**Building a COVID-19 Social Distancing Monitoring System.**

In this project we’re going try out an experiment on social distancing using Matlab. We’re going to give video as input and we will know whether we’re in a safe distance from others or not.

Let’s divide and conquer:

1. First we will know the functions used.
2. Then we detect people in an image.
3. Later we detect people in video.
4. Followed by finding distance between people to check whether they are safe or in danger.
5. Finally we save all the frames as a video.

Functions used in this project

Make a video out of it

Find the pixels distance between detected

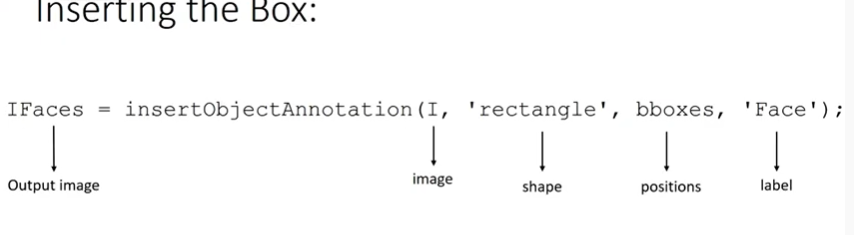
Detect people in video

Detect people in an image

Now Let’s make ourselves familiar with some functions:

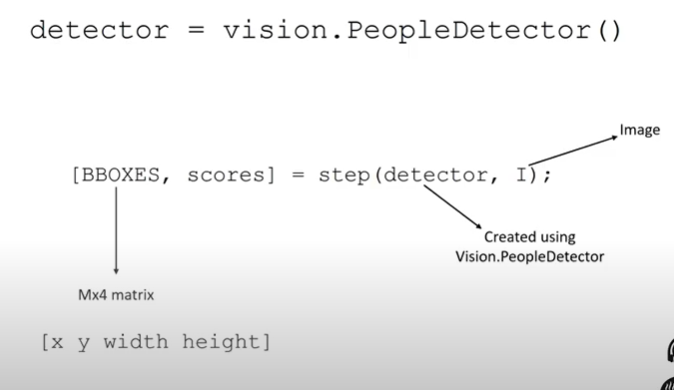
1. first one is **insert object annotation function:**

* for input imageite we put shapes gives with given label at the points given as b box points to get something like this



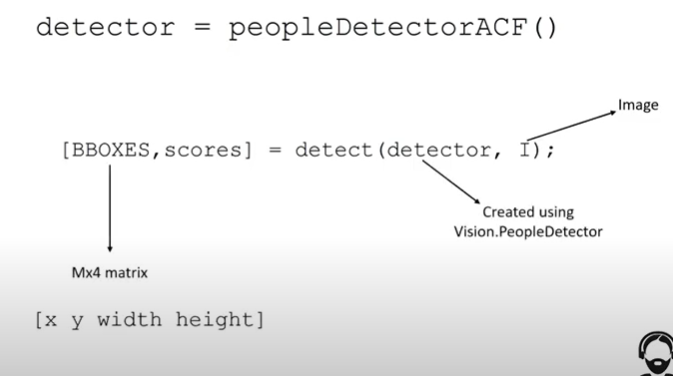
1. Second function is form **computer vision toolbox**:

* it is **vision.PeopleDetector()**
* Its help to detect people. we detect the people in image eye with the detector using step function. It gives bboxes which contain initial positions width and height of the people detected in the image.



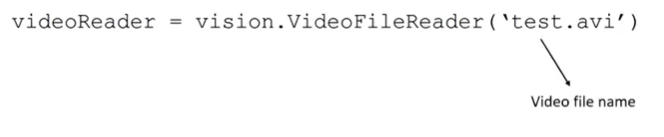
1. Another Function for Similar Operation is **peopleDetectorACF()**:

* That also detect people. Have we use detect function to find people



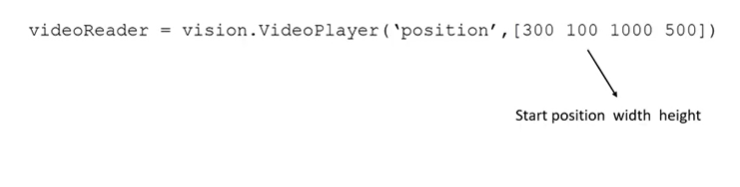
1. Next Function is vision.**VideoFileReader(‘video file name’):**

* It helps in reading a video file given.



