Journal for Educators, Teachers and Trainers
The PROFESIOLab electronic, peer-reviewed, open-access Magazine

ISSN 1989-9572

DOI:10.47750/jett.2021.12.01.028

DEEP LEARNING-BASED SYSTEM FOR NON-HELMET RIDER DETECTION AND LICENSE PLATE RECOGNITION TO ENHANCE ROAD SAFETY

Sapthagiri Vienala[1],
Dhiravath Sumitha[1],
Kaki Ajay[1]

Journal for Educators, Teachers and Trainers, Vol.12(1)

https://jett.labosfor.com/

Date of reception: 2 Dec 2021

Date of revision: 1 Jan 2021

Date of acceptance: 30 Jan 2021

Sapthagiri Vienala, Dhiravath Sumitha, Kaki Ajay. (2021). DEEP LEARNING-BASED SYSTEM FOR NON-HELMET RIDER DETECTION AND LICENSE PLATE RECOGNITION TO ENHANCE ROAD SAFETY. Journal for Educators, Teachers and Trainers, Vol.12(1). 231-239.

Journal for Educators, Teachers and Trainers
The LabOSfor electronic, peer-reviewed, open-access Magazine

Journal for Educators, Teachers and Trainers
The PROFESIOLab electronic, peer-reviewed, open-access Magazine

Journal for Educators, Teachers and Trainers, Vol. 12(1)

ISSN 1989 –9572

https://jett.labosfor.com/

DEEP LEARNING-BASED SYSTEM FOR NON-HELMET RIDER DETECTION AND LICENSE PLATE RECOGNITION TO ENHANCE ROAD SAFETY

Sapthagiri Vienala[1], Dhiravath Sumitha[1], Kaki Ajay[1]
[1]Department of Electronics and Communication Engineering
[1]Sree Dattha Group of Institutions, Sheriguda, Hyderabad, Telangana.

Abstract
It is essential to have efficient automatic helmet detection in order to improve the safety of drivers on the road. On the other hand, the automated systems that are now in use frequently fall short in terms of efficiency, accuracy, and speed when it comes to object recognition and classification. This study presents a Non-Helmet Rider detection system with the objective of automating the identification of traffic offenses linked to helmet non-compliance. Additionally, the system is designed to retrieve the license plate numbers of automobiles. Object Detection by Deep Learning is the fundamental principle that underpins this system. This deep learning system functions on three different levels. Individuals, motorbikes and mopeds (at the first level), helmets (at the second level), and license plates (at the third level) are all detected by the system. YOLOv2 allows for the detection of license plates. After that, the number that is printed on the license plate is extracted using a technique called optical character recognition (OCR). The application of all of these strategies is carried out while taking into account the conditions and constraints that have been set, with particular emphasis paid to the process of extracting license plate numbers. As a result of the fact that this system processes image inputs, the speed of execution is of essential significance.
Keywords: Deep Learning, Artificial Intelligence, YOLOv2, YOLOv3, Optical Character Recognition, Road Safety, Non-Helmet Rider Detection, License Plate Extraction.

1. Introduction

A helmet reduces the chances of skull getting decelerated, hence sets the motion of the head to almost zero. Cushion inside the helmet absorbs the impact of collision and as time passes head comes to a halt. It also spreads the impact to a larger area, thus safeguarding the head from severe injuries. More importantly it acts as a mechanical barrier between head and object with which the rider comes into contact. Injuries can be minimized if a good quality full helmet is used. Traffic rules are there to bring a sense of discipline, so that the risk of deaths and injuries can be minimized significantly. However strict adherence to these laws is absent. Hence efficient and feasible techniques must be created to overcome these problems. Manual surveillance of traffic using CCTV is an existing methodology. But here so many iterations must be performed to attain the objective and it demands a lot of human resources. Therefore, cities with millions of populations having so many vehicles running on the roads cannot afford this inadequate manual method of helmet detection. So here we propose a methodology for full helmet detection and license plate extraction using YOLOv2, YOLOv3 and OCR. Basically, helmet detection system involves the following steps such as collection of datasets, moving object detection, background subtraction, object classification using neural networks.

