Helmet Rule Violation Detection for Motorcyclists using a Custom Tracking Framework and Advanced Object Detection Techniques

Viet Hung Duong [*] Quang Huy Tran [*†] Huu Si Phuc Nguyen [*]
Duc Quyen Nguyen Tien Cuong Nguyen
VNPT AI, VNPT Group
{hungdv,tqhuy,phucnhs,quyennd2000,nguyentiencuong}@vnpt.vn

Abstract

The use of helmets by motorcyclists is an effective way to reduce the risk of head injuries and fatalities in case of accidents. However, many countries still face the challenge of enforcing the helmet rule and ensuring compliance among riders. In this paper, we propose a novel framework that can differentiate between the driver and passengers and detect helmet rule violations for each rider empowered by computer vision and deep learning techniques. In the real-world scenario, there are many small and obstacle objects in each frame, which is a significant challenge, even with state-of-the-art detectors. To address this challenge, we employ an additional head detection module and a custom tracking algorithm that leverage auxiliary information such as moving direction, to improve detection performance on small and obstacle objects. This solution results in a significant improvement of 16% on mAP. Our complete framework achieves a final score of 69.97% on the 2023 AI City Challenge - Track 5 [18] and ranks third among the competing teams.

1. Introduction

Motorcycles have become a widely used form of transportation due to their affordability and ease of use [21]. According to the World Health Organization, wearing a helmet can reduce the risk of head injury by 69% and death by 42% in the event of a crash [3]. Despite the imposing of laws requiring helmet use, compliance remains low, particularly in developing countries [1, 2, 28]. Thus, developing automatic helmet detection systems to ensure traffic safety and reduce the cost of law enforcement is of utmost importance. To this end, numerous systems have been developed for monitoring motorbikes and detecting violations of the helmet rule. For instance, previous works in [6, 7] have shown the early success of automated helmet violation detection systems by utilizing Computer Vision and Deep Learning techniques to help to solve the problem of helmet law violations.

In the last few years, with the booming development of Artificial Intelligence and Deep Learning, many state-of-the-art object detection techniques have been proposed and achieved high accuracy across numerous tasks [5, 11, 12, 16, 22, 22, 24–26]. These techniques help to tackle challenging problems of Traffic Management Systems, including detecting and counting vehicles for density estimation, speed estimation, vehicle classification, and violation detection [9, 17, 29]. However, current methods still face some common challenges in real transportation scenarios such as class imbalance, occlusion, and viewing perspective, which exist in the 2023 AI City Challenge - Track 5 problem.

In this paper, we propose a novel framework to detect helmet rule violations for motorcyclists ap-

[*] Equally-contributed authors.
[†] Corresponding author.

plied to the 2023 AI City Challenge - Track 5. As illustrated in Fig.1, our proposed system consists of three main modules as follows:

- Object detection module: Two detection modules are proposed, with one module localizing human heads, and the other detecting motorbikes and their corresponding drivers and passengers along with information regarding whether they are wearing helmets. Within each module, we opt for ensemble techniques, then a boxes-fusion algorithm combines outputs from sources to produce the final result.
- Object association module: Since the detection modules identify objects such as head, motorbike, driver, and passenger, independently, we develop an object association module to attach them together. This module is also responsible for counting the number of humans on the motorcycle.
- Post-processing for tracking module: This module deals with the challenge of correctly detecting far and small objects by re-attaching the small object's class. Due to the similarities between classes and low resolution, common Re-Identification (ReID) [34] methods are incapable of delivering high detection accuracy. Thus, we propose a post-processing technique for the tracking algorithm, which utilizes object attributes such as vehicle direction, and the number of passengers to boost the overall accuracy from 53.95% to 69.97%.

