INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY, BANGALORE

ASSIGNMENT 1
DS/NC/ESD 863 Machine Perception

Image Analysis

Anoop Toffy - MT2016016
Athul Suresh - MT2016030

Instructor :
Dr. Dinesh Babu

February 1, 2017

Contents

List of Figures

List of Tables

1 Starters 1

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

1.1 Observations

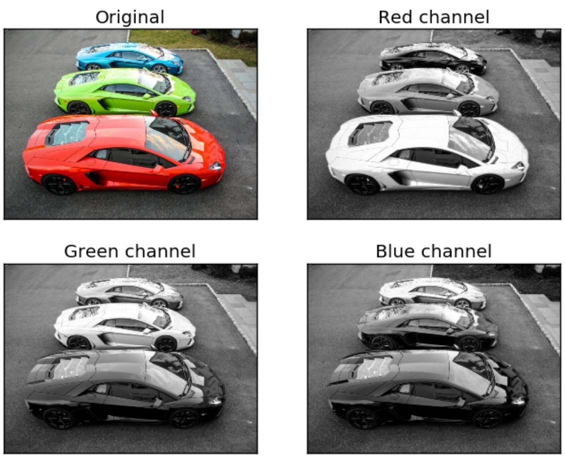


Figure 1: Extracting R,G,B channels with gray color maps

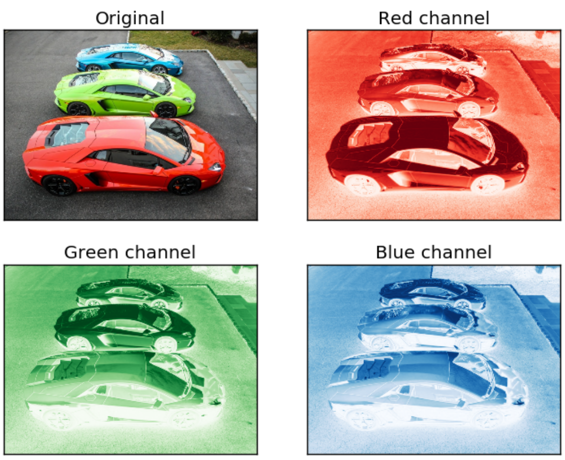


Figure 2: Extracting R,G,B channels with corresponding color maps

1.2 Inference

Each image is represented in python as an $n*m*c$ nd-array where $n*m$ is the resolution of the image and c is the number of channels. On splitting the channels we get 3 $n*m$ images. Each pixel in the image has an intensity value ranging from 0 to 255 showing the intensity of the color in that particular(eg. Red) channel. Combination of the basic R, G, B colors produces all the other different colors.If a particular image contains 3 channels, it is interpreted as (R,G,B) by matplotlib. If there is only a single channel, then we specify a color map which is basically a mapping of pixel values to colors. In Figure 1, gray scale color map is used where 0 corresponds to black or absence of that particular color and 255 corresponds to white or where the color is present in full intensity. In order to make the representation more intuitive, Figure 2 uses a different color map for each channel where white represents the absence of that color (pixel value = 0) and dark shade of the corresponding color represents it's maximum intensity (pixel value = 255).
The picture below represents various color maps

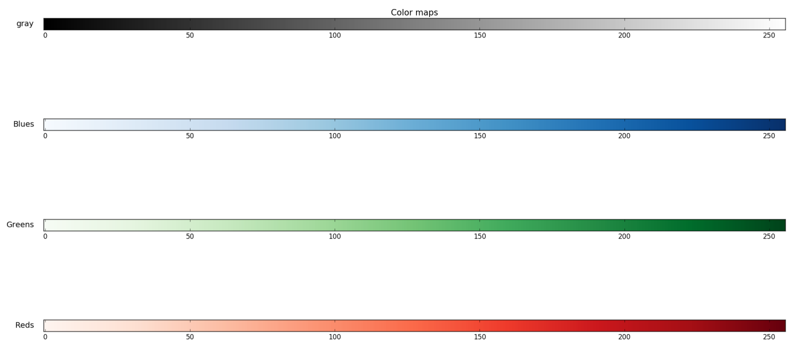


Figure 3: Pixel Values

2 Starters 2

Convert Image into HSL and HSV. Write the expressions for computing H, S and V/L ? (Write clear comments and observations)

