Detection of Pedestrians at Far distance.

Rudy Bunel, Franck Davoine, Philippe Xu

Detection of Pedestrians at Far Distance

Rudy Bunel, Franck Davoine and Philippe Xu

Abstract— **Pedestrian detection is a well-studied problem. Even though many datasets contain challenging case studies, the performances of new methods are often only reported on cases of reasonable difficulty. In particular, the issue of small scale pedestrian detection is seldom considered. In this paper, we focus on the detection of small scale pedestrians, i.e., those that are at far distance from the camera. We show that classical features used for pedestrian detection are not well suited for our case of study. Instead, we propose a convolutional neural network based method to learn the features with an end-to-end approach. Experiments on the Caltech Pedestrian Detection Benchmark showed that we outperformed existing methods by more than 10% in terms of log-average miss rate.**

I. Introduction

In order to reach reliable advanced driver assistance systems and safe autonomous driving, a robust pedestrian detection is mandatory. Due to its high commercial interest and to the possibility to make the roads safer, this has been an intense area of research in the past decades. The availability of several public benchmark datasets [1], [2], [3] has allowed the research community to compare many detection systems in challenging scenarios. This has lead to the emergence of new methods based on advanced features such as the classical HOGs [1] or Integral Channel Features [4] among many others. These progresses have brought impressive results in cases of reasonable difficulty. However, difficult situations such as occluded or distant people remain challenging. While occlusion has been studied at several occasions [5], the detection of pedestrian at far distance has been seldom explored.

The size of the target objects is known to be an important factor when looking to perform detection and is often the primary explanation when analyzing the failure modes of a method [6], [7]. Notably, most pedestrian detectors fail at detecting people with an apparent size on the image of less than 30 pixels. Throughout this paper, we will refer to them as *small* pedestrians [8]. While the problem could possibly be alleviated by using higher resolution cameras, as suggested in [9], this would prove more expensive, both in terms of hardware costs as well as computationally due to the increase in image sizes.

Our work aims at improving the state of the art in small scale pedestrian detection, which will be necessary in future autonomous vehicles. Indeed, in order to allow travel at high speeds, detection of obstacles should be made as early as possible.

In this paper, we will first review the state of the art and show the limitations of classical methods. Then, we will focus on our proposed method based on Convolutional Neural Networks (CNNs). Finally, experimental results on the Caltech Pedestrian Detection Benchmark [2] are reported in Section V. We show an improvement of more than 10% in terms of log-average miss rate.

II. State of the art

Classically, pedestrian detection consists in extracting features, such as HOG [1], shapelet features [10] or color self similarity [11] and using classifiers such as Adaboost [12], [4] or linear SVM [1], [11], [10]. In order to handle pose variations, the use of Deformable Parts Models [13] is also a popular method yielding good results by decomposing a pedestrian into a coarse global and some local appearance models, together with geometric constraints on their deformation. Other methods for pedestrian detection leverage features from motion as in [14]. Originally, all these methods were developed under the assumption of a particular human appearance structure, covering shapes, colors and sub-parts. However, the limited resolution of small pedestrian prevents recovering these features.

The detection of small objects has often been studied in the context of aerial imaging [15]. In this context, the successful detection is usually due to the relatively simple background. In the case of urban scenes captured from a vehicle, these approaches are not applicable.

As opposed to the use of traditional handcrafted features, the idea to use Deep Learning and end-to-end learning methods in order to build a good representation for pedestrian detection is emerging. In [16], unsupervised learning is used to give a proper initialization of the network weights before fine-tuning the CNN in a supervised manner. In [17], Ouyang and Wang learn jointly the feature extraction, the deformation handling and the scoring of candidates. In [18], handcrafted features were used the different stages of the cascade of classifiers were learned together.