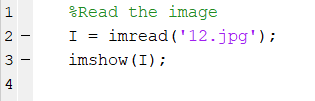
1. Another function is **vision.VideoPlayer(‘position’,[300 100 1000 500]):**

* As we expect it helps to play the video
* Here position represent the windows starting positions and width and height of the video player



**Detect people in image:**

1. Read the image **img1.jpg** available in the matlab and display it:

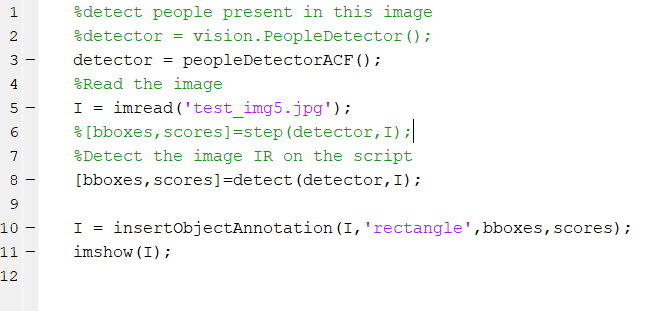


1. Now we will detect people present in this image:

* For this create a detector using **vision.PeopleDetector()**
* Then step the detector on the image to find the bboxes and scores of the detected boxes
* After detecting we draw rectangles at that points.

1. Instead of vision.PeopleDetector() we will use another function to detect the people because in vision.PeopleDetector () not detect all people.

* We use **peopleDetectorACF()** funcrion to detect people in an image
  + 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.





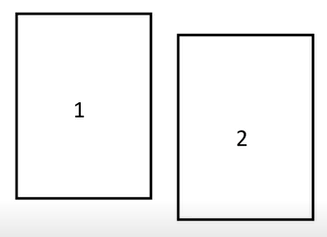
1. We got all of them these values are the scores of the images. We can also replace them with number. Instead of scores here we give number from one to the total people present in the image

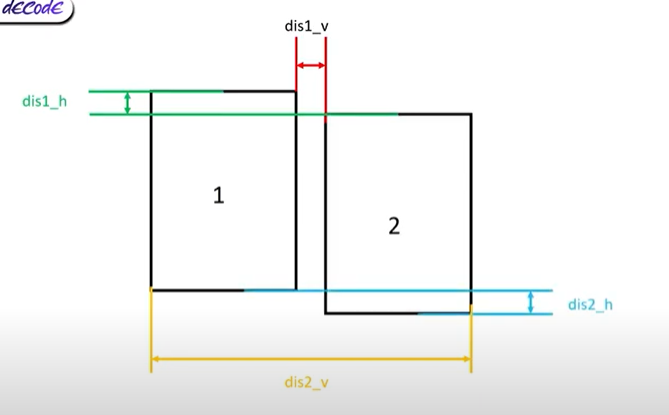




1. Now we will find the distances from first person to the remaining persons. [Remember here we are calculating the distance from person one only, we are find who are near to person one only]

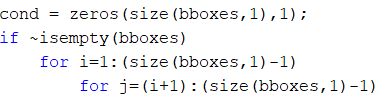
How we do that?

