VSNet: Vehicle State Classification for Drone Image with Mosaic Augmentation and Soft-label Assignment*

Youlkyeong Lee[1] [0000−0002−5806−7502], Jehwan Choi[2] [0009−0005−8494−2170], and Kanghyun Jo[3] [0000−0001−8317−6092]

Department of Electrical, Electronic and Computer Engineering
University of Ulsan, Ulsan, 44610, South Korea
yklee00815@gmail.com[1], choijh1897@gmail.com[2], acejo@ulsan.ac.kr[3]

Abstract. Numerous architectures are under development to comprehend object information and background in images by analyzing features extracted through Convolutional Neural Networks (CNNs). Autonomous driving requires understanding diverse information and collecting data from heterogeneous environments to generalize classification models. However, the patterns of feature maps extracted through convolution layers in drone image data, which encompass assorted types of vehicles and road shapes, tend to be simple, leading to overfitting during model training. To prevent overfitting, this study applies Mosaic Augmentation to increase data diversity and brings generalization to the data. This data augmentation method randomly combines four selected images to create a new mosaic image. Soft-label Assignment is used to determine the labels of the mosaic images. The dataset is collected using a drone flying along roads, and approximately 4,000 images are used for training. In the experiment, the classification performance of vehicle status is listed based on the weight of the loss function of the soft label and hard label. Having achieved an accuracy of 83.41%, the effectiveness of the proposed method is compared with dilated residual networks in terms of improving model performance.

Keywords: Drone image · Transportation system · Vehicle state · Classification.

1 Introduction

Generating desired information through algorithms using various CNN models is crucial for collecting and analyzing image-based traffic information. Utilizing drones for data analysis provides a different perspective on the images compared to the black box or installed cameras in conventional vehicles. Therefore, a drone capturing a wider area at once is essential for comprehending

* This results was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-003).

comprehensive vehicle movement and flow. Understanding vehicle movement is necessary and can be categorized into three basic states: normal state, where the vehicle moves along the lane; lane-changing state, where the vehicle changes its lane; and stop state, where the vehicle comes to a halt.

This research aims to design a CNN model that classifies three vehicle movement states through drone images and applies mosaic data augmentation [1] and soft label assignment [2]. The dataset is collected data using a drone, which captured images with a bird's-eye view of the road. This work utilizes mosaic augmentation to increase data diversity and prevent overfitting during model training. This data augmentation technique randomly combines four selected images to generate a new mosaic image. Soft-label Assignment is used to determine the labels of the mosaic images. These techniques demonstrate the potential of drones for traffic information analysis and the effectiveness of the proposed methodology in improving classification model performance for autonomous driving systems.

2 Related Work

2.1 Autonomous Vehicle dataset for object detection

Images and annotation data collected in the past environment on various roads have been continuously accumulated. With the advancement of object classification and detection technology, autonomous driving technology is being developed rapidly. The Cityscapes [3] and KITTI [4] datasets were created as datasets for autonomous vehicle research [5–10]. The dataset generated image collection and annotation data for traffic conditions on the road through cameras installed in the vehicle. The KITTI dataset also includes 3D bounding box location and camera calibration information through a 3D laser scanner.

2.2 Drone-based dataset for object detection

Stanford drone dataset [11] is the first public aerial image dataset using drones. This dataset contains ten kinds of tracking information (Track ID, (xmin, ymin), (xmax, ymax), frame, lost, occluded, generated, label) about objects on the road in the video image. Images taken at eight locations on the Stanford campus were collected. The targets are six classes (Bicyclist, Pedestrian, Skateboarder, Cart, Car, and Bus). However, the annotation quality of the bounding box of the object is roughly expressed, which has a problem with the performance of the object detection algorithm. The VisDrone[12] dataset is a large-scale drone image produced by AISKYEYE team at Lab of Machine Learning and Data Mining, Tianjin University, China. The dataset aims to develop applications that can be used for computer vision through drones. Through cameras installed in drones, 288 video images were collected from 14 urban areas in China. It produced 2.6 million bounding boxes, including ten classes (pedestrian, person, bus, car, van, truck, bicycle, awning tricycles, motorcycles, and tricycles).

