Face Mask Detection System with Percentage for Analyzing Discretion Factor

By

Jone
ID: CSE1803015064

Md Sahin Bissas
ID: CSE1803015052

Lovely Akter
ID: CSE1803015087

Supervised by
Khadija Islam

Submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SONARGAON UNIVERSITY (SU)

September 2022

APPROVAL

The project titled **"Face Mask Detection System with Percentage for Analyzing Discretion Factor"** submitted by Jone (CSE1803015064), Md. Sahin Bissas (CSE1803015052), Lovely Akter (CSE1803015087) to the Department of Computer Science and Engineering, Sonargaon University (SU), has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering approved as to its style and contents.

Board of Examiners

---------- **Supervisor**

Khadija Islam
Lecturer,
Department of Computer Science and Engineering
Sonargaon University (SU)

---------- **Examiner 1**

(Examiner Name & Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

---------- **Examiner 2**

(Examiner Name & Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

---------- **Examiner 3**

(Examiner Name & Signature)
Department of Computer Science and Engineering
Sonargaon University (SU)

DECLARATION

We, hereby, declare that the work presented in this report is the outcome of the investigation performed by us under the supervision of **Khadija Islam, Lecturer,** Department of Computer Science and Engineering, Sonargaon University, Dhaka, Bangladesh. We reaffirm that no part of this [thesis or project] has been or is being submitted elsewhere for the award of any degree or diploma.

Countersigned

(Khadija Islam)
Supervisor

Signature

Jone
ID: CSE1803015064

Md Sahin Bissas
ID: CSE1803015052

Lovely Akter
ID: CSE1803015087

ABSTRACT

The novel Coronavirus has brought a new normal life in which the social distance and wearing of face masks plays a vital role in controlling the spread of the virus. But most of the people are not wearing face masks in public places which increases the spread of viruses. This may result in a serious problem of increased spreading. Hence to avoid such situations we have to scrutinize and make people aware of wearing face masks. Humans cannot be involved in this process, due to the chance of getting affected by corona. Hence here comes the need for artificial intelligence (AI), which is the main theme of our project. Our project involves the identification of persons wearing face masks and not wearing face masks in public places by means of image processing and AI techniques and sending alert messages to authority persons. The object detection algorithms are used for identification of persons with and without wearing face masks which also gives the count of persons wearing mask and not wearing face mask and Internet of Things (IOT) is utilized for sending alert messages. The alert messages are sent to the authority persons through mobile notification and Email. 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 and LED's.

ACKNOWLEDGMENT

At the very beginning, we would like to express my deepest gratitude to the Almighty Allah for giving us the ability and the strength to finish the task successfully within the scheduled time.

We are auspicious that we had the kind association as well as supervision of **Khadija Islam**, Lecturer, Department of Computer Science and Engineering, Sonargaon University whose hearted and valuable support with best concern and direction acted as necessary recourse to carry out our project.

We would like to convey our special gratitude to **Bulbul Ahamed,** Associate Professor and Head**,** Department of CSE, and **Prof. Dr Md Alamgir Hossain, Dean**, Faculty of Science and Engineering for his kind concern and precious suggestions.

We are also thankful to all our teachers during our whole education, for exposing us to the beauty of learning.

TABLE OF CONTENTS

CHAPTER 6

LIST OF FIGURES

CHAPTER 1
INTRODUCTION OF FACE MASK DETECTION

1.1 Introduction

The novel coronavirus covid-19 had brought a new normal life. India is struggling to get out of this virus attack and the government implemented a lockdown for the long way. Lockdown placed a pressure on the global economy. So, the government gave relaxations in lockdown. Declared by the WHO that a potential speech by maintaining distance and wearing a mask is necessary. The biggest support that the government needs after relaxation is social distancing and wearing of masks by the people. But many people are getting out without a face mask this may increase the spread of covid-19. Economic Times India has stated that " Survey Shows that 90 percent Indians are aware, but only 44 percent wearing a mask ". This survey clearly points that people are aware but they are not wearing the mask due to some discomfort in wearing and carelessness. This may result in the easy spreading of covid-19 in public places. The world health organization has clearly stated that until vaccines are found the wearing of masks and social distancing are key tools to reduce spread of virus. So, it is important to make people wear masks in public places. In densely populated regions it is difficult to find the persons not wearing the face mask and warn them. Hence, we are using image processing techniques for identification of persons wearing and not wearing face masks. In real time images are collected from the camera and it is processed in Raspberry Pi embedded development kit. 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. The Internet of Things (IOTs) can be used for connecting objects like smartphones, Internet TVs, laptops, computers, sensors and actuators to the Internet where the devices are linked together to enable new forms of communication between things and people, and between things themselves. Intimation messages are sent to authority persons by means of IOT.

1.2 Objective

My objective is to study a model with the end goal that it picks the right class. The principal aim of this project is to develop a Deep learning model (CNN) with the help of various frameworks to predict in real-time if a person(s) is wearing a mask or not. The project also aims to create a strong and accurate surveillance system by using modern concepts of AI and Computer Vision. It is indirectly useful to all citizens as this system can be adopted at various public and private organizations to monitor people not wearing a mask and keep a regular check on those who violate. There are no efficient face mask detection applications to detect whether the person is wearing a face mask or not. This increases the demand for an efficient system for detecting face masks on people for transportation means, densely populated areas, residential districts, large-scale manufacturers, and other enterprises to ensure safety. This project uses machine learning classification using OpenCV and TensorFlow to detect face masks on people.

