**Software team task for agv**

-by Priyakant gautam

* There are six task given form which I have attempeted the task 1
* Which is further classified into two sub task

## Subtask 1

* Optical flow(spares optical flow)

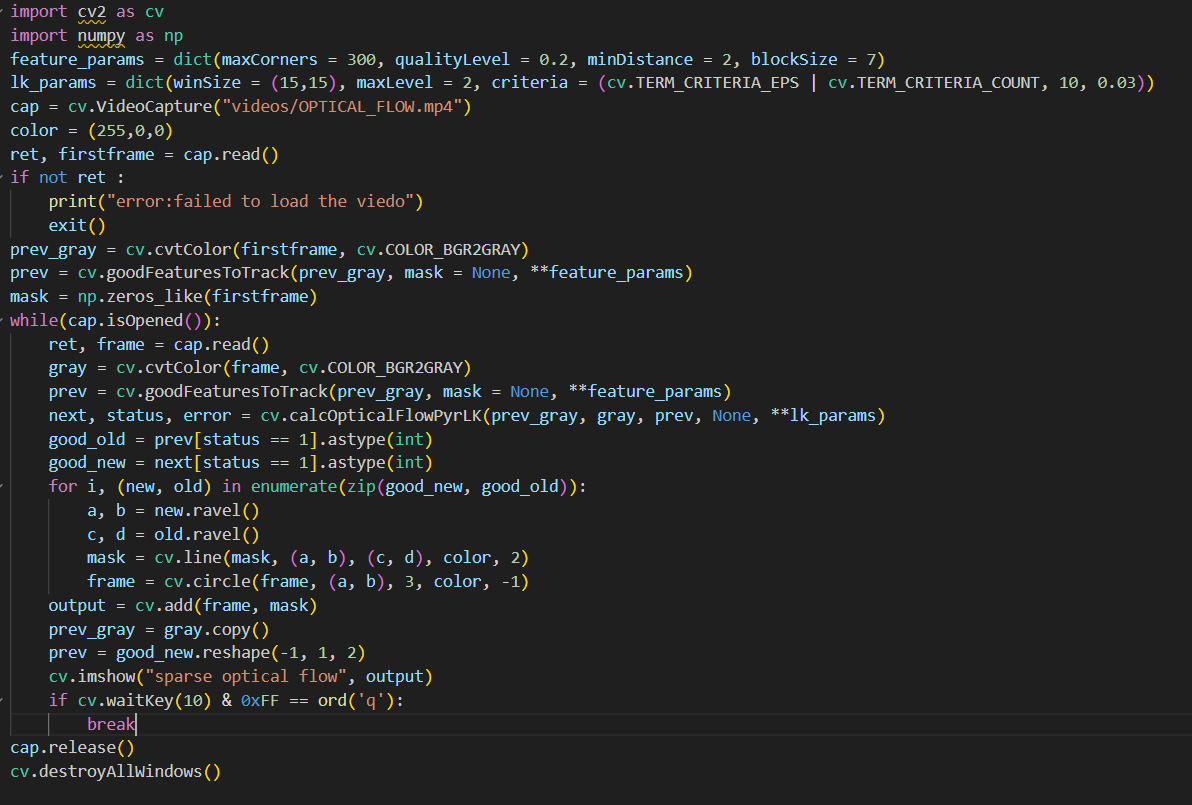
## Motto of this sub taks-:

* In this task we have to detect the optical flow

## Content of task

* We have to detect spares optical flow using lucas cascade method and the basic cv2 function and also numpy
* Firstly,I have go though the given reserch paper fro which I get basic idea of optical flow and the all other method which will we used in this task by this reserch paper I have figure out the code to find spares optical flow this code is given the attached rescerch paper

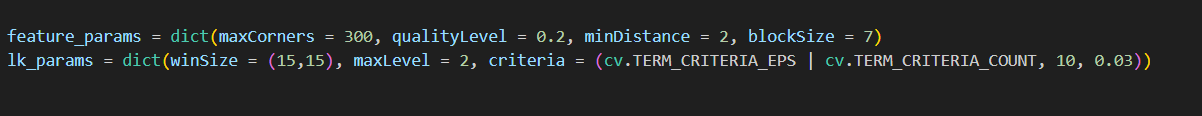
### The first demo code -:



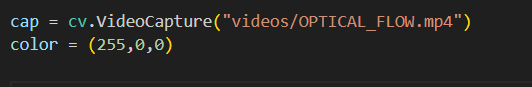
### Now the breakdown of this code-:

### 

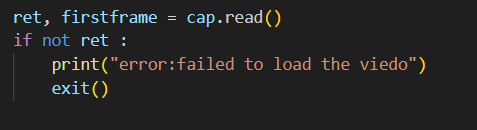
* First we import cv2 as cv and numpy as np

1. 

