Thermal-Based Pedestrian Detection Using Convolutional Neural Networks and Multi-Regions Dropout Technique

Hsiang-Ying Wang, Kuan-Yi Li, Chia-Jen Lin, Sin-Ye Jhong, Hsien-I Lin, and Yung-Yao Chen*

***Abstract*—Pedestrian detection is one of the most common research topics in the area of computer vision. It has a significant impact on the safety of autonomous driving and related surveillance applications. Recently, because of the rising of Convolutional Neural Networks (CNN) in deep learning techniques, object detection algorithms have made a significant breakthrough in their accuracy and robustness. However, most detection algorithms can only perform stably under the environments where lighting is sufficient. Nighttime pedestrian detection remains a challenging problem. In this paper, we present a thermal-based framework to solve the nighttime pedestrian detection problem, which utilizes the thermal camera and extends the Faster R-CNN method. Moreover, a multi-scale model scheme is used to enrich the learning information of pedestrian features, and a feature region segmentation method is added to solve the occlusion issue. From the experimental results, it demonstrates that our proposed method achieves higher performance compared to current classic deep learning pedestrian detection methods.**

***Index Terms*—Thermal imaging, pedestrian detection, convolutional neural networks (CNN), multi-regions dropout (MRD), modified ROI pooling**

I. Introduction

Object detection and object detection are two of the most essential research topics in the areas of computer vision and smart robots [1, 2]. Among different objects, human beings might be the most common object which must be concerned. As a result, pedestrian detection is one of the most active topics in the computer vision. For the applications such as self-driving cars, automatic surveillance systems, and intelligent robots, many computer vision methods involving people have been proposed recently.

Zhang et al. [3] proposed a remote pedestrian detection algorithm, which is based on edge information and CNN. Because the pedestrian information is not clear in the remote imaging, the method of [3] utilizes the edge features of shallow layers, and combines them with grayscale images to replace the original color images. Liu et al. [4] proposed a new perspective to solve the pedestrian detection problem, in which pedestrian detection is considered as a high-level semantic feature detection task. Therefore, the method of [4] effectively reduces the use of sliding-window classifiers, which is commonly used in the traditional methods. Lin et al. [5] proposed a graininess-aware deep feature learning method to solve the pedestrian detection problem, in which fine-grained information is combined with the convolutional features. In addition, a pedestrian attention mechanism is presented in [5] to determine the ROI (region of interest) of pedestrian candidates efficiently. Li et al. [6] proposed a novel pedestrian detection method, which can effectively detect pedestrians even under hazy weather. The method of [6] utilizes the you-only-look-once method, and adds a weighted combination layer to integrate multiscale feature maps. Liu et al. [7] proposed a refined pedestrian detection method, which can be used in a crowd. To refine the bounding boxes, the method of [7] presents a novel Non-Maximum Suppression (NMS) scheme, which can adaptively adjust the suppression threshold according to the target density. In addition, a sub-network is added to learn the density scores, which is able to be embedded into the one-stage detector or the two-stage detector. Du et al. [8] proposed a fast pedestrian detection method, which is based on a deep neural network fusion architecture. The method of [8] has the advantage of parallel processing of multiple networks that increases its computational efficiency. Furthermore, a single shot deep convolutional network is presented to find the pedestrian candidates with different sizes. There are also several interesting pedestrian detection methods which rely on the Lidar sensing [9, 10].

Occlusion rises another challenging problem for the topic of pedestrian detection. Occlusion problem occurs when the pedestrian is completely or partially hidden by other objects. For example, two persons are walking past each other in the pathway or the crowded occlusion (occluded by a group of people). As a result, we only can see part of the human body, which increase the difficulty of detecting the existence of a pedestrian. Zhou and Yuan [11] proposed a CNN-based framework to deal with the occlusion problem involving pedestrian detection, which estimates the degree of occlusion by regressing multiple boxes for the full body localization and the partial body localization. Therefore, the method of [11] has two branches in its CNN framework. In addition, a new criterion is presented to select positive training examples to alleviate the heavy occlusion problem. Pang et al. [12] proposed an occluded pedestrian detection method, which can effectively cope with the problem of intra- and inter-class occlusion at the same time. The method of [12] presents a novel mask-guided attention network, which addresses most on the visible pedestrian regions, and meanwhile, suppresses the occluded regions by modulating the full-body characteristics. Zhang et al. [13] proposed an occlusion-aware R-CNN framework to detect pedestrians in a crowd. The method of [13] presents a novel aggregation loss to enforce the proposals to be as near to their neighboring objects as possible. In addition, an occlusion-aware ROI pooling unit is

This paper was first submitted on January 17, 2021.