The existing system monitors the traffic violations primarily through CCTV recordings, where the traffic police must investigate the frame where the traffic violation is happening, zoom into the license plate in case rider is not wearing helmet. But this requires lot of manpower and time as the traffic violations frequently and the number of people using motorcycles is increasing day-by-day. What if there is a system, which would automatically look for traffic violation of not wearing helmet while riding motorcycle/moped and if so, would automatically extract the vehicles' license plate number. Recent research has successfully done this work based

on machine learning methods etc. But these works are limited with respect to efficiency, accuracy or the speed with which object detection and classification is done.
Rest of the paper is organized as follows: Section 2 details about literature survey, section 3 details about the proposed methodology, section 4 details about the results with discussion, and section 5 concludes article with references.

2. LiteratureSurvey

Chowdhury, Pinaki Nath, et al (2020) [11] proposed U-Net-CNN which was used for enhancing contrast of license plate pixels. Since the difference between pixels that represented license plates and pixels that represented background is too due to low light effect, the special property of U-Net that extracted context and symmetric of license plate pixels to separated them from background pixels irrespective of content. This process resulted in image enhancement. Rashid, Amr E et al. (2019) [12] proposed a fast algorithm for automatic license plate detection system for the Egyptian license plates that achieved a high detection rate wihigh-qualityed for a high-quality image from expensive hardware. The system captured images of the vehicles with a digital camera. An algorithm for the extraction of the license plate has been explained and designed using Mat lab. We achieved about 96% detection rate for small dataset. Kim, Jung-Hwan, et al (2019) [13] proposed a Gaussian blur filter which was used to remove noise in the image, and they detected the license plate edge using modified Canny algorithm. Second, they determined license plate candidate image using morphology and support vector machine. It recognized numbers and characters using k-nearest neighbour classifier. Abolghasemi, Vahid, et al. (2019) [14] Proposed two different image enhancement methods (using intensity variance and edge density) were proposed. They aimed to increase contrast of plate-like regions to avoid missing plate location especially in poor quality images. The MNS (multimodal neighbour hood signature) method is used. A well-organized database, consisting of car images with different known distances and viewing angles, was prepared to verify the performance of plate detection algorithm. Amaanullah, Rizki Rafiif, et al (2022) [15] proposed a three transfer learning models, namely DenseNet121, MobileNetV2, and NAS Net Mobile models. The experiment in this research was carried out using the data on number plates in the parking lot. The accuracy calculation counted the number of correctly recognized characters divided by the total characters on the number plate. The experimental results show that the DenseNet121 model produced the best accuracy, 96.42%.
Khan, Ishtiaq Rasool, et al (2022) [16] proposed a transfer learning approach to train the recently released YOLOv5 object detecting framework to detect the LPs and the alphanumerics. Next, we train a convolutional neural network (CNN) to recognize the detected alphanumerics. The proposed technique achieved a recognition rate of 92.8% on a challenging proprietary dataset collected in several jurisdictions of Saudi Arabia. This accuracy is higher than what was achieved on the same dataset by commercially available Sighthound (86%), Plate Recognizer (67%), Open ALPR (77%), and a state-of-the-art recent CNN model (82%). The proposed system also outperformed the existing ALPR solutions on several benchmark datasets. Tham, Mau-Luen, et al. (2021) [17] proposed OpenCV image processing is invoked to segment the characters of each image instance before feeding them into the Tesseract optical character recognition (OCR) engine for character recognition. Fifth, a weighted selection algorithm is designed to choose the best car plate number among the pooled samples. Lastly, the entire solution is deployed in the Up Squared board and powered by the popular IoT Node-Red. Sharma, Jitendra, et al (2019) [18] proposed a new methodology for `License Plate Recognition' based on wavelet transform function. This proposed methodology compares with Correlation based method for detection of number plate. Empirical result shows that better performance in comparison of correlation-based technique for number plate recognition. Here, it is modified the Matching Technique for numberplate recognition by using Multi-Class RBF Neural Network Optimization. Ho, Wing Teng, et al. (2019) [19] proposed two-stage method to detect license plates in real world images. An initial set of possible license plate character regions were obtained by the first stage classifier and then passed to the second stage classifier to reject non-character regions. 36 Ada boost classifiers (each trained with one alpha-numerical character, i.e., A... Z, 0...9) served as the first stage classifier. In the second stage, a support vector machine (SVM) trained on scale-invariant feature transform (SIFT) descriptors obtained from training sub-windows were employed. Gunawan, Dani, et al. (2019) [20] proposed a K-Nearest Neighbour (KNN) to recognize the license plate. The research was conducted on 125 license plate images which were divided into 100 training images and 25 testing images. The success rate of this research was 92.86%. The condition of the license plate and light intensity influenced the recognition result.

