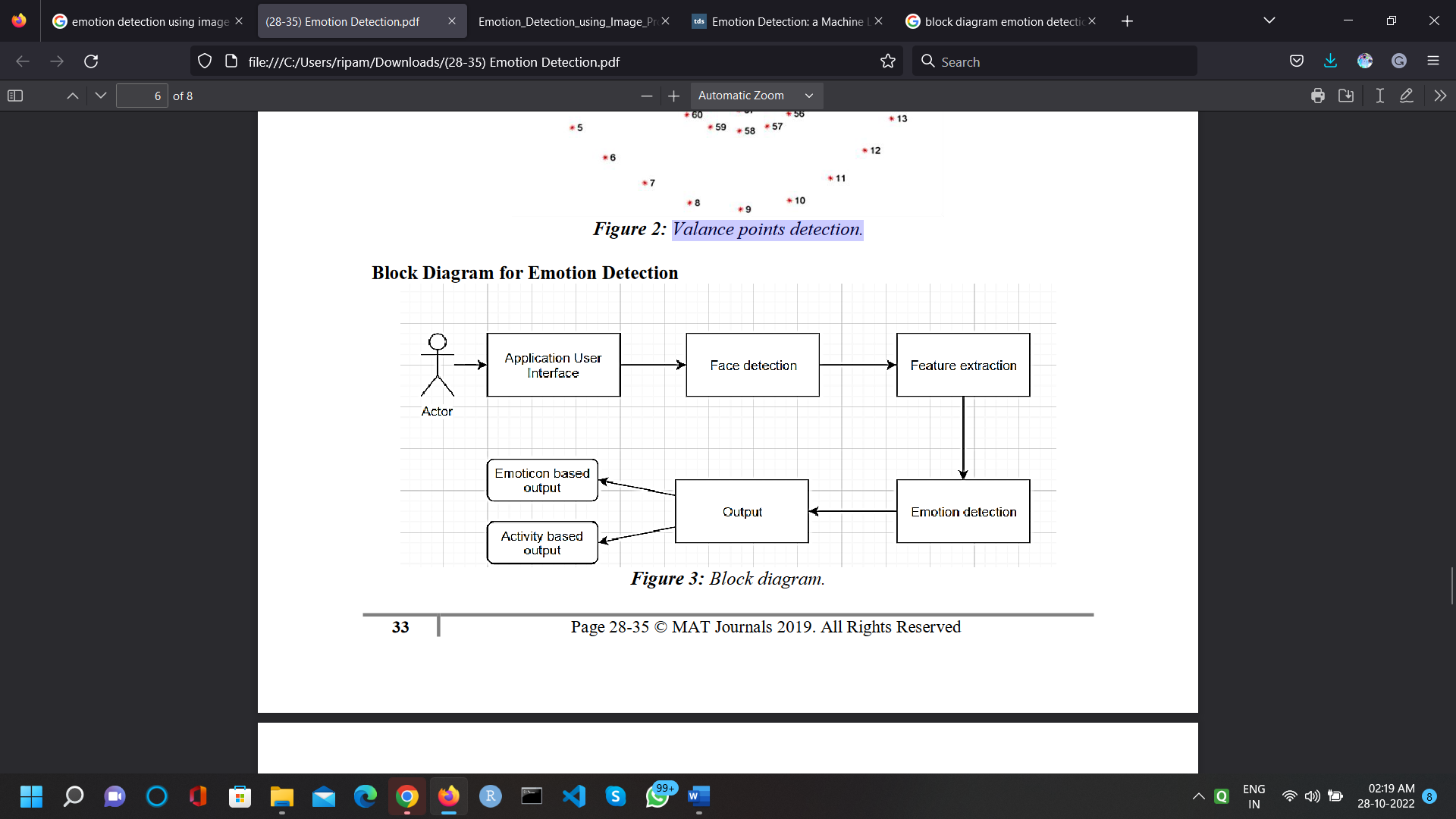
**WORKING PRINCIPLE**

* We first import the OpenCV library.
* Now start webcam in the second line using the Video Capture function of cv2.
* Then, include haarcascade files in the python file.
* Video is nothing but a series of images so we will run an infinite while loop for the same.
* Then we are reading images from the video through read ().
* As feature recognition is more accurate in Gray images, we will convert the image to Gray image using cvtColor () and BGR2GRAY which are basic OpenCV functions.
* Now we will read faces using an already included haarcascade file and detect Multiscale () function where we pass Gray image, Scale Factor, and minNeighbors.
* Scale Factor: Parameter specifying zoom image, accuracy depends on it so we will keep it close to 1 but not very close as if we take 1.001(very close to 1), then it would detect even shadows so 1.1 is good enough for the face.
* minNeighbors: Parameter specifying how many neighbours each rectangle should have to retain it.
* If it detects a face, we will draw an outer boundary of the face using rectangle () method of cv2 containing 5 arguments: image, initial point (x, y), an endpoint of principal diagonal (x + width, y + height), colour of the rectangular periphery and last parameter is the thickness of drawn rectangular periphery.
* If the face is detected then we will similarly detect a smile and if a smile is detected too, we will print Image<cnt> saved in the cmd/terminal and then we have to provide the location of the folder in which we want to save the images.
* To save the images we will use imwrite() which takes 2 parameters- location and image.
* To prevent memory overflow, we will just save 2 images in one run and thus useif statement which breaks the loop if cnt>=2.
* To break infinite loop, we have used an if statement which becomes true when we press ‘q’ denoting ‘quit’.
* At last, we will release the video.
* Do not forget to destroy all the windows.

