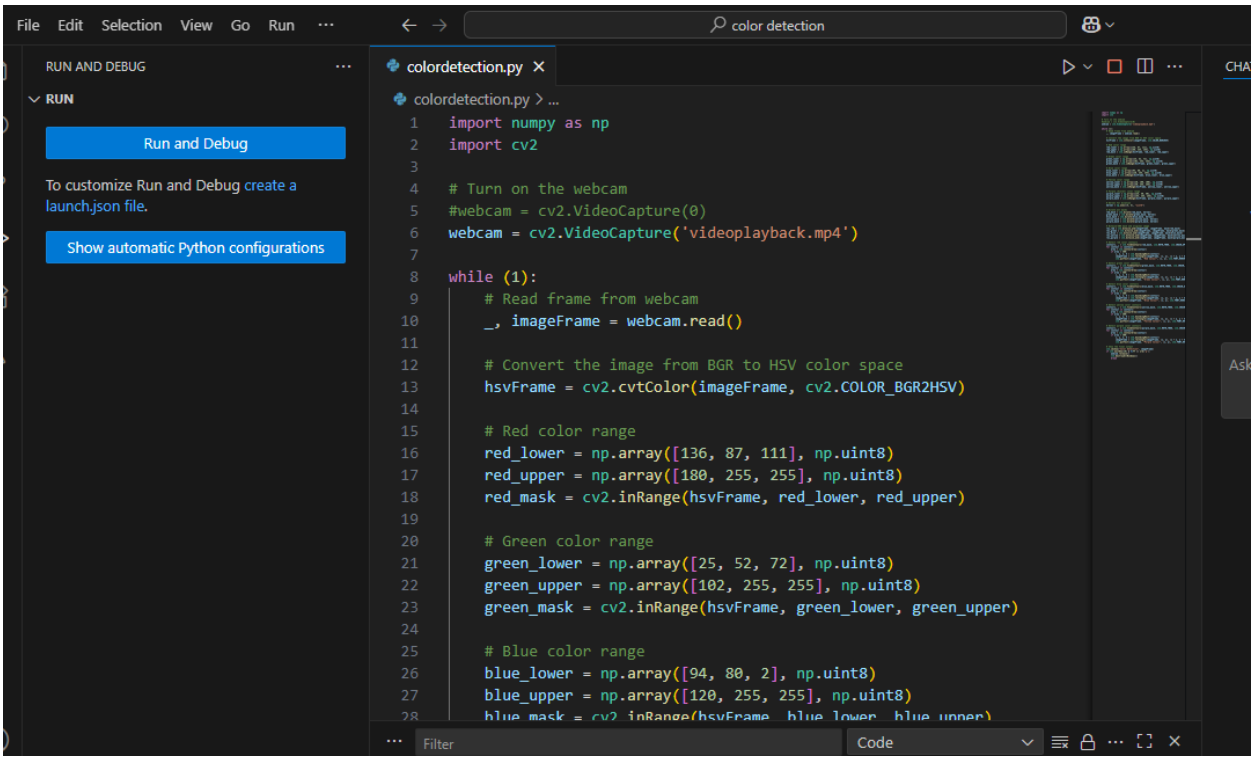
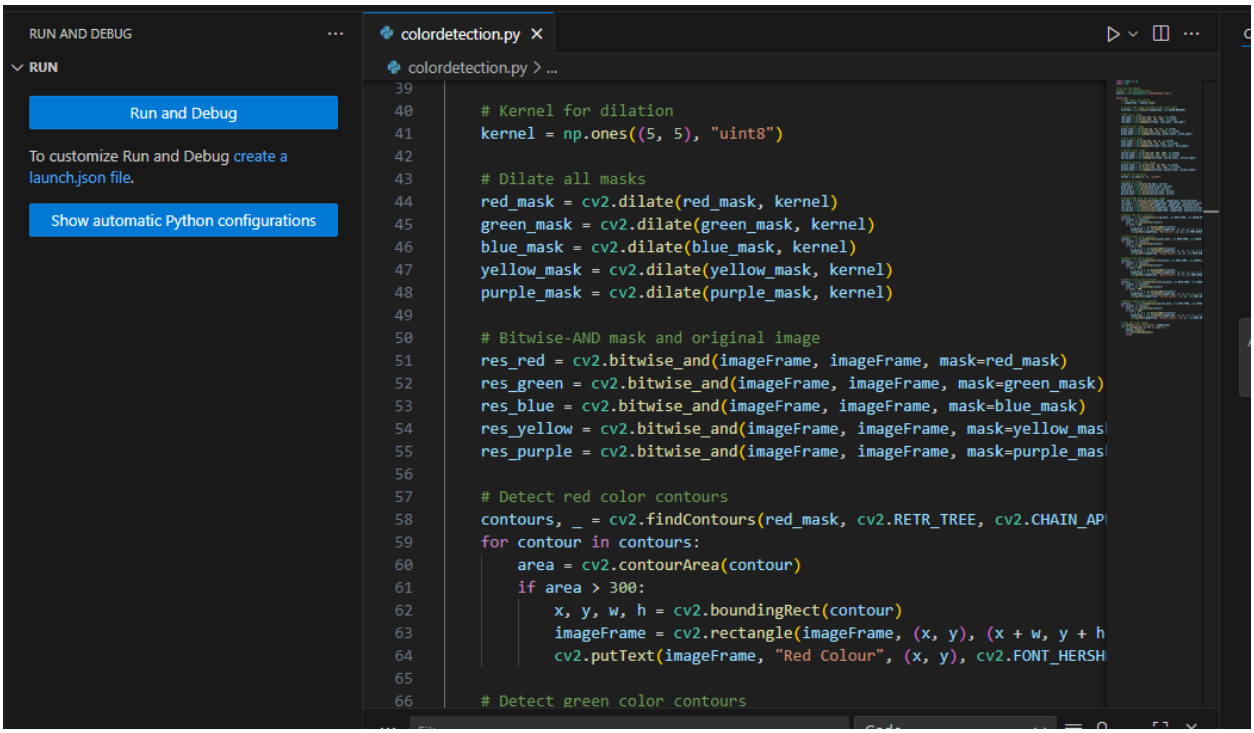
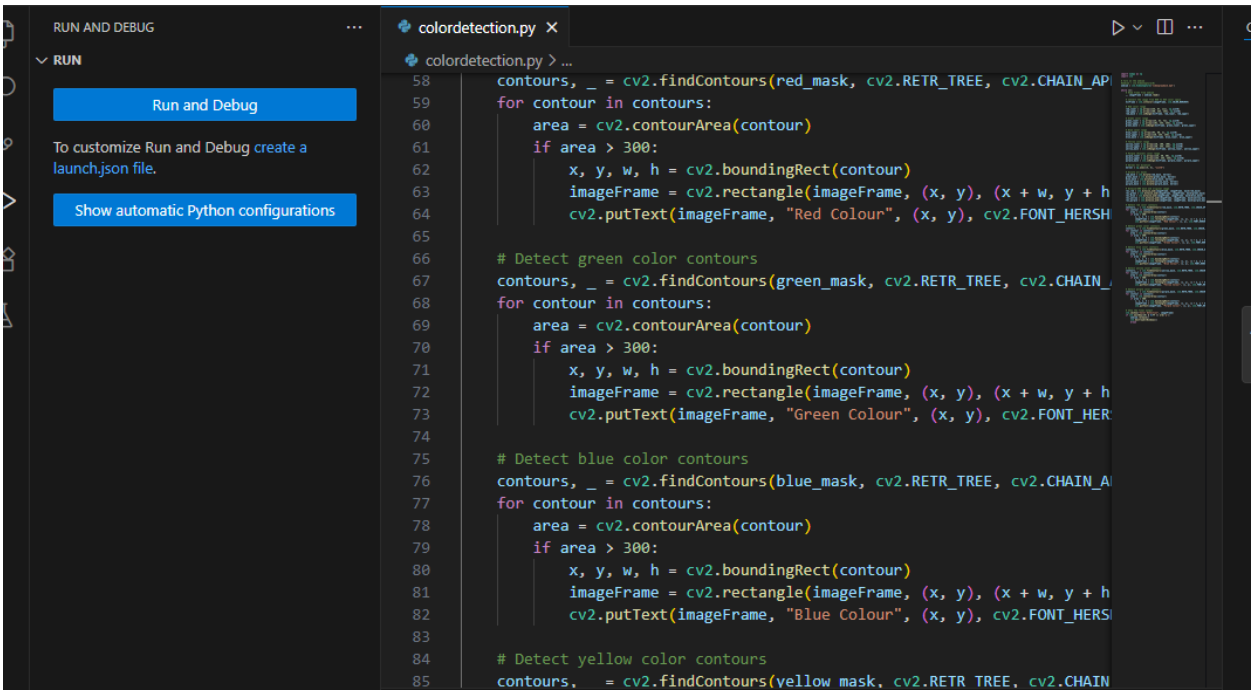
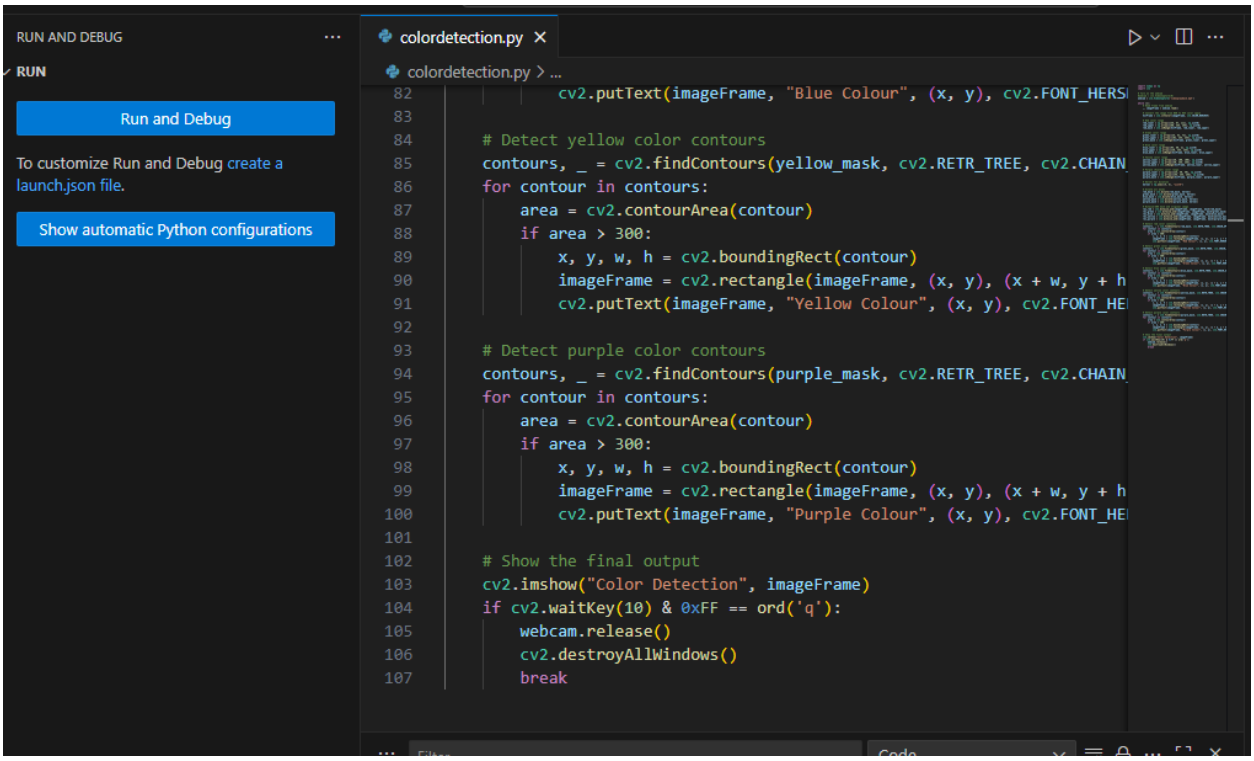
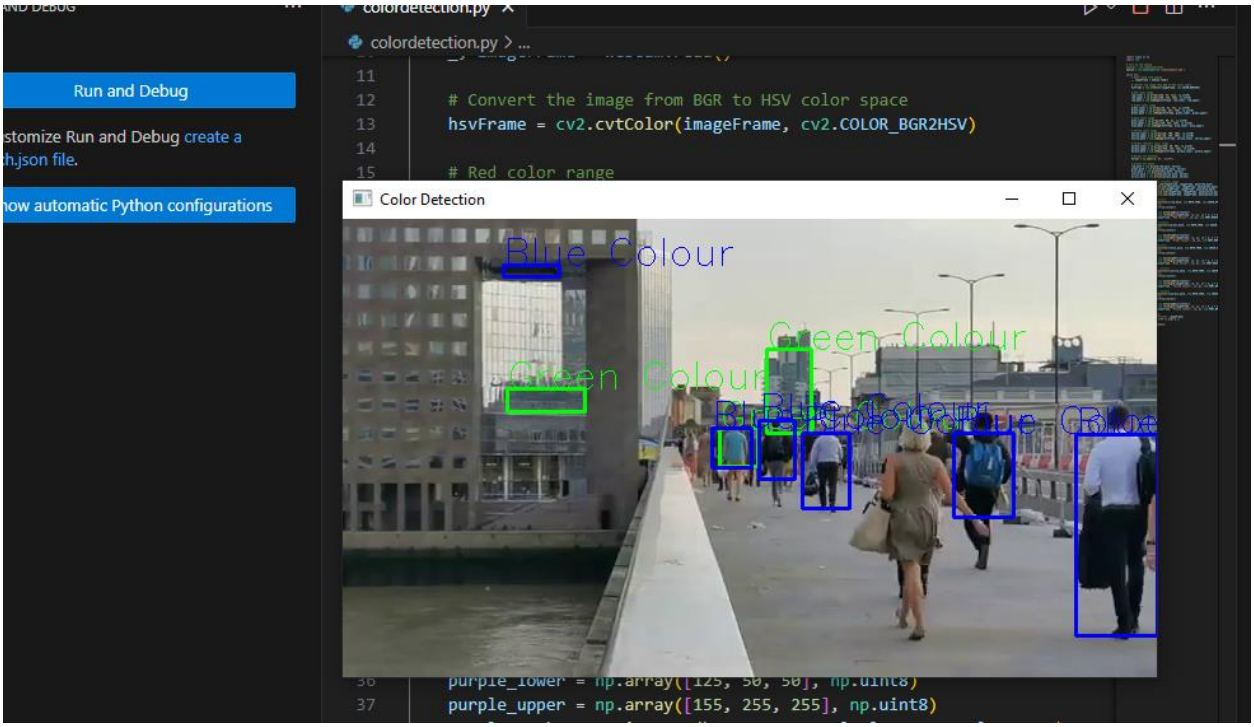

File Edit Selection View Go Run
color detection
RUN AND DEBUG
RUN
Run and Debug
To customize Run and Debug create a launch.json file.
Show automatic Python configurations
colordetection.py
colordetection.py > ...
1 import numpy as np
2 import cv2
3
4 # Turn on the webcam
5 #webcam = cv2.VideoCapture(0)
