Simple Real-Time Human Detection Using a Single Correlation Filter

David S. Bolme Yui Man Lui Bruce A. Draper J. Ross Beveridge
Computer Science Department
Colorado State University
Fort Collins, CO 80521, USA
{bolme,lui,draper,ross}@cs.colostate.edu

Abstract

This paper presents an extremely simple human detection algorithm based on correlating edge magnitude images with a filter. The key is the technology used to train the filter: Average of Synthetic Exact Filters (ASEF). The ASEF based detector can process images at over 25 frames per second and achieves a 94.5% detection rate with less than one false detection per frame for sparse crowds. Filter training is also fast, taking only 12 seconds to train the detector on 32 manually annotated images. Evaluation is performed on the PETS 2009 dataset and results are compared to the OpenCV cascade classifier and a state-of-the-art deformable parts based person detector.

1. Introduction

One of the simplest ways to detect targets in images is to convolve an image with a filter or template that responds to the target. The output of the convolution should produce a large response where the target is present and a suppressed response over the background. Targets are then detected where the convolution output exceeds a threshold. The primary advantages of this approach is that it is extremely simple and very fast.

The success of the filter-based object detection depends on the ability of the filter to distinguish between targets and background. A typical way to produce a filter is to crop a template of the target from a training image. Unfortunately, templates based on one image often do not capture appearance variation adequately and therefore only perform well in highly controlled object detection scenarios. To compensate, there are a number of techniques to produce filters from large numbers of templates and therefore more accurately represent targets appearance. For example, a filter can be produced by averaging templates. Unfortunately, such a filter often fails to adequately discriminate between targets and background.

More sophisticated methods based on Synthetic Discriminant Functions (SDF)s [13] can also be used to produce filters that respond well to the training templates and produce sharp and stable peaks. One problem with SDFs is that they do not consider the entire convolution output during training. Instead they emphasize only one point in the output when the filter is aligned with the target. These techniques emphasize good peaks for targets but have much less control when it comes to suppressing peaks for background objects with similar appearances.

Recently, a new concept for training filters was introduced called Average of Synthetic Exact Filters (ASEF) [3]. ASEF considers the entire output of the filter under a full convolution operation. By exploiting the Convolution Theorem, ASEF provides a mechanism where the entire output for a full training image can be specified. Producing an ASEF filter is much more like deconvolution than prior techniques. In [3] it was shown that ASEF filters were much better at locating eyes on a face because the filters were much better at suppressing the response of other facial features. This study will show that ASEF filters are able to produce good target/background separation on a more general detection problem, namely the PETS 2009 dataset[8].

Detectors based on ASEF filters have many advantages. Training only requires a small number of hand annotated images and a few seconds of computation time. The resulting detector is tuned specifically to the camera setup. Detection is much simpler than competing techniques and based on the highly regular convolution, which means that it is ideally suited for embedded systems or existing signal processing chips. Filter-based detection is many times faster than competing techniques, while its accuracy is comparable or better.

The rest of this paper is organized as follows. Section 2 discusses other person detection techniques and how they relate to the work presented here. Section 3 discusses the process of creating a filter based detector and the method used to learn the ASEF filter. Section 4 compares the filter based detector to a morphable parts based approach and a cascade based classifier. Section 5 summarizes the findings.

978-1-4244-5504-1/09/$25.00 ©2009 IEEE

2. Related Work

This paper compares the ASEF filter to two publicly available detectors. The first detector is based on the Viola and Jones cascade classifier. This classifier is interesting because it is a good object detection algorithm and is fast enough for real time systems[14]. The original context of this work was in the area of face detection. Viola *et. al.* also adapted this algorithm to the problem of people detection [15]. In that study, detection was based on both visual features and motion features computed between video frames. The detector was also fast enough for real time detection, reporting a speed of 4 frames per second. In this paper the OpenCV[16] implementation of the cascade detector was retrained on the PETS data with good results.

The second detector is based on a deformable parts model is based on the work of Felzenszwalb *et.al.* [7]. This detector adopts many ideas from [5] such as Histogram of Oriented Gradient (HOG) based features and using and used a Support Vector Machine (SVM) like classifier. The primary improvement of this method is that it also uses deformable parts models in addition to holistic matching to improve detection accuracy. While accurate, this detector is too slow for real time detection and takes a few seconds to process each frame.

