EAST WEST UNIVERSITY

Course Title: Digital Speech & Image Processing

Course Code: ICE472

Topic: Project Report

Project Title: Building a COVID-19 Social Distancing Monitoring System.

Submitted To

Muhammad Suhail Najeeb

Lecturer

Department of Electronic & Communication Engineering

Submitted By

Name: Joy Chandra Gope

ID: 2017-3-50-004

Department of ECE

Semester: Spring-2021

Table of Contents

A. GENERAL INFORMATION

Project Overview

- An application system based on the Windows Platform.

  System name or title: Building a COVID-19 Social Distancing Monitoring System.
- System code: MATLAB
- System category:
  - ***Major application:*** Performs clearly defined functions for which there is a Social Distancing Monitoring System whenever we detect people in an image. Later we detect people in video. Followed by finding distance between people to check whether they are safe or in danger.
  - ***General support system:*** Provides general frames support for we save all the frames as a video.

Project References

- ✓ Build your Social Distancing Detection Tool using Deep Learning (analyticsvidhya.com)
- ✓ Social Distancing Monitoring System - File Exchange - MATLAB Central (mathworks.com)

Points of Application

- **Information**

The points of application function information that may be needed by the user for troubleshooting purposes are currently not available.

- **Coordination**

The list of applications that require coordination between the project and its specific support function (e.g., executable coordination, video security, etc.) are currently not available.

- **Save frames as a video feature**

Save all the frames as a video feature including responsible for record the monitoring information available.

B. APPLICATION SUMMARY

Application Configuration

This is windows monitor network application that whenever we are detecting people in an image. Here also we give video as input, so it detects the people in it and find the distance between them to figure it out whether they are safe or in danger. If two or more persons are close to each other then it says that they are in danger, otherwise they are labeled as safe with a specific color tracking around them.

Store data Flows

Users input image and video by using a system, which has a dedicated for inserting valuable information function. Detect people present in every frame, store the calculating the distance from who are near to person, frame record in front of system are supported, and users may change a input image or video after it has been monitoring; similar windows system that may have been the user was trying to save the data will then be automatically capture the recent frame rate.

User Access Levels

The Primary user and other user not be able to modify Social Distancing Monitoring application data and information.

C. FUNCTION REQUIREMENT

For this project, we need all of the following functions which name is exactly the name of the function:

- insertObjectAnnotation()

- peopleDetectorACF('caltech-50x21')

- vision.VideoFileReader(‘video name’)

- vision.VideoPlayer(‘position’,[300 100 1000 500])

- Videowriter('Newfilename.avi')
- Open(writerObj)
- im2frame(I)
- writeVideo (writerObj, frame)
- Close(writerObj)

D. METHODOLOGY

This section provides a general walkthrough of the system application. The logical arrangement of the information shall enable the functional personnel to understand the sequence and flow of the system application.

Logging Keys

This feature describes in general terms the system application first captures the user input image, as well as when the application running in the background it will store the user input video file. When user going to monitoring social distance from a image and video file then automatic store in windows specific folder which means monitoring system that records that particular monitoring video. After store the monitoring information then user get these information.

Detect people in image

To detect people present in image, we will be using the **peopleDetectorACF('caltech-50x21')** function, which is for create a detector. In this feature using detect function we pass the image to the detector created using peopleDetectorACF which gives the bboxes and scores. After detecting we draw rectangles at that points. Then we will find the distances from first person to the remaining persons. How we do that?

- Consider we have two persons here we represent them using rectangle.
- Let's consider the distance between adjacent sides is **dis1_v** and the distance between the outer sides is **dis2_v** distance between the top sides is **dis1_h**. similarly between downsides is **dis2_h.** We find the distances from person one to the person i and here we take the absolute value because sometime they may

overlap which may result a negative distance between them so to avoid the problems we use absolute here.

- If any one of the vertical distances is less than 75 pixels and any one of the horizontal distances is less than 50 pixels then we change the color of the ith persons and first person rectangle to red otherwise we change the color to green.
- Remember that we are comparing only with respect to person one not every person. After that we labeled as safe and danger.

Detecting in Video:

Content of the detecting in video so that we can record maybe what's going on if someone close to another perso. Create a video player using vision.VideoPlayer that is palys it [300 100] width and height of [1000 500]. While file reader is completed get a frame from the videoReader. Then step it to the videoPlayer and end the while loop. Then we detect people present in every frame as we did for one frame just before. Change the image to double. Step the image to detector to find the bboxes and scores, if bboxes is not empty then we put rectangle at every bboxes points available along with scores then change the color and run the script again.

After that we have to find who are close to other person and show they are in danger. For check we use this condition array filled with zeros initially (**Condition = [0 0 0 0 0]**). Number of zeros depends on the number of bboxes present in the image. In this case we take five zero start with first box

- We find the distances between one and two as they are close we increase the elements corresponding to those positions. Here we increase first and second element by one.

