Pedestrian detection using HoG features

Sarthak Ahuja
IIIT-Delhi
sarthak12088@iiitd.ac.in

Prateekshit Pandey
IIIT-Delhi
prateekshit12078@iiitd.ac.in

Abstract

Human Detection in Images is a contemporary Computer Vision problem, still welcoming improved solutions. This subset area of object detection has seen many attempts made towards efficient implementation and in this project proposal we describe one based on Histogram of Oriented Gradients which proves to be superior than the rest in terms of both Detection rate and Error rate when using a Linear SVM Classifier.

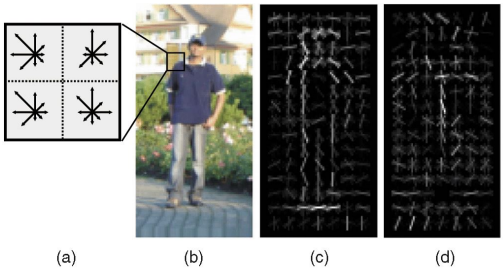

Figure 1. Histogram of Oriented Gradients

1. Motivation

The problem of detecting pedestrians in an image or a video finds applications in countless domains like crowd estimation, congestion analysis, surveillance videos, robotic vision, and self-driven vehicles. Owing to their complex shapes and poses, detecting Humans is a challenging task which requires the need of some robust feature which can distinguish them from the scene. Histogram of Oriented Gradients are one example of such a feature source, which we aim to implement and evaluate as a part of this project.

2. Previous Work

Owing to its age, this problem has a lot of literature published. Two of the most important approaches include using HAAR wavelet descriptors as input parameters or using a parts based method containing detectors for various parts of the body. An extension of this algorithm can be seen as the skeleton modelling done by kinect using RGB-D images. The approach we propose to implement is much simpler than the above mentioned methods and is proven to provide significantly higher performance in the real world.

2.1. Histogram of Oriented Gradients for Human Detection

The basic idea behind this approach is capturing the object appearance and shape by characterizing it using Local intensity Gradients and Edge Directions. The image is densely divided into small spacial regions called cells. (Figure 1)For each cell a 1-D histogram of gradient directions/edge directions is computed and later all cell data is combined to give a complete HoG descriptor of the window. The variety of colors and illumination in the surrounding makes normalization inevitable. We further describe the normalization technique as a part of our approach later in the report. In their work, Triggs and Dalal[1] use the MIT and Inria Dataset(1805)[2][3]. They generate a sufficiently large negative set by sampling out patches from person-free images. Later in their work, they provide a detailed comparison with other state-of-the-art methods- Generalized Haar Wavelets, PCA-SIFT and Shape Context methods showing great superiority of the HOG implementation over the rest. Besides the usual square R-HOG Blocks, comparison has even been shown after carrying out the training and testing with vertical descriptors (2x1 cell), horizontal descriptor (1x2 cell) and C-HOG geometry Blocks. Learning from their inferences we look to implement the proposed system.

3. Problem Statement/Objective

We aim to implement an Object Detection System for detecting and marking Pedestrians in a scene. We intend to implement a system based on supervised learning of HoG Features extracted from the MIT and INRIA Pedestrian database and classified using a linear SVM.

