Detection and Number Plate Recognition of Non-Helmeted Motorcyclists using YOLO

SUBMISSION DATE / POSTED DATE

23-08-2022 / 29-08-2022

CITATION

Padalia, Dishant (2022): Detection and Number Plate Recognition of Non-Helmeted Motorcyclists using YOLO. TechRxiv. Preprint. https://doi.org/10.36227/techrxiv.20561343.v1

DOI

10.36227/techrxiv.20561343.v1

Detection and Number Plate Recognition of Non-Helmeted Motorcyclists using YOLO

Dishant Padalia
Department of Electronics and Telecommunication Engineering
K.J. Somaiya College of Engineering
Mumbai, India
dishant.padalia@somaiya.edu

***Abstract*—Motorcycles are the most common mode of transport as they are affordable and low-maintenance vehicles. Motorcyclists were roughly 29 times more likely than passenger car passengers to die in an accident per vehicle mile travelled in 2019. One of the leading causes of fatal motorcycle accidents is the rider's failure to wear a helmet. According to section 129 of the motorcycle vehicle act, the Government has made it mandatory for two-wheeler drivers to wear helmets while driving. Still, many traffic rule violators do not obey them. In most developing countries, traffic police manually monitor motorcyclists at road junctions. Still, this method is inefficient as it does not apply on highways where the probability of accidents is highest due to speeding. This paper presents an automatic surveillance system for detecting two-wheeler drivers without helmets and recognizes their License plate numbers in the system. Firstly, the system detects motorcycles in the image or live video using the You Only Look Once (YOLO) algorithm. It again applies this algorithm to detect whether the driver is helmeted or not for the detected motorcycles. Finally, the motorcycle's number plate is detected for identified motorcyclists without a helmet, and the characters are extracted using Optical Character Recognition.**

Keywords—Helmet Detection, Number Plate Recognition, YOLO, Neural Networks

I. INTRODUCTION

In developing countries, most of the population uses Motorcycles for daily commuting. Motorcycles, by definition, are significantly less crashworthy than closed automobiles. In addition, they are less visible to other cars and pedestrians and less stable than four-wheel vehicles. Motorcycle riders and passengers are more sensitive to weather and road conditions than drivers in enclosed cars.

In the population-based observational study [1] in India, helmet observations were recorded for 68,229 motorcyclists and 21,777 pillion riders. About 22.6% of motorcycle drivers and only 1.1% of pillion riders were found to be wearing helmets. Motorcycle riders lack the safety of an enclosed vehicle when they crash, making them more likely to be injured or killed. Helmets are approximately 37% effective in avoiding motorcycle deaths and around 67% effective in preventing brain injuries. Helmets are required because severe head injuries are typical among fatally wounded motorcycle riders.

Currently, the Traffic Police capture photographs of the Riders not wearing helmets using their Handy smartphones and then manually input the Vehicle License Plate into the system to take strict action against these violators. This procedure is inefficient owing to a lack of police officers and excessive manual work. Also, now almost all the cities have CCTV surveillance cameras on the majority of the Road Junctions. These, however, require human participation and are not computerized.

The technique presented in this study automates the surveillance of non-helmeted motorcyclists. It identifies bikers who are not wearing helmets and recognizes license plates from CCTV footage in real time. The You Only Look Once [2] (YOLO) architecture is used that is built using Convolutional Neural Networks.

II. RELATED WORK

The first proposed system for detection of non-helmeted motorcycle riders was done by Chiu et al. [3]. This system uses segmentation techniques to extract moving objects from the input image and subtract the background image. It also uses vehicle occlusion for separating merged vehicles. Lastly, for helmet detection, uses Canny Edge Detection to detect the edges of the helmet and check if it is a circle or not. The main shortcoming of this approach is that the motorcycles are moving, and as a result, the image data might be distorted by motion blur; hence it reduces the precision of the edges in the images. Another disadvantage of this technique is that it frequently leads to misclassification since circular items are categorised as helmets while certain helmets with slightly different shapes are not identified as helmets.

