Automated License Plate Recognition for Non-Helmeted Motor Riders Using YOLO and OCR

Chunduru Anilkumar[1,*], Meesala Shobha Rani[2], Venkatesh B[3] and G. Srinivasa Rao[4]

[1]*Department of Information Technology, GMR Institute of Technology, Rajam, Andhra Pradesh, India*
[2]*School of Computer Science and Engineering, REVA University, Bangalore, Karnataka India*
[3]*Associate Professor, Department of Computer Science and Engineering, BVRIT HYDERABAD College of Engineering for Women, Hyderabad, India*
[4]*Department of Information Technology, RVR & JC College of Engineering, Guntur Andhra Pradesh, India*
E-mail: anilkumar.ch@gmrit.edu.in
[*]*Corresponding Author*

Received 05 July 2023; Accepted 03 October 2023;
Publication 29 March 2024

Abstract

One of the leading causes of serious injuries in motorcycle accidents is the failure to wear a helmet, pressing the need for effective measures to encourage riders to use helmets. To stop these frequent violations of business regulations, regular observation is necessary. The proposed system is for detecting traffic violations in India related to riding motorcycles without helmets. The system utilizes deep learning-based object detection using YOLO and OCR techniques to automatically detect non-helmet riders and extract license plate numbers. The proposed system aims to improve efficiency and accuracy in detecting violations by automating the process and reducing the need for manpower. The system involves three levels of object detection: person and motorcycle/moped, helmet, and license plate. OCR is used to

Journal of Mobile Multimedia, Vol. 20_2, 239–266.
doi: 10.13052/jmm1550-4646.2021

extract the license plate number, and all techniques are subject to predefined conditions and constraints. The system is designed to operate on video input to ensure high-speed execution, and it intends to offer a complete solution for both detection of helmet and extracting the license plate number.

Keywords: Object detection, helmet detection, license plate number, YOLO, optical character recognition.

1 Introduction

The impact of a collision is absorbed by the helmet's cushion, and the head eventually is brought to a halt as time passes. It disseminates the impact over a wider range, wearing a protective helmet while participating in activities that pose a risk of head injury is one of the best ways to protect the head. It serves primarily in the role of mechanical impediment the rider's head and whatever he contacted. Wearing a high-quality full-face helmet can help to reduce injuries. To instill discipline, traffic rules are in place, thereby reducing the risk of fatalities and injuries. As a result, efficient and feasible solutions to these problems must be developed. Manual traffic surveillance using CCTV is an existing method. Manual methods of helmet detection can be time-consuming and inefficient, especially in cities with high traffic volumes, as the population and number of vehicles grow. Furthermore, human error can lead to inaccuracies in helmet law enforcement, potentially causing harm to riders who are not wearing helmets. To address these issues, technology-based solutions such as Algorithms for computer vision and machine learning have been developed to automate the detection of helmet process. These solutions can identify non-helmeted riders quickly and accurately, alerting the appropriate authorities. By combining YOLO and OCR, you can build a system capable of detecting helmet wear and extracting vehicle license plate information, providing a comprehensive solution for helmet law enforcement and road safety. Implementing such a system can aid in improving the efficiency and accuracy of helmet law enforcement while also promoting road safety.

2 Related Work

2.1 Helmet Detection

The use of video surveillance of detecting the automated motorcycle. To detect the moving objects, the authors in proposed a multi-task learning

(MTL) method which uses a convolution neural network (CNN) to identify and track individual motorcycles. The HELMET dataset is used, which contains 91,000 frames from nearly ten thousand different motorcycles were collected from twelve observation sites in Myanmar. The proposed MTL (Multi-task Learning) approach has been successful in improving the efficiency of the detection process for tracked motorcycles, and the processing rate exceeded 8 frames per second on standard consumer hardware. Furthermore, by using this method, the detection of the number of riders and their helmet usage can achieve a weighted average F-measure of 67.3% [1]. An improved version of YOLO algorithm which is YOLOv5-HD. It is to detect helmet use on motorcyclists, and the results show that the method was effective in determining whether riders were wearing helmets. The training data is taken for this stage which consists of motorcycle images that taken from the original image of traffic monitoring, with the helmet or head positioned in the upper half of the image. A smaller model with an input size of 320×320 is used to meet real-time processing requirements [2]. This paper addresses the difficulties of object detection in crowded scenes with multiple heads present. To address these issues, the paper employed data enhancement technology, which entails generating additional training data by augmenting existing data. This technique can help to increase the variety and complexity of the training data, as well as make the model more resistant to variations in real-world scenarios. Post-processing refers to a set of techniques used to refine and improve the results after the initial detection process [3].

Figure 1 describes about the moving vehicle detection. It is the process of identifying vehicles in motion within a given scene, typically using computer vision techniques. Background subtraction is a common method used in this process, in which a model of the stationary background is subtracted from the current frame to identify any pixels that are significantly different, and thus potentially part of a moving vehicle. After that the image is being processed.

A system to detect the motorcycle automatically. In this system, support vector machines (SVMs) are utilized to analyze image data of the head region

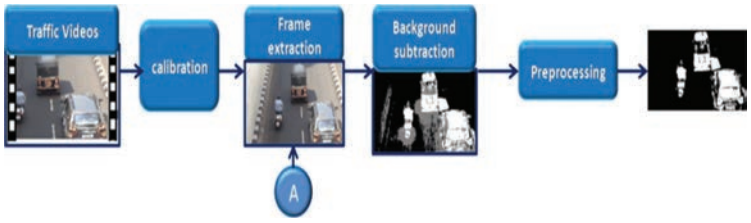

Figure 1 Moving vehicle detection module.

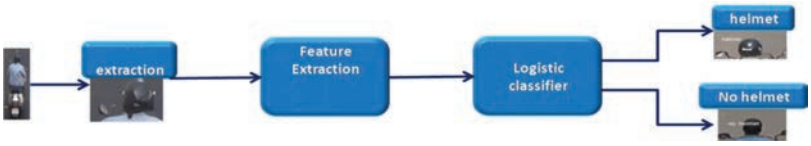

Figure 2 Helmet detection module.

of motorcycle riders. The system trains the SVMs on histograms extracted from both static photographs and individual image frames taken from video data [4]. The system has generated tracks of motorcycle riders' heads and classified each head region based on whether a helmet is present; After obtaining individual classifier results, the tracks are categorized as a complete entity based on the average of those results. By using the trained classifier, it is used to classified and isolated to identify the heads of the riders. The system has been tested on static photographs as well as tracking data, and Based on the results, it can be concluded that the system can effectively distinguish between helmet-wearing and non-helmet-wearing riders in both scenarios [5].

In Figure 2 explains about helmet detection, once the person riding the motorcycle is identified, feature extraction can be used to determine whether they are wearing a helmet or not. Feature extraction involves analyzing the characteristics of an image or video frame to extract relevant information that can be used to classify the image or object of interest. For helmet detection, features such as color, texture, and shape can be extracted from the region of interest and then used to classify whether a helmet is present or not. Machine learning algorithms can be trained on a dataset of helmet and non-helmet images, and then used to automatically extract these features and make accurate predictions about whether a person is wearing a helmet or not.

