Semantic Part RCNN for Real-World Pedestrian Detection

Mengmeng Xu[1] Yancheng Bai[2] Sally Sisi Qu[1] Bernard Ghanem[1]
{mengmeng.xu,sisi.qu,bernard.ghanem}@kaust.edu.sa baiyancheng20@gmail.com
[1] Visual Computing Center, King Abdullah University of Science and Technology (KAUST)
[2] Institute of Software, Chinese Academy of Sciences (CAS)

Abstract

Recent advances in pedestrian detection, a fundamental problem in computer vision, have been attained by transferring the learned features of convolutional neural networks (CNN) to pedestrians. However, existing methods often show a significant drop in performance when heavy occlusion and deformation happen because most methods rely on holistic modeling. Unlike most previous deep models that directly learn a holistic detector, we introduce the semantic part information for learning the pedestrian detector. Rather than defining semantic parts manually, we detect key points of each pedestrian proposal and then extract six semantic parts according to the predicted key points, e.g.*, head, upper-body, left/right arms and legs. Then, we crop and resize the semantic parts and pad them with the original proposal images. The padded images containing semantic part information are passed through CNN for further classification. Extensive experiments demonstrate the effectiveness of adding semantic part information, which achieves superior performance on the Caltech benchmark dataset.*

1. Introduction

Pedestrian detection is a very active research field and has attracted distinct attention in the computer vision community, since it is an essential step towards many real-world applications, including intelligent surveillance, autonomous driving and pedestrian retrieval [21, 37, 23], *etc*. Pedestrian detection has been extensively studied over the past few decades, and huge progress has been made with the emergence of deep convolutional neural networks [20, 34, 17]. In light of the dramatic success of Faster RCNN [33] in generic object detection, most proposed pedestrian detection methods follow this framework. There are two stages in the pipeline of Faster RCNN: firstly, a region proposal network (RPN) is proposed to find candidate pedestrian locations; secondly, a deep region classifier neural network (RCNN) is deployed to classify these proposals. Notably, RPN shares the same *conv* features with the classification network, thus enabling nearly cost-free region proposals. These two stages are learned end-to-end. The detection methods based on Faster RCNN [33] have recently registered further improvement in both detection performance and computational efficiency.

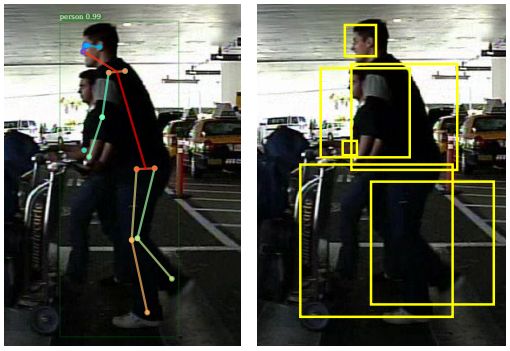

Figure 1. **An example of our constructed semantic part proposal.** We first detect N key points of each proposal from a key-point detection model. Then we extract six semantic parts according to the predicted key points, *e.g*., head, upper-body, left/right arms and legs. After that, we crop and resize the semantic parts and pad them with the original proposal images. Self-occluded key-points (*e.g*.right arm) are not shown in the image.

With the great success of the Faster RCNN based pedestrian detection methods [39, 4, 45, 43], a large group of pedestrian detectors reach convincing performance, and many of them fall under the umbrella of small scale pedestrian detection [25, 35]. It has been shown in recent years that these techniques such as using hyper-resolution or attention model can reach high recall on real word images. However, we want to draw attention to the other two fatal issues: deformation and occlusion. Here we define the deformation as the changes of a pedestrian's appearance or gesture, and the occlusion is defined by the hidden part of the human body occluded by other parts or objects. In real-world applications, *e.g*. autonomous driving, more focus should be directed on the occluded and deformed pedestrian detection problem. We must give a high recall on a person standing in front of a car or walking from a hidden corner.

Unfortunately, the deep CNN classifier in most methods [4, 45, 3] utilizes holistic representations to describe the candidate proposals, which cause their incapability to handle these two common issues in the real-world pedestrian detection. Other part-based models define human parts manually and classify each part to help the final detection. For example, DeepParts [39] constructs an extensive part pool and automatically selects the most discriminative parts to represent proposals. PCN [43] divides candidate regions into small regular grids. Those methods employ less semantic parts that degrade the performance to some extent.

To address the occlusion and deformation problem in pedestrian detection, we introduce the semantic part information in RCNN to help classify proposals. Since it is expensive and time-consuming to exhaustively label all human semantic parts (*e.g*. face, arm, leg, *et al*.) in pedestrian datasets, it is not reasonable to depend on part-level annotations to build reliable body part detection models. To unravel this problem, we use a pseudo label method generally used in the weakly supervised learning domain. We transfer the pedestrian gesture knowledge from a well-trained human pose estimation model to generate the pseudo body part prediction. Then we use the semantic part information to improve our pedestrian detector.

Figure 1 shows how we construct semantic parts in a pedestrian proposal. We first apply an off-the-shelf key point detector to find all the possible human key-points in a given image. We extract six semantic parts for each pedestrian proposal according to the predicted key points, *e.g*., head, upper-body, left/right arms and legs. After that, the semantic parts are padded to the original proposal image. The padded image contains the semantic part information, thus, making the final detector robust to occlusion and deformation. Our model does not use explicitly given human part annotations or key-point annotations that usually need to be done manually with high cost of labour and time. In our experiments, we demonstrate that adding the semantic part information transferred from other datasets(in our case, COCO) is of great use to achieve superior performance, especially on near and heavy occluded pedestrians.