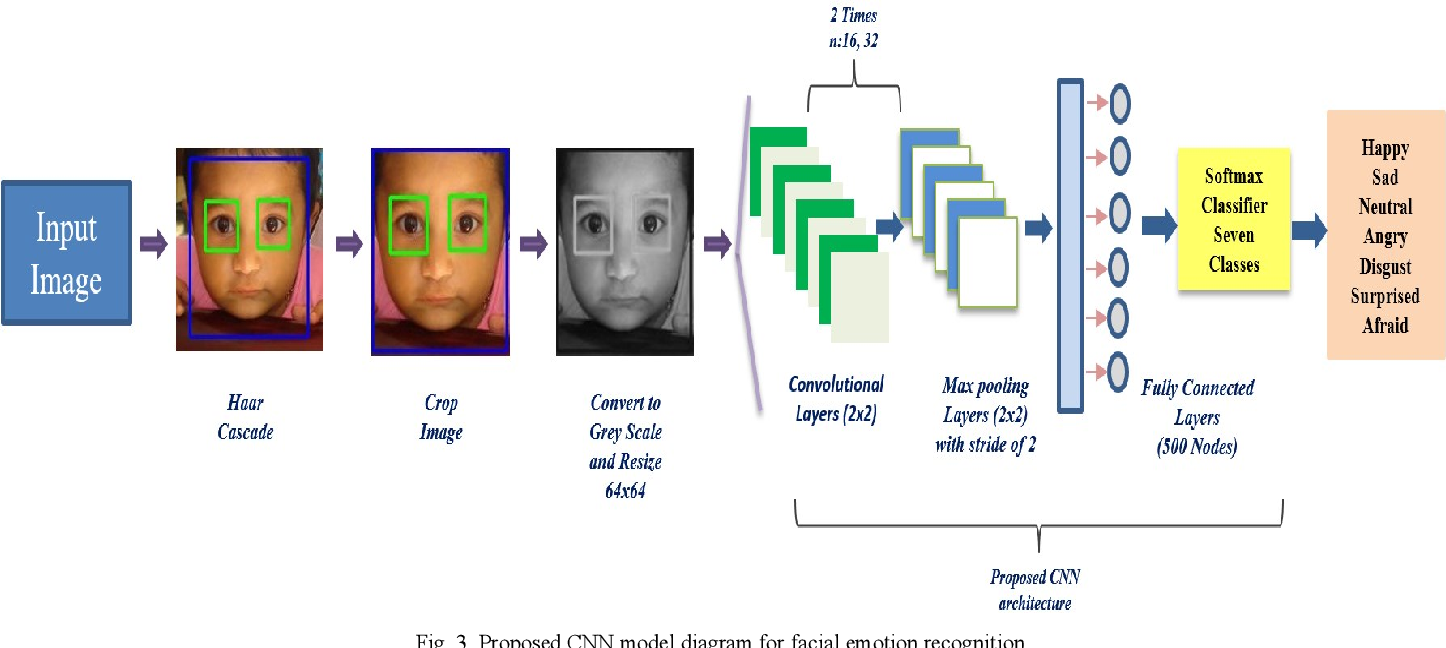
**Flow chart for Emotion Detection**



**Block Diagram for Emotion Detection**



**CNN Model Diagram for Emotion Detection**

****

**Requirement**

#### **Functional Requirements-**

Functional Requirements are those that refer to the functionality of the system, i.e., what services it will provide to the user. Non-functional (supplementary) requirements pertain to other information needed to produce the correct system and are detailed separately.

#### **Non-Functional Requirements-**

In addition to the obvious features and functions that you will provide in your system, there are other requirements that don't actually DO anything, but are important characteristics nevertheless. These are called "non-functional requirements" or sometimes "Quality Attributes." For example, attributes such as performance, security, usability, compatibility are not a "feature" of the system, but are a required characteristic. You can't write a specific line of code to implement them; rather they are "emergent" properties that arise from the entire solution. The specification needs to describe any such attributes the customer requires. You must decide the kind of requirements that apply to your project and include those that are appropriate.

Each requirement is simply stated in English. Each requirement must be objective and quantifiable; there must be some measurable way to assess whether the requirement has been met.

Often deciding on quality attributes requires making tradeoffs, e.g., between performance and maintainability. In the APPENDIX you must include an engineering analysis of any significant decisions regarding trade-offs between competing attributes.

#### **Here are some examples of non-functional requirements-**

**Performance Requirements-**

Requirements about resources required, response time, transaction rates, throughput, benchmark specifications or anything else having to do with performance.For better performance the application will restrict the document size to 5 MB.

**Operating Constraints-**

List any run-time constraints. This could include system resources, people, needed software, The application must run without any manual intervention.

# **FEASIBILITY STUDY**

#### **You should provide a feasibility report in the following format-**

* **Product ::** A general statement of the product; give a brief description of what the proposed system will do, highlighting where the proposed system meets the specified business requirements of the organization.
* **Technical Feasibility ::** Will the proposed system perform to the required specification? Outline technical systems options you propose to use, which will give a technical solution satisfying the requirements and constraints of the system, as outlined in the terms of reference.
* **Social Feasibility ::** Consideration of whether the proposed system would prove acceptable to the people who would be affected by its introduction. Describe the effect on users from the introduction of the new system; consider whether there will be a need for retraining the workforce. Will there be a need for relocation of some of the workforce? Will some jobs become deskilled? Will the current workforce be able to perform effectively any new tasks introduced by the proposed system? Describe how you propose to ensure user co-operation before changes are introduced.
* **Economic Feasibility ::** Such as improved management information and better customer service. Illustrate the Consider the cost/benefits of the proposed system. Detail the costs that will be incurred by the organization adopting the new system; consider development costs and running costs. Detail benefits that the new system will bring, direct economic benefits such as reduced costs, and indirect benefits cost/benefit of the new system by applying a suitable cost/benefit analysis method such as the payback method.
* **Market Research ::** A comprehensive market research for identifying a need for the product. Detail all market research you carried out, listing sources of information. Justify any conclusions you have drawn from your research. Identify the potential customer base for your product, together with evidence of customer need for the product. Describe how you propose to compete with similar products on the market.
* **Alternative Solution ::** Consideration of alternative solutions should be documented. At least two alternative business or technical systems options should be considered. Detail the differences between these options and the proposed system. Justify your choice of the proposed system and the reasons for rejecting the alternative options At this point, all of the planning for the project has been done and if the feasibility study has shown that the project is likely to succeed within its constraints, then it only remains for us to start the requirements analysis and thus proceed with the project.

# **PROJECT PLANNING**

**Project planning is concerned with identifying the following for every project:**

* Activities
* Milestones
* Deliverables.

A plan must be drawn up to guide the development towards the project goal. A plan is drawn up at the start of a project. This plan should be used as the driver for the project. The initial plan is not static, and must be modified as the project progresses. Planning is required for development activities from specification through to delivery of the system.

# **PROJECT SCHEDULING**

The project scheduling can be represented diagrammatically with the help of GANTT chart. A Gantt chart is a matrix which lists on the vertical axis all the tasks to be performed. Each row contains a single task identification which usually consists of a number and name. The horizontal axis is headed by columns indicating estimated task duration, skill level needed to perform the task, and the name of the person assigned to the task, followed by one column for each period in the project's duration. The graphics portion of the Gantt chart consists of a horizontal bar for each task connecting the period start and period ending columns.

