MACHINE PERCEPTION ASSIGNNMENT 1 REPORT

BY

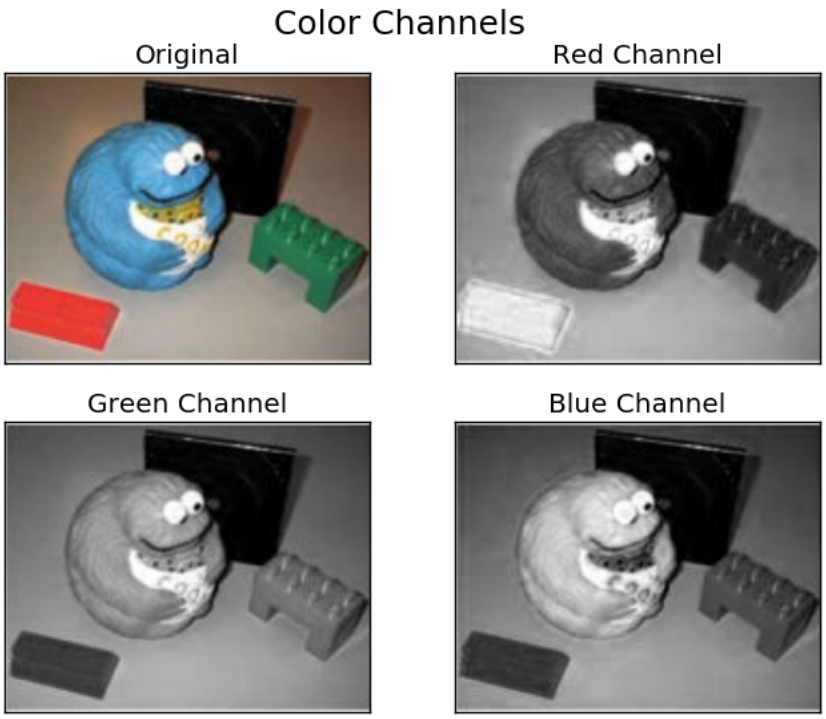
A NEHA (MT2016015)

K DEEPIKA RAJ (MT2016529)

QUESTION 1:

Choose an RGB image (Image1); Plot R, G, and B separately (Write clear comments and observations)

Observations:



Each color image consists of Blue, Green and Red channels, here cv2.split will help us analysing the image with a specific color channel.

In red channel image, red pixels look bright compared to other pixels. Same follows for Green and Blue channels images.

RGB space may be visualised as a cube with the three axis corresponding to red, green and blue.

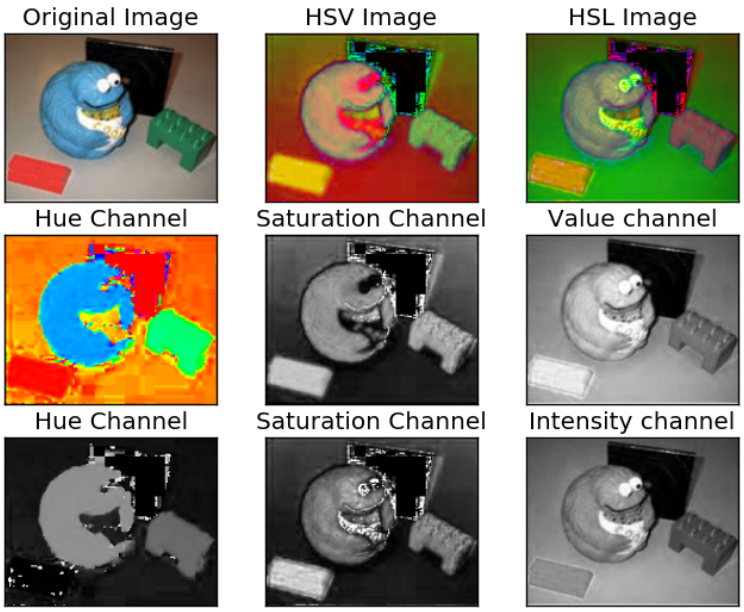
Each pixel of image is represented by 24 bits wherein 8 bits is contributed by each color channel R, G and B. Hence each channel is can have a value from 0 to 255.

QUESTION 2:

Convert Image 1 into HSL and HSV. Write the expressions for computing H, S and V/I.