Contributions.

(1) Semantic part information based on key points on a pedestrian is introduced to deal with deformation and occlusion in pedestrian detection, which carries much importance in real-world image applications .

(2) To incorporate the semantic part information, we crop and resize the semantic parts and pad them with the original images into large ones, which contain both holistic and partial information, lending more credence to the robustness of our model.

(3) The proposed pedestrian detector outperforms the state-of-the-art methods on the popular benchmark (Caltech dataset [10]) in the default setting.

2. Related Work

2.1. Object Detection

In recent years, computer vision community, without doubt, are making inroad to the fields of image classification and scene recognition through the agency of the outstanding performance of CNNs [20, 34, 47]. Generic object detectors based on CNNs, *e.g*., the Region-based CNN (RCNN) [13], Faster RCNN [33] and other variants have been introduced [13, 14, 33, 2, 1]. In [45], Liliang *et al*. find that R-CNN performs well on the pedestrian detection task, but Fast R-CNN presents a much worse result because the resolution in the last convolution layers is too low for small pedestrians. Researches on simultaneous detection and segmentation [12, 11] also show that object detection can be improved by using segmentation as a strong cue. Garrick *et al*. [3] propose a novel framework using weak box-based segmentation masks to address the issue of lacking pixel-wise segmentation annotations, instead of using a separate segmentation network.

We take SDS-RCNN [3] as the basic pedestrian detector for the convenience, which follows the Region-based CNN architecture. We first train an RPN to propose pedestrian candidates. Then we crop the original images and train an RCNN model as a binary classifier. Worth mentioning, our model can be further improved if we substitute the baseline by any off-the-shelf pedestrian detector, especially most recent ones, *e.g*., TLL-TFA [35].

2.2. Pedestrian Detection

Many works have been done to improve the pedestrian detectors by finding small scale pedestrians [4, 35]. In [4], multi-scale RPN is used to deal with scale variation and contextual information is introduced in the RCNN classifier. In [35], Song, *et al*. propose a novel method that integrates somatic topological line localization with temporal feature aggregation for detecting multi-scale pedestrians. However, those methods highly rely on the holistic representations to model the proposals, leading to an inferior performance on heavily-occluded or deformable pedestrians. Compared to these approaches, our detector harnesses the semantic part information extracted from original images to represent proposals, which is more robust to occlusion and deformation.

Pioneering work to focus on body parts to improve the pedestrian detector has shed light on the underlying potential of this method [39, 43]. Tian *et al*. [39] propose 45 complementary part detectors and manually construct a part pool. This pool covers all the scales of different body parts and the important parts are automatically chosen for occlusion handling. In [43], Wang *et al*. take manual grids as parts, and use LSTM for part communication in view of its capability of learning long-term dependencies. [25] encodes fine-grained attention masks into convolutional fea-

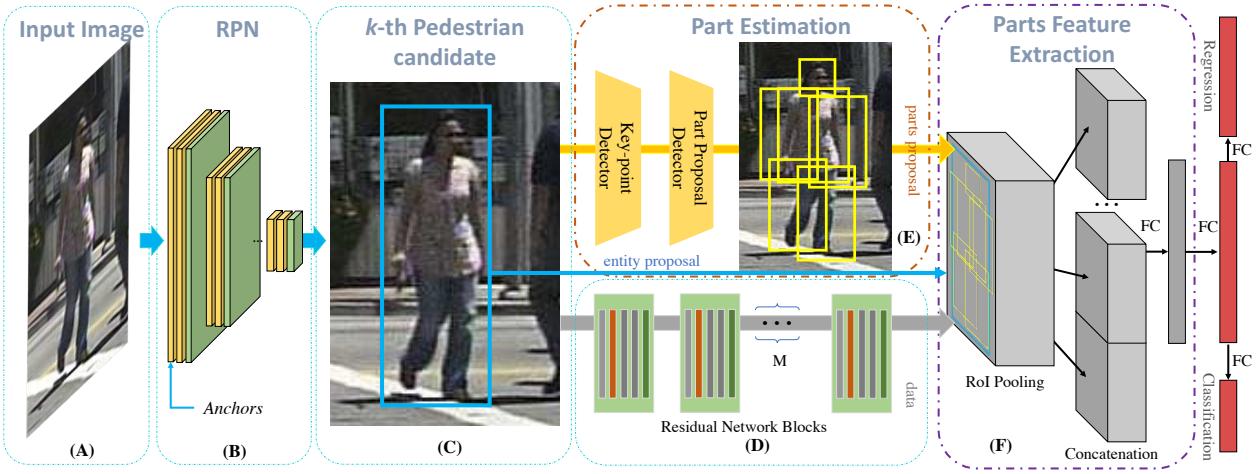

Figure 2. **The basic architecture of our Semantic Part Region based Convolutional Neural Networks (SP-RCNN).** The RPN+BF [45] method for pedestrian detection is shown by blue rounded rectangles. Given an input image, a region proposal network is used to find top K possible pedestrian candidates. Then We adapt two blocks from Region based CNN model to better classify the candidates. In **Part Estimation block** (shown as the orange dashed and rounded rectangle), we collect predictions from the key-point detector and the part proposal detector from input images to draw part bounding boxes. In **Part Feature Extraction block** (shown as the purple dashed and rounded rectangle), we use RoI Pooling on feature map from original entity proposal and part proposals (the blue rectangle and yellow rectangles on the surface of RoI Pooling cube), and concatenate the feature vectors. We also explore other methods to extract feature in Sec.3.3 and Sec.4.2. Finally, we add full connect layers to classify if a candidate pedestrian is a real person and to regress its boundary.

ture maps to focus on pedestrians. Compared to these methods, we locate the key points of pedestrian proposal and extract semantic parts. Finally, we pad parts with the original proposal regions, which represent each proposal with holistic and partial information, thus enabling the final detector to reach superior performance.