Hsiang-Ying Wang, Kuan-Yi Li, Chia-Jen Lin, and Hsien-I Lin are with the Grad. Inst. of Automation Technology, National Taipei Tech. Univ., Taipei, Taiwan.

Sin-Ye Jhong is with the Dept. Engineering Science, National Cheng Kung Univ., Tainan, Taiwan.

Yung-Yao Chen is with the Dept. Electronic and Computer Engineering, National Taiwan Univ. of Science and Technology, Taipei, Taiwan. (corresponding author, email: yungyaochen@gapps.ntust.edu.tw)

(a)

(b)

(c)

Fig. 1. Example of (a) optical image, (b) thermal image, and (c) thermal camera.

designed to integrate the information of prior structure into the visibility prediction of a human body.

Compared with daylight-based computer vision, nighttime computer vision is a difficult task because the luminance in nighttime is very weak [14]. As a result, thermal camera has become an alternative solution to capture clear images and has many applications [15, 16]. As shown in Fig. 1(a), the appearance of a pedestrian has poor visibility from an optical CMOS (Complementary Metal-Oxide-Semiconductor) camera at nighttime. However, using a thermal camera can achieve significant visibility of pedestrian features, as shown in Fig. 1(b). The thermal camera used in this study is FLIR Boson 640, as shown in Fig. 1(c). Many thermal-based pedestrian detection methods have been proposed recently. Baek et al. [17] proposed a thermal-based pedestrian detection method, which uses a new feature called thermal-position-intensity-histogram of oriented gradient to extract the pedestrian features efficiently. In addition, the method of [17] also presents the additive kernel SVM scheme. Ma et al. [18] proposed a pedestrian detection method, which uses the information of unmanned aerial vehicle (UAV) thermal imagery. The method of [18] especially solves the difficulties of low-resolution of imagery and image instability due to UAV.

II. PROPOSED METHOD

2.1 Proposal Method

The architecture of our MRD-RCNN is shown in Fig. 2. It mainly consists of feature extraction, region proposal, ROI pooling, and Multiple Regions Dropout (MRD [19]) phases. For extracting a generic CNN features, similar to other works, we use a classification network trained with ImageNet. In the Pooling layer, we select appropriate boxes among them for training and inference in consideration of the data balance between foreground and background samples.

In the MRD network, several region proposals are generated based on using different kernel sizes. We then decompose the generated proposals with defined factors to enhance the diversity of region proposals. As a result, the entire object region is decomposed into several small regions, and the entire object detection mode can be viewed as the combination of several part models. Finally, the combined feature maps are used for object regression and classification.

2.1.1 Convolutional Layer

In the convolutional layer stage, for the input thermal image, the shared feature map is extracted through convolution operation. The architecture adopted here is the VGG16 [20] deep convolutional neural network model, which mainly uses the first 17 convolutional layers as feature extraction. However, due to the different processing objects, we adjusted the network. As shown in Table 1, in order to perform pedestrian detection, we fixed the input image to a 416×416 size image and reduced the number of uses of max Pooling from 4 times to 3 times to avoid over-compression of the feature map. Therefore, the obtained thermal image sharing feature map is with the size of 52×52, which is then combined with the subsequent RPN [21] and ROI Pooling.

Table 1 Modified VGG16

Layer	Type/Repeat	# of Filter	Size/Sride
1	Convolutional/2	64	3×3
3	Max Pooling	-	2×2
4	Convolutional/2	128	3×3
6	Max Pooling	-	2×2
7	Convolutional/3	256	3×3
10	Max Pooling	-	2×2
11	Convolutional/3	512	3×3
14	Convolutional/3	512	3×3