In object detection benchmarks, the best performances are achieved by methods relying on CNNs. These approaches are now widely based on the "Region with CNN feature" paradigm (R-CNN) [19]. Starting from a number of potential regions of interest, it consists in extracting features from each region proposal with a CNN pre-trained on a larger dataset, then in using class specific classifiers to identify the objects and finally in improving the localization by fitting

* This work was carried out and funded in the framework of the Labex MS2T. It was supported by the French Government, through the program "Investments for the future" managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02)

The authors are with Sorbonne Universités, Université de Technologie de Compiègne, CNRS, Heudiasyc Laboratory, Compiègne, France. Rudy Bunel is also with École Centrale Paris, Châtenay-Malabry, France. bunel.rudy@gmail.com, franck.davoine@hds.utc.fr, philippe.xu@hds.utc.fr

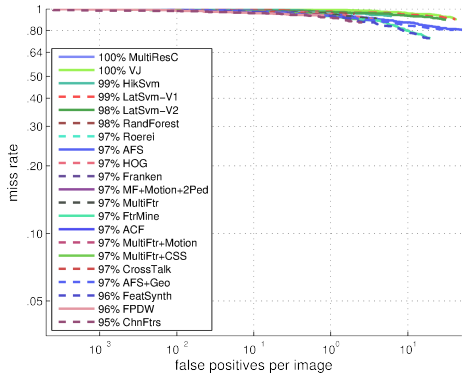


Fig. 1: Detection results for the far scale group.

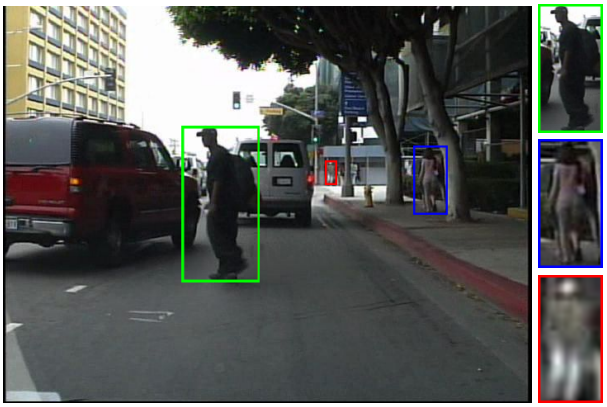


Fig. 2: Urban scene with various sized pedestrians: near (in green), medium (in blue) and far (in red) scales.

bounding-box regressors. The application of these methods to pedestrian detection in [20] yields competitive results on the Caltech Pedestrian Dataset and points that a critical element of success is the use of pedestrian specific detectors for region proposal instead of class-agnostic ones. However, in the specific case of far scale pedestrians, even at high proportions of false positives, detectors still have too low recall rates to be used as region proposal generators.

Another way to use CNNs for detection is the one proposed in [21], where the network is applied in a sliding window manner. High scoring windows become candidate bounding boxes, whose locations are improved by a regression network. Another similar approach is the Large-Field-of-View network [22], used to perform a fast high-recall pedestrian detection that serves as the first stage in a cascade of networks. Our method is similar to these approaches but has been specifically designed for small pedestrians.

III. Detection of small pedestrians

In the Caltech dataset, pedestrians are grouped into three scales, based on their height: near (80 or more pixels), medium (between 30-80 pixels) and far (30 pixels or less). The far scale corresponds to our so called small pedestrians. A first look at the performance of existing methods shows the difficulty of detecting them. Figure 1 depicts the performance of several algorithms in the far scale case. We can clearly see that the results are far from being satisfactory. Typically, at a rate of one False Positive Per Image (FPPI), the best algorithm barely reaches 5% detection rate.

Those methods still learn from pedestrians within their image context, at a predefined scale, typically 64 by 128 pixels. Detecting bigger or smaller pedestrians requires image resizing operations, giving rise to artifacts as shown in Figure 2. Upsampled small pedestrians appear blurry, with low information content.

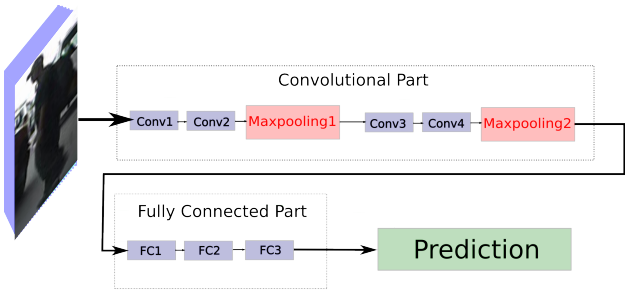
As a result, most models fail at detecting small pedestrians, as they differ significantly from the training samples. To tackle multi-resolution issues, Park et al. [23] proposed to learn different HOG models for different scales. However, the detection of small pedestrians remained unsuccessful. This can be explained by the lack of reliable contour information. In the particular case of HOG features, the restricted numbers of pixels available means that either the histograms are built with very few gradient values, or that each cell corresponds to a large part of the pedestrian. For these reasons, we decided to not base ourselves on the classical features but instead to learn features with an end-to-end approach, by making use of CNNs.