1.3 Project Background

The proposed framework centers around how to recognize the individual on a picture/video stream wearing a face mask with the assistance of a PC vision and machine learning algorithm by utilizing the OpenCV, TensorFlow, Keras and PyTorch libraries.

Approach:

1. Dataset of around 300 images with & without face mask are collected out of which 150 images are used for training the model and 150 images for testing the model.
2. Apply a face mask detector created using a machine learning algorithm over the picture/live video stream.
3. The system will show warning to the violator.

1.4 Why this detection project

Deep learning is an important breakthrough in the AI field. It has recently shown enormous potential for extracting tiny features in image analysis. Due to the COVID-19 epidemic, some deep learning approaches have been proposed to detect patients infected with coronavirus. In this context, and unlike bacterial pneumonia, many other types of lung infections caused by viruses are called viral pneumonia. These viruses, such as the COVID-19, infect the lungs by blocking the oxygen flow, which can be life-threatening. This motivated researchers to develop many frameworks and schemes based on AI tools in the fight against this dangerous virus. Hence, we divide this section into two sections to provide an in-depth overview of the proposed techniques.

1.5 Aim and Scope of the Project

Face mask detection is the identification of whether or not a person is wearing a mask. In actuality, the challenge is reverse engineering of face detection, in which the face is identified using various machine learning techniques for security, authentication, and surveillance. Face detection is a critical area in Computer Vision and Pattern Recognition. In the past, a substantial body of research has contributed complex algorithms for face identification. The initial study on face detection was conducted in 2001, with the use of handwritten features and traditional machine learning methods to build successful classifiers for detection and recognition. Create a unique object recognition approach that combines one-stage and two-stage detectors to reliably recognize objects in realtime from video streams, using transfer learning at the back end. To crop facial regions from uncontrolled real-time photos with changes in face size, orientation, and backdrop, an improved affine transformation is created. This phase aids in locating the individual who is breaching the facemask rules in public areas/offices. Making an impartial facemask dataset with an imbalance ratio close to one. The suggested approach uses less memory, making it suitable for embedded devices used for surveillance.

CHAPTER 2

BACKGROUND OF FACE MASK DETECTION

2.1 Introduction

As well as writing Python, Keras, OpenCV and TensorFlow and many so on tools may be needed to test a new app. Face Mask Detection has been developed by python, Keras, OpenCV, TensorFlow. This project helps for detecting the mask and non-masking people.

2.2 Languages and Tools

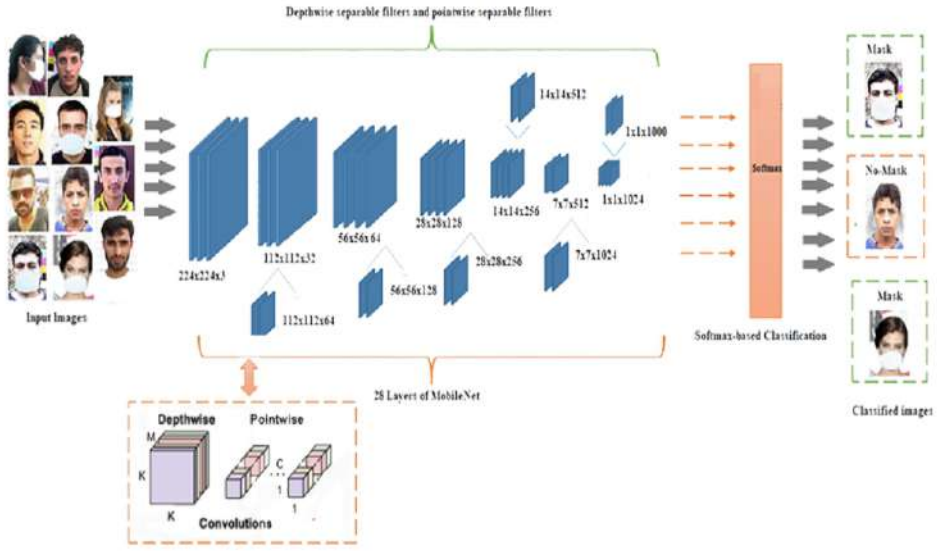

Fig-1.1: Languages & Tools

2.2.1 Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. Python's simple, easy to learn syntax emphasizes readability and therefore reduces the cost of program maintenance. Python supports

modules and packages, which encourages program modularity and code reuse. The Python interpreter and the extensive standard library are available in source or binary form without charge for all major platforms, and can be freely distributed.

Often, programmers fall in love with Python because of the increased productivity it provides. Since there is no compilation step, the edit-test-debug cycle is incredibly fast. Debugging Python programs is easy: a bug or bad input will never cause a segmentation fault. Instead, when the interpreter discovers an error, it raises an exception. When the program doesn't catch the exception, the interpreter prints a stack trace. A source level debugger allows inspection of local and global variables, evaluation of arbitrary expressions, setting breakpoints, stepping through the code a line at a time, and so on. The debugger is written in Python itself, testifying to Python's introspective power. On the other hand, often the quickest way to debug a program is to add a few print statements to the source: the fast edit-test-debug cycle makes this simple approach very effective.

2.2.2 keras

Keras is a high-level, deep learning API developed by Google for implementing neural networks. It is written in Python and is used to make the implementation of neural networks easy. It also supports multiple backend neural network computation.
Keras is relatively easy to learn and work with because it provides a python frontend with a high level of abstraction while having the option of multiple back-ends for computation purposes. This makes Keras slower than other deep learning frameworks, but extremely beginner-friendly.