In parallel with Chiu's work, Chiverton et al. [4] proposed a system that uses an SVM classifier with a linear kernel to differentiate between head and helmet, which is trained based on features such as shape and reflective property of helmets where the tops of helmets are shinier than the bottom half. The limitation of this system is that it does not first detect and segment out all the motorcyclists in the image and then detect whether they are helmeted since the system is only relevant when we can get the number plates of not helmeted motorcyclists and penalize them.

Silva et al. [5], [6] came up with a system that first does Background Subtraction using only the grayscale information between the current frame and the background image. The bikers in the image are then identified using an SVM classifier trained on features extracted by an Local Binary patterns descriptor, which encodes local primitives such as edges, curves, stains, and plain regions. The helmet is then classified using SVM and Random Forest on features taken from the Circular Hough Transform [7], Histogram Oriented Gradients, and LBP descriptors.

Waranusast et al. [8] created a method that employs the K-Nearest Neighbors classifier. To distinguish motorcycles from other moving objects, it uses three features from the moving connected region: The area of the bounding rectangle, the Aspect ratio of the bounding rectangle, and the Standard deviation of hue. Then, using these attributes, the KNN classifier determines if the moving item is a motorcycle. Finally, KNN is used to determine if the extracted head is wearing a helmet or not. The features used are based on each head quadrant's circularity, average intensity, and average hues.

In 2016, Doungmala et al. [9] proposed a real-time vision-based helmet-wearing monitoring system. The system used a moving object detection method for full and half helmet detection using a variant of the Adaboost classifier. The proposed method used visual face/nose/mouth / left eye / right eye detection using a haar-like feature and circle Hough transform to identify complete and half helmet classes. The main drawback of this system is that it does not detect motorcycles first; instead directly detects only the helmets.

K. Dahiya et al. [10] created a method for detecting helmets in surveillance footage using an SVM classification model to distinguish between motorcyclists and non-motorcyclists and another SVM classifier to classify helmet or non-helmet. They developed three features - HOG, SIFT, and LBP - and determined that HOG with a linear kernel outperforms all other combinations.

C. Vishnu et al. [11] suggested a method based on Convolutional Neural Networks. In this approach, adaptive background subtraction is applied to detect moving objects. Then these moving objects are given to a CNN classifier to classify them as motorcyclists or non-motorcyclists. The non-motorcyclists are discarded, and the motorcyclists are passed to another CNN classifier to classify the motorcyclist as with-helmet or without-helmet.

In parallel with C. Vishu's work, Y. Kulkarni et al. [12] devised a framework that incorporates Convolutional Neural Networks and Transfer Learning. They have used CNN layers on top of VGG-16 [13], a pre-trained model trained on ImageNet [14]. If the moving object supplied into this classifier is categorized as a motorcycle, it is sent to the head localization algorithm. They built a similar classifier for helmet and non-helmet classification. If the head region is classified as non-helmet, the original biker picture is provided for number plate detection; otherwise, the frame is rejected. Optical Character Recognition (OCR) recognizes the characters on the number plate. The main flaw of this proposed approach is that there are many motorcycles in a single frame during traffic, and this CNN classifier will not be able to detect multiple overlapping motorcycles in a single frame.

III. Proposed Work

Most researchers have proposed methods in which classifiers are built using hand-crafted features, but coming up with perfect features that work in all scenarios is difficult. While some have used edge detection techniques for the segmentation and detection of motorcycles in the image, the major drawback is that it does not work when multiple overlapping motorcycles are in a single frame. Convolutional neural network models require a lot of processing power and time. Therefore, the YOLO (You Only Look Once) algorithm is used in this proposed system.

A. Working of YOLO

Joseph Redmond et al. [2] proposed YOLO (i.e. You Only Look Once) in 2015. YOLO is an object detection algorithm that handles the entire image in a single iteration and predicts multiple bounding box coordinates and class probabilities for these boxes. YOLO was proposed to address the issues encountered by object recognition models at the time like Fast R-CNN. Fast Region based Convolutional Neural Networks [15] was a cutting-edge model at the time, but it had limitations, namely that it cannot be used in real-time because prediction usually takes 2-3 seconds.

Object Detection in YOLO is treated as a regression problem; it requires only a single forward propagation to predict bounding boxes and classes on the image. The method employs a single end-to-end trained neural network that receives an image as input and directly predicts bounding boxes and labels.