6 webcam = cv2.VideoCapture('videoplayback.mp4')
7
8 while (1):
9 # Read frame from webcam
10 _, imageFrame = webcam.read()
11
12 # Convert the image from BGR to HSV color space
13 hsvFrame = cv2.cvtColor(imageFrame, cv2.COLOR_BGR2HSV)
14
15 # Red color range
16 red_lower = np.array([136, 87, 111], np.uint8)
17 red_upper = np.array([180, 255, 255], np.uint8)
18 red_mask = cv2.inRange(hsvFrame, red_lower, red_upper)
19
20 # Green color range
21 green_lower = np.array([25, 52, 72], np.uint8)
22 green_upper = np.array([102, 255, 255], np.uint8)
23 green_mask = cv2.inRange(hsvFrame, green_lower, green_upper)
24
25 # Blue color range
26 blue_lower = np.array([94, 80, 2], np.uint8)
27 blue_upper = np.array([120, 255, 255], np.uint8)
Filter
Code

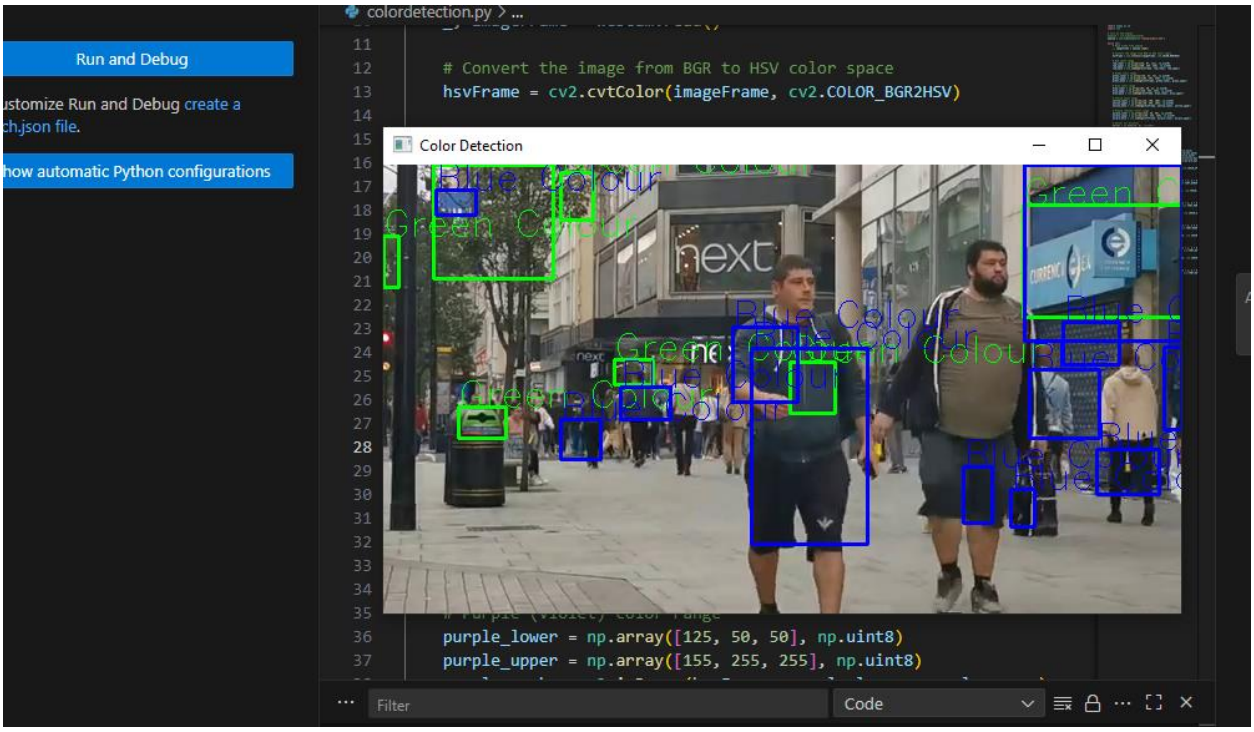

RUN AND DEBUG
RUN
Run and Debug
To customize Run and Debug create a launch.json file.
Show automatic Python configurations
colordetection.py
colordetection.py > ...
39
40 # Kernel for dilation
41 kernel = np.ones((5, 5), "uint8")
42
43 # Dilate all masks
44 red_mask = cv2.dilate(red_mask, kernel)
