Journal of Theoretical and Applied Information Technology
15th December 2021. Vol.99. No 23

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

LEARNING TO LOCATE OBJECTS IN CROWDED SCENES BASED ON FULLY CONVOLUTIONAL NETWORK

HOANH NGUYEN

Faculty of Electrical Engineering Technology, Industrial University of Ho Chi Minh City, Ho Chi Minh City, Vietnam

E-mail: nguyenhoanh@iuh.edu.vn

ABSTRACT

Locating objects in crowded scenes is a challenging problem since objects such as pedestrians or vehicles often gather and occlude each other. This paper proposes a new approach for locating pedestrians in crowded scenes based on fully convolutional network. First, ResNest, which combines the channel-wise attention strategy with multipath network layout to extract feature from images, is used as the backbone network to extract features from input image. ResNest is a simple architecture that achieves better speed-accuracy trade-offs than state-of-the-art CNN architectures without incurring excessive computational costs. Since the features produced by the backbone network often have small receptive fields and weak representation capabilities, the feature enhancement model is then designed to refine the features efficiently. The feature enhancement model is attached behind the backbone network and makes features deeper and more expressive than before. Based on the enhanced feature pyramid, the detection head including three branches is adopted to predict the classification score for each point on the feature pyramid, regress the distances from the point to the four sides of a bounding box, and predict the center-ness score which is multiplied by the classification score to rank the bounding box in NMS. Experimental results on the CityPersons dataset show the effectiveness of the proposed method on locating objects in crowded scenes.

Keywords: *Fully Convolutional Neural Network, ResNest, Object Detection, Pyramid Network, Crowded Scenes*

1. INTRODUCTION

Object detection is an important research topic in computer vision field with various applications, such as autonomous driving, video surveillance, and robotics. Object detection predicts a series of bounding boxes enclosing object instances in an image. Traditionally, scanning an image in a sliding-window paradigm is a common practice for object detection. In this paradigm, designing hand-crafted features is of critical importance for state-of-the-art performance, which still remains as a difficult task. Recent advances in object detection are driven by the success of deep convolutional neural networks (CNNs), which uses the bounding box regression techniques to accurately localize the objects based on the deep features. With the development of object detection, currently popular object detectors can be categorized by whether they use anchor boxes or not, including anchor-based detectors and anchor-free detectors. Anchor-based detectors inherit the ideas from traditional sliding-window and proposal-based detectors such as Fast R-CNN [1]. Faster-RCNN [2] proposed Region Proposal Network (RPN) to generate proposals in a unified framework. Beyond its success on generic object detection, numerous adapted Faster-RCNN detectors were proposed and demonstrated better accuracy for pedestrian detection [3]. However, when the processing speed is considered, Faster-RCNN is still unsatisfactory because it requires two-stage processing. Alternatively, as a representative one-stage detector, SSD [4] discards the second stage of Faster-RCNN and directly regresses the default anchors into detection boxes. Recently, Cascade R-CNN [5] has proved that Faster R-CNN can be further improved by applying multi-step ROI-pooling and prediction after RPN. Besides, another recent work called RefineDet [6] suggests that ROI-pooling can be replaced by a convolutional transfer connection block after RPN.