The overall period for the development of the project is 30 days. According to the CNN Model, classification, Regration the different phases are described in the following Gantt chart. The required days for every six different phases of the model have already mentioned above in the effort estimation table. In the table the effort percentage and effort duration is mentioned. On the basis of those days the Gantt chart is made. Color is used for indicating the different phases of CNN Model, classification, Regration. After the chart the use of every colour for each phase is described in a box.

**The Gantt chart to be followed for the development of the project is given below:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task** | **Person(s) Responsible** | **Week 1** | **Week 2** | **Week 3** | **Week 4** |
| **Communication** | Ripam,  Arijit,  Mritanka |  |  |  |  |
| **Quick Plan** | Ripam,  Arijt. |  |  |  |  |
| **Modeling Quick Design** | Arijti Mandel |  |  |  |  |
| **Coding** | Ripam , Mritnaka, Arijt |  |  |  |  |
| **Deployment** | Ripam Kundu |  |  |  |  |

**GANTT chart**

#### **Note :**

**Stage 1:** Preliminary Investigation & Feasibility Study.

**Stage 2:** System Analysis

**Stage 3:** System Design

**Stage 4:** Coding

**Stage 5:** Testing

**Stage 6:** Implementation & Maintenance

# **PROJECT PLANNING & SCHEDULING**

Project planning can be represented diagrammatically with the help of **Program Evaluation and Review Technique (PERT) chart.** PERT chart describe task, duration, and dependency information.

A PERT chart may have multiple parallel or interconnecting networks of tasks. If the scheduled project has milestones, checkpoints, or review points (all of which are highly recommended in any project schedule), the PERT chart will note that all tasks up to that point terminate at the review node. It should be noted at this point that the project review, approvals, user reviews, and so forth all take time. This time should never be underestimated when drawing up the project plan. It is not unusual for a review to take 1 or 2 weeks. Obtaining management and user approvals may take even longer.

#### **According to Project planning following phases are covered**

**Preliminary Investigation & Feasibility Study**

**System Analysis**

**Testing**

**Implementation & Maintenance**

**System Design**

**Coding**

**kjj**

#### **Phases are described as follows :**

#### **Preliminary Investigation & Feasibility Study ::** Before considering the system analysis the software demand must be clarified. It also be checked that project is economically, technically, operationally feasible.

#### **System Analysis ::** Software demand is accepted; its cost, precedence, completion time, and personnel requirement are estimated. Some tools are also used in analysis.

#### **System Design ::** In this stage it is decided that technically problem is solved in which way. No of modules, algorithms etc are decided in this phase.

#### **Coding ::** This phase is most vital for a project. In this phase complete project code is developed.

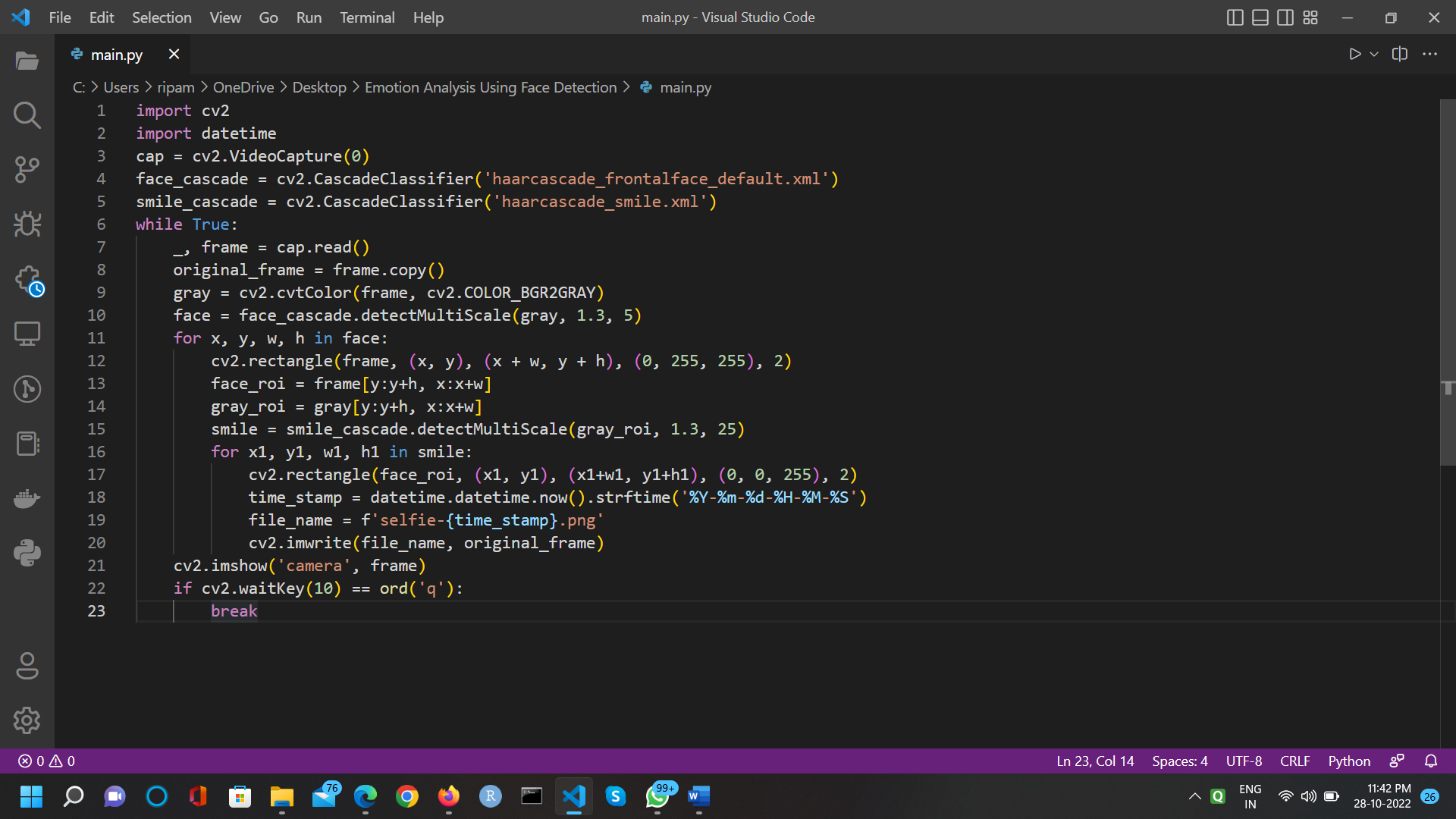
#### **Testing ::** As the coding phase proceeds, every unit module is tested before being delivered to the next phase. Final Software is then tested under System Testing.