45 green_mask = cv2.dilate(green_mask, kernel)
46 blue_mask = cv2.dilate(blue_mask, kernel)
47 yellow_mask = cv2.dilate(yellow_mask, kernel)
48 purple_mask = cv2.dilate(purple_mask, kernel)
49
50 # Bitwise-AND mask and original image
51 res_red = cv2.bitwise_and(imageFrame, imageFrame, mask=red_mask)
52 res_green = cv2.bitwise_and(imageFrame, imageFrame, mask=green_mask)
53 res_blue = cv2.bitwise_and(imageFrame, imageFrame, mask=blue_mask)
54 res_yellow = cv2.bitwise_and(imageFrame, imageFrame, mask=yellow_mas
55 res_purple = cv2.bitwise_and(imageFrame, imageFrame, mask=purple_mas
56
57 # Detect red color contours
58 contours, _ = cv2.findContours(red_mask, cv2.RETR_TREE, cv2.CHAIN_AP
59 for contour in contours:
60 area = cv2.contourArea(contour)
61 if area > 300:
62 x, y, w, h = cv2.boundingRect(contour)
63 imageFrame = cv2.rectangle(imageFrame, (x, y), (x + w, y + h
64 cv2.putText(imageFrame, "Red Colour", (x, y), cv2.FONT_HERSH
65
66 # Detect green color contours


RUN AND DEBUG
RUN
Run and Debug
To customize Run and Debug create a launch.json file.
