Realtime Social Distance Detection and Human Count using Deep Learning

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering

by

Mr. Salman Ansari (Roll no. 49)

Mr. Parth Desai (Roll no. 53)

Mr. Hitesh Gosavi (Roll no. 12)

Under the guidance of

Asst. Prof. Sunil Katkar

DEPARTMENT OF COMPUTER ENGINEERING

VIDYAVARDHINI'S COLLEGE OF ENGINEERING AND TECHNOLOGY

K. T. MARG, VASAI ROAD (W.) DIST-THANE, PIN: 401202

(Affiliated to University of Mumbai)

2021-2022

A project report on

Realtime Social Distance Detection and Human Count using Deep Learning

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering

by

Mr. Salman Ansari (Roll no. 49)

Mr. Parth Desai (Roll no. 53)

Mr. Hitesh Gosavi (Roll no. 12)

Under the guidance of

Asst. Prof. Sunil Katkar

DEPARTMENT OF COMPUTER ENGINEERING

VIDYAVARDHINI'S COLLEGE OF ENGINEERING AND TECHNOLOGY

K. T. MARG, VASAI ROAD (W.) DIST-THANE, PIN: 401202

(Affiliated to University of Mumbai)

2021-2022

CERTIFICATE

This is to certify that the project entitled **"Realtime Social distance detection and human count using Deep Learning"** is a bonafide work of **"Salman S Ansari (Roll No. 49), Parth J Desai (Roll No. 53) and Hitesh K Gosavi (Roll No. 12)"** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of "**Bachelor of Engineering**" in "**Computer Engineering**".

_____

Asst. Prof. Sunil Katkar
(Guide)

_____ _____

Dr. Megha Trivedi
(Head of Department)

Dr. Harish Vankudre
(Principal)

Project Report Approval for B.E.

This project report entitled **'Realtime Social distance detection and human count using Deep Learning'** by **'Salman S Ansari, Parth J Desai and Hitesh K Gosavi'** is approved for the degree of '**Bachelor of Engineering**' in '**Computer Engineering**'.

Examiners

1. _____

2. _____

Date:

Place:

Declaration

We declare that this written submission represents our ideas in our own words and where other's ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Salman S Ansari (49)

Parth J Desai (53)

Hitesh K Gosavi (12)

Date:

Acknowledgements

It is said that “learning is a never-ending process.” While working on the project we have undergone the same experience of learning new things as we proceeded in our goal of building a Glove based sign language translator which could cater to the need of the physically challenged people.

Working on the project was a new experience for us. As it opened a new gateway wherein, we had as opportunity to work on a totally new concept as far as the engineering syllabus is concerned where most of the concepts are to be learned by rote.

The joy of working in a new domain and learning new things was welcome experienced for the four of us and all we have to say is that we have cherished all the moments as they came by, right from working on project to the making this report.

We would like to thank our Principal **Dr. Harish Vankudre** for constant motivation and support to excel and having faith in our ability. We would also like to thank our professor **Dr. Megha Trivedi** (Head of Department of Computer Engineering) for providing her views of the subject.

We would like to thank **Asst. Prof. Sunil Katkar,** who guided us and shared their knowledge & invaluable experience about the topic and gave their precious time towards solving our difficulties. We would also like to thank our college management for providing us with the facilities and infrastructure for working on the project.

Salman S Ansari (49)

Parth J Desai (53)

Hitesh K Gosavi (12)

Date:

Abstract

A method for social distancing detection using deep learning to estimate the distance between people to lessen the impact of coronavirus pandemic. The practice of social distancing introduced in China and recently used around the world to control COVID19. Many nations have come up with solutions to use technology to help overcome pandemic loss. The model is trained on COCO dataset. For the detection of people from top view, transfer learning is used to increase efficiency of the model.

YOLOv3 has been used in our model due to its predictive accuracy. YOLOv3 has a tight network structure, comes in handy for multiscale detections. The proposed method can accurately measure social distance between individuals in the video. We use bounding box information to calculate the centroid distance. We use the Euclidean distance formula and compute the distance between every detected box. After computing the distance, a fixed threshold value is used to determine whether the distance between two centroids is less than the fixed threshold value or not. If two individuals are in proximity and distance between them is less than the threshold value, then they are put in violate set and color is updated to red. A centroid tracking algorithm is adopted to help track those individuals who violate / cross the threshold of social distancing. We observe that total social distance violations and total human count are displayed on the output screen.

Table of Content

List of Figures

Chapter 1

Introduction

1.1 Problem Definition

COVID-19 originated from China and has spread to almost every part of the world since December 2019. WHO declared it as a pandemic when the virus was detected in more than 11 countries. On October 7, 2020, more than 35,00,000 confirmed cases of COVID-19 were reported, including more than 1,00,000 deaths. [1]

The number of people infected from coronavirus is shown in the figure 1.1. Many health organizations have been continuously searching for vaccines and drugs to deal with the virus, although no achievement have been reported so far. The world community have been searching for ways to stop the spreading of the coronavirus. Many have even switched to technological based solutions like CNN. Deep Convolution Neural Networks (CNNs) have shown considerable performance in terms of object detection. It is nothing but localization and classification. [2]

The coronavirus is known to spread when an infected person sneezes. When people breathe in the contaminated air, they are also infected with the virus. The virus reaches the lungs and starts killing cells inside the lungs. Studies have showed that people who are infected and have no symptoms are mainly involved in the spread of the virus. Therefore, it is necessary to maintain a minimum distance of 1.5-2 meters from others even if they show no signs of symptoms.

All these circumstances favor the need to develop social distance detectors.

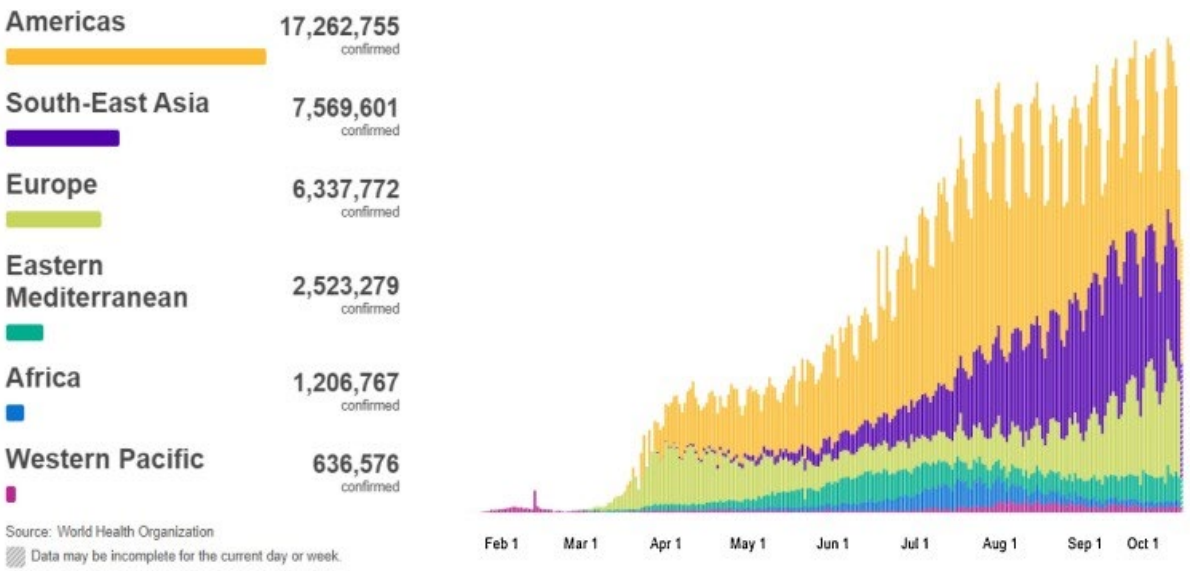

Fig 1.1 Covid Cases Worldwide as of October 2020

1.2 Aim and Objective

To develop a tracking framework for social distance tracking using top view. To implement YOLOv3 algorithm for human detection and creating a bounding box for the same. In addition, non-maxima suppression algorithm to improve model performance.

To map the distance between each pair of bounding boxes, Euclidean distance formula is used. This is done after centroid calculation for each box. A fixed threshold value for social distance is specified in pixel value for an approximate distance (i.e., for 2 meters). Using an algorithm to track the pair which violates the social distancing norms. Also, count the number of bounding boxes to show human count.

1.3 Motivation

Computer vision and machine learning have shown promising results in various applications developed for everyday life. Object detection have seen significant improvements over the past few decades. Distance based methods are employed to calculate the distance between people.

Most of the methods that have been developed so far are based on front or side view recording. This requires the camera to be fixed correctly to map pixels to distance to measure (e.g., feet, meters, etc.). Such technological development will be useful in addressing the current covid situation.

COVID-19 pandemic has shown ill effects on both social and economic life. The most difficult task is to restore back the normal life keeping the risk of infection to minimum. To mitigate the spread of virus and reduce the risk of getting infected, social distancing has shown to be effective in community. Multi-class object detection in deep learning research has shown phenomenal output on complex datasets. Nguyen et al. provided a complete study on current developments and human detection trials. [3]

Fig 1.2 Social Distancing

Chapter 2

Literature Review

2.1 Existing System

Following an increase of the COVID-19 pandemic since the end of 2019, social distancing has proved an extremely effective method to prevent the spreading of coronavirus and was adopted as a standard practice in early 2020. In the beginning, the number of new cases increased exponentially, with six to eight thousand new cases every day in the month of February.