* 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**

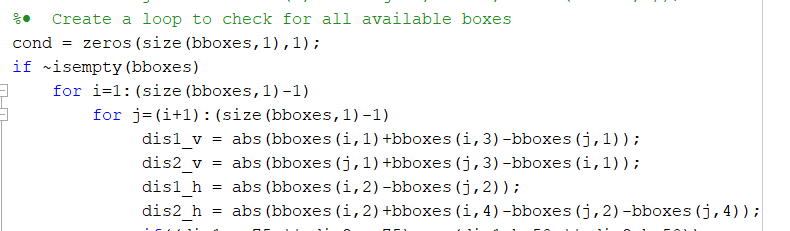


Code:

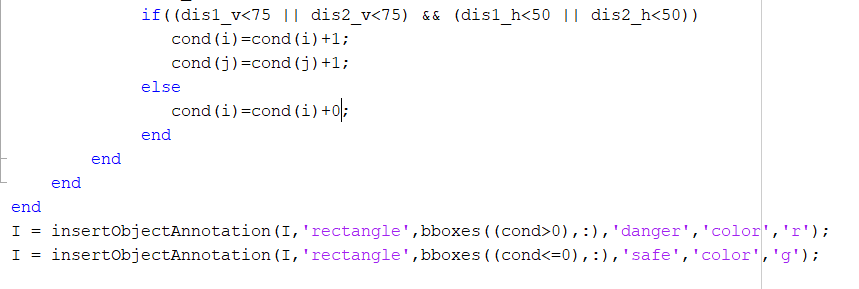
* Create a loop to check for all available boxes:

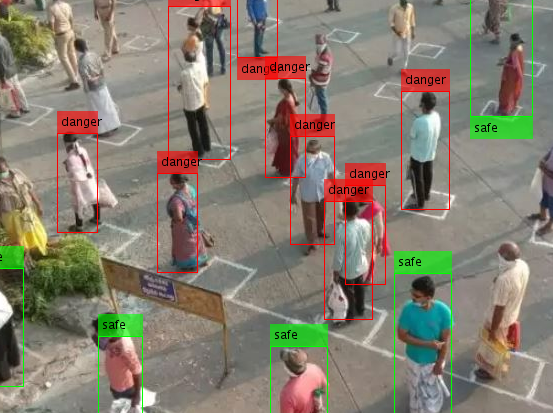


* Now find the distances from person one to the person i and here we take the absolute value because sometime they may overlap which may result a negative distance between them so to avoid the problems we use absolute here.



* 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

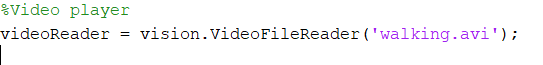




* We can see the person’s rectangle closer to the first person become red whereas the person who are far from person one has green color rectangles around them. Remember that we are comparing only with respect to person one not every person. After that we labeled as safe and danger.

**Detecting in Video:**

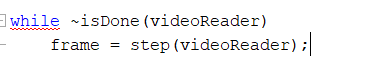
1. Read the file into video reader using vision.VideoFileReader(‘video name’)



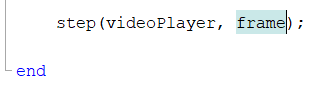
1. Create a video player using vision.VideoPlayer that is palys it [300 100] width and height of [1000 500]



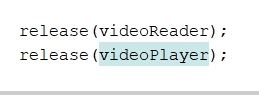
1. While file reader is completed get a frame from the videoReader

