**MODEL 3:BODY DETECTION**

**1 INTRODUCTION**

In this section we explore human computer interaction under the heading of body detection. Detecting human beings in images is a challenging task, for their varying height, colour and race provide computers a complex calculation of accounting for such diversity.

However body detection is a powerful tool in human computer interaction and finds applications in self driving cars, detecting pedestrians crossing the road, in surveillance systems etc.

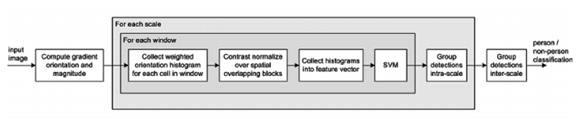
Here we use the concept of Histogram of Oriented Gradients (HOG) to extract features of the human body and then use a linear Support Vector Machine(SVM) in order to detect human bodies in a video stream.

**2 REVIEW OF LITERATURE**

In 2001 the Haar Cascade classifier also known as the viola jones algorithms were considered the state of the art for feature extraction ,however since then more complicated and robust algorithms for feature extraction have been explored. Gavrila and Philomen built a pedestrian detector by extracting edges in images and matching them to a set of learned exemplars using chamfer distance. SIFT appearance detectors are popular approach for object detection.

SIFT algorithm was proposed by David L Gowe in 2004.Ronfard built an articulated body detector y incorporating SVM based limb classifiers over 1st and 2nd order Gaussian filters in a dynamic programming framework. Finally Dalal and Triggs introduced the use of HOG feature extraction to detect pedestrians on the MIT pedestrian database.

**3 METHODOLOGY**

****

In this project we make use of opencv and python. OpenCV ships with a pre-trained HOG plus Linear SVM model that can be used to perform pedestrian detection in both images and video streams. We use this pre trained classifier on a video stream, from a web camera.

*A COMPUTING HOG*

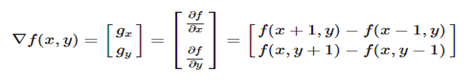
The Histogram of Oriented Gradients (HOG) is an efficient way to extract features out of the pixel colours for building an object recognition classifier

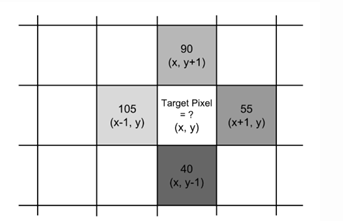
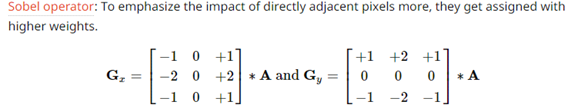
*1.Calculating image gradient vector*

The [image gradient vector](https://en.wikipedia.org/wiki/Image_gradient) is defined as a metric for every individual pixel, containing the pixel color changes in both x-axis and y-axis. The definition is aligned with the gradient of a continuous multi-variable function, which is a vector of partial derivatives of all the variables. Suppose f(x, y) records the color of the pixel at location (x, y), the gradient vector of the pixel (x, y) is defined as follows:

Magnitude is the L2-norm of the vector, g=g2x+g2y−−−−−−√g=gx2+gy2.

Direction is the arctangent of the ratio between the partial derivatives on two directions, θ=arctan(gy/gx)θ=arctan⁡(gy/gx).



Now calculating this gradient magnitude and arctan for every pixel is a heavy computation cost. Hence predefined operators exist for this purpose which convolve on the image hence we call them kernels ,example prewitt ,sobel etc ,in our code we use sobel kernel 

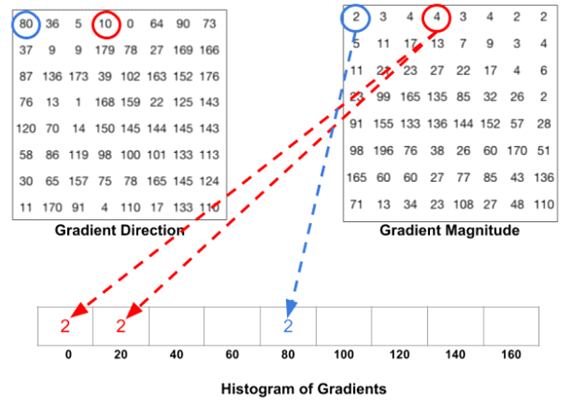
1. *Calculating HOG*

1)Pre-process the image, including resizing and color normalization.

2) Compute the gradient vector of every pixel, as well as its magnitude and direction.

3) Divide the image into many 8x8 pixel cells. In each cell, the magnitude values of these 64 cells are binned and cumulatively added into 9 buckets of unsigned direction (no sign, so 0-180 degree rather than 0-360 degree; this is a practical choice based on empirical experiments).For better robustness, if the direction of the gradient vector of a pixel lays between two buckets, its magnitude does not all go into the closer one but proportionally split between two. For example, if a pixel’s gradient vector has magnitude 8 and degree 15, it is between two buckets for degree 0 and 20 and we would assign 2 to bucket 0 and 6 to bucket 20.

4) Then we slide a 2x2 cells (thus 16x16 pixels) block across the image. In each block region, 4 histograms of 4 cells are concatenated into one-dimensional vector of 36 values and then normalized to have an unit weight. The final HOG feature vector is the concatenation of all the block vectors. It can be fed into a classifier like SVM for learning object recognition tasks.



*3.Training HOG*

Step1

Sample *P* positive samples from your training data of the object(s) you want to detect and extract HOG descriptors from these samples.