This was mainly due to the practice of social distancing recently introduced and used around the world to control COVID-19. CNN have shown to get almost perfect result and phenomenal performance on multiple benchmarks for image recognition [4]. Until now, technology-based solutions are employed to help recover from pandemic loss. It presents an overview of many developing technologies, including GPS, Wi-Fi, Bluetooth, etc. that could be useful in a variety of real social distancing scenarios. To detect crowds, some researchers have employed drones and other surveillance cameras.

The ODTS (Object Detection and Tracking System) can also be used in fields to monitor dynamic movement of specific object like in this case humans. We know that ODTS is a combination of R-CNN which gives better accuracy. [5]

Hence, object detection algorithms like R-CNN by Girshick et al [6] and YOLO by Redmon et al [7] have been developed to detect multiple classes in a single frame. The most popular object identification algorithm, YOLO (You Only Look Once) is fast in both speed and accuracy.

2.2 Proposed System

A framework model is introduced to continuously check if social distancing is followed or not. YOLOv3 is used because of its robustness and superior performance. YOLOv3 is an open-source object detection algorithm [8] which is employed to detect people in the feed received from a viewing device. The model uses a single-phase network to draw bounding boxes and estimate class probabilities. Transfer learning is applied to enhance the detection of people from the top view.

Centroid of each box detected is calculated repeatedly. We use the Euclidean distance formula to calculate the distance between each bounding box. After this, a fixed threshold is set and checked if distance between every pair of box is greater than the threshold value. If distance between a pair of boxes is less than the defined threshold value, then the pair is put in violate set and color is updated to red. For tracking, a tracking algorithm is adopted to help track those individuals who violate/cross the threshold of social distancing. We observe that total social distance violations and total human count are displayed on the output screen.

As YOLOv3 has high accuracy for object detection, we will use it for human detection. It has a tight architecture for multiple object detection.

From the literature, we conclude that there is certain amount of work to monitor social distance in public settings. Majority of the work is based on frontal or side view of the camera. Therefore, we present a top view social distance tracking framework which overcomes occlusion issues and will play a vital role in monitoring of individuals whether social distancing norms are being followed or not.

Chapter 3

Project Description

3.1 Modules

3.1.1 Object Detection

The first step is to capture the frame we get from the viewing device. Then detect the humans in the frame using the person class and find their centroid.

Once the people are detected, centroid of each bounding box is calculated. We use the Euclidean distance formula and compute the distance among all detected boxes.

After calculating the distance with the help of centroid, a threshold value which is predefined is used to check whether the distance between two centroids of the detected box is less than the number of pixels which are configured. If the distance between a pair is less than the fixed threshold value, then the pair is put in a violate set and the color of the box is updated to red. For tracking, a centroid tracking algorithm is adopted to help track those individuals who violate/cross the threshold of social distancing. We see the total number of social distance violations detected and human count in the final output screen.

3.1.2 Social Distance Detection

Once the people have been detected in the frames, the next step is to calculate their centroid and calculate the distance between each bounding box. The bounding box coordinates (x, y) are used to calculate the centroid. After all calculations, each bounding box is given a unique number to identify and keep track of it.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

By using the Euclidean distance formula, we measure the distance between the centroids of the box detected. For each upcoming frame, we first compute the centroids then compute the distance between each pair of detected boxes. Each centroid calculated information is stored in a list. We define a threshold value to check if a pair of boxes are separated by more than N pixels. If the pair of boxes are separated by a distance less than the threshold value, then are put in a violate set. The color of the bounding box is set to red. If the current index set by violation exists, the color is kept red. A centroid tracking algorithm is used to keep track in the video frame.

$$c = \begin{cases} red & d < t \\ green & d \geq t \end{cases}$$

The algorithm also helps to keep track of people (i.e., if people are following social distancing norms). We see the total number of social distance violations detected and human count in the final output screen.

3.1.3 Realtime Human Count

We count the number of bounding boxes in the frame. We check if the number of bounding boxes is > (greater) or < (less) than the maximum limit of people. If it is greater than the limit value, an alarm is raised, indicating that the maximum number of people has been exceeded than the limit and changes the color of the text to red. If it's not greater than the limit, then it keeps the color to green.

The model displays the total number of human count that is detected by the model.

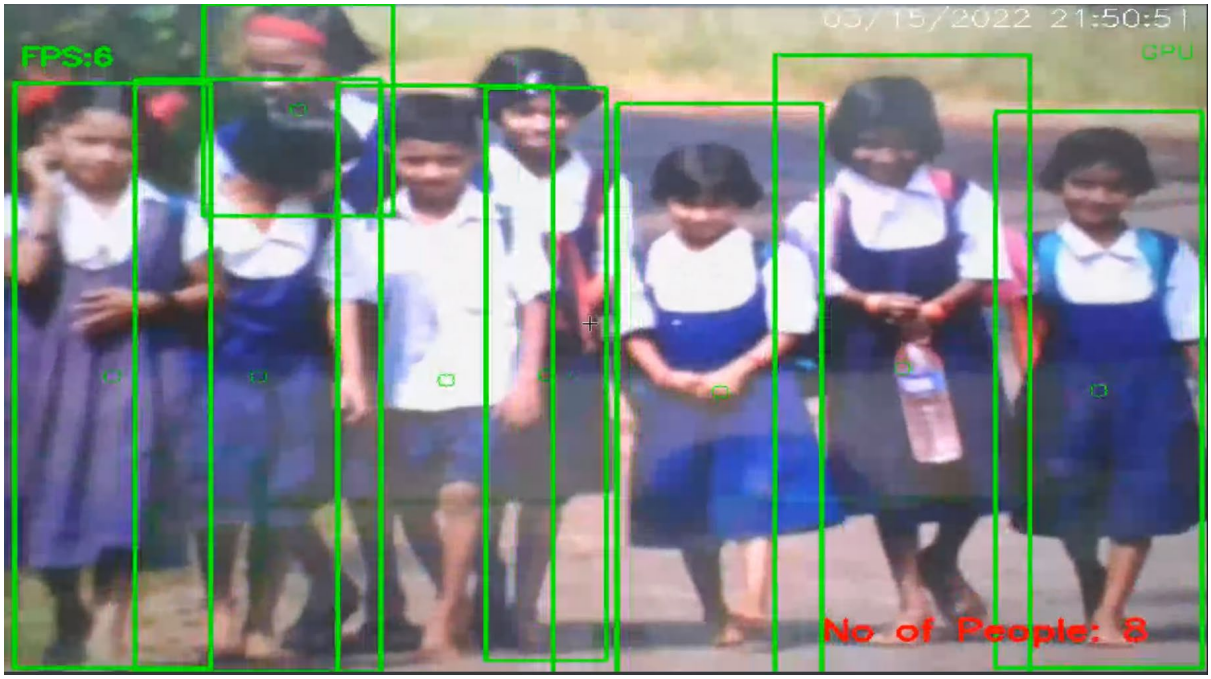

Fig 3.1 School children count

Chapter 4

Analysis

4.1 Functional Requirements

The functional requirements include functionality required by the users from the systems. The different functionalities that the user can use includes usage of GPU (graphics card) by setting up CUDA for the same. This will allow the code to be executed on the GPU (graphics card) rather than CPU (processor). The user can also select the functionality which he likes to use i.e., Social Distance Detection, Human Count or Both at the same time. The user also has the flexibility to use the device camera or mobile camera. For usage of mobile camera, the user must download a third-party application from the play store to utilize the functionality. The user can also choose to record the data in a word, excel or a text file. The user can then start the application and see whether social distance norms are being followed or not.

4.2 Non-Functional Requirements

- **Safety**

  Humans are error prone, but the negative effects of common errors should be limited. E.g., Users can make a mistake while entering the URL from the third-party application, but URL is validated and will inform the user if there is any error in the URL entered.

- **Reliability**

  The system performs according to given specifications and is reliable enough to provide accurate results.

- **Performance Issue**

  The system runs smoothly on a computer with 8GB RAM or more and versions of Intel i5 processors and above. Application windows load in fraction of seconds providing optimal performance of the system.

- **Usability**

  It is very easy for the user to use the application as everything is GUI based with ample of information for the user to navigate easily.

- **Maintainability**

  A record is maintained of all the errors that may occur.