Data validation is tested through VisDrone challenge [13–15] and various kinds of research [16–18] are utilized. The Institut für Kraftfahrwesen Aachen research team had built a drone-based road user trajectory dataset for various situations. The test for vehicles related to autonomous driving is conducted based on the scenario. Therefore, we present reliable and high-quality data criteria. The highD [19] is a large-scale vehicle trajectory dataset for German high roads. It includes six locations, 16.5 hours, and 110,000 trajectory information. In inD [20], automated vehicles require data-based analysis methods to understand complex environments. By collecting road images using drones, it was proposed to collect road trajectories and natural road conditions through vehicle movement. Finally, the dataset provides a dataset including road conditions and vehicles, bicycles, and pedestrians over four kinds of German intersections. The roundD [21] includes the movement trajectories of cars, vans, trucks, buses, pedestrians, bicycles, and motorcycles in three traffic circles in Germany. In addition, positions, headings, speeds, accelerations, and classes of objects were extracted from the video and provided as data.

3 Proposed Algorithm

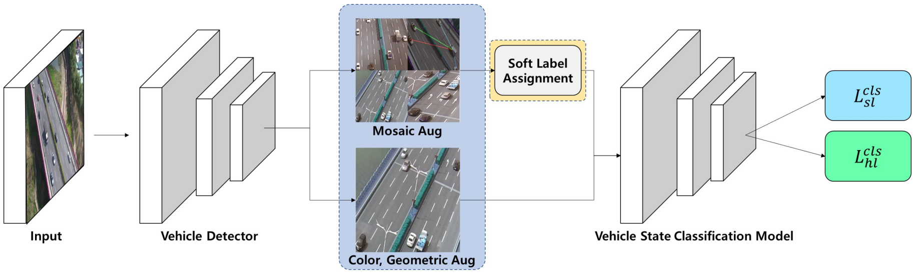


Fig. 1. An overview process of vehicle state classification that contains object detection, data augmentation (mosaic augmentation), soft label assignment, and VSNet (vehicle state network)

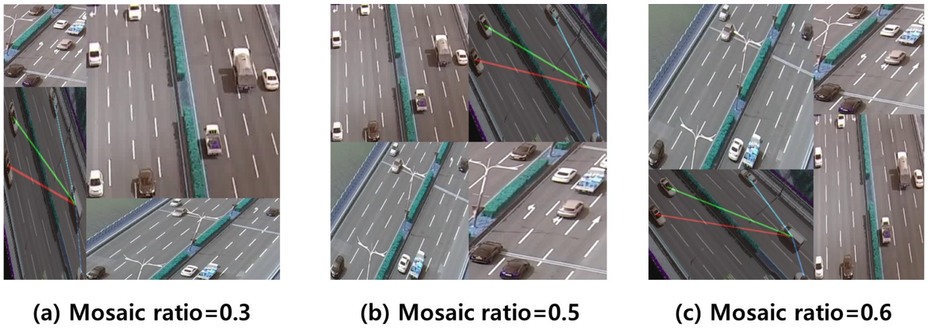
Fig. 1 illustrates the overall process for classifying vehicle status. The process consists of four components: 1) Vehicle detection with YOLOv5, 2) Mosaic data augmentation, 3) Soft-label Assignment, and 4) a network for vehicle state classification. These four components work together to determine the movement status of vehicles. This chapter provides an explanation of the proposed methods in detail.

3.1 Vehicle detection

This study first presents an approach for detecting vehicles using YOLOv5 [22], an advanced object detection algorithm that has achieved state-of-the-art performance on a variety of visual recognition tasks. YOLOv5 is an abbreviation for ”You Only Look Once version 5”, and is an extension of the original YOLO algorithm with improvements in speed and accuracy. It is based on a deep neural network architecture that efficiently extracts features from images and predicts object bounding boxes and class probabilities in a single forward pass.

The YOLOv5 algorithm comprises two main components: a feature extraction backbone and a detection head. The backbone network is built on efficient architecture, which has been shown to be highly efficient and effective in a wide range of vision tasks. The detection head employs anchor boxes and grid cells to predict object locations and classes at multiple scales. To adapt YOLOv5 for vehicle detection task, the train fine-tunes the model on a custom dataset of drone flight images using transfer learning. Vehicle types are limited to car, truck, and bus. Specifically, this work initializes the network with pre-trained weights on the COCO dataset [23] with advanced data augmentation and optimization techniques.