2.2 License Plate Detection

A convolutional neural network was employed to detect the license plate of a motorcycle rider without a helmet in a video stream. To avoid false positives caused by helmeted riders exiting the video frames, the centroid tracking method with a horizontal reference line was applied. The overall detection rate of LPs was 98.52%. The YOLOv2 algorithm is used in this article to detect license plates (LPs) of non-helmeted motorcyclists in real time [6]. The responsibility of predicting the object's coordinates present in a particular cell of the input image lies with that cell itself. The YOLOv2 model underwent training using a dataset consisting of three types of objects,

namely Person, Helmet, and Plate. To label the dataset, a labelling software was employed which enabled human annotators to identify and categorize objects in images by manually marking their locations [7]. The OKM-CNN model is a robust deep learning technique for recognizing vehicle license plates (VLPR). The approach combines optimal K-means clustering (OKM) based segmentation with convolutional neural network (CNN) based recognition [8]. The model comprises three primary stages, namely license plate detection, segmentation using OKM clustering, and license plate number recognition using CNN. For detecting and locating license plates, the model employs the Improved Bernsen Algorithm (IBA) and Connected Component Analysis (CCA) models [9]. The license plate detection process involves using a trained model to detect the license plate in each motorcycle image. A possible rephrased version of the statement could be: "To develop the model, a dataset containing 832 images of bikes and mopeds, along with annotated license plate bounding boxes, was collected [10]. The annotated images were utilized to train the model, which can then detect the license plate in each input image and create a bounding box around it. The detected bounding box's information, such as the top-left and bottom-right coordinates, class name, and detection confidence, is stored in a .json file. In case of multiple bounding boxes, a confidence threshold of 0.5 is set, and the one with the highest confidence is selected. The license plate image is subsequently extracted using the bounding box coordinates and stored separately [11].

Figure 3 Describes about the motorcycle segmentation. Contour detection is the process of identifying and extracting an object's outline or boundary from an image or video. It is common in computer vision applications

Figure 3 Motorcycle segmentation module.

like object recognition and tracking. Based on their characteristics, metric classification could be used to categorize several types of motorcycles. It checks whether the motorcycle is tracked before or not. If not tracked before it will add the tracking point and it will classify the object.

2.3 License Plate Number Extraction

The process of obtaining license plate images from motorcycle photographs involves utilizing the bounding box coordinates stored in a .json file. In the case of multiple bounding boxes, a confidence threshold of 0.5 is set, and the bounding box with the highest confidence is selected [12]. The extracted image is then rotated by a fixed angle (in this case, 6 degrees) and rescaled to improve accuracy when optical character recognition (OCR) is applied. Finally, the OCR is applied to the pre-processed image to obtain an output. The image that was rotated was resized to a suitable scale to enable OCR to accurately recognize the characters in the image [13–15]. A method for extracting the license plate number from an image. First, the model detects whether the rider is wearing a helmet, and if not, the model finds the associated person class, motorbike, and license plate. Once the license plate is located, it is cropped and fed into an optical character recognition (OCR) model to recognize the text [16]. The OCR module generates a list of license plate numbers along with their corresponding confidence values, and the license plate with the highest confidence value is saved in a text file for future use [17–19]. The proposed approach comprises three distinct modules for license plate recognition. The first module detects the license plate and highlights it with a bounding box in the original image. The license plate extraction module crops the license plate image using the bounding box coordinates and feeds it to the Optical Character Recognition (OCR) module for further processing [20–23]. The OCR module first converts the image to grayscale and identifies the possible areas where the characters lie. The characters extracted from the license plate image are recognized using an Optical Character Recognition (OCR) system and compared to those identified by the K-Nearest Neighbors algorithm for matching. Finally, recognized characters are displayed on screen and added to a text file for further use [24–26].

2.4 Yolo Algorithm

The YOLO (You Only Look Once) algorithm was first introduced in 2016 by Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, all

researchers from the University of Washington and Facebook AI Research. In its initial release, YOLO utilized a solitary convolutional neural network (CNN) for direct prediction of class probabilities and bounding boxes from full-sized input images. This was a significant departure from previous object detection algorithms, which typically used multiple stages of processing and more complex architectures to achieve high accuracy [27, 28]. The authors of YOLO claimed that this single-stage approach made their algorithm faster and it outperformed other contemporary state-of-the-art object detection algorithms at the time, such as R-CNN and Fast R-CNN. However, the first version of YOLO suffered from some limitations, such as low recall on small objects and difficulty handling overlapping objects. In 2017, the authors released a significantly improved version of the algorithm, called YOLOv2 [29]. This version incorporated several key improvements, including anchor boxes for better handling of object scales and aspect ratios, batch normalization for improved accuracy, and a multi-scale training strategy to improve recall on small objects. YOLOv2 also significantly outperformed the original version on standard object detection benchmarks. In 2018, the authors released YOLOv3, which further improved the accuracy and speed of the algorithm. YOLOv3 introduced a larger network architecture with skip connections, which improved the model's ability to detect small objects and increased its accuracy on the COCO benchmark by about 10% compared to YOLOv2. YOLOv3 also added a feature called "swish" activation, which improved the model's speed and accuracy by using a smooth nonlinearity instead of the traditional rectified linear unit (ReLU) activation function. Since its inception, the YOLO algorithm has gained significant traction in the field of computer vision and has emerged as one of the most popular object detection techniques, being employed in diverse applications such as self-driving vehicles, robotics, and surveillance.

Working of Yolo Algorithm

The YOLO (You Only Look Once) algorithm is a real-time object detection system that utilizes a solitary convolutional neural network (CNN) for direct prediction of bounding boxes and class probabilities from full-sized input images. The algorithm works in several steps, which are as follows:

1. **Input Image:** The algorithm takes a full-sized input image and resizes it to a fixed size, typically 416×416 pixels.
2. **Feature Extraction:** The input image, resized to a specific dimension, is fed into a convolutional neural network, which extracts a set of feature

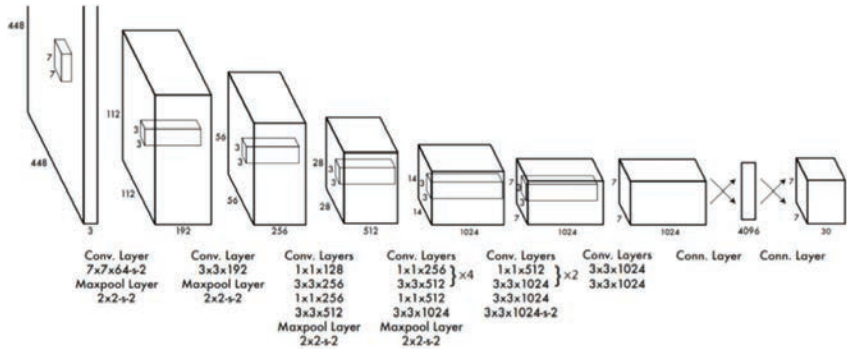

Figure 4 Yolo architecture.

maps that capture information about the objects in the image at different scales.

