SNRU Journal of Science and Technology

Journal home page: snrujst.snru.ac.th

Fast Classifying Non-helmeted Motorcyclists by Using Convolutional Neural Networks

Kietikul Jearanaitanakij*, Karnnumart Iamthammarak, Nattakitt Wangcharoen
Department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, 10520 Thailand
***Corresponding Author**: kietikul.je@kmitl.ac.th

Received: 12 October 2020; **Revised:** 7 December 2020; **Accepted:** 9 December 2020; **Available online:** 1 January 2021

Abstract

According to statistics from ThaiRoads foundation, fifty percent of motorcyclists in Thailand ignore helmet while riding. This reckless behavior may lead to a severe injury when they encounter an accident. Most existing non-helmeted detection techniques analyze all moving objects in the video frame without ignoring unrelated pieces. They need to classify a large number of moving objects in the buffer, resulting in a delay in the execution time. We propose the fast system to accurately detect non-helmeted motorcyclists from the surveillance cameras by using the object filtering technique along with two convolutional neural networks (CNNs). The first CNN identifies motorcycles among filtered moving objects while the second CNN detects non-helmeted motorcyclists. The experimental results on the video stream dataset captured from surveillance cameras at King Mongkut's Institute of Technology Ladkrabang indicate that the proposed system not only improves an execution time but also produces a higher classification accuracy, comparing among other algorithms.

Keywords: Convolutional neural network; Classification; Deep learning; Road safety; Non-helmeted motorcyclist

1. Introduction

A helmet is an equipment that can protect motorcyclists from a serious head injury when they encounter an accident. According to the information from Thailand Development Research Institute (TDRI), there are about 20 million motorcycles in Thailand. Around 70% of traffic deaths are motorcycle accidents and the numbers rise steadily. Instead of being concerned about their safety, most Thai people wear helmets because of being afraid of getting fined by the police. Due to a limited number of the police officers who cannot monitor all motorcyclists in different areas, it would be a good idea to have an automated system that can identify riders who do not wear a helmet. This system not only reduces the officer's workload but also enforces a good habit for those motorcyclists to wear a helmet while driving.

There are efforts that use CNN to detect the helmet wearing. Vishnu et al. [1] extract the moving objects from the video frame and use the first CNN, i.e., Krizhevsky et al [2], to classify whether the object is a motorcycle. Afterward, another CNN will identify non-helmeted rider from the image which contains only the head region of a person. Mistry et al. [3] use person class detection instead of motorcycles to increase helmet detection accuracy. Their method captures the back of motorcyclists and reports the license plate number if there is no helmet detected. Raj et al. [4] use AlexNet to classify non-helmeted violations and LeNet to classify characters in the license plate. Although their system obtains good accuracy for violation classification, most errors in the system happen when the front rider wears a helmet, but the back person does not. Forero [5] uses image processing along with artificial intelligence techniques to detect non-helmeted motorcyclists. However, this approach requires many passive sensors to maintain high precision. Boonsirisumpun et al. [6] apply various pre-trained CNN models to identify non-helmeted riders. However, due to the small size of the image dataset, their system does not produce the optimal classification accuracy. Long et al. [7] use the extension of the Single Shot multi-box Detector (SSD) architecture to detect workers who wear a safety helmet. Their system outperforms the traditional SSD architecture for safety helmet wearing detection. Dasgupta et al. [8] implement automated helmet detection for multiple riders by using the YOLOv3 model to identify multiple riders. Their model works quite well for helmet detection in different scenarios with high accuracy. Khan et al. [9] utilize a transfer learning by using MobileNet-SSD with additional layers to classify helmet and non-helmeted motorcyclists. MobileNet-SSD is pre-trained on the Common Objects in Context (COCO) dataset. Siebert and Lin [10] develop helmet detection by using RetinaNet [11] which simultaneously performs object location and identification. Their experimental results reveal a fairly high accuracy. Kumar et al. [12] recently reported that using only one neural network per image to detect helmet (or non-helmet) outperforms R-CNN [13, 14] and Fast R-CNN [15] as they use multiple networks.