3.2 Mosaic data augmentation



(a) Mosaic ratio=0.3 **(b) Mosaic ratio=0.5** **(c) Mosaic ratio=0.6**

Fig. 2. With the several mosaic ratios, mosaic augmentation generates mixed 4 images to a new image.

Mosaic data augmentation is proposed in YOLOv4 [1] as a technique for augmenting data. This method involves selecting four images from the dataset and arranging them in a manner that is determined by the mosaic ratio, denoted by $\mathcal{M}_r$ so that they are represented as a single image in Fig. 2. $\mathcal{M}_r$ is chosen randomly from the range [0.3, 0.7]. $I(i)$ is image among dataset at index i. Index, i is randomly selected. Every $I(i)$ is resized to 512×512. w_i and h_i denote the width and height of $I(i)$. $I(n)$ contains 4 images that n is an order, $n = 0, \ldots, 3$. Based on $\mathcal{M}_r$, the width and height sizes of each of the four images are determined as follows:

$$I(n) = \begin{cases} \text{new } \mathrm{w}_0 = \mathcal{M}_r \times 512, \ \text{new } \mathrm{h}_0 = \mathcal{M}_r \times 512 & \text{if } n \text{ is } 0 \\ \text{new } \mathrm{w}_1 = 512 - \text{new } \mathrm{w}_0, \ \text{new } \mathrm{h}_1 = 512 - \text{new } \mathrm{h}_3 & \text{if } n \text{ is } 1 \\ \text{new } \mathrm{w}_2 = \mathcal{M}_r \times 512, \ \text{new } \mathrm{h}_2 = 512 - \text{new } \mathrm{h}_0 & \text{if } n \text{ is } 2 \\ \text{new } \mathrm{w}_3 = 512 - \text{new } \mathrm{w}_2, \ \text{new } \mathrm{h}_3 = \mathcal{M}_r \times 512 & \text{if } n \text{ is } 3 \end{cases} \tag{1}$$

This approach enables the model to learn from multiple images simultaneously, improving generalization by incorporating diverse contextual information into a single image. Mosaic is applied by selecting random numbers in quantity equal to the batch size for each iteration.

3.3 Soft label assignment

After applying mosaic augmentation in this study, a method of soft label assignment for label allocation is proposed. In the mosaic image, four original images correspond to four labels. A soft label is created by referring to label smoothing [2]. The soft label, $\mathcal{S}(x)$ is shown in Eq. (2). In training sample x, $h_i(k|x)$ represents the hard label distribution of the four images at classes, $k \in 0, 1, 2$ and index of distribution i. The hyperparameter α is assigned a weight value between 0 and 1. The value of K, which denotes the number of images, is 4. Eq. (2) are defined as the following:

$$\mathcal{S}(x) = \frac{1}{N} \sum_{i=1}^{N} \{(1 - \alpha) h_i(k|x) + \alpha/K\} \tag{2}$$

The ground truth label distribution multiples the weight α and interpolates the hard label through the α/K. The label of the mosaic image generates a soft label by calculating the average value from $(1-\alpha)h_i(k|x)+\alpha/K$. Eq. (2) represents the ground truth soft label, which adjusts the ground truth label distribution by applying label smoothing to mosaic images for classification models. In conclusion, these methods adapt the use of mosaic augmentation and soft label assignment resulting in improved classification model performance.

3.4 Vehicle state classification

The proposed model comprises the Wide Area Feature Extraction (WAFE) module and Deformable Residual (DR) module. These modules play critical roles in extracting and focusing on feature information. The following section provides a detailed layer-by-layer explanation of these modules.

Wide area feature extraction module (WAFE module) To classify the state of the target vehicle, the input image considers the position and state of the surrounding vehicles. Fig. 2 shows that the vehicles in the image are mostly separated. To exclude unnecessary information like background, the first convolutional layer passes a 5×5 kernel size with 64 filters, a stride of 4, and a dilated

