Improved YOLOv5 Network with CBAM for Object Detection Vision 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
Sam Ratulangi university
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

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

***Abstract*—Recently, drones have been applied and used in various tasks in many fields. Due to the development of camera technology and hardware, it is possible to detect objects using deep learning technology in real time. Most of the object sizes of videos and images taken by drones are very small objects. The dataset that collects these image data is the VisDrone dataset. As deep learning technology develops a lot, object detection accuracy is getting higher and higher. However, it is still hard to perform object detection in real-time and show high object detection accuracy. In this paper, we combine YOLOv5 and CBAM to increase object detection accuracy by focusing more on the features required for object detection. The model is trained using the VisDrone dataset, and the mAP value is measured at 22.56mAP, which is 2.45mAP higher than the original YOLOv5.**

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

I. Introduction

Recent developments in drone technology have been advancing quickly and enabled to support of various intelligent systems, such as autonomous video monitoring, surveying, and mapping [1]. Computer vision based on artificial intelligence becomes the main essence behind the success of this technology. Numerous vision drone works can be performed, such as object detection and classification, which results in an excellent performance. This fact drives researchers and practitioners to develop and implement this technology in various fields such as military [2], transportation [3], agriculture [4], mining industry [5], etc.

With the massive success of deep learning, many novel object detection techniques have been proposed in the last few years. Commonly, these techniques can be divided into two categories, two-stage and single-stage mechanisms. The single-stage detection mechanism, such as You Only Look Once (YOLO) [6], considers object detection as a regression case. It conducts the localization and classification in the same stage. Meanwhile, the two-stage detection mechanism, such as Faster R-CNN [7] proposed to perfect the previous version, R-CNN [8] and Fast R-CNN [9], first extracts the regions of interest (ROI) from the input images. Then, it performs bounding box regression and classification within these ROIs. As a result, the single-stage detection mechanism has a little lower accuracy but a higher detection speed than the two-stage detection mechanism. Therefore, a detector using a single-stage method, such as YOLO, with a higher speed, is very suitable to be applied in a vision drone which often moving.

YOLOv5 [10] is a relatively new YOLO version that delivers satisfactory performance. This detector utilizes the Feature Pyramid Network (FPN) [11] technique to incorporate features with diverse levels. This technique drives the system to detect an object of various sizes. Several researchers attempted to implement and upgrade the YOLOv5 by proposing new blocks [12] or employing attention modules, such as Convolutional Block Attention Module (CBAM) [13], to enhance detection performance. Zhu et al. [14] used CBAM to highlight the information from the feature map and added it to YOLOv5 to detect boulders from planetary images. Wang et al. [15] employed CBAM to enhance YOLOv5 in detecting helmets worn by construction workers. This module can improve the characterization capability of target features.

Another work [16] tried to escalate YOLOv5 by using CBAM, combined with Bidirectional Feature Pyramid Network (BiFPN), to perform Synthetic Aperture Radar (SAR) ship detection. This module is located before the Spatial Pyramid Pooling (SPP) layer to improve the features extractor capability. Yang et al. [17] fused CBAM with YOLOv5 to conduct student in-class behavior detection. This module is put into the Neck module of YOLOv5, effectively extracting robust features. The works show that employing attention modules, such as CBAM, can improve detection performance.

In this work, we propose an improved YOLOv5s network with CBAM to increase the performance of object detection.