IV. Our model

A. Convolutional Neural Networks

CNNs, as particular cases of Deep Neural Networks (DNNs), are trained in a fully supervised, end-to-end manner to learn a hierarchy of features. An advantage of CNN-based methods is that no prior domain knowledge is required. A DNN consists of multiple levels of feature extractors, arranged in a feed forward manner, that apply parameterized transformation functions in order to transform their inputs into higher level representations. CNNs use convolutional filters as transformation functions, meaning that the output of a filter is a linear function of the input. This allows the network to take advantage of the two dimensional structure of the data. On top of these filters, CNNs also contain non-linear activation units, intertwined between convolutional layers so as to obtain a richer global function than a simple linear function. Examples of activation units are the Rectified Linear Unit (ReLU), sigmoid or hyperbolic tangent. The output of the activation function is used as input for the following layer in the hierarchy. We also make use of sub-sampling layers (max-pooling, average pooling) that reduce the dimension of the input and provide small scale translation invariance.

We train our network using the back-propagation algorithm. The results of the forward pass of our samples through the network are compared to the corresponding labels with the use of a differentiable loss function. This loss function can be simple, such as a cross-entropy in the case of binary classification like ours or can be more complex for other tasks. The comparison between the output of our network and the labels using the loss function returns an error. Using the chain rule, it is possible to differentiate the loss function that we chose and obtain the derivatives with respect to the



	Layer	Train time size	Test time size
Input		3 x 48 x 32	3 x 480 x 640
Convolutional part	*conv1* - 5 x 5 Convolution, filter depth 25	25 x 44 x 28	25 x 476 x 636
	conv2 - 5 x 5 Convolution, filter depth 50	50 x 40 x 24	50 x 472 x 632
	maxpooling1 - 2 x 2 MaxPooling, stride 2	50 x 20 x 12	50 x 236 x 316
	conv3 - 5 x 5 Convolution, filter depth 75	75 x 16 x 8	75 x 232 x 312
	conv4 - 5 x 1 Convolution, filter depth 100	100 x 12 x 8	100 x 228 x 312
	maxpooling2 - 2 x 2 MaxPooling, stride 2	100 x 6 x 4	100 x 114 x 156
Fully connected part	*fc1* - 6 x 5 Convolution, filter depth 1200	1200 x 1 x 1	1200 x 108 x 152
	fc2 - 1 x 1 Convolution, filter depth 600	600 x 1 x 1	600 x 108 x 152
	fc3 - 1 x 1 Convolution, filter depth 1	1 x 1 x 1	1 x 108 x 152

TABLE I: Our CNN architecture.

filter weights of our network. Using this gradient, we can then optimize the loss function on the training dataset to generate appropriate features. The optimization can be done with any gradient descent type algorithm. In order to increase the convergence speed, the gradient is not computed on the whole training set but approximated on mini-batches, so as to perform more frequent updates.

B. Architecture

The template of pedestrians that we want to detect is set to 16 by 32 pixels. We designed our network to have receptive fields (the size of an input element that matches one prediction) of size 32 by 48 pixels, in order to include contextual information around the target. It is known that including this margin leads to significant improvement [1]. Given the limited size of the receptive fields, we chose to use small kernels and pooling regions. Our network is composed of two main parts. The first consists in four convolutional layers and two max-pooling layers that extract features. The second part contains three fully connected layers to predict the score using the features. The complete architecture is detailed in Table I.

Each of the layers in our network is followed by a Parametric ReLU (PReLU) activation unit [24]. These units have the low computational cost of the ReLU layers but also provide a way to learn parameters for the activation function, leading to improved accuracy.

