MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

Project Report

on

Face Mask Detection System

Submitted By:

Akansh Verma (0901AM223D01)

Vishal Soni (0901AM211064)

Faculty Mentor:

Prof. Shubha Mishra

CENTRE FOR ARTIFICIAL INTELLIGENCE

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE

GWALIOR - 474005 (MP) est. 1957

JULY-DEC. 2023

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

CERTIFICATE

This is certified that **Vishal Soni** (0901AM211064) & **Akansh Verma** (0901AM223D01) has submitted the project report titled Face Mask Detection System under the mentorship of **Prof.Subha Mishra** in partial fulfilment of the requirement for the award of degree of Bachelor of Technology in AIML from Madhav Institute of Technology and Science, Gwalior.

23/11/2023

Prof. Shubha Mishra
Faculty Mentor
Assistant Professor
Centre for Artificial Intelligence

23/11/23

Dr. R. R. Singh
Professor & Head
Centre for Artificial Intelligence

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

DECLARATION

I hereby declare that the work being presented in this project report, for the partial fulfilment of requirement for the award of the degree of Bachelor of Technology in AIML at Madhav Institute of Technology & Science, Gwalior is an authenticated and original record of my work under the mentorship of **Prof. Subha Mishra** , Centre for Artificial Intelligence.

I declare that I have not submitted the matter embodied in this report for the award of any degree or diploma anywhere else.

Vishal Soni
(0901AM211064)
3rd Year,
Centre for Artificial Intelligence

Akansh Verma
(0901AM223D01)
3rd Year,
Centre for Artificial Intelligence

MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE GWALIOR

(A Govt. Aided UGC Autonomous Institute Affiliated to RGPV, Bhopal)

NAAC Accredited with A++ Grade

ACKNOWLEDGEMENT

The full semester project has proved to be pivotal to my career. I am thankful to my institute, **Madhav Institute of Technology and Science** to allow me to continue my disciplinary/interdisciplinary project as a curriculum requirement, under the provisions of the Flexible Curriculum Scheme (based on the AICTE Model Curriculum 2018), approved by the Academic Council of the institute. I extend my gratitude to the Director of the institute, **Dr. R. K. Pandit** and Dean Academics, **Dr. Manjaree Pandit** for this.

I would sincerely like to thank my department, **Centre for Artificial Intelligence,** for allowing me to explore this project. I humbly thank **Dr. R. R. Singh**, Coordinator, Centre for Artificial Intelligence, for his continued support during the course of this engagement, which eased the process and formalities involved.

I am sincerely thankful to my faculty mentors. I am grateful to the guidance of **Prof.Subha Mishra**, Assistant Professor , Centre for Artificial Intelligence for his continued support and guidance throughout the project. I am also very thankful to the faculty and staff of the department.

Vishal Soni
(0901AM211064)
3rd Year,
Centre for Artificial Intelligence

Akansh Verma
(0901AM223D01)
3rd Year,
Centre for Artificial Intelligence

TABLE OF CONTENT:

Abstract

This report proposes a Face Mask Detection Using OpenCV. This pandemic is causing a worldwide emergency in healthcare. This virus particularly spreads via droplets which emerge from someone infected with coronavirus and poses a risk to others.The risk of transmission is highest in public places. one of the satisfactory ways to live safe from getting inflamed is carrying a face mask in open territories as indicated with the aid of the arena fitness business enterprise (WHO) on this task, we endorse a method which employs TensorFlow and OpenCV to hit upon face mask on people.A bounding container drawn over the face of the person describes weather the man or woman is carrying a mask or no longer.If a person's face is saved within the database, it detects the name of the person that isn't carrying face masks and an e-mail might be sent to that individual caution them that they are not sporting a masks as a way to take precautions .if name not saved in database then we directly imposes live web camera for detecting the person's whether he/she mask wearing or not. If a person's face is saved within the database, it detects the name of the person that isn't carrying face masks and an e-mail might be sentto that individual caution them that they are not sporting a masks as a way to take precautions. Many businesses and organization need to adapt and protect an infected person by detecting whoever does not wear masked face.

Keywords – Tensorflow, Opencv ,Covid 19, image processing ,mask, no mask, pandemic, safety, computer vision.

सार:

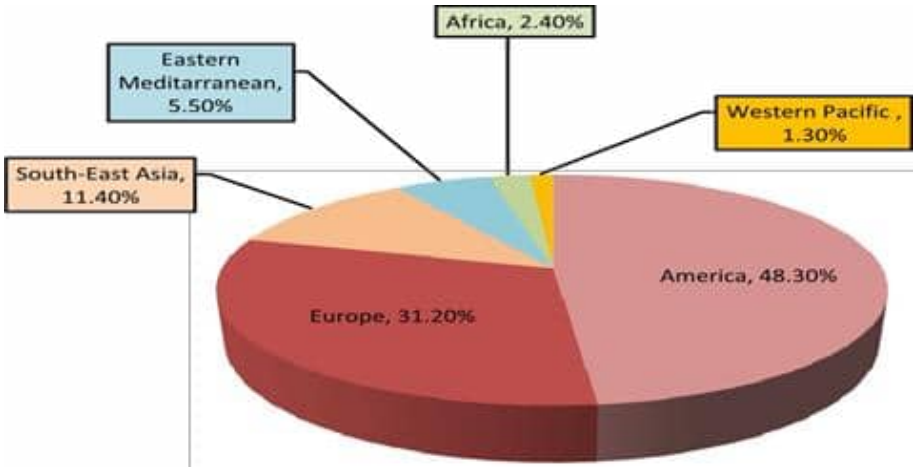
इस रिपोर्ट में हम एक ओपनसीवी (OpenCV) का उपयोग करके फेस मास्क डिटेक्शन (Face Mask Detection) प्रस्तुत कर रहे हैं। यह पैंडेमिक स्वास्थ्य में वैश्विक आपातकालीन स्थिति पैदा कर रहा है। यह वायरस विशेषकर उन बूंदों के माध्यम से फैलता है जो कोरोनावायरस से संक्रमित व्यक्ति से निकलती हैं और दूसरों के लिए खतरा पैदा करती है। संक्रमण का खतरा सबसे अधिक सार्वजनिक स्थानों में होता है। संक्रमित होने से बचने का एक सत्यापन किया गया तरीका खुले क्षेत्रों में एक फेस मास्क पहनना है, जैसा कि विश्व स्वास्थ्य संगठन (WHO) ने सुझावित किया है। इस परियोजना में, हम एक ऐसी विधि की सिफारिश करते हैं जो मानवों पर फेस मास्क की पहचान के लिए TensorFlow और OpenCV का उपयोग करती है। व्यक्ति के चेहरे पर एक बाउंडिंग बॉक्स खींचा जाता है जो यह बताता है कि व्यक्ति एक मास्क पहन रहा है या नहीं। यदि किसी के चेहरे का डेटाबेस में संरक्षित है, तो यह मास्क नहीं पहनने वाले व्यक्ति का नाम पहचानता है और उस व्यक्ति को एक ईमेल भेजा जाता है जिसमें उन्हें चेतावनी दी जाती है कि वे मास्क नहीं पहन रहे हैं और सावधानी बरतनी चाहिए। यदि नाम डेटाबेस में संरक्षित नहीं है, तो हम सीधे लाइव वेब कैमरा का उपयोग करके व्यक्ति को मास्क पहने या नहीं पहने की पहचान करते हैं। यदि किसी के चेहरे का डेटाबेस में संरक्षित है, तो यह मास्क नहीं पहनने वाले व्यक्ति का नाम पहचानता है और उस व्यक्ति को एक ईमेल भेजा जाता है जिसमें उन्हें चेतावनी दी जाती है कि वे मास्क नहीं पहन रहे हैं और सावधानी बरतनी चाहिए। कई कंपनियां और संगठन एक संक्रमित व्यक्ति की सुरक्षा करने के लिए अपनाने और संरक्षित करने की आवश्यकता है, जो मास्क पहने वाले व्यक्ति को पहचानकर किसी को भी संक्रमित होने से बचा सकती है।

