FACE MASK DETECTION

A Mini Project Report

Submitted to the Faculty of Engineering of
JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY KAKINADA, KAKINADA

In partial fulfillment of the requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY
In
COMPUTER SCIENCE AND ENGINEERING

By

P.SREYA
(18481A05H8)

K.POOJASRI
(18481A05C4)

P.YESHWANTH
(18481A05G2)

P.SIVARAMAKUMARI
(19481A0522)

Under the Enviable and Esteemed Guidance of
Smt T.NAGAMANI, M.Tech(Ph.d)
Assistant Professor, Department of CSE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

GUDLAVALLERU ENGINEERING COLLEGE
(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
SESHADRIRAO KNOWLEDGE VILLAGE
GUDLAVALLERU – 521356
ANDHRA PRADESH

2020-21

GUDLAVALLERU ENGINEERING COLLEGE

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
SESHADRI RAO KNOWLEDGE VILLAGE, GUDLAVALLERU

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project report entitled **"Face Mask Detection"** is a bonafide record of work carried out by **P.Sreya (18481A0H8), P.Yeshwanth (18481A05G2), K.PoojaSri (18481A05C4), P.Sivaramakumari (19485A0522)** under the guidance and supervision of **Mrs T.Nagamani** in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering of Jawaharlal Nehru Technological University Kakinada, Kakinada** during the academic year 2020-21.

Project Guide
(Mrs T.Nagamani)

Head of the Department
(Dr. M. BABU RAO)

External Examiner

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of any task would be incomplete without the mention of people who made it possible and whose constant guidance and encouragements crown all the efforts with success.

We would like to express our deep sense of gratitude and sincere thanks to **Mrs.T.Nagmani, Assistant Professor** in the Department of Computer Science and Engineering for **her** constant guidance, supervision and motivation in completing the project work.

We feel elated to express our floral gratitude and sincere thanks to **Dr. M. Babu Rao**, Head of the Department, Computer Science and Engineering for his encouragements all the way during analysis of the project. His annotations, insinuations and criticisms are the key behind the successful completion of the project work.

We would like to take this opportunity to thank our beloved principal **Dr. G. V. S. N. R. V. Prasad** for providing a great support for us in completing our project and giving us the opportunity for doing project.

Our Special thanks to the faculty of our department and programmers of our computer lab. Finally, we thank our family members, non-teaching staff and our friends, who had directly or indirectly helped and supported us in completing our project in time.

Team Members

18481A05H8
18481A05G2
18481A05C4
19485A0522

ABSTRACT

The statistical data acquired by the World Health Organization (WHO), show that the COVID 19 pandemic has heartlessly affected the life of human beings and even the economic state of the world where over 7 million of people have been infected with the virus. The pandemic led to governments all over the world to impose lockdowns so as to prevent the transmission of the Coronavirus. So the actual safety measures being advocated by WHO are wearing of face masks at public places so that the spread of the virus can be prevented. For creating a safety atmosphere which contributes to public safety, we propose an effective computer vision based system using Artificial Intelligence. A hybrid system model using classical and deep learning for facial mask recognition and detection will be implemented. We will be using a face mask detection dataset which will consist of images with mask and images without mask, OpenCV to detect faces from a livestream through the Webcam in real-time. The image dataset will be used to build the face mask detection system. The system will be implemented with deep learning using python, Opencv. Keras and Tensorflow. The main goal of the system is to identify whether the detected individual on the video or image is wearing a mask or is without the mask with the approach of computer vision and machine/deep learning. In hospitals, offices, stores, malls and any other public places it will be used to alert an individual without the mask to put it on so as to curb the transmission of the COVID-19 virus.

The proposed system model will be built with deep learning algorithms that are diversified with some geometric methods so as to build a robust system covering the following three features of tracking, detection and validation. The system shall support the society through offering lower spread of COVID-19 and time saving. The system will be effectively implemented during this current condition where lockdowns have been eased to allow public gatherings, mall shopping, church gathering and reopening of schools. This automation of checks will minimize the number of manpower for inspections at public gatherings hence can be used at any situation and period of time.

INDEX

CHAPTER-I
INTRODUCTION

The 209th report of the world health organization (WHO) published on 16th August 2020 reported that coronavirus disease (COVID-19) caused by acute respiratory syndrome (SARS-CoV2) has globally infected more than 6 Million people and caused over 379,941 deaths worldwide. The key to control COVID-19 pandemic is to maintain social distancing, improving surveillance and strengthening health systems. Recently, a study on understanding measures to tackle COVID-19 pandemic carried by the researchers at the University of Edinburgh reveals that wearing a face mask or other covering over the nose and mouth cuts the risk of Coronavirus spread by avoiding forward distance travelled by a person's exhaled breath by more than 90%. The findings reveal that near universal adoption (80%) of even weak masks (20% effective) could prevent 17–45% of projected deaths over two months in New Work and reduce the peak daily death rate by 34–58%. Their results strongly recommend the use of face masks in the general public to curtail the spread of Coronavirus. Further, with the reopening of countries from COVID-19 lockdown, Government and Public health agencies are recommending face masks as essential measures to keep us safe when venturing into public. To mandate the use of facemasks, it becomes essential to devise some techniques that enforce individuals to apply a mask before exposure to public places.

Face mask detection refers to detect whether a person is wearing a mask or not. In fact, the problem is reverse engineering of face detection where the face is detected using different machine learning algorithms for the purpose of security, authentication and surveillance. Face detection is a key area in the field of Computer Vision and Pattern Recognition. In recent years, face detection methods based on deep convolutional neural networks (CNN) have been widely developed to improve detection performance.