In this paper, we propose a non-helmeted motorcyclist detection system that takes the video stream from the surveillance camera. We apply the adaptive background subtraction [16] to capture all moving objects and use the object filtering technique to keep only motorcycle-like objects. Later, we use the first CNN to identify the motorcycle objects and the second CNN to detect non-helmeted riders. In contrast to other works, the proposed system filters unrelated objects, e.g., car, truck, bus, by comparing the dimension ratio of the object with a suitable threshold before entering the first CNN. The experiments are conducted on the video dataset captured from the surveillance cameras installed around the area of King Mongkut's Institute of Technology Ladkrabang. The experimental results indicate that the proposed system significantly saves an execution time of the system, yet still produces a high classification accuracy.

2. Materials and methods

Related CNN Architectures

Three CNN architectures related to our research are explained. 1) ResNet-v2 is an image classification model with various numbers of layers. It is an extension of the ResNet architecture in [17]. ResNet-152-v2 was introduced by He et al. [18] with the key difference, compared to [17], is to perform the batch normalization before each weight layer of the residual unit. The residual learning unit in ResNet diminishes the vanishing gradient of deep neural networks without increasing the complexity. Fig.1 shows the residual unit of ResNet-v2. 2) Inception-ResNet-v2 is a combination of the Inception module and the Residual structure [19]. One block of Inception-ResNet contains different sized convolutional filters which are merged by residual connections. The compress view of Inception-ResNet-v2 is shown in Fig.2. 3) Inception-v3 [20], an extension of GoogLeNet [21], is the first runner up of Image Classification in ILSVRC 2015. The traditional 7×7 convolution is factored into three 3×3 convolutions to significantly reduce the computational cost in the network. By using the grid reduction technique, the inception part is reduced the grid-size while expanding the filter banks as shown in Fig.3.

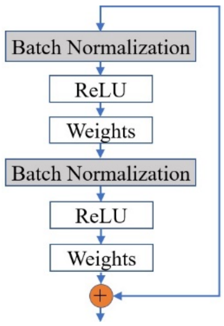

Fig. 1 Residual unit

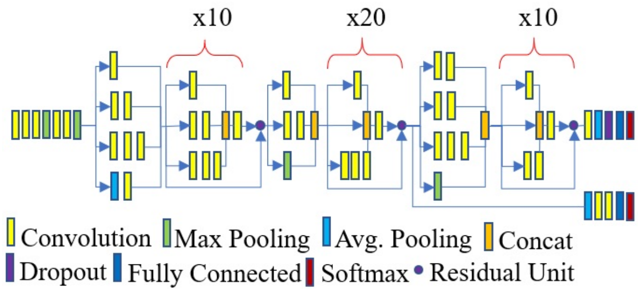

Fig. 2 Inception-ResNet-v2

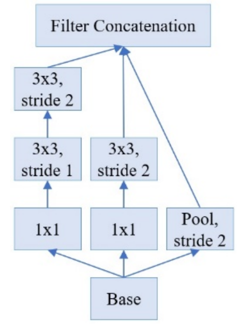

Fig. 3 Inception part

We employ ResNet-v2, Inception-ResNet-v2, and Inception-v3 as candidates in designing our system because they are successors of well-known CNNs proposed in ImageNet Large Scale Visual Recognition Challenge (ILSVRC). Substantial improvements have been made so that their top-5 accuracies are above 90% on the ImageNet dataset. Table 1 shows the comparison among three CNN architectures in various aspects.

Table 1 Comparison among three CNNs.

Architecture	Depth	No. of parameters	Size (MB)	Year
ResNet-v2	152	60M	232	2016
Inception-ResNet-v2	572	56M	215	2016
Inception-v3	159	24M	92	2015

Adaptive Background Subtraction

Background subtraction is a process of detecting the moving object which does not fit the statistical model of the scene. The scene without the moving objects exhibits the normal behavior which can be described by a probability density function. Part of the image that does not conform to the density function is considered as the moving object. Zoran [16] developed the adaptive background subtraction by using Gaussian mixture probability density. The number of components per pixel can be adaptively adjusted to the observed scene. The algorithm can reduce the processing time and improve the image segmentation. It is also robust to the change of illumination, noises, and high-frequency objects such as tree leaves and rain.

Proposed System

The proposed system can be described, along with a flowchart in Fig.4, as the following steps.

Step 1: Apply an adaptive background subtraction [16] to locate all moving objects.

Step 2: For each moving object, measure the vertical size (Y) and the horizontal size (X). If the dimension ratio Y/X is greater a certain threshold (Θ) then proceeds to the next step. Otherwise, ignore that object since it is unlikely a motorcycle.