Chapter1: Introduction

The pandemic had a huge effect on human lives. The Covid-19 result in the lack of 10millions and damaged the lives of billions of humans. The virus spreads through close contact of humans and in crowded/overcrowded places. Among them cleaning hands, maintaining a safe distance, wearing a mask, refraining from touching eyes, nose, and mouth are the main, where wearing a mask is the simplest one. Unfortunately, people are not following these rules properly which is resulting in speeding the spread of this virus. The solution can be to detect the people not wearing mask and informing their authorities. Its negative result turned into felt by nearly all business establishments for example like education system, religion, tourism, employment, entertainment, and different industries face the losses due to this pandemic. According to World Health Organization, 56.5 million people get infected with virus and 1.44 million people loss their life in 2020 because of the virus came in 2019.From the very basic hygiene requirements to the treatments in the hospitals, people are doing all they could for his or her personal and the society's protection; face masks are one of the private shielding gadget. Humans put on face masks once they step out in their houses and authorities strictly ensure that people are carrying face mask even as they may be in companies and public places After the one person get infected, it almost takes fourteen days to the virus to grow in the human and affect them and within the interval, it flare to almost every body who's come in contact with that affected person. So, itis very hard to stop the spread of coronavirus among peoples. Coronavirus flare through droplets produced from coughing and sneezing through an affected person. This transmit the virus to any person who come indirect contact with the person affected by corona virus because of this, the virus spreads easily among the persons.

1.1 Introduction

The present scenario of COVID-19 demands an efficient face mask detection application. The main goal of the project is to implement this system at entrances of colleges, airports, hospitals, and offices where chances of spread of COVID-19 through contagion are relatively higher. Reports indicate that wearing face masks while at work clearly reduces the risk of transmission. It is an object detection and classification problem with two different classes (Mask and Without Mask). A hybrid model using deep and classical machine learning for detecting face mask will be presented. A dataset is used to build this face mask detector using Python, OpenCV, and TensorFlow and Keras. While entering the place everyone should scan their face and then enter ensuring they have a mask with them. If anyone is found to be without a face mask, beep alert will be generated. As all the workplaces are opening. The number of cases of COVID-19 are still getting registered throughout the country. If everyone follows the safety measures, then it can come to an end. Hence to ensure that people wear masks while coming to work we hope this module will help in detecting it.



number of cases of COVID-19 are still getting registered throughout the country. If everyone follows the safety measures, then it can come to an end. Hence to ensure that people wear masks while coming to work we hope this module will help in detecting it.

Fig 1.1: Cumulative death cases reported by the world health organization (WHO) from different regions globally are shown in this pie chart.

1.2 Formulation of Problem

A face masks detector machine can be carried out to check this.Face masks detection method to identify whether a person is sporting a mask or now not.The face mask detection is a technique to find out whether the person is wearing a mask or not. In this project We will build a real-time system to detect whether the person on the webcam is wearing a mask or not. We will train the face mask detector model using Keras and OpenCV.. We will develope the face mask detector model for detecting whether person is wearing a mask or not. We will train the model using Keras with network architecture. Training the model is the first part of this project and testing using webcam using OpenCV is the second part. The dataset we are working on consists of 1376 images with 690 images containing images of people wearing masks and 686 images with people without masks. we will test the results of face mask

detector model using OpenCV.The proposed model can be integrated with computer or laptop cameras allowing it to detect people who are wearing masks and not wearing masks. The model has been put together using deep learning and classical machine learning techniques with opencv, tensor flow and keras.