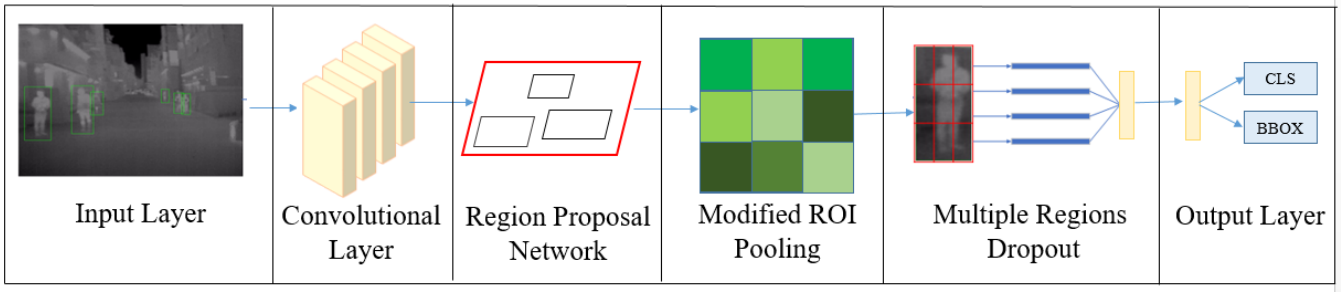


Fig. 2. The architecture of the proposed MRD-CNN.

2.1.2 *Region Proposal Network*

The main purpose of the RPN stage is to generate an appropriate region proposal. The operation method applies a 3x3 filter to convolve on the feature map, and applies different scales and anchor ratios to each position to generate an anchor frame for foreground/background classification. The bounding box regression scheme is used to find preliminary regional proposals.

The detailed process is described as follows. After obtaining an anchor frame with positive and negative sample labels, the anchor frames are used to train the RPN, in which the loss function can be expressed by

$$L(p_i,t_i)_{\text{RPN}} = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i,t_i^*) \quad (1)$$

The loss function is composed of and two parts: the classification loss L_{cls} , which is the output of the softmax layer; and the regional loss L_{reg}, which is the error between the offset of the anchor box and the true value used for the training of the bounding box regression. The index i indicates the anchor frame number, p_i is the foreground probability of the anchor frame, and p_i^* is the anchor frame label (If it is marked as positive sample, set as 1. If it is marked as negative sample, set as 0), where the anchor frame is corresponding to the positive sample real frame coordinate information. Since the normalization process uses the same number of N_{cls}= 256 (one mini-batch) and N_{reg}= 256 (total number of anchor points), the adjustment parameter λ is set to 1. The classic cross-entropy loss function (cross-entropy loss) in logistic regression is used in L_{cls}, which is defined by the following formula:

$$L_{cls}(p_i, p_i^*) = -log\left[p_i^* p_i + (1-p_i^*)(1-p_i)\right] \quad (2)$$

There are two constraints: first, only when the anchor box is a positive sample ($p_i^*=1$) will it participate in the calculation; second, only the foreground is needed to modify the border. Therefore, when defining the positive and negative sample labels, an anchor box can only match with a real box, and the real box can match multiple anchor boxes.

However, the detailed formula for finding anchor boxes is followed as function (3). We can optimize the model by calculating the $smooth_{L1}$ loss of the target and the prediction.

$$L_{reg}(t_i,t_i^*) = \sum_{i\in\{x,y,w,h\}} smooth_{L1}(t_i - t_i^*) \quad (3)$$

$$t_x = \frac{x - x_a}{w_a},\ t_x^* = \frac{x^* - x_a}{w_a} \quad (4)$$

$$t_y = \frac{y - y_a}{h_a},\ t_x^* = \frac{y^* - y_a}{h_a} \quad (5)$$

$$t_w = log\left(\frac{w}{w_a}\right),\ t_w^* = log\left(\frac{w^*}{w_a}\right) \quad (6)$$

$$t_h = log\left(\frac{h}{h_a}\right),\ t_h^* = log\left(\frac{h^*}{h_a}\right) \quad (7)$$

In the above formulas, *x*, *y*, *w*, *h* are respectively the center coordinates, the width, and the height of the bounding box. In (4), x is the prediction box, x_a is the anchor box; meanwhile, x^* is the real box, and t_i is the relative prediction box. For the offset of the anchor frame, t_i^* is the offset of the real frame to the anchor frame. Finally, the symbol $smooth_{L1}$ is used to calculate the difference between the two as follows.

$$smooth_{L1}(x) = \begin{cases} 0.5x^2, & if\ |x|<1 \\ |x|-0.5, & otherwise \end{cases} \quad (8)$$

The main purpose of the above bounding box regression is to predict an offset for all positive sample anchor frames, which is as close as possible to the offset of the anchor frame to the real frame, to achieve the purpose of regional proposal. In addition, it can tell the classifier whether it is possible, and these area proposals are sent to the subsequent ROI pooling processing.