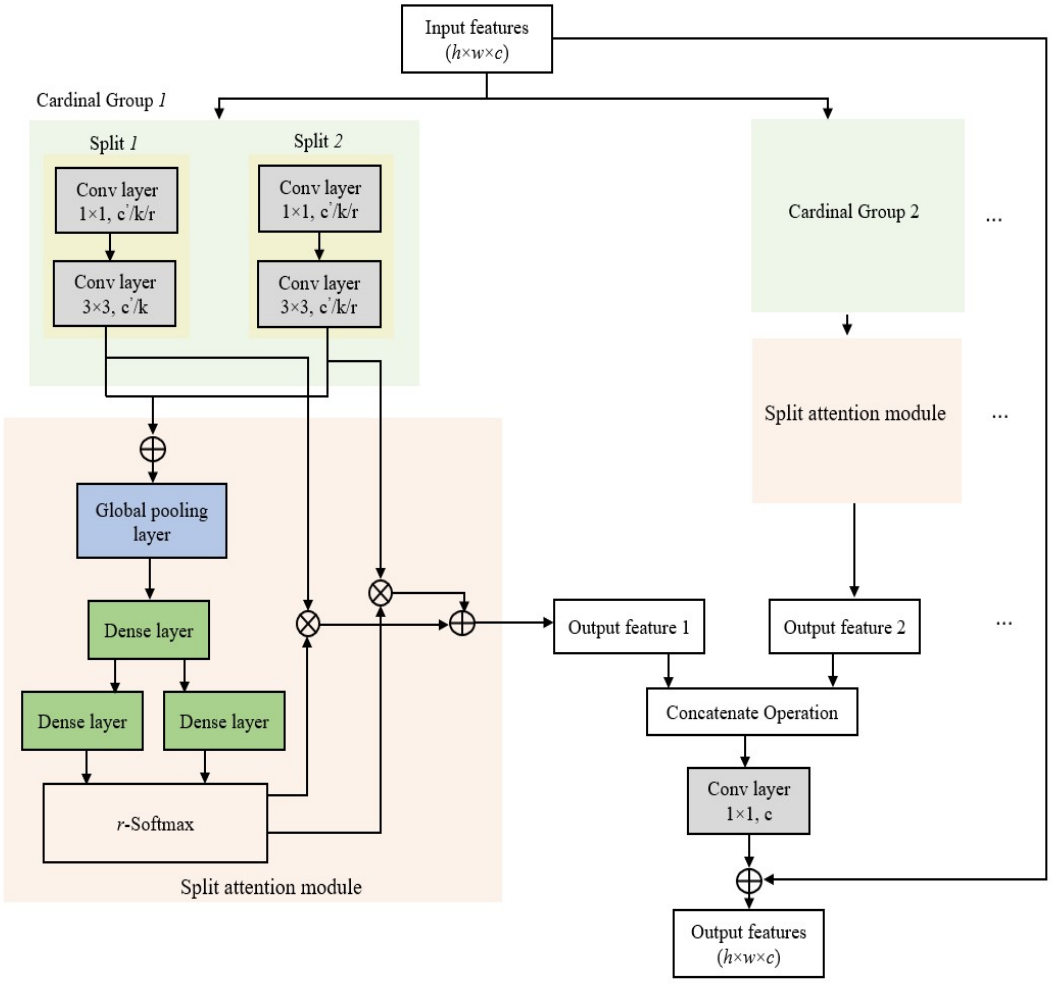

Figure 1: The Structure of ResNest Block.

More recently, anchor-free detectors have attracted substantial attention due to their novelty and simplicity. The most popular anchor-free detector might be YOLOv1 [7]. Instead of using anchor boxes, YOLOv1 predicts bounding boxes at points near the center of objects. Only the points near the center are used since they are considered to be able to produce higher quality detection. However, since only points near the center are used to predict bounding boxes, YOLOv1 suffers from low recall as mentioned in YOLOv2 [8]. As a result, YOLOv2 employs anchor boxes as well. Another family of anchor-free detectors formulates the object detection problem as a key-point or a semantic-point detection problem, including CornerNet [9], CenterNet [10], ExtremeNet [11], and Reappoints [12]. Another type of anchor-free detectors, including DenseBox [13], FASF [14], FoveaBox [15], and FCOS [16] are similar to anchor-based one-stage methods, but they remove the usage of anchor boxes. Instead, they classify each point on the feature pyramids [17] into foreground classes or background, and directly predict the distances from the foreground point to the four sides of the ground-truth bounding box, to produce the detection.