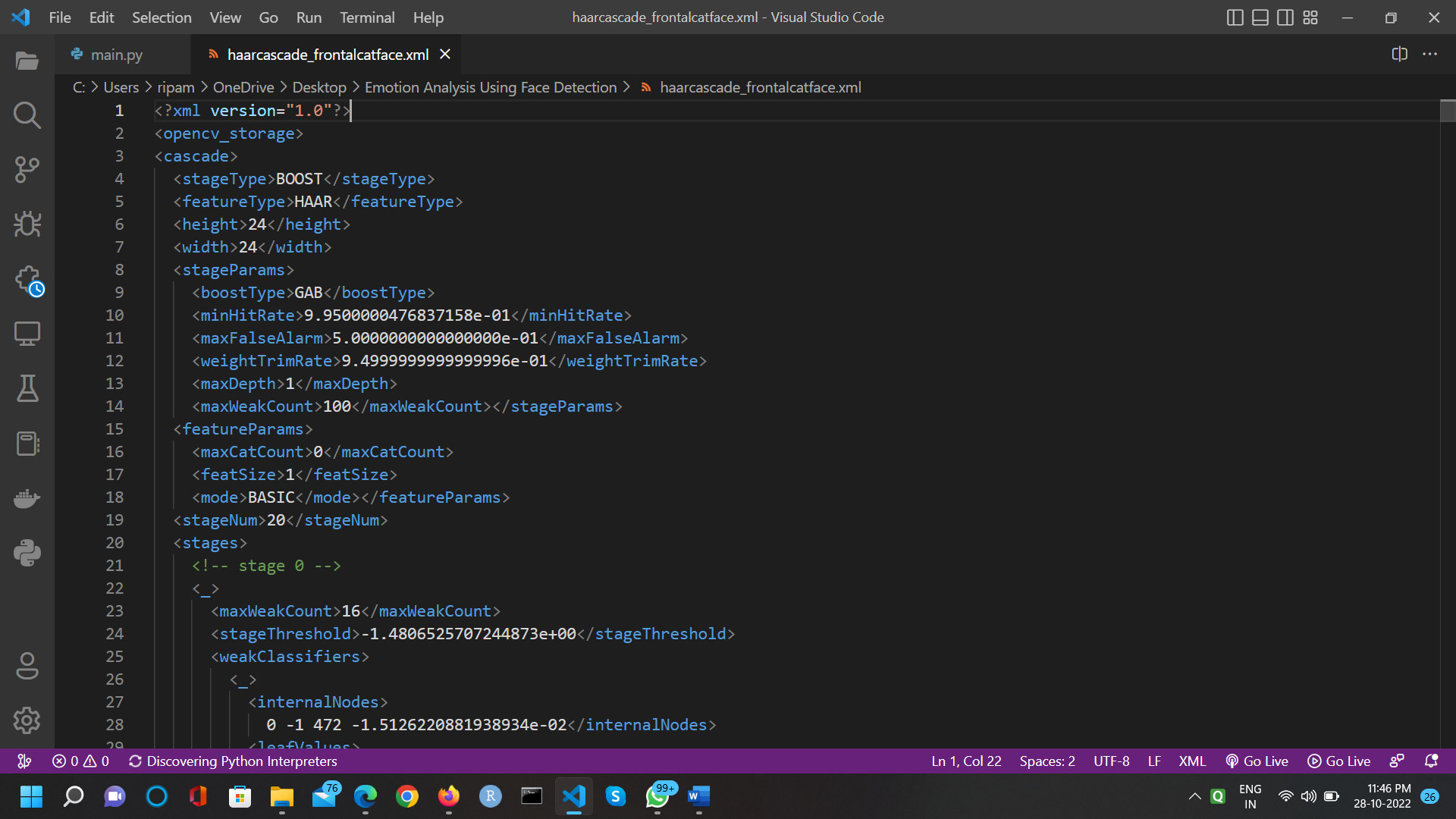
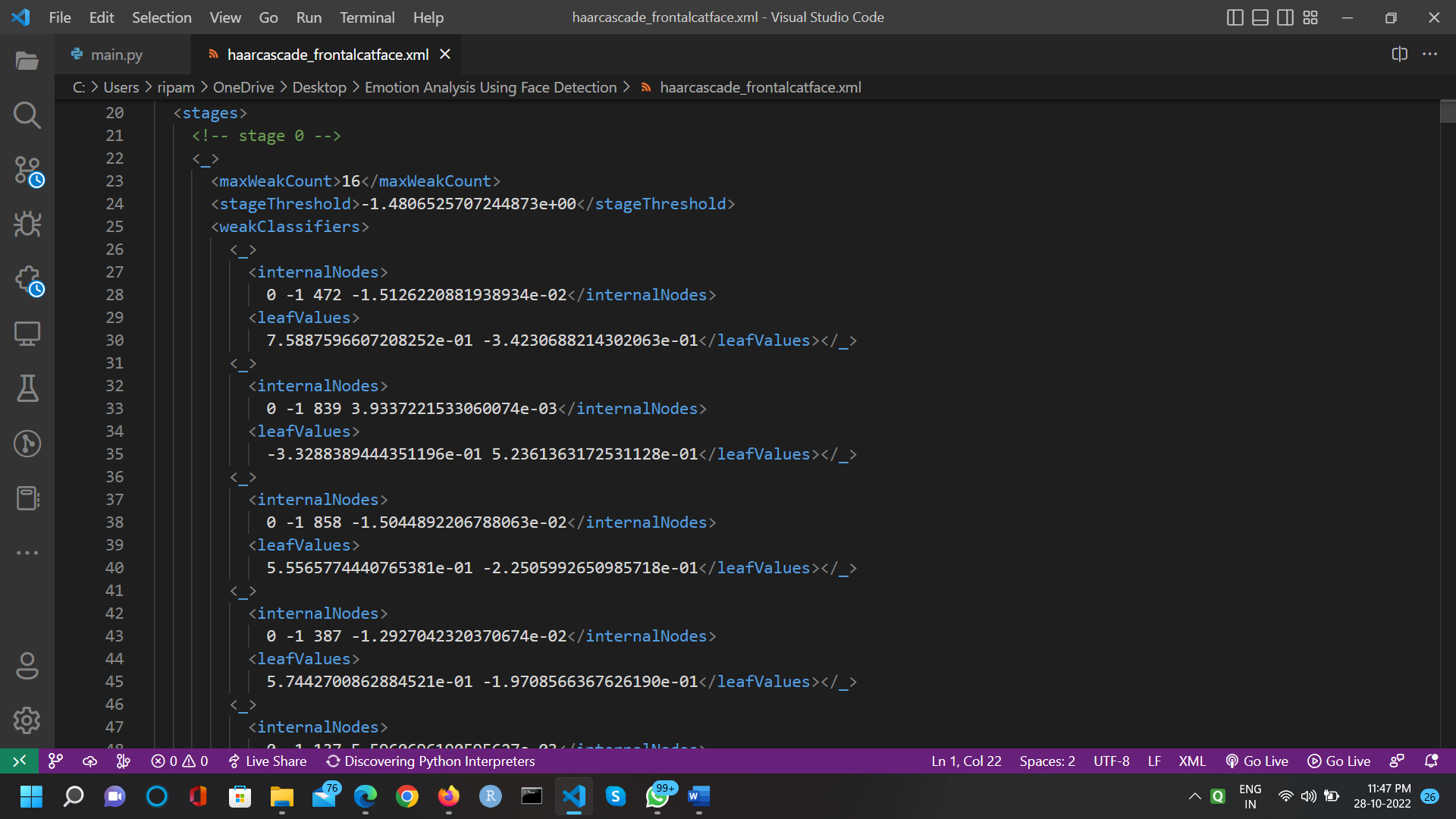
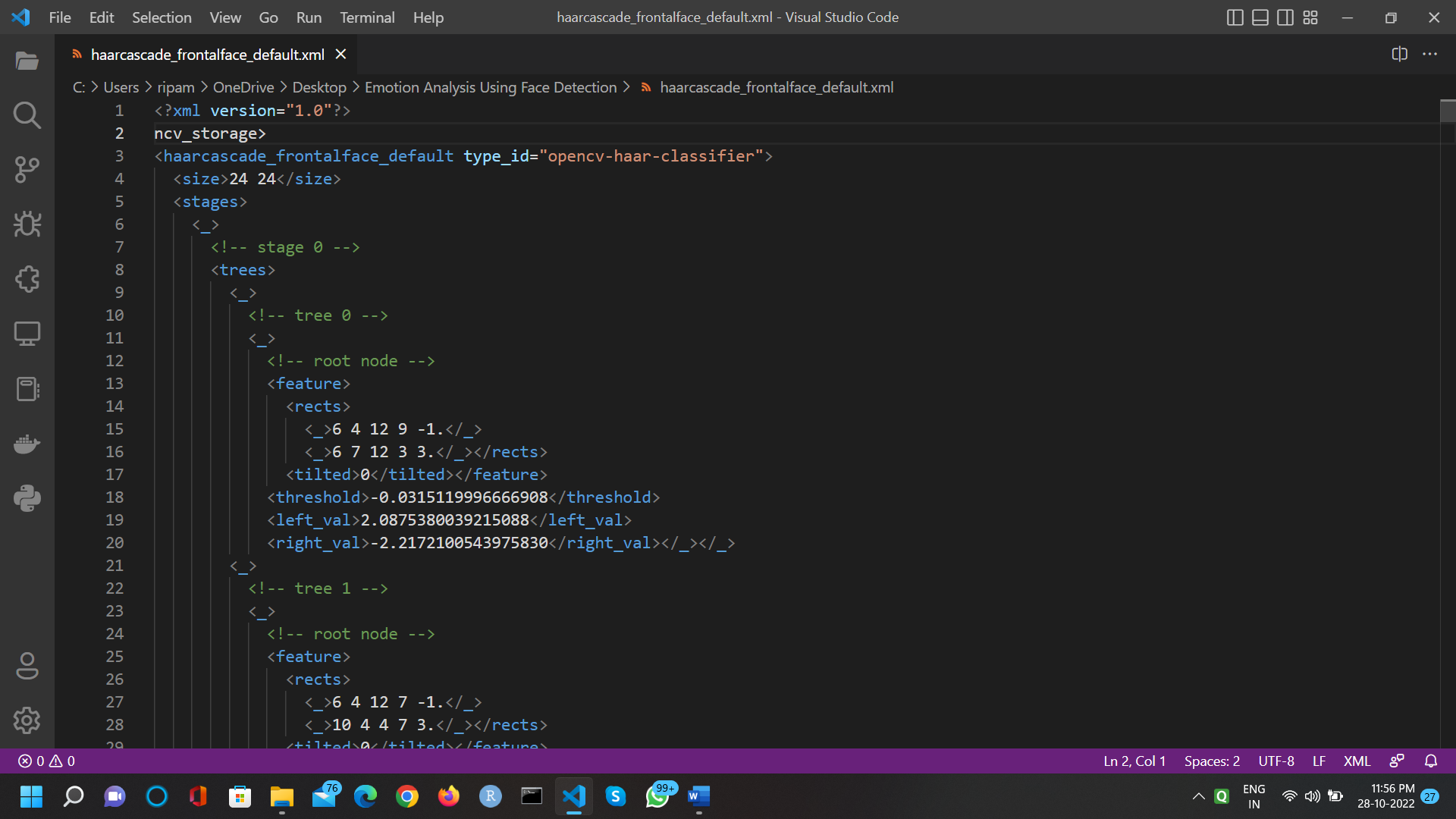
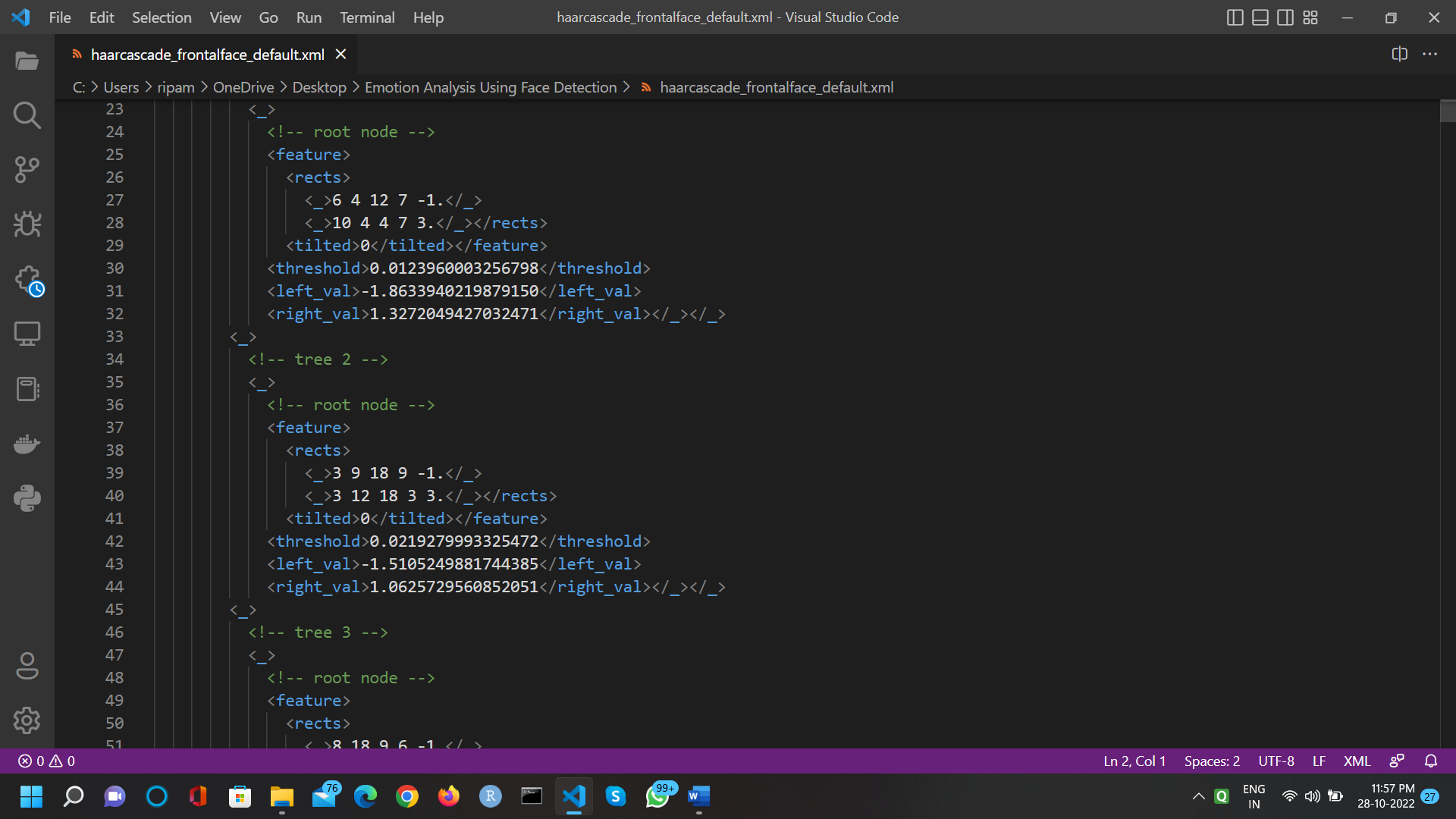
