Multi-label Learning of Part Detectors for Heavily Occluded Pedestrian Detection

Chunluan Zhou Junsong Yuan
School of Electrical and Electronic Engineering
Nanyang Technological University, Singapore
czhou002@e.ntu.edu.sg, jsyuan@ntu.edu.sg

Abstract

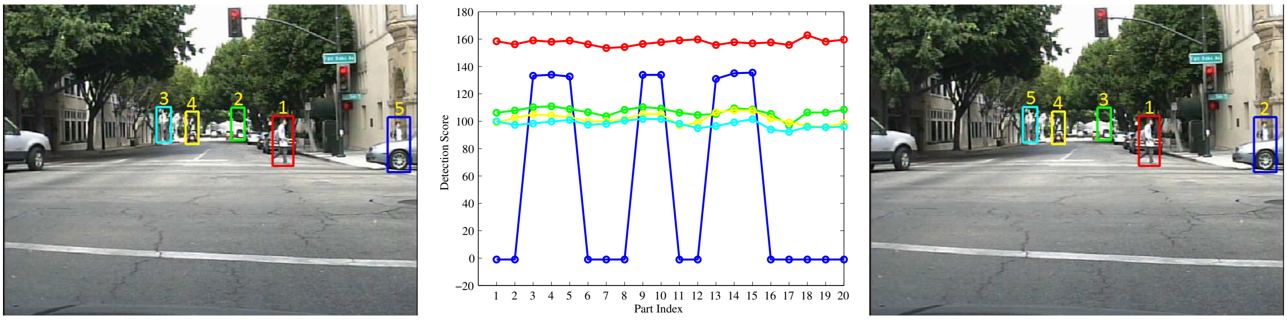
Detecting pedestrians that are partially occluded remains a challenging problem due to variations and uncertainties of partial occlusion patterns. Following a commonly used framework of handling partial occlusions by part detection, we propose a multi-label learning approach to jointly learn part detectors to capture partial occlusion patterns. The part detectors share a set of decision trees via boosting to exploit part correlations and also reduce the computational cost of applying these part detectors. The learned decision trees capture the overall distribution of all the parts. When used as a pedestrian detector individually, our part detectors learned jointly show better performance than their counterparts learned separately in different occlusion situations. The learned part detectors can be further integrated to better detect partially occluded pedestrians. Experiments on the Caltech dataset show state-of-the-art performance of our approach for detecting heavily occluded pedestrians.

1. Introduction

Occlusions present a great challenge for pedestrian detection in real-world applications. Most state-of-the-art pedestrian detection approaches [20, 35, 33, 4] train a full-body detector and show promising performance for detecting pedestrians which are non-occluded or slightly occluded. These approaches would often suffer when pedestrians are heavily occluded. For example, RPN+BF [33] achieves a log-average miss rate of 9.6% on the *Reasonable* subset of the Caltech dataset [8], but its performance drops dramatically to 74% on the *Heavy* subset in which pedestrian examples are heavily occluded. Since most of the body of a heavily occluded pedestrian is invisible, a full-body detector would probably be misled by the background region inside its detection window so that it tends to miss the pedestrian. As shown in Fig. 1(a-b), a heavily occluded pedestrian (Blue bounding box) is only ranked at 5th among five detections by a full-body detector.

A common solution to occlusion handling for pedestrian detection is to learn a set of part/occlusion-specific detectors which can be properly integrated to detect partially occluded pedestrians [10, 9, 16, 14, 19, 17, 36, 26]. When a full-body detector fails to detect a partially occluded pedestrian, the detectors of the parts which are still visible may give high detection scores (See Fig. 1(b-c)). For this solution, the reliability of part detectors is of great importance, since part detectors are its building blocks. Usually, part detectors are learned independently [10, 9, 16, 14, 19, 36, 26]. This way of learning part detectors has two drawbacks: (1) Correlations among parts are ignored during learning, which would affect the reliability of the learned part detectors; (2) The computational cost of applying a set of part detectors increases linearly with the number of parts. In [17], part detector learning and integration are done in a single convolutional neural network. However, this approach only uses class labels and part detectors are learned implicitly in a weakly supervised fashion. We believe part-level supervision can be exploited to further improve its performance.

In this paper, we propose a multi-label learning approach to jointly learn part detectors. The goal of joint learning is to (1) improve the part detectors by exploiting correlations among parts, *e.g.* some parts of the body tend to appear/disappear together, and (2) reduce the computational cost of applying the learned part detectors for pedestrian detection. Since the combination of boosting and decision trees works well for pedestrian detection [6, 35, 5, 33], we choose decision trees to form our part detectors. However, instead of training a set of decision trees for each part detector, we only construct one set of decision trees which are shared by all the part detectors. To exploit part correlations, these decision trees are learned and combined to capture the overall distribution of all the parts. We adapt AdaBoost.MH [22], which is a multi-class, multi-label version of AdaBoost, to learn these decision trees. When used for pedestrian detection individually, the part detectors learned



(a) (b) (c)

Figure 1. Occlusion handling. (a) Top five detections of a full-body detector. The heavily occluded pedestrian (Blue bounding box) is only ranked at 5th. (b) Scores of the five detections in (a) given by 20 part detectors. Each curve shows the 20 scores of one detection in the same color. The first detector is the full-body detector. Figure 2 shows the 20 parts. (c) The five detections in (a) are re-ranked by properly integrating the 20 part detectors. The heavily occluded pedestrian (Blue bounding box) is ranked at 2nd.

by our approach show better performance than their counterparts learned separately in different occlusion situations. By proper integration, the learned part detectors can be used to handle occlusions, which further improves the performance for detecting partially occluded pedestrians. The effectiveness of our approach is demonstrated on the Caltech dataset [8]. We apply the proposed multi-label learning approach to channel features [15] and features learned by a convolutional neural network [33] respectively. Using channel features our approach improves state-of-the-arts for detecting pedestrians in different occlusion situations while using deep learning features our approach shows comparable performance for detecting pedestrians that are non-occluded or slightly occluded and achieves the best performance for detecting partially occluded pedestrians, especially the heavily occluded ones.

