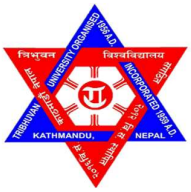


Tribhuvan University

Faculty of Humanities and Social Science

FACE MASK DETECTION IN OPENCV

A PROJECT REPORT

Submitted to

Department of Computer Application

Kathmandu BernHardt College

In partial fulfillment of requirements for the Bachelors in Computer Application

Submitted by:

Name	Roll No.
Ashish Poudel	4610520

September, 2022

Under supervision of

Kumar Prasun

ABSTRACT

The pandemic and the variants have affected millions of people globally. With no effective cure, prevention is the only solution to slow down the rapid transmission of the virus. Face masks play a vital role in prevention of the transmission. Real time detection of proper face masks is crucial in this time. This system helps in identifying proper face mask automatically compared to the manual detection. COVID-19 epidemic has swiftly disrupted our day-to-day lives affecting the international trade and movements. Wearing a face mask to protect one's face has become the new normal. In the near future, many public service providers will expect the clients to wear masks appropriately to partake of their services. Therefore, face mask detection has become a critical duty to aid worldwide civilization. The suggested technique successfully recognizes the face in the image or video and then determines whether or not it has a mask on it. As a surveillance job performer, it can also recognize a face together with a mask in motion as well as in a video. The technique attains excellent accuracy. It investigates optimal parameter values for the Convolutional Neural Network model in order to identify the existence of masks accurately without generating over-fitting. Based on the count of persons wearing and not wearing face masks the status is obtained. Depending upon the status warning is done by means of using buzzer.

Keywords: *VS Code, Desktop App, Algorithm, CNN, OpenCV*

ACKNOWLEDGEMENT

I would like to express our sincere gratitude to Project Supervisor **Mr. Kumar Prasun** and mentor sir, he continuously helped and guided us through different problems faced during this whole documentation process. Without his support, it would have been difficult to work on. The satisfaction and success of the completion of this task would be incomplete without heartfelt thanks to people whose constant guidance, support and encouragement made this work successful.

I would like to thank Lecturer **Mr. Sagar KC** and Head of Department **Mr. Ram Babu Mahato** sir for assisting in the different scope of the programs. I would like to thank Kathmandu BernHardt College for providing all the resources and ideal environment for the project developing process and encourage us to develop such a project. I would also like to share our gratitude to Tribhuvan University for providing the opportunity to evaluate our programming skills through this project.

I sincerely express our whole hearted thanks to the principal Mr. Sahadev Sigdel for his constant encouragement and moral support during the course of this project. I owe our sincere thanks for furnishing every essential facility for doing this project. I sincerely thank our guides for valuable help and guidance throughout the project. I wish to express our sincere thanks to the project coordinator and all staff members, Department of Computer Application for their valuable help and guidance rendered to us throughout the project. Above all I am grateful to all our classmates and friends for their friendly cooperation and their exhilarating company.

Ashish Poudel

6-2-461-55-2018

Table OF CONTENTS

Title **Page No.**

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
BC	Binary Classification
CASE	Computer Aided Software Engineering
CDC	Centre for Disease Control and Prevention
CNN	Convolutional Neural Network
GPU	Graphical Processing Unit
IOU	Intersection Over Union
MATLAB	Matrix Laboratory
MC	Multicast Classification
ML	Machine Learning
ODD	Object-Oriented Design
OpenCV	Open-Source Computer Vision Library
SciPy	Scientific Python
SDL	Software Development Lifecycle
SSD	System Sequence Diagram
SVM	Support Vector Machine
UML	Unified Modelling Language
YOLO	You Only Look Once

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Introduction

Face Mask Detection System is a system that detects whether or not the person is wearing face mask. Face detection is a key area in the field of Computer Vision and Pattern Recognition. In recent decades, facial recognition has become the object of research worldwide. In addition, with the advancement of technology and the rapid development of artificial intelligence, very significant advances have been made. For this reason, public, private companies and many other sectors use facial recognition systems to identify and control the access of people in airports, schools, offices, and other places. On the other hand, with the spread of the COVID-19 pandemic, government entities have established several biosafety regulations to limit infections. Among them is the mandatory use of face masks in public places, as they have been shown to be effective in protecting users and those around them [1]. 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. The proposed method used here is carried out in two steps. The first step is to train the face mask detector using transfer learning. The second step is to use this trained face mask detector on images or videos of people to identify if they are wearing a mask. Therefore, it has become necessity to develop such precise system. It is based on the real time facial images analysis for visualizing the person's face.

1.2 Problem Statement

As the spread of the virus occurs through physical contact, conventional recognition systems (such as fingerprints) or typing a password on a keyboard become insecure. Thus, facial recognition systems are the best option, as they do not require physical interaction as in other cases. However, the use of the face mask within these systems has represented a great challenge for artificial vision, because at the time of facial recognition, half of the face is covered and several essential data are lost. This clearly denotes the need to create algorithms that recognize a person when they are wearing a face mask. This has made it necessary to implement new strategies to achieve robustness in the current systems.

Nepal, along with rest of the world is struggling to get out of this virus attack and the government implemented lockdown for the long way. But still new variants are on rise and many people are getting out without a face mask this may increase the spread of covid-19. Centers for Disease Control and Prevention (CDC) has stated that " Survey Shows that 90

percent Nepalese are aware, but only 64 percent wearing a mask " [2]. This survey clearly points that people are aware but they are not wearing the mask due to some discomfort in wearing and carelessness. Hence, object detection techniques for identification of persons wearing and not wearing facemasks are being used. The real time images from the camera are compared with the trained dataset and detection of wearing or not wearing a mask is done. The trained dataset is made by using machine learning technique which is the deciding factor of the result. The algorithm created by means of using a trained dataset will find the persons with and without wearing face masks.

1.3 Objectives

The objectives of project are:

- To develop an object detection method from real-time video streams
- To localizing the person who violates the health norms in public areas
- To optimize trained model faster & accurately in order to keep people alert

1.4 Scope and Limitations

1.4.1 Scope of Project

- The system is useful for academic institutions, hospitals & commercial organizations
- Automatic alert system makes people alert to wear mask.
- Reduce spread of air borne diseases.
- Mass surveillance improves smarter border control

1.4.2 Limitations

There are some limitations related to the project:

- It cannot correctly classify partially hidden faces.
- The irregularities in images, like those with insufficient light need proper attention.

1.5 Development Methodology

Considering that the proportion of face masks in the image is often not uniform in size, they often do not occupy the whole image. The features at the highest level have relatively wealthy semantic information, the candidate region's features can only be obtained by pooling the target region in the last convolution layer. Therefore, in order to solve the problem of multi-scale detection, the feature pyramid network / waterfall model is introduced that takes a single-scale image of an arbitrary size as input, and outputs proportionally sized feature maps at multiple levels.

Dataset Collection: The dataset was collected from Kaggle Repository and was split into training and testing data after its analysis.

Training a model to detect face masks: A default OpenCV module was used to obtain faces followed by training a Keras model to identify face mask.

Detecting the person not wearing a mask: A open CV model was trained to detect the names of the people who are not wearing masks by referring the database.

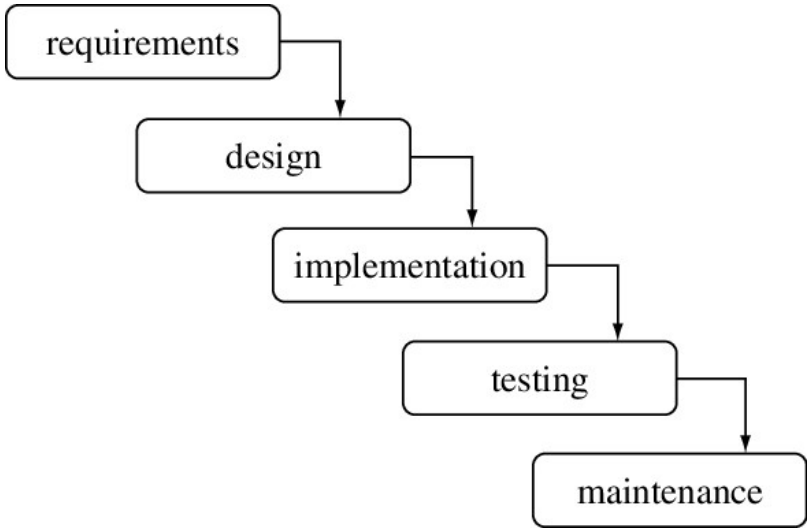


Figure 1: Waterfall Model in Face Mask Detection

Some situations where the use of Waterfall model is most appropriate in this project are: -

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

1.6 Organization of Report

The material presented in the project is organized into five chapters.

