Fast and Accurate Pedestrian Detection using Dual-Stage Group Cost-Sensitive RealBoost with Vector Form Filters

Chengju Zhou
School of Computer Science and Engineering, Nanyang Technological University
Singapore
zhou0271@e.ntu.edu.sg

Meiqing Wu
School of Computer Science and Engineering, Nanyang Technological University
Singapore
meiqingwu@ntu.edu.sg

Siew-Kei Lam
School of Computer Science and Engineering, Nanyang Technological University
Singapore
assklam@ntu.edu.sg

ABSTRACT

Despite significant research efforts in pedestrian detection over the past decade, there is still a ten-fold performance gap between the state-of-the-art methods and human perception. Deep learning methods can provide good performance but suffers from high computational complexity which prohibits their deployment on affordable systems with limited computational resources. In this paper, we propose a pedestrian detection framework that provides a major fillip to the robustness and run-time efficiency of the recent top performing non-deep learning Filtered Channel Feature (FCF) approach. The proposed framework overcomes the computational bottleneck of existing FCF methods by exploiting vector form filters to efficiently extract more discriminative channel features for pedestrian detection. A novel dual-stage group cost-sensitive RealBoost algorithm is used to explore different costs among different types of misclassification in the boosting process in order to improve detection performance. In addition, we propose two strategies, selective classification and selective scale processing, to further accelerate the detection process at the channel feature level and image pyramid level respectively. Experiments on the Caltech and INRIA datasets show that the proposed method achieves the highest detection performance among all the state-of-the-art non-CNN methods and is about 148X faster than the existing best performing FCF method on the Caltech dataset.

CCS CONCEPTS

• **Computing methodologies → Object detection**;

KEYWORDS

Pedestrian detection, cost-sensitive RealBoost, vector form filter, embedded vision

1 INTRODUCTION

Pedestrian detection is an essential task in many autonomous driving and video surveillance systems [2, 17]. It is well recognized that vision based pedestrian detection is a challenging problem [2, 11, 17] due to high intra-class variation, highly cluttered background, inconsistent illumination, etc. This is aggravated by the fact that real-world applications necessitate that pedestrians are detected at high speed with high accuracy, while running on inexpensive systems with tight computational constraints. Recently, the Filtered Channel Feature (FCF) methods [8, 27, 41, 42] gained huge attention when they demonstrated high performance on well established pedestrian benchmarks. However, the existing FCF methods employ cost-insensitive Adaboost algorithm that assigns the same cost to different types of misclassification, which limits their detection performance potential. Furthermore, existing FCF methods either employ large numbers of filters [42] or perform filtering over high resolution channels [41], both of which incur high computational complexity.

In this paper, we employ vector form filters instead of the traditional matrix form filters which are widely used in existing FCF based approaches [27, 41, 42], to accelerate the most time-consuming step of FCF based methods (i.e. channel filtering step). Experimental results reveal that the proposed vector form filters not only resulted in significant reduction in run-time, but also led to better detection performance. Specifically, our proposed method employs three vector form filters (filter size of 1×1, 2×1 and 1×2 respectively) in each stage, for each of the 10 aggregated feature channels (3 LUV color channels, 1 gradient magnitude channel, and 6 channels for Histogram of Oriented Gradients (HOG)). The filtered channels are used to assemble a feature pool that is used for training the detector.

To solve the performance drawbacks of existing FCF methods, we develop a dual-stage group cost-sensitive RealBoost algorithm that explores different costs for different types of misclassification during the boosting process. Concretely, the training samples that are wrongly classified in the first stage are divided into different groups where each group is assigned a set of costs based on the posterior probability estimation from the first stage. The costs of each group correspond to penalties that enforces the boosting algorithm in the second stage to give larger emphasis to the harder training samples.

Finally, to further reduce the detection time without compromising on the detection performance, we incorporated two strategies: 1) *selective scale processing*, where the proposed dual-stage detection method is performed on a reduced number of image pyramid levels, and 2) *selective classification*, where an initial classification based on a coarse sliding window pattern is performed to obtain

MM'17, , October 23–27, 2017, Mountain View, CA, USA.

ACM ISBN 978-1-4503-4906-2/17/10…$15.00
https://doi.org/10.1145/3123266.3123303

responses that will determine if further classification at a finer granularity is necessary. We evaluated the proposed approach using the widely used Caltech and INRIA datasets and demonstrated that it can achieve the best detection performance and fastest run-time among all the state-of-the-art non-CNN methods.

The rest of the paper is organized as follows. A review of existing top-performing methods in pedestrian detection is presented in Section 2. Section 3 introduces the proposed dual-stage detection framework and two acceleration strategies: selective scale processing and selective classification. Section 4 presents the experimental results on the Caltech and INRIA datasets to demonstrate the effectiveness and efficiency of the proposed approach over state-of-the-art methods. We conclude the paper in Section 5.

2 RELATED WORK

Pedestrian detection can be typically decomposed into two steps, i.e. feature representation and classification. Based on the classification strategy, existing methods can be divided into three categories [2]: DPM (Deformable Part Models) variants [14, 15, 19, 36, 37], Convolutional Neural Network (CNN) [12, 32, 38, 39], and Decision Forest (DF) [8, 20, 27, 40, 42]. The current ranking on the Caltech evaluation platform shows that the top-performing methods belong to the latter two categories. Even though [18], [23] and [30] significantly improve the speed on the task of general object detection, they rely on very deep convolution neural networks (e.g. VGG [33]) that incur high computation complexity and typically require high-end Graphic Processing Units (GPUs) to meet real-time constraints. Experiments undertaken in [5, 22] show that energy and thermal constraints will limit the maximum achievable accuracy and run-time of deep learning algorithms on embedded GPUs for applications e.g. autonomous driving. As such, CNN-based methods are currently not well suited for realization on affordable and mass volume deployable embedded platforms that have tight computational constraints. In the remaining section, we will present a detailed review of the DF based methods for pedestrian detection.

All the current top-performing DF based methods are based on the Filtered Channel Features (FCF) framework [42]. In this framework, the detection process are decomposed into three correlated steps: aggregated channel extraction, filtering over aggregated channels and object classification. Existing FCF methods differ from the filters used in filtering step. The pioneering work in FCF [10] proposed a pedestrian detector called Integral Channel Feature (ICF) detector, where the sum of rectangular regions are adopted as feature vectors. In [9], FPDW is proposed to accelerate ICF. To further improve the detection performance, the authors in [8] proposed Aggregated Channel Feature (ACF) in which features are single pixel lookups from aggregated channels.