Step 3: Classify the object by Inception-ResNet-v2 whether it is a motorcycle or something else, e.g., pedestrian, bicycle.

Step 4: Classify the motorcycle object in step 3 to identify non-helmeted riders by using 2nd CNN.

The filtering condition (Y/X > Threshold Θ) in step 2 significantly reduces the number of objects to be classified in step 3. As a result, the system can ignore irrelevant objects and improve execution time. The processes to determine all parameters and settings will be demonstrated later in this section.

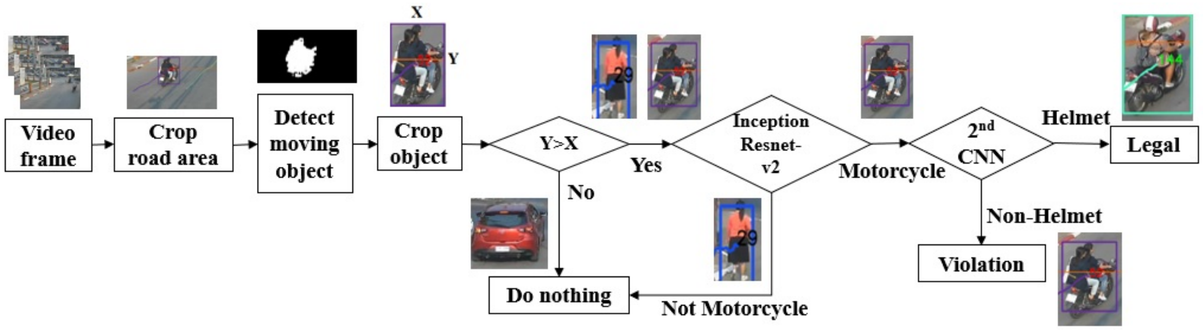

Fig. 4 Flowchart of the proposed system.

Comparative Method

Since the method in [1] will be mainly compared with the proposed system, its architecture is briefly described as follows. The method in [1] applies AlexNet [2] as the first CNN to classify whether the object is a motorcycle. Afterward, another CNN identifies non-helmeted rider from the image which contains only the head region of a person. The framework of [1] is illustrated in Fig.5.

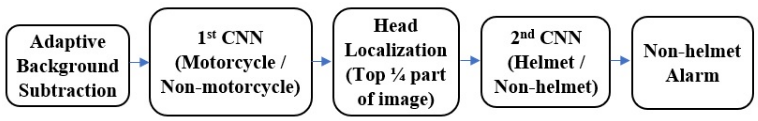

Fig. 5 Framework proposed by Vishnu et al. [1]

Although the proposed method and [1] look similar in strategy, three differences between them are listed as follows. First, we filter unlike-motorcycle objects before entering the first CNN. Second, Vishnu et al. [1] crop the top one-fourth part of the image to locate the region of the motorcyclist's head before feeding it to the second CNN. Third, Vishnu et al. [1] use AlexNet for both CNNs while we employ Inception-ResNet-v2 and Inception-v3.

Fig. 6 Sample objects from three classes of the dataset

Dataset

To create the dataset for training and testing, we gather mp4 video streams (720p, 30 FPS) from the surveillance cameras installed at roadsides around the area of King Mongkut's Institute of Technology Ladkrabang. All moving objects are extracted from each video frame by the adaptive background subtraction technique [16]. Afterward, Inception-ResNet-v2 is used to classify those objects to collect only motorcycle images. Since Inception-ResNet-v2 is a well-known pre-trained model for classifying 1,000 targets (including motorcycle), using it to classify just motorcycle and non-motorcycle is a very easy task as it produces nearly 100% accuracy. For each motorcycle object, we manually tag the ground truth helmeted/non-helmeted. A total of 13,800 images are partitioned into 4,600 images for each of 3 classes, e.g., helmeted rider, non-helmeted rider, and other objects (non-motorcycle). Fig. 6 shows sample objects from each class of our dataset. All passengers in a helmet class must wear helmet. If there is at least one non-helmeted passenger, that object will be categorized as non-helmeted.