We also briefly investigated the person detector from [5]. This method is simpler than the Parts Based model and carefully investigated HOG based features as a basis of person detection. The performance of the detector seemed to be similar to [7], but was also slower.

In [9], the problem of accurate object detection in crowded scenarios is discussed. Leibe *et.al.* point out that many pedestrian detection techniques have been evaluated on isolated people and as a result those detectors often fail in crowded or complex real world situations. They propose an iterative detection system that both detects and segments people in a crowded scene. They also suggests that partial occlusions in crowded scenes may be too difficult for detectors based on simple features or models. In this work, we have seen evidence to the contrary. The simple ASEF filter based detector handled partial occlusion better than the more complex Part Model based detector in many situations. However, all the detectors tested in this paper failed as the crowd density increased. In those cases segmentation is a probably a better strategy for locating individuals in those dense groups.

A number of detection methods have already been presented on the PETS 2009 dataset[6], however most of those rely on tracking or multiple views to more accurately locate people in the presence of crowds or occlusion. Tracking and detection are not contradictory. In fact, detection is often an important part of tracking systems and is typically used to initialize or maintain tracks. The work discussed here focuses on improving detection in individual images.

Stalder *et.al.* [12] test a single frame detector, which offers good solutions to the difficulties encountered in the PETS 2009 dataset: namely occlusion and dense groups. Instead of scanning the image using a single detection function, they learn a separate classifier function for every location in the image. While this improves the detection accuracy by simplifying the classification task, it is also not a generalized detection algorithm such as the ASEF filter detector discussed here.

3. Methods

To detect people in video, thousands of detection windows must be evaluated every second. The vast majority of those windows correspond to background and must be disregarded. Consequently, a classifier must be constructed that reliably rejects the vast majority of detection windows presented to it while simultaneously avoiding the mistake of rejecting windows in which a person is actually present. This can be a major challenge.

In principle, it makes sense to train a detector on every possible detection window in every frame of labeled video. However, doing this for commonly used types of detectors such as the Viola Jones cascade classifier is often too computationally demanding. Instead, these algorithms are trained iteratively using boosting. While this type of boosted training is clever, and can after many iterations generate very good detectors, the process is hard to automate and in practice can be problematic.

In contrast, the techniques presented here lend themselves naturally to efficient training over every possible detection window. This is because the classifier is based upon convolution, and training can be accomplished efficiently by exploiting the Convolution Theorem. To be concrete the filter presented in this study was trained on 3,145,728 detection windows in under 12 seconds. The rest of this section discusses the filter based detection algorithm including regions of interest, preprocessing, training the ASEF correlation filter, and finally filtering and detection.

3.1. Size Normalization and Preprocessing

One challenge in creating a filter based detector is the problem of scale changes. In the PETS2009 dataset, View001 the heights of people vary from a minimum of 50 pixels to a maximum of 150 pixels. This presents two problems. The first is for training, which assumes the people are approximately the same size. The second is in testing, where the filter needs to be applied at multiple scales. The solution to both these problems leverages the geometry of the camera setup. Because the camera is elevated and looking down at an approximately planar scene, the y location of the person is a good predictor of a persons height. The approach here is to divide the scene into regions with approximately constant scale (See Figure 1). The four regions are then rescaled so that people are approximately the same height in each. Figure 2 shows that the rescaled regions have much less variance in proportion to the average height than the full frame. These regions also focus the detection effort on the side walk which covers most of the action in the videos.[1]

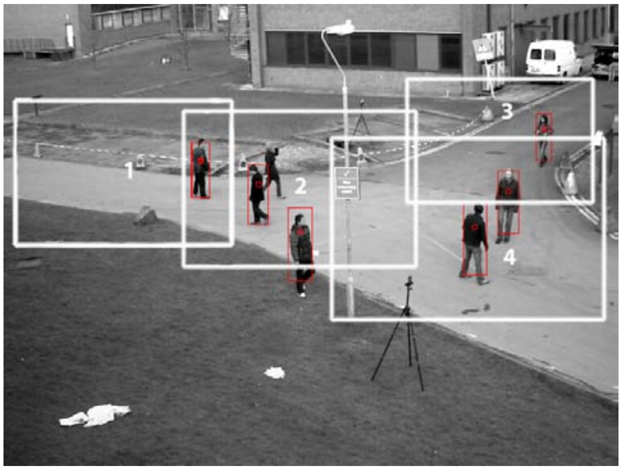


Figure 1: This image shows the four detection regions which cover the sidewalk.