2. Background

2.1. Object detection

Object detection algorithms can be categorized into two main groups: two-stage and one-stage methods. Two-stage methods, such as Mask R-CNN [12], R-CNN [11], FPN [16] and Fast R-CNN [10], first generate region proposals and then classify the proposals as objects or backgrounds. One-stage methods, such as YOLO [5, 24–26], and EfficientDet [32], directly predict the class and location of objects in a single shot. Both methods have their strengths and weaknesses and have been improved significantly over the years. Additionally, recent advancements in object detection including the use of attention mechanisms, transformer-based models, and efficient backbones have been studied and obtained promising results.

2.2. Multiple object tracking

Multi-object tracking (MOT) aims to associate detected objects across video frames. MOT algorithms typically consist of two main components: detection and association. Detection is the process of finding the objects in each frame, while the association process links the detection results across frames to form consistent trajectories. Simple Online Real-time Tracker (SORT) [4] is a simple tracking approach based on the Hungarian Algorithm [14] for data association and utilizes the Kalman Filter [13] to fuse the location of the predicted tracklets and the detection boxes to improve the results. In our system, we exploit the SORT tracking algorithm by extending its features, including motorbike direction. These features help our post-processing for tracking module to accurately reassign passenger positions.

3. System Architecture

Our proposed system processes video streams frame-by-frame through three components, including Object detection, Object association, and Post-processing for tracking module as depicted in Fig. 1. The object detection component is responsible for detecting all necessary objects in each frame, while the object association component connects each driver/passenger to the corresponding motorcycle and identifies the number of humans on the motorcycle. Last but not least, we design a post-processing for tracking approach to utilize object information to accurately reassign human classes, resulting in significant improvement in overall system performance.

3.1. Object detection module

This module consists of two models. The first model is Helmet Detection for Motorcyclists to detect 7 different object classes including motorbike,