Chapter 1 represents the problem statement, objectives, scope and limitations of the project.

Chapter 2 describes the fundamental theories and concepts as well as information about existing system, journals and references.

Chapter 3 summarizes the keynote on system analysis and design where description of use case diagram, performance and reliability, different feasibility analysis, diagrams, database as well as architectural design are set out.

Chapter 4 provides an account on implementation and testing, tools used for preparation of the project. Test cases for unit testing as well as integration testing are done. Implementation details of modules are traced.

Chapter 5 presents brief summaries of outcome of the project, conclusion, reviews as well as future recommendations, improvements that can be done on upcoming days and feedback of systems, stability of the project. The implementation of web-based application in the project are described.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

In existing research, the most frequently used methods for face mask detection and masked face recognition are transfer learning of pre-trained deep learning models and support vector machine (SVM) [3]. But it takes long training time and huge memory requirements for a large dataset. Whereas in transfer learning, overfitting and negative transfer are the most alarming limitations. Also, due to the lack of accurate labelled datasets, masked facial recognition is very challenging. To resolve these issues of transfer learning and SVM, a novel model is proposed that can be used to reliably detect face masks (binary classification) and recognize masked faces (multiclass classification).

Existing research works on face mask detection and masked facial recognition have certain limitations, i.e., the absence of a unified method and dataset to tackle both problems, low accuracy of masked face recognition, less uncovered face exposure that makes it difficult to capture enough facial landmarks, etc. The proposed technique is robust to variations in face angles, lightning conditions, gender, skin tone, age, types of masks, occlusions (glasses), etc.

In the authors have developed a method to identify how a person is wearing the face mask. They were able to classify three categories of facemask-wearing condition namely correct facemask-wearing, incorrect facemask- wearing, and no facemask-wearing. This method achieved over 98 % accuracy in detection. Nicolae-Cătălin Ristea, Radu Tudor Ionescu proposed a novel data augmentation approach for mask detection from speech. Original and translated utterances were changed over into spectrograms were given as inputs to a bunch of ResNet neural organizations with different depths. In the authors have employed a GAN-based network using two discriminators for the removal of face mask from a face and reconstruct the face without the face mask using the CelebA dataset.

2.2 Literature Review

For this project, research and reviews some of the related websites and applications are done. In 2001, Viola and Jones proposed a real-time object detection called the Viola-Jones object detection framework. This process can determine competitive object detection rates in real-time to solve various detection problems. Moreover, it was used primarily in face

detection. Although this algorithm was very slow in training, it could detect faces in real-time with impressive speed. The method divided the image into many smaller sub-regions and required checking many different positions and scales because an image had many faces of different sizes.

An effective and economic approach to the use of AI in a manufacturing setting to build a secure environment. Using a face mask detection dataset, the system will use Open CV to perform real-time face detection from a live stream from our webcam. Using Keras, Python, Tensorflow and Open CV, and, it will build a COVID-19 face mask detector with computer vision. Using computer vision and CNN, it is aimed to decide whether or not the person in the image or video streaming is wear a mask [4].

Existing works on face mask detection can be categorized into conventional machine learning (ML) methods, deep learning (DL) based methods, and hybrid methods. The deep learning-based methods were utilized in the majority of studies on face mask detection, while conventional ML-based methods by Nieto-Rodríguez are limited. In Nieto-Rodríguez et al, an automated system was designed that activates an alarm in the operational room when the healthcare workers do not wear the face mask. This system used the Viola and Jones face detector for face detection and Gentle AdaBoost for face mask detection [5].

In the content based image classification using deep learning, Joseph Redmon et.al proposed You Only Look Once (YOLO) algorithm for real time object detection.

Sanzidul Islam et.al 2020, gave a deep learning based assistive System to classify COVID-19 Face Mask which is implemented in rasbperrypi-3 [6].

Velantina et.al 2020, made an COVID-19 facemask detection by means of using Caffe model.

Senthilkumar et.al 2017, compared the two most frequently used ma-chine learning algorithms K-Nearest Neighbor and Support Vector Ma-chine in his work for face recognition.

Senthilkumar et.al 2018, proposed a new and fast approach for face recognition [7].

With this, based on the above literature surveys, an effective algorithm for face mask detection has been made. The details are elaborated in forthcoming chapters.

CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

In the object-oriented approach, the focus is on capturing the structure and behaviour of information systems into small modules that combines both data and process. The main aim of Object-Oriented Design (OOD) is to improve the quality and productivity of system analysis and design by making it more usable.

3.1.1 Requirement Analysis

Requirement analysis results in the specification of operational characteristics of software: indicates interface of software with other system elements and establishes constrains the software must meet. The requirement analysis is mainly categorized into two types functional and non-functional:

i. Functional Requirement

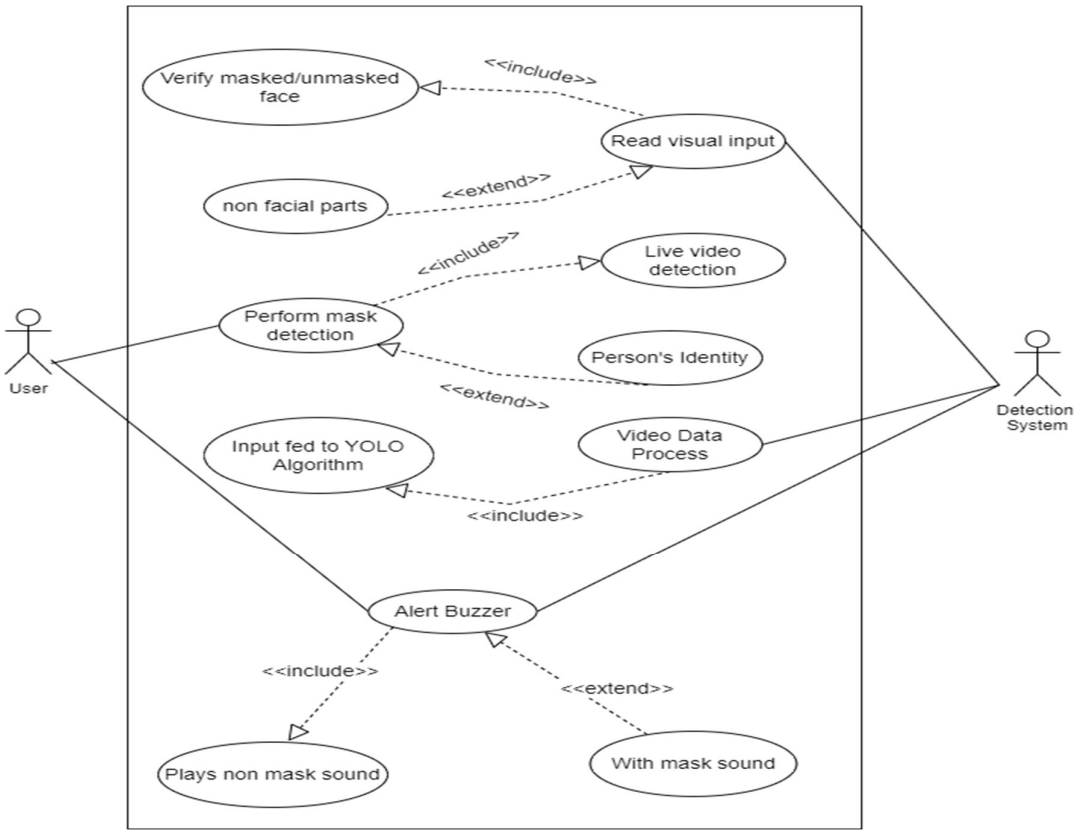


Figure 2: Use Case Diagram of Face Mask Detection System

The image captured is used in the face recognition process. This face image is analysed and considered as a high dimensional vector. This vector is then compared to all the face images in the dataset, looking for a match. Model parameters can be further optimized by retraining this already trained model on some new dataset.

- **Functional Requirements of Face Mask Dataset:**

➢ The system must have an unbiased 'with_mask' dataset.

➢ The dataset must have over 500+ images in both 'with_mask' and 'without_mask' classes.

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

- **Functional Requirements of Face Mask Detector:**

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

➢ The system must be able to detect faces in images or video stream.

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

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

➢ Correctly able to detect masks in 'png', 'jpg', 'jpeg', and 'gif' format images.

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

➢ The results must be viewed by showing the probability along with the output of 'Mask' or 'No Mask'.

ii. Non-functional Requirement

- **Adaptability:**

  If user faces video streaming display, he/she should first wear mask otherwise buzzer will ring. The capacity to be modified for a new use or purpose.