2. Related work

In the past decade, many efforts have been made to improve pedestrian detection [8, 2]. Two major categories of pedestrian detection approaches are channel-feature based approaches [7, 1, 14, 6, 34, 20, 15, 35] and deep-learning based approaches [16, 17, 12, 31, 27, 26, 4, 3, 33]. For the former, decision trees are usually learned by applying boosting to channel features to form a pedestrian detector. Pedestrian detection is carried out in a sliding-window fashion. For the latter, a deep neural network is trained to either form a pedestrian classifier [16, 17, 12, 27, 26, 3] or generate features which are combined with other types of classifiers for pedestrian detection [31, 4, 33]. This category of approaches usually perform detection by classifying a set of pedestrian proposals.

Many approaches have been proposed to handle occlusions for pedestrian detection. The approach in [13] adopts an implicit shape model to generate a set of pedestrian hypotheses which are further refined using local and global cues to obtain their visible regions. In [29], a pedestrian template is divided into a set of blocks and occlusion reasoning is conducted by estimating the visibility status of each block. A probabilistic framework [18] is proposed to exploit multi-pedestrian detectors to aid single-pedestrian detectors, which can handle partial occlusions especially when pedestrians occlude each other. In [25, 21], a set of occlusion patterns are discovered to capture occlusions of single pedestrians and multiple pedestrians and then a deformable part model [11] is employed to learn a mixture of occlusion-specific detectors. A widely used occlusion handling strategy for pedestrian detection is to learn a set of part detectors which are then fused properly for detecting partially occluded pedestrians [30, 23, 10, 9, 16, 14, 19, 17, 36, 26]. In these approaches part detectors are usually learned separately, while in our approach part detectors are learned jointly so as to exploit part correlations for improving these detectors and reduce the computational cost of applying multiple part detectors for pedestrian detection.

3. Joint learning of part detectors

3.1. Part representation

We model the whole body of a pedestrian as a rectangular template without distinguishing different viewpoints. The template is divided into an $H \times W$ grid. Any rectangular sub-region in the template is considered as a part. Mathematically, a part can be expressed as a 4-tuple $p = (l, t, r, b)$, where (l, t) and (r, b) are the coordinates of the top-left and bottom-right corners of the part respectively with $0 \leq l < r \leq W$ and $0 \leq t < b \leq H$. In our implementation, we set $H = 6$ and $W = 3$. According to prior knowledge that pedestrians are usually occluded from the left, right or bottom, we manually design a pool of parts as shown in Figure 2. The part pool can be expressed as $\mathcal{P} = \{p_k | 1 \leq k \leq K\}$, where $p_k = (l_k, t_k, r_k, b_k)$ and $K = 20$. For pedestrian detection, images are usually

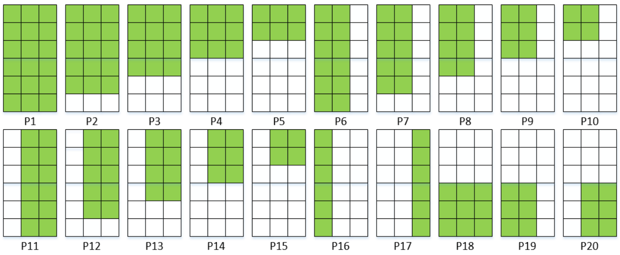


Figure 2. Part pool. Green regions denote parts. The first part is the whole body which is modeled as a template of 6 × 3 cells. Parts 2 to 17 are designed to handle situations where occlusions occur from the left, right or bottom and the last three parts are used for detecting the lower body.

represented by a set of feature maps, *e.g.* locally decorrelated channel features (LDCF) [15] and convolutional channel features (CCF) [31]. To represent a part on a pedestrian, a natural way is to crop regions that correspond to the part from the feature maps. One drawback of this representation is that small parts on a pedestrian are difficult to be reliably detected as the information from the small parts is relatively limited compared with that from large parts, especially when the pedestrian is small. Instead, we represent a part using features from the whole body. Features from the surrounding region of the part can be taken as its context. In this way, all the parts are represented by the same features.

3.2. Multi-label formulation

Let $\mathcal{X}$ be an instance space which consists of image regions. For each part $p_k \in \mathcal{P}$, we want to learn a detector $d_k : \mathcal{X} \rightarrow \mathbb{R}$ such that for an image region $x \in \mathcal{X}$, $d_k(x) > 0$ if the image region contains p_k and $d_k(x) \leq 0$ otherwise. A direct solution is to learn the K part detectors separately. However, this solution ignores correlations among the parts. For example, according to the part definition in Section 3.1, if a part appears in an image region, any smaller part within this part should also be visible. To exploit potential correlations among the parts, we propose a multi-label learning approach to learn the part detectors jointly.