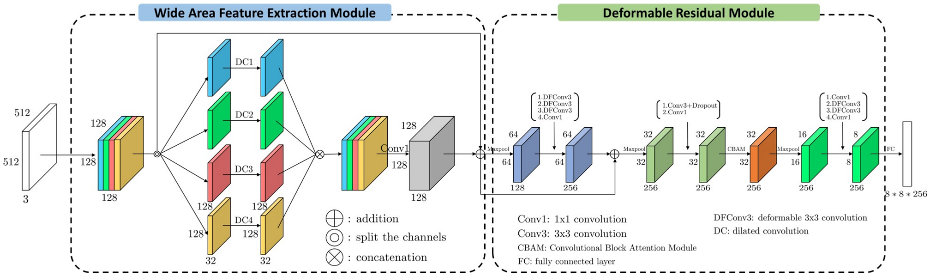


Fig. 3. The framework of the proposed method. Object detection generates bounding boxes of vehicles. Then, the condition module re-identifies and matches the vehicle ID in sequential images. In the condition module, the moving distance between the current and the next frame of each vehicle is computed, and the direction of each vehicle is computed. Subsequently, the selection module decides to remove or keep the detection result by considering the position of each vehicle.

ratio of 3. To stabilize the learning process on the feature map, batch normalization (BN) [24] is performed after all convolutional layers, and the proposed network employs Gaussian Error Linear Units (GELU) [25] as the activation function. The feature map is further processed by dividing the channels into four groups, and each group is passed through four kinds of dilated ratio, [1,3,5,7] of 3×3 convolutional layer. The four groups of outputs are concatenated, and a 1×1 kernel is applied. Additionally, a residual block is used to incorporate previous information before the maxpooling operation into the feature map.

Deformable residual module (DR module) Deformable residual is modified from deformable convolutional layer [26] to extract flexible spatial information through output feature from WAFE module.

As illustrated in Fig. 4(a), the traditional 3×3 convolutional layer has a fixed receptive field in the image area, represented by the red and blue dots. However, in the image data used for vehicle detection, the vehicles are often separated from each other. Therefore, using a fixed receptive field would extract feature information that includes unnecessary background information. To address this issue and perform more effective convolutional operations, deformable convolution is employed. Fig. 4(b) shows how deformable convolution generates an offset as the convolutional layer and performs convolution operations through the offset information.

$$\mathbf{y}(\mathbf{p}_0) = \sum_{\mathbf{p}_n \in \mathcal{R}} \mathbf{w}(\mathbf{p}_n) \cdot \mathbf{x}(\mathbf{p}_0 + \mathbf{p}_n + \triangle \mathbf{p}_n) \tag{3}$$

Eq. (3) represents the offset that determines the kernel position in deformable convolution. $\mathbf{y}$ represents the output feature map. The kernel grid, $\mathcal{R}$, is defined as the receptive field, where $\mathcal{R} = (-1,-1), (-1,0), \ldots, (0,1), (1,1)$. The convolution occurs at the pixel position of the input image $\mathbf{x}$, which is $\mathbf{p}_0$, and at

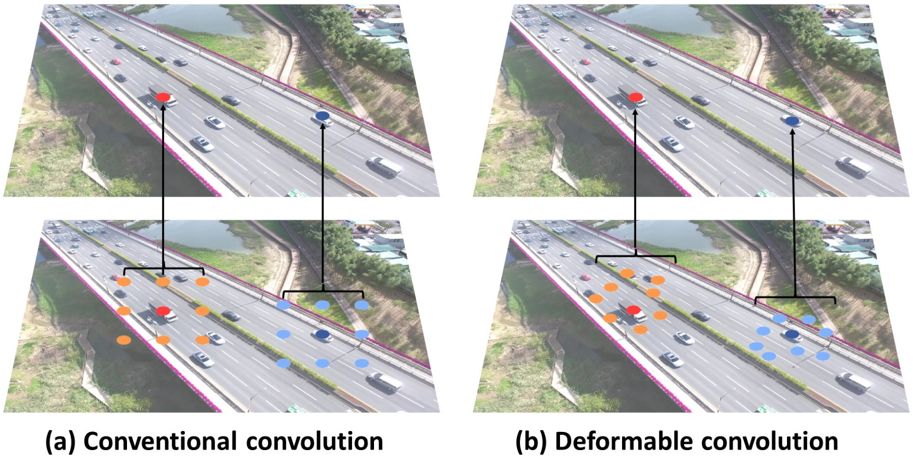


Fig. 4. (a) 3 × 3 conventional convolution, (b) 3 × 3 deformable convolution, In deformable convolution, deep red and dark blue dots are focused on the vehicle in the input image.

the individual positions in $\mathcal{R}$, which are $\mathbf{p}_n$, along with the offset, $\triangle\mathbf{p}_n$. In particular, the offset $\triangle\mathbf{p}_n$ value is generated based on the convolution layer value and is trained in each iteration. Thus, $\mathbf{p}_0 + \mathbf{p}_n + \triangle\mathbf{p}_n$ ultimately determines the position of the input value, and the convolution operation is performed by multiplying it with the convolution kernel weight, $\mathbf{w}(\mathbf{p}_n)$, at that position.