Following the success of ACF, more recent works e.g. LDCF [27], InformedHaar [40], Checkerboards [42] and RotatedFilters [41] are proposed. LDCF employed learned PCA eigenvectors as filters to remove correlation in local neighbourhoods so that they are well suited for orthogonal decision trees. In [42], a naive set of filters named Checkerboards is proposed to check how much the "informed" design of the filters is effective compared to other possible choices. Motivated by the orientated channels in HOG [8, 27] and the effectiveness of multiple scales in SquaresChnFtrs [2], RotatedFilters [41] proposed a set of rotated filters that are tailored towards different oriented HOG channels and integrated more local information by repeating each filter over multiple scales. Even though the Checkerboards [42] and RotatedFilters [41] achieved promising performance, the reported run-time for Checkerboards and RotatedFilters on workstation with single thread execution are only 43 and 16 seconds per frame respectively on the Caltech dataset [11], which prohibits their applicability in many real-time applications that run on tightly constrained embedded systems. In [1] and [4], the authors proposed to use GPU to accelerate the pedestrian detection and achieved notably improvement on detection speed. However, their work focuses on hardware acceleration which is orthogonal with our work. In addition, the performance of their work is considerably weak comparing top-performing FCF methods.

All of the above mentioned methods [8, 27, 40–42] focus on feature representation and employ cost-insensitive Adaboost learning algorithm [16] when training the detector. The cost-insensitive Adaboost assumes that different types of misclassification have equal importance and hence are assigned the same cost. However, this assumption does not usually hold in real-world applications [24]. For instance, in a pedestrian protection system for autonomous driving, mis-detection of pedestrians may induce deadly accidents. Various cost-sensitive boosting methods have been proposed to assign different costs to different types of misclassification including AdaCost [13], CSB0, CSB1, CSB2 [35], asymmetric-AdaBoost [25] and AdaC1, AdaC2, AdaC3 [34]. All of these algorithms are heuristic in nature, and they attempt to achieve cost-sensitivity by direct manipulation of the weights and confidence parameters of Adaboost [16]. Moreover, these algorithms are designed to deal with class-level cost-sensitivity by only assigning different costs to inter-class misclassification. This can lead to failure in capturing the large intra-class variants (examples of large intra-class variants are shown in Fig. 2(d)) which are commom in object detection tasks.

In order to better handle multi-resolution pedestrian detection, [43] proposed a group cost-sensitive Adaboost which explores different costs for different resolution subsets from positive samples during training. However, the groups are formulated based on the resolution of positive training samples, and hence they are affected by the quality of the positive samples that are often subjected to annotation errors [41]. The negative samples with complex background have much larger intra-variants than positive samples but are not explored. Consequently, the detection performance of [43] (20.20% log-average miss rate(MR)) is inferior comparing with existing FCF methods (e.g. Checkerboards [42] with MR of 18.47%).

2.1 Main Contributions

In this paper, we focus on improving the detection performance and at the same time, reducing the computational complexity of pedestrian detection. Fig. 1 shows the training and testing procedure of our proposed dual-stage pedestrian detection framework. The main contributions of this paper are summarized as follows:

1) Our work is the first to use vector form filters in the FCF framework. Unlike existing FCF methods that employ matrix form filters [27, 41, 42], our proposed vector form filters can

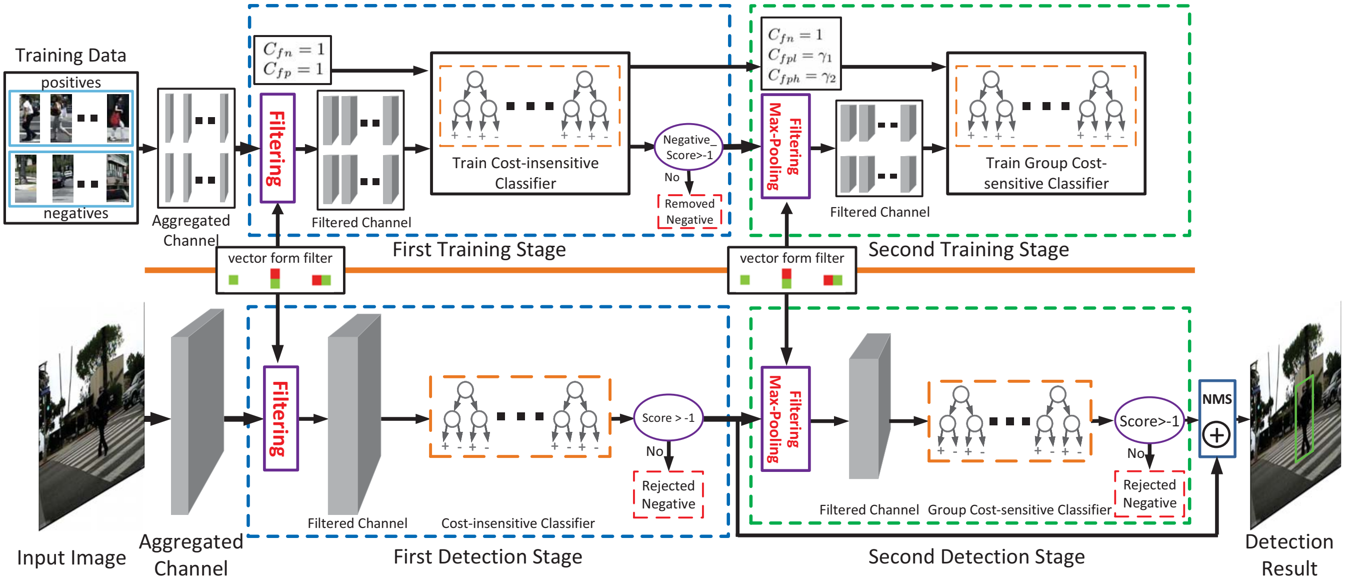


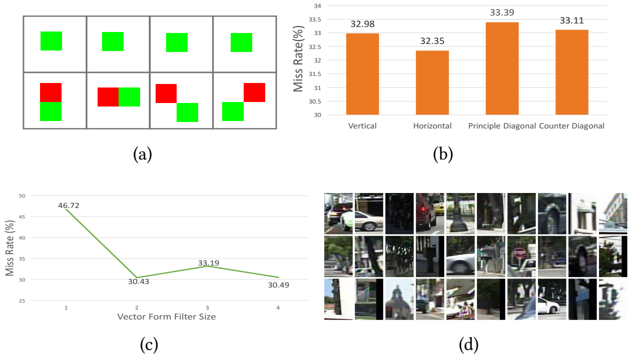
Figure 1: Proposed framework. In the first training stage, the cost of each group is the same. The groups are assigned different costs based on the posterior probability estimation for training in the second stage. During testing, the aggregated channels are extracted from the images and fed into our dual-stage detector. The final score comprises of average scores from the first and second stage and the detection result is obtained after Non-Maximum Suppression (NMS).

provide more discriminative features while at the same time leading to lower computational complexity.

