SUMMARY Pedestrian detection is a critical problem in computer vision with significant impact on many real-world applications. In this paper, we introduce an fast dual-task pedestrian detector with integrated segmentation context (DTISC) which predicts pedestrian location as well as its pixel-wise segmentation. The proposed network has three branches where two main branches can independently complete their tasks while useful representations from each task are shared between two branches via the integration branch. Each branch is based on fully convolutional network and is proven effective in its own task. We optimize the detection and segmentation branch on separate ground truths. With reasonable connections, the shared features introduce additional supervision and clues into each branch. Consequently, the two branches are infused at feature spaces increasing their robustness and comprehensiveness. Extensive experiments on pedestrian detection and segmentation benchmarks demonstrate that our joint model improves the performance of detection and segmentation against state-of-the-art algorithms.

key words: pedestrian detection, feature integration, image segmentation

1. Introduction

Pedestrian detection has drawn much attention in a wide range of tasks, including autonomous driving, video surveillance, and urban monitoring [29], [30], [33]. With advanced detectors built on deep convolutional neural networks (CNNs), pedestrian detection [3], [21], [31] have achieved great progress and extensive performance has been achieved on certain benchmarks [7], [27]. However, two significant challenges still hinder the advance of pedestrian detection in industrial applications.

First, pedestrian detection suffers more from occlusion than general object detection. In real-world scenarios, pedestrians tend to gather closely and occlude each other with high overlaps. The main challenge of crowd occlusion is that it significantly increases the difficulty in pedestrian localization. When detecting closely-laied instances, CNN-based detectors may fail to locate each individual accurately since the boundaries between targets are blurred during convolution and pooling operations [21]. As shown in the right part of Fig. 1, the target pedestrian (A) is highly overlapped by another target (B). The detector is apt to get confused and generate an false positive prediction such as (F). Even worse, the primary prediction is further processed by non-maximum suppression (NMS) which tends to treat pedestrian (A) as a false prediction and suppress it. Consequently, images with heavy intra-class occlusion increase the sensitivity on the threshold of NMS to balance false positive predictions and missed detections. Another critical issue is the trade-off between speed and accuracy for real-world applications. Many frameworks for pedestrian occlusion are integrated by several subdetectors [28], [37], whose computational cost increases with the number of subdetectors. Certain detectors adopt the Faster-RCNN framework [14], [26], [38] to predict pedestrian localization and estimate the occlusion part. However, given the heavy computation of Region Propose Network (RPN), detectors with the RPN module can not exhibit satisfactory speed.

In this paper, we aim to improve the pedestrian detection performance in crowded scenes with semantic segmentation information. Intuitively, the two tasks are highly related and we expect that knowledge from either task will make the other easier. Considering both accuracy and speed, we build our pedestrian detector on one-stage framework. This novel unified network is able to jointly predict pedestrian locations and their pixel-wise segmentation via one inference, which consists of three branches, namely, a detection branch, a segmentation branch and an integration. The integration branch is used as a bridge to enrich the features between the detection and segmentation branches. The proposed bi-directional structure can jointly optimize the two tasks. During the integration process, both the two branches share high-level features as extra illumination and exhibit considerable improvement on the corresponding task. To utilize the rich information from datasets on different objectives, we optimize the network using an iterative training scheme that only requires one kind of ground truth at a time. The time cost of inference is competitive when dealing with a mission requiring two kinds of predictions, such as auto pilot. We performed extensive ablation experiments on CrowdHuman benchmark [27] and evaluated the pedestrian segmentation performance on the Pascal VOC [9] benchmark. In summary, the contributions of this work are as follows:

• We propose a dual-task integration network for simultaneously supervision on pedestrian and segmentation, with the goal of improving pedestrian detection under heavy crow occlusion. Both the tasks are processed in one shot and no extra inputs are required for inference.
• We demonstrate that pedestrian detection and segmentation tasks are complementary, and can benefit each
other through feature propagation in our bi-directional framework. A shallow integration branch is proposed to improve the feature communication process.