Although numerous researchers have committed efforts in designing efficient algorithms for face detection and recognition, there exists an essential difference between 'detection of the face under mask' and 'detection of mask over face'. As per available literature, very little body research is attempted to detect masks over face. Thus, our work aims to

develop a technique that can accurately detect mask over the face in public areas (such as airports. railway stations, crowded markets, bus stops, etc.) to curtail the spread of Coronavirus and thereby contributing to public healthcare. Further, it is not easy to detect faces with/without a mask in public as the dataset available for detecting masks on human faces is relatively small leading to the hard training of the model. So, the concept of transfer learning is used here to transfer the learned kernels from networks trained for a similar face detection task on an extensive dataset. The dataset covers various face images including faces with masks, faces without masks, faces with and without masks in one image and confusing images without masks. With an extensive dataset containing 3,833 images, our technique achieves outstanding accuracy of 93.2%.

1.2 PROBLEM STATEMENT

The main problem and goal is to facilitate the designing and implementation of a face mask detection system model. The proposed system model should have the capability of detecting the faces with face mask and those without face masks. The system model shall be integrated with video surveillance systems used in health facilities, ATM banking systems and public places for monitoring of the face masks.

1.3 EXISTING SYSTEM

The trend of wearing face masks in public is rising due to the COVID- 19 coronavirus epidemic all over the world. At the initial stage of COVID, people were manually checked by the security guards or any other person near the entry point. And they were allowed if they were wearing mask. But whenever there is a huge crowd or gathering of people it would be very difficult to track who are wearing a mask and who are not wearing a mask as it would be a time consuming process as each person need to be checked individually.

1.4 DISADVANTAGES

- Whenever there is a power failure then our project model will not work as it needs continuous power supply
- The accuracy of our project will be based upon the camera resolution when the camera resolution was low then the detection of face mask will be only upto little extent.
- Whenever there are obstacles like covering the face with hands or any cloth then it will be treated as wearing mask.

1.5 PROPOSED SYSTEM

We introduce deep learning-based analysis frameworks like TensorFlow, Keras which help to detect the person whether wearing a mask or not. Using a combination of face detection, object tracking, image classification, and video analysis we develop a robust system that can detect the presence and absence of face masks in images as well as real-time videos. This can be mainly used in security cameras so that whenever there is a huge crowd we might easily identify who is wearing a mask and ensure a safe working environment.

1.6 ADVANTAGES

- Manual Monitoring is very difficult for officers to check whether the people are wearing masks or not. So in our technique, We are using a webcam to detect peoples faces and to prevent virus transmission.
- It has fast and high accuracy
- This system can be implemented in ATMs, Banks, Hospitals etc
- We can keep people safe from our technique.

CHAPTER 2

REQUIREMENT ANALYSIS

2.1 FUNCTIONAL REQUIREMENTS

2.1.1 Functional Requirements of Face Mask Dataset:

R1. The system must have an unbiased ‘with_mask’ dataset.

R2. The dataset must have over 1500+ images in both ‘with_mast’ and ‘without_mask’ classes.

R3. The dataset must not re-use the same images in training and testing phases.

2.1.2 Functional Requirements of Face Mask Detector

R4. The system must be correctly able to load the face mask classifier model.

R5. The system must be able to detect faces in images or video streams.

R6. The system must be able to extract each face's Region of Interest (ROI).

R7. There must not be any object between the system and the face of the user for a successful face detection and hence the face mask detection.

R8. The end position of the face must be fit inside the webcam frame and must be closer to the camera.

R9. Correctly able to detect masks in ‘png’, ‘jpg’. ‘jpeg’, and ‘gif format images.

R10. The system must be able to detect face masks on human faces on every frame in a live video.

R11. The results must be viewed by showing the probability along with the output of ‘Mask’ or ‘No Mask’.

2.2 NON FUNCTIONAL REQUIREMENTS

2.2.1 Product Operation

R1. The face should be localized by detecting the facial landmarks and the background must be ignored.

R2. The system will be implemented in Python script with an accuracy of the model of over 90%.

R3. The user must not move his/her face out of camera's sight in order to get correct results.

R4. The background must not be too bright or too dark while detecting the face mask.

2.2.2 Product revision

R5. The system must be portable and can be applied to embedded devices with limited computational capacity (ex., Raspberry Pi, Google Coral, NVIDIA Jetson Nano, etc.).

R6. The output response operation must be fast and under 5 seconds per person.

R7. The system must be able to correctly detect more than one face if present, and hence the presence of a mask in the frame.

2.2.3 Product transition

R8. The system should be easy for usability and self-descriptive for maintenance purposes.

R9. The system must be platform independent and flexible for updates.

2.3 SOFTWARE REQUIREMENTS

A. Operating System - Windows(7 or above)
B. Language - Python 3
C. IDLE - PyCharm or any python editors

2.4 HARDWARE SPECIFICATIONS

A. Processor - Intel core I3 or above
B. Hard disk - 16GB or above
C. Raspberrypi - Model B or above
D. Raspberrypi Camera - Module 2 or above
E. Speakers for alerting