C. Learning and prediction

To train the network, we perform stochastic gradient descent. Our loss function is the binary cross entropy,

$$\delta(c,p) = c\log(p) + (1-c)\log(1-p) \tag{1}$$

where c is the class of our sample and $p \in [0,1]$ our prediction after passing through a sigmoid.

In order to improve regularization and prevent overfitting, we use dropout [25]. The gradient step size is reduced gradually and we perform early stopping by monitoring the objective function on a validation set.

During the evaluation phase, our model that makes a 32 by 48 pixels window correspond to a single score prediction is applied in a fully convolutional manner. This is equivalent to performing a sliding window on the whole image. Instead of applying the convolutions to a patch sized input, we apply them to the whole image. The last layers that were considered fully connected during the training phase are equivalent to convolutional layers with a kernel size corresponding to the size of the training samples at this stage. For the layers with a kernel bigger than one, this is more efficient than processing each window separately due to the shared computations between overlapping regions. Furthermore, a lot of work has been done recently to optimize the convolution operations [26], [27]. Applying our CNN in a fully convolutional manner generates a two dimensional prediction matrix where each element corresponds to the prediction for a window in the input image. In order to detect pedestrians of different sizes, we feed in the input images at various scales.

D. Detection boxes filtering

Our network returns a score for each detection window at all positions and scale. We only keep Bounding Boxes (BB) with a positive score. A bounding box is defined by a five dimensional vector

$$BB = (x, y, w, h, s), \tag{2}$$

where (x, y) represent location coordinates, (w, h) the dimensions of the box, and s the score.

Some pedestrians may have activated the output for several neighboring predictions so we need to perform a Non Maximum Suppression (NMS) step to remove duplicates that would be considered as false positives. Typically, a pair of bounding boxes (BB_i, BB_j) is supposed to correspond to a unique pedestrians if the overlap;

$$a = \frac{\text{area}(BB_i \cap BB_j)}{\text{area}(BB_i \cup BB_j)}, \tag{3}$$

is above a threshold t.

A simple solution consists in greedily selecting the highest scoring bounding box and then removing all the bounding boxes with enough overlap [13]. NMS can be formalized as a clustering problem using (3) as a distance measure. Given a cluster $(BB_1, BB_2, \ldots, BB_k)$, this method is equivalent to representing this cluster by a unique box defined as

$$BB_{greed} = (x_M, y_M, w_M, h_M, s_M), \tag{4}$$

where

$$M = \text{argmax}_i(s_i). \tag{5}$$

It means that we only keep the BB with the highest score for each cluster. This method has the issue that we do not take advantage of the additional information given by neighboring bounding boxes. Intuitively, we would expect that a cluster with many high scored BBs should be represented by a

highly scored one. We propose two other strategies for taking this information into consideration.

The first one, so called *Vote strategy*, is similar to the greedy Non Maximum Suppression except that we update the score of the maximum bounding box by adding the score of the boxes that got suppressed. This can be expressed as

$$BB_{vote} = \left(x_M, y_M, w_M, h_M, \sum_{i=1}^{k} s_i \right), \quad (6)$$

where M is defined as previously (5). This allows regions with a high density of detections to be favored. However we observed some correct detections that all got suppressed by another non-matching one, leading to a false negative with a very high confidence.

In order to solve this problem, a so-called *Merge strategy*,consists in merging all the detections, instead of suppressing the lowest scoring ones. Rather than updating only its score, we also update the other parameters of the BB. To do so, we perform a weighted average of the coordinates, height and width. In order to give more importance to the highest scored boxes, we use the scores as weights. This can be formalized as follows:

$$BB_{merged} = \left(\sum_{i=1}^{k} x_i \frac{s_i}{C}, \sum_{i=1}^{k} y_i \frac{s_i}{C}, \sum_{i=1}^{k} w_i \frac{s_i}{C}, \sum_{i=1}^{k} h_i \frac{s_i}{C}, C \right), \quad (7)$$

where C is a normalization constant defined as

$$C = \sum_{i=1}^{k} s_i. \quad (8)$$

The results given by these strategies are compared in section V.