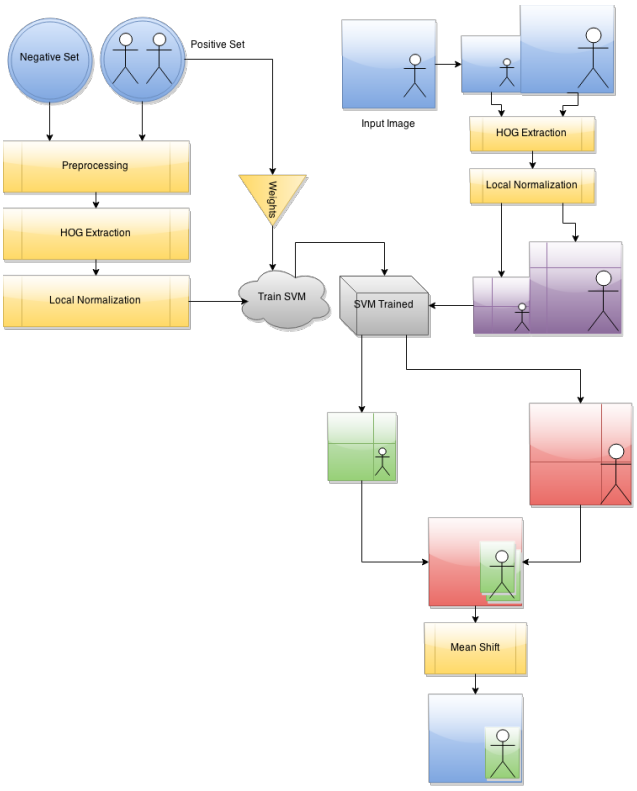

Figure 2. Flowchart for proposed algorithm

4. Proposed Approach

Our implementation will be closely based on the works on Dalal and Triggs with little modification. Figure 2 shows the process flow we intend to follow.

4.1. Image Acquisition

Images acquired from the standard dataset were used for testing. Other than that picture captured from mobile phone cameras were used to validate the system.

4.2. Preprocessing

We will be working and processing the images in the gray scale space as the above cited paper suggests and gives no distinct advantage of using the RGB or LAB color spaces. Apart from that we apply gamma normalization to improve the intensity of the image. This is done as images taken from the mobile device had low illumination.

4.3. Dataset

We will use the MIT and (Figure 3)INRIA Pedestrian data set for Training and Testing of our SVM. Instead of the Hold-Out method used in the cited approach we plan to use k-cross validation. We have a data set of approx 2100 images. (900 MIT +1200 INRIA). We try all combinations of the datasets to identify the best one.

4.4. HOG Extraction

We will be implementing the HoG Feature Extraction procedure from scratch following the implementation given by Dalal and Triggs:

1. Gradient Computation: The most common method to compute gradient is to simply apply the point discrete derivative mask in both horizontal and vertical directions. This method requires filtering the intensity data of the image with the kernels [-1 0 1] in both horizontal and vertical directions.

2. Orientation Binning: A cell histogram is created by weighted quantization of the orientation of each pixel of the the cell into pre-defined orientation-based bins. The cells are usually square in shape (for convenience we will stick with rectangular), but they can be rectangular or circular. The weighting of the orientations can be either by using the gradient magnitude itself.

3. Block Division and Normalization: The cells must be grouped together in order to factor in changes in illumination and contrast. The complete HOG descriptor is then the vector of the components of the normalized cell histograms from all of the block regions. Two main block geometries exist: rectangular R-HOG blocks and circular C-HOG blocks. R-HOG blocks are generally square grids, represented by three parameters: the number of cells per block, the number of pixels per cell, and the number of channels per cell histogram. C-HOG has two variants: ones with a single, central cell and the ones with angular divided cells, which can be described by number of angular/radial bins, radius if center bin and expansion factor for radius of additional radial bins. These blocks are normalized by four prominent methods: L1 norm, L1 norm square root, L2 norm and L2 norm followed by clipping (L2 Hys). We will experiment and choose the one that works best.

4.5. Training the Classifier

Based on the literature survey done, we chose a linear kernel SVM in our project. We kept the error cost to the default value of 1. We attempted using weighted SVMs but were not able to implement it as a part of this project due to technical difficulties with MATLAB.

