

# Real-Time Helmet Detection and Number Plate Extraction Using Computer Vision

Jyoti Prakash-Borah, Prakash Devnani, Sumon Kumar-Das,  
Advaita Vetagiri, Partha Pakray\*

National Institute of Technology, Silchar,  
India

{jyoti20\_ug, prakash20\_ug, sumon20\_ug,  
advaita21\_rs, partha}@cse.nits.ac.in

**Abstract.** In the contemporary landscape, two-wheelers have emerged as the predominant mode of transportation, despite their inherent risk due to limited protection. Disturbing data from 2020 reveals a daily toll of 304 lives lost in India in road accidents involving two-wheeler riders without helmets, emphasizing the urgent need for safety measures. Recognizing the crucial role of helmets in mitigating risks, governments have made riding without one a punishable offense, employing manual strategies for enforcement with limitations in speed and weather conditions. In today's world of advancing technology, we can leverage the power of computer vision and deep learning to tackle this problem. This can eliminate the need for constant human surveillance to be kept on riders and can automate this process, thus enforcing law and order as well as making this process efficient. Our proposed solution utilizes video surveillance and the YOLOv8 deep learning model for automatic helmet detection. The system employs pure machine learning to identify helmet types with minimal computation cost by utilizing various image processing algorithms. Once the helmet-less person is detected, the number plate corresponding to the rider's motorcycle is also detected and extracted using computer vision techniques. This number plate is then stored in a database thus allowing further intervention to be done in this matter by the authorities to ensure penalties and enforce safety rules properly. The model developed achieves an overall accuracy score of 93.6% on the testing data, thus showcasing good results on diverse datasets.

**Keywords.** Image dataset, YOLOv8, deep learning model, object detection, image processing algorithms.

## 1 Introduction

The field of Artificial Intelligence (AI) encompasses a diverse range of technologies and applications, with its roots in creating intelligent systems that can perform tasks that typically require human intelligence.

One prominent subfield, Computer Vision, focuses on endowing machines with the ability to interpret and understand visual information from the world, opening up possibilities for applications in image analysis, video processing, and augmented reality.

Within the realm of Computer Vision, the You Only Look Once (YOLO) algorithm stands out as a groundbreaking approach to object detection. YOLO's innovation lies in its unified, real-time processing capabilities, achieved through a single neural network that can simultaneously predict bounding boxes and class probabilities for objects within an image.

The influential paper introducing YOLO, authored by Joseph Redmon and Santosh Divvala in 2016 [16], has since garnered widespread attention and has become a foundational reference in the field of computer vision, influencing subsequent developments and applications of object detection technologies. The increasing population of India in the last 30 years is leading to the use of more vehicles.



**Fig. 1.** A biker without helmet



**Fig. 2.** A biker with helmet

According to Statista, as of 2023, the current population of India is 1.429 billion<sup>1</sup>. A study by Financial Express found that the majority of India's population is middle-class, which is about 31% of the population in 2020–2021 and is expected to rise to 61% by 2046–47<sup>2</sup>, a two-wheeler is the most sought-after vehicle in India.

<sup>1</sup>Statista: [www.statista.com/statistics/263766/total-population-of-india/](https://www.statista.com/statistics/263766/total-population-of-india/)

<sup>2</sup>Financial Express: [tinyurl.com/56thtnfh](https://tinyurl.com/56thtnfh)

Two-wheeler domestic sales rose from 13.57 million in the financial year 2022 to 15.86 million in the financial year 2023<sup>3</sup>, as suggested by data from Statista. The increasing use of two-wheelers without helmets and reckless driving is leading to the deaths of riders. A news article by the Times of India shows that, in the year 2021, 47,000 Indians died in two-wheeler accidents due to not wearing helmets<sup>4</sup>.

Head injuries sustained by riders who do not wear helmets are a major cause of these deaths. Addressing this issue requires a comprehensive approach that combines technology and law enforcement. A study shows that using surveillance cameras in traffic has led to decreased road accidents.

Times of India reports that, in the state of Kerala in India, the use of surveillance cameras led to a decrease in accidents from 1,669 deaths in road accidents from June 5, 2022, to October 31, 2022. However, it dropped to 1,081 during the same period in 2023 after the installation of AI cameras<sup>5</sup>.

A study found out that wearing helmets lowers the death rate chances by 37% and the head injury rate chances by 69% [10]. So there is a need to automate the process of helmet detection for proper law enforcement and to reduce deaths by two-wheelers.

The implementation of an automated system for monitoring helmet usage and identifying license plate numbers of non-compliant two-wheelers is a crucial step toward enhancing road safety. AI and computer vision algorithms can analyze real-time CCTV camera footage, enabling the detection of riders without helmets and the retrieval of their license plate numbers.