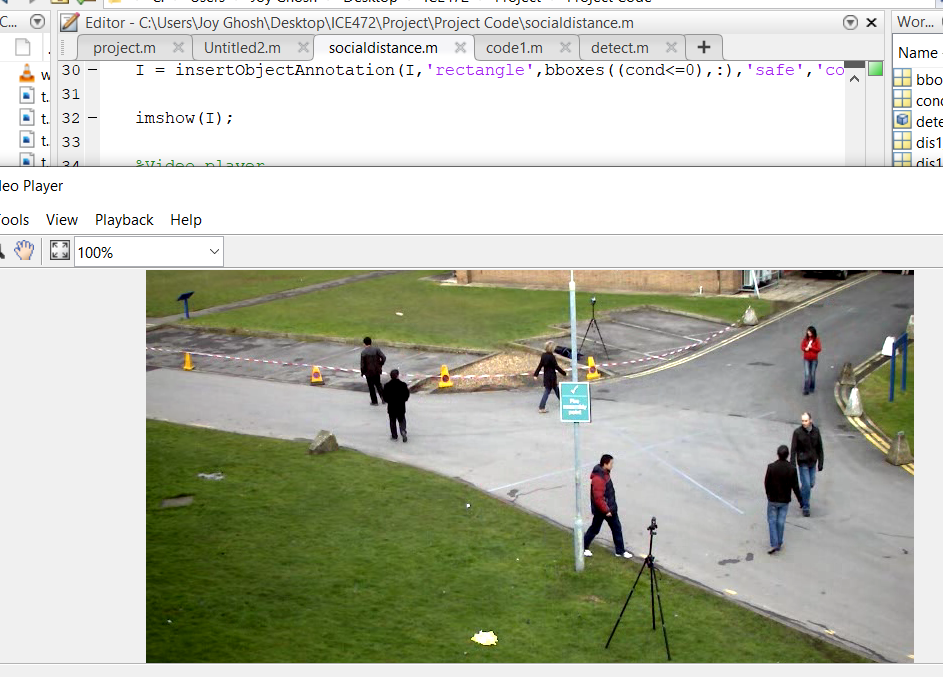


1. Then step it to the videoPlayer and end the while loop



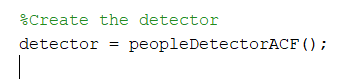
1. Release them at the end



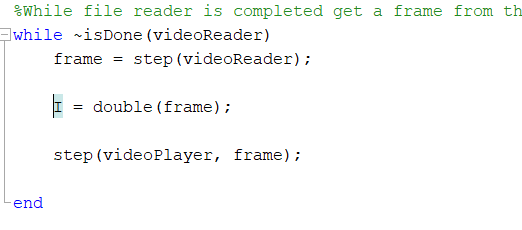


Here the frame rate comparatively high.

1. Now we detect people present in every frame as we did for one frame just before
2. Create the detector before the while loop



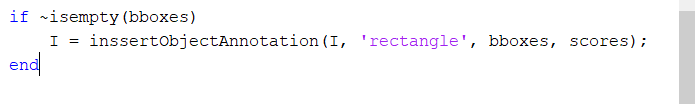
1. Change the image to double



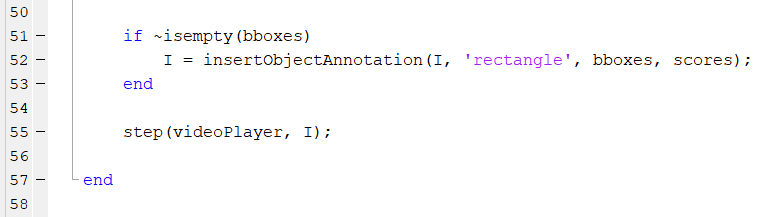
1. Step the image to detectpr to find the bboxes and scores



1. If bboxes is not empty then we put rectangle at every bboxes points available along with scores



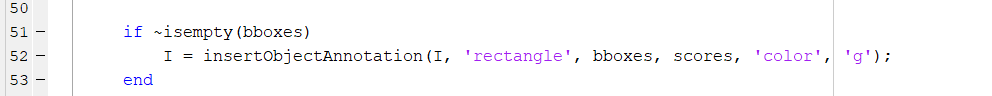
1. S





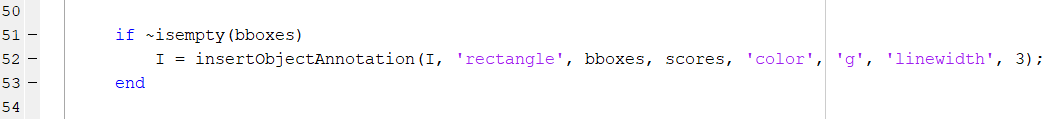
It can detect a people though there are some errors