3. Proposed Methodology

Figure1 shows the proposed block diagram, which is used to identify the person, bike rider with having helmet, and extracts the number plate. The YOLO (You Only Look Once) model is a popular real-time object detection algorithm used for identifying objects in images or videos. For person identification using the YOLO model, the algorithm is trained on a dataset that contains labeled images of people.The YOLO model works by dividing the input image into a grid of cells and predicting a bounding box and class probabilities for each cell. To identify a person, the YOLO model uses a pre-trained convolutional neural network (CNN) to extract features from the image. The extracted features are then used to predict the bounding box and class probabilities for each cell.Once the YOLO model predicts the bounding box and class probabilities for each cell, a post-processing step is performed to merge overlapping bounding boxes and remove false positives. The result is a list of bounding boxes and class probabilities for each person in the input image.

Bike Rider Helmet Detection using YOLO Model: To detect whether a bike rider is wearing a helmet or not, the YOLO model can be trained on a dataset of labeled images of bike riders with and without helmets. The YOLO model works by first extracting features from the input image using a pre-trained CNN, then predicting the bounding box and class probabilities for each cell in the grid.To detect whether a bike rider is wearing a helmet or not, the YOLO model can be trained to recognize the shape and color of the helmet. The YOLO model can also be trained to recognize the shape of the rider's head without a helmet.Once the YOLO model predicts the bounding box and class probabilities for each cell, a post-processing step is performed to merge overlapping bounding boxes and remove false positives. The result is a list of bounding boxes and class probabilities for each bike rider in the input image, along with a label indicating whether they are wearing a helmet or not.

Bike Number Plate Extraction using YOLO Model: To extract the number plate of a bike using the YOLO model, the algorithm is trained on a dataset of labeled images of bikes with visible number plates. The YOLO model works by first extracting features from the input image using a pre-trained CNN, then predicting the bounding box and class probabilities for each cell in the grid.To extract the number plate of a bike, the YOLO model can be trained to recognize the shape and color of the number plate, as well as any patterns or characters that may be present. Once the YOLO model predicts the bounding box and class probabilities for each cell, a post-processing step is performed to merge overlapping bounding boxes and remove false positives.The result is a list of bounding boxes and class probabilities for each number plate in the input image, along with the characters or patterns recognized in the plate. The extracted number plate can then be used for further analysis or processing, such as identifying the owner of the bike or verifying the bike's registration.

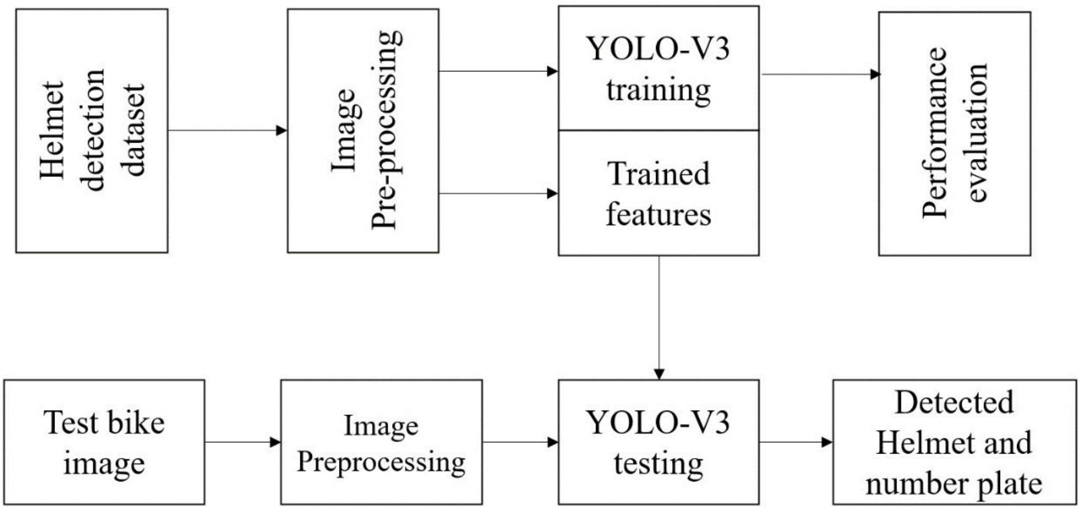

Figure 1. Proposed block diagram.