3. **Bounding Box Prediction:** For each cell in the feature map, the algorithm predicts a fixed number of bounding boxes, each of which consists of 4 coordinates (x, y, w, h) that define the center point, width, and height of the box. The coordinates are predicted relative to the coordinates of the cell, so they are always between 0 and 1.
4. **Objectness Score:** For each bounding box, the algorithm predicts an objectness score, which indicates the probability that the box contains an object.
5. **Class Prediction:** For each bounding box, the algorithm predicts a class probability distribution, which indicates the probability that the object inside the box belongs to each of the possible object classes.
6. **Non-Maximum Suppression:** To remove duplicate bounding boxes, the algorithm utilizes a method known as non-maximum suppression (NMS). NMS works by selecting the bounding box with the highest objectness score for each object and discarding any overlapping boxes with a lower score.
7. **Output:** The algorithm outputs a set of bounding boxes, each of which is associated with a class probability and an objectness score. These bounding boxes represents objects detected in the input image.

Overall, the YOLO algorithm is designed to be fast and efficient, allowing it to process images in real-time on a wide range of devices, including mobile phones and embedded systems. It is also able to detect a wide range of object categories with high accuracy, making it a popular choice for applications such as surveillance, autonomous vehicles, and robotics.

Multiple Generations of YOLO Object Detection Models

Yolo Object Detection Models are classified in to two types.

- Single Shot Object Detection
- Two Shot Object Detection

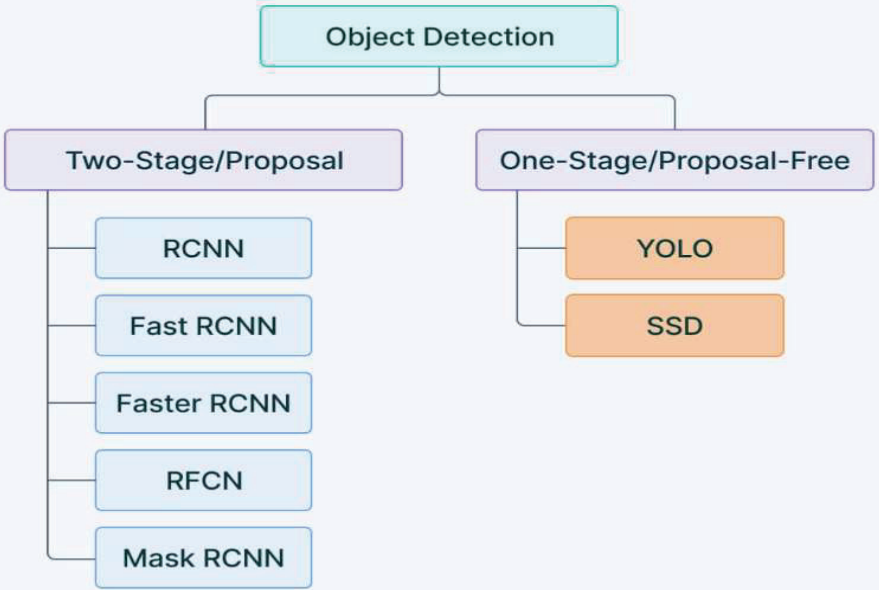

Figure 5 Yolo object detection models.

Single Shot Object Detection

When making predictions about the presence and placement of objects in an image, single-shot object detection only makes one pass over the input picture. The entire image is processed in a single pass, which increases their computing efficiency.

Two Shot Object Detection

Two passes of the input image are used in two-shot object detection to make assumptions about the existence and placement of objects. A collection of suggestions or prospective item positions are generated during the first pass, and these ideas are then refined during the second run to produce final predictions.

3 Data Collection

Motor Cycle Helmet and License Plate Detection Image Dataset contains 1476 images of three distinct classes i.e., Helmet, Plate, Rider for the objective of helmet, Plate and Rider Detection.

Bounding box annotations are provided in the YOLO v7 PyTorch format.

These total images are annotated into 3580 files. The bounding boxes are annotated in a .txt file format, where each line corresponds to a description of a single bounding box. A typical annotation file format looks like:

```
3 0.51015625 0.38984375 0.2375 0.73671875
1 0.540625 0.48203125 0.08125 0.071875
3 0.24609375 0.12578125 0.07734375 0.21484375
1 0.2453125 0.1640625 0.025 0.02265625
3 0.17578125 0.09140625 0.0640625 0.178125
1 0.18046875 0.11484375 0.025 0.01796875
3 0.4109375 0.178125 0.1 0.30625
```

Figure 6 Annotation.txt.

Figure 6 explains:

- Each row describes an object
- The .txt file format consists of columns for class, x_center, y_center, width, and height
- The coordinates of the bounding boxes are normalized to the dimensions of the image, meaning that the values are adjusted to a range between 0 and 1.
- The class numbers are indexed from one.

The Dataset partitioned into train, test, and validation sets. These contains 88%, 4%, 8% of the data respectively.

Table 1 Formation of dataset

Images	1476		
Annotations	3581(2.4 Per Image on Average)		
	Classes		
Annotation Classification	Helmet	Plate	Rider
	1079	1231	1271
Partition of Dataset	Training Set	Testing Set	Validation Set
	3100 (88%)	146 (4%)	296 (8%)

4 Database Connection

Fire Base

Firebase is a cloud-based platform that offers several services, such as authentication, real-time database, cloud storage, hosting, and machine learning features. While Firebase has many applications, it can also be utilized for image recognition tasks, like helmet detection and number plate extraction.

To detect helmets, you can make use of Firebase's machine learning capabilities, specifically its ML Kit, which provides pre-built machine learning models for image recognition, including object detection. You can either train your own model or use one of ML Kit's pre-built models. After selecting or training the model, you can utilize Firebase's Cloud Functions to perform the detection on the server-side, where the model can be deployed.

For number plate extraction, you can use Firebase's Cloud Vision API, which offers optical character recognition (OCR) for extracting text from images. By using the API, you can identify the number plate in an image and extract its text, which you can then store in Firebase's real-time database or cloud storage for further processing.

Overall, Firebase is an excellent choice for object detection and image recognition tasks due to its powerful and scalable platform. By using Firebase's machine learning capabilities and APIs, you can easily incorporate object detection and image recognition functions into your application.

5 Proposed Methodology

Our proposed approach for License Plate Recognition for Non-Helmeted Motor Riders consists of 3 steps. In the First step, we use the YOLO for the detection of the objects which are helmets, riders, and number plates in our case. Then in the second step, we took out the license plate object for those riders who are not wearing the helmet. And finally, the taken-out object is fed to the OCR Model for extracting the characters in the license plate object.

5.1 Detection of Objects Using YOLOv7

YOLOv7 is incredibly fast while having similar accuracy and is thus suitable for real-time applications. The steps taken in training the model are:

1. Cloned the YOLOv7 repository from https://github.com/WongKinYiu/yolov7.git

2. Downloaded the weights from the repository https://github.com/WongKinYiu/yolov7/releases/download/v0.1/yolov7.pt
3. Adapted the file of configuration settings. Here we called the yolov7.yaml configuration file
4. Adapted the data file. The data.yaml file in our case is –
    a. nc:3
    b. names: [‘helmet’, ‘rider’, ‘plate’]
5. Prepared the dataset into yolo format and split the data into train test and val folders
6. Model Training: The arguments passed here are pretrained downloaded weights file,config file from step 3, data file from step 4 and the number of epochs i.e., 55 in our case
7. Model Testing: The arguments passed here are trained weights file obtained from step 6, confidence, and the source video for testing.