2.2 *Proposed Modified ROI Pooling*

In the ROI pooling stage, in order to obtain better inspection quality, we compared the original ROI pooling and ROI align schemes, and then combined the advantages of the above methods. As shown in Fig. 3, the red dotted line indicates the position of the candidate image in the feature map. It can be seen from the figure that the red dotted line is no longer on the

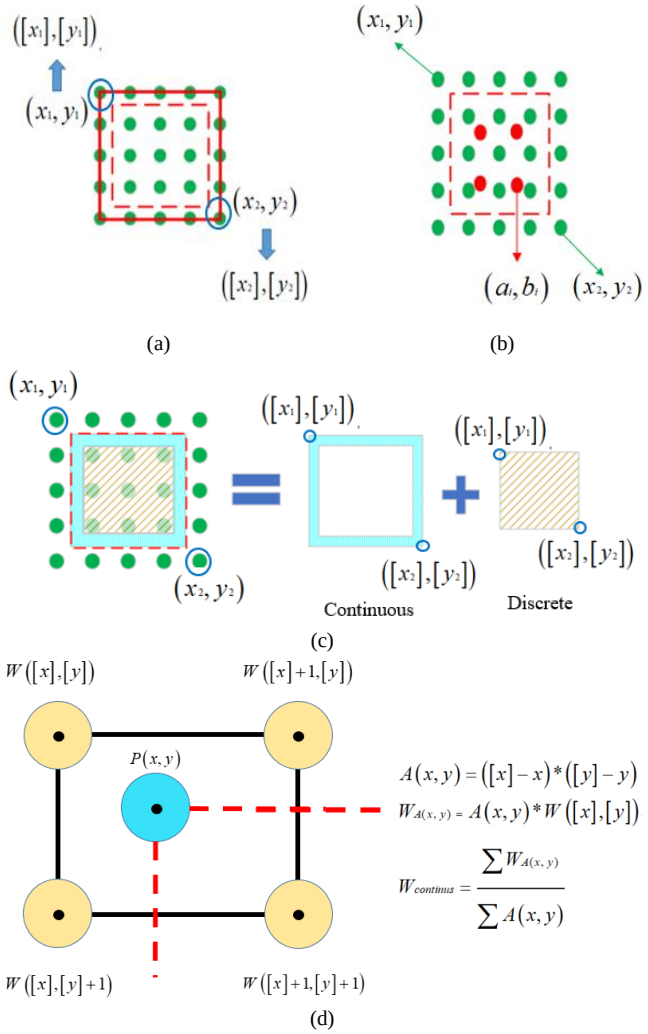


Fig. 3. Illustration of different types of pooling: (a) ROI pooling, (b) ROI align, (c) modified ROI pooling, and (d) continuous-domain region pooling.