4.3 Hardware & Software Requirements

Hardware Requirements

- Intel i5 processor
- RAM – 8GB
- Hard disk – 100GB
- Monitor, Mouse, and Keyboard
- Webcam
- Graphic processor – Maxwell processor or higher (optional)

Software Requirements

- Programming Languages – Python
- Operating System – Windows 8 and above
- Python libraries and Packages
- CUDA enabled for GPU (optional)
- IP Webcam (Third Party Application for mobile) (optional)

Chapter 5

System Design

5.1 Flowchart for Social Distance Detection

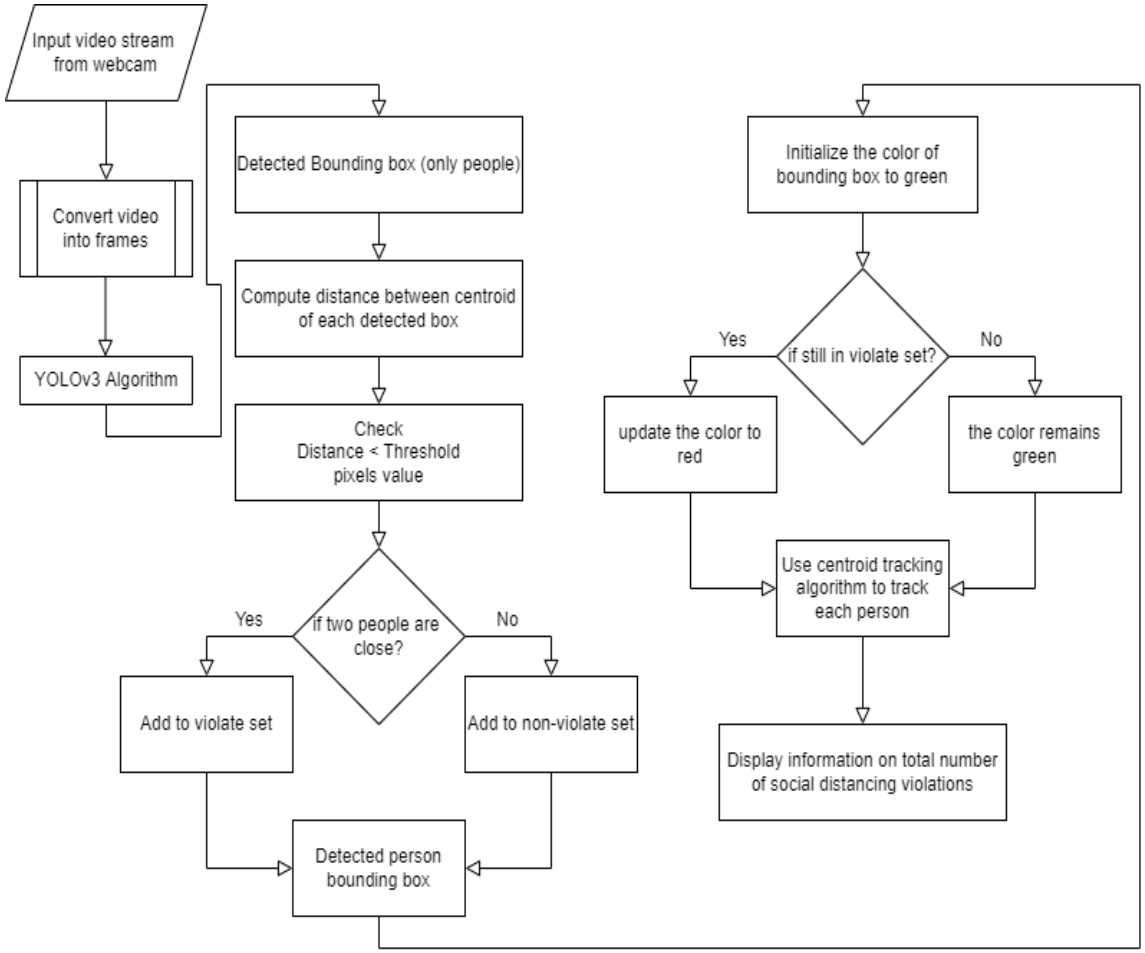

Fig 5.1

5.2 Flowchart for Human Count

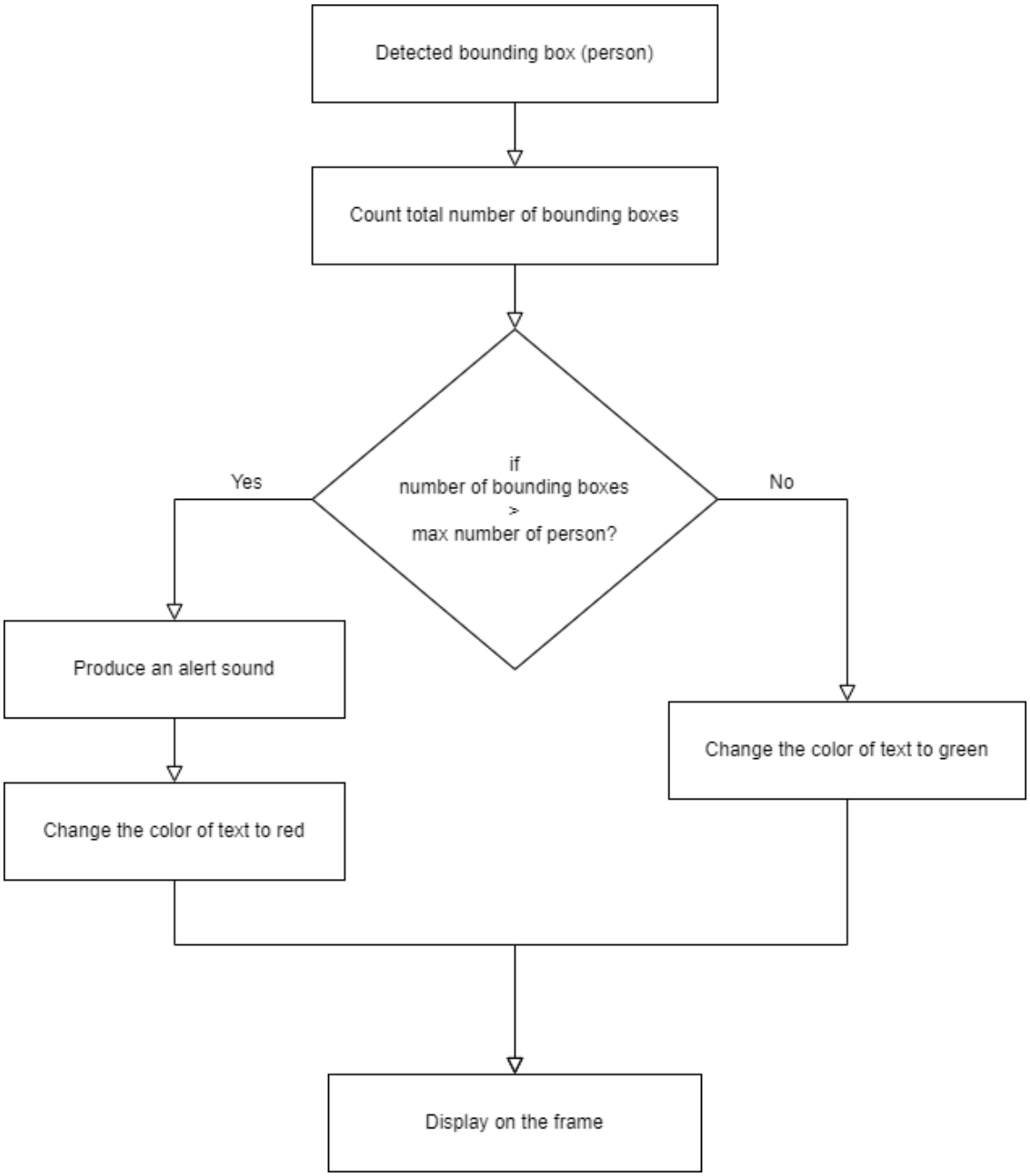

Fig 5.2

5.3 Data Flow Diagram

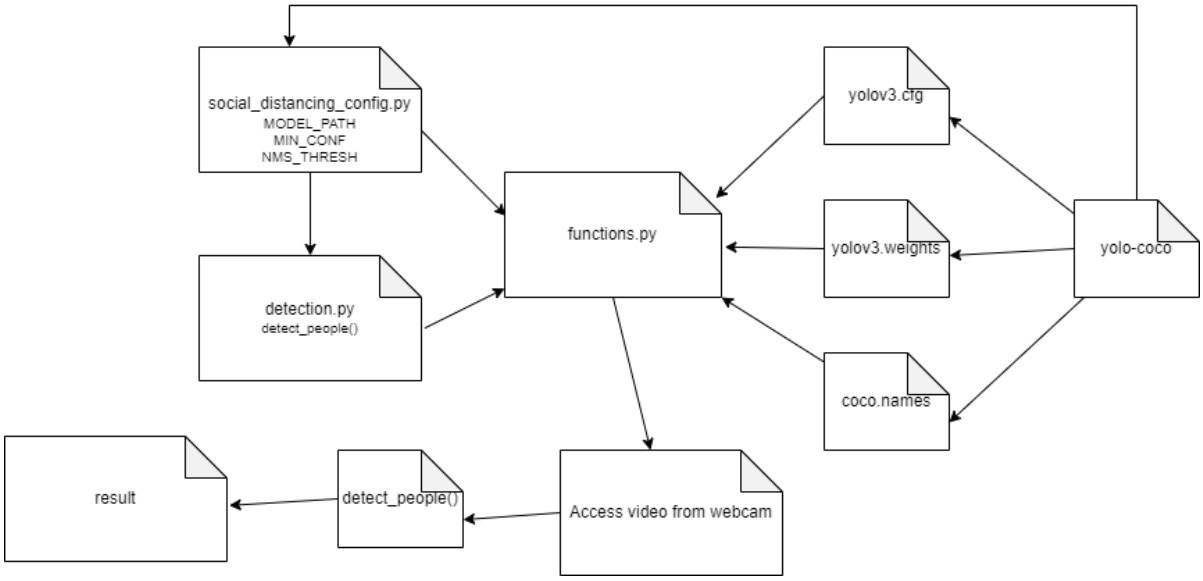

Fig 5.3

Chapter 6

Methodology

6.1 Implementation Methodology

To detect social distance among people in public spaces, a detection tool is designed with user friendly GUI. In this theory, CNN and computer vision techniques are used. YOLOv3, an open-source network-based algorithm is deployed to detect people. From result, a certain class (person class) is used, and rest of the classes are ignored. In this way, the most suitable bounding box for each person can be drawn, and the data is further used for measurement of distance.