(Write clear comments and observations)

Observations:



To get Hue Channel from HSV image, use the HSV image and fill Saturation and Value channel to maximum (255) value . In OpenCV, for HSV, Hue range is [0,179], Saturation range is [0,255] and Value range is [0,255].

Cmax = max(*R*', *G*', *B*')

Cmin = min(*R*', *G*', *B*')

Δ = *Cmax* - *Cmin*

$$H = \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left(\frac{G'-B'}{\Delta} mod 6\right) & , C_{max} = R' \\ 60^\circ \times \left(\frac{B'-R'}{\Delta} + 2\right) & , C_{max} = G' \\ 60^\circ \times \left(\frac{R'-G'}{\Delta} + 4\right) & , C_{max} = B' \end{cases}$$

$$S = \begin{cases} 0 & , C_{max} = 0 \\ \frac{\Delta}{C_{max}} & , C_{max} \neq 0 \end{cases}$$

$$V = Cmax$$

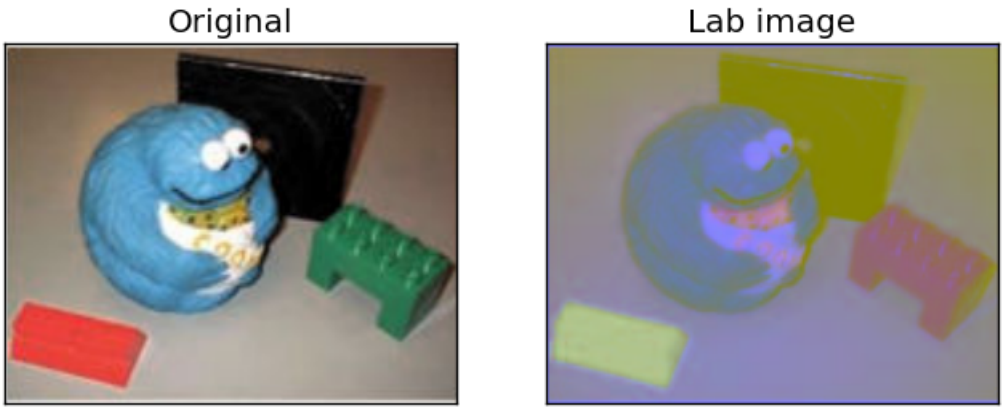
L = (Cmax + Cmin) / 2

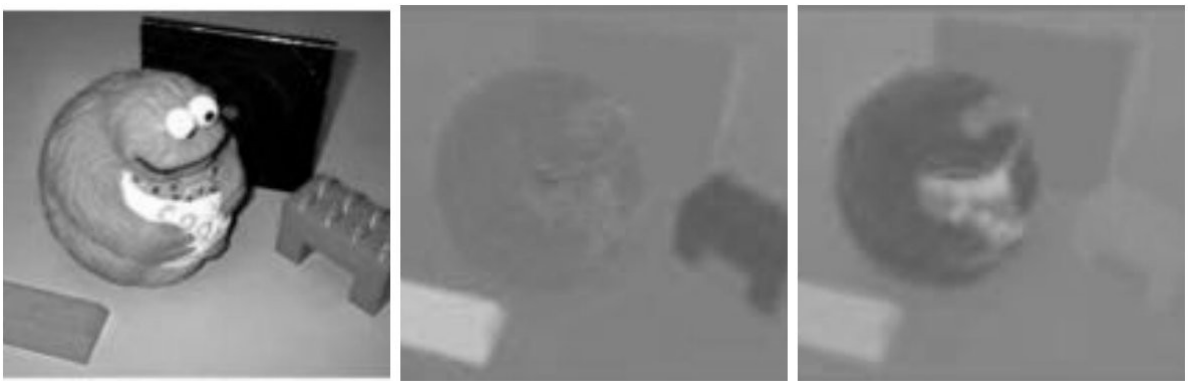
R', *G*', *B*' are normalised values of R, G , B. To compute R, G, B values use cv2.split() funtion.

QUESTION 3:

Convert Image 1 into L*a*b* and plot.