Figure 7 Predictions after model training.

Architecture of YOLOv7

Figure 8 describes about a back-bone network, a neck-network, and a head-network comprise the YOLOv7 architecture. The back-bone network oversees retrieving characteristics from the input image, whilst the neck network fuses the features to minimize noise and enhance accuracy. The algorithm predicts the bounding boxes and class probabilities of the objects present in the image by the head network using these attributes.

Figure 9 describes about the CSPDarknet53 architecture, which is a modified version of the Darknet-53 architecture used in YOLOv5, is utilized as the backbone network in YOLOv7. CSPDarknet53 is a lightweight, high-performance backbone network that leverages cross-stage partial connections to increase information flow between network stages.

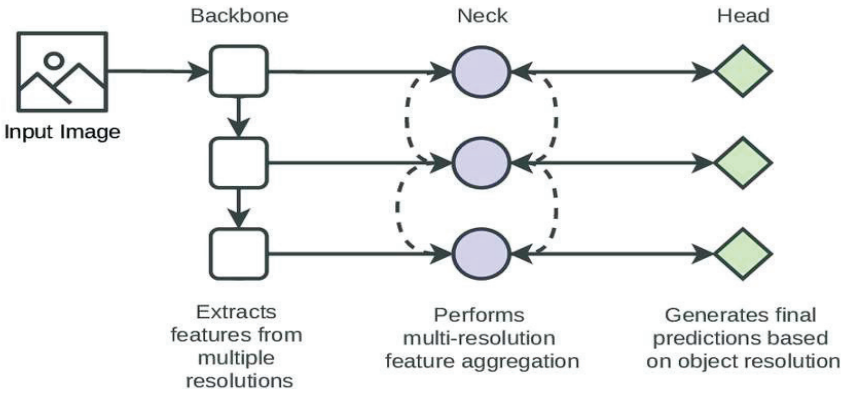

Figure 8 YOLOv7 architecture.

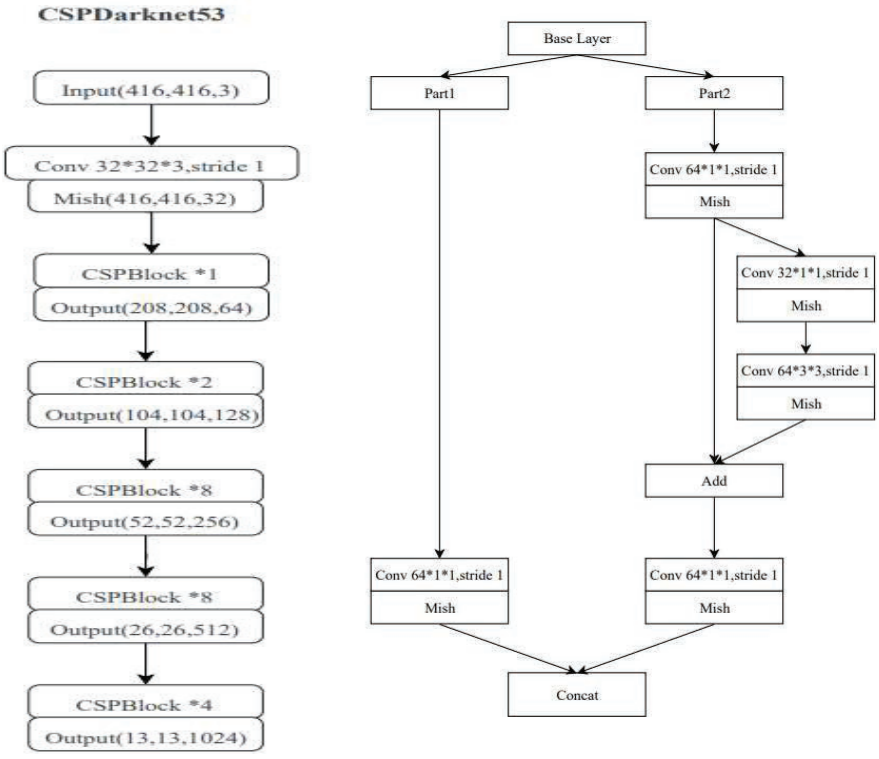

Figure 9 Back bone network in YOLOv7.

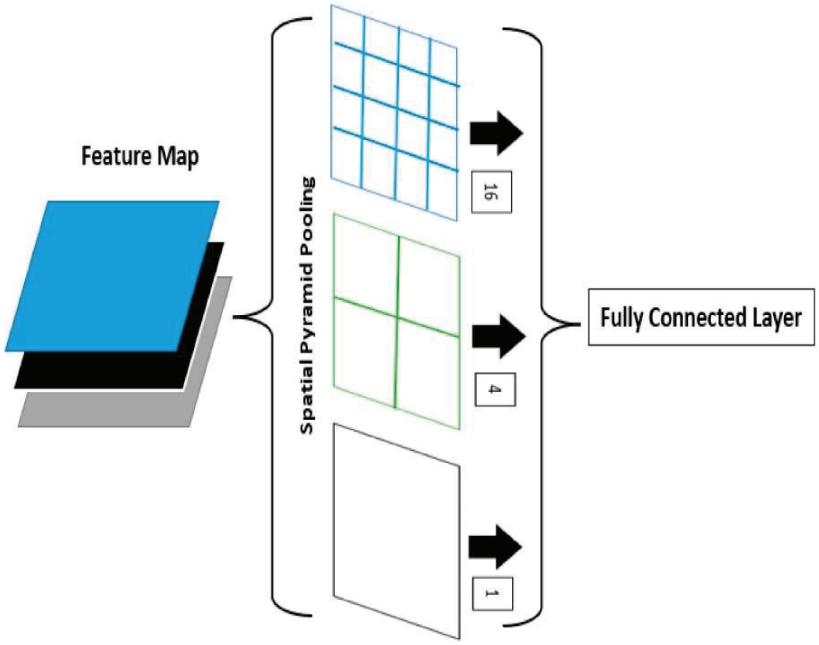

Figure 10 Spatial pyramid pooling (SPP).

Figure 10 describe about the neck network in YOLOv7 fuses features from different layers of backbone network using Spatial Pyramid Pooling (SPP) and Path Aggregation Network (PAN) modules. SPP improves the model's ability to detect objects of many sizes, whereas PAN enhances the model's capacity to understand and utilize spatial relationships between different elements in the input data.

Figure 11 describe about Anchor boxes are used by the head network in YOLOv7 to forecast the bounding boxes of objects in an image. It also employs a modified version of the YOLOv5 head network, which contains an attention mechanism that aids in model accuracy. YOLOv7 is not restricted to a single head. The lead head oversees the ultimate output, while the auxiliary head oversees aiding training in the middle layers.