- The face should be localized by detecting the facial landmarks and the background must be ignored.
- The user must not move his/her face out of camera's sight in order to get correct results.
- **Portable:**

  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.).

- **Performance:**

  Depends on compatibility & architecture of device. Almost runs on all desktop devices.

- **Maintainability:**

  System is enabled of coordination which means the register information will maintained and taken into database system.

- **Usability:**

  It is easy to use the system

- **Security:**

  Secure and good privacy qualities

- **Compatibility:**

  The system runs on web platform of the operating system.

3.1.2 Feasibility Analysis

- **Technical Feasibility:**

  Technical feasibility assesses the current resources (hardware and software) and technologies, which are required to accomplish user requirements. It requires a computer with python anaconda installed. Today every organization has computer, so it is not an extra cost.

- **Economic Feasibility**

  In this project work, the system developed is a webcam application; which requires all the basic hardware and software support as required by other application. To integrate real surveillance may require software and manpower with developing skills. The proposed model is cost effective.

- **Schedule Feasibility**

  The project to be completed, realistic and achievable under a deadline according to strategy. It is developed within time limit. Hence, it is feasible in respective schedule.

- **Operational Feasibility**

  The proposed system performs effective outcome. It used to dig into the data and quickly conduct experiments to establish baseline performance on a task. The system recognizes a person without wearing a mask which requires no human involvements.

3.1.3 Object Modelling: Object & Class Diagram

Class diagram shows your classes and their relationships. An Object Model Diagram shows the interaction between objects at some point, during run time. A Class Diagram will show what the Objects in your system consist of (members) and what they are capable of doing (methods) mostly static.

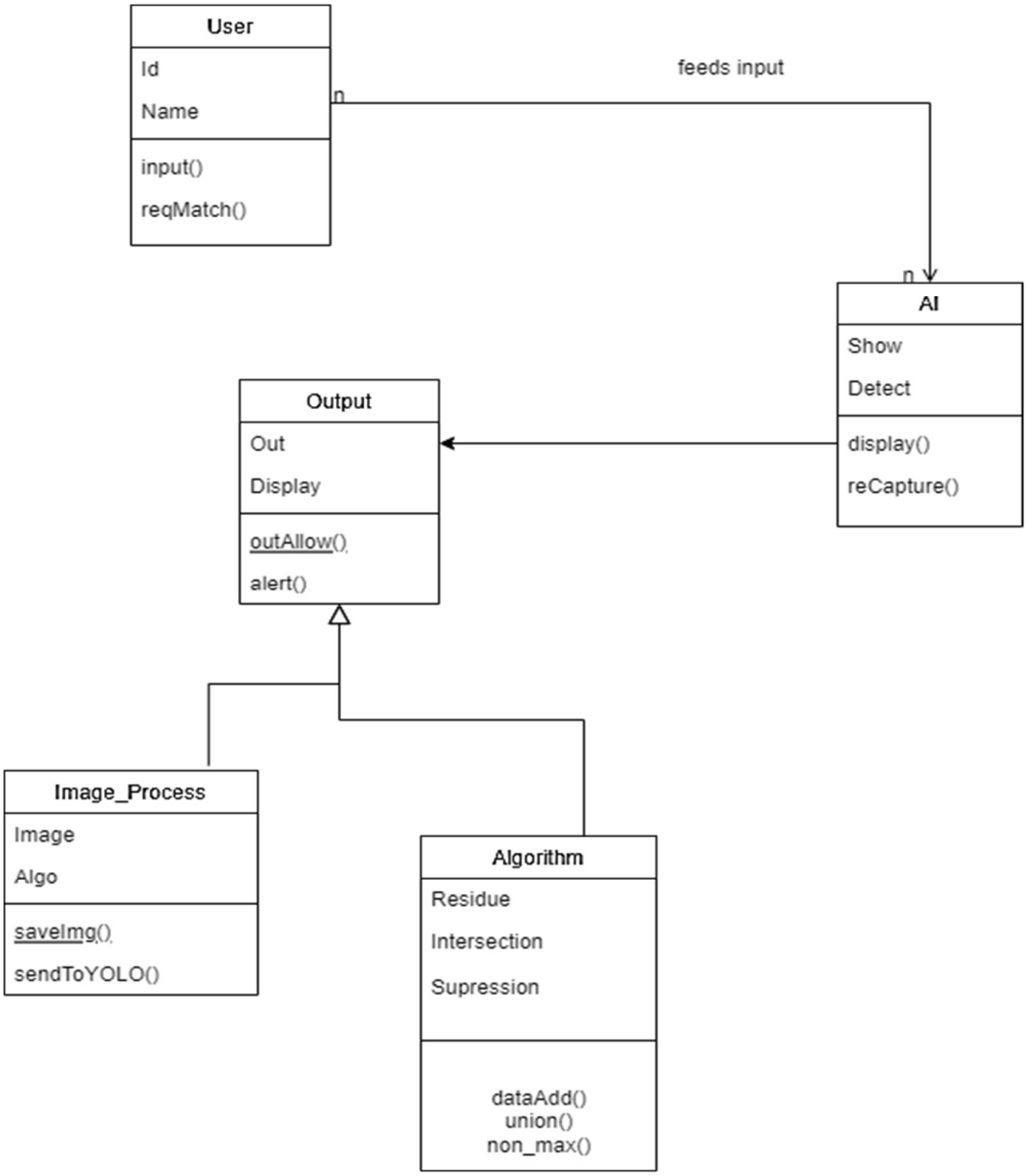


Figure 3: Class Diagram of Face Mask Detection System

3.1.4 Dynamic Modelling: State & Sequence Diagram

A sequence diagram or system sequence diagram (SSD) shows process interactions arranged in time sequence in the field of software engineering. It depicts the processes involved and the sequence of messages exchanged between the processes needed to carry out the functionality. 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.

