Adaptive Deformation Handling for Pedestrian Detection

Hak Kyoung Kim YongHyun Kim DaiJin Kim
Department of Creative IT Engineering,
Pohang University of Science and Technology (POSTECH), Republic of Korea
{khk88,gkyh0805,dkim}@postech.ac.kr

Abstract

Despite the abundance of successful models for pedestrian detection, many are limited in their ability to handle deformations, such as large appearance variations. In view of insufficient number of models with the ability to handle deformations, we propose a simple strategy, which incorporates deformation handling with a spatial pyramid method in basic classifier learning. By using the max pooling method, this approach aggregates a set of randomly-selected basic features from a local region. The spatial pyramid method has been integrated to our method to construct a richer feature in a local region. We show how to train the model with this deformation handling method using a boosting process. Our best detector outperforms the state-of-the-art of pedestrian detection on the INRIA and the Caltech-USA datasets. It achieves a log average miss rate of 12.21% on the INRIA and a log average miss rate of 24.03% on the Caltech-USA datasets.

1. Introduction

Pedestrian detection is a key problem for many applications such as robotics, video surveillance and automated driver assistance. Over the past few years, researchers in the area of pedestrian detection have made great progress. However, the performances of current systems are not efficient enough for application. The main challenge is how to deal with the large variability of human shapes and of appearances caused by pose, clothing, occlusions and lighting changes.

Most early pedestrian research focused on modeling the global appearance of pedestrians. For example, there is Viola and Jones, which uses the Haar-wavelet feature [18], and Histogram of Oriented Gradient (HOG) [5]. Boosting method and sliding window technique have become fundamental of pedestrian detection research and is applied in recent researches [7, 8, 6]. However, these fundamental methods lack the ability to cope with large variations and to detect pedestrians in complex environment settings.

In order to enhance these fundamental methods to handle the lack of ability to cope with variations, recent research have brought forth the Deformable Part Model (DPM) [13]. The DPM exploits the fact that the overall figure of the pedestrian might change, but parts of the pedestrian, such as the arms and legs, do not have significant change. By distinctively training each part, the learning model is able to combine information to detect the pedestrian. However, as DPM requires each parts of the pedestrian to be trained individually, additional steps are required and processing speed increases.

Many recent pedestrian detection researches suggest methods that use the pooling method [4], which is mainly used in object recognition. These methods set the pooling region and either averages or maximizes features to be reordered in the region. They can fulfill the needs of handling deformations in a degree, as the pooling method reduces the effect of spatial constraints within the region. But, if the low-level features of the pooling region have heterogeneous properties, information loss could occur and lead to low performance.

In order to pool features without the decrease in performance and efficiently handle large deformations, Wang *et al*. suggested regionlet [19] in object recognition. Although regionlet is similar to other recognition researches in terms of using the basic pooling method, regionlet does not pool features from the entire pooling region, but pools selective features which describe fine-grained spatial information inside object. In other words, regionlet not only models relative spatial layouts inside object, but also tolerates deformations by aggregating responses of selected features.

Inspired by regionlet and recent success of spatial pooling on object recognition, we propound a new framework to detect pedestrians. To devise an innovative strategy to handle deformations, we use the regionlet's selected feature pooling method to create a new feature pool. Moreover, additional improvements are achieved through the integration of regionlet and the spatial pyramid method [3, 15]. The spatial pyramid method tries to increase the feature's discriminative characteristics through various scales of fea-

978-1-4799-6683-7/15 $31.00 © 2015 IEEE
DOI 10.1109/WACV.2015.28

tures, and this is a similar concept to the multi-resolution [21] proposed in pedestrian detection. The existing multi-resolution method scales the input image to calculate the feature, but this takes a lot of time. However, we don't use the scaling. We efficiently find the feature based on the pooling operation. The method proposed was evaluated on the INRIA, Caltech-USA, TUD-Brussels, ETH datasets and our approach outperforms all reported pedestrian detectors in Caltech-USA datasets.

The rest of this paper is organized as following: Section 2 discusses related work. Section 3 introduces baseline detector, ACF. Section 4 shows feature extraction and training method for deformation handling. In Section 5, we explains experiment parameters and report the results. Section 6 concludes the paper.