CHAPTER 3
DESIGN

3.1 SYSTEM ARCHITECTURE

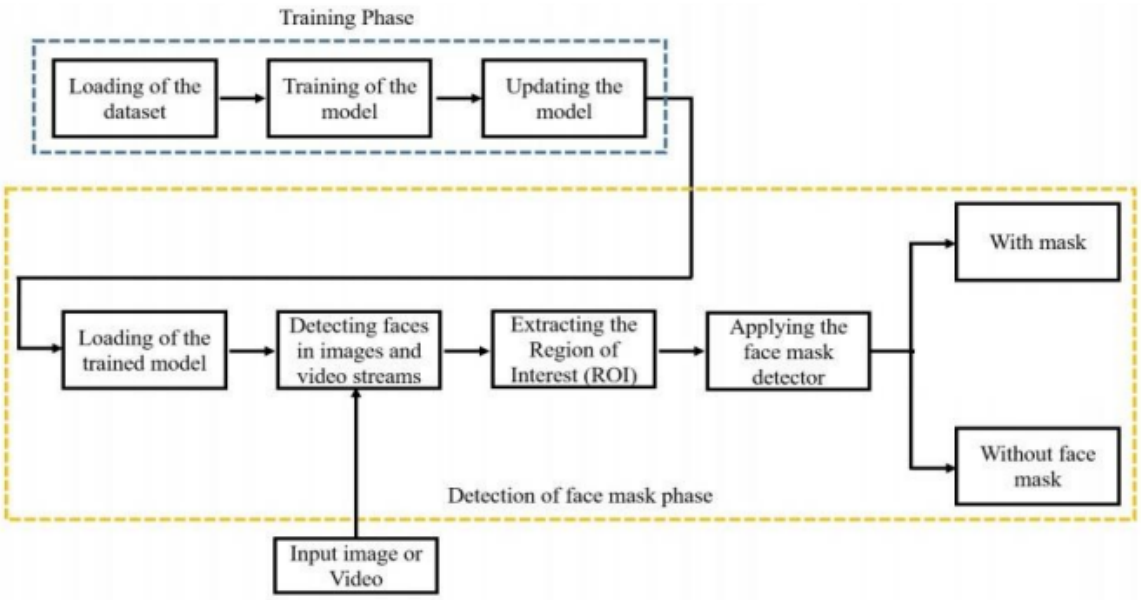


Fig: 3.1.1 System Architecture

Load the dataset which contains images of faces with masks and without masks. Train the Face Mask Detection Classifier on Image Dataset using the MobileNetV2 model which is compatible with any embedded devices. Save the Face Detection Classifier Model. Now load the trained model. Use a Live webcam Video stream to detect the face and extract the ROI(Region of Interest of the Face). Apply the face mask Classifier to each face ROI to determine whether the person is wearing a mask or not. If the person is not wearing a mask then it will give an alert.

3.2 UML DIAGRAMS

The Unified Modeling Language (UML) is a standard language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling

of large and complex systems. The UML is a very important part of developing objects oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. Using the UML helps project teams communicate, explore potential designs, and validate the architectural design of the software.

Goals Of UML:

The primary goals in the design of the UML were:

1. Provide users with a ready-to-use, expressive visual modeling language so they can develop and exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of programming languages and development processes.
4. Provide a formal basis for understanding the modelling language.
5. Encourage the growth of the OO tools market.
6. Support higher-level development concepts such as collaborations, frameworks, patterns and components. Integrate best practices.

3.2.1 Use Case Diagram

A use case diagram is the primary form of system/software requirements for a new software program underdeveloped. Use cases specify the expected behavior (what), and not the exact method of making it happen (how). A key concept of use case modelling is that it helps us design a system from the end user's perspective. It is an effective technique for communicating system behavior in the user's terms by specifying all externally visible system behavior.

3.2.1.1 Use Case Diagram captures:

- It only summarizes some of the relationships between use cases, actors, and systems.
- It does not show the order in which steps are performed to achieve the goals of each use case.

3.2.1.2 Purpose of Use Case Diagram:

- Specify the context of a system
- Capture the requirements of a system
- Validate a systems architecture
- Drive implementation and generate test cases
- Developed by analysts together with domain experts

