Efficient Multi-Receptive Pooling YOLOv5 with Coordinate Attention Module for Object Detection on Drone

Jinsu An
Department of Electrical, Electronic and Computer Engineering
University of Ulsan
Ulsan, Korea
jinsu5023@islab.ulsan.ac.kr

Muhamad Dwisnanto Putro
Department of Electrical Engineering
Universitas Sam Ratulangi
Manado, Indonesia
dwisnantoputro@unsrat.ac.id

Adri Priadana
Department of Electrical, Electronic and Computer Engineering
University of Ulsan
Ulsan, Korea
priadana3202@mail.ulsan.ac.kr

Junmyeong Kim
Department of Electrical, Electronic and Computer Engineering
University of Ulsan
Ulsan, Korea
kjm7029@islab.ulsan.ac.kr

Kang-Hyun Jo
Department of Electrical, Electronic and Computer Engineering
University of Ulsan
Ulsan, Korea
acejo@ulsan.ac.kr

***Abstract*—Object detection is the most basic and significant research in computer vision in images, and it is a study to discriminate the position and class of an object. This operation has been continuously researched for the past few years. Object detection performance based on accuracy is gradually improving due to the recent development of hardware such as GPU computing power and cameras. Object detection operations grafted in drones can be implemented in many domains. To perform object detection algorithms in real-time in drones, the applied network must be lightweight. For an algorithm capable of real-time operation on low-cost devices, this paper proposes Efficient Multi-Receptive Pooling YOLOv5 with Coordinate(CAM). Efficient Residual Bottleneck and Efficient Multi-Receptive Pooling make the model lighter by reducing the number of parameters, and the CAM improves the object detection rate of the model. The model is trained using the VisDrone dataset, and the mAP value increased by about 19% to 20.6 mAP, and the number of parameters decreased by about 6% to 1,663,599.**

***Index Terms*—Object Detection, Drone Vision, Convolutional Neural Network (CNN), Efficient Module, Attention Modules**

I. Introduction

By dint of their capacity to carry out increasingly complicated tasks like monitoring and surveillance operations, drone usage and interest have surged in recent years. Drones can complete various complex tasks autonomously due to technological support, such as artificial intelligence and computer vision. Many vision drone tasks, including object detection and identification, can be performed and provide outstanding performance. It attracts the attention of many researchers and practitioners to continue to develop this technology as a solution in various fields such as the mining industry [1], factory [2], transportation [3], etc.

The tremendous success of deep learning has led to the development of numerous outstanding object detection techniques in recent years. You Only Look Once (YOLO) [4], especially the fifth version (YOLOv5) [5], which applies a single-stage detection mechanism, is one of the most popular architectures for object detection. Compared to other techniques that implement two-stage detection, such as Faster R-CNN [6], YOLO produces a slightly inferior accuracy but can operate at a higher detection speed as a result of performing the localization and classification operation in the one stage. Therefore, this technique is perfect for a vision drone technology that needs fast detection mode.

Lately, several works have concentrated on improving a YOLOv5 architecture on the VisDrone dataset to increase its performance or make it more efficient. Pruned-YOLO [7] improved YOLOv5 architecture by applying an iterative channel pruning mechanism. It achieves a sufficient balance between accuracy and efficiency. Zhan et al. [8] redesigned the YOLOv5 anchor size, reduced the feature dimension, and utilized squeeze-and-excitation (SE) as an attention module. This mechanism can effectively increase the detection speed and improve the detection precision.

Another work [9] escalated YOLOv5 architecture by employing Strip Bottleneck (SPB) block to create an efficient detector called SPB-YOLO. It gains a satisfactory trade-off between accuracy and speed. Kim et al. [10] proposed an enhancement of YOLOv5 with an efficient channel attention pyramid module called ECAP-YOLO. This module is used to bargain the small object issues in the VisDrone dataset.