The model divides the image into grid cells, where each cell is responsible for predicting K bounding boxes. Each bounding box can be described using five descriptions, class probability (pc), the centre of the box (bx, by), width (bw), height (bh) and value c corresponding to the class of the object. Confidence scores reflect the presence or absence of an object in the bounding box. IoU is intersection over union; it calculates the overlap of predicted and ground truth bounding boxes. IoU is the ratio of area of intersection to the area of union of the ground truth and predicted bounding box.

$$confidence\ score = p(Object) \cdot IoU$$

$$IoU = \frac{A \cap B}{A \cup B}$$

where:
p(Object) – Probability of presence of object
A – Ground Truth Bounding Box
B – Predicted Bounding Box

One of the most common problems with YOLO is that rather than detecting an object just once, they might detect it multiple times. Non-max suppression is a technique that can help to solve this problem. In this technique, we take the boxes with maximum probabilities and suppress the close-by boxes with non-maximum probabilities.

B. Model Architecture

In this paper, we have used YOLOv5. The network architecture of YOLOv5 comprises three parts: Backbone, Neck and Head. Model Backbone mainly extracts essential features from the given input image. CSPNet (i.e. Cross Stage Partial Network) is a backbone of the YOLOv5 model. Model Neck is used to generate feature pyramids. The feature pyramid improves the recognition of visual objects at many sizes and aids the model's performance on unseen data. In YOLOv5, PANet (i.e. Path Aggregation Network) is used for the model neck. The model Head is primarily responsible for detection. It creates final output vectors with class probabilities, confidence scores, and bounding boxes by using anchor boxes on features.

GIoU [16] (i.e. Generalized Intersection over Union) Loss Function is used in YOLOv5. IoU loss only operates when the predicted bounding boxes overlap with the ground truth box and therefore does not offer a moving gradient in circumstances when the bounding boxes are not overlapping. On the other hand, GIoU loss maximizes the overlap region between the ground truth and predicted bounding boxes. In non-overlapping scenarios, it gradually increases the size of the predicted box to overlap with the target box. GIoU loss outperforms IoU loss in terms of accuracy.

In Fig. 1, we can observe that both (a) and (b) cases have IoU of zero since there is no overlap between the ground truth and predicted bounding box, but example (b) is a better prediction than (a) as it is closer to the ground truth. For this reason, GIoU is used as it increases as the prediction moves towards the ground truth. GIoU ranges from -1 to 1. Example (a) has a GIoU score of -0.75, while (b) has a GIoU score of -0.34.

$$GIoU = \frac{A \cap B}{A \cup B} - \frac{|C \backslash (A \cup B)|}{|C|} = IoU - \frac{|C \backslash (A \cup B)|}{|C|}$$

where A and B are the ground truth and prediction bounding boxes respectively. C is the smallest convex hull that encloses both A and B.

In the middle layers, the Leaky ReLU [17] activation function is used, whereas the sigmoid activation function is used in the final detection layer. In YOLOv5 is can be trained using Adam or SGD optimization function. The optimization function used in this study is SGD. Fig. 2 represents the YOLOv5 Architecture.

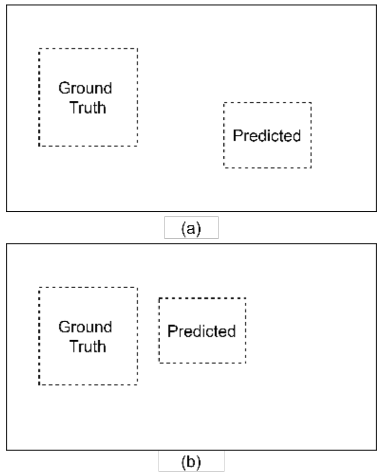

Fig. 1. Two examples (a) and (b) with ground truth and prediction bounding boxes. (a) has IoU and GIoU scores of 0 and -0.75, while (b) has IoU and GIoU scores of 0 and -0.34.

YOLOv5 has four versions based on the depth of the model, namely YOLOv5s, YOLOv5m, YOLOv5l and YOLOv5x. The version used in this study is YOLOv5l. As the depth of the model decreases, the detection is faster, but the accuracy is compromised in that version. For this reason, the YOLOv5l model is used as it is very accurate and is quite fast in real-time detection.

