# Pedestrian Detection and Tracking in Images and Videos

Azar Fazel
Stanford University
azarf@stanford.edu

Viet Vo
Stanford University
vtvo@stanford.edu

## Abstract

*The increase in population density and accessibility to cars over the past decade has led to extensive computer vision research in recognition and detection to promote a safer environment. Primarily, much of the research focuses on detecting pedestrians in order to reduce the chance of collision, and to improve traffic control. The need for increased surveillance at the work place and at home also promotes research in this area. We implemented a pedestrian detection and tracking algorithm by using histogram of oriented gradients (HOG) features and a linear support vector machine (SVM) and Random Forest classifier. Our goal was to analyze and generate different bounding boxes for people within static images, and finally apply this strategy to localize and track people within videos. We benchmarked different HOG parameters to find the best model, and furthered our experimentation by comparing the effectiveness of SVM versus Random Forests. Our implementation was able to achieve 80% accuracy in static images, and was able to track pedestrians in videos if the detected pedestrians' poses do not vary significantly.*

## 1. INTRODUCTION

Just in the United States, 5,000 of the 35,000 annual traffic crash fatalities involve pedestrians [1]. Computer vision research in the area of pedestrian detection is becoming increasingly crucial as more intelligent motor vehicles are introduced into the streets. However, pedestrian tracking and detection is inherently a hard problem to solve due to high intra-class variability and partial occlusions. Our goal was to benchmark different feature parameters from HOG, and compare the success of SVM versus Random Forests for pedestrian detection. We implemented our algorithms on Python and utilized several computer vision packages from OpenCV, machine learning packages from sklearn, and imaging processing packages from scikit-image. In order to train our model, we used a combination of 5,400 positive 64 x 128 images from the Inria's Person dataset [2], PETA dataset [3], and the MIT database [4]. Our negative images were also from these databases and images from the Daimler Mono dataset [5], containing a total of 2,100 images. For each non-pedestrian image, 10 random windows of 64 x 128 pixels were extracted for training, giving a total of 21,000 negative images. This trained model was then used to test the detection accuracy on images, and track pedestrians in videos.

## 2. PREVIOUS WORK

Many techniques are being used today for pedestrian detection. One such technique that is similar to HOG is Scale Invariant Feature Transform (SIFT). This technique generates features by using Difference-of-Gaussians (DoG) in an image's scale-space pyramid to find interesting local keypoints. Each keypoint will have a orientation vector, and is invariant to scale and rotation. Due to the high dimensionality of SIFT features, principal component analysis is often used in conjunction with SIFT [10]. Although SIFT can be effective for detecting human features, Dalal and Triggs explained in their paper that locally normalized HOG descriptors were more effective [6]. They experimentally showed that the dense grid of uniformly spaced cells and overlapping local contrast normalization improved the detection performance as compared to SIFT. In fact, HOG features have been shown to have 1 to 2 orders of magnitude less false positives

then other approaches.

The Deformable Parts Model (DPM) is another technique for object detection that performs well at classifying highly variable object classes. In this technique, for each image, a HOG feature pyramid is formed by varying the scale of the image, and defining a root and parts filter. The root filter is coarse and is used to capture the general shape of the object, while higher resolution part filters are used to capture small parts of the object. Objects are then detected by computing the overall score for each root location based on the best possible placement of the parts [7].

The other technique in pedestrian detection is Convolutional Neural Networks (CNN). This technique shows outstanding power in addressing the pedestrian detection problem, especially in the context of autonomous driving. In CNN, it learns which convolution parameters can produce better features to easily predict an optimal output. Then it uses these features by extracting them from the last fully connected layers to train an SVM model for pedestrian detection[11][12].