In this work, we propose Efficient Multi-Receptive Pooling YOLOv5 with CAM. The main contributions of this work are

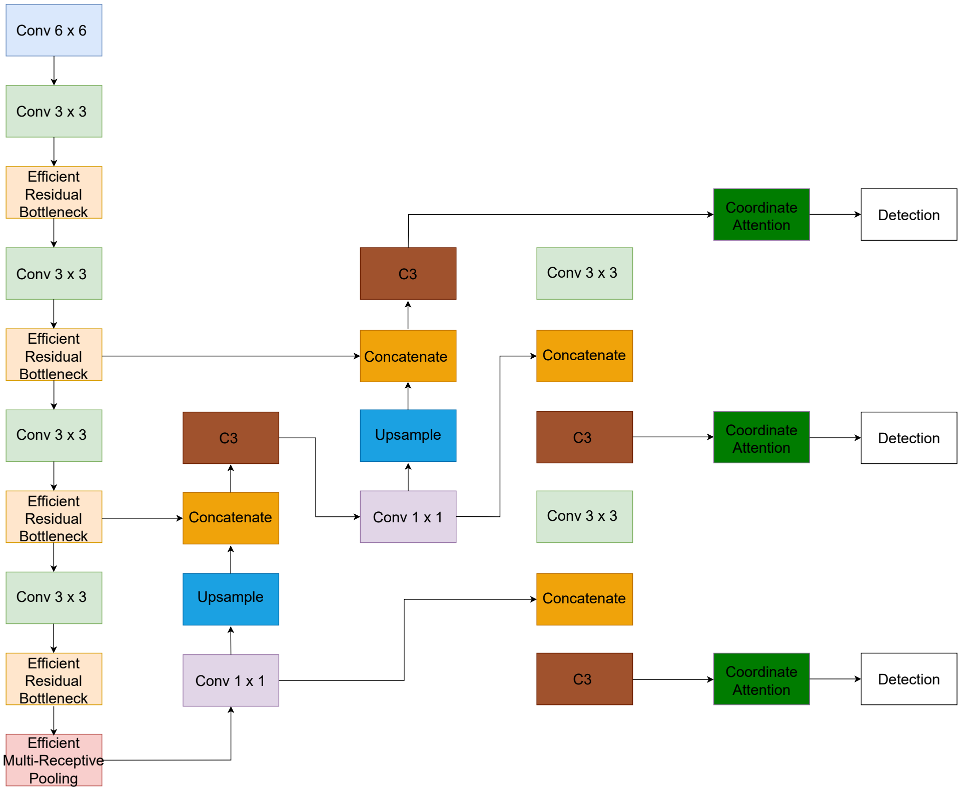

Fig. 1. The proposed architecture. A backbone module is used to extract object features with the proposed efficient methods. Besides, the PANet(Neck) and detection(Head) modules help the detector identify the location of the object in multi-scale variants.

summarized as follows:

1) A real-time object detection method is proposed for quickly locating an object that can operate on a low-cost device.
2) A structure of combination block is introduced by applying the Coordinate Attention Module to the original YOLOv5 network.

II. Proposed Architecture

The proposed architecture has three main modules as shown in the proposed Architecture Fig. 1. The first Efficient Residual Bottleneck (ERB) [11] and the second Efficient Multi-Receptive Pooling (EMRP) [11] are used in the backbone of YOLOv5, which corresponds to the baseline. The third Coordinate Attention Module (CAM) [12] is applied to the Path Aggregation Network (PANet) corresponding to the Neck part. CAM is applied before each detector, which is part from the Neck to the Head.

