Vehicle State Classification from Drone Image

Youlkyeong Lee, Seongmin Kim, Jehwan Choi, Kanghyun Jo
Dept. of Electrical, Electronic and Computer Engineering
University of Ulsan, Ulsan, Korea
{yklee00815, dailysmile3347, choijh1897}@gmail.com, acejo@ulsan.ac.kr

2023 IEEE International Conference on Industrial Technology (ICIT) | 979-8-3503-3650-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICIT58465.2023.10143168

***Abstract*—The use of drone views helps to create a safe transportation system by providing various traffic information. This paper aims to identify changes in ground vehicle movements, such as stopping, lane changing, and safety, by tracking vehicles on the road. The vehicle condition is determined in two ways. First, the collected drone images are refined, and the data is augmented using the mixup method to identify the state of the vehicle. Second, a proposed learning model, the Wide Area Feature Extraction (WAFE), and Deformable Residual Module (DRM) are used. WAFE generates features by extracting objects across a wide area. DRM utilizes a deformable convolutional layer to extract features, incorporating information from the previous layer to create a feature map with receptive field flexibility. The experimental results indicate an 88.6% accuracy for the vehicle state classification, with the model containing a total of 1.27M parameters. This represents a significant improvement over DRN_C_26, with a decrease 95% in the total number of parameters and a difference of 1.06% in accuracy.**

***Index Terms*—Drone image, transportation system, vehicle state, classification, object detection**

I. Introduction

As the use of drones increases, research in various transportation fields is exploring new ways to collect and analyze traffic-related information. Traffic information encompasses a range of data, including vehicle speeds collected through sensors, traffic card systems, bus management systems, and information on the volume and types of traffic, such as bus, taxi, bicycle, pedestrian, etc. In traditional traffic information systems, cameras installed in limited areas captured traffic information. However, drones offer the advantage of being able to move to different locations and capture aerial views of large areas. Recently, several drone aerial image datasets, such as VisDrone [1], highD [2], inD [3], roundD [4], etc., have been developed to collect information on the movement of vehicles on the road to support the safe operation of autonomous vehicles. These datasets provide a bird's-eye view of vehicle distribution, making it possible to quickly understand the state of vehicles on the road and support efficient traffic flow and safe autonomous driving research.

Recent research in convolutional neural networks analyzes the features of objects within images and continues to develop performance in the fields of object detection [5], [6], [7], re-identification [8], [9], [10], motion prediction [11], [12], and classification [13], [14]. Safe autonomous vehicle research requires the detection and prediction of dynamic changes in the movement of surrounding vehicles. It also requires the response of the road view of the target vehicle or the sensor installed in the vehicle. This paper proposes a study to classify the state of the target vehicle using the region of interest of the detected vehicle. The paper presents a novel classification model along with data augmentation techniques. The results show that the classification learning model increases the diversity of limited data while also ensuring real-time performance. The dataset was produced for the purpose of vehicle state classification. It was created by extracting images collected by drones during flight.

The analysis of the state of a vehicle on the road through drone images requires the interaction of classification and detection algorithms. Vehicle detection is the first necessary step in determining the condition of a vehicle, which can achieve high performance through the use of various object detection algorithms. Ongoing research in real-time applications is actively occurring. The YOLO series [7], [15] of algorithms produce both object classification and localization in a single-stage, with YOLOv8 [16] (2023) representing the latest performance advances. Additionally, the Re-Identification (Re-Id) method plays a critical role in identifying and tracking vehicles across multiple continuous images. Re-Id is a popular research topic in intelligent surveillance systems for tracking people [9], [10] and vehicles [8], and numerous studies are underway. With the growing demand for autonomous driving technology, the area of vehicle motion detection is rapidly evolving. Several studies have investigated real-time vehicle and lane detection for vehicle movement detection [17], as well as vehicle motion estimation based on CCTV data [18], [19]. Furthermore, a growing body of research is focused on self-driving vehicle technology using AI [20]. This paper proposes a convolutional neural network for vehicle motion classification using images collected by drones, exploring data augmentation techniques, and proposed vehicle state classification models.

