CAM-VISION
(AN OBJECT DETECTOR)

A Project-I Report

Submitted in partial fulfillment of requirement of

the Degree of

BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING

BY

Arpit Paliwal **EN19CS301073**

Arpit Purohit **EN19CS301074**

Aroopv Krishna Patidar **EN19CS301068**

Under the Guidance of

Ms. SNEHAL MOGHE

Department of Computer Science & Engineering
Faculty of Engineering
MEDI-CAPS UNIVERSITY, INDORE- 453331

NOVEMBER 2022

Report Approval

The project work **"Cam-Vision (An Object Detector)"** is hereby approved as a creditable study of an engineering subject carried out and presented in a manner satisfactory to warrant its acceptance as prerequisite for the Degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approve any statement made, opinion expressed, or conclusion drawn therein; but approve the "Project Report" only for the purpose for which it has been submitted.

Internal Examiner
Name: Designation
Affiliation

External Examiner
Name: Designation
Affiliation

Declaration

We hereby declare that the project entitled **“Cam-Vision”** submitted in partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Business Systems, Department of Computer Science and Engineering completed under the supervision of **Ms. Snehal Moghe, Assistant Professor and Department of Computer Science & Engineering,** Faculty of Engineering, Medi-Caps University, Indore is an authentic work.

Further, we declare that the content of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted to any other Institute or University for the award of any degree or diploma.

Signature and name of the student(s) with date

Arpit Paliwal (EN19CS301073)
Arpit Purohit (EN19CS301074)
Apoorv Krishna Patidar(EN19CS301068)

Date: November 2022

Certificate

I **Snehal Moghe** certify that the project entitled **“Cam-Vision”** submitted in partial fulfillment for the award of the degree of Bachelor of Technology/Master of Computer Applications by **Arpit Paliwal, Arpit Purohit and Apoorv Krishna Patidar** is the record carried out by them under my guidance and that the work has not formed the basis of award of any other degree elsewhere.

____

Ms. Snehal Moghe

CSE Department

Medi-Caps University, Indore

____

Dr. Pramod S. Nair

Head of the Department

Computer Science & Engineering

Medi-Caps University, Indore

Acknowledgements

I would like to express my deepest gratitude to Honorable Chancellor, **Shri R C Mittal**, who has provided me with every facility to successfully carry out this project, and my profound indebtedness to **Prof. (Dr.) Dileep K. Patnaik**, Vice Chancellor, Medi-Caps University, whose unfailing support and enthusiasm has always boosted up my morale. I also thank **Prof. (Dr.) D K Panda,** Pro Vice Chancellor**, Dr. Suresh Jain,** Dean Faculty of Engineering, Medi-Caps University, for giving me a chance to work on this project. I would also like to thank my Head of the Department **Dr. Pramod S. Nair** for his continuous encouragement for betterment of the project.

I express my heartfelt gratitude to my Internal Guide/Project coordinator **Ms. Snehal Moghe** without whose continuous help and support, this project would ever have reached completion.

THANKS AGAIN TO ALL WHO HELPED ME.

Without their support this report would not have been possible.

Arpit Paliwal
Arpit Purohit
Apoorv Krishna Patidar

B.Tech. III rd. Year
Department of Computer Science & Engineering
Faculty of Engineering
Medi-Caps University, Indore

Executive Summary

We built our project for surveillance purposes to reduce unwanted happenings like robbery, not wearing a helmet while driving a 2-wheeler, not wearing a mask when required, like this there are many more events or happening that can be tracked by our project as requested by the client.

So, our main motive to build this project is to reduce human efforts, error and to build more accurate results than humans by using some technologies.

To build this project we have downloaded some dependencies on our system like PYQT-5, TensorFlow, open-CV, Tf-Record, proto-buf and many more…

We have created our own road map to build this project.

Here it is:

1.) Created a Labelle image system using PYQT -5 for custom labeling image, then we labeled our dataset for training and testing.
2.) Next, we train and run a custom object detector for customized detection.
3.) Now using transfer learning, we trained our own custom detection.
4.) Setting up proper workflow and directory structure.
5.) Making detection of objects in real time using our trained model.

While making all these things possible in the backend we were also building our frontend side by side using HTML, CSS, JavaScript.

After completing backend and frontend we connected both using Django.

We have followed an iterative model to complete each stage of the project.

So, this was a short summary of our project and the method we have used to accomplish this project.

Table of Contents

List Of Figures

Abbreviations

- GUI : Graphical User Interface.
- openCV: Open Source Computer Vision.
- numpy: numerical python.
- UI: User Interface.
- SDLC: Software development life cycle.

Chapter 1

1.1 Introduction

Real time object detection could bring revolution in the world of sur- veillance, security and mankind. Real time object detector can detect any object as per the requirement like bag, helmet, mask, human, animal or anything you want to track down. It's a machine learning project where we train our model to detect certain objects with the help of a labeled image.
Then we capture a real time view with the help of C.C.T.V. or webcam or any type of camera. Then our model will talk about object status. This object detector can be used anywhere for surveillance as per requirement of the client. Let's see some scenarios as examples where real time object detec- tor can be beneficial.

1.2 Literature Review

The aim of object detection is to detect all instances of objects from a known class, such as people,cars or faces in an image. Generally, only a small number of instances of the object are present in the image, but there is a very large number of possible locations and scales at which they can occur and that need to somehow be explored. Each detection of the image is reported with some form of poseinformation. This is as simple as the location of the object, a location and scale, or the extent of the object defined in terms of a bounding box. In some other situations, the pose information is more detailed and contains the parameters of a linear or non-linear transformation. For example, face detection in a face detector may compute the locations of the eyes, nose and mouth, in addition to the bounding box of the face.
In various fields, there is a necessity to detect the target object and also track them effectively while handling occlusions and other included complexities. Many researchers (Almeida and Guting 2004,Hsiao-Ping Tsai 2011, Nicolas Papadakis and Aure lie Bugeau 2010) attempted various approaches in object tracking. The the nature of the techniques largely depends on the applicationdomain.

1.3 0bjectives

1) In this ongoing pandemic of covid people are supposed to wear mask and need to maintain social distance. So, our government plan a lock- down for this which was a quite best optimal decision to control the rate of spread of the virus. But lockdown cannot be stretched for long period of time. So, the government decided to terminate the lockdown with guidelines like to maintain social distance, keep sanitizing hands, and wear masks all time

while traveling out. But some casual people who don’t take this thing seriously the don’t wear mask while traveling they go just like that everywhere including public places like mall, railway station, hospital etc. while they en- ter such places the guard at place keep on telling them to wear mask which people listen at gate but further going ahead, they remove the mask. Even the guard has the chance to get infected by that person or the person who is not wearing the mask has same probability of getting infected if anyone of them is not wearing mask. So, in this scenario we will build our real time object detector which will detect the mask whether the person is wearing or not. We will detect this with the help of camera placed at entrance of any public place such as gate of mask. Now the gate can be opened if it’s merged with the program or it can open from the control room where person can easily watch on screen and can control the entrance or door from just sitting there. Like this we can put camera everywhere in place suppose any one removed his mask after getting entry than our model detects and that can be seen from the control room and strict action can be taken against that person for not following guideline

.