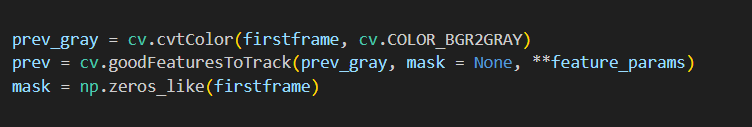
* We make some variable to use them further in the code these variable contains the some features which is required by the function in which we will use them further

1. 

* In this we capture the video frame by frame andv the address of video is(“video/optical\_flow.mp4”)
* And the color is also a variable which we will use further and (255,0,0) this color code is green

1. 

* By this line of code we are just want to clarify that video is successfully ret or not and if not then it will exit the code and print the error:failed to load the video

1. 

* Now in this-:
* Prev\_grey int this we make the video in gray color to reduce its complicity and easy to detect the required pixel
* Prev in this we track the corner of as mentioned condition by feature\_params it is based on shi Tomashi corner detection method   
  now the parameter which is used-:   
  1.prev\_gray-:the gray scale image to detect corner

2. Mask-: no specific region is selected  
 3. max corner-:maximum number of corner to detect

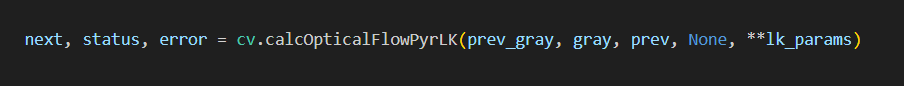
4. Qualiytylevel-:the minimum level of quality to detect the corner  
 5. mindistance-: Minimum distance between detected features.

6. blockSize-:Size of the area for computing corners.

* Np.zeroes\_like make an array of zeroes with same shape and size as firstframe.  
  This mask is later used to draw **optical flow paths** by overlaying lines on the original video frame.

1. 

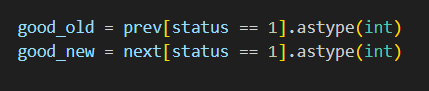
* Cap.isOpened() check the video is successfully opened
* And the while loop ensure that the frames only considered while the video is open

1. 

* This function compute spares optical flow btw two consecutive frames using lucas Kanade methode
* It tracks the movement of detected feature points from the prev\_gray frame to the gray frame.
* Parameter-:

1.prev\_gray=previous gray scale image  
 2.gray=current gray scale image  
 3.prev=the set of point to detect  
 4.lk params

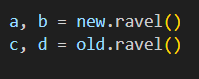
* It return the next frame and statues and error

1. 

* This ensure that only valid tracked point is used for visualition
* Astype(int) just convert the floating point num to interger num
* Good\_old take valid prev tracked point
* Good\_new take valid next tracked point

1. 

* Zip(good\_new,good\_old)-: Pairs up the new and old positions of each successfully tracked feature.
* Enumerate-:add the index(i) which is helpful for the loop
* Zip()-:pair like[(new1,old1), (new2,old2) …….]

1. 

* .ravel()-:convert the multi dimentional array into 1d array
* It convert the new and old array which are 2d array into separate x and y coordinates
* New.ravel() converts [x\_new,y\_new]into two seperates value a=x\_new,b=y\_new
* Similarly for old.ravel()
* The (x,y) coordinates are extracted as individual values to be used in drawing the function.

1. 

* It is used to draw a line from(c,d) to (a,b) of the given color the thickness of line is 2

1. 

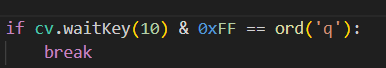
* For each tracked feature, a small red dot id drawn at its new position.
* This visually represent the tracked point on the video.

1. 

* Cv.add() combines two image by adding their pixels value
* Cv.add(frame,mak)merges them,so the motion lines appear on top of the frames in video.

1. 