A. The Backbone

YOLOv5's framework has three main components: It consists of Backbone, Head, and Neck. The Backbone extracts the features of the image and transfers them to the Neck through the Head. Neck creates a feature pyramid by collecting feature maps extracted from Backbone. Finally, it is composed of an output layer that detects objects in the Head. CSPDarknet53 is used as the Backbone, CSPDarknet is a lightweight network structure based on the Darknet53 structure. In the first convolution layer, the feature map is divided into two paths to deliver information in a balanced way, and various types of operations are performed in each path. The structure of CSPDarknet is largely divided into two paths. One is the same path as the existing Darknet53 structure, and the other is a path through a deeper network. These two paths are concatenated in the last layer and merged into a single feature map. This structure contributes to achieving both weight reduction and performance improvement of the model. Therefore, in YOLOv5 using CSPDarknet, the CSPDarknet structure serves as the Backbone. PANet is used for the Neck, and $B \times (5 + C)$ output layer is used for the Head. B is the number of bounding boxes, and C is the class score. Among them, the $C3$ layer of CSPDarknet53 used in the Backbone is improved to lighten the deep learning object detection model. CAM was applied before detection by adding a CAM layer to each end of the PANet connected from the Neck to the Head.

B. The Efficient Residual Bottleneck

Efficient Residual Bottleneck (ERB) as shown in Fig. 2 is an improved layer of the C3 layer used in YOLOv5. The C3 layer is the CSP bottleneck with 3 convolutions and consists of a bottleneck and 3 convolutional layers. In order to object detection algorithms in real-time on drones using low-cost devices, the number of parameters in deep learning object detection networks needs to be reduced. To reduce the number of parameters, we adjusted the convolution of the C3 layer from 3 to 2 and changed the order of concatenation and addition of feature maps. The proposed network provides an improved backbone for extracting object features and distinguishing essential elements from the background. Apply a series of convolutional layers sequentially using efficient modules. The light block applies residual techniques to maintain the quality of feature maps, resulting in high performance in final predictions. To avoid gradient degradation and avoid saturation of the training process, each convolution operation sequentially uses SiLU activations and batch normalization.

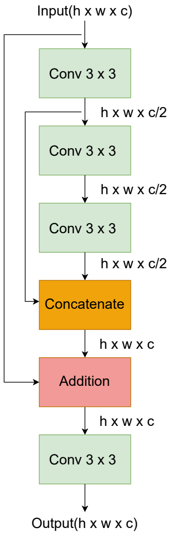

Fig. 2. Efficient Residual Bottleneck.

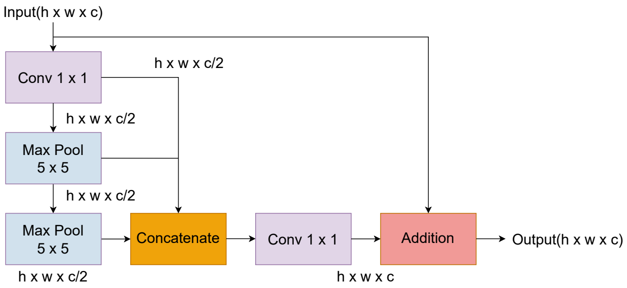

Fig. 3. Efficient Multi-Receptive Pooling, less complexity by double receptive pooling & addition pathways.

C. Efficient Multi-Receptive Pooling

We introduce efficient multi-receptive pooling as shown in Fig. 3, improved in SPPF to capture the difference of spatial information using simple convolution and cascade pooling. Convolutional and Two Sequential Pooling are applied to provide various receptive areas. It can increase feature selection options in multi-perspective combinations and use simple convolution to obtain a single spatial domain. Two pools with a window size of 5×5 are used sequentially to capture the maximum of the features. Combining features from different receptive domains will increases the diversity of information, allowing the network to know more about the types of features. Convolution operations are then applied to blend the various pieces of information. Residual techniques are used in this module to ensure that different feature pooling results achieve the expected quality and reduce the error rate of the filtering process.

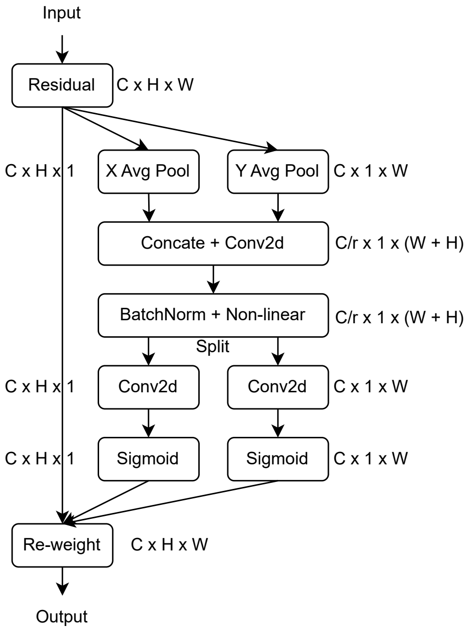

Fig. 4. Coordinate Attention Module.