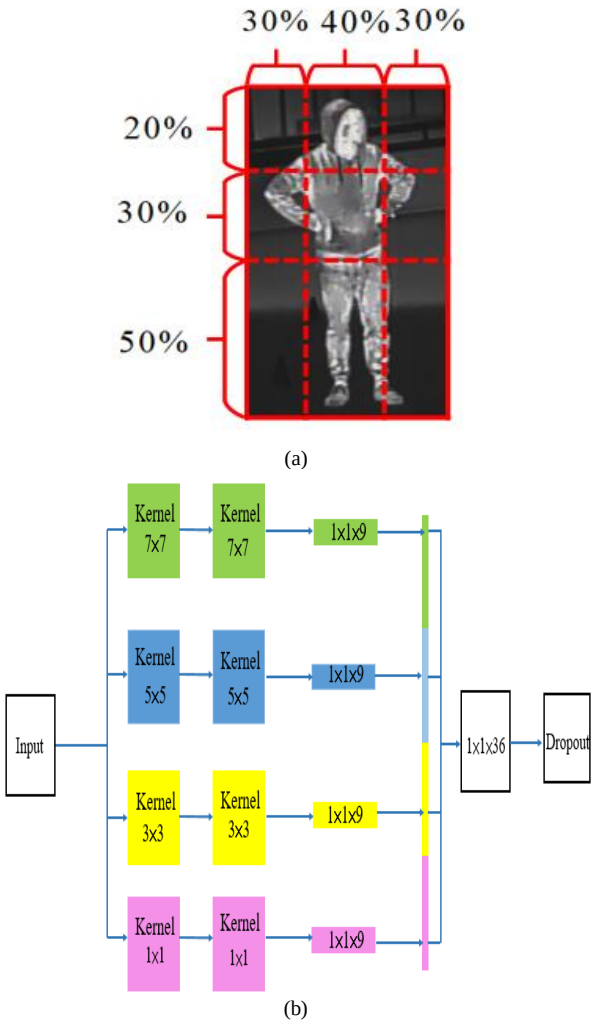


Fig. 4. (a) Pedestrian area decomposition ratio definition, and (b) MRD architecture flow chart.

physical points (i.e., the green points). As shown in Fig. 3(a), the ROI pooling method is relatively rough, and its strategy is to directly perform quantization and integer processing, which tends to result in a loss of accuracy. As shown in Fig. 3(b), the ROI align method first performs interpolation, dividing the candidate image area into N sub-regions (the example shown in Fig. 3b is one sub-region, represented by 4 solid red dots), and finally performing an average process on these 4 sub-regions during pooling.

Modified ROI Pooling: As shown in Fig. 3(c), we divide the problem into the discrete-domain regions and the continuous-domain regions for processing. The former method is the same as the ROI pooling method. The regions are summed and averaged. On the other hand, for continuous-domain regions, interpolation scheme is used as shown in Fig. 3(d). After the method of planting, the values of the area are integrated and final merged to obtain the output result. Compared with other methods, our proposed ROI pooling method improves the accuracy of the ROI definition.

2.3 Multiple Regions Dropout

The concept of multi-scale processing is to start from the image pyramid, and obtain various size changes of objects by inputting multi-scale features. When the feature maps are obtained through RPN, because different convolution sizes have different fields of view, we use 1×1, 3×3, 5×5, and 7×7 convolution sizes to sample the feature maps at different scales. Since pedestrians have a similar appearance, the individual feature maps are disassembled here, and divided into nine areas based on the nine sub-blocks shown in Fig. 4(a).

As shown in Fig. 4(b), because there are 4 scales, the features are extracted into 36 pedestrian features representing different proportions. The MRD layer is composed of the merge layer and the dropout layer, which are used to simulate different occlusion features during training. It will randomly discard 36 neurons representing the pedestrian features previously arranged by the merged layer. By doing so, the network can learn random occlusion features without providing occlusion data, and conduct occlusion and non-occlusion training in the input layer at the same time.

This method can not only achieve the purpose of learning occlusion features without increasing pedestrian information, but also avoid model confusion (occlusion and non-occlusion) caused by directly training two different pedestrian features, which will make the model performance worse. As shown in Fig. 5, under the control of predefined dropout rate, we can use

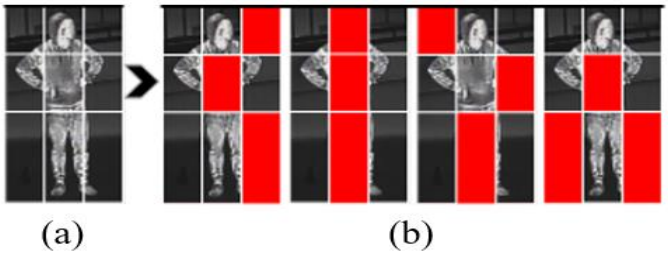


Fig. 5. Training pedestrian types: (a) Input, and (b) Train Type.

MRD network to obtain pedestrians in different shades for network training.

III. Experimental Results

In this section, the extended model used is based on the Faster R-CNN [22] framework, which uses TensorFlow. The network is initialized with Glorot and trained with 80,000 iterations using the SGD [23] optimizer, with a learning rate of 0.0001 and a batch size of 32. No data augmentation is used during training. All experiments are run on a single GPU 2080Ti and CPU Intel Core i7 8700 4.6GHz computer.

3.1 Dataset

In this subsection, in order to effectively evaluate the feasibility and effectiveness of the night pedestrian detection system, we use the night pedestrian detection database (KAIST) [24]. The data has been re-corrected, and the detailed information of this KAIST database is provided in Table 2. The training set contains a total of 21307 characters, and the test set contains 3390 characters. Fig. 6 provides the difference between entire pedestrian and occluded pedestrians in the database.