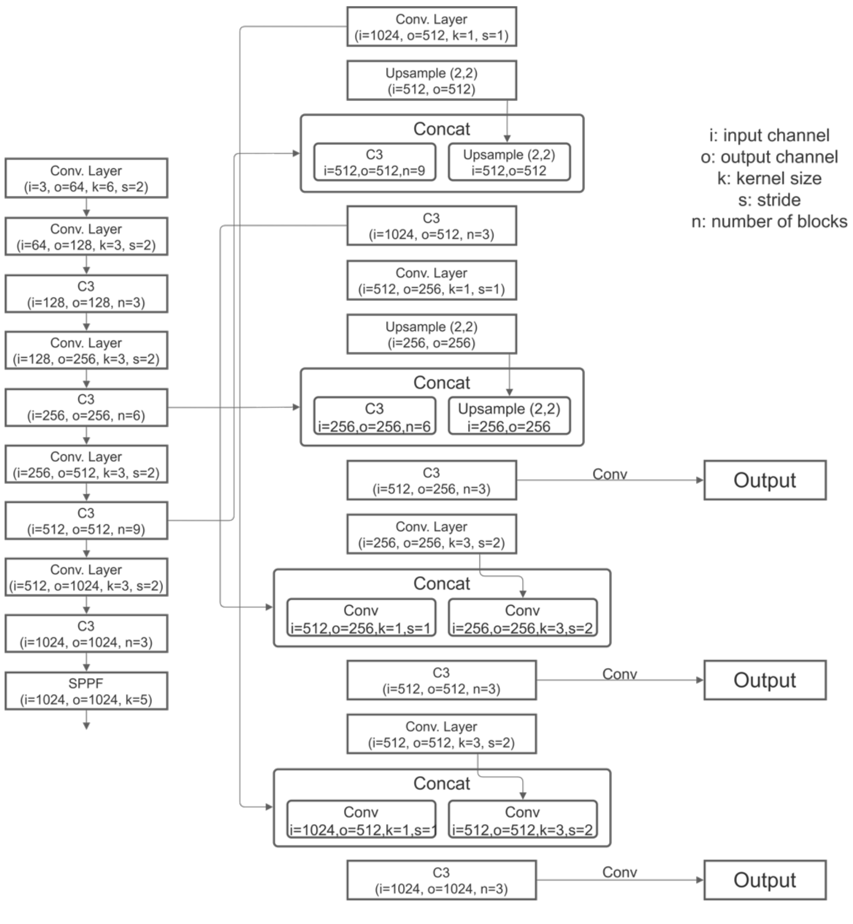

Fig. 2. YOLOv5 Architecture

C. Dataset

There is no publicly available dataset for this study. So the dataset used in this research was acquired from the Road Traffic Department. The dataset consists of images captured from the CCTV cameras at the traffic junctions. The dataset contains 202 images, and the dimensions of each image are 1920 x 1080 x 3. The dataset was then split into separate train and test datasets. The training dataset consisted of 182 images, and the testing dataset consisted of 20 images.

D. Data Preprocessing

Before training the model, the images in the training dataset had to be manually annotated. The data annotation was done using the makesense.ai [18] website. The website generates annotations in a text file for each image. It contains class, starting coordinates (i.e., top-left x, y coordinates), and the ending coordinates (i.e., bottom-right x, y coordinates) for each bounding box in the image. The coordinates are normalized between 0 and 1. Since the proposed methodology consists of two stages, one for motorcycle detection and another for helmet and license plate detection, the dataset was annotated twice. For the first stage, the target class is a motorcycle, and for the second stage, the target classes are helmet, non-helmet, and license plate.

E. Methodology

The flow chart of the proposed system is illustrated in Fig. 2. YOLOv5 was used for the real-time detection of motorcycles. There are two stages in the proposed approach. In the first stage, each frame is taken as the input to the YOLO model, which detects all the motorcycles in the frame. Then in the second stage, each detected motorcycle is extracted and passed to the next YOLO model, which detects the helmet, non-helmet, and license plate in the input motorcycle image. If the motorcyclist is non-helmeted, the license plate is extracted from the frame and passed for Optical Character Recognition to extract the characters on the license plate. The output is recorded and can be used to fine motorcycle riders. The characters on the number plate are recognized using Google's Tesseract [19] Optical Recognition.