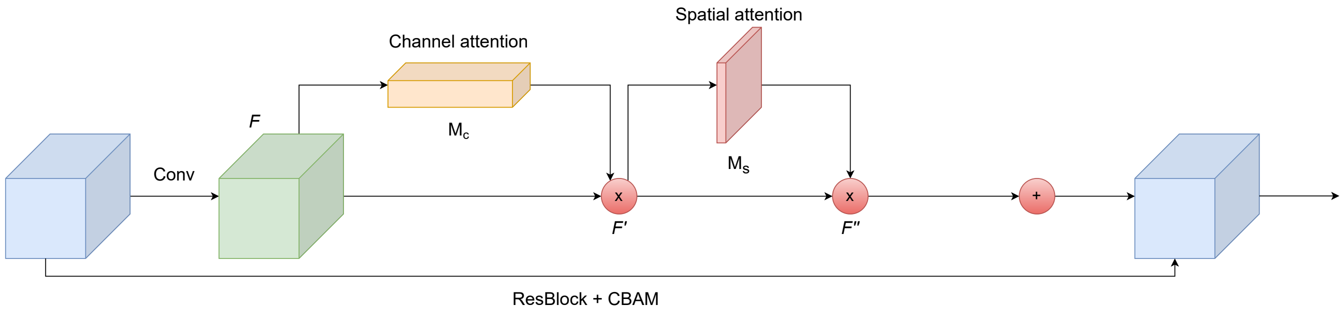

Fig. 1. CBAM Structure grafted with ResBlock

The main contributions of this work are summarized as follows:

1) A structure of the combination block is introduced by applying the CBAM to the original YOLOv5 network.
2) By applying CBAM that combines two attention modules, Channel Attention and Spatial Attention, an accurate result of 22% mAP can be obtained in drone tasks.

II. Proposed Architecture

The proposed architecture applied Convolutional Block Attention Module (CBAM) [13] to the Original YOLOv5 shown in the Fig 3. CBAM is a follow-up study of the BAM module, which maximizes the performance of the attention module. Since CBAM is a light and general module, it can be freely attached to any CNN architecture, and the entire model can be trained end-to-end.

A. The Backbone

YOLOv5 consists of three frameworks: Backbone, Neck, and Head. In the Backbone, the features of the image are extracted and transmitted to the Head through the Neck. The Neck collects the extracted features to create a feature pyramid, and the Head finally configures the output layer that detects the object. CSPDarknet53 is used as the Backbone, PANet is used for the Neck, and $B \times (5 + C)$ output layer is used for the Head. B is the $\#$ bounding boxes, and C is the class score. The object detection rate is improved by connecting CBAM to each last $C3$ layer from the Neck to the Head.

This framework refers to the study of [10]. It applies several convolution operations to distinguish the meaningful elements of the object sequentially. C3 (Three Convolution layers) is used several times in the different stages to generate the high and low-level frequency features. The C3 module utilizes the bottleneck technique efficiently by dividing the initial feature map by a simple convolution operation. The main part is processed at the residual bottleneck, while others are combined to enrich the information variety. This architecture is claimed to be light computational and parameter due to the convolution operation with compressed input channels. The entire network implements the C3 module in stages 2, 3, 4, and 5. These stages implement a different number of bottleneck operations on the Backbone, such as 1, 2, 3, and 1, respectively.

B. Channel Attention Module

Fig. 2 describes Channel Attention Module used in this work. The channel attention module is a step of encoding which channels to focus on, and the channel attention module has a slightly different squeeze process from SENet [18]. In SENet, the feature maps received from the previous convolutional block are encoded into $1 \times 1 \times C$ vectors through global average pooling. Meanwhile, in CBAM, values are obtained by global max-pooling and global average pooling as shown in Fig. 2, and encoded by pooling. Each of the two vectors is MLPed to apply nonlinearity. After being added, it is finally encoded as a randomized value through sigmoid. The final encoded value Mc is a value generated from the input feature map. It is a value expressed as a probability of which feature map is an important feature, considering among different C feature maps. Multiply Mc by the input feature map F to generate F'.

Intuitively, this attention module obtains stimuli from two different types of pooling to obtain a specific feature representation from the input channel extraction facility. Channel extraction applies a ratio to the middle part to suppress parameters while generating a diverse selection of channel representation features. The trained weights provide several judgment scores for the initial information employed to update the input features.

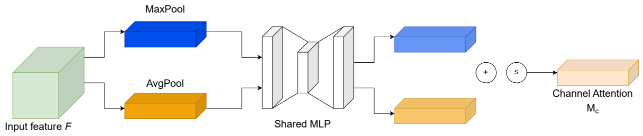

Fig. 2. Channel Attention Module

C. Spatial Attention Module

You can see the Spatial Attention Module in Fig. 4. The Spatial Attention Module is the stage of encoding which area to focus on the essential feature. After performing average pooling and max-pooling on the channel axis, they both are concatenated to create a feature map of $Hx \times W \times 2$. And, for spatial attention, Ms of $Hx \times W \times 1$ is generated by performing 7×7 Conv. Ms is multiplied by F' generated by the Channel Attention Module to create F''.