The Convolutional Block Attention Module (CBAM) [27] allows for complementary attention of both channel-wise and spatial-wise information and is applied through the output of three deformable convolutional layer operations. The fully connected layer receives the feature map that has been calculated by two deformable convolutional layers.

Loss function During training, hard and soft labels are utilized to adjust the loss function. For hard labels, the original loss function uses in this study is Focal Loss [28], which helps to balance the training process and prevent bias towards one class when dealing with data imbalance issues. For soft labels, the loss function is the mean squared error (MSE). Since the value of the soft labels is a float number, it has been computed average value. The proposed total loss function, $\mathcal{L}$ is as follows:

$$\mathcal{L} = \alpha_l \mathcal{L}_{hl}^{cls} + (1 - \alpha_l)\mathcal{L}_{sl}^{cls} \tag{4}$$

$\mathcal{L}_{hl}^{cls}$ is the loss calculated for the hard label, and $\mathcal{L}_{sl}^{cls}$ represents the loss result for the soft label. The parameter, α_l assigns weights to the hard and soft losses. Since soft labels are selected less frequently than hard labels, the weight of hard labels is higher. α_l is 0.9

4 Experiment

Drone Image Dataset: The dataset for drone images is captured from a top-down perspective, and vehicle detection is performed using YOLOv5 large model on the collected images. After detection, the image is cropped based on the five vehicles surrounding the target vehicle. The dataset consists of three classes: lane_change, safe, and stop, and the total number of training and test data is shown in Table 1.

Configuration Details In this study, the Adam optimizer [29] is employed and the learning rate is set to 0.001. Epoch is 200. Four NVIDIA RTX 3090 GPU, each with 24GB of memory, are used, with a batch size of 16.

Table 1. Information of train and test dataset for vehicle state classification

Class	**train**	**test**	**Total**
lane_change	**860**	**214**	**1,074**
safe	**1,241**	**310**	**1,551**
stop	**1,222**	**305**	**1,525**
Total	**3,323**	**829**	**4,152**

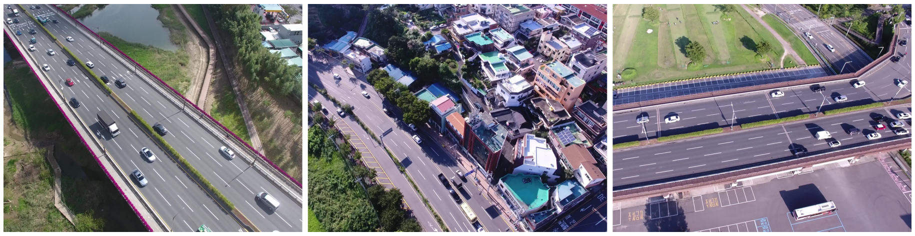


Fig. 5. An overview vehicle state classification

Object Detection In this study, YOLOv5 [22] is adopted as the object detection algorithm, and car and truck are the two classes considered for train and test. Table 3 presents the object detection performance for the train and test datasets. The training on 9,776 images a performance of 95.75 mAP(AP_{50}) and 83.8 mAP($AP_{50:95}$), while testing on 2,200 images a performance of 91.8 mAP(AP_{50}) and 80.3 mAP($AP_{50:95}$). Utilizing this detector, other traffic videos are analyzed to identify and extract vehicle information, including their position and class.