II. Proposed Algorithm

A. Object Detection

Vehicle detection is a key factor in evaluating the safety and condition of the vehicle. This study employs YOLOv5 [15] for vehicle detection. The model was trained using a combination of self-processed drone flight images and the VisDrone [1] dataset. The YOLOv5 detector, which has demonstrated outstanding performance in real-time object detection, will be utilized by the vehicle status classifier for real-time vehicle detection in future applications.

Authorized licensed use limited to: University of Ulsan. Downloaded on June 14,2023 at 01:17:11 UTC from IEEE Xplore. Restrictions apply.

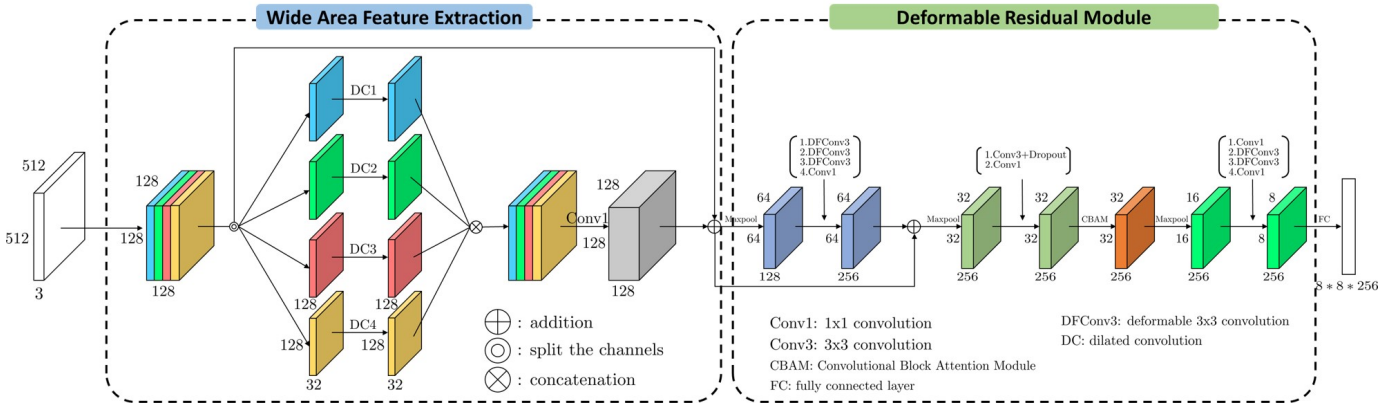


Fig. 1: The framework of the proposed method. Object detection generates bounding boxes of vehicles. Then, the condition module re-identifies and matches the vehicle ID in sequential images. In the condition module, the moving distance between the current and the next frame of each vehicle is computed, 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.

B. Data Generation and Augmentation

1) Data Generation: The object detector generates image data based on the detected vehicle for classification. One of the detected vehicles is randomly chosen and selected as a target vehicle. The method for extracting a cropped image area for vehicle state classification is referred to as Algorithm 1. A vehicle $V_{t,i}$ is selected as the target vehicle by randomly selecting one of the detected vehicles, $V_{d,j}$. The Euclidean distance, $D_i(V_{t,i}, V_{d,j})$ is calculated between the target vehicle and the surrounding vehicles. Top5(List_{D_i}) is selected as a short distance of Top5. The minimum values of x_i and y_i, as well as the maximum values of x_i and y_i, are selected from the list. The cropped area of the original image is then selected, as illustrated in Fig 2(b). The goal is to understand

Algorithm 1 cropped area for classification

Data: detected vehicle bbox: $V_{d,j}$, target vehicle bbox: $V_{t,i}$
Result: $\text{I}[x_{d,i,min} : x_{d,i,max}, y_{d,i,min} : y_{d,i,max}]$
for $V_{t,i}$ **do**
  for $V_{d,j}$ **do**
    $List_{D_i} = D_i(V_{t,i}, V_{d,j})$
  end
end
$\text{Top5}(\text{List}_{D_i}) = \text{Top5}_{D_i}$
$\rightarrow \min(\text{Top5}_{D_i}) = (x_{d,i,min}, y_{d,i,min})$
$\rightarrow \max(\text{Top5}_{D_i}) = (x_{d,i,max}, y_{d,i,max})$
$\text{points} = [(x_{d,i,min}, y_{d,i,min}), (x_{d,i,max}, y_{d,i,max})]$
$\text{cropped I} = \text{I}[x_{d,i,min} : x_{d,i,max}, y_{d,i,min} : y_{d,i,max}]$

the state of the vehicle by analyzing the characteristics of the surrounding vehicles within a specific, cropped area. Fig 2(c) shows the linear distance between the target vehicle and the surrounding vehicles. Fig 2(d) generates a heatmap to represent the positions of the five surrounding vehicles.