2. METHODOLOGY

2.1 Feature Extraction Network

ResNest [18] is a simple architecture which combines the channel-wise attention strategy with multipath network layout to extract feature from images. ResNest architecture can capture cross-channel feature correlations, while preserving independent representation in the meta structure. As a result, ResNest architecture achieves better speed-accuracy trade-offs than state-of-the-art CNN architectures without incurring excessive computational costs. Inspired by the ResNest architecture, this paper adopts ResNest as the feature extraction network of the model. ResNest

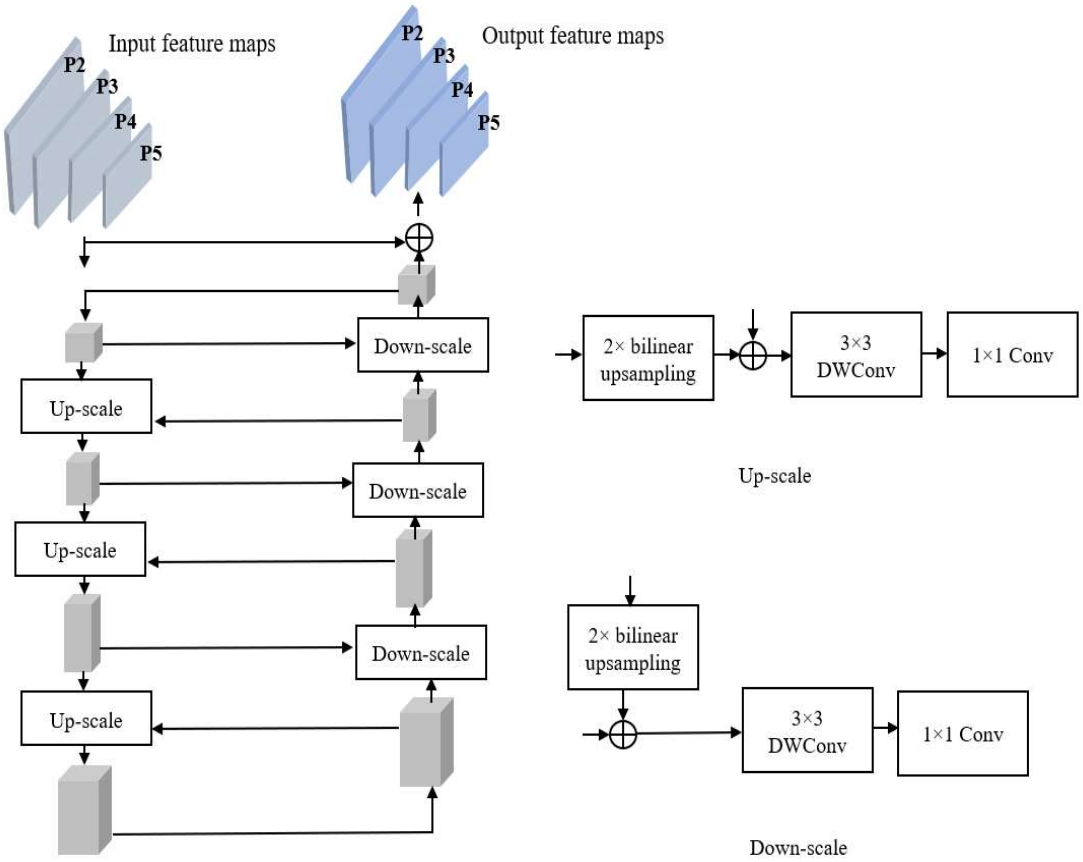

Figure 2: The Structure of The Feature Pyramid Enhancement Module.

architecture is based on the ResNet-D model [19] and ResNest block.

ResNest block is the main block in the ResNest architecture. Figure 1 shows the structure of the ResNest block. In ResNest block, input feature maps are divided into n separate groups (n=4 in this paper as in original model). The resulting feature groups are regarded as cardinal groups. ResNest introduced a new radix hyperparameter r that indicates the number of splits within each cardinal group, thus the total number of feature groups t is calculated as follow:

$$t = k.r \quad (1)$$

where r=2 in this paper as in original model.

In each individual group, a series of transformations, including a 1×1 convolution layer followed by a 3×3 convolution layer, is applied to generate the intermediate representation of each group. Split attention module is used to fuse feature maps in each split groups. In split attention module, the feature maps of each split group are first fused via element-wise summation across multiple splits. The fused feature maps are then fed into a global average pooling across spatial dimensions to effectively collect global context information with embedded channel-wise statistics. Two fully connected layers with BN and ReLU activation are used to generate each feature map channel as a weighted combination over splits. The output feature maps generated by split attention module from each cardinal group are then concatenated along the channel dimension.