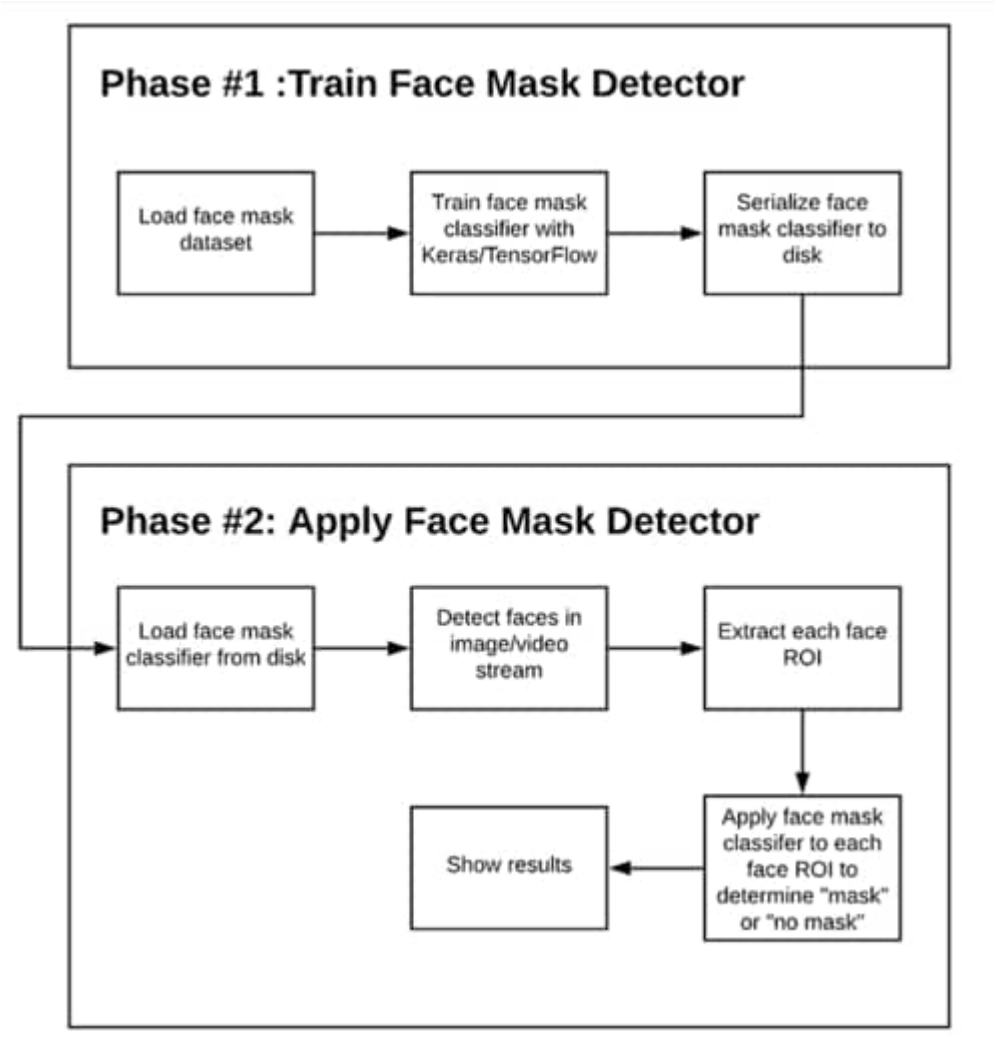


Fig 1.2: Two-phase COVID-19 face mask detector

In order to train a custom face mask detector, we need to break our project into two distinct phases, each with its own respective sub-steps (as shown by **Figure 1** above):

1. Training: Here we'll focus on loading our face mask detection dataset from disk, training a model (using Keras/TensorFlow) on this dataset, and then serializing the face mask detector to disk

2. Deployment: Once the face mask detector is trained, we can then move on to loading the mask detector, performing face detection, and then classifying each face as with_mask or without_mask

1.3 Tool and Technology Used

Category: Machine Learning

Programming Language:

Python Tools & Libraries:

OpenCV,kerasFront End: NA

Back End : python

Chapter3: Implementation

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.

A face masks detector machine can be carried out to check this. Face masks detection method to identify whether a person is sporting a mask or now not. The face mask detection is a technique to find out whether the person is wearing a mask or not. In this project We will build a real-time system to detect whether the person on the webcam is wearing a mask or not. We will train the face mask detector model using Keras and OpenCV.. We will develope the face mask detector model for detecting whether person is wearing a mask or not. We will train the model using Keras with network architecture. Training the model is the first part of this project and testing using webcam using OpenCV is the second part. The dataset we are working on consists of 1376 images with 690 images containing images of people wearing masks and 686 images with people without masks. we will test the results of face mask detector model using OpenCV.The proposed model can be integrated with computer or laptop cameras allowing it to detect people who are wearing masks and not wearing masks. The model has been put together using deep learning and classical machine learning techniques with opencv, tensor flow and keras.

First a base model is generated. This is done my using Keras . First a base model is generated and a head model is generated on top of that. The generated version is then trained with the labeled dataset by way of splitting it into portions. One portion incorporates 75 percentage photographs and it's far used for training. The closing component includes the remaining 25 percent of pictures and is used for testing the version accuracy. After the model is trained, it is able to be used for detection of facemask on human faces. A person without the background. This face is given because the enter to the model which we skilled in advance. This outputs weather there is a mask or not. Another model is educated with the faces of humans. The images used for the training of the model are provided with the name and email address of that person because the labels of those snap shots. that is performed by the usage of Open CV. while an input image is given to the CV version, it detects the face of someone and asks the consumer to provide the name and e mail address of that man or woman so one can be saved in the database. The output of the primary version is given because the input to this version. This face can be in comparison with the folks gift in the database. And if his face matches, then a bounding field will be drawn over his face along with his name on it and an email and Sms could be sent to him that he isn't always wearing a masks. Else, only the phrases "mask" may be gift underneath the bounding box if the individual is sporting a masks and "No mask" if the person is not carrying one. that is done by means of the usage of OpenCV. When an input image is given to the CV model, it detects the face of a person and asks the user to provide the call and e mail address of that man or woman so as to be stored within the database. The output of the first model is given as the enter to this model. This face could be compared with the humans present in the database. And if his face suits, then a bounding box can be drawn over his face along with his call on it and an email and Sms can be sent to him that he isn't always sporting a mask. Else, simplest the phrases "masks" can be gift under the bounding field if the person is

wearing a mask and "No mask" if the character isn't always carrying one.

THE PROPOSED METHOD

The proposed method consists of a cascade classifier and a pre-trained CNN which contains two 2D convolution layers connected to layers of dense neurons. The algorithm for face mask detection is as follows:

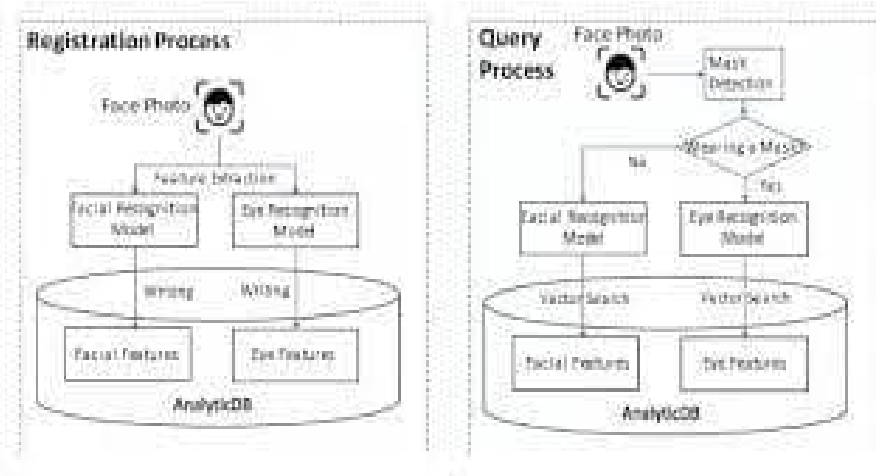


Fig 3.1

A. Data Processing

Data preprocessing involves conversion of data from a given format to much more user friendly, desired and meaningful format. It can be in any form like tables, images, videos, graphs, etc. These organized information fit in with an information model or composition and captures relationship between different entities . The proposed method deals with image and video data using Numpy and OpenCV.