## 3. TECHNICAL APPROACH

Using our dataset of positive and negative images, we extracted features using the histogram of oriented gradients technique described by Dalal and Triggs. This technique divides the image into dense equal sized overlapping blocks. Each of these blocks are further divided into cells which will be used to find a 1-D histogram of gradient edge orientations over the pixels of the cell. For this project, we experimented with block sizes that were 2x2 and 4x4, and cells sizes that were 8x8 and 16x16 in order to find the best combination. For our histograms, we used 9 orientation bins across all experiments. Histograms for each block are combined and finally normalized to have better invariance to illumination and shadowing [6]. Figure 1 shows an example of extracted HOG features for a pedestrian.

In order to train the model, the feature vectors for the images were fed into a linear SVM classifier. This model was then used to classify pedestrians from non-pedestrians. We implemented a sliding window approach to exhaustively search static images for windows with the scores greater than 0.2. The scores for each window was calculated using the weight and bias

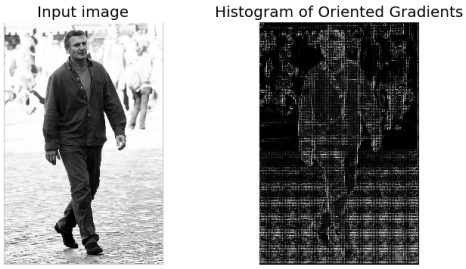


Figure 1: Example of HOG features, the right picture is the original image and the left one is the extracted HOG features.

found from our SVM model. Since our sliding window was kept at a constant size of 64x128 pixels, we implemented an image pyramid approach for our detection. In this approach, for each image, we scaled down the image by 15% of its original size for several iterations until the size is below a threshold of 64 pixels for width and 128 pixels for height. For each iteration, our detector window searched the entire scaled image and calculated scores using our SVM model. Once this algorithm was finished, scaled bounding boxes was displayed on the original image and non-maximal suppression applied to eliminate redundant boxes.

In order to reduce false positive rates, we mined for hard negative examples using our negative training data. We extracted all false positive objects found within negative images and included these examples into our training data for retraining the classifier.

Due to the exhaustive search performed during HOG feature extraction, the time complexity for object detection is very high. This poses a problem for pedestrian tracking in videos because detection rates would be too slow. In order to remedy this problem, objects that are moving will be extracted from each frame using background subtraction [8]. Using this method, we detected motion by segmenting moving objects from the background and passing these smaller images into our model instead of passing the whole frame for detection. The nth frame can be represented as $I_n$ which is its intensity value. $I_{n-1}$ will correspond to the previous frame. Doing a pixelwise subtraction, we get the equation

$$M_n = \begin{cases} I_n(i,j) & \Delta(i,j) \geq T_{threshold} \\ 0 & \Delta(i,j) < T_{threshold} \end{cases}$$

where i and j are pixel positions and $M_n$ is the motion image. By finding the motion image, we can dramatically reduce the complexity of our computation [9]. Using these motion images, we were be able to run our model on video frames much faster than when we did not have any motion detection.
Figure 2 provides a summary of the steps we have in our detection algorithm.

## 4. EXPERIMENTS AND RESULTS

To obtain the model with the highest accuracy, we tried two different classifiers: SVM and Random Forest. To evaluate our models, we tested them on validation set including 1,000 pedestrian images. For the SVM classifier, we investigated different regularization parameters(C) to get the highest accuracy. The regularization parameter tells the SVM optimization how much we want to avoid misclassifying each training example. For large values of C, the optimization will choose a smaller-margin hyperplane if that hyperplane does a better job of getting all the training points classified correctly. Conversely, a very small value of C will cause the optimizer to look for a larger-margin separating hyperplane, even if that hyperplane misclassifies more points. We got the highest accuracy for the SVM model when the regularization parameter has the value of 0.001.