2) We propose a novel dual-stage group cost-sensitive RealBoost learning algorithm for training the pedestrian detector. The posterior probability estimation of the negative training samples that are misclassified in the first stage are used to divide the negative samples into different groups. Each group is assigned a different cost and are trained in the second stage using proposed group cost-sensitive RealBoost algorithm.
3) In order to further accelerate the detection process, we incorporate strategies for selective classification and selective scale processing, to avoid unnecessary computations at the aggregated channel level and image pyramid level respectively. Results show that these strategies lead to significant execution time reduction while maintaining good detection performance.
4) We perform extensive evaluations on the proposed framework using the widely known Caltech and INRIA datasets. Our proposed framework achieves the best detection performance (i.e. 14.62% MR on Caltech dataset) among all the state-of-the-art non-CNN methods. In addition, the proposed method runs 148.8 times faster than the best performing FCF method in the literature (i.e. Checkerboards [42]).

3 PROPOSED METHOD

The analysis undertaken in [41, 42] reveals that the computational bottleneck of existing FCF methods lies in filtering over aggregated channels. This is due to fact that the existing top performing FCF methods either rely on large numbers of filters [42], or utilize high resolution feature maps [41]. In addition, existing FCF approaches adopt cost-insensitive Adaboost learning strategy which limits its the detection performance potential. Our work aims to address the drawbacks of existing FCF approaches by adopting vector form filters and a group cost-sensitive RealBoost learning strategy. The proposed vector form filters (Section 3.1) were able to extract more discriminative pedestrian features which contribute to improved detection performance. At the same time, they contribute to lowering the computational complexity of the filtering step. The proposed dual-stage group cost-sensitive RealBoost learning strategy (Section 3.2) enforces the boosting process to pay larger emphasis to the harder training samples, which led to higher detection performance. Finally, we employ strategies to accelerate the detection process



(a)

(b)

(c)

(d)

Figure 2: a) Four types of vector form filters, from left to right: vertical, horizontal, principal diagonal and counter diagonal, b) average MR of the vector form filters on Caltech 10x validation set, c) average MR with varying vertical and horizontal vector form filter sizes on Caltech 10x validation set, and d) large intra-class variants observed in the hard negative samples.

using selective scale processing and selective classification (Section 3.3).

3.1 Vector Form Channel Filter Bank

Existing top-performing FCF methods exploit matrix form filters to extract discriminative features. For instance, LDCF employed 4 filters with size 5×5 [27] and RotatedFilters utilized 9 filters with size 16×16 [41] per channel respectively. The utilization of matrix form filters contribute to high computational complexity in the filtering process. To alleviate this computational bottleneck, we propose to use vector form filters and investigated the impact of different vector form filters on the detection performance. We setup a validation environment by splitting the Caltech 10x training set into five training sets and one testing set as suggested by [11]. The parameters used in the cross-validation is identical to those used in the experiments described in Section 4. The log-average miss rate (MR), which is calculated as False Positive Per Image (FPPI) in $[10^{-1}, 10^{-0}]$, is used to evaluate the detection performance.

Four types of vector form filters as shown in Fig. 2(a) are investigated, i.e. vertical, horizontal, principal diagonal and counter diagonal. The filters are generated to enable gradient detection in different orientations for a given size. Note that the filters shown in the first row of Fig. 2(a) are 1x1 vector filters with weight 1 which means the aggregated channel feature is used. The detection performance of the four types of filters on cross-validation experiments are shown in Fig. 2(b). It can be observed that the detection performance of diagonal filters are worse than the vertical and horizontal filters, which implies that the feature combination in the vertical or horizontal orientations are more discriminative than the diagonal orientations. Based on the results of the cross-validation experiments, we proposed to use vertical and horizontal filters with same length in the detection framework.

The detection performance with varying vertical and horizontal filter sizes is shown in Fig. 2(c). It can be observed that the average MR increases when the filter size is larger than two. This phenomenon is mostly caused by overfitting as larger filters capture pixel differences at larger distances and hence, are less correlated. Based on these results, we employ three vector form filters with size 1×1, 2×1 and 1×2 in the proposed framework. Our first stage pedestrian detector using these three vector form filters achieves MR of 15.66%, which outperforms all existing FCF methods (e.g. Checkerboards with MR of 18.47%, and RotatedFilter with MR of 19.20%). This demonstrates that simple vector form filters can achieve better detection performance compared to the commonly used matrix form filters.

3.2 Dual-Stage Group Cost-sensitive RealBoost

In this subsection, we first give a formal definition of RealBoost and then we introduce our proposed dual-stage group cost-sensitive RealBoost algorithm.

Detection via RealBoost: Given a set of training samples for pedestrian detection $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x} = (x_1, \ldots, x_d)^T \in \mathbb{R}^d$ is the feature vector of each training sample and $y \in \{-1, 1\}$ is the label of the training sample. The detector aims to learn a function $G(\mathbf{x})$ that maps feature space to label space and can be expressed as:

$$G(\mathbf{x}) = sgn[F(\mathbf{x})] \tag{1}$$

$F(\mathbf{x})$ is a predictor, $sgn[.]$ is the sign function that returns 1 if $F(\mathbf{x}) > 0$ and -1 otherwise. In RealBoost, the predictor $F(\mathbf{x})$ is learned from a linear combination of weak learners in a greedy forward stagewise fashion [16]:

$$F(\mathbf{x}) = \sum_{m=1}^{M} f_m(\mathbf{x}) \tag{2}$$

The predictor is updated at each iteration according to:

$$F^{(t)}(\mathbf{x}) = F^{(t-1)}(\mathbf{x}) + f^{(t)}(\mathbf{x}) \tag{3}$$

where $f^{(t)}(\mathbf{x})$ is the learned weak learner in iteration t. The detector $G(\mathbf{x})$ is optimal if it minimizes the risk $E_{\mathbf{X},Y}[Loss(\mathbf{x}, y)]$, where $Loss(\mathbf{x}, y)$ is a loss function that measures the misclassification. In RealBoost [16], the zero-one loss is used to evaluate the misclassification and can be expressed as:

$$Loss(\mathbf{x}, y) = \begin{cases} 0, & \text{if } G(\mathbf{x}) = y \\ 1, & \text{if } G(\mathbf{x}) \neq y \end{cases} \tag{4}$$

Cost-sensitive RealBoost: The zero-one loss in Eq. 4 is cost-insensitive as it assigns identical cost for different types of misclassification: false positive ($y = -1, G(\mathbf{x}) = 1$) and false negative ($y = 1, G(\mathbf{x}) = -1$). However in many real-world applications, the classification techniques involve dramatically varying costs associated with different types of misclassification, which motivates the need for classification algorithms that focus on varying costs for different misclassification. In [26], a cost-sensitive RealBoost algorithm is proposed that assigns different costs for false positive and false negative, which can be expressed as:

$$Loss_c(\mathbf{x}, y) = \begin{cases} 0, & \text{if } G(\mathbf{x}) = y \\ C_{fn}, & \text{if } y = 1, G(\mathbf{x}) = -1 \\ C_{fp}, & \text{if } y = -1, G(\mathbf{x}) = 1 \end{cases} \tag{5}$$

where $C_{fn} > 0$ and $C_{fp} > 0$ are the cost for false negative and false positive respectively. The loss function in Eq. 5 is actually a class level cost-sensitive loss and a generalization of loss function in Eq. 4 since it reduces to zero-one loss when $C_{fn} = C_{fp}$.