a) **Data Visualization:** Data visualization is the process of transforming abstract data to meaningful representations using knowledge communication and insight discovery through encodings. It is helpful to study a particular pattern in the dataset . The total number of images in the dataset is visualized in both categories – 'with mask' and 'without mask'. The statement categories=os.listdir(data path) categorizes the list of directories in the specified data path. The variable categories now looks like: ['with mask', 'without mask'] Then to find the number of labels, we need to distinguish those categories using labels=[i for i in range(len(categories))]. It sets the labels as: [0, 1] Now, each category is mapped to its respective label using label dict=dict(zip(categories,labels)) which at first returns an iterator of tuples in the form of zip object where the items in each passed iterator is paired together consequently. The mapped variable label dict looks like: f'with mask': 0, 'without mask': 1g

b) **Conversion of RGB image to Gray image:** Modern descriptor-based image recognition systems regularly work on grayscale images, without elaborating the method used to convert from color-to-grayscale. This is because the colortograyscale method is of little consequence when using robust descriptors. Introducing nonessential information could increase the size of training data required to achieve good performance. As grayscale rationalizes the algorithm and diminishes the computational requisites, it is utilized for extracting descriptors instead of working on color images instantaneously . Fig. Conversion of a RGB image to a Gray Scale image of 100 x 100 size We use the function cv2.cvtColor(input image, flag) for changing the color space. Here flag determines the type of conversion . In this case, the flag cv2.COLOR BGR2GRAY is used for gray conversion. Deep CNNs require a fixed-size input image. Therefore we need a fixed common size for all the images in the dataset. Using cv2.resize() the gray scale image is resized into 100 x 100.

c) **Image Reshaping:** The input during relegation of an image is a three-dimensional tensor, where each channel has a prominent unique pixel. All the images must have identically tantamount size corresponding to 3D feature tensor. However, neither images are customarily coextensive nor their corresponding feature tensors . Most CNNs can only accept fine-tuned images. This engenders several problems throughout data collection and implementation of model. However, reconfiguring the input images before augmenting them into the network can help to surmount this constraint. The images are normalized to converge the pixel range between 0 and 1. Then they are converted to 4 dimensional arrays using data=np.reshape(data,(data.shape[0], img size,img size,1)) where 1 indicates the Grayscale image. As, the final layer of the neural network has 2 outputs – with mask and without mask i.e. it has categorical representation, the data is converted to categorical labels.

B. Training of Model

Deep Learning:

Deep learning is a branch of machine learning which is based on artificial neural networks. It is capable of learning complex patterns and relationships within data. In deep learning, we don't need to explicitly program everything. It has become increasingly popular in recent years due to the advances in processing power and the availability of large datasets. Because it is based on artificial neural networks (ANNs) also known as deep neural networks (DNNs). These neural networks are inspired by the structure and function of the human brain's biological neurons, and they are designed to learn from large amounts of data. Convolutional Neural Networks (CNNs): CNNs are a type of deep neural network particularly effective for image-related tasks.

In face mask detection, CNNs can learn hierarchical representations of image features, enabling the model to recognize patterns associated with faces and masks.

a) Feature Extraction: Deep learning models, especially CNNs, excel at automatically extracting relevant features from images. These features can include facial landmarks, textures, and spatial relationships, which are essential for accurate mask detection.

b) Transfer Learning: Pre-trained models, such as those trained on large datasets like ImageNet, can be utilized as a starting point for face mask detection models. Transfer learning allows leveraging the knowledge gained from one task (e.g., general image recognition) to another task (e.g., face mask detection), even when the target task has a smaller dataset.

c) Object Detection Architectures: Face mask detection involves locating and classifying objects in an image, making object detection architectures like Faster R-CNN (Region-based Convolutional Neural Network) or YOLO (You Only Look Once) suitable for this task. These architectures enable the model to identify and localize faces and masks within an image.

d) Real-time Detection: Deep learning models optimized for speed, such as Single Shot Multi Box Detector (SSD) or Mobile Net, are commonly used for real-time face mask detection applications. These models are designed to provide fast inference while maintaining reasonable accuracy.

e) Edge Computing: For applications where real-time processing is critical, edge computing can be employed to deploy the face mask detection model on edge devices (e.g., cameras or IoT devices) rather than relying solely on centralized servers.

METHODOLOGY

System design The major requirement for implementing this project using python programming language along with Deep learning ,Machine learning , Computer vision and also with python libraries. The architecture consists of Mobile Net as the backbone, it can be used for high and low computation scenarios. We are using Algorithm in our proposed system. Implementation: We have four modules Datasets Collecting : We collect no of data sets with face mask and without masks. we can get high accuracy depends on collecting the number of images . Datasets Extracting: We can extract the features using mobile net v2 of mask and no mask sets Models Training: We will train the the model using open cv,keras (python library). Facemask Detection : We can detect Pre processing image and also detect via live video . If people wear mask, it will permit them, if not then it will give the buzzer to wear mask to prevent them from virus transmission.

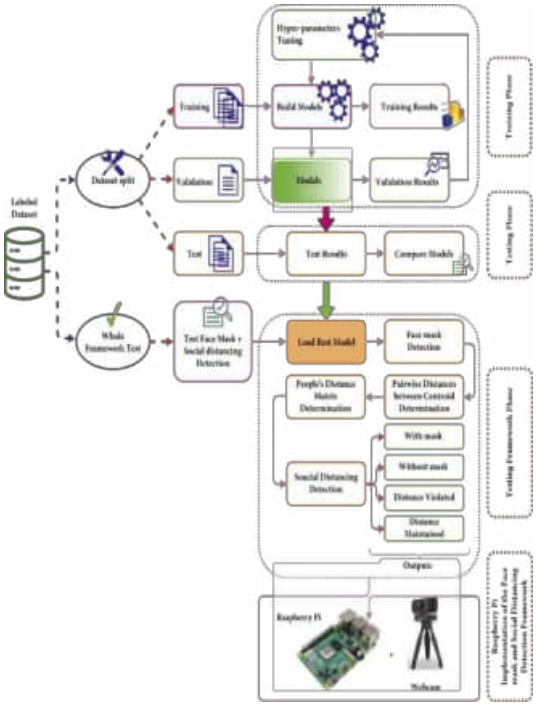


Fig 3.2 Proposed framework for the face mask and social distancing.