2.1 Observations

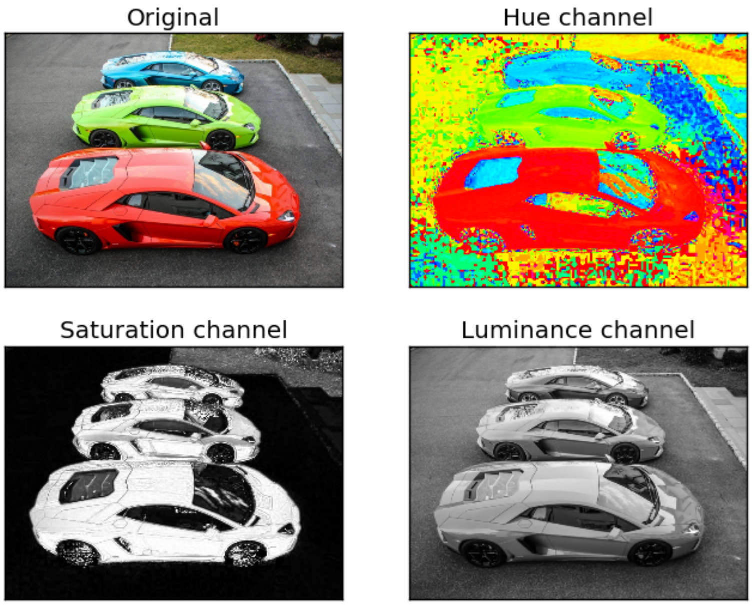


Figure 4: Pixel Values

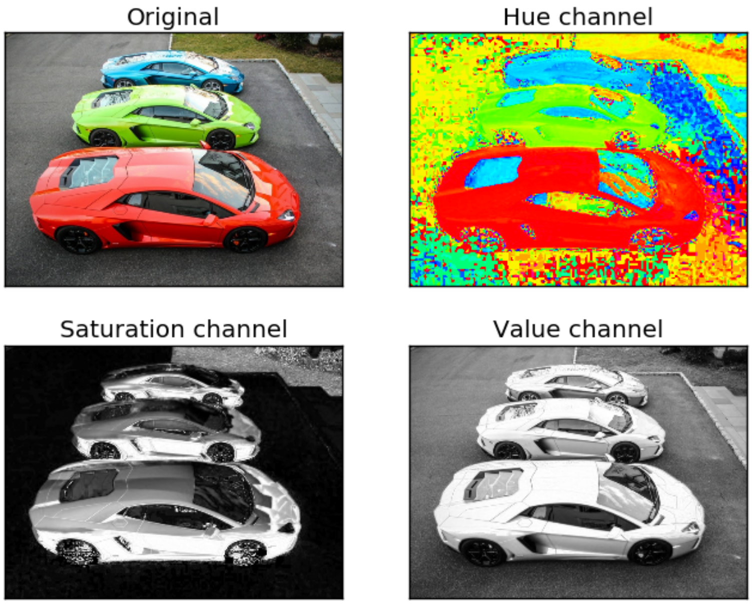


Figure 5: Pixel Values

2.2 Inference

2.2.1 Converting RGB to HSV/ L

The individual components of the HSV and HSL can be visualized with the help of the following figures. [1]

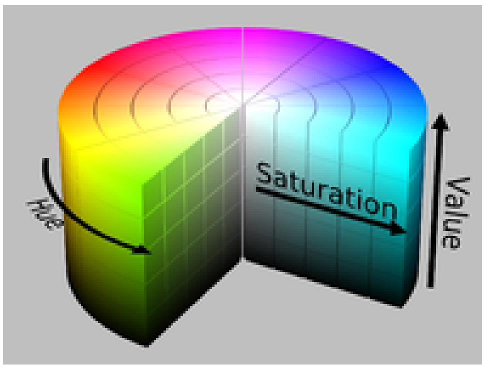


Figure 6: HSV Cylinder

[1] https://en.wikipedia.org/wiki/HSL_and_HSV

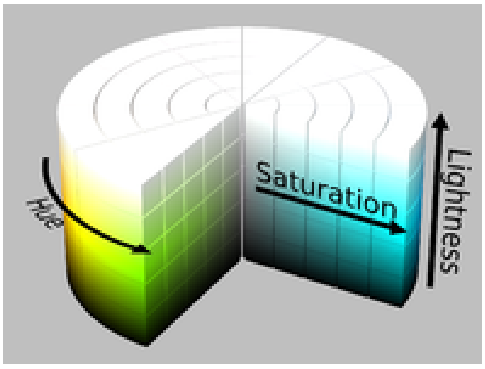


Figure 7: HSL Cylinder

The individual components of the HSL/HSV can be computed from the respective R,G,B values of the given image using the following equations. The expression below gives the angle that the Hue components makes with the saturation components.

