PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

IMAGE PROCESSING

ROBOFEST WORKSHOP

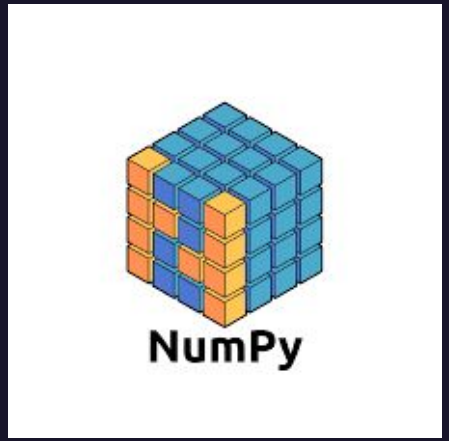
PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

WHAT IS IMAGE PROCESSING?

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter



```
img = cv2.imread(path)
```

```
cap = cv2.VideoCapture(n)
```

```
cv2.imshow("NameOfWindow", img)
```

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

How are colors defined in Image Processing?

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

RGB

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

HSV

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

```
new_img = cv2.cvtColor(img, function)
```

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

DRAWING SHAPES

```
imageFrame = cv2.rectangle(imageFrame, (x1, y1), (x2, y2), color_rgb,
                           thickness)
imageFrame = cv2.circle(imageFrame, (x1, y1), radius, color_rgb,
                        thickness)
```

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

WRITING TEXT

```
cv2.putText(img, text, org, fontFace, fontScale, color,
                                    thickness=...)
```

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

Using Numpy

```
np.array([...], np.data_type)
```

```
np.ones(5, 5)
```

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

MASKING

```
cv2.inRange(img, color_lower, color_upper)
```

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

DILATION

```
cv2.dilate(img, kernel=to_convolute, iterations=no_of_times)
```

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

BITWISE OPERATIONS

```
cv2.bitwise_and(img2, img1, mask)
```

```
cv2.bitwise_or(img2, img1, mask)
```

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

CONTOURS

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

```
contours, hierarchy = cv2.findContours(image, mode, method[, contours[, hierarchy[, offset]]])
```

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

```
area = cv2.contourArea(contour)
```

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

FINDING BOUNDING RECTANGLE

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

```
x, y, w, h = cv2.boundingRect(contour)
```

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

Let’s Begin the Project!

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

```
# Python code for Red Color Detection
import numpy as np
import cv2
# Capturing video through webcam
webcam = cv2.VideoCapture(0)
```

```
# Start a while loop
while(1):

    # Reading the video from the
    # webcam in image frames
    _, imageFrame = webcam.read()
```

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

```
    # Convert the imageFrame in
    # BGR(RGB color space) to
    # HSV(hue-saturation-value)
    # color space
    hsvFrame = cv2.cvtColor(imageFrame, cv2.COLOR_BGR2HSV)
```

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

```
    # Set range for red color and
    # define mask
    red_lower = np.array([136, 87, 111], np.uint8)
    red_upper = np.array([180, 255, 255], np.uint8)
    red_mask = cv2.inRange(hsvFrame, red_lower, red_upper)
    kernel = np.ones((5, 5), "uint8")
    red_mask = cv2.dilate(red_mask, kernel)
```

PES UNIVERSITY

IEEE Robotics & Automation Society
PES UNIVERSITY
Student Chapter

```
# Creating contour to track red color
    contours, hierarchy = cv2.findContours(red_mask, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)

    for pic, contour in enumerate(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), (0, 0, 255), 2)
            cv2.putText(imageFrame, "Red Colour", (x, y), cv2.FONT_HERSHEY_SIMPLEX, 1.0,
                                                                    (0, 0, 255))
```

```
    cv2.imshow("RED COLOR DETECTOR", imageFrame)
    if cv2.waitKey(10) & 0xFF == ord('q'):
        webcam.release()
        cv2.destroyAllWindows()
        break
```

THANK YOU

Reach me in the below Socials:

: linkedin.com/in/jacob-sanoj-b83442182/

: instagram.com/_jacstagram__/

hub.com/jacob-02

anoj23@gmail.com