In addition to replace Residual block with ResNest block in the ResNet-D model [19], ResNest model also adopts two effective modifications:

- The first 7×7 convolutional layer is replaced with three consecutive 3×3 convolutional layers, which have the same receptive field size with a similar computation cost as the original design.
- A 2×2 average pooling layer is added to the shortcut connection prior to the 1×1 convolutional layer for the transitioning blocks with stride of two.

In addition, instead of using strided convolution at the transitioning block, ResNest architecture uses an average pooling layer with a kernel size of 3×3. ResNest model captures cross-channel feature correlations, while preserving independent representation in the meta structure. ResNest block performs a set of transformations on low dimensional embeddings and concatenates their

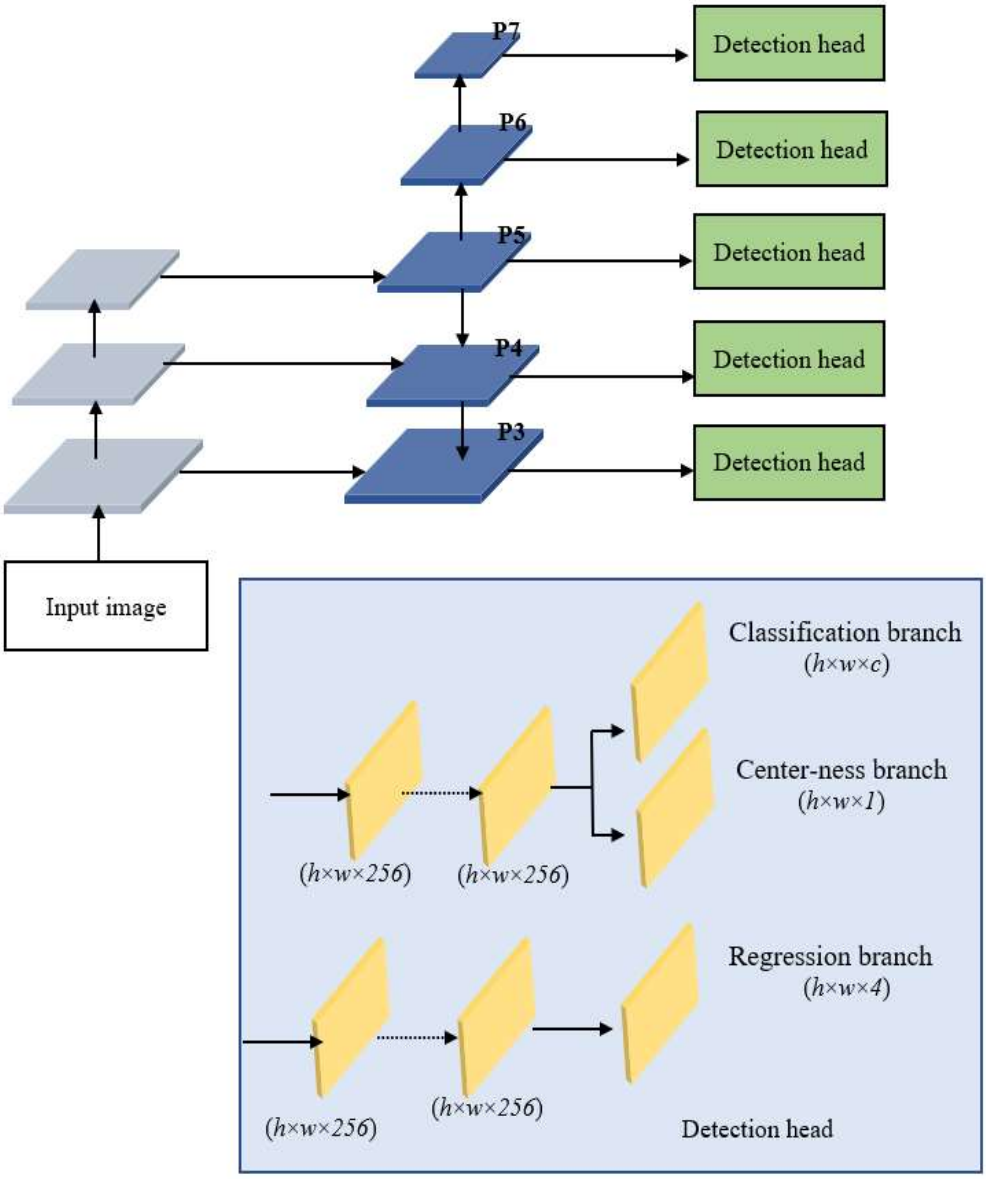

Figure 3: FCOS Architecture Based on FPN.