Proposed Group Cost-sensitive RealBoost: The class-level cost-sensitive loss function in Eq. 5 can only capture variants of inter-class misclassification and is not well suited for pedestrian detection. In pedestrian detection, the negatives are often obtained from complex background through hard negative mining which leads to huge variants in the negative set [39] as shown in Fig. 2(d). In order to explore the variants in negatives and improve the detection performance, we proposed a group cost-sensitive RealBoost algorithm which allocates different costs based on the difficulty of negatives. Specifically, the negative set is divided into two distinct groups based on the posterior probability estimation $\eta(\mathbf{x}) = P_{\mathbf{X},Y}(1|\mathbf{x})$: negative samples with low posterior probability ($\eta(\mathbf{x}) < P_t$, denoted as $\mathbf{x}_l$), and negative samples with high posterior probability ($\eta(\mathbf{x}) >= P_t$, denoted as $\mathbf{x}_h$). The group cost-sensitive loss function can then be expressed as:

$$Loss_{gc}(\mathbf{x}, y) = \begin{cases} 0, & \text{if } G(\mathbf{x}) = y \\ C_{fn}, & \text{if } y = 1, G(\mathbf{x}) = -1 \\ C_{fpl}, & \text{if } y = -1, G(\mathbf{x}_l) = 1 \\ C_{fph}, & \text{if } y = -1, G(\mathbf{x}_h) = 1 \end{cases} \tag{6}$$

where $C_* > 0$. Note that this group cost-sensitive loss reduces to zero-one loss if C_{fn}, C_{fpl}, C_{fph} are 1 [16] and becomes class level cost-sensitive loss when $C_{fpl} = C_{fph}$ [26]. In object detection, it is essential to ensure that true positives can be correctly classified, as mis-detections are harder to recover. Therefore, the cost for false negative (C_{fn}) should be larger than false positive (C_{fpl} and C_{fph}). The optimal value of different costs can be chosen experimentally through cross-validation on training set. Note that the optimal costs are decided by the ratio C_{fpl}/C_{fn} and C_{fph}/C_{fn}, where C_{fn} can be set to one and the search becomes two-dimensional. In order to obtain posterior probability estimation $\eta(\mathbf{x})$, we propose a two-stage detection framework in which each of the negatives that are wrongly classified in first stage is assigned a corresponding posterior probability as follows:

$$\eta(\mathbf{x}) = \frac{e^{2F(\mathbf{x})}}{1 + e^{2F(\mathbf{x})}} \tag{7}$$

where $F(\mathbf{x})$ is the predictor trained in the first detection stage.

Note that the hypotheses that pass the first detection stage should have negligible difference in pixel values from the filtered feature channels. In order to distinguish harder negatives that are wrongly classified in first stage, we need to expand the receptive field of information extraction to integrate richer local information. Generally, there are two ways to expand the receptive field: use of larger filters and feature channel shrinkage. Larger filters can capture information from larger regions while feature channel shrinkage enables the same filter to cover a larger region. However, larger filters entail more filtering operations, which will not only increase the computational complexity but also increase the risk of overfitting as shown in Fig. 2(c). Therefore, in the second stage, we chose to use same filters with the first stage and perform feature channel shrinkage prior to filtering. Specifically, we adopt a 2 × 2 max-pooling with a stride of 2, downsampled in both the vertical and horizontal orientations. The feature channel shrinkage operation leads to 2 × 2 receptive field expansion from the local regions if we use the same filters as in the first stage while keeping some degrees of invariance with respect to translations and distortions. From the perspective of receptive field, 1 pixel represents a 4×4 pixel region in the first stage, and 8×8 pixel region in the second stage. The former one is determined by the shrinkage factor when computing aggregated channel feature while the second one is determined by both the shrinkage factor and the max-pooling operation (2×2 with stride of 2) adopted prior to filtering in the second stage.

The optimal detector $G(\mathbf{x})$ can be learnt through minimizing risk $E_{\mathbf{X},Y}[Loss(\mathbf{x}, y)]$ with respect to the corresponding loss defined in Eq. 4, Eq. 5 and Eq. 6. However, these minimizations are difficult to achieve. In RealBoost [16], exponential loss is employed to approximate zero-one loss, which induces the minimization of empirical risk as:

$$R(F) = \sum_{i=1}^{N} w_i e^{-yF(\mathbf{x})} \tag{8}$$

where w_i is weight of loss for training sample $\mathbf{x}_i$ and uniform weight distribution ($w_i = 1/N$) is adopted, which means each training sample share same importance. In the proposed method, we adopt RealBoost [16] and the exponential loss to approximate group cost-sensitive loss define in Eq. 6. After training the first detector with the minimization problem in Eq. 8, each negative that are wrongly classified is assigned a posterior probability estimation $\eta(\mathbf{x})$ using Eq. 7. The negative set is then separated into groups with low and high posterior probability based on the threshold P_t. With the costs $C_{fn} = 1, C_{fpl} = \gamma_1, C_{fph} = \gamma_2$ selected from cross-validation experiments, the group cost-sensitive detector can be trained by minimizing group cost-sensitive empirical risk based on exponential loss as:

$$\begin{aligned} R_{gc}(F) = & \sum_{y_i=1} w_i e^{-y_i C_{fn} F(\mathbf{x}_i)} \\ & + \sum_{\substack{y_i=-1 \\ \mathbf{x}\in\mathbf{x}_l}} w_i e^{-y_i C_{fpl} F(\mathbf{x}_i)} + \sum_{\substack{y_i=-1 \\ \mathbf{x}\in\mathbf{x}_h}} w_i e^{-y_i C_{fph} F(\mathbf{x}_i)} \end{aligned} \tag{9}$$

The above problem can be optimized using greedy forward stage-wise fashion [16]. After we have estimated $F^{(t-1)}(\mathbf{x})$, the weak learner $f^{(t)}(\mathbf{x})$ can be learned by solving the following:

$$\begin{aligned} f^{(t)}(\mathbf{x}) = & \arg\max_f -[\sum_{y_i=1} -w_i y_i C_{fn} e^{-y_i C_{fn} F^{(t-1)}(\mathbf{x}_i)} \\ & + \sum_{\substack{y_i=-1 \\ \mathbf{x}\in\mathbf{x}_l}} -w_i y_i C_{fpl} e^{-y_i C_{fpl} F^{(t-1)}(\mathbf{x}_i)} \\ & + \sum_{\substack{y_i=-1 \\ \mathbf{x}\in\mathbf{x}_h}} -w_i y_i C_{fph} e^{-y_i C_{fph} F^{(t-1)}(\mathbf{x}_i)}] f^{(t)}(\mathbf{x}_i) \\ & = \arg\max_f \sum_{y_i=1} w_{i_{fn}}^{(t)} y_i f^{(t)}(\mathbf{x}_i) \\ & + \sum_{\substack{y_i=-1 \\ \mathbf{x}\in\mathbf{x}_l}} w_{i_{fpl}}^{(t)} y_i f^{(t)}(\mathbf{x}_i) + \sum_{\substack{y_i=-1 \\ \mathbf{x}\in\mathbf{x}_h}} w_{i_{fph}}^{(t)} y_i f^{(t)}(\mathbf{x}_i) \end{aligned} \tag{10}$$

