# LAB Manual UCS615: IMAGE PROCESSING

- **Write a program (WAP) to convert Grayscale image to binary image.**

  Case 1: Assume Mean Intensity Image as threshold value.

  Case 2: Input the threshold value from user.

- **WAP to convert RGB (Color) Image to grayscale image.**

  Case 1: By taking mean average of three planes.

  Case 2: Input the weightage of three planes, i.e., R, G, B from the user. The weightage is a value between 0 and 1, and sum of all three weightages is equal to 1.

  (Example: Weightage of R, G, B is 0.7, 0.2, 0.1, respectively.)

- **WAP to draw a border around the input image (or in other words pad the input image).**

  Input the width of the (in terms of pixel count) and the desired color of the border from the user.

  Try this for both binary and grayscale image.

- **WAP for Image Compliment.**

  Try this for both binary and grayscale image.

- **WAP to enhance the input image using Log transform.**

  Assume c=1.

- **WAP to enhance the input image using Power law/gamma transform.**

  Assume c=1, and input the value of gamma from the user.

- **WAP to enhance the input image using Intensity level slicing (Contrast stretching).**

  The input range of intensity values and the desired range of intensity values in the output image are obtained from the user.

  (Example: The input range of intensity values from 80 to 120 is stretched to 50-150 in the output image)

- **WAP to enhance the input image using histogram equalization.**

- **WAP to match the histogram of the input image with that of reference image using histogram matching technique.**

- **WAP to smooth the input image using:**

1. Averaging filter (un-weighted)
2. Weighted filter given by h(x,y)=max(|x|,|y|)
3. Gaussian filter

# LAB Manual UCS615: IMAGE PROCESSING

**Note:** Input the filter size and other parameters such as sigma in case of Gaussian filter, method of padding (replication, zero padding, etc.) from the user.

- **WAP to sharpen the input image using:**

1. Laplacian filter
2. Gradient operators of Sobel and Canny edge detector
3. Unsharp masking. To obtain blurred image, use the Gaussian filter of size 5x5 and sigma=3.

- **WAP to generate following noises using equations derived from PDFs of noise distributions. Compare your output with those generated using inbuilt function *imnoise*. Plot the histograms of generated noise to determine the shape of the distribution.**

1. Uniform noise
2. Gaussian noise
3. Erlang noise
4. Exponential noise
5. Rayleigh noise

- **WAP to implement the local binary pattern on gray scale image.**

- **WAP to denoise the input image corrupted by Gaussian and Speckle noise by using following filters (Use $3 \times 3$ or $5 \times 5$ window).**

  **(a) Weiner filter**
  **(b) Median filtering**
  **(c) Gaussian filter**
  **(d) Bilateral filter**

- **WAP to denoise the input image using adaptive median filter. Input the initial window size and maximum size of window (Smax) from the user. Compare the performance of median filter and adaptive median filter, using the PSNR values.**

- **WAP to threshold the image using Otsu's global threshold for segmentation of input gray scale image.**

- **WAP to use random values in confusion matrix and compute the different quantitative metrics (Accuracy, precision, recall, F1-score, MCC) for $2 \times 2\ and\ 3 \times 3$ matrix.**

- **WAP to implement basics ingredients of CNN by taking random values in $3 \times 3$ matrix.**

  **(a)** Convolution and kernel
  **(b)** Pooling (Max and Min pooling)
  **(c)** Padding
  **(d)** Activation function
  **(e)** Fully connected layer
  **(f)** Global average pooling
  **(g)** Dense layer
  **(h)** Drop out (BN, Early stopping, $L_1\ and\ L_2$ regularization)
  **(i)** Fully connected layers
  **(j)** Testing overfitting and under fitting issues

# LAB Manual UCS615: IMAGE PROCESSING

- **WAP to implement the custom CNN (using above ingredients) by taking the small dataset having 600 images.**
- **WAP to implement the VGG16 network by taking the small dataset having 600 images.**
- **WAP to implement the MobileNet by taking the small dataset having 600 images.**
- **WAP to implement the ResNet by taking the small dataset having 600 images.**