A 2-Stage Model for Vehicle Class and Orientation Detection with Photo-Realistic Image Generation

Youngmin Kim*, Donghwa Kang*, Hyeongboo Baek†
Dept. of Computer Science and Engineering Incheon National University(INU) Incheon, Republic of Korea
{winston1214, anima0729, hbbaek}@inu.ac.kr

2022 IEEE International Conference on Big Data (Big Data) | 978-1-6654-8045-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/BigData55660.2022.10020472

***Abstract*—We aim to detect the class and orientation of a vehicle by training a model with synthetic data. However, the distribution of the classes in the training data is imbalanced, and the model trained on the synthetic image is difficult to predict in real-world images. We propose a two-stage detection model with photo-realistic image generation to tackle this issue. Our model mainly takes four steps to detect the class and orientation of the vehicle. (1) It builds a table containing the image, class, and location information of objects in the image, (2) transforms the synthetic images into real-world images style, and merges them into the meta table. (3) Classify vehicle class and orientation using images from the meta-table. (4) Finally, the vehicle class and orientation are detected by combining the pre-extracted location information and the predicted classes. We achieved 4th place in IEEE BigData Challenge 2022 Vehicle class and Orientation Detection (VOD) with our approach. Our code and project material will be available at https://github.com/inu-RAISE/VOD_Challenge.**

***Index Terms*—Object Detection, Classification, Image-to-Image Translation, GAN**

I. Introduction

A large amount of data is required to train using a Transformer [1] or Convolution Neural Networks (CNN) [2] based model, which is in the spotlight in the field of computer vision. In addition, to apply this model to the real world using this model, it is necessary to collect and construct data from the real environment. However, this work has a challenge because it takes a lot of time and cost. To solve this challenge, we study using virtual synthetic data, which is relatively inexpensive, and then study to improve the model that can operate in a real-world driving environment.

Our proposed approach follows: First, a table containing the object information of the image is built, and the location information of the object is extracted. After that, the styles of the synthesized object images are transformed into photo-realistic images. Using the transformed photo-realistic images and synthetic images, the vehicle class and orientation are classified, a nd l ocation i nformation i s c ombined t o achieve detection in real-world images. In this way, we achieved an accuracy that is about $8\%p$ higher than that of the existing object detection model trained only on synthetic data.

*Co-leading authors.
†Corresponding author.

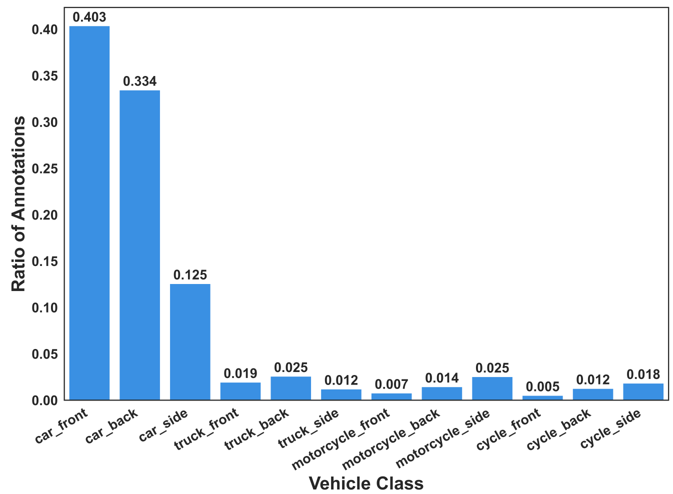

Fig. 1. Number of annotations by class

This paper is summarized as follows:

- We propose a two-stage model that detects the class and orientation of a vehicle by building a meta-table containing information about each object in images.
- We propose a method for better prediction in the real world with photo-realistic image generation.

In this paper, we introduce our proposed approach to detect vehicle class and orientation, and the rest of the paper is organized as follows: In Section 2, we survey synthetic datasets and related studies on object detection. In Section 3, we introduce our detailed approach. We share our experimental environment and experimental results in Sections 4 and 5, respectively. Finally, we conclude our paper in Section 6 by discussing our findings and the limitations of our paper.