Show automatic Python configurations
colordetection.py
colordetection.py > ...
contours, _ = cv2.findContours(red_mask, cv2.RETR_TREE, cv2.CHAIN_AP
for contour in contours:
area = cv2.contourArea(contour)
if area > 300:
x, y, w, h = cv2.boundingRect(contour)
imageFrame = cv2.rectangle(imageFrame, (x, y), (x + w, y + h
cv2.putText(imageFrame, "Red Colour", (x, y), cv2.FONT_HERSH
Detect green color contours
contours, _ = cv2.findContours(green_mask, cv2.RETR_TREE, cv2.CHAIN_
for contour in contours:
area = cv2.contourArea(contour)
if area > 300:
x, y, w, h = cv2.boundingRect(contour)
imageFrame = cv2.rectangle(imageFrame, (x, y), (x + w, y + h
cv2.putText(imageFrame, "Green Colour", (x, y), cv2.FONT_HER
Detect blue color contours
contours, _ = cv2.findContours(blue_mask, cv2.RETR_TREE, cv2.CHAIN_A
for contour in contours:
area = cv2.contourArea(contour)
if area > 300:
x, y, w, h = cv2.boundingRect(contour)
imageFrame = cv2.rectangle(imageFrame, (x, y), (x + w, y + h
cv2.putText(imageFrame, "Blue Colour", (x, y), cv2.FONT_HERS
Detect yellow color contours
contours, _ = cv2.findContours(yellow_mask, cv2.RETR_TREE, cv2.CHAIN


RUN AND DEBUG
RUN
Run and Debug
To customize Run and Debug create a launch.json file.
Show automatic Python configurations
colordetection.py
colordetection.py > ...
cv2.putText(imageFrame, "Blue Colour", (x, y), cv2.FONT_HERS
Detect yellow color contours
contours, _ = cv2.findContours(yellow_mask, cv2.RETR_TREE, cv2.CHAIN
for contour in contours:
area = cv2.contourArea(contour)
if area > 300:
x, y, w, h = cv2.boundingRect(contour)
imageFrame = cv2.rectangle(imageFrame, (x, y), (x + w, y + h
cv2.putText(imageFrame, "Yellow Colour", (x, y), cv2.FONT_HE
Detect purple color contours
contours, _ = cv2.findContours(purple_mask, cv2.RETR_TREE, cv2.CHAIN
for contour in contours:
area = cv2.contourArea(contour)
if area > 300:
x, y, w, h = cv2.boundingRect(contour)
imageFrame = cv2.rectangle(imageFrame, (x, y), (x + w, y + h
cv2.putText(imageFrame, "Purple Colour", (x, y), cv2.FONT_HE
# Show the final output
cv2.imshow("Color Detection", imageFrame)
if cv2.waitKey(10) & 0xFF == ord('q'):
webcam.release()
cv2.destroyAllWindows()
break


colordetection.py > ...
nd Debug
d Debug create a
Python configurations
11
12 # Convert the image from BGR to HSV color space
13 hsvFrame = cv2.cvtColor(imageFrame, cv2.COLOR_BGR2HSV)
14
15 # Red color range
Color Detection
Red Colour
Blue Colour
Green Colour
Yellow Colour
purple_lower = np.array([125, 50, 50], np.uint8)


colordetection.py
colordetection.py > ...
Run and Debug
stomize Run and Debug create a
h.json file.
now automatic Python configurations
11
12 # Convert the image from BGR to HSV color space
13 hsvFrame = cv2.cvtColor(imageFrame, cv2.COLOR_BGR2HSV)
14
15 # Red color range
Color Detection
Blue Colour
Green Colour
purple_lower = np.array([125, 50, 50], np.uint8)
37 purple_upper = np.array([155, 255, 255], np.uint8)


Run and Debug
ustomize Run and Debug create a
ch.json file.
how automatic Python configurations
colordetection.py > ...
# Convert the image from BGR to HSV color space
hsvFrame = cv2.cvtColor(imageFrame, cv2.COLOR_BGR2HSV)
Color Detection
Blue Colour
Green Colour
next
purple_lower = np.array([125, 50, 50], np.uint8)
purple_upper = np.array([155, 255, 255], np.uint8)
Filter
Code


Run and Debug
ustomize Run and Debug create a
ch.json file.
Show automatic Python configurations
# Convert the image from BGR to HSV color space
hsvFrame = cv2.cvtColor(imageFrame, cv2.COLOR_BGR2HSV)
Color Detection
Red Colour
Blue Colour
purple_lower = np.array([125, 50, 50], np.uint8)
purple_upper = np.array([155, 255, 255], np.uint8)