3.1 Image preprocessing

Digital image processing is the use of computer algorithms to perform image processing on digital images. As a subfield of digital signal processing, digital image processing has many advantages over analogue image processing. It allows a much wider range of algorithms to be applied to the input data — the aim of digital image processing is to improve the image data (features) by suppressing unwanted distortions and/or enhancement of some important image features so that our AI-Computer Vision models can benefit from this improved data to work on. To train a network and make predictions on new data, our images must match the input size of the network. If we need to adjust the size of images to match the network, then we can rescale or crop data to the required size.

we can effectively increase the amount of training data by applying randomized augmentation to data. Augmentation also enables to train networks to be invariant to distortions in image data. For example, we can add randomized rotations to input images so that a network is invariant to the presence of rotation in input images. An augmented Image Datastore provides a convenient way to apply a limited set of augmentations to 2-D images for classification problems.

we can store image data as a numeric array, an Image Datastore object, or a table. An Image Datastore enables to import data in batches from image collections that are too large to fit in memory. we can use an augmented image datastore or a resized 4-D array for training, prediction, and classification. We can use a resized 3-D array for prediction and classification only.

There are two ways to resize image data to match the input size of a network. Rescaling multiplies the height and width of the image by a scaling factor. If the scaling factor is not identical in the vertical and horizontal directions, then rescaling changes the spatial extents of the pixels and the aspect ratio.
Cropping extracts a subregion of the image and preserves the spatial extent of each pixel. We can crop images from the center or from random positions in the image. An image is nothing more than a two-dimensional array of numbers (or pixels) ranging between 0 and 255. It is defined by the mathematical function f(x,y) where x and y are the two co-ordinates horizontally and vertically.
Resize image: In this step-in order to visualize the change, we are going to create two functions to display the images the first being a one to display one image and the second for two images. After that, we then create a function called processing that just receives the images as a parameter.
Need of resize image during the pre-processing phase, some images captured by a camera and fed to our AI algorithm vary in size, therefore, we should establish a base size for all images fed into our AI algorithms.
Morphological Operations: Morphological Operations is a broad set of image processing operations that process digital images based on their shapes. In a morphological operation, each image pixel is corresponding to the value of other pixel in its neighborhood. By choosing the shape and size of the neighborhood pixel, you can construct a morphological operation that is sensitive to specific shapes in the input image. Morphological operations apply a structuring element called strel in Matlab, to an input image, creating an output image of the same size.
Types of Morphological operations:

- Dilation: Dilation adds pixels on the object boundaries.
- Erosion: Erosion removes pixels on object boundaries.
- Open: The opening operation erodes an image and then dilates the eroded image, using the same structuring element for both operations.
- Close: The closing operation dilates an image and then erodes the dilated image, using the same structuring element for both operations.

The number of pixels added or removed from the object in an image depends on the shape and size of the structuring element used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image. The rule used to process the pixels defines the morphological operation as a dilation or an erosion.

3.2 YOLO Model

Object detection is a phenomenon in computer vision that involves the detection of various objects in digital images or videos. Some of the objects detected include people, cars, chairs, stones, buildings, and animals. This phenomenon seeks to answer two basic questions:

1. What is the object? This question seeks to identify the object in a specific image.
2. Where is it? This question seeks to establish the exact location of the object within the image.

Object detection consists of various approaches such as fast R-CNN, Retina-Net, and Single-Shot MultiBox Detector (SSD). Although these approaches have solved the challenges of data limitation and modeling in object detection, they are not able to detect objects in a single algorithm run. YOLO algorithm has gained popularity because of its superior performance over the object detection techniques.
YOLO Definition: YOLO is an abbreviation for the term 'You Only Look Once'. This is an algorithm that detects and recognizes various objects in a picture (in real-time). Object detection in YOLO is done as a regression problem and provides the class probabilities of the detected images.
YOLO algorithm employs convolutional neural networks (CNN) to detect objects in real-time. As the name suggests, the algorithm requires only a single forward propagation through a neural network to detect objects. This means that prediction in the entire image is done in a single algorithm run. CNN is used to predict various class probabilities and bounding boxes simultaneously. The YOLO algorithm consists of various variants. Some of the common ones include tiny YOLO and YOLOv3.
Importance of YOLO: YOLO algorithm is important because of the following reasons:

- Speed: This algorithm improves the speed of detection because it can predict objects in real-time.
- High accuracy: YOLO is a predictive technique that provides accurate results with minimal background errors.
- Learning capabilities: The algorithm has excellent learning capabilities that enable it to learn the representations of objects and apply them in object detection.