2) Data Augmentation: Data augmentation techniques are utilized on the image data used for vehicle state classification, resulting in a weighted generalization of the learned model

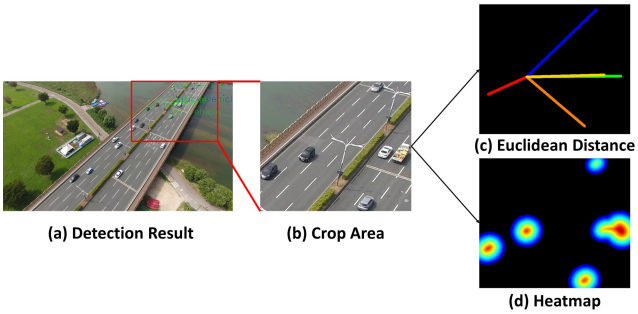


Fig. 2: Data Generation: (a)Detection result by YOLOv5, (b)crop area, (c)euclidean distance for line, (d)heatmap for bbox location

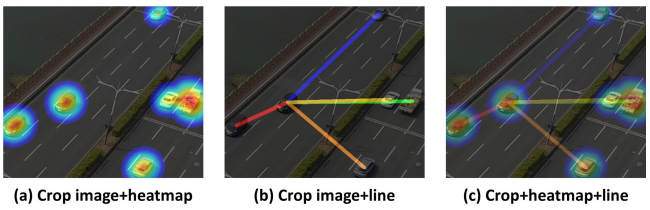


Fig. 3: mixup: (a)crop area + heatmap, (b)crop area + line, (c)crop area + heatmap + line

by increasing the diversity of the data. Fig 2 demonstrates the application of three data combinations through the mixup[7], [21] technique. The mixup process works as follows: a value for $\alpha_i = \text{rand(N)}$ is randomly selected from a range of 0 to 1. The number of α_i values is determined based on the number of input images.

$$p_i = \frac{e^{\alpha_i}}{\sum_{j=1}^{k} e^{\alpha_i}} \text{for } i = 1, 2, \ldots, N \tag{1}$$

In Eq 1, the overall probability of α_i and p_i values is determined using the softmax function, which serves as the

Authorized licensed use limited to: University of Ulsan. Downloaded on June 14,2023 at 01:17:11 UTC from IEEE Xplore. Restrictions apply.

weight for the pixel values of the image. The following steps are taken to generate the input image I_i as described in Eq 2:

$$I_i = \sum_{k=1}^{N} I_k * p_k \tag{2}$$

N is 3 that are cropped image, linear distance, heatmap. k is the index of kinds of images.

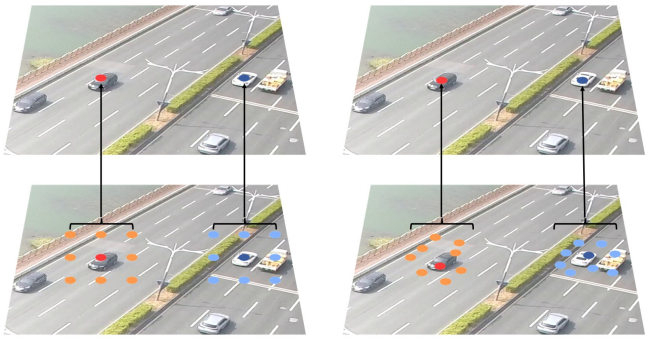
C. Proposed architecture

The proposed architecture in this paper is designed for Vehicle State Classification, as illustrated in Fig 1. The classifier consists of a Wide Area Feature Extraction (WAFE) backbone and a Deformable Residual Module (DRM), and has a total of 1,265,022 parameters. Detailed information is shown in Table 1.

