IEICE TRANS. INF. & SYST., VOL.E105–D, NO.3 MARCH 2022

PAPER

Discriminative Part CNN for Pedestrian Detection

Yu WANG[†a)], **Cong CAO**[††], ***Nonmembers, and*** **Jien KATO**[†], ***Member***

SUMMARY Pedestrian detection is a significant task in computer vision. In recent years, it is widely used in applications such as intelligent surveillance systems and automated driving systems. Although it has been exhaustively studied in the last decade, the occlusion handling issue still remains unsolved. One convincing idea is to first detect human body parts, and then utilize the parts information to estimate the pedestrians' existence. Many parts-based pedestrian detection approaches have been proposed based on this idea. However, in most of these approaches, the low-quality parts mining and the clumsy part detector combination is a bottleneck that limits the detection performance. To eliminate the bottleneck, we propose Discriminative Part CNN (DP-CNN). Our approach has two main contributions: (1) We propose a high-quality body parts mining method based on both convolutional layer features and body part subclasses. The mined part clusters are not only discriminative but also representative, and can help to construct powerful pedestrian detectors. (2) We propose a novel method to combine multiple part detectors. We convert the part detectors to a middle layer of a CNN and optimize the whole detection pipeline by fine-tuning that CNN. In experiments, it shows astonishing effectiveness of optimization and robustness of occlusion handling.

key words: *pedestrian detection, occlusion handling, parts mining, parts detectors, Discriminative Part CNN*

1. Introduction

Pedestrian detection is a significant task in computer vision. In recent years, it is widely used in applications such as intelligent surveillance systems and automated driving systems. Although it has been exhaustively studied in the last decade [1]–[5], the occlusion situation still remains a very challenging issue to be dealt with.

As the samples shown in Fig. 1, the occlusion situation means that a partial or the whole pedestrian is blocked by other objects. Such a situation is closely related to potential risks in the real world. For example, for automated driving systems, the pedestrians that hide behind a wall or a parked car are the most dangerous objects in the street. When they suddenly appear from a very close distance, the brake or steering operations may be too late. Because it is hard to detect fully occluded pedestrians, the precise detection at the moment when they appears is very important, especially when only body parts become visible. In order to deal with this issue, one convincing idea is to first detect human body parts, and then utilize the parts information to estimate the pedestrians' existence.

Fig. 1 The image samples that show occlusion situation in the Caltech Pedestrian dataset [6].

Many parts-based pedestrian detection approaches have been proposed in the literature, most of them focus on part detector construction. In Bourdev et al.'s work [7], Poselet is proposed and human detection is implemented based on the configurations of human body parts. However, this approach utilizes annotations of human body parts in training, which are expensive and difficult to obtain. Tian et al. propose Deepparts in [8], which constructs part clusters simply using the relative position within the pedestrian bounding box, and trains one weak detector for each part cluster. Although it does not need additional annotations for training, since the relevance within each part cluster is weak, it is necessary to consume large computational cost to train and implement a lot of such weak detectors to achieve a good performance. Similar to [7] and [8], many parts-based approaches have to make a compromise between paying expensive annotation costs for getting good part clusters and paying high computational cost for using a large number of weak detectors.

Different with these approaches, this paper proposes Discriminative Part CNN (DP-CNN) that employs a complex mining method to find discriminative part clusters without any extra annotations. With the resulting high-quality part clusters, it becomes easy to construct robust part detectors. We also notice that most existing approaches, including the works mentioned above, do not pay much attention to part detector combination. Many approaches use brutally a linear SVM that may not exert the potential of multiple part detectors. In this work, we implement part detectors as a layer in the feature extractor CNN, which makes use of their potentials and achieves astonishing optimization effect.

The proposed DP-CNN makes full use of the parts information to deal with the occlusion issue in the pedestrian

Manuscript received March 15, 2021.
Manuscript revised July 21, 2021.
Manuscript publicized December 6, 2021.

[†]The authors are with the College of Information Science and Engineering, Ritsumeikan University, Kusatsu-shi, 525–8577 Japan.
[††]The author is with the Graduate School of Informatics, Nagoya University, Nagoya-shi, 464–8601 Japan.
a) E-mail: ywang@nagoya-u.jp

DOI: 10.1587/transinf.2021EDP7057

detection task. Our approach has two main contributions: (1) We propose a body parts mining method based on both convolutional layer features and body part subclasses. The mined part clusters own discriminative and representative characteristics which help to construct powerful pedestrian detectors. By mining part clusters, expensive annotations of human body parts are no longer a pre-condition for training part detectors, therefore, utilizing parts information becomes easier and cheaper. (2) We propose to convert the part detectors to a middle layer of the CNN that used for feature extraction, and optimize the whole detection pipeline by fine-tuning that CNN. This makes utilizing parts information becomes more computational efficient, and shows astonishing effectiveness for optimization and occlusion handling.

2. Related Work

This section introduces the related works in three aspects. First, we introduce the R-CNN based approaches for pedestrian detection, which we also adapt as the basic pipeline in this work. Then, we introduce some mid-level visual element mining methods that are closely related to the one we proposed to mine pedestrian body parts. Finally, we compare our approach with the previous parts-based pedestrian detection works to show the differences.

2.1 Pedestrian Detection with CNN

Similar to many other object detection tasks, pedestrian detection is usually approached using the sliding window paradigm. It means to slide a window over all the scales and positions of an image, extract image features from each detection window, and apply a pre-trained classifier to do the pedestrian/non-pedestrian classification. Over the past decade, many research efforts have been devoted to the feature extraction phase, and a lot of feature extraction methods have been proposed. The hand-crafted feature proposals such as HOG [1] and channel features [2], [3] have made great achievements in early years.

In recent years, with the rapid development of Convolutional Neural Networks (CNNs), the performance of pedestrian detection has been improved significantly. For instance, Hosang et al. [9] show that both small and large CNN models can reach good performance by carefully exploring the design space and the critical implementation choices. Zhang et al. [10] provide a detailed analysis of state-of-the-art pedestrian detectors, and use the insights to construct an R-CNN [11] pipeline to improve the detection performance. Tian et al. [12] jointly optimize pedestrian detection with semantic tasks, such as pedestrian attribute recognition and scene category classification. Zhang et al. [13] show that using the Region Proposal Networks (RPN) in a R-CNN pipeline is more efficient than hand-crafted proposals, and Brazil et al. [14] suggest that adding an additional segmentation task can further improve the detection accuracy.

Including the researches mentioned above, most of the CNN based approaches are based on the two-phase R-CNN pipeline. The R-CNN paradigm is similar with sliding window paradigm except using a fast but coarse detector instead of searching exhaustively. In this work, we also utilize the R-CNN pipeline like many other previous works. However, we mainly focus on the occlusion handling in the second classification phase, by emphasizing on the effective usages of human body parts information.

2.2 Mid-Level Visual Element Mining

Mid-level visual element mining aims to discover clusters of representative and discriminative image patches. According to former works, it is useful for image classification tasks [15]–[17]. Recently, Li et al. [18] utilize the CNN fully-connected layer features as the input of the Apriori mining algorithm [19], and use the mined clusters to construct concept detectors which are further used to encode images. Their approach show state-of-the-art performances on both scene categorization and object classification tasks.

These researches give us inspirations on how to mine discriminative body parts for the pedestrian detection task. However, different with these prior works, our task deals with small-sized pedestrian images in which body parts are sometimes hard to distinguish. Therefore, we adopt more local-adapted CNN convolutional layer features and utilize body part subclasses in the mining process to make sure high-quality body part clusters can be discovered.