2.3. Human Pose Estimation

The goal of human pose estimation is to localize human anatomical key points (e.g., elbow, wrist, etc.) or parts (e.g., head, arm, leg *et al.*). Most traditional solutions adopt the probabilistic graphical model or the pictorial structure model [32, 44]. In contrast, deep learning based methods are currently the dominant solutions [15, 24, 38, 28, 31, 22, 27]. More specifically, there are two mainstream methods: regressing the position of key points [41, 5], and estimating heatmaps for body parts [7, 36].

Our proposed method does not train a pose estimator because of the lack of key point annotations on most general pedestrian datasets. However, our experiment shows that a well-trained model on other datasets can be generalized to this problem. Without bells and whistles, we use Mask-RCNN[17] trained on COCO[26] to transfer human part knowledge to our model.

3. Proposed method

In this section, firstly we will introduce the conceptual overview of our proposed semantic part based region convolutional neural networks (SP-RCNN) for pedestrian detection. Secondly, we will give a detailed description of each component of our proposed SP-RCNN system. Finally, the details about how to implement the proposed detector in practice will be provided.

3.1. Overview of SP-RCNN

As illustrated in Figure 2, the whole system consists of three components.

(i) The first component is the region proposal network, which is used to find the candidate pedestrian and can be of any typical pedestrian detector like PCN [43] or recent TLL-TFA [35]. In our experiments, a general RPN is adopted to find proposals from input images.

(ii) The second component is the key point detector, which locates the specific points on each proposal and then semantic parts are extracted from the proposal image. We crop semantic part images, resize and pad them with the original proposal image to form a large image.

(iii) The formed large image is passed through the final RCNN classifier for classification and regression. This image contains the semantic partial information, rendering the detector more robustness to deformation and occlusion.

3.2. Part Estimation Model

Applying part estimation can build a more robust detector for deformable objects (*e.g.*animals, human, robots). Even though such objects have variable appearances in different scenes, we can still find the minimal fixed-shape parts

Table 1. **Mapping from key-point IDs to part IDs.** To obtain each part, we find the positions of predicted key-points corresponding to the parts assigned by this table. Then we define part bounding boxes from Eq. 1

Part ID	Name	Key-Point ID
0	Head	0,1,2,3,4
1	R-Arm	6,8,10
2	L-Arm	5,7,9
3	R-Leg	12,14,16
4	L-Leg	11,13,15
5	Body	5,6,11,12

from the entity. For example, if all parts of an object are predicted as negative or absent, the object is negative. In the occluded case, a general object detection model would project the partly hidden object onto an abnormal feature map, resulting in confusion for the following classifier. If we have accurate part estimation, we can still give a powerful prediction from visible body parts.

Ideally, a part detection network can be built to locate the object part bounding boxes from each candidate proposal. However, training such a model requires a considerable number of part annotations, which are expensive and time-consuming. Alternatively, we can apply human pose estimator to reveal body parts from pedestrian images.

3.2.1 Key-Point Detection

Key-point detection is one of the direct ways to estimate human pose. COCO[26] dataset defines $K = 17$ points to describe people's gesture and MASK-RCNN[17] takes the fifth position on COCO's leaderboard in the challenge. We use this CNN model to provide human pose priors from every pedestrian candidate during all of the training, validation and test processes.

The left image in Fig. 1 is from the prediction of Mask-RCNN. These points are lined in different colours to indicate their connection and absence. In this model, each k-th point the highest score pixel location in k-th predicted hot map (*e.g.*left eye, right foot). Mask-RCNN does not use specific domain knowledge related to pedestrian, while still achieves good precision and efficiency. In fact, since most pose estimation models are trained on a large scale of dataset (*e.g.*COCO with $200k$ images and $250k$ person) and obtain high accuracy in most occasions, any state-of-the-art key-point detector can be used such as [6] and [30].

3.2.2 Semantic Parts

Based on the possible deformability of people, we slice a pedestrian into six parts: head, left and right arms, left and right legs, and main body. Head is a particular discriminative class, although it is always connected with body rigidly.

Tab. 1 shows the projection from key-points to human parts. Given a bounding box around a person as a proposal, Mask-RCNN returns visible key-points (left image in Fig. 1). We assign each part to a subset of these points, shown in Tab. 1. To get an accurate part bounding box, we draw a minimal square box covering all of its corresponding points, and constrain the centre of the box to be equal the centroid of key points inside one subset. Proposal padding to increase the size of the square boxes is applied to guarantee semantic parts inside part proposals. Eq.1 gives the formula to compute bounding box parameters, where $\mathbf{x}_k$ is the coordination of k-th key-point, r is the padding constant.

$$\mathbf{c}=\frac{1}{N}\sum_{k=1}^{N}\mathbf{x}_k,\quad a=\min_k|\mathbf{c}-\mathbf{x}_k|_2+r. \tag{1}$$

At the end, we obtain the part proposal boxes $(x_1, y_1, x_2, y_2) = (\mathbf{c}_1 - a, \mathbf{c}_2 - a, \mathbf{c}_1 + a, \mathbf{c}_2 + a)$, as shown in the right picture in Fig. 1.