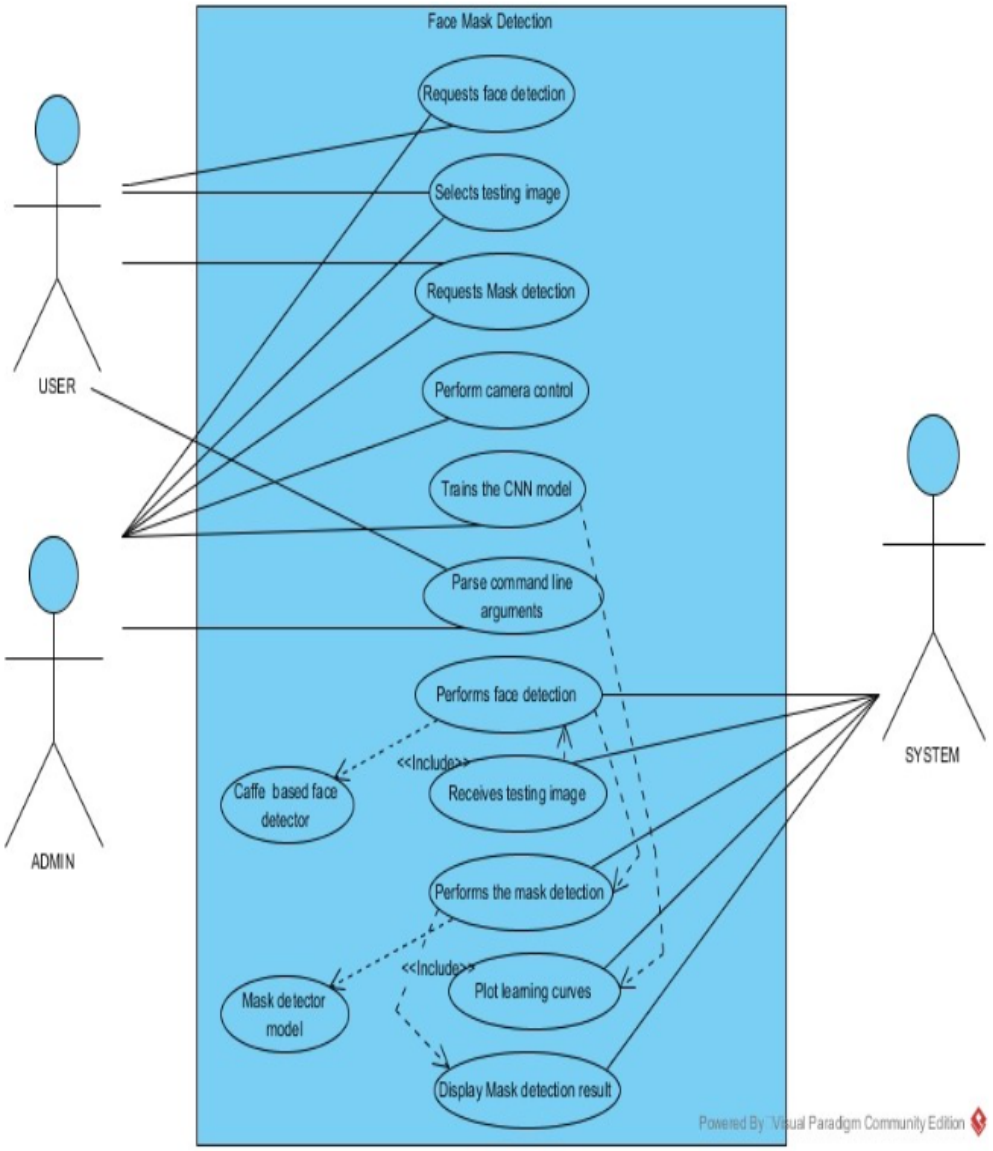


Fig 3.2.1.2.1 Use Case Diagram

3.2.2 Class Diagram

Class diagrams are the main building blocks of every object-oriented method. The class diagram can be used to show the classes, relationships, interface, association, and collaboration. Class diagrams are a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

3.2.2.1 Purpose of Class Diagram:

- Shows static structure of classifiers in a system
- Diagram provides a basic notation for other structure diagrams prescribed by UML
- Helpful for developers and other team members too
- Business Analysts can use class diagrams to model systems from a business perspective

3.2.2.2 A UML class diagram is made up of:

- A set of classes
- A set of relationships between classes

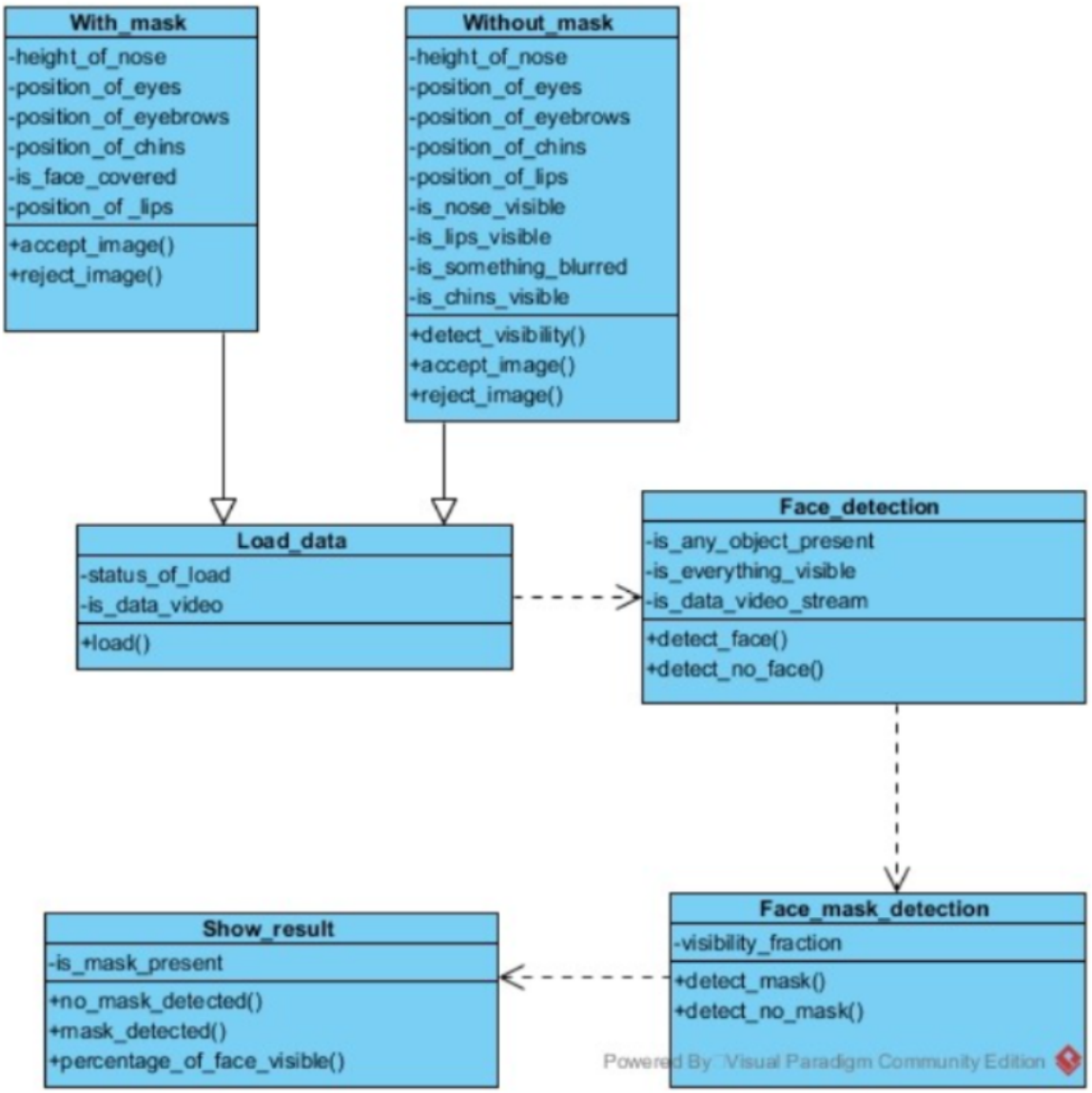


Fig 3.2.2.2.1 Class Diagram

3.2.3 Activity Diagram

Activity Diagrams describe how activities are coordinated to provide a service which can be at different levels of abstraction. Typically, an event needs to be achieved by some operations, particularly where the operation is intended to achieve a number of different things that require coordination, or how the events in a single use case relate to one another, in particular, use cases where activities may overlap and require coordination.