Keras allows you to switch between different back ends. The frameworks supported by Keras are:

- TensorFlow
- Theano
- PlaidML
- MXNet
- CNTK (Microsoft Cognitive Toolkit)

Out of these five frameworks, TensorFlow has adopted Keras as its official high-level API. Keras is embedded in TensorFlow and can be used to perform deep learning fast as it provides inbuilt modules for all neural network computations. At the same time, computation involving tensors, computation graphs, sessions, etc can be custom made using the TensorFlow Core API, which gives you total flexibility and control over your application and lets you implement your ideas in a relatively short time.

2.2.3 TensorFlow

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow. TensorFlow is a Python-friendly open-source library for numerical computation that makes machine learning and developing neural networks faster and easier. TensorFlow can train and run deep neural networks for handwritten

digit classification, image recognition, word embeddings, recurrent neural networks, sequence-to sequence models for machine translation, natural language processing, and PDE (partial differential equation)-based simulations, competing with frameworks such as PyTorch and Apache MXNet. Best of all, TensorFlow can anticipate production at scale using the same models that were used for training. TensorFlow also comes with a large library of pre-trained models that you may utilize in your own applications. You may also utilize TensorFlow Model Garden code as examples of excellent practices while training your own models.

TensorFlow applications may be launched on almost any suitable target: a local system, a cloud cluster, iOS and Android devices, CPUs or GPUs. If you utilize Google's cloud, you can accelerate TensorFlow by running it on Google's own TensorFlow Processing Unit (TPU) hardware. TensorFlow-generated models, on the other hand, may be installed on almost any device and used to offer predictions. TensorFlow 2.0, launched in October 2019, improved the framework in many areas based on user input, making it easier to use (for example, by leveraging the comparatively basic Keras API for model training) and more performant. A new API makes distributed training easier to implement, and support for TensorFlow Lite allows models to be deployed on a wider range of systems. To take full use of new TensorFlow 2.0 capabilities, code developed for previous versions of TensorFlow must be rewritten—sometimes very slightly, sometimes extensively.

2.2.4 OpenCV

OpenCV-Python is a library of Python bindings designed to solve computer vision problems. Python is a general-purpose programming language started by Guido van Rossum that became very popular very quickly, mainly because of its simplicity and code readability. It provides a wide range of features, including object detection, face recognition, and tracking.

2.2.5 PyCharm

PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development.

The main reason PyCharm for the creation of this IDE was for Python programming, and to operate across multiple platforms like Windows, Linux, and macOS. The IDE comprises code analysis tools, debugger, testing tools, and also version control options. It also assists developers in building Python plugins with the help of various APIs available. The IDE allows us to work with several databases directly without getting it integrated with other tools. Although it is specially designed for Python, HTML, CSS, and JavaScript files can also be created with this IDE. It also comes with a beautiful user interface that can be customized according to the needs using plugins. PyCharm created this IDE primarily for Python programming and to run on different platforms such as

Windows, Linux, and macOS. The IDE includes code analysis tools, a debugger, testing tools, and version management. It also supports developers in creating Python plugins by utilizing the numerous APIs accessible. The IDE enables us to work directly with several databases without requiring integration with additional tools. Although it is specifically built for Python, it can also produce HTML, CSS, and JavaScript files. It also has a stunning user interface that can be adjusted via plugins to meet your specific needs. PyCharm is a well-known Python IDE. A code editor and a compiler are included in an Integrated Development Environment (IDE) for developing and compiling programs in one or more programming languages. Furthermore, an IDE has a plethora of tools that aid in complete software development.

2.2.6 Machine Learning

Machine learning (ML) is a discipline of artificial intelligence (AI) that provides machines with the ability to automatically learn from data and past experiences while identifying patterns to make predictions with minimal human intervention. Machine learning methods enable computers to operate autonomously without explicit programming. ML applications are fed with new data, and they can independently learn, grow, develop, and adapt. Machine learning derives insightful information from large volumes of data by leveraging algorithms to identify patterns and learn in an iterative process. ML algorithms use computation methods to learn directly from data instead of relying on any predetermined equation that may serve as a model. When starting your educational journey, it's critical to first grasp how to study ML. We've divided the learning process into four knowledge domains, with each supplying a core piece of the ML jigsaw. We've found books, videos, and online courses that can help you advance your skills and equip you to apply ML in your projects. Begin by browsing our resource collection and following our guided curriculums to expand your knowledge. Machine learning is significant because it provides organizations with insights into trends in consumer behavior and company operating patterns, as well as assisting in the development of new products. Machine learning is fundamental to the operations of many of today's biggest organizations, like Facebook, Google, and Uber. For many businesses, machine learning has become a crucial competitive differentiation. The data scientist must train the algorithm with both labeled inputs and intended outputs in supervised machine learning. The following tasks benefit from supervised learning algorithms. Binary classification is the division of data into two groups. Choosing between more than two categories of replies is referred to as multi-class categorization. Predicting continuous values using regression modeling. Combining the predictions of numerous machine learning models to get an accurate forecast is known as assembling. Unsupervised machine learning techniques do not need labeled data. They search through unlabeled data for patterns that may be used to classify data points into subgroups. The vast majority of deep learning methods, including neural networks, are unsupervised. The following tasks are well suited to unsupervised learning algorithms. Clustering is the process of dividing a dataset into groups based on similarities. Anomaly detection is the discovery of unexpected data points in a data set. Association mining is the process of identifying groups of things in a data collection that commonly appear together. Dimensionality reduction refers to the process of reducing the number of variables in a data source.