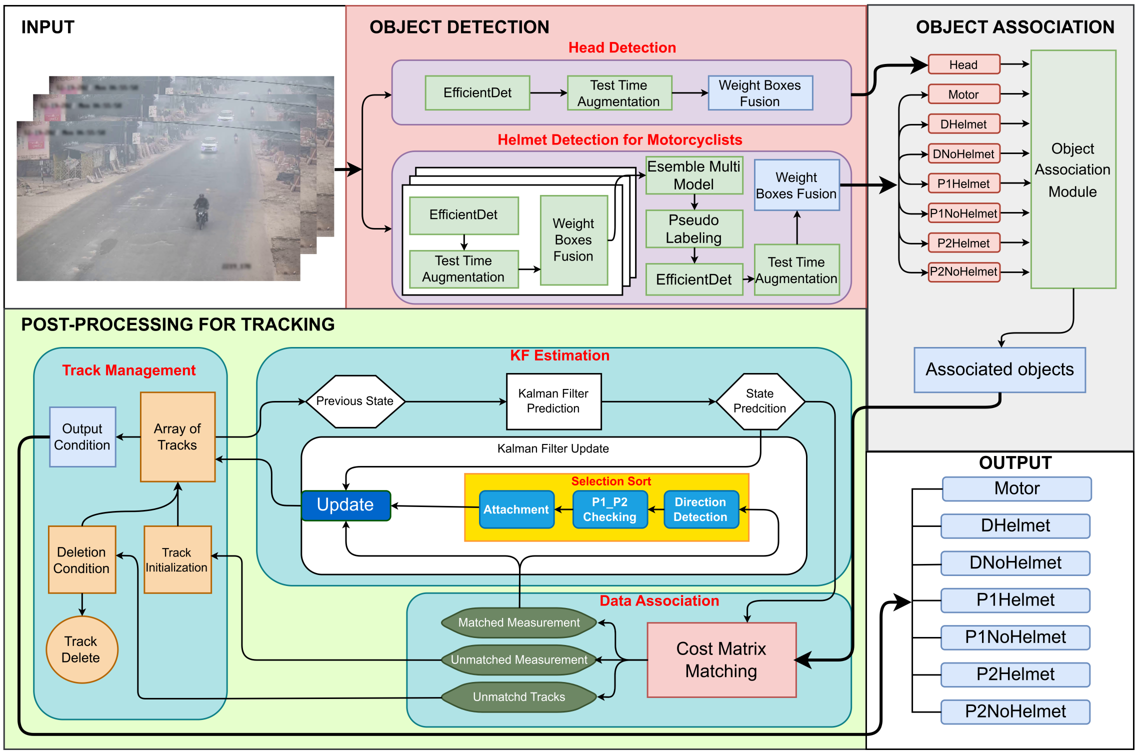


Figure 1. System Architecture. The object detection block comprises two modules for detecting the heads and helmets of motorcyclists. The object association block associates the outputs from these models to match their corresponding motorbikes with head and human objects. Subsequently, the post-processing for tracking block tracks all the motorbikes and employs our extension block, the Selection Sort (the yellow block), to reassign the corrected class for each human's box on those motorbikes before returning the final results.

driver, and passenger. However, there is a class imbalance problem that results in poor detection performance in minor classes. Therefore, we use the second model, which is Head Detection to detect the head of each rider and utilize its output as additional information for the first model, which helps to improve the overall detection performance.

Helmet Detection for Motorcyclists. We use EfficientDet [31], a scalable and efficient object detection framework, for our task. It uses EfficientNet [31] as the backbone network, a bi-directional feature pyramid network (BiFPN) to fuse and enhance features, and a compound scaling method to scale all network components. It is one of the state-of-the-art models on the COCO dataset [15]. We detect 7 classes: Motorbike, DHelmet, DNoHelmet, P1Helmet, P1NoHelmet, P2Helmet, and P2NoHelmet. We experiment with the three largest variants of EfficientDet (D5, D6, and D7) and different input scales (512 to 1024) to obtain the best performance.

Data augmentation. To improve the accuracy of our predictions during inference, we utilize Test-Time Augmentation [27]. This technique enables the model to detect objects from various perspectives, which increases its ability to identify objects that may not be visible from certain angles.

Assembling predicted boxes and pseudo-labeling. We use models ensembling and pseudo-labeling to improve the performance and generalization of EfficientDet D6. Models ensembling combines multiple models to reduce variance and bias, increase robustness and stability, and handle complex and large-scale data. We compare two ensembling techniques: Non maximum suppression (NMS) [19] and Weighted boxes fusion (WBF) [30], and find that WBF is superior (see Table 3). Pseudo-labeling is a semi-supervised technique that uses labeled data to annotate unlabeled data and then trains the model with those newly generated labels. Pseudo-labeling enables the model to exploit a larger amount of data. We combine ensemble and pseudo-labeling techniques to enhance our model's accuracy and generalization. We ensemble the results from the best models to generate pseudo labels, then train an EfficientDet model for a few epochs with these pseudo labels and select it as our final model. To avoid eliminating rare objects, we use a very low threshold with WBF.

Head detection. We adapt EfficientDet for training the head detection model, as it demonstrates excellent performance for Helmet Detection for Motorcyclists. Due to time constraints, we opt not to incorporate pseudo-labeling and instead train only one head detection model. During the inference phase, we utilize Test-Time Augmentation (TTA) and Weighted boxes fusion (WBF), as discussed previously, to generate the final predictions for the model.

3.2. Object association module

As described in previous sections, the object detection module comprises two models that output all detected objects, including Head, Motorbike, DHelmet, DNoHelmet, P1Helmet, P1NoHelmet, P2Helmet, and P2NoHelmet. However, these objects are independent, making it challenging to exploit their relational information. To address this, we have devised an additional module for object association to group related objects together, thereby enabling the management of a single tracking ID for each group. On top of that, the information on the relationship between objects is also employed to enhance the accuracy of the post-processing for tracking modules. In particular, after assigning the output objects from detection modules to motor objects (if their class is 'motorbike') and head objects (if their class is 'heads') or human objects (for all other classes), the object association module identifies all possible pairs of human-motor and human-head and links them together. This is performed by calculating the overlap areas and relative positions of the bounding boxes with respect to the motorbikes. As a result, the output is a list of motorbikes attached with their corresponding humans and heads, which serves as the input for the subsequent module to improve detection results.