Observations:





Spltting into indiviual channels : L, a, b channels

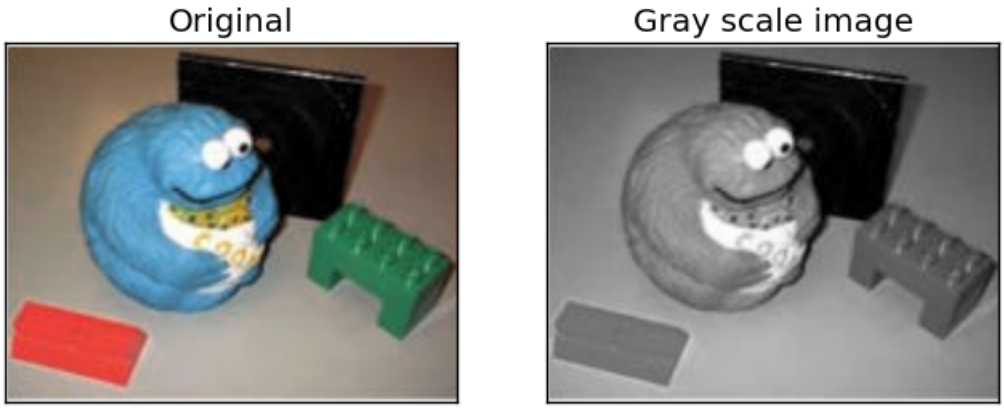
A Lab color space is a color-opponent space with dimensions L for lightness and a and b for the color-opponent dimensions, based on nonlinearly compressed (e.g. CIE XYZ) coordinates.

The three coordinates of CIELAB represent the lightness of the color (L* = 0 yields black and L* = 100 indicates diffuse white; specular white may be higher), its position between red/magenta and green (a*, negative values indicate green while positive values indicate magenta) and its position between yellow and blue (b*, negative values indicate blue and positive values indicate yellow).

QUESTION 4:

Convert Image 1 into Grayscale using the default OpenCV function. Write the expressions used for the conversion.

Observations:



Expression for converting BGR image to gray scale image is :

RGB to Gray: 0.299 * R+0.58 * G+0.114 * B

Default OpenCV function for conversion :

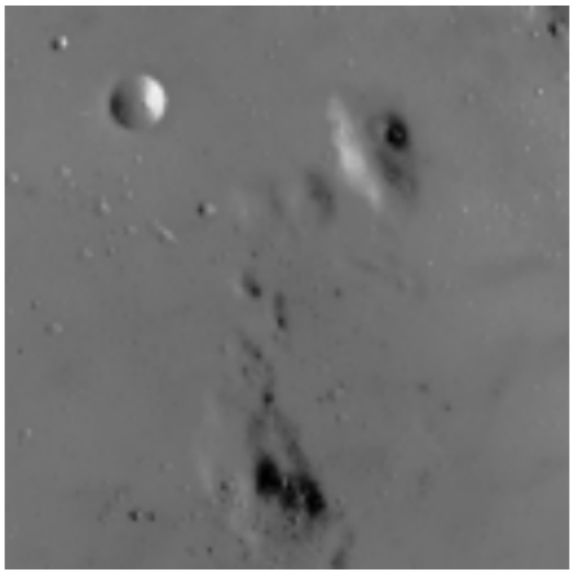
cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

QUESTION 5:

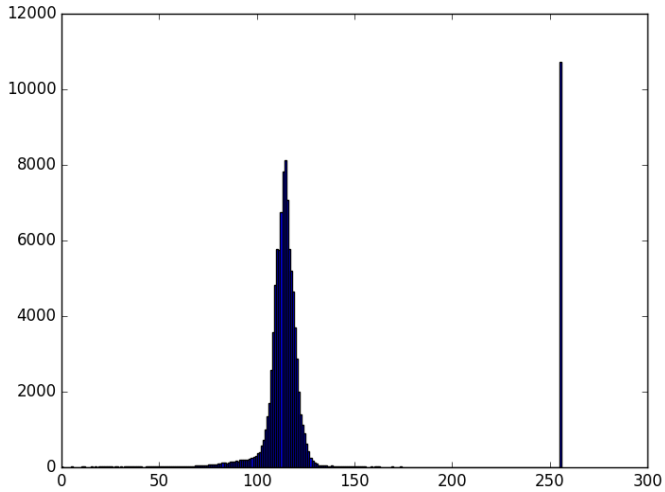
Take a grayscale image (Image 3) and illustrate

1) Whitening
2) Histogram equalization

Observations:

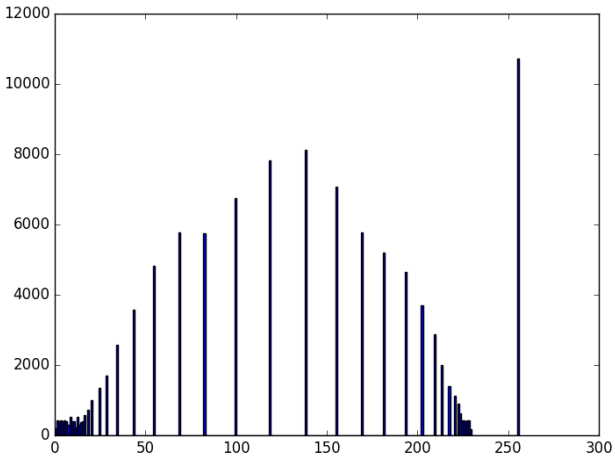


Input Image

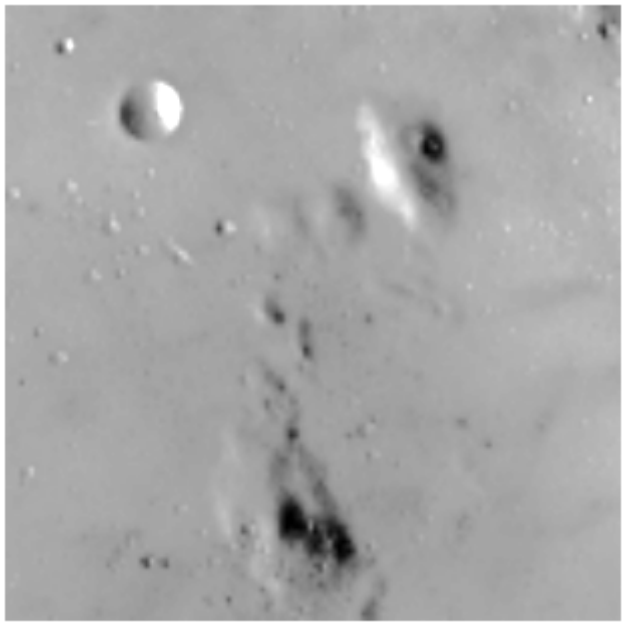


Histogram of the input image

Output after histogram equalisation



Histogram of output image



Output after Whitening

Depending upon the histogram of the image, the histogram is streched to either ends so that the image will have pixels from all regions of the image.

Unlike the input image, the output image obtained by Histogram equalisation has uniform distribution intensities

Expression for whitening a image :

Output_Pixels = (Input_pixels – Mean) / Standard Deviation

where Output_Pixels coresspond to the whitened image and Input_pixels coresspond to input image.

Mean and Standard Deviation are calculated using numpy.

QUESTION 6:

Take a low illumination noisy image (Image 4), and perform Gaussian smoothing at different scales. What do you observe w.r.t scale variation?

Observations:

Convolution with low pass filter removes noise which are high frequency compents. Guassian function which creates a distribution of values around the center point. This results in a kernel in which pixels near the center contribute more towards the new pixel value than those further away.

By varying the kernel size we can determine how many pixels to sample during the convolution and the sigma will define how much to modulate them by.

By increasing the SD of the kernel reduces the amplitude substantially, in other words by increasing the standard deviation we increasing the burring radius.

QUESTION 7:

Take an image and add salt-and-pepper noise. Then perform median filtering to remove this noise.

Observations:

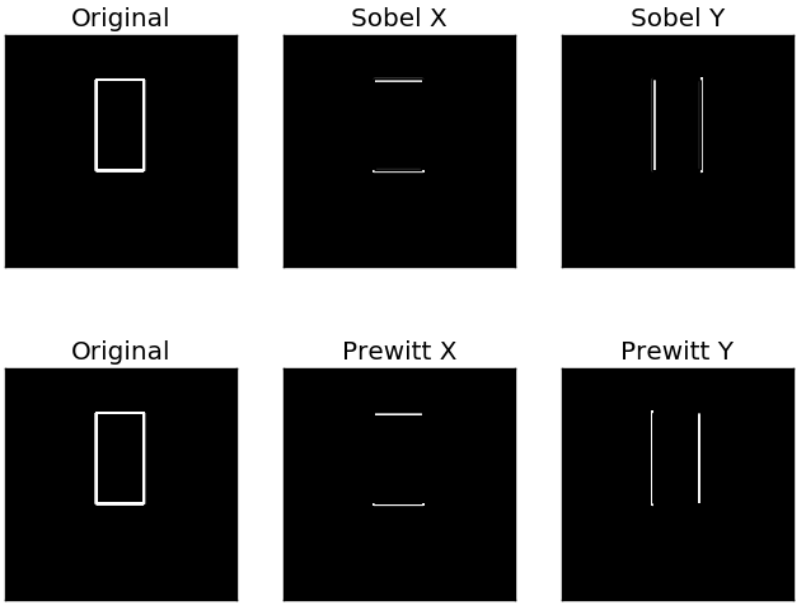
Median filtering removes high freqeuncy componets, by replacing the high pixel value with the median of the neighbouring pixels.