For Random Forest, we examined different number of trees in training the model. Random Forest uses bagging (picking a sample of observations rather than all of them) and random subspace method (picking a sample of features rather than all of them) to grow a tree. If the number of observations is large, but the number of trees is too small, then some observations will be predicted only once or even not at all. If the number of predictors is large but the number of trees is too small, then some features can be missed in all subspaces used. Both cases results in the decrease of random forest predictive power. But the last is a rather extreme case, since the selection of subspace is performed at each node. In general, the more trees we use the better get the results. However, the improvement decreases as the number of trees increases, i.e. at a certain point the benefit in prediction performance from learning more trees will be lower than the cost in computation time for learning these additional trees. For our dataset, Random Forest provided the best accuracy with 1,000 trees.

Furthermore, in order to tune the hyper parameters for HOG features, we extracted them using different block sizes and cell sizes. Table 1 shows the results of these experiments. As seen in this table, the block size and the cell size have a significant affect on the accuracy of our models. In the other words, the effectiveness of the models strongly depends on the HOG feature parameters. Also from the table, we can see that the Random Forest outperforms SVM in all the cases except when the block size is 2x2 and the cell size is 4x4 which the accuracy of the SVM model is higher than the Random Forest. According to these results, we can conclude that there is no optimal configuration for HOG features and it depends on the dataset we are using. In order to reduce false positive rates in our models, we exhaustively searched all 2,100 negative images and extracted 5,800 windows with the size of 64x128 pixels as false positive objects and then retrain our model with the new augmented set. Using 1,000 new negative images for validation, the original model had a false positive rate of 0.005% while the new model with hard negative mining had a 0% false positive rate. Most of the false positives came from objects that are erect and skinny such as poles and trees. However, our hard negative mined model eliminated many of these false positives. An example of this improvement is seen in Figure 3.

As mentioned in the section 3, since we found multiple bounding boxes for each object, we used non-maximal suppression to remove the redundant bounding boxes. Figure 4 shows an example of using non-maximal suppression for two images.

For the purpose of background subtraction, we calculated a reference image using a Gaussian Mixture-based background/foreground segmentation algorithm. Then, we subtracted each new frame from this image to compute a foreground mask. The result is a binary segmentation of the image which highlights regions of non-stationary objects. This way we were be able to get the segmentation of moving regions in image sequences in Real-time. Figure 5 shows an example of the background subtraction for one frame of a video.

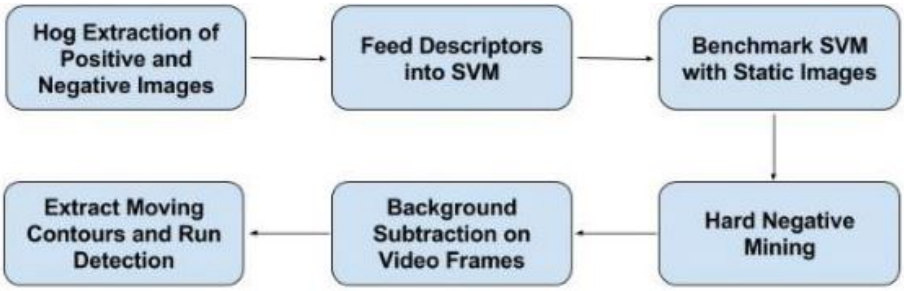


Figure 2: Flow Chart of Major Steps for Pedestrian Detection and Tracking.

| Block size | Cell Size | SVM Accuracy | Random Forest Accuracy |
|---|---|---|---|
| 2 | 8 | 80.8% | 67.7% |
| 2 | 16 | 69.7% | 81.1% |
| 4 | 8 | 66.7% | 69.1% |
| 4 | 16 | 0% | 80.4% |

Table 1: The accuracy of SVM and Random Forest models using different HOG parameters

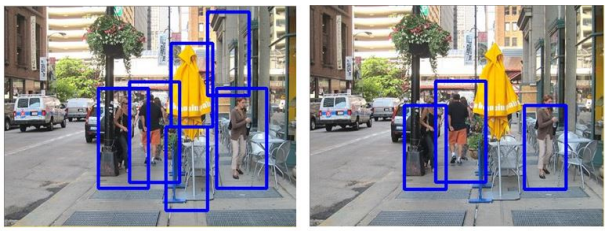

Figure 3: Reduction in false positive rates using hard negative mining. Poles and other erect patterns were eliminated in the new SVM model trained with the augmented negative dataset.