3.3. Post-processing for tracking module

Accurately detecting the number of people on a motorbike is a challenging problem for object detection models, particularly when the number exceeds two. The reason is that when the motorcycle approaches the camera at a complete angle, distinguishing between individuals on that motorcycle becomes more difficult. Furthermore, due to the limited number of instances of Passenger 1 (P1) and Passenger 2 (P2) in the training dataset (approximately 5500 instances for P1 and only 70 instances for P2, see table 2), these classes are often misclassified as the driver (D) when the model localizes only one human associated with the motorbike. This misclassification significantly reduces the accuracy of the detection results. To address this issue, we propose a novel post-processing for tracking module that corrects misclassified cases in the aforementioned classes while retaining the Helmet or NoHelmet class obtained from the object detection model. Our proposed module uses a SORT-based algorithm, which is illustrated in the post-processing for tracking block as shown in Fig. 1. The Track Management and Data Association [20] steps in this block remain the same as in the SORT algorithm. Additionally, we have integrated the Kalman Filter (KF) Estimation combined with our extension algorithms (the Selective Sort module) to improve detection results. In particular, not only updating the object's id, but our Selective Sort module also enables to update object's additional attributes in each frame for reassigning the human's position on motorbikes. Specifically, the process details are as follows:

(1) Direction detection: Figure 2 illustrates motorbike's direction. In particular, algorithm 1

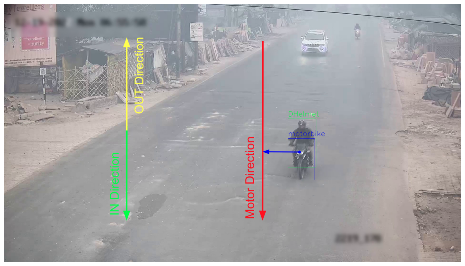


Figure 2. Illustration of the motorbike detection. The direction of the motorbike is assigned as IN direction (green arrow) or OUT direction (yellow arrow) according to the motorbike's position throughout its appearance in the video.

Algorithm 1: Direction Detection

```
Input  : $centers$: list of center points'
         coordinator for 1 motorbike
Output: 1 (IN) / 0 (OUT) / $None$:
         direction of motorbike
$checks \leftarrow []$;
for $i$ in [1: $len(centers)$] do
    $checks$.append($centers_i$ - $centers_{i-1}$);
    if $len(checks) \geq 3$ then
        break;
end
if $len(checks) < 3$ then
    return $None$
else
    $num_T \leftarrow count\_true\_in\_checks$;
    $num_F \leftarrow count\_false\_in\_checks$;
    return $num_T > num_F$
```

checks the coordinates of the motorbike between the current and previous frames to determine its direction (d). If the motorbike moves towards the camera, the direction will be 1 (IN direction). If the motorbike moves away from the camera, the direction will be 0 (OUT direction). The algorithm continues to execute until the direction is determined.

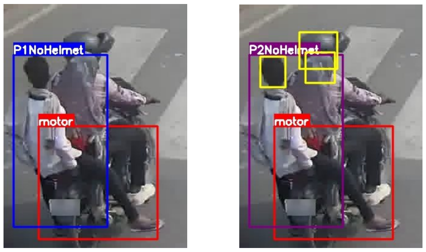
(2) P1_P2_Checking: This algorithm is designed to determine the number of passengers on motorbikes. Algorithm 2 counts the number of human heads presented on each motorbike in each frame. If a motorbike has 3 heads, it is identified as having two passengers. Similarly, if a motorbike has 2 heads, it is identified as having one passenger.

(3) Finally, algorithm 3 reassigns the correct class for human objects based on results from the P1_P2_Checking algorithm and direction detection algorithm. First, human boxes are sorted by their coordinates to get their relative position with respect to the motorcycle. Subsequently, they are assigned to the appropriate class as designated in algorithm 3. Note that once the direction of the motorbike is accurately detected and identified if P1 or P2 is on that motorbike, the reassignment of classes based on the factors mentioned above will be carried out throughout the entire duration of that motorbike ID in the video.

