e-Yantra Robotics Competition Plus

(eYRC+ Pilot)

<Please enter your team id here>

Team leader name	**JIGER MAHENDRA JAIN**
College	**SARDAR PATEL INSTITUTE OF TECHNOLOGY**
e-mail	**JIGER13@GMAIL.COM**
Date	**8/12/2014**

Scope of the Task (5)

Describe the task assigned.

We were supposed to get the trajectory of bullet and detect which balloon is been targeted
The scope of this task was to do image processing in such a way that we can actually find
The position of gun not only for 1 image but for a set of 10 images without much error
And thus find the balloon which is been targeted!

Camera and Image Processing (8)

Write down the answers to the following questions.

1. What is the resolution (size) of the picture taken from your camera?
2. What is the resolution (size) of the test image assigned in the task?
3. What is the use of thresholding an image?
4. What is the use of color masks?

Answers:

1. The Resolution of the image captured by camera is 480 x 640 pixels
2. The Resolution of test image is 475 x 770 pixels
3. By Thresholding an image we can differentiate between two pixel value either black or white by converting grayscale image into binary image and by setting a threshold we can fetch the part of the image which is required.
4. By Masking technique we can fetch a particular part of image which falls under specified Limits of colour in the code i.e colour to be passed, other all part of the image is masked and thus we can get an image with that specified colour part as white and other part being masked by black colour

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:
    a. filename: a color image (Test color image is in folder "Task1_Practice/test_images". Pick first image to perform the experiment.)
    b. g: an integer

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

```
import numpy as np
import cv2

    def color_grayscale(filename,g):
        '''
        In this function we get 2 values filename which gives the name of the
    file
        and an integer g which used to set which type of image we need
        grayscale or normal
        for that purpose we have used if and else conditional statements
        and thus performed the required task
        '''
        #add your code here
        pic = "test_images/"+str(filename)+".jpg"
        Image = cv2.imread (pic)
        if (g==1):
            img = cv2.cvtColor(Image,cv2.COLOR_BGR2GRAY)
        else :
            img = image
        return(img)

if __name__ == "__main__":
    #checking output for single image
     yoo = color_grayscale(1,0) ## filename 1 and to show grayscale image thus g=1
     cv2.imshow('image1',yoo)
    ##########################################
    ## Close and exit
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

2. Write a function in python to return only the red portions of the image based on the appropriate HSV range.

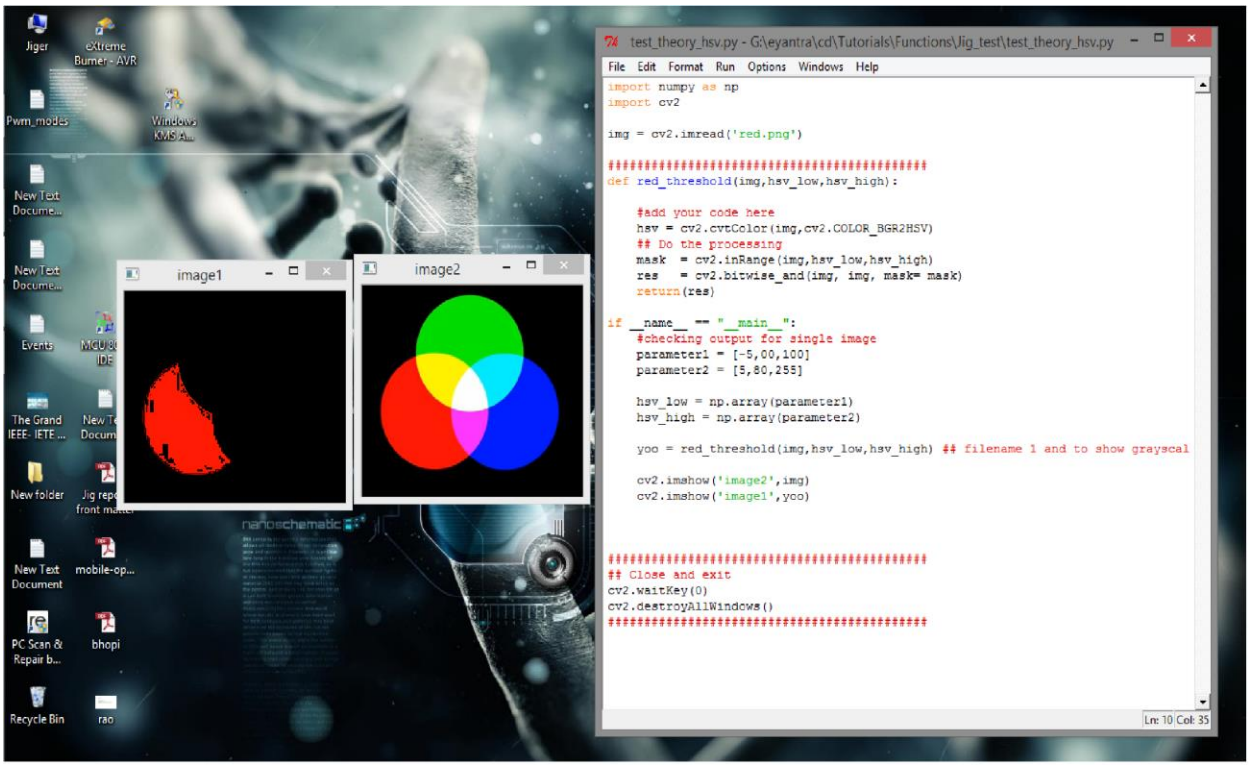
```
import numpy as np
import cv2

img = cv2.imread('red.png')

##########################################
def red_threshold(img,hsv_low,hsv_high):
    '''
      Here in this function we have taken 3 parameters 1st is the image itself
      2nd is the lower value of red hsv parameter and 3rd is the higher value
      Red parameter
'''
    #add your code here
    hsv = cv2.cvtColor(img,cv2.COLOR_BGR2HSV)#converting to hsv
    ## Do the processing
    mask  = cv2.inRange(img,hsv_low,hsv_high)#masking image to take only red
    res   = cv2.bitwise_and(img, img, mask= mask)

    #showing image in original part by taking bitwise AND with original
    #image with mask image and get only red part of original image
    return(res)

if __name__ == "__main__":
    #checking output for single image
    parameter1 = [-5,00,100]  #value of lower red parameter hsv
    parameter2 = [5,80,255]   #value of higher red parameter hsv
    '''
      These values of parameters are found by trial and error
      We can also find by
      #red = np.uint8([[[0,0,255]]]) ##red colour Bgr
      #hsv_red = cv2.cvtColor(green,cv2.COLOR_BGR2HSV)
      #print hsv_red
      We get [0,255,255]
      For red value hsv
      To set parameters we take
      #Low_hsv  =[H-10,100,100]
      #High Hsv =[H+10,255,255]
      This way we set the hsv value
   '''
    ##Convert the parameters into a form that OpenCV can understand
    hsv_low = np.array(parameter1)
    hsv_high = np.array(parameter2)
    yoo = red_threshold(img,hsv_low,hsv_high)
    cv2.imshow('image2',img)
    cv2.imshow('image1',yoo)
## Close and exit
cv2.waitKey(0)
cv2.destroyAllWindows()
##########################################
```



Output only red part of image is taken