BENEFITS

1. **Public Health Compliance:**
    - **Enforcement of Safety Measures:** Face mask detection systems help enforce public health guidelines by ensuring compliance with mask-wearing regulations, contributing to the overall safety of communities.

2. **Disease Transmission Mitigation:**
    - **Reduced Transmission Risk:** By identifying individuals not wearing masks, the system aids in minimizing the risk of disease transmission, especially in crowded public spaces such as transportation hubs, healthcare facilities, and commercial establishments.

3. **Healthcare Settings:**
    - **Patient and Staff Safety:** In healthcare facilities, the system enhances safety measures by ensuring that both patients and healthcare providers are adhering to recommended protective measures.

4. **Educational Institutions:**
    - **Secure Learning Environment:** In schools and universities, the face mask detection system contributes to creating a secure learning environment, promoting the health and well-being of students, teachers, and staff.

5. **Commercial Spaces:**
    - **Employee and Customer Safety:** Businesses benefit from the system by reinforcing safety protocols for employees and customers, fostering a secure and confidence-inspiring environment.

6. **Transportation Hubs:**
    - **Public Transportation Safety:** The system can be deployed in transportation hubs to monitor and enforce mask-wearing compliance on public transportation, reducing the risk of virus spread in confined spaces.

3.1 Module Description

PYTHON: Python's simplicity, related to its easily readable code, gives programmers confidence in the projects they work on. It's stable, flexible, and gives developers access to a variety of tools that make their jobs easier. Machine Learning projects rely on complex algorithms, leaving very little room for error.

OPENCV: OpenCV is a great tool for image processing and performing computer vision tasks. It is an open-source library that can be used to perform tasks like face detection, objection tracking, landmark detection, and much more. It supports multiple languages including python, java C++.OpenCV provides a real-time optimized Computer Vision library, tools, and hardware. It also supports model execution for Machine Learning (ML).

TENSORFLOW: TensorFlow is an end-to-end open source platform for machine

learning. It is an open source artificial intelligence library, using data flow graphs to build models. It allows developers to create large-scale neural networks with many layers. TensorFlow is mainly used for: Classification, Perception, Understanding, Discovering, Prediction and Creation.

Dataset: Data collection is the process of gathering and measuring information from countless different sources. In order to use the data we collect to develop practical artificial intelligence (AI) and machine learning solutions, it must be collected and stored in a way that makes sense for the business problem at hand.

Train a model to catch face masks: We will build a real-time system to detect whether the person on the webcam is wearing a mask or not. We will train the face mask detector model using Keras and OpenCV.

Detect the individual not wearing face mask: Model was educated to discover the names of the people who are not sporting mask by referring the database.

Numpy,cv2: OpenCV is a widely usedopen-source library for computer vision. It includes several ready to use computer vision algorithms. Python is becoming the standard programming language for AI and NumPy. provides data structures used to deploy OpenCV with Python.

WORKFLOW DIAGRAM:

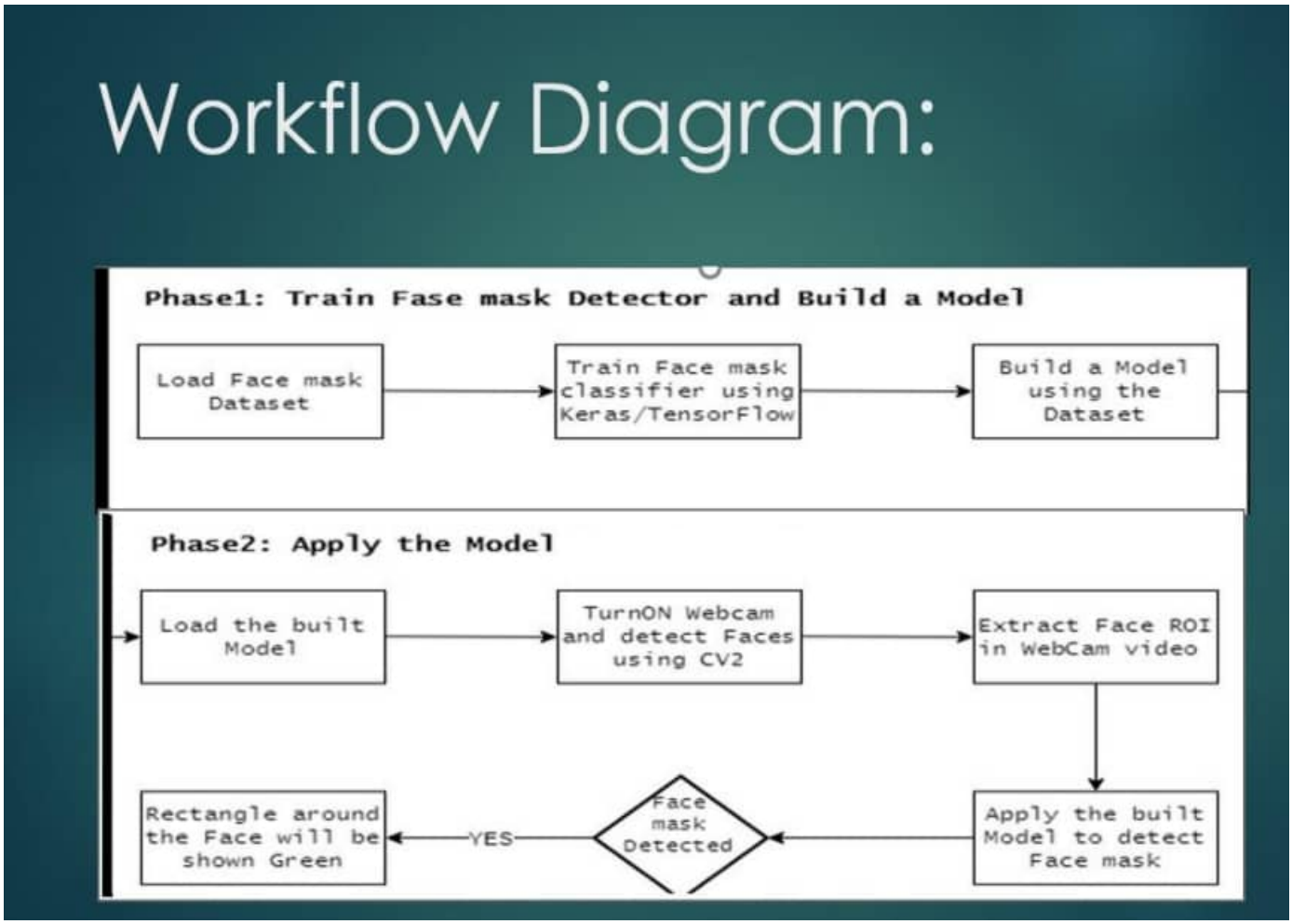


Fig 3.1: FlowChart of the project

Fig 3.2: Person With Mask and Without Mask.