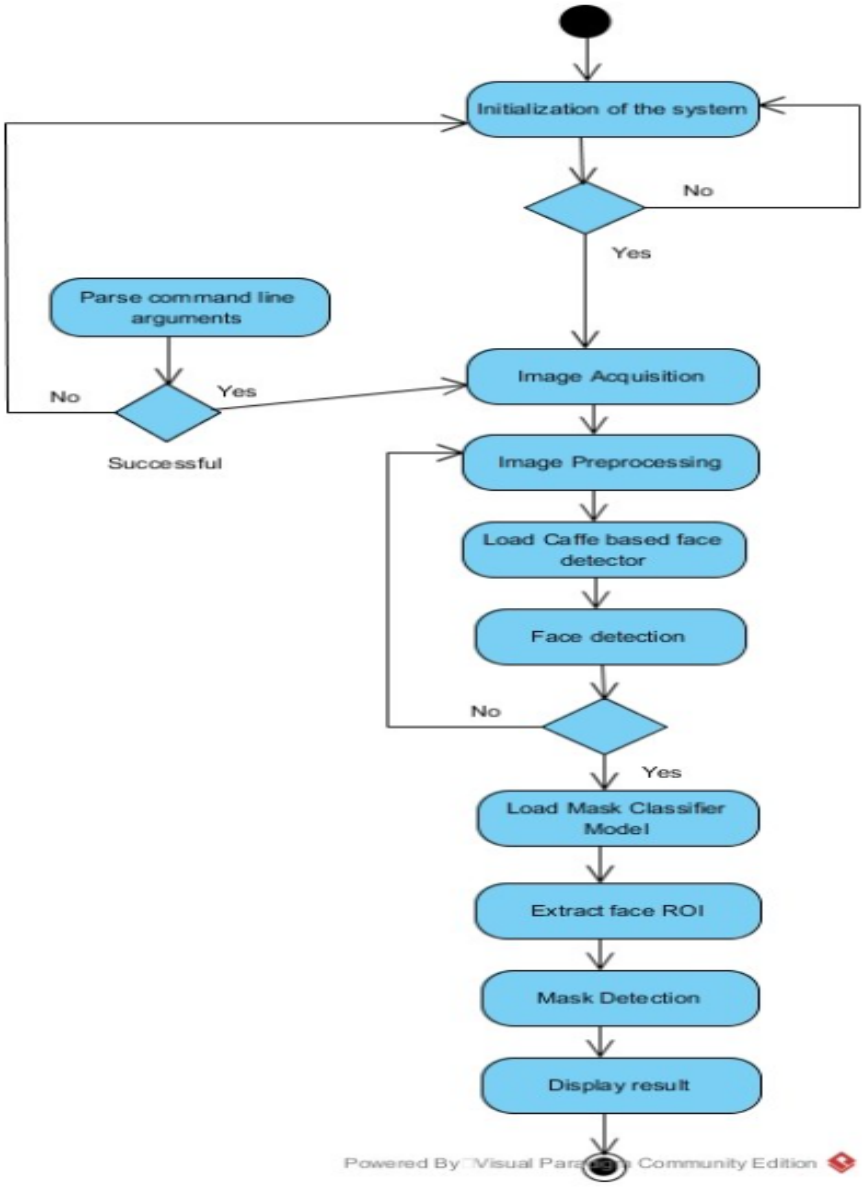
3.2.3.1 Purpose of Activity Diagram:

- Draw the activity flow of a system.

- Describe the sequence from one activity to another.
- Describe the parallel, branched and concurrent flow of the system.

3.2.3.1 Activity diagram can be used for:

- Modeling workflow by using activities.
- Modeling business requirements.
- High level understanding of the system's functionalities.
- Investigating business requirements at a later stage.