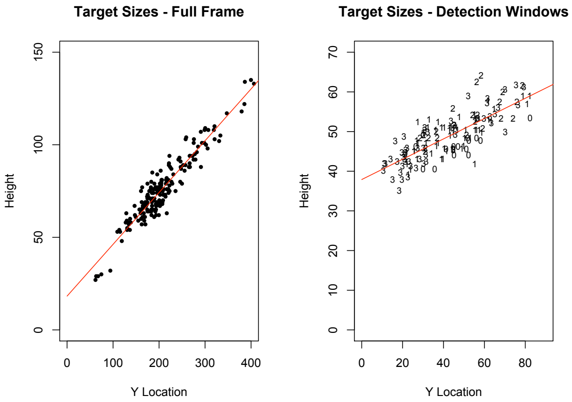
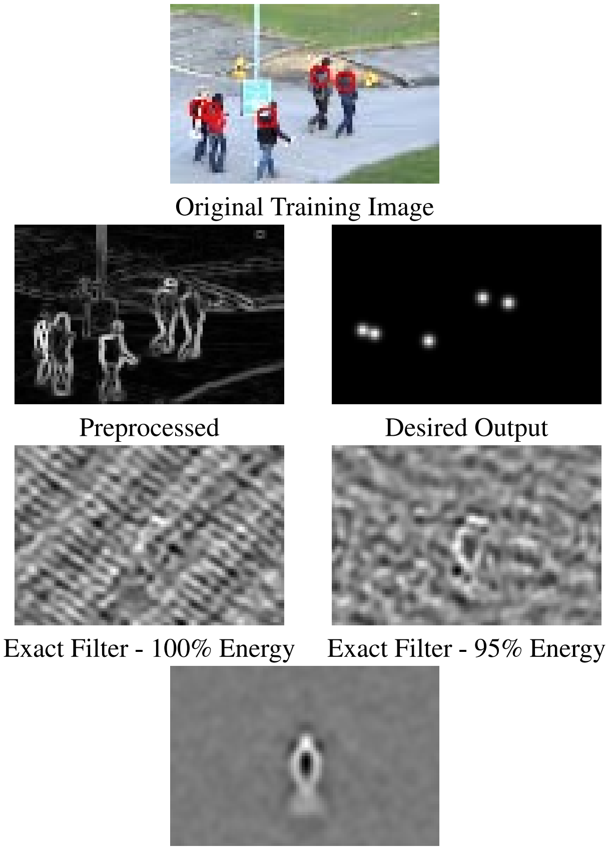


Figure 2: The left plot confirms that the height of people in the full frame can be approximated nicely using a linear model. This is expected given the approximately plainer ground. The right plot shows the height of those people in the rescaled detection regions. In the rescaled regions there is much less variation in the person height.

Another challenge stems from the fact that a person's appearance is greatly affected by clothing. Many detection algorithms solve this problem by focusing on gradient based features[17, 5, 11]. The gradient based features focus the detection process on edge information and detection is therefore less dependent on the absolute intensity of the pixels. Images are therefore preprocessed to produce the gradient magnitude for each pixel in the detection region using standard Sobel operators (See Figure 3). This step creates a new image where the people are defined primarily by there outline. The images are then value normalized by taking the log of the pixel values and scaling the image to have a mean value of zero and unit length in a manor identical to [3].

3.2. Training an ASEF Filter


Original Training Image

Preprocessed

Desired Output

Exact Filter - 100% Energy

Exact Filter - 95% Energy

ASEF Filter (Average of 512 Exact Filters)

Figure 3: This image shows the intermediate steps when computing an ASEF filter.

Examples of the filter training process can be found in Figure 3 and is discussed in detail in [3]. The filters are implemented in Python using the PyVision library[2]. ASEF filters learn the mapping from a source image to a target im-

[1] Having a uniform size also simplifies filter training and ensures that image is a size compatible with an FFT.

age. More formally, they take in an image $f \in \mathbb{R}^{P \times Q}$ and maps it to a new image $g \in \mathbb{R}^{P \times Q}$. That mapping is parameterized by a filter $h \in \mathbb{R}^{P \times Q}$ and transformation can be expressed as a convolution:

$$g = f \otimes h \tag{1}$$

When training a standard detector, each detection window is labeled as either being a positive example if a person is present or a negative example if it corresponds to background. In contrast, the ASEF filter is trained on complex scenes that contain both positive and negative samples. The entire image is labeled with peaks with a values of 1.0 where a person is present and a values of 0.0 for background. The ASEF process learns a mapping from the training images to the labeled outputs.