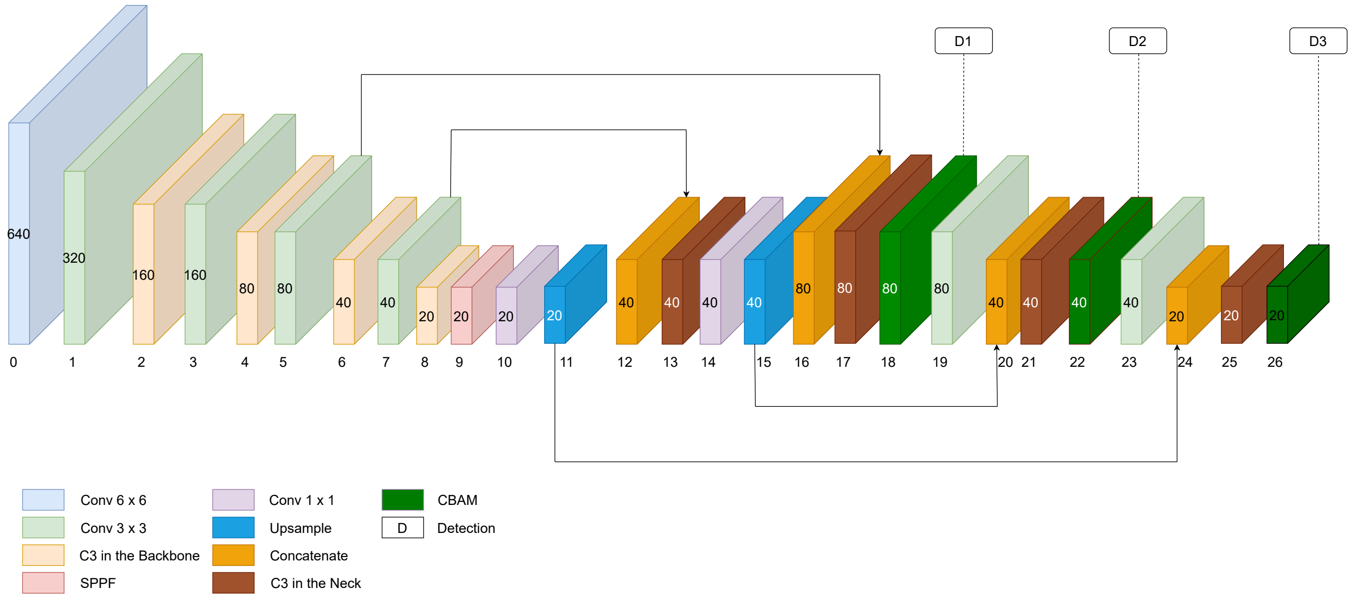

Fig. 3. Proposed Architecture. YOLOv5 small with CBAM

This attention module employs two pools to generate feature summaries in the spatial dimension. The representation variation helps the network to acquire different feature information, which is a process of increasing feature diversity. Two layers of spatial features are extracted using a weighted filter operation using a large kernel size to obtain a broad coverage of spatial area information. Subsequently, a single spatial feature is applied as an activation probability to generate a weighted mask which is used to update the input map.

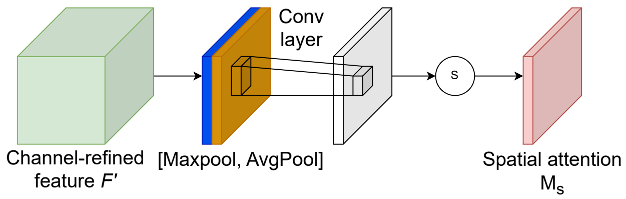

Fig. 4. Spatial Attention Module

D. Loss function

The loss function of YOLOv5 consists of the sum of three loss values: Class, Objectness, and Location loss. Class loss is the loss of ascertaining the class well. Objectness loss is the loss to confirm whether or not an existing object is in the grid. Location loss is the regression loss for finding well the center point (x,y), width, and height of the bounding box. Class loss and objectness loss are the addition part of a sigmoid layer to binary cross-entropy, which is used in multi-label classification problems. Class loss works to mitigate the difference in the number of data between classes, and objectness loss works to solve the imbalance of the presence/absence of an object. However, the objectness loss is added by placing a larger weight on the scale that finds smaller objects. Next, location loss uses CIoU loss, and IoU-based loss functions (GIoU, CIoU, DIoU, etc.), not MSE, which are usually used in regression, were used in the experiment. After finding the three losses, all are added, and this time, the weights are multiplied and added. Weight is a hyperparameter. It expresses as follows:

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

The proposed network uses λ_{coord}, λ_{obj}, and λ_{cls} are regression, object, and classification constant parameters by 0.05, 0.5, and 1, respectively. To obtain complete score loss, it calculates L_{coord} as a coordinate and size boxes loss, L_{obj} as an objectness loss, and L_{cls} as a classification loss separately with applying to the whole grid (G) and anchors (A). Additionally, it implements 1_{ga}^{obj} as an indicator to activate this function when there is an object of a prediction.

III. Implementation Details

This session describes experiments with the VisDrone dataset through the proposed architecture. As an experimental environment, the model is implemented using PyTorch in a Linux environment. When training the deep learning model, training is conducted using an Intel Xeon Gold CPU and Nvidia Tesla A100 40GB GPU.

TABLE I
DETECTION RESULT COMPARISONS ON VISDRONE DATASET

Model	**AP**	**AP50**	**Backbone**
TridentNet [19]	22.51	43.29	ResNet101
CenterNet-Hourglass [20]	22.36	41.76	Hourglass-104
retinaplus [21]	20.57	40.57	ResNeXt-101
ERCNNs [22]	20.45	41.2	ResNeXt-101
SAMFR-Cascade RCNN [23]	20.18	40.03	SERexNeXt-50
Cascade R-CNN++ [23]	18.33	33.5	SERexNeXt-50
EnDet	17.81	37.27	ResNet101-fpn
DCRCNN [24]	17.79	42.03	ResNeXt-101
Cascade R-CNN+ [23]	17.67	34.89	ResNeXt-101
ODAC	17.42	40.55	VGG
DA-RetianNet [25]	17.05	35.93	ResNet101
MOD-RETINANET [21]	16.96	33.77	ResNet50
DBCL [26]	16.78	31.08	Hourglass-104
ConstraintNet [20]	16.09	30.72	Hourglass-104
CornetNet* [27]	17.41	34.12	Hourglass-104
Light-RCNN* [28]	16.53	32.78	ResNet101
FPN* [29]	16.51	32.2	ResNet50
Cascade R-CNN* [30]	16.09	31.91	ResNeXt-101
DetNet59* [31]	15.26	29.23	ResNet50
RefineDet* [32]	14.9	28.76	ResNet101
RetinaNet* [21]	11.81	21.37	ResNet101
YOLOv5s	**20.11**	**35.7**	**Improved CSPDarknet53**
YOLOv5s with CBAM	**22.56**	**36.8**	**Improved CSPDarknet53**

IV. EXPERIMENTAL RESULTS

A. Evaluation on VisDrone Datasets

VisDrone 2019 dataset was created by AISKYEYE, a team from the Machine Learning and Data Mining Lab of Tianjin University in China. The dataset consists of 288 videos with 261,908 frames and 10,209 static images shot by various drones equipped with cameras. Filmed in 14 different cities thousands of kilometers away in China, it captures a wide range of aspects: environments such as urban and rural areas, objects such as pedestrians, vehicles, and bicycles, and population densities such as neighborhoods and crowded scenes.

The proposed method tested the object detection performance on the VisDrone dataset. The VisDrone dataset consists of 288 video clips (261,908 images) and 10,209 static photos collected from multiple cameras mounted on drones. This dataset has a total of 10 classes (pedestrian, people, bicycle, car, van, truck, tricycle, awning-tricycle, bus, and motor). The proposed model is trained, validated, and tested with 6,471, 1,610, and 1,610 images, respectively.

An object detection model is evaluated through a dataset by extracting and learning features of various objects included in the dataset by using 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 mAP value of the proposed method is 22.56. it shows better performance compared with other models in the table.1

V. CONCLUSION

In this paper, in order to improve the object detection performance of YOLOv5, object detection is performed by combining YOLOv5 and CBAM. Object detection performance is improved by adding CBAM between the neck and head of YOLOv5. The train is conducted on the VisDrone dataset, and the mAP value on the VisDrone dataset is 22.56, showing better performance compared to models such as TridentNet and CenterNet, which were tested on VisDrone.