5.2 Localizing License Plate for Non-Helmeted Riders

If the helmet is discovered, this step is unnecessary. This step is carried out by figuring out whether the helmet class's is present in the frame of the video. If the helmet class is not present in the frame, then we extract the license plate

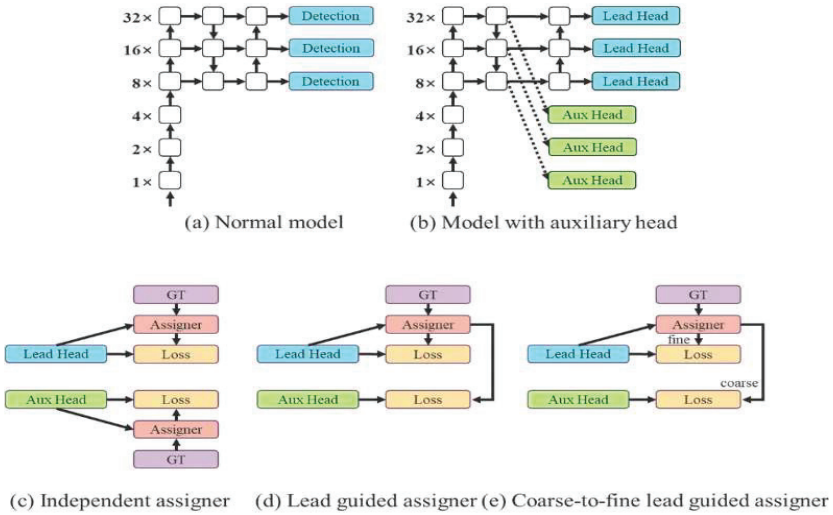

Figure 11 Head network in YOLOv7.

coordinates in the same frame if detected. Using the coordinates, we localize the bounding box of license plate and store it in a separate folder.

5.3 Extracting the Text from License Plate Using OCR

We iterate through each image in the stored folder which we did in step-2 and gave it to tesseract to obtain text from it. Pre-processing is needed before applying OCR straight to the retrieved license plate image to get higher accuracy. Hence, we applied dilation and erosion to each image to remove the noise and used bilateral filter to deal with blur images.

Bilateral filter is defined as

$$I^{filtered}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|)$$

And the term, W_P is defined as

$$W_p = \sum_{x_i \in \Omega} f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|)$$

Where,

I^{filtered} is the image after filtering,

I, is the unique image to be screened as input,

x are the coordinates of the currently being filtered pixel,
Ω stands for the window that is centered in x.
f_r is the range kernel for minimizing intensity differences,
g_s is the spatial (or domain) kernel used to smooth out coordinate discrepancies,

The connected component analysis of the number plate extracts the outlines of the characters, that is, the character to be identified is supplied in white text and the rest of the image is black. Once the character outlines have been retrieved, the image is transformed into blobs. Blobs are small areas of a scanned image that differ in properties such as brightness or color from the surrounding region.

5.4 OCR Working

OCR or Optical Character Recognition is an algorithm that enables computers to recognize printed or handwritten text characters from digital images or scanned documents. OCR has many practical applications, including digitizing paper documents, processing forms, and invoices, and improving accessibility for the visually impaired.

Figure 12 describe about the OCR process begins with pre-processing, which involves enhancing the image quality by removing noise, adjusting brightness and contrast, and correcting skew. Next, the OCR algorithm

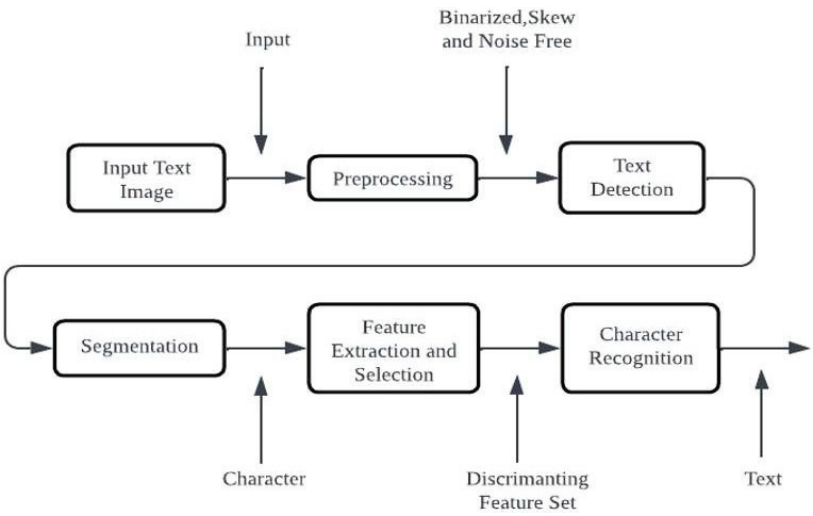

Figure 12 OCR architecture.

locates the regions of the image that contain text using edge detection or other techniques that differentiate text from the background. The OCR algorithm then analyzes each character within the text regions and identifies the corresponding character or symbol. Once the text is recognized, it is post-processed to correct errors and improve accuracy through language model-based correction, spell-checking, and other techniques. OCR algorithms can be classified into diverse types based on their inputs, outputs, and level of complexity in recognizing text. These include Handwritten OCR, Intelligent OCR, Zone OCR, and Hybrid OCR. OCR technology has advanced significantly and is now highly accurate, reliable, and efficient. Modern OCR algorithms can recognize a wide range of fonts and languages and can be easily integrated into various applications and workflows.

6 Performance Evaluation

The experimental setup involved using a Windows 10 PC along with GPU runtime provided by Google Collaboratory, and a 64-bit operating system. Google Drive was used for storing data, with 15GB of storage space available. The training and validation procedure of the experimental model were performed using the pyTorch framework with the CUDA backend. To aid in object detection and processing of image, the OpenCV library was used, while NumPy was used for multi-dimensional arrays, mathematical functions, and other utility tools.

Figure 13 explains about the annotated images are fed into the model as input. If the person is with helmet, it reads the next frame. If a non-helmet rider is found the model detects the license plate. After detecting the license plate, the next step would be to extract the plate from the image by using segmentation techniques. This may involve tasks such as isolating the plate from the rest of the image and enhancing the image to improve the readability of the characters on the license plate. Once the plate has been extracted, it can be further processed to perform tasks such as recognition of characters to read the alphanumeric characters on the plate by using Optical Character Recognition. The output of the model can provide information about whether a person is riding a motorbike in the input data.

The metric "mAP@0.5" represents the average precision of predicted bounding boxes that have an IoU overlap of at least 0.5 with the ground truth bounding boxes and is commonly used to measure the accuracy of object detection models. Based on analysis of the results, the overall mAP@0.5 score indicates that the model correctly identifies objects in approximately