YOLO algorithm working: YOLO algorithm works using the following three techniques:

- Residual blocks
- Bounding box regression
- Intersection Over Union (IOU)

Residual blocks: First, the image is divided into various grids. Each grid has a dimension of S x S. The following Figure 2 shows how an input image is divided into grids. In the Figure 2, there are many grid cells of equal dimension. Every grid cell will detect objects that appear within them. For example, if an object center appears within a certain grid cell, then this cell will be responsible for detecting it.

Figure 2. Example of residual blocks.

Bounding box regression: A bounding box is an outline that highlights an object in an image.Every bounding box in the image consists of the following attributes:

- Width (bw)
- Height (bh)
- Class (for example, person, car, traffic light, etc.)- This is represented by the letter c.
- Bounding box center (bx,by)

The following Figure 3 shows an example of a bounding box. The bounding box has been represented by a yellow outline. YOLO uses a single bounding box regression to predict the height, width, center, and class of objects. In the image above, represents the probability of an object appearing in the bounding box.

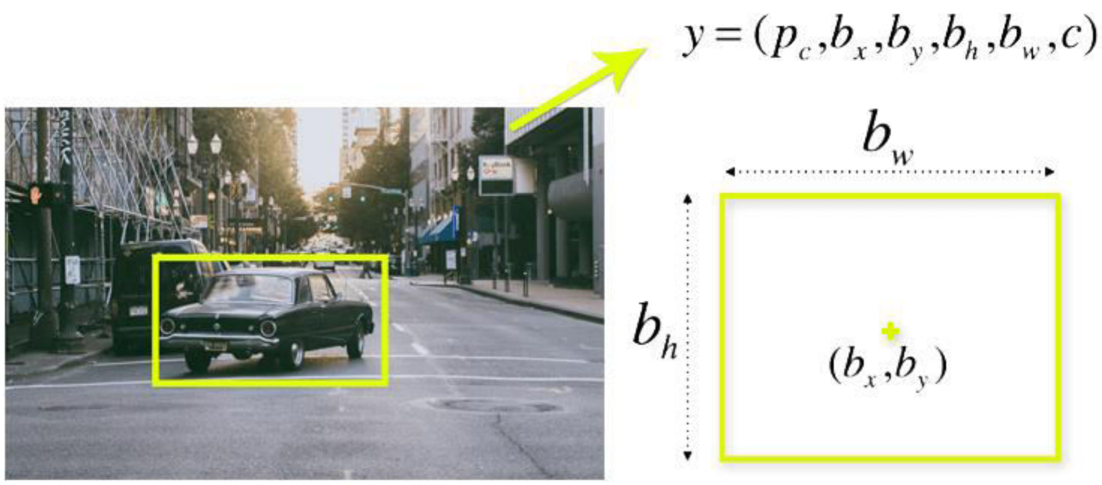

Figure 3. Bounding box regression

Intersection over union (IOU): Intersection over union (IOU) is a phenomenon in object detection that describes how boxes overlap. YOLO uses IOU to provide an output box that surrounds the objects perfectly. Each grid cell is responsible for predicting the bounding boxes and their confidence scores. The IOU is equal to

1 if the predicted bounding box is the same as the real box. This mechanism eliminates bounding boxes that are not equal to the real box.

Combination of the three techniques: The following image shows how the three techniques are applied to produce the final detection results.

