

**e-Yantra Robotics Competition Plus**

**(eYRC+ Pilot)**

**eYRCPlus-PS2#2526**

|  |  |
| --- | --- |
| **Team leader name** | **Shreyas J kumar** |
| **College** | **RNS Institute of Technology** |
| **e-mail** | eyrc1rn14ec@gmail.com |
| **Date** | **29th Nov ,2015** |

**Image Processing** (8)

Write down the answers to the following questions.

1. What is the resolution(size) of the test image assigned in the task?
2. What is the use of thresholding an image?
   * + 1. (747,900)

🡪width=900pixels

🡪height= 747 pixels

2. We threshold the image to differentiate the foreground from the background to the required level

Explain the algorithm used to perform the task given in practice\_test folder.

1. create reference data file

2. import to the required program

3. convert to grayscale

4. blur

5. dialate

6.threshold

7. locate the numbers

8. put in equivalent responses

9. display

**Software used**  (7)

Write down the answers to the following questions.

1. Write a function in python to open a color image and convert the image into grayscale. You are required to write a function *color\_grayscale(filename,g)* which takes two arguments:
   1. filename: a color image (Test color image is in folder “Task1\_Practice/test\_images”. Pick first image to perform the experiment.)
   2. g: an integer

Output of program should be a grayscale image if g = 1 and a color image otherwise.

**# Import OpenCV & numpy**

**import numpy**

**import cv2**

**def color\_grayscale(filename,g):**

'''

filename-- input color image stored as file(Test color image is in folder

“Task1\_Practice/test\_images”. Pick first image to perform the experiment.)

g-- int 0 or 1

returnsimg-- grayscale of input image if g=1 else color image

'''

#imgt is image matrix that stores input image

**imgt = cv2.imread(filename)**# filename is string passed

**if g==1:** #checks if g is equals to 1 so that its converted to grayscale

**img = cv2.cvtColor(imgt, cv2.COLOR\_BGR2GRAY)**

#img is converted

**else:**

#if g is not equal to 1 then it output matrix is the temporary matrix imgt

**img=imgt**

**return(img)** # gives the caller function the resultant image matrix