1. Change the color and run the script again



Remember videoReader doesn’t stop even we close the window so we have to stop the process manually using command prompt clc

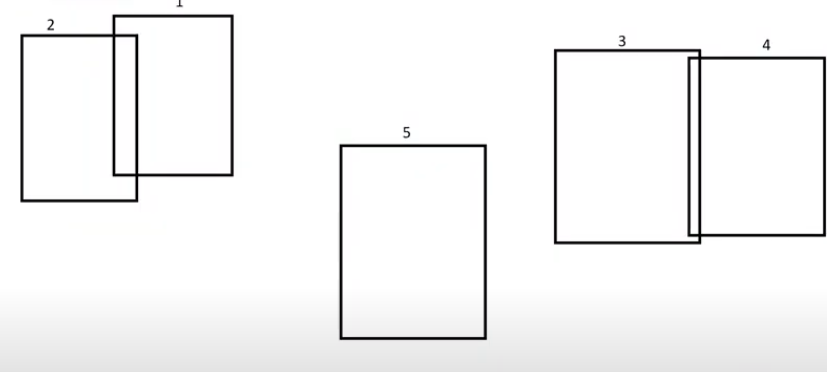
1. Now increase the line width





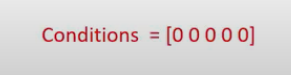
**Safe and Danger:**

1. Consider there are some boxes detected like this in an image



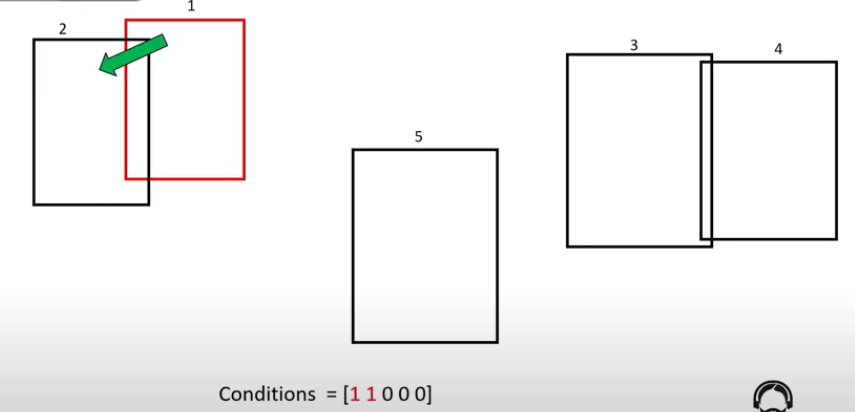
Now we have to find who are close to other person and show they are in danger.

For check we use this condition array filled with zeros initially

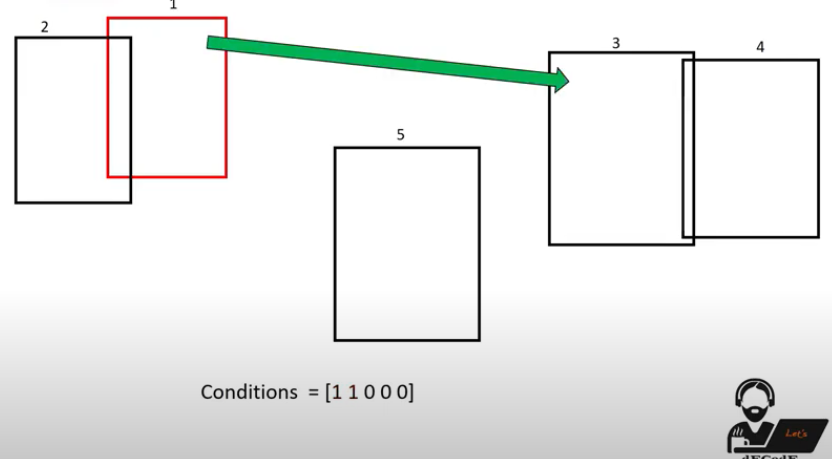


Number of zeros depends on the number of bboxes present in the image. In this case we take five zero start with first box