II. Related Works

A. Synthetic Dataset

Collecting labels that fit images and objects in the real world is an expensive and challenging process in object detection using deep learning. For this reason, a relatively inexpensive synthetic dataset is used to collect a relatively inexpensive dataset and perform object detection. Kumar *et al*. [3]–[5] collected a dataset that includes vehicle class and orientation and location information to identify traffic flow. This image dataset has a total of 63,066 synthetic images for training

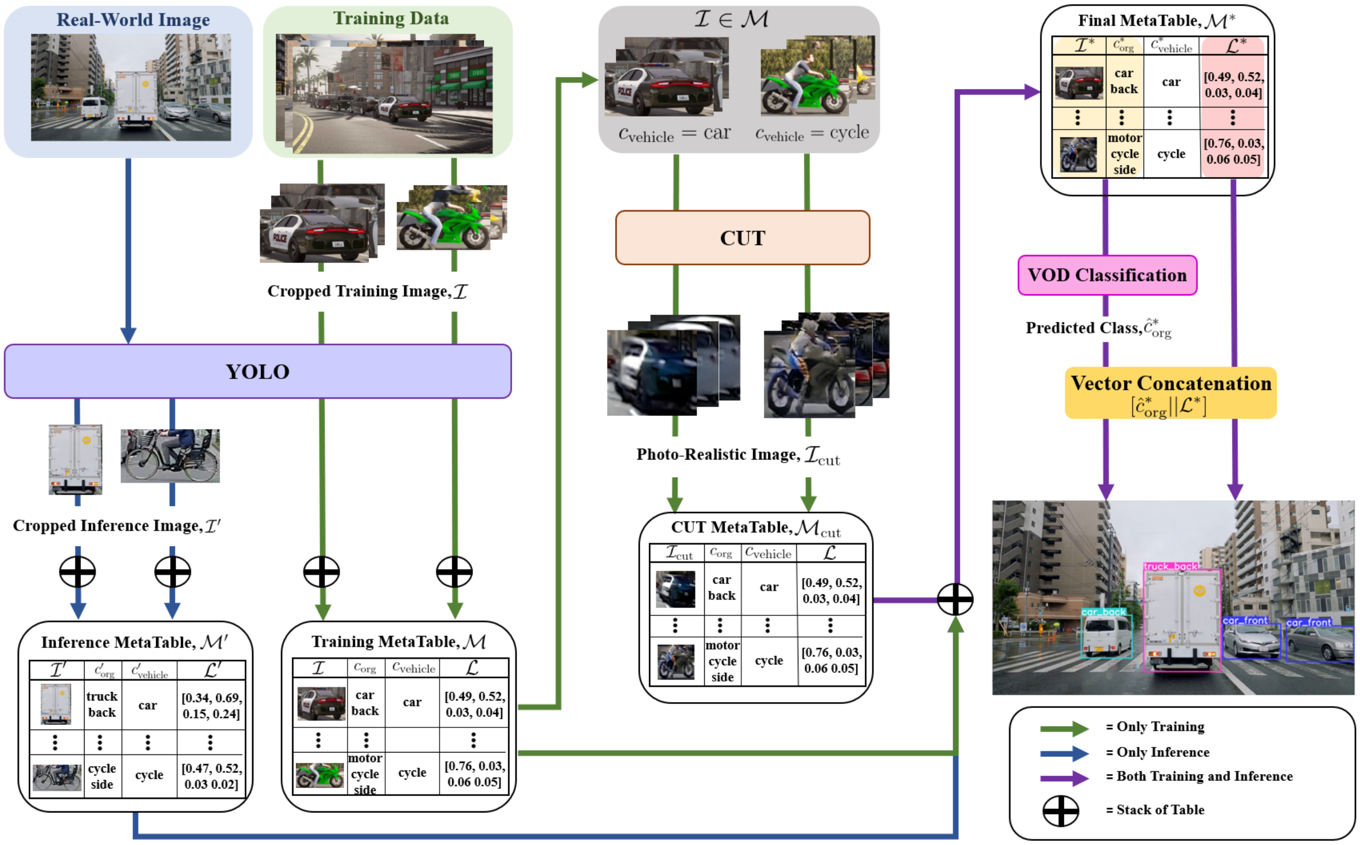

Fig. 2. Overview of our proposed VOD approach.

at 1920 x 1080 resolution. Also, a total of 3,000 real-world images were provided for testing at 1920 x 1080 and 1280 x 720 resolutions, a number of 1,500, respectively. There are 12 classes in the dataset, consisting of four types of vehicles (car, truck, motorcycle, cycle) and three orientations (back, front, side). The number of annotations for each class is shown in Fig. 1.