$$\theta = \cos^{-1}\left\{\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2+(R-B)*(G-B)]^{\frac{1}{2}}}\right\}$$

- Hue

  The "attribute of a visual sensation according to which an area appears to be similar to one of the perceived colors: red, yellow, green, and blue, or to a combination of two of them". In the above figure that HSV color map has been used to display the Hue Channel.

$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases}$$

- Luminance (Y or L), Intensity (I)

  The radiance weighted by the effect of each wavelength on a typical human observer, measured in SI units in candela per square meter (cd/m2). Often the term luminance is used for the relative luminance, Y/Yn, where Yn is the luminance of the reference white point.

$$I = \frac{1}{3}(R+G+B)$$

- Saturation

  The "colorfulness of a stimulus relative to its own brightness".

$$S = 1 - \frac{3}{(R+G+B)}[min(R,G,B)]$$

- Lightness, Value (V)

  The "brightness relative to the brightness of a similarly illuminated white".

$$V = max(R,G,B)$$

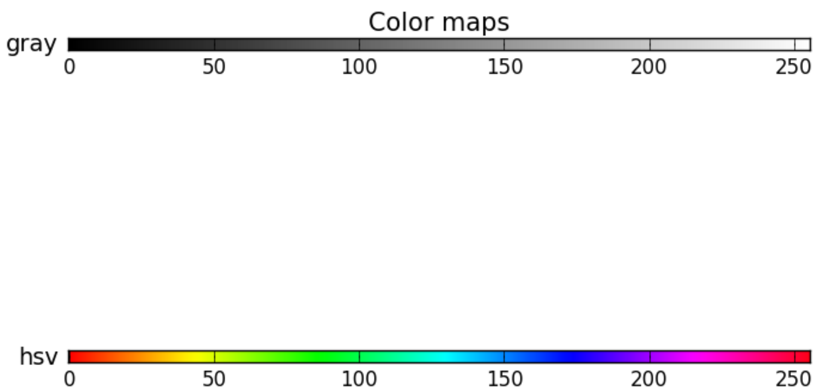


Figure 8: Pixel Values

3 Starters 3

Convert Image into $L^*a^*b^*$ and plot ?

3.1 Observations

Figure 9: Color image to Lab Image

3.2 Inference

The $L^*a^*b^*$ color space is derived from the CIE XYZ tristimulus values. The $L^*a^*b^*$ consists of a luminosity layer L^*, chromaticity-layer a^* indicating where color falls along the red-green axis, and chromaticity-layer b^* indicating where the color falls along the blue-yellow axis. All of the color information is in the a^* and b^* layers. $L^*a^*b^*$ is designed to approximate human vision. It aspires to perceptual uniformity, and its L^* component closely matches human perception of lightness.

4 Starters 4

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

4.1 Observations

Figure 10: Color image to Lab Image

4.2 Inference

Converting an image to Grayscale

To convert an RGB image to gray scale image, we use the following technique[2]

- We first apply gamma correction (denoted by C_{srgb}) to get the luminance or tristimulus values of individual channels, where C_{srgb} represents any of the three gamma-compressed sRGB (standard RGB color space) primaries (R_{srgb}, G_{srgb}, B_{srgb} also in range of [0,1]) and C_{linear} is

[2] `https://en.wikipedia.org/wiki/Grayscale`

the corresponding linear-intensity value (R_{linear}, G_{linear}, and B_{linear}, also in range [0,1]).

$$C_{linear} = \begin{cases} C_{srgb}, & \text{if } C_{srgb} \leq 0.04045 \\ \left(\frac{(C_{srgb} + 0.055)}{1.055}\right)^{2.4} & \text{if } C_{srgb} > 0.04045 \end{cases}$$