If the proposal does not include the whole pedestrian, or some part of the human is invisible, we will draw a small square around the proposal edge or the only visible key-point. These failure predictions will damage the final classification, but the remained part proposals are still informative enough to build a robust model.

Furthermore, when the pedestrian candidate proposal is much larger than the size of a real human body, it gives little detriment on part prediction. Since the inaccurate human candidate can cause a high location error, we also consider the original entity proposal to provide a reference on localization regression.

3.3. Parts Feature Extraction

In Fig. 2, the Parts Feature Extraction block is a straightforward way to build semantic parts based on RCNN. We adapt the RoI pooling layer from Fast RCNN to a **part** RoI pooling layer. Given a feature map and M proposals, the feature map is cropped to M tensors according to the proposal positions, and reshaped into M feature vectors. In our case, $M = 7$, because we use features from both the six semantic parts and the original candidate pedestrian (entity). Then the M short vectors are concatenated to a lengthy one. Finally, as most CNN object detectors, we use two fully connected layers to get the prediction confidence score and regressed pedestrian location.

We also compare above methods with our solution that obtains the part regions by cropping on the raw RGB data layer. Fig. 4 shows padding semantic parts around the entity for a $448 \times 448 \times 3$ input image for Res50 backbone. We resize the entity data as a tall rectangular to try to keep the aspect ratio. This image is downsampled to $224 \times 224 \times 3$ for VGG16 backbone.

There are two advantages of our proposed solution. Firstly, part proposal size becomes larger when we get pro-

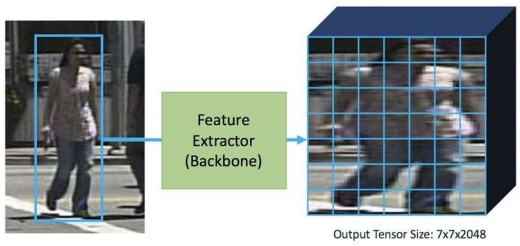

Figure 3. **Visualization of a women image in both original size (left) and in 7×7 grid (right).** Due to the rounding process, if the part region proposal (*e.g.* head) is not accurate, the cropped part feature will change a lot.

posals from raw RGB images. For example, Fig. 3 visualizes a women image patch in both original size and feature map size (7×7). Due to roundness, if the head region proposal varies from one element to two, the pooled part feature will change a lot. Secondly, cropping the part region on the raw RGB data layer is more memory-efficient. Concatenating seven part feature vectors can increase the size of the weights in the fully connected layer by seven times. In our implementation, our choice saves the GPU memory from over $8GB$ GPU to $3.79GB$. Further discussion is in Sec. 4.2 .

3.4. Implementation Details

We use ResNet-50[18] (Res50) as the backbone network for our SP-RCNN detector. The weights of the filters of newly-added layers are initialized by randomly drawing from a zero-mean Gaussian distribution with standard deviation 0.01. Biases are initialized at 0. All other layers are initialized using a model pre-trained on imagenet [9]. The mini-batch size of 16 is employed for SP-RCNN. The learning rate is initially set to 0.0005 and then reduced by a factor of 10 after every $40k$ mini-batches. Training is terminated after a maximum of $120k$ iterations. We also use a momentum of 0.9 and a weight decay of 0.0005. To detect the key points, we use the Mask-RCNN with backbone *X-101-64x4d-FPN* [16] as our key point detector. Our system is implemented in Caffe [19] and its source code will be made publicly available.

4. Experiments and Analysis

In this section, we will experimentally validate our proposed method. Firstly, we conduct ablation experiments to verify the effectiveness of the semantic part information in pedestrian detection. Then we evaluate the proposed SP-RCNN detector on the public pedestrian detection benchmark, *e.g.*, Caltech dataset [10], with comparison against state-of-the-art methods.

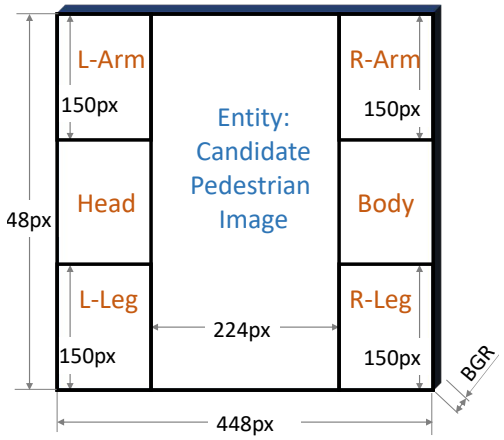

Figure 4. **2-D concatenation of the entity and the part data.** When we extract semantic parts from the proposal images, we crop, resize and pad semantic part images with the original candidate pedestrian (entity) image to form a large image.

4.1. Datasets and Evaluation Metrics

Training and Validation Datasets. To train and evaluate our SP-RCNN detector, we use the well-known large-scale pedestrian detection benchmark, the Caltech pedestrian detection dataset [10], mainly used to conduct a comprehensive analysis and ablation experiments. The entire dataset is collected from a 640×480 video recorder on a moving vehicle and divided into three subsets, including training, validation and test subsets, of which have $43k$, $1.0k$ and $4.0k$ images, respectively.