Specifically, we learn a function $F : \mathcal{X} \rightarrow 2^{\mathcal{P}}$ to predict an arbitrary set of parts $P \subseteq \mathcal{P}$ which appear in an given image region x. Let $\mathcal{D} = \{(x_i, l_i, B_i^v, B_i^f)|1 \leq i \leq N\}$ be a set of training examples, where $x_i \in \mathcal{X}$ is an image region, $l_i \in \{-1, 1\}$ indicates whether the image region contains a pedestrian and if so, B_i^v and B_i^f are two bounding boxes specifying the visible portion and full body of the pedestrian respectively. To learn F, we need to construct for each instance x_i a label vector $\mathbf{Y}_i = (y_{ik}) \in \{-1, 1\}^K$ for $1 \leq k \leq K$, where y_{ik} indicates whether the image region x_i contains the part p_k. When a part is only partially visible on a pedestrian, it is not trivial to assign 1 or -1 to the part. Wrong assignment of part labels may cause the learning of part detectors to fail. So, we introduce a cost vector $\mathbf{C}_i = (c_{ik}) \in \mathbb{R}^K$ for $1 \leq k \leq K$ to soften the label assignment, where c_{ik} $(0 \leq c_{ik} \leq 1)$ defines the cost incurred if a wrong prediction is made on x_i for p_k. For $l_i = -1$, we set $y_{ik} = -1$ and $c_{ik} = 1$. For $l_i = 1$, we set $y_{ik} = 1$ and determine the cost c_{ik} based on B_i^v and B_i^f. We first standardize the full-body bounding box B_i^f as in [8]: the bounding-box is adjusted such that after the adjustment the ratio of its width to its height is 0.41 with its height and center coordinates unchanged. Denote by R_i^f the standardized full body. Then, any image contents inside the visible portion B_i^v but outside the standardized full body R_i^f are discarded to obtain a new visible portion R_i^v. Let R_{ik}^p be the image region of the part p_k on the instance x_i. We calculate the intersection over union (IOU) between R_{ik}^p and R_i^v denoted by I_{ik}, and the proportion of R_{ik}^p covered by R_i^v denoted by O_{ik}. Finally, the cost c_{ik} is determined as follows:

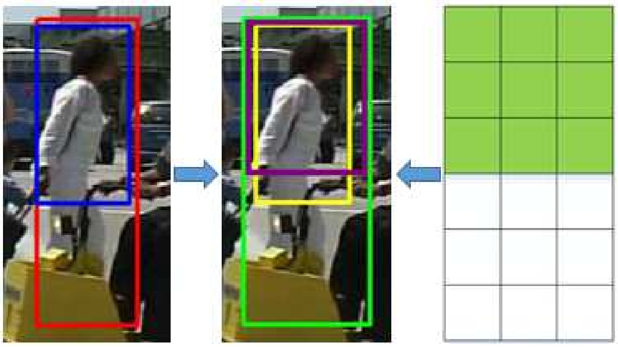


Figure 3. Example labeling. The blue and red bounding boxes are the visible portion and full body of the pedestrian example respectively. The green bounding box is the standardized full body and the yellow bounding box is the new visible portion inside the standardized full body. The purple bounding box shows the image region of the upper-body part on the pedestrian example. The cost vector is calculated based on the purple and yellow bounding boxes.

$$c_{ik} = \begin{cases} O_{ik} & O_{ik} \geq 0.7; \\ I_{ik} & O_{ik} < 0.7 \text{ and } I_{ik} \geq 0.5; \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

In the first case, the majority of the part p_k is visible on the instance x_i, so a large cost is set to prevent the part from being wrongly predicted. In the second case, the IOU between the part and visible portion is over 0.5. We thus consider the part to be visible on the instance and the cost of wrongly classifying it depends on the IOU. In the third case, the part is largely occluded. We set $c_{ik} = 0$ to discard this training example for the k-th part. Figure 3 illustrates how a pedestrian example is labeled. $\mathcal{D}_F = \{(x_i, \mathbf{Y}_i, \mathbf{C}_i)|1 \leq i \leq N\}$ forms the training set for learning F. We identify F with a two-argument function $H : \mathcal{X} \times \mathcal{P} \rightarrow \mathbb{R}$ such that

$p_k \in F(x)$ if $H(x, p_k) > 0$ and $p_k \notin F(x)$ otherwise. For any predicate π, let $[\pi]$ be 1 if π holds and 0 otherwise. We learn H by minimizing the following loss function:

$$L(H, \mathcal{D}_F) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} c_{ik}[\text{sign}(H(x_i, p_k)) \neq y_{ik}], \quad (2)$$

where $\text{sign}(H(x_i, p_k)) = 1$ if $H(x_i, p_k) > 0$ and $\text{sign}(H(x_i, p_k)) = -1$ otherwise.

3.3. Learning via boosting

Since the combination of boosting and decision trees has shown promising performance for pedestrian detection [6, 35, 5, 33], we choose decision trees to form our part detectors. We consider two approaches to minimizing the loss function $L(H, \mathcal{D}_F)$ in Eq. (2) for learning the part detectors.

The first approach learns the part detectors separately. Define the training loss related to the k-th part detector by

$$L_k(H, \mathcal{D}_F) = \frac{1}{N} \sum_{i=1}^{N} c_{ik}[\text{sign}(H(x_i, p_k)) \neq y_{ik}]. \quad (3)$$

$L(H, \mathcal{D}_F)$ can be decomposed as

$$L(H, \mathcal{D}_F) = \sum_{k=1}^{K} L_k(H, \mathcal{D}_F). \quad (4)$$

$L(H, \mathcal{D}_F)$ can be minimized by minimizing $L_k(H, \mathcal{D}_F)$ for $1 \leq k \leq K$ separately. Let $Q_k = \sum_{i=1}^{N} c_{ik}$. We normalize the costs associated with the k-th part by Q_k to obtain $\mathbf{D}_k = (c_{1k}/Q_k, ..., c_{Nk}/Q_k)$. $\mathbf{D}_k$ can be considered as a distribution over the training examples of the k-th part in $\mathcal{D}_F$. Boosting can be applied to learn and combine T decision trees to form a detector for the k-th part with example weights initialized to $\mathbf{D}_k$. This learning approach has two limitations: (1) Correlations among the parts are ignored; (2) The computational costs of training and testing increase linearly with the number of parts.