where

$$w_i^{(t)} = \begin{cases} w_i C_{fn} e^{-C_{fn} F^{(t-1)}(\mathbf{x}_i)}, & \text{if } y_i = 1 \\ w_i C_{fpl} e^{C_{fpl} F^{(t-1)}(\mathbf{x}_i)}, & \text{if } y_i = -1 \text{ and } \mathbf{x} \in \mathbf{x}_l \\ w_i C_{fph} e^{C_{fph} F^{(t-1)}(\mathbf{x}_i)}, & \text{if } y_i = -1 \text{ and } \mathbf{x} \in \mathbf{x}_h \end{cases} \tag{11}$$

is the weight of training samples for different groups at iteration t. The weight updating rule becomes cost-insensitive if $C_* = 1$ and class level cost-sensitive if $C_{fpl} = C_{fph}$. Compared to the cost-insensitive weight updating rule, the weight updating rule in Eq. 11 enables more emphasis on groups with higher cost. The proposed group cost-sensitive RealBoost for pedestrian detection is presented in Algorithm 1.

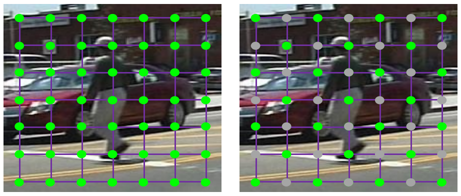


Figure 3: (a) Traditional sliding window based classification, and (b) proposed selective classification.

3.3 Strategies to Accelerate Detection

In order to further reduce the computation time of the proposed detection method, we introduce two acceleration strategies that are applied to the aggregated channel and image pyramid level: selective classification and selective scale processing.

Selective Classification: Traditional sliding window based methods perform classification at regular spaced image locations at a fine granularity, which often induces unnecessary computations since the responses for hypotheses at close proximity are usually highly correlated. The authors in [21] define that an object has a

Algorithm 1 Group Cost-sensitive RealBoost for Pedestrian Detection

```
Input: Training set $(\mathbf{x}_i, y_i)_{i=1}^{N}$, weight of loss for training samples $\{w_i\}_{i=1}^{N}$, a set of weak learner
    $\{f_i(\mathbf{x})\}_{i=1}^{K}$, group cost $\{C_{fn}, C_{fpl}, C_{fph}\}$, posterior probability threshold $P_t$, number of
    iteration $T$ and terminate threshold of empirical risk $\epsilon$
Output: Detector $G(\mathbf{x})$
Set $C_* = 1$;
Set $F_1^{(0)}(\mathbf{x}) = 0$, $w_i^{(1)} = w_i C_*$, $i = 1, \ldots, N$;
for $t_1 = 1$ to $T$ do
    Choose optimal weak learner $f_1^{(t_1)}(\mathbf{x})$ through solving problem in Eq. 10
    Update predictor $F_1^{(t_1)}(\mathbf{x}) = F_1^{(t_1-1)}(\mathbf{x}) + f_1^{(t_1)}(\mathbf{x})$;
    Update weights through Eq. 11;
    if $R_{gc}(F_1^{(t_1)}(\mathbf{x})) < \epsilon$ then
        break;
    end if
end for
Obtain $\eta(\mathbf{x})$ from $F_1^{(t_1)}(\mathbf{x})$ and divide negatives into two groups according to threshold $P_t$.
Set $C_{fn} = 1$, $C_{fpl} = \gamma_1$, and $C_{fph} = \gamma_2$;
Set $F_2^{(0)}(\mathbf{x}) = 0$, $w_i^{(1)} = w_i C_*$, $i = 1, \ldots, N$;
for $t_2 = 1$ to $T$ do
    Choose optimal weak learner $f_2^{(t_2)}(\mathbf{x})$ through solving problem in Eq. 10
    Update predictor $F_2^{(t_2)}(\mathbf{x}) = F_2^{(t_2-1)}(\mathbf{x}) + f_2^{(t_2)}(\mathbf{x})$;
    Update weights through Eq. 11;
    if $R_{gc}(F_2^{(t_2)}(\mathbf{x})) < \epsilon$ then
        break;
    end if
end for
return detector $G(\mathbf{x}) = sgn[F_1^{(t_1)}(\mathbf{x}) + F_2^{(t_2)}(\mathbf{x})]$
```

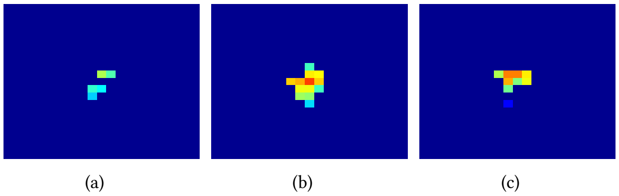

(a) (b) (c)

Figure 4: Hypothesis responses in three successive scales of the image pyramid. Image scale (b) has the highest positive response (indicated by the red region).

region of support (ROS) i.e. the neighbours of a positive location remain positive as illustrated in Fig. 4(b). This means that if the neighbours of a particular hypothesis are all positive, the hypothesis is most likely to be positive. On the contrary, a hypothesis tends to be negative if all of its neighbours are negative. Besides, most of these neighbours will be eliminated as NMS (Non-Maximum Suppression) only selects the hypothesis with highest response. This motivates us to exploit the responses from neighbours to reduce the number of classification operations.

Fig. 3 illustrates the traditional sliding window and the proposed sliding window approach for classification. The proposed sliding window approach first runs the classifier at coarse granularity (locations with green points). Their responses are then used to decide whether the classifier should be performed on the left neighbours (locations with gray points) or not. The pedestrian candidate is directly classified as positive if all its neighbours (up, bottom, right and left neighbours) are positive, and negative if all its neighbours are negative. Compared with the traditional sliding window method, our proposed method leads to notably lesser classification operations.

Selective Scale Processing: Existing image pyramid based methods first construct several pyramid levels of different image scales and then perform the detection on each pyramid level to detect objects of different sizes. The need to perform detection on different pyramid levels incur high computational complexity. In order to reduce the computational complexity, the Feature Approximation method is proposed in [8] which approximates multi-resolution image features via extrapolation from adjacent scales. However, the Feature Approximation method achieves faster detection at the cost of significant reduction in detection performance.

Our proposed strategy exploits the fact that the hypothesis response at adjacent scales are correlated since the ratio of adjacent scale factors are usually close to 1. Similar to ROS [21], we define this phenomena as *scale of support* (SOS), which means that a true positive will induce positive responses at adjacent scales. This phenomena is illustrated in Fig. 4, where the true positive with highest response in image scale Fig. 4(b) also has positive responses at the adjacent scales. Most of the positive responses from adjacent scales will be discarded in the NMS process. This motivates us to perform the detection process only on alternate scales of the pyramid in order to reduce the computational complexity.