2.3 Pedestrian Detection Based on Body-Parts

Recent researches suggest that using body part detectors [7], [8], [20], [21] helps improving the pedestrian detection performance. In a relatively earlier work [7], Bourdev et al. propose the idea of Poselets, which are defined as body part clusters under different viewpoint and pose conditions. Poselets are learned from manually labelled 3D joint keypoints and are very discriminative. However, because heavily dependent on manual annotations, such an approach does not generalize well to datasets and tasks without annotations.

In a recent work, Tian et al. propose Deepparts [8], in which each body part cluster is constructed using image patches of a same spatial location within the pedestrian bounding boxes. This approach does not need additional annotations for training while achieves state-of-the-art pedestrian detection performance. However, because the relevance within each body part cluster is weak, in order to achieve a good performance, it is necessary to train and implement a large number of weak detectors.

Including above mentioned researches, most of the existing parts-based approaches [7], [8], [20], [21] mainly focus on the construction of part detectors, but hardly pay enough attention to the combination of multiple part detectors. For example, [8] and [20] both use a linear SVM to combine multiple part detection scores to a pedestrian score. By contrast, we believe the detection performance could be

further improved with a more ingenious combination strategy.

The proposed DP-CNN is different from prior works in two aspects: (1) It mines high-quality body part clusters without any additional annotations. This makes robust part detectors can be trained in an easier and faster way. (2) It implements multiple part detectors as a CNN middle layer. This makes fine-turning of the whole pipeline, from feature extraction to part detector combination, become possible thus lead to further global optimization.

3. Basic Idea

In this study, we focus on two key phases of the parts-based pedestrian detection approach: part detector construction and part detector combination.

As discussed before, the mining quality is very important for part detector construction. In previous works, high-quality mining methods usually rely on rich annotations, which require exhaustive manual efforts and do not generalize well. By contrast, low-quality mining methods do not require additional annotations but usually lead to part clusters of low internal relevance and weak part detectors. In this work, we propose a novel mining method which discovers discriminative part clusters without additional annotations.

On the other hand, in this work, we pay more attention to the part detector combination phase. We reshape the learned part detectors, which are actually linear classifiers, to 2D filters and plug them back to the CNN as a middle layer. The resulting CNN then implements all part detectors through a single forward pass, and can be further optimized as a whole model in an end-to-end fashion.

The two-steps pipeline which reflects above ideas is shown in Fig. 2. Specifically, in the first step, we conduct rule mining to gather body part clusters using CNN convolutional layer features, and train Linear Discriminant Analysis (LDA) detectors for selected part clusters. In the second step, we transform LDA detectors to a CNN layer and plug it back to the CNN that is used to extract convolutional layer features, then train the renewed model to pursue further optimization.

3.1 Part Detector Construction

Overall, we mine body parts from pedestrian and background images using convolutional layer features and association rule mining. The mining process is similar with MDPM [18], which is designed for the scene classification task. Because the body parts of pedestrians are small and less distinguishable, our mining task is more challenging. We take the following two strategies to ensure the mining quality.

- Mine body parts using convolutional layer features, which not only can better represent local image patches but also are of lower dimensionality.
- Divide training images/features of body parts into subclasses according to their known size and location, and conduct parts mining within each subclass respectively.

3.2 Part Detector Combination

Most parts-based pedestrian detection methods focus on how to construct part detectors rather than how to combine them. We propose a novel approach which transforms the part detectors to a CNN middle layer and trains the resulted

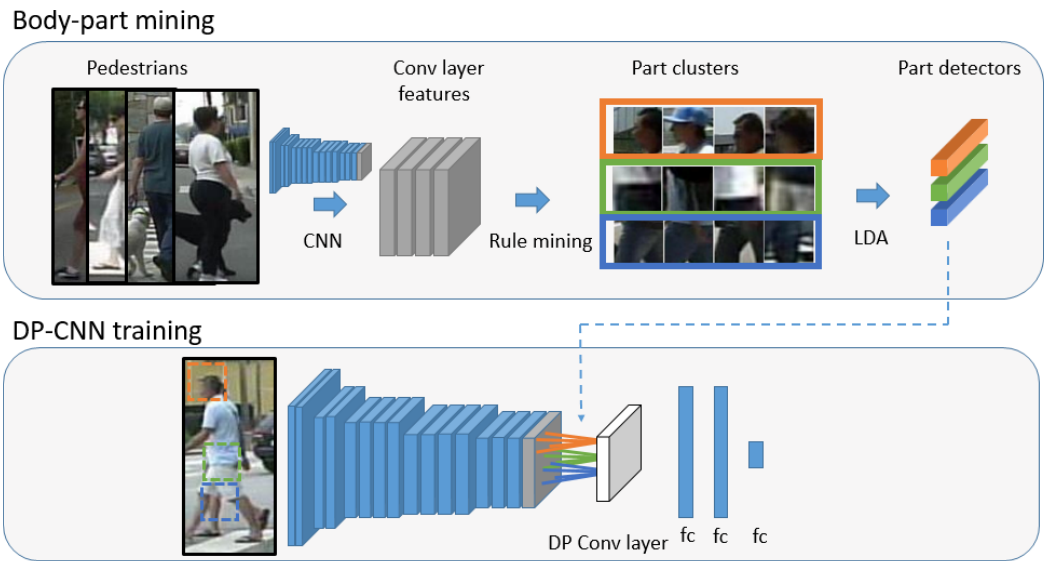


Fig. 2 The two-steps pipeline of the proposed DP-CNN. The first step mines body part clusters using convolutional layer features, and trains Linear Discriminant Analysis (LDA) detectors for selected part clusters. On the other hand, in the second step, the LDA detectors are converted to a CNN layer and plugged back to the CNN that is used to extract convolutional layer features. The resulting CNN model is then fine-turned to pursue further optimization.

CNN model end-to-end to achieve further optimization.

Thanks to the use of convolutional layer features in the mining phase, the part detectors can be converted to 2D filters and plugged back to the CNN model as a layer (after the last convolutional layer that is used for feature extraction). By adding additional fully-connected layers and retrain the whole model, not only a classifier for combining part detection results can be learnt, but also the parameters of the layers for feature extraction can be finely optimized.

4. Part Detector Construction

The implementation of part detector construction can be summarized into three steps: 1) extract CNN features from image patches; 2) mine body part clusters; and 3) train part detectors of body part clusters.

4.1 Local Feature Extraction

In recent years, many authors realize that fully-connected layer features of a pre-trained CNN can represent images much precisely than the traditional hand-crafted features [18]. However, because the pedestrian images are usually small sized (e.g. under an average height of 80 pixels in the Caltech Pedestrian dataset [6]), it is unreasonable to use the fully-connected layer features. For example, resize a small part of the pedestrian image (e.g. 32×32) to feed a pre-trained CNN model (e.g. 227×227 of AlexNet) seems to be inefficient and overqualified.

Therefore, we propose to use the last convolutional layer's activations of the whole image as local features to represent images patches. Given an image of size $w \times h \times 3$, each d-dimensional feature in the resulted $w' \times h' \times d$ activation/feature map corresponds to a local image patch. Section 6.2.1 shows that these convolutional layer features have some nice properties for mining body part clusters.

4.2 Body Parts Mining

The body parts mining is conducted mainly based on the association rule mining algorithm [22]. The algorithm is motivated by the market basket analysis and aims to discover a collection of if-then rules from the transactions. In our mining task, the rule means "if a specific collection of feature dimensions are active, then it is an image of pedestrian".