2) Road’s accidents are very life risking; risk increases if accidents occur with two- wheeler. Even the death rate of road accidents is more than covid. This risk could be reduced if a person wears a helmet, but some people don’t take it seriously they don’t wear helmet. So, we built our model which detects helmets with the help of cameras placed on crossings, road side, corners and in streets. All the things will be recorded on the government database. Using that database, the government can collect fines or challans from the people who were not wearing helmets while riding. By this people will be in fear of paying a fine so they will be wearing a helmet.

1.4 Significance

Our project mission is to implement our idea at schools, colleges, airports, hospitals, etc.

Where chances of spread of COVID-19 through contagion are relatively higher.

To reduce numbers of accidents on roads by making people wear helmets to avoid paying challans.

To stop students taking bags to the libraries as it is a major concern towards safety.

To detect motion in a storeroom, warehouse or a place where important things are kept where chance of robbery is more so it can be placed there to detect little motion if someone is present or not.

1.5 Research Design

The research design adopted for this Project is descriptive research design. The descriptive research design focuses on the accurate description of the variables present in the problem.

The proposed system is made using jupyter notebook. All the desired python libraries were convenient to use in this IDE. For this project the following modules and libraries were used i.e. tensorflow, numpy, Datetime, matplotlib, winsound,Object detection. We have created a live GUI for interacting with them as it gives a design and interesting look for object detection.

1.6 Sources Of Data

This project is created using python library and Google support.the Source of Data for the project was external. We used a dataset loaded from kaggle to feed the neurons and train the model. for input data this project will be using a camera for input in the form of images which will be converted to numerical input by python using numpy and tensorflow libraries.

1.7 Chapter Scheme

In this chapter ,we get to understand the basic introduction of "CAM-VISION".As we know Python is an emerging and effective language for machine learning so it is convenient to write a script for object detection in Python. The instructions for the program can be handled as per the requirement of the user. Object detection is about detecting the objects and labeling them from the image input. This is commonly used in various security programs and surveillance applications. In Python there is a library called numpy which allows us to convert images into desired numerical input. It was an interesting task to train the program/model for object detection. It became easier for the trained model to recognize objects as we trained it with more and more labeled data. During the training program the model was able to recognize objects like masks, helmets and people with or without them. As the current scenario, the model is able to detect masks,helmets and people while it can be useful for a wide range of application domains such as security and surveillance.

Chapter 2

2.1 Experimental setup

To build our project there are lot of things we have used in our pc(with inbuilt HD Camera) here they are:

Hardware Dependencies	Capacity
ROM	Up to 5 GB
RAM	At least DDR4 4 GB
CPU	Intel core i5 10th Gen
GPU	Nvidia MX 130 (2 GB)
Windows	Windows 10

Software Dependencies	Version
TensorFlow	2.7.0
OpenCV	4.5.4
Python	3.8
PYQT	5
Protoc	3.19.1
Visual C++ build tool	2015
HTML	5
CSS	3
JavaScript	ES2015
Django	3.2.9

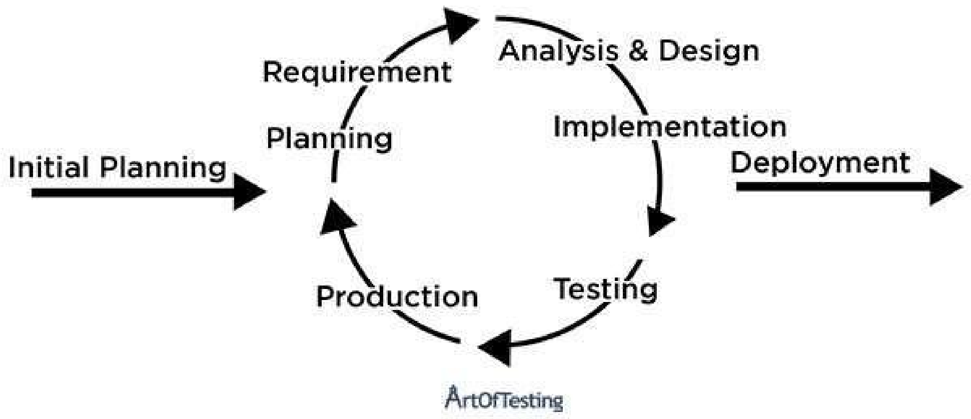
2.2 Procedures adopted

To complete our project, we have followed an iterative model.

Iterative model gives an exact performance of the development of software as a life cycle. It primarily focuses on preliminary growth and design and then gains momentum slowly with more complexity as well as meet requirements until the final software is built entirely. So, basically, the iterative development model is an approach of segmenting any large software development process into smaller portions.

There are major reasons to use this SDLC like:

- Parallel development can be planned.
- Progress can be measured.
- Less costly to change the scope/requirements.
- Testing and debugging during smaller iteration is easy.



