

Implementation of Social Distancing Detection and Alerting System Using Artificial Intelligence

Anuradha M sandi¹, ShivaniShetkar², Anand², Sneha²

¹ Professor, Department of Electronics & communication Engineering, GNDECBidar, University of VTU, India

² Student, Department of Electronics & communication Engineering, GNDECBidar, University of VTU, India

² Student, Department of Electronics & communication Engineering, GNDECBidar, University of VTU, India

² Student, Department of Electronics & communication Engineering, GNDECBidar, University of VTU, India

ABSTRACT

The COVID-19 pandemic forced governments across the world to impose lockdowns to prevent virus transmissions. This resulted in the shutdown of all economic activity and accordingly the production at manufacturing plants across most sectors was halted. While there is an urgency to resume production, there is an even greater need to ensure the safety of the workforce. Reports indicate that maintaining social distancing at work clearly reduces the risk of transmission. We decided to use computer vision to monitor activity and detect violations which trigger real time alerts and mail with violation photos attached to concerned authority regarding rules violation as evidence.

The adoption of public health-informed hygienic practices can have a large impact on community transmission of COVID-19. This paper proposes a practical deep learning computer vision framework methodology for social distancing detection for detection and tracking of people in public spaces. The detection tool was developed to alert people to maintain a safe distance with each other by evaluating a video feed. The video frame from the camera was used as input, and the open-source object detection pre-trained model based on the YOLOv3 algorithm was employed for pedestrian detection.

KEYWORDS: Covid-19, Social distancing, Opencv, Python, Computer vision, YOLOv3, Haar Cascade Object Detector

Date of Submission: 21-08-2021

Date of acceptance: 05-09-2021

I. INTRODUCTION

Over time and space, technology is widely spreading across the world. It is used to facilitate our living skills in our daily life. Technology has been around surveillance systems for decades. In recent years, video-surveillance systems have become a main interest in people's life such as government agencies, business, and private possessions. Nowadays, people seek for better image quality, less in cost, better in size and scalability. For safety issues, cameras can monitor real-time occurrences, collect data, and come out with analyzing the behavior of people. Monitoring is often performed through consecutive frames which are extracted from the video. To take advantage of video surveillance, it is an essential thing to propose an algorithm that is simple and fast to detect human activities. It's an approach that combines the needs of the market and the simplicity. The major idea of this is to suggest a sufficient algorithm that deals with analyzing the content of a video to classify events between normal and abnormal ones, using a simple software algorithm. According to data obtained by the World Health Organization, the global pandemic of COVID-19 has severely impacted the world and has now infected more than eight million people worldwide. Following safe social distancing is one of the enhanced safety protocols need to be followed in public places in order to prevent the spread of the virus. To create safe environment that contributes to public safety in colleges and Schools, we propose an efficient computer vision based approach focused on the real-time automated monitoring of people to detect social distancing in public places by implementing the model to monitor activity and detect violations through camera. We have chosen our university a place to test our results in. We aim to solve this issue in order to be aware of anything that could happen to provide the safety due to Covid-19 pandemic, so we need to avoid social gatherings.