D. Coordinate Attention Module

In the YOLOv5 architecture, an attention module is needed as a magnifier before delivering the feature map to the detection head. This module will make the network more focused on the specific area of the features map that contributes more to the detection result. This architecture utilizes a Coordinate Attention Module (CAM) [12] shown in Fig. 4, which involves two average pooling operations to aggregate features of each channel along the horizontal and vertical coordinates to generate a pair of 1D feature maps. The average pooling operation along the horizontal coordinate can be formulated as

$$z_c^h\left(h\right) = \frac{1}{W} \sum_{0 \leq i \leq W} x_c\left(h, i\right), \tag{1}$$

where x_c is the c-th channel of input and z_c^h is the average pooling operation result at the height h. Likewise, the average pooling operation along the vertical coordinate can be formulated as

$$z_c^w\left(w\right) = \frac{1}{H} \sum_{0 \leq i \leq H} x_c\left(j, w\right), \tag{2}$$

where z_c^w is the average pooling operation result at the width w. This pair of operations enables the network to record long-range dependencies along one spatial direction and keep exact location information along one another spatial direction. It will allow the networks better precisely locate the objects of interest. To effectively capture inter-channel relationships, the two average pooling operation results, z^h and z^w, concatenation operation followed by a shared 1×1 convolutional operation function F_1 are applied to them illustrated as

$$\mathbf{f} = \delta(F_1(\mathbf{z}^h \oplus \mathbf{z}^w)), \tag{3}$$

where δ is a non-linear activation function and $\oplus$ is a concatenation operation along the spatial dimension. $\mathbf{f} \in \mathbb{R}^{C/r \times (H+W)}$ is the output feature map where r is the reduction ratio used to regulate the block size. Further, this output $\mathbf{f}$ is split into two separate parts $\mathbf{f}^h \in \mathbb{R}^{C/r \times H}$ and $\mathbf{f}^w \in \mathbb{R}^{C/r \times W}$. A 1×1 convolutional operation is applied to each part to individually transform $\mathbf{f}^h$ and $\mathbf{f}^w$ followed by sigmoid activation function which can be formulated as

$$\mathbf{g}^h = \sigma(F_h(\mathbf{f}^h)), \tag{4}$$

$$\mathbf{g}^w = \sigma(F_w(\mathbf{f}^w)), \tag{5}$$

where σ is a sigmoid activation function. This process will generate attention weights. Finally, The outputs $\mathbf{g}^h$ and $\mathbf{g}^w$ are then expanded by applying a channel-wise broadcast multiplication operation illustrated as

$$y_c\,(i,j) = x_c\,(i,j) \times g_c^h\,(i) \times g_c^w\,(j)\,. \tag{6}$$

E. Loss Function

The loss function of YOLOv5 is used to improve the model's prediction during training by calculating the difference between the bounding box predicted by the model and the ground truth bounding box. The loss function of YOLOv5 consists of three types: Localization loss, Confidence loss, and Class loss. Localization loss calculates the difference between the position of the bounding box predicted by the model and the ground truth bounding box. This loss function uses Mean Square Error (MSE) as a loss function for the coordinates and size of the center of the predicted bounding box. Confidence loss calculates the Intersection over Union (IoU) difference between predicted bounding boxes. It is calculated as a binary cross-entropy loss function between the confidence of the predicted bounding box and the confidence of the ground truth. Class loss computes the difference between the object class predicted by the model and the ground truth class. This loss function is calculated as a multi-class cross-entropy loss function. The three loss functions are combined to finally calculate the loss of the model's prediction result. Train the model to minimize this value.