By increasing the kernel size, we can achieve better filtering.

QUESTION 8:

Create binary synthetic images to illustrate the effect of Prewitt (both vertical and horizontal) plus Sobel operators (both vertical and horizontal)

Observations:



Sobel and Prewitt operators are used for edge detection ass they are high pass filters.
Sobel and Prewitt operate in the same way except that Sobel is not isotropic in its response, in other words Sobel is baised towards particular set of directions.

Sobel operator kernels :

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Prewitt operator kernels :

-1	0	+1
-1	0	+1
-1	0	+1

Gx

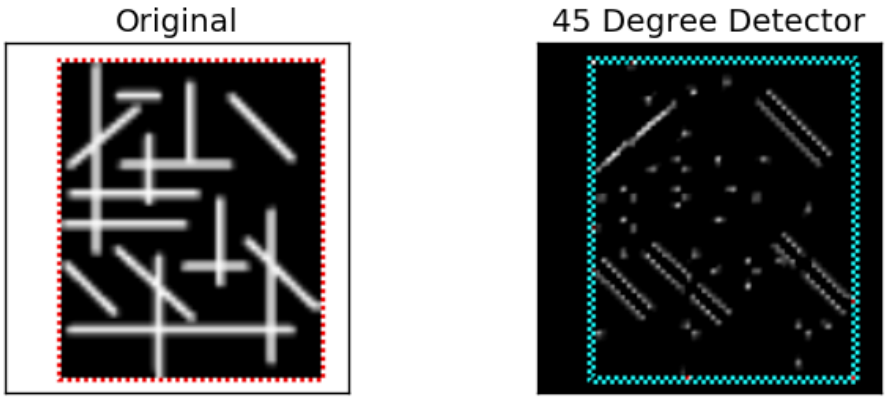
+1	+1	+1
0	0	0
-1	-1	-1

Gy

QUESTION 9:

What filter will you use to detect a strip of 45 degrees

Observations:



A convolution based technique which produces an image description of the thin lines in an input image is utilised to detect a strip of 45 degrees.

Kernel for convolution :

-1	-1	2
-1	2	-1
2	-1	-1

QUESTION 10:

Take an image and observe the effect of Laplacian filtering

Can you show edge sharpening using Laplacian edges?

Observations:

Laplacian operator is senstive to noise, hence we perform Guassian blurring before convoluting with laplacian kernel.

Kernels used for convolution is :