Figure 3 depicts examples of the reassignment process. One of the most challenging misdetection cases occurs when three humans sit on a motorbike in the OUT direction while these humans overlap and occlude with each other illustrated in Fig. 3a. In this case, the detection module detects only one human on the motorbike and misclassifies him as P1NoHelmet (the blue box) while there are actually one driver and two passengers. To address this issue, our post-processing for tracking module reassigns the human position from P1 to P2 while preserving the helmet attribute (Helmet and NoHelmet) from the detection results. The reassignment process involves several algorithms. First, algorithm 1 identifies the motorbike direction (OUT direction in this example), while algorithm 2, based on the head attributes associated with the motor, determines that P2 is on the motorbike as depicted in Fig. 3b. Finally, algorithm 3 reassigns the corrected class P2NoHelmet (h_1'), the purple box, for the misclassified box P1NoHelmet (h_1), according to sub-table P2 in table 1. In this case, the algorithm corresponds with the ”1 human” column as only one human (P1NoHelmet) is detected.

Table 1. Class transform on different cases. d is the direction of motorbike, h_i human with original class from the detection module, h'_i human after being assigned class. The class values are defined in table 2

	P1			P2					
	1 human	2 humans		1 human	2 humans		3 humans		
d	$h_1 \to h'_1$	$h_1 \to h'_1$	$h_2 \to h'_2$	$h_1 \to h'_1$	$h_1 \to h'_1$	$h_2 \to h'_2$	$h_1 \to h'_1$	$h_2 \to h'_2$	$h_3 \to h'_3$
0	2,6 4 3,7 5	4,6 2 5,7 3	2,6 4 3,7 5	2,4 6 3,5 7	- -	2,4 6 3,5 7	2,4 6 3,5 7	2,6 4 3,7 5	4,6 2 5,7 3
1	4,6 2 5,7 3	2,6 4 3,7 5	4,6 2 5,7 3	- -	2,4 6 3,5 7	- -	4,6 2 5,7 3	2,6 4 3,7 5	2,4 6 3,5 7



(a) Detection output

(b) Final output

Figure 3. Illustration for reassignment process. Figures 3a and 3b depict examples of the OUT direction case.

Algorithm 2: P1_P2_Checking

```
Input  : $bbox\_head$: list of heads attached
         to 1 motorbike
         $P\_type$: $P1$ or $P2$
Output: P_type
if $P\_type$ = $P1$ then
    $num\_heads \leftarrow 2$;
else
    $num\_heads \leftarrow 3$;
for $heads$ in $bbox\_head$ do
    if $len(heads)$ = $num\_heads$ then
        $counter \leftarrow counter + 1$;
    if $counter$ = $3$ then
        return $P\_type$;
end
return $None$;
```

Algorithm 3: Attachment 1 motorbike on 1 frame

```
Input  : $center$: center point's coordinator
         $bbox\_head$: list of head's
coordinator
         $bbox\_human$: list of human body
coordinator and class
Output: Update class in $bbox\_human$
if $d$ is $None$ then
    update $d$ with algorithms 1
if $is\_P_1$ is $None$ then
    update $is\_P_1$ with algorithms 2
if $is\_P_2$ is $None$ then
    update $is\_P_2$ with algorithms 2
if $is\_P_2 \neq None$ then
    $bbox\_human \leftarrow$
      $sorted(bbox\_human)$;
    if $len(bbox\_human) = 1$ then
        update class according to table 1
    if $len(bbox\_human) = 2$ then
        update class according to table 1
    if $len(bbox\_human) = 3$ then
        update class according to table 1
else if $is\_P_1 \neq None$ then
    $bbox\_human \leftarrow$
      $sorted(bbox\_human)$;
    if $len(bbox\_human) = 1$ then
        update class according to table 1
    if $len(bbox\_human) = 2$ then
        update class according to table 1
```