Evaluation Metrics. We use the log average miss rate (MR) to summarize detector performance. MR is the average miss rate at nine false positive per images (FPPI) rates evenly spaced in log-space in the range 10^{-2} to 10^{0} [10]. In the analysis section, the default (a.k.a. *reasonable*) setting only considers pedestrians whose size in images are larger than 50% pixels and at least 65% area of which is visible. The positive detection must have at least 0.5 intersection over union (IOU). In the heavily occluded performance test, the pedestrians visible area is constrained to $[20\%, 65\%]$. Section 4.3 gives more details on evaluation settings.

4.2. Ablation Study

In order to verify our semantic part based RCNN, we examine the impact of three main components: Part RoI Pooling, Part Feature Extraction and Part Estimator.

A. Part RoI Pooling. In this subsection, we evaluate the effect of our Part RoI Pooling layer. For each pedestrian candidate proposal, its corresponding feature vector is the concatenation of the entity and parts feature vectors. The function of part RoI pooling layer is to crop data by part proposals that are represented by a K by $4 \times 7 = 28$ matrix, the position (a four-dimension vector) of seven-part

Table 2. **Ablation experiments evaluated using the Caltech test set.** Each experiment reports the log average miss rate (MR) with one improvement disabled at a time. From this data we can conclude that (1) our proposed part ROI pooling improves the classifier in most cases; (2) Cropping on the RGB images keeps more information and leads to best performance; (3) A well-trained keypoint detector works better than designing grids as fixed body part positions.

Backbone	Part	Fused	RoI Pooling	MR
Res50	✗	✗	Res4_5_sum	18.60
Res50	✓	✗	Res4_5_sum	15.45
Res50	✗	✗	RGB	11.80
Res50	✓	✗	RGB	10.88
Res50	✗	✓	RGB	7.24
Res50	✓	✓	RGB	**7.13**
VGG16	✗	✓	RGB	7.39
VGG16	✓	✓	RGB	7.76
Res50	grid	✗	RGB	11.29
Res50	grid	✓	RGB	7.20

proposals from K pedestrian candidates. To verify the vital role of our part RoI pooling layer, we replace it by normal RoI pooling to observe the performance of our model in the absence of this layer. In this case, the part information is not used and the model acts as a generic binary object detection model as in [40]. From the first block in Tab.2, the improvement contributed by Part RoI Pooling layer is obvious. The log average miss rate falls from 18.60% to 15.45% significantly. If we crop the original RGB image directly to obtain the parts, shown as the second block of the table, we can reduce $\sim 1\%$ of MR. We also compare the benefit of part RoI pooling when cropping RGB images in the sate-of-the-art model, SDS-RCNN[3], in which authors infuse detection with weak segmentation supervision. On the Res50 backbone, using part information can improve 0.1 point(from 7.24 to 7.13) in Caltech test set under default reasonable settings. Interestingly, when we use a VGG16 backbone to replace Res50, the part proposals decrease its performance from 7.39 to 7.76 due to the low-resolution issue discussed in Sec. 3.3.

B. Feature Extraction on Different Levels. In this subsection, we compare the performance of our model applying part RoI pooling on the summation layer of Res50 in the last residual block (Res4_5_sum) or raw RGB images. Experiments substantiate that it is better to set part RoI pooling at the beginning, which means we extract and concatenate the raw RGB images. To get a feature map, we need to resize the candidate pedestrian images to 224×224 or 448×448 to fit the input size for VGG16 or Res50 backbones, respectively. Then the resized image passes through 5 to 6 2-by-2 max pooling layers and get a 7-by-7 feature map. Extracting the part features from such a low-resolution feature map is intractable. Observed from the 2nd and 4th rows in Tab. 2,

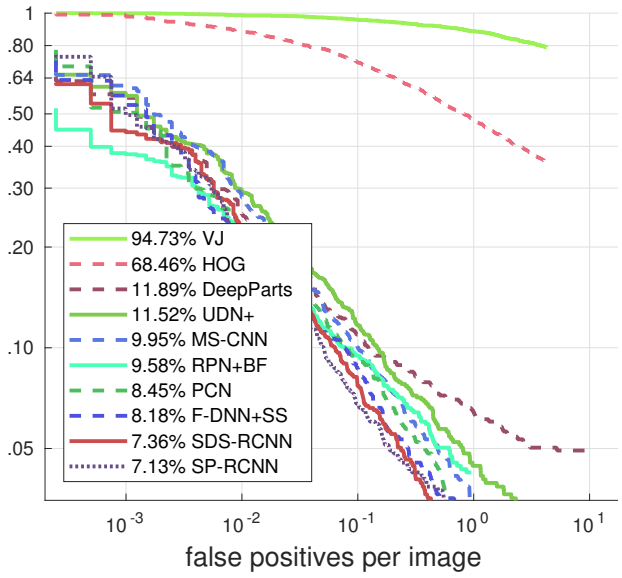

Figure 5. **The miss rate - FPPI curve on Caltech Pedestrian test set under default evaluations (legends indicate MR).** Our SP-RCNN reaches the lowest MR among all the methods. (Lower is better)

changing part RoI pooling from Res4_5_sum to raw RGB data can bring us at least 4.6 percent decrease on the average log missing rate. Without semantic part information (row 1 and row 3), aforementioned conclusion still stands, which is consistent with many works[45, 3].