The first and second stage YOLOv5 model were trained in Google Colab for 150 epochs each with a batch size of 2.

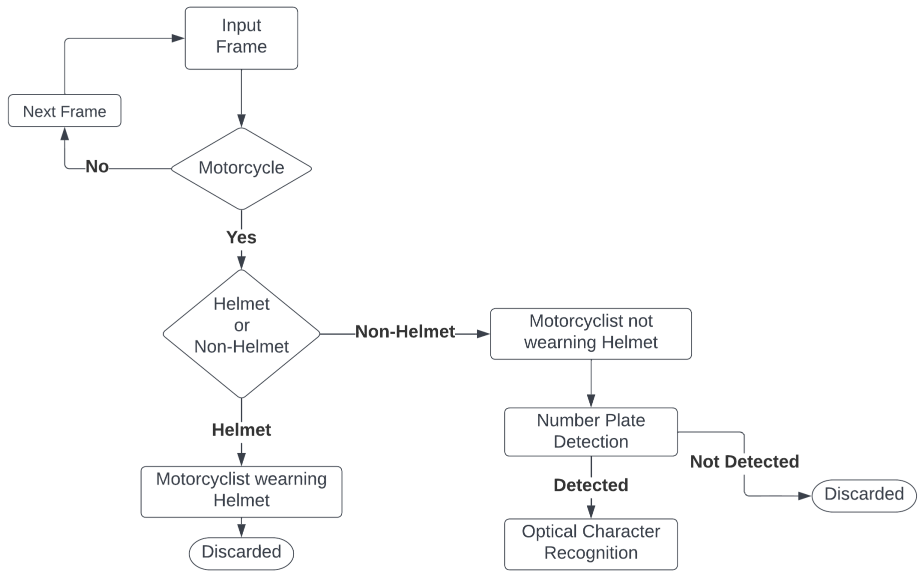

Fig. 3. Flowchart of the proposed methodology

IV. Experiments and Results

A. Evaluation Metrics

To evaluate the model's performance, the metrics used are Precision, Recall, Mean Average Precision (mAP) and F1 score.

Precision - It is the ratio of correctly predicted positive observations to the total predicted positive observations.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

Recall - It is the ratio of correctly predicted positive observations to the total actual positive observations.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Mean Average Precision (mAP) - It is a standard metric to analyze the accuracy of the object detection model. Average Precision is the weighted mean of Precision at each threshold and is calculated for each class. Mean Average Precision is calculated by taking the mean of AP over all classes. In this study, two of the most widely used mAP metrics are used, mAP_0.5 and mAP_0.5:0.95. Here mAP_0.5 means the average Precision for each class is calculated for an IoU threshold of 0.5. While mAP_0.5:0.95 means the average Precision for each class is calculated for IoU thresholds from 0.5 to 0.95.

F1 Score - It calculates the balance between precision and recall.

$$F1\ Score = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

B. Results

In our experiment, the motorcycle detection model achieved a precision of 95.6%, recall of 99.5%, F1 score of 97.51%, mAP_0.5 score of 99.32% and mAP_0.5:0.95 score of 73.93%. Whereas the helmet, non-helmet and license plate detection model achieved a precision of 99.69%, recall of 99.87%, F1 score of 99.77%, mAP_0.5 score of 99.5% and mAP_0.5:0.95 score of 94.77%.

TABLE 1. PERFORMANCE OF THE TWO YOLO MODELS

Detection Model	Performance			
	Precision	Recall	mAP_0.5	mAP_0.5:0.95
Motorcycle	95.6	99.5	99.32	73.93
Helmet/ Non-Helmet/ Number Plate	99.69	99.87	99.5	94.77

Fig. 4 demonstrates the entire methodology using an example. Firstly, the frame is captured from CCTV footage. In step (a), the input frame is passed to the first YOLO model to detect the motorcycles in the image. The model detects all the motorcycles in the image with the confidence score as shown in the image. Then in step (b), these detected motorcycles are cropped to get individual images of each motorcycle. In step (c), these separate images are passed individually to the second YOLO model to detect the helmet, non-helmet and number plate in each figure. We notice that the right motorcyclist is wearing a helmet while the left one is not, and the model has detected that appropriately. It has also detected the number plates in each of the images. Since the right rider is wearing a helmet, that number plate is discarded. For the non-helmeted rider, the number plate is extracted from the image. It is passed for optical character recognition (OCR), and the recognized text is stored in the database so that the concerned rider can be penalized.

