# Pedestrian Detection in Low Resolution Videos

Hisham Sager
Colorado School of Mines
Golden, CO 80401
hsager@mines.edu

William Hoff
Colorado School of Mines
Golden, CO 80401
whoff@mines.edu

## Abstract

*Pedestrian detection in low resolution videos can be challenging. In outdoor surveillance scenarios, the size of pedestrians in the images is often very small (around 20 pixels tall). The most common and successful approaches for single frame pedestrian detection use gradient-based features and a support vector machine classifier. We propose an extension of these ideas, and develop a new algorithm that extracts gradient features from a spatiotemporal volume, consisting of a short sequence of images (about one second in duration). The additional information provided by the motion of the person compensates for the loss of resolution. On standard datasets (PETS2001, VIRAT) we show a significant improvement in performance over single-frame detection.*

## 1. Introduction

Pedestrian detection in images or video is an important area of research, and has many commercial applications. One scenario is the case of stationary outdoor surveillance cameras, which are mounted in a high position and look down upon a large public area such as a street or plaza. In these scenarios, the size of pedestrians in the images is often small, and detection can be challenging.

The performance of current pedestrian detection approaches drops as resolution decreases. According to evaluations of the state of art pedestrian detectors (*e.g.* [1] and [2]); detection performance degrades rapidly at far scales; *i.e.*, where pedestrians are 30 pixels tall or less. In this case, nearly all pedestrians are missed by even the best detectors. One way to compensate for the loss of information due to low resolution is to use a sequence of images for detection. Motion information is a powerful cue for recognition. For example, Figure 1 shows a motivating example for the work described here. In a single low resolution frame, it is difficult to identify the object in the image, but it is much easier in a sequence of images in which a subject is performing a recognizable movement; *i.e.*, walking.

Figure 1: Sequence of images from a low resolution video of a walking person.

In this work, we propose a method to detect moving pedestrians in low resolution videos taken by stationary outdoor surveillance cameras. We form a spatiotemporal volume, consisting of a short sequence of images (about one second in duration). We extract gradient-based features from this volume, and train a support vector machine classifier to recognize people from the feature vectors. On standard datasets we show a significant improvement in performance over single-frame detection.

The remainder of this paper is as follows. We discuss previous work in Section 2. The proposed method is presented in Section 3. Section 4 gives a detailed description of datasets, experiments, discussion and evaluation. The conclusion and future work are summarized in Section 5.

## 2. Previous Work

There is an extensive body of literature on pedestrian detection. Most work focuses on pedestrian detection in single high resolution images. Instead of an explicit model, an implicit representation is learned from examples, using machine learning techniques. These approaches typically extract features from the image and then apply a classifier to decide if the image contains a person. Typically, the detection system is applied to sub-images over the entire image, using a sliding window approach. A multi-scale approach can be used, to handle different sizes of the person in the window.

The most common and successful approaches for single frame pedestrian detection use gradient-based features. The Dalal-Triggs detector [3] used a histogram of oriented

gradient (HOG) features to detect people. Felzenszwalb, *et al*. introduced the idea of deformable part filters for detection [4].

Most work on pedestrian recognition focuses on detection with cameras mounted on moving cars. In this scenario, pedestrians can appear at a wide range of sizes in the image. Park, *et al.* addressed the problem of detecting pedestrians at multiple resolutions [5]. They integrated a rigid HOG-based template for low resolution, with a deformable parts model for high resolution. They found that the part-based model is not useful for pedestrian heights less than 90 pixels.

Contextual information can improve recognition, since in traffic scenes pedestrians are often around vehicles [6]. Our work does not use contextual information, since we wanted to make our approach more general and not limit our domain to traffic scenes.

In our domain, which is the detection of pedestrians using stationary outdoor surveillance cameras, we can use the additional information provided by image sequences to improve performance. One approach is to check 2D object detections for consistency with scene geometry and convert them to 3D tracks [7]. Other methods applicable to tracking in image sequences include active contours, particle filters, and level sets, as well as intensity-based techniques often applied to tracking faces and whole bodies [9].