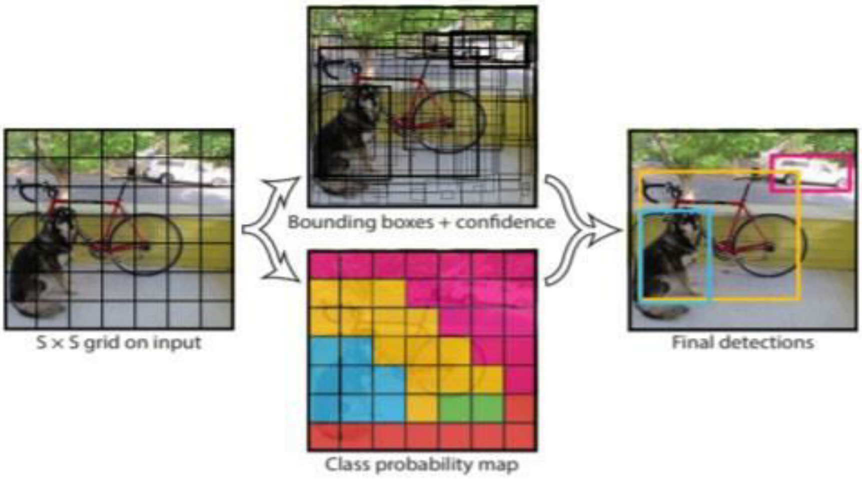

Figure 4. Combination of three modules.

First, the image is divided into grid cells. Each grid cell forecasts B bounding boxes and provides their confidence scores. The cells predict the class probabilities to establish the class of each object. For example, we can notice at least three classes of objects: a car, a dog, and a bicycle. All the predictions are made simultaneously using a single convolutional neural network. Intersection over union ensures that the predicted bounding boxes are equal to the real boxes of the objects. This phenomenon eliminates unnecessary bounding boxes that do not meet the characteristics of the objects (like height and width). The final detection will consist of unique bounding boxes that fit the objects perfectly. For example, the car is surrounded by the pink bounding box while the bicycle is surrounded by the yellow bounding box. The dog has been highlighted using the blue bounding box.

The YOLO algorithm takes an image as input and then uses a simple deep convolutional neural network to detect objects in the image. The architecture of the CNN model that forms the backbone of YOLO is shown below.

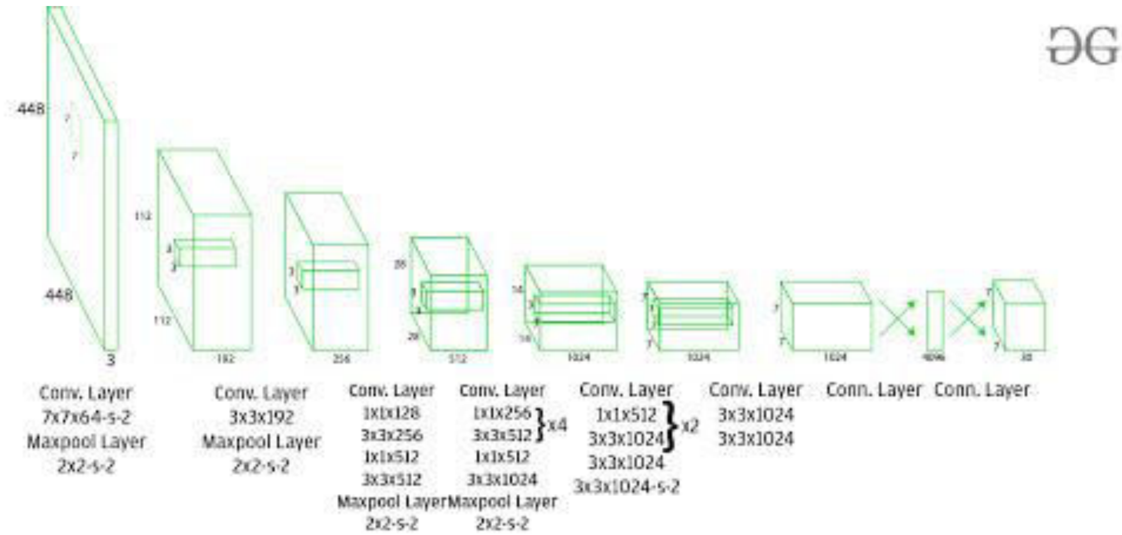

Figure 5. YOLO Layers.