To address the limitations of the separate learning approach, we propose another approach to learn the K part detectors jointly. Instead of learning T decision trees for each part detector, we only learn T decision trees which are shared by all the part detectors. We adapt AdaBoost.MH [22], which is a multi-class, multi-label version of AdaBoost, to learn H of the form:

$$H(x, p) = \sum_{t=1}^{T} \alpha_t h_t(x, p), \quad (5)$$

where $h_t : \mathcal{X} \times \mathcal{P} \to \mathbb{R}$ is a weak classifier which is a decision tree in our case and α_t is a weight associated with h_t. First, we consider a simplified case in which $c_{ik} = 1$ for $1 \leq i \leq N$ and $1 \leq k \leq K$. AdaBoost.MH can be directly applied to minimize $L(H, \mathcal{D}_F)$. The idea of AdaBoost.MH is to reduce the multi-label learning problem to a binary classification problem for which AdaBoost can be used to obtain H. Each training example $(x_i, \mathbf{Y}_i, \mathbf{C}_i) \in \mathcal{D}_F$ is replaced with K training examples $((x_i, p_k), y_{ik})$ for $1 \leq k \leq K$. Note that since $c_{ik} = 1$ for all i and k, $\mathbf{C}_i$ in the example $(x_i, \mathbf{Y}_i, \mathbf{C}_i)$ can be ignored. y_{ik} is the binary label for (x_i, p_k). $\mathcal{D}_B = \{((x_i, p_k), y_{ik}) | 1 \leq i \leq N, 1 \leq k \leq K\}$ forms the training set for the binary classification problem. H is learned by minimizing the following loss function:

$$L(H, \mathcal{D}_B) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \frac{1}{K}[\text{sign}(H(x_i, p_k)) \neq y_{ik}]. \quad (6)$$

AdaBoost.MH maintains a sequence of weight matrices $(\mathbf{W}^1, ..., \mathbf{W}^T)$ through T stages where $\mathbf{W}^t = (w_{ik}^t) \in \mathbb{R}^{N \times K}$ for $1 \leq t \leq T$ with w_{ik}^t the weight of the training example (x_i, p_k) at stage t. $\mathbf{W}^1$ is initialized to $w_{ik}^1 = \frac{1}{NK}$ for all i and k. At each stage t, AdaBoost.MH learns a weak classifier h_t and a weight α_t based on $\mathbf{W}^t$. With h_t, example weights are updated as follows:

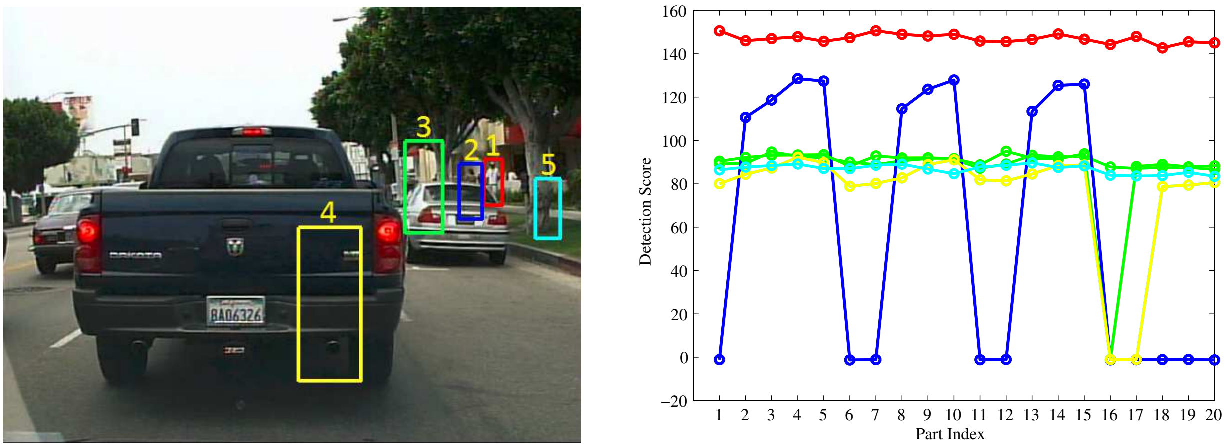
$$w_{ik}^{t+1} = \frac{w_{ik}^t exp(-\alpha_t y_{ik} h_t(x_i, p_k))}{Z_t}, \quad (7)$$

where $Z_t = \sum_{i,k} w_{ik}^t exp(-\alpha_t y_{ik} h_t(x_i, p_k))$ is a normalization factor. The training error of the learned H is bounded by

$$L(H, \mathcal{D}_F) \leq \prod_{t=1}^{T} Z_t. \quad (8)$$

Now we introduce how to minimize $L(H, \mathcal{D}_F)$ for the general case. Note that when $\frac{1}{K}$ in Eq. (6) is replaced with c_{ik}, the loss function in Eq. (6) is exactly the loss function in Eq. (2). It is easy to verify that by initializing $\mathbf{W}^1$ to $w_{ik}^1 = \frac{c_{ik}}{N}$ for all i and k, AdaBoost.MH can be used to minimize $L(H, \mathcal{D}_F)$ in Eq. (2). The upper loss bound given in Eq. (8) still holds.

Next, we describe how to learn a decision tree h_t at stage t given example weights $\mathbf{W}^t$. Starting with a root node, we construct h_t greedily. At the beginning, all training examples fall in the root node with sample weights $\mathbf{W}^t$. We examine one leaf node at a time. If a leaf node reaches a predefined maximum depth or the training examples falling in the node are pure enough, we stop branching the leaf node. Otherwise, we choose a feature and a threshold which have the minimum weighted classification error on the training examples at the leaf node. With the chosen feature and threshold, the training examples are split into two subsets each of which is assigned to one new leaf node. The two leaf nodes are the children of the current leaf node. Assume h_t has M leaf nodes and the instance space $\mathcal{X}$ is partitioned into X_1,...,X_M with X_j the set of instances falling in the



(a) (b)

Figure 4. Part scores on top 5 scoring image regions. The number above each region denotes its ranking. For the fully visible pedestrian (Red bounding box), all part detectors give high detection scores consistently (Red curve). For the partially occluded pedestrian (Blue bounding box), only the detectors of visible parts (*e.g.* P3, P4, P5, P9, P10, P14 and P15) output high detection scores (Blue curve). Background regions (Green, yellow and cyan bounding boxes) receive relatively low scores from all the part detectors (Green, yellow and cyan curves).

j-th leaf node. For an instance $x \in X_j$, h_t is defined to output

$$h_t(x, p_k) = \frac{1}{2}\ln(\frac{S^+_{jk}}{S^-_{jk}}), \tag{9}$$

where $S^+_{jk} = \sum_{x_i \in X_j} w^t_{ik}[y_{ik} = 1]$ and $S^-_{jk} = \sum_{x_i \in X_j} w^t_{ik}[y_{ik} = -1]$. After the decision tree is constructed, it can be proved that h_t defined in Eq. (9) minimizes Z_t with $\alpha_t = 1$ (See [22] for more details).