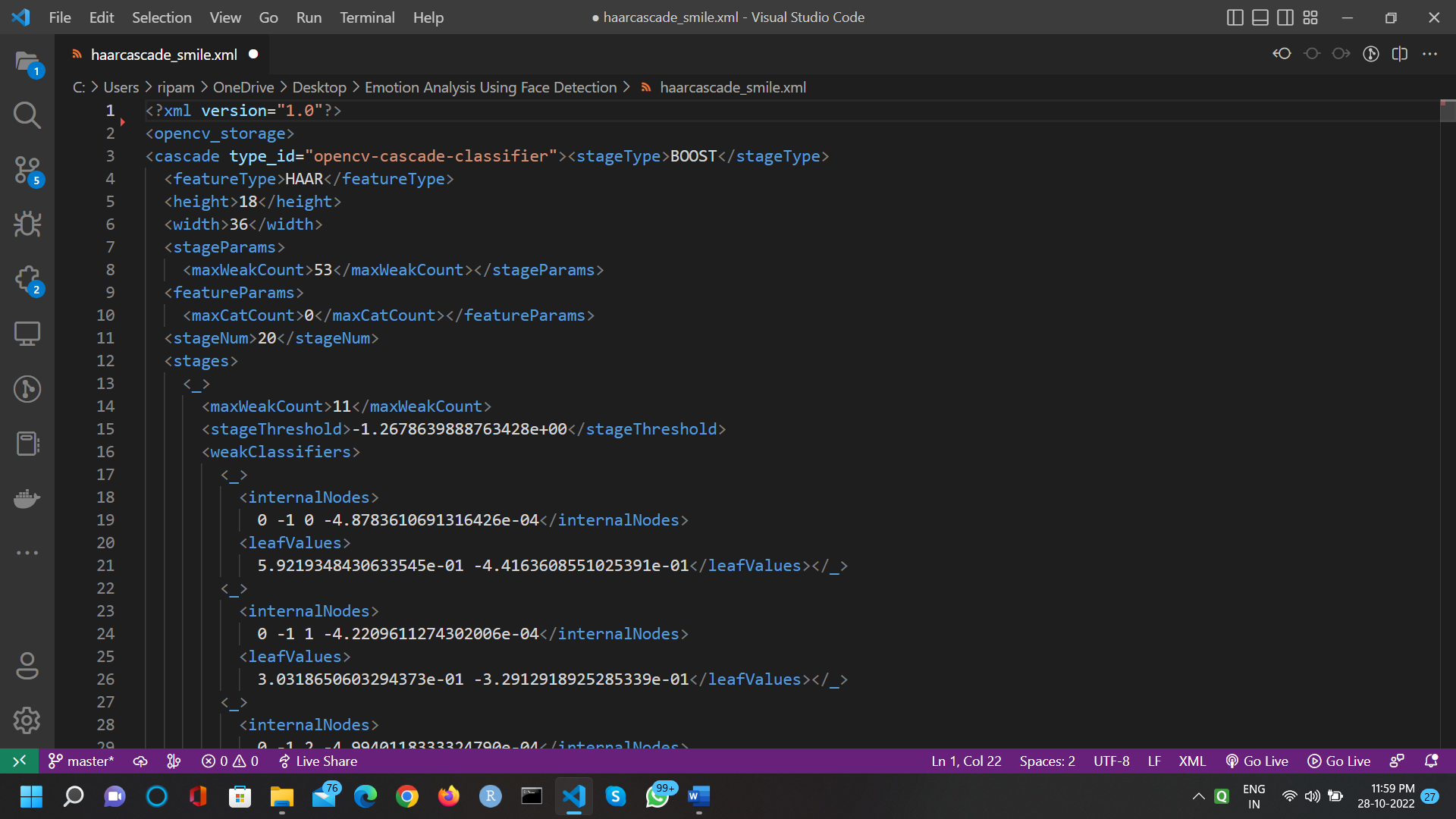
#### **Implementation & Maintenance ::** Once the Systempassesall the test during System testing the system is delivered to the College where it has to be implemented successfully. Maintenance phase includes addition of modules, modification of the hardware and software.