II. BLOCK DIAGRAM

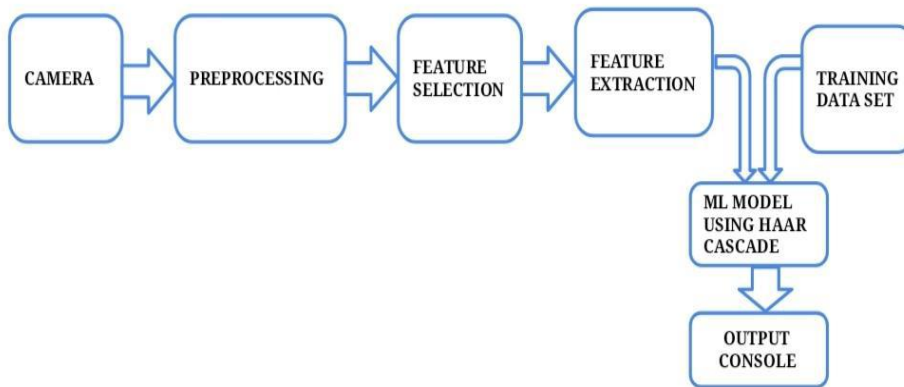


Figure- 1:Block diagram

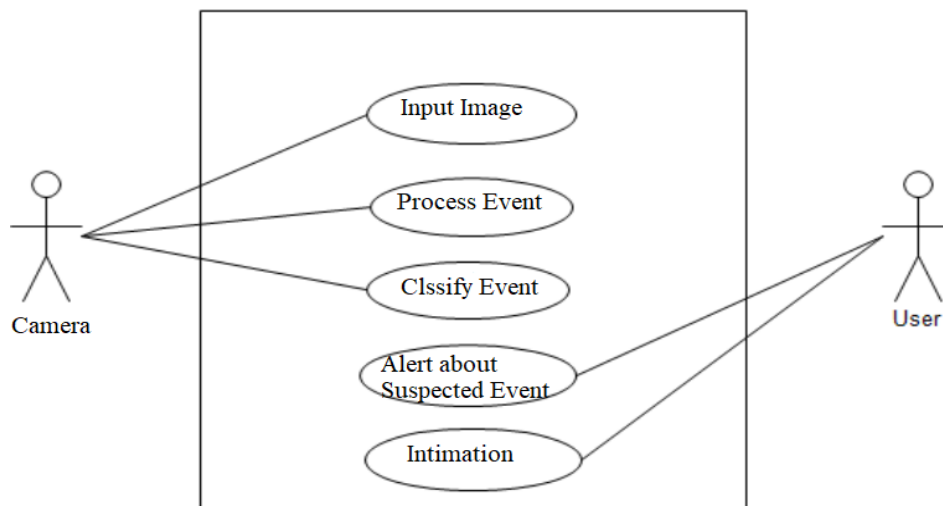


Figure- 2:User case diagram

III. TECHNICAL SPECIFICATIONS

HARDWARE

- System : Intel i3/i5 2.4 GHz.
- Hard Disk : 500 GB
- Ram : 4/8 GB

SOFTWARE

- Operating system : WindowsXP/Windows 7/8/10.
- Packages Used : OpenCV Python
- Coding Language : Python
- IDE : Python

DESCRIPTION ABOUT MODULES

1. Python IDE

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. The Python interpreter is easily extended with new functions and

data types implemented in C or C++ (or other languages callable from C). Python is also suitable as an extension language for customizable applications.

2. OpenCV

OpenCV is a library of programming functions mainly aimed at real-time computer vision. It has a modular structure, which means that the package includes several shared or static libraries. We are using an image processing module that includes linear and non-linear image filtering, geometric image transformations (resize, affine and perspective warping, and generic table-based remapping), color space conversion, histograms, and so on. Our project includes libraries such as Viola-Jones or Haar classifier, LBPH (Lower Binary Pattern histogram) face recognizer, Histogram of oriented gradients (HOG).

OpenCV(open source computer vision library)

Open source PC vision library and AI programming library .OpenCV is a cross-stage library utilizing which can grow constant PC vision applications. It chiefly centers around picture handling, video catch and examination including highlights like face identification and item location.

It can be used as:

```
import OpenCV as  
>>>pip install cv2
```

OpenCV-Python

Python is a general purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability.

Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in the background) and second, it is very easy to code in Python. This is how OpenCV-Python works, it is a Python wrapper around original C++ implementation. And the support of Numpy makes the task more easier. NumPy is a highly optimized library for numerical operations. It gives a MATLAB-style syntax. All the OpenCV array structures are converted to-and-from NumPy arrays. So whatever operations you can do in NumPy, you can combine it with OpenCV, which increases number of weapons in your arsenal. Besides that, several other libraries like SciPy, Matplotlib which supports NumPy can be used with this. So OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems.

3. Keras

Keras is an incredible simple to-utilize python library for creating and assessing profound learning models. It wraps the effective numerical calculation libraries theano and tensorflow and permits you to characterize and prepare neural system models in a couple of short lines of code.

The way it is used:

```
import Kereas as  
>>>pip install keras
```

4. Tensorflow

Tensorflow is the essential programming device in profound learning. It is an open source man-made consciousness library, utilized for information stream diagrams to manufacture models. Tensorflow is principally utilized for: discernment, arrangement, finding, getting, creation and forecast

The way it is used:

```
import Tensorflow as  
>>>pip install tensorflow
```

5. Imutils

Imutils are a progression of comfort capacities to make essential picture preparing capacities, for example, interpretation, turn, resizing, skeletonization, and showing

Matplotlib pictures simpler with OpenCV and both Python 2.7 and Python 3. The way it is used:

```
import Imutils as  
>>>pip install imutils
```

6. Pillow

Pillow is a Python Imaging Library (PIL), which includes support for opening or controlling or sparing pictures. The present adaptation Pillow recognizes and peruses an enormous number of configurations. Compose suppon are deliberately limited to the most regularly utilized trade or introduction designs.