- We develop an iterative training scheme to train the joint model. The infused network achieves the satisfactory performance on both tasks while having little impact on inference efficiency. The speed of the proposed network increases its competitiveness when dealing with two tasks simultaneously.

2. Related Work

2.1 Object Detection

Deep CNNs have achieved great success in the domain of object detection. Recently, object detectors mainly fall into two series of frameworks. The first series is the two-stage detector such as Fast-RCNN [11] and Faster-RCNN [26]. These methods generally consist of two key modules, namely, a RPN and a sub-network for classification. RPN first generates a set of object candidates at each input image scale. The second module involves the sub-network which handles classification and bounding box regression for each per-proposal region. The two-stage detectors exhibit state-of-the-art performance on many benchmarks but are also limited in terms of speed in the region proposal process. Giving priority to inference speed, several researchers have shown interest on one-stage detectors, such as SSD [19] and YOLO [23]. These detectors remove RPN and densely make predictions at different input image scales and locations, combining the object detection and bounding box regression in a single stage. The latter series of detectors focus on speed, thus compromising on performance.

2.2 Pedestrian Detection

As an extensively studied field in object detection, pedestrian detection can inherit a lot of successful techniques form object detection frameworks [19], [24], [26]. In [26] and [34], CNN-generated features were used instead of hand-crafted ones are used as input for a boosted decision forest. In [37], a multilabel learning structure was proposed for joint part detectors to address the heavy occlusion issue, but its performance is not being satisfactory in detecting pedestrians in a slightly occluded situation. A bi-box regression approach was proposed in [38]; it consists of a full body estimation branch and visible part estimation branch, but it is insufficient in terms of cost computation. With the RPN module, the two-stage-based detectors show great potential in high-accuracy pedestrian detection. In [15], a scale-aware pedestrian detector based on Fast R-CNN was proposed to improve detection performance in natural scenes where pedestrians are on different spatial scales. [34] investigated issues involving Faster R-CNN for pedestrian detection and presented a modified network with boosted forest. To meet the demand of real-time applications, several researchers have shown interest on one-stage pedestrian detectors [19], [23], [24] which remove the RPN module to make the entire pipeline a single-stage procedure. Molchanov et al. [22] improves YOLO [23] by replacing fully-connected layers with convolution layers and fine-tuned this CNN on a specialized dataset to improve detection of small instances. Hou et al. [13] worked on multiple multispectral pedestrian detectors based on SSD [19].

2.3 Feature Integration

Integrating diverse features has been a prevalent method in pedestrian detection [1], [5], [21]. Bai et al. [1] combines
depth information with a conventional pedestrian detector based on the HOG feature. Similarly, Costea et al. [5] classified each pixel to obtain semantic segmentation channels and then detected pedestrians using multisresolution and semantic channels. Recently, Mao et al. [21] compared the effect of different kinds of features on helping pedestrian detection, highlighting that the segmentation context is highly effective and proposed HyperLearner, which adds a channel feature network on the Faster R-CNN, concatenating the detection feature with an extra segmentation feature before the RPN. However, few of researchers have worked on feature integration on the one-stage-based CNN network.

2.4 Simultaneous Detection and Segmentation

As two arresting tasks in computer vision, researchers have explored the combination of detection and segmentation. A line of research aims to explicitly improve the detection performance by utilizing segmentation features as additional supervision. Fidler et al. [10] proposed a deformable model that uses semantic segmentation masks to generate feature maps to improve detection. Inspired by this work, Du et al. [8] proposed an ensemble structure, referred to as F-DNN+SS, using an embedding network to predict the segmentation mask to filter unnecessary background proposals during post-processing. In comparison with its baseline model, this work improves the 0.47% miss rate on Caltech [7] benchmark but 8x slower for inference. Brazil et al. [3] fused the segmentation information into detection branch with shared feature maps and a weak box-based segmentation box mask was proposed to address the efficiency.