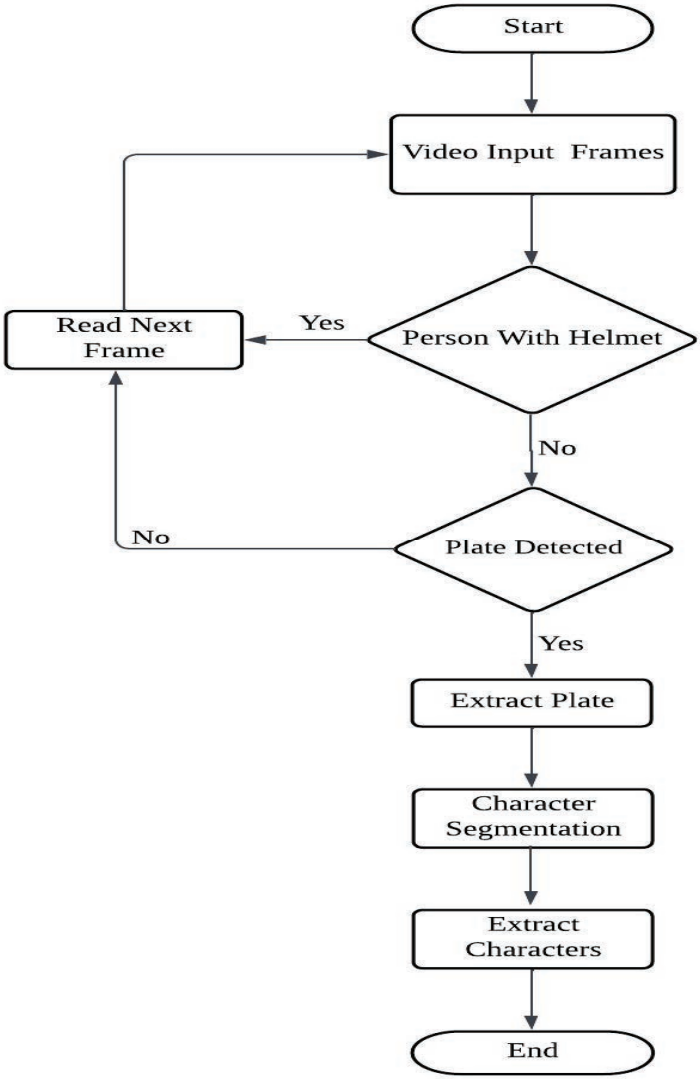

Figure 13 Implementation architecture overview.

58% of cases. Additionally, the precision value for the "Plate" class is the highest at 0.654, indicating that the model is correct when identifying plates about 65.4% of the time.

On the other hand, "mAP@0.5:0.95" represents the average precision across IoU thresholds between 0.5 to 0.95, with all detections considered.

Table 2 Performance evaluation

Class	Images	Labels	Precision	Recall	mAP@0.5	mAP@0.5:0.95:100%
All	296	677	0.604	0.64	0.58	0.289
Helmet	296	188	0.62	0.626	0.6	0.286
Plate	296	247	0.654	0.761	0.683	0.314
Rider	296	236	0.539	0.532	0.457	0.267

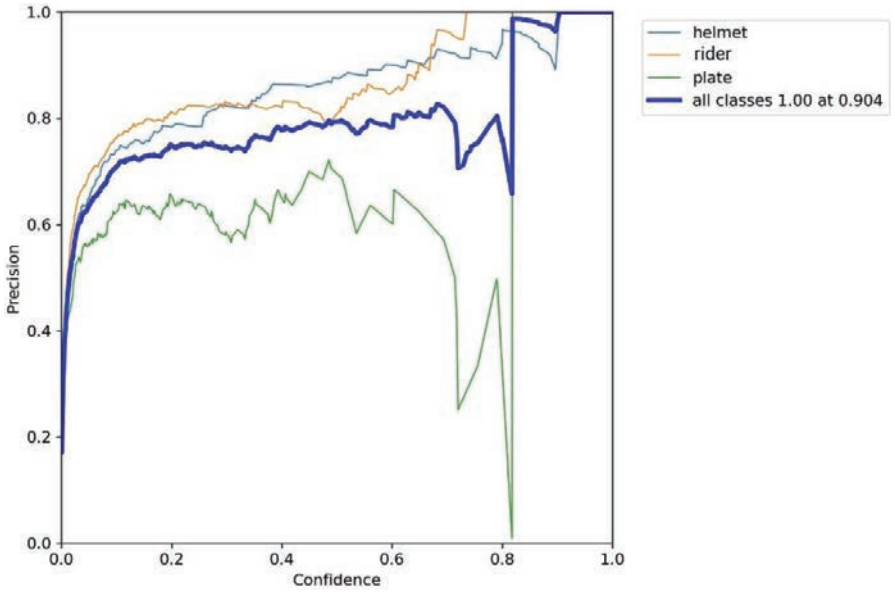

Figure 14(a) Precision vs confidence.

The mAP@0.5:0.95 for all classes combined is 0.289, indicating that the model's accuracy decreases when detecting objects with high confidence. Overall, the model's performance in identifying "Plate" objects is better than its performance in identifying "helmet" objects, and its overall performance is considered decent.

The label 14(a) shows the precision (P) versus confidence (C) graph, The label 14(b) the recall (R) versus confidence (C).

The label 14(c) is the mean average precision based on comparing the truth bounding box and detection box, The tradeoff between precision and recalls for various thresholds is depicted by the precision-recall curve. It is frequently employed when there is a significant imbalance in the courses. and 14(d) the IDF1 score at 62% with confidence of 0.032, advocates the balancing between the P and R based on Motorcycle Helmet and License

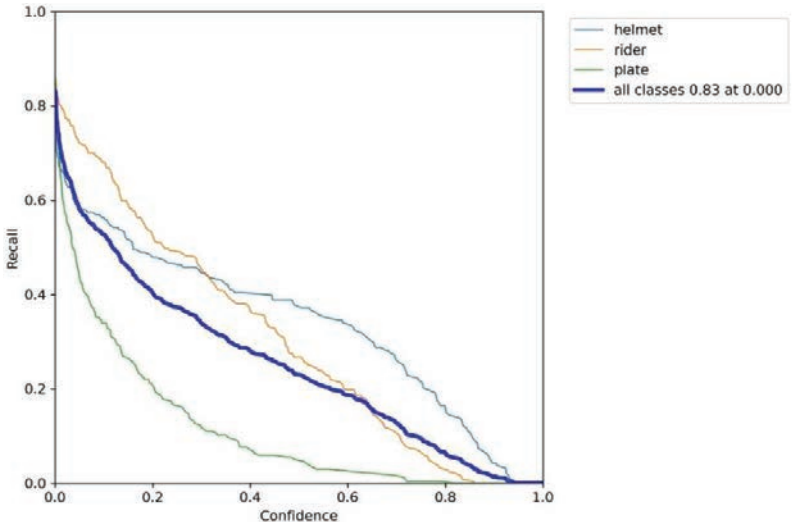

Figure 14(b) Recall vs confidence.

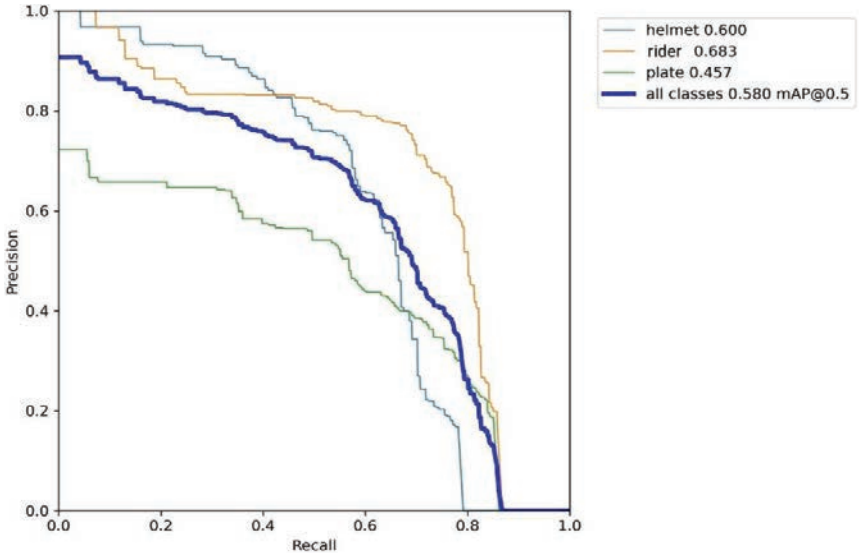

Figure 14(c) Precision vs recall.