Other approaches use features that are similar to Haar wavelets [8-11]. Viola and Jones [9] popularized this approach and showed its applicability to face detection. The features are differences of rectangular regions in the images. These are simple and very fast to compute. Although each feature is not very discriminatory, a large number of features can to be chained together to achieve good performance. In [10] Viola and Jones use Haar-like wavelets to compute features in pairs of successive images for pedestrian detection.

Jones and Snow [11] extended the above algorithm to make use of 10 images in a sequence. This algorithm is the closest one to our approach, since it uses a relatively long sequence. They used two types of Haar-like features: Features applied within each frame, and differences of features between two different frames. On the PETS2001 dataset, their detector achieves a detection rate from 84% to 93% with a very low false positive rate of $10^{-6}$. They were able to detect pedestrians down to a size of 20 pixels tall.

To get better performance, one might try to extend the Jones and Snow method to work on longer sequences of images. However, in this case the number of potential Haar-like features grows to an unmanageable amount. Because of the large number of feature hypotheses that need to be examined at each stage, the training time can be quite slow (in the order of weeks).

Another approach to pedestrian detection in image sequences is to extract local features from the spatiotemporal volume of images. The work of [12] extracts spatiotemporal interest points. The interest point detector is composed of a 2D Gaussian smoothing kernel, applied along the spatial dimensions, and a quadrature pair of 1D Gabor filters applied temporally. This work was developed to recognize actions in videos. Conceivably these approaches could be adapted to detect pedestrians instead. However, in low resolution image sequences, it would be difficult to extract local features, since the volume is so small.

In summary, existing approaches for pedestrian detectors do not perform well for very low resolutions (*i.e.*, less than 30 pixel tall pedestrians). One exception is the Jones and Snow algorithm, which detected people down to 20 pixels tall. Our approach, described in the next section, is to use a sequence of images for detection (as Jones and Snow), but over a relatively longer time period. Instead of 10 images, we utilize up to 32 images. The additional information provides better detection performance, as described in Section 4.

## 3. The Method

In the proposed algorithm, we extract HOG features from a volume of images containing up to 32 frames. This corresponds to 1 to 2 seconds, depending on the camera frame rate. This time duration is enough to capture an appreciable fraction of a gait cycle, for normal walking speeds.

The proposed detector follows a sliding window paradigm which entails feature extraction, binary classification, and non-maximum suppression. A multi-scale approach is used, to handle different sizes of the person in the window. The algorithm has 3 primary components: (1) formation of spatiotemporal volumes, (2) feature extraction, and (3) classification. These are described in the subsections below.

### 3.1. Formation of Spatiotemporal Volumes

To form spatiotemporal volumes, we extract subwindows (or "slices") from the video, at a fixed position in the image, for up to 32 frames. Thus, each volume contains up to 32 slices, representing about 1 second of motion, depending on the frame rate. The slice window size was chosen to be 32 × 32 pixels. This size is large enough so the pedestrian remains within the window throughout the sequence, at normal walking speeds. The detector is trained to detect pedestrians with a height of approximately 20 pixels. This allows a border of 6 pixels in width around the pedestrians.