During installation, the camera is fixed at an inclined angle, 8-10 feet from the ground. This methodology assumes that pedestrians walk on the plane in the video image. A fixed threshold is set, any pair violating the threshold will be put in the violate set and update color to red. The entire application is developed using Python. Figure 6.1 shows the pipelining and figure 6.2 shows the process of detection tool.

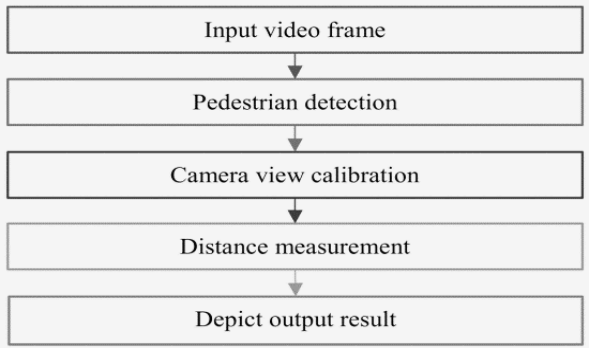

Fig 6.1 Pipeline for Social Distance Detection

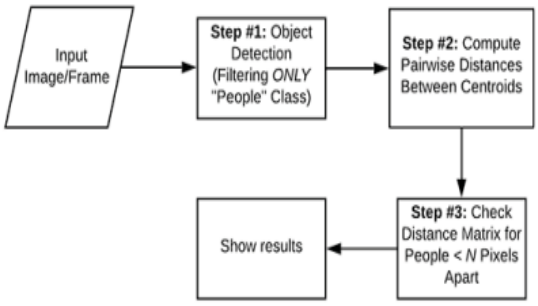

Fig 6.2 Methodology of Video Social Distancing

6.2 Sample Code

* Function to choose functionality

```
      def ChooseOption():
              x1.withdraw()
              global f1
              f1=Toplevel(w)
              f1.focus_set()
              f1.overrideredirect(False)
              f1.attributes("-fullscreen",True)
              f1.state('zoomed')
              f1.resizable(0,0)
              f1.configure(bg='blue')
              e = Example(f1)
              e.pack(fill=BOTH, expand=YES)
              config.USE_SOCIAL_DISTANCE=False
              config.USE_HUMAN_COUNT=False
              config.USE_BOTH=False

              def CUDASetup():
                  f1.attributes('-disabled',True)
                  result = messagebox.askquestion("Information", "Are you
sure? You will be directed to an external webpage.")
                  if result=="yes":

webbrowser.open('https://towardsdatascience.com/installing-tensorflow-with-
cuda-cudnn-and-gpu-support-on-windows-10-60693e46e781')
                  else:
                      pass
                  f1.attributes('-disabled',False)
                  f1.focus_set()

              def on_closing():
                  f1.attributes('-disabled',True)
                  resultquit = messagebox.askquestion("Quit", "Do you want
to quit?")
                  if resultquit=="yes":
                      f1.attributes('-disabled',False)
                      w.destroy()
                  else:
                      f1.attributes('-disabled',False)
                      f1.focus_set()

              f1.protocol("WM_DELETE_WINDOW", on_closing)

              def Back():
                  f1.withdraw()
                  x1.deiconify()
                  x1.state('zoomed')
                  x1.overrideredirect(False)

              menubar = Menu(f1)
              filemenu = Menu(menubar, tearoff=0)
              filemenu.add_command(label="Quit", command=on_closing)
              menubar.add_cascade(label="File", menu=filemenu)
              helpmenu = Menu(menubar, tearoff=0)
              helpmenu.add_command(label="CUDA Setup...",
command=CUDASetup)
              helpmenu.add_separator()
```

```
            helpmenu.add_command(label="Credits...", command=about)
            menubar.add_cascade(label="Help", menu=helpmenu)
            f1.config(menu=menubar)
            backImage=PhotoImage(file = r"Images/BACK.png")
            backButton=Button(f1,image=backImage, compound =
TOP,command=Back)
            backButton.place(x=100,y=700)
            backButton.image=backImage
            tip.bind_widget(backButton,balloonmsg="This will open the
previous page.")

            def SocialDistance():
                if(config.USE_SOCIAL_DISTANCE==True):
                    f1.attributes('-disabled',True)
                    messagebox.showwarning("Warning","Already Selected")
                    f1.attributes('-disabled',False)
                    f1.focus_set()
                else:
                    f1.attributes('-disabled',True)
                    messagebox.showinfo("Select Functionality","Social
Distancing Only Enabled.")
                    f1.attributes('-disabled',False)
                    f1.focus_set()
                    config.USE_SOCIAL_DISTANCE=True
                    config.USE_BOTH=False
                    config.USE_HUMAN_COUNT=False
                    HumanCountButton['state']=DISABLED
                    BothButton['state']=DISABLED
                    nextButton['state']=NORMAL

            def HumanCount():
                if(config.USE_HUMAN_COUNT==True):
                    f1.attributes('-disabled',True)
                    messagebox.showwarning("Warning","Already Selected")
                    f1.attributes('-disabled',False)
                    f1.focus_set()
                else:
                    f1.attributes('-disabled',True)
                    messagebox.showinfo("Select Functionality","Realtime
Human Count Only Enabled.")
                    f1.attributes('-disabled',False)
                    f1.focus_set()
                    config.USE_SOCIAL_DISTANCE=False
                    config.USE_BOTH=False
                    config.USE_HUMAN_COUNT=True
                    SocialDButton['state']=DISABLED
                    BothButton['state']=DISABLED
                    nextButton['state']=NORMAL

            def Both():
                if(config.USE_BOTH==True):
                    f1.attributes('-disabled',True)
                    messagebox.showwarning("Warning","Already Selected")
                    f1.attributes('-disabled',False)
                    f1.focus_set()
                else:
                    f1.attributes('-disabled',True)
                    messagebox.showinfo("Select Functionality","Both
Social Distancing and Realtime Human Count Enabled.")
                    f1.attributes('-disabled',False)
                    f1.focus_set()
```

```
                    config.USE_SOCIAL_DISTANCE=False
                    config.USE_BOTH=True
                    config.USE_HUMAN_COUNT=False
                    SocialDButton['state']=DISABLED
                    HumanCountButton['state']=DISABLED
                    nextButton['state']=NORMAL


            def switchFunc():
                if(config.USE_SOCIAL_DISTANCE==True):
                    CamSelect()
                else:
                    MaxPerson()

            def SwitchStateRender():
                if SocialDButton['state']==NORMAL and
HumanCountButton['state']==NORMAL and BothButton['state']==NORMAL:
                    f1.attributes('-disabled',True)
                    messagebox.showwarning("Warning", "No option is
selected!")
                    f1.attributes('-disabled',False)
                    f1.focus_set()
                else:
                    SocialDButton['state']=NORMAL
                    HumanCountButton['state']=NORMAL
                    BothButton['state']=NORMAL
                    nextButton['state']=DISABLED
                    config.USE_SOCIAL_DISTANCE=False
                    config.USE_BOTH=False
                    config.USE_HUMAN_COUNT=False

            SocialDImage=PhotoImage(file = r"Images/Social_Distance.png")
            SocialDButton=Button(f1,image=SocialDImage, compound =
TOP,command=SocialDistance,state=NORMAL)
            SocialDButton.place(x=100,y=200)
            SocialDButton.image=SocialDImage
            tip.bind_widget(SocialDButton,balloonmsg="Only Social
Distance violations will be displayed.")
            HumanCountImage=PhotoImage(file = r"Images/Human_Count.png")
            HumanCountButton=Button(f1,image=HumanCountImage, compound =
TOP,command=HumanCount, state=NORMAL)
            HumanCountButton.place(x=565,y=200)
            HumanCountButton.image=HumanCountImage
            tip.bind_widget(HumanCountButton,balloonmsg="Only Realtime
Human Count will be displayed.")
            BothImage=PhotoImage(file = r"Images/BOTH.png")
            BothButton=Button(f1,image=BothImage, compound =
TOP,command=Both, state=NORMAL)
            BothButton.place(x=1030,y=200)
            BothButton.image=BothImage
            tip.bind_widget(BothButton,balloonmsg="Both Social Distance
and Human Count will be displayed.")
            resetImage=PhotoImage(file = r"Images/RESET.png")
            resetButton=Button(f1,image=resetImage, compound =
TOP,command=SwitchStateRender)
            resetButton.place(x=650,y=675)
            resetButton.image=resetImage
            tip.bind_widget(resetButton,balloonmsg="This will deselect
the option.")
            titleImage=PhotoImage(file =
r"Images/SELECT_FUNCTIONALITY.png")
```

```
            titleLabel=Label(f1,image=titleImage, compound = TOP)
            titleLabel.place(x=415, y = 100)
            titleLabel.image=titleImage
            nextImage=PhotoImage(file = r"Images/NEXT.png")
            nextButton=Button(f1,image=nextImage, compound =
TOP,command=switchFunc)
            nextButton.place(x=1250,y=700)
            nextButton.image=nextImage
            tip.bind_widget(nextButton,balloonmsg="This will proceed to
the next page.")
            nextButton['state']=DISABLED
```