For future work, we plan to use ERB (Efficient Residual Bottleneck) and EMRP (Efficient Multi-Receptive Pooling) layers instead of C3 and SPPF layers in the Backbone part of the original YOLOv5. It can make the network more efficient. we have the plan to increase the detection rate of objects by using Attention Modules such as CBAM and to create an efficient network that can perform real-time calculations on low-cost devices. And we plan to use additional detectors to detect tiny size objects.

REFERENCES

[1] T.-Z. Xiang, G.-S. Xia, and L. Zhang, "Mini-unmanned aerial vehicle-based remote sensing: Techniques, applications, and prospects," *IEEE Geoscience and Remote Sensing Magazine*, vol. 7, no. 3, pp. 29–63, 2019.

[2] I. Verdiesen, A. Aler Tubella, and V. Dignum, "Integrating comprehensive human oversight in drone deployment: a conceptual framework applied to the case of military surveillance drones," *Information*, vol. 12, no. 9, p. 385, 2021.

[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] A. Hafeez, M. A. Husain, S. Singh, A. Chauhan, M. T. Khan, N. Kumar, A. Chauhan, and S. Soni, "Implementation of drone technology for farm monitoring & pesticide spraying: A review," *Information processing in Agriculture*, 2022.

[5] 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.

[6] 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.

[7] 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.

[8] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2014, pp. 580–587.

[9] R. Girshick, "Fast r-cnn," in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2015, pp. 1440–1448.

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

[11] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 936–944.

[12] 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)*. IEEE, 2022, pp. 1–4.

[13] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.

[14] L. Zhu, X. Geng, Z. Li, and C. Liu, "Improving yolov5 with attention mechanism for detecting boulders from planetary images," *Remote Sensing*, vol. 13, no. 18, p. 3776, 2021.

[15] L. Wang, Y. Cao, S. Wang, X. Song, S. Zhang, J. Zhang, and J. Niu, "Investigation into recognition algorithm of helmet violation based on yolov5-cbam-dcn," *IEEE Access*, vol. 10, pp. 60 622–60 632, 2022.

[16] Y. Guo, S. Chen, R. Zhan, W. Wang, and J. Zhang, "Sar ship detection based on yolov5 using cbam and bifpn," in *IGARSS 2022-2022 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2022, pp. 2147–2150.