Table 2. The mAP performance of YOLOv5 on drone train and test dataset

Class	Images	Instance	mAP@50	mAP@50:95
all_train	9,776	309,470	95.75	83.8
car_vehicle	9,776	277,263	97.2	86.0
truck_vehicle	9,776	32,207	94.3	81.6
all_test	2,200	85,398	91.8	80.3
car_vehicle	2,200	78,765	96.1	85.3
truck_vehicle	2,200	6,633	87.5	75.4

VSNet Performance The performance of the network for classifying the final vehicle state in Table 3 is compared to the Dilated Residual Network (DRN) [30]. DRN is a classification model derived from ResNet [31] and is a network that replaces the convolutional layers with dilated convolutional layers. Proposed model utilizes dilated and deformable convolutional layers to extract features from a wide area, and therefore, its network is compared with DRN, which is composed of dilated convolutional layers. DRN has four types, A, B, C, and D, with additional dilated blocks and skip-connections. In this paper, types C and D are used, and type D is a simplified version of type C.

Table 3. Comparison result with dilated residual networks (DRN) and vehicle state network (VSNet) for data augmentation.

Method	Data augmentation		#para	Acc(%)
	Mosaic	Color		
DRN_C_26	-	-	21,126,584	89.62
DRN_C_42	-	-	31,234,744	81.78
DRN_D_22	-	-	16,393,752	87.69
DRN_D_38	-	-	26,501,912	86.49
DRN_D_54	-	-	35,809,176	89.26
Proposed	-	-	1,273,504	72.85
Proposed	-	O	1,273,504	79.73
Proposed	O	-	1,273,504	81.25
Proposed	O	O	1,273,504	83.41

Compared to DRN_D_22, the first proposed model shows a 16.9% difference in accuracy results, but it reduces the number of parameters by 92.2%. In addition, the proposed model presents the results of applying mosaic and color data augmentation. When both augmentations are applied, it shows the best performance among the results presented, with an accuracy of 83.41%. Furthermore, compared to the DRN_C_42 model, it achieves a 1.63% higher accuracy and saves 96% of the parameters.

Table 4 presents the accuracy performance of the proposed model according to the soft label values. The highest accuracy performance of 83.41% is achieved when α is set to 0.7. As α gradually decreases, the performance decreases as well. This is because the soft label values differ from the original hard label values, leading to differences in learning performance.

Table 4. According to Soft label, α, accuracy of vehicle state classification.

Method	**Soft label=α**	**Acc(%)**
Proposed	**0.3**	**79.62**
Proposed	**0.4**	**81.73**
Proposed	**0.5**	**83.12**
Proposed	**0.6**	**82.57**
Proposed	**0.7**	**83.41**

5 Conclusion

This study applies mosaic augmentation and soft-label assignment techniques to classify vehicle states using drone images. Mosaic augmentation combines existing images to create a new image, increasing the amount of data and improving generalization for a limited dataset. Additionally, soft-label assignment is used to generate labels for the mosaic images in vehicle state classification. These two techniques contribute to smooth training and enhance the accuracy performance of the proposed classification model.

Acknowledgment

This results was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(MOE)(2021RIS-003)

References

1. Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *ArXiv*, abs/2004.10934, 2020.
2. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, 2015.
3. Marius Cordts, 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. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

4. Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32:1231 – 1237, 2013.
5. Liang-Chieh Chen, Huiyu Wang, and Siyuan Qiao. Scaling wide residual networks for panoptic segmentation. *ArXiv*, abs/2011.11675, 2020.
6. Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Y. Qiao. Vision transformer adapter for dense predictions. *ArXiv*, abs/2205.08534, 2022.
7. Jiacong Xu, Zixiang Xiong, and S. Bhattacharyya. Pidnet: A real-time semantic segmentation network inspired from pid controller. *ArXiv*, abs/2206.02066, 2022.
8. Shichao Li, Zengqiang Yan, Hongyang Li, and Kwang-Ting Cheng. Exploring intermediate representation for monocular vehicle pose estimation. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1873–1883, 2020.
9. Yifan Zhang, Qijian Zhang, Zhiyu Zhu, Junhui Hou, and Yixuan Yuan. Glenet: Boosting 3d object detectors with generative label uncertainty estimation. *ArXiv*, abs/2207.02466, 2022.
10. Yu Hong, Hang Dai, and Yong Ding. Cross-modality knowledge distillation network for monocular 3d object detection. *ArXiv*, abs/2211.07171, 2022.
11. Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *European Conference on Computer Vision*, 2016.
12. Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian, Heng Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1, 2020.
13. Yaru Cao, Zhi-Yang He, Lujia Wang, Wenguan Wang, Yixuan Yuan, Dingwen Zhang, Jinglin Zhang, Pengfei Zhu, Luc Van Gool, Junwei Han, Steven C. H. Hoi, Qinghua Hu, Meilin Liu, Chong Cheng, Fanfan Liu, Guojin Cao, Guozhen Li, Hongkai Wang, Jianye He, Jun-Jun Wan, Qianxiang Wan, Qi Zhao, Shuchang Lyu, Wenzhe Zhao, Xiaoqiang Lu, Xingkui Zhu, Yingjie Liu, Yixuan Lv, Yujing Ma, Yuting Yang, Zhe Wang, Zhenyu Xu, Zhipeng Luo, Zhimin Zhang, Zhiguang Zhang, Zihao Li, and Zixiao Zhang. Visdrone-det2021: The vision meets drone object detection challenge results. *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 2847–2854, 2021.
14. Dawei Du, Longyin Wen, Pengfei Zhu, Heng Fan, Qinghua Hu, Haibin Ling, Mubarak Shah, Junwen Pan, Ali Al-Ali, Amr M. Mohamed, Bakour Imene, Bin Dong, Binyu Zhang, Bouchali Hadia Nesma, Chenfeng Xu, Chenzhen Duan, Ciro Castiello, Corrado Mencar, Dingkang Liang, Florian Krüger, Gennaro Vessio, Giovanna Castellano, Jieru Wang, Junyu Gao, Khalid Abualsaud, Laihui Ding, Lei Zhao, Marco Cianciotta, and Muhammad Saqib. Visdrone-cc2021: The vision meets drone crowd counting challenge results. pages 2830–2838, 2021.
15. Heng Fan, Dawei Du, Longyin Wen, Pengfei Zhu, Qinghua Hu, Haibin Ling, Mubarak Shah, Junwen Pan, Arne Schumann, Bin Dong, Daniel Stadler, Duo Xu, Filiz Bunyak, Guna Seetharaman, Guizhong Liu, and V. Haritha. Visdrone-mot2021: The vision meets drone multiple object tracking challenge results. *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 2839–2846, 2021.
16. Jinwang Wang, Chang Xu, Wen Yang, and Lei Yu. A normalized gaussian wasserstein distance for tiny object detection. *ArXiv*, abs/2110.13389, 2021.

17. Youlkyeong Lee, Qing Tang, Je-Woo Choi, and Kanghyun Jo. Low computational vehicle re-identification for unlabeled drone flight images. *IECON 2022 – 48th Annual Conference of the IEEE Industrial Electronics Society*, pages 1–6, 2022.
18. Youlkyeong Lee, Qing Tang, Je-Woo Choi, and Kanghyun Jo. Low computational vehicle lane changing prediction using drone traffic dataset. *2022 International Workshop on Intelligent Systems (IWIS)*, pages 1–4, 2022.
19. Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 2118–2125, 2018.
20. Julian Bock, Robert Krajewski, Tobias Moers, Steffen Runde, Lennart Vater, and Lutz Eckstein. The ind dataset: A drone dataset of naturalistic road user trajectories at german intersections. *2020 IEEE Intelligent Vehicles Symposium (IV)*, pages 1929–1934, 2019.
21. Robert Krajewski, Tobias Moers, Julian Bock, Lennart Vater, and Lutz Eckstein. The round dataset: A drone dataset of road user trajectories at roundabouts in germany. *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6, 2020.
22. Glenn R. Jocher, Alex Stoken, Jiří Borovec, NanoCode, Ayushi Chaurasia, TaoXie, Liu Changyu, Abhiram, Laughing, tkianai, yxNONG, Adam Hogan, lorenzomammana, AlexWang, Jan Hájek, Laurentiu Diaconu, Marc, Yonghye Kwon, Oleg, wanghaoyang, Yann Defretin, Aditya Lohia, ml ah, Ben Milanko, Ben Fineran, D. P. Khromov, Ding Yiwei, Doug, Durgesh, and Francisco Ingham. ultralytics/yolov5: v5.0 - yolov5-p6 1280 models, aws, supervise.ly and youtube integrations. 2021.
23. Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.
24. Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, 2015.
25. Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv: Learning*, 2016.
26. Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 764–773, 2017.
27. Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In-So Kweon. Cbam: Convolutional block attention module. In *European Conference on Computer Vision*, 2018.
28. Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:318–327, 2017.
29. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
30. Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.
31. Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.