  (**Condition = [1 1 0 0 0]**).
- Now we find the distance between one and three as they are far we don't make any changes to conditions matrix.

  (**Condition = [1 1 0 0 0]**).
- Similarly we find 4 and 5 also
- As they also far from 1 no changes make to the condition matrix.
- Next check for 2. We find the distance between 2 & 3. Remember we already find the distance between 1 & 2 so we neglect that has 2 & 3 far no changes made to condition matrix.

  (**Condition = [1 1 0 0 0]**).

- Similarly, 2 to 4 & 5 also
- After that we check 3 with respect to 4 and 5.

(**Condition = [1 1 1 1 0]**)

- Has 3 & 4 are close we increase the element 3 & 4 in condition matrix by 1.
- 3 & 5 far to each other so no changes made.
- Similarly we check for the remaining boxes also.
- Finally get the element & condition matrix with 0 we consider it is safe otherwise they are in danger as they close to other people.

Create loop to check for every person with others. Then check for the distance if they are close to each other then we increase the corresponding elements in conditions array otherwise we keep it as zero. Then draw the boxes around them. If the element is greater than zero then we put red color around it along with label danger, if the element is zero then we put green color around it along with label safe. Release them at the end.

After run we see that here two persons are detected as one. To overcome it we mentioned the specific detector here **detector = peopleDetectorACF('caltech-50x21').**

Save this frames as video

Provide some useful functions to save video.

- Videowriter('NewVision.avi') creates new video file writer
- writerObj frame rate sets the frame rate of the video
- Open(writerObj) starts writing video
- writeVideo (writerObj, frame) adds the frame to the writer object
- Close(writerObj) stops writing video
- Convert the image to frame (im2frame(I))
- Then add the frame to the Videowriter
- Finally close writing video

It's saving the video frame by frame. we find delay in the video here but there will be no frame drop in the video as we set the frame rate to 8.

E. SPECIAL INSTRUCTIONS FOR ERROR CORRECTION

As a condition of your use of the application, you will not use the other services for any purpose that is prohibited. You may not use the other services in any manner that could damage, disable, overburden, or impair any windows system.