It can be used as follows:

```
import pillow as  
>>>pip install pillow
```

7. Image processing module

The purpose of image processing is divided into 5 groups. They are:

1. Visualization- Observe the objects that are not visible.
2. Image sharpening and restoration- To create a better image.
3. Image retrieval- Seek for the image of interest.
4. Measurement of pattern– Measures various objects in an image.
5. Image Recognition– Distinguish the objects in an image.

IV.RESULT AND DISCUSSION

Using the Anaconda platform we have created a virtual system environment with different libraries and version. Basically with anaconda we can build and deploy deep learning models that use neural networks. We have built and trained a neural network model. Initially we can see in Fig 3 the anaconda navigator which includes applications , packages and environment. Anaconda is popular because it brings many of the tools used in data science and machine learning with just installation. It also uses the concept of creating environments so as to isolate different libraries and versions.

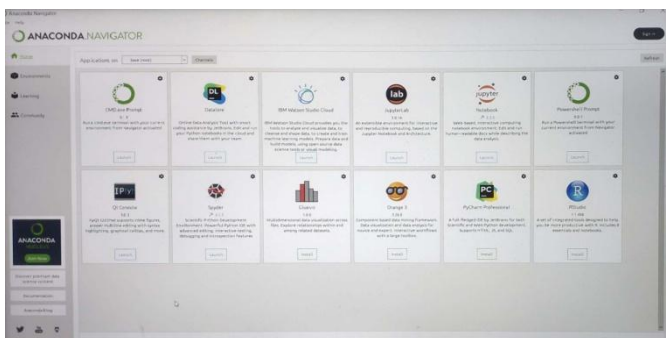


Fig -3: Anaconda navigator.

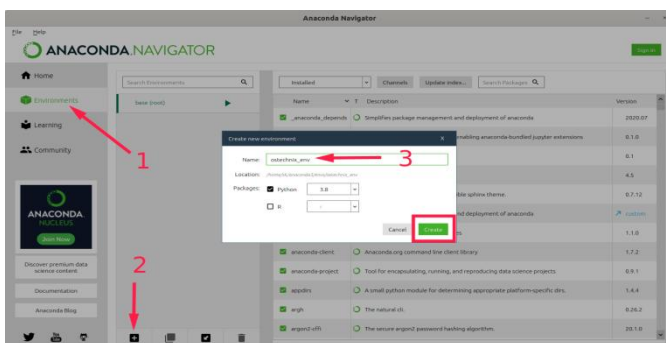


Fig -4: Creating an environment by choosing packages.

figure 4, represents creating our own environment using packages first from the navigator window moving to the environment section then by creating an environment with a name suitable to the project and there by choosing which packages we want to use. Upon creation of the environment in that terminal we need to install all the additional libraries such as python, opencv, tensorflow, imutils, pillow, keras.

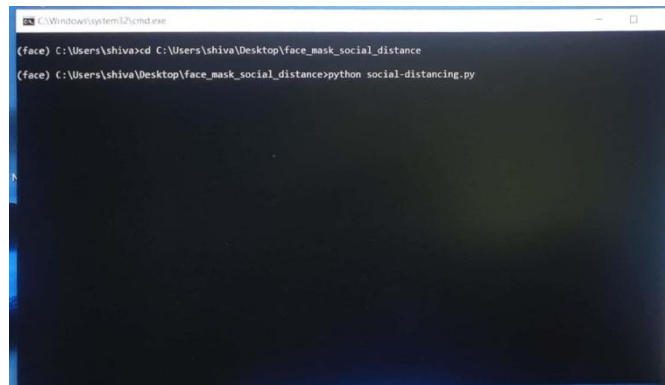


Fig -5: Created environment terminal

Fig- 5, represents the terminal upon opening the terminal. Firstly we need to change the path and provide the path for the execution of the code file. After changing the path we need to run the file by providing filename along with py extension.

Fig- 6, Shows the final result of the social distancing model. It is splitted into four modules

- Capture the image
- Identify the person from the image
- Social Distance
- SMS module