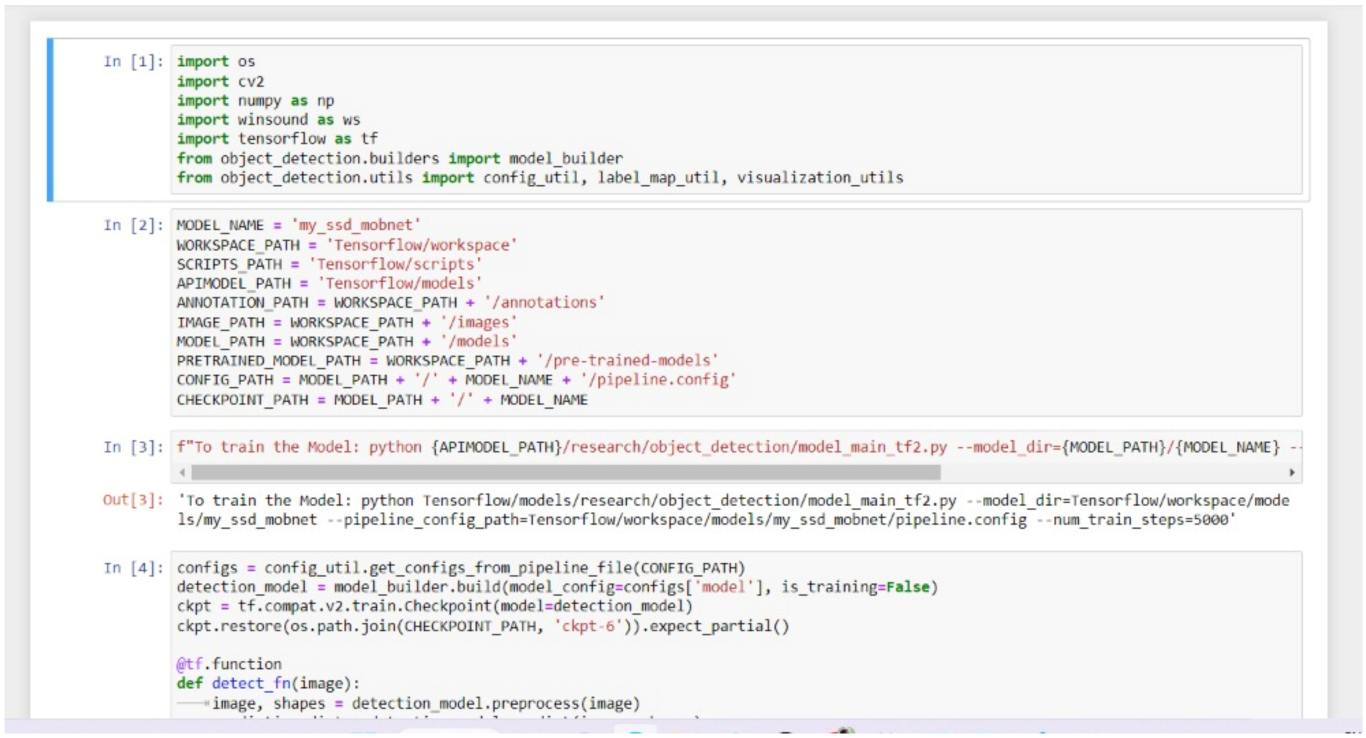
2.3 Technology Used

Type	Technology
Backend	Python
Frontend	HTML, CSS, JavaScript
Connectivity	Django, DTL, ORM

Chapter 3

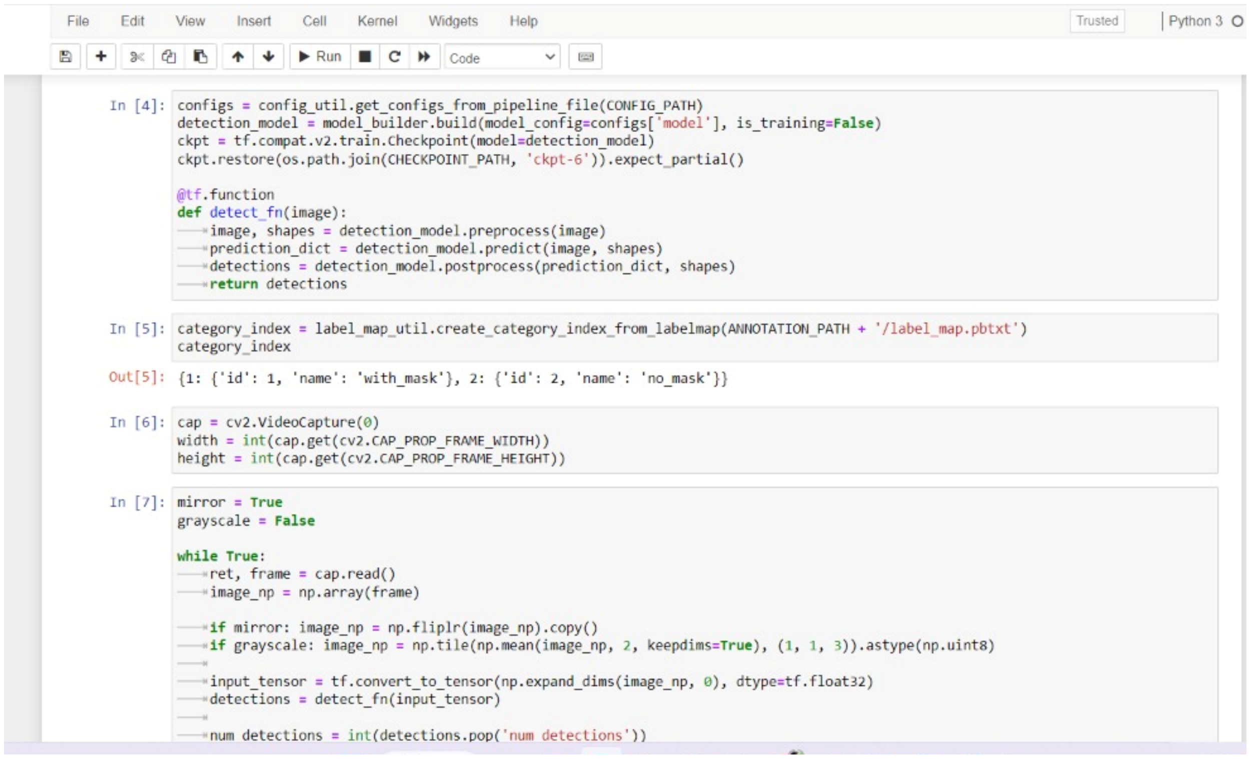
3.1 System Overview

CAM-VISION is an object detection program that can perform tasks of detecting objects on which it is trained.proposed the system is programmed using python libraries like tensorflow and numpy while helped the model in training with the labeled which was annotated using data of over thousand images and later used to train the data with these labeled images.
the code used in the program was written in jupyter notebook as shown in the images below:



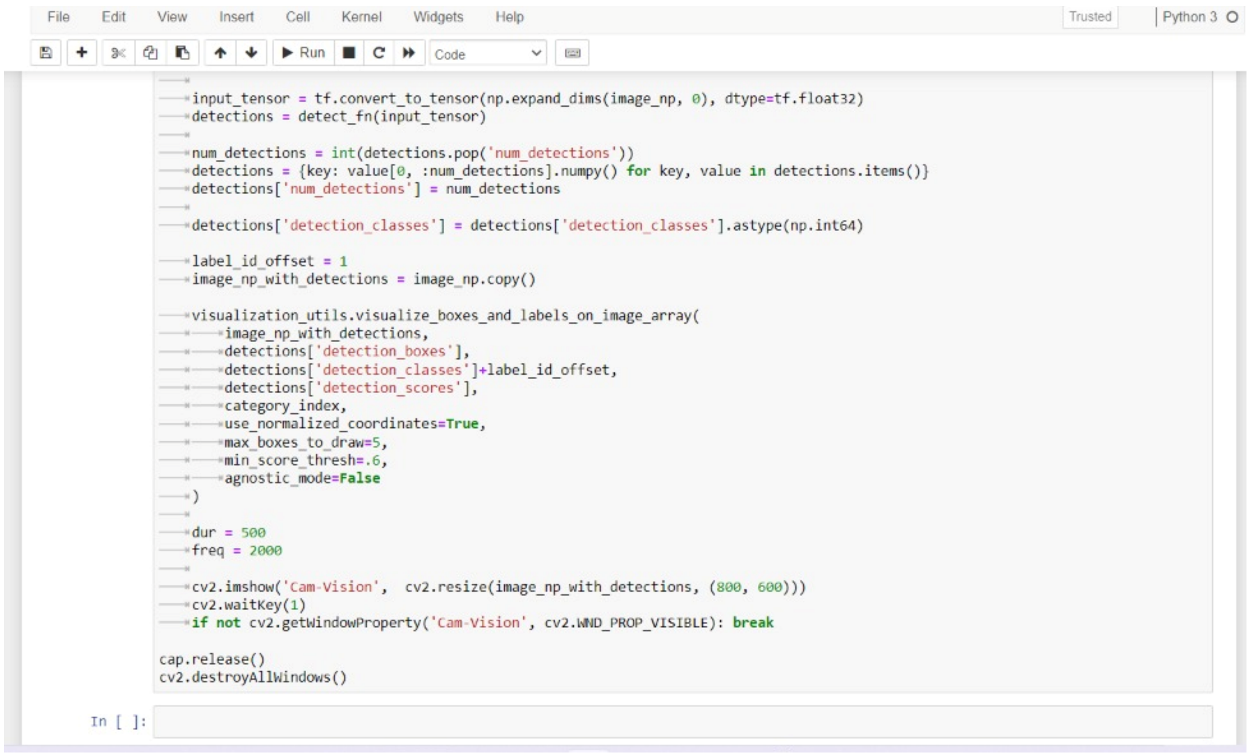
```
In [1]: import os
        import cv2
        import numpy as np
        import winsound as ws
        import tensorflow as tf
        from object_detection.builders import model_builder
        from object_detection.utils import config_util, label_map_util, visualization_utils

In [2]: MODEL_NAME = 'my_ssd_mobnet'
        WORKSPACE_PATH = 'Tensorflow/workspace'
        SCRIPTS_PATH = 'Tensorflow/scripts'
        APIMODEL_PATH = 'Tensorflow/models'
        ANNOTATION_PATH = WORKSPACE_PATH + '/annotations'
        IMAGE_PATH = WORKSPACE_PATH + '/images'
        MODEL_PATH = WORKSPACE_PATH + '/models'
        PRETRAINED_MODEL_PATH = WORKSPACE_PATH + '/pre-trained-models'
        CONFIG_PATH = MODEL_PATH + '/' + MODEL_NAME + '/pipeline.config'
        CHECKPOINT_PATH = MODEL_PATH + '/' + MODEL_NAME

In [3]: f"To train the Model: python {APIMODEL_PATH}/research/object_detection/model_main_tf2.py --model_dir={MODEL_PATH}/{MODEL_NAME} -

Out[3]: 'To train the Model: python Tensorflow/models/research/object_detection/model_main_tf2.py --model_dir=Tensorflow/workspace/mode
        ls/my_ssd_mobnet --pipeline_config_path=Tensorflow/workspace/models/my_ssd_mobnet/pipeline.config --num_train_steps=5000'

In [4]: configs = config_util.get_configs_from_pipeline_file(CONFIG_PATH)
        detection_model = model_builder.build(model_config=configs['model'], is_training=False)
        ckpt = tf.compat.v2.train.Checkpoint(model=detection_model)
        ckpt.restore(os.path.join(CHECKPOINT_PATH, 'ckpt-6')).expect_partial()

        @tf.function
        def detect_fn(image):
            image, shapes = detection_model.preprocess(image)
```

Image 3.1:loading libraries and loading the data



```
In [4]: configs = config_util.get_configs_from_pipeline_file(CONFIG_PATH)
        detection_model = model_builder.build(model_config=configs['model'], is_training=False)
        ckpt = tf.compat.v2.train.Checkpoint(model=detection_model)
        ckpt.restore(os.path.join(CHECKPOINT_PATH, 'ckpt-6')).expect_partial()

        @tf.function
        def detect_fn(image):
            image, shapes = detection_model.preprocess(image)
            prediction_dict = detection_model.predict(image, shapes)
            detections = detection_model.postprocess(prediction_dict, shapes)
            return detections

In [5]: category_index = label_map_util.create_category_index_from_labelmap(ANNOTATION_PATH + '/label_map.pbtxt')
        category_index

Out[5]: {1: {'id': 1, 'name': 'with_mask'}, 2: {'id': 2, 'name': 'no_mask'}}

In [6]: cap = cv2.VideoCapture(0)
        width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
        height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

In [7]: mirror = True
        grayscale = False

        while True:
            ret, frame = cap.read()
            image_np = np.array(frame)

            if mirror: image_np = np.fliplr(image_np).copy()
            if grayscale: image_np = np.tile(np.mean(image_np, 2, keepdims=True), (1, 1, 3)).astype(np.uint8)

            input_tensor = tf.convert_to_tensor(np.expand_dims(image_np, 0), dtype=tf.float32)
            detections = detect_fn(input_tensor)

            num_detections = int(detections.pop('num_detections'))
```

Figure 3.2: using opencv for image processing and setting the window



```
            input_tensor = tf.convert_to_tensor(np.expand_dims(image_np, 0), dtype=tf.float32)
            detections = detect_fn(input_tensor)

            num_detections = int(detections.pop('num_detections'))
            detections = {key: value[0, :num_detections].numpy() for key, value in detections.items()}
            detections['num_detections'] = num_detections

            detections['detection_classes'] = detections['detection_classes'].astype(np.int64)

            label_id_offset = 1
            image_np_with_detections = image_np.copy()

            visualization_utils.visualize_boxes_and_labels_on_image_array(
                image_np_with_detections,
                detections['detection_boxes'],
                detections['detection_classes']+label_id_offset,
                detections['detection_scores'],
                category_index,
                use_normalized_coordinates=True,
                max_boxes_to_draw=5,
                min_score_thresh=.6,
                agnostic_mode=False
            )

            dur = 500
            freq = 2000

            cv2.imshow('Cam-Vision',  cv2.resize(image_np_with_detections, (800, 600)))
            cv2.waitKey(1)
            if not cv2.getWindowProperty('Cam-Vision', cv2.WND_PROP_VISIBLE): break

        cap.release()
        cv2.destroyAllWindows()

In [ ]:
```

Figure 3.3: Evaluating the object detection and closing the window.