V. Experiments

We performed all of our experiments on the Caltech Pedestrian Dataset, using the Matlab toolbox and evaluation software provided by [2]. Our CNN implementation is based on the open source Torch framework [28]. Codes are available at the authors website.[1]

A. Training Data

The Caltech Pedestrian Dataset is composed of 11 sets of images taken from a camera embedded into a vehicle driving in normal conditions in urban areas. The dataset contains 250 000 frames and 2300 unique pedestrians in total, presenting varying sizes, aspect and occlusion ratios.

We performed our training on the first five sets (noted 0 to 4 in the distributed dataset) and kept the sixth one for validation and early stopping. To gather our positives samples, we extracted from each frame all the pedestrians that did not present occlusion and were not part of a group of people for which no precise annotation is available. This led to about 44,000 samples of small pedestrians. Due to the fact that the performance of neural network is heavily dependant on the volume of data available during training, we also performed data augmentation in order to extend our training set. Medium and near scale pedestrians resized by downsampling provided an additional 33,000 positive samples. For each pedestrian, we used their original images, the mirrored version around the vertical axis, as well as the results of small deformations consisting in translation and scaling. While these generated examples are very close to the ones already found in the dataset, they help the network to learn a more generalized version of the pedestrians.

A batch of 100,000 negative examples were randomly sampled from all the images at each iteration. In order to learn the more ambiguous cases, after each epoch of the stochastic gradient procedure, we hold the samples that resulted in high penalty from the loss function, i.e., negative samples considered pedestrians with a high confidence, to be reused in the next epoch, in addition to the new batch of random crops. This forced the network to be more exposed to them. This is similar to hard negatives mining.

B. Evaluation protocol

Like in the PASCAL detection challenge [29], the match between proposed detection and ground truth is determined by the overlap criteria, defined in (3). Usually, a detection is deemed a true positive if this overlap score with a bounding box is higher than a threshold $t = 0.5$. Ground truth bounding boxes can only be matched by a single detection, any additional detection are considered as false positives. Occulted and tightly grouped people were ignored during the evaluation, meaning that they did not count as true positives when detected but did not penalize the score when missed. The performance criterion is the log average miss rate, evaluated for nine levels of false positives per image, spaced evenly in log scale between 10^{-2} and 10^{0}.

C. Results

The results obtained on the *far* subset of the Caltech dataset are shown in the left side of Figure 3. The best results were obtained with the *Merge* strategy. In terms of log average miss rate, we observes a gain of 7%. The *Vote* and *Greedy* schemes performed slightly worse but they were still significantly better than existing methods. We also present the same evaluation but using a lower overlap threshold, therefore corresponding to a relaxation in the precise localisation expectation. During the evaluation, images were processed at a rate of 1.34 frames per second on a desktop computer equipped with a GPU. Speed wasn't an objective of this project but this could be improved by making use of recent results in speeding up the evaluation of CNNs [30].

With deeper analysis of our failures, we observe that many detections were considered as false positives because of the fixed threshold, even though they present a reasonable overlap with the ground truths. Examples of such cases are shown in Figure 5.

In the case of small pedestrians, tiny misalignment of bounding boxes can translate to great reduction in overlap score. For this reason, we also tested our performance with a threshold of $t = 0.25$. This comes down to a relaxation of the localization criterion. These results are shown on the right side of Figure 3. Under this setting, the Vote and

[1] https://github.com/bunelr/utc-caltech

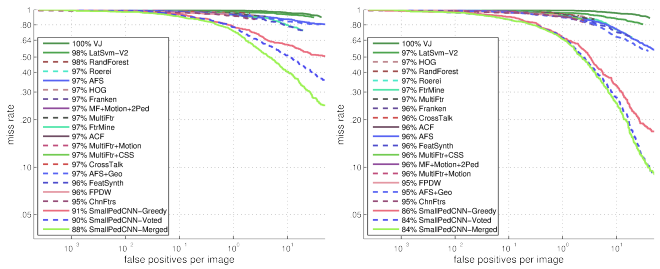


Fig. 3: Evaluation Results on far scale pedestrians of the Caltech dataset for $t = 0.5$ (left) and for $t = 0.25$ (right).