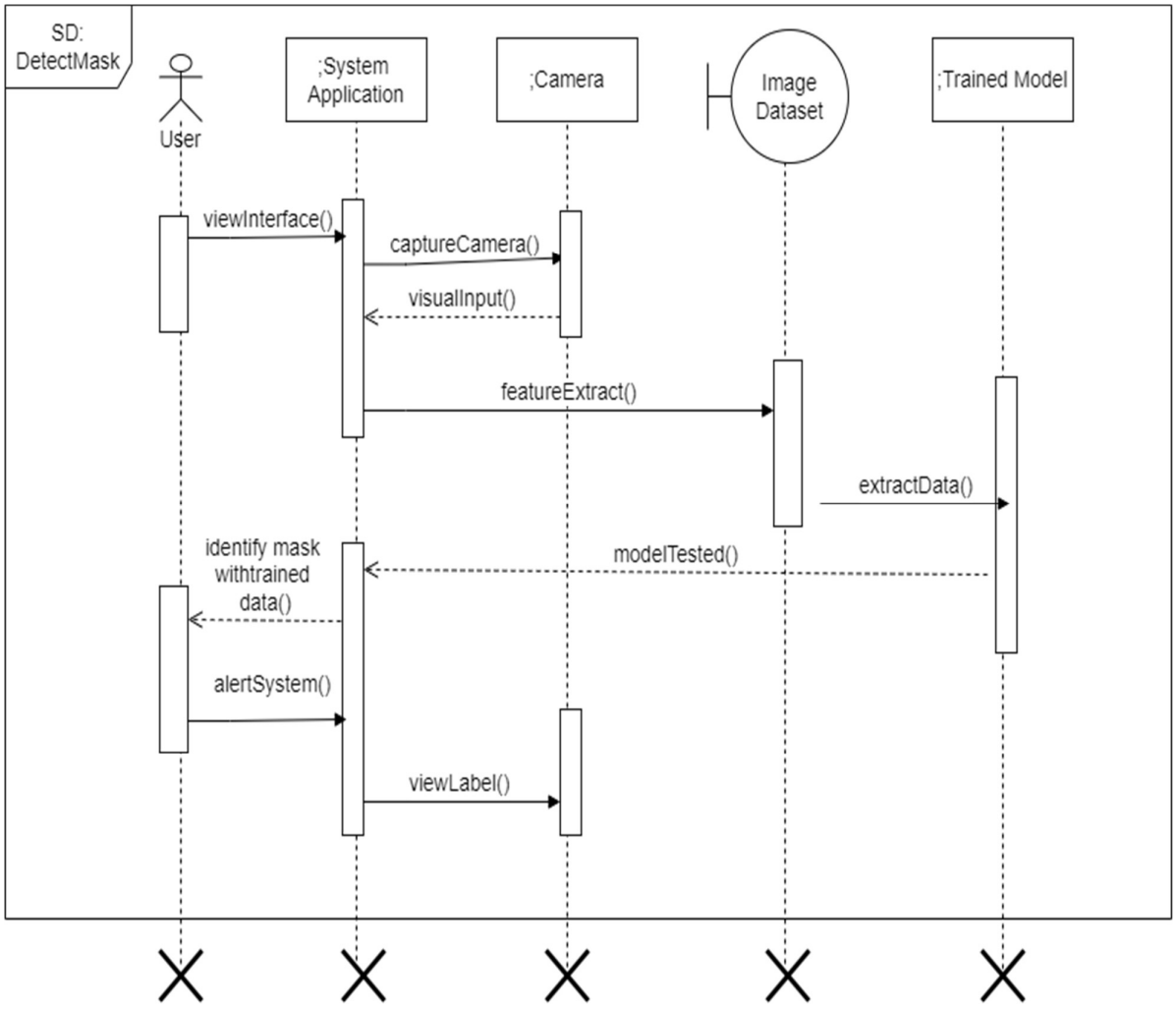


Figure 4: State & Sequence Diagram of Face Mask Detection System

3.1.5 Process Modelling: Activity Diagram

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. It describes 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. It is also suitable for modelling how a collection of use cases coordinates to represent business workflows. Model workflows between/within use cases. Model complex workflows in operations on objects. Model in detail complex activities in high-level activity. Activity diagrams are often used in business process modelling. They can also describe the steps in a use case diagram.

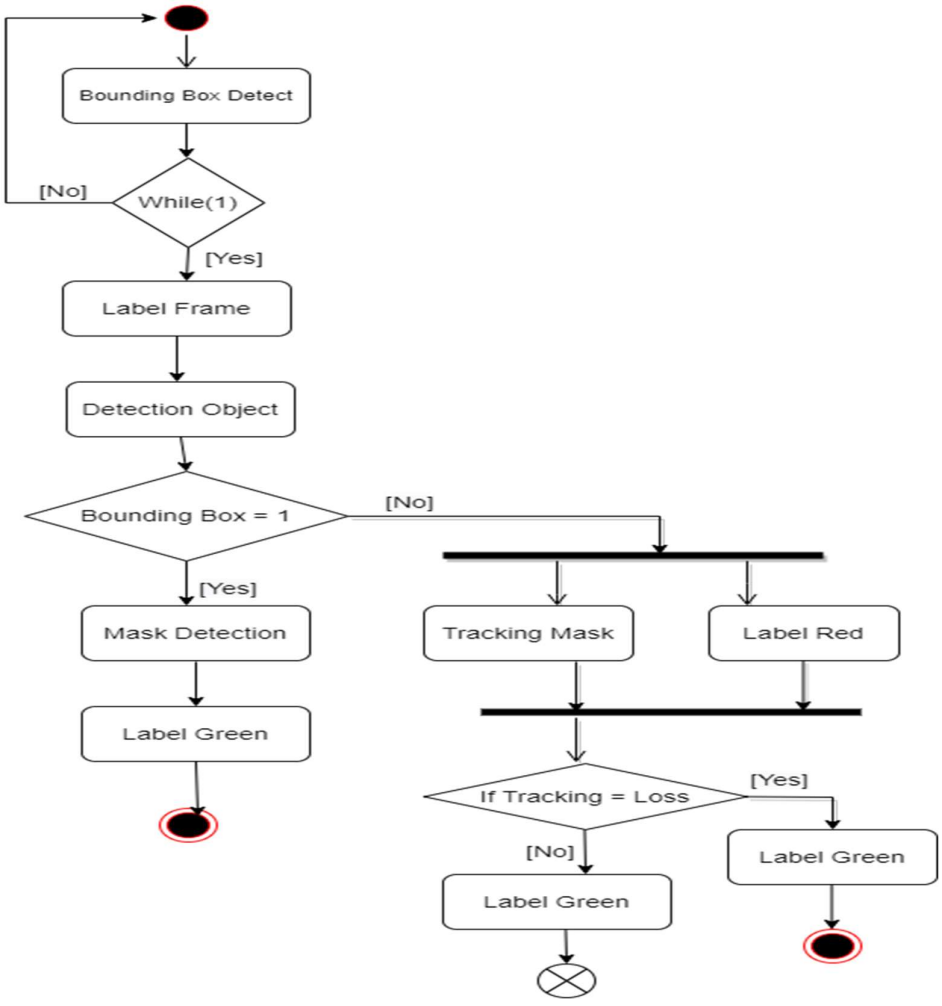


Figure 5: Activity Diagram of Face Mask Detection System

3.2 System Design

3.2.1 Refinement of Classes and Object

Generalisation is the refinement of a class into more refined classes. Generalisation allows the developer to model objects into hierarchical structures based on their similarities. The class being refined is called super-class and the refined versions of it are called sub-classes. Each sub-class inherits the attributes and operations from their super-class. Methods and attributes can then be refined and the sub-class also adds specific attributes and operations. A discriminator is a variable of enumeration type, which indicate which property of an object is being abstracted. The most important use of inheritance is the conceptual simplification it makes trough the reduction of independent features in the system. By defining a feature with the same name a subclass can override a superclass feature. Overriding is a process of refining and replacing the overridden feature. Generalisation is used for extension and restriction.

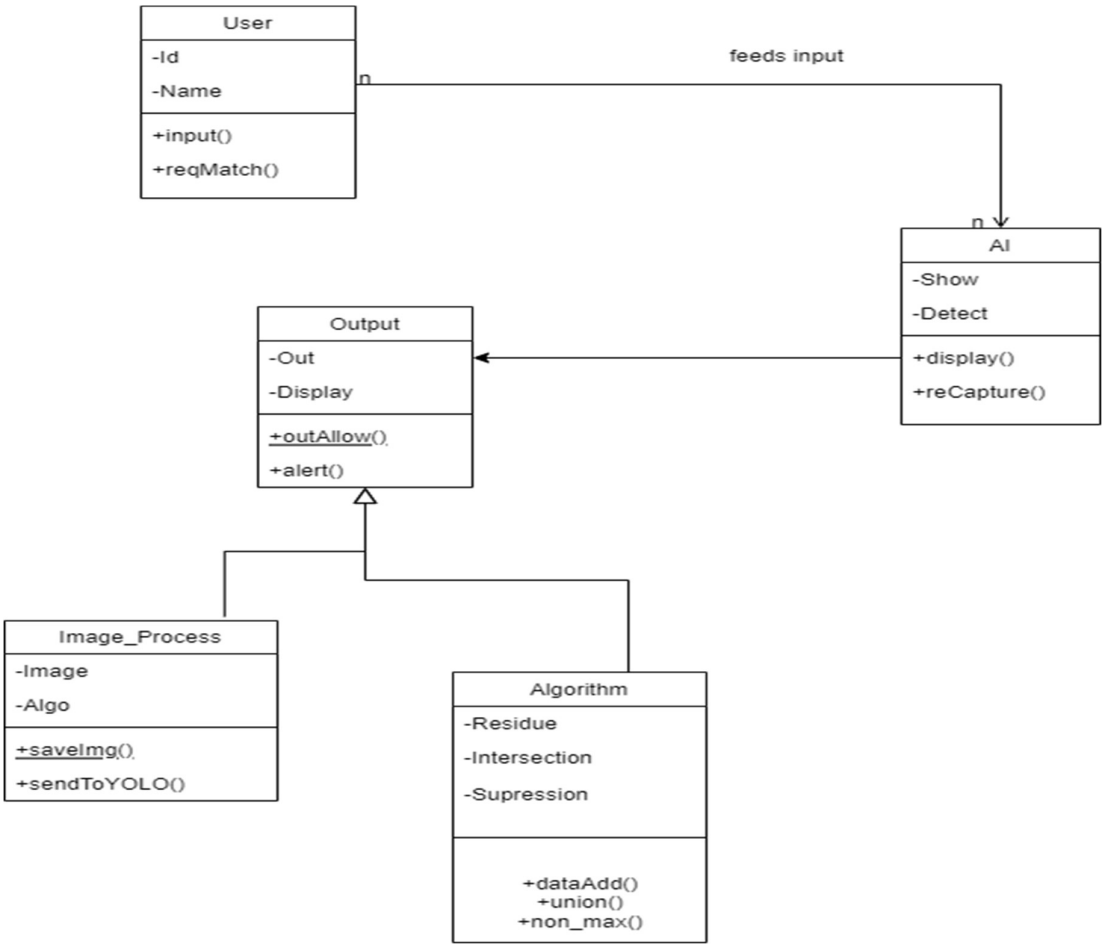


Figure 6: Physical Class Diagram of Detector System

3.2.2 Component Diagram

A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical components in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development. The purpose of this diagram is different. Component diagrams are used during the implementation phase of an application. However, it is prepared well in advance to visualize the implementation details. Initially, the system is designed using different UML diagrams and then when the artifacts are ready, component diagrams are used to get an idea of the implementation.