Table 2 KAIST database information.

KAIST	Images	Label	Number
Train set	7601	Entire	18230
		Occluded	3077
Test set	2252	Entire	2910
		Occluded	480

3.2 Experimental Results

In order to prove the effectiveness of the proposed method, we conducted data analysis on different sets of data, as shown in Table 3. First, the performance of detecting complete pedestrian is discussed. Under the same benchmark, our proposed method is compared with other representative methods, including Faster RCNN, and YOLOv3 [25]. During the experiment, our method is slightly less accurate than the original Faster R-CNN. But the recall rate has risen sharply, which should represent a degree of generalization of the proposed model.

The test results are further analyzed. The performance of detecting occluded pedestrian is discussed as follows. As shown in Table 4, through the test of the occluded objects, we use all the original parameters and codes for data training and testing. We have greatly exceeded the comparison target in each index, which means that the model is effective for the occluded objects with stronger identification learning ability. As shown in Table 5, although the current detection method is still far from the human standard, our proposed MRD-RCNN is obviously more correct than other methods.

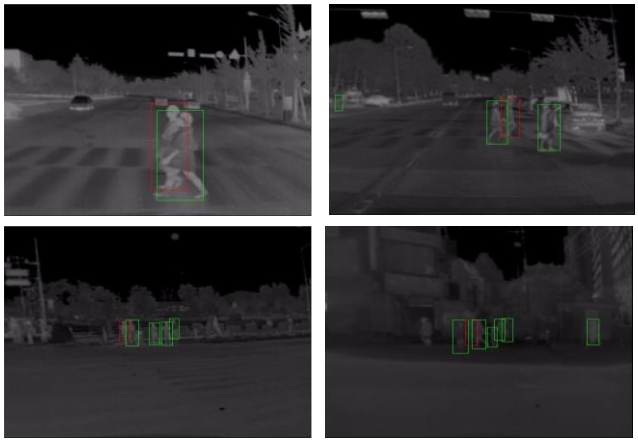


Fig. 6. Pedestrian display (red box represents the occluded pedestrians, and green box represents the entire pedestrians).

IV. Conclusions

In this paper, a thermal-based pedestrian detection method is proposed, which uses convolutional neural network and multi-regions dropout techniques. From the experimental results, we prove that the proposed thermal-based pedestrian detection method has stable performance under the nighttime cases and even the heavy occlusion conditions. In addition, the proposed method validates it superiority over other current pedestrian detection methods.

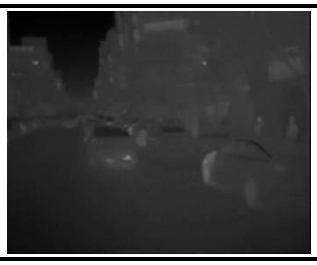
Table 3 Entire pedestrian output results.

Methods	Label	TP	FP	Recall	Precision	F1-Measure
Faster R-CNN	Entire	1660	540	57%	**75%**	65%
YoloV3	Entire	1850	1026	63%	64%	63%
Our	Entire	2177	820	**74%**	72%	**73%**

Table 4 Occluded pedestrian output results.

Methods	Label	TP	FP	Recall	Precision	F1-Measue
Faster R-CNN	Occluded	172	62	35%	73%	48%
YoloV3	Occluded	255	128	53%	66%	59%
Our	Occluded	340	108	**70%**	**75%**	**73%**

Table 5 Comparison among detection results using different methods.

A			
B			
C			

REFERENCES

[1] H. Lin, Y. Chen, and Y. Chen, "Robot vision to recognize both object and rotation for robot pick-and-place operation,"*2015 International Conference on Advanced Robotics and Intelligent Systems*, pp. 1-6, 2015.

[2] C. Hsia, and C. Lai, "Embedded vein recognition system with wavelet domain," *Sensors and Materials*, pp. 3221-3234, 2020.

[3] C. Zhang, N. Tan, and Y. Lin, "Remote Pedestrian Detection Algorithm Based on Edge Information Input CNN," *2019 3rd High Performance Computing and Cluster Technologies Conference*, pp. 190-194, 2019.