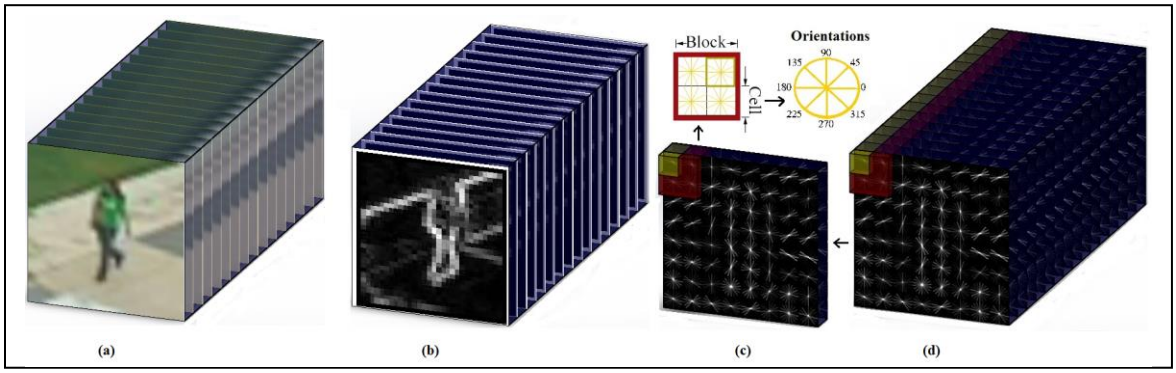


Figure 2: Spatiotemporal Volume. (a) Volumetric positive example. (b) Gradient. (c) Computed HOG for one slice. (d) Volumetric HOG descriptor; with block (shown in red color), and cell (shown in yellow color)

Figure 2(a) shows an example of a spatiotemporal volume of images. The extraction process is repeated at multiple scales. We used a pyramid consisting of 6 levels, where each level of the pyramid differs from the previous level by a factor of 0.75.

### 3.2. Feature Extraction

The feature extraction method is then applied to the series of slices that make up the volume (Figure 2). It starts by dividing each 32 × 32 pixel slice into square cells (typically 4 × 4 pixels each), and computes a histogram of gradient directions in each cell. We use 9 bins for the gradient directions, which represent unsigned directions from 0°-180°.

Following the method of Dalal [3], cells are grouped into blocks, where each block consists of 2x2 cells. Blocks may overlap; *i.e.*, they may share cells. We discuss the benefits of overlapping blocks in Section 4. We normalize the gradients within each block. Feature vector normalization improves accuracy and makes them more invariant to changes in illumination or shadowing. Next, the features from all the blocks in all slices are concatenated into a single volumetric feature vector.

The feature vector size is determined by the number of blocks in each slice and the number of slices per volume. Using cells of size 4x4 pixels, with no overlap between blocks, the feature vector size is 576 features multiplied by the number of slices. If overlap is allowed, there are more blocks in each slice and the feature vector size is correspondingly larger.

Following the method of Felzenszwalb [4], we use principal components analysis (PCA) to reduce the dimensionality of the features. Feature vectors are transformed to principal component space, and only those principal components which account for the most variance in the data are kept. Using lower dimensional features produces models with fewer parameters, which speeds up the training and detection algorithms, while keeping detection performance about the same. In the learning stage, we collect a large number of 36-dimensional HOG features (*i.e.* for each block) and perform PCA on them. The eigenvalues indicate that the linear subspace spanned by the top *n* eigenvectors (typically 10 to 15) can capture the essential information in the features.

### 3.3. Classification

The final step is to develop a recognition system using supervised learning methods. In this work, we trained a support vector machine (SVM) classifier. The SVM classifier is a binary classifier that looks for an optimal hyperplane as a decision function. Once trained on image sequences containing both positive and negative examples of pedestrians, the SVM classifier can make decisions regarding the presence of that object in a test image sequence. We used a freely available SVM-based classifier (the OSU-SVM MATLAB toolbox) for development and testing. K-fold cross-validation is used for parameter selection, by partitioning the training data into 5 equally sized segments, and then iterations of training and validation are performed to pick the best parameters for the SVM kernels. We experimented with two kernels – a linear kernel and a radial basis function kernel. Although the non-linear kernel gives slightly more accurate results, for simplicity and speed we use the linear kernel as the baseline classifier throughout this study.

Figure 3 shows the result of weighting the HOG descriptor of the example in Figure 2(d) by positive SVM weights. The classification decision is based on the result in this figure. As can be seen, the strongest features correspond to the general outline of the person.

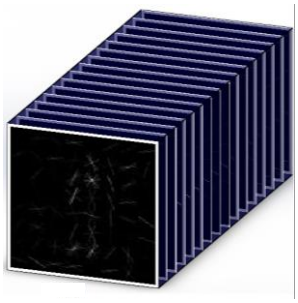

Figure 3: The HOG descriptor of the example of Figure 2 (d) weighted by the positive SVM weights.