- Then, linear luminance is calculated as a weighted sum of the three linear-intensity values. The sRGB color space is defined in terms of the CIE 1931 linear luminance Y_{linear}, which is given by

$$Y_{linear} = 0.2126R_{linear} + 0.7152G_{linear} + 0.0722B_{linear}$$

5 Starters 5

Take a Grayscale image and illustrate

- Whitening
- Histogram equalization

5.1 Observations

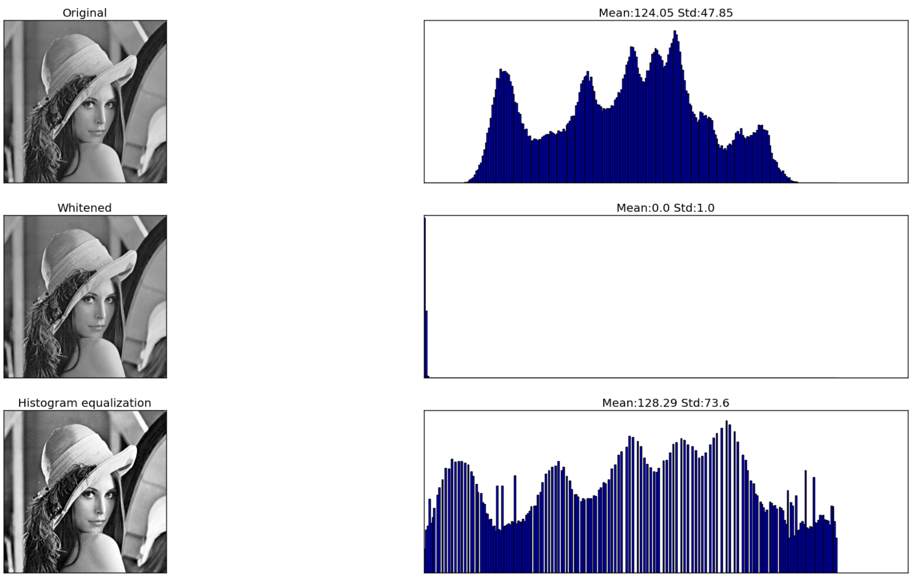


Figure 11: Whitening and Histogram equalization

5.2 Inference

- Whitening

  The whitening removes the variation in the mean(makes $\mu = 0$) and standard deviation ($\sigma = 1$) of the pixel intensities. An image is converted to gray scale prior to whitening so that we have only the intensities of the respective pixels. Then we apply the following operation on to each pixel, $x_{i,j}$

$$\mu = \frac{\sum_{i=1}^{I} * \sum_{j=1}^{J} * x_{i,j}}{IJ} \quad \sigma^2 = \frac{\sum_{i=1}^{I} * \sum_{j=1}^{J} * (x_{i,j} - \mu)^2}{IJ}$$

  Then μ and σ are used to transform each pixel intensities :

$$x_{i,j} = \frac{x_{i,j} - \mu}{\sigma}$$

- Histogram equalization

  The goal of histogram equalization is to modify the statistics of the intensity values so that all of their moments take predefined values. To this end, a nonlinear transformation is applied that forces the distribution of pixel intensities to be flat.