TABLE 1: Vehicle State Classification Network Convolutional layer configurations. **c**: channel, **k**: kernel size, **s**: stride, **d**: dilated ratio, **p**: padding

No	Contents	size	c	k	s	d	p
0	Input image	512×512	3	-	-	-	-
1	Initial Block	128×128	64	3	4	3	6
2	Initial Block	128×128	128	1	1	1	0
3-1	Dilated conv3	128×128	32	3	1	1	1
3-2	Dilated conv3	128×128	32	3	1	3	3
3-3	Dilated conv3	128×128	32	3	1	5	5
3-4	Dilated conv3	128×128	32	3	1	7	7
4	conv1	128×128	128	1	1	1	0
5	maxpool	64×64	-	-	-	-	-
6	Deformable conv3	64×64	128	3	1	1	1
7	Deformable conv3	64×64	128	3	1	1	1
8	Deformable conv3	64×64	128	3	1	1	1
9	maxpool	32×32	-	-	-	-	-
10	conv3+dropout	32×32	128	3	1	1	1
11	conv1	32×32	256	1	1	1	0
12	CBAM	32×32	256	reduction ratio=16			
13	maxpool	16×16	-	-	-	-	-
14	conv1	16×16	128	1	1	1	0
15	Deformable conv3	16×16	128	3	2	1	1
16	Deformable conv3	8×8	128	3	1	1	1
17	conv1	8×8	256	1	1	1	0
18	fc	16,384					
Total Parameters:		1,265,022					

1) WAFE Backbone: The input image classifies the state of the target vehicle based on the position and state of the vehicles surrounding it. As shown in Fig 2(b), the objects in the image are mainly separated. To exclude information from unnecessary areas, the initial convolutional layer uses a 5×5 kernel size with 64 filters, a stride 4, and a dilated ratio 3. The number of channels is increased from 64 to 128 through a 1×1 kernel, allowing for the sharing of information between channels about the reduced feature map size. The activation function used in this proposed network is Gaussian Error Linear Units (GELU) [22]. Batch normalization (BN) is performed after all convolutional layers to stabilize the learning progress on the feature map. Maxpooling downsamples the feature map by half. The channels are divided into four groups and a 3×3 convolutional layer with dilated ratios of [1,3,5,7] is applied respectively. The four groups are concatenated and a 1×1 kernel is applied. This architecture generates a feature map by extracting features from flexible regions, rather than using three times 3×3 deformable convolutional layers on existing fixed responsive fields. Additionally, the feature map incorporates previous information before the maxpooling operation, using a residual block.



(a) standard convolution (b) deformable convolution

Fig. 4: (a)standard convolution, (b)deformable convolution, In deformable convolution, deep red and dark blue dots are flexible to compute the pixels in the input image.

2) Deformable Residual Module: The deformable convolutional layer [23] is informed to generate flexible spatial information through features from WAFE. As demonstrated in Fig 4(a), the traditional 3×3 convolutional layer has a fixed 3×3 receptive field in the image area. In contrast, Fig 4(b) generates an offset as the convolutional layer and performs convolution operations through the offset information. Vehicle features in the image data are distributed over a wide area, making it more effective to extract features from a flexible area rather than a fixed receptive field. As indicated in Eq 3, sampling using a regular grid, $\mathcal{R} = \{(-1,-1),(-1,0),\ldots,(0,1),(1,1)\}$ for the input feature map $\mathbf{x}$, location in the pixel, $\mathbf{p}_0$, and offset, $\triangle\mathbf{p}_n$ are added to the pixel multiplication position $\mathbf{p}_0 + \mathbf{p}_n + \triangle\mathbf{p}_n$ of the traditional convolution.

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

The Convolutional Block Attention Module (CBAM) [24] 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.

III. Experiment

Drone Image Dataset: The drone image dataset is captured from a bird's eye view perspective. Using the collected images and a YOLOv5 large model, the data is able to detect vehicles. Subsequently, the image is cropped from the original image based on the five vehicles surrounding the target vehicle using Algorithm 1. The dataset comprises three classes: lane_change, safe, and stop, with the total number of training and test data shown in Table 2 below.