CHAPTER 3
METHODOLOGY

3.1 Introduction

This project is based on the functional design method, which simplifies the project's design by describing its flow, use cases, and implementation in a modular manner. In this project, for example, there are various modules with distinct functionality and sub functionalities/modules. To create a flawless operating application, all of the components are planned, implemented, and integrated together.

3.2 Iterative Software Model

When there are complex pictures in the dataset, this paper explains mask detection using MATLAB. For mask detection, MATLAB defined the Faster R-CNN method and Dataset allotment. This article uses facial recognition software to manage complicated images. The Faster R-CNN approach is applied in both the security and medical systems. Face restriction, color changes, brightness changes, and contrast adjustments were all balanced in the suggested work. Face limitation of a human picture is achieved by segmentation and feature extraction. For identifying Mask identification and social distance, we used the RCNN, Fast RCNN, and Faster RCNN algorithms. Convolutional neural network regions Using image mixing, pixel prediction, and particular improvements. The major goal was to solve various and multitask image detection issues at high speeds. The methodology utilized for face detection and Unmask person detection in a face database dataset.

3.3 Proposed Methodology

The system must have an unbiased 'with_mask' dataset.The dataset must have over 1500+ images in both 'with_mask' and 'without_mask' classes.The dataset must not re-use the same images in training and testing phases.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.The system must be able to extract each face's Region of Interest (ROI).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.The end position of the face must be fit inside the webcam frame and must be closer to the camera.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'.

3.4 System Planning

The gadget can identify up to ten persons who are not wearing face masks. at the same time the frame (if allowed by the computing capacity). The module does not recognize (identify) faces; it

cannot distinguish one individual from another. It just compares one person's face to the faces in a database discovers persons in the frame who are not wearing face masks When an infringement event (no mask) is identified, the module temporarily shows the person's face in the client application with a red square and records a matching event in the event log. Repeated detection of the violation by the same person will be achievable only after this person has vanished from the screen for at least three seconds (when the individual exits the frame or totally hides his/her face).

3.5 System Analysis

Combining two types of photos in a single dataset is an efficient way to identify illness. Sedik et al. published two deep learning models in this area in [21]: CNN and ConvLSTM. Two datasets are assumed in order to simulate the models. The first dataset contains CT pictures, whereas the second has X-ray images. COVID-19 and non-COVID-19 picture categories are included in each dataset. To validate the suggested models, the picture categories COVID-19 and pneumonia were categorized.

3.6 Requirement Analysis

The first CNN-based model has five convolutional layers (CNVLs) and five pooling layers (PLs). The classification network is made up of two layers: fully connected layer (FC) and classification layer. The second design is a hybrid. It combines ConvLSTM and CNN simultaneously. The categorization network is also included in the initial model. To simplify the complexity of the planned deep learning structure, the three processes that comprise the two modalities are training, validation, and testing. In training, an optimization mechanism is required. Sedik et al. employed the Adam optimizer to minimize the differences between the actual and estimated objectives. This sort of model must be handled with caution. The accuracy, Matthew's correlation coefficient (MCC), and F1-score of the suggested models are all measured. The evaluation procedure takes into account specificity, negative predictive value (NPV), sensitivity, and positive predictive value (PPV).

3.7 System Requirement

If utilized appropriately, the COVID-19 mask detector we're constructing here today might assure your and others' safety (but I'll leave that up to medical experts to decide on, implement, and disseminate in the wild). It just compares one person's face to the faces in a database discovers persons in the frame who are not wearing face masks When an infringement event (no mask) is identified, the module temporarily shows the person's face in the client application with a red square and records a matching event in the event log. Repeated detection of the violation by the same person will be achievable only after this person has vanished from the screen for at least three seconds (when the individual exits the frame or totally hides his/her face).

CHAPTER 4

REQUIREMENTS ANALYSIS AND SYSTEM SPECIFICATIONS

4.1 Software Requirements

The software requirements features are follows -

4.1.1 Data Requirements

Data requirements specify the set of data that is involved in any project. The login credentials for registering the application and the item's details are the most important pieces of information for this project. The program will not be able to complete the transaction without this information.

4.1.2 Functional Requirements

The properties that must be present in the final system are known as functional requirements. To use any mobile application, we must first download it from the Google Play Store. Depending on the store or seller, the application could be free or charged. The user must register and login to the application after it has been installed by giving login credentials in order to use it. He or she logs in after that.

4.1.3 Performance Requirements

The performance requirements that should be considered while creating any system include response speed, scalability, platform dependencies, and tolerance. When the user interacts with the application, the application or system should be able to reply rapidly. When we wish to expand the application, it should be constructed in such a way that it is scalable enough to accept new features. The software's application from the project's design phase onwards, it should run in all of the required software and hardware requirements. In addition, the program's tolerance rate (fault tolerance) should be set higher in the event of network challenges, connectivity issues, and when the application crashes or quits. When the system is up and running, it should be able to provide information to the user about any of those difficulties.

4.2 Testing and Maintainability Requirements

In a test environment, the application should be able to meet all of the conceivable good and bad test scenarios. When a user uses an application, it should be created in such a way that it does not have any faults or crashes. When we expand the code or add new functionalities to the existing application, it should be able to extend itself.