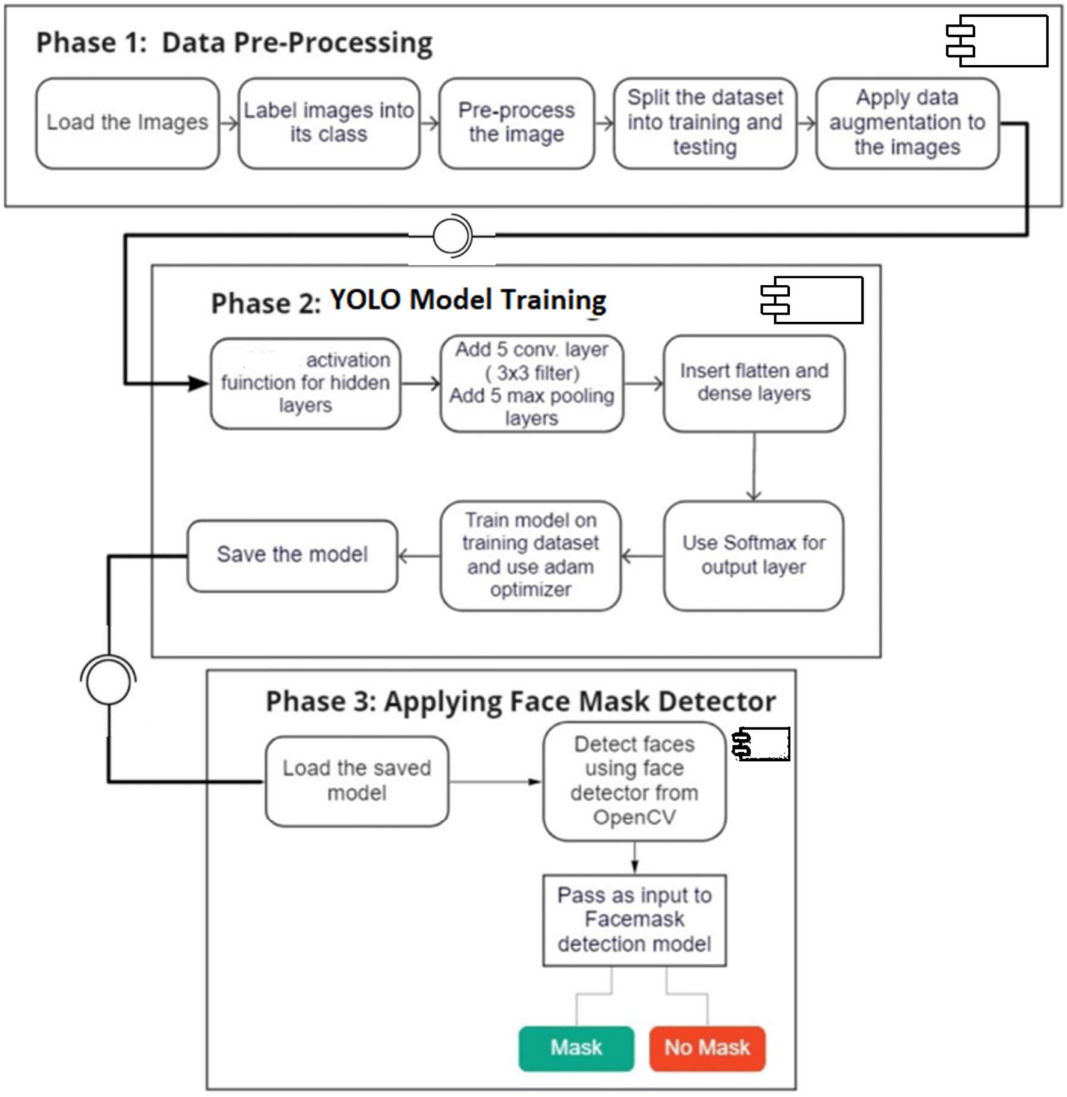


Figure 7: Component Diagram of Face Mask Detection System

3.2.3 Deployment Diagram

In the last step the model is integrated into a web-based application that is hosted in order to be shared easily with other users and they can upload their image or live video feed to the model to recognize facial masks and then get the predicted result. An open-source python library, is used to design and construct a simple web app that allows users to submit a picture with a single click of a button and receive the outcome in a matter of seconds. They show the structure of the run-time system. They capture the hardware that will be used to implement the system and the links between different items of hardware. They model physical hardware elements and the communication paths between them. They can be used to plan the architecture of a system. They are also useful for Document the deployment of software components or nodes

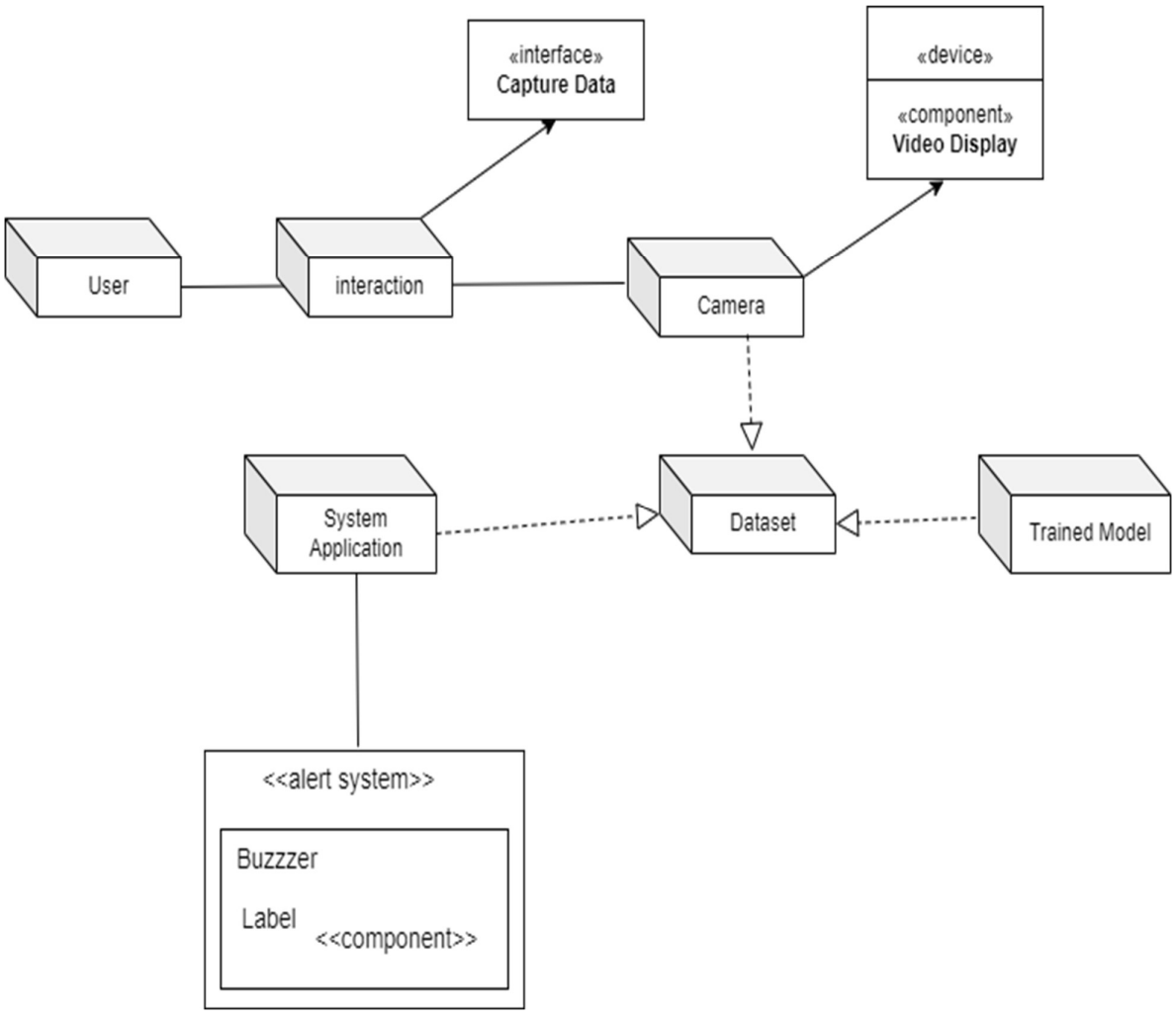


Figure 8: Deployment Diagram of Face Mask Detection System

3.3 Algorithm Details

YOLO is an algorithm that uses neural networks to provide real-time object detection. This algorithm is popular because of its speed and accuracy. It has been used in various applications to detect traffic signals, people, parking meters, and animals.