More formally, for each training image f_i, a synthetic output g_i will be generated which contains a peak for each person in the image. The peaks in g_i will take the shape of two dimensional Gaussians:

$$g_i(x, y) = \sum_p e^{-\frac{(x-x_p)^2+(y-y_p)^2}{\sigma^2}} \tag{2}$$

where (x_p, y_p) is the location of person p in the training image, and σ controls the radius of the peak.

Next, for each training image, an exact filter h_i will be computed which exactly maps the image f_i to g_i. This computation is efficient in the Fourier domain. The Convolution Theorem states that convolution in the spatial domain becomes an element-wise multiplication in the Fourier domain. Therefore the problem can be transformed from

$$g_i = f_i \otimes h_i \tag{3}$$

in the spatial domain, to

$$G_i = F_i \odot H_i \tag{4}$$

in the Fourier domain, where G_i, F_i, and H_i are the Fourier transforms of their lower case counterparts, and $\odot$ explicitly indicates an element-wize multiplication. The exact filter[2], H_i, can now be quickly computed by solving Equation 4:

$$H_i = \frac{G_i}{F_i} \tag{5}$$

where the division is also performed element-wise.

The resulting filter could be considered a weak classifier that performs perfectly on a single training image. It does not, however, generalize well to the larger dataset. As seen in Figure 3, the exact filter looks more like noise than a template that will respond to a person's outline. To produce a more general classifier, exact filters are computed for every training image and then averaged. The motivation for averaging exact filters can be found in the literature for bootstrap aggregation or bagging[4]. Aggregating a collection of simple filters converges on a filter that minimizes the variance error. A more intuitive way to think about the averaging process is that it keeps features that are consistent across many filters while averaging out features that are idiosyncratic to a single instance. Therefore, the final ASEF filter is computed as:

$$h = \frac{1}{N} \sum_{i=1}^{N} h_i = \frac{1}{N} \mathcal{F}^{-1} \left(\sum_i H_i \right) \tag{6}$$

where N is the number of training images. Averaging has some nice properties which makes training an ASEF filter fast and easy to compute: it does not overfit the training data, it only requires a single pass for each image, and it only requires enough memory to store one filter.

One limitation of ASEF is that it typically requires a large number of training images to converge to a good filter. Two methods were used to reduce the total number of frames required for training. The first, which was adopted from [3], is to use duplicate training images that have been perturbed by small scales, small rotations, small translations, and reflections. This increases the number of training images and also encourages the filter to be more robust to small changes in rotation and scale.

This paper introduces a second technique which improves the stability of the exact filters. In Equation 5, frequencies in the training image f_i that contain very little energy are weighted heavily in corresponding the exact filter. These frequencies can cause the exact filter to become unstable, and in the extreme case where the energy is zero cause a divide by zero error. To correct for this problem the exact filters are constructed using the largest frequencies in F_i that contain 95% of the total energy.

Removing the small frequencies appears to remove much of the “noise” in the exact filter (See Figure 3). In tests, this heuristic allowed ASEF filters to be trained on fewer images without adversely affecting their accuracy or appearance.

3.3. Filter Based Object Detection

Object detection using a filter is simple and fast. The four detection regions are rescaled and preprocessed using the same procedure as the training. This produces four gradient magnitude images which are convolved with the ASEF filter using the FFT method, as in Equation 2. The resulting correlation output should have a peaks where people

[2] The notation differs slightly from that in [3] which considered correlation instead of convolution and therefore used the notation H_i^*. The method here will learn a 'matched' filter which means that it will be flipped on both the x and y axes in the spatial domain. This flipping can also be performed by taking the complex conjugate in the Fourier domain. The filter will also be centered on the pixel (0,0).