outputs as in a multi-path network. Each transformation incorporates channel-wise attention strategy to capture interdependencies of the feature map.

2.2 Feature Pyramid Enhancement Module

The features generated by the backbone network usually have small receptive fields and weak representation capabilities [17]. To overcome this problem, FPN [17] proposed to enhance multi-scale feature maps by fusing the low-level and high-level information. However, FPN increases the computational cost while the receptive fields are still unchanged. This paper proposes a feature pyramid enhancement module based on PAN++ [20] that can improve the feature maps generated by the backbone network efficiently. Figure 2 shows the structure of the proposed feature pyramid enhancement module. It consists of two stages, including up-scale enhancement stage and down-scale enhancement stage. The up-scale enhancement stage is applied to the input feature maps to enhances the input feature maps with strides of 32, 16, 8, and 4 pixels, respectively. The down-scale enhancement stage takes the feature pyramid generated by up-scale enhancement stage as input, and the enhancement is

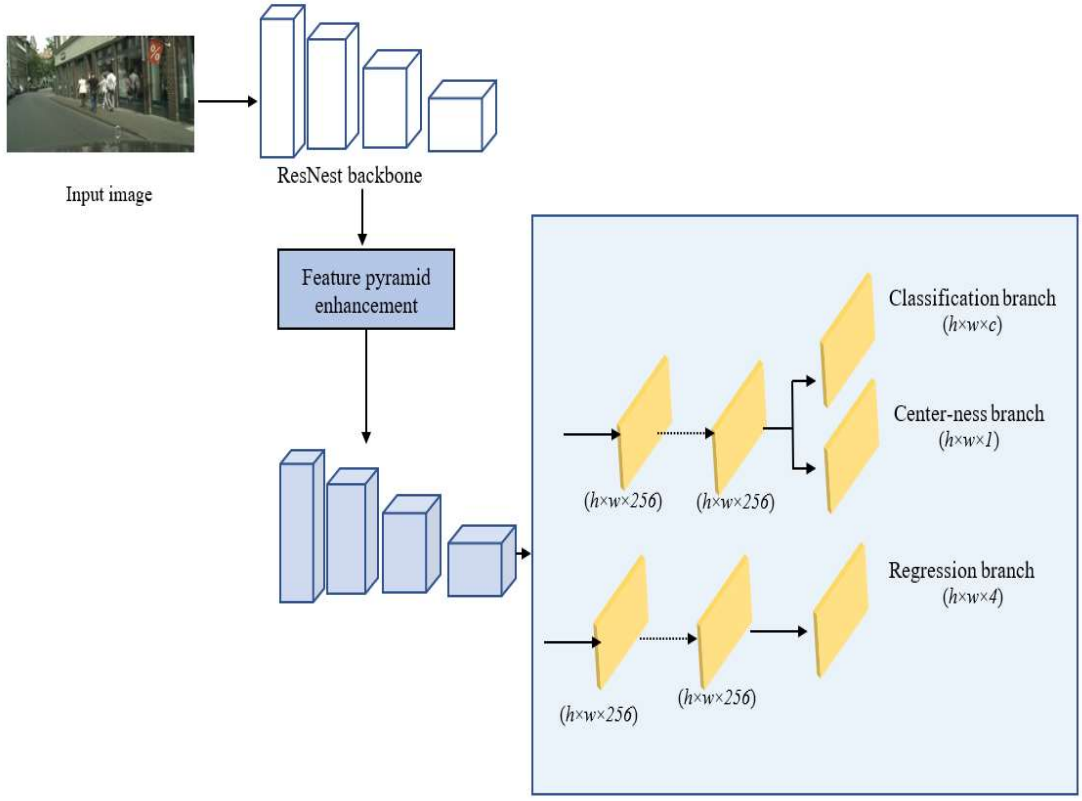

Figure 4: The Proposed Model.