Another series of research aimed to improve the performance of both tasks. Dai et al. [6] proposed a three-stage cascaded structure to improve the performance in instance segmentation based on Faster R-CNN while paying little attention to inference speed. Li et al. [16] conducted an extension work of [6], which consisted of several sub-models to sequentially handle different regions to improve speed and accuracy. Easy and confident regions are processed by earlier sub-models, and hard regions are fed to next sub-models for further processing. Our work attempts to improve both pedestrian detection and segmentation in a unified framework, exerting additional effort on the detection task.

3. Pedestrian Detection with Integrated Segmentation Context

Our goal is to improve the pedestrian detection performance on a one-stage-based CNN network with additional segmentation contexts. Towards this end, we design a unified model with two main branches: a detection branch based on YOLOv3 [25], a segmentation branch based on DeepLabv3+ [4], and a side integration branch for joining the two branches. The integration branch aims to enhance the communication of shared features, which naturally serve as strong cues for both tasks. Given that only few of datasets have both detection and segmentation annotations, we initialize the weights of the main branches from pretrained models [4], [25]. The entire DTISC is then optimized on detection and segmentation datasets via a two-stage training strategy, extending the specific class information obtained from two different datasets. In the following sections, we will first briefly introduce the fundamental models for the main branches and how we establish the integration branch to joint the two branch making semantic features flow in bi-direction. The overall structure of our proposed model is shown in Fig. 2.

3.1 Detection Branch

Our detection branch is derived from YOLOv3 [25]. YOLOv3 is a typical efficient one-stage detector and has a satisfactory accuracy. The network takes Darknet-53 [25] as its backbone to generate low-level features and adopt sub detection module to make predictions on multi-scales. We construct our detection branch based on its main formation. Modifications are made on the detection module and the loss function to fit pedestrian detection.

Darknet-53 has five convolutional blocks, each consisting of several convolutional layers, batch normalization, Leaky ReLU and shortcut links. To maintain more details, Darknet-53 replaces the conventional pooling layer with down-sampling to scale the size of input image. Inspired by feature pyramid network (FPN) [17], the detection module takes feature maps from block3 to block5 to make predictions on three different scales, where these maps are with sizes of 1/8, 1/16, and 1/32 of the input image size. Hand-picked anchor boxes [24] generated via dimension clusters are substituted for RPN in the consideration of speed. For each anchor $i$, the detection branch aims to minimize the joint loss with three parts, which is defined as follows:

$$L = \lambda_c \sum_i L_c(c_i, g_t) + \lambda_b \sum_i L_b(b_i, g_t) + \lambda_w L_w (1)$$

The first term of Eq. (1) is the classification loss $L_c$, which is a binary cross-entropy loss between pedestrian and background. The labeling policy we adopted in this work takes a pedestrian at anchor $i$ as positive if it has an IoU with a groundtruth box over 0.5 and otherwise background. The second term is the sum of the squared error loss of coordinates for bounding box regression. The bounding is denoted as a 4-tuple $[b_x, b_y, b_w, b_h]$ at anchor $i$ where $(b_x, b_y)$ is the left corner of the bounding box, and $b_w, b_h$ are the width and height of the bounding box after scales operation and log-transformation. Specifically, we only predict the offsets with this 4-tuple to make the convergence of the network during the training process easier. The third term is the prediction score by logistic regression. After the raw prediction process, we reduce the multiple detections of the same ground truth by soft non-maximum suppression suppression (SoftNMS) [2]. Unlike the greedy NMS algorithm, Soft-NMS only decays the detection score rather than directly refuse it if two detections have a significant overlap which is more robust for crowded scenarios.
3.2 Segmentation Branch

Considering precision and effectiveness, we apply the DeepLabv3+ [4] as our segmentation branch baseline model. In DeepLabv3+, Atrous Spatial Pyramid Pooling (ASPP) is used to control grid scales rather than pooling operation since it allows easy control of resolution of feature maps and adjust filter’s field-of-view to keep more semantic information [4]. When ASPP is applied on the input feature map $x$, its corresponding output feature map $y$ can be defined as:

$$y[i] = \sum_k x[i + r \cdot k]w[k]$$

(2)

where $i$ stands for the location in $y$; $w$ stands for a convolution filter; and $r$ refers the scaling factor by which we sample the input feature maps. When $r = 1$, ASPP degenerates to a convention convolution operator.