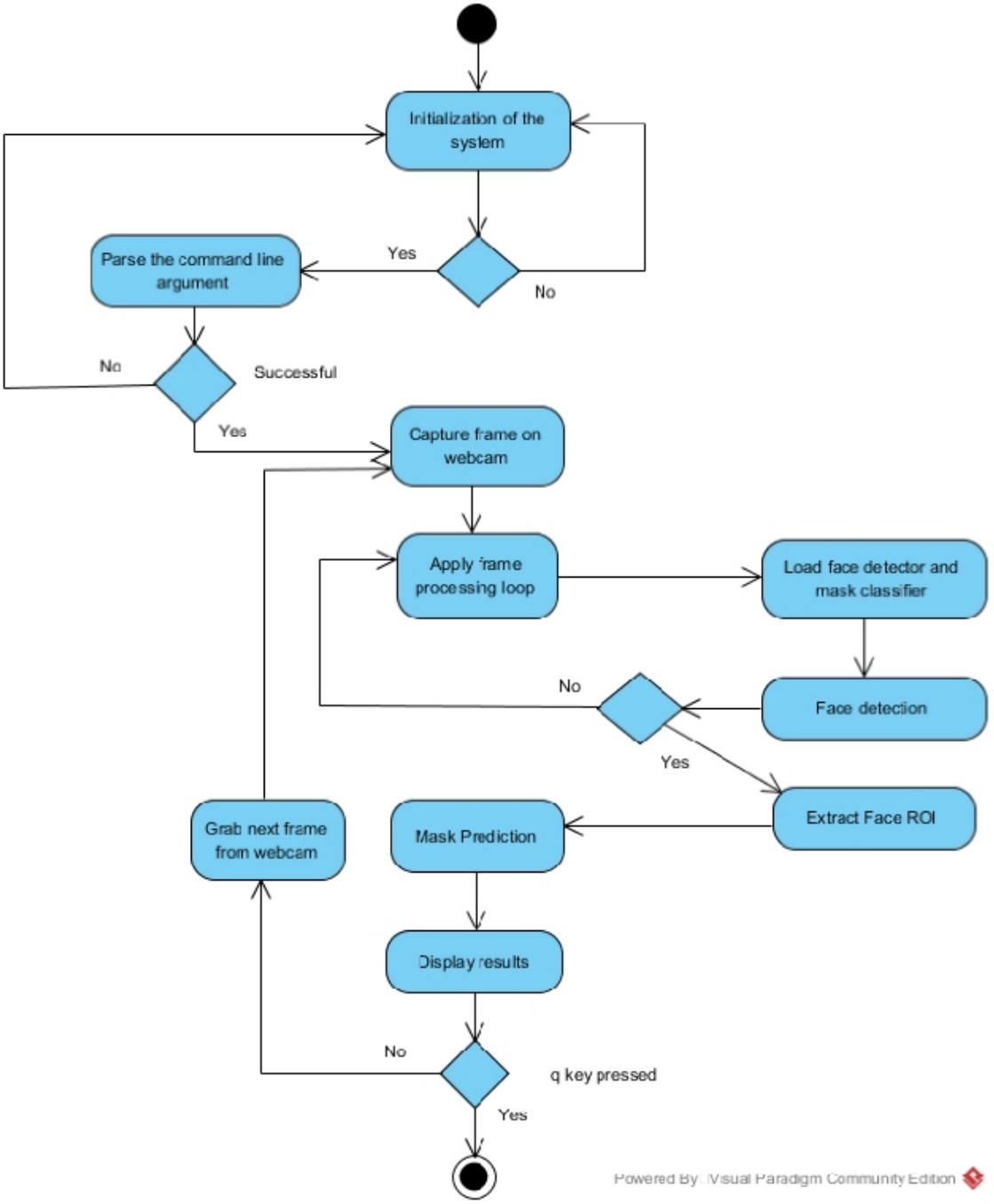


Fig 3.2.3.1 Activity Diagram

3.2.4 Sequence Diagram

Sequence Diagrams are interaction diagrams that detail how operations are carried out. They capture the interaction between objects in the context of a collaboration. Sequence Diagrams are time focus and they show the order of the interaction visually by using the vertical axis of the diagram to represent time what messages are sent and when.

3.2.4.1 Sequence Diagrams captures:

- The interaction that takes place in a collaboration that either realizes a use case or an operation (instance diagrams or generic diagrams)
- High-level interactions between user of the system and the system, between the system and other systems. or between subsystems (sometimes known as system sequence diagrams)

3.2.4.2 Purpose of Sequence Diagram

- Model high-level interaction between active objects in a system
- Model the interaction between object instances within a collaboration that realizes a use case
- Model the interaction between objects within a collaboration that realizes an operation
- Either model generic interactions (showing all possible paths through the interaction) or specific instances of a interaction (showing just one path through the interaction)

3.2.4.3 Sequence Diagram at a Glance

Sequence Diagrams show elements as they interact over time and they are organized according to object (horizontally) and time (vertically). Like all other diagrams, sequence diagrams may also contain notes and constrains.

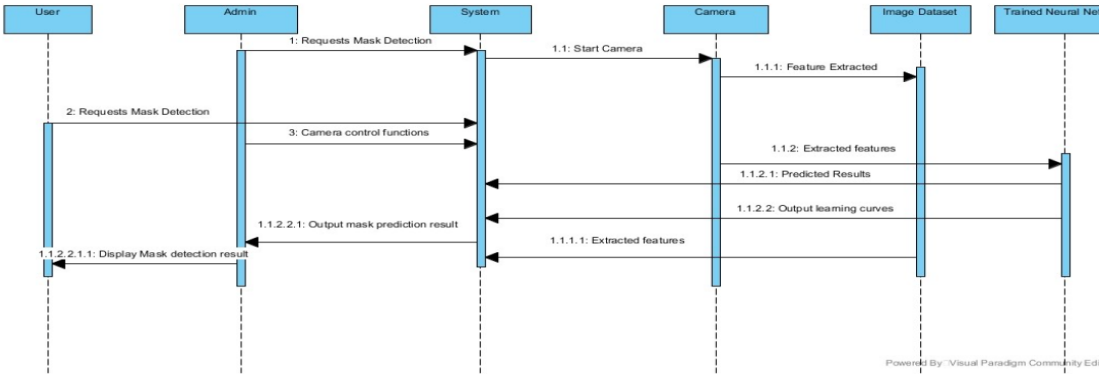


Fig 3.2.4.3.1 Sequence Diagram

CHAPTER 4

IMPLEMENTATION

4.1 TECHNOLOGIES DESCRIPTION:

Python:

Python is an interpreted high-level programming language for general- purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. C Python is managed by the non-profit Python Software Foundation.

Transfer Learning:

Transfer learning is the field of machine learning in which it helps in recognizing and detecting.As our project deals with face mask detection we use the transfer learning.When we want train the model over the large datasets we use transfer learning.

In transfer learning, we first train a base network on a base dataset and task, and then we repurpose the learned features, or transfer them, to a second target network to be trained on a target dataset and task. This process will tend to work if the features are general, meaning suitable to both base and target tasks, instead of specific to the base task.

MobileNetV2 :

MobileNetV2 is a convolutional neural network architecture that seeks to perform well on mobile devices. It is based on an inverted residual structure where the residual connections are between the bottleneck layers.The main benefit is it has low memory requirement and it is lightweight.The model has the faster execution and the accurate results.

MobileNet-v2 hass 53 layers deep. You can load a pre trained version of the network trained on more than a million images from the ImageNet database . The pretrained network can classify images into 1000 object categories, such as keyboard, mouse, pencil, and many animals. As a result, the network has learned rich feature representations for a wide range of images. The network has an image input size of 224-by-224.

MobileNetV2 is a very effective feature extractor for object detection and segmentation. For example, for detection when paired with the newly introduced SSDLite the new model is about 35% faster with the same accuracy than MobileNetV1

Libraries:

Numpy:

Numpy is a python library “Numerical python” which is used of working with arrays as well as variations such as masks and matrices, which can be used for various math operations.It is also useful in linear algebra, random number capability etc.

Scipy:

SciPy is a scientific computation library that uses NumPy underneath. SciPy stands for Scientific Python. It provides more utility functions for optimization, stats and signal processing

Keras:

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library.It cannot handle low-level computations, so it makes use of the backend library to resolve it. The backend library act as a high-level API wrapper for the low-level API, which lets it run on TensorFlow, CNTK, or Theano.

Matplot:

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.Pyplot is a plotting library used for 2D graphics in python programming language. It can be used in python scripts, shell, web application servers and other graphical user interface toolkits.