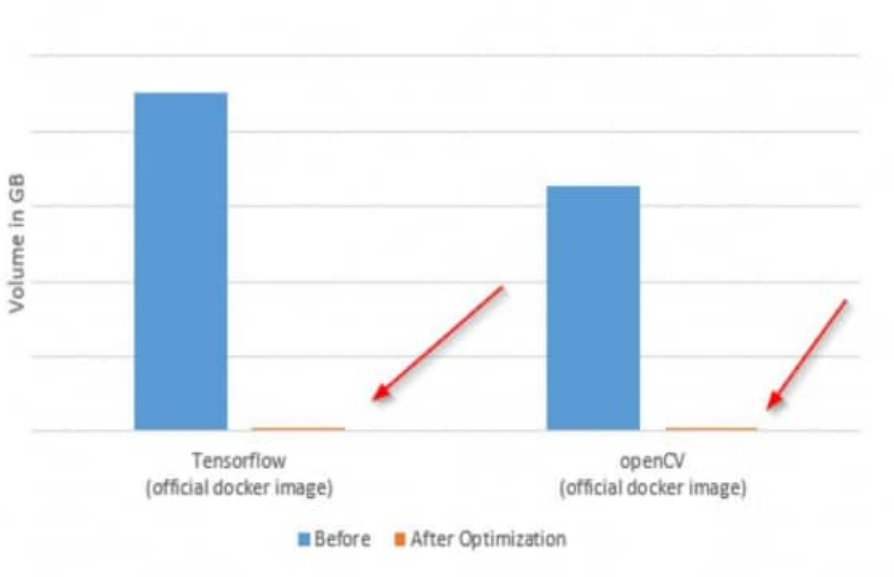


Fig 3.3 Face Mask Detection With ML.

Chapter4: Experiment and results

Fig 4.1

```
import tensorflow as tf
import cv2
import os
import matplotlib.pyplot as plt
import numpy as np
```

Python

```
img_array= cv2.imread("datasets\\with_mask\\00000_Mask.jpg")
```

Python

```
plt.imshow(img_array)
```

Python

```
<matplotlib.image.AxesImage at 0x13557c5ff90>
```

Fig 4.2

```
#normalize the data
X = X/225.0
```

```
import pickle

pickle_out = open("X.pickle","wb")
pickle.dump(X, pickle_out)
pickle_out.close()

pickle_out= open("y.pickle","wb")
pickle.dump(y,pickle_out)
pickle_out.close()
```

```
pickle_in = open("X.pickle", "rb")
X = pickle.load(pickle_in)

pickle_in = open("y.pickle", "rb")
y= pickle.load(pickle_in)
```