## 4. Experiments and Results

We chose two standard datasets (PETS2001 and VIRAT) to evaluate our algorithm. These datasets contain images taken from stationary surveillance cameras, since that was our target application. Also, the PETS2001 dataset was used by Jones and Snow, and we wanted to compare our results to theirs, since their algorithm was the closest one to our approach. In both datasets, data was partitioned into testing and training sets.

To extract positive examples from the training videos, the following procedure was followed. A pedestrian was selected in one of the images and a square subwindow was extracted from the image, surrounding the pedestrian. This subwindow was scaled such that the person was 20 pixels tall, and the subwindow size was 32 × 32 pixels. Next, a sequence of subwindows was extracted from the images following this image, at the same fixed place in the image, and the subwindows were similarly scaled. A total of 32 such slices were assembled into a spatiotemporal volume, representing a single positive example. We placed the starting position of the window to ensure that the person remained in the 32x32 window throughout the duration of the 32 slice sequence. Negative examples were also extracted from the training images. These were spatiotemporal volumes of the same size as the positive examples, but sampled randomly from person-free areas of the scene.

We then applied the detector to the test videos. In principle, the detector should be applied at each pixel, in each image. However, our prototype system was relatively slow and it was inconvenient to apply it at every pixel. Instead, we applied it to randomly selected points in the image where pedestrians appeared, as well as person-free points. Speeding up our implementation was beyond the scope of this work; however, in principle it should be able to run in close to real time, since the operations it uses are similar to other real-time systems based on HOG features and SVM classifiers.

### 4.1. PETS2001 Dataset

The PETS2001 dataset contains 16 video sequences of length of about 2 to 4 minutes each, with a frame rate of 25 frames/second, and frame size of 768 pixels in width and 576 pixels in height. Half of the videos are designated as training, and half as testing. These sequences are taken by a stationary camera mounted on a high vantage point. It looks down upon a street and parking lot in front of a building. Cars and pedestrians periodically move through the scene.

We doubled the size of the dataset by flipping all frames of each video, *i.e*. taking the mirror image of each frame. The training set consisted of 26,000 positive slices, and 26,000 negative slices. Figure 4 shows some examples of positive sequences. Each sequence represents 1.28 seconds of activity.

Figure 4: Positive examples from the PETS2001 dataset, sub-sampled from 32 frame sequences.

We applied our detector to a test set, consisting of the same number of positive and negative examples as in the training set; *i.e.* about 26,000 positive examples and 26,000 negative examples. Figure 5 shows an image from this test set, with some detection examples.

Using the detector with parameters tuned for the best performance, we achieved a detection rate of 96% with a false positive rate (FPR) of $10^{-6}$. Detection rate is defined as

$$DR = \frac{TP}{TP + FN}$$

where *TP* is the number of true positives and *FN* is the number of false negatives. At the same FPR, the Jones and Snow detector [11] achieved a detection rate of 93% on the same dataset.

### 4.2. VIRAT Dataset

The second dataset is the VIRAT dataset. From this

dataset, we used 30 video sequences of length of about 0.5 to 5 minutes each, with a frame rate of 30 frames/second, and frame size of 1280 pixels in width and 720 pixels in height.

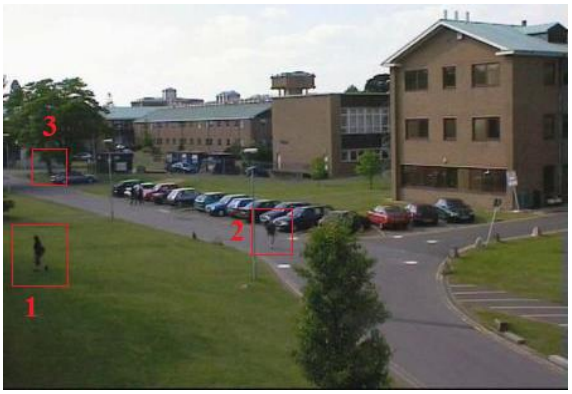


Figure 5: Example detections, PETS2001 dataset. (1) and (2) are true positives; (3) is a false positive.

Similar to the PETS2001, the sequences are taken by a stationary camera mounted on a high vantage point. It looks down upon a scene containing a street and parking lot. Cars and pedestrians periodically move through the scene.