The first 20 convolution layers of the model are pre-trained using ImageNet by plugging in a temporary average pooling and fully connected layer. Then, this pre-trained model is converted to perform detection since previous research showcased that adding convolution and connected layers to a pre-trained network improves performance. YOLO's final fully connected layer predicts both class probabilities and bounding box coordinates. YOLO divides an input image into an S × S grid. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object. Each grid cell predicts B bounding boxes and confidence scores for those

boxes. These confidence scores reflect how confident the model is that the box contains an object and how accurate it thinks the predicted box is.
YOLO predicts multiple bounding boxes per grid cell. At training time, we only want one bounding box predictor to be responsible for each object. YOLO assigns one predictor to be “responsible” for predicting an object based on which prediction has the highest current IOU with the ground truth. This leads to specialization between the bounding box predictors. Each predictor gets better at forecasting certain sizes, aspect ratios, or classes of objects, improving the overall recall score.
One key technique used in the YOLO models is non-maximum suppression (NMS). NMS is a post-processing step that is used to improve the accuracy and efficiency of object detection. In object detection, it is common for multiple bounding boxes to be generated for a single object in an image. These bounding boxes may overlap or be located at different positions, but they all represent the same object. NMS is used to identify and remove redundant or incorrect bounding boxes and to output a single bounding box for each object in the image.

4. Results and Discussion

In Figure 6, YOLO model detected image contains person and bike. Further, the application detected that person is not wearing helmet and its extracted number from vehicle and display in beside text area. The extracted number plate is KA-09 HB-0164. In Figure 7, screen application detected person is wearing helmet and that label is displaying around his head and application stop there itself and not scanning number plate.

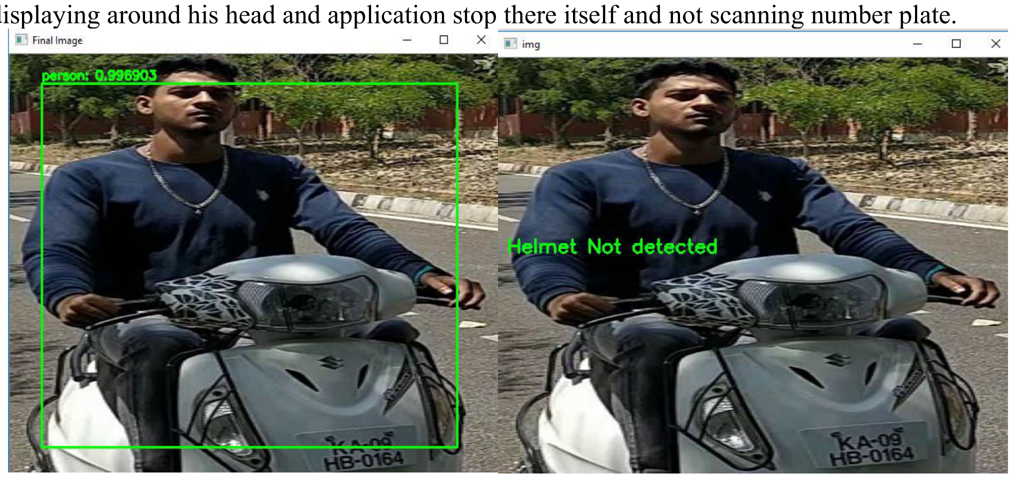

Figure 6. Person detected without helmet.

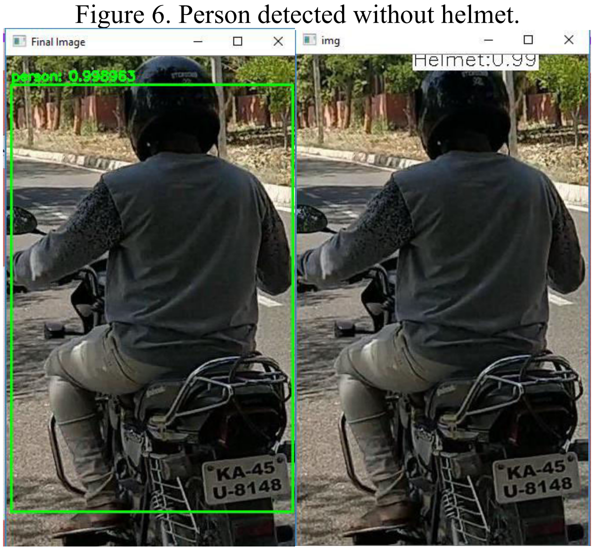

Figure 7. Person detected with helmet.

5. Conclusion

A Non-Helmet Rider Detection system is developed where a video file is taken as input. If the motorcycle rider in the video footage is not wearing helmet while riding the motorcycle, then the license plate number of that motorcycle is extracted and displayed. Object detection principle with YOLO architecture is used for motorcycles, person, helmet, and license plate detection. OCR is used for license plate number extraction if rider is not wearing helmet. Not only the characters are extracted, but also the frame from which it is also extracted so that it can be used for other purposes. All the objectives of the project were achieved satisfactorily.