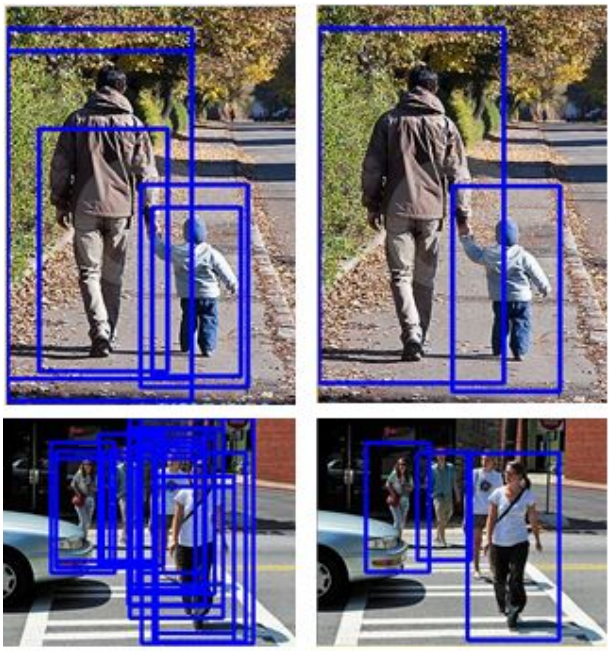

Figure 4: Redundant bounding boxes were eliminated using non-maximal suppression on these images.

To track the pedestrian in videos, after applying the background subtraction and getting the foreground mask, we found the contours for each frame and then computed the bounding boxes for each contour of that frame. Since all the training images have the size of 64 x 128, we re-sized the contours whenever their heights and widths were smaller than our training image sizes. This was accomplished by adding some padding to the heights of widths of the contours. Afterward, we applied our classifier on that contour to see if the contour is a pedestrian or not. We then used the non-maximal suppression technique to remove multiple bounding boxes for each object. Figure 6 shows the results for one frame of a video. As seen in this figure, there are six pedestrian in the frame and the classifier detected 3 of them. The others either are occluded or they are in a pose that the classifier can not detect. To observe if

Figure 5: Example of applying background subtraction on one frame of a video.

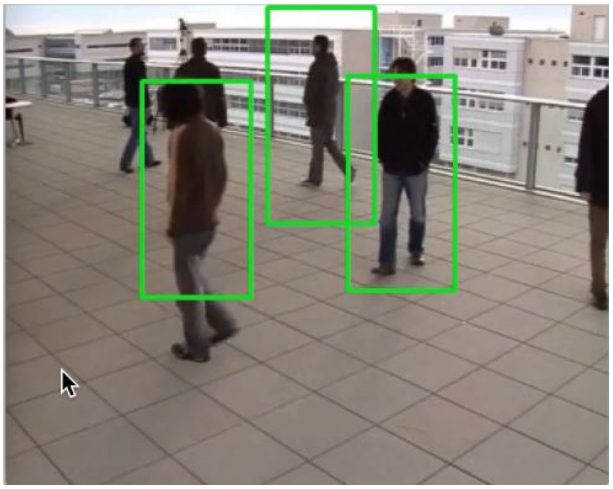

Figure 6: Pedestrian detecting and tracking in one frame of a sample video.

there are any confusion for the classifier when there are moving objects other than the pedestrians, we tested it on the videos that have different type of moving objects. Figure 7 is an example of a frame that contains pedestrians, motorcycles and a truck. As shown in the figure, the classifier only detected pedestrians and excluded the truck and motorcycles.