YOLO is an abbreviation for the term 'You Only Look Once'. This is an algorithm that detects and recognizes various objects in a picture (in real-time). Object detection in YOLO is done as a regression problem and provides the class probabilities of the detected images.

YOLO algorithm employs convolutional neural networks (CNN) to detect objects in real-time. As the name suggests, the algorithm requires only a single forward propagation through a neural network to detect objects.

This means that prediction in the entire image is done in a single algorithm run. The CNN is used to predict various class probabilities and bounding boxes simultaneously.

The YOLO algorithm consists of various variants. Some of the common ones include tiny YOLO and YOLOv3.

YOLO algorithm is important because of the following reasons:

a. Speed:
   This algorithm improves the speed of detection because it can predict objects in real-time.
b. High accuracy:
   YOLO is a predictive technique that provides accurate results with minimal background errors.
c. Learning capabilities:
   The algorithm has excellent learning capabilities that enable it to learn the representations of objects and apply them in object detection.

The techniques of algorithm are:

a. Residual Block
   x → Input H(x) → Expected Output
   F(x) → Mapping learned by the network
   New Output = F(x) + x ……. (i)
   H(x) = F(x) + x ……. (ii)
   F(x) = H(x) -x ……....(iii)

b. Bounding Box Regression
   $y = (p_c , b_x , b_y , b_h , b_w , c)$ ……. (iv)
   $b_x=\sigma(t_x)+c_x b_x=\sigma(t_x)+c_x$ …… (v)
   $b_y=\sigma(t_y)+c_y b_y=\sigma(t_y)+c_y$ ……... (vi)
   $b_w=p_w\, e^{tw}$ …… (vii)
   $b_h=p_h\, e^{th}$
   where (cx,cy) are the grid cell coordinates and (p_w,p_h) the anchor dimensions.

c. Intersection over Union (IoU) = $\frac{\text{area of overlap}}{\text{area of union}}$ ……. (viii)

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1 Implementation

4.1.1 Tools Used (CASE Tools, Programming Languages, Database Platforms)

The different types of CASE tools used in the project are:

- **Diagram Tool:**

  The components of the system, and the flow of the data and control between these components are demonstrated by diagram tools by using graphs. "Draw.io" and "Creately" are the diagram tool used in the project.

- **Version Control Tool:**

  An instance of software is released under one version. CASE tools help in this by automatic tracking, version management and release management. Git is used as configuration management tool.

- **Coding Tools:**

  These tools assist in coding modules and algorithm with all allied elements like libraries, functions, graphic and so on. Visual Studio Code, Anaconda, PyTorch, Opencv, Matplotlib, etc. are used as development tools in the project.

The different tools & programing language used in project:

- **Python Language:**

  The Python library provides base-level items, so developers do not have to write code from scratch every time. Machine learning requires continuous data processing, and Python libraries allow you to access, process, and transform your data. These are some of the most extensive libraries available for AI and ML.

- **Tensorflow:**

  TensorFlow provides a collection of workflows to develop and train models using Python or JavaScript, and to easily deploy in the cloud, on-prem, in the browser, or on-device no matter what language you use. The tf. data enables you to build complex input pipelines from simple, reusable pieces.

- **Keras:**

  Keras is based on minimal structure that provides a clean and easy way to create deep learning models based on TensorFlow or Theano. Keras is designed to quickly define deep learning models. Well, Keras is an optimal choice for deep learning applications.

- **Imutils:**

  Imutils is Python basic image processing functional package to do image translation, rotation, resizing, skeletonization, or blur amount detection. Imutils also check to find functions if you already have NumPy, SciPy, Matplotlib, and OpenCV installed.

- **Numpy:**

  Numpy is one of the most commonly used packages for scientific computing in Python. It provides a multidimensional array object, as well as variations such as masks and matrices, which can be used for various math operations. It consists of many functions. In many domains like Scientific Computing, Deep Learning and Financial analysis, the system works with high dimensional arrays. You may think Python's data structures like lists, tuples and dictionaries can be used to store such data.

- **OpenCV:**

  Open Source Computer Vision Library is an open source computer vision and machine learning software library. It was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. is a huge open-source library for computer vision, machine learning, and image processing. It 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.

- **Matplotlib:**

  Developers can also use matplotlib's APIs to embed plots in GUI applications. Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy. As such, it offers a viable open-source alternative to MATLAB. It supports various types of graphical representations like Bar Graphs, Histograms, Line Graph, Scatter Plot, Stem Plots, etc. Matplotlib can be used in multiple ways including Python scripts, the Python and iPython shells, Jupyter Notebooks. Matplotlib is a 2-D plotting library.

- **Scipy:**

  SciPy is a scientific computation library that uses NumPy underneath. It stands for Scientific Python. It provides more utility functions for optimization, stats and signal processing. It is an open-source Python library which is used to solve scientific and mathematical problems. It is built on the NumPy extension and allows the user to manipulate and visualize data with a wide range of high-level commands.

- **PyGame:**

The pygame library is an open-source module for the Python programming language specifically intended to help you make games and other multimedia applications. It adds functionality on top of the excellent SDL library. This allows you to create fully featured multimedia programs in the python language. It is highly portable and runs on nearly every platform and operating system.

4.1.2 Implementation Details of Modules (Description of procedures/functions)

Create a function to find the objects and assign the bounding box thresholds. Here the system uses the "NMSBoxes" function from the OpenCV darknet module to do the non-max suppression. And then using the OpenCV rectangle function the system draw the bounding boxes and using putText function the system add the labels to the output.

Considering the scenario where a 10 X 10 grid with two anchors per grid is used, and there are 3 different object classes. So the corresponding y labels will have a shape of 10 X 10 X 16. Now, suppose if 5 anchor boxes per grid is used, the number of classes has been increased to 5. So the target is to be 10 X 10 X 10 X 5. This is how the training process is done – taking an image of a particular shape and mapping it with a 10 X 10 X 16 target (this may change as per the grid size, number of anchor boxes and the number of classes).

The new image is be divided into the same number of grids which is chosen during the training period. For each grid, the model predicts an output of shape 10 X 10 X 16. The 16 values in this prediction will be in the same format as that of the training label. The first 8 values will correspond to anchor box 1, where the first value is the probability of an object in that grid. Values 2-5 is the bounding box coordinates for that object, and the last three values tell us which class the object belongs to. The next 8 values will be for anchor box 2 and in the same format, i.e., first the probability, then the bounding box coordinates, and finally the classes. Finally, the Non-Max Suppression technique will be applied on the predicted boxes to obtain a single prediction per object. That brings to the end of the theoretical aspect of understanding how the YOLO algorithm works, starting from training the model and then generating prediction boxes for the objects. Finally, the IoU and Non-Max Suppression is done to avoid selecting overlapping boxes.