Our approach will be using state of the art YOLOv8<sup>6</sup> model to extract the number plates of the without-helmet bike riders and store them in a database. This information can then be used to enforce helmet usage regulations and educate riders about the importance of helmet safety.

<sup>3</sup>Two-wheeler: [www.statista.com/statistics/318023/two-wheeler-sales-in-india/](https://www.statista.com/statistics/318023/two-wheeler-sales-in-india/)

<sup>4</sup>Times of India: [tinyurl.com/mw9advuw](https://tinyurl.com/mw9advuw)

<sup>5</sup>Times of India: [tinyurl.com/2twhy25](https://tinyurl.com/2twhy25)

<sup>6</sup>YOLOv8: [github.com/ultralytics/ultralytics](https://github.com/ultralytics/ultralytics)

## 2 Related Work

Numerous domains, including pose detection, decision-making, self-driving vehicles, computer vision, and digital image processing techniques. The use of deep learning models has demonstrated success in a variety of fields, including healthcare [18], social sciences [11], earth sciences [2], etc.

R. Meenu et al. [12], carried out research where they were performing helmet detection and number plate extraction using Faster Region-Convolutional Neural Network (Faster R-CNN). They used CCTV footage and then split it into frames for analysis. Their methodology was split into four stages: motorcycle detection, head detection, helmet detection, and then number plate detection.

They utilized image processing algorithms like the Gabor wavelet filter to get accurate head positions. They achieved an accuracy of around 92%, depending on the quality of the CCTV cameras. However, cases of false detection are not addressed in the solution. Kunal Dahiya et al. [3] applied algorithms like background subtraction to detect only moving motorcycles and deal with false detection rates.

They also used Gaussian models to deal with various environmental detection challenges. Further, after extracting the foreground layer, many image processing algorithms were applied, like a noise filter and a Gaussian filter, and a binary image was obtained. Furthermore, objects were detected only based on a threshold area range that can be likely classified as a motorcycle.

They used techniques like Histogram of Oriented Gradients (HOG) and scale-invariant feature transformation for feature extraction. For classification, they used a Support Vector Machine(SVM). To remove false detection, they also consolidated the results using the information from the past frames.

They achieved a frame processing time of 11.58 ms and a frame generation time of around 33 ms, implying high efficiency. However, there is a lack of comprehensive evaluation on a diverse range of datasets, thus limiting the generalizability of the results.



**Fig. 3.** Example of the original image and various image augmentations applied

Pushkar Sathe et al. [17] used yolov5 for helmet detection with an accuracy of 0.995 mean Average Precision(mAP) score [15]. They are using two methods to check if the rider is wearing a helmet.

**Table 1.** Total number of images from different sources

Sources	Total Images
Outside the campus	12 videos collected
Online sources including Google and news articles	3600
Data from the private repository of Roboflow	3155

Firstly, they check with the help of overlapping boxes of the helmet, numberplate, and the person and verify through a set of conditions if the person is wearing a helmet or not. The second method uses a range of motorcycle coordinates to check for helmets. Finally, they are using EasyOCR for character recognition of number plates.

However, this suffers from the lack of inclusion of a diverse dataset to make the model more generalizable. J Mistry et al. [13] used YOLOv2 for first detecting persons in a frame, citing the better performances in detecting a person rather than a motorcycle of the model. It then proceeds to detect the helmet, and if it is not found, then it goes for the number plate.

For no number plate detected, the model infers that the person detected is a pedestrian. The model achieved a 0.9470 value accuracy for helmet detection. However, this model also suffers from generalizability as not all cases of numberplates, riders, and helmet positions are discussed. M.M. Shidore and S.P. Narote [19] worked on techniques for efficient and accurate extraction of number plates from vehicles.

They used image processing techniques like histogram equalization and grey-scale conversions to deal with low-resolution images. Candidate number plate areas were extracted, and then true number plate areas were extracted. Character regions were enhanced, and background pixels were weakened.

Further character segmentation is done to get information about each number plate character. Then, finally, SVM was used to classify each character properly. The final results showcased an accuracy of around 85%. However, there is no mention of the dataset used for training and

testing the system, which could be a limitation in evaluating the performance of the proposed approach. Waranusast et al. (2013) [21], in their work suggested a four step process to automatically identify motorcycles and determine whether they are wearing helmets or not.

Utilizing machine vision methodologies, the system employs algorithms to extract dynamic entities from the scene, distinguishing between motorcycles and other objects. Following this differentiation, it proceeds to enumerate and segment the heads of riders.

Subsequently, a comprehensive analysis is conducted to determine helmet usage, facilitated by a K-Nearest Neighbor (KNN) classifier. This classifier utilizes distinct features extracted from the segmented head regions to discern whether a helmet is present or not.