$$\begin{aligned} L_{MB} = \lambda_{coord} \sum_{g=1}^{G^2} \sum_{a=1}^{A} 1_{ga}^{obj} L_{coord} + \\ \lambda_{obj} \sum_{g=1}^{G^2} \sum_{a=1}^{A} 1_{ga}^{obj} L_{obj} + \\ \lambda_{cls} \sum_{g=1}^{G^2} \sum_{a=1}^{A} 1_{ga}^{obj} L_{cls} \end{aligned} \tag{7}$$

III. Implementation Setup

In this session, we describe the experiments of YOLOv5 network with Coordinate Attention Module on VisDrone dataset. As an experimental environment, the model is implemented using PyTorch in a Linux environment. When training the deep learning model, training was conducted using Intel Xeon Gold CPU and Nvidia Tesla A100 40GB GPU.

IV. Experimental Results

A. Evaluation on Datasets

The VisDrone dataset is a large-scale object detection and tracking dataset based on high-resolution video images captured by multiple cameras mounted on drones. This dataset contains video images taken in various environments, mainly in cities, coastal areas, agricultural lands, and mountainous areas. There are a total of 10 classes (pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, motor), and it consists of 288 (261,908 images) video clips and 10,209 static photos. The dataset contains video images in a variety of conditions, including day and night, sunny and cloudy, and supports up to 1080p. The VisDrone 2019 dataset can be used to solve various computer vision problems such as object detection, tracking, and velocity estimation. Since it contains videos taken on a large scale, in high resolution, and under various conditions, it can be usefully used for the development and performance evaluation of object detection and tracking algorithms. The proposed method tested the object detection performance on the VisDrone dataset. The VisDrone dataset contains many objects of tiny size. In order to detect small-sized objects, a high-resolution image is required or a method capable of extracting features of the object well is required. An object detection model is evaluated through a dataset by extracting and learning the features of various objects included in the dataset. To evaluate the model, we use Average Precision (AP) to measure the accuracy of the predicted bounding box, derive AP for each class, and finally calculate the mean Average Precision (mAP) value for all classes. As a result, the proposed method shows 20.6mAP with about 19% higher mAP compared to the original YOLOv5n, and the number of parameters is 1,663,599, which is about 6% less.

B. Runtime Efficiency

In this paper, Efficient Residual Bottleneck (ERB) and Efficient Multi-Receptive Pooling (EMRP) are applied to YOLOv5 to make the network more efficient. ERB and EMRP are created by improving C3 and SPPF layer, which correspond to the Backbone of YOLOv5, and through this method, the number of parameters of the network could be effectively reduced. In addition, when the Coordinate Attention Module is applied, the parameters of the network increase, but the increase in parameters is prevented by minimizing the number of repetitions of ERB performed in Backbone. As a result, it is possible to reduce the number of parameters and improve performance through the combination of ERB, EMRP, and CA. Compared with the original YOLOv5s, the number of parameters is about 76% less, the GFLOPs are quarter, and