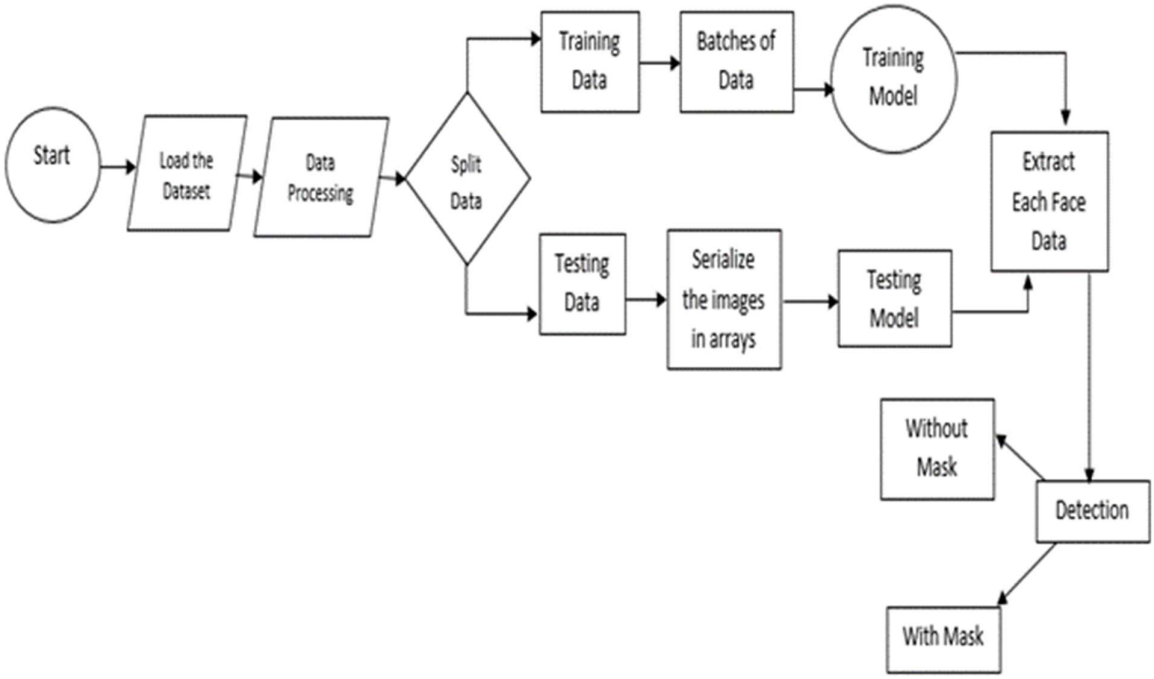


Figure 9: Working Mechanism of face mask detection system

When the system will be started, the webcam should be activated. The model extracts dataset after cleaning the dataset and only process ahead the frames of images containing faces ignoring the background or any other things. The system should detect the user's face mask. After the detection of user's face, the system should detect whether the user's wearing a mask or not. If the user will be detected wearing mask, then the system should show green box around user's face and keep monitoring else the system should display a red alert kind of box-interface to the user. Then, again the system will start monitoring the user.

Here YOLO machine learning technique is used to identify persons wearing and not wearing facemasks. Joseph Redmon et al. introduced You look only once also known as YOLO in 2015. The algorithm applies a single neural network to the full image, and then divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities. YOLO is extremely fast. It scans the entire image during training and also during testing. It learns generalizable representations of objects so that when it is trained on natural images and tested, the algorithm performs excellently when compared to other top detection methods. This algorithm employs convolutional neural networks (CNN) to detect objects in real-

time. As the name suggests, the algorithm requires only a single forward propagation through a neural network to detect objects.

The algorithm works using the following three techniques:

- Residual blocks:

  First, the image is divided into various grids. Each grid has a dimension of 10 x 10. The following image shows how an input image is divided into grids. In the image above, there are many grid cells of equal dimension. Every grid cell will detect objects that appear within them. For example, if an object center appears within a certain grid cell, then this cell will be responsible for detecting it.

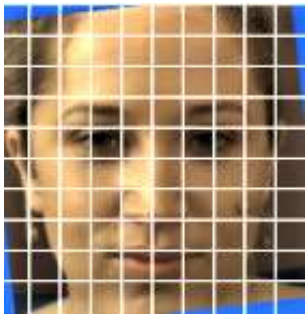


Figure 10: Grid Blocks

- Bounding box regression:

  Each bounding box can be described using four descriptors: Center of the box (bx, by), Width (bw), Height (bh), Value c corresponding to the class of an object. Along with that the system predict a real number P_c, which is the probability that there is an object in the bounding box. YOLO uses a single bounding box regression to predict the height, width, center, and class of objects. In the image above, represents the probability of an object appearing in the bounding box.

$$y = (p_c, b_x, b_y, b_h, b_w, c)$$

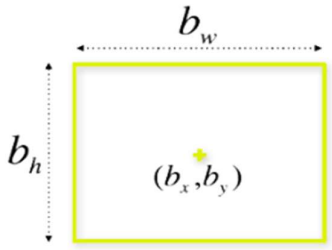


Figure 11: Bound Box [Source: https://www.v7labs.com]

- Intersection Over Union (IOU):

  To resolve the problem Non-max suppression, it eliminates the bounding boxes that are very close by preforming the IoU (Intersection over Union) with the one having the highest class probability among them.

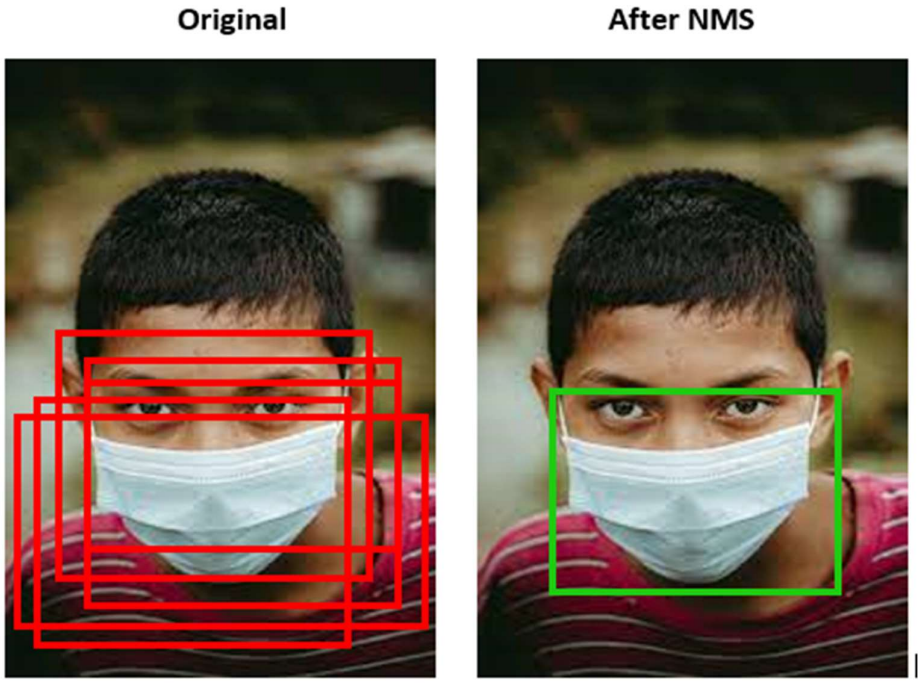


Figure 12: Before and after Non Max Suppression

When working with the YOLO algorithm, it is not searching for interesting regions in our image that could potentially contain an object. Instead, image is splitted into into cells, typically using a 10×10 grid. Each cell is responsible for predicting 5 bounding boxes (in case there is more than one object in this cell). Therefore, it is arrived at a large number of 1805 bounding boxes for one image. Rather than seizing the day with #YOLO, it is looked to seize object probability. The exchange of accuracy for more speed isn't reckless behaviour, but a necessary requirement for faster real-time object detection. IOU is mainly used in applications related to object detection, where train a model to output a box that fits perfectly around an object. There's a green box, and a blue box. The green box represents the correct box, and the blue box represents the prediction from our model. The aim of this model would be to keep improving its prediction, until the blue box and the green box perfectly overlap, i.e. the IOU between the two boxes becomes equal to 1.

4.2 Testing

It's a process of executing a program or application with the intent of finding the software bugs. It can be also stated as the process of validating and verifying that a software program or application or product meets the business and technical requirements that guided its design and development. Testing can be done using varieties of level.

4.2.1 Test Cases for Unit Testing

The modules of application are tested alone in an attempt to find any errors in its code. In this web application, here is taken different procedures and interface as the small unit and tested individually.

Table 1: Test Cases for Unit Testing

S.N.	Test Cases	Test Data Input	Expected Outcome	Obtained Test Result
1.	Check with valid data	Stable human face	Mask detected	Pass
2.	Check with empty input	No any object	Idle system	Nothing to detect
3.	Check with animals' face	Dog's face	Object is invalid	Different object appeared than dataset
4.	Check unstable input	With rapid movement face	Mask Not Detected	Fail
5.	Check playing videos	Human face motion picture in movies	Should be able to detect details	Depends upon image stability

4.2.2 Test Cases for System Testing

System testing during development involves integrating components to create a version of the system and then testing the integrated system. System testing checks that components are compatible, interact correctly, and transfer the right data at the right time across their interfaces.

Table 2: Test Cases for System Testing

S.N.	Test Cases	Test Data Input	Expected Outcome	Obtained Test Result
1.	Near/Far Image Blocks	Stay near & far from camera	Categorize Residual blocks.	Hard to detect far objects
2.	Centre of box in object	Place different shaped faces	Probability that there is object	Bounding box determined
3.	Multiple objects together	Different aged people	Call classification function	Mask detected
4.	Putting Mask on other parts of face	Mask under the chin	Must throw required input not matching error	Buzzer Alert
5.	Use others things than mask	Using hands / hankey & other objects to cover face	Should be able to identify proper mask	(Without Mask) Fail

CHAPTER 5: CONCLUSION AND FUTURE RECOMMENDATIONS

5.1 Lesson Learnt/Outcome

Working with a project has been a great experience in learning new things and gaining practical knowledge. During the period of project, it is known that theory knowledge is incomplete without practical knowledge. But to work in practical environment, the theory knowledge is very essential. Some of the lessons learnt from the timeline of project are:

- Work in mentorship and make quality group decision.
- Importance of coordination and cooperation in the work environment.
- Understanding the difference between theoretical knowledge and practical world.