4. Experiments

4.1. Dataset

Data preprocessing. Our work utilizes the dataset [23] provided by the 2023 AI City Challenge - Track 5, which contains 100 20-second videos recorded by 21 cameras in various locations across India. Each video is captured at Full HD resolution (1920 x 1080) and 10 fps, making a total of approximately 20000 images. Annotations are provided for each motorbike, driver, passenger, and the information regarding whether they are wearing helmets. As shown in Fig. 4, the dataset cannot be used straightaway, since some bounding boxes are mislabeled and incorrectly labeled. Thus, we perform a data-cleaning process to remove incorrect bounding boxes and annotate additional objects to increase the consistency between images. The distributions of the original and modified datasets are shown in Table 2.

Table 2. Distributions of the original and modified datasets

ID	Class	Number of Instances	
		Original	Modified
1	Motorbike	31135	36268
2	DHelmet	23260	25213
3	DNoHelmet	6856	8437
4	P1Helmet	94	132
5	P1NoHelmet	4280	5369
6	P2Helmet	0	0
7	P2NoHelmet	40	70

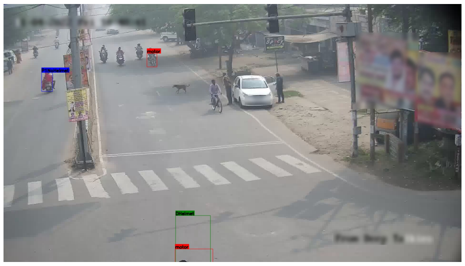


Figure 4. Mislabeled examples in the original dataset

Annotation. As presented in Table 2, there is barely any instance of the second passenger in the original dataset, and the number of samples of the passenger wearing a helmet is very small. Due to this class imbalance issue, it is difficult to generalize patterns of objects belonging to the P1Helmet, P2Helmet, and P2NoHelmet classes. Hence, we rely on additional algorithms to help our models detect instances of these classes. We annotate all human heads, which can be used to detect the number of passengers. The total number of human heads annotated is 103,929. We use labelImg [33] to manually label and correct flawed bounding boxes.

4.2. Implementation details

We use a COCO-pretrained model and fine-tune it on the AI City Challenge dataset for 50 epochs using Adam optimizer. The learning rate is set to 5e-4 and decayed by a cosine schedule. We run all experiments on one DGX node with 8 NVIDIA A100-40GB GPU.

We train the Helmet Detection for Motorcyclists model using a 5-fold cross-validation scheme on the dataset, where the folds are split by video id. We also vary the image size among 512, 640, 768, 896, 1024 and the model size among D5, D6, D7, leading to more than 15 models in total.

After the training process completes, we select the top 10 models and use them to generate pseudo labels on the test set by applying weighted boxes fusion with an NMS threshold of 0.5 and a score threshold of 0.25. We then fine-tune the EfficientDet D6 with the generated labels and image size of 640 for 20 epochs using a learning rate of $8e^{-5}$.

With the head detection model, we employ an EfficientDet D7 pre-trained on the COCO dataset. The input image size is set to 768 pixels. We set the learning rate to 5e-8, which is a small value that prevents overfitting and ensures convergence. We train the model for 10 epochs, which is sufficient to achieve good performance on our dataset.

4.3. Experiments Results

4.3.1 Evaluation metrics

The evaluation metric used for AICityChallenge - Track 5 is mean Average Precision (mAP), which calculates the area under the Precision-Recall curve over all the object classes. This measure is introduced in the PASCAL VOC 2012 competition [8].

4.3.2 Object detection models

Helmet Detection for Motorcyclists. We evaluate our proposed method, which is described in sections 4.2, on various combinations of model scales and image size scales.