* Function for the processor selection

```
def SelectRenderer():
        w.withdraw()
        global x1
        x1=Toplevel(w)
        x1.focus_set()
        x1.overrideredirect(False)
        x1.attributes("-fullscreen",True)
        x1.state('zoomed')
        x1.resizable(0,0)
        x1.configure(bg='blue')
        e = Example(x1)
        e.pack(fill=BOTH, expand=YES)
        config.USE_GPU=False
        config.USE_CPU=False
        if os.path.exists("Converted/AllData-Converted.pdf"):
            os.remove("Converted/AllData-Converted.pdf")
        if os.path.exists("Converted/AllData-Converted.xlsx"):
            os.remove("Converted/AllData-Converted.xlsx")

        def CUDASetup():
            x1.attributes('-disabled',True)
            result = messagebox.askquestion("Information", "Are you sure?
You will be directed to an external webpage.")
            if result=="yes":
                webbrowser.open('https://towardsdatascience.com/installing-
tensorflow-with-cuda-cudnn-and-gpu-support-on-windows-10-60693e46e781')
            else:
                pass
            x1.attributes('-disabled',False)
            x1.focus_set()

        def on_closing():
            x1.attributes('-disabled',True)
            resultquit = messagebox.askquestion("Quit", "Do you want to
quit?")
            if resultquit=="yes":
                x1.attributes('-disabled',False)
                w.destroy()
            else:
                x1.attributes('-disabled',False)
```

```
                x1.focus_set()

        x1.protocol("WM_DELETE_WINDOW", on_closing)

        def Back():
            x1.withdraw()
            w.deiconify()
            w.state('zoomed')
            w.overrideredirect(False)

        menubar = Menu(x1)
        filemenu = Menu(menubar, tearoff=0)
        filemenu.add_command(label="Quit", command=on_closing)
        menubar.add_cascade(label="File", menu=filemenu)
        helpmenu = Menu(menubar, tearoff=0)
        helpmenu.add_command(label="CUDA Setup...", command=CUDASetup)
        helpmenu.add_separator()
        helpmenu.add_command(label="Credits...", command=about)
        menubar.add_cascade(label="Help", menu=helpmenu)
        x1.config(menu=menubar)

        def CPU():
            if(config.USE_CPU==True):
                x1.attributes('-disabled',True)
                messagebox.showwarning("Warning", "Already Selected")
                x1.attributes('-disabled',False)
                x1.focus_set()
            else:
                x1.attributes('-disabled',True)
                messagebox.showinfo("Select CPU or GPU", "CPU is
Selected.")
                x1.attributes('-disabled',False)
                x1.focus_set()
                config.USE_CPU=True
                config.USE_GPU=False
                gpuButton['state']=DISABLED
                nextButton['state']=NORMAL


        def GPU():
            def is_cuda_cv(): # 1 == using cuda, 0 = not using cuda
                try:
                    count = cv2.cuda.getCudaEnabledDeviceCount()
                    if count > 0:
                        return 1
                    else:
                        return 0
                except:
                    return 0

            if(is_cuda_cv()==1):
                if(config.USE_GPU==True):
                    x1.attributes('-disabled',True)
```

```
                    messagebox.showwarning("Warning", "Already Selected")
                    x1.attributes('-disabled',False)
                    x1.focus_set()
                else:
                    x1.attributes('-disabled',True)
                    messagebox.showinfo("Select CPU or GPU", "GPU is
Selected.")
                    x1.attributes('-disabled',False)
                    x1.focus_set()
                    config.USE_GPU=True
                    config.USE_CPU=False
                    cpuButton['state']=DISABLED
                    nextButton['state']=NORMAL
            else:
                x1.attributes('-disabled',True)
                messagebox.showerror("Error", "CUDA not enabled for GPU on
this device.")
                x1.attributes('-disabled',False)
                x1.attributes('-disabled',True)
                result = messagebox.askquestion("Setup", "Do you want to
check the steps to setup CUDA? You will be redirected to an external
webpage.")
                if result=="yes":

webbrowser.open('https://towardsdatascience.com/installing-tensorflow-with-
cuda-cudnn-and-gpu-support-on-windows-10-60693e46e781')
                else:
                    pass
                x1.attributes('-disabled',False)
                x1.focus_set()
                config.USE_GPU=False
                config.USE_CPU=False
                cpuButton['state']=NORMAL
                gpuButton['state']=DISABLED
                nextButton['state']=DISABLED
                resetButton['state']=DISABLED



        labelMainImage=PhotoImage(file = r"Images/SELECT_CPU_OR_GPU.png")
        labelMainLabel=Label(x1,image=labelMainImage, compound = TOP)
        labelMainLabel.place(x=420, y = 100)
        labelMainLabel.image=labelMainImage
        cpuImage=PhotoImage(file = r"Images/CPU.png")
        cpuButton=Button(x1,image=cpuImage, compound =
TOP,command=CPU,state=NORMAL)
        cpuButton.place(x=200,y=200)
        cpuButton.image=cpuImage
        tip.bind_widget(cpuButton,balloonmsg="This will use the machine
processor for computation.")
        gpuImage=PhotoImage(file = r"Images/GPU.png")
        gpuButton=Button(x1,image=gpuImage, compound = TOP,command=GPU,
state=NORMAL)
```

```
        gpuButton.place(x=900,y=200)
        gpuButton.image=gpuImage
        tip.bind_widget(gpuButton,balloonmsg="This will use the machine
Graphics Card for computation.")

        def SwitchStateRender():
            if cpuButton['state']==NORMAL and gpuButton['state']==NORMAL:
                x1.attributes('-disabled',True)
                messagebox.showwarning("Warning", "No option is selected!")
                x1.attributes('-disabled',False)
                x1.focus_set()
            else:
                config.USE_GPU=False
                config.USE_CPU=False
                cpuButton['state']=NORMAL
                gpuButton['state']=NORMAL
                nextButton['state']=DISABLED

        resetImage=PhotoImage(file = r"Images/RESET.png")
        resetButton=Button(x1,image=resetImage, compound =
TOP,command=SwitchStateRender)
        resetButton.place(x=650,y=600)
        resetButton.image=resetImage
        tip.bind_widget(resetButton,balloonmsg="This will deselect the
option.")
        nextImage=PhotoImage(file = r"Images/NEXT.png")
        nextButton=Button(x1,image=nextImage, compound =
TOP,command=ChooseOption)
        nextButton.place(x=1250,y=700)
        nextButton.image=nextImage
        tip.bind_widget(nextButton,balloonmsg="This will proceed to the
next page.")
        nextButton['state']=DISABLED
        backImage=PhotoImage(file = r"Images/BACK.png")
        backButton=Button(x1,image=backImage, compound = TOP,command=Back)
        backButton.place(x=100,y=700)
        backButton.image=backImage
        tip.bind_widget(backButton,balloonmsg="This will open the previous
page.")
```

Chapter 7

Result

The results of the Social Distance framework test were visualized with a previously trained model. The results are compared using multiple cctv footages. The people move freely in the video footages.

The size of the person also varies in different places. Since the model only considers the human class (person class); therefore, only a human-like object will be recognized by the model. It performs well and recognizes the bounding boxes of people of different sizes with green rectangles. For example, people whose boxes are marked green indicate that they meet the social distance threshold value. The model is tested for multiple people. In below images, after the person has been recognized, the distance between each recognized limit frame is measured to verify whether the person in the scene violates social distance or not. Pair whose boxes are marked red indicate that they have crossed the social distance threshold value. However, a person's appearance may not be same at all places; thus, the model provides false results. The reason could be that the model may not have been trained for top view, which can be misleading to the model.