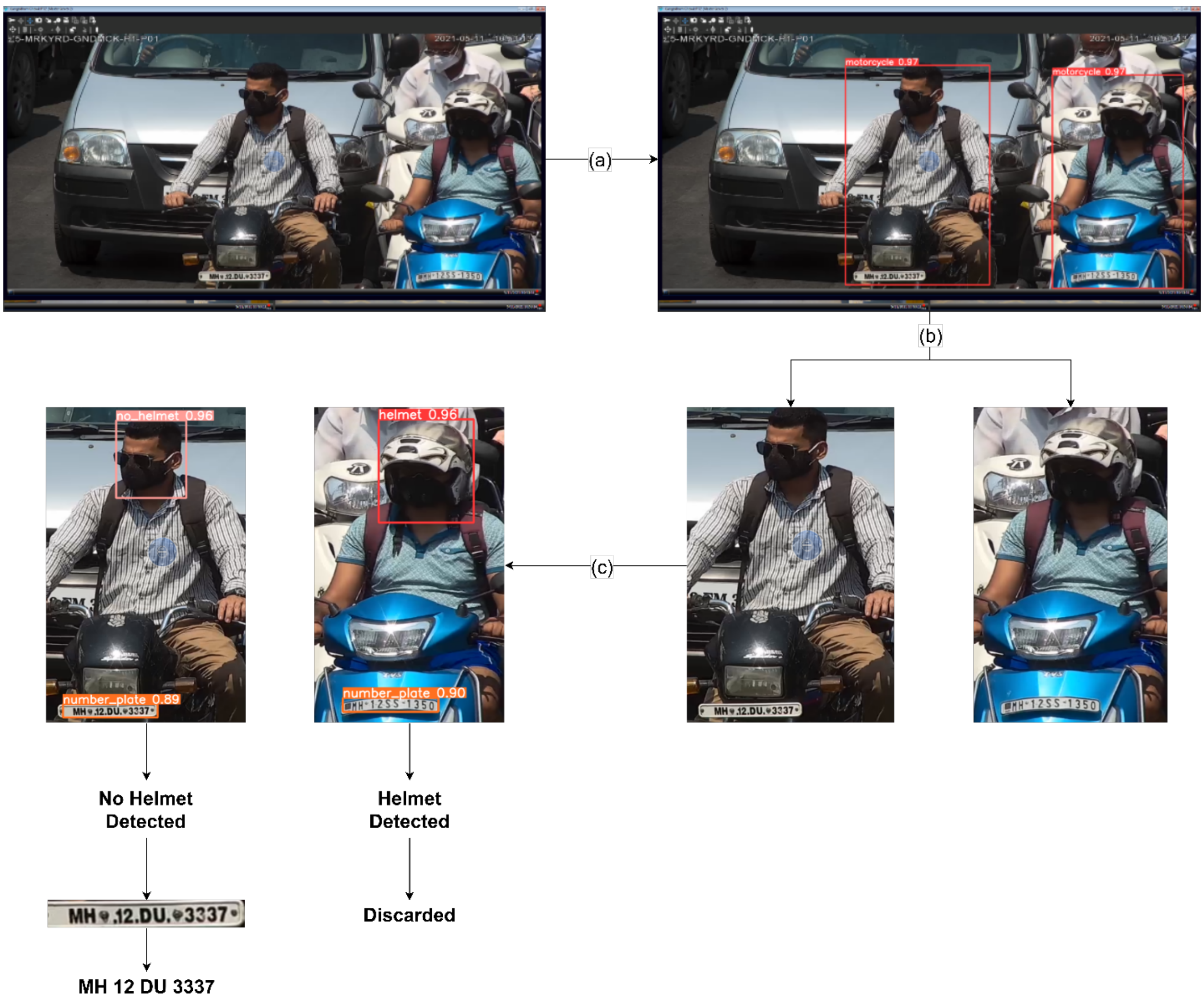

Fig. 3 Example of the proposed methodology. (a) Input frame from CCTV is passed to the first YOLO model, (b) Detected motorcycles are cropped, and (c) Each motorcycle image is given separately to the second YOLO model for detection of helmet, non-helmet and number plate. The number plate of non-helmeted riders is passed for OCR.