Deep learning model for training - Transfer learning

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
```

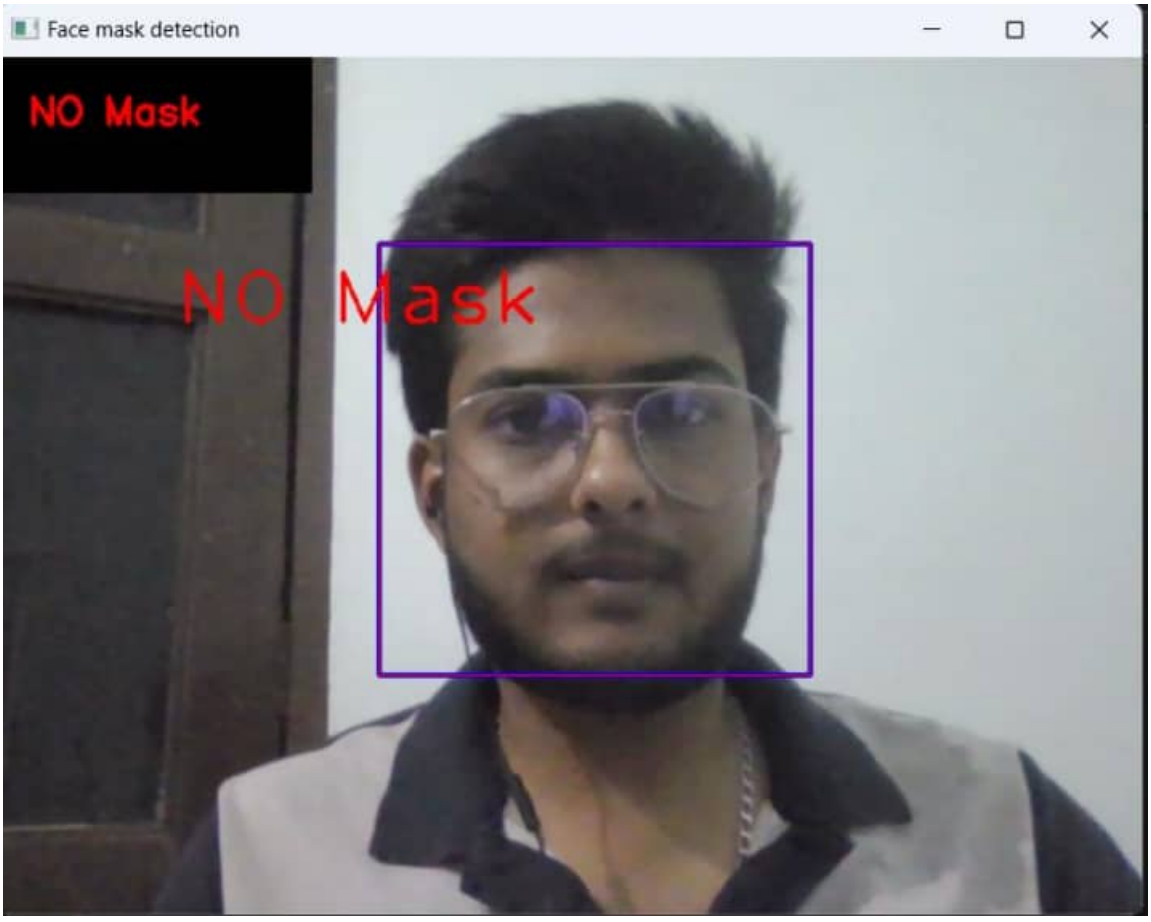
Fig 4.3: Dataset with mask

Fig 4.4: Dataset with mask and no mask

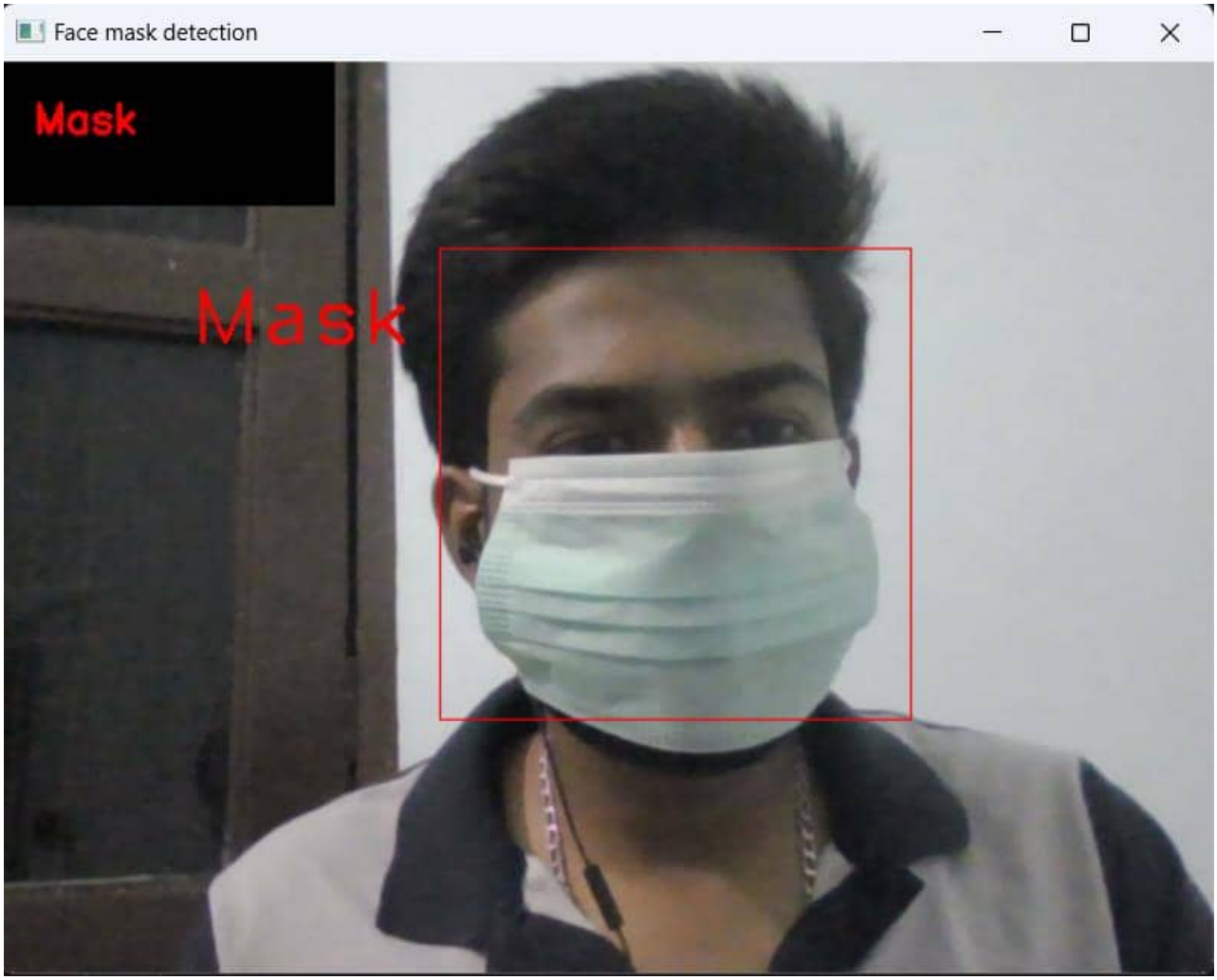
RESULTS:

By preserving a reasonable proportion of different classes, the dataset will partition into training and testing set. The dataset will comprises of 1315 samples in total where 80% will be used in training phase and 20% will get used in testing phase. The develop architecture will trained for 10 epochs since further training results will cause overfitting on the training data. Overfitting generally occurs when a model learns the unwanted patterns of the training samples. Hence, training accuracy increases but test accuracy decreases.

Fig 4.3: When The Person Not Wearing the Mask . A bounding box drawn over the face of the person describes weather the person is wearing a mask or not. If a person's face is stored in the database, it detects the name of the person who is not wearing face mask.



Mask:



Code :

```
import tensorflow as tf
import cv2
import os
import matplotlib.pyplot as plt
import numpy as np

img_array= cv2.imread("datasets\\with_mask\\00000_Mask.jpg")
plt.imshow(img_array)
plt.imshow(cv2.cvtColor(img_array, cv2.COLOR_BGR2RGB))
img_array.shape

Datadirectory = "datasets/" ##training dataset
Classes = ["with_mask","without_mask"]  ## List of classes
for category in Classes:
    path=os.path.join(Datadirectory,category)
    for img in os.listdir(path):
        img_array = cv2.imread(os.path.join(path,img))
        #blacktorgb = cv2.cvtColor(img_array,cv2.COLOR_GRAY2RGB)
        plt.imshow(cv2.cvtColor(img_array,cv2.COLOR_BGR2RGB))
        plt.show()
        break
    break

img_size = 224 ##ImageNet = 224*224

new_array = cv2.resize(img_array,(img_size,img_size))
plt.imshow(cv2.cvtColor(new_array,cv2.COLOR_BGR2RGB))
plt.show()

training_Data = [] ##data

def create_training_Data():
    for category in Classes:
        path = os.path.join(Datadirectory,category)
        class_num = Classes.index(category) ## 0  1,
        for img in os.listdir(path):
            try:
                img_array=cv2.imread(os.path.join(path,img))
                new_array=cv2.resize(img_array,(img_size,img_size))
                training_Data.append([new_array,class_num])
            except Exception as e:
                pass
create_training_Data()

print(len(training_Data))


import random
random.shuffle(training_Data)
```

```
X= [] #data features
y= [] #label

for features,label in training_Data:
    X.append(features)
    y.append(label)

X = np.array(X).reshape(-1,img_size,img_size,3)

import pickle

pickle_out = open("X.pickle","wb")
pickle.dump(X, pickle_out)
pickle_out.close()

pickle_out= open("y.pickle","wb")
pickle.dump(y,pickle_out)
pickle_out.close()

pickle_in = open("X.pickle", "rb")
X = pickle.load(pickle_in)

pickle_in = open("y.pickle", "rb")
y= pickle.load(pickle_in)
```

Deep learning model for training - Transfer learning