Through this iterative process, the system effectively identifies motorcyclists, segments their heads, and evaluates helmet compliance. However, the paper does not discuss the model performance under different lighting conditions or presence of occlusion.

Rupesh Chandrakant et al. (2022) [7] used a pre-trained model that uses the YOLO algorithm to detect whether the rider is wearing a helmet or not. Weights were tweaked as per the requirements. The authors created the dataset to ensure relevant data availability.

An accuracy of 96% and a frame detection time of around 1.35 sec were achieved. However, there is a lack of diversity in the dataset, including variations in lighting conditions, camera angles, and different types of helmets, which may limit the generalizability of the model.

V, Sri Uthra, et al. (2020) [20] presented significant findings where the paper proposed a motorcycle detection and classification method, helmet detection and helmet detection, and license plate recognition. Vehicle Classification was performed using an SVM classifier.

Helmet detection was done by applying Convolutional Neural Network (CNN) algorithms to extract image attributes, followed by classification using the SVM classifier. License plate recognition was done using Optical Character Recognition (OCR).

**Algorithm 1: Extract Number Plate From the Frame**


---

**Data:** YOLO predictions for image or frame: predictions, YOLO class names: classNames, YOLO class indexes: classIndexes, Image dimensions: imageWidth, imageHeight

**Result:** Cropped license plate regions and extracted text stored in a database

1 Initialize empty arrays: motorcycleBboxes, licensePlateBboxes, withoutHelmetBboxes;

2 **foreach** object in predictions **do**

3     Extract class index, x\_center, y\_center, bbox\_width, bbox\_height of the object;

4     Convert normalized YOLO coordinates to image coordinates using imageWidth and imageHeight;

5     Calculate bounding box lower left and upper right corner points (x\_min, y\_min, x\_max, y\_max);

6     **if** classNames[class index] is “motorcycle” **then**

7         Append (x\_min, y\_min, x\_max, y\_max) to motorcycleBboxes;

8     **else**

9         **if** classNames[class index] is “licensePlate” **then**

10             Append (x\_min, y\_min, x\_max, y\_max) to licensePlateBboxes;

11         **else**

12             **if** classNames[class index] is “withoutHelmet” **then**

13                 Append (x\_min, y\_min, x\_max, y\_max) to withoutHelmetBboxes;

14             **else**

15                 **foreach** motorcycleBbox in motorcycleBboxes **do**

16                     **foreach** licensePlateBbox in licensePlateBboxes **do**

17                         **foreach** withoutHelmetBbox in withoutHelmetBboxes **do**

18                             **if** CheckInsideBoundingBox(licensePlateBbox, motorcycleBbox) **and**  
                               CheckInsideBoundingBox(withoutHelmetBbox, motorcycleBbox) **then**

19                                 Crop the license plate region from the image;

20                                 Send the cropped license plate to an OCR for text extraction;

21                                 Store the extracted text in a database;

---

The system utilized background subtraction and feature extraction using Wavelet Transform. The accuracy for motorcycle classification is 93%, for helmet classification is 85%, and license plate recognition is about 81%. The paper, however, didn't mention computational requirements.

Adil Afzal et al. (2021) [1] introduce a deep learning-based methodology for the automatic detection of helmet wear by motorcyclists in surveillance videos. Leveraging the Faster R-CNN model, the approach involves two phases: helmet detection using the Region Proposal Network (RPN) and subsequent recognition of the detected helmets.

Trained on a self-generated dataset from three distinct locations in Lahore, Pakistan, the methodology achieves a notable 97.26% accuracy in real-time surveillance video analysis. Its strengths lie in the effective utilization of deep

learning techniques, the accuracy afforded by the Faster R-CNN model, and the realism added by the use of a self-generated dataset from actual surveillance footage.

However, limitations include the lack of detailed information on addressing challenges like low resolution and varying weather conditions, limited generalizability to other locations or datasets, and a lack of discussion on the computational requirements and scalability of the proposed methodology.

Further, Mamidi Kiran Kumar et al. (2023) [9] use the YOLO Darknet deep learning framework to automate the detection of motorcycle riders wearing helmets from images, simultaneously triggering alerts for non-compliance. Through bounding boxes and confidence scores, the model identifies regions of interest like riders, helmets, and number plates.