2. Related Work

Aggregated Channel Feature (ACF), which is proposed in [6], is the most frequently used method in recent pedestrian detection research. Benenson *et al*. [1] proposed the method which uses slightly changed ACF and the GPU to process 50 frames per second. An even higher detection rate is achieved by stereo information (Veryfast). Moreover, [2] obtained better detection result by using the multi-scale model based on ACF. This method achieves high accuracy because of the different size of pedestrian models have different information (Roerei). W. Nam *et al*. [20] improved detection performance by adjusting the parameters of ACF and they proposed a feature transform method that removes correlations in local neighborhoods (LDCF).

Fundamentally, pedestrians have large intra-class variations caused by pose and occlusion. In [13, 12], P. Felzenszwlab *et al*. suggested Deformable Part Models(DPM) to cope with variations by combining rigid root filter and deformable part filters. The DPM only performs well for high resolution objects. J. Yan *et al*. in [22], proposed a multi-task deformable part model for multi-resolution pedestrian detection. In order to handle occlusions and spatial interactions [21, 16, 11] were suggested. [21] proposed a unified probabilistic framework to globally describe multiple pedestrians in crowded scenes. In [16], M. Mathias *et al*. trained a set of occlusion-specific classifiers called Franken-classifiers, and each Franken-classifier was trained for a certain amount and type of occlusion (Franken). Occlusion inference was done using depth information in [11].

Our approach is also related to object recognition and classification area. Most general objects have large variations depending on the pose. Researches in these area suggest spatial pyramid to efficiently represent image [3, 15, 14]. This method has emerged as a popular framework to encapsulate sophisticated feature. It has similar meaning to multi-resolution methods [2, 22]. X. Wang *et al*. in [19], presented regionlet. This method selectively choose a set of features and then pools them to handle deformations. We were motivated by this method which achieved the high performance in object recognition and we propose a novel framework for pedestrian detection.

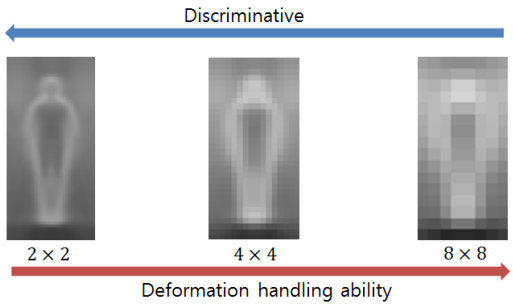

Figure 1: Characteristics of trained models according to different block size. The model which has high discriminative property is vulnerable to deformations (left). Large block size allows the model to cover large variations, but is not sensitive enough for detection (right).

3. Baseline Detector

We will briefly explain the low-level feature we have used. Although the proposed method is capable of using HOG, SIFT and other various features, we have employed the Aggregated Channel Feature (ACF), which seems to have the best performance in detecting pedestrians. Given an input image, linear and non-linear transformations are used to compute multiple registered image channels, then features are efficiently computed. Several channels that include different information lead to high performance in pedestrian detection. ACF uses two-step smoothing to accumulate neighborhood information before and after feature extraction, and uses Adaboost [18] with multiple rounds of bootstrapping. Similar to [6], we used the basic channels like LUV colors (3 channels), gradient magnitude (1 channel) and gradient histogram (3 channels). We divide each channel into several blocks of fixed size, and then compute local sums of each block, which are used as features.

Original ACF shows a outstanding performance of 40% in the Caltech Pedestrian Dataset [9], but W. Nam *et al*. achieved 30% performance by tuning the parameters slightly in [20], increasing the tree depth and increasing the number of samples. We explain the relationship between tree depth and the number of samples in §4. We adopt this method and call this altered detector as advanced-ACF.

The reason ACF has the best performance is that it uses information acquired from various channels and applies smoothing to concentrate surrounding information. However, ACF only focuses on a local area and cannot detect pedestrians well when large deformations occur. To resolve this problem, we propose our new method.

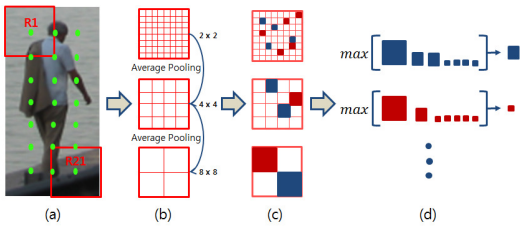

Figure 2: Illustration of Feature extraction strategy. (a) Red-rectangles are deformation handling regions, and green-points are center of that region. (b),(c) and (d) show how to make deformation handling feature.

