



## A Smart System for Detection of Helmets and Number Plate Recognition using AI

Prof.Pramod V Patil<sup>1</sup>, Prof. Sachin S Patil<sup>2</sup>, Prof.Prasanna Patil<sup>3</sup>

<sup>1,2,3</sup>Department of Electronics and communication

Engineering

HSIT Nidasoshi,

<sup>1</sup>[pramodpatil.ece@hsit.ac.in](mailto:pramodpatil.ece@hsit.ac.in)

<sup>2</sup>[sachinpatil.ece@hsit.ac.in](mailto:sachinpatil.ece@hsit.ac.in)

### Abstract

Now a day's two wheelers is the most preferred mode of transport. Motorcycles have always been the primary mode of transportation in developing countries. Motorcycle accidents have increased in recent years. One of the main reasons for fatalities in accidents is that a motorcyclist does not wear a protective helmet. The most common way to ensure that motorcyclists wear a helmet is by traffic police to manually monitor motorcyclists at road junctions or through CCTV footage and to penalize those without a helmet. But it requires human intervention and effort.

This system proposes an automated system for detecting motorcyclists who do not wear a helmet and a system for retrieving motorcycle number plates from CCTV video footage. First, the system classifies moving objects as motorcycling or non-motorcycling. In the case of a classified motorcyclist, the head portion is located and classified as a helmet or non-helmet. Finally, the motorcyclist without a helmet is identified. Further we have developed a system which identifies the number plates and extracts the characters of the number plate using OCR algorithm.

**Keywords**—YOLO, OCR, ANPR, Helmet detection, and Number plate recognition

### INTRODUCTION

We come across numerous incidents where motorcyclist's gets even rely injured simply because they didn't wear safety helmets. This negligence has caused a threat to many innocent lives. In India, six two-wheeler riders die every hour in road accidents. Despite having strict traffic regulations, people still neglect the importance of wearing a helmet. Also,

We used object detection deep learning algorithms like YOLO v5 the existing surveillance system requires significant human assistance and humans are prone to make mistakes. So, automating this system is highly desirable. For doing so, We are dealing with a variety of motorcyclists with distinct colors of clothes, helmets, and angles of motorcyclists. To achieve this, we require a deep neural



network that will help to determine the motorcyclist very accurately. We mainly aim to collect a database of all the motorcyclists who have violated the rules. One of the key problems we faced was determining whether the person is wearing a helmet or not and to differentiate between biker and pedestrian. To solve this problem, we have come up with two methods for finding the same. In the first method, we check the overlap between the classes and create a link between the bike rider and the number plate. In the second method, depending upon the height of the motorcycle bounding box, it will check if a helmet class exists at a particular distance above the motorcycle.

In conclusion, the system will determine whether the motorcyclist is wearing a helmet or not. And depending up on the answer, it would extract the number plate soft hose motor cyclists who are not wearing helmets.

## II. LITERATURESURVEY

### A. *Helmet Detection*

For the past few years, many algorithms and models have been used for helmet detection .J.Chiverton *etal.*[1]used the background subtraction method to separate the background of the bikers and then isolate the head of the biker and identify the features of the helmet using SVM with a linear kernel. Z.Chen *et al.* in [2] used a Multi-dimensional Gaussian Kernal Density Transform (MDGKT) and a self-adaptive Gaussian mixture model for background subtraction. In the above-mentioned papers, the model will show errors in case there are any changes in background illumination or environmental conditions. To overcome this, authors in [5] have used multiple Gaussian models.