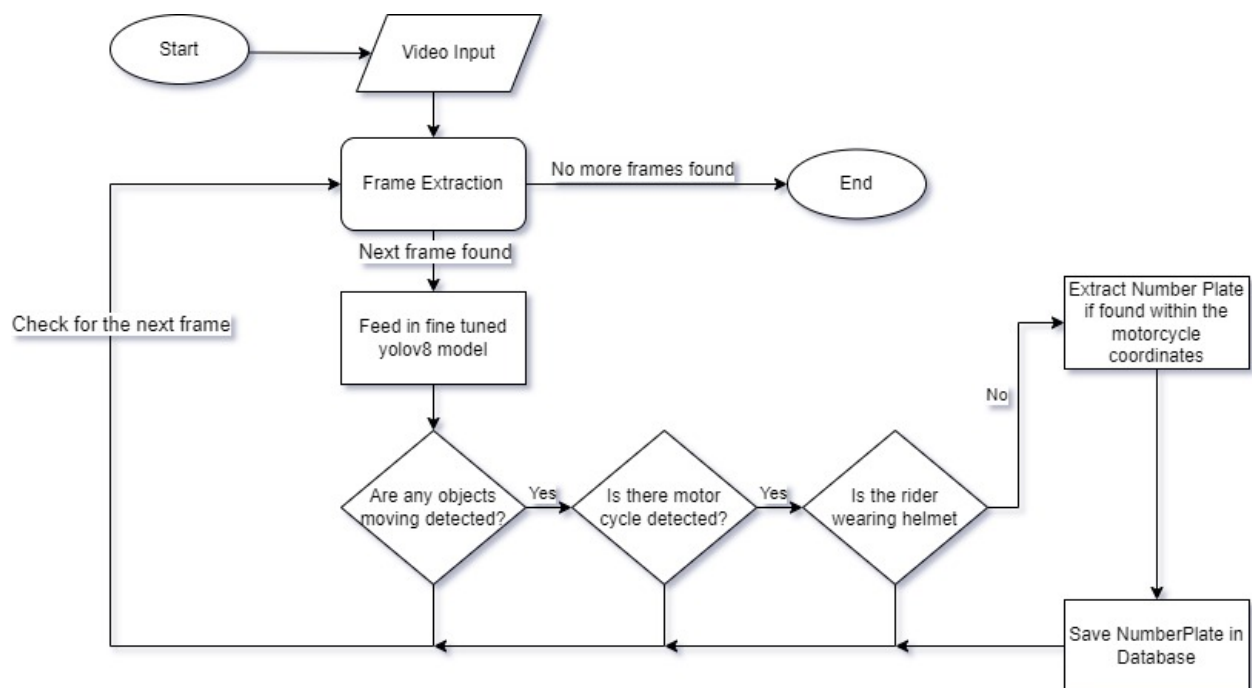


Fig. 4. Flowchart of the proposed solution

The dataset used for training encompasses a diverse collection of images with 80 object categories, capturing a broad spectrum of real-world scenarios. The strengths of the model lie in its automated and efficient solution for helmet detection, eliminating the need for manual checks, and its utilization of the YOLO Darknet framework, enabling real-time detection and alert generation.

However, the limitations include the absence of detailed information on performance metrics or evaluation results, making it challenging to assess the model's accuracy, and a lack of specificity about the training dataset, raising concerns about its representativeness and potential biases.

### 3 Dataset

For our work we collected various images by ourselves and annotated them, we used state-of-the-art YOLOv8 for its ability to detect images in a single pass, its speed and efficiency in object detection tasks, we fine-tuned it.

In the subsequent sections, we provide detailed insights into our fine-tuning methodology, including the selection of hyperparameters, the augmentation strategies employed, and the evaluation metrics used to assess the model's performance. Our objective was to harness the power of YOLOv8 to deliver precise and efficient object detection for our application.

#### 3.1 Dataset Statistics

Our dataset<sup>7</sup> was compiled based on both online sources and self-collected data. Since there was no such public repository for bike riders' images, we scrapped various news articles to get images of interest. Figure 2 shows samples of images from our dataset. First, a total of 3155 images were sourced from online, enriching our dataset with diverse visual data for comprehensive model training. These images were already annotated to serve our purpose.

<sup>7</sup>Dataset Link: [github.com/Jyoti764/Helmet-Violation-Detection-Dataset](https://github.com/Jyoti764/Helmet-Violation-Detection-Dataset)

**Table 2.** Evaluation Metrics

Class	Images	Instances	Box(P)	R	mAP50	mAP50-95	Correct Instances
all	726	2600	0.932	0.907	0.936	0.751	2402
licensePlate	726	762	0.946	0.966	0.964	0.755	737
motorcycle	726	819	0.924	0.939	0.952	0.845	778
withHelmet	726	686	0.902	0.834	0.887	0.672	586
withoutHelmet	726	333	0.955	0.888	0.939	0.733	301

Further, we collected 12 videos from outside the National Institute of Technology, Silchar, campus. The images were then annotated using the Roboflow<sup>8</sup> online annotation tool. Also, various image augmentation techniques were applied so that we could further diversify our dataset and ensure the model remained robust and had good generalizability.

Techniques like flips, rotation, blur, and adjusting the values of RGB channels were employed to achieve a total of 3600 images among the self-collected data. In total, we amassed a total of 6755 images, among which 3600 were self-collected and self-annotated, and 3155 were outsourced from Roboflow as in 1.