With proven effectiveness in image segmentation, the encoder-decoder structure is adopted in our network, where the encoder module captures multi-scales feature information by down-sampling the image at different resolutions. The decoder module up-samples these rich feature information for sharp segmentation (shown in the lower branch in Fig. 2). Based on the structure, we find that the detection branch and the segmentation branches may have certain connections with each other because they share a downsampling-to-upsampling structure and similar feature maps scales in the upsampling stage. Details for the feature communication are introduced in the next section.

We maintain the main properties and replace its encoder from DeepLabv3 to Resnet-51 as a trade-off of performance and speed. Similar with detection branch, we use feature maps of 1/4, 1/8, and 1/16 of the input image as the input for shared features. The output of the segmentation branch is the pixel-wise possibilities by softmax denoted as:

$$p(x) = \frac{e^{a(x)}}{\sum_y e^{a(y)}}$$

(3)

where $x$ denotes a pixel of the image and $a(x)$ denotes the direct network output of $x$, and $p(x)$ denotes the possibility of positive prediction. Using Eq. (3), the segmentation branch is optimized by a cross-entropy loss:

$$L_s = \sum_i w(x) \log(p(x))$$

(4)

3.3 Feature Integration

To establish communication between the two branches mentioned above, we create a shallow network to jointly learn high-level information, as shown in Fig. 3. The integration branch takes the following bi-directional input: feature maps from the detection branch with sizes of 1/8, 1/16 and 1/32 of the input image, and feature maps from the segmentation branch with sizes of 1/4, 1/8, 1/16 of input image. Extra feature maps generated by the integration branch ensure that the two branches are balanced at each scale. For instance, the features at the 1/16 scale from the segmentation branch are first down-sampled to obtain the feature maps at 1/32 scale to match the detection features, and then concatenated to the detection module. Note that,
the fused feature maps are not directly used for predictions but flowed through a few convolutional layers, accompanied with batch-normalization layers and activation layers. Therefore, we optimize our unified network with the following loss function combined by Eqs. (1) and (4): \( L = L_d + \lambda L_s \). In all experiments we set \( \lambda = 0.5 \).

We find that the too many changes in the base models have a huge impact on the performance and go against of easy implementation. As a result, we only make a few of modifications on the two branches and maintain the scales of the feature maps used in the baseline models. The scale transformation in integration branch is for the following considerations. In contrast to prior work utilizing final prediction to guide other tasks, we design our network to fuse information in the feature spaces. An obvious reason is that useful feature representations can be learned for both tasks. For instance, the single detection branch cannot learn segmentation representations, which offers a clue to distinguish two close targets. Another consideration we take is the trade-off between speed and accuracy. With the increased model capacity, we do not add excessive computation burden on the model by limiting the integration branch within three convolutional layers.

4. Experiments and Results

In this section, we present the experimental results of DTISC on several challenging pedestrian detection and segmentation benchmarks. We choose CrowdHuman [27], Caltech [7] for pedestrian detection and Pascal VOC [9] and Semantic Boundaries Dataset (SBD) [12] for segmentation. Comparative experiments and ablation analysis are conducted on the CrowdHuman dataset. We refer to our basic model as DTISC, which does not include an integration branch. By contrast, the shared feature maps are plainly scale transformed. The complete model is referred to as DTISC+IB which contains an integration branch. Detailed experiment results are reported in Table 1.