As these papers used geometrical features to identify a helmet, it can detect any other object as a helmet with similar geometric features. In the above-mentioned papers, the helmet detection accuracy rate is low and is not practically feasible to compute such models. With the development in computer hard ware and GPU, object detection using deep learning models is more accurate and faster than traditional methods. Among all the models, [4]R-CNN is a region proposal-based convolution neural networks algorithm that uses CNN to detect the object. A. Adilet *al.* in [9] have used the Faster R-CNN method for helmet detection The model works in two phases, one is the Region Proposal Network (RPN) which draws anchor boxes on images to detect objects and the most feasible one which consists of the biker's head is used for recognition in the second phase. Another CNN-based algorithm is detect ron 2 used by Ritik Singh *etal.*in [11] which uses masked annotation of objects and detects objects based on their features. Since it detects based on features, it detects other round objects such as a bald person as a helmet.You Look Only Once(YOLO)is a deep learning algorithm that can be implemented in real-time, as used by authors in[6].It forms a grid on the image and if the center point of the object falls on any one of the grids it selects that grid to make a bounding box and detects



the object. Siebert *et al.* [7] proposed the SSD algorithm (combination of the anchor box of Faster R-CNN and bounding box regression of YOLO) with high accuracy and high computing speed but a complex algorithm. Authors in [12] have used YOLO v5 to perform object detection on the COCO dataset. YOLO v5 shows promising results in real-time detection.

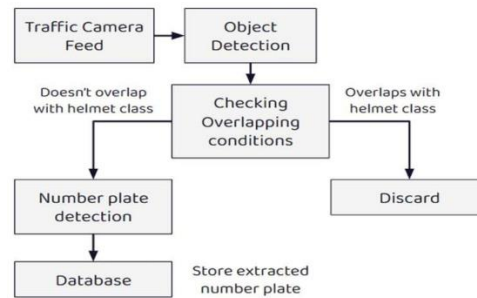
Deep Learning algorithms are more efficient and accurate than traditional methods. Traditional methods of deep learning consume more disk space and time, with slow speed and less accuracy. Hence, we present a deep learning method of object detection (YOLOv5) for object detection.

### ***B. Number plate Recognition***

Authors in [3] have developed an Automatic Number Plate Recognition (ANPR) with Optical Character Recognition (OCR), where mechanical or electronic conversion of scanned images takes place. According to the survey conducted by authors in [3] Image binarization, Sliding concentric window (SCW), Sobel operator, Canny edge operator, Hough Transform (HT), Probabilistic neural network (PNN) and dichromatic imaging with a color-discrete characteristic approach can provide promising results for number plate detection. The conventional number plate recognition systems use sliding window approaches. The major drawback of this system was unnecessary computation. [8] Here, author Yon ten J. *et al.* used YOLO v2 for number plate detection and ANPR for recognizing number plate characters but the computational speed was less and generated error due to bad illumination. For character recognition, author Tushar Goel *et al.* [10] used Python-Tesseract which is an optical character recognition (OCR) tool for python i.e. it will recognize and read the text embedded in images. A study mentioned in [18] shows a comparative study between Tesseract OCR, Easy OCR, and Keras OCR. All three show promising results but Easy OCR is a light weight model with good results and performance. Hence, we present number plate recognition with Easy OCR which is best suited for natural geographical conditions.

### **III. MATERIALS AND METHODOLOGY**

In our project, we have trained two YOLOv5 models. The first model is of Helmet and Person class and the second is of Number Plate and Motorcycle class using the transfer learning method. As shown in the flowchart in Fig 1, we take frames of images from the camera feed and store them in a folder. Further, we used to give these images as input to our trained YOLOv5 model of helmet and motorcycle for object detection. Later the output of this model is fed as input to the trained YOLO v5 model of person and helmet. After detection, we obtain the bounding box coordinates of each detected object. These coordinates are then used to check to overlap between all 4 classes.



**Fig.1.Flow chart for detecting person not wearing a helmet**

The obtained coordinates from object detection are first converted to normal form. Once done, it is stored in a data frame. We have proposed two methods to check if the biker is wearing a helmet or not. The first method is to check over lapping between detected classes. In order to check overlap, we check if the area of inter section between the two boxes is positive. This is done 3 times, firstly over lapping between Helmet and Person is checked. If overlap exists between Motorcycle and Number Plate then overlapping between Person and Motorcycle is checked. Further, it will check for Helmet coordinates. If helmet coordinates are present then we check for overlapping and if it overlaps then the biker is wearing a helmet. And if helmet coordinates are not present then the biker is not wearing a helmet and we crop the number plates using number plate coordinates. This cropped image is then given as input to the Easy OCR model for character recognition. Thus, we obtain the Number Plate characters of the biker not wearing a helmet. In the second method, we set a range using motorcycle coordinates and check if the helmet coordinates fall inside this range. If it does, then the biker is wearing a helmet and if it doesn't then we extract the number plate characters as mentioned in the first method.