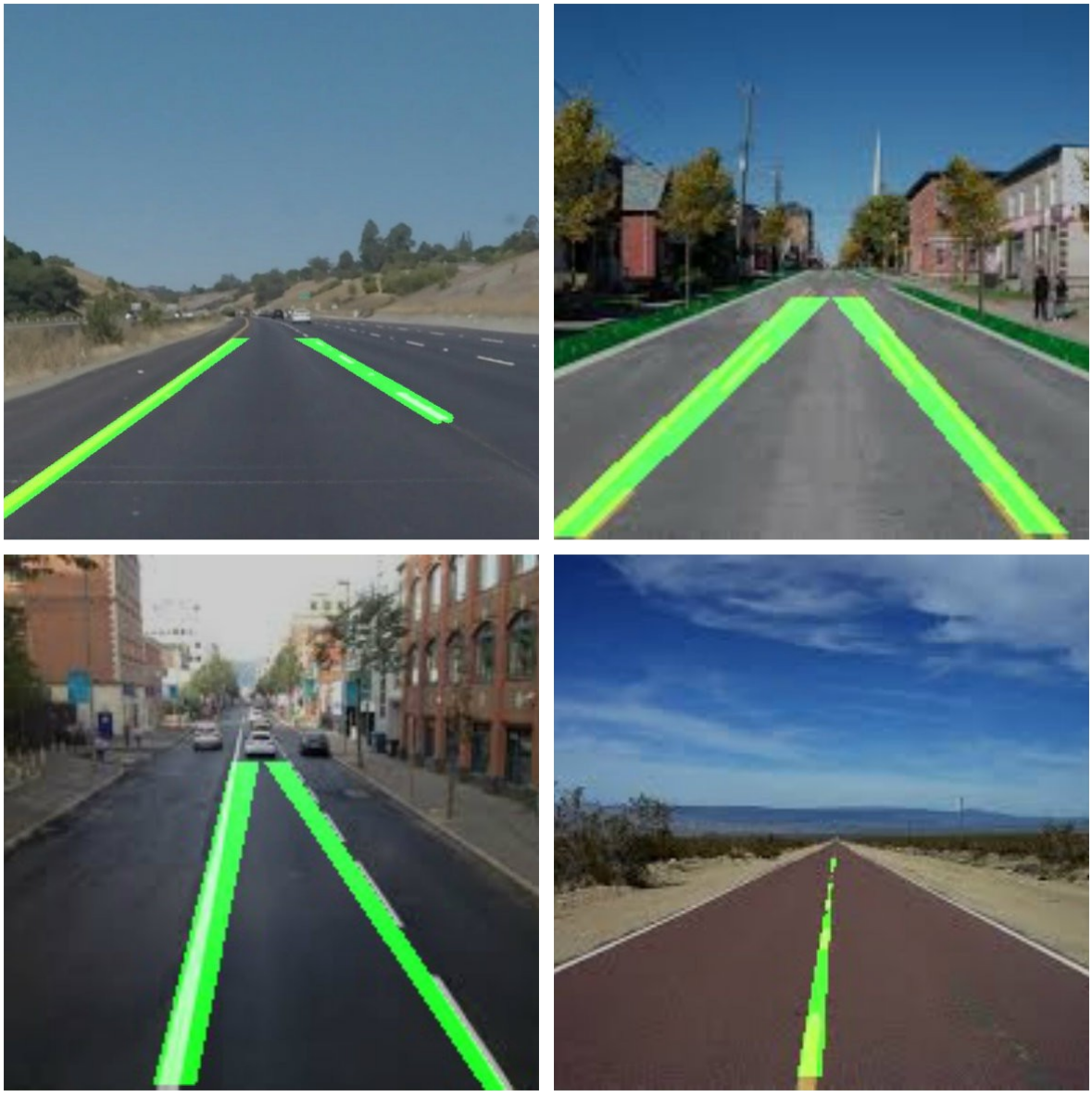
0	−1	0
−1	4	−1
0	−1	0

For better results, a single kernel named LOG (Laplcian of Gaussian) can be used.

QUESTION 11:

Detect Road land markers

Observations:



To start with we performed Guassian Blurring to remove noise. For detecting road lanes mere Canny Edge Detector was not useful as output of Canny edge detector was noisy, i.e. it contained multiple edge fragments corresponding to a single whole feature. So after Canny Edge Detection the image was subjected to Probablisitic Hough Transform which focuses not only what features are but also how many of them exist.

Probalisitic Hough Tranform was useful in detecing incomplete and discontinuos edges also.

QUESTION 12:

Classify modes: Night; Portrait; Landscape

Design features, use NN

Observations:

To start with we read all images (59) from the folder and assigned class label to each image.

Landscape : Class 1
Night : Class 2
Potrait : Class 3

Feature Description :

Image Descriptor : Raw pixel descriptor
Output : 1 D Array of numbers corresponding to the raw RGB pixel intensities of images.

Training of data :