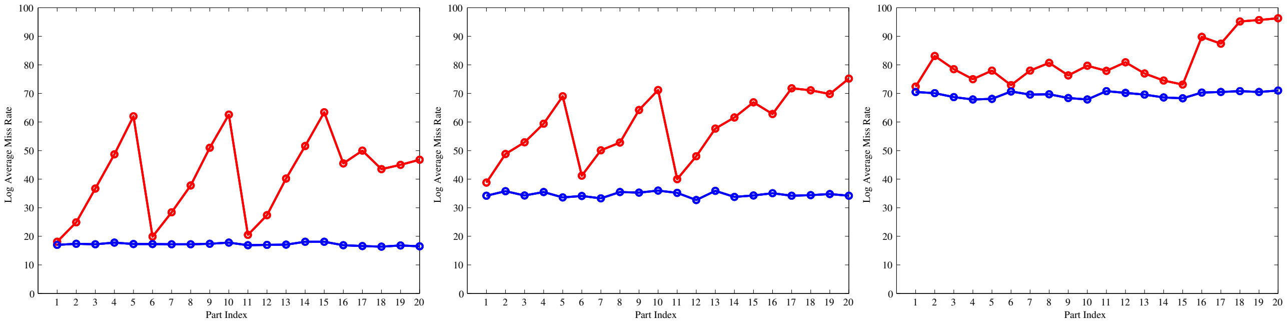
According to the above adaptation of AdaBoost.MH for minimizing $L(H, \mathcal{D}_F)$, the costs $\mathbf{C} = (c_{ik}) \in \mathbb{R}^{N \times K}$ after normalization can be considered as a distribution over $\mathcal{D}_B$. The decision trees are learned to capture the overall distribution of all the parts. Part correlations are exploited by sharing the decision trees among these parts. When taken as a pedestrian detector individually, the part detectors learned jointly show better performance than those learned separately as demonstrated in Section 5. For detection, applying the part detectors with shared decision trees is much faster as it only involves a computational cost of K instead of $K \times T$ decision trees.

4. Occlusion handling with part detectors

In a particular scene, pedestrians may be occluded by each other or other objects. Simply applying a full body detector usually does not work well when pedestrians are heavily occluded. As we do not know in advance which parts are occluded, a simple yet effective way to handle occlusions is to apply a set of part detectors. For a candidate region in an image, K part detectors would give K detection scores. We need to integrate these detection scores properly to give a final score indicating how likely the candidate region contains a pedestrian. We propose a heuristic integration method based on two observations: (1) For a partially occluded pedestrian, detectors of those parts which are inside or have large overlap with the visible region of the pedestrian would probably give high detection scores, while the other part detectors may give low detection scores due to occlusions; (2) For a non-pedestrian region, part detectors tend to output low detection scores. Figure 4 illustrates the two observations. To output a final score for a candidate image region, we choose the top S scores from the K detectors and then calculate the average of these S scores (S is set to 15 in our experiments). We do not take the maximum among the K scores as the final detection score, as some detectors may produce noisy detection scores, which would result in wrong detections. Choosing the top S scores makes the prediction more robust.

5. Experiments

We conduct two experiments using hand-crafted channel features and features learned by a convolutional neural network (CNN) respectively. We evaluate our approach on the Caltech dataset [8] which is commonly used for evaluating pedestrian detection approaches and provides both visible portion and full body annotations. Following the standard evaluation protocol, we use video sets S0-S5 for training and video sets S6-S10 for testing. A log-average miss rate is used to summarize the detection performance and is calculated by averaging miss rates at 9 false positive per-image (FPPI) points sampled evenly in the log-space ranging from 10^{-2} to 10^{0}. As the purpose of our approach is to han-



(a) *Reasonable* (b) *Partial* (c) *Heavy*

Figure 5. Results of part detectors using different part representations.

dle occlusions, we evaluate it on three subsets: *Reasonable*, *Partial* and *Heavy*. In the *Reasonable* subset, only pedestrians with at least 50 pixels tall and under no or partial occlusion are used for evaluation. This subset is widely used for evaluating pedestrian detection approaches. In the *Partial* and *Heavy* subsets, pedestrians are at least 50 pixels tall and are partially occluded (1-35 percent occluded) and heavily occluded (36-80 percent occluded) respectively.

5.1. Experiments with channel features

We choose locally decorrelated channel features (LDCF) [15] which are frequently used for pedestrian detection in recent years to represent the parts in our approach. We use the same setting as in [15]: 4 filters of size 5 × 5 are learned to locally decorrelate aggregated channel features (ACF) [6] of 10 channels to generate LDCF of 40 channels. We sample training data from video sets S0-S5 at an interval of 3 frames. Pedestrian examples which are at least 50 pixels tall and occluded not more than 70% are collected as positive examples. Five rounds of bootstrapping are adopted to train 64, 512, 1024, 2048 and 4096 decision trees respectively. The maximum depth of a decision tree is 5.