4. Deformation handling Method

In pedestrian detection, models created by machine learning technics essentially contain spatial information by location of features from local parts. ACF, our baseline detector, also contains the local information based on the blocks. To be brief, we extract the features from sub-rectangles with fixed location in an image and combine their information to decide whether it is a pedestrian. Therefore, the size of sub-rectangles has a decisive effect on the pedestrian model because different block-sizes include different amounts and kinds of information (Figure 1). Features extracted from small areas have good localization ability, but are weak in the presence of variations. Features extracted from large areas can cover large variations well, but cannot achieve accurate localization. If we want to cover large deformations, we need a model that uses large rectangles, but using large rectangles decreases the detection rate due to loss of discriminative information from the dataset. This motivates us to suggest a new framework that combines [19] and [15] to detect pedestrians when large deformation occur. First, to use diverse information, we compute features with various block sizes using the average pooling; this method is called spatial pyramid method. By using the average pooling, we don't need to repeatedly calculate features over different block sizes. In addition, to reduce effect of spatial constraints from blocks, a randomly-selected set of features are aggregated by the max pooling in the defined deformation handling region, namely the model trained by this method adaptively use different features at same stage according to max function per every sliding window. This method efficiently integrates features with discriminative characteristics in the large region. Thus, localization information is preserved, making it possible to detect pedestrians even in the presence of large variations.

4.1. Feature extraction method

Figure 2 shows overall feature extraction strategy. As shown in figure 2(a), image I is a set of regions R_i with their location indices $i = 1, \cdots, N$. R_i is the definition of the deformation handling region of each location. The size of the region (16×16) is defined by considering the maximum range of deformations. The deformation handling region can be set as the whole input image, but this requires high computation power and wastes memory space. Our feature extraction method takes two steps. The first step uses a average pooling method based on baseline feature ACF to extract a pool of spatial pyramid features (Figure 2(b). This pool includes more information than the original one-size block feature pool. Each R_i is divided into a minimum block size (2×2) and then we computed the low-level feature (ACF). Blocks in the region R_i are represented as x_{ij}, where $j = 1, \cdots, M$ indicates the relative position index. S consists of indices of cells in the spatial pyramid (In our case, we use three pyramid levels. S is 64+16+4 = 84), with $\boldsymbol{M}_S$ denoting the set of block locations in the spatal pyramid region S. Let f and g denote some low-level feature extraction function (ACF) and pooling operator (average pooling), respectively. The vector $\boldsymbol{SPF}_i$ that represents the spatial pyramid feature pool is obtained by sequentially extracting features and pooling over R_i.

$$\alpha_{ij} = f(x_{ij}), \quad j = 1, \cdots, M \tag{1}$$

$$F_s = g(\{\alpha_{ij}\}_{j \in \boldsymbol{M}_S}), \quad s = 1, \cdots, S \tag{2}$$

$$\boldsymbol{SPF}_i = [F_1 \cdots F_S] \tag{3}$$

The second step is to integrate features of a set of features from $\boldsymbol{SPF}_i$ into a 1-dimensional feature. This process makes the feature invariant to deformations. A randomly-selected set of features is represented as $\{F_k\}_{k \in r_S}$, where r_S is the indices set of randomly-selected features from the spatial pyramid, then generates the deformation handling feature $DHF(R_i)$ for the region. The equation is defined as the following:

$$DHF(R_i) = \sum_{k \in r_S} \beta_k F_k \tag{4}$$

$$subject \;\; to \;\; \beta_k \in \{0, 1\}, \quad \sum_{k \in r_S} \beta_k = 1$$

where β_k is either 0 or 1. Only one of a set of features will contribute to representing the region. This means that if the deformation occurs along the locations of selected features, we would use only one feature that is most discriminative to input image. Eq.4 can be easily denoted by the max operation.

$$DFH(R_i) = max_{k \in r_S}(F_k) \tag{5}$$

In our framework, we repeated the second stage to construct a deformation handling feature pool, and then concatenate this feature pool with the spatial pyramid feature pool for training. We can construct an overcomplete feature pool with endlessly high dimensions using the feature