F. RESULT AND OTHER FEATURE

Detect people in image

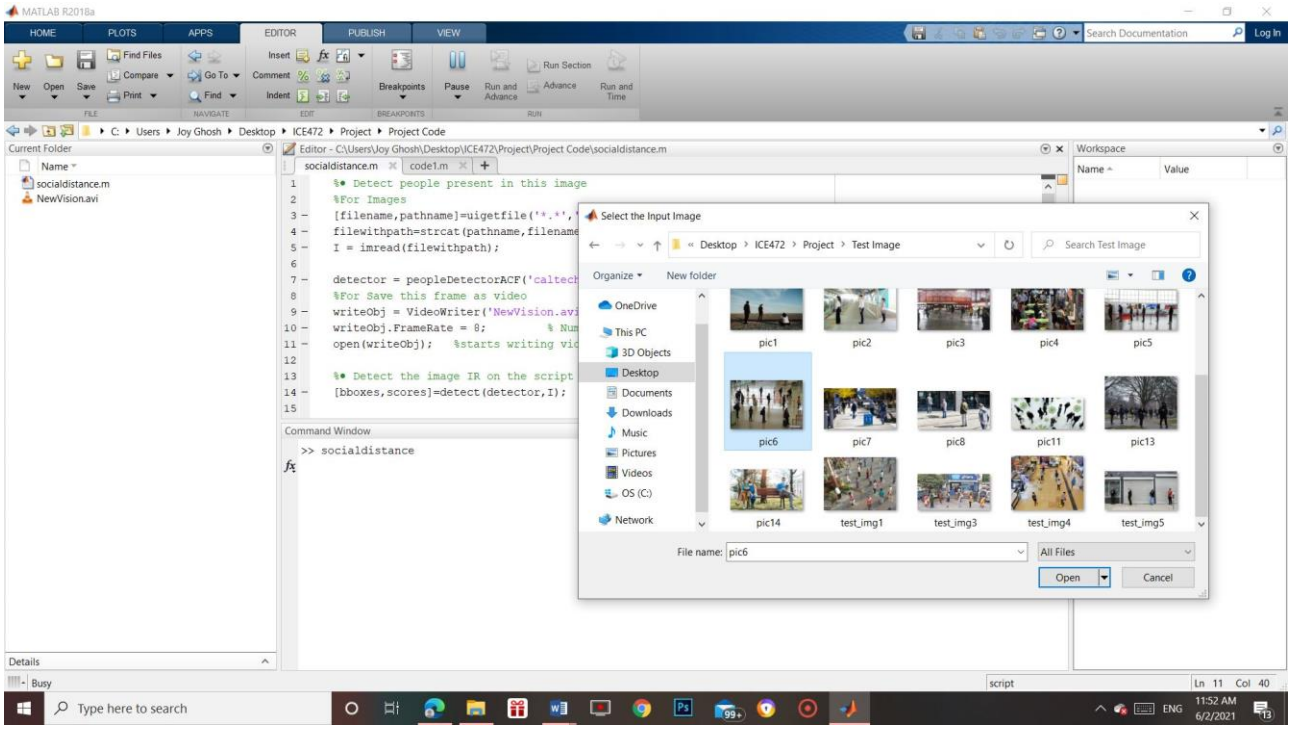


Fig: input image

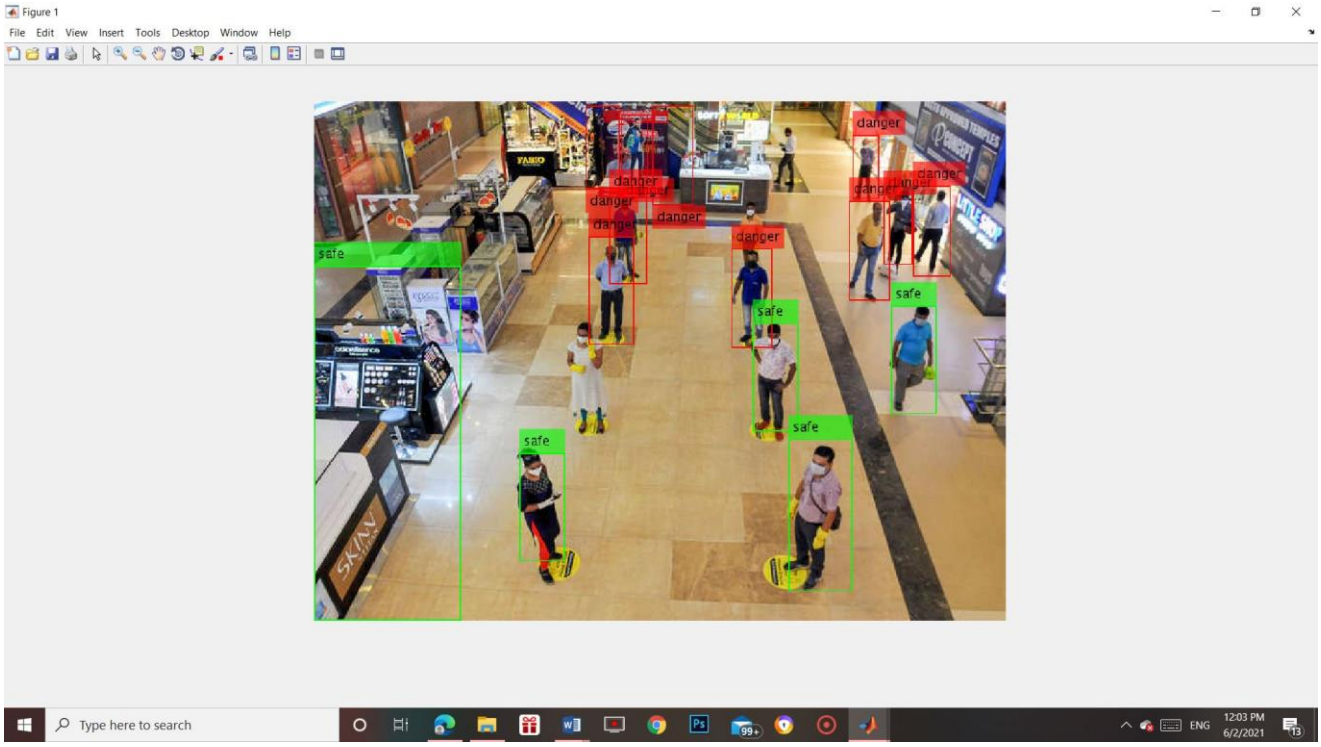


Figure 1
File Edit View Insert Tools Desktop Window Help
danger
safe
Type here to search
ENG
12:03 PM
6/2/2021