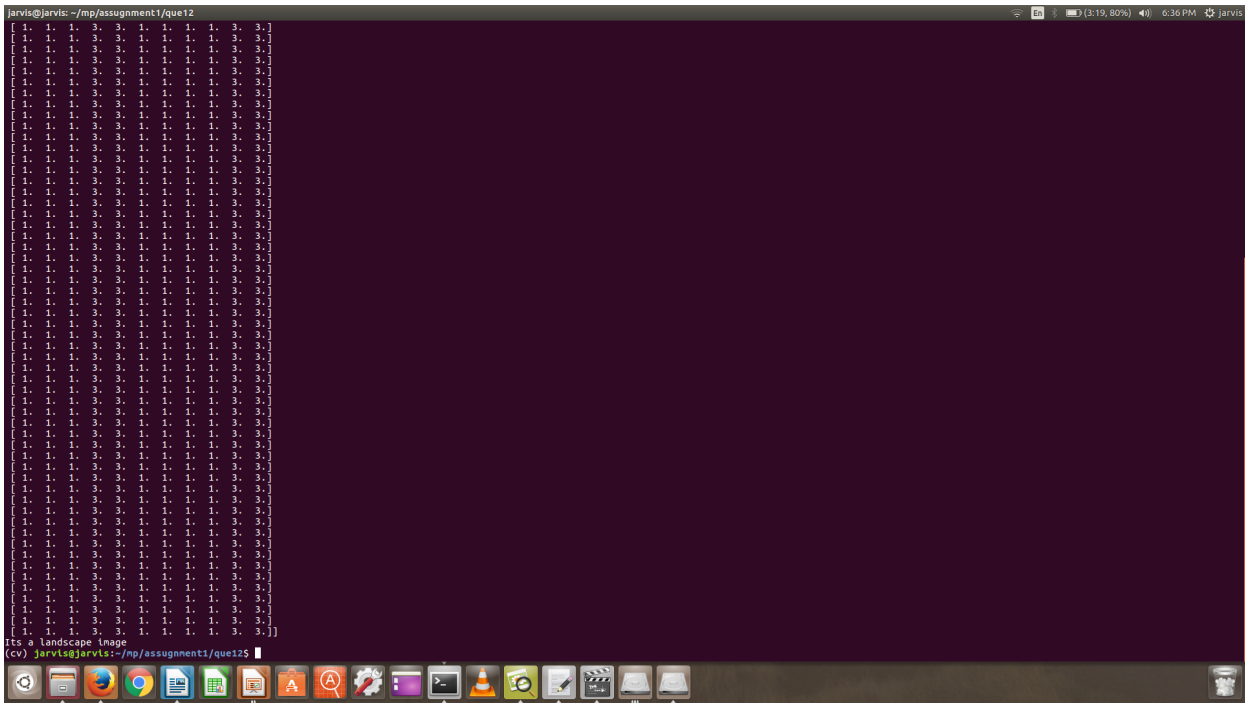
We have utilsed the function cv2.ml.Knearest_create() to implement K Nearest Neighbour Algorithm wherein k = 11.

Prediction :

The results are analysed based on the maximum no of votes from a class.
If in the list of neighbours there are more from a praticular class, the image is classified in to that particular class.
We made sure that all the training as well as test images are of float32 data type.