The FCAV dataset [6] is an image of a video game-based simulation engine as virtual data, and was created based on GTA V [7]. This dataset contains images of various situations because it was collected assuming driving in 4 weather conditions (sun, fog, rain, haze) and 4-time ranges (day, night, morning, dusk). There are a total of 205,879 training datasets for FCAV, consisting of three classes (person, car, and motorbike). For each class, there are 58 objects for a person, 1,125,624 objects for a car class, and 15,902 objects for a motorbike. We use only the training dataset, and the person class is excluded.

B. Object Detection

The object detection task is to detect object instances of a specific class in images to perform both class classification and object localization. The methods for object detection can be categorized into two main types: There is a single-stage model that performs class classification and object localization simultaneously, and a two-stage model proceeds with class classification after object localization. The single-stage model has a relatively fast inference speed, and representative models include SSD [8] and YOLO [9]. On the other hand, the two-stage model has relatively high performance, and the representative model is the RCNN series [10]–[12].

We used YOLOv5[1], the latest version of YOLO, which has high accuracy and fast inference time on the COCO dataset [13].

III. Proposed Approach

To avoid learning bias due to imbalanced classes in the training data set, we used a method to solve the problem by separating it into regional proposals and vehicle class classification. Also, we transform the synthetic image into a photo-realistic image to make better predictions from real-world images. For this, we crop the image by object using the label of the training data and define training meta-table, $\mathcal{M} = \{\mathcal{M}_1, \cdots, \mathcal{M}_n\}$, with information about the number of n objects. It is built with the object's images dictionary, $\mathcal{I} = [\mathcal{I}_1, \cdots, \mathcal{I}_n]$, classes dictionary, $\mathcal{C} = [\mathcal{C}_1, \cdots, \mathcal{C}_n]$, and location dictionary, $\mathcal{L} = [\mathcal{L}_1, \cdots, \mathcal{L}_n]$, according to $\mathcal{I}, \mathcal{C}, \mathcal{L} \subset \mathcal{M}$.

Our approach using meta-table is illustrated in Figure 1 and summarized in the following four steps. 1) It stacks the image,

[1]https://github.com/ultralytics/yolov5

Fig. 3. Examples of transform result from synthetic image to photo-realistic image

class, and location information of the object extracted from the training image in the meta-table. 2) Transform synthetic images to photo-realistic images and creates a new meta-table, $\mathcal{M}_{\text{cut}}$. 3) The image classification model classifies the class and orientation of the vehicle by the image and class in the final meta-table $\mathcal{M}^*$, which combines $\mathcal{M}_{\text{cut}}$ and the meta-table $\mathcal{M}$ of the training data. 4) Finally, the vehicle class and orientation are detected by combining the predicted class and the location information of the $\mathcal{M}^*$.

A. Meta-table Representation

1) Training: It builds the training image dictionary, $\mathcal{I} = [\mathcal{I}_1, \cdots, \mathcal{I}_n]$, location dictionary of training data, $\mathcal{L} = [\mathcal{L}_1, \cdots, \mathcal{L}_n]$, and class dictionary, $C = [\mathcal{C}_1, \cdots, \mathcal{C}_n]$, of the object with the given label information of the training dataset. First, object images obtained by cropping only the object from the training datasets with the given normalized object location information $((x, y, w, h) \in \mathcal{L})$ stacked on $\mathcal{I}$. Then, after dividing the class dictionary into 12 classes, c_{org}, including the class and direction of the vehicle and the class, $c_{vehicle}$, defined as the car or motorbike, stack them in C. It follows $\mathcal{C}_i = [c^i_{\text{org}}, c^i_{\text{vehicle}}] \in \mathcal{C}$, where i means i-th. In summary, i-th $\mathcal{M}$ follows:

$$\mathcal{M}_i = [\mathcal{I}_i, x_i, y_i, w_i, h_i, c^i_{\text{org}}, c^i_{\text{vehicle}}] \tag{1}$$

The FCAV dataset [6] is also stacked in M after constructing a dictionary in the same way. Using the $\mathcal{M}$ configured in this way, YOLOv5, an object detection model, is trained. At this time, we use c_{vehicle}, which is binary class information, not c_{org}, which contains all class information. As shown in Fig.1, since the training dataset has an imbalanced distribution, we do not predict objects as binary classes but instead, predict one class each. i.e., object detection is performed about cars and motorcycles, respectively. Through this, it is possible to more accurately predict the location information about cycles and motorcycles with a relatively small number of objects, which is a method of resolving the data imbalance.