Configuration Details: The images used in the learning

Authorized licensed use limited to: University of Ulsan. Downloaded on June 14,2023 at 01:17:11 UTC from IEEE Xplore. Restrictions apply.

TABLE 2: Drone 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

process were resized to 512×512. A learning rate of 0.001 is utilized, and the optimizer employed is Adam [25]. Focal loss [26] is chosen as the loss function. Four NVIDIA A100 GPUs, each with 40GB of memory, are used, with a batch size of 32.

Object Detection: This paper utilizes YOLOv5 [15] as the object detection algorithm. The training and testing of this work focuses on two classes: car and truck. The object detection performance of the training and testing datasets is shown in Table 3. The results obtained from the 9,776 training images are 95.75 mAP(AP_{50}) and 83.8 mAP($AP_{50:95}$). Similarly, the results obtained from the 2,200 testing images are 91.8 mAP(AP_{50}) and 80.3 mAP($AP_{50:95}$). Using this detector, other video clips of traffic on roads are analyzed to detect vehicles and extract information such as position and class.

TABLE 3: 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

The vehicle state classification architecture incorporates two residual blocks before maxpooling. Table 4 presents the performance of the network based on each residual module. ✓is used to indicate whether the residual is True or False. It shows the order of listing according to performance. The first achieved 77.6% accuracy when the second residual block was used. The accuracy improved by 3.0% when the two residual blocks were not used. Using all residual blocks showed a performance improvement of 4.9%. However, using only the first residual block resulted in the highest performance improvement of 8.0%. The feature map added through the second residual block caused a performance reduction of approximately 3% on the overall performance. Additionally, the feature map used in the network contributed to a performance improvement of approximately 5%.

TABLE 4: Comparison of accuracy with the residual module in proposed network.

Method	Residual module		Acc(%)
	First residual	Second residual	
Proposed			80.6(+3.0)
	✓		85.6(+8.0)
		✓	77.6
	✓	✓	82.5(+4.9)

TABLE 5: Comparison of accuracy with various data augmentation methods.

Method	Residual		Data augmentation			#para	Acc(%)
	1st	2nd	Color	Geo	Mixup		
DRN_B_22	-	-	-	-	-	16.39M	87.69
DRN_B_38	-	-	-	-	-	26.50M	86.49
DRN_B_54	-	-	-	-	-	35.80M	89.26
DRN_C_26	-	-	-	-	-	21.13M	89.62
DRN_C_42	-	-	-	-	-	32.23M	81.78
Proposed	✓				✓	1.27M	81.9
	✓		✓		✓	1.27M	88.6
	✓			✓	✓	1.27M	75.8
	✓		✓	✓	✓	1.27M	85.6
	✓					1.27M	72.8
	✓		✓			1.27M	79.7
	✓			✓		1.27M	81.3
	✓		✓	✓		1.27M	81.1
	✓	✓			✓	1.27M	73.5
	✓	✓	✓		✓	1.27M	86.0
	✓	✓		✓	✓	1.27M	84.4
	✓	✓	✓	✓	✓	1.27M	82.8
	✓	✓				1.27M	67.0
	✓	✓	✓			1.27M	82.2
	✓	✓		✓		1.27M	66.3
	✓	✓	✓	✓		1.27M	76.8

As a classification model of an adapting flexible receptive field, Dilated residual networks (DRN) [27] has a total of 21.13M parameters(DRN_C_26). The proposed model has approximately 95% fewer parameters. Table 5 describes the performance comparison between DRN and the proposed model using various data augmentation techniques. Image transformations with brightness=0.5, contrast=0.5, saturation=0.5, and hue=0.5 are applied. Geo transform refers to the geometrical transformation of an image, applying random rotation of -10 to +10 degrees. Mixup refers to an image synthesized with the original for heatmap and line plotting images. DRN_C_26 achieves the highest accuracy, 89.62%. Two residual conditions from the proposed model are taken to compare the performance of applying data augmentation. The first proposed model (1st=True, 2nd=False) using color Transform and mixup shows the highest performance, with an accuracy of 88.6%. Similarly, the second proposed model (1st=True, 2nd=True) achieves 86.0%. Compared to the DRN_C_26 model, the performance is 1.02% different. However, the proposed model has 95% fewer parameters than the comparative model. It is a lightweight improved model through effective feature extraction. During data augmentation, color and mixup play an important role in improving performance. From the first proposed model (1st=True, 2nd=False, Color=True, Geo=False, Mixup=True), the color conversion shows a 6.7% performance difference, and mixup has an 8.9% difference in performance.