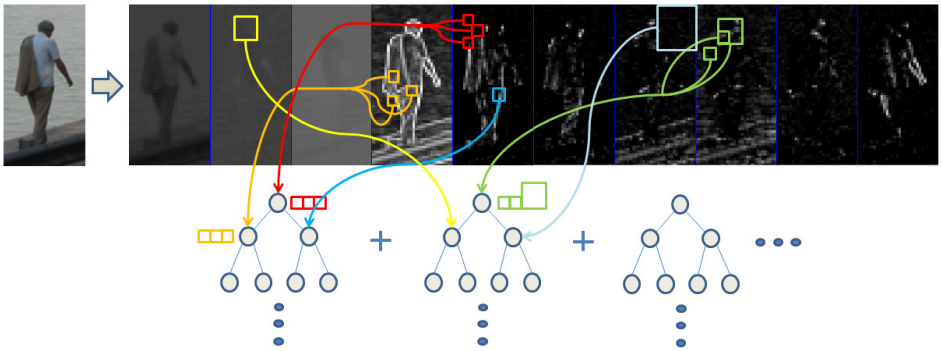

Figure 3: Visualization of the training scheme from multiple registered image channels (LUV colors, Gradient magnitude, Gradient histograms). Each weak classifier consist of N-depth tree. When boosting select deformation handling feature, corresponding node hold the feature locations in the way of lookup table.

extraction method, but for practical implementation we restrict the feature dimension by limiting the number of repeating second stage. With this feature pool, we use the boosting method to train the model. The parameters for feature extraction and extension will be explained in §5.

4.2. Training based on Tree structure

Boosting method offers a fast, effective approach to train the model given a large feature pool. We use RealBoost [17] based on a tree structure. This means that the a weak classifier consists of an N-depth tree. (N is the number of tree depth). Figure 3 shows an example of our classifier being trained. Each circle denotes node that is a binary tree, and the collection of these trees forms an N-depth tree, which separates the training images (positive images + negative images) recursively based on the feature corresponding to the node. Therefore the data are analyzed N times to decide whether they are positive or negative. Conceptually, an N-depth tree is comparable to an N-dimensional decision boundary. (We can't claim that it is an optimal N-dimensional decision boundary because we use a maximum of 2^N-1 features.) Therefore, if we increase the depth of the tree, the discriminative character of the weak classifier increases. But in pactice, this is not always true. If the dataset is unlimited, the performance of the classifier would increase relatively with the depth of the tree. However, if the dataset is finite and tree depth is relatively high, the classifier would have an overfitting problem. Thus, if the tree depth is unlimited, eventually, the training dataset will be accurately classified as positive or negative, but will have no effect on newly-input testing data. The overfitting problem is related to feature dimension. According to the curse of dimensionality [10] theory, if dimension increase, the volume of the space exponentially increase. Thus, the data becomes sparse. This means that decision boundary can be easily overfitted. [8] reports that the 2-depth tree produces the best performance in pedestrian detection, but our method produces the optimal performances with 3-depth tree in the INRIA and over 5-depth tree in Caltech-USA when using same number of data. This is because deformation handle feature integrates multiple features. It has similar effects of reducing the dimensionality. This is why the deeper depth of the tree, in the method we proposed, shows high performance.

Our approach is slightly different from the existing learning approaches. First, as described in figure 3, block size is not static. This means the feature is trained from the spatial pyramid feature pool to obtain diverse information. Moreover, many features can be mapped on one node. This means the deformation handling method. Each of training images uses just one feature which is largest in condidates to train model, namely If the deformation handling feature is selected from the boosting method, the node saves the location information of a set of features as the lookup table. We apply max function among feature values that corresponds to location information when we apply this model. Similar to most algorithms that fundamentally use boosting, we use the cascading method and multiple stages of bootstrapping.

5. Experiment

We evaluate the performance of our algorithm on the Caltech-USA, INRIA, ETH, and TUD-Brussels datasets. We use Caltech pedestrian benchmark software that provides comparison of many state-of-the-art methods. Detection performance is measured by the miss rate (MR) related to false positives per image (FPPI) [9]. In our implementation, we utilize ACF as a baseline detector. We train just two detectors: One is trained using the INRIA dataset for the INRIA, ETH, TUD-Brussels test datasets; the other one is trained using the Caltech-USA training dataset for the Caltech-USA. Negative images are collected from INRIA background images for the INRIA detector while neg-