OpenCV:

OpenCV is a huge open-source library for computer vision,machine learning, and image processingIt can process images and videos to identify objects, faces, or even the handwriting of a human. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection .

Imutlis:

Imutils are a series of convenience functions to make basic image processing functions such as translation, rotation, resizing, skeletonization, and displaying Matplotlib images easier with OpenCV and both Python 2.7 and Python 3 .

4.2 INSTALLATION STEPS:

4.2.1 Installation of Python*:*

Step 1: Select Version of Python to Install.

Step 2: Download Python Executable Installer.

Step 3: Run Executable Installer.

Step 4: Verify Python was Installed.

Step 5: Verify Pip was Installed.

- All libraries use “pip install command” in command prompt.

4.3 PROCEDURE FOR EXECUTION

4.3.1 Project directory

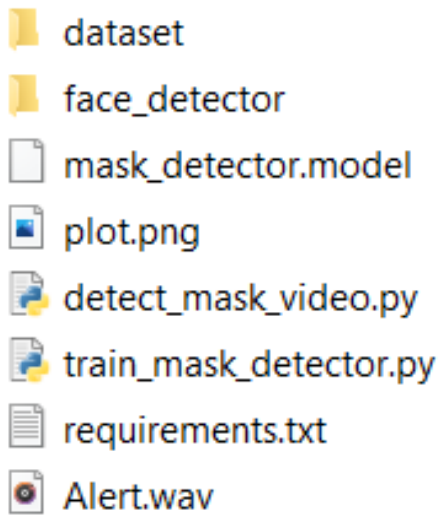


Fig 4.3.1.1 Project Directory

I. dataset: This folder has training dataset that contains images of faces with mask and without mask.

II. face_detector: It contains the models that will recognize the faces in the video feed.

III. mask_detector.model: After completion of training, the face mask detector model is saved into this file. This file is used for detecting face masks.

IV. plot.png: This file contains visualization of accuracy and loss rates in each iteration while training.

V. detect_mask_video.py: This python file contains the code that will take video feed, detect faces, and apply mask detector model to the faces to get results.

VI. train_mask_detector.py: This python file has the logic for training the datasets using transfer learning and MobileNetV2 model. After completion of training, the model is saved in mask_detector.model file.

VII. requirements.txt: This file contains all the libraries required to run the model including version numbers. You need to install each library in this file using the command “pip requirements.txt” in the terminal.

VIII. Alert.wav: This is the audio file for generating alert whenever the model recognizes a person not wearing a mask

Virtual Environment :

- Python applications will often use packages and modules that don't come as part of the standard library. Applications will sometimes need a specific version of a library, because the application may require that a particular bug has been fixed or the application may be written using an obsolete version of the library's interface.
- This means it may not be possible for one Python installation to meet the requirements of every application. If application A needs version 1.0 of a particular module but application B needs version 2.0, then the requirements are in conflict and installing either version 1.0 or 2.0 will leave one application unable to run.
- The solution for this problem is to create a virtual environment

4.3.2 Steps for execution:

- Goto project directory in the terminal.
- Now run "**python3 -m venv maskDetector-env"** command in the command prompt which will create the "maskDetector-env" directory if it doesn't exist, and also create directories inside it containing a copy of the Python interpreter.
- Run "**tutorial-env\Scripts\activate.bat**" command in the terminal which will activate the virtual environment that is created.
- Now run "**pip requirement.txt**" in the terminal to install required libraries to run the model.
- Now run "**python train_mask_detector.py"** in the terminal to train the face mask model. This will generate two files plot.png and mask_detector.model. This file needs to be run only once for the first time to generate the model.
- Now run "**detect_mask_video.py"** in the terminal to identify if a person is wearing a mask or not and generate alerts accordingly.
- To stop the process enter 'q' which will end the execution.
- After execution is done, we have to deactivate the virtual environment using the **"deactivate.bat"** command.

CHAPTER 5

TESTING

5.1 TEST PLANS

A test plan documents strategy that will be used to verify and ensure that a product or system meets its design specification and other requirements. A test plan is usually prepared by or with significant input from the engineer.This document describes the plans for testing the architectural prototype of System. In my Project The Trained Model has to be tested to get the Desired Output.I use Different Classes of images for testing the system.

5.2 UNIT TESTING

In computer programming, unit testing is a software testing method by which individual units of source code, sets of one or more computer program modules together with associated control data, usage procedures, and operating procedures, are tested to determine whether they are fit for use. In our system,

• Test the preprocessing module work properly to preprocess the dataset

• Test to check whether the training of the images work properly.

• Test to check whether the model recognizes the face mask Accurately.

5.3 INTEGRATION TESTING

Integration testing (sometimes called integration and testing) is the phase in software testing in which individual software modules are combined and tested as a group. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

• Check whether the model takes the input image.

• Check whether the System plots the features of a given image.

• Check whether the model recognizes the face mask accurately

5.4 SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.

5.5 BLACK BOX TESTING

In this testing by knowing the internal operation of a product, tests can be conducted to ensure that "all gears mesh" that is the internal operation performs according to specification and all internal components have been adequately exercised .It fundamentally focuses on the functional requirements of the software. The steps involved in the black box testing case design are

- Graph Based testing methods
- Equivalence partitioning
- Boundary value Analysis
- Comparison testing & Test results