4.2.1 Association Rule Mining

First, we briefly introduce the fundamental of association rule mining [22]. Let $I = \{i_1, i_2, \ldots i_n\}$ be a set of n items. Let $D = \{t_1, t_2, \ldots t_m\}$ be a database of m transactions. Each transaction $t \in D$ contains a subset of the items in I. A rule is defined as $\{X \rightarrow Y\}$, where $X, Y \subseteq I$. The *support* value of X reflects the quantity defined as:

$$supp(X) = \frac{|\{t|t \in D, X \subseteq t\}|}{m}, \tag{1}$$

where $|\cdot|$ measures cardinality. $supp(X)$ shows how frequently the item set X appears in the database D. On the other hand, the *confidence* value of the rule $\{X \rightarrow Y\}$ reflects the quantity defined as:

$$conf(X \rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)}. \tag{2}$$

It shows how often the rule $\{X \rightarrow Y\}$ (the co-occurrence of X and Y) has been found to be true in the database.

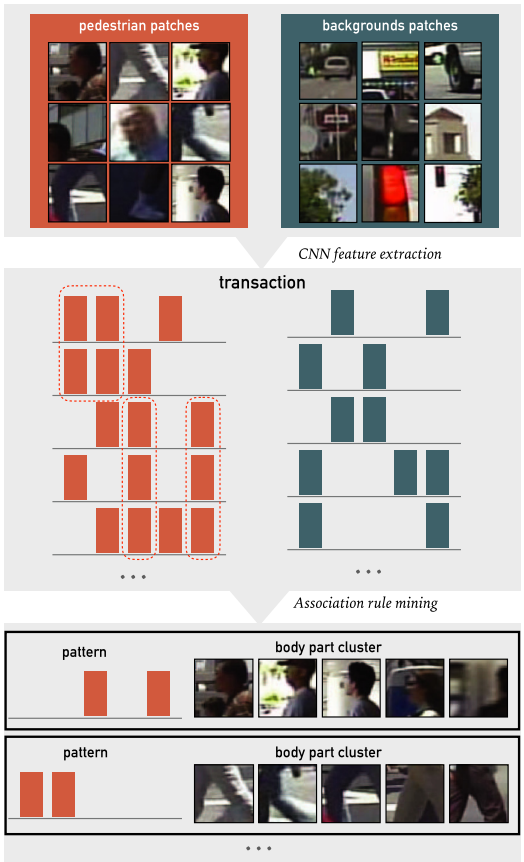


Fig. 3 Pipeline of body parts mining. Given image patches sampled from both "pedestrian" and "background", we create transactions using their CNN features. Mid-level patterns (rules) of "pedestrian" are then discovered via association rule mining. Body part clusters are then constructed by retrieving image patches that agree with related patterns.

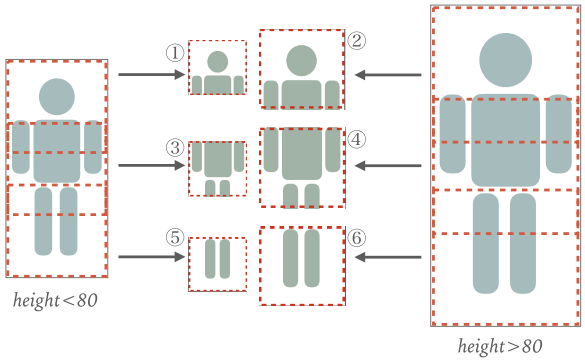
4.2.2 Pattern Mining on Pedestrians

We treat every image patch as a transaction, and its d-dimensional feature has d independent items. For each transaction, we keep n items which have the largest activations and add a pedestrian/non-pedestrian label as the $(n+1)$th item. For example, if the feature vector of a pedestrian image patch has the 1st, the 10th and the dth dimensions as its top items, then the transaction becomes $\{1, 10, d, pedestrian\}$.

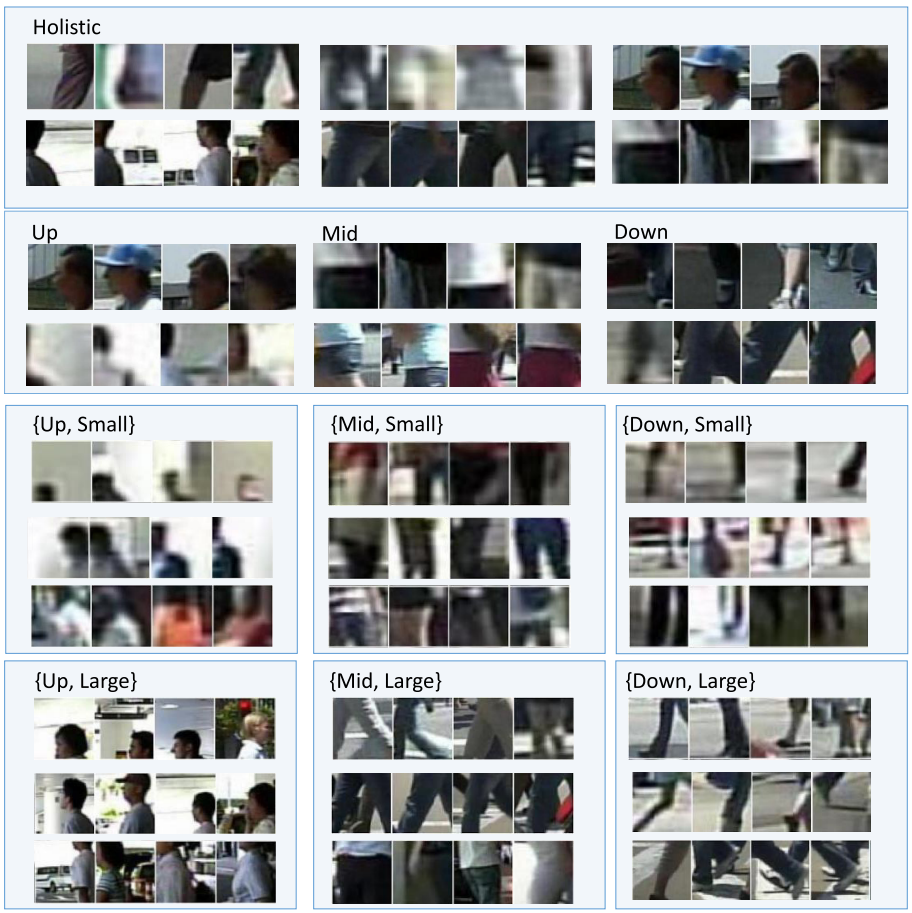
We use the Apriori algorithm [19] to find a set of rules P that satisfy the following two conditions:

$$supp(P) > supp_{min}, \tag{3}$$

$$conf(P \rightarrow pedestrian) > conf_{min}, \tag{4}$$



(a)



(b)

Fig. 4 (a) The subclasses for the "UpMidDown+Scales" setting. (b) Examples of part clusters that mined for each subclass.

where, $supp_{min}$ and $conf_{min}$ are predefined thresholds that define the minimum requirements on representativeness and discriminativeness of a rule. Each rule corresponds to a cluster of image patches which agree with that rule.

4.2.3 Mining Parts from Subclasses

In MDPM [18], the mining is conducted to discriminate the holistic collection of pedestrian image patches from the background image patches. However, because image patches of pedestrian body parts are usually small and of low resolution, different body parts (e.g. arm and leg) with similar visual appearances may be end up in a same part cluster after the mining is converged. In order to get more discriminative part clusters, we propose to divide the holistic collection of pedestrian image patches into body part sub-

classes according to the known size and location, and conduct mining for each subclass under this weak supervision. Specifically, we define two types of subclasses for mining discriminative part clusters (see Fig. 4 (a) for an illustration).