2) Inference: Unlike the training process, the inference process cannot directly build meta-table because there is no provided location information and class information about objects. Therefore, by distinguishing the car and motorcycle classes, the pretrained YOLOv5 model predicts the class and location of objects for each of the two classes. The new inference dictionary $\mathcal{L}'$ and $\mathcal{C}'$ are represented based on class and location information predicted by YOLOv5. At this time, $\mathcal{C}'$ is represented only by binary class c'_{vehicle}, unlike the training process. Then, the image is cropped by objects using $\mathcal{L}'$, a location information dictionary, and a dictionary $\mathcal{I}'$ is made from the cropped images. In summary, the meta-table in the inference process, $\mathcal{M}'$ follows:

$$\mathcal{M}' = [\mathcal{I}', \mathcal{L}', c'_{\text{vehicle}}] \tag{2}$$

B. Synthetic Image to Photo-Realistic Image

There is a clear difference between real-world data and artificially created synthetic data. For this reason, training a model with synthetic images does not apply well to real-world images. Therefore, we help train the classification model through image-to-image translation that transforms synthetic images into a photo-realistic images.

Image-to-image translation model, such as Pix2Pix [14], uses a training set of aligned image pairs to learn mappings between input and output images. However, collecting a training set of image pairs is very difficult and costly. So we used an image-to-image translation model that can be used even on unpaired image training sets. Among the unpaired image-to-image translation models, we used CUT [15], which achieved a state of the arts (SOTA) in self-supervised representation learning. We transformed the $\mathcal{I}$ in $\mathcal{M}$ into photo-realistic images using CUT. Examples of the results are shown in Fig. 3. A new image dictionary $\mathcal{I}_{\text{cut}}$ is built as images transformed into photo-realistic images, and $\mathcal{M}_{\text{cut}} = \{\mathcal{M}^1_{\text{cut}}, \cdots, \mathcal{M}^n_{\text{cut}}\}$ is represented as class, $\mathcal{C}$, and location information, $\mathcal{L}$ of $\mathcal{M}$ in which each image, i.e. $\mathcal{I}_{\text{cut}}$ and $\mathcal{I}$ are matched. The meta-table $\mathcal{M}_{\text{cut}}$ that contains the information of the image converted to the style as real-world follows:

$$\mathcal{M}^i_{\text{cut}} = [\mathcal{I}^i_{\text{cut}}, \mathcal{C}^i, \mathcal{L}^i] \tag{3}$$

C. Vehicle Class and Orientation Detection

We use meta-table to detect the class and orientation of the vehicle. The final meta-table, $\mathcal{M}^* = \{\mathcal{I}^*, c^*_{\text{org}}, c^*_{\text{vehicle}}, \mathcal{L}^*\}$ is represented by combining $\mathcal{M}$ and $\mathcal{M}_{\text{cut}}$ during the training process. Likewise in the inference process, $\mathcal{M}^*$ is expressed as a combination of $\mathcal{M}'$ and $\mathcal{M}_{\text{cut}}$. We use EfficientNet [16] with high accuracy in the image classification task to classify vehicle class and orientation. This model is a stable image classification model that maintains high accuracy in various datasets and is the base of many states the of art models. Therefore, EfficientNet predicts 12 classes after that and combines the predicted class, c^*_{org}, with location information, $\mathcal{L}^*$, to detect the class and orientation of the vehicle. EfficientNet is divided into b0 to b7 according to the number of parameters. As the number increases, the number of parameters in the model

(a) **(b)**

Fig. 4. Examples of predicted vehicle class and direction detection result. (a) is the result images for the first test set and (b) is the result images for the second test set.

increases. We focused on improving classification accuracy by choosing b7, which has high accuracy but a slow processing speed. In addition, to prevent overfitting and improve accuracy, the final classification architecture was designed through an ensemble process 5 times.