### ***A. Object Detection***

#### ***1) Collection of Dataset***

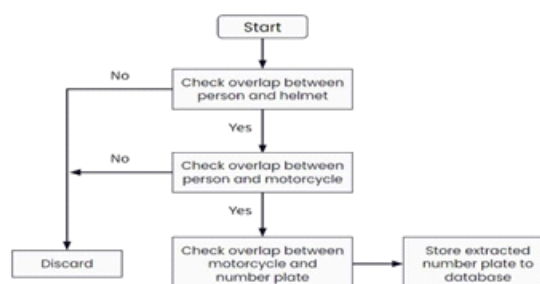
According to the requirements, we decided we needed 4 classes in total, i.e. Helmet, Number plate, Motorcycle, and Person. We have used 4,000 images in total to train these classes, 1,000 images for each class. We obtained the images and their annotation files using the Open Image Dataset (OIDv6)[13]. For annotating the images we used, makesense.ai [14] and LabelImg[15] tools. Makesense.ai is a free-to-use online tool whereas LabelImg is python generated offline tool for labeling photos. Since we are using the YOLOv5 algorithm we have obtained the labels in the Yolo format as a ".txt" file.

We also used RoboFlow[16] for pre-processing and augmenting our data set to increase the dataset. Once our data set was ready, we split it into train-test folders.



### 1) Training Testing

We have used transfer learning for training our model. Transfer learning is when we use previously trained model weights and add new layers of custom weights during the training. It helps in increasing the accuracy of our model without increasing the training time of the model. In our model, we have used yolov5 pre-trained weights by Ultralytics[17] which were trained on the COCO 2017 dataset. We installed different dependencies like matplotlib, PyTorch, etc. for training and testing purposes, and for plotting the metrics of our training we have used w and b[19]. W and b (Weights and Biases) is a tool that is used to monitor the progress of our training in real-time. Metrics in the w and b site show the variation in precision and loss of our model. Next, we have changed hyper parameters in the configuration file according to our custom dataset. After saving we proceeded with training. We propose two methods for finding out whether a person is wearing a helmet or not. In method 1, we first detect all four classes - helmet, person, motorcycle, and number plate using YOLOv5. Training the model with a higher batch size requires more GPU. Therefore, we have kept the batch size value as 8. We kept the learning rate as small as possible (i.e. 0.001) to obtain the best accuracy at each step size. We observed while training, that the AP of the model, when trained with all four classes, was less compared to the mAP of the model when trained with two classes at a time. Thus, we split our models into two - one for detecting helmet and person classes and the other model for detecting motorcycle and number plates. We trained with YOLO v5s weight and finally obtained an mAP value of 0.943 for the Motorcycle and Number plate model, and for Person and Helmet, we obtained an mAP value of 0.945.



**Fig2. Flow chart for detecting if a person is wearing a helmet (Method1)**

After object detection, we get bounding box coordinates for all the four classes- helmet, number plate, motorcycle, and person. YOLOv5 gives the coordinates in YOLO format. For making the further process easier, we need to convert these coordinates from YOLO to PASCAL VOC. Once we get all coordinates for all the bounding boxes, we aim to find the biker not wearing a helmet to extract his/her number plate. To check this we use the Overlapping method. The overlapping function helps





us to calculate the intersection area between the 2 bounding boxes. For doing so, we first count the number of occurrences of Person class and then check if the Helmet class and Person class overlap, and if it does, it means that the biker is wearing a helmet as shown in Figure 3.a. and we can discard that coordinate and move to the next one. If no overlap is found, the person can be a regular pedestrian or a non-helmeted rider. To rule out the person being a pedestrian, we would again check the overlap between Person class and Motorcycle class and if it doesn't overlap, it means it is a regular pedestrian. If the overlap exists, then this implies that the person is riding the bike as shown in Figure 3.b. Further for the final step, we need to extract its number plate. So again, we check the overlap between the Motorcycle class and Number plate (Figure3.c). Thus, we have successfully extracted the number plate of those bike riders who are not wearing helmets.