3. Results and Discussions

Searching for a Suitable Value of the Threshold Θ

Determining a suitable value of the threshold for the dimension ratio (Y/X) is not trivial because the angle of the surveillance camera influences a proper value of a threshold. It is necessary to find a suitable threshold when a new camera is installed. This step is worth doing since this threshold can significantly eliminate the non-motorcycle objects and it is a single time setup. We illustrate a sample experiment for one camera by creating a new dataset from its video stream. The adaptive background subtraction is applied to identify all moving objects resulting in 52,512 moving objects. For each moving object, if $Y/X > \Theta$ then move this motorcycle-like object into the first pool (motorcycle-like). Otherwise, move it into the second pool (unlike a motorcycle). Next, we apply the pre-trained Inception-ResNet-v2 to classify objects in both pools and measure the percentage that Inception-ResNet-v2 detects motorcycles (bikes). To find a suitable value of Θ, the number of moving objects left in the first pool should be small (comparing to the total number of moving objects) while the percentage of motorcycles detected in the second pool (unlike a motorcycle) must be low.

Table 2 Determining a suitable value of the threshold Θ.

	1st Pool (like bike)		2nd Pool (unlike bike)	
Θ	**No. of objects**	**No. of bikes detected**	**No. of objects**	**No. of bikes detected**
0.75	52,342	68.23	170	0.00
1.00	43,121	82.82	9,391	0.05
1.25	**39,150**	91.22	**13,362**	**0.05**
1.50	20,286	92.99	32,226	52.28
2.00	5,120	19.80	47,392	73.22

Table 2 shows the number of moving objects and the percentage of motorcycles detected in each pool for various values of Θ. If the value of Θ is too small, e.g., 0.75, most moving objects will be thrown into the first pool resulting in many non-motorcycle objects are still unfiltered. On the other hand, if the value of Θ is too large, e.g., 2.00, moving objects in the first pool will be aggressively filtered as most motorcycles are thrown into the second pool. As a result, the suitable value of Θ is 1.25 since it can filter a significant number of unrelated objects (25.44% of 52,512 objects) and produce only 0.05% of the motorcycles detected in the second pool.

Selecting 2nd CNN Architecture for Helmeted / Non-helmeted Classification

Three candidate pre-trained models for the second CNN are ResNet-152-v2, Inception-ResNet-v2, and Inception-v3. For each pre-trained model, we append a two-neuron output layer which is fully connected to its original output layer. In the preliminary training process, the number of training epochs to fine-tune each model is fixed at 100 and only weights connected to an output layer can learn. Half of the motorcycle images (helmeted & non-helmeted) from the dataset are partitioned into 3,450 training images and 1,150 test images with an equal number of target classes (helmeted and non-helmeted). The specification of a machine for training CNNs are Intel Core i9, 64 GB RAM, and GPU RTX-2080 8 GB. We use five-folds cross validation to determine the early-stop training point before each candidate model begins to overfit. The accuracies and losses of 3 CNN models are shown in Fig. 7 – 9. According to the loss graphs from 3 models, the divergence between the training loss and the validation loss begins after training for 10 epochs. Therefore, this is a good place to early stop the training to avoid the overfitting problem. After a real training, we measure the accuracies from 3 models by using a test set. The test accuracies of ResNet-152-v2, Inception-ResNet-v2, and Inception-v3 are 93.79%, 95.21%, and 95.67%, respectively. Both Inception-ResNet-v2 and Inception-v3 achieve the best accuracy.

