LAB # 10: Texture Based Descriptor

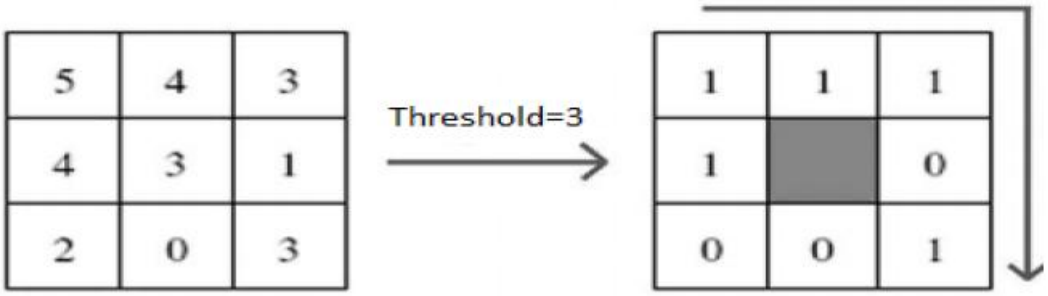
Objective:

The objective of this lab is to develop an understanding of texture based descriptors and how they can be further used

Theory:

Texture Based Descriptors can be useful for gleaning such information from an image that can provide good features for the said image.

Local Binary Pattern (LBP) extracts local texture features from an image by comparing the intensity of a central pixel to the intensities of its surrounding neighbors. These comparisons result in binary values, which are then used to form a binary pattern representing the local texture. Different variants of the LBP can be used to tackle issues such as rotation and scale. Rotated LBP is such an example.

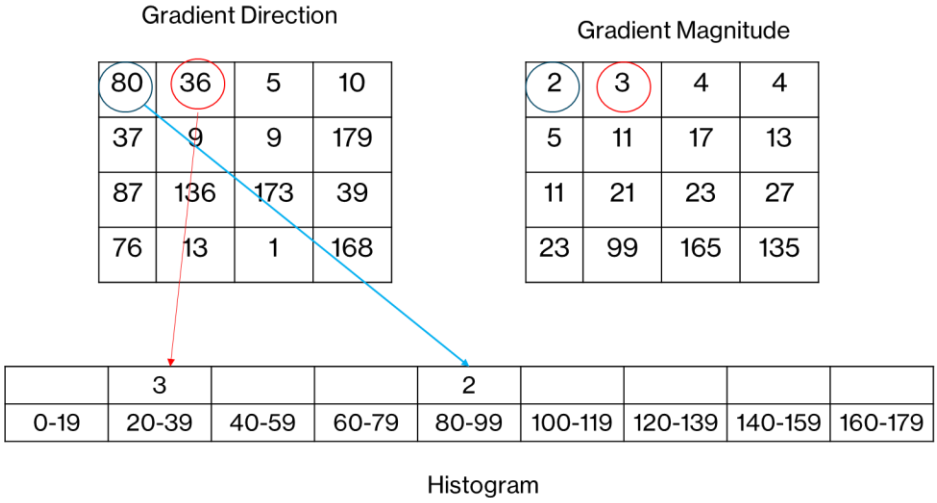


Similar to Local Binary Patterns (LBP), Histogram of Oriented Gradients (HOG) can also be used to extract image features. However, while LBP focuses on local texture patterns, HOG captures the distribution of gradient orientations and magnitudes within small local regions (cells) of an image. These histograms provide a representation of the local shape and structure of objects which can then be used for classification tasks among other things. To compute HoG for an image, the following steps can be used:

1. Compute the gradient magnitude and the gradient direction of the image using Sobel X and Sobel Y filters as done in one of the previous labs.
2. We look at a local group of pixels in the image by using the following convention:

a. A n x n cell. For example, an 8x8 cell. This simply is the size of the window.
b. An m x m block consisting of adjacent cells. For example, 2x2 cells for a total of 4 cells in a block. So, the size of the block will be 16x16. Here we have considered non-overlapping blocks. We can also have some percentage of overlap in the blocks.