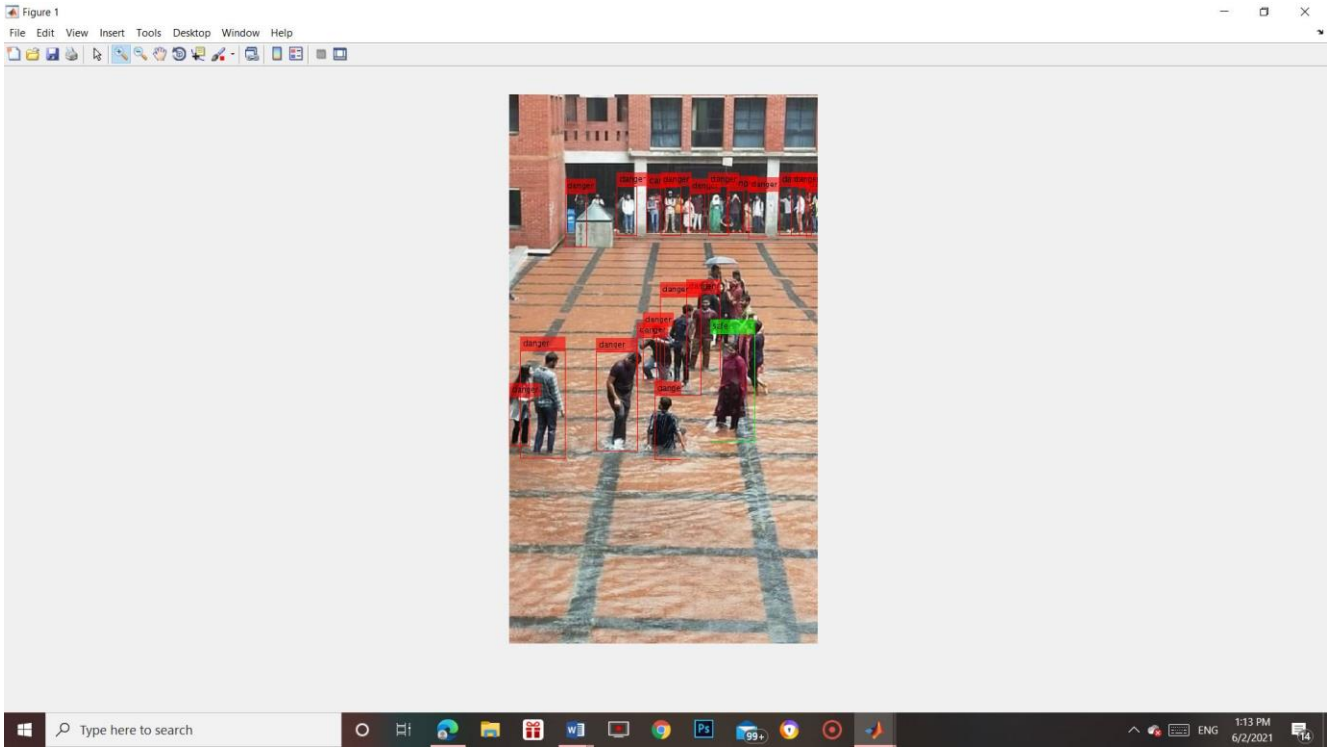


Figure 1
File Edit View Insert Tools Desktop Window Help
Type here to search
ENG
1:13 PM
6/2/2021