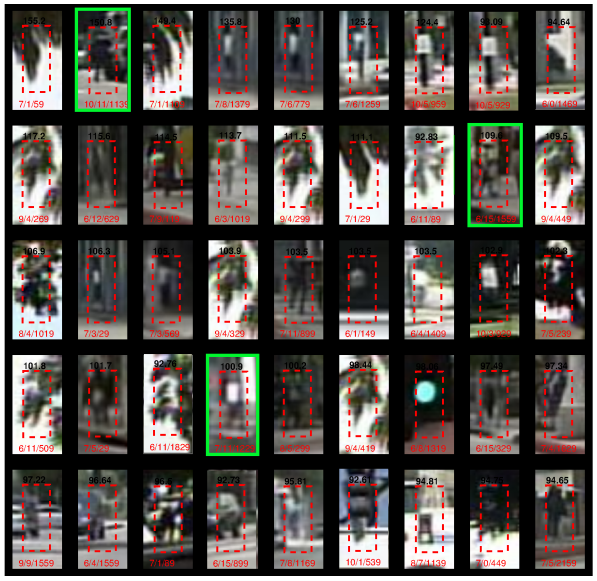


Fig. 4: Worst False Positives. Highlighted are some true positives that correspond to pedestrians non annotated in the ground truth data.

Merge strategies led to the same results, a reduction of more than 10% in miss rate. The difference between these two in the former case ($t = 0.5$) demonstrates that using all boxes to determine the location coordinates of the cluster representative improves localization. It is interesting to notice that existing methods do not benefit as much from the relaxed criterion on the localization.

In Figure 6, we show some detection examples. The detectors were set up to have a 1 FPPI rate on the far configuration. We can observe that our method have a much higher recall rate than both HOG and Channel Features [4]. Allowing a higher number of false positives, our methods can reach miss rates lower than 10% while existing methods barely attain 60%. Analyzing the false positives detections also led to some insights. Figure 4 shows the highest scoring false positives. Among those, almost a third consist in leaves of tree that are wrongly detected as pedestrians. Other elements source of confusions are traffic signs, lamp posts and similar elements of urban architecture. We can see that for a significant ratio of false positives, even a human would be fooled by their visual appearance. This hints that we may not hope to achieve much better improvements by relying only on the visual appearance of pedestrians.

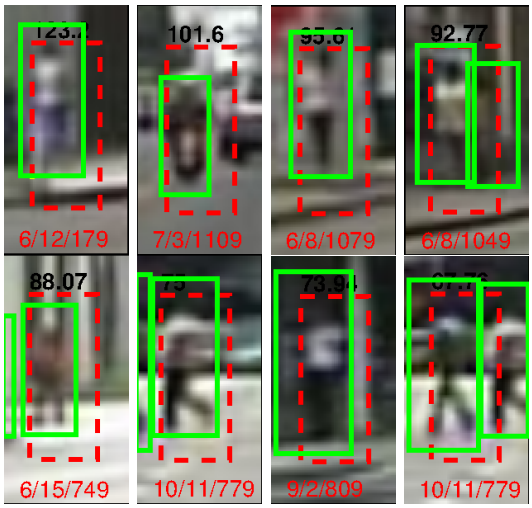


Fig. 5: Examples of overlap between 0.25 and 0.5. The red boxes are our detections and the green ones are ground truths

VI. Conclusion

We improved the state of the art in small pedestrians detection by reducing by an order of magnitude the numbers of false positives, using a simple convolutional neural network. The best localisation is obtained using our merging strategy. As a future direction, we can consider more complex approaches, such as regression based localisation [19]. Incorporating additional information in our model such as motion [14] thermal information [31], or semantic attributes [32] would yield further improvements. Improving the model by taking into account the relation between different frames, for example using Recurrent Neural Networks [33].

References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, vol. 1, June 2005, pp. 886–893.

[2] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 304–311.

[3] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research*, 2013.

[4] P. Dollar, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," in *Proceedings of the British Machine Vision Conference*, 2009, pp. 91.1–91.11.

[5] W. Ouyang and X. Wang, "A discriminative deep model for pedestrian detection with occlusion handling," in *IEEE International Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 3258–3265.