- **UpMidDown**: Divide pedestrian image patches based on their known location (within the bounding box) into three subclasses: {*Up, Mid, Down, Background*}.
- **UpMidDown+Scales**: Divide each UpMidDown subclass into small (Height < 80) and large (Height > 80) ones: {*Up-small, Up-large, Mid-small, Mid-large, Down-small, Down-large, Background*}.

Because the image patches of subclasses all belong to the pedestrian class, the discriminitiveness between subclasses is not important for the pedestrian detection task. Therefore, in practice, we conduct mining to discriminate image patches of each subclass from the background image patches to find rules $\{X \rightarrow attribute\}$, where *attribute* can be replaced by the subclasses that defined above. The *support* then becomes:

$$supp(X) = \frac{|\{t|t \in D', X \subseteq t\}|}{|D'|}, \tag{5}$$

where the database D' is defined as:

$$D' = \{t|t \in D, attribute \subseteq t\} \cup \{t|t \in D, background \subseteq t\}. \tag{6}$$

Additionally, we do not mine rules from the background image patches because their large data quantity and huge diversity in appearance make the mining hard to converge.

4.3 Train Part Detectors

With the rules, we first retrieve the image patches to construct one part cluster for each specific rule. Then, we use the patches of the part clusters to train related detectors. In this step of training, we make use of Linear Discriminant Analysis (LDA) [23]. We also utilize the merging algorithm proposed in [18] to remove the redundancy that caused by overlaps between different part clusters.

Specifically, training of part detectors is done by a recursive process. In each iteration, we first select the largest part cluster and trains its LDA detector. Then, we run this detector on remaining image patches to find positive detections, add them to the training set, and update the LDA detector. The iteration is repeated until no more image patches can be added to the training set. The output of this merging procedure is a clean set of part clusters, each with a corresponding LDA detector.

For each subclass, detectors trained from the part clusters of top k cover rates (number of image patches) are selected for further combination. In Fig. 4 (b), we show some examples of the part clusters that mined for each subclass using the Caltech Pedestrian dataset [6]. It can be confirmed that image patches in the same cluster show visual similarity and are related to similar semantic concepts. The trend is especially obvious in the "UpMidDown+Scales" setting, which confirms the effectiveness of our mining strategy.

5. Part Detector Combination

In this section, we introduce two methods for part detector combination. One is the traditional shallow method, which encodes an image as a vector of part detection scores, and learns a linear SVM classifier to predict the pedestrian score from that vector. Another is the proposed deep method, which transforms the part detectors to a middle layer of the CNN, and fine-tune the resulted CNN to predict the pedestrian score. We use the VggNet19 [24] as an example to facilitate our discussion, and will also show experiment results of other models in Sect. 6.

5.1 The Shallow Method

Similar to [18], we encode images using part detectors. In order to reduce computational cost, not all part detectors are used. We first rank the mined rules based on their cove rate within each subclass, and then select the detectors corresponding to the rules of the top k highest cover rates. A same number of detectors are selected from each subclass. Stack them together lead to to set of $N = nk$ part detectors, where n is the number of subclasses.

We first run the N detectors at each location on the $W \times H \times 512$ feature map to get a $W' \times H' \times N$ new feature map. Then, apply max pooling five times (one covers the whole image and the other four cover every $H/4$ of the feature map) to get a $1 \times 1 \times 5N$ feature vector. Finally, we train a linear SVM to predict pedestrian scores from these $1 \times 1 \times 5N$ vectors. The pipeline is illustrated in Fig. 5 (a).

The shallow method is originally designed to follow the reasonable idea of part detector combination, and achieved good performance in our preliminary experiments. However, we also found several issues or improvable aspects of it.

One serious issue is the high memory consumption in the parts mining and SVM training phases. Both phases require to keep a large quantity of data points in the memory. In practice, when training on the Caltech Pedestrian dataset, only a small subset of the data can be loaded into the memory at a same time. This means the performance of this method is restricted by the limited quantity of training data that can be used. Another improvable aspect is the separated learning and processing of different steps within the whole pipeline. As highlighted in [25], the CNN model itself is a process of combing local image information to high-level concepts, and it can achieve better performance rather than the methods that manipulate each step separately. Additionally, the CNN training does not have much memory limitations because it is a batch-based training process. Therefore, we propose the deep method and discuss it in the next section.

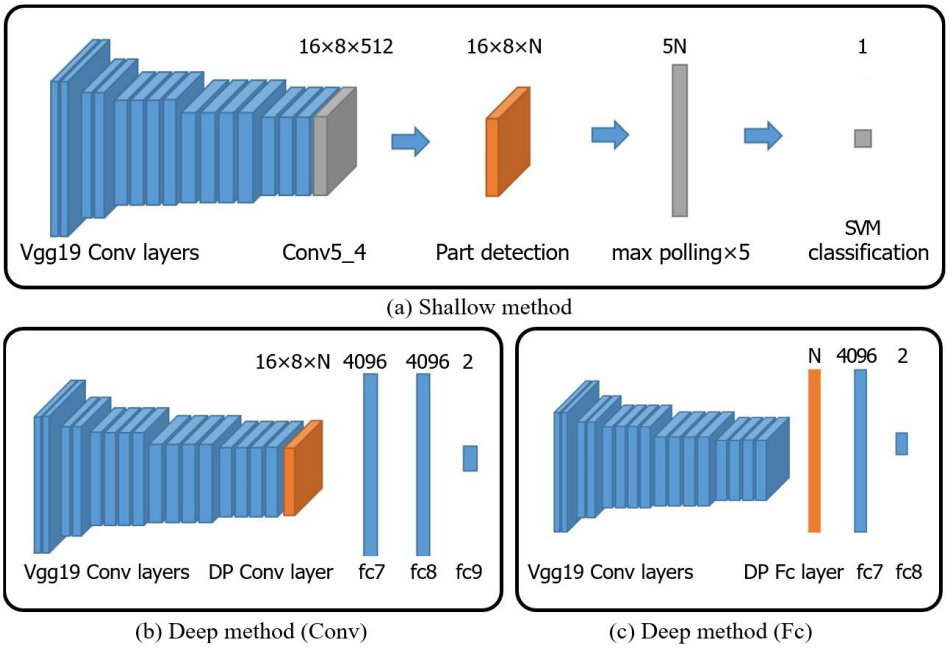


Fig. 5 Shallow and deep methods for part detector combination. (a) Shallow method: encode the image by pooling part detection scores, and predict the pedestrian score using a linear SVM. (b) Deep method (Conv): Transform part detectors to a convolutional layer and retrain the whole CNN model. (c) Deep method (Fc): Transform part detectors to a fully-connected layer and retrain the whole CNN model.

5.2 The Deep Method

As shown in Fig. 5 (a), the shallow method is surprisingly similar to the processing pipeline in a CNN model. Since we use the convolutional layer feature to construct part detectors, processing steps such as feature extraction, part detection, part detector combination and classification can be seamlessly merged to a CNN and implemented through a single forward pass. In Fig. 5 (b) and (c), we show two ways for such kind of manipulation: one transforms part detectors to a convolutional layer; the other transforms part detectors to a fully-connected layer. The merged model is a standard CNN, which can be fine-tuned using the popular mini-batch stochastic gradient descent algorithm to achieve further optimization in an end-to-end fashion.

5.2.1 Convert Part Detectors to a Convolutional Layer

- **Input size:** We fix the input size to 256×128, which adapts the average aspect ratio of pedestrian images in the Caltech Pedestrian dataset as well as the input size of the base model VggNet19 [24].
- **Convolutional layer:** As show in Fig. 5 (b), we keep all convolutional layers of the VggNet19 and insert a DP (Discriminative Part) convolutional layer on top of them. The DP convolutional layer is of size $1 \times 1 \times 512 \times N$, and is converted from the N 512-dimensional part detectors according to the following rule:

$$convW(1, 1, :, i) = L_i, \tag{7}$$

where, $convW(1, 1, :, i)$ denotes the weight of the ith filter in the DP convolutional layer and L_i means the 512 dimensional weight of the ith part detector.