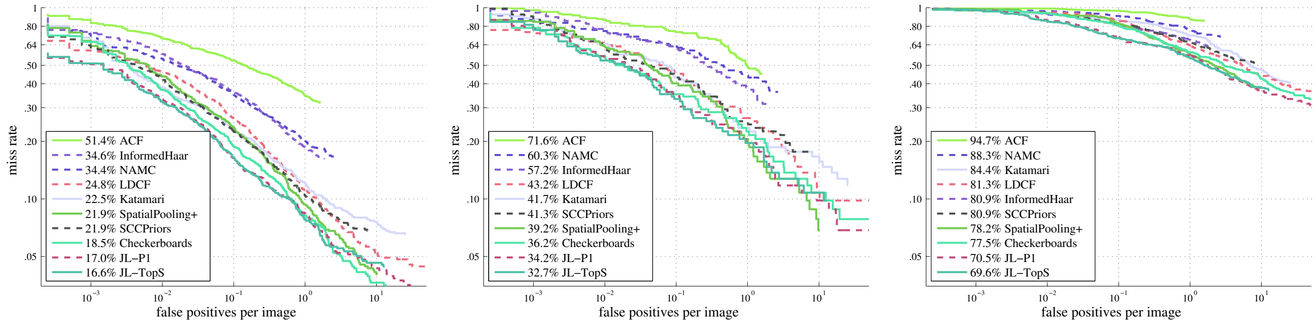
Figure 5 shows the results of part detectors learned using different part representations. PR1 denotes the representation method in which a part is represented by the features from its own image region. The part detectors using PR1 are learned independently. PR2 is the representation method in which all the parts share the features from the whole body. The part detectors using PR2 are learned jointly using our multi-label formulation. It can been seen that the performances of the part detectors learned using PR1 vary largely from part to part on the *Reasonable*, *Partial* and *Heavy* subsets. The detectors of small parts (*e.g.* P5, P10 and P20) usually perform worse than those of large parts (*e.g.* P1, P6 and P11) since with PR1, the information from the small parts is relatively limited compared with that from the large parts (See Fig. 2 for part correspondence). The part detectors with PR2 perform much better than those with PR1. The performances of different part detectors with PR2 do not change much on the three subsets. Although these part detectors show similar performances, they do behave differently. The example distribution of each part is captured by its detector. When a pedestrian example is occluded, those parts inside or have large overlap with the visible portion usually get large detection scores while the other parts tend to have low detection scores (See the blue curve in Fig. 4).

Method	*Reasonable*	*Partial*	*Heavy*
SL-P1	18.2	36.1	72.1
JL-P1	**17.0**	**34.2**	**70.5**
SL-P4	18.6	39.7	69.9
JL-P4	**17.8**	**35.5**	**67.9**
SL-P6	19.2	37.7	72.1
JL-P6	**17.3**	**34.1**	**70.7**
SL-P11	19.2	42.4	73.8
JL-P11	**16.9**	**35.2**	**70.8**
Avg. Imp.	+1.6	+4.2	+2.0

Table 1. Comparison of separate learning (SL) and joint learning (JL). P1, P4, P6 and P11 are four typical parts shown in Fig. 2. The last row shows the average improvements on the three subsets brought by joint learning.

Table 1 shows the results of the detectors of four typical parts (P1, P4, P6 and P11) learned by two different approaches, separate learning (SL) and joint learning (JL). SL learns part detectors separately by minimizing Eq. (3), while JL learns all part detectors jointly by minimizing Eq. (2). For the four parts, the detectors learned by JL perform better than their counterparts learned by SL on all the three subsets, which shows the effectiveness of sharing decision trees to exploit correlations among the parts. The average improvements on the three subsets brought by JL are 1.6%, 4.2% , and 2.0% respectively.

Table 2 shows the results of different approaches using channel features. LDCF-P1 is our implementation of LDCF [15] which only uses fully visible pedestrian examples as positive examples. SL-P1 and JL-P1 are two full-body detectors learned by SL and ML respectively. LDCF-P1 and



(a) *Reasonable*

(b) *Partial*

(c) *Heavy*

Figure 6. Comparison with state-of-the-art channel-feature based approaches.

Method	*Reasonable*	*Partial*	*Heavy*
LDCF-P1	18.1	38.8	72.4
SL-P1	18.2	36.1	72.1
JL-P1	**17.0**	**34.2**	**70.5**
JL-Max	17.5	34.8	**68.8**
JL-TopS	**16.6**	**32.7**	69.6

Table 2. Results of different approaches using channel features.

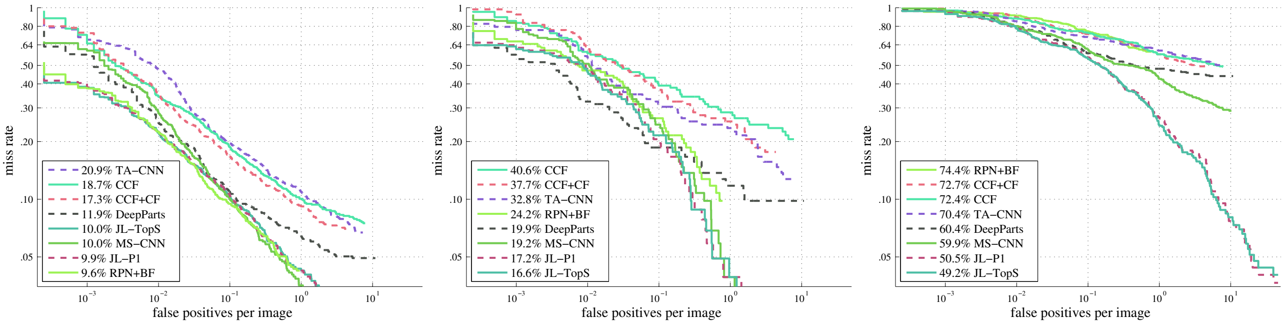
SL-P1 have similar performances on the *Reasonable* and *Heavy* subsets, but SL-P1 performs better than LDCF-P1 on the *Partial* subset since SL-P1 uses additional partially occluded pedestrian examples for training according to the definition of the misclassification cost in Eq. (1). JL-P1 outperforms SL-P1 on *Reasonable*, *Partial* and *Heavy* by 1.2%, 1.9% and 1.6% respectively. JL-Max and JL-TopS are two part detector integration approaches. JL-Max takes the maximum from 20 detection scores, while JL-TopS averages S highest detection scores. JL-TopS performs 0.8% worse than JL-Max on *Heavy* but outperforms JL-Max on *Reasonable* and *Partial* by 0.9% and 2.1% respectively. Overall, JL-TopS works more robustly. JL-TopS outperforms JL-P1 on the three subsets by 0.4%, 1.5% and 0.9% respectively, which demonstrates that the performance can be further improved by properly integrating the part detectors.

