**PROBLEM**

To detect the speed of the pedestrian who is crossing diagonally approaching the camera

**SOLUTION**

**REQUIREMENT**

* OpenCV
* Time

**PROCEDURE**

**OBJECT IDENTIFICATION:**

* Detection of the object in motion using grayscale convertion guassianblur , threshold, dilate which are the functions of OpenCV.
* Then using contour function which finds the object in motion and draws a line over the moving object
* Using the boundingrect() function I can draw a rectangle contour over the moving object

**DISTANCE CALCULATION:**

* I had an idea of using triangle similarity to calculate the distance of the object but we need input such as focal length, pixel , width

**D= (F\*W)/P**

* Then I chose distance formula using coorinates(x,y)
* For that I need two points i.e initial position of the object (x1,y1)and the point where the object moved i.e the final position of the object(x2,y2)
* DISTANCE FORMULA = sqrt((x2-x1)^2+(y2-y1)^2) or sqrt((x2-x1)^2+(y2-y1)^2+(z2-z1)^2)
* From the contour moment function we can get the centroid position i.e the midpoint of the rectangle contour(Cx,Cy)
* when the object is detected first I will store the centroid as (x1,y1)
* when the object is detected last I will store the centroid as (x2,y2)
* And z1 is the distance of the camera from the object and taking z2=0 as it is at the origin
* by applying these in the distance formula I would get the distance

**TIME CALCULATION:**

* when the object is detected first I will store that time as start\_time using time.time() function which records the time in seconds
* when the object is detected last I will store that time as end\_time using time.time() function
* The difference of end\_time and start\_time we will get the actual time

**Detected\_time = end\_time - start\_time**

**SPEED CALCULATION:**

* By coverting the distance from mm to feet we will get the distance in feet
* Speed= distance/time using this formula we can find the speed of the object in feets/sec