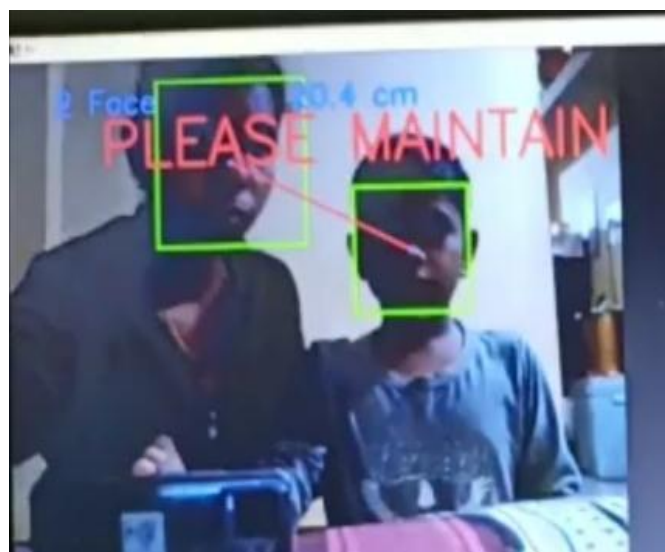


Fig -6:Detection of Social distancing

Capture the image

The first step is to initialize the web camera and start capturing the video frame in that we are using opencv(cv2) and then it takes the video and face of the person image. It will capture the image.

Identify the person from the image

The goal of face detection is to determine if there are any faces in the image or video. If multiple faces are present, each face is enclosed by a bounding box and thus we know the location of the faces. The videos are basically made up of frames, which are still images. We perform face detection for each frame in a video. So when it comes to detecting a face in still image and detecting a face in a real-time video stream, there is not much difference between them.

We will be using the Haar Cascade algorithm, also known as Viola-Jones algorithm to detect faces. It is basically a machine learning object detection algorithm which is used to identify objects in an image or video. In OpenCV, we have several trained Haar Cascade models which are saved as XML files. Instead of creating and training the model from scratch, we use this file. We are going to use "haarcascade_frontalface_alt2.xml" file in this project.

Social distance

In this module If the user could mark two points on the frame that are 6 feet apart, using extrapolation, one could find the distance between different points on the frame. This would have been true if the camera was equidistant to all the points on the plane where the pedestrians were walking. The closer the pedestrians are to the camera the bigger they are. The closer the two points (which are the same number of pixels apart) on the frame to the camera, the smaller is the actual distance between them.if both persons closer the two points it will display please maintain social distance. An alert message will be sent to the user.

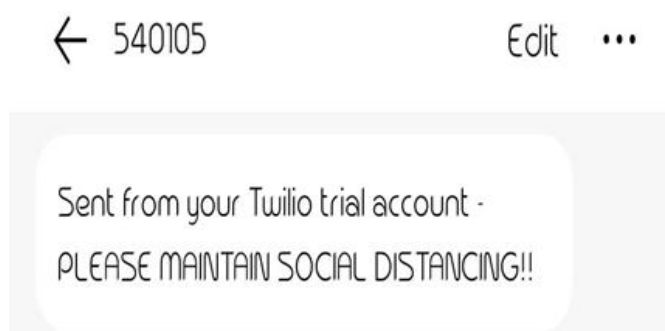


Fig-7: Alert message sent from twilio message module.

SMS module

From fig-7 In this module we will be sending an sms to the concerned authority with the image showing the level of accuracy. In the webcam we will be taking the frame and storing it in location and will be sharing it to the respective concerned authority. Here we configured our sms as the domain to send msg here we are using the twilio module.

V.CONCLUSION

This model is more reliable in providing security. This is also a much anticipated solution to the problem of illegal or rules violation in any places and also it can be used to monitor social distance. It deals with Interfacing of camera to capture live face image, monitoring the social distancing further Detecting unusual Activity and sending alerts to the consult person that if social distance is not maintained then sending an alert about the violation to the authorized person.The outcomes were expectations based. Still, we have a vision of building a hardware model along with unusual event detection.

REFERENCES

- [1]. <https://blog.usejournal.com/social-distancing-ai-using-python-deep-learning-c26b20c9aa4c>.
- [2]. Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object detection with deep learning: A review," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 11, pp. 3212–3232, 2019.
- [3]. S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.
- [4]. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [5]. X. Chen and A. Gupta, "An implementation of faster rcnn with study for region sampling," *arXiv preprint arXiv:1702.02138*, 2017.
- [6]. N. S. Punn and S. Agarwal, "Crowd analysis for congestion control early warning system on foot over bridge," in *2019 Twelfth International Conference on Contemporary Computing (IC3)*.IEEE, 2019, pp. 1–6.
- [7]. A. Brunetti, D. Buongiorno, G. F. Trotta, and V. Bevilacqua, "Computer vision and deep learning techniques for pedestrian detection and tracking: A Survey," *Neurocomputing*, vol. 300, pp. 17–33, 2018.
- [8]. Pias, "ObjectDetectionanddistancemeasurement," <https://github.com/paul-pias/Object-Detection-and-Distance-Measurement>, 2020.
- [9]. Kartik J. Srimadhavan V. 2013 SMS Alert and Embedded Network Video Monitoring Terminal.