Figure 6 compares the proposed approach with state-of-the-art approaches using channel features, ACF [6], InformedHaar [34], NAMC [28], LDCF [15], Katamari [2], SpatialPooling+ [20], SCCPriors [32] and Checkerboards [35]. Our approach achieves the best performance among these channel-feature based approaches. Our full-body detector (JL-P1) already outperforms Checkerboards on all the three subsets. By properly integrating part detectors, JL-TopS outperforms Checkerboards on the three subsets by 1.9%, 3.5% and 7.9% respectively. The advantage of our approach over Checkerboards which only uses a full-body detector is more significant on the *Partial* and *Heavy* subsets, which shows the effectiveness of learning and integrating part detectors for occlusion handling. More results are provided in the **supplementary material**.

5.2. Experiments with CNN features

Recently, several approaches using CNN features have achieved the state-of-the-art performance for pedestrian detection [3, 33, 26, 4]. The proposed multi-label learning approach also applies to CNN features. We use a region proposal network (RPN) from [33] for feature extraction and then learn a set of part detectors jointly as described in Section 3.3. RPN+BF [33] also adopts a similar framework in which a set of decision trees are learned to form a full-body detector using CNN features from the RPN. The major differences between RPN+BF and our approach are two-fold: (1) our approach jointly learns the full-body detector with the other part detectors to exploit part correlations; (2) our approach further integrates the part detectors to better handle occlusions. We sample training data from video sets S0-S5 at an interval of 3 frames as in [33]. Pedestrian examples which are at least 50 pixels tall and occluded not more than 70% are collected as positive examples. These positive examples are also used for training the RPN (See [33] for the network architecture and training procedure of the PRN). To speed up training and testing, we use the RPN to generate pedestrian proposals. About 1000 proposals and 400 proposals per image are generated for training and testing respectively. Six rounds of bootstrapping are adopted to train 64, 128, 256, 512, 1024 and 2048 decision trees respectively. The maximum depth of a decision tree is 5. On an NVIDIA K5200 GPU, it takes about 0.6s (0.5s for feature extraction and 0.1s for detection) to test the jointly learned part detectors on a 480 × 640 image, while it takes about 2.2s (0.5s + 1.7s) to apply 20 separately learned detectors. Excluding the time for feature extraction, the speedup factor of the jointly learned part detectors is close to 20×.

Table 3 shows the results of different approaches using CNN features. RPN+BF-P1 is our implementation of RPN+BF [33]. SL-P1 and JL-P1 are two full-body detectors learned by separate learning (SL) and joint learning



(a) *Reasonable*

(b) *Partial*

(c) *Heavy*

Figure 7. Comparison with state-of-the-art CNN-feature based approaches.

Method	*Reasonable*	*Partial*	*Heavy*
RPN+BF-P1	10.1	18.9	58.9
SL-P1	10.3	18.0	56.6
JL-P1	**9.9**	**17.2**	**50.5**
JL-Max	10.3	17.2	**48.4**
JL-TopS	**10.0**	**16.6**	49.2

Table 3. Results of different approaches using CNN features.

(JL) respectively. SL-P1 outperforms slightly worse than RPN+BF-P1 on the *Reasonable* subset but outperforms it on the *Partial* and *Heavy* subsets. The use of some partially occluded pedestrian examples for training makes SL-P1 achieve better performance for occluded pedestrian detection. JL-P1 outperforms SL-P1 on the three subsets by 0.4% (*Reasonable*), 0.8% (*Partial*) and 6.1% (*Heavy*) respectively. The performance improvement on *Heavy* is significant. In our multi-label learning approach, the full-body detector (JL-P1) is learned jointly with the other part detectors by sharing decision trees. These decision trees are learned to capture the overall distribution of pedestrian examples including heavily occluded ones. When the full-body detector is learned independently, most heavily occluded pedestrian examples are ignored, which makes SL-P1 perform relatively poorly on *Heavy*. JL-Max and JL-TopS are two part detector integration approaches which take the maximum from 20 detection scores and the average of top S detection scores as the final detection score respectively. JL-Max has better performance on *Heavy*, while JL-TopS performs better on *Reasonable* and *Partial*. Compared with JL-P1, JL-TopS performs slightly worse (0.1%) on *Reasonable* but achieves performance gains of 0.6% and 1.3% on *Partial* and *Heavy* respectively. Since JL-P1 already works well for detecting pedestrians which are non-occluded or slightly occluded, integrating the other part detectors with the full-body detector does not help. The improvement of JL-TopS over JL-P1 on *Partial* and *Heavy* justifies that the other part detectors provide complementary information for handling occlusions.