Fig-4.1: Testing requirements

4.3 Validation

Before releasing an application to users, it is critical to validate it. If information supplied by users is not validated, it may be redundant, formatted incorrectly, and unmaintainable. For instance, we can validate a mobile number so that it only contains digits and characters. If the validation isn't done, there's a probability the user will make a mistake. The authorized person is unable to contact the appropriate person in the event of an emergency. Validations for all fields that are used to save information in any application are also essential. In this application, I validated the Login Page and the Home Page multiple times. I checked all of the login information that the user needs to sign up for the first time on the Login Page. Error warnings are displayed when fields such as login, email, password, and cellphone number are not correct. The username must not contain any digits, the password must be at least six letters long, the email must be a legitimate address, and if the email address has already been registered, an error notice will appear stating that the email address already exists. The mobile number should only contain digits, and while scanning the item, the scan process should be completed correctly otherwise, a message will be displayed. When looking for an item by name, the search field should not contain any digits. Following face detection, an efficient CNN-based architecture is employed for face mask detection training and testing. There are numerous existing designs for training purposes, which were previously described in the literature review section. In this paper, we introduced a bespoke architecture for detecting whether or not a person is wearing a face mask. Facial feature analysis is used to detect face masks in the suggested study. The suggested lightweight CNN model comprises four convolutions, one completely connected and one output layer, and a nonlinear activation function.

CHAPTER 5
SYSTEM DESIGN

5.1 Introduction

According to one source, using face masks can help reduce Covid-19 transmission. Covid-19 is the most recent pandemic virus to threaten human health in the twentieth century. Covid-19's rapid growth has prompted WHO to declare it a worldwide pandemic in 2020. In less than six months, Covid-19 infected almost 5 million people in 188 countries. The virus spreads through close interaction in heavily crowded areas. Because of the Coronavirus pandemic, there has been extraordinary worldwide scientific cooperation. Machine learning uses enormous quantities of data to estimate Covid-19dispersion, act as a brief notice system for potential pandemics, and classify susceptible groups. Many nations have laws mandating people to wear face masks when in a crowd. We have a tendency to create these regulations and laws in response to the exponential growth of occurrences and deaths in a variety of fields. As a result, recognizing face masks is a challenging task. Because of the spread of the coronavirus illness, several nations have implemented laws such as "No admittance without a mask." Face mask detection is a crucial problem in security and the prevention of Covid-19. In the medical field, a mask reduces the risk of infection from an infected person, whether or not they exhibit symptoms. Face mask detection is used in many places, including airports, hospitals, businesses, and educational institutions. Face recognition without a mask is easier, but face recognition with only a mask is more difficult because masking face feature extraction is more difficult than conventional face feature extraction. The covered face lacks several facial features, such as the nose, lips, and chin.