Fig 3.a Checking Overlap between helmet and person class Method1



Fig3.b Checking Overlap between person and motorcycle Method 1

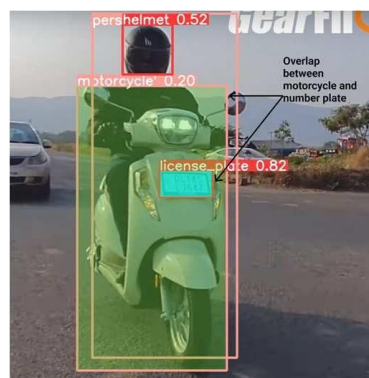


Fig 3.c Checking Overlap between the motorcycle and number plate



In the second method, instead of 4 classes, we require only 3 in this case. So, we have trained the model with 3 classes (helmet, motorcycle, number plate) and obtained an mAP of 0.847. Further to check whether the biker is wearing a helmet or not, is by considering a motorcycle coordinate and set range based on the height and width of the bounding box and check whether the helmet coordinates fall inside the coordinate range of the motorcycle. X-coordinates of the helmet will fall inside the x-coordinate range of the motorcycle where as they-coordinates of the helmet will be on an average 40% above they-coordinates of the motorcycle according to our data set. So, by setting that range we can check whether they-coordinates of the helmet fall inside the range as shown using a yellow-colored shade in Fig.3.d. If it does, it will display “Biker is wearing a helmet” as output and if it does not, it will display “Biker is not wearing a helmet” and check for overlapping between Motorcycle class and Number plate class and extract the characters from the number plate.



Fig 3.d Checking if the helmet is present 40% above the height of the motorcycle- Method2

### A. Character Recognition

As for the final step we have used Easy OCR, we pass the cropped image of the number plate to the OCR model. This model converts the text into a machine-readable format and saves it to the database. At first, it crops the images of the number plate after it is detected and then performs image processing on the image. Further, it checks each character and recognizes its pattern and then it matches with its data and stores the characters in text format. In Easy OCR, if the image captured by the camera doesn't have proper alignment then this library will try to detect the text at every possible angle.

## II. RESULTS AND DISCUSSION

We trained our model with 4 classes - helmet, person, motorcycle, and number plate, 1,000 images per class and obtained an mAP of 0.56. Later, we trained the model with only 3 classes - number plate, motorcycle and helmet. For this case, we obtained an mAP of 0.847. Subsequently, we also trained for 2



models for 2 classes each-one with number plate and motorcycle and the other with helmet and person. We observed the mAP of 0.943 and 0.945 respectively. All the results are shown in Table 1 below. In the above experiment, we observed that, as the number of classes increases, it, in turn, decreases the accuracy of the model. Thus, to tackle this problem, we used 2 models with 2 classes each.

TABLE 1: Mean Average Precision for trained models

No. of classes	mAP
4 classes(h,p,np, m)	0.56
3 classes(np, m,h) <b>Method2</b>	0.847
2 classes(np,m) <b>Method1</b>	0.943
2 classes(h,p) <b>Method1</b>	0.945

np=number plate, h=helmet=motorcycle, p=person, nh=nohelmet

We trained our model with a data set and chose the best epoch that suits our criteria. We achieved different accuracies for different sets of classes as shown in Table 1.

Fig 4.a and b represent the precision and loss graphs that are generated by W and B. We observed an mAP of 0.9 and class loss of 0.00004656 for 2 classes at 150 epochs. Mean Average Precision (mAP) increases with an increase in epochs and loss decreases with an increase in epochs (Fig 4).