- **Fully-connected layers:** We add two normal fully-connected layers of 4096 filters, one fully-connected layer of 2 filters, and one softmax layer to get the binary pedestrian classification outputs.

5.2.2 Convert Part Detectors to a Fully-Connected Layer

- **Input size:** We fix the input size to 256×128 for the same reason that mentioned above.
- **Convolutional layers:** As show in Fig. 5 (c), we simply keep all of the convolutional layers of the VggNet19.
- **Fully-connected layers:** We add one DP fully-connected layer of N filters, one normal fully-connected layer of 4096 filters, one fully-connected layer of 2 filters and one softmax layer on top of the convolutional layers. These layers together help to implement the part detectors and predict the final pedestrian classification scores. Specifically, the DP fully-connected layer is initialized randomly at first. Then, some of its random weights are replaced using the weights of the part detectors. Because the part detectors that mined from the subclasses also have known location information (Up, Mid or Down),they are used to substitute the weights of the corresponding locations in the DP fully-connected layer. The DP fully-connected

layer is of size $h \times w \times 512 \times N$, and the substitution of its weights using part detectors is implemented according to the following rule:

$$fcW(h_i, w_i, :, i) = L_i, \tag{8}$$

where h and w equals to the height and width of the feature maps from the previous convolutional layer. $fcW(h_i, w_i, :, i)$ is a $1 \times 1 \times 512$ vector at the location (h_i, w_i) of the ith filter in the DP fully-connected layer. $fcW(h_i, w_i, :, i)$ is replaced by the 512 dimensional weight of the ith part detector. (h_i, w_i) represents the ith part detector's known location (in anther word, in which subclass the ith part detector have been constructed). In this work, $h_i \in \{h_{up}, h_{mid}, h_{down}\}$ as described in Sect. 4.2.3, and w_i is not divided into subclasses in the body parts mining phase.

6. Experiments

In this section, we evaluate different aspects of the proposed approach through experiments. We first conduct ablation studies on the part detector construction method and the part detector combination method using VggNet19 on the Caltech Pedestrian dataset. Then, we extend the experiments by using additional CNN models and the KITTI dataset.

6.1 Dataset and Evaluation Criterion

The experiments are mainly conducted on the Caltech Pedestrian dataset, which has approximately ten hours of 640×480 videos that were taken from an urban environment in CA, USA. In this dataset, about 350,000 pedestrian bounding boxes of 2,300 unique pedestrians are annotated. Strictly follow the evaluation protocol of [6], [26], we use sequences 00 to 05 for training and sequences 06 to 10 for testing.

In the experiments, we use the Caltech1x subset (every 30th frame) or the Caltech10x subset (every 3th frame) for different training tasks. For the body parts mining and part detector training tasks, due to the high computational cost and the large memory consumption, we use the Caltech1x subset. For the CNN retraining task, we use the much larger Caltech10x subset. In all experiments, we use ground truth pedestrian images that annotated by bounding boxes as the positive training data, and the false positive regional proposals obtained by LDCF [27] as the negative training data.

To evaluate a detection approach, we utilize LDCF [27] to get pedestrian proposals first, and then implement that approach to update the scores of raw proposals to improved ones. The detection performance is evaluated on the improved detection scores. Like many previous studies, we use log-average miss rate [6], [26] as the evaluation metric, and report results on five subsets of different difficulties, namely *Reasonable*, *Near Scales*, *Medium Scales*, *Partial Occlusion* and *Heavy Occlusion*, of the Caltech Pedestrian dataset.

To confirm some of the conclusions we get from the Caltech Pedestrian dataset, we also implement one set of experiments on the KITTI dataset [28]. KITTI is another major dataset for pedestrian detection which consists of 7,481 training images and 7,518 testing images. We only evaluate the pedestrian detection task, and ignore car and cyclist labels which are also included in the KITTI dataset. Three subsets including *Easy*, *Moderate* and *Hard* are generally used to evaluate the pedestrian detection performance under different difficulties. We evaluate pedestrian detection performance using the PASCAL criteria on the KITTI dataset.

6.2 Evaluation on Part Detector Construction

We first compare the proposed parts mining method to the traditional mining method of MDPM [18]. Our method is different from the MDPM method in two aspects. First, our method uses the CNN convolutional layer features, while the MDPM uses the CNN fully-connected layer features. Second, we propose to divide training images of body parts into subclasses and conduct mining within each subclasses, while the MDPM conduct mining only using the holistic collection of images. We evaluate these two aspects in the following experiments.

6.2.1 Evaluation on CNN Feature Representation

The objective of this experiment is to confirm if convolutional layer features are useful image representations for the pedestrian detection task. The CNN model that used for feature extraction is a VggNet19 [24] that per-trained on the ImageNet dataset. In this experiment, we compare three different representations produced by the VggNet19.

- **Fc.:** Resize the image to 224×224 (standard input size of the VggNet19), then feed it to the VggNet19 and take the the 4,096 dimensional feature of the last fully-connected layer as the representation.
- **Conv.:** Resize the image to 256×128, then feed it to the CNN model and take the feature map from the last convolutional layer. Reshape the resulted $16 \times 8 \times 512$ feature map to a 65,536 dimensional vector, and take it as the representation.
- **Conv. + Max Pooling:** Conduct max pooling on the $16 \times 8 \times 512$ convolutional layer feature map, and use the pooled feature vector as the representation.

Using these features, we train linear SVMs to classify pedestrians from backgrounds. The classification scores of the SVMs are then used to update the raw scores of detection proposals found by LDCF. This experiment is conducted on the Caltech1x subset, and the results are reported on the *Reasonable* subset. The log-average miss rate is used as the performance metric.

The results in Table 1 show that although the max pooling version of the convolutional layer features perform slightly worse than the fully-connected layer features, the performance of the raw convolutional layer features is much

Table 2 The effectiveness of part detectors that trained using three different mining methods. The final pedestrian scores are predicted using linear SVMs. MR: Log-average miss rate (%).

Detectors	Reasonable	Near Scales	Medium Scales	Partial Occlusion	Heavy Occlusion
Holistic	46.32	18.31	74.95	59.12	87.91
UpMidDown	41.16	15.41	73.87	53.05	84.46
UpMidDown+Scales	39.34	16.69	72.53	48.21	82.33

Table 1 Comparison of pedestrian detection performance using Fc., Conv. and Conv. + Max Pooling features. Results are reported on the *Reasonable* subset of Caltech Pedestrian dataset, using Log-average miss rate (%) as the metric.

Method	MR
Fc.	62.38
Conv.	41.32
Conv. + Max Pooling	65.4

better. This suggests that the convolutional layer features preserve more information that can be used to better represent pedestrians. Moreover, to represent small image patches, the convolutional layer features are superior because they do not only have small size of receptive fields, but also omit the steps of downsampling and resizing, which could easily lead to information loss and distortion.

6.2.2 Evaluation on Mining Using Subclasses

The objective of this experiment is to confirm if conducting parts mining using body part subclasses helps to construct better part detectors. In this experiment, the **UpMidDown** method and the **UpMidDown+Scales** method are compared with the **Holistic** method. The **Holistic** method is actually a reimplementation of the mining method of MDPM, and can be treated as the baseline method. Considering the fact that the number of part detectors may directly affect the final performance, we select the same number of 600 part detectors for each method. The selection is implemented by calculating the detectors' cover rate of the training images as described in Sect. 4.3. We use the shallow method to get the final detection score from multiple part detectors. In Table 2, we report the results on the *Reasonable*, *Near Scales*, *Medium Scales*, *Partial Occlusion* and *Heavy Occlusion* subsets of the Caltech Pedestrian dataset.