5.2 System Architecture Design

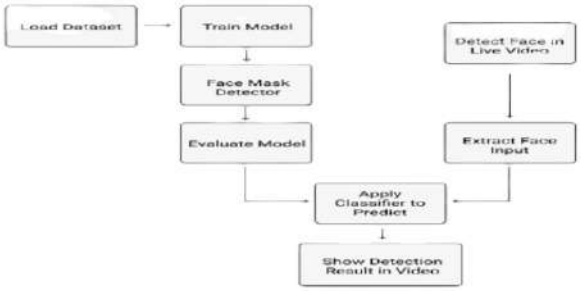

Fig-5.1: System Architecture Design

5.3 Website Diagram:

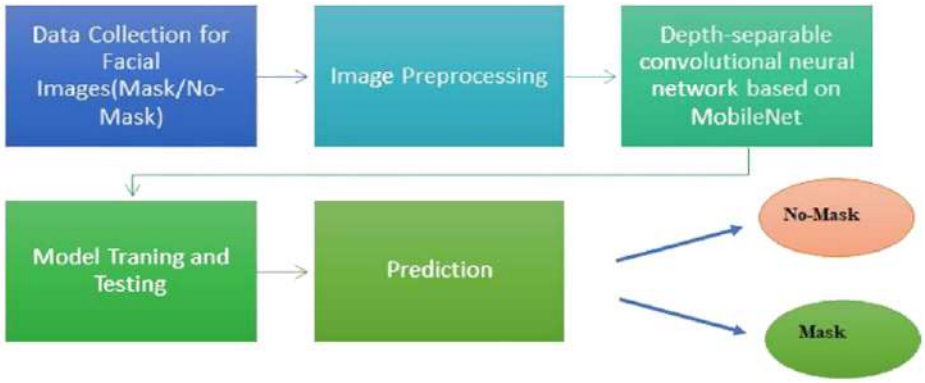

Fig-5.2: Website Diagram

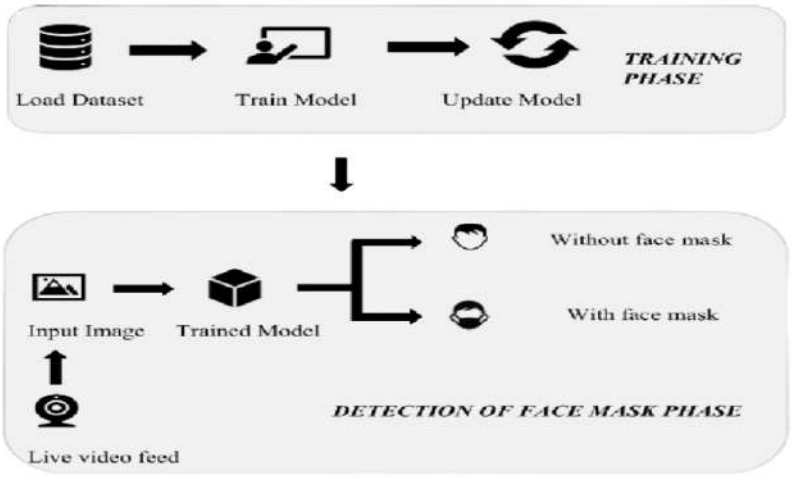

Fig-5.3: Database

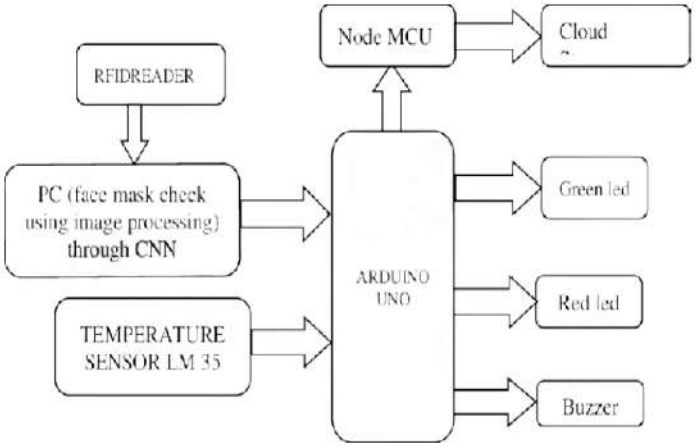

Fig-5.4: Application Diagram

5.4 Flowchart:

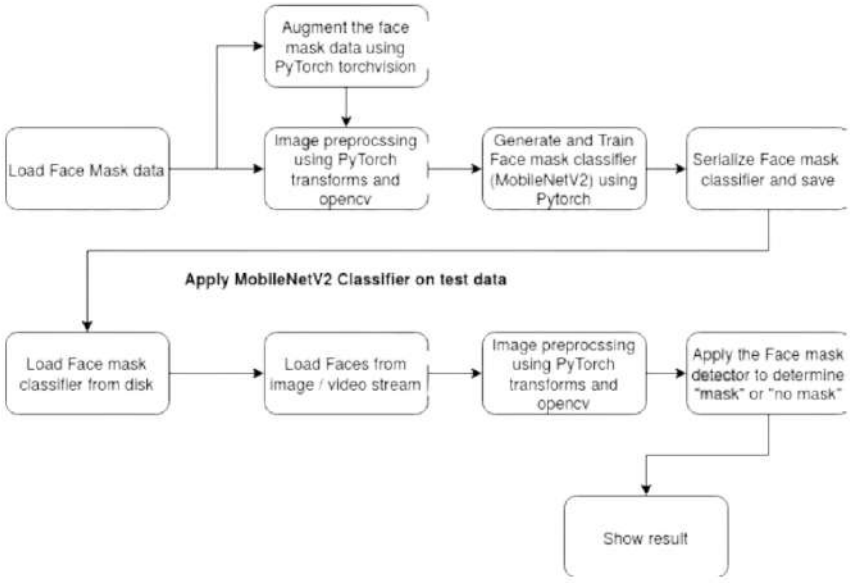

Fig-5.5: Flowchart

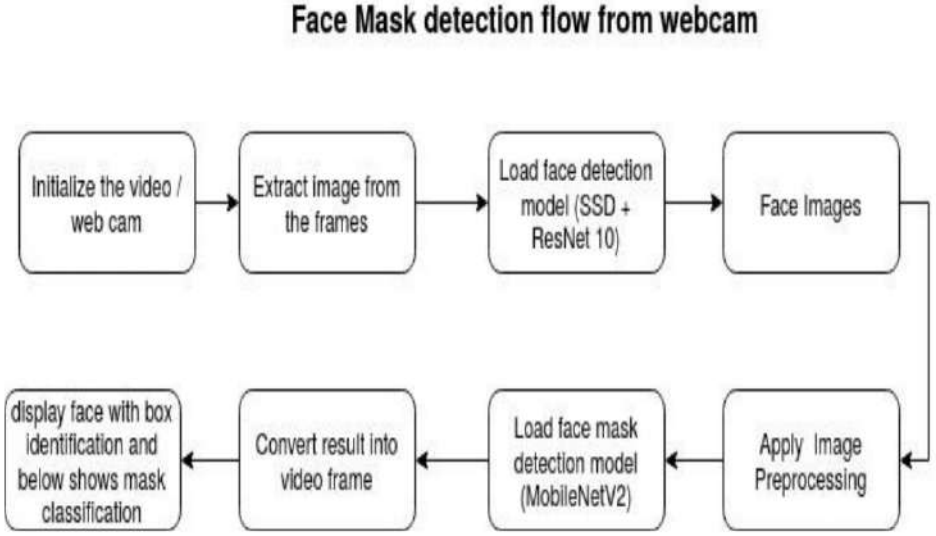

Fig-5.6: Admin Interface Design

5.5. User Interface Design

Face Mask Detection Platform uses an Artificial Network to determine whether or not a user is wearing a mask. To detect persons without masks, the software may be linked to any existing or new IP mask detection cameras. App users may also add faces and phone numbers to get notifications if they are not wearing a mask. If the camera records an unidentified face, an alert can be issued to the administrator. If the face mask detection program determines that a user was not wearing a mask, AI notifications with the user's photo are issued. It enables the program to execute automatically and enforces the mask's use.