Step2

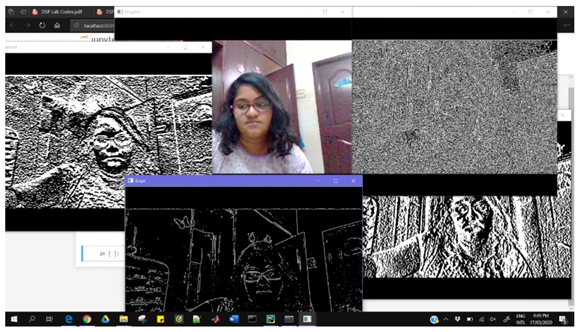
Sample *N* negative samples from a *negative training set* that ***does not contain*** any of the objects you want to detect and extract HOG descriptors from these samples as well. In practice *N >> P.*

Step3

Train a Linear Support Vector Machine on your positive and negative samples.

Step4

Apply to dataset

IMAGE GRADIENTS 

1.Original

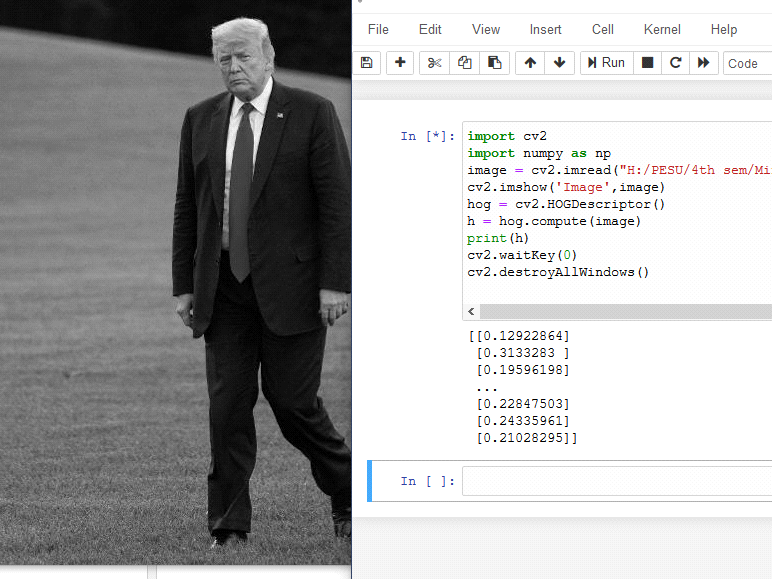
2.Horizontal Gradient

3.Vertical Gradient

4.Laplacian

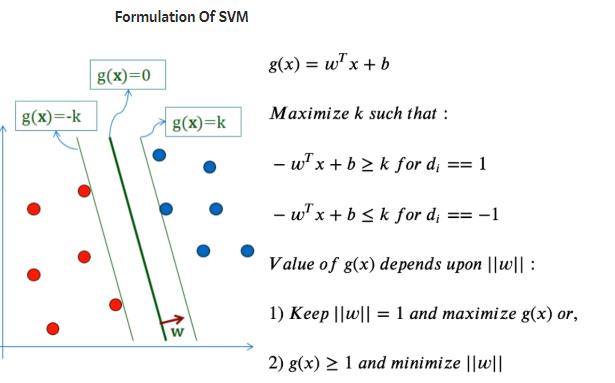
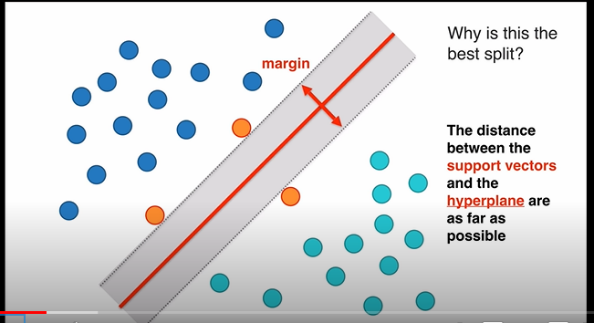
5.Edge detection

Sample HOG calculation of image



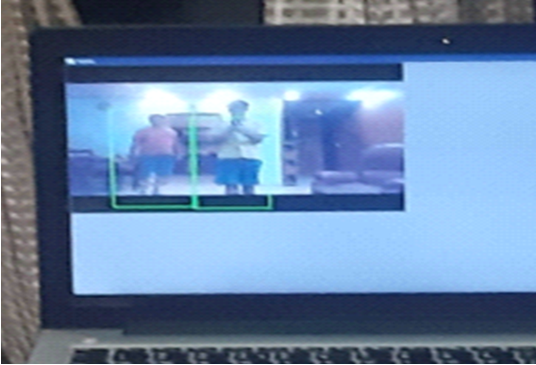
*3.Support Vector Machine*

SVM or support vector machine is the classifier that maximizes the margin. The goal of a classifier in our example below is to find a line or (n-1) dimension hyper-plane that separates the two classes present in the n-dimensional space.



The seperating line is called hyper plane and the nearest points to the hyperplane are called support vectors. A hyperplane is a line for 2d plane for 3d and so on. This is also know as constrained optimization constrained because the points cannot be on the line optimum because it is the widest separation. SVM are based on lagranges multipliers.

1. **RESULT**

****

1. **REFERENCES**

<https://www.pyimagesearch.com/2014/11/10/histogram-oriented-gradients-object-detection/>

N. Dalal, B. Triggs, “Histogram of oriented gradients for human detection,” IEEE Conference on Computer Vision and Pattern recognition, vol. 2, pp. 886–893, June 2005.

GL. Gan, J. Cheng, “Pedestrian detection based on HOG-LBP Feature,” International Conference on Computational Intelligence and Security,pp. 1184-1187, Dec 2011