C. Part Estimator. In this subsection, we evaluate the effect of Mask-RCNN key point detector with other methods. Mask-RCNN predicts 17 key-points from each pedestrian candidate. In fact, any off-the-shelf human pose estimation detector can be used such that our model owns great flexibility. To explore the importance of our pose estimator, we design a **grid** method to manually assign part bounding boxes. Similar to the part branch in PCN[43], *grid* method divides the entity proposals equally into 3×2 blocks. Then we pad the 6 blocks on the pedestrian candidate to train/test the classifier. The last blocks in Tab. 2 show that the handcrafted parts also work well. $Grid$ method can increase the performance from 11.80% MR to 11.29% MR. Fused with weak segmentation, *grid* part reaches 7.20% MR. However, to reach our best MR (7.13%), using a well-trained keypoint detector, such as Mask-RCNN, is still preferred.

4.3. Comparison with State-of-the-art Methods

We compare our results in Caltech dataset with two traditional methods [42, 8] and seven deep learning based approaches [39, 29, 4, 40, 43, 11, 3] in Tab. 3, Fig. 5 and Tab. 3 and Fig. 6.

Among those approaches, our SP-RCNN obtains a log average miss rate of 7.13% on the default setting, which beats our baseline method SDS-RCNN[3]. We achieve zero

Table 3. **Detailed breakdown performance comparisons of our models and other state-of-the-art models on several evaluation settings.** Underlined rates are the second best records (Lower is better). Our method is based on RPN+BF[40] and SDS-RCNN[3]. However, this semantic part information can also be used in other most recent works [46, 48, 25, 35] with improvement on find heavily occluded pedestrians and small pedestrians.

Methods	Default	All	Scl.large	Scl.near	Occ.heavy
VJ[42] Paul *et al*. (2001)	94.7	99.5	86.2	89.9	98.8
HOG[8] Navneet *et al*. (2005)	68.4	90.4	37.9	44.0	96.0
DeepParts[39] Yonglong *et al*. (2015)	11.8	64.8	4.37	4.78	60.4
RPN+BF[40] Cosmin *et al*. (2015)	9.57	64.7	1.18	2.26	74.4
MS-CNN[4] Zhaowei *et al*. (2016)	9.95	61.0	1.99	2.60	59.9
UDN+[29] Wanli *et al*. (2017)	11.5	64.8	1.05	2.08	70.3
F-DNN+SS[11] Xianzhi *et al*. (2017)	8.17	50.3	1.70	2.82	53.8
SDS-RCNN[3] Garrick *et al*. (2017)	7.36	61.5	0.97	2.15	58.5
PCN[43] Shiguang *et al*. (2018)	8.45	61.9	**0.00**	1.51	55.8
FRCNN+ATT-vbb[46] Shanshan *et al*. (2018)	10.33	-	-	-	45.2
PDOE+RPN[48] Chunluan *et al*. (2018)	7.60	-	-	-	44.4
GDFL[25] Chunze *et al*. (2018)	7.85	-	-	-	**43.2**
TLL-TFA [35] Tao *et al*. (2018)	7.40	**37.6**	0.72	-	-
SP-RCNN (ours)	**7.13**	60.4	**0.00**	**1.11**	53.3

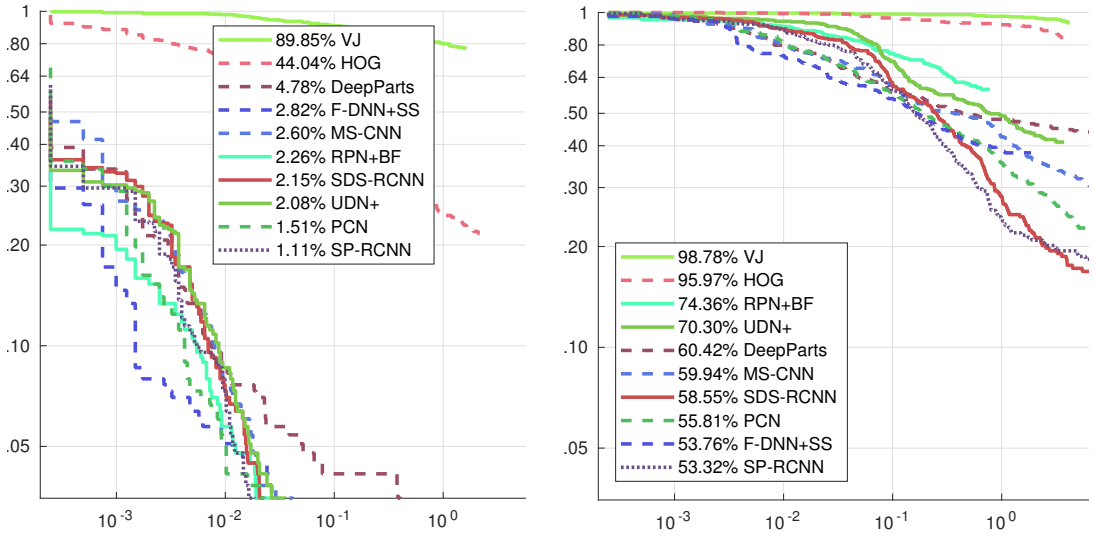

Figure 6. **The miss rate - FPPI curve on Caltech Pedestrian test set under near and heavily occluded evaluations. Legends indicate MR(log average miss rate).** *Left*: Near scale setting where we only consider the size of pedestrians larger than 80 pixels. *Right*: Heavily occluded setting with human occluded area from 20% to 65%. We keep miss rate and FPPI in the same range to show there is huge difference on the difficulty of solving these two problems. Our method denoted in blue dot curves gives less MR in both cases.