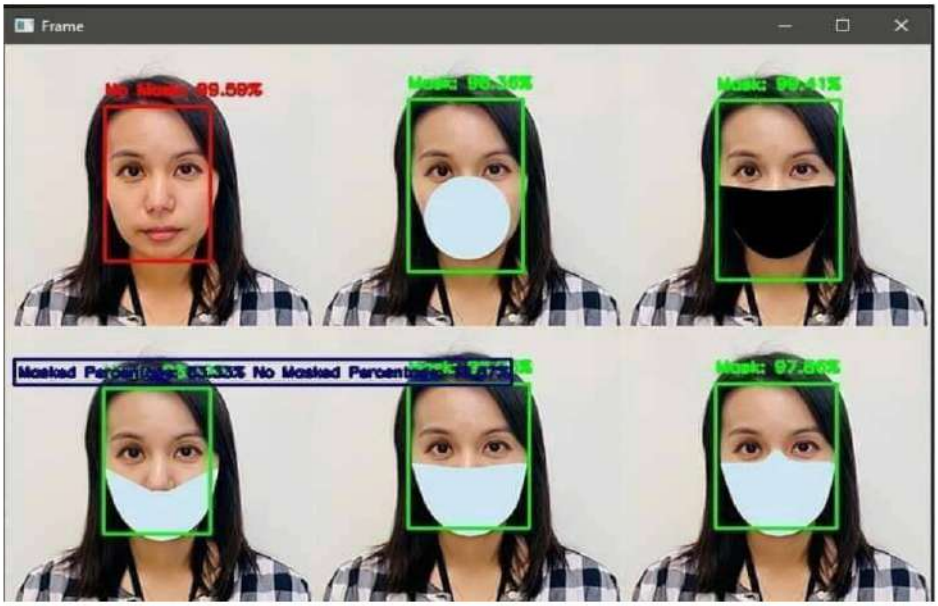

Fig-5.7: User Interface Design

Fig-5.8: User Interface Design

Fig-5.9: User Interface Design

Fig-5.10: User Interface Design

5.6. Admin Interface Design

Convolutional Neural Networks (CNNs) are deep neural networks that are inspired by biological events. A CNN is made up of multiple components, including a convolutional layer, a pooling layer, and a fully connected layer, and it uses the backpropagation method to learn spatial patterns of input independently and fluidly. Because the CNN kernels are shared across full picture locations, it is extremely parameter-efficient. Because of these characteristics, CNN is a viable alternative for computer vision challenges. Deep learning technologies have flourished in recent years as a result of significant developments in GPU computing capabilities. Object identification appears to be a vital job that has received a lot of interest in computer vision.

Fig-5.11: Admin Interface Design

5.7. Database:

The proposed study is divided into two parts: the first is data collecting and dataset preparation, and the second is the presentation of a unique Deep mask Net model building for face mask detection and masked facial recognition. We suggested a unique Deep mask Net model that may be utilized for face mask detection as well as masked facial recognition. Fig. Figure 1 depicts the fundamental process of the suggested technique. The model is made up of ten learnt layers, six convolutional and four fully connected. The three face mask classifier models were developed. trained using our dataset Images from the dataset Faces, both masked and unmasked, were adhered. from publicly available picture collections domain, as well as some data collected from the World Wide Web.