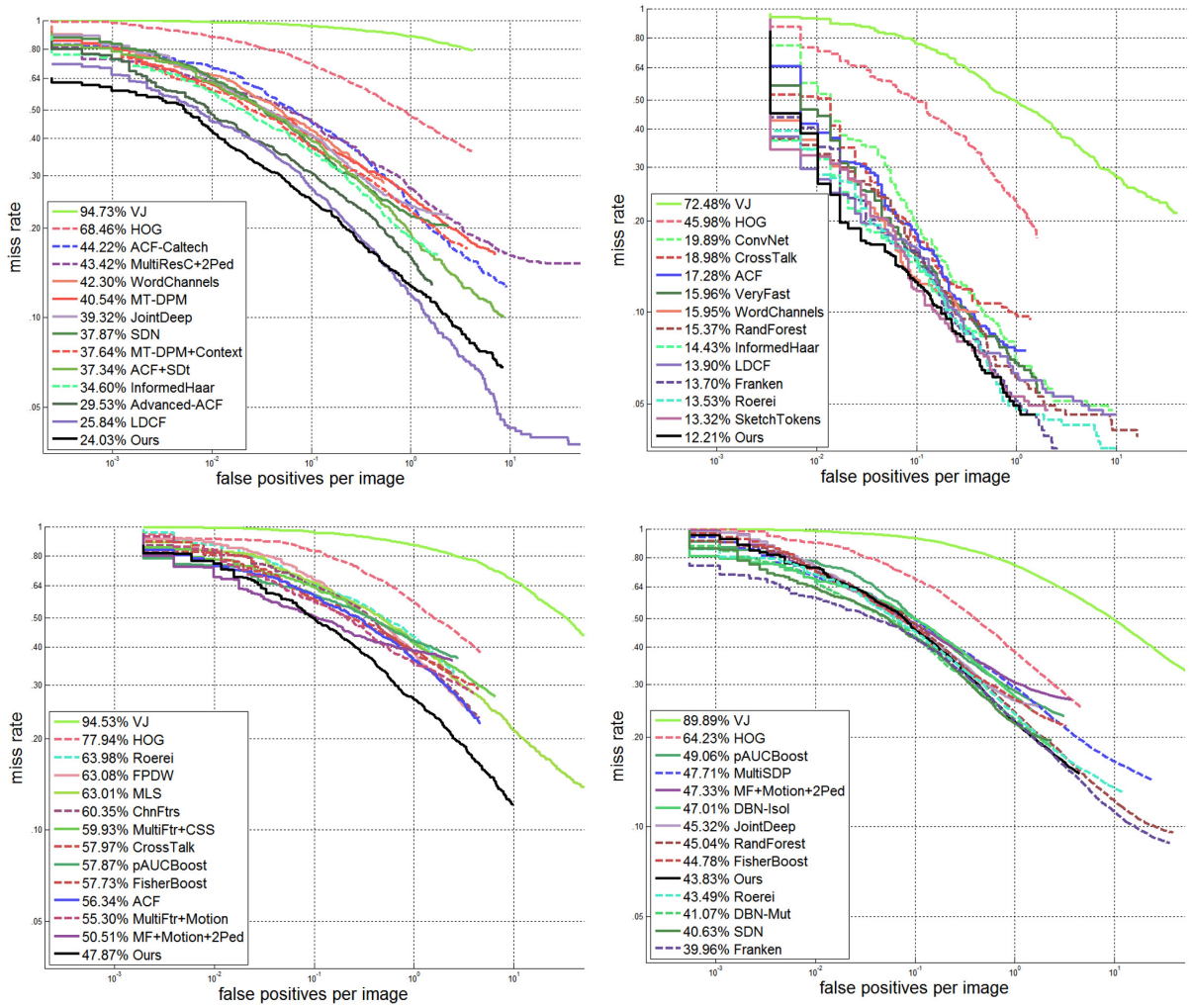

Figure 4: Miss rate vs. false positives per image on four datasets; Legend: log-average miss rate; (a) Caltech-USA, (b) INRIA, (c) TUD-Brussels, (d) ETH.

ative data are collected from the Caltech-USA training set with pedestrians cropped out. We use RealBoost with multiple rounds of bootstrapping to train 2048 N-depth decision trees.

5.1. Parameter setting

For INRIA, each training sample is resized to a resolution of 64x128 pixels. We use $12k$ negative images to train a detector. The block size is set to 2×2 with 10 channels. Therefore, the feature dimension of the baseline detector is 20480 ($64/2\times128/2\times10$). We set the stage of the spatial pyramid to 3. Therefore the dimension of the spatial pyramid feature pool becomes 26880 ($20480+20480/4+20480/16$). We define 21 regions with 32×32 pixels to construct the deformation handle feature pool. We arbitrarily set the number of selected features to 3 and repeat the deformation handling feature extraction process 70 times. We generate additional feature pool that have 14700 ($70\times21\times10$) dimensionality.