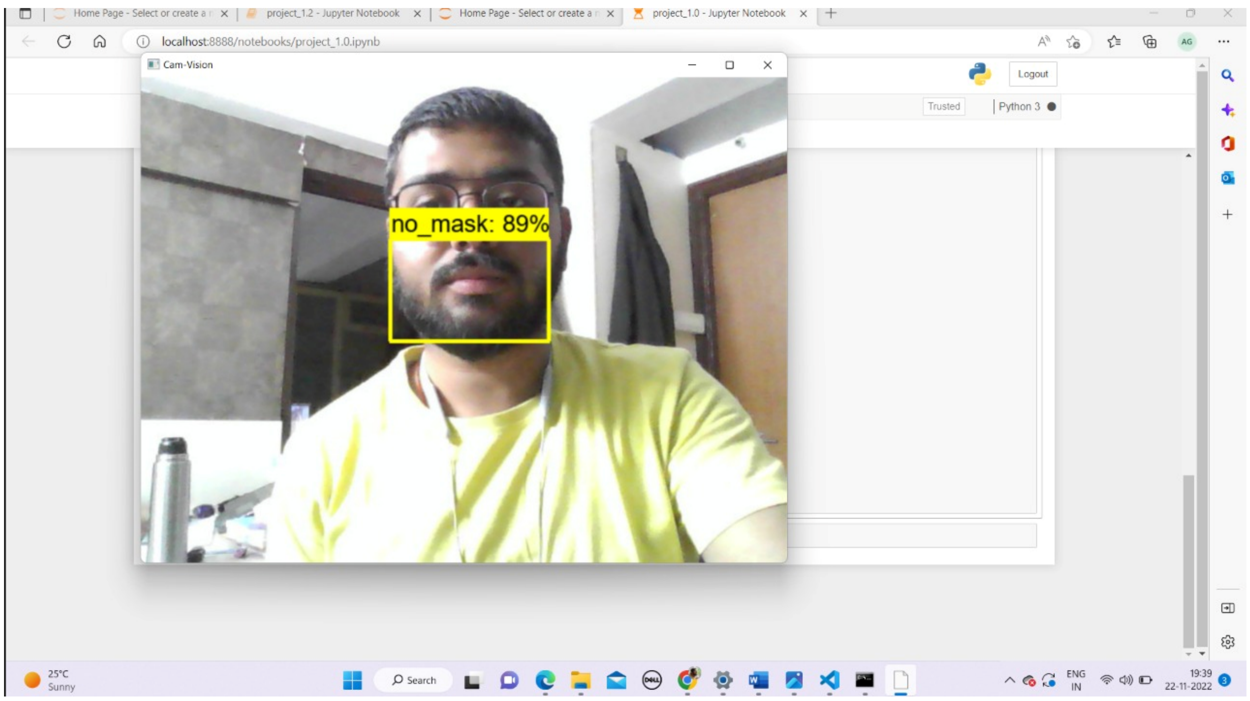


Figure 3.4 : running program window detecting objects.

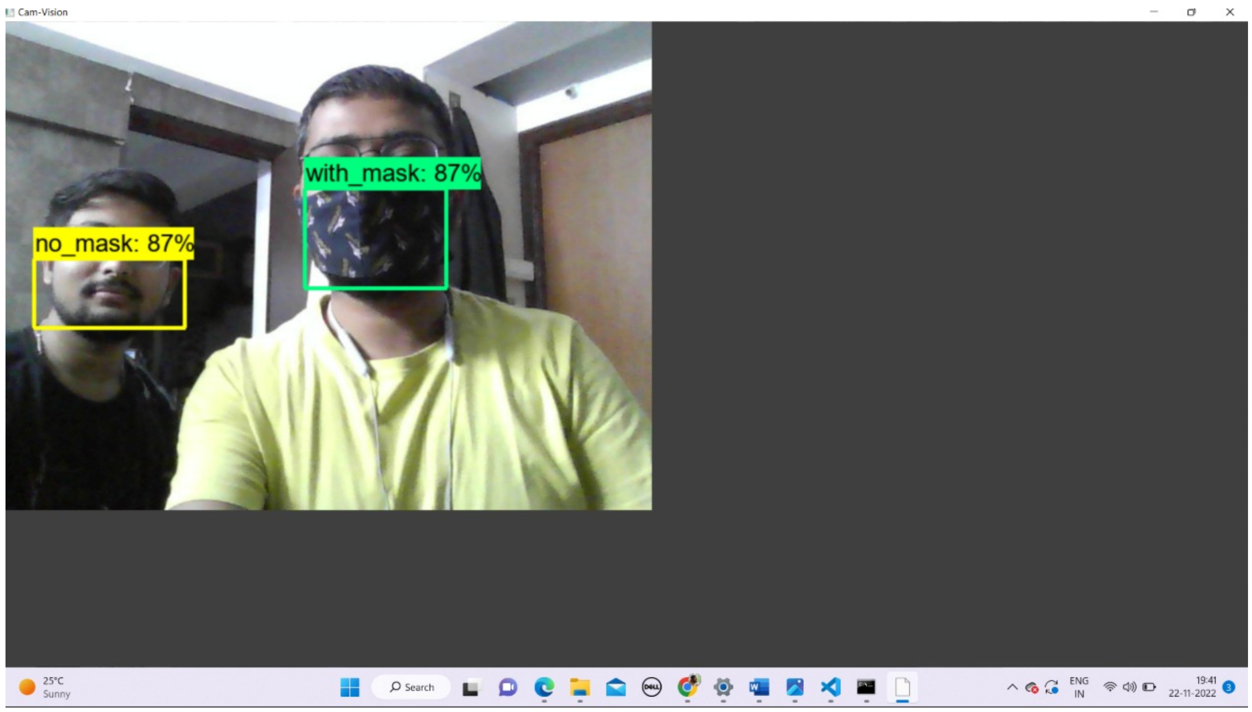


Figure 3.5: working out put window with accurate detection.

3.2 Functional requirements

- 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 streams.
- The system must be able to extract each face region of interest.
- The system must be able to detect masks at every frame of video.

3.3 Non-Functional requirements

- 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.
- The background must not be too bright or too dark while detecting the face mask.
- The system must be able to correctly detect more than one face if present, and hence the presence of a mask in the frame.

3.4 Security

For secure access to the site for every personal machine we have added login support with multiple users accessibility within we make sure that he/she was login properly and can't access any modules by manipulating the search URL path or using any SQL Injections.