### 3.2 Augmentations Applied on Images

We used various augmentation techniques to improve the training and diversify the dataset.

We applied horizontal flip, coloured images were augmented to grayscale images to simulate nighttime CCTV video feeds or images, rotation was applied with a magnitude between  $-15^\circ$  to  $+15^\circ$  shear was done randomly with a magnitude between  $-16^\circ$  to  $+16^\circ$  in the horizontal direction and  $-23^\circ$  to  $+23^\circ$  in the vertical direction, hue and saturation of the images were changed between  $-25^\circ$  to  $+25^\circ$  gaussian blur was applied to the extent of 0.75 pixels, brightness of images were changed between  $-25\%$  to  $+25\%$  and lastly noise was added to 5% of the pixels.

Figure 3 shows various augmentations applied on a sample image from the dataset.

<sup>8</sup>Roboflow: [roboflow.com/](https://roboflow.com/)

### 3.3 Dataset Annotation and Validation

We utilized the online Roboflow annotation tools to label the images nicely in YOLO format for the image annotation. This annotation format is useful for object detection tasks, as it divides the image in a grid and assigns bounding boxes to objects in those grid cells.

The Roboflow annotation tools provided us with an interactive interface to accurately mark and label objects of interest in the images. Also, we used Roboflow's generation tools to apply augmentations. For dataset validation, we inspected and verified the annotated dataset using the built-in validation features of the Roboflow tool.

The tool provides a visual graphic of the annotations, allowing us to quickly verify the completeness of the labelled objects. This manual validation step was important for ensuring the dataset's quality and removing potential errors in annotations.

## 4 Methodology

Our first step involves segmenting the video into consecutive frames and then applying some image processing techniques for better inference. We are using the Open Source Computer Vision Library (OpenCV)<sup>9</sup> library to first read the video as consecutive frames.

Then, for each frame, we are first resizing the frame to the YOLO input standard size (480) and then applying the following transformations:

<sup>9</sup>OpenCV: [github.com/opencv/opencv](https://github.com/opencv/opencv)

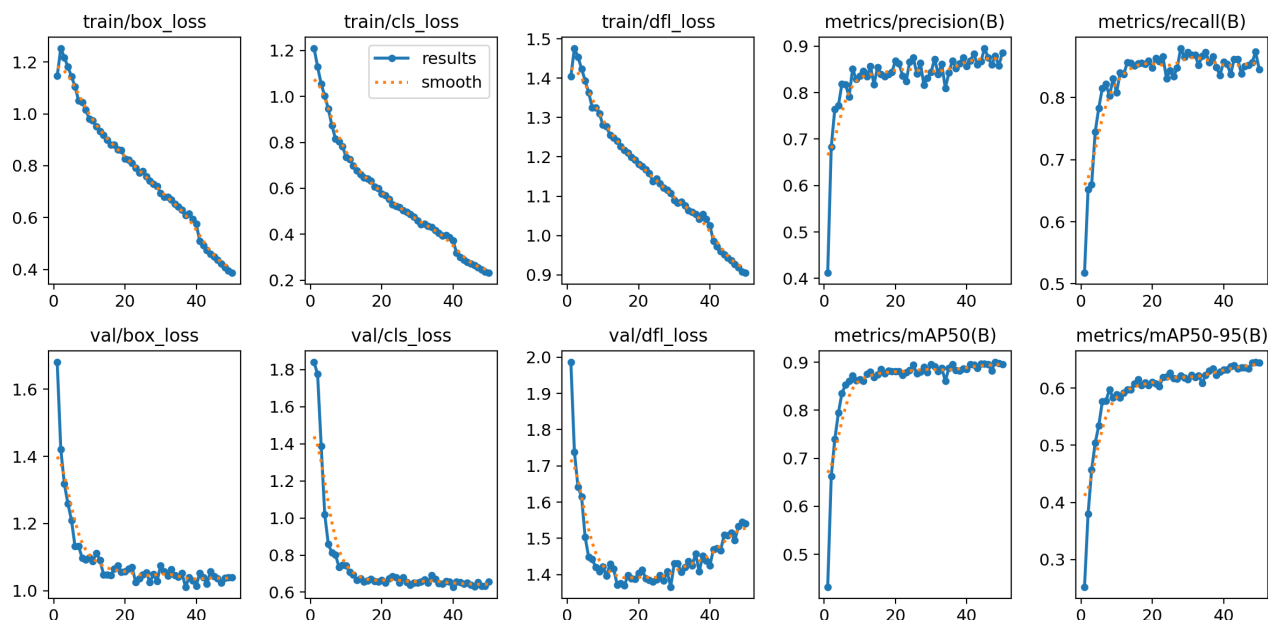


Fig. 5. Training and Validation Metrics

- Grayscale conversion: Grayscale conversion uses the values of the RGB channel and then calculates the pixel value using the following formula:

$$\text{Grayscale} = 0.299 R + 0.587 G + 0.114 B. \quad (1)$$

- Histogram Equalization [4]: This method is performed after grayscale conversion. It is a method that improves an image's contrast to stretch out the intensity range.