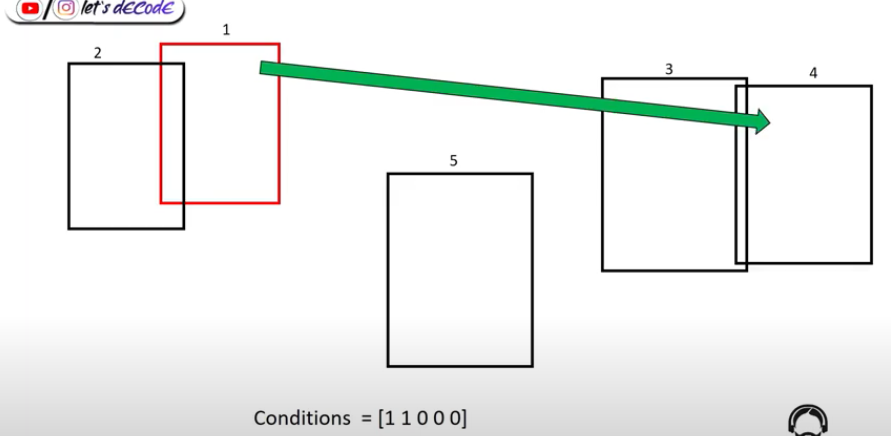
* + - We find the distances between one and two as they are close we increase the elements corresponding to those positions. Here we increase first and second element by one

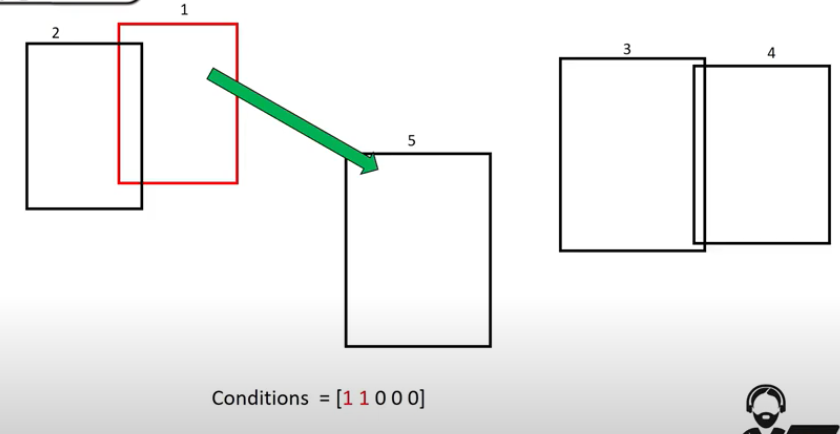


* + - Now we find the distance between one and three as they are far we don’t make any changes to conditions matrix.