5.2 Conclusion

With the completion of the project, the experience of working in a team and maintaining cooperative behaviour has been got. The use and implementation of recognition algorithm has been learnt by the project. The project is done by using Python and its libraries in Visual Studio Code. Programming with very minor knowledge was a challenging task but was very much worthwhile to learn and experience. The article proposed an efficient real-time deep learning-based technique to automate the process of detecting masked faces, where each masked face is identified in real-time with the help of bounding boxes.

The extensive trials were conducted with popular models, namely. It has better precision, for applying this in real-world surveillance cameras, it would be preferred to use the model with this algorithm as it performs single-shot detection and has a much higher frame rate than Faster-RCNN or any other state-of-the-art object detection algorithms. Which model to use, also depends on the resources available. If high-end GPUs are available on the deployed devices, faster R-CNN must be used. YOLOv3 can be deployed on mobile phones also. Since this approach is highly sensitive to the spatial location of the camera, the same approach can be fine-tuned to better adjust with the corresponding field of view. These models can be used along with surveillance cameras in offices, metros, railway stations and crowded public areas to check if people are following rules and wearing marks. The trained weights provided by the authors can be further improved by training on larger datasets and can then be used in real-world applications.

5.3 Future Recommendations

This application can be used in any working environment like any public place, station, corporate environment, streets, shopping malls, examination centres, etc., where accuracy and precision are highly desired to serve the purpose. This technique can be used in smart city innovation and it would boost up the development process in many developing countries. Our analysis of the current circumstance presents a chance to be more ready for the next crisis, or to evaluate the effects of huge scope social change.

In future, number of cameras and scanning displays can be increased to achieve positive outcomes. Its just desktop app for now and can be developed as android and iOS application.

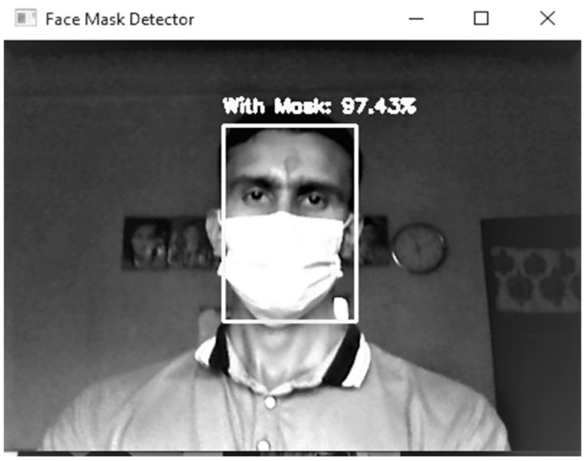
Since different network architecture performs better in mask detection tasks, the model helps in yielding to large dataset performance.

The regularities in fast moving images, like those with insufficient light and side angle, can be given proper attention in upcoming days by different techniques of image processing & image stabilization algorithms.

Also, high accuracy in the least possible time can be achieved. Additionally, the video analysis can be made simple, including motion blur, transitioning between frames, etc.

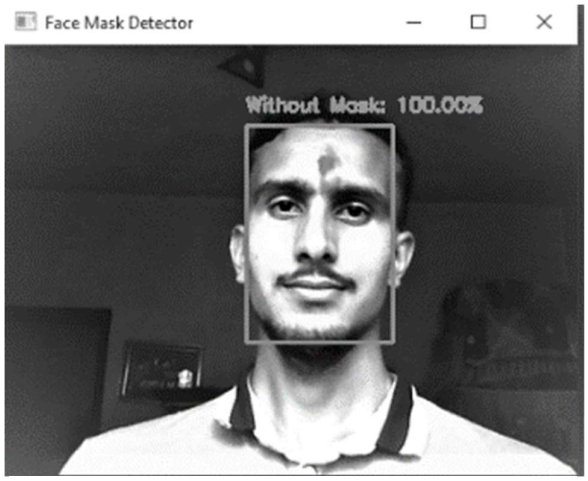
APPENDIX

Outcome Screenshots:



1
Check with valid data
Stable human face
Mask detected

Figure 13: Check with valid data



2
Check without wearing mask
Stable human face
Mask not detected, alarm on

Figure 14: Check without mask on

3
Check with wearing mask on half face
Stable human face
Mask not detected, alarm on

Figure 15: Check with mask on half face

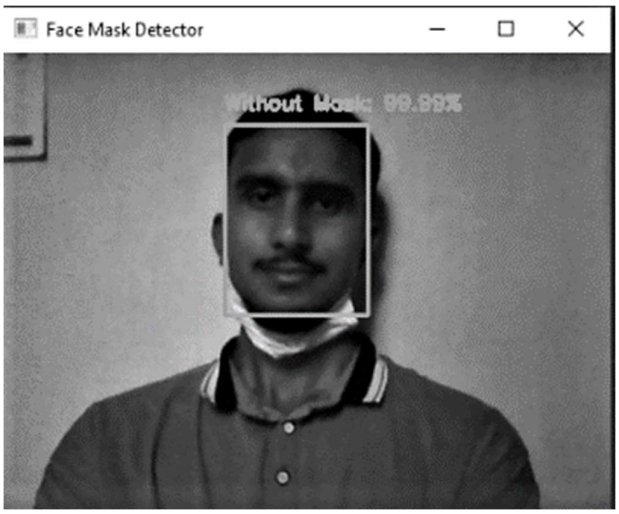


Figure 16: Check with mask under the chin

4
Check with wearing mask under the chin
Stable human face
Mask not detected, alarm on

Figure 17: Use others objects than mask

5
Using hands / hankey & other objects to cover face
Stable human face
Mask not detected, alarm on

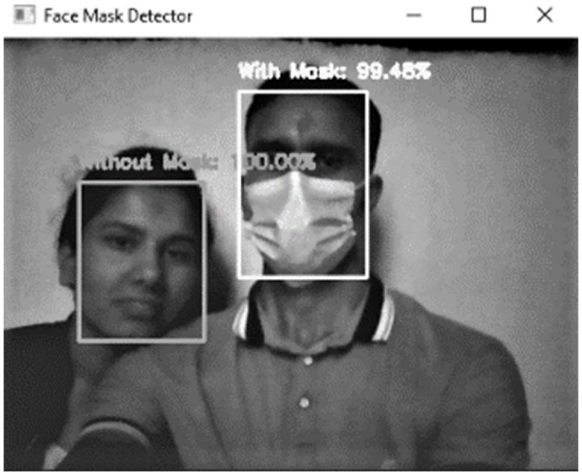


Figure 18: Testing Multiple faces together

6
Multiple faces together/ Different aged people
Stable human face
Bounding box determined, Mask detected

REFERENCES

[1] A. Voulodimos, "PubMed," 01 February 2018, *Past, Present, and Future of Face Recognition: A Review,* Available: https://pubmed.ncbi.nlm.nih.gov/29487619/. [Accessed 27 January 2022].

[2] A. Benzaoui, "MDPI," 02 March 2020, *Facial Recognition System for People with and without Face Mask in Times of the COVID-19 Pandemic,* Available: https://www.mdpi.com/2079-9292/9/8/1188. [Accessed 27 January 2022].

[3] Biplab Ketan, "ScholarWorks," 24 April 2017, *Review of constraints on vision-based gesture recognition for human–computer interaction,* Available: https://scholarworks.iupui.edu/handle/1805/17801. [Accessed 05 February 2022].

[4] Byoungchul Ko, "Research Gate" January, 2018, *A Brief Review of Facial Emotion Recognition Based on Visual Information,* Available:. [Accessed 05 February 2022].

[5] Egger M, "Semantics Scholar", 28 June 2018, *A Review of Emotion Recognition Using Physiological Signals,* Available: https://www.semanticscholar.org/. [Accessed 05 February 2022].

[6] Gotel, O. C. Z., and Finkelstein, A. C. W., *An analysis of the requirements traceability problem, Requirements Engineering, 2013*, Proceedings of the First Annual Conference on Requirements Engineering, Colorado Springs, Colorado, 94–101, April 18–22, 2014. (Most Influential Paper Award given 2017.)

[7] C. Quazi, "U-Texas," 02 March 2020, *Machine Learning Project,* Available: https://www.cs.utexas.edu/~mooney/cs391L/paper-template.html. [Accessed 05 February 2022].