4.6. Weighted SVM

The figure above shows the result of averaging all the image gradients in the training set. We observe that pedestrians are actually being classified based on their silhoutte

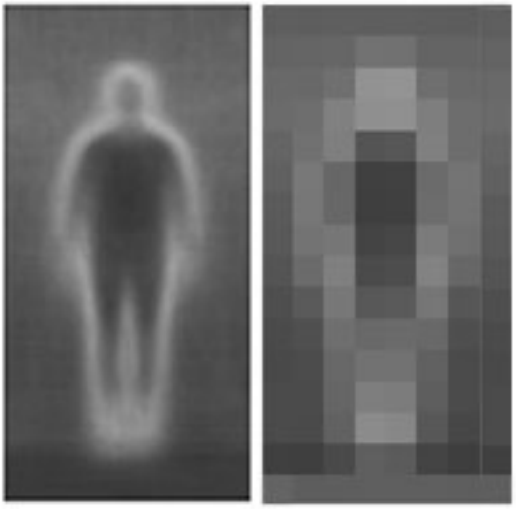

Figure 3. Average gradient image: original and sampled

boundaries rather than the whole image itself. On the contrary, rest of the image is actually proving to be counter productive as the svm is being trained on them as well. The variation in the texture found on the clothes and the random texture in the background can be dealt with by giving them lower weights and giving more weight to these regions around the silhoutte. We implement this by sampling the gradient image into the same number of blocks as in the case of HOG extraction. All the histogram inputs from one block are given the corresponding weight as in the sampled image.

4.7. Accuracy

In our verification experiment we perform 5-fold cross validation on multiple combinations of the datasets. The ccuracies received were as follows:

Patch Size	Combined DB	MIT DB	INRIA DB
8 x 16	96.8521%	92.213%	91.927%
4 x 8	94.213%	90.125%	89.312%

4.8. Sliding Window

To detect pedestrian in a given image, we applied a sliding window approach to predict the presence of a pedestrian in a window which kept sliding over the complete image. The window has been shifted ("slided") with a step length equal to the length and width of the block size respectively in the vertical and horizontal direction.

This process is computationally heavy, given that the gradient magnitudes and orientations for each patch needs to be computed for each window during sliding. To speed up the process, we divide the whole image into the blocks of the given block size, and compute gradient histograms over them before applying the sliding window. Thus, after computing these histograms before-hand, now while applying sliding window, we just need to consider the subset of blocks which belong to the window and concatenate their hostograms to get our feature vector.

4.9. Clustering the Results

There were a lot of resultant overlapping patches, owing to that fact that the sliding window would capture the same individual multiple times while sliding over it. To address this issue, we tried applying mean-shift clustering over the center points of the detected patches.

Figure 4. Example images from INRIA dataset

5. Results

On testing over a wide set of images collected in the image. the system works reasonably robustly. Link to the code.

6. Challenges

1. Slow Speed: Owing to MATLABs internal structure the sliding window was translating very slowly even though we had used significant preprocessing steps to reduce computation as much as possible.

2. Deciding Depth and Quantization of Scale Space: Since we were not implementing dense scaling due to its computational complexity, we had to choose among the different scales we want our algorithm to work on manually. This information is usually determined after knowing the exact location of the camera.

3. Finding the appropriate block size: Due to our preprocessing step aimed towards improving the computational complexity of the code, we were forced to move

the sliding window with a displacement of the block size. This could have potentially lead to a fall in the detection rate as the pedestrian may not come in the center of the window(high accuracy region).

4. Issues while application of weights to the SVM training process: Our code has the complete weight vector in the required format ready. We could not include it in the system as matlab had some technical difficulties in doing so.

5. Clustering: Our idea of clustering the window centers did not work out well as it required a dependent parameter - the window size in case of mean shift and 'k' in the case of k-means. Due to a diverse variety in the scale of the pedestrians and the case of congested pedestrians, clustering does more harm than good. It works reasonably well in the case of a uniform scale sparse set of pedestrians.

7. Future Work

As part of future work we look forward to applying the deformable parts model approach to the system. It includes training individual classifiers for different body parts and later during the detection phase combining their outputs and classifying based on some energy minimization in their spatial consistency. We also look forward to integrate online learning to our code to make the classifier more robust over time. The current system only work on images. We can possibly extend it to videos by integrating motion HOG to our code.

8. References

[1] Histograms of Oriented Gradients for Human Detection,Dalal, Navneet and Triggs, Bill,Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). [2] MIT Dataset: http://cbcl.mit.edu/software-datasets/PedestrianData.html
[3] INRIA Dataset: http://pascal.inrialpes.fr/data/human/