References

[1]. Marzuki, P., et al. "A design of license plate recognition system using convolutional neural network." International Journal of Electrical and Computer Engineering 9.3 (2019): 2196.

[2]. Ma, Lixin, and Yong Zhang. "Research on vehicle license plate recognition technology based on deep convolutional neural networks." Microprocessors and Microsystems 82 (2021): 103932.

[3]. Vaiyapuri, Thavavel, et al. "Automatic vehicle license plate recognition using optimal deep learning model." Computers, Materials & Continua 67.2 (2021).

[4]. 4. Huang, Zhao-Kai, Hao-Wei Tseng, and Cheng-Lun Chen. "Application of Extreme Learning Machine to Automatic License Plate Recognition." 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA). IEEE, 2019.

[5]. Neto, Edson Cavalcanti, et al. "Brazilian vehicle identification using a new embedded plate recognition system." Measurement 70 (2019): 36-46.

[6]. Halin, Alfian Abdul, et al. "License plate localization using a Naïve Bayes classifier." 2020 IEEE International Conference on Signal and Image Processing Applications.IEEE,2020.

[7]. Akhtar, Zuhaib, and Rashid Ali. "Automatic number plate recognition using random forest classifier." SN Computer Science 1 (2020): 1-9.

[8]. Sathiyabhama, B., et al. "Tracing of vehicle region and number plate detection using deep learning." 2020 International conference on emerging trends in information technology and engineering (ic-ETITE). IEEE, 2020.

[9]. Tabrizi, Sahar S., and Nadire Cavus. "A hybrid KNN-SVM model for Iranian license plate recognition." Procedia Computer Science 102 (2019): 588-594.

[10]. Singh, Jaskirat, and Bharat Bhushan. "Real time Indian license plate detection using deep neural networks and optical character recognition using LSTM tesseract." 2019 international conference on computing, communication, and intelligent systems (ICCCIS). IEEE, 2019.

[11]. Chowdhury, Pinaki Nath, et al. "A new U-net based license plate enhancement model in night and day images." Pattern Recognition: 5th Asian Conference, ACPR 2019, Auckland, New Zealand, November 26–29, 2019, Revised Selected Papers, Part I 5. Springer International Publishing, 2020.

[12]. Rashid, Amr E. "A fast algorithm for license plate detection." 2019 International Conference on Signal Processing, Image Processing & Pattern Recognition. IEEE, 2019.

[13]. Kim, Jung-Hwan, et al. "License plate detection and recognition algorithm for vehicle black box." 2019 International Automatic Control Conference (CACS). IEEE, 2019.

[14]. Abolghasemi, Vahid, and Alireza Ahmadyfard. "An edge-based color-aided method for license plate detection." Image and Vision Computing 27.8 (2019): 1134-1142.

[15]. Amaanullah, Rizki Rafiif, et al. "Comparative Transfer Learning Techniques for Plate Number Recognition." 2022 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom). IEEE, 2022.

[16]. Khan, Ishtiaq Rasool, et al. "Automatic License Plate Recognition in Real-World Traffic Videos Captured in Unconstrained Environment by a Mobile Camera." Electronics 11.9 (2022): 1408.

[17]. Tham, Mau-Luen, and Wei Kun Tan. "IoT based license plate recognition system using deep learning and Open VINO." 2021 4th International Conference on Sensors, Signal and Image Processing. 2021.

[18]. Sharma, Jitendra, et al. "A hybrid technique for license plate recognition based on feature selection of wavelet transform and artificial neural network." 2019 International Conference on Reliability Optimization and Information Technology (ICROIT). IEEE, 2019.

[19]. Ho, Wing Teng, Hao Wooi Lim, and Yong Haur Tay. "Two-stage license plate detection using gentle Adaboost and SIFT-SVM." 2019 First Asian Conference on Intelligent Information and Database Systems. IEEE, 2019.

[20]. Gunawan, Dani, W. Rohimah, and R. F. Rahmat. "Automatic number plate recognition for Indonesian license plate by using K-nearest neighbour algorithm." Series: Materials Science and Engineering. Vol. 648. No. 1. IOP Publishing, 2019.