V. Conclusion

Motorcycle accidents are the most common road accidents leading to many deaths. One of the most significant reasons for deaths during motorcycle accidents is the rider not wearing a helmet. Many laws are passed making it mandatory for two-wheeler drivers to wear helmets, but still, many motorcyclists do not obey them. The current systems are very inefficient. In this research, we have proposed a practical framework for detecting non-helmeted motorcyclists from CCTV footage using YOLO; it is very fast and effective in real-time. After the non-helmeted motorcyclists are detected, the number plate characters are extracted using optical character recognition and stored in a database so that the concerned individuals can be penalized.

References

[1] S. Wadhwaniya, S. Gupta, L. K. Josyula, G. Gururaj, and A. A. Hyder, "The validity of self-reported helmet use among motorcyclists in India," *WHO South-East Asia journal of public health*, vol. 4, no. 1, pp. 38–44, 2015.

[2] 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*, 2016, pp. 779– 788.

[3] Chung-Cheng Chiu, Min-Yu Ku, and Hung-Tsung Chen, "Motorcycle detection and tracking system with occlusion segmentation," in *Eighth International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'07)*, 2007, pp. 32–32.

[4] J. Chiverton, "Helmet presence classification with motorcycle detection and tracking," *IET Intelligent Transport Systems*, vol. 6, no. 3, pp. 259– 269, 2012.

[5] R. Silva, K. Aires, T. Santos, K. Abdala, R. Veras, and A. Soares, "Automatic detection of motorcyclists without helmet," in *2013 XXXIX Latin American Computing Conference (CLEI)*, 2013, pp. 1–7.

[6] R. R. V. e Silva, K. R. T. Aires, and R. de Melo Souza Veras, "Helmet detection on motorcyclists using image descriptors and classifiers," in *2014 27th SIBGRAPI Conference on Graphics, Patterns and Images*, 2014, pp. 141–148.

[7] A. S. Hassanein, S. Mohammad, M. Sameer, and M. E. Ragab, "A survey on Hough transform, theory, techniques and applications," *arXiv preprint arXiv:1502.02160*, 2015.

[8] R. Waranusast, N. Bundon, V. Timtong, C. Tangnoi, and P. Pattanathaburt, "Machine vision techniques for motorcycle safety helmet detection," in *2013 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013)*, 2013, pp. 35–40.

[9] P. Doungmala and K. Klubsuwan, "Helmet wearing detection in Thai- land using Haar like feature and circle hough transform on image processing," in *2016 IEEE International Conference on Computer and Information Technology (CIT)*, 2016, pp. 611–614.

[10] K. Dahiya, D. Singh, and C. K. Mohan, "Automatic detection of bike- riders without helmet using surveillance videos in real-time," in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 3046–3051.

[11] C. Vishnu, D. Singh, C. K. Mohan, and S. Babu, "Detection of motor- cyclists without helmet in videos using convolutional neural network," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 3036–3041.

[12] Y. Kulkarni, S. Bodkhe, A. Kamthe, and A. Patil, "Automatic number plate recognition for motorcyclists riding without helmet," in *2018 International Conference on Current Trends towards Converging Tech- nologies (ICCTCT)*, 2018, pp. 1–6.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[14] J. Deng, W. Dong, R. Socher, Li-Jia Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, 2009, pp. 248–255.

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

[16] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized intersection over union: A metric and a loss for bounding box regression," in *Proceedings of the IEEE/CVF conference on com- puter vision and pattern recognition*, 2019, pp. 658–666.

[17] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[18] P. Skalski, "Make Sense," 2019.

[19] R. Smith, "An overview of the Tesseract OCR engine," in *Ninth international conference on document analysis and recognition (ICDAR 2007)*, vol. 2, 2007, pp. 629–633.