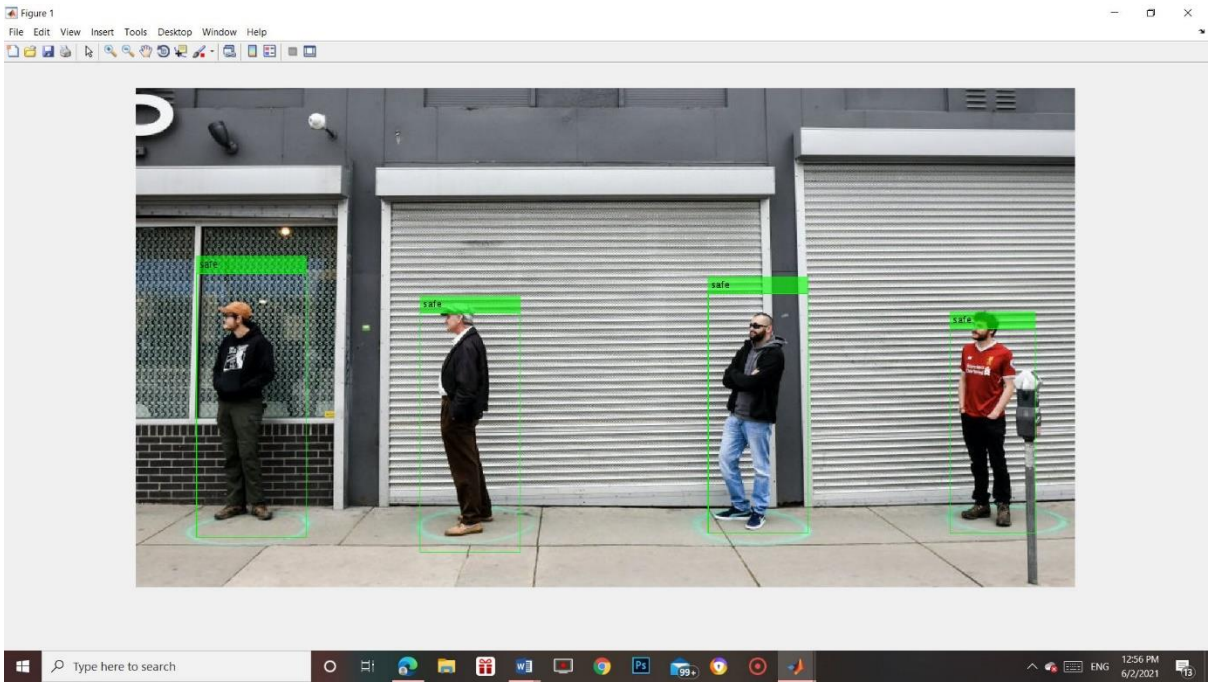


Fig: detect social distance in image

Detect people in video