6 Starters 6

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

6.1 Observations

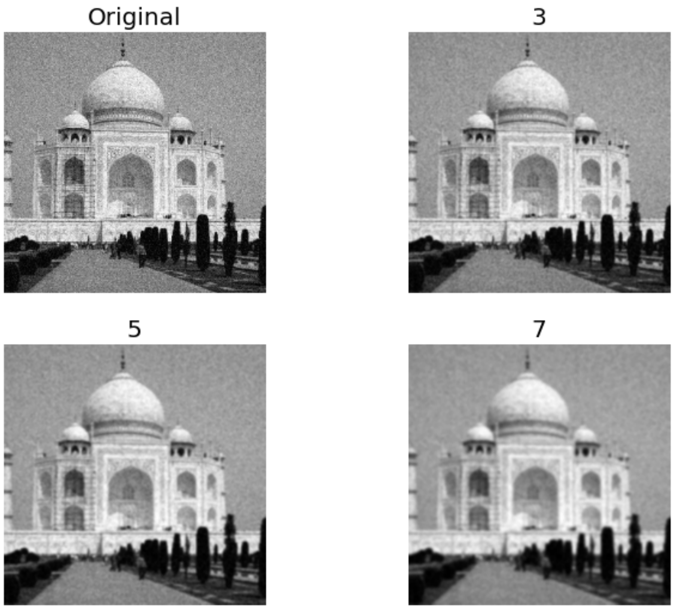


Figure 12: Gaussian smoothing with different kernel size

6.2 Inference

The Gaussian filter is used to remove noise from an image by smoothing/blurring it. The level of smoothing depends on the kernel size we are using. The higher the kernel size the more blur the image would become as show in the Observation section. Gaussian filter modifies the input signal by convolution with a Gaussian function.
The 2D Gaussian function is given by:

$$g(x,y) = \frac{1}{2\pi\sigma^2} * exp^{-\dfrac{x^2+y^2}{2\sigma^2}}$$

where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution.

7 Starters 7

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

7.1 Observations

Figure 13: Image Enhancement (Noise removal)

7.2 Inference

The image is mixed with salt and pepper noise and there after median kernel is applied to filter the noise. The median filter runs through each element of the signal (in this case the image) and replaces each pixel with the median of its neighbouring pixels. It is a nonlinear noise reduction technique. We use median filter as a pre-processing technique in image analysis.

8 Starters 8

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

8.1 Observations

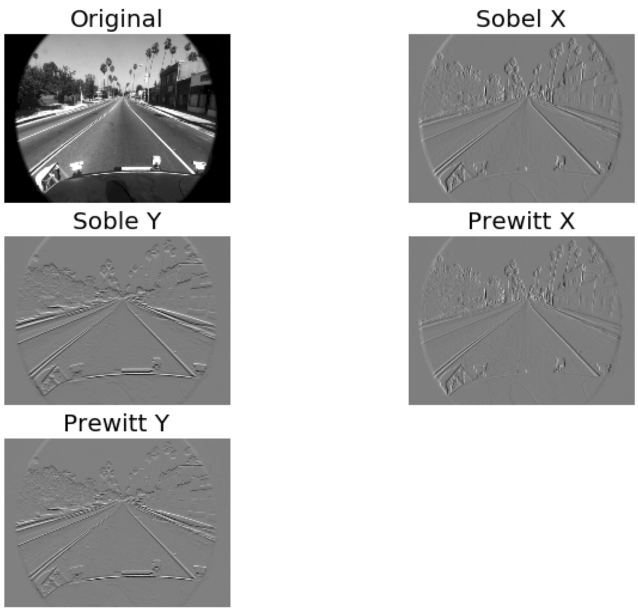


Figure 14: Prewitt and Sobel Operators)

8.2 Inference

The prewitt and sobel operators works like a first order derivatives and calculates the difference is pixel intensities in an edge region. Both can be applied horizontally as well as vertically depending up on the type of kernel we are using. The vertical kernel can be used to find the horizontal edges and horizontal kernel is used to find the vertical edges.

The Prewitt Kernels are given by :
The Sobel Kernels are given by :

-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
0	0	0
1	1	1

Table 1: Vertical mask(Left Table) and Horizontal mask(Right Table) of Prewitt Operator

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

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

Table 2: vertical Mask(Left Table) and horizontal Mask(Right Table) of Sobel Operator