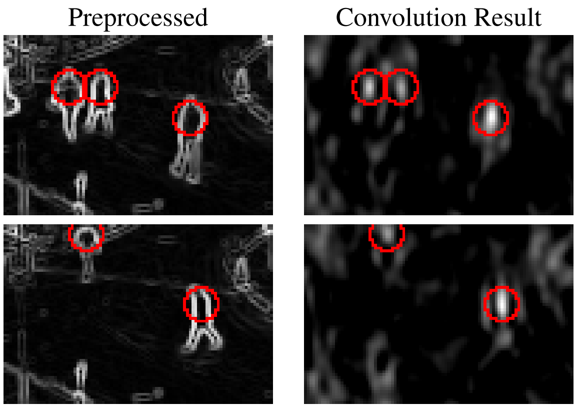


Figure 4: This shows the preprocessed images (left) and the result of the convolution with the ASEF Filter (right). Detected peaks are highlighted.

are present, and suppressed responses to image background. The correlation output is then scanned for local maxima. Any maxima that exceeds a user defined threshold is considered a detection. See Figure 4 for examples.

4. Results

To better understand the performance of the filter based detector a careful analysis was conducted on two sequences from the PETS 2009 test set. For all tests View_001 was used. The first sequence, S2L1_T1234, represents an easy detection problem where people tend to be sparsely distributed around the frame. The second sequence, S1L1_T1357, represent a more crowded scene with a moderately dense crowd walking from the right to left along the side walk.

As mentioned before, for evaluation purposes two detectors were used for comparison. The first is based on the well known Ada-Boost Cascade Object Detector [14]. The OpenCV implementation[16, 14, 10] was retrained on the PETS 2009 dataset with the same preprocessing that was used for the filter based method used in this paper. The only difference is that detection regions were rescaled to twice the size (196 × 128) because the cascade failed to accurately detect people on the small images used by the filter. The speed of this algorithm should compare to the person detector created by Viola *et.al.* [15]. This detector also provides a basis of comparison for good object detector trained on the same training data as the filter based detector.

Matlab code was also obtained for the parts based person detector described in [7] which evaluated very well for detection accuracy. This implementation was used ”out of the box” with no retraining and should therefore represent state-of-the-art performance in terms of accuracy.

4.1. Training Time

The ASEF filter is trained on 32 frames taken from View_001 of training sequence ”Time 14-03”. Each frame is divided into 4 detection regions, and each detection region is randomly perturbed 4 times. Thus, the ASEF filter is trained on 512 total images ($32 \times 4 \times 4 = 512$). Each training window is 96×64 pixels. This gives a grand total of 3,201,024 pixels or detection windows. Training took approximately 11.5 seconds running on an Apple MacBook Pro with a 2.4Ghz Intel Core 2 Duo processor. This included reading in the original frames, extracting and randomly permuting the detection windows, computing the ASEF filter, and writing the trained filter to disk.

A training time comparison between ASEF and the OpenCV cascade classifier can be found in Table 1. The Cascade training was terminated when it reached a time limit of six hours and had trained to a depth of 13 nodes.

Table 1: Total training time for ASEF and the OpenCV Cascade Classifier

Method	# Tiles	Size	Training Time (min:sec)
ASEF	512	96x64	0:12
Cascade	512	192x128	361:01

4.2. Detection Speed

The most obvious advantage of filter based detectors is the speed at which they can process images. Figure 5 compares the rate at which the detectors processed frames in the S1L1_T1357 sequence. The ASEF filter detector is the clear winner with an median rate of 25.37 frames per second. The Viola and Jones based Cascade comes in second with a median rate of 6.75 frames per second which is actually very close to the frame rate of the video (7 frames per second). The parts based method was much slower than real time and took on average 5.2 seconds to process each frame.

4.3. Detection Accuracy

The methods to measure detection accuracy were adopted from [6]. Accuracy is measured in terms of **correct** detections, **missed** detections, and **false** detections. To determine if a ground truth rectangle G and a detection rectangle D constituted a correct detection, a simple measure called overlap ratio was used:

$$overlap = \frac{|G \bigcap D|}{|G \bigcup D|} \tag{7}$$

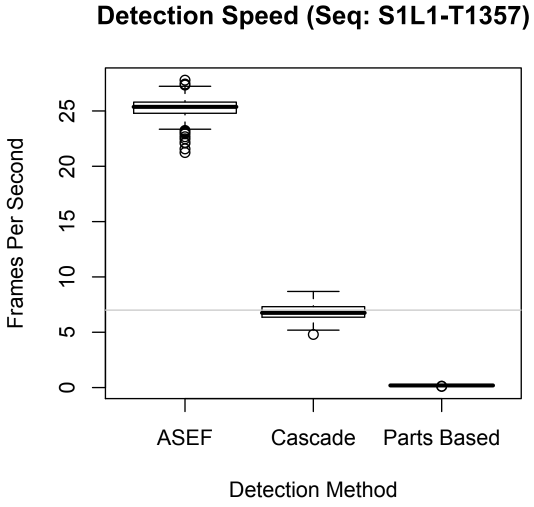


Figure 5: This shows detection speed for each algorithm for each frame in the S1L1_T1357 testing sequence.