4.1 Benchmark Comparison

The CrowdHuman is a recently-released pedestrian dataset that consists of 15000, 4370 and 5000 images crawled from the Internet for training, validation and testing, respectively. The training set is annotated with \(~340K\) bounding boxes and has an average of \(~22.6\) pedestrian instances for each image. Heavy-occlusion issues are emphatically pointed out, and the distribution of different occlusion levels is balanced. Each individual is annotated with three categories of bounding box, including human head, human visible-region, and human full-body region. The models are trained and evaluated on human visible-region settings. The predictions are ignored if they match no target box over a minimum IoU threshold of 0.5. The Caltech benchmark comprises approximately 2.5 hours of urban autodriving. The training set contains 42782 decomposed frames with 13674 person instances. We follow the new annotations provided by [35] and conduct evaluations on the test set of 4024 images. Following the standard evaluation metric [7] of each dataset, we adopt the log average miss rate sample against a false positive per image (FPPI) range of \([10^{-2}, 10^{0}]\) (denoted as \(MR^{-2}\)) measurement for both datasets, and the mean Average Precision (mAP) [9] for the Crowdhuman additionally.

For the segmentation task, we choose one of the most prevalent datasets, Pascacal VOC [9], for training and validation. Following the common split, we use a union of VOC2007 and VOC2012 as our training data, containing 17125 of 20 categories of which 1464 images are annotated with pixel-level segmentation labels. To fully utilize these images, the SBD [12] provides another 8498 boundary-level annotations of images from the PascalVOC dataset. Given that our task is on pedestrian detection and segmentation, we collect 3898 images containing pedestrian instances for segmentation training. To make the pedestrian segmentation subset, we further modify the boundary-level annotations from the SBD into pixel-level and regard the pedestrian instance region as positive and the others as background.

| Table 1 | Pedestrian detection results on CrowdHuman. |
|---------|---------------------------------------------|
|          | Recall | AP  | MR$^{-2}$ | FPS |
| YOLOv3 [25] | 88.94  | 76.60 | 66.83 | 29  |
| RetinaNet [18] | 90.96  | 77.19 | 65.47 | 10  |
| FPN      | 91.51  | 85.60 | 55.94 | 5   |
| DTISC    | 91.63  | 80.04 | 58.26 | 22  |
| DTISC + IB(1) | 92.05  | 81.92 | 56.97 | 22  |
| DTISC + IB(2) | 92.30  | 82.55 | 56.21 | 22  |
| DTISC + IB(3) | 92.37  | 82.71 | 55.93 | 22  |
4.2 Implementation and Results

4.2.1 Implementation Details

The entire training process can be divided into two stages. First, we optimize the detection and segmentation branches on CrowdHuman and our custom segmentation subset to obtain coarse models. To jointly optimize the whole network, we fix all the weights of the segmentation branch and continue to optimize the detection and integration branches on CrowdHuamn. When the detection branch is fully optimized, we then freeze the weights from the detection branch and similarly to finetune the segmentation branch. This iterative learning process is performed three rounds. Figure 4 shows the change of the combined loss during iterative training. Detailed results of different rounds on the pedestrian detection task is shown in Table 1.

To start training the joint model, we initialize the detection branch from original Darknet-53. Following the training configurations in [25], we set the momentum to 0.9 and the weight decay to 0.005. The learning rate is set to $10^{-3}$ for the first 70k iterations, $10^{-4}$ for next 25k iterations, and the last 15k iterations are trained with $10^{-5}$. To adapt our situation, we train the segmentation branch from scratch with 20k iterations. The initial learning rate is set to $10^{-3}$ and decreased by a factor 10 after 10k and 15k. We adopt Stochastic Gradient Descent (SGD) as the optimizer for each training procedure. After gaining the coarse models, we continue to jointly optimize the entire networks. At each iterative learning stage, we focus on one task in the branch and fix the other task branch as a feature generator through propagation. The entire training process is performed on a machine with one GTX 1080Ti GPU.