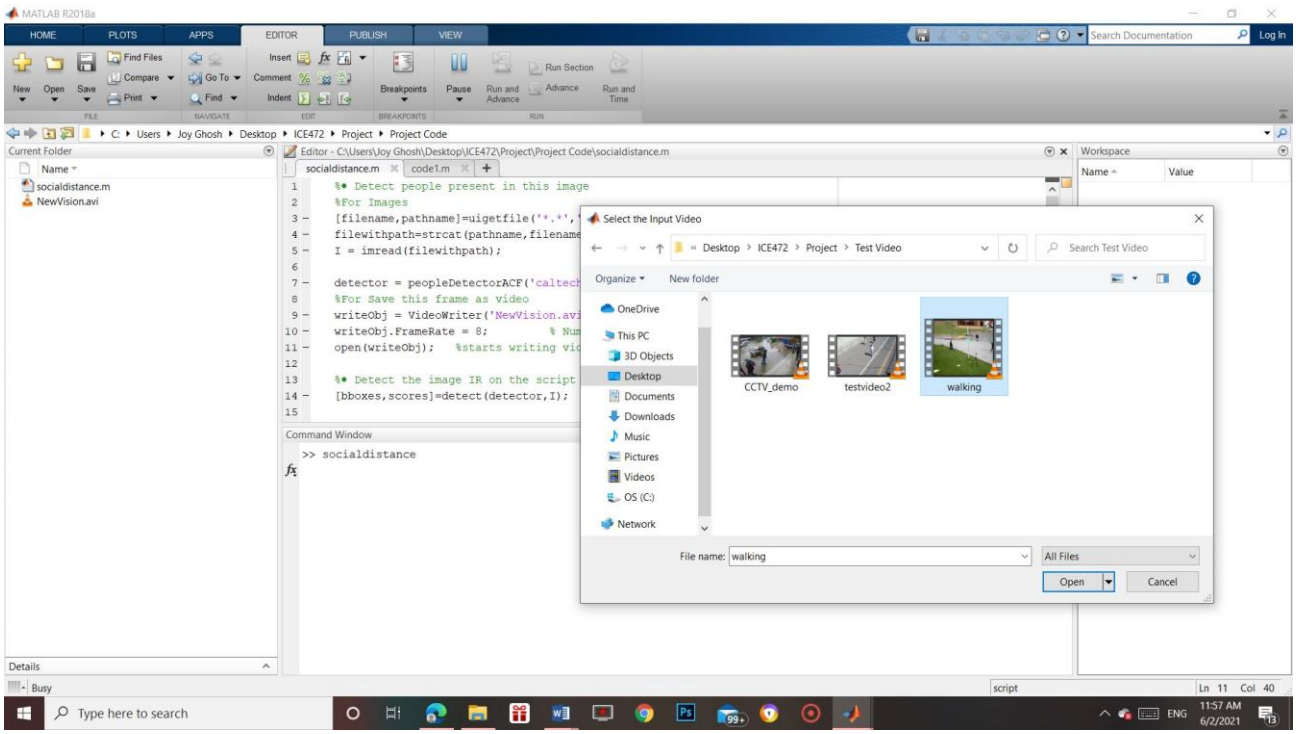
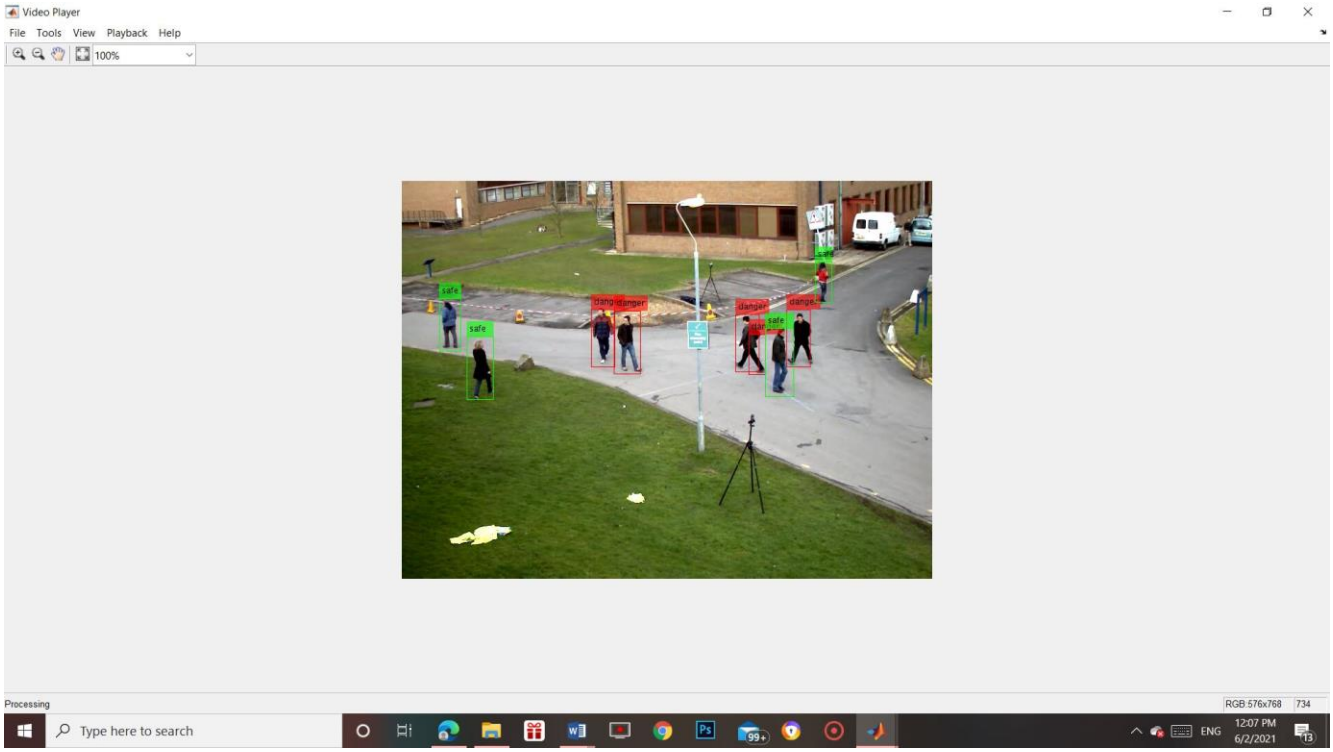
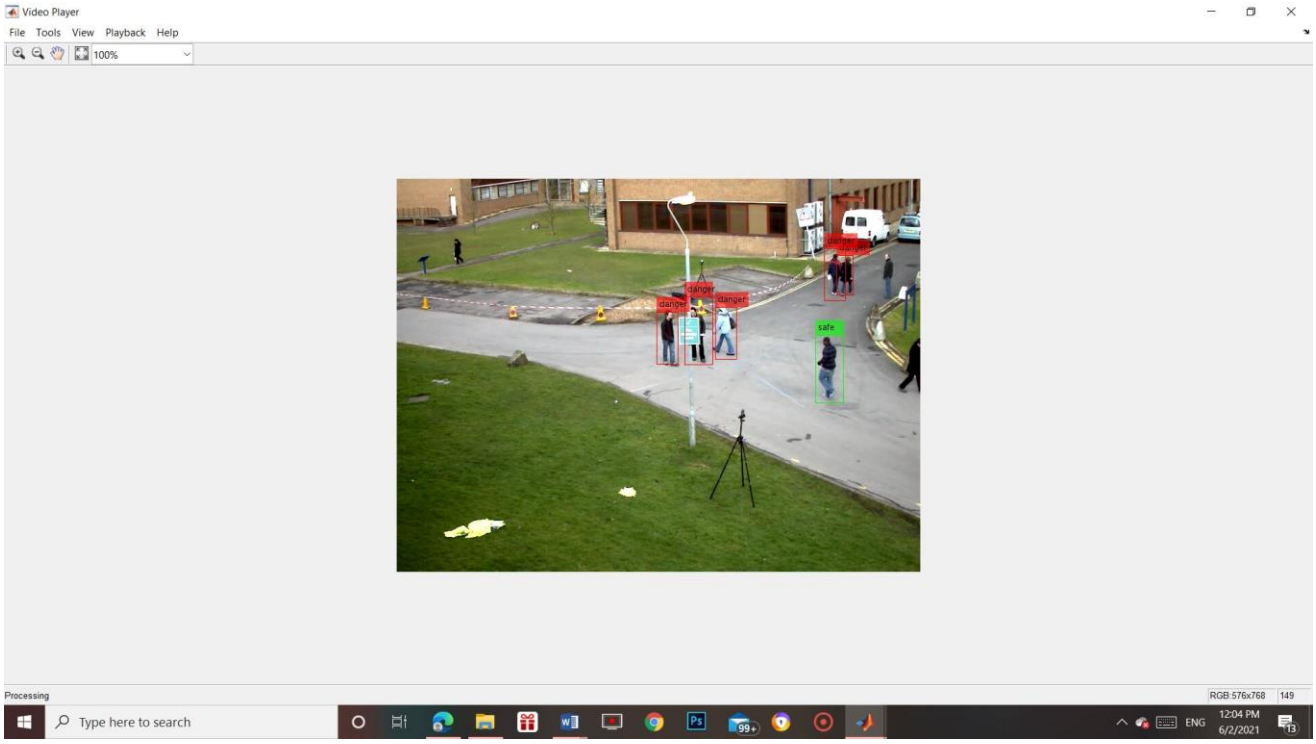