9 Starters 9

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

9.1 Observations

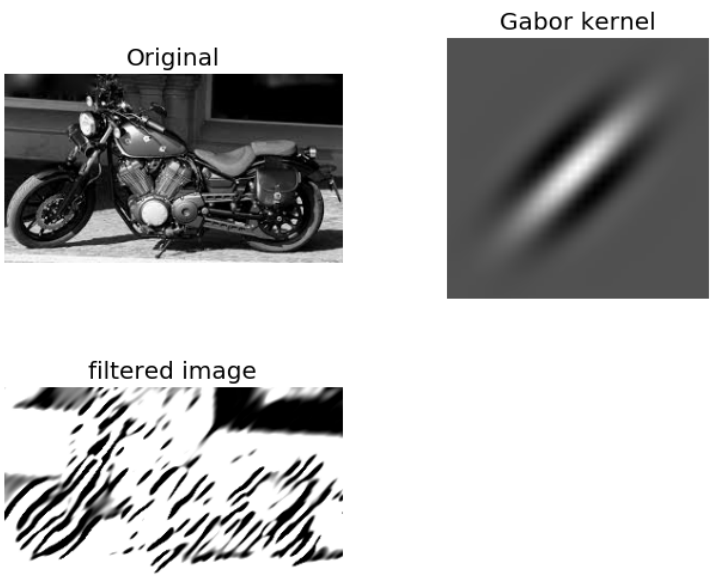


Figure 15: Gabor 45 degree filter

9.2 Inference

In the above observation we use Gabor filter for detecting 45° strips. The Gabor filter is a linear filter for edge detection. Since the frequency and orientation of Gabor filter is much similar to human visual perception it is used in image analysis.

The Gabor filter is defined by a sinusoidal wave multiplied by a Gaussian function. The mathematical representation of 2D Gaussian kernel function used in Gabor filter is given below :

$$g(x, y; \lambda, \theta, \phi, \sigma, \gamma) = exp\left(\frac{-(x^2 + \gamma'^2 y'^2)}{2\sigma^2}\right) exp\left(i\left(2\pi\frac{x'}{\lambda} + \phi\right)\right)$$
$$\text{where, } x' = x\cos(\theta) + y\sin(\theta)\ y' = -x\sin(\theta) + y\cos(\theta)$$

λ is the wavelength of the sinusoidal factor, θ represents the orientation of the normal to the parallel stripes of the Gabor function, ϕ is the phase offest, σ is the standard deviation of the Gaussian envelope and γ is the spatial aspect ratio.

10 Starters 10

- Take an image and observe the effect of Laplacian filtering
- Can you show edge sharpening using Laplacian edges ?

10.1 Observations

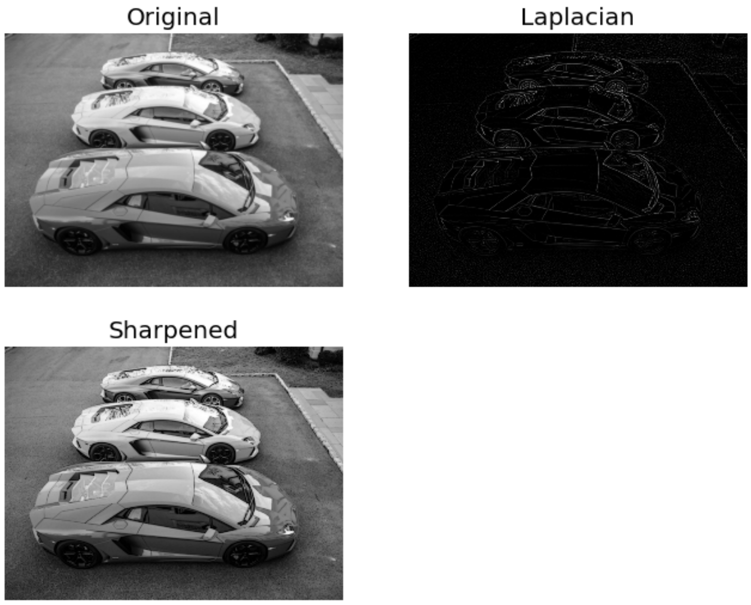


Figure 16: Image Enhancement (Noise removal)

10.2 Inference

The Laplacian $L(x, y)$ of an image with pixel intensity values $I(x, y)$ is given by :

$$L(x, y) = \frac{\partial^2 I}{\partial^2 x^2} + \frac{\partial^2 I}{\partial^2 y^2}$$

The Laplacian filtering helps to detect the edges of an images. It is a second order derivative operator. We apply laplacian on a smoothened gray scale image. Laplacian operator applied on the image above filters Inward edges. The Kernel of the laplacian is given by