[6] B. Pepik, R. Benenson, T. Ritschel, and B. Schiele, "What is holding back convnets for detection?" in *German Conference on Pattern Recognition*, 2015.

[7] D. Hoiem, Y. Chodpathumwan, and Q. Dai, "Diagnosing error in object detectors," in *European Conference on Computer Vision*, 2012, vol. 7574, pp. 340–353.

[8] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, vol. 34, no. 4, pp. 743–761, April 2012.

[9] R. N. Rajaram, E. Ohn-Bar, and M. M. Trivedi, "An exploration of why and when pedestrian detection fails," in *IEEE Conference on Intelligent Transportation Systems*, 2015.

Fig. 6: Comparison of pedestrian detectors. From left to right, HOG, ChnFtrs and SmallPedCNN-merged detections. All detectors are setup to have a 1 FPPI rate on the *far* subset of the dataset. Green, blue and red boxes correspond respectively to Detections, Misses and False Positives. Solid lines correspond to ground truths and dotted lines to detections.

[10] P. Sabzmeydani and G. Mori, "Detecting pedestrians by learning shapelet features," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.

[11] S. Walk, N. Majer, K. Schindler, and B. Schiele, "New features and insights for pedestrian detection," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2010, pp. 1030–1037.

[12] P. Dollar, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532–1545, Aug 2014.

[13] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence,*, vol. 32, no. 9, pp. 1627–1645, Sept 2010.

[14] D. Park, C. Zitnick, D. Ramanan, and P. Dollar, "Exploring weak stabilization for motion feature extraction," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2013, pp. 2882–2889.

[15] S. Razakarivony and F. Jurie, "Discriminative autoencoders for small targets detection," in *International Conference on Pattern Recognition*, 2014, pp. 3528–3533.

[16] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, "Pedestrian detection with unsupervised multi-stage feature learning," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2013, pp. 3626–3633.

[17] W. Ouyang and X. Wang, "Joint deep learning for pedestrian detection," in *IEEE International Conference on Computer Vision*, Dec 2013, pp. 2056–2063.

[18] X. Zeng, W. Ouyang, and X. Wang, "Multi-stage contextual deep learning for pedestrian detection," in *IEEE International Conference on Computer Vision*, Dec 2013, pp. 121–128.

[19] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 580–587.

[20] J. Hosang, M. Omran, R. Benenson, and B. Schiele, "Taking a deeper look at pedestrians," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2015.

[21] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," in *International Conference on Learning Representations*, April 2014.

[22] A. Angelova, A. Krizhevsky, and V. Vanhoucke, "Pedestrian detection with a large-field-of-view deep network," in *IEEE International Conference on Robotics and Automation*, May 2015, pp. 704–711.

[23] D. Park, D. Ramanan, and C. Fowlkes, "Multiresolution models for object detection," in *European Conference on Computer Vision*, 2010.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *IEEE International Conference on Computer Vision (to appear)*, 2015.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[26] N. Vasilache, J. Johnson, M. Mathieu, S. Chintala, S. Piantino, and Y. LeCun, "Fast convolutional nets with fbfft: A GPU performance evaluation," in *International Conference on Learning Representations*, 2015.

[27] S. Chetlur, C. Woolley, P. Vandermersch, J. Cohen, J. Tran, B. Catanzaro, and E. Shelhamer, "cudnn: Efficient primitives for deep learning," *arXiv preprint arXiv:1410.0759*, 2014.

[28] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, 2011.

[29] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, 2010.

[30] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Speeding up convolutional neural networks with low rank expansions," in *British Machine Vision Conference, BMVC*, 2014.

[31] S. Hwang, J. Park, N. Kim, Y. Choi, and I. So Kweon, "Multispectral pedestrian detection: Benchmark dataset and baseline," in *IEEE Conference on Computer Vision and Pattern Recognition)*, June 2015.

[32] Y. Tian, P. Luo, X. Wang, and X. Tang, "Pedestrian detection aided by deep learning semantic tasks," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2015.

[33] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," *arXiv preprint arXiv:1411.4389*, 2014.