Fig 7.1 Social Distance Violations on a street

Fig 7.2 Social Distance Violations on a street

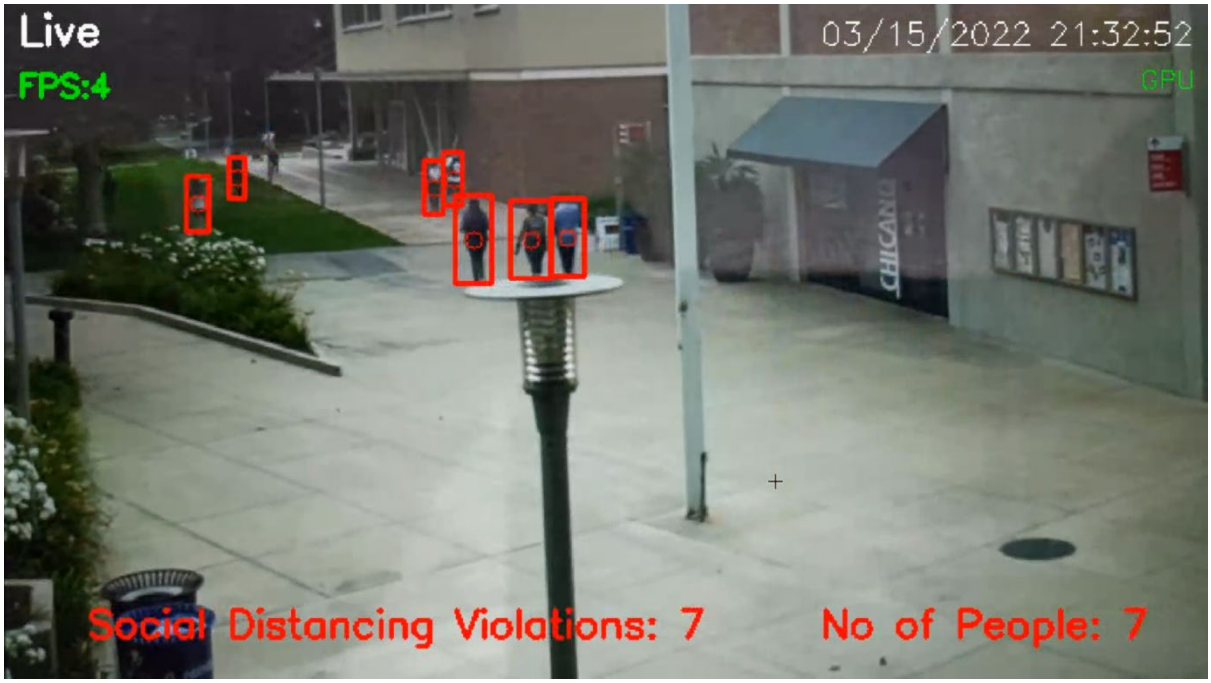

Fig 7.3 Social Distance Violations and Human Count outside a College

Fig 7.4 Social Distance Violations and Human Count outside a Campus

Fig 7.5 Human Count on a street

Fig 7.6 Human Count on an intersection

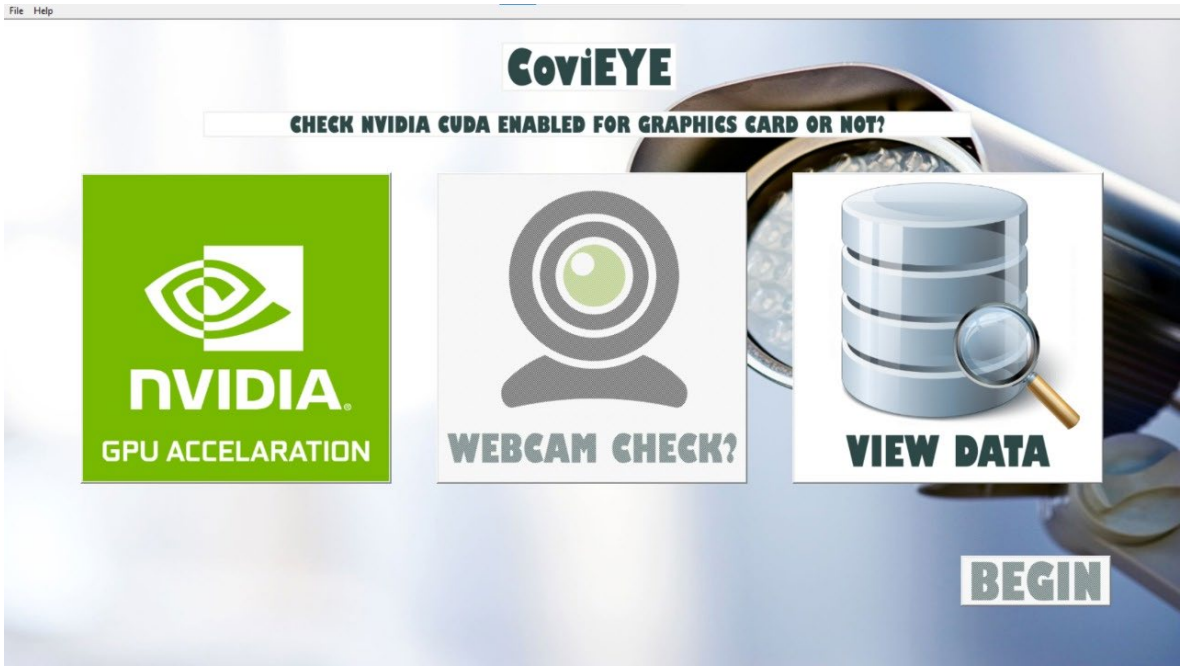