3.5 User interface

Fig 3.6 Home Page

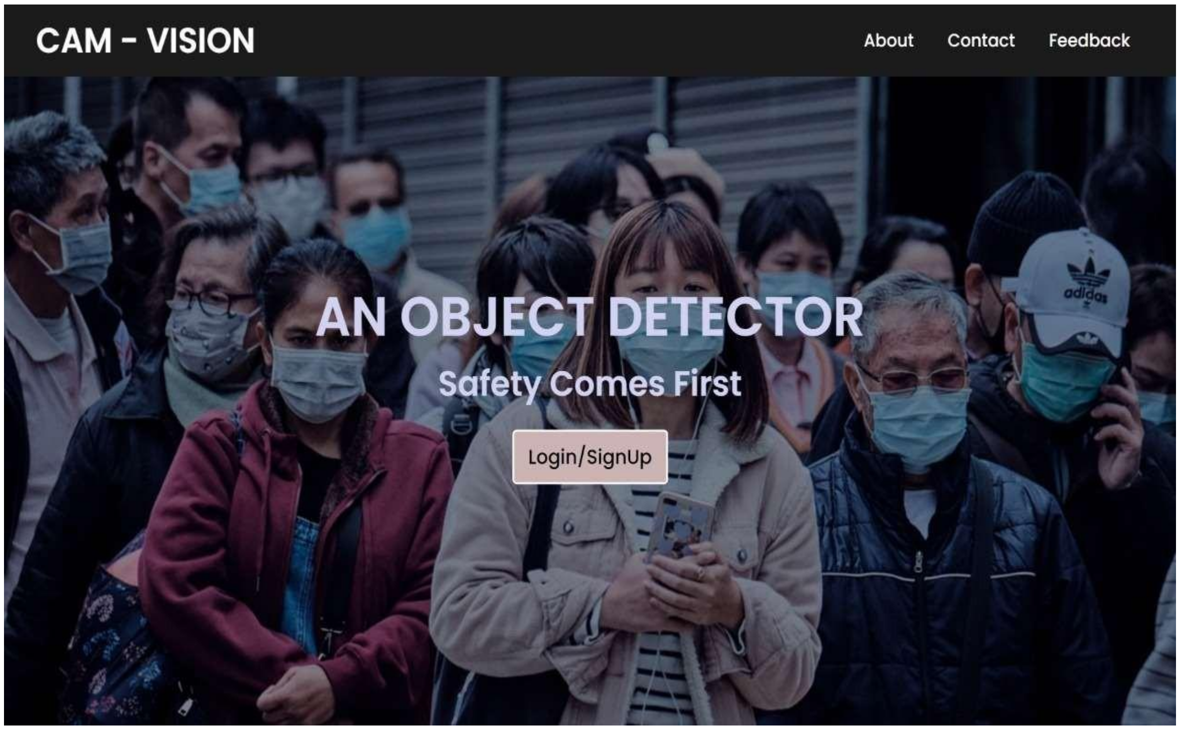


Fig 3.6: Sign-in Page

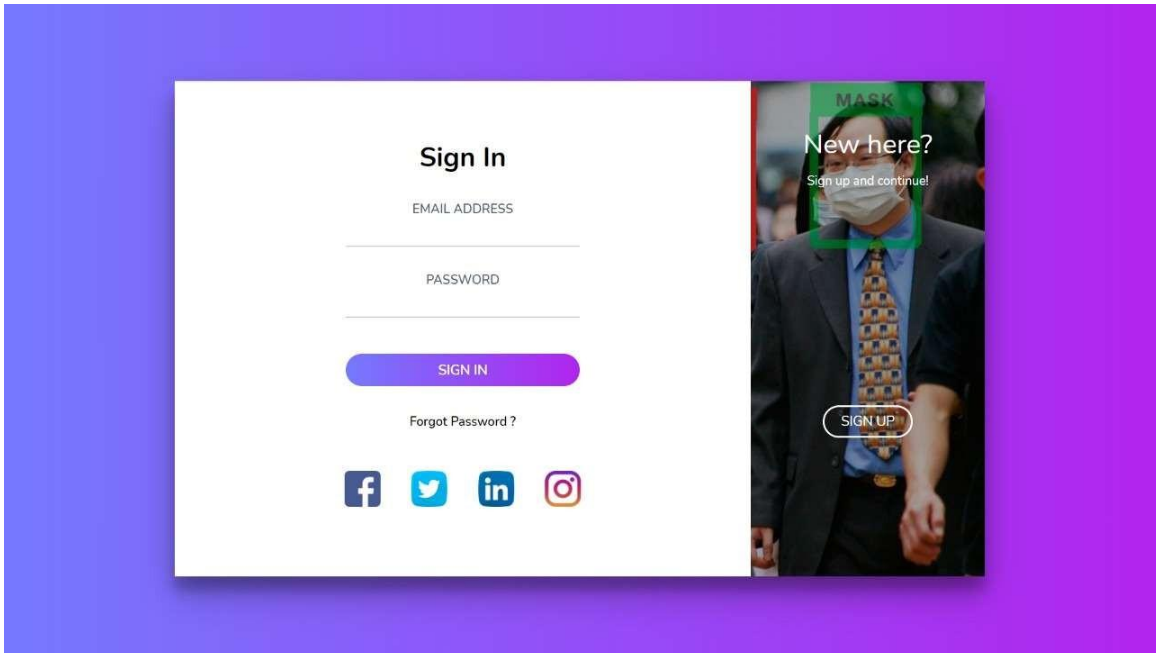


Fig 3.7: Sign-up Page

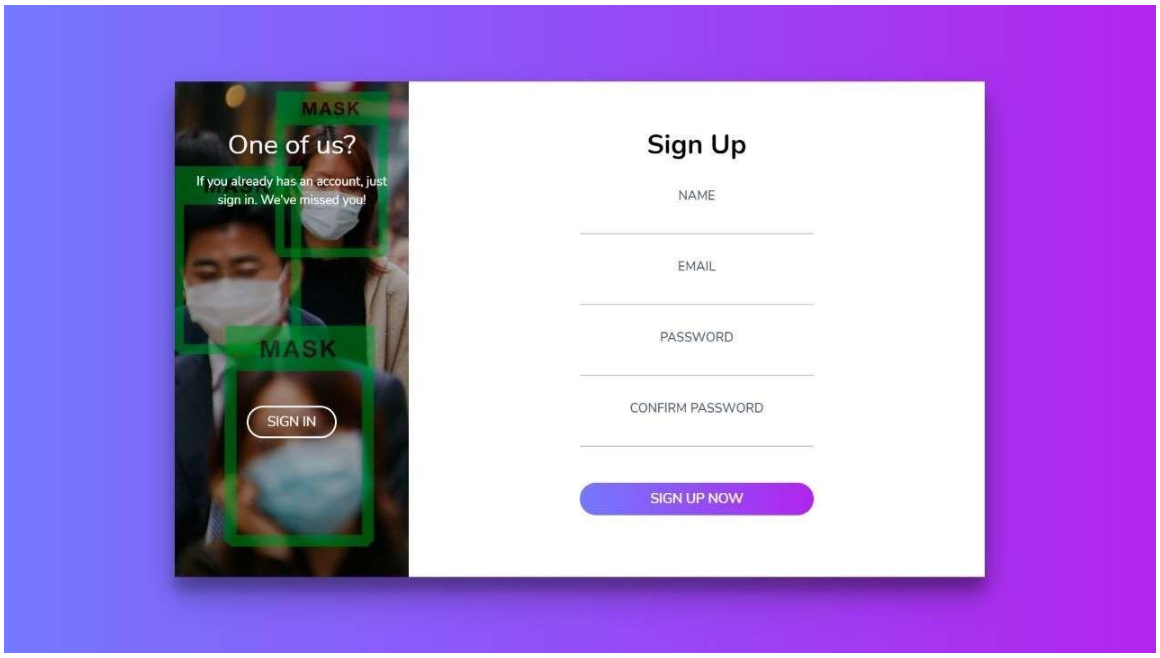
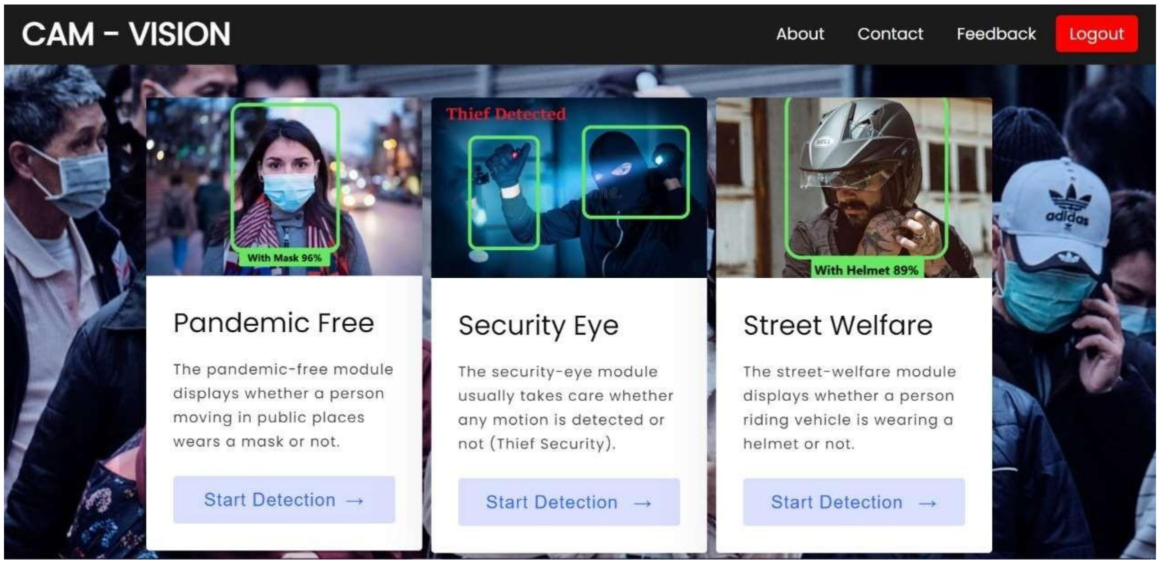