In [6] an overlap ratio exceeding 0.60 was determined to be a successful detection. This criteria however presented a number of issues. The most important had to do with the size of the detection rectangles.

To obtain ground truth data, the center of the torso was manually located for each person in 50 randomly selected frames from each of the two video sequences. A detection rectangle was then approximated by estimating the height from the model shown in Figure 2. The width was also estimated to be approximately $1/3$ the size of the height. This rectangle size is a good approximation of the size of a person in a video sequence. This same method was used to estimate the detection size for every detection from the filter based detector. Under these circumstances the metric appears to adequately measure correct detections.

The Cascade and Parts Model detectors returned detection rectangles that were often much larger than the person, and therefore the overlap ratio was very low. The result was that the computed detection rates for both algorithms were very low even in the easy test sequence. When viewing images the algorithms were clearly detecting a majority of the people in the video. In order to measure the detection accuracy for these algorithms a number of adjustments were required. First, the detection rectangle were resized using the same model used by the ground truth and the ASEF detector. Second, an overlap threshold of 0.6 was found to be too high because of high variance in the positional accuracy of the Cascade and Parts Model detectors. Therefore, a threshold of 0.4 is used in this study which seems to more accurately reflect the performance of the detectors.

One other difficulty with the Cascade and Parts Based detectors is that they both scan a range in scale space, which produces false detections at inappropriate scales. Because the filter based method only considers a small range in scale space it is less prone to this type of error. To level the playing field, detection from those two algorithms were only evaluated if their height was between 0.7 and 1.3 times the estimated height of a person at that location. Likewise, the ASEF and Cascade detectors only searched for people in the region near the sidewalk. This gave the Parts based method an advantage because it detects people for the full frame. Therefore detections and truths were only evaluated if the lower center of the bounding box fell within the regions R1 and R2 for the people counting scenario.

A greedy method was used to associate detection regions with ground truth regions. Starting with the detection with the highest score, each detection region was matched to the truth region with the highest overlap if it exceeded the overlap threshold. This was counted as a correct detection and that detection and ground truth region was removed from consideration in further matches. Any detection or truth regions that were not matched were counted as errors, either a missed detection or a false detection. The correct detection rate is counted as the number of correct divided by the total number of truth, and false detections are reported as the average number of false detections per frame.

The results of the accuracy evaluation can be found in Figure 6. As expected, all the detectors perform well on the sparsely populated scenario. The ASEF detector has a clear advantage and achieves a 94.5% correct detection rate with less than one false detection per frame. This is 15% higher than the other two detectors. On the crowded scenario all the detectors do poorly. This is most likely due to overlapping people and occlusion making detecting individuals difficult. In this harder case the Parts Based detector has a small advantage with a detection rate of 50.7%.

For a more qualitative analysis we selected a threshold for each algorithm which produced no more than 1.0 false detection per frame on the easier sequence. To better understand the situations in which the detectors had difficulty we carefully studied the detection results for the ASEF and Part Model algorithms for each manually annotated frame. Some notable examples are shown in Figure 7. ASEF had very few missed detections in the sparse video sequence with the most common cause being partial occlusion. It also had difficulty when a person was partially outside of the detection regions. The Part Model algorithm had more difficulty with occlusion but also had difficulty detecting people near the top of the frame. This is probably because the size of the people were smaller than the lower bound on the detector. False detections for both algorithms seemed to be detections that were offset from the true location of a person, or a bad response to two or more people in close proximity.

For the densely packed crowd the most common issue was missed detections. In many cases one person was almost totally occluded by another. In these cases both detectors tended to produce only one detection and therefore the other person was missed. Both algorithms almost always missed some of the detections for densely packed groups of people and occasionally missed all of the detections in a group. The Parts Based model seems to have a higher detection rate in the densely packed groups which may account for its slightly higher detection rates in Figure 6. As stated before, the solution to this problem is probably not to produce better detectors but instead to segment those groups using methods similar to [9] or by adding trackers which can follow people through occlusion.