The training set consisted of 32,000 positive slices, and 32,000 negative slices. Positive and negative examples were extracted in the same way described for the PETS2001 dataset. Figure 6 shows some examples of positive sequences. Each sequence represents 1.06 seconds of walking.

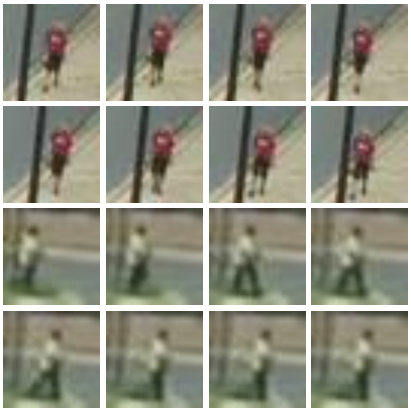

Figure 6: Positive examples (VIRAT dataset) sub-sampled from 32 frame sequences.

We applied our detector to a test set, consisting of the same number of positive and negative examples as in the training set. Figure 7 shows an image from this test set, with some detection examples.

On the VIRAT dataset, our best tuned detector achieves a detection rate of 93% with a false positive rate of $10^{-6}$.

### 4.3. Performance Study and Discussion

We studied the effects of the choices of various detector parameters on the performance.

Figures 8 and 9 show the effect of the number of slices per volume on the detection accuracy, for the two different datasets. Accuracy is defined as

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}.$$

The results show that including more slices improves the accuracy significantly, up to 20% over just processing a single frame. The case where the number of slices is equal to one represents the standard single frame pedestrian detector method.

The improvement increases with the number of slices, until a total of 16 slices is reached. After that, adding more time duration does not improve the accuracy. One possible reason is that for the datasets that we used, there is enough motion in 16 frames for the classifier to tell whether the tested example is a pedestrian or not.


Figure 7: Example detections, VIRAT dataset. (1) is a true positive; (2) is a false negative; (3) is a false positive.

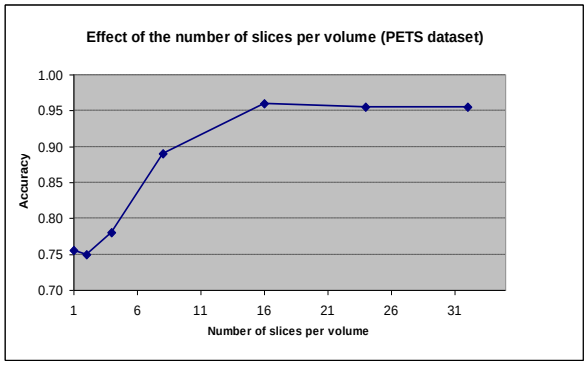


Figure 8: The effect of the number of slices per volume on detection accuracy, for PETS2001.

Figure 10 shows the effect of cell size on detection accuracy. The experiments show that a smaller cell size outperforms the use of a large cell size. In very low resolution images, a small cell size may be better able to capture the details of a pedestrian's body.

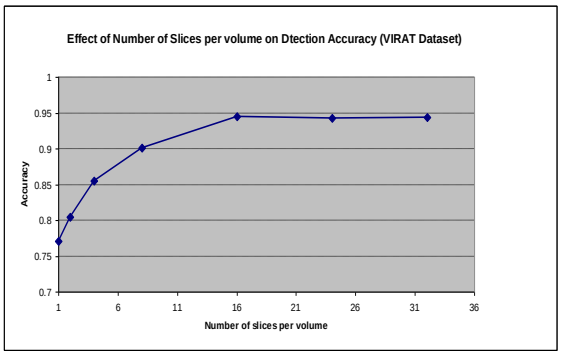

Figure 9: The effect of the number of slices per volume on detection accuracy, for VIRAT.

In addition, our experiments confirmed that local normalization is essential for good performance. Normalization over blocks improves the accuracy detection by a rate of 3% (*i.e*, from 93% to 96% for the PETS2001 database). In addition, normalization over slices (instead of blocks in the standard HOG) improves the accuracy of detection by 2% over normalization over blocks.