On the *Reasonable* subset, the results show that the part detectors trained using body part subclasses perform better than the part detectors trained using only the holistic set. In addition, by comparing the results of **UpMidDown** and **UpMidDown+Scales**, it is obvious that more detailed definition of subclasses results in better part detectors. The part detectors mined from six subclasses of **UpMidDown+Scales** get the best result of 39.34%.

The results of the *Near Scales* and the *Medium Scales* subsets also show a similar trend. The proposed methods outperform the baseline on these two subsets. The **UpMidDown+Scales** method outperforms the **UpMidDown** method on the *Medium Scales* subset. However, the results on the *Near Scales* subset show that the **UpMidDown** method outperforms the **UpMidDown+Scales** method 1.28%. It means using scales in mining is helpful for detecting small-sized/faraway pedestrians but brings negative affects for handling large-sized/near pedestrians.

The results of the *Partial Occlusion* subset and the *Heavy Occlusion* subset show the same trend comparing to the *Reasonable* subset. Both proposed methods outperform the baseline by a considerable margin. Moreover, the **UpMidDown+Scales** method outperforms **UpMidDown** method under both conditions, which indicates that a fine definition of body part subclasses is very useful for occlusion handling.

Overall, this experiment confirms that parts mining using body part subclasses is effective for training good part detectors. It does not bring additional runtime cost, but lead to improved pedestrian detection performance.

6.3 Evaluation on Part Detector Combination

In this section, we evaluate different strategies for part detector combination. We start with an exhaustive study using the VggNet19, and then extend the experiments by also using the AlexNet and the VggNet16.

6.3.1 Evaluation Using VggNet19

We evaluate a series of deep and shallow methods for combining multiple part detectors to predict the final pedestrian scores. Because most of the deep methods do not have the issue of large memory consumption, we use the Caltech10x subset to train the deep models. On the other hand, because training SVMs for the shallow methods requires to keep all data points in the memory, we use the smaller Caltech1x subset to train shallow classifiers. For all methods in this experiment, the base part detectors are mined using the **UpMidDown+Scales** setting, which have been confirmed to have the best performance in earlier experiments. The results on the *Reasonable*, *Partial Occlusion* and *Heavy Occlusion* subsets are reported. The details of the methods we compare in this experiment are described as below and in Table 3, with the quantitative results summarized in Table 4.

- **Shallow:** The best performed setting in Sect. 6.2.2. It can be considered as the MDPM [18] approach combined with our proposed mining method.
- **ConvDP:** Convert the part detectors to a convolutional layer and retrain the resulted model. The details of the model architecture can be found in Table 3.
- **FcDP:** Convert the part detectors to a fully-connected

Table 3 Architectures of deep methods with different base models. The convolutional and fully-connneted layer parameters are denoted as "Conv-[filter size]-[number of channels]", and "Fc-[number of channels]".

ConvNet Configuration					
AlexNet		VggNet16		VggNet19	
ConvDP	FcDP	ConvDP	FcDP	ConvDP	FcDP
Input: 256 × 128 RGB Image					
AlexNet Layer 1 ... AlexNet Layer 5		VggNet16 Layer 1 ... VggNet16 Layer 13		VggNet19 Layer 1 ... VggNet19 Layer 16	
Conv-1-600 Max Pooling		Conv-1-600 Max Pooling		Conv-1-600 Max Pooling	
FC-4096 FC-4096	FC-600 FC-600	FC-4096 FC-4096	FC-600 FC-600	FC-4096 FC-4096	FC-600 FC-600
Fc-2					
SoftMax					

Table 4 Comparison of shallow and deep methods for part detector combination. All methods are implemented using VggNet19 as the base model. Results are reported on the Caltech Pedestrian dataset in log-average miss rate (%).

Method	Reasonable	Partial Occl.	Heavy Occl.
Shallow	39.34	48.21	82.33
Shallow+FT	29.74	39.10	68.71
FT	18.10	28.48	65.46
ConvDP	17.47	30.73	66.58
FcDP	17.14	28.12	62.33
ConvDP+FT	16.63	28.48	61.38
FcDP+FT	16.65	27.59	64.92

layer and retrain the resulted model. The details of the model architecture can be found in Table 3.

- **FT:** Fine-tune the VggNet19 on the Caltech10x subset for binary classification, and use the resulted model to directly predict pedestrian scores. The model is tweaked to accept images of size 256×128, which correspond to the aspect ratio of pedestrian images.
- **Shallow+FT:** The approach is the same to the **Shallow** approach. It uses a fine-tuned VggNet19 from the **FT** approach, while the **Shallow** approach is implemented using the VggNet19 that pre-trained on ImageNet.
- **ConvDP+FT:** The **ConvDP** approach that implemented using the fine-tuned VggNet19 of **FT**.
- **FcDP+FT:** The **FcDP** approach that implemented using the fine-tuned VggNet19 of **FT**.

Comparing the baseline method **Shallow** to the two proposed deep methods **ConvDP** and **FcDP**, we observe a significant performance gap. Both deep methods outperform the shallow method over 20% MR on all three subsets. It is also obvious that using a fine-tuned VggNet19 to replace the pre-trained one used in these methods does not change the trend. The performance gap between the **Shallow+FT** to **ConvDP+FT** and **FcDP+FT** is narrower, but still remains over 10% MR on the *Reasonable* and *Partial Occlusion* subsets. These results indicate that the proposed deep part detector combination methods are superior than the shallow part detector combination method.

Table 5 Number of parameters (in millions).

Methods	AlexNet	VggNet16	VggNet19
Original	41	99	104
ConvDP	71	110	116
FcDP	19	54	60

Comparing between the two proposed deep methods, the **FcDP** slightly outperforms **ConvDP** on all three subsets. However, replacing the pre-trained VggNet19 by a fine-tuned one leads to a different result, in which **ConvDP+FT** outperforms **FcDP+FT** on the *Reasonable* and *Heavy Occlusion* subsets. We hypothesize that this is because of the trade off between the importance of location information and the model capacity. Comparing to **ConvDP** which converts part detectors to a convolutional layer, **FcDP** converts part detectors to a fully-connected layer and also preserves the location information of part detectors. This makes **FcDP** work better than **ConvDP**. On the other hand, as summarized in Table 5, **ConvDP** has relatively more parameters than **FcDP**. During the training process, both **ConvDP+FT** and **FcDP+FT** have been fine-turned for two times, one for the feature extraction layers and the other for the whole models. The two-stage fine-tuning makes **ConvDP+FT** can achieve a better optimization and bring out its full potential.