[4] W. Liu, S. Liao, W. Ren, W. Hu, and Y. Yu, "High-Level Semantic Feature Detection: A New Perspective for Pedestrian Detection," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5187-5196, 2019.

[5] C. Lin, J. Lu, G. Wang, J. Zhou, "Graininess-Aware Deep Feature Learning for Pedestrian Detection," *2018 European Conference on Computer Vision*, pp. 732-747, 2018.

[6] G. Li, Y. Yang and X. Qu, "Deep Learning Approaches on Pedestrian Detection in Hazy Weather," *IEEE Transactions on Industrial Electronics*, pp. 8889-8899, 2020.

[7] S. Liu, D. Huang, and Y. Wang, "Adaptive NMS: Refining Pedestrian Detection in a Crowd," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6459-6468, 2019.

[8] X. Du, M. El-Khamy, J. Lee, and L. Davis, "Fused DNN: A Deep Neural Network Fusion Approach to Fast and Robust Pedestrian Detection," *2017 IEEE Winter Conference on Applications of Computer Vision*, pp. 953-961, 2017.

[9] H. Tang, S. Chien, W. Cheng, Y. Chen , and K. Hua, "Multi-cue pedestrian detection from 3D point cloud data," *2017 IEEE International Conference on Multimedia and Expo*, pp. 1279-1284, 2017.

[10] T. Lin *et al*., "Pedestrian Detection from Lidar Data via Cooperative Deep and Hand-Crafted Features," *2018 IEEE International Conference on Image Processing*, pp. 1922-1926, 2018.

[11] C. Zhou, J. Yuan, "Bi-box Regression for Pedestrian Detection and Occlusion Estimation," *2018 European Conference on Computer Vision*, pp. 135-151, 2018.

[12] Y. Pang, J. Xie, M. H. Khan, R. M. Anwer, F. Shahbaz Khan, and L. Shao, "Mask-Guided Attention Network for Occluded Pedestrian Detection," *2019 IEEE/CVF International Conference on Computer Vision*, pp. 4967-4975, 2019.

[13] S. Zhang, L. Wen, X. Bian, Z. Lei, S. Z. Li, "Occlusion-aware R-CNN: Detecting Pedestrians in a Crowd," *2018 European Conference on Computer Vision (ECCV)*, pp. 637-653, 2018.

[14] C. Hsia, S. Yen, and J. Jang, "An intelligent IoT-based vision system for nighttime vehicle detection and energy saving," *Sensors and Materials*, pp. 1803-1814, 2019.

[15] Y. Chen, W. Chen and H. Ni, "Image segmentation in thermal images," *2016 IEEE International Conference on Industrial Technology*, pp. 1507-1512, 2016.

[16] S. Chien, F. Chang, C. Tsai and Y. Chen, "Intelligent all-day vehicle detection based on decision-level fusion using color and thermal sensors," *2017 International Conference on Advanced Robotics and Intelligent Systems*, pp. 76-76, 2017.

[17] J. Baek, S. Hong, J. Kim, E. Kim, "Efficient Pedestrian Detection at Nighttime Using a Thermal Camera," *Sensors*, pp.1850, 2017.

[18] Y. Ma, X. Wu, G. Yu, Y. Xu, "Pedestrian Detection and Tracking from Low-Resolution Unmanned Aerial Vehicle Thermal Imagery," *Sensors*, pp.446, 2016.

[19] H. Wu and X. Gu, "Towards dropout training for convolutional neural networks," *Neural Networks*, pp. 1-10, 2015.

[20] K. Simonyan, Z. Andrew, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *2015 International Conference on Learning Representations*, 2015.

[21] Z. Zhong, S. Lei, and H. Qiang, "An anchor-free region proposal network for Faster R-CNN-based text detection approaches," *International Journal on Document Analysis and Recognition* , pp. 315-327, 2019.

[22] S. Ren, et al, "Faster r-cnn: Towards real-time object detection with

region proposal networks," *IEEE transactions on pattern analysis and machine intelligence*, pp. 1137-1149, 2016.

[23] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv*, 2016.

[24] Y. Choi, et al. "KAIST multi-spectral day/night data set for autonomous and assisted driving," *IEEE Transactions on Intelligent Transportation Systems*, pp. 934-948, 2018.

[25] J. Redmon and F. Ali, "Yolov3: An incremental improvement," *arXiv preprint arXiv*, 2018.