conducted from 4-stride to 32-stride, respectively. Finally, the output feature maps are generated by the elementwise addition result of the input feature maps and the feature maps generated by the down-scale enhancement stage. In both up-scale and down-scale stages, the separable convolution [21] is adopted instead of the regular convolution to enlarge the receptive field. Separable convolution includes a 3×3 depth-wise convolution followed by a 1×1 projection.

2.3 FCOS: Fully Convolutional Detection Network

FCOS [16] is a fully convolutional one-stage object detection framework. FCOS is based on anchor-free branches that takes advantages of all points in a ground truth bounding box to predict the bounding boxes and the low-quality detected bounding boxes are suppressed by the proposed "center-ness" branch. As a result, FCOS is able to provide comparable recall with anchor-based detectors. Figure 3 shows the structure of FCOS framework. FCOS is built on FPN [17] and detect different sizes of objects on different levels of feature maps. To be more specific, FCOS uses five levels of feature maps defined as {P3; P4; P5; P6; P7}. While {P3, P4, P5} are produced by the backbone CNNs' feature maps C3, C4 and C5 followed by a 1×1 convolutional layer with the top-down connections. {P6, P7} are produced by applying one convolutional layer with the stride being 2 on P5 and P6, respectively. As a result, the feature levels {P3, P4, P5, P6, P7} have strides {8, 16, 32, 64, 128}, respectively. Unlike anchor-based detectors, which assign anchor boxes with different sizes to different feature levels, FCOS directly limit the range of bounding box regression for each level. There are three branches in the detection head of FCOS. The first branch predicts the classification score for each point on the feature map. The second branch regresses the distances from the point to the four sides of a bounding box. The final branch predicts the center-ness score which is multiplied by the classification score to rank the bounding box in NMS. The center-ness branch is proposed to suppress low-quality detected bounding boxes without introducing any hyper-parameters. This branch can down-weight the scores of bounding boxes far from the center of an object. As a result, with high probability, low-quality bounding boxes might be filtered out by the final non-maximum suppression process, improving the detection

performance remarkably. To verify the effectiveness of the proposed modules, this paper builds the proposed model based on the FCOS model and evaluate it on CityPersons benchmark [22].

2.4 The Proposed Model

The proposed model based on ResNest architecture, feature pyramid enhancement module, and FCOS model for locating objects in crowded scenes is show in Figure 4. The ResNest architecture first extracts feature maps from input images. Four feature maps are generated by conv2, conv3, conv4, and conv5 layers of the backbone network, whose resolutions are 1/4, 1/8, 1/16 and 1/32 compared with the input image, respectively. For computational efficient, the channel number of each feature map is reduced to 128 via 1×1 convolution layers. These reduced feature maps are considered as thin feature maps. Since the thin features produced by the backbone network often have small receptive fields and weak representation capabilities, the feature enhancement model is used to refine the features efficiently. The feature enhancement model is attached behind the backbone network and makes features deeper and more expressive than before. Based on the enhanced feature pyramid, the detection head including three branches is adopted to predict the classification score for each point on the feature pyramid, regress the distances from the point to the four sides of a bounding box, and predict the center-ness score which is multiplied by the classification score to rank the bounding box in NMS.

3. RESULTS AND DISCUSSION

3.1 Dataset and Metrics

This paper uses CityPersons dataset [22] for evaluating the proposed model. CityPersons dataset was built upon the Cityscapes dataset [23], which was recorded across 18 different cities in Germany with 3 different seasons and various weather conditions. The dataset includes 5000 images (2975 for training, 500 for validation, and 1525 for testing) with approximately 35000 manually annotated persons plus 13000 ignore region annotations. Both the bounding boxes and visible parts of pedestrians are provided and there are approximately 7 pedestrians in average per image. Importantly, it includes a large number of objects in crowded scenes. The proposed model is trained on this training subset and evaluated on the validation subset. For evaluation metrics, this paper reports performance using standard average-log miss rate (MR) in experiments. It is computed over the false positive per image (FPPI) range of $[10^{-2}, 10^{0}]$ [6]. This paper also selects MR^{-2} and its lower value reflects better detection performance. All experiments are conducted based on mmdetection [24], with RTX 3070 GPU for training. The backbone network is pretrained on ImageNet [9] and all added layers are randomly initialized with the xavier method. The network is totally trained for 240k iterations, with the initial learning rate of 0.0001 and decreased by a factor of 10 after 160k iterations.