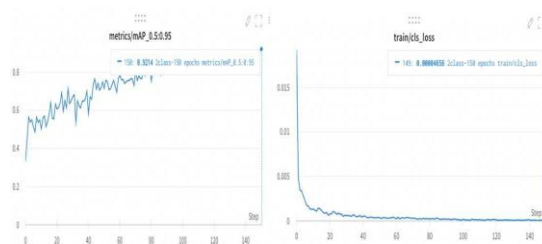


Fig 4. W and b metrics graph

The Mean Average Precision (mAP) for the 4 class detection model was 0.56, and for the 2-class models was 0.995 and 0.945 for the first and the second model respectively. We got to see that the less the number of classes, the more accurate is our model so we opted for two methods. Thus, as the number of classes increases, the accuracy of the model decreases.



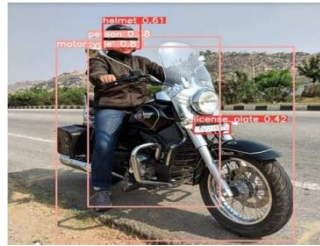


Fig.5.The result after testing using method 1 with 4 classes

In Method1, we trained two different models with two classes each and observed higher accuracy as shown in Fig. 5. Further, we used an overlapping method to check whether the biker is wearing a helmet or not. Consider Fig.6.a, there are two number plates, two motorcycles, and two people in the same image. Since there are multiple objects of the same class present in the image, we consider the coordinates of the first motorcycle and check the overlapping with each number plate. The overlapping number plate and motorcycle will give a positive response as “overlap” as shown below.



Fig.8.Output of method 2 after checking if the biker is wearing helmet or not

After checking for the helmet-wearing bikers, we extracted the number plates of the bikers not wearing helmets using Easy OCR as shown in Fig. 9.



Fig.9.Easy OCR output after number plate recognition

## II. CONCLUSION

We use the YOLOv5 Algorithm which is considered to be the best in real-time object detection algorithm. We have used two methods for detecting whether the motorcyclist is wearing a helmet or not and proceeded with the extraction of the number plate. This number plate character was displayed in text format. This text format is stored in a sheet for violating the law of not wearing a helmet. We



achieved a mAP of 0.995 for detecting objects and overlapping conditions make our work different from any others. Our project makes this system autonomous and makes it easier for the authority to penalize people.

### REFERENCES

- [1] J.Chiverton,“Helmetpresenceclassificationwithmotorcycledetectionandtracking,”IntelligentTransportationSystems(IET),vol.6, no.3,September2012.
- [2] Z.Chen,T.Ellis,andS.Velastin,“Vehicledetection,tracking and classification in urban traffic,” in Procs. of theIEEE Int. Conf. on Intelligent Transportation Systems (ITS),Anchorage, AK,Sept 2012.
- [3] C.Patel,D.Shah,A.Patel,“AutomaticNumberPlateRecognitionSystem”,ArticleinInternationalJournalofComputer Applications,May2013.
- [4] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, “FasterRCNN”Submittedon4Jun2015(v1),lastrevised6Jan2016.
- [5] K.Dahiya,D.Singh,andC.K.Mohan,“Automaticdetectionofbike-riderswithouthelmetusingsurveillancevideos in real-time,” in 2016 International Joint Conference onNeuralNetworks(IJCNN),July2016.
- [6] FanWu,GuoqingJin,“HelmetDetectionBasedOnImproved YOLO V3 Deep Model”, in IEEE 16th InternationalConferenceonNetworking,SensingandControl(ICNSC),May2019.
- [7] Felix Wilhelm Sieberta, Hanhe Linb "Deep Learning BasedSafetyHelmetDetectioninEngineeringManagementBasedonConvolutional Neural Networks", Department of Psychologyand Ergonomics,Berlin,Germany, 2019.
- [8] YontenJ.,PanomkhawnR.,Rattapoom W.,“Real-timenumber plate detection for non-helmetedmotorcyclist usingYOLO”,Received29November2019;receivedinrevisedform3June2020;accepted31July2020,Availableonline22August2020.