MR when people is higher than 100 pixels. That is because our method is based on human key-point estimation. It acts better when the pedestrian is close to the camera. If we consider pedestrians taller than 80 pixels, shown in Scl.near column in Fig. 6, SP-RCNN further increases 0.4 points (from 1.5% to 1.1%) than the closest PCN[43] method that uses both part and context information. In the occlusion test, our SP-RCNN still has a good record when the pedestrian is half-visible or heavily occluded, with different competitors SDS-RCNN[11] and F-DNN+SS[11].

Fig. 6 also plots the miss rate - FPPI curve on Caltech Pedestrian test set under near and heavily occluded evaluations. Our method denoted in blue dot curves gives less log average miss rate in both cases. We keep the range of miss rate and FPPI in the same scale to show there is a huge difference on the difficulty in the two problems. Comparing both plots in Fig. 6, the miss rate - FPPI curve in heavily occluded setting is still in the upper area with high MR, indicating that more research works are required to handle with the heavily occluded pedestrian detection problem.

We also list four most recent works [46, 48, 25, 35] in Tab. 3. [46, 48, 25] get high improvement on finding heav-

Figure 7. **Comparison of pedestrian detection results with other methods.** The first column shows the input images with ground-truths annotated with red rectangles. The next four columns show the detection results in green bounding boxes of PCN [43], F-DNN+SS [11], SDS-RCNN [3] and our SP-RCNN, respectively. The red number on the top of each detection bounding box is the confident score for each method (Please zoom in to see the red confidence scores). The four rows give pedestrian examples in four cases: regular, small, near-scale and heavily occluded. In each row, we only plot the detection with higher confidence than a certain threshold for better comparison. Our SP-RCNN results (last column) are comparatively more accurate in all cases.

ily occluded pedestrians and [35] gets very low MR in *all* setup by finding many small-scale pedestrians. This paper mainly shows the performance of our method based on RPN+BF[40] and SDS-RCNN[3], while it is promising to achieve much better performance if applied on later better baselines. Flexibility of our method cannot be ignored and denied in this aspect, for example, applying our semantic part information in those most recent works.

4.4. Qualitative Results

We visualize some of the detection results in Fig. 7 from the four state-of-the-art methods[40, 43, 11, 3]. As shown in the first and last images of the second row, the manually grid part method [43] cannot detect heavily occluded pedestrians. It is meaningful to find that our method is better than F-DNN+SS[11] shown in the third column at detecting heavy occluded pedestrian (one person standing behind a tree in the first row), given that this method is addressing occlusion problem specifically. Comparing the fourth column [3] with our method in the fifth column, SDS-RCNN[3] classifies a traffic light pole as a pedestrian with very high confidence (0.92, please zoom in Fig. 7 to see confidence scores over every green bounding box) while our method can avoid this false alarms by using semantic part information.

5. Conclusion

In this paper, we propose the semantic part based region convolutional neural networks (SP-RCNN) to deal with the deformation and occlusion problems in pedestrian detection. In SP-RCNN, we use the human pose estimation detector to locate key points of each pedestrian candidate. Then we extract semantic parts and pad them with the original proposal images. The padded images are resized and passed through the RCNN for both classification and localization tasks. Extensive experiments on the Caltech dataset demonstrate that adding semantic part information is of great importance and use to achieve superior performance, especially on close and heavy occluded pedestrians.

Acknowledgments This work was supported by the King Abdullah University of Science and Technology (KAUST) Office of Sponsored Research and by Natural Science Foundation of China, Grant No. 61603372.

References

[1] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem. Finding tiny faces in the wild with generative adversarial network. In *CVPR*, pages 21–30, 2018. 2

[2] Yancheng Bai, Yongqiang Zhang, Mingli Ding, and Bernard Ghanem. Sod-mtgan: Small object detection via multi-task generative adversarial network. In *ECCV*, 2018. 2

[3] Garrick Brazil, Xi Yin, and Xiaoming Liu. Illuminating pedestrians via simultaneous detection & segmentation. *CoRR*, abs/1706.08564, 2017. 2, 6, 7, 8

[4] Zhaowei Cai, Quanfu Fan, Rogerio S Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. In *European Conference on Computer Vision*, pages 354–370. Springer, 2016. 1, 2, 6, 7

[5] Joao Carreira, Pulkit Agrawal, Katerina Fragkiadaki, and Jitendra Malik. Human pose estimation with iterative error feedback. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4733–4742, 2016. 3

[6] Yilun Chen, Zhicheng Wang, Yuxiang Peng, Zhiqiang Zhang, Gang Yu, and Jian Sun. Cascaded pyramid network for multi-person pose estimation. *CoRR*, abs/1711.07319, 2017. 4

[7] Xiao Chu, Wei Yang, Wanli Ouyang, Cheng Ma, Alan L Yuille, and Xiaogang Wang. Multi-context attention for human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1831–1840, 2017. 3