4 RESULTS AND DISCUSSION

In this section, we evaluate the detection performance and execution time of the proposed method on two widely used datasets. To ensure a fair comparison for the execution time, we implemented all the methods on the same platform, i.e. 3.5GHz Intel Xeon E5-1650 CPU with single thread execution. We have not relied on GPUs in our experiments.

Datasets: Our experiments are based on two public datasets: Caltech [11] [1] and INRIA [7] [2]. For the Caltech dataset [11], the training data is augmented by extracting one of every 3 frames instead of every 30 frames from the raw videos, which is similar to the approach adopted by [42]. 42782 images are used to train our model. The Caltech test set consists of 4024 images which includes 1014 positive images. The evaluation metric is MR on False Positive Per Image (FPPI) in $[10^{-2}, 10^{-0}]$ under *reasonable* setup (pedestrians that are at least 50 pixels tall and at least 65% visible [11]). In addition, we also tested our model on the new annotations of Caltech test set provided by [41], which has corrected some errors in the original annotations. We denote the results of the original and new annotations as MR and MR$_{\_N}$ respectively. For the INRIA dataset [7], there are 614 positive images and 1218 negative images in the training set. The trained model is evaluated on 288 testing images using MR on FPPI ranges of $[10^{-2}, 10^{-0}]$.

Model Parameters: We use a model with size 64×128 when training the detector. For each stage, four rounds of hard negative mining (32, 512, 1024, 2048, 4096 trees respectively) are used and 100000 negatives are added to the training set at each round. During decision tree learning, we randomly selected 1/16 features from a large feature pool and the depth of the decision tree is limited to 5. The strides of both sliding window and aggregated channel shrinkage factor are 4, and each image is upsampled by one octave. The optimal value of costs for different groups are selected from

[1] http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/

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

$C_{fn} = 1, C_{fpl} \in [0.8 : 0.05 : 1)$ and $C_{fph} \in [0.8 : 0.05 : 1)$ while keeping $C_{fpl} < C_{fph}$ based on the cross-validation experiments.

4.1 Evaluation of Acceleration Strategies

In this subsection, we analyze the impact of the acceleration strategies discussed in Section 3.3 on the execution time and detection performance for the proposed cost-sensitive pedestrian detection framework on Caltech dataset [11]. We also implemented the Feature Approximation method in [8] within our framework to show the advantages of our acceleration strategies.

As discussed in Section 3, the computational bottleneck of existing FCF methods lies in the filtering step. It can be observed from Table 1 that the proposed method has led to significant reduction in run-time for the filtering step, as this is no longer the bottleneck of the detection process. The first row of Table 1 shows that our proposed method runs at 0.411 seconds per image on the Caltech dataset without any form of acceleration. It is noteworthy that the proposed method without acceleration is already the fastest among all existing methods that have MR lower than 15% on the Caltech dataset. The Feature Approximation method can significantly reduce the execution time as shown in the second row of Table 1, but at the cost of about 2% loss in MR. It can be observed from the third row of Table 1, that unlike the Feature Approximation, the proposed Selective Classification strategy simultaneously achieves lower execution time and higher detection performance than the proposed method without acceleration. To further reduce the execution time, we employ the proposed Selective Scale Processing strategy by selecting 12 alternate scales from the entire 27 scales for running the detection algorithm. When both the Selective Scale Processing and Selective Classification strategies are employed, the proposed method obtains a reduction in MR (last row of Table 1), most probably due to discarding some isolated hypotheses which have no *region of support* (ROS). With the combined acceleration strategies, our method can run at 0.291 second per frame while achieving MR of only 14.62% on the Caltech dataset. These results demonstrate the effectiveness of our acceleration strategies compared to the Feature Approximation method. In the remaining paper, only results of the proposed method with Selective Classification and Selective Scale Processing are reported.

4.2 Comparison with State-of-the-art Methods on the Caltech dataset

For Caltech dataset, The optimal value of C_{fpl} and C_{fph} in the proposed method are 0.85 and 0.9 respectively. Compared with traditional cost-insensitive loss in dual-stage detection framework, there is about 0.4% decline on MR when using proposed group cost-sensitive loss. Fig. 5 compares the detection performance between the proposed method and state-of-the-art methods on the Caltech dataset. It can be observed that the proposed method is superior to all other non-CNN methods. Compared to existing state-of-the-art FCF methods, the MR of our method is signficantly lower i.e. 14.62% whereas the MR of RotatedFilter [41] and Checkerboards [42] are 19.20% and 18.47% respectively. The proposed method still outperforms existing FCF methods when the evaluations are undertaken on the new annotations of Caltech test set. Although the proposed vector form filters are much simpler than those used in RotatedFilter and Checkerboards, they result in better detection performance. This demonstrates the effectiveness of the proposed vector filters and the cost-sensitive learning algorithm.

Among the CNN based methods, RPN+BF [12] achieves the lowest MR when preparing the paper. Though compACT-Deep [3], RPN+BF [39] and Fused DNN [12] have a better performance than our proposed method, their performance are achieved using pre-trained very deep models (e.g., VGG [33]) on ImageNet [31] that requires a large amount of convolution operations. The RPN+BF runs at about 0.5 second per frame on the Tesla K40 GPU (which is reported to have 10x computation power of parallel processing on a 16-core 3.1GHz CPU [3]). As highlighted in [29], discrete GPUs e.g. the Tesla K40 GPU are not suitable for automotive systems as they consume high power (which require active cooling) and occupy substantial space. Integrated GPUs (e.g. Jetson TX1), which are the preferred platforms for embedded systems, have significantly lower computation capabilities. For example, the run-time of CNN algorithms on the Jetson TX1 are reportedly 7X slower than the run-time on the Tesla K40 [28]. As such, existing CNN algorithms for pedestrian detection may not be able to meet the run-time requirements or suffer from accuracy loss if they are deployed on embedded platforms. On the other hand, the proposed method on a CPU with only a single processing thread can already achieve much lower run-time than the CNN based methods on GPUs with high detection performance.

Table 2 compares the computation time of the proposed method with state-of-the-art FCF methods. Note that here, we only perform comparisons with methods that have released their models, as we can execute them on a common platform. It can be observed that ACF has the fastest execution time but suffers from a very high MR. The ACF is faster than our method since they utilize a smaller model size (64 × 32) and perform detection on the original image while we exploit larger model size (128 × 64) and perform detection on upsampled image. In addition, the ACF method does not perform the channel filtering step which is the most time consuming step in the FCF methods. The RotatedFilters [41] and Checkerboards [42] achieve much lower MR compared with ACF and LDCF but have very high execution time. As discussed in Section 3, this is either caused by the need to filter over high resolution channels or the need for a large amount of filters. As shown in Table 2, the proposed method runs about 57.1 and 148.8 times faster than RotatedFilter [41] and Checkerboards [42] respectively while still achieving a much lower MR. The proposed method also runs significantly faster than the current top performing non-CNN method in the literature, NNNF [6] (i.e. 0.877 seconds per frame on a similar platform) while achieving lower MR. These results clearly demonstrate that our proposed method achieves the best trade-off between detection performance and speed among all the state-of-the-art pedestrian detection methods.