TABLE I
DETECTION RESULT COMPARISONS ON VISDRONE DATASET

Model	AP	AP50	Backbone
Cascade R-CNN++ [13]	18.33	33.5	SERexNeXt-50
EnDet	17.81	37.27	ResNet101-fpn
DCRCNN [14]	17.79	42.03	ResNeXt-101
Cascade R-CNN+ [13]	17.67	34.89	ResNeXt-101
ODAC	17.42	40.55	VGG
DA-RetianNet [15]	17.05	35.93	ResNet101
MOD-RETINANET [16]	16.96	33.77	ResNet50
DBCL [17]	16.78	31.08	Hourglass-104
ConstraintNet [18]	16.09	30.72	Hourglass-104
CornetNet* [19]	17.41	34.12	Hourglass-104
Light-RCNN* [20]	16.53	32.78	ResNet101
FPN* [21]	16.51	32.2	ResNet50
Cascade R-CNN* [22]	16.09	31.91	ResNeXt-101
DetNet59* [23]	15.26	29.23	ResNet50
RefineDet* [24]	14.9	28.76	ResNet101
RetinaNet* [16]	11.81	21.37	ResNet101
YOLOv5n	**17.3**	**31.4**	**Improved CSPDarknet53**
YOLOv5n w ERB [11]	**17.0**	**31.7**	**Improved CSPDarknet53**
YOLOv5n w EMRP [11]	**17.3**	**31.5**	**Improved CSPDarknet53**
YOLOv5n w CAM	**15.4**	**28.0**	**Improved CSPDarknet53**
YOLOv5n w ERB&CAM	**18.3**	**32.8**	**Improved CSPDarknet53**
YOLOv5n w EMRP&CAM	**20.6**	**34.9**	**Improved CSPDarknet53**

Model	# parameters	GFLOPs	AP
YOLOv5s	7,046,599	15.9	20.1
YOLOv5s w ERB	6,871,559	15.5	19.5
YOLOv5s w EMRP	6,915,527	15.8	19.3
YOLOv5n	1,777,477	4.2	17.3
YOLOv5n w ERB	1,733,447	4.1	17
YOLOv5n w EMRP	1,744,679	4.2	17.3
YOLOv5n w CAM	1,696,367	3.9	15.4
YOLOv5n w ERB&CAM	1,652,367	3.8	18.3
YOLOv5n w EMRP&CAM	1,663,599	3.9	20.6

the performance is similar at 20.6mAP. Compared with the original YOLOv5n, the number of parameters is reduced by about 6%, GFLOPs are also faster at 3.9, and the performance is improved by about 19%.

V. CONCLUSION

This paper proposes a YOLOv5 network that enables real-time object detection. It shows higher performance by applying an Efficient Residual Bottleneck and The Coordinate Attention Module. The C3 layer is improved with an Efficient Residual Bottleneck to reduce the number of computations. The coordinate Attention Module is also applied to enhance the performance of object detection. In this work, the VisDrone dataset is used as a training set. The mAP value is 20.6mAP, about 19% higher than the original YOLOv5, and the number of parameters is 1,663,599, about 6% less.

Additional detectors will be employed to increase the object detection rate in future work. YOLOv5 uses three detectors, which detect large, medium, and small size objects. Objects in the VisDrone dataset have a lot of tiny size objects. Therefore, we plan to detect objects of tiny size using an additional detector. As the number of layers in the network increases, the number of parameters required for calculation increases. However, it is expected that the proposed method can be applied to reduce the number of parameters and increase the object detection rate by using additional detectors.

VI. ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the government(MSIT).(No.2020R1A2C2008972)

REFERENCES

[1] J. Shahmoradi, E. Talebi, P. Roghanchi, and M. Hassanalian, "A comprehensive review of applications of drone technology in the mining industry," *Drones*, vol. 4, no. 3, p. 34, 2020.

[2] O. Maghazei, T. H. Netland, D. Frauenberger, and T. Thalmann, "Automatic drones for factory inspection: The role of virtual simulation," in *Advances in Production Management Systems. Artificial Intelligence for Sustainable and Resilient Production Systems: IFIP WG 5.7 International Conference, APMS 2021, Nantes, France, September 5–9, 2021, Proceedings, Part IV*. Springer, 2021, pp. 457–464.

[3] Y. Lee, Q. Tang, J. Choi, and K. Jo, "Low computational vehicle re-identification for unlabeled drone flight images," in *IECON 2022–48th Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2022, pp. 1–6.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2016, pp. 779–788.

[5] G. Jocher, A. Stoken, and J. Borovec, "ultralytics/yolov5: v3.0." [Online]. Available: https://doi.org/10.5281/zenodo.3983579

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2016.

[7] J. Zhang, P. Wang, Z. Zhao, and F. Su, "Pruned-yolo: Learning efficient object detector using model pruning," in *International Conference on Artificial Neural Networks*. Springer, 2021, pp. 34–45.