Fig: input video


Video Player
File Tools View Playback Help
100%
safe
danger
Processing
RGB 576x768
734
Type here to search
ENG
12:07 PM
6/2/2021


Video Player
File Tools View Playback Help
100%
danger
safe
Processing
RGB 576x768
149
Type here to search
ENG
12:04 PM
6/2/2021

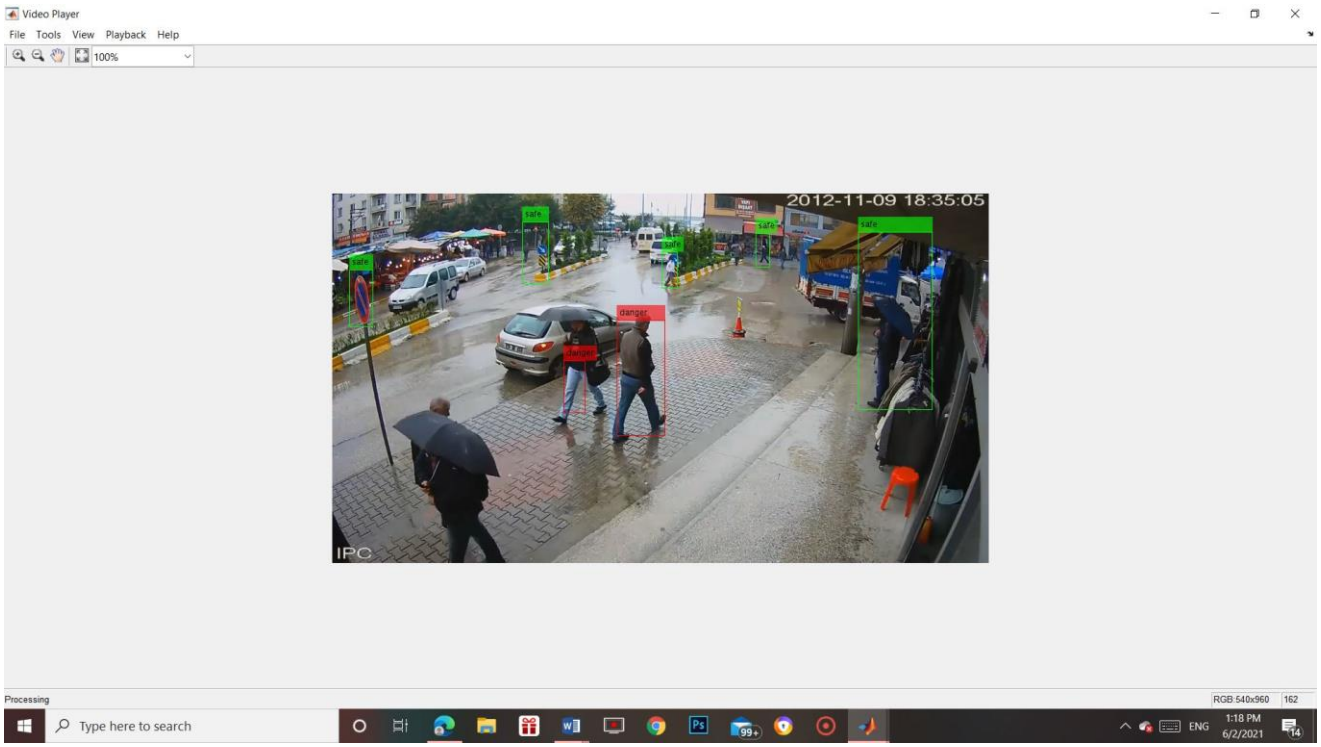
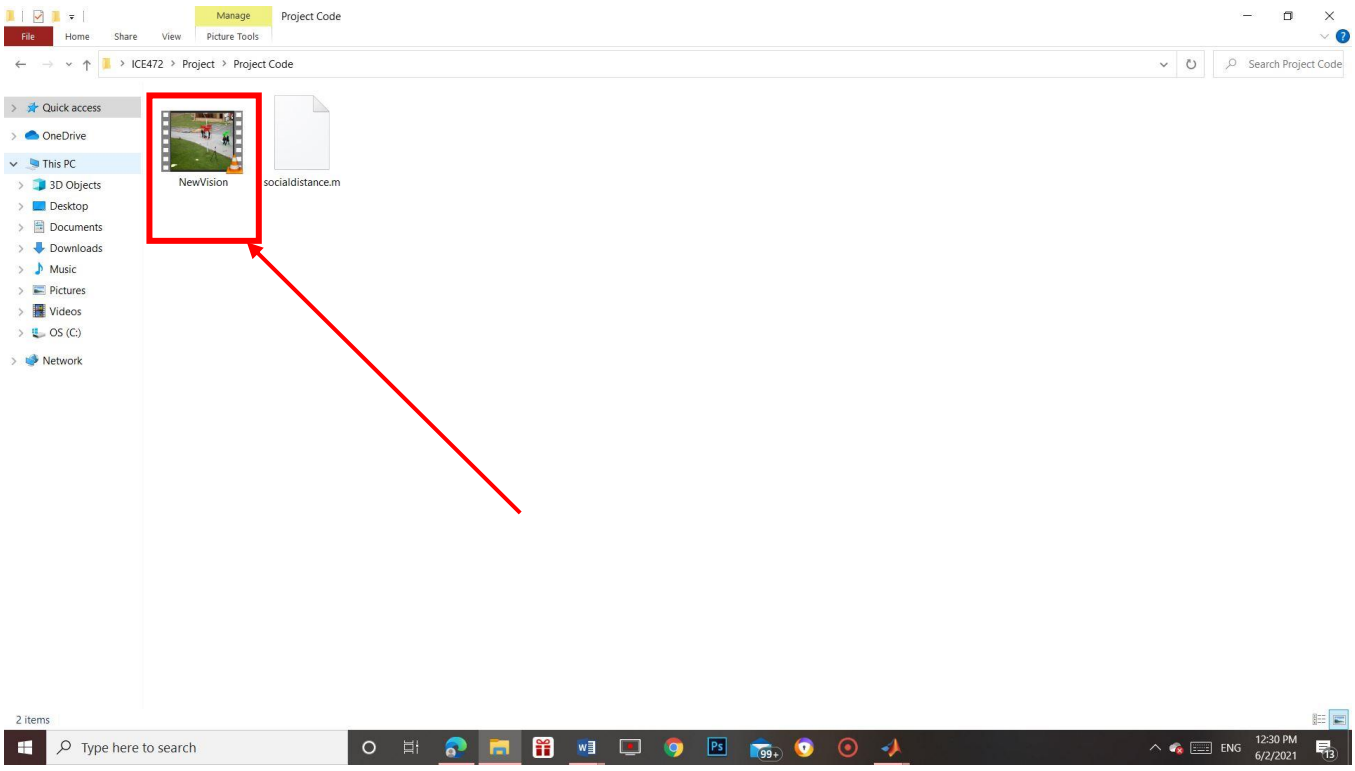