Figure 7 compares our approach with some state-of-the-art CNN-feature based approaches, TA-CNN [27], CCF [31], CCF+CF [31], DeepParts [26], CompACT-Deep [4], MS-CNN [3] and RPN+BF [33]. On the *Reasonable* subset, JL-P1 performs comparably to the top two approaches RPN+BF and MS-CNN which also only use a single full-body detector. This is because the three approaches use similar deep convolutional neural networks (variants of VGG-16 [24]). On the *Partial* and *heavy* subsets, JL-P1 outperforms the most competitive approach MS-CNN by 2.0% and 9.4% respectively. The advantage of JL-P1 over MS-CNN on *Heavy* is significant, which shows the effectiveness of learning the full-body detector with the other part detectors. By properly integrating the jointly learned part detectors, JL-TopS further improves the performance for partially occluded pedestria detection. JL-TopS achieves the same performance on *Reasonable* as MS-CNN and outperforms MS-CNN on *Partial* and *heavy* by 2.6% and 10.7% respectively. More results are provided in the **supplementary material**.

6. Conclusions

In this paper, we propose a multi-label learning approach to learn part detectors jointly. AdaBoost.MH is adapted to learn a set of decision trees which are shared by all the part detectors. Thanks to the sharing of decision trees, part correlations are exploited and the computational cost of applying these part detectors is reduced. The learned decision trees capture the overall distribution of all parts. The effectiveness of our approach is validated on the Caltech dataset. The proposed approach is applied to channel features and CNN features and shows promising performance for detecting partially occluded pedestrians, especially the heavily occluded ones. The part detectors learned jointly by the proposed approach also perform better than their counterparts learned separately.

Acknowledgement. This work is supported in part by Singapore Ministry of Education Academic Research Fund Tier 2 MOE2015-T2-2-114 and Tier 1 RG27/14.

References

[1] R. Benenson and M. M. Seeking the strongest rigid detector. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[2] R. Benenson, M. Omran, J. Hosang, and B. Schiele. Ten years of pedestrian detection, what have we learned? In *ECCV 2014 Workshop*, 2014.

[3] Z. Cai, M. Saberian, and Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *European Conference on Computer Vision (ECCV)*, 2016.

[4] Z. Cai, M. Saberian, and N. Vasconcelos. Learning complexity-aware cascades for deep pedestrian detection. In *International Conference on Computer Vision (ICCV)*, 2015.

[5] A. Costea and S. Nedevschi. Semantic channels for fast pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] P. Dollar, R. Appel, S. Belongie, and P. Perona. Fast feature pyramids for object detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 2014.

[7] P. Dollar, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In *British Machine Vision Conference (BMVC)*, 2009.

[8] P. Dollar, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 2012.

[9] G. Duan, H. Ai, and S. Lao. A structural filter approach to human detection. In *European Conference on Computer Vision (ECCV)*, 2010.

[10] M. Enzweiler, A. Eigenstetter, B. Schiele, and D. Gavrila. Multi-cue pedestrian classification with partial occlusion handling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[11] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transaction on Pattern Analysis and Machine Intelligence (PAMI)*, 2010.

[12] J. Hosang, M. Omran, R. Benenson, and B. Schiele. Taking a deeper look at pedestrians. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[13] B. Leibe, E. Seemann, and B. Schiele. Pedestrian detection in crowded scenes. In *IEEE Conference on Computer Vision and Pattern Recogntion (CVPR)*, 2005.

[14] M. Mathias, R. Benenson, R. Timofte, and L. Van Gool. Handling occlusions with franken-classfiers. In *International Conference on Computer Vision (ICCV)*, 2013.

[15] W. Nam, P. Dollar, and J. Han. Local decorrelation for improved pedestrian detection. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

[16] W. Ouyang and X. Wang. A discriminative deep model for pedestrian detection with occlusion handling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[17] W. Ouyang and X. Wang. Joint deep learning for pedestrian detection. In *International Conference on Computer Vision (ICCV)*, 2013.

[18] W. Ouyang and X. Wang. Single-pedestrian detection aided by multi-pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[19] W. Ouyang, X. Zeng, and X. Wang. Modeling mutual visibility relationship in pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[20] S. Paisitkriangkrai, C. Shen, and A. Hengel. Strengthening the effectiveness of pedestrian detection with spatially pooled features. In *European Conference on Computer Vision (ECCV)*, 2014.

[21] B. Pepik, M. Stark, P. Gehler, and B. Schiele. Occlusion patterns for object class detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[22] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Journal of the Royal Statistical Society, Series B*, 1999.

[23] V. Shet, J. Neumann, V. Ramesh, and L. Davis. Bilattice-based logical reasoning for human detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, 2014.

[25] S. Tang, M. Andriluka, and B. Schiele. Detection and tracking of occluded people. In *British Machine Vision Conference (BMVC)*, 2012.

[26] Y. Tian, P. Luo, X. Wang, and X. Tang. Deep learning strong parts for pedestrian detection. In *International Conference on Computer Vision (ICCV)*, 2015.

[27] 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 (CVPR)*, 2015.

[28] C. Toca, M. Ciuc, and C. Patrascu. Normalized autobinomial markov channels for pedestrian detection. In *British Machine Vision Conference (BMVC)*, 2015.

[29] X. Wang, T. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *International Conference on Computer Vision (ICCV)*, 2009.

[30] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *International Conference on Computer Vision (ICCV)*, 2005.

[31] B. Yang, J. Yan, Z. Lei, and S. Li. Convolutional channel features. In *International Conference on Computer Vision (ICCV)*, 2015.

[32] Y. Yang, Z. Wang, and F. Wu. Exploring prior knowledge for pedestrian detection. In *British Machine Vision Conference (BMVC)*, 2015.

[33] L. Zhang, L. Lin, X. Liang, and K. He. Is faster r-cnn doing well for pedestrian detection? In *European Conference on Computer Vision (ECCV)*, 2016.

[34] S. Zhang, C. Bauckhage, and A. Cremers. Infromed haar-like features improve pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[35] S. Zhang, R. Benenson, and B. Schiele. Filtered channel features for pedestrian detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[36] C. Zhou and J. Yuan. Learning to integrate occlusion-specific detectors for heavily occluded pedestrian detection. In *Asian Conference on Computer Vision (ACCV)*, 2016.