5.8 Application Diagram:

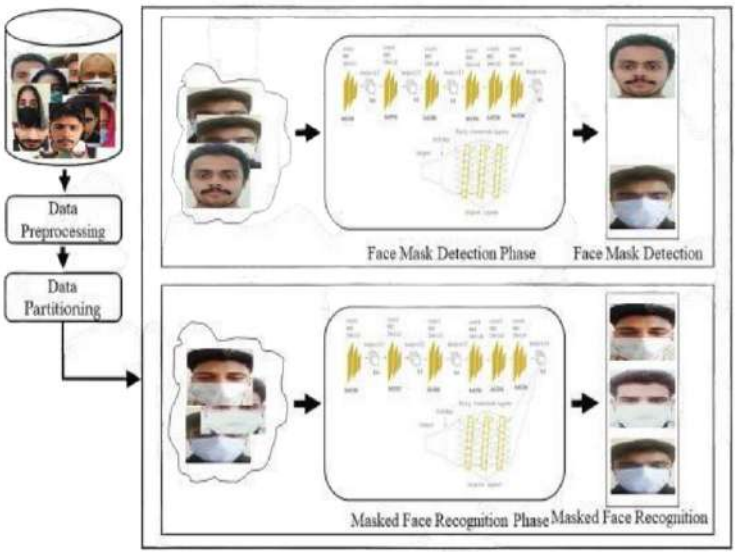

Fig-5.12: Application Diagram

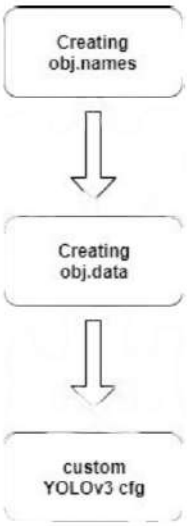

Fig-5.13: Application Diagram

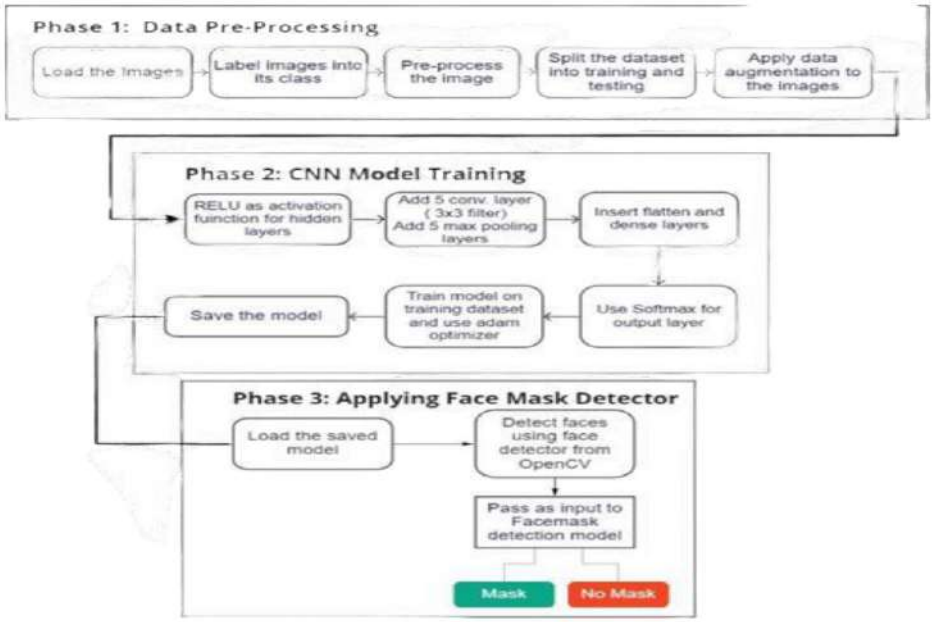

Fig-5.14: Application Diagram

CHAPTER 6
CONCLUSION AND FUTURE WORKS

6.1. Conclusion

As the innovations are blossoming with developing patterns, this model gives a novel face mask detector which can add to public medical care. The architecture comprises MobileNet as the spine, it may be utilized for high and low computation situations. So as to extricate more vigorous highlights, here transfer learning is used to embrace loads from a comparative undertaking face recognition, which is trained on an exceptionally enormous dataset. This model has utilized OpenCV, TensorFlow, Keras, PyTorch and CNN to identify if individuals were wearing face masks. The models were tested with pictures and continuous video streams. The exactness of the model is accomplished and the optimization of the model is a ceaseless cycle and this model creates an exceptionally precise arrangement by tuning the hyper parameters. This particular model could be utilized as a utilization case for edge investigation. Moreover, the proposed technique accomplishes best in class results on a public face mask dataset. By the advancement of face mask detection, we can recognize if the individual is wearing a face cover and permit their entrance would be of extraordinary assistance to the general public.

6.2. Future Work

Face mask detection is an object recognition technique that use image processing algorithms. Classical image processing and deep learning-based image analysis are the two primary types of digital image processing. Deep learning-based techniques use models that mirror the workings of the human brain, as opposed to traditional image analysis, which uses complicated formulae to detect and interpret images. The majority of previous research has relied on Deep Learning models. Kaur et al CNN-based.'s technique properly recognizes the face in the image or video.Mata (12) improved the model's performance by data augmentation. To extract the facial area as a ROI, a CNN model that can distinguish between ROIs with and without a face mask must be created. Toppo et al. (1) used Mobile NetV2 to create a technique for identifying face masks that integrates three separate face detector models to verify the model's accuracy and assess its performance. The conclusion of the trained model enables for implementation on low-power devices, allowing the mask detection approach to be included faster than earlier tactics.Balaji et al. (13) used a VGG-16 CNN model created in Keras/TensorFlow and Open-CV to detect persons who were not wearing face masks in government offices to detect people who were not wearing face masks. Fan et al. (3) proposed two more approaches to adjust for the model's small weight. A unique residual contextual awareness module for critical face mask areas is used to find better mask discrimination features. A two-stage synthetic Gaussian heat map regression is utilized to identify better mask discrimination features. According to ablation research, these tactics can increase feature engineering and hence the efficacy of numeric identification. The proposed model outperforms previous models for AIZOO and Moxa3K.

References:

[1] C. M. Astley and J. S. Brownstein, "Mask-wearing and control of SARSCoV-2 Transmission in the USA: a Cross-sectional study," The Lancet Digital Health, vol. 3, pp. e148-e157, 2021.

[2] "Coronavirus Disease 2019 (COVID-19) – Symptoms", Centers for Disease Control and Prevention.

[3] L. Cuimei, Q. Zhiliang, J. Nan and W. Jianhua, "Human face detection algorithm via Haar cascade classifier combined with three additional classifiers," in 2017 13th IEEE International Conference on Electronic Measurement Instruments (ICEMI), 2017.

[4] M. Jiang, X. Fan and H. Yan, "RetinaMask: A Face Mask Detector," 2020.

[5] M. Jiang, X. Fan and H. Yan, "RetinaMask: A Face Mask Detector", arXiv.org, 2020. [Online].

[6] Parsons, J. J., & Oja, D. (2007). New Perspectives on Computer Concepts, 10th Edition, Comprehensive (New Perspectives Series) (10th ed.). Thomson Course Technology.

[7] Sommerville; I. (2021). Software Engineering by Sommerville Ian (2015–08-20). Pearson; 10 editions (2015–08-20).

[8] Roger S. Pressman. (2021). Software Engineering Software Engineering: A Practitioner"s Approach 6th edition (6th Revised edition). McGraw-Hill Science/Engineering.

[9] Moroney, L. (2017). The Definitive Guide to Firebase: Build Android Apps on Google"s Mobile Platform (1st ed.).

[10] Singh, H., & Tanna, M. (2018). Serverless Web Applications with React and Firebase: Develop real-time applications for web and mobile platforms.

[11] Mishra, S. K. (2020). Fundamentals of Android App Development: Android Development for Beginners to Learn Android Technology, SQLite, Firebase and Unity (English Edition). BPB Publications.