Fig 3.8: Object Selection Page



Pandemic Free

The pandemic-free module displays whether a person moving in public places wears a mask or not.

Security Eye

The security-eye module usually takes care whether any motion is detected or not (Thief Security).

Street Welfare

The street-welfare module displays whether a person riding vehicle is wearing a helmet or not.

Fig 3.9: Pandemic Free (Mask Detector)

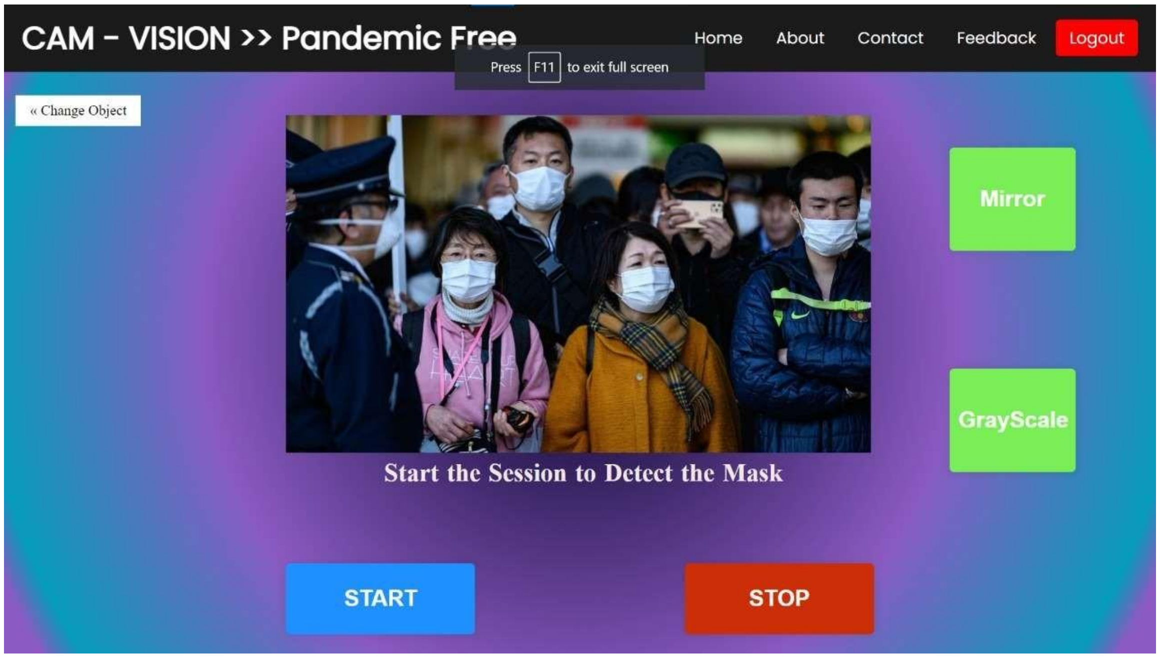

CAM - VISION >> Pandemic Free
Home
About
Contact
Feedback
Logout
Press F11 to exit full screen
« Change Object
Mirror
GrayScale
Start the Session to Detect the Mask
START
STOP

Fig 3.10: Security Eye (Motion Detector)

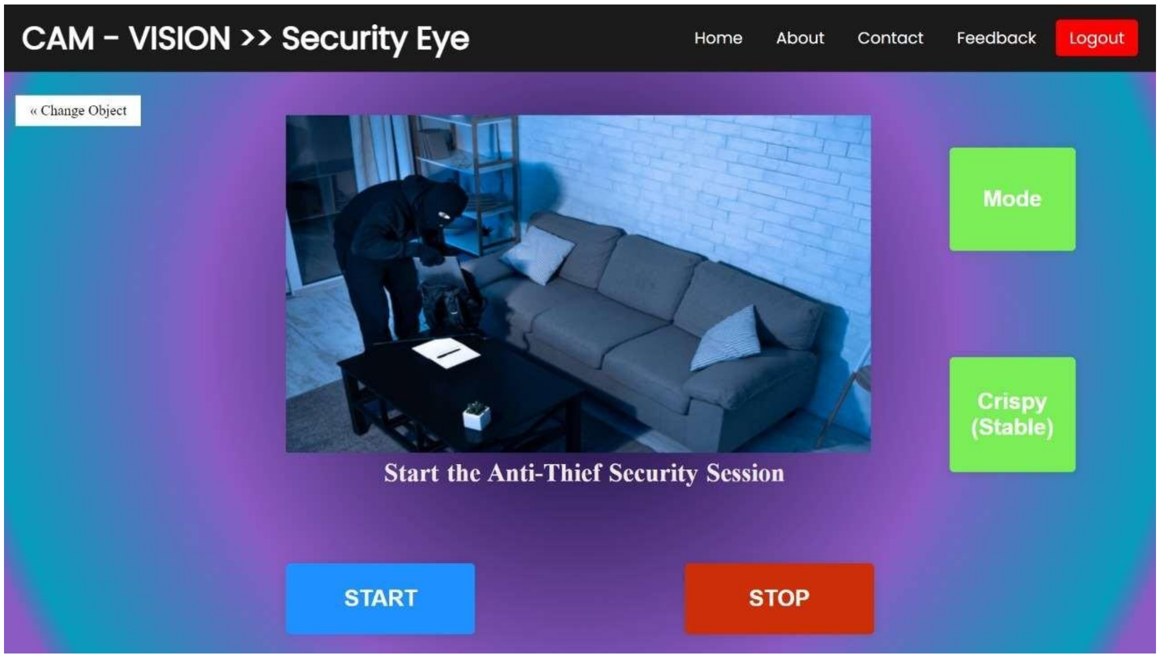


Fig 3.11: Street Welfare (Helmet Detector)