```
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers

model = tf.keras.applications.mobilenet.MobileNet() #pre trained model

base_input = model.layers[0].input

base_output = model.layers[-4].output

Flat_layer = layers.Flatten()(base_output)
final_output = layers.Dense(1)(Flat_layer)
final_output= layers.Activation('sigmoid')(final_output)

new_model = keras.Model(inputs = base_input,outputs= final_output)

new_model.compile(loss ="binary_crossentropy",optimizer="adam",metrics=["accuracy"])

new_model.fit(X,Y, epochs=1,validation_split=0.1)

evaluation = new_model.evaluate(X, Y)
print("Loss:", evaluation[0])
print("Accuracy:", evaluation[1])

new_model.save('face_mask_detection_model.h5')
```

```
new_model = tf.keras.models.load_model('face_mask_detection_model.h5')

faces = faceCascade.detectMultiScale(gray,1.1,4)
for x,y,w,h in faces:
    roi_gray = gray[y:y+h, x:x+w]
    roi_color = frame[y:y+h, x:x+w]
    cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2)
    facess = faceCascade.detectMultiScale(roi_gray)
    if len(facess) == 0:
        print("Face not detected")
    else:
        for (ex,ey,ew,eh) in facess:
            face_roi = roi_color[ey:ey+eh, ex:ex + ew]

plt.imshow(cv2.cvtColor(frame,cv2.COLOR_BGR2RGB))
plt.imshow(cv2.cvtColor(face_roi,cv2.COLOR_BGR2RGB))

final_image = cv2.resize(face_roi,(224,224))
final_image = np.expand_dims(final_image,axis=0)
final_image = final_image/255.0
```

Predictions

```
import cv2
path  = "haarcascade_frontalface_default.xml"
font_scale = 1.5
font = cv2.FONT_HERSHEY_PLAIN

import numpy as np
import tensorflow as tf

# ... (Previous code)

cap = cv2.VideoCapture(1)
if not cap.isOpened():
    cap = cv2.VideoCapture(0)
if not cap.isOpened():
    raise IOError("Cannot open webcam")

while True:
    ret, frame = cap.read()
    faceCascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml')
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(gray, 1.1, 4)

    for x, y, w, h in faces:
        roi_gray = gray[y:y+h, x:x+w]
        roi_color = frame[y:y+h, x:x+w]
        cv2.rectangle(frame, (x, y), (x+w, y+h), (255, 0, 0), 2)
        facess = faceCascade.detectMultiScale(roi_gray)
```

```
        if len(facess) == 0:
            print("Face not detected")
        else:
            for (ex, ey, ew, eh) in facess:
                face_roi = roi_color[ey:ey+eh, ex:ex+ew]

    final_image = cv2.resize(face_roi, (224, 224))
    final_image = np.expand_dims(final_image, axis=0)
    final_image = final_image / 255.0

    Predictions = new_model.predict(final_image)

    font_scale = 1.5
    font = cv2.FONT_HERSHEY_PLAIN

    if Predictions  [0][0] > 0.5:
        status = "Mask"
        x1, y1, w1, h1 = 0, 0, 175,75
        cv2.rectangle(frame, (x1, y1), (x1 + w1, y1+h1), (0, 0, 0), -1)
        cv2.putText(frame, status, (x1 + int(w1/10), y1+int(h1/2)), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 0, 255),
2)
        cv2.putText(frame, status, (100, 150), font, 3, (0, 0, 255), 2, cv2.LINE_4)
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 0, 255))
    else:
        status = "Face Mask"
        x1, y1, w1, h1 = 0, 0, 175,75
        cv2.rectangle(frame, (x1 + int(w1/10), y1 + int(h1/2)), cv2.FONT_HERSHEY_SIMPLEX, 0.7, (0, 255, 0), 2)
        cv2.putText(frame, status, (100, 150), font, 3, (0, 0, 255), 2, cv2.LINE_4)
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0))

    cv2.imshow('Face mask detection ', frame)

    if cv2.waitKey(2) & 0xFF == ord('q'):
        break

cap.release()
cv2.destroyAllWindows()
```

Chapter 5: Conclusion and Future Scope

The work proposed in the system focuses on the important challenge faced by the world during the current times due to the ongoing COVID-19 pandemic. The proposed research work has successfully combined the face mask detection model with the person identification model, which is also be able to send mail notifications to the registered people on our platform who are not wearing a mask. Also, this research work has successfully detected multiple people without wearing a mask or with a mask in a single frame of video. This third eye technology focuses on the complicated work of detecting multiple people at once to ensure that people stay safe in these troubled times by ensuring that they follow the guidelines which are issued by the government.

5.2: Future Scope

With the growing quantity of COVID cases all around the global, a machine to update human beings to check masks on the faces of humans is significantly wanted. This system satisfies that need. This machine may be employed in public places like railway stations and department shops. it is going to be of a top notch help in corporations and big establishments where there will be quite a few workers. This gadget may be of a high-quality assist there as it is easy to attain and save the information of the personnel working in that organisation and will very clean locate the folks who are now not sporting the masks and a mail will sent to that respective man or woman to take Precautions not sporting masks, if there data already stored in system.

REFRENCE:

[1] Wuttichai Vijitkunsawat, Peerasak Chantngram on" Machine Learning Algorithms for Face Mask Detection" doi: 10.1109/InCIT5058.2020.9310963

[2] Susanto Febri ,Alwan Putra , Riska Analia ,Ika Karlina Laila Nur Suciningtyas" Face Mask Detection For Preventing the Spread of COVID-19" doi: 10.1109/ESCI50559.2021.9396783

[3] Wei Bu∗†, Jiangjian Xiao†, Chuanhong Zhou∗, Minmin Yang‡, Chengbin Peng† DOI:10.1109/TSP52935.2021.9522677

[4] Sahana Srinivasan ,Rujula Singh R ,Ruchita R Biradar ,Revathi "Face Mask Detection on Surveillance video datasets" DOI: 10.1109/ICCIS.2017.8274819

[5] Ravi Kishore Kodali and Rekha Dhanekula" FACE MASK DETECTION USING DEEP LEARNING" doi: 10.1109/ESCI50559.2021.9396783

[6] Baluprithviraj.K.N ,Bharathi.K.R ,Chendhuran.S" Artificial Intelligence based Smart Door with Face Mask Detection"

[7] Gayatri Deore, Ramakrishna Bodhula,Dr. Vishwas Udpikar, Prof. Vidya More" Masked Face Detection Approach in Video Analytics" Doi: 10.1109/CASP.2016.7746164