4.3 Comparison with State-of-the-art Methods on the INRIA dataset

Compared to the Caltech dataset, INRIA [7] has much lesser training images. There are 42782 images for hard negative mining in Caltech but only 1218 image in INRIA dataset. Therefore, we change some

[3] https://www.nvidia.com/content/tesla/pdf/nvidia-tesla-k40-2014mar-lr.pdf

Table 1: Average execution time per frame (seconds) and MR of proposed method on Caltech test with different acceleration strategies.

Feature Approximation	Selective Classification	Selective Scale Processing	Aggregated Channel (s)	Filtering (s)	Classification (s)	Total Time (s)	MR (%)
			0.218	0.061	0.132	0.411	14.75
√			0.071	0.060	0.113	0.244	17.01
	√		0.213	0.058	0.120	0.391	14.58
	√	√	0.153	0.044	0.094	0.291	14.62

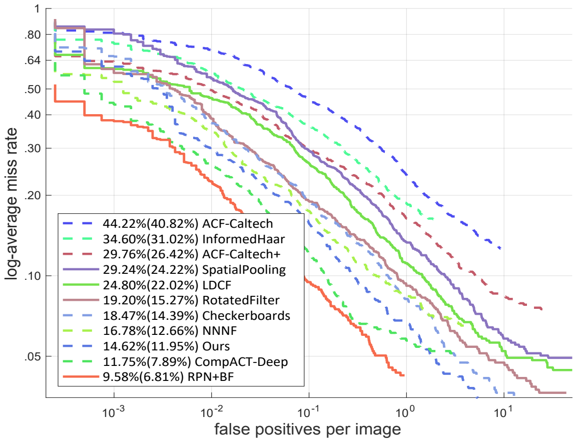


Figure 5: Detection performance comparison with state-of-the-art methods on Caltech dataset (legend indicates MR(MR$_{\_N}$)).

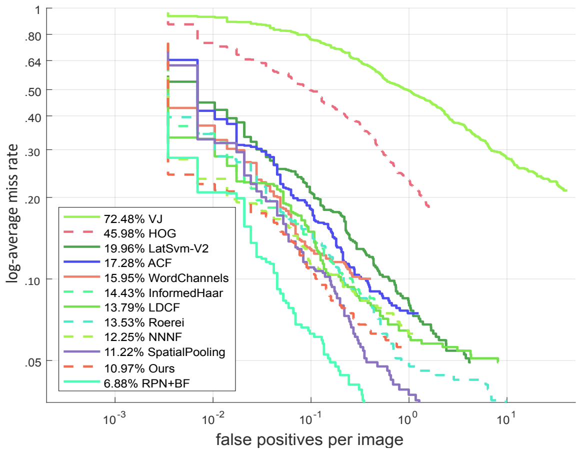


Figure 6: Detection performance comparison with state-of-the-art methods on the INRIA dataset.

Table 2: Average execution time per frame (seconds) and MR of filtered channel feature methods on Caltech dataset.

	Aggregated Channel (s)	Filtering (s)	Classification (s)	Total Time (s)	MR(MR$_{\_N}$) (%)
ACF	0.045	-	0.061	0.106	29.76(26.42)
LDCF	0.046	0.220	0.035	0.301	24.80(22.02)
RotatedFilters	0.384	9.309	6.931	16.625	19.20(15.27)
Checkerboards	0.496	22.117	20.709	43.319	18.47(14.39)
Ours	0.153	0.044	0.094	0.291	14.62(11.95)

Table 3: Average execution time per frame (seconds) and MR of filtered channel feature methods on INRIA dataset.

	Aggregated Channel (s)	Filtering (s)	Classification (s)	Total Time (s)	MR (%)
ACF	0.024	-	0.012	0.036	17.28
LDCF	0.051	0.243	0.105	0.399	13.79
Ours	0.075	0.027	0.019	0.121	10.97

parameters to adjust the smaller dataset. Each stage detector is trained via three rounds of hard negative mining (32, 128, 512, 4096 trees respectively) and 20000 negatives are added to each round. The depth of decision tree is limited to 2 when training weak learner. Note that we only perform comparisons on the execution time with FCF methods that have released their models for the INRIA dataset.

The detection performance of the proposed method and the state-of-the-art methods are shown in Fig. 6. It can be observed that our proposed method achieves best detection performance (MR is 10.97%) among all the non-CNN methods. The MR of the proposed method is 6.31% and 2.82% lower than ACF [8] and LDCF [27] respectively. The detection performance of our proposed method is much better than NNNF [6], which employs more complex Haar-like features. These results on the INRIA dataset further confirm that the simple vector form filters and cost-sensitive learning strategy can lead to significant detection performance improvement. The execution time of ACF [8], LDCF [27] and proposed method are shown in Table 3. Even though ACF runs at 0.036 seconds per image, the high MR limits its practicality in real-world applications. LDCF achieves a much lower MR but its execution time is about 3 times higher than the proposed method and its MR is also about 2.82% higher than the proposed method. These results further demonstrate the effectiveness and efficiency of the proposed method.

5 CONCLUSION

We proposed an accurate and run-time efficient pedestrian detection method that exploits vector form filters to capture more discriminative features and a group cost-sensitive RealBoost algorithm to explore the intra-class variants of training samples to improve detection performance. The vector form filters are used in a dual-stage detection framework that relies on a learned cascade detector to capture the discriminative features from aggregated channels of different resolutions in two stages. The combination of simple vector form filters and the cost-sensitive learning framework resulted in the best detection performance among all state-of-the-art non-CNN based methods. In addition, the proposed method achieves about one order of magnitude speedup over existing filtered channel feature methods. In order to further improve the speed of detection, we adopted a selective classification strategy which is inspired by the principle of *region of support* (ROS), and proposed a selective scale processing strategy which is motivated by the principle of *scale of support* (SOS). These strategies led to further speedup improvement in our proposed method without compromising on the detection performance.

REFERENCES

[1] Rodrigo Benenson, Markus Mathias, Radu Timofte, and Luc Van Gool. 2012. Pedestrian detection at 100 frames per second. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on.* IEEE, 2903–2910.

[2] Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. 2014. Ten years of pedestrian detection, what have we learned?. In *European Conference on Computer Vision.* Springer, 613–627.

[3] Zhaowei Cai, Mohammad Saberian, and Nuno Vasconcelos. 2015. Learning complexity-aware cascades for deep pedestrian detection. In *Proceedings of the IEEE International Conference on Computer Vision.* 3361–3369.