Fig: detect social distance in video

Save this frames as video



G. CONTRIBUTIONS

Saving the video frame by frame. we find delay in the video here but there will be no frame drop in the video as we set the frame rate to 8. Also, After run we see that here two persons are detected as one. To overcome it we mentioned the specific detector here **detector = peopleDetectorACF('caltech-50x21').**

Create loop to check for every person with others. First one gives the reference person. Second loop helps us to move to check with every other person. We find the distance between the ith and jth person. Then check for the distance if they are close to each other then we increase the corresponding elements in conditions array otherwise we keep it as zero. Then draw the boxes around them.

H. GOAL OF THIS PROJECT

The goal of this project is to prevent the spread of COVID-19. Because the biggest cause of concern is that COVID-19 spreads from person to person through contact or if you're within close proximity of an infected person. The only way to prevent the spread of COVID-19 is Social Distancing. Keeping a safe distance from each other is the ultimate way to prevent the spread of this disease. For example, in our project we are detecting people in an image. Here also we give video as input, so it detects the people in it and find the distance between them to figure it out whether they are safe or in danger. If two or more persons are close to each other then it says that they are in danger, otherwise they are labeled as safe with a specific color tracking around them.

I. FUTURE ENHANCEMENT

Platform: In future for better development of this Social Distancing Monitoring System, we are thinking that in future we will try to add more feature like showed better results for real-time performance vs other object detectors. Additionally, later on we will use this technique on versatile cameras, e.g., mounted on a self-sufficient autonomous drone system, and more effective to capture fast actions of the detected objects from different angles. This is a small approach only but not an accurate model. In future we can find the way for remove these errors.

J. CONCLUSION

An social distancing monitoring system is designed, developed, and demonstrated to detect logging keys, Detect people in image, Detect people in video, and Save this frames as video details based on the windows platform which is one of the useful method. We can see the boxes

around people with colors red comes when people are close otherwise we have green color. They are some small errors in this model also sometimes it shows boxes even there is no one and sometimes even two persons are close it show they are safe. So this is a small approach only but not an accurate model.

The experience of working in project related functions and integration of feature developed independently, with just requirement specifications, is a very important achievement for us.