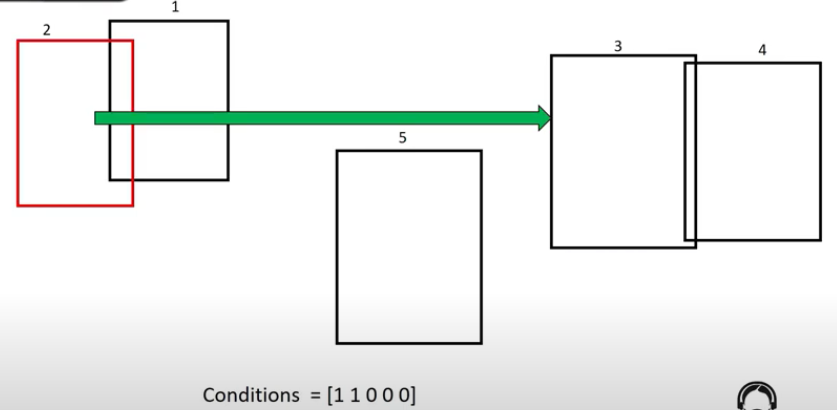
* + - Similarly we find 4 and 5 also



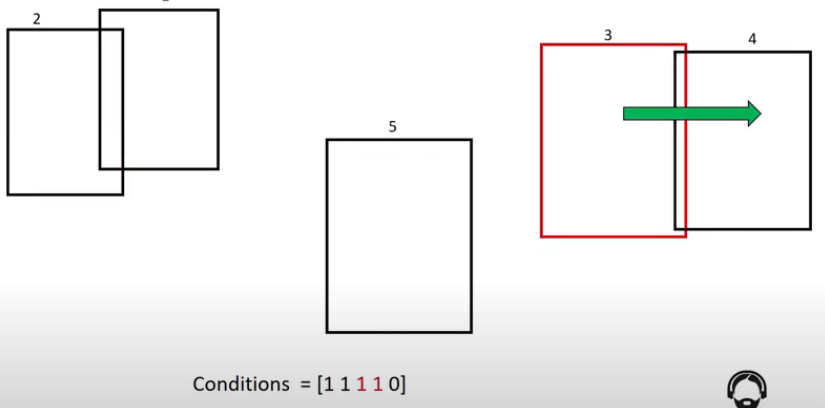


As they also far from 1 no changes make to the condition matrix.

* + - 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.



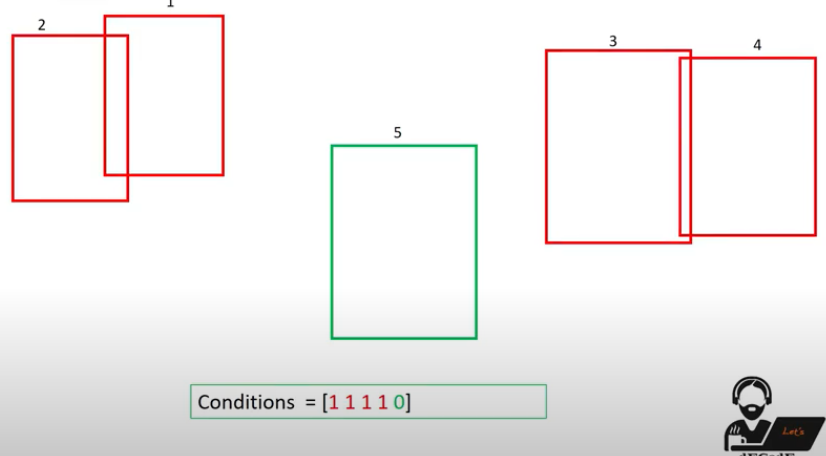
* + - Similarly 2 to 4 & 5 also
    - After that we check 3 with respect to 4 and 5



* + - 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.

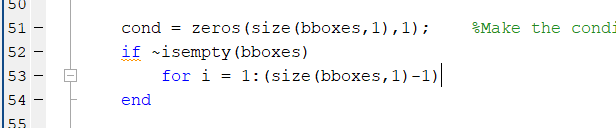


Code:

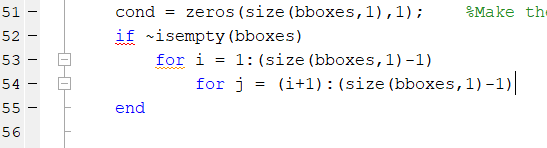
1. Make the condition matrix with size equal to the number of bboxes detected



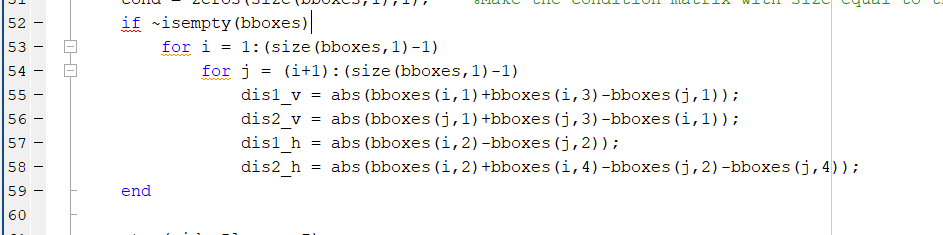
1. Create loop to check for every person with others. Here we need two loops
   * First one gives the reference person