Table 2: This table shows the detection rates for 1.0 false detections per frame and the associated thresholds.

Algorithm	Recall	False Det.	Thresh
Sequence: S2L1_T1234			
ASEF	0.945	0.84	**0.1066**
Cascade	0.720	0.78	-
Parts Based	0.799	0.92	**-1.103**
Sequence: S1L1_T1357			
ASEF	0.469	0.90	0.1015
Cascade	0.520	*2.16*	-
Parts Based	0.507	1.00	-0.5464

5. Conclusions

This paper demonstrates that a simple filter based person detector can perform well on a difficult detection problem. The most notable feature of the detector is its speed which runs at over 25 frames per second. We have also compared that detector to the more sophisticated method of [7] and showed that results are comparable or better for the two scenarios tested.

This technique also has some limitations. The ASEF filter was trained specifically on View001. It remains to be tested whether a filter trained for one view will perform well for other cameras. Another problem is that the current technique does not scan multiple scales in the same way as the Cascade or Parts Based detectors. These issues will be studied in more detail in future work.

References

[1] Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11):1475–1490, 2004.

[2] David S. Bolme. Pyvision - computer vision toolkit. WWW Page: http://pyvision.sourceforge.net, 2008.

[3] David S. Bolme, Bruce A. Draper, and J. Ross Beveridge. Average of synthetic exact filters. In *CVPR*, pages 2105–2112, Miami Beach, Florida, June 2009.

[4] L. Breiman. Bagging Predictors. *Machine Learning*, 24(2):123–140, 1996.

[5] N. Dalai, B. Triggs, I. Rhone-Alps, and F. Montbonnot. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

[6] A. Ellis, A. Shahrokni, and J. Ferryman. Overall evaluation of the pets2009 results. In *PETS*, pages 117–124, 2009.

[7] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. *CVPR*, pages 1–8, 2008.

[8] J. Ferryman and A. Shahrokni. An overview of the pets2009 challenge. In *PETS*, pages 25–30, 2009.

[9] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *CVPR*, volume 1, pages 878–885, 2005.

[10] R. Lienhart, A. Kuranov, and V. Pisarevsky. Empirical Analysis of Detection Cascades of Boosted Classifiers for Rapid Object Detection. *Pattern Recognition: 25th Dagm Symposium, Magdeburg, Germany, September 10-12, 2003: Proceedings*, pages 297–304, 2003.

[11] O. Sidla, Y. Lypetskyy, N. Brandle, and S. Seer. Pedestrian detection and tracking for counting applications in crowded situations. In *IEEE International Conference on Video and Signal Based Surveillance, 2006. AVSS'06*, pages 70–70, 2006.

[12] Severin Stalder, Helmut Grabner, and Luc Van Gool. Exploring context to learn scene specific object detectors. In *PETS*, pages 63–70, 2009.

[13] B.V.K. Vijaya Kumar, A. Mahalanobis, and R.D. Juday. *Correlation Pattern Recognition*. Cambridge University Press, 2005.

[14] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *CVPR*, pages 511–518, 2001.

[15] P. Viola, M.J. Jones, and D. Snow. Detecting pedestrians using patterns of motion and appearance. *IJCV*, 63(2):153–161, 2005.

[16] Willow Garage. Opencv libarary, April 2009. `http://opencv.willowgarage.com`.

[17] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *ICCV*, volume 1, pages 90–97, 2005.