Fig 7.7 Home Screen of the application

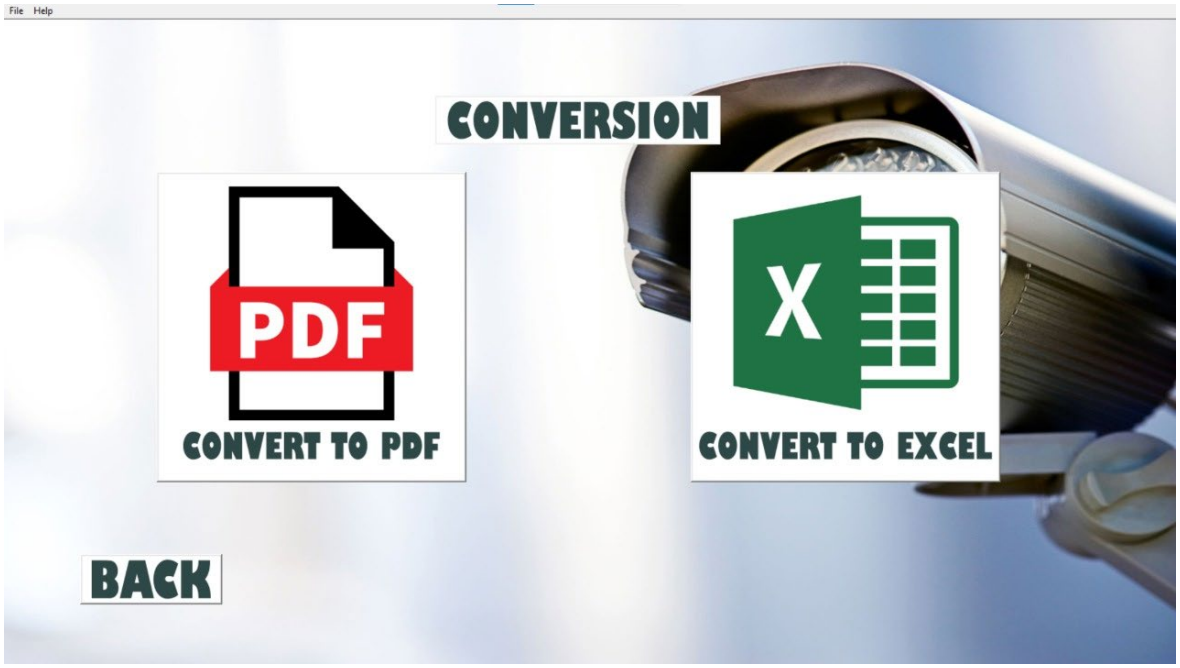

Fig 7.8 Data Conversion Screen

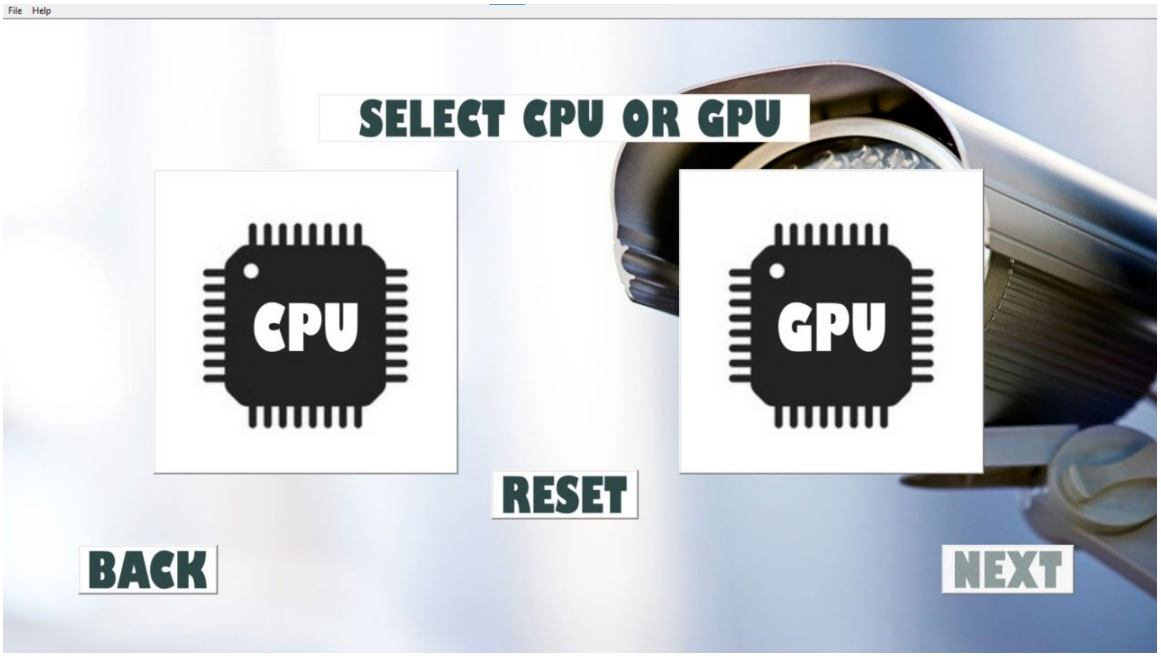

Fig 7.9 Processor Selection Screen

Fig 7.10 Maximum Person Count Limit Screen

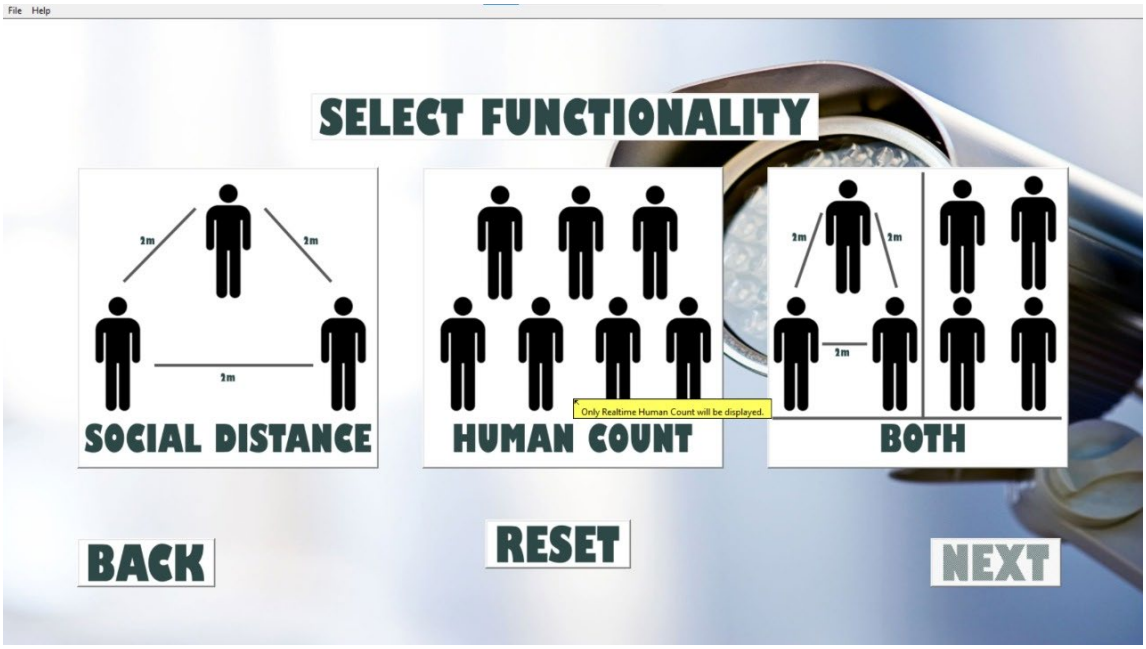

Fig 7.11 Functionality Selection Screen

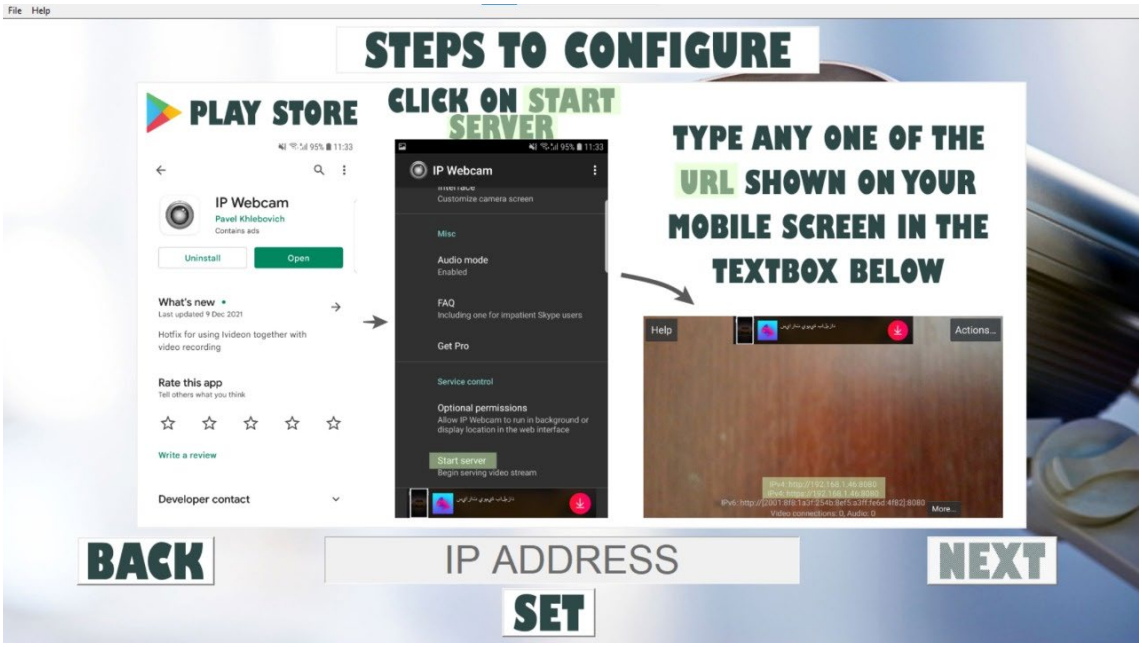

Fig 7.12 IP Webcam Setup Screen (Third party Application)