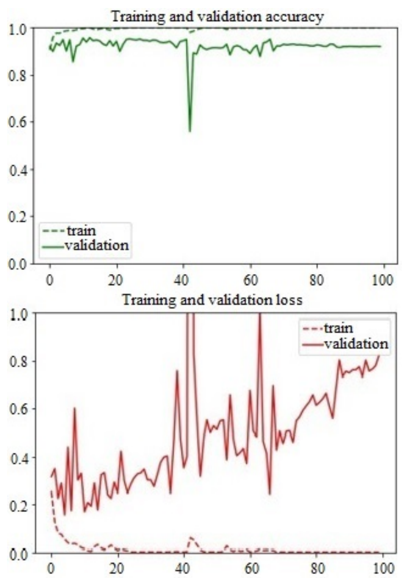

Fig. 7 ResNet-152-v2

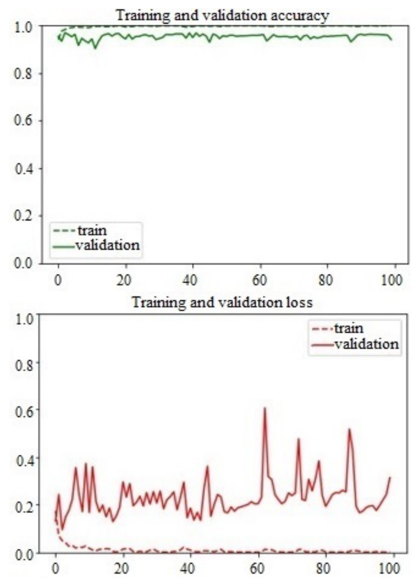

Fig. 8 Inception-ResNet-v2

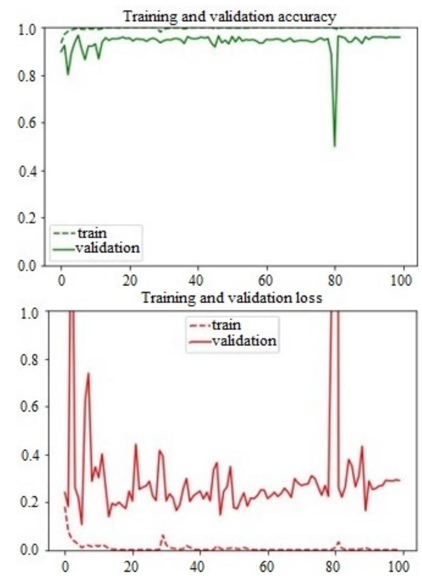

Fig. 9 Inception-v3

Table 3 Confusion matrices of Inception-ResNet-v2 and Inception-v3.

	Inception-ResNet-v2		Inception-v3	
Prediction \ Target	**Helmeted**	**Non-helmeted**	**Helmeted**	**Non-helmeted**
Helmeted	96.67	3.33	98.25	**1.75**
Non-helmeted	2.25	97.75	2.92	97.08

In order to break the tie, we consider the confusion matrices in Table 3. Since the objective of the proposed system is to detect non-helmeted riders, the system must maintain high true negative (the non-helmeted rider is detected as non-helmeted) and low false positive (non-helmeted biker is detected as helmeted). As a result, we choose Inception-v3 to predict a non-helmeted rider since it produces only 1.75% false positive rate.

Experiment on the Complete Dataset

The dataset is partitioned into 10,350 training images and 3,450 test images with an equal number of classes (non-helmeted rider, helmeted rider, and other objects). To compare with the state-of-the-art method, we implement the framework in Vishnu et al. [1] and train it with our dataset. For the sake of a fair comparison, we replace AlexNet in [1] with Inception-ResNet-v2 and Inception-v3 as we used in our system since they are more recent architectures which outperform AlexNet. Let us call this modified version of Vishnu et al. [1] as MVishnu. To be congruent with the framework in [1], we separately train the second CNN (Inception-v3) in MVishnu by using only the upper one-fourth part of images. We also implement a variation of the proposed system without a filtering condition to understand the impact of the filter. Five-fold cross validation is employed to ensure stable statistical results. The experimental results for training helmeted/non-helmeted classification and overall testing among three frameworks are shown in Table 4.

Table 4 Results from helmet/non-helmeted training and overall testing.

	Fold1	**Fold2**	**Fold3**	**Fold4**	**Fold5**	**Avg.**	**Test Acc.**	**Time(sec)**
MVishnu	95.43	96.12	95.69	94.95	95.08	95.45	92.52	13.58
Proposed (No filter)	97.04	96.82	96.74	97.10	96.87	96.91	95.50	14.20
Proposed (Filter)	97.11	96.79	96.35	97.73	97.47	97.09	95.65	10.04

The execution time of the proposed system (with filter) is significantly lower than that of MVishnu by (13.58 – 10.04)*100/13.58 = 26.07%. Although motorcycle objects in MVishnu are cropped to one-fourth size, the time overhead for passing those objects along deep layers of the second CNN is still high. Moreover, passing all moving objects though the first CNN is time-consuming. In our two variations, the filtering condition can significantly reduce the execution time by (14.20 – 10.04)*100/14.20 = 29.29%. The advantage of having a low execution time is the ability to interactively classify all objects in the video frame without a delay. For example, given 3,450 test objects and an average of 8.50 moving objects per frame, the number of video frames is 3450/8.50 = 406 frames. The number of frames per second that the proposed system can process is 406/10.04 = 40.44 FPS. This frame rate is higher than 30 FPS of the surveillance camera. On the other hand, MVishnu can process only 406/13.58 = 29.89 FPS, which means that a 0.11 frame is delayed for every second when operating on 30 FPS camera. This accumulated delay will cause a frame skip when the video buffer is full, resulting in a miss detection of some objects. An insightful implication of the proposed system is the ability to operate under a real situation, especially in a place where most people use motorcycles for their transportation. It can interactively classify objects in the high frame-rate camera to capture high-speed motorcycles without any lag of an execution. As a result, none of the motorcycle is undetected during the monitoring period.