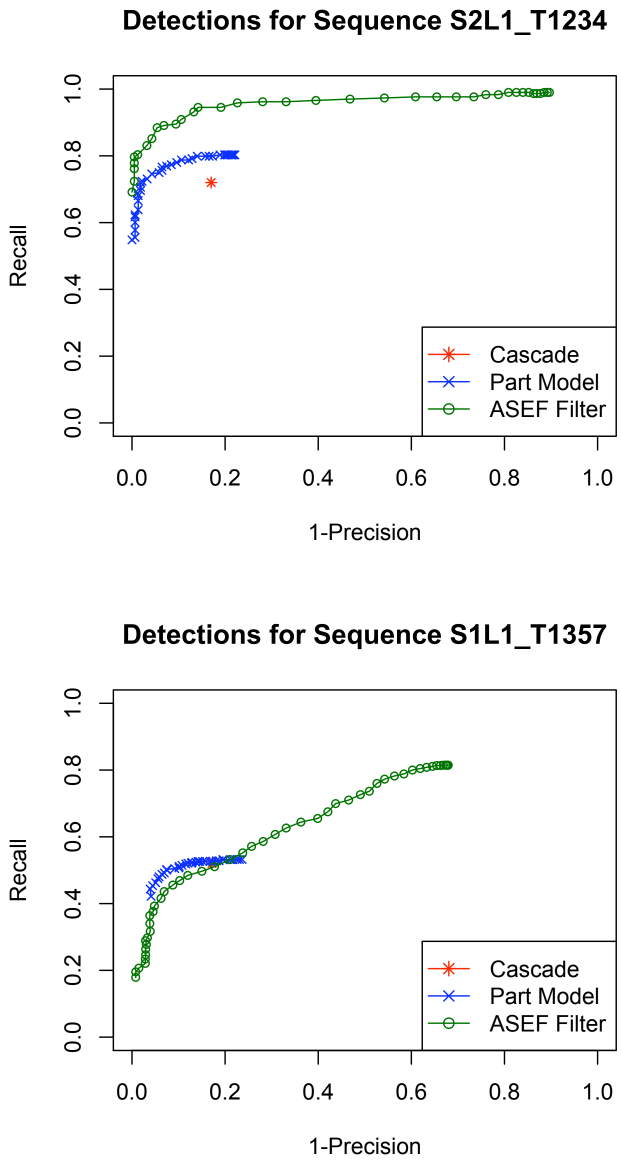


Figure 6: These plots show quanitative results of the detection algorithm using the standard Recall / 1-Precision curves suggested in [1]. The top plot shows that ASEF has a clear advantage in the sparse crowd of sequence S2L1-T1234. The bottom plot shows that all detectors have difficulty on the high density crowd of sequence S1L1-T1357.

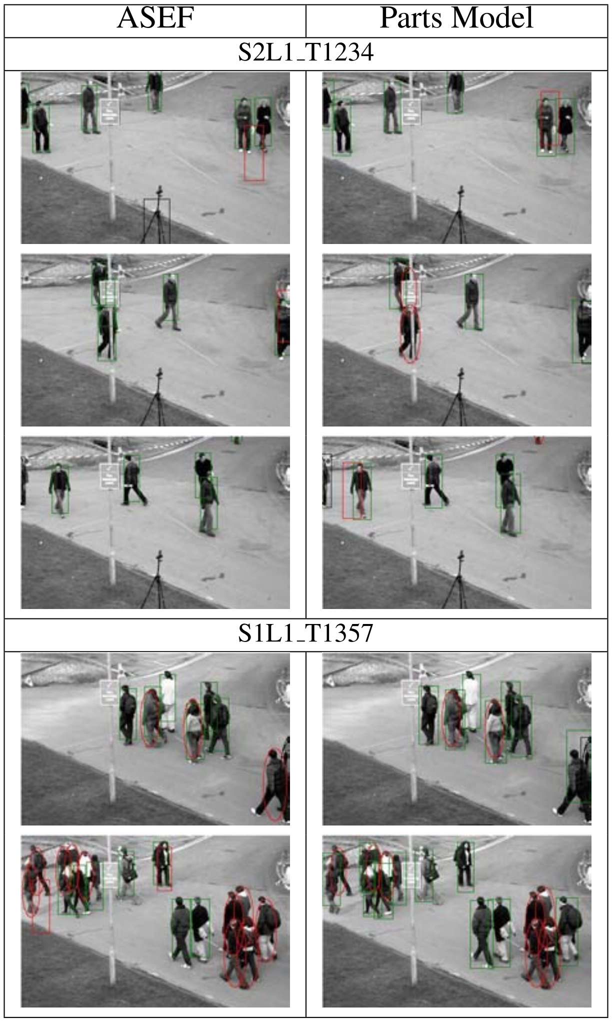


Figure 7: This figure shows some interesting detection challenges in the PETS dataset. Green rectangles indicate successes, red rectangles indicate false detections, and red ellipses indicate missed detections. From top to bottom: example false detections from people walking in close proximity, occlusion from the background, partial occlusion from by people, total occlusion by people, and densely packed groups of people.