Equalization implies mapping one distribution (the given histogram) to another distribution (a wider and more uniform distribution of intensity values) so the intensity values are spread over the whole range. The remapping should be the cumulative distribution function to accomplish the equalisation effect. To use this as a remapping function, we have to normalize such that the maximum value is 255.

- Gaussian blur [5]: This blur focuses on taking a weighted mean, where neighbourhood pixels that are closer to the central pixel contribute more “weight” to the average. This generally helps in removing noise from our image.

Then, we are using background subtraction [6] to separate the forward mask from the image. Then, to enhance the quality of the mask, we apply morphological transformations and then extract the contours beyond a threshold. Thus, the first step is complete, as we have the bounding boxes of all the moving objects in the frame.

This ensures that, in any case, non-moving objects shouldn't get selected in a frame. Our second step involves passing the frame through our fine-tuned YOLOv8 model. To prevent repeated boxes from being sent, we are first taking the union of intersecting boxes. Then, all the bounding boxes of moving objects corresponding to that frame are sent to the model. The model detects the image in four classes of objects, namely: “Motorcycle”, “WithHelmet”, “WithoutHelmet”, and “NumberPlate”.

With reference to the algorithm 1 and figure 4, our model first stores the information of the bounding boxes of the number plates, motorcycles, and WithoutHelmet classes. Then, for every motorcycle's bounding box, we are only considering the top 40% section of the box, as this serves as the most likely region where we are going to find a rider's head.





**Fig. 6.** Inferences from the model. (a) A biker with Helmet. (b) A Biker without Helmet. (c) Extracted Numberplate. Here (c) is the numberplate extracted from (b), i.e. without helmet biker

Then, for each without Helmet and Numberplate class, we check if both exist within the motorcycle's bounding boxes. If this is the case, then we are sure that one of the riders on the bike is not wearing a helmet, and the numberplate detected also belongs to that motorcycle itself. So, the Numberplate coordinates can be extracted and saved for further inference.

#### 4.1 Model Parameters

We used Adam Optimizer [8] during the training process. The learning rate, momentum, and weight decay are set to 0.00125, 0.8, and 0.0005 for 104 weights and 0.0 for 97 weights, respectively. The number of epochs was 55, and the batch size was set to 16.

We evaluate the mean Average Precision of the object detection to measure the performance of our model. The Intersection Over Union (IOU) [14] threshold range for measuring the accuracy of predicted bounding boxes relative to ground truth has been set to 0.50 to 0.95, with an interval of 0.05.

## 5 Results

In object detection, precision, recall, and mAP are commonly used metrics to evaluate the performance of a model such as YOLO. Precision, recall, and mAP can be defined as follows:

Precision is a measure of the accuracy of positive predictions made by an object detection model. It is defined as the ratio of true positives to the total predicted positives. The precision formula for object detection is given by:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \quad (2)$$

True Positives are correctly predicted positive instances, and false positives are those predicted as positive but actually negative. In the context of object detection, a "positive" prediction typically means the model correctly identified and localized an object of interest.

Recall, also known as sensitivity or true positive rate, is a measure of the ability of an object detection model to capture all relevant instances. It is defined as the ratio of true positives to the total actual positives. The recall formula for object detection is given by:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \quad (3)$$

where true positives are the correctly predicted positive instances and false negatives are the instances that are actually positive but were predicted as negative. Recall helps assess how well the model captures all instances of the objects in the dataset. The mAP at IoU 0.5 is calculated by averaging the precision values at a specific IoU threshold (commonly set to 0.5) for each class.

TARGET \ OUTPUT	licensePlate	motorcycle	withHelmet	withoutHelmet	background
	licensePlate	737 25.99%	0 0.00%	0 0.00%	0 0.00%
motorcycle	0 0.00%	778 27.43%	1 0.04%	0 0.00%	83 2.93%
withHelmet	0 0.00%	0 0.00%	586 20.66%	7 0.25%	89 3.14%
withoutHelmet	0 0.00%	0 0.00%	7 0.25%	301 10.61%	17 0.60%
background	25 0.88%	41 1.45%	92 3.24%	25 0.88%	0 0.00%