IV. Experiments Setup

We used YOLOv5, CUT, and EfficientNet models to perform VOD, and the hyperparameters for each model are varied. First, YOLOv5 fine-tuned the YOLOv5x6 model, a pretrained weight on the COCO dataset. Furthermore, we set the image size to 640 x 482, batch size to 16, and epoch to 20. Next, since the size of the predicted object or a given object varies in the CUT, the image size is set to 256 x 256, the epoch is set to 100, and it is trained from scratch without using a pretrained model. Finally, a model pretrained with ImageNet is used, and the image is augmented with rotation and noise to prevent overfitting due to many parameters. The image size is set to 384 x 384 to proceed with training and inference. All our experiments were performed in an environment with two A6000 GPUs.

V. Results

We report the performance of our approach on the VOD challenge evaluation system, with metrics as follows:

$$WmAP = \sum_{k=1}^{12} w^k \times AP;\ where, \sum_{k=1}^{12} = 1 \tag{4}$$

$$score = 0.45 \times \underset{v_1}{WmAP} + 0.55 \times \underset{v_2}{WmAP} \tag{5}$$

This metric derives the final score by giving different weights for each class and data set. In this Table I, we list the results of comparing the prediction made with vanilla YOLOv5 and our method. As shown in Table I, our approach proved to be better than the prediction with YOLOv5 trained on synthetic data only. Since YOLOv5 was trained only on synthetic data, the model could not predict well on real-world images. Moreover, since localization and classification of 12 classes were performed simultaneously, the model had no choice but to solve complex problems and was difficult to predict. On the other hand, our approach consists of two-stage and separates localization and classification so that the model can solve problems easily. In addition, it was transformed into a photo-realistic image and trained to make predictions in real-world images. Examples of the predicted class and bounding box using our approach is shown in Fig. 4.

TABLE I
THE PERFORMANCE ON IEEE BIGDATA 2022 VOD CHALLENGE TEST SET

Method	WmAP(Test 1)	WmAP(Test 2)	Weighted Average Score
YOLOv5	0.282	0.285	0.284
Ours	**0.374**	**0.357**	**0.365**

VI. Conclusion

We proposed a method to detect the class and orientation of the vehicle in images of a real-world using synthetic data. To train by dividing into 2-stage model, we stacked each object's region, label, and image information. We represented them in a meta-table and tried to solve the dataset imbalance by simplifying the class to predict the region information of the objects. Furthermore, our model can better predict and detect images in real-world data by converting the object images in the meta-table into photo-realistic images. We have demonstrated that our approach is better than prediction with vanilla YOLOv5 and that synthetic images alone can make good enough predictions. The main limitation of our approach is the need for a more accurate prediction of the car's orientation. However, our approach can provide insights that can improve accuracy with feature adaptions.

Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF-2022R1A4A3018824).

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[3] A. Kumar, T. Kashiyama, H. Maeda, and Y. Sekimoto, "Citywide reconstruction of cross-sectional traffic flow from moving camera videos," in *2021 IEEE International Conference on Big Data (Big Data)*, pp. 1670–1678, IEEE, 2021.

[4] A. Kumar, T. Kashiyama, H. Maeda, H. Omata, and Y. Sekimoto, "Citywide reconstruction of traffic flow using the vehicle-mounted moving camera in the carla driving simulator," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2292–2299, IEEE, 2022.

[5] A. Kumar, T. Kashiyama, H. Maeda, H. Omata, and Y. Sekimoto, "Real-time citywide reconstruction of traffic flow from moving cameras on lightweight edge devices," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 192, pp. 115–129, 2022.

[6] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 746–753, IEEE, 2017.

[7] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, "Playing for data: Ground truth from computer games," in *European conference on computer vision*, pp. 102–118, Springer, 2016.

[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, pp. 21–37, Springer, 2016.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[11] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, pp. 1440–1448, 2015.

[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[13] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.

[14] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

[15] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive learning for unpaired image-to-image translation," in *European conference on computer vision*, pp. 319–345, Springer, 2020.

[16] M. Tan and Q. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," in *International conference on machine learning*, pp. 6105–6114, PMLR, 2019.