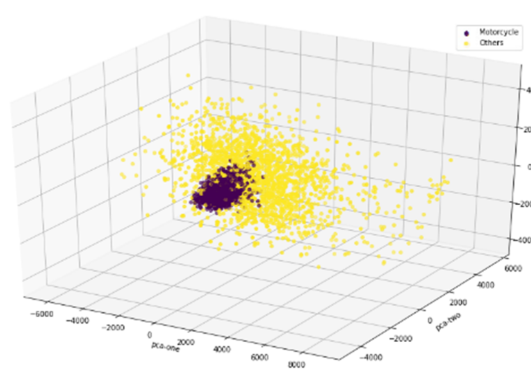

Fig. 10 PCA plot for motorcycle & other objects.

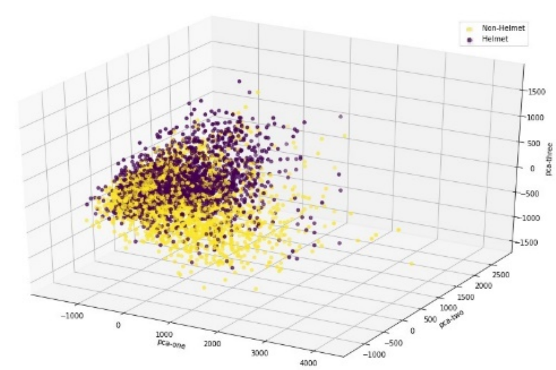

Fig. 11 PCA plot for helmet & non-helmet.

There are a couple of discussions which are worth mentioning. Firstly, the distribution of objects in Principal Component Analysis (PCA) supports the fundamental concept of the proposed system. Fig. 10 and Fig. 11 illustrate the 3D plots of 3 principal components from the dataset. Two obvious clusters in Fig. 10 indicate that motorcycles and other objects are easy to classify. Therefore, applying the filtering condition in this step can significantly reduce the number of unrelated objects before classifying with the first CNN. On the contrary, classifying helmet/non-helmet is more difficult since two classes are overlapped, as shown in Fig.11. Therefore, it is necessary to fine-tune the second CNN to discern the differences between the two classes.

Fig. 12 Images misclassified by MVishnu.

Fig. 13 Images misclassified by our system.

As the second discussion, we investigate the validation and test accuracies from Table 4 which two variations of the proposed system are slightly better than MVishnu. Fig. 12 illustrates some misclassified objects by MVishnu. These images are the case when a motorcycle is vertically aligned with the object in the background and their movements are in the same direction and velocity. The upper one-fourth part of Fig.12 (a – c) returns cars, while Fig.12 (d – f) returns helmets which are the wrong predictions since non-helmeted riders locate in the lower half of images. The proposed system does not struggle with this problem since it brings the whole image to the second CNN.

For the last discussion, it is interesting to look at some misclassified cases by the proposed system. Fig.13 (a – d) shows some false-positive test images; non-helmeted motorcyclists are classified as helmeted. We can reduce this kind of false by adding more various images, e.g., the rider wears a cap, hair net, and veil into the dataset. In addition, Fig.13(e) reveals the limitation of the proposed system. It fails to filter the courier motorcycle which carries a large baggage because the dimension ratio is less than a certain threshold. However, this kind of object seldomly occurs in our dataset (about 0.50%) and the usual environment.

4. Conclusion

We propose a fast non-helmeted motorcyclist detection system by using the filtering condition and two CNNs. The first CNN (Inception-ResNet-v2) identifies motorcycle-like objects while the second CNN (Inception-v3) classifies whether the riders are wearing helmets. The proposed system filters a lot of unrelated objects by comparing the dimension ratio of the object with a threshold before passing them to the first CNN. As a contribution of the proposed system, its fast classification produces a smooth and real-time non-helmet monitoring system without any undetected moving object. In addition, it produces higher classification accuracy comparing to the state-of-the-art method. The experimental results on the image dataset captured from the surveillance cameras show that the proposed system can quickly classify images with high accuracy and low false-positive rate. A limitation of our system is that it misclassifies the rare case when a motorcycle carries large baggage at the back as we discussed in the previous section. One possible future work of the proposed method is to train the second CNN with more various images, i.e., the non-helmet riders who wear a cap, hair net, and veil, to reduce the false-positive predictions.