Table 3: Laplacian Kernel for Inward Edge

0	1	0
1	-4	1
0	1	0

11 Main Course 11

Detect Road lane markers

11.1 Observations

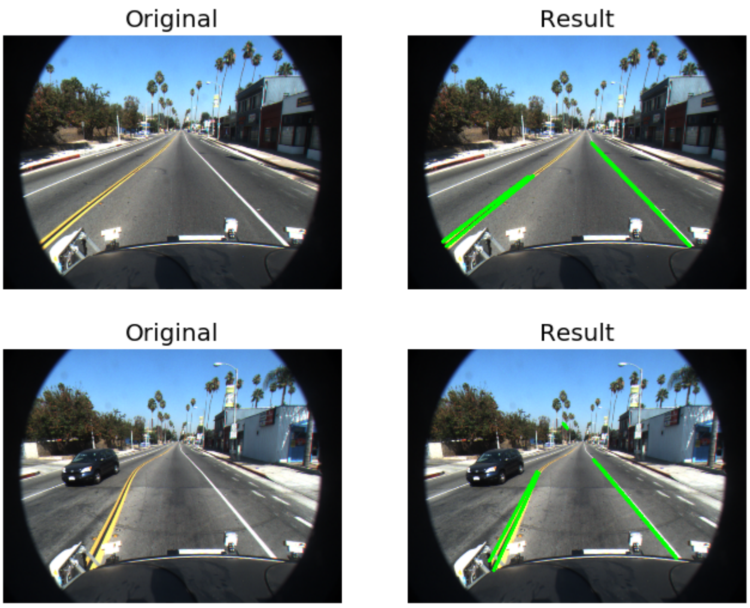
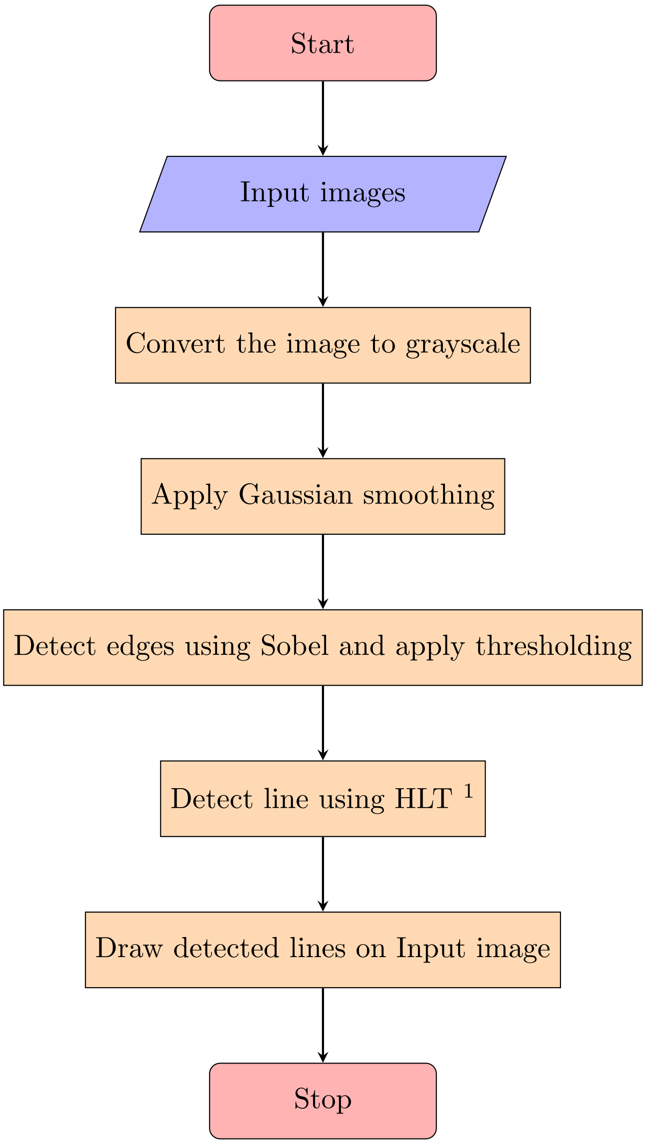


Figure 17: Lane Detection

11.2 Inference

The lane detection work using the following algorithms



[1]HLT stands forHough Line Transform

12 Main Course 12

Classify modes:

- Night
- Portrait
- Landscape

Design features (use NN)

12.1 Observations

Figure 18: Image Classification

12.2 Inference

The training images are resized to a standard size of 100x100 pixels. The feature vector consists of the flattened image matrix, where each pixel value represents a feature. The k-NN classifier, trained using this feature set and the associated labels, is then used on the test set with a k value of 3. The classifier has an accuracy of about 75% .