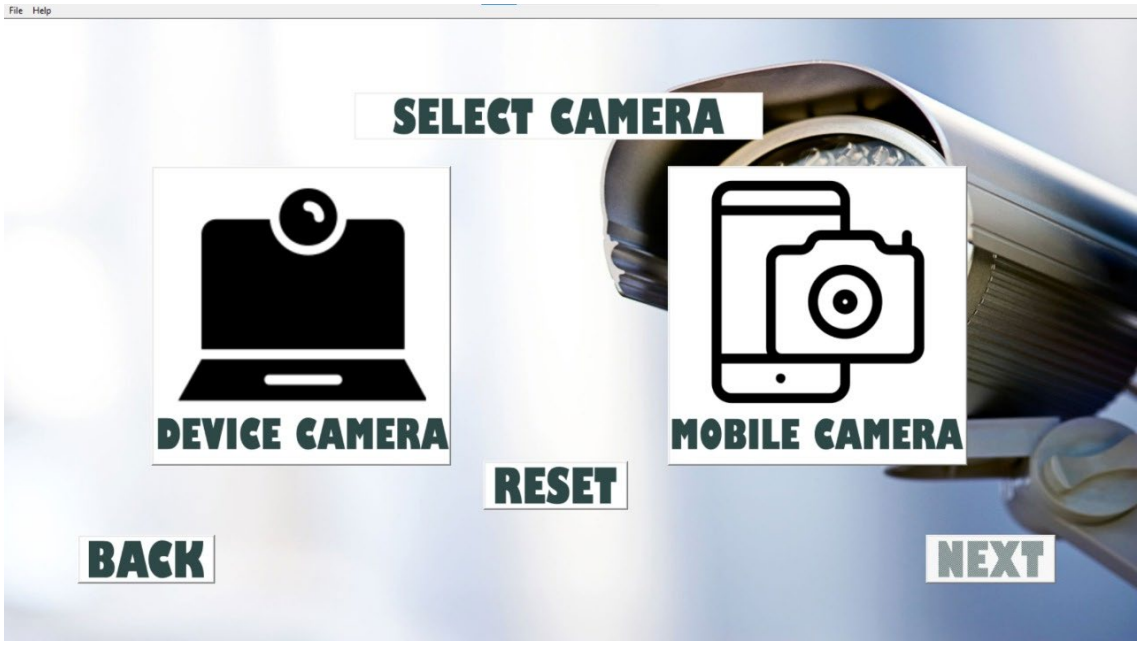

Fig 7.13 Select camera Screen

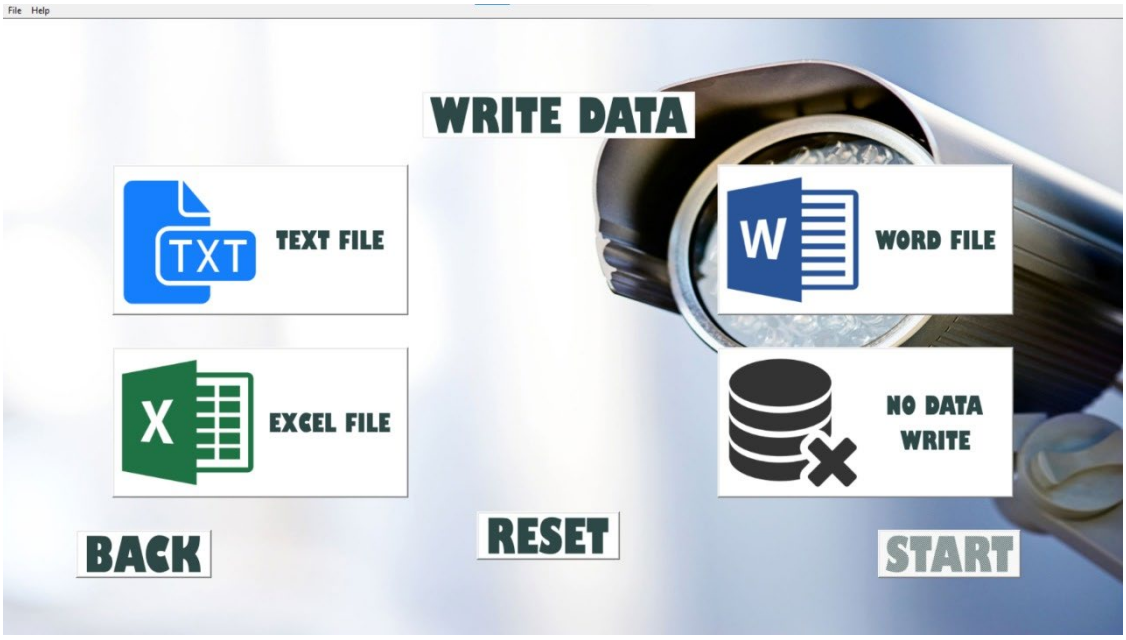

Fig 7.14 Data Write Screen

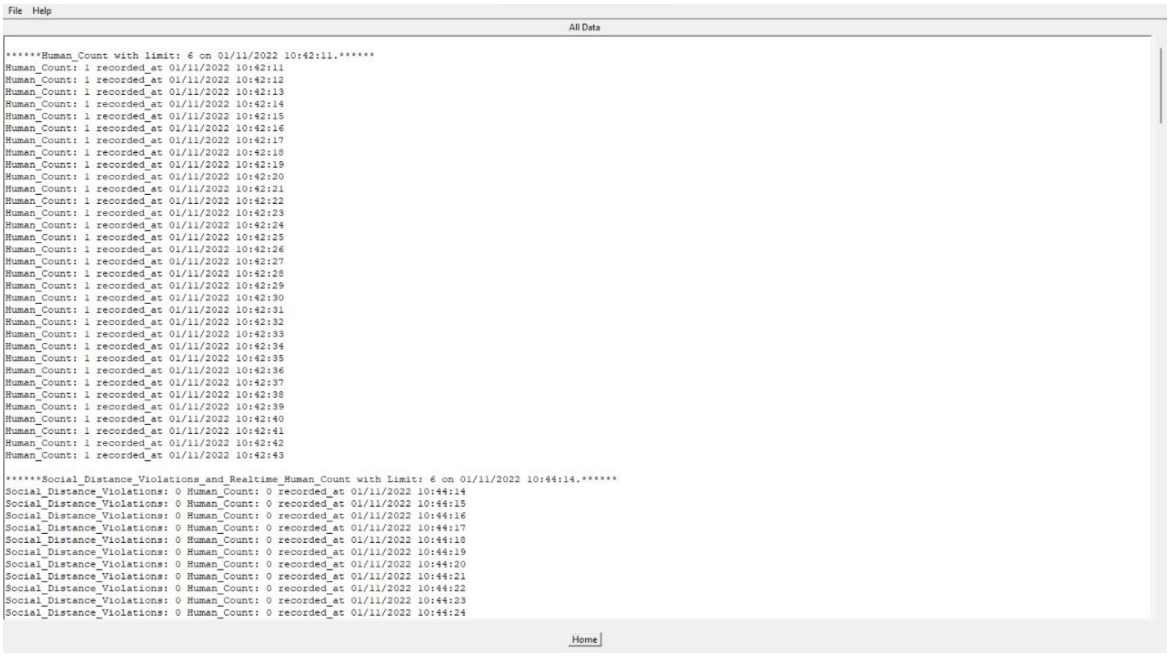

Fig 7.15 View Data Screen

Chapter 8

Conclusion

A methodology is proposed to detect social distancing through a deep learning model. The model can detect people not following social distancing by calculating the distance between them. The model is validated by using multiple cctv footages. The proposed method can detect social distancing and can be developed further and used in universities, malls and workplaces. In addition, further optimizations can be made in the detection algorithm, also integrating mask and body temperature detection within the same application. Improvement can also be made in the calibration of the viewing device and computational power of the hardware.

References

[1] W.H. Organization (2020). https://www.who.int/emergencies/diseases/novel-corona-virus-2019 (Accessed 02 May 2020).

[2] D. Garg, P. Goel, S. Pandya, A. Ganatra and K. Kotecha, "*A Deep Learning Approach for Face Detection using YOLO*," 2018 IEEE Punecon, 2018, pp. 1-4, doi: 10.1109/PUNECON.2018.8745376.

[3] D.T. Nguyen, W. Li, P.O. Ogunbona, "*Human detection from images and videos: A survey*", Pattern Recognition, 51:148-75, 2016.

[4] A. Krizhevsky, I. Sutskever, G.E. Hinton, "*Imagenet classification with deep convolutional neural networks*", In Advances in neural information processing systems, pp. 1097-1105, 2012.

[5] K. B. Lee and H. S. Shin, "*An Application of a Deep Learning Algorithm for Automatic Detection of Unexpected Accidents Under Bad CCTV Monitoring Conditions in Tunnels,*" 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), 2019, pp. 7-11, doi: 10.1109/Deep-ML.2019.00010.

[6] R. Girshick, J. Donahue, T. Darrell, J. Malik. "*Rich feature hierarchies for accurate object detection and semantic segmentation.*" In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580-587. 2014.

[7] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, "*You only look once: Unified, real-time object detection*", In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788. 2016.

[8] Y. C. Hou, M. Z. Baharuddin, S. Yussof and S. Dzulkifly, "*Social Distancing Detection with Deep Learning Model,*" 2020 8th International Conference on Information Technology and Multimedia (ICIMU), 2020, pp. 334-338, doi: 10.1109/ICIMU49871.2020.9243478.

Plagiarism Report

ORIGINALITY REPORT

22%	17%	15%	15%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.coursehero.com Internet Source	5%
2	Imran Ahmed, Misbah Ahmad, Joel J.P.C. Rodrigues, Gwanggil Jeon, Sadia Din. "A deep learning-based social distance monitoring framework for COVID-19", Sustainable Cities and Society, 2021 Publication	4%
3	en.wikipedia.org Internet Source	3%
4	Submitted to Chandigarh College of Engineering & Technology , CCET Student Paper	1%
5	www.bygonebytes.co.uk Internet Source	1%
6	Submitted to CSU, San Jose State University Student Paper	1%
7	ijarcce.com Internet Source	1%

8	www.ijsrd.com Internet Source	1 %
9	upcommons.upc.edu Internet Source	1 %
10	recentscientific.com Internet Source	1 %
11	Bohan Wei, Yao Zhang, Yufan Pu, Yiliang Sun, Shihan Zhang, Hongyu Lin, Changfan Zeng, Yahui Zhao, Kejun Wang, Zhiyong Chen. "Recursive-YOLOv5 Network for Edible Mushroom Detection in Scenes with Vertical Stick Placement", IEEE Access, 2022 Publication	1 %
12	www.theseus.fi Internet Source	<1 %
13	kupdf.net Internet Source	<1 %
14	Submitted to De Montfort University Student Paper	<1 %
15	P. Baskaran, R. Rengarajan, S. Naveen Kumar, V. Vijay. "Chapter 20 COVID-19 Regulation Analysis Using Deep Learning", Springer Science and Business Media LLC, 2022 Publication	<1 %
16	Submitted to University System of Georgia (USG)	<1 %

	Student Paper	
17	Submitted to Shakarim State University Semey Student Paper	<1 %
18	Imran Ahmed, Misbah Ahmad, Joel J.P.C. Rodrigues, Gwanggil Jeon, Sadia Din. "A Deep Learning-Based Social Distance Monitoring framework for COVID-19", Sustainable Cities and Society, 2020 Publication	<1 %
19	codingdict.com Internet Source	<1 %
20	Nikhil Raote, Mohd Saad Khan, Zaid Siddique, Amiya Kumar Tripathy, Phiroj Shaikh. "Campus Safety and Hygiene Detection System using Computer Vision", 2021 International Conference on Advances in Computing, Communication, and Control (ICAC3), 2021 Publication	<1 %
21	aiktcdspace.org:8080 Internet Source	<1 %
22	hdl.handle.net Internet Source	<1 %
23	"Sustainable Communication Networks and Application", Springer Science and Business Media LLC, 2022	<1 %

	Publication	
24	annalsofrscb.ro Internet Source	<1 %
25	core.ac.uk Internet Source	<1 %
26	eprints.utem.edu.my Internet Source	<1 %
27	mafiadoc.com Internet Source	<1 %
28	www.slideshare.net Internet Source	<1 %