3.2 Detection Results

This paper compares the proposed method with state-of-the-art detectors, including Adapted Faster RCNN [22], RepLoss [25], and OR-CNN [26] on both the validation and testing sets of the CityPersons dataset. The results are shown in Table 1 and Table 2. Detection results on the original image size and up-sampled image are compared. It should be noted that it is a common practice to up-sample the image to achieve a better detection accuracy, but with the cost of more computational expense. This paper only tests the proposed method on the original image size as detecting objects in crowded scenes is more critical on both accuracy and efficiency. As shown in Table 1, the proposed model achieves the best detection results on the validation set of the CityPersons dataset. To be more specific, the proposed model reduces MR^{-2} by 4.6%, 2.4%, and 2.0% compared with Adapted Faster-RCNN, RepLoss, and OR-CNN, respectively on the reasonable subset at scale ×1. The results demonstrate the superiority of the proposed method in locating pedestrian in crowded scenes. On the testing set, the proposed method achieves the best performance, with an improvement of 1.82%, 0.33%, 0.17% compared with Adapted Faster-RCNN, RepLoss, and OR-CNN, respectively. The results demonstrate the ability of the proposed method to handle occlusion issues in crowded scenes.

To demonstrate the effectiveness of the proposed method under various occlusion levels, this paper follows the strategy in [25] [26] to divide the reasonable subset in the validation set (occlusion < 35%) into three subsets: partial subset with 10% < occlusion ≤ 35%, bare subset with occlusion ≤ 10%, and heavy subset with the occlusion ratio larger than 35%. The results on these three subsets are reported in Table 1. As shown in Table 1, The proposed method outperforms the state-of-the-art methods consistently across all three subsets, i.e., reduces 5.1% MR^{-2} on bare subset, 1.7% MR^{-2} on partial

Table 1: Detection Results on The CityPersons Validation Set.

Scale	Model	Backbone	Reasonable	Heavy	Partial	Bare
×1	Adapted Faster-RCNN	VGG-16	15.4	-	-	-
	RepLoss	ResNet-50	13.2	56.9	16.8	7.6
	OR-CNN	VGG-16	12.8	55.7	15.3	6.7
	Proposed model	ResNest	**10.8**	**50.6**	**13.6**	**5.5**
×1.3	Adapted Faster-RCNN	VGG-16	12.8	-	-	-
	RepLoss	ResNet-50	11.6	55.3	14.8	7.0
	OR-CNN	VGG-16	11.0	51.3	13.7	5.9

Table 2: Detection Results on The CityPersons Testing Set.

Model	Backbone	Scale	Reasonable
Adapted Faster-RCNN	VGG-16	×1.3	12.97
RepLoss	ResNet-50	×1.3	11.48
OR-CNN	VGG-16	×1.3	11.32
Proposed model	ResNest	×1	11.15

subset, and 1.2% MR^{-2} on heavy subset. Figure 5 shows some examples of detection results of the proposed method on the CityPersons testing set.

4. CONCLUSIONS

This paper proposes a new framework for locating objects in crowded scenes based on fully convolutional network. In the proposed framework. ResNest structure is used as the backbone network to extract features from input image. The feature enhancement model is designed to refine the features generated by the backbone efficiently. The detection head including three branches is adopted to predict the classification score for each point on the feature pyramid, regress the distances from the point to the four sides of a bounding box, and predict the center-ness score which is multiplied by the classification score to rank the bounding box in NMS. Experimental results on the CityPersons dataset show that the proposed model achieves the best detection results compared with state-of-the-art detectors.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 5: Visualization of Detection Results of The Proposed Method on The CityPersons Testing Set.