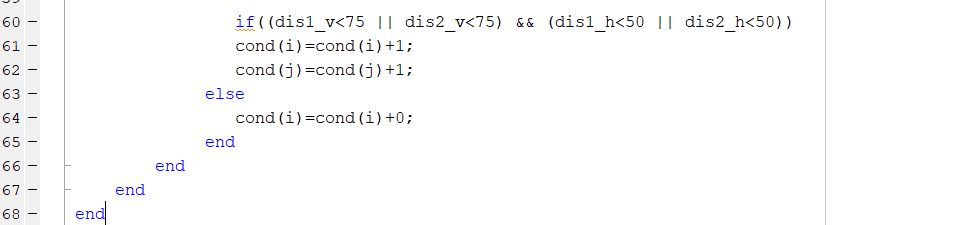
* + Second loop helps us to move to check with every other person



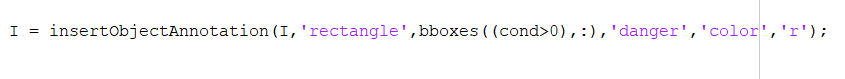
* + Now we find the distance between the ith and jth person as we did before



* + Now 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



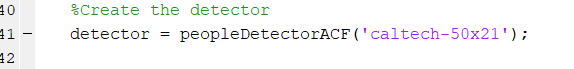
* + Then draw the boxes around them. If the element is greater than zero then we put red color around it along with label danger



* + If the element is zero then we put green color around it along with label safe



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

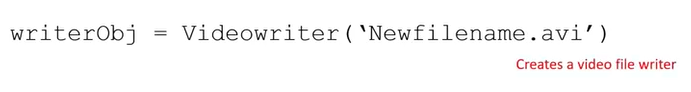




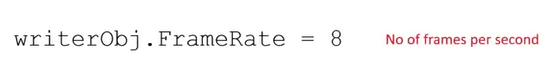
**Save this frames as video**

Here some useful functions to save video

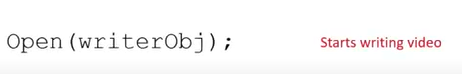
1. **Videowriter(‘Newfilename.avi’)** creates new video file writer



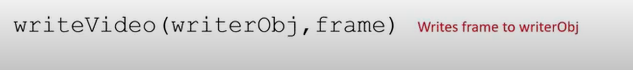
1. **writerObj** frame rate sets the frame rate of the video



1. Open(writerObj) starts writing video



1. writeVideo (writerObj, frame) adds the frame to the writer object

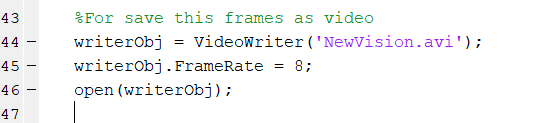


1. Close(writerObj) stops writing video

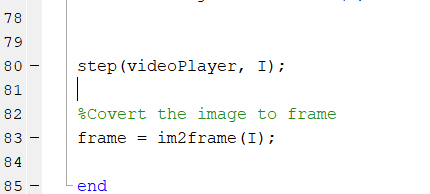


Code:

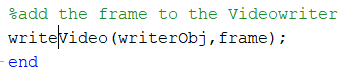
1. Before while loop



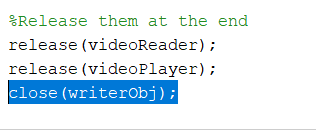
1. Convert the image to frame



1. Then add the frame to the Videowriter



1. 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.

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.

In future we can find the way for remove these errors.