CHAPTER 6

SCREENSHOTS

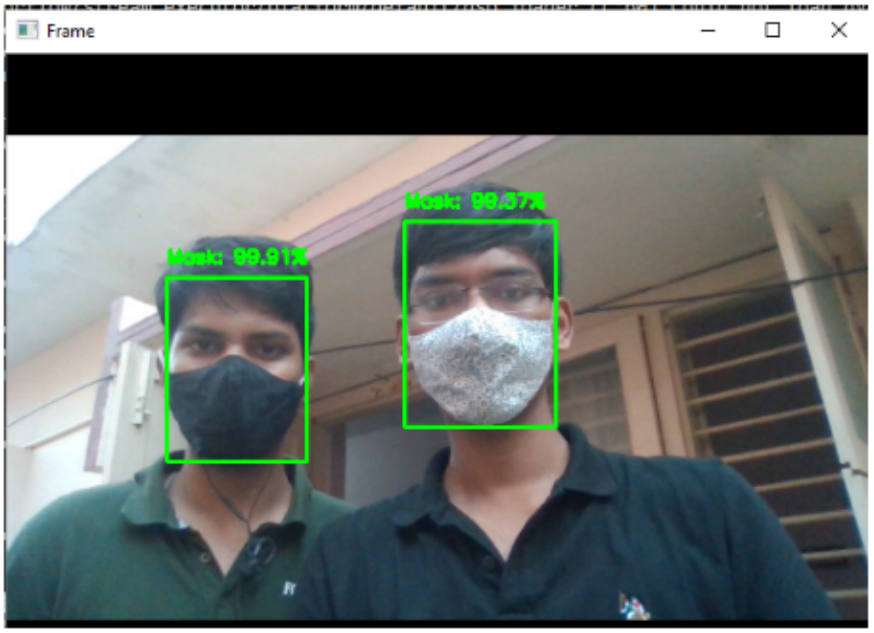


Fig 6.1 Persons are wearing the masks

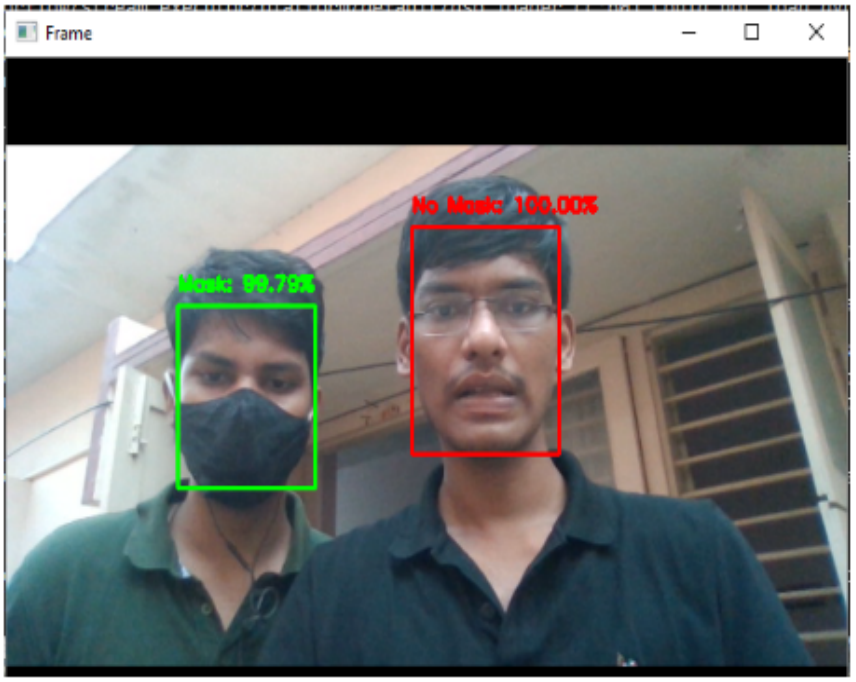


Fig 6.2 Person is not wearing a mask. In this case an alert is generated.

CHAPTER 7

CONCLUSION AND FUTURE SCOPE

7.1 Conclusion:

The model so developed is related to the detection of face masks being put on by the people, from the images as well as from live video streams. On training the model using OpenCV, Keras and Tensorflow an accuracy of 94% was achieved. Post training, the classifier was subjected to images and live video streams. The faces were recognized in the images and videos and the faces were extracted. Subsequently, the face mask classifier was applied and the required results were obtained. Also the model so developed is least complicated in structure and gives instantaneous results and hence can be deployed or used at various places.

7.2 Future Scope:

We can enhance the model by adding social distancing to the current model which can identify whether people are maintaining required distancing(6 feet).

REFERENCES

[1]-https://www.investopedia.com/terms/d/deep-learning.asp

[2]-https://medium.com/softway-blog/building-a-facial-recognition-machine-learning-model-using-tensorflow-6e62fb349794.

[3]-https://www.pyimagesearch.com/2020/05/04/covid-19-face-mask-detector-with-opencv-keras-tensorflow-and-deep-learning/

GUDLAVALLERU ENGINEERING COLLEGE

(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
SESHADRI RAO KNOWLEDGE VILLAGE, GUDLAVALLERU

Department of Computer Science and Engineering

Program Outcomes (POs)

Engineering Graduates will be able to:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions., component, or software to meet the desired needs.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. **Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. **Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO1: Design, develop, test and maintain reliable software systems and intelligent systems.

PSO2: Design and develop web sites, web apps and mobile apps.

PROJECT PROFORMA

Classification of Project	Application	Product	Research	Review
	√			

Note: Tick Appropriate category.

Project Outcomes	
Outcome 1	Synthesize and apply prior knowledge of mathematics, computer science and engineering to design and implement solution to open-ended problems
Outcome 2	Design engineering solution to problems utilizing a system approach.
Outcome 3	Use different tools for design, implementation, testing, Data transformation and Documentation.
Outcome 4	Develop better interpersonal communication skills, team work and leadership quality.
Outcome 5	Acquire writing and oral presentation skills.

Mapping Table

Project Outcomes	Program Outcomes (POs)												PSOs	
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
Outcome 1	3	2	2	2	2	2	2	2	2	2	3	2	2	2
Outcome 2	3	1	2	2	1	3	3	3	3	2	2	3	2	2
Outcome 3	2	1	1	2	3	2	2	3	3	3	2	1	2	2
Outcome 4	2	2	1	1	1	2	1	1	3	2	2	1	1	2
Outcome 5	2	2	1	1	1	2	2	1	2	3	2	2	1	1

Note: Map each project outcomes with POs and PSOs with either 1 or 2 or 3 based on level of mapping as follows:

1-Slightly (Low) mapped 2-Moderately (Medium) mapped 3-Substantially (High) mapped