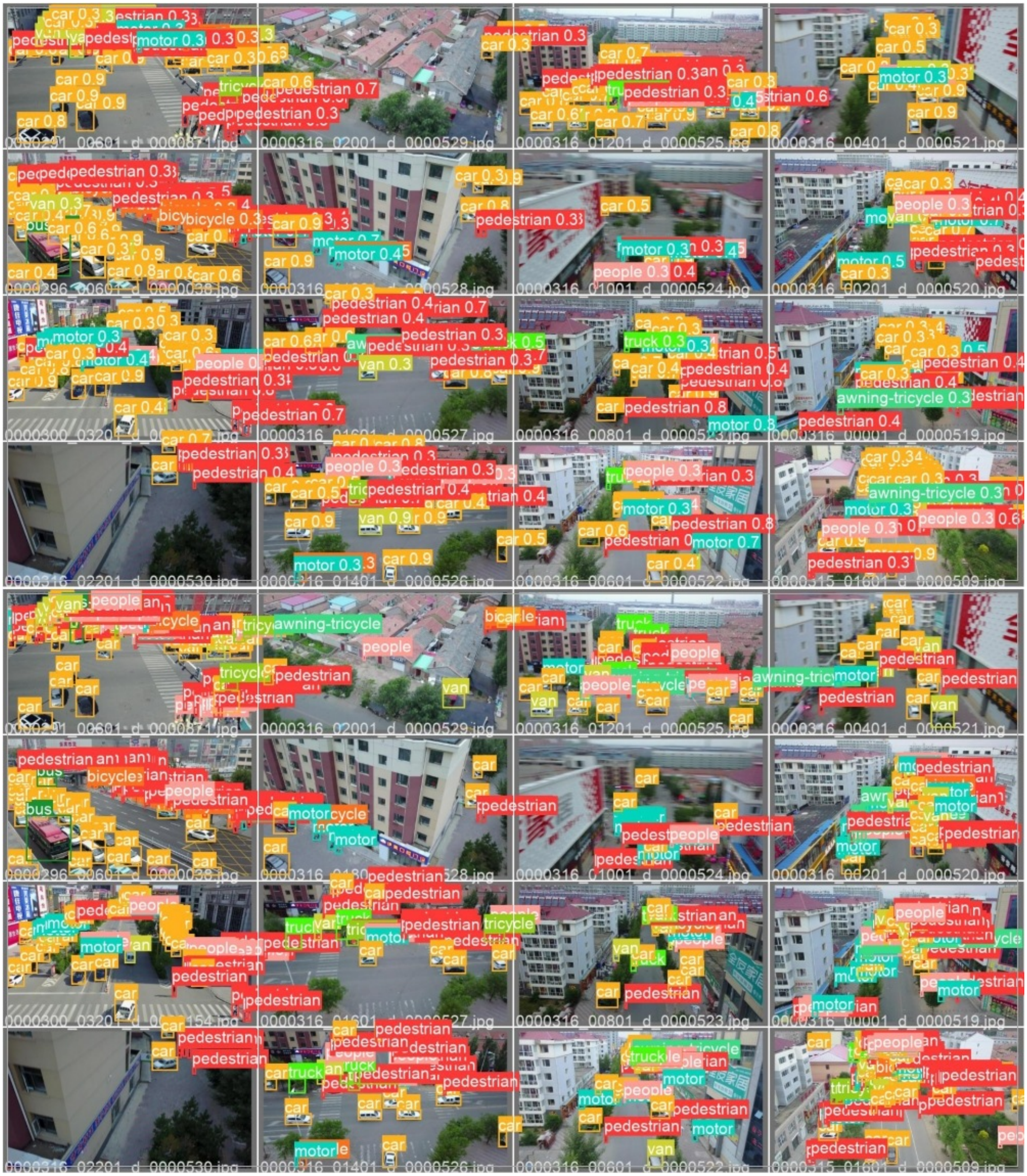
0000291_00601_d_0000871.jpg
0000316_02001_d_0000529.jpg
0000316_01201_d_0000525.jpg
0000316_00401_d_0000521.jpg
0000296_00601_d_0000030.jpg
0000316_01801_d_0000528.jpg
0000316_01001_d_0000524.jpg
0000316_00201_d_0000520.jpg
0000300_03201_d_0000154.jpg
0000316_01601_d_0000527.jpg
0000316_00801_d_0000523.jpg
0000316_00001_d_0000519.jpg
0000316_02201_d_0000530.jpg
0000316_01401_d_0000526.jpg
0000316_00601_d_0000522.jpg
0000315_01601_d_0000509.jpg

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

[17] S.-Q. Yang, Y.-H. Chen, Z.-Y. Zhang, and J.-H. Chen, “Student in-class behaviors detection and analysis system based on cbam-yolov5,” in *2022 7th International Conference on Intelligent Computing and Signal Processing (ICSP)*. IEEE, 2022, pp. 440–443.

[18] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” *CoRR*, vol. abs/1709.01507, 2017. [Online]. Available: http://arxiv.org/abs/1709.01507

[19] Y. Li, Y. Chen, N. Wang, and Z. Zhang, “Scale-aware trident networks for object detection,” 2019.

[20] X. Zhou, D. Wang, and P. Krähenbühl, “Objects as

points," *CoRR*, vol. abs/1904.07850, 2019. [Online]. Available: http://arxiv.org/abs/1904.07850

[21] 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

[22] N. Xie, S. Li, and J. Zhao, "Ercnn: Enhanced recurrent convolutional neural networks for learning sentence similarity," in *Chinese Computational Linguistics*, M. Sun, X. Huang, H. Ji, Z. Liu, and Y. Liu, Eds. Cham: Springer International Publishing, 2019, pp. 119–130.

[23] 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.

[24] 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.

[25] 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.

[26] 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

[27] 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

[28] 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

[29] 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

[30] 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

[31] 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

[32] 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