Fig. 7. Confusion Matrix

The precision at IoU is calculated using the precision-recall curve. The formula is given by:

$$\text{mAP@50} = \frac{1}{C} \sum_{i=1}^C \text{AP}_i^{50}, \quad (4)$$

where  $C$  is the total number of classes and  $\text{AP}_i^{50}$  is the Average Precision at IoU 0.5 for class  $i$ . The mAP from IoU 0.5 to 0.95 with a step of 0.05 is calculated by averaging the precision values over a range of IoU thresholds for each class. The precision at each IoU threshold is calculated using the precision-recall curve. The formula is given by:

$$\text{mAP@50:95} = \frac{1}{C} \sum_{i=1}^C \frac{1}{10} \sum_{t=50}^{95} \text{AP}_i^t, \quad (5)$$

where  $C$  is the total number of classes,  $t$  represents the IoU threshold (from 50 to 95 with a step of 5), and  $\text{AP}_i^t$  is the Average Precision at IoU  $t$  for class  $i$ .

## 5.1 Testing Results

The mAP serves as a performance metric, with higher values generally indicating better overall object detection accuracy. Further analysis and adjustments may be considered to optimize and enhance model performance.

## 5.2 Training Results

The model training dataset comprises a total of 6755 images. The dataset is divided into three subsets: the testing, validation, and training sets. The testing set consists of 726 images, serving as a separate portion for assessing the model's performance. The validation set, consisting of 755 images, is employed for fine-tuning and parameter optimization during the training process.

The majority of the dataset, totalling 5274 images, forms the training set, providing the foundation for training the model to recognize and generalize patterns from the input images. Figure 5 shows metrics for training and validations.

Our model underwent evaluation on a diversity dataset containing a total of 726 images with a total of 2600 instances across all classes, achieving promising results across all classes. Figure 6 shows some of the inferences from our model. The overall performance, as indicated by the “all” class, demonstrated high mAP50 of 93.6% and mAP50-95 of 75.1%, contributing significantly to the robustness of the model.

The motorcycle class also exhibited strong performance, achieving a mAP50 of 95.2%. Additionally, the model performed well in identifying instances of withHelmet and withoutHelmet, showcasing its versatility in handling diverse scenarios in object detection tasks. Further, the overall performance metrics are shown in the Table 2 and figure 7. However, our model showed variations in performance across different classes.

Though licensePlate and motorcycle classes achieved outstanding results, the withHelmet and withoutHelmet classes showed lower precision and recall values, indicating potential room for optimization. The model speed, with preprocessing, takes 0.8 milliseconds, inference takes 29.2 milliseconds and postprocessing consumes 3.5 milliseconds per image, showing its efficiency in real-time applications.

In summary, our model with YOLOv8 architecture demonstrated high accuracy in detecting and localizing objects across multiple classes. The detailed class-wise metrics provide insights into the model’s strengths and areas for refinement, informing potential adjustments or fine-tuning strategies to enhance its overall performance.

## 6 Conclusion

This paper presented the development and evaluation of our fine-tuned YOLOv8 model for detecting without helmets bike riders and extracting their number plates. We employed various augmentation techniques to improve the accuracy and robustness of our model. The result shows a high mAP50 score of 0.936 on the testing data, correctly labelling the majority of the classes regardless of lighting and weather conditions of the images or videos showcasing

the working of the model under diverse scenarios. Our model can also be efficiently deployed in real-time applications to monitor traffic in cities and highways. This model will help law enforcement agencies enforce laws on helmets properly and reduce the incidence of fatalities resulting from failure to wear helmets, undeniably contributing to saving lives.

Further improvements can be made by increasing the size of the dataset. We anticipate that our efforts will serve as a catalyst for additional investigations in this field, fostering the creation of models that are more precise and more efficient in enhancing safety for individuals on motorcycles, including riders, passengers, and fellow commuters on the road.

## Acknowledgments

We extend our thanks to the Computer Science & Engineering Department at the National Institute of Technology Silchar for enabling us to pursue our research and experiments. Additionally, we acknowledge the support and resources offered by the Center for Natural Language Processing (CNLP) and Artificial Intelligence (AI) laboratories and the conducive research environment.

## Declarations

**Data Availability** The authors declare that their data will be made available on request.

**Conflict of Interest** The authors declare that they have no conflict of interest.

## References

1. Afzal, A., Draz, H. U., Khan, M. Z., Khan, M. U. G. (2021). Automatic helmet violation detection of motorcyclists from surveillance videos using deep learning approaches of computer vision. *Proceedings of the International Conference on Artificial Intelligence*, pp. 252–257. DOI: 10.1109/ICA152203.2021.9445206.