[4] Victor Campmany, Sergio Silva, Antonio Espinosa, Juan Carlos Moure, David Vázquez, and Antonio M López. 2016. GPU-based pedestrian detection for autonomous driving. *Procedia Computer Science* 80 (2016), 2377–2381.

[5] Alfredo Canziani, Adam Paszke, and Eugenio Culurciello. 2016. An Analysis of Deep Neural Network Models for Practical Applications. *arXiv preprint arXiv:1605.07678* (2016).

[6] Jiale Cao, Yanwei Pang, and Xuelong Li. 2015. Pedestrian Detection Inspired by Appearance Constancy and Shape Symmetry. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015).

[7] Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, Vol. 1. IEEE, 886–893.

[8] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. 2014. Fast feature pyramids for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 8 (2014), 1532–1545.

[9] Piotr Dollár, Serge J Belongie, and Pietro Perona. 2010. The Fastest Pedestrian Detector in the West.. In *BMVC*, Vol. 2. 7.

[10] Piotr Dollár, Zhuowen Tu, Pietro Perona, and Serge Belongie. 2009. Integral channel features. (2009).

[11] Piotr Dollar, Christian Wojek, Bernt Schiele, and Pietro Perona. 2012. Pedestrian detection: An evaluation of the state of the art. *IEEE transactions on pattern analysis and machine intelligence* 34, 4 (2012), 743–761.

[12] Xianzhi Du, Mostafa El-Khamy, Jungwon Lee, and Larry Davis. 2017. Fused DNN: A deep neural network fusion approach to fast and robust pedestrian detection. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on.* IEEE, 953–961.

[13] Wei Fan, Salvatore J Stolfo, Junxin Zhang, and Philip K Chan. 1999. AdaCost: misclassification cost-sensitive boosting. In *Icml.* 97–105.

[14] Pedro F Felzenszwalb, Ross B Girshick, and David McAllester. 2010. Cascade object detection with deformable part models. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on.* IEEE, 2241–2248.

[15] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. 2010. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence* 32, 9 (2010), 1627–1645.

[16] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. 2000. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics* 28, 2 (2000), 337–407.

[17] David Geronimo, Antonio M Lopez, Angel D Sappa, and Thorsten Graf. 2010. Survey of pedestrian detection for advanced driver assistance systems. *IEEE transactions on pattern analysis and machine intelligence* 32, 7 (2010), 1239–1258.

[18] Ross Girshick. 2015. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision.* 1440–1448.

[19] Ross Brook Girshick. 2012. *From rigid templates to grammars: Object detection with structured models.* Citeseer.

[20] Alejandro González, Gabriel Villalonga, Jiaolong Xu, David Vázquez, Jaume Amores, and Antonio M López. 2015. Multiview random forest of local experts combining rgb and lidar data for pedestrian detection. In *Intelligent Vehicles Symposium (IV), 2015 IEEE.* IEEE, 356–361.

[21] Giovanni Gualdi, Andrea Prati, and Rita Cucchiara. 2010. Multi-stage sampling with boosting cascades for pedestrian detection in images and videos. In *European Conference on Computer Vision.* Springer, 196–209.

[22] Shaoshan Liu, Jie Tang, Zhe Zhang, and Jean-Luc Gaudiot. 2017. CAAD: Computer Architecture for Autonomous Driving. *arXiv preprint arXiv:1702.01894* (2017).

[23] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2015. Ssd: Single shot multibox detector. *arXiv preprint arXiv:1512.02325* (2015).

[24] Aurlie C Lozano and Naoki Abe. 2008. Cost-sensitive Boosting with p-norm Loss Functionsand its Applications. *MI lecture note series* 12 (2008), 65–74.

[25] Hamed Masnadi-Shirazi and Nuno Vasconcelos. 2007. Asymmetric boosting. In *Proceedings of the 24th international conference on Machine learning.* ACM, 609–619.

[26] Hamed Masnadi-Shirazi and Nuno Vasconcelos. 2011. Cost-sensitive boosting. *IEEE Transactions on pattern analysis and machine intelligence* 33, 2 (2011), 294–309.

[27] Woonhyun Nam, Piotr Dollár, and Joon Hee Han. 2014. Local decorrelation for improved detection. *NIPS* (2014).

[28] Varun Praveen OEdwin L. Weill, Jesse Tetreault and Melissa Smith. [n. d.]. 50. DeepROAD: A Multifaceted Deep Learning Suite for Real-Time Optimized Autonomous Driving. ([n. d.]).

[29] Nathan Otterness, Ming Yang, Sarah Rust, Eunbyung Park, James H Anderson, F Donelson Smith, Alex Berg, and Shige Wang. [n. d.]. An Evaluation of the NVIDIA TX1 for Supporting Real-time Computer-Vision Workloads. ([n. d.]).

[30] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 779–788.

[31] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* 115, 3 (2015), 211–252.

[32] Pierre Sermanet, Koray Kavukcuoglu, Soumith Chintala, and Yann LeCun. 2013. Pedestrian detection with unsupervised multi-stage feature learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 3626–3633.

[33] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[34] Yanmin Sun, Andrew KC Wong, and Yang Wang. 2005. Parameter inference of cost-sensitive boosting algorithms. In *International Workshop on Machine Learning and Data Mining in Pattern Recognition.* Springer, 21–30.

[35] Kai Ming Ting. 2000. A comparative study of cost-sensitive boosting algorithms. In *In Proceedings of the 17th International Conference on Machine Learning.* Citeseer.

[36] Jiaolong Xu, Sebastian Ramos, David Vázquez, and Antonio M López. 2014. Domain adaptation of deformable part-based models. *IEEE transactions on pattern analysis and machine intelligence* 36, 12 (2014), 2367–2380.

[37] Junjie Yan, Zhen Lei, Longyin Wen, and Stan Z Li. 2014. The fastest deformable part model for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2497–2504.

[38] Bin Yang, Junjie Yan, Zhen Lei, and Stan Z Li. 2015. Convolutional channel features. In *Proceedings of the IEEE International Conference on Computer Vision.* 82–90.

[39] Liliang Zhang, Liang Lin, Xiaodan Liang, and Kaiming He. 2016. Is Faster R-CNN Doing Well for Pedestrian Detection?. In *European Conference on Computer Vision.* Springer, 443–457.

[40] Shanshan Zhang, Christian Bauckhage, and Armin B Cremers. 2014. Informed haar-like features improve pedestrian detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition.* 947–954.

[41] Shanshan Zhang, Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. 2016. How Far are We from Solving Pedestrian Detection?. In *IEEE Conference on Computer Vision and Pattern Recognition.* IEEE Computer Society.

[42] Shanshan Zhang, Rodrigo Benenson, and Bernt Schiele. 2015. Filtered channel features for pedestrian detection. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE, 1751–1760.

[43] Chao Zhu and Yuxin Peng. 2016. Group cost-sensitive boosting for multi-resolution pedestrian detection. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence.* AAAI Press, 3676–3682.