The results also indicate some other trends. Firstly, comparing **ConvDP** and **FcDP** to **FT**, while **FcDP** always has better performance than **FT**, **ConvDP** sometimes underperforms **FT**. This shows that the simplest deep method **FT**, which is implemented without using any explicit parts information, is already a very powerful pedestrian detection approach. Secondly, by using the fine-tuned model in **FT** as the base model of **ConvDP** and **FcDP**, we see **ConvDP+FT** and **FcDP+FT** achieve steady improvements over **FT**, **ConvDP** and **FcDP**. This confirms in advance that the

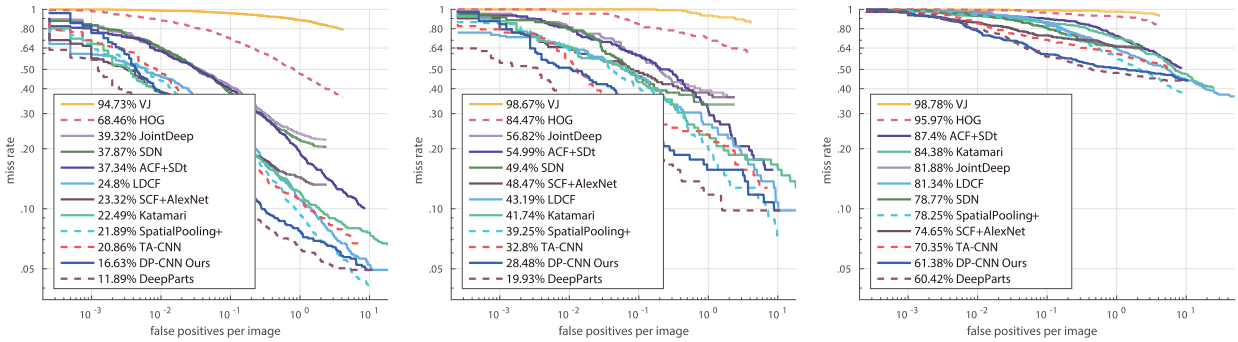


Fig. 6 Comparing between the proposed DP-CNN with related works on the Caltech Pedestrain dataset. DP-CNN outperforms most of the related methods, with notable high performance on the *Heavy Occlusion* subset.

Table 6 Comparison of deep methods implemented using different base models. Results are reported on the Caltech Pedestrian dataset in log-average miss rate (%).

Model	Reasonable	Partial Occl.	Heavy Occl.
FT(AlexNet)	26.76	45.05	79.49
ConvDP+FT(AlexNet)	26.38	40.14	73.76
FcDP+FT(AlexNet)	25.96	41.78	70.62
FT(VggNet16)	20.88	33.92	71.15
ConvDP+FT(VggNet16)	18.61	30.05	66.01
FcDP+FT(VggNet16)	19.18	30.95	70.04
FT(VggNet19)	18.10	28.48	65.46
ConvDP+FT(VggNet19)	16.63	28.48	61.38
FcDP+FT(VggNet19)	16.65	27.59	64.92

parts information is very useful and plays an important role in pedestrian detection. The best approach we confirmed in this experiment is **ConvDP**, which utilizes part information in a deep way and of sufficient model capacity.

6.3.2 Evaluation on other CNN Models

In order to confirm whether the proposed deep methods generalize well to other CNN models, we conduct additional experiments using the AlexNet [29] and the VggNet16 [24] as base models. AlexNet is a classic CNN model which consists of 5 convolutional layers and 3 fully-connected layers. It is much shallower comparing to the VggNet19. On the other hand, VggNet16 shares a same design language with VggNet19 but has three less convolutional layers.

Because the shallow methods are not comparable to deep methods as shown in previous experiments, in this experiment, we only implement three deep methods, namely **FT**, **ConvDP+FT** and **FcDP+FT**, using the additional base models. The implementation is exactly the same as described in Sect. 6.3.1. The details of network architecture and parameter quantity can be found in Table 3 and Table 5. Note again that **ConvDP+FT** and **FcDP+FT** utilize a same fine-tuned model of **FT** as the start point for the steps of body parts mining and part detector combination.

The results are shown in Table 6. Comparing the **FT** to **ConvDP+FT** and **FcDP+FT**, we observe that the proposed parts-based approaches outperform the standard fine-turning approach in all subset and base model combinations. This confirms that the superiority of our approaches does not depend on the architecture of base model and generalize well. On the other hand, the comparisons between **ConvDP+FT** and **FcDP+FT** using different base models do not always lead to a consist conclusion. Basically, **ConvDP+FT** works better when using deeper models (VggNet16 and VggNet19), while **FcDP+FT** works better when the model is relatively shallow (AlexNet). Additionally, comparing all the three deep methods with themselves using different base models, it is easy to observe that the deeper the base model is, the better the final performance can be.

6.4 Overall Evaluation

In previous experiments we confirmed that: (1) the two contributions of the proposed parts-based approach lead to superior pedestrian detection performances; (2) a deeper base model makes the two configurations of the proposed approach, namely **ConvDP+FT** and **FcDP+FT**, work better; and (3) in case of using relatively deeper base models, **ConvDP+FT** works better than **FcDP+FT**. Base on these observations, we take the **ConvDP+FT(VggNet19)** configuration to represent the proposed Discriminative Part Convolutional Neural Network (DP-CNN), and compare it with related approaches on the Caltech Pedestrian dataset, including VJ [30], HOG [1], ACF+SDT [31], Jointdeep [25], SDN [32], LDCF [27], SCF+AlexNet [9], Katamari [33], SpatialPooling+ [34], TA-CNN [12] and Deepparts [8].

The evaluation results on the Caltech Pedestrian dataset are shown in Fig. 6. Our DP-CNN achieves 16.63% MR on the *Reasonable* subset, 28.48% MR on the *Partial Occlusion* subset and 61.38% MR on the *Heavy Occlusion* subset. These results outperform most of the related approaches. We also note that the proposed approach does not outperform the DeepParts [8] approach, which explicitly utilizes the human body parts information and closely related to our approach. However, comparing to DeepParts, our DP-CNN has an obvious strength that we believe make

Table 7 A comparison between the DP-CNN and the DeepParts in terms of the number of parameters and the runtime cost. The runtime cost is evaluated for processing 100 images. The mean time of 5 runs is reported.

Methods	Number of Parameters	Runtime Cost
DP-CNN	116 (millions)	0.679 (seconds)
DeepParts	602 (millions)	97.505 (seconds)

Table 8 The AP (%) of different methods on the KITTI validation dataset. Compression of the fine-tuning method with two deep methods using VggNet19 as the base model.

Model	Easy	Moderate	Hard
FT(VggNet19)	60.79	53.04	45.42
ConvDP+FT(VggNet19)	61.89	54.11	46.11
FcDP+FT(VggNet19)	60.96	54.06	46.18

valuable contributions. The final CNN model of DP-CNN has a very simple architecture: a standard VggNet19 model with one more convolutional layer. This architecture is simple and slim. In contrast, DeepParts uses a number of 45 GoogleNets to train a lot of weak part detectors, which not only makes the training difficult, but also lead to high runtime cost. A detailed comparison between the DP-CNN and the DeepParts in terms of the number of parameters and the runtime cost is shown in Table 7. Because the DeepParts is not open-sourced, we reimplemented it based on the descriptions in [8], then calculated its number of parameters and estimated its runtime performance. We used a single Titan X GPU for computation, and confirmed that the proposed DP-CNN is about 144 times faster comparing to the DeepsParts in practice.

6.5 Evaluation on KITTI

Through the above experiments, the proposed approach have been exhaustively evaluated on the Caltech Pedestrian dataset. In this section, we conduct one more set of experiments on the KITTI dataset to verify if the conclusions from previous experiments also hold. Because of such a purpose, this experiment adopts a quite simple configuration. We followed [35] to split the KITTI TrainVal set to training set and validation set, and report the evaluation results on the validation set. The regional proposals we used are generated using the LDCF that trained on the Caltech Pedestrian dataset. We implemented three methods as in Sect. 6.3.2 using VggNet19 as the base model.

Different with the MR (the lower the better) that used in previous experiments, we adopt AP (the higher the better) as the performance metric in this experiment. As shown in Table 8, the results indicate that the trends on the KITTI dataset are consistent with these we observed in Sect. 6.3.1. On the KITTI validation dataset, the proposed parts-based approaches steadily outperforms the standard fine-turning approach on all subsets. Additionally, comparing between **ConvDP+FT** and **FcDP+FT**, it is also obvious that **ConvDP+FT** works better in general.