**Screenshot**

**Code**

* **Main.py**



* **haarcascade\_frontalcatface.xml**
* **haarcascade\_frontalface\_default.xml**
* **haarcascade\_smile.xml**

**Testing Overview**

The best results came by using the VGG which were right about 68–70% of the time, but even the linear models did a very good job. Although 50% accuracy does not seem like a lot, it is still more than if you would pick up a picture and a label at random. There, you would be right about 20% of the time.

The VGG, however, performs even better than humans do in this particular dataset. The difference between the CNNs and the MLPs was that the CNNs were extracting features that they deemed important by themselves, while we were feeding either pixels or landmarks as features to the MLPs.

Humans are used to taking in nonverbal cues from facial emotions. Now computers are also getting better to reading emotions. So how do we detect emotions in an image? We have used an open source data set — Face Emotion Recognition (FER) from Kaggle and built a CNN to detect emotions. The emotions can be classified into 7 classes — happy, sad, fear, disgust, angry, neutral and surprise.

Model — We built a 6 layered Convolutional Neural Network (CNN) in Keras and use image augmentations to improve model performance.

The figures attached below are the real time images captured when the person has used the emotion detection application. The application takes input in real time i.e. dynamic images. The application detects and monitors the face even after it has calculated the percentage value of different emotions hence when the face expression changes the corresponding changes are reflected in the emotion detection which leads to the values of the percentages to change accordingly. Emoticons are displayed right next to the face and they are in accordance with the emotion detected.

The Test Team will work closely with the Development Team to achieve a high-quality design and user interface specifications based on user requirements. The Test Team is responsible for visualizing test cases and raising quality issues and concerns during meetings to address issues early enough in the development cycle.

The Test Team will work closely with Development Team to determine whether or not the application meets standards for completeness. If an area is not acceptable for testing, the code complete date will be pushed out, giving the developers additional time to stabilize the area.

Since the application interacts with a back-end system component, the Test Team will need to include a plan for integration testing. Integration testing must be executed successfully prior to system testing.

#### **Test Objective ::** The objective our test plan is to find and report as many bugs as possible to improve the integrity of our program. Although exhaustive testing is not possible, we will exercise a broad range of tests to achieve our goal. Our user interface to utilize these functions is designed to be user-friendly and provide easy manipulation of the tree. The application will only be used as a demonstration tool, but we would like to ensure that it could be run from a variety of platforms with little impact on performance or usability.

#### **Software testing ::** Is an investigation conducted to provide stakeholders with information about the quality of the product or service under test Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation test techniques include, but are not limited to, the process of executing a program or application with the intent of finding software bugs (errors or other defects).

Software testing can be stated as the process of validating and verifying that a software program/application/product:

* meets the requirements that guided its design and development;
* works as expected; and
* Can be implemented with the same characteristics.

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed. As such, the methodology of the test is governed by the software development methodology adopted.

Different software development models will focus the test effort at different points in the development process. Newer development models, such as Agile, often employ test driven development and place an increased portion of the testing in the hands of the developer, before it reaches a formal team of testers. In a more traditional model, most of the test execution occurs after the requirements have been defined and the coding process has been completed.

Testing can never completely identify all the defects within software Instead, it furnishes a *criticism* or *comparison* that compares the state and behavior of the product against oracles—principles or mechanisms by which someone might recognize a problem. These oracles may include (but are not limited to) specifications, contracts, comparable products, past versions of the same product, inferences about intended or expected purpose, user or customer expectations, relevant standards, applicable laws, or other criteria.

Every software product has a target audience. For example, the audience for video game software is completely different from banking software. Therefore, when an organization develops or otherwise invests in a software product, it can assess whether the software product will be acceptable to its end users, its target audience, its purchasers, and other stakeholders. **Software testing** is the process of attempting to make this assessment.

A study conducted by NIST in 2002 reports that software bugs cost the U.S. economy $59.5 billion annually. More than a third of this cost could be avoided if better software testing was performed.

The separation of debugging from testing was initially introduced by Glen ford J. Myers in 1979. Although his attention was on breakage testing ("a successful test is one that finds a bug") it illustrated the desire of the software engineering community to separate fundamental development activities, such as debugging, from that of verification. Dave Gelperin and William C. Hetzel classified in 1988 the phases and goals in software testing in the following stages.

#### **Scope ::** A primary purpose of testing is to detect software failures so that defects may be discovered and corrected. Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions. The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do. In the current culture of software development, a testing organization may be separate from the development team. There are various roles for testing team members. Information derived from software testing may be used to correct the process by which software is developed.

#### **Functional vs. non-functional testing ::** Functional testing refers to activities that verify a specific action or function of the code. These are usually found in the code requirements documentation, although some development methodologies work from use cases or user stories. Functional tests tend to answer the question of "can the user do this" or "does this particular feature work."

Non-functional testing refers to aspects of the software that may not be related to a specific function or user action, such as scalability or other performance, behavior under certain constraints, or security. Non-functional requirements tend to be those that reflect the quality of the product, particularly in the context of the suitability perspective of its users.

#### **Defects and failures ::** Not all software defects are caused by coding errors. One common source of expensive defects is caused by requirement gaps, e.g., unrecognized requirements that result in errors of omission by the program designer. A common source of requirements gaps is non-functional requirements such as testability, scalability, maintainability, usability, performance, and security.

Software faults occur through the following processes. A programmer makes an error (mistake), which results in a defect (fault, bug) in the software source code. If this defect is executed, in certain situations the system will produce wrong results, causing a failure. Not all defects will necessarily result in failures. For example, defects in dead code will never result in failures. A defect can turn into a failure when the environment is changed. Examples of these changes in environment include the software being run on a new computer hardware platform, alterations in source data or interacting with different software. A single defect may result in a wide range of failure symptoms.