[8] W. Zhan, C. Sun, M. Wang, J. She, Y. Zhang, Z. Zhang, and Y. Sun, "An improved yolov5 real-time detection method for small objects captured by uav," *Soft Computing*, vol. 26, pp. 361–373, 2022.

[9] X. Wang, W. Li, W. Guo, and K. Cao, "Spb-yolo: An efficient real-time detector for unmanned aerial vehicle images," in *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*. IEEE, 2021, pp. 099–104.

[10] M. Kim, J. Jeong, and S. Kim, "Ecap-yolo: Efficient channel attention pyramid yolo for small object detection in aerial image," *Remote Sensing*, vol. 13, no. 23, p. 4851, 2021.

[11] J. An, M. D. Putro, and K.-H. Jo, "Efficient residual bottleneck for object detection on cpu," in *2022 International Workshop on Intelligent Systems (IWIS)*, 2022, pp. 1–4.

[12] Q. Hou, D. Zhou, and J. Feng, "Coordinate attention for efficient mobile network design," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 13 713–13 722.

[13] Z. Cai and N. Vasconcelos, "Cascade r-cnn: High quality object detection and instance segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 5, pp. 1483–1498, 2021.

[14] S. Chakraborty, S. Aich, A. Kumar, S. Sarkar, J.-S. Sim, and H.-C. Kim, "Detection of cancerous tissue in histopathological images using dual-channel residual convolutional neural networks (dcrcnn)," in *2020 22nd International Conference on Advanced Communication Technology (ICACT)*, 2020, pp. 197–202.

[15] G. Pasqualino, A. Furnari, G. Signorello, and G. M. Farinella, "An unsupervised domain adaptation scheme for single-stage artwork recognition in cultural sites," *Image and Vision Computing*, p. 104098, 2021.

[16] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *CoRR*, vol. abs/1708.02002, 2017. [Online]. Available: http://arxiv.org/abs/1708.02002

[17] Y. Wu, Z. Cheng, Z. Xu, and W. Wang, "Segmentation is all you need," *CoRR*, vol. abs/1904.13300, 2019. [Online]. Available: http://arxiv.org/abs/1904.13300

[18] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *CoRR*, vol. abs/1904.07850, 2019. [Online]. Available: http://arxiv.org/abs/1904.07850

[19] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," *CoRR*, vol. abs/1808.01244, 2018. [Online]. Available: http://arxiv.org/abs/1808.01244