Test cases are given below

test_case1.jpg

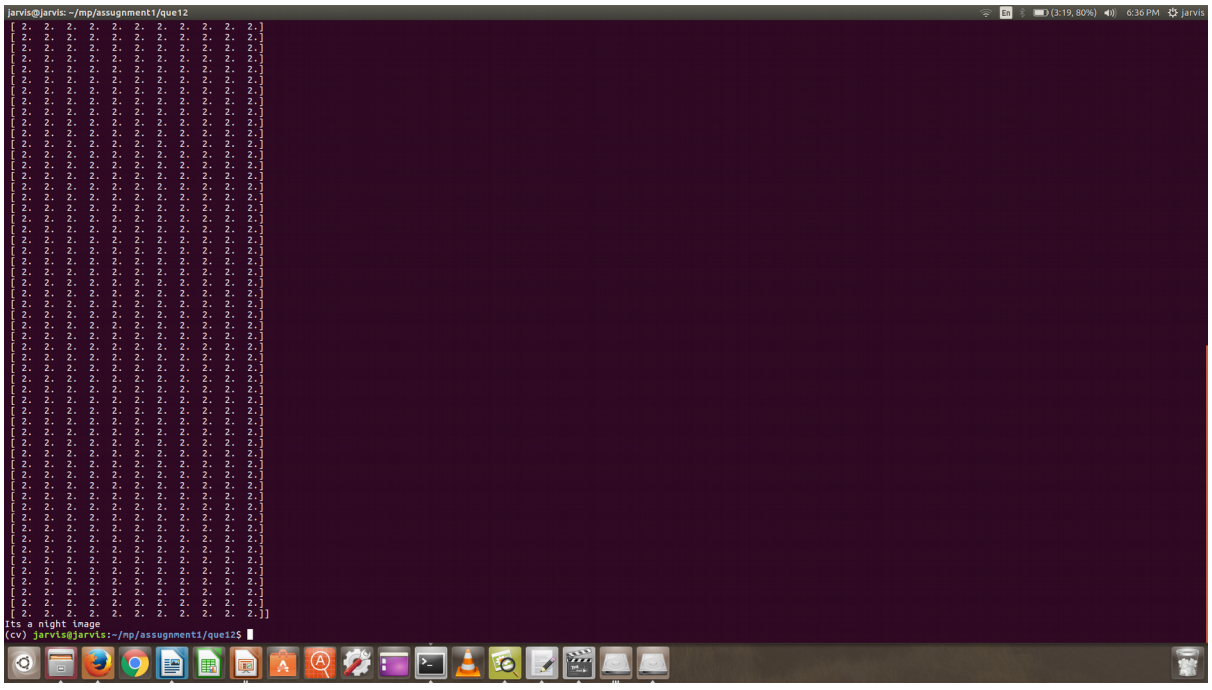


Corresponding output for the input image : It is a landscape image

In test_case1:

From the terminal output we can analyse that most neighbours are from Class 1
Hence KNN decides to categorise test_case1 as Landscape.

test_case2.jpg

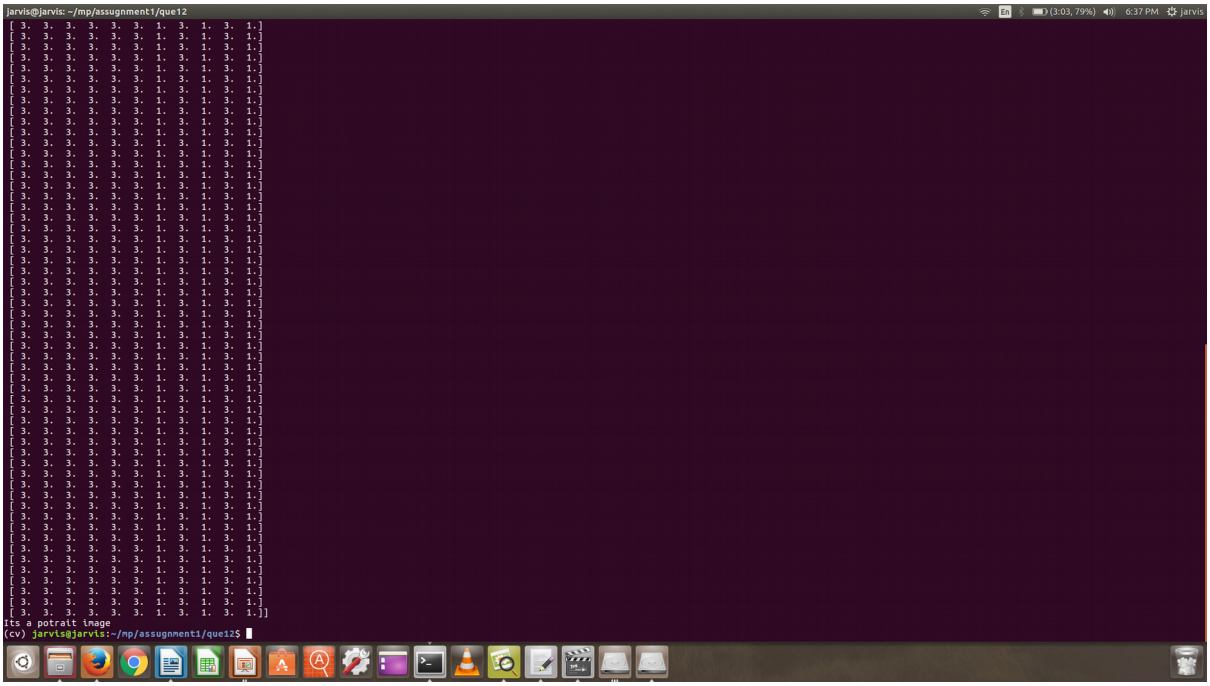


Corresponding output for the input image : It is a night image

In test_case2:

From the terminal output we can analyse that most neighbours are from Class 2
Hence KNN decides to categorise test_case2 as Night.

test_case3.jpg



Corresponding output for the input image : It is a potrait image

In test_case3:

From the terminal output we can analyse that most neighbours are from Class 3
Hence KNN decides to categorise test_case3 as Potrait.