REFERENCES:

[1] Girshick, Ross. "Fast r-cnn." In *Proceedings of the IEEE international conference on computer vision*, pp. 1440-1448. 2015.

[2] Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: towards real-time object detection with region proposal networks." *IEEE transactions on pattern analysis and machine intelligence* 39, no. 6 (2016): 1137-1149.

[3] Zhang, Liliang, Liang Lin, Xiaodan Liang, and Kaiming He. "Is faster R-CNN doing well for pedestrian detection?." In *European conference on computer vision*, pp. 443-457. Springer, Cham, 2016.

[4] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single shot multibox detector." In *European conference on computer vision*, pp. 21-37. Springer, Cham, 2016.

[5] Cai, Zhaowei, and Nuno Vasconcelos. "Cascade r-cnn: Delving into high quality object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 6154-6162. 2018.

[6] Zhang, Shifeng, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z. Li. "Single-shot refinement neural network for object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4203-4212. 2018.

[7] Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You only look once: Unified, real-time object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779-788. 2016.

[8] Redmon, Joseph, and Ali Farhadi. "YOLO9000: better, faster, stronger." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263-7271. 2017.

[9] Law, Hei, and Jia Deng. "Cornernet: Detecting objects as paired keypoints." In *Proceedings of the European conference on computer vision (ECCV)*, pp. 734-750. 2018.

[10] Duan, Kaiwen, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. "Centernet: Keypoint triplets for object detection." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 6569-6578. 2019.

[11] Zhou, Xingyi, Jiacheng Zhuo, and Philipp Krahenbuhl. "Bottom-up object detection by grouping extreme and center points." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 850-859. 2019.

[12] Yang, Ze, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. "Reppoints: Point set representation for object detection." In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9657-9666. 2019.

[13] Huang, Lichao, Yi Yang, Yafeng Deng, and Yinan Yu. "Densebox: Unifying landmark localization with end to end object detection." *arXiv preprint arXiv:1509.04874* (2015).

[14] Zhu, Chenchen, Yihui He, and Marios Savvides. "Feature selective anchor-free module for single-shot object detection." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 840-849. 2019.

[15] Kong, Tao, Fuchun Sun, Huaping Liu, Yuning Jiang, Lei Li, and Jianbo Shi. "Foveabox: Beyound anchor-based object detection." *IEEE Transactions on Image Processing* 29 (2020): 7389-7398.

[16] Tian, Zhi, Chunhua Shen, Hao Chen, and Tong He. "Fcos: Fully convolutional one-stage object detection." In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9627-9636. 2019.

[17] Lin, Tsung-Yi, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. "Feature pyramid networks for object detection." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117-2125. 2017.

[18] Zhang, Hang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun et al. "Resnest: Split-attention networks." *arXiv preprint arXiv:2004.08955* (2020).

[19] He, Tong, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. "Bag of tricks for image classification with convolutional neural networks." In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 558-567. 2019.

[20] Wang, Wenhai, Enze Xie, Xiang Li, Xuebo Liu, Ding Liang, Yang Zhibo, Tong Lu, and

Chunhua Shen. "PAN++: Towards Efficient and Accurate End-to-End Spotting of Arbitrarily-Shaped Text." *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021).

[21] Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).

[22] Zhang, Shanshan, Rodrigo Benenson, and Bernt Schiele. "Citypersons: A diverse dataset for pedestrian detection." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3213-3221. 2017.

[23] Cordts, Marius, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. "The cityscapes dataset for semantic urban scene understanding." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213-3223. 2016.

[24] Chen, Kai, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun et al. "MMDetection: Open mmlab detection toolbox and benchmark." *arXiv preprint arXiv:1906.07155* (2019).

[25] Wang, Xinlong, Tete Xiao, Yuning Jiang, Shuai Shao, Jian Sun, and Chunhua Shen. "Repulsion loss: Detecting pedestrians in a crowd." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7774-7783. 2018.

[26] Zhang, Shifeng, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z. Li. "Occlusion-aware R-CNN: Detecting pedestrians in a crowd." In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 637-653. 2018.