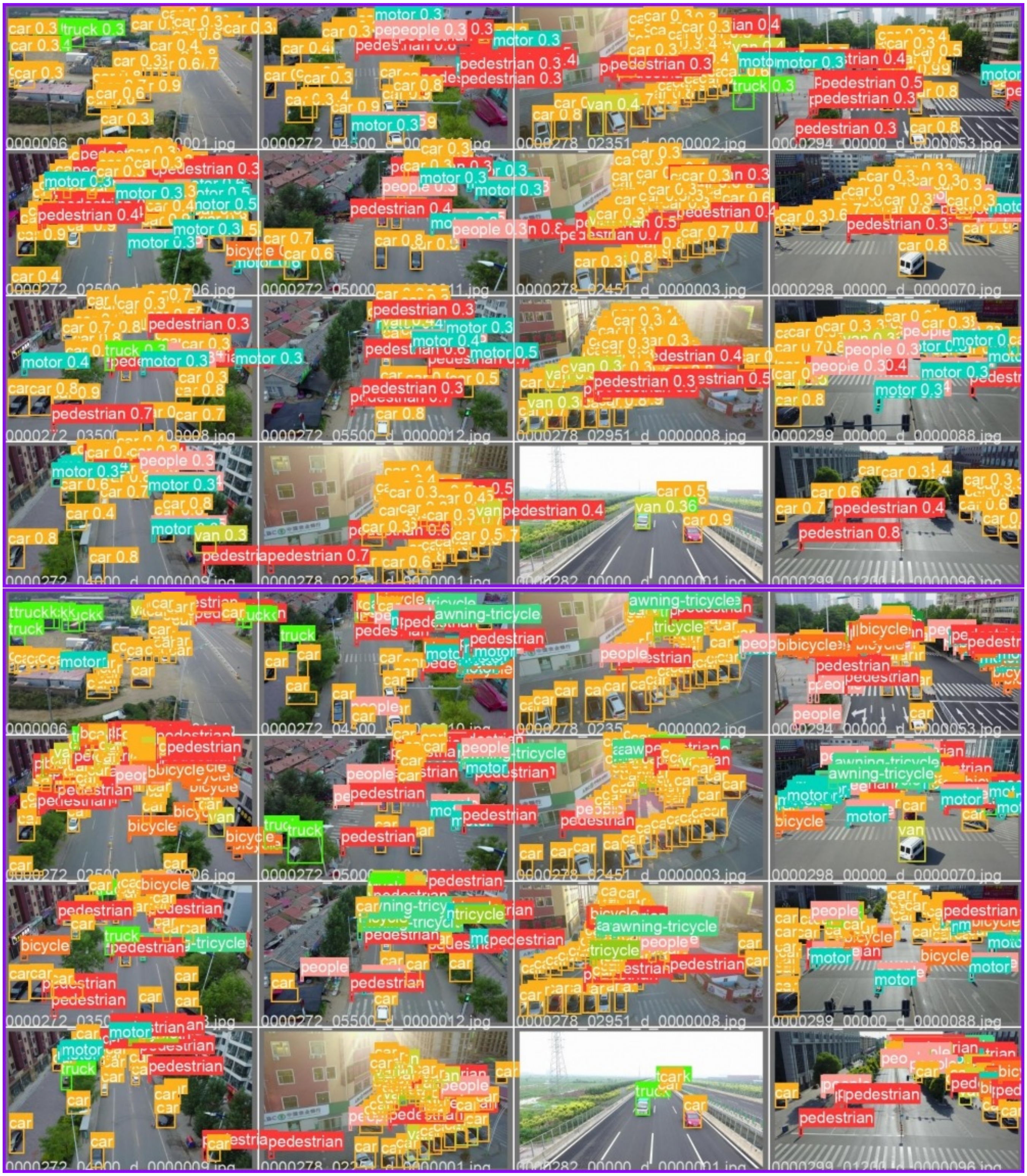
0000006_0...001.jpg
0000272_04500_...
0000278_02351_d_0000002.jpg
0000294_00000_d_0000053.jpg
0000272_02500_...06.jpg
0000272_05000_...11.jpg
0000278_02451_d_0000003.jpg
0000298_00000_d_0000070.jpg
0000272_03500_...008.jpg
0000272_05500_d_0000012.jpg
0000278_02951_d_0000008.jpg
0000299_00000_d_0000088.jpg
0000272_04000_d_0000009.jpg
0000278_0225..._0000001.jpg
0000282_00000_d_0000001.jpg
0000299_01200_d_0000096.jpg

Fig. 5. Visualization of Prediction and Ground-truth Result on VisDrone Dataset

[20] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Light-head R-CNN: in defense of two-stage object detector," *CoRR*, vol. abs/1711.07264, 2017. [Online]. Available: http://arxiv.org/abs/1711.07264

[21] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," *CoRR*, vol. abs/1612.03144, 2016. [Online]. Available: http://arxiv.org/abs/1612.03144

[22] Z. Cai and N. Vasconcelos, "Cascade R-CNN: delving into high quality object detection," *CoRR*, vol. abs/1712.00726, 2017. [Online].

Available: http://arxiv.org/abs/1712.00726

[23] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun, "Detnet: A backbone network for object detection," *CoRR*, vol. abs/1804.06215, 2018. [Online]. Available: http://arxiv.org/abs/1804.06215

[24] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li, "Single-shot refinement neural network for object detection," *CoRR*, vol. abs/1711.06897, 2017. [Online]. Available: http://arxiv.org/abs/1711.06897