2. **Camps-Valls, G., Reichstein, M., Zhu, X., Tuia, D. (2020).** Advancing deep learning for earth sciences: From hybrid modeling to interpretability. *IEEE International Geoscience and Remote Sensing Symposium*, pp. 3979–3982. DOI: 10.1109/IGARSS39084.2020.9323558.
3. **Dahiya, K., Singh, D., Mohan, C. K. (2016).** Automatic detection of bike-riders without helmet using surveillance videos in real-time. *Proceedings of the International Joint Conference on Neural Networks*, pp. 3046–3051. DOI: 10.1109/IJCNN.2016.7727586.
4. **Garg, P., Jain, T. (2017).** A comparative study on histogram equalization and cumulative histogram equalization. *International Journal of New Technology and Research*, Vol. 3, No. 9.
5. **Gedraite, E. S., Hadad, M. (2011).** Investigation on the effect of a gaussian blur in image filtering and segmentation. *Proceedings of the International Symposium on Electronics in Marine*, pp. 393–396.
6. **Goyal, K., Singhai, J. (2017).** Review of background subtraction methods using gaussian mixture model for video surveillance systems. *Artificial Intelligence Review*, Vol. 50, No. 2, pp. 241–259. DOI: 10.1007/s10462-017-9542-x.
7. **Jaiswal, R., Srushti, C., Deo, V. (2023).** Helmet detection using machine learning. *International Journal of Emerging Technologies and Innovative Research*, Vol. 9, pp. d10–d17.
8. **Kingma, D., Ba, J. (2015).** Adam: A method for stochastic optimization. pp. 1–15. DOI: 10.48550/arXiv.1412.6980.
9. **Kiran-Kumar, M., Sanjana, C., Shireen, F., Harichandana, D., Sharma, M., Manasa, M. (2023).** Automatic number plate detection for motorcyclists riding without helmet. *E3S Web of Conferences*, Vol. 430, pp. 01038. DOI: 10.1051/e3sconf/202343001038.
10. **Liu, B. C., Ivers, R., Norton, R., Boufous, S., Blows, S., Lo, S. K. (2008).** Helmets for preventing injury in motorcycle riders. Wiley. DOI: 10.1002/14651858.cd004333.pub3.
11. **Lundberg, I., Brand, J. E., Jeon, N. (2022).** Researcher reasoning meets computational capacity: Machine learning for social science. *Social Science Research*, Vol. 108, pp. 102807. DOI: 10.1016/j.ssresearch.2022.102807.
12. **Meenu, R., Sinta, R., Smrithi, P. P., Swathy, S., Alphonsa, J. (2020).** Detection of helmetless riders using faster R-CNN. *International Journal of Innovative Science and Research Technology*, Vol. 5, No. 5, pp. 1616–1620.
13. **Mistry, J., Misraa, A. K., Agarwal, M., Vyas, A., Chudasama, V. M., Upla, K. P. (2017).** An automatic detection of helmeted and non-helmeted motorcyclist with license plate extraction using convolutional neural network. *Proceedings of the 7th International Conference on Image Processing Theory, Tools and Applications*, pp. 1–6. DOI: 10.1109/ipta.2017.8310092.
14. **Nowozin, S. (2014).** Optimal decisions from probabilistic models: The intersection-over-union case. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 548–555. DOI: 10.1109/cvpr.2014.77.
15. **Padilla, R., Netto, S. L., da-Silva, E. A. B. (2020).** A survey on performance metrics for object-detection algorithms. *Proceedings of the International Conference on Systems, Signals and Image Processing*, pp. 237–242. DOI: 10.1109/iwssip48289.2020.9145130.
16. **Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016).** You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788. DOI: 10.1109/CVPR.2016.91.
17. **Sathe, P., Rao, A., Singh, A., Nair, R., Poojary, A. (2022).** Helmet detection and

number plate recognition using deep learning. IEEE Region 10 Symposium, pp. 1–6. DOI: 10.1109/tensymp54529.2022.9864462.

**18. Shailaja, K., Seetharamulu, B., Jabbar, M. A. (2018).** Machine learning in healthcare: A review. Proceedings of the 2nd International Conference on Electronics, Communication and Aerospace Technology, pp. 910–914. DOI: 10.1109/ICECA.2018.8474918.

**19. Shidore, M. M., Narote, S. P. (2011).** Number plate recognition for indian vehicles. International Journal of Computer Science and Network Security, Vol. 11, No. 2, pp. 143–146.

**20. Sri, U. V., Sariga, D. V., Vaishali, K. S., Padma, P. S. (2020).** Helmet violation

detection using deep learning. International Research Journal of Engineering and Technology, Vol. 7, pp. 3091–3095.

**21. Waranusast, R., Bundon, N., Timtong, V., Tangnoi, C., Pattanathaburt, P. (2013).** Machine vision techniques for motorcycle safety helmet detection. Proceedings of the 28th International Conference on Image and Vision Computing New Zealand, pp. 35–40. DOI: 10.1109/IVCNZ.2013.6726989.

*Article received on 16/11/2023; accepted on 29/01/2024.  
\* Corresponding author is Partha Pakray.*