3. Next, we divide the angle range into bins. For example, if the angles go from 0 to 180 and we want to have 9 bins, then each bin will be 20 degrees wide.
4. For each pixel in each cell, we look at the angle from the Gradient Direction image for the image and depending on what bin it falls into (step 3), we add the magnitude of that pixel to that bin as shown below. The gradient magnitude is summed up for every pixel that falls into a particular bin:



5. From step 4 above, we get a histogram for each cell. For all the cells in a block, the histograms are concatenated together resulting in a longer feature vector.
6. Normalization can be applied to the feature vector of each block using the L2 normalisation (or some other normalization) as given below. Values greater than 0.2 even after normalization can also be clipped.

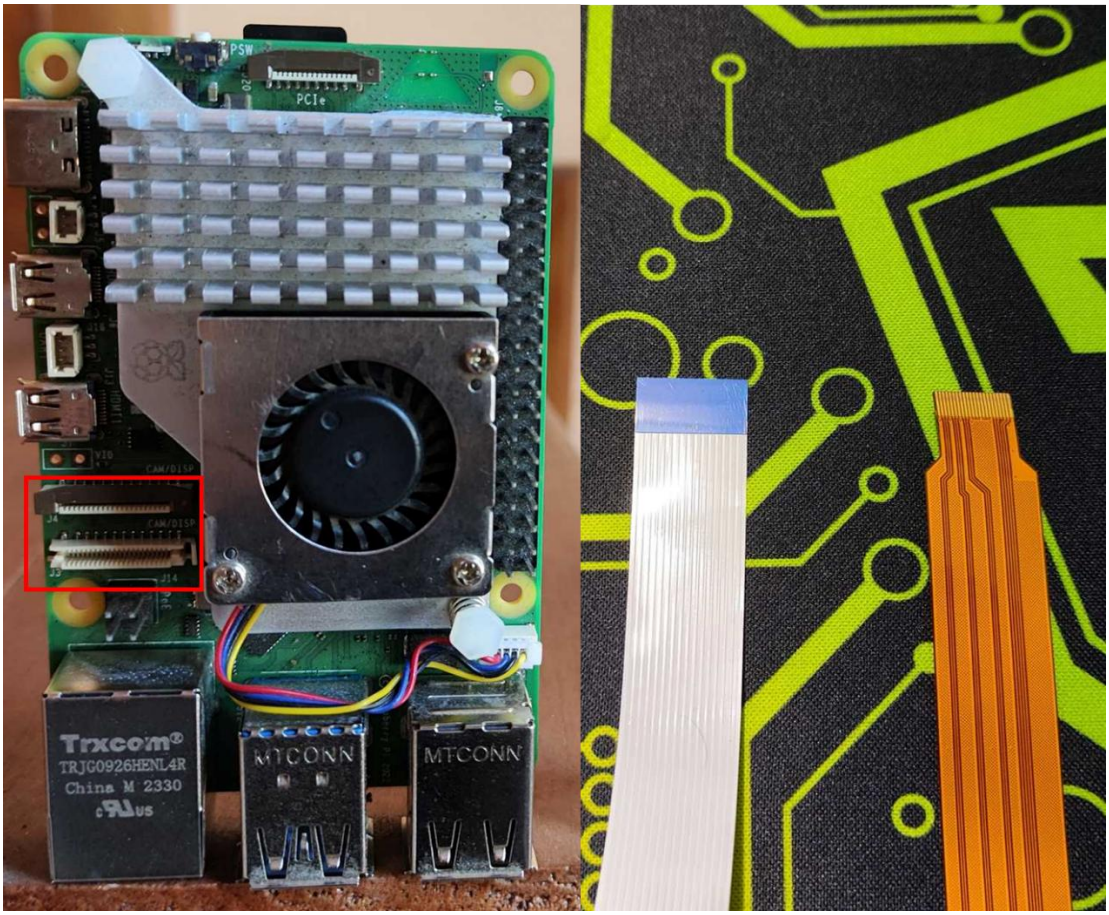
$$L2 - norm : v \longrightarrow v/\sqrt{||v||_2^2 + \epsilon^2}$$

7. Finally, we just concatenate the histograms of all blocks as feature vectors to get the HoG descriptor for the image.

Setting up the Raspberry Pi with the Camera:

We are already aware of how to set up the Raspberry Pi 5 for Raspberry Pi OS as we have covered it in a previous lab.