To generate pseudo labels, we employ Weighted boxes fusion (WBF) to combine the outputs of our top 10 models. Using ensemble only, the model achieves 47.53% mAP. We use the pseudo labels to fine-tune an EfficientDet D6 model with an input image size of 768 pixels for a few epochs. We further improve our results by applying Test-time augmentation (TTA) and ensembling the predictions of our best models. Table 4 shows the difference in accuracy when using pseudo labeling. Our performance was 53.95% mAP on the full test set.

Table 3. Comparison of the performance of different ensemble methods on the training set of fold 1, following the experimental setup described in Section 4.2.

Method	WBF	NMS
mAP	44.46	40.72

Head detection. To perform head detection, we adapt the EffcientDet D7 architecture and train it on images of size 768x768 without using any pseudo labels. We employ Test-time augmentation (TTA) and Weighted box fusion (WBF) to improve the robustness and diversity of the predictions. We split 20% of the training dataset for validation and tune the score threshold in the interval of [0.1, 0.3] to optimize the mAP on the validation dataset, which reaches 62.6%. Our head detection model improves the performance of our previous model (Helmet Detection for Motorcyclists), which fails to detect cases of two passengers sitting behind a driver.

Comparison with other teams. We evaluate our solution on the Track 5 evaluation system. As shown in Table 5, our solution obtains 69.97% mAP and ranks 3rd among 30+ teams.

Table 4. Ablation study on impact of applied methods: Ensemble(Ens), Pseudo Labeling (Ps), and our Post-processing for Tracking (PPT) respectively. The first row is the baseline results from EfficientDet-D6 with image size of 768.

Ens	Ps	PPT	mAP
			44.09(baseline)
✓			47.53 (+3.44)
✓		✓	67.85 (+23.76)
✓	✓		53.59 (+9.5)
✓	✓	✓	**69.97 (+25.88)**

Table 5. Leaderboard of Track 5 in the AI City Challenge 2023.

Team ID	mAP
58	83.4
33	77.54
37 (Ours)	**69.97**
18	64.22
16	63.89

4.3.3 Post-processing for Tracking module

Table 4 compares the accuracy of different methods. Using the tracking module increases the accuracy by 16.02%, from 53.95% to 69.97% on the final leaderboard, outperforming methods using only ensemble and pseudo labeling. The tracking module proves to be effective in detecting difficult classes, which have very few samples in the training data and thus are hard to detect with Ensemble + Pseudo labeling method.

5. Conclusion

This paper presents a novel pipeline for Violation of Helmet Rule detection for Motorcyclists, which leverages the EfficientDet model to identify motorbike, driver, passenger, and head instances. Object association and post-processing for tracking modules are applied to enhance accuracy. Our method is evaluated on the AICityChallenge - Track 5 dataset and has ranked 3 with 69.97% mAP, demonstrating the effectiveness of our solution.

References

[1] A.M. Bachani, Y.W. Hung, S. Mogere, D. Akunga, J. Nyamari, and A.A. Hyder. Helmet wearing in kenya: prevalence, knowledge, attitude, practice and implications. *Public Health*, 144:S23–S31, 2017. Supplement: Global Road Safety: Monitoring Risks and Evaluating Programs. 1

[2] Abdulgafoor M. Bachani, Casey Branching, Chariya Ear, Douglas R. Roehler, Erin M. Parker, Sotheary Tum, Michael F. Ballesteros, and Adnan A. Hyder. Trends in prevalence, knowledge, attitudes, and practices of helmet use in cambodia: results from a two year study. *Injury*, 44:S31–S37, 2013. Global Road Safety: Updates from ten low- and middle-income countries. 1

[3] Liu BC, Ivers R, Norton R, Boufous S, Blows S, and Lo SK. Helmets for preventing injury in motorcycle riders. *The Cochrane database of systematic reviews: CD004333*, 2008. 1

[4] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, sep 2016. 2

[5] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *CoRR*, abs/2004.10934, 2020. 1, 2

[6] Aphinya Chairat, Matthew N. Dailey, Somphop Limsoonthrakul, Mongkol Ekpanyapong, and Dharma Raj K.C. Low cost, high performance automatic motorcycle helmet violation detection. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 3549–3557, 2020. 1