IV. Conclusion

The paper presents a novel approach to classifying the state of a target vehicle using aerial images captured over a wide road area. The collected drone image data serves as the input image, and a portion of the image is cropped based on the target vehicle, thereby increasing data diversity through the mixup method. The proposed vehicle state model comprises

Authorized licensed use limited to: University of Ulsan. Downloaded on June 14,2023 at 01:17:11 UTC from IEEE Xplore. Restrictions apply.

a WAFE and a DRM, which extract far-flung information between vehicles in a wide area and select features in any area using a deformable convolutional layer, respectively. The entire learning model contains 1.27M parameters and achieves a detection performance of 88.6% in the experiment. In the future, a vehicle state classification model that combines vehicle tracking will be developed.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the government(MSIT).(No.2020R1A2C200897212)

REFERENCES

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] Jiale Cao, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, Yanwei Pang, and Ling Shao. D2det: Towards high quality object detection and instance segmentation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11482–11491, 2020.

[6] Hwanjun Song, Deqing Sun, Sanghyuk Chun, Varun Jampani, Dongyoon Han, Byeongho Heo, Wonjae Kim, and Ming-Hsuan Yang. Vidt: An efficient and effective fully transformer-based object detector. *ArXiv*, abs/2110.03921, 2021.

[7] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *ArXiv*, abs/2004.10934, 2020.

[8] 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.

[9] Qing Tang and Kang-Hyun Jo. Unsupervised person re-identification via nearest neighbor collaborative training strategy. *2021 IEEE International Conference on Image Processing (ICIP)*, pages 1139–1143, 2021.

[10] Yixiao Ge, Dapeng Chen, Feng Zhu, Rui Zhao, and Hongsheng Li. Self-paced contrastive learning with hybrid memory for domain adaptive object re-id. *ArXiv*, abs/2006.02713, 2020.

[11] Yifu Zhang, Chunyu Wang, Xinggang Wang, Wenjun Zeng, and Wenyu Liu. Fairmot: On the fairness of detection and re-identification in multiple object tracking. *Int. J. Comput. Vis.*, 129:3069–3087, 2021.

[12] 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.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015.

[14] 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.

[15] 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.

[16] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. YOLO by Ultralytics, 1 2023.

[17] Bipul Neupane, Teerayut Horanont, and Jagannath Aryal. Real-time vehicle classification and tracking using a transfer learning-improved deep learning network. *Sensors (Basel, Switzerland)*, 22, 2022.

[18] Ling Huang, Hengcong Guo, Rong hui Zhang, Hai wei Wang, and Jianping Wu. Capturing drivers' lane changing behaviors on operational level by data driven methods. *IEEE Access*, 6:57497–57506, 2018.

[19] Benjamin Coifman and Lizhe Li. A critical evaluation of the next generation simulation (ngsim) vehicle trajectory dataset. *Transportation Research Part B-methodological*, 105:362–377, 2017.

[20] Abu Jafar Md Muzahid, Syafiq Fauzi Kamarulzaman, Md. Arafatur Rahman, Saydul Akbar Murad, Md Abdus Samad Kamal, and Ali H. Alenezi. Multiple vehicle cooperation and collision avoidance in automated vehicles: survey and an ai-enabled conceptual framework. *Scientific Reports*, 13, 2023.

[21] Hongyi Zhang, Moustapha Cissé, Yann Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *ArXiv*, abs/1710.09412, 2017.

[22] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv: Learning*, 2016.

[23] 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.

[24] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In-So Kweon. Cbam: Convolutional block attention module. In *European Conference on Computer Vision*, 2018.

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[26] 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.

[27] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated residual networks. In *Computer Vision and Pattern Recognition (CVPR)*, 2017.

Authorized licensed use limited to: University of Ulsan. Downloaded on June 14,2023 at 01:17:11 UTC from IEEE Xplore. Restrictions apply.