5. References

[1] C. Vishnu, D. Singh, C.K. Mohan, S. Babu, Detection of motorcyclists without helmet in videos using convolutional neural network, International Joint Conference on Neural Networks (IJCNN), Anchorage, USA. 14 – 19 May 2017, 3036 – 3041.

[2] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, COMMUN. ACM. 60(2) (2017) 84 – 90.

[3] J. Mistry, A.K. Misraa, M. Agarwal, A. Vyas, V.M. Chudasama, K.P. Upla, An Automatic Detection of Helmeted and Non-helmeted Motorcyclist with License Plate Extraction Using Convolutional Neural Network, Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA), Montreal, Canada. 28 November – 1 December 2017, 1 – 6.

[4] K.C.D. Raj, A. Chairat, V. Timtong, M.N. Dailey, M. Ekpanyapong, Helmet violation processing using deep learning, International Workshop on Advanced Image Technology (IWAIT), Chiang Mai, Thailand. 7 – 9 January 2018, 1 – 4.

[5] M.A.V. Forero, Detection of motorcycles and use of safety helmets with an algorithm using image processing techniques and artificial intelligence models, MOVICI-MOYCOT: Joint Conference for Urban Mobility in the Smart City, Medellin, Colombia.18 – 20 April 2018, 1 – 9.

[6] N. Boonsirisumpun, W. Puarungroj, P. Wairotchanaphuttha, Automatic Detector for Bikers with no Helmet using Deep Learning, International Computer Science and Engineering Conference (ICSEC), Chiangmai, Thailand. 21 – 24 November 2018, 1 – 4.

[7] X. Long, W. Cui, Z. Zheng, Safety Helmet Wearing Detection Based on Deep Learning, IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chengdu, China. 15 – 17 Mar 2019, 2495 – 2499.

[8] M. Dasgupta, O. Bandyopadhyay, S. Chatterji, Automated Helmet Detection for Multiple Motorcycle Riders using CNN, IEEE Conference on Information and Communication Technology, Allahabad, India. 6 – 8 Dec 2019, 1 – 4.

[9] F. Khan, N. Nagori, A. Naik, Helmet Presence Detection on Motorcyclists Using Image Processing and Machine Vision Techniques, IEJRD. 5(5) (2020) 1 – 7.

[10] F.W. Siebert, H. Lin, Detecting motorcycle helmet use with deep learning, Accid. Anal. Prev. 134(1) (2020) 1 – 28.

[11] T. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal Loss for Dense Object Detection, IEEE PAMI. 42(2) (2017) 318 – 327.

[12] S. Kumar, N. Neware, A. Jain, D. Swain, P. Singh, Automatic Helmet Detection in Real-Time and Surveillance Video, Machine Learning and Information Processing. Advances in Intelligent Systems and Computing. 1101(1) (2020) 51 – 60.

[13] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), OH, USA. 23 – 28 June 2014, 580 – 587.

[14] R. Girshick, J. Donahue, T. Darrell, J. Malik, Region-Based Convolutional Networks for Accurate Object Detection and Segmentation, IEEE PAMI. 38(1) (2015) 142 – 158.

[15] R. Girshick, Fast R-CNN, IEEE International Conference on Computer Vision (ICCV), Santiago, Chile. 7–13 Dec 2015, 1440 – 1448.

[16] Z. Zoran, Improved adaptive gaussian mixture model for background subtraction, Proceeding International Conference Pattern Recognition (ICPR), Cambridge, England. 23 – 26 August 2004, 28 – 31.

[17] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), NV, USA. 27 – 30 June 2016, 770 – 778.

[18] K. He, X. Zhang, S. Ren, J. Sun, Identity Mappings in Deep Residual Networks, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, Springer International Publishing, Amsterdam, 9908 (2016) 630 – 645.

[19] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, USA. 4 – 9 February 2017, 4278 – 4284.
[20] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the Inception Architecture for Computer Vision, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas. 27 – 30 June 2016, 2818 – 2826.
[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston. 7 – 12 June 2015, 1 – 9.