[8] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005. 6, 7

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 5

[10] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *CVPR*, June 2009. 2, 5

[11] Xianzhi Du, Mostafa El-Khamy, Jungwon Lee, and Larry Davis. Fused dnn: A deep neural network fusion approach to fast and robust pedestrian detection. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 953–961. IEEE, 2017. 2, 6, 7, 8

[12] Sanja Fidler, Roozbeh Mottaghi, Alan Yuille, and Raquel Urtasun. Bottom-up segmentation for top-down detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3294–3301, 2013. 2

[13] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 2

[14] Ross B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015. 2

[15] Georgia Gkioxari, Alexander Toshev, and Navdeep Jaitly. Chained predictions using convolutional neural networks. In *European Conference on Computer Vision*, pages 728–743. Springer, 2016. 3

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017. 5

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 1, 3, 4

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 5

[19] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, 2014. 5

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2

[21] Da Li and Zhang Zhang. Large-scale pedestrian retrieval competition. *CoRR*, abs/1903.02137, 2019. 1

[22] Jianshu Li, Jian Zhao, Yunchao Wei, Congyan Lang, Yidong Li, Terence Sim, Shuicheng Yan, and Jiashi Feng. Multiple-human parsing in the wild. *arXiv preprint arXiv:1705.07206*, 2017. 3

[23] Kai Li, Zhengming Ding, Kunpeng Li, Yulun Zhang, and Yun Fu. Support neighbor loss for person re-identification. In *2018 ACM Multimedia Conference on Multimedia Conference*, pages 1492–1500. ACM, 2018. 1

[24] Ita Lifshitz, Ethan Fetaya, and Shimon Ullman. Human pose estimation using deep consensus voting. In *European Conference on Computer Vision*, pages 246–260. Springer, 2016. 3

[25] Chunze Lin, Jiwen Lu, Gang Wang, and Jie Zhou. Graininess-aware deep feature learning for pedestrian detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 732–747, 2018. 1, 2, 7

[26] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 3, 4

[27] Ting Liu, Tao Ruan, Zilong Huang, Yunchao Wei, Shikui Wei, Yao Zhao, and Thomas Huang. Devil in the details: Towards accurate single and multiple human parsing. *CoRR*, abs/1809.05996, 2018. 3

[28] Xuecheng Nie, Jiashi Feng, Yiming Zuo, and Shuicheng Yan. Human pose estimation with parsing induced learner. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2100–2108, 2018. 3

[29] Wanli Ouyang, Hui Zhou, Hongsheng Li, Quanquan Li, Junjie Yan, and Xiaogang Wang. Jointly learning deep features, deformable parts, occlusion and classification for pedestrian detection. *IEEE transactions on pattern analysis and machine intelligence*, 2017. 6, 7

[30] George Papandreou, Tyler Zhu, Nori Kanazawa, Alexander Toshev, Jonathan Tompson, Chris Bregler, and Kevin P. Murphy. Towards accurate multi-person pose estimation in the wild. *CoRR*, abs/1701.01779, 2017. 4

[31] Xi Peng, Zhiqiang Tang, Fei Yang, Rogerio S Feris, and Dimitris Metaxas. Jointly optimize data augmentation and

network training: Adversarial data augmentation in human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2226–2234, 2018. 3

[32] Leonid Pishchulin, Mykhaylo Andriluka, Peter Gehler, and Bernt Schiele. Poselet conditioned pictorial structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 588–595, 2013. 3

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 1, 2

[34] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 1, 2

[35] Tao Song, Leiyu Sun, Di Xie, Haiming Sun, and Shiliang Pu. Small-scale pedestrian detection based on topological line localization and temporal feature aggregation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 536–551, 2018. 1, 2, 3, 7, 8

[36] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. *arXiv preprint arXiv:1902.09212*, 2019. 3

[37] Yifan Sun, Liang Zheng, Weijian Deng, and Shengjin Wang. Svdnet for pedestrian retrieval. *CoRR*, abs/1703.05693, 2017. 1

[38] Wei Tang, Pei Yu, and Ying Wu. Deeply learned compositional models for human pose estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 190–206, 2018. 3

[39] Yonglong Tian, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning strong parts for pedestrian detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1904–1912, 2015. 1, 2, 6, 7

[40] Cosmin Toca, Mihai Ciuc, and Carmen Patrascu. Normalized autobinomial markov channels for pedestrian detection. In *BMVC*, pages 175–1, 2015. 6, 7, 8

[41] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1653–1660, 2014. 3

[42] Paul A. Viola and Michael J. Jones. Robust real-time face detection. *International Journal of Computer Vision*, 57:137–154, 2001. 6, 7

[43] Shiguang Wang, Jian Cheng, Haijun Liu, and Ming Tang. Pcn: Part and context information for pedestrian detection with cnns. *arXiv preprint arXiv:1804.04483*, 2018. 1, 2, 3, 6, 7, 8

[44] Yi Yang and Deva Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR 2011*, pages 1385–1392. IEEE, 2011. 3

[45] Liliang Zhang, Liang Lin, Xiaodan Liang, and Kaiming He. Is faster R-CNN doing well for pedestrian detection? *CoRR*, abs/1607.07032, 2016. 1, 2, 3, 6

[46] Shanshan Zhang, Jian Yang, and Bernt Schiele. Occluded pedestrian detection through guided attention in cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6995–7003, 2018. 7

[47] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *NIPS*, 2014. 2

[48] Chunluan Zhou and Junsong Yuan. Bi-box regression for pedestrian detection and occlusion estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 135–151, 2018. 7