7. Conclusion

In this paper, we proposed the DP-CNN, which is a parts-based approach for pedestrian detection. DP-CNN is featured with: (1) a high-quality body parts mining method, which utilizes convolutional layer features as the image representation and conducts body parts mining using finely defined body part subclasses; and (2) a novel deep method for combining multiple body part detectors, which enables the whole pipeline, from feature extraction to part detector combination, to be optimized in an end-to-end fashion.

Through the experiments, we exhaustively evaluated the effectiveness of the two featured aspects of DP-CNN, and confirmed: (1) the body parts mining method helps to train high performance part detectors without parts annotations; and (2) the part detector combination method helps to achieve better global optimization of detection performance. By conducting additional experiments using more base models on more datasets, we further provided evidences which indicate the proposed approach also generalizes well. The representative configuration of DP-CNN is compared with many related works, and its high-performance and high-efficiency have also been confirmed.

In the future work, we will develop DP-CNN in the following two directions: (1) implement the core ideas of DP-CNN in single shot detection methods, such as SSD [36] and YOLO [37], to achieve further computational efficiency; (2) develop methods to integrate the parts mining process, which is done independently in its current form, to the whole detection pipeline to achieve further global optimization.

Acknowledgments

This work is supported by the JSPS Grant-in-Aid for Early Career Scientists (No. 20K19831).

References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05) - Volume 1 - Volume 01, CVPR '05, Washington, DC, USA, pp.886–893, IEEE Computer Society, 2005.

[2] P. Dollar, Z. Tu, P. Perona, and S. Belongie, "Integral channel features," Proceedings of the British Machine Vision Conference, pp.91.1–91.11, BMVA Press, 2009. doi:10.5244/C.23.91.

[3] P. Dollar, R. Appel, S. Belongie, and P. Perona, "Fast feature pyramids for object detection," IEEE Trans. Pattern Anal. Mach. Intell., vol.36, no.8, pp.1532–1545, Aug. 2014.

[4] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, "Pedestrian detection at 100 frames per second," Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), CVPR '12, Washington, DC, USA, pp.2903–2910, IEEE Computer Society, 2012.

[5] S. Zhang, C. Bauckhage, and A.B. Cremers, "Informed haar-like features improve pedestrian detection," Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR '14, Washington, DC, USA, pp.947–954, IEEE Computer Society, 2014.

[6] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," PAMI, vol.34, no.4, pp.743–761, 2012.
[7] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations," International Conference on Computer Vision (ICCV), 2009.
[8] Y. Tian, P. Luo, X. Wang, and X. Tang, "Deep learning strong parts for pedestrian detection," ICCV, 2015.
[9] J.H. Hosang, M. Omran, R. Benenson, and B. Schiele, "Taking a deeper look at pedestrians," CoRR, abs/1501.05790, 2015.
[10] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "How far are we from solving pedestrian detection?," CoRR, abs/1602.01237, 2016.
[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," Computer Vision and Pattern Recognition, 2014.
[12] Y. Tian, P. Luo, X. Wang, and X. Tang, "Pedestrian detection aided by deep learning semantic tasks," CoRR, abs/1412.0069, 2014.
[13] L. Zhang, L. Lin, X. Liang, and K. He, "Is faster r-cnn doing well for pedestrian detection?," European conference on computer vision, pp.443–457, Springer, 2016.
[14] G. Brazil, X. Yin, and X. Liu, "Illuminating pedestrians via simultaneous detection and segmentation," Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, pp.1–10, 2017.
[15] S. Singh, A. Gupta, and A.A. Efros, "Unsupervised discovery of mid-level discriminative patches," European Conference on Computer Vision, 2012.
[16] C. Doersch, A. Gupta, and A.A. Efros, "Mid-level visual element discovery as discriminative mode seeking," Advances in Neural Information Processing Systems (NIPS), pp.494–502, 2013.
[17] M. Juneja, A. Vedaldi, C.V. Jawahar, and A. Zisserman, "Blocks that shout: Distinctive parts for scene classification," IEEE Conference on Computer Vision and Pattern Recognition, 2013.
[18] Y. Li, L. Liu, C. Shen, and A. van den Hengel, "Mid-level deep pattern mining," CVPR, pp.971–980, 2015.
[19] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," Proceedings of the 20th International Conference on Very Large Data Bases, VLDB '94, San Francisco, CA, USA, pp.487–499, Morgan Kaufmann Publishers Inc., 1994.
[20] L.D. Bourdev, F. Yang, and R. Fergus, "Deep poselets for human detection," CoRR, abs/1407.0717, 2014.
[21] H. Cho, P.E. Rybski, A. Bar-Hillel, and W. Zhang, "Real-time pedestrian detection with deformable part models," IEEE Intelligent Vehicles Symposium, Aug. 2012.
[22] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," SIGMOD Rec., vol.22, no.2, pp.207–216, June 1993.
[23] B. Hariharan, J. Malik, and D. Ramanan, "Discriminative decorrelation for clustering and classification," ECCV (4), pp.459–472, 2012.
[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," CoRR, abs/1409.1556, 2014.
[25] W. Ouyang and X. Wang, "Joint deep learning for pedestrian detection," The IEEE International Conference on Computer Vision (ICCV), Dec. 2013.
[26] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: A benchmark," CVPR, June 2009.
[27] W. Nam, D. Piotr, and J.H. Han, "Local decorrelation for improved pedestrian detection," Advances in neural information processing systems, pp.424–432, 2014.
[28] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," Conference on Computer Vision and Pattern Recognition (CVPR), 2012.
[29] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems 25, ed. F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, pp.1097–1105, Curran Associates, Inc., 2012.
[30] P. Viola and M.J. Jones, "Robust real-time face detection," Int. J. Comput. Vision, vol.57, no.2, pp.137–154, May 2004.
[31] D. Park, C.L. Zitnick, D. Ramanan, and P. Dollar, "Exploring weak stabilization for motion feature extraction," 2013 IEEE Conference on Computer Vision and Pattern Recognition, pp.2882–2889, 2013.
[32] P. Luo, Y. Tian, X. Wang, and X. Tang, "Switchable deep network for pedestrian detection," The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2014.
[33] R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Ten years of pedestrian detection, what have we learned?," ECCV, CVRSUAD workshop, 2014.
[34] S. Paisitkriangkrai, C. Shen, and A. van den Hengel, "Pedestrian detection with spatially pooled features and structured ensemble learning," CoRR, abs/1409.5209, 2014.
[35] X. Chen, K. Kundu, Y. Zhu, A. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," Advances in neural information processing systems, pp.424–432, 2015.
[36] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A.C. Berg, "Ssd: Single shot multibox detector," European conference on computer vision, pp.21–37, Springer, 2016.
[37] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.7263–7271, 2017.

Yu Wang received the Ph.D. degree in Engineering from Nagoya University in 2013. He is currently an assistant professor with the College of Information Science and Engineering, Ritsumeikan University. He is a member of IEEE.

Cong Cao received the B.S. degree in IoT Engineering from SouthWest JiaoTong University in 2014. He is a Master's student with the Graduate School of Information Science, Nagoya University. He is a student member of IEEE.

Jien Kato received the M.E. and Ph.D. degrees in Information Engineering from Nagoya University in 1990 and 1993, respectively. She is a professor with the College of Information Science and Engineering, Ritsumeikan University. Her research interests include object recognition, visual event recognition and machine learning. She is a member of IEICE, IPSJ and JSAI, and also a senior member of IEEE.