4.2.2 Pedestrian Detection Results

Table 1 shows the pedestrian detection results on the CrowdHuman benchmark. Several representative baselines are listed in the upper section. The lower section presents the results of our proposed model with and without the integration branch (IB) and those for the three joint learning rounds. To validate the effectiveness of our model, we compare the original YOLOv3 and the two other detectors adopted as baselines in CrowdHuman [27]. In comparison with the original YOLOv3, we boost the AP by 3.44% solely with a segmentation branch. The the second section in Table 1 indicates that the performance is constantly improved during each joint learning round, finally resulting in 2.67% improvement. In comparison with another one-stage detector RetinaNet [18], our model outperforms this baseline on both Recall and AP. This result confirms our assumption that segmentation context is useful for locating each pedestrian instance. Remarkable improvement can be observed in the lower part of Table 1 when introducing the integration branch. Figure 5 shows some visual comparisons of detection results and an extra segmentation context. Intuitively, our detector can well locate each individual under crowd occlusion. Our model predicts one detection result in 0.06 seconds on a 1080Ti GPU with 12 GB of memory, whereas the original YOLOv3 takes 0.035 seconds to infer one image. We also report the experiment result on the Caltech Reasonable setting on Table 2. In comparison with other state-of-the-art detectors, our model achieves 5.26% MR which surpasses most methods. In Table 3, we further validate the superiority on detecting pedestrians in heavy occlusion scenes by carrying out an experiment on the Caltech Heavy-occlusion setting.

4.2.3 Segmentation Results

Table 4 shows the Mean Squared Error (MSE) of the proposed DTISC and the comparisons of other superior methods, including our baseline model Deeplabv3+ for the segmentation branch. Our task is focused on pedestrian segmentation. Hence, we finetune part of the methods to adjust the issue. In comparison with the baseline model, we boost the MSE by 0.017. We also compare the proposed model with prevalent approaches, including FCN [20] and PSPNet [36]. Results show that our model, with and without the integration branch, performs better than against other methods.

5. Conclusion

We present a dual-task infusion framework for joint super-
For each input image, we show the detection results of the original YOLOv3, our DTISC model on the CrowdHuman dataset. In comparison with its counterparts, our model predicts targets with better location, especially for serious occlusion regions. The last column is the results of our segmentation branch.

**Table 2** Pedestrian detection results on *Reasonable* setting.

|        | DeepParts | FasterRCNN | RPN+BF | F-DNN |
|--------|-----------|------------|--------|-------|
| MR \(^2\) F-DNN+SS | 9.89      | 8.72       | 7.37   | 7.32  |
| MR \(^2\) ALPNet    | 6.61      | 6.13       | 6.04   | 5.26  |

**Table 3** Pedestrian detection results on *Heavy-occlusion* setting.

|        | DeepParts | FasterRCNN | RPN+BF | F-DNN |
|--------|-----------|------------|--------|-------|
| MR \(^2\) F-DNN+SS | 55.5      | 53.1       | 54.6   | 55.8  |
| MR \(^2\) ALPNet    | 52.0      | 51.1       | 47.2   | 44.4  |

**Table 4** Segmentation results on Pascal VOC.

|        | FCN | SegNet | PSPNet |
|--------|-----|--------|--------|
| MSE    | 0.106 | 0.074 | 0.066 |
| MSE    | 0.057 | 0.044 | 0.040 |

### Acknowledgments

This research was partially supported by National Natural Science Foundation (NSFC) of China under project No. 61906206.

### References

[1] Z. Bai, P. Wang, and Z. Li, “Stereo visual pedestrian detection based on stixel-world and feature fusion,” Chinese Journal of Scientific Instrument, vol.38, no.11, pp.2822–2829, 2017.

[2] N. Bodla, B. Singh, R. Chellappa, and L.S. Davis, “Soft-NMS—Improving object detection with one line of code,” International
Conference on Computer Vision, pp.5562–5570, 2017.

[3] G. Brazil, X. Yin, and X. Liu, “Illuminating pedestrians via simultaneous detection & segmentation,” Proc. IEEE International Conference on Computer Vision, pp.4950–4959, 2017.

[4] L.C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, “Encoder-decoder with atrous separable convolution for semantic image segmentation,” European Conference on Computer Vision, pp.833–851, 2018.

[5] A.D. Costea and S. Nedevschi, “Semantic channels for fast pedestrian detection,” Computer Vision & Pattern Recognition, 2016.

[6] J. Dai, K. He, and J. Sun, “Instance-aware semantic segmentation via multi-task network cascades,” Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.3150–3158, 2016.