Furthermore, to evaluate our classifier, we tried 5 different videos with different duration. The total number of the pedestrians in these videos was 40 and our classifier detected 24 of them which means it has 60% accuracy. As a side note, we should mention that finding videos that have both moving pedestrians and moving non-pedestrian objects was difficult since since the cameras of most videos were not fixed and so the detection was not possible. We instead evaluated our classifier on a few number of videos. To watch a complete demo of the performance of our classifier, please refer to the Youtube link that we have provided in section 6.

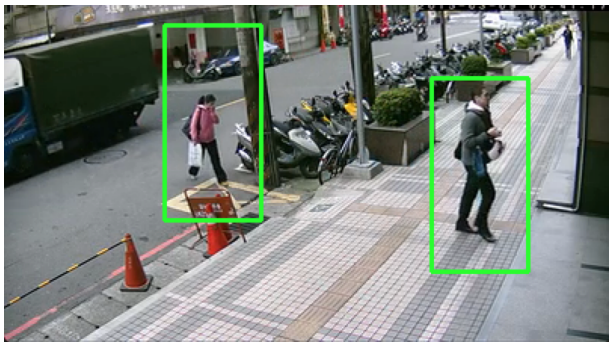

Figure 7: Pedestrian detecting and tracking in a frame with different type of moving objects.

## 5. CONCLUSION AND FUTURE WORK

We have demonstrated that HOG feature descriptors combined with SVM or Random forest and negative hard mining provides an effective strategy for pedestrian detection and tracking. The main draw backs of this approach is that our model is unable to track a large variety of human poses, and can only track pedestrians after some delay is added to the video due to the high complexity of HOG extraction. For many videos, occlusion is often present while pedestrians are moving in the scene, causing difficulties in detection. In the future, other techniques for pedestrian tracking can be added to our system such as optical flow and Kalman filtering. Tracking humans is inherently a difficult problem in the computer vision society, but solving this problem can greatly reduce the number of annual motor vehicle casualties, and reduce crime rates through improved surveillance systems at home and at work.

## 6. GITHUB AND YOUTUBE LINKS

Our GitHub Code:
`https://github.com/afazel/CS231A_Project`
Our Youtube Video on Pedestrian Tracking:
`https://www.youtube.com/watch?v=0lEJIh6dWAE`

## References

[1] *2014 Motor Vehicle Crashes: Overview*. U.S. Department of Transportation, March 2016.

[2] http://pascal.inrialpes.fr/data/human/

[3] http://mmlab.ie.cuhk.edu.hk/projects/PETA.html

[4] http://cbcl.mit.edu/software-datasets/PedestrianData.html

[5] S. Munder and D. M. Gavrila. *An Experimental Study on Pedestrian Classification*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no. 11, pp.1863-1868, November 2006

[6] Dalal Navneet, Triggs Bill *Histograms of Oriented Gradients for Human Detection*. International Conference on Computer Vision & Pattern Recognition - June 2005.

[7] Felzenszwalb Pedro, Girshick Ross, McAllester David and Ramanan Deva. *Object Detection with Discriminatively Trained Part Based Models*.

[8] Nan Lu, Jihong Wang, Q.H. Wu and Li Yang *An Improved Motion Detection Method for Real-Time Surveillance* . Annalen der Physik, 322(10):891921, 1905.

[9] Nan Lu, Jihong Wang, Q.H. Wu and Li Yang *Histograms of Oriented Gradients for Human Detection*. IAENG International Journal of Computer Science, 35:1.

[10] Zickler Stefan, and Efors Alexei *Detection of Multiple Deformable Objects using PCA-SIFT*. Carnegie Mellon University. 2007.

[11] Canyameres Masip, Sergi, and Antonio Manuel Lpez Pea *On the use of Convolutional Neural Networks for Pedestrian Detection*. 2015.

[12] Szarvas, M., Yoshizawa, A., Yamamoto, M., and Ogata, J. *Pedestrian detection with convolutional neural networks*. Intelligent Vehicles Symposium, 2005.