IMAGE PROCESSING

PYTHON

The project was carried as following:

(Refer to the headings in the python code indicated in capital letters. Also, the attached image was the setup of the project.)

# Importing libraries:

* The numpy library is imported for carrying out and to support all the computations.
* Cv2 is imported for OpenCV.
* Serial is imported to tie it with PySerial. PySerial is a module for accessing the serial port. Arduino IDE takes up input from the serial port. Hence, eventually it interfaces Arduino IDE and Python.

# Baud Rate

It is selected as 9600 i.e. bits sent per second to the serial port. The baud rate of Arduino is also set at 9600.

# Transition time

In case, no reqd. color value is found immediately from the frames, the moment of that color comes out to be zero. For calculating the centroid, the moment is to be used in the denominator, hence an error pops up. The time approximately is the time taken by two frames to appear.

# Color Detection and Contour making

* *Calibration*: For finding out the RGB values of the ball and bot, I first took a screenshot from the video. I then opened the file in the software Photoshop and by clicking on several points for bot and ball; I noted the range of values for both. There’s a feature in the software for the same.
* *Masking*: A separate window(s) with only 2 colors-black and white. All the pixels having RGB value in the specified range would be displayed white and rest black.
* *Drawing Contours and centroid:* The findcontour() function makes a numpy array of all the contours of all the white objects of the mask window. The contour approximation method employed here is ‘simple’. It implies that not all points are stored; rather the minimum number of points needed to draw the contour are stored, e.g. 4 for rectangle, 2 for a line, etc. It saves memory. A variable ‘i’ is initiated which corresponds to the number of white objects found. Thus to eliminate noise, a particular area range is selected viz. determined empirically. Thus, we finally have the required colored patch. The drawcontour() function draws a contour, here- in the window ‘frame’; at the points found by findcontour () function; of green color; and of thickness 2 pixels. The centroid is calculated using the moments’ formula. Once found, a circle is drawn about to locate it.
* *Translation of the bot to catch the ball:* The centroids of the ball and the bot are compared. The signal right, left or nothing were sent to the serial port accordingly which was then read by Arduino. The code written in Arduino IDE was to simply direct which pins to be high or low. The pins were connecting to the L293D motor driver.

ARDUINO

* Two files have been attached namely Arduino Code first and Arduino Code corrected.
* The first one works just fine too. But we observed that the speed of the motors was really less as compared when plugged to a battery supply. The motor drivers were fine too. The reduction of speed was only observed when the motors were run according to image processing.
* We then observed that the camera from which the video was being captured had a frame rate of 100fps. Whereas the Baud rate set for PySerial and Arduino was 9600. This meant that out of 9600 times a second the date was sent by PySerial to Arduino, only 100 of them had the actual information. It therefore unknowingly created PWM with a DC average 1/96th time of the maximum DC output.
* We therefore found out that since we had used ‘if loop’ in the Arduino code, the relevant signal was only passed if there was an incoming byte from PySerial. So, we inserted a nested ‘while loop’ in the ‘if loop’. And things got fine! ☺