We evaluated detector performance with different block overlapping schemes (*i.e.*, 0.5 and 0.75 overlapping factor). The experiments show that the use of overlapping blocks in the descriptor improves performance by around 4% (*i.e*, from 92% to 96% for the PETS2001 database). Overlapping the blocks allows a feature to contribute to the decision more than one time, whereas if there are no overlapping blocks, a cell is coded only once in the final descriptor.

The results described in Section 4.2 were obtained using a cell size of 4x4 pixels, block overlap of 0.75, and 16 slices per volume.

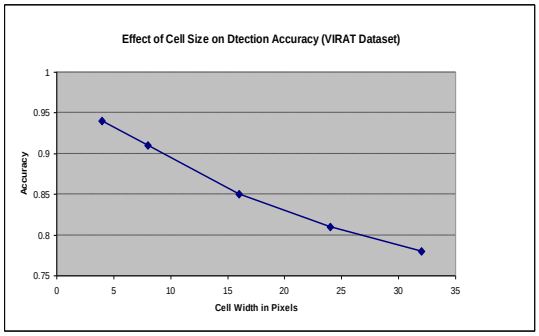

Figure 10: The effect of cell size on detection accuracy.

## 5. Conclusions and Future Work

We have shown that pedestrians can be detected in low resolution video if their motion is viewed over a sufficiently long sequence of images. The duration should be long enough to capture a complete gait cycle; *i.e.*, typically 1 second or more should be long enough. We have built a normalized volumetric gradient-based feature set that allows pedestrians to be discriminated in low resolution videos, where the size of the pedestrians is only about 20 pixels tall. By using sequences long enough to contain a full gait cycle, almost 20% improvement over single frame detection can be obtained.

We studied the effect of various parameters on the detector performance, and found that including more frames in feature vector improves the performance significantly, up to about 16 frames. On a standard dataset, our detector has a better detection rate than a previously published detector that uses Haar-like features [11], at the same low false positive rate.

In future work, we plan to extend our detector so that it can work with non-stationary cameras; particularly on aerial image sequences. The duration of a second or so is reasonable for aerial imagery - even though the camera may be moving, a person is often in the field of view during that time duration. Since our current algorithm extracts slices that are stationary with respect to the ground, we will need to register the images to compensate for the motion. Finally, we plan to optimize our implementation so that it can run in close to real time.

## 6. References

[1] P. Dollár, *et al*. "Pedestrian detection: A benchmark." In CVPR, pp. 304-311, 2009.

[2] P. Dollár, *et al*. "Pedestrian detection: An evaluation of the state of the art." In PAMI, pp. 743-761. 2012.

[3] N. Dalal, *et al*. "Histograms of oriented gradients for human detection." In CVPR, pp. 886-893, 2005.

[4] P. Felzenszwalb, *et al*. "A discriminatively trained, multiscale, deformable part model." In CVPR, pp. 1-8. 2008.

[5] D. Park, *et al*. "Multiresolution models for object detection." In ECCV, pp. 241-254. 2010.

[6] J. Yan, *et al*. "Robust multi-resolution pedestrian detection in traffic scenes." In CVPR, 2013.

[7] B. Leibe, *et al*. "Coupled object detection and tracking from static cameras and moving vehicles." PAMI, Transactions on, vol. 30, no. 10. pp. 1683-1698, 2008.

[8] C. Papageorgiou, *et al*. "A trainable system for object detection." Int'l Journal of Computer Vision vol. 38, no. 1. pp. 15-33. 2000.

[9] P. Viola, *et al*. "Robust real-time face detection." Int'l Journal of Computer Vision 57, no. 2 (2004): 137-154.

[10] P. Viola, *et al*. "Detecting pedestrians using patterns of motion and appearance." International Journal of Computer Vision 63, no. 2 (2005): 153-161.

[11] M. Jones, *et al*. "Pedestrian detection using boosted features over many frames." In ICPR, pp. 1-4., 2008.

[12] P. Dollár, *et al*. "Behavior recognition via sparse spatio-temporal features." In Visual Surveillance and Performance Evaluation of Tracking and Surveillance Workshop, pp. 65-72. 2005.