* Display the video in a window
* Name of video is sparse optical flow

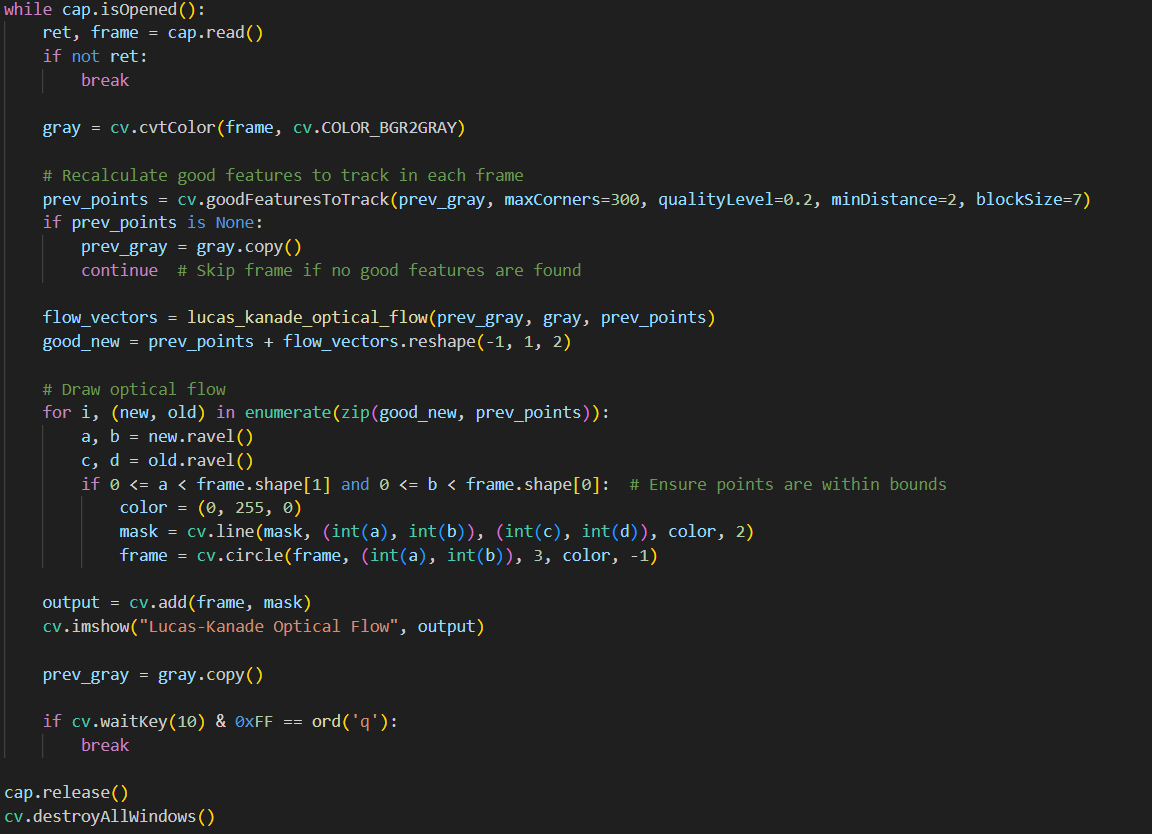
1. 

* It just command when to break the loop and by which key we can break and finish the task

## But in above code we have used the cv.calcOpticalFlowPyLK() but according to the task we have to implement Lucas Kanade methode manually so

### The final code -: first half

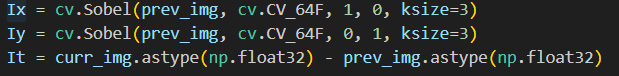
The other half is on other page-:



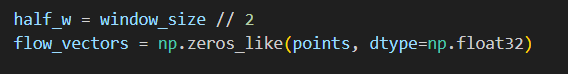
### Key point-:

* Now me use lucas Kanade optical flow manually
* We make afunction to measure lucas Kanade optical flow which is doing same work as cv2.CalcOpticalFlowPyLk() function.

## Breakdown of code-:

1. 

* In this line code we compute gradients
* Ix=compute the gradient of prev\_img in the x direction.
* Iy=compute the gradient in the y direction.
* It=compute the difference btw prev image and current image

1. 

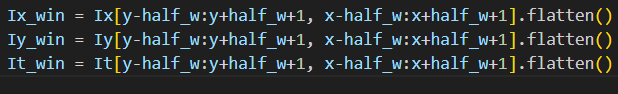
* Half\_w=This calculates half the window size.
* Flow\_vector=creates and array to store the optical flow of each point.

1. 

* The function iterates over each feature point detected in the previous frame.

1. 

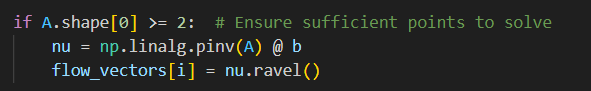
* This condition check if the window around the point extend outside the image boundaries if yes it skip the point processing.

1. 

* These lines extract a small window
* Then flattened into 1d arrayfor easier matrix operation.
* Ix\_win=gradient in the x-direction
* Iy\_win=gradient in the y-direction
* It\_win=change in intensity over time
* This provide the required matrix.

1. 

* This step construct the Ax=b system
* A is matrix where row represent [Ix,Iy] .
* B is a column vector representing negative temporal gradient.
* This equation helps us to determine how the features points are moving.

1. 

* A.shape[0] is the number of rows we need atleast two unknown point
* Np.linalg.pinv(A) solve using pseudoinverse methode
* Flow vector store the value in array

## The remaining code is same as the first just change is we call the function now.

# subtask 2

* I am facing the problem in loading the iam2thor file in docker and not able to complete it.