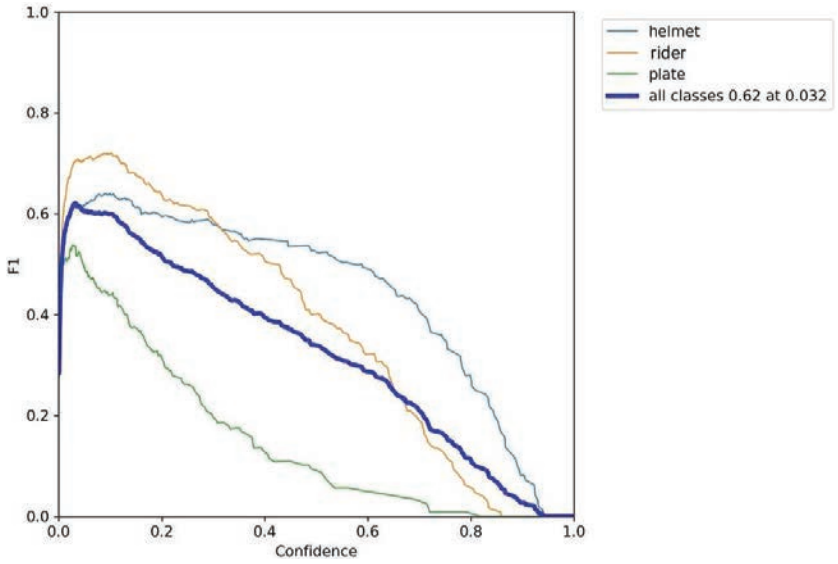

Figure 14(d) F1 vs confidence.

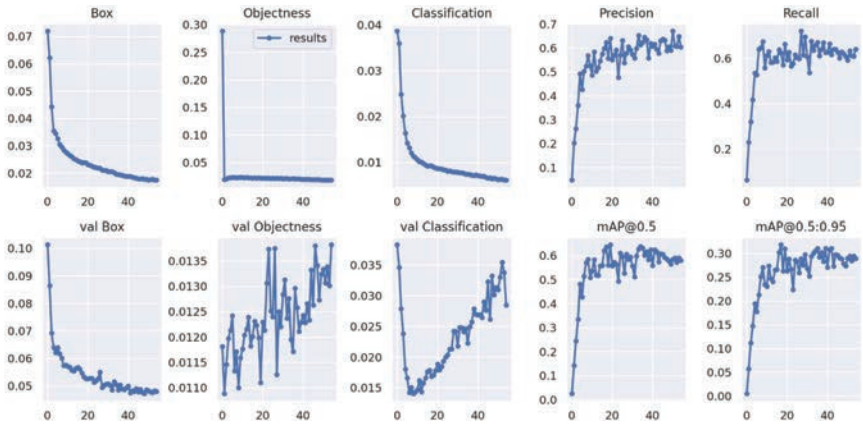

Figure 15 Training and validation loses of Yolov7.

Plate Detection Image Dataset. The mAP for all classes is medium and accurate model detections at 58% with a threshold of 0.5. The P and R are high at 100%, and 83.0%, respectively, and more confidence at 0.904 and 0.0, respectively, for all classes.

Figure 15 shows both training and validations losses of the Yolov7 algorithm's object detector and classification on 55 epochs for Motorcycle Helmet

and License Plate Detection Image Dataset. The precision and recall metrics in the training and validation phase converge at the highest of 60.4% accuracy, whereas the mAP converges at 95% with a 0.5 threshold.

7 Conclusion

The results displayed earlier demonstrate that YOLOv7 object detection is suitable for real-time processing, and it accurately classified and localized all object classes. Our proposed end-to-end model was successfully developed and equipped with capabilities for automation and deployment for monitoring. We utilized OCR technique to extract number plates and handled multiple riders without helmets. All libraries and software used in the project are open source, providing flexibility and cost-effectiveness. Our main goal was to address the issue of ineffective traffic management, and we are confident that if adopted by traffic management departments, our solution would streamline their operations and enhance efficiency. Future scope includes integration with ITS, machine learning algorithms for forecasting, facial recognition technology, enhancement of number plate extraction techniques, and cloud-based deployment. Continued research and development have the potential to revolutionize the field of traffic management.

References

[1] Hahne Lin, Jeremiah Deng, Deike Albers, Felix Wilhelm Siebert, "Helmet Use Detection of Tracked Motorcycles using CNN-based Multi-task Learning", IEEE Access, Vol 04, April 2019.

[2] Jia, Wei, et al., "Real-time automatic helmet detection of motorcyclists in urban traffic using improved YOLOv5 detector", IET Image Processing, Vol 15, Issue: 14, Dec 2021.

[3] J. Chiverton, "Helmet presence classification with motorcycle detection and tracking", IET Intelligent Transport Systems, Vol 06, Issue: 3, March 2020.

[4] Jamtsho, Yonten, Panomkhawn Riyamongkol, and Rattapoom Waranusast, "Real-time license plate detection for non-helmeted motorcyclist using YOLO", Ict Express 7.1, August 2021.

[5] Irina Valeryevna Pustokhina, Denis Alexandrovich Pustokhin, Joel J. P. C. Rodrigues, Deepak Gupta, "Automatic Vehicle License Plate Recognition using Optimal K-Means with Convolutional Neural Network for Intelligent Transportation Systems", IEEE Access, Oct 2017.

[6] Prajwal, M. J., et al. "Detection of non-helmet riders and extraction of license plate number using Yolo v2" International Journal of Innovative Technology and Exploring Engineering (IJITEE) (2019).

[7] Allamki, Lokesh, et al. "Helmet detection using machine learning and automatic License Plate Recognition." International Research Journal of Engineering and Technology (IRJET) 6.12 (2019).

[8] M. Darji, J. Dave, N. Asif, C. Godawat, V. Chudasama and K. Upla, "Licence Plate Identification and Recognition for Non-Helmeted Motorcyclists using Light-weight Convolution Neural Network," 2020 International Conference for Emerging Technology (INCET).

[9] S. Du, M. Ibrahim, M. Shehata, W. Badawy, Automatic license plate recognition (ALPR): A state-of-the-art review, IEEE Trans. Circuits Syst. Video Technol. 23(2) (2013) 311–325, http://dx.doi.org/10.1109/TCSVT.2012.2203741.

[10] Shi, X., Zhao, W., and Shen, Y. (2005, May). Automatic license plate recognition system based on color image processing. In *International Conference on Computational Science and Its Applications* (pp. 1159–1168). Berlin, Heidelberg: Springer Berlin Heidelberg.

[11] Silva, R., Aires, K., Santos, T., Abdala, K., Veras, R., and Soares, A. (2013, October). Automatic detection of motorcyclists without helmet. In *2013 XXXIX Latin American computing conference (CLEI)* (pp. 1–7). IEEE.