[7] P. Dollár, C. Wojek, B. Schiele, and P. Perona, “Pedestrian detection: A benchmark,” CVPR, June 2009.

[8] X. Du, M. El-Khamy, J. Lee, and D. Davis, “Fused dnn: A deep neural network fusion approach to fast and robust pedestrian detection,” 2017 IEEE winter conference on applications of computer vision (WACV), pp.953–961, IEEE, 2017.

[9] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” International Journal of Computer Vision, vol.88, no.2, pp.303–338, June 2010.

[10] S. Fidler, R. Mottaghi, A. Yuille, and R. Urtasun, “Bottom-up segmentation for top-down detection,” Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.3294–3301, 2013.

[11] R.B. Girshick, “Fast r-cnn,” International conference on computer vision, pp.1440–1448, 2015.

[12] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, “Semantic contours from inverse detectors,” International Conference on Computer Vision (ICCV), 2011.

[13] Y.L. Hou, Y. Song, X. Hao, S. Yan, and M. Qian, “Multispectral pedestrian detection based on deep convolutional neural networks,” IEEE International Conference on Signal Processing, 2018.

[14] C. Li, S. Dan, R. Tong, and T. Min, “Illumination-aware faster r-cnn for robust multispectral pedestrian detection,” Pattern Recognition, 85, 2018.

[15] J. Li, X. Liang, S.M. Shen, T. Xu, and S. Yan, “Scale-aware fast r-cnn for pedestrian detection,” IEEE Trans. Multimedia, vol.20, no.4, pp.985–996, 2015.

[16] X. Li, Z. Liu, P. Luo, C.C. Loy, and X. Tang, “Not all pixels are equal: Difficulty-aware semantic segmentation via deep layer cascade,” Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.3193–3202, 2017.

[17] T.Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” Computer Vision and Pattern Recognition, pp.936–944, 2016.

[18] T.Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, “Focal loss for dense object detection,” International Conference on Computer Vision, pp.2999–3007, 2017.

[19] 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, 2016.

[20] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” Computer vision and pattern recognition, pp.3431–3440, 2015.

[21] J. Mao, T. Xiao, Y. Jiang, and Z. Cao, “What can help pedestrian detection?,” Computer Vision & Pattern Recognition, 2017.

[22] V.V. Molchanov, B.V. Vishnyakov, Y.V. Vizilter, O.V. Vishnyakova, and V.A. Knyaz, “Pedestrian detection in video surveillance using fully convolutional yolo neural network,” Automated Visual Inspection and Machine Vision II, 10334Q, 2017.

[23] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” Computer Vision and Pattern Recognition, pp.779–788, 2015.

[24] J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” IEEE Conference on Computer Vision & Pattern Recognition, 2017.

Chen Chen is currently a Ph.D. student in the College of System Engineering at the National University of Defense Technology. He received his B.S. degree from LanZhou University, China in 2013. His current research interest focuses on pedestrian detection and object segmentation.
Huaxin Xiao is currently a lecturer in the College of System Engineering at the National University of Defense Technology. He received his B.S. degree from the University of Electronic Science and Technology of China, China in 2012 and Ph.D. degree from National University of Defense Technology, China in 2018. He was a visiting student at the National University of Singapore from 2016 to 2018. He received the winner prize of object localization task in ILSVRC 2017. His current research interest focuses on saliency detection, image/video object segmentation.

Yu Liu received his B.S. degree from Northwestern Polytechnical University, Xian, China in 2005. He then received his MSC on image processing and Ph.D. on computer graphics from the University of East Anglia, Norwich, UK, in 2007 and 2011, respectively. He is currently an associate professor in the College of System Engineering at the National University of Defense Technology. His research interests include image/video processing, computer graphics, and visual haptic technology.

Maojun Zhang received his B.S. and Ph.D. degrees in system engineering from National the University of Defense Technology, Changsha, China, in 1992 and 1997, respectively. He is currently a professor in the College of System Engineering at the National University of Defense Technology. His current research interest focuses on computer vision, information system engineering and system simulation.