Today, we are going to interface a camera with the Pi 5 for video capture. The Pi 5 comes with 2 MIPI/CSI connectors that can be used for this purpose. However, we do need to use the right cable (picture below).



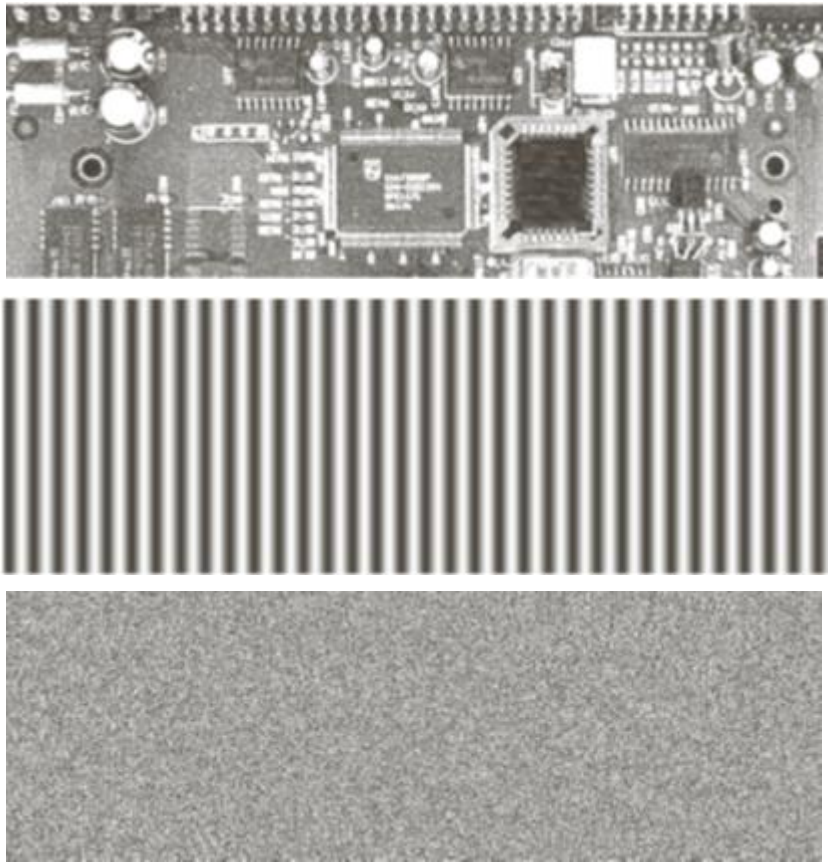
You can follow the tutorial here to get the camera up and running: https://www.xda-developers.com/connect-a-camera-module-to-raspberry-pi-5/

Once the camera is accessible, you can use the following Python commands to get the input from the camera in your Python script:

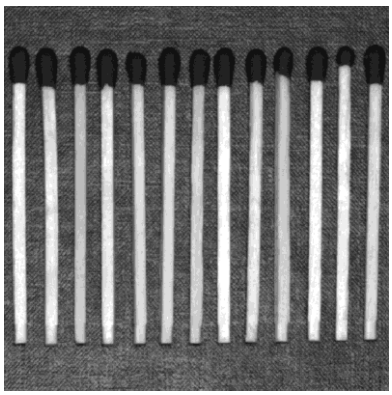
```
cap = cv2.VideoCapture(0)

while True:

    ret, frame = cap.read()

    cv2.imshow("Camera ", frame)

cap.release()
```

Lab Tasks:

Lab Task 1: Apply Local Binary pattern algorithm for following image. Then all apply Rotated Local Binary Pattern for the same image. Plot the histograms of both variants of LBP. Use $LBP_{(8,1)}$ for this task.



Lab Task 2: Calculate HoG for the image shown below. Show the HoG descriptor that you get for the image.



Lab Task 3: You need to run the LBP and HoG on input captured by the camera and display the results.