# **Finding faults early ::** It is commonly believed that the earlier a defect is found the cheaper it is to fix it. The following table shows the cost of fixing the defect depending on the stage it was found. For example, if a problem in the requirements is found only post-release, then it would cost 10–100 times more to fix than if it had already been found by the requirements review. Modern continuous deployment practice and cloud-based services may cost less for re-deployment and maintenance than in the past.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Cost to fix a defect | | Time detected | | | | |
| Requirements | Architecture | Coding | System test | Post-release |
| Time introduced | Requirements | 1× | 3× | 5–10× | 10× | 10–100× |
| Architecture | - | 1× | 10× | 15× | 25–100× |
| Construction | - | - | 1× | 10× | 10–25× |

#### **Compatibility ::** A common cause of software failure (real or perceived) is a lack of its compatibility with other application software, operating systems (or operating system versions, old or new), or target environments that differ greatly from the original (such as a terminal or GUI application intended to be run on the desktop now being required to become a web application, which must render in a web browser). For example, in the case of a lack of backward compatibility, this can occur because the programmers develop and test software only on the latest version of the target environment, which not all users may be running. This results in the unintended consequence that the latest work may not function on earlier versions of the target environment, or on older hardware that earlier versions of the target environment was capable of using. Sometimes such issues can be fixed by proactively abstracting operating system functionality into a separate program module or library.

#### **Input combinations and preconditions ::** A very fundamental problem with software testing is that testing under all combinations of inputs and preconditions (initial state) is not feasible, even with a simple product. This means that the number of defects in a software product can be very large and defects that occur infrequently are difficult to find in testing. More significantly, non-functional dimensions of quality (how it is supposed to be versus what it is supposed to do)—usability, scalability, performance, compatibility, reliability—can be highly subjective; something that constitutes sufficient value to one person may be intolerable to another.

#### **Static vs. dynamic testing ::** There are many approaches to software testing. Reviews, walkthroughs, or inspections are considered as static testing, whereas actually executing programmed code with a given set of test cases is referred to as dynamic testing. Static testing can be (and unfortunately in practice often is) omitted. Dynamic testing takes place when the program itself is used for the first time (which is generally considered the beginning of the testing stage). Dynamic testing may begin before the program is 100% complete in order to test particular sections of code (modules or discrete functions). Typical techniques for this are either using stubs/drivers or execution from a debugger environment. For example, spreadsheet programs are, by their very nature, tested to a large extent interactively ("on the fly"), with results displayed immediately after each calculation or text manipulation.

#### **Software verification and validation-**

**Software testing is used in association with verification and validation:**

* Verification: Have we built the software right? (i.e., does it match the specification).
* Validation: Have we built the right software? (i.e., is this what the customer wants).
* The terms verification and validation are commonly used interchangeably in the industry; it is also common to see these two terms incorrectly defined. According to the IEEE Standard Glossary of Software Engineering Terminology:
* Verification is the process of evaluating a system or component to determine whether the products of a given development phase satisfy the conditions imposed at the start of that phase.
* Validation is the process of evaluating a system or component during or at the end of the development process to determine whether it satisfies specified requirements.

#### **The software testing team ::** Software testing can be done by software testers. Until the 1980s the term "software tester" was used generally, but later it was also seen as a separate profession. Regarding the periods and the different goals in software testing, different roles have been established: manager, test lead, test designer, tester, automation developer, and test administrator.

# **Process Overview**

#### **The following represents the overall flow of the testing process ::**

1. Identify the requirements to be tested. All test cases shall be derived using the current Program Specification.
2. Identify which particular test(s) will be used to test each module.
3. Review the test data and test cases to ensure that the unit has been thoroughly verified and that the test data and test cases are adequate to verify proper operation of the unit.
4. Identify the expected results for each test.
5. Document the test case configuration, test data, and expected results.
6. Perform the test(s).
7. Document the test data, test cases, and test configuration used during the testing process. This information shall be submitted via the Unit/System Test Report (STR).
8. Successful unit testing is required before the unit is eligible for component integration/system testing.
9. Unsuccessful testing requires a Bug Report Form to be generated. This document shall describe the test case, the problem encountered, its possible cause, and the sequence of events that led to the problem. It shall be used as a basis for later technical analysis.
10. Test documents and reports shall be submitted. Any specifications to be reviewed, revised, or updated shall be handled immediately.

**Future scope**

Understanding the meaning of facial expressions is an essential human survival tool, but one that generates scientific controversy. Some contend that we recognize anger most easily1, since it may represent an existential threat; others that we recognize happiness most easily and anger most slowly, since the march of civilization has altered our priorities over time.

The lexicon of facial expression is rich in ambiguities, as facial cues may be insincere, misinterpreted, or in some way at a tangent to their intent. Experts in the understanding of facial expression remain split even on whether the Mona Lisa's enigmatic smile is sincere or forced (apparently a matter of context).

The ability to decipher the true intent and emotional response of a person from their facial expressions, notwithstanding their attempts to mask or deceive what they feel, is an evolutionary advantage of great interest to a range of sectors, from physicians through to marketers and political analysts. Unsurprisingly, there's a lot of money in it.

When you drive your car for a long time, what happens if you fall asleep. It leads to accidents, but what if your steering vibrates and makes a beep sound when falling asleep or feeling drowsy. This is due to the technological advancements in computer vision and artificial intelligence in the automobile industry. This blog talks about the future of emotion recognition in machine learning and AI.

**CONCLUSION**

At first, there were many reasons why the model didn’t work, it would detect non-smiling faces with his smile too. I had to work with the scale factor, as well as the minimum neighbours to get a perfect result. In this project, we have developed a python project to detect the smile and capture selfies using OpenCV And we have used OpenCV because it’s a popular machine learning library in the field of computer vison

.

We have developed an android based application that enables the user to detect his emotion and have connected his emotion into an output based system, which can either results in the corresponding emoticon, or a suggestion based output which would suggest activities to the user. A unique feature of the developed application is that it calculates the percentages of different emotions that the user is experiencing and the respective emotions are also displayed alongside when the emotion is calculate. Fig. 4 and 5 demonstrate the calculated percentage of emotion. The developed application is simple and user friendly, also the entire process has been developed to work quickly yet give an accurate output.

It was a wonderful and learning experience for me while working on this project. This project took me through the various phases of project development and gave me real insight into the world of software engineering. The joy of working and the thrill involved while tackling the various problems and challenges gave me a feel of developer’s industry. It was due to this project I came to know how professional software’s are designed. I enjoyed each and every bit of work I had put into this project. The project is further extendable.

**REFERENCE**

* <https://analyticsindiamag.com/my-fun-project-using-opencv/>
* [www.analyticsindiamag.com](http://www.analyticsindiamag.com)
* [Books-OpenCV](Project%20Documentation.docx)
* <https://www.tutorialpoint.com>
* <https://www.w3schools.com>
* <https://www.codeproject.com/>
* <https://sourcecode.com/>
* <https://data-flair.training/blogs/>