[7] Shuai Chen, Jinhui Lan, Haoting Liu, Chengkai Chen, and Xiaohan Wang. Helmet wearing detection of motorcycle drivers using deep learning network with residual transformer-spatial attention. *Drones*, 6(12), 2022. 1

[8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 7

[9] Ruben J Franklin and Mohana. Traffic signal violation detection using artificial intelligence and deep learning. In *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, pages 839–844, 2020. 1

[10] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 2

[11] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014. 1, 2

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 2

[13] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960. 2

[14] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 2

[15] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014. 3

[16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1, 2

[17] Vishal Mandal, Abdul Rashid Mussah, Peng Jin, and Yaw Adu-Gyamfi. Artificial intelligence-enabled traffic monitoring system. *Sustainability*, 12(21), 2020. 1

[18] Milind Naphade, Shuo Wang, David C. Anastasiu, Zheng Tang, Ming-Ching Chang, Yue Yao, Liang Zheng, Mohammed Shaiqur Rahman, Meenakshi S. Arya, Anuj Sharma, Qi Feng, Vitaly Ablavsky, Stan Sclaroff, Pranamesh Chakraborty, Sanjita Prajapati, Alice Li, Shangru Li, Krishna Kunadharaju, Shenxin Jiang, and Rama Chellappa. The 7th AI City Challenge. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2023. 1

[19] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *Proceedings of the 18th International Conference on Pattern Recognition - Volume 03*, ICPR '06, pages 850–855, Wash-

ington, DC, USA, 2006. IEEE Computer Society. 4

[20] Ricardo Pereira, Guilherme Carvalho, Luís Garrote, and Urbano J. Nunes. Sort and deep-sort based multi-object tracking for mobile robotics: Evaluation with new data association metrics. *Applied Sciences*, 12(3), 2022. 4

[21] Jakapong Pongthanaisawan and Chumnong Sorapipatana. Relationship between level of economic development and motorcycle and car ownerships and their impacts on fuel consumption and greenhouse gas emission in thailand. *Renewable and Sustainable Energy Reviews*, 14(9):2966–2975, 2010. 1

[22] Pulak Purkait, Cheng Zhao, and Christopher Zach. Spp-net: Deep absolute pose regression with synthetic views. *arXiv preprint arXiv:1712.03452*, 2017. 1

[23] Mohammed Shaiqur Rahman, Jiyang Wang, Senem Velipasalar Gursoy, David Anastasiu, Shuo Wang, and Anuj Sharma. Synthetic Distracted Driving (SynDD2) dataset for analyzing distracted behaviors and various gaze zones of a driver, 2022. arXiv:2204.08096. 6

[24] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1, 2

[25] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 1, 2

[26] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 1, 2

[27] Divya Shanmugam, Davis Blalock, Guha Balakrishnan, and John Guttag. Better aggregation in test-time augmentation, 2021. 3

[28] Felix Wilhelm Siebert, Deike Albers, U Aung Naing, Paolo Perego, and Chamaiparn Santikarn. Patterns of motorcycle helmet use – a naturalistic observation study in myanmar. *Accident Analysis Prevention*, 124:146–150, 2019. 1

[29] Felix Wilhelm Siebert and Hanhe Lin. Detecting motorcycle helmet use with deep learning. *Accident Analysis Prevention*, 134:105319, 2020. 1

[30] Roman A. Solovyev and Weimin Wang. Weighted boxes fusion: ensembling boxes for object detection models. *CoRR*, abs/1910.13302, 2019. 4

[31] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946, 2019. 3

[32] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. *CoRR*, abs/1911.09070, 2019. 2

[33] Tzutalin. labelimg labeling tool. 7

[34] Mang Ye, Jianbing Shen, Gaojie Lin, Tao Xiang, Ling Shao, and Steven C. H. Hoi. Deep learning for person re-identification: A survey and outlook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(6):2872–2893, 2022. 2