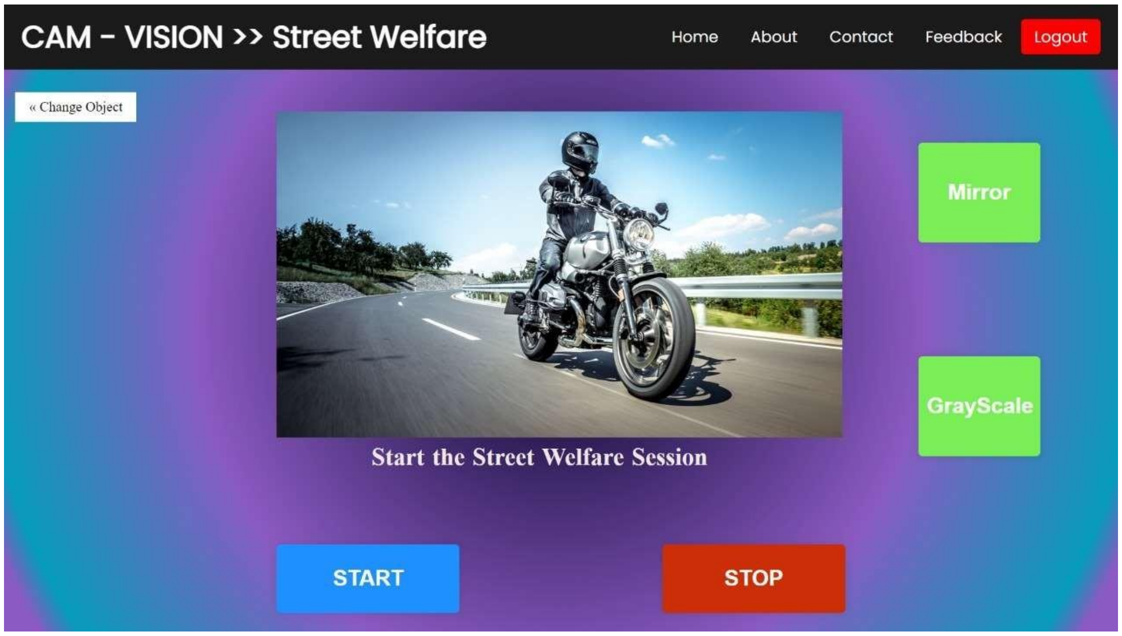
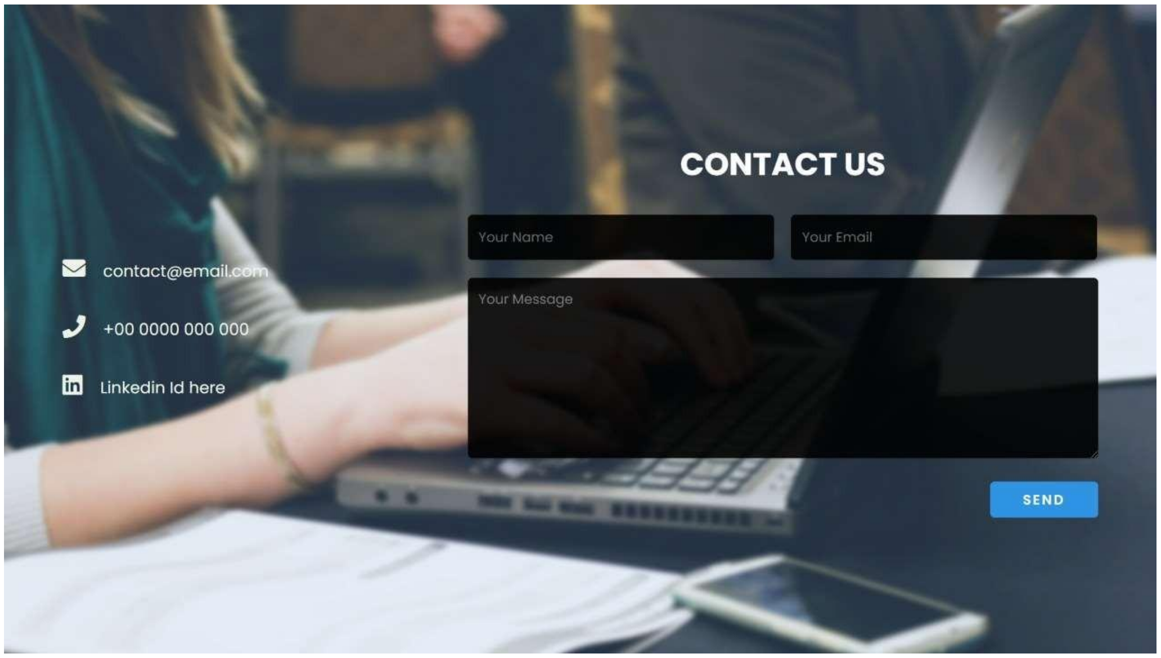


Fig 3.12 : Contact us



Chapter 4

4.1 Conclusion

This was all about our project. Our project has vast industrial as well as personal use. Our project mission is to implement our idea at schools, colleges, airports, hospitals, etc. etc. where chances of spread of COVID-19 through contagion are relatively higher. To reduce numbers of accidents on roads by making people wear helmets to avoid paying challans. To stop students taking bags to the libraries as it is a major concern towards safety.to prevent robbery at storehouse warehouse etc by detecting little motion. At last, by binding all the things together our main motive is for surveillance and for security purposes. Suppose in the future any pandemic occurs like corona then we will be prepared for all things.

4.2 Future Scope

To create a better future, we should take steps in the present. This project will help the future in surveillance, safety and security. Since these things are never going to end. By that time we will also try to create some function which can be more beneficial for the user. Technology is growing with great speed. I hope in future we may get some good technology or idea to create some function.

4.3 Bibliography

1. https://tensorflow-object-detection-apitutorial.readthedocs.io/en/latest/index.html
2. https://getbootstrap.com/
3. https://docs.djangoproject.com/en/3