For Caltech-USA, we alter the sampling interval from 30 to 4 based on [20]. This change increases the capacity of baseline detector. The resolution of the pedestrian model from Caltech-USA is 32×64. Therefore, the dimension of the baseline detector and of the spatial pyramid feature pool from 10 channels are 5120 ($32/2\times64/2\times10$) and 6720 ($5120+5120/4+5120/16$) respectively. In order to create a deformation handling feature pool, except for setting the region size to 16×16, all parameters are the same as those of INRIA. By increasing the number of data samples, we become independent of the overfitting problem, even if when using weak classifiers with high-depth trees. In practice, we get the best result when we use depth 5.

5.2. Performance

In Figure 4, we compare our detector with state-of-the-art detectors. We use ROC curves for 14 high rank detectors with MR against FPPI. To summarize detector performance, we use the log-average miss rate [9]. We denote the baseline detector trained using INRIA as ACF, and that trained using Caltech-USA as ACF-caltech. For Caltech-USA, by increasing the amount of data and changing some parameters, the advanced baseline detector (Advanced-ACF) achieves 29.53% MR. Our method gives about 5% improvement compared to Advanced-ACF. It achieves 24.03% MR and outperforms all other models. Our model also achieves good performance on TUD-Brussels dataset. The log-average miss rate on TUD-Brussels is 47.87%. Our model does not perform well in ETH dataset that contains many images of highly-occluded people. In our approach, we don't use any special methods to detect highly-occluded people, whereas Franken [11] uses partial trained model. In terms of speed, we minimize computation cost using a lookup table when computing deformation handling feature. Therefore, the runtime of our method is similar to ACF.

6. Conclusion

Combining the selective feature pooling method and the spatial pyramid method, we propose a new pedestrian detector that can successfully identify pedestrians, despite large variations. Our model achieves the best detection performance on the Caltech-USA, INRIA, and TUD-Brussels datasets.

Acknowledgement

This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the "IT Consilience Creative Program" (NIPA-2014-H0201-14-1001) supervised by the NIPA(National IT Industry Promotion Agency)

The research was supported by the Implementation of Technologies for Identification, Behavior, and Location of Human based on Sensor Network Fusion Program through the Ministry of Trade, Industry and Energy (Grant Number: 10041629)

References

[1] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool. Pedestrian detection at 100 frames per second. In *Computer Vision and Pattern Recognition*, 2012. 2

[2] R. Benenson, M. Mathias, T. Tuytelaars, and L. Van Gool. Seeking the strongest rigid detector. In *Computer Vision and Pattern Recognition*, 2013. 2

[3] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *Computer Vision and Pattern Recognition*, 2010. 1, 2

[4] Y.-L. Boureau, N. Le Roux, F. Bach, J. Ponce, and Y. LeCun. Ask the locals: multi-way local pooling for image recognition. In *International Conference on Computer Vision*, 2011. 1

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition*, 2005. 1

[6] P. Dollár, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. 2014. 1, 2

[7] P. Dollár, S. Belongie, and P. Perona. The fastest pedestrian detector in the west. In *BMVC*, 2010. 1

[8] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In *BMVC*, 2009. 1, 4

[9] 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*, 34(4):743–761, 2012. 2, 4, 6

[10] D. L. Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture*, pages 1–32, 2000. 4

[11] M. Enzweiler, A. Eigenstetter, B. Schiele, and D. M. Gavrila. Multi-cue pedestrian classification with partial occlusion handling. In *Computer Vision and Pattern Recognition*, 2010. 2, 6

[12] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained , multiscale, deformable part model. In *Computer Vision and Pattern Recognition*, 2008. 2

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645, 2010. 1, 2

[14] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *International Conference on Computer Vision*, 2005. 2

[15] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition*, 2006. 1, 2, 3

[16] M. Mathias, R. Benenson, R. Timofte, and L. V. Gool. Handling occlusions with franken-classifiers. In *International Conference on Computer Vision*, 2013. 2

[17] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine learning*, 37(3):297–336, 1999. 4

[18] P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004. 1, 2

[19] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *International Conference on Computer Vision*, 2013. 1, 2, 3

[20] N. Woonhyun, P. Dollár, and H. Joon Hee. Local decorrelation for improved detection. In *arXiv*, 2014. 2, 5

[21] J. Yan, Z. Lei, D. Yi, and S. Z. Li. Multi-pedestrian detection in crowded scenes: A global view. In *Computer Vision and Pattern Recognition*, 2012. 2

[22] J. Yan, X. Zhang, Z. Lei, S. Liao, and S. Z. Li. Robust multi-resolution pedestrian detection in traffic scenes. In *Computer Vision and Pattern Recognition*, 2013. 2