[12] D. Huang, C. Shan, M. Ardabilian, Y. Wang, L. Chen, Local binary patterns and its application to facial image analysis: A survey, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 41(6) (2011) 765–781, http://dx.doi.org/10.1109/TSMCC.2011.2118750.

[13] Waranusast, R., Bundon, N., Timtong, V., Tangnoi, C., and Pattanathaburt, P. (2013, November). Machine vision techniques for motorcycle safety helmet detection. In *2013 28th International conference on image and vision computing New Zealand (IVCNZ 2013)* (pp. 35–40). IEEE.

[14] C.-C. Chiu, M.-Y. Ku, H.-T. Chen, Motorcycle detection and tracking system with occlusion segmentation, in:Eighth International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'07), 2007, p. 32, http://dx.doi.org/10.1109/WIAMIS.2007.60.

[15] Chiu, C. C., Wang, C. Y., Ku, M. Y., and Lu, Y. B. (2006, June). Real time recogniton and tracking system of multiple vehicles. In *2006 IEEE Intelligent Vehicles Symposium* (pp. 478–483). IEEE.

[16] Wen, C. Y., Chiu, S. H., Liaw, J. J., and Lu, C. P. (2003, October). The safety helmet detection for ATM's surveillance system via the modified Hough transform. In *IEEE 37th Annual 2003 International Carnahan Conference on Security Technology, 2003. Proceedings.* (pp. 364–369). IEEE.

[17] Mistry, J., Misraa, A. K., Agarwal, M., Vyas, A., Chudasama, V. M., and Upla, K. P. (2017, November). An automatic detection of helmeted and non-helmeted motorcyclist with license plate extraction using convolutional neural network. In *2017 seventh international conference on image processing theory, tools and applications (IPTA)* (pp. 1–6). IEEE.

[18] J. Redmon, A. Farhadi, Yolo9000: Better, faster, stronger, 2016, ArXiv Prepr. ArXiv161208242, Dec. 2016, [Online]. Available: http://arxiv.org/abs/1612.08242 (Accessed: Mar. 05, 2019).

[19] A. Hirota, N.H. Tiep, L. Van Khanh, N. Oka, Classifying helmeted and non-helmeted motorcyclists, in: Advances in Neural Networks – ISNN 2017, Cham, 2017, pp. 81–86, http://dx.doi.org/10.1007/978-3-319-59072-1_10.

[20] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, 2015, ArXiv150602640 Cs, http://arxiv.org/abs/1506.02640.

[21] Huang, Y. P., Lai, S. Y., and Chuang, W. P. (2004, March). A template-based model for license plate recognition. In *IEEE International Conference on Networking, Sensing and Control, 2004* (Vol. 2, pp. 737–742). IEEE.

[22] Lin, C. H., Lin, Y. S., and Liu, W. C. (2018, April). An efficient license plate recognition system using convolution neural networks. In *2018 IEEE International Conference on Applied System Invention (ICASI)* (pp. 224–227). IEEE.

[23] Ullah, I., and Lee, H. J. (2016, December). An approach of locating Korean vehicle license plate based on mathematical morphology and geometrical features. In *2016 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 836–840). IEEE.

[24] Omran, S. S., and Jarallah, J. A. (2017, March). Iraqi car license plate recognition using OCR. In *2017 annual conference on new trends in information & communications technology applications (NTICT)* (pp. 298–303). IEEE.

[25] Babu, K. M., and Raghunadh, M. V. (2016, May). Vehicle number plate detection and recognition using bounding box method. In *2016*

International Conference on Advanced Communication Control and Computing Technologies (ICACCCT) (pp. 106–110). IEEE.

[26] N. Rana and P. K. Dahiya, "Localization techniques in ANPR systems: A-state-of-art," Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 7, no. 5, pp. 682–686, May 2017.

[27] Liang, G., Shivakumara, P., Lu, T., and Tan, C. L. (2015, August). A new wavelet-Laplacian method for arbitrarily-oriented character segmentation in video text lines. In *2015 13th international conference on document analysis and recognition (ICDAR)* (pp. 926–930). IEEE.

[28] Khare, V., Shivakumara, P., Raveendran, P., Meng, L. K., and Woon, H. H. (2015). A new sharpness based approach for character segmentation in License plate images. In Proceedings of the 3rd IAPR Asian conference on pattern recognition (ACPR) (pp. 544–548). IEEE.

Biographies

Chunduru Anilkumar received his B.Tech (Computer Science and Engineering) from Jawaharlal Nehru Technological University Kakinada, Andhra Pradesh in 2012. He received his Master of Technology in Computer Science and Engineering from Jawaharlal Nehru Technological University Hyderabad, Telangana in 2014. From 2016 onwards he is pursuing a part time Doctor of Philosophy (Ph.D) in School of Computer Science and Engineering from Vellore Institute of Technology, Vellore. He is currently working as an Assistant Professor in the Department of Information Technology at GMR Institute of Technology, Rajam, Srikakulam, Andhra Pradesh. He has published more than 30 technical papers in various international journals, conferences, and book chapters in Computer Science. He is associated with many professional bodies like ISTE, IACSIT, IAENG, CSTA (ACM), SDIWC, UACEE, CRSI and IEEE. He is in the editorial board reviewer of several international journals like International Journal of Grid and High-Performance Computing (IJGHPC), International Journal of Digital Crime

and Forensics (IJDCF), Journal of Information Technology Research (JITR), International Journal of Cyber-Physical Systems (IJCPS) these journals are indexed in Scopus, SCI. His research includes Cloud computing, Network security, Fog Computing, Information Security, Cybernetics, Machine Learning.

Meesala Shobha Rani received her Ph.D. from Vellore Institute of Technology in 2021, Vellore, India. She completed her M.Tech from the Karunya University, Coimbatore, India and B.Tech from JNTU, Anantapur. She is working as Assistant Professor in the School of Computer Science and Engineering at REVA University, Bangalore. She has published more than 20 technical papers in various international journals, conferences, patents, and book chapters in Computer Science. Her research includes Sentiment Analysis and Opinion Mining, Opinion Spam Detection, Text Mining, Intrusion Detection, Data Mining, Artificial Intelligence, Machine Learning, Deep Learning, and Evolutionary Optimization Techniques.

Venkatesh B obtained his B. Tech from JNTU, Anantapur in 2009. He received his Master of Technology from JNTU, Anantapur in 2013. He received his Ph.D. in 2021 from Vellore Institute of Technology, Tamil Nadu. Currently, he is working as an Associate Professor in the Department

of CSE at BVRIT HYDERABAD College of Engineering for Women, Hyderabad, India. His areas of interest are data mining, Deep learning, Machine Learning, and Network Security. He has published more than 10 papers in National and International Journals.

G. Srinivasa Rao graduated in B. E(CSE) from Marathwada university, India in the year 1989, received master's degree M. S in software systems from Birla Institute of Technology and Science, Pilani in 1996 and M. Tech (CSE) from JNTU Hyderabad in the year 2008, and he is pursuing Ph.D. from JNTUH, Hyderabad. He has 32 years of teaching Experience. Presently he is working as Associate Professor in the department of Information Technology, RVR & JC College of Engineering, Guntur. His research interest includes Image and Signal Processing, algorithms, and web technologies.