RES MILITARIS
REVUE EUROPEENNE D ETUDES EUROPEAN JOURNAL OF MILITARY STUDIES
Social Science Journal

AUTOMATED DETECTION OF HELMET VIOLATIONS USING DEEP LEARNING IN TRAFFIC REGULATION SYSTEMS

P. Deepthi[1*], T. Shiva Prasad[1], T. Bhanu Prakas[1], G. Uday Sai Teja[1], J. Vinay[1]

[1]Department of Computer Science and Engineering, Sree Dattha Institute of Engineering and Science, Sheriguda, Hyderabad, Telangana, India

[*]Corresponding E-mail: deepthi.pola14@sreedattha.ac.in

ABSTRACT

In the current situation, India faces numerous challenges in traffic regulations that can be mitigated through innovative solutions. Riding motorcycles or mopeds without wearing a helmet is a prevalent traffic violation that has led to an increase in accidents and fatalities. The existing system primarily relies on CCTV recordings to monitor traffic violations, requiring traffic police to identify the violation and zoom in on the license plate if the rider is not wearing a helmet. This process demands significant manpower and time due to the frequent traffic violations and the rising number of motorcycle users. An automated system that detects helmet violations and extracts the vehicle's license plate number would significantly enhance efficiency. Recent research has utilized machine learning methods to address this issue, although these systems often face limitations in terms of efficiency, accuracy, or the speed of object detection, resulting in lower performance. To address these challenges, a Non-Helmet Rider Detection system has been developed. This system aims to automate the detection of helmet violations and the extraction of vehicle license plate numbers. The core principle involves Object Detection using Deep Learning at three levels. Initially, the system detects a person and motorcycle/moped, followed by helmet detection at the second level using the You Only Look Once (YOLO) model.

Keywords: Traffic regulation, Computer vision, YOLO model, Object detection.

1. INTRODUCTION

Helmet detection for riders is a crucial aspect of road safety and is typically implemented through various technological means, such as computer vision systems, deep learning algorithms, and surveillance cameras. This technology is designed to identify and verify whether a motorcycle or scooter rider is wearing a helmet while on the road. The process of helmet detection begins with the deployment of cameras strategically positioned at key points on roads, intersections, or highways, often as part of a broader traffic management or safety system. These cameras continuously capture real-time video footage of the traffic, including riders on two-wheelers. Computer vision algorithms, powered by machine learning models like Convolutional Neural Networks (CNNs), play a pivotal role in helmet detection. These algorithms analyze the incoming video stream, identifying and tracking the presence of riders in the frame. Once a rider is detected, the system focuses on the head region to determine whether they are wearing a helmet. It does so by recognizing the distinct shape and color of a helmet. If a helmet is detected on the rider's head, the system registers a positive result; otherwise, it flags the rider as non-compliant.

Helmet detection systems often include advanced features such as motion analysis, which helps in distinguishing between riders and pedestrians, and can even evaluate the proper fastening of the helmet straps. Additionally, some systems incorporate real-time alerts and notifications to inform law enforcement or relevant authorities about non-compliant riders. The importance of helmet detection lies in its contribution to road safety. Wearing helmets significantly reduces the risk of head injuries during accidents, making it a crucial safety measure for motorcycle and scooter riders. By enforcing helmet-wearing regulations through automated detection systems, governments and traffic authorities aim to decrease the number of head injuries and fatalities on the road.

2. LITERATURE SURVEY

Marzuki, P., et al. (2019) [1] proposed an improved Convolutional Neural Network (CNN) algorithm approach for license plate recognition system. The main contribution of this work is on the methodology to determine the best model for four-layered CNN architecture that has been used as the recognition method. Two types of accuracies are taken at different stages. The first accuracy is taken at the preprocessing part the system and the second one is taken at the classification stage. The classification result was taken according to the number of characters successfully recognized. It described that the preprocessing part achieved 74.7% out of 300 samples tested which does not achieve the expectation level. Ma, Lixin, and Yong Zhang (2021) [2] proposed convolutional neural network. The cycle juice used was based on information obtained from the original. Experimental results showed that the method had a high accuracy rate of 94% and no return rate of 88% for the image data block.

Vaiyapuri, Thavavel, et al (2021) [3] proposed the technique, it had a total of four major processes namely preprocessing, License Plate (LP) localization and detection, character segmentation, and recognition. Hough Transform (HT) was applied as a feature extractor and SSA-CNN algorithm was applied for character recognition in LP. This study introduced a robust DL-based VLPR model using SSA-CNN model. The proposed model had a total of four major processes namely preprocessing, LP localization and detection, HT-based character segmentation, and SSA-CNN based recognition. The input image was pre-processed to make it compatible with further processing.

Huang, Zhao-Kai, Hao-Wei Tseng, and Cheng-Lun Chen (2019) [4] proposed new method for vehicle license plate recognition on the basis of Extreme Learning Machine. ELM was a new category of neural networks which possesses compelling characteristics essential for license plate recognition, such as low computational complexity, fast training, and good generalization (as opposed to traditional neural networks). The proposed method studied incorporation of three ELMs (i.e., basic ELM, incremental ELM, and enhanced incremental ELM) into a typical pipeline for automatic license plate recognition. In the preliminary study (under Windows PC with MATLAB), the success rate of recognition was 87.5% with execution time of 0.3s. A comparative study also showed that the proposed method outperformed conventional approaches (template matching, edge statistics, RBF, and SVM) in terms of accuracy and speed.

Neto, Edson Cavalcanti, et al (2019) [5] proposed a new system to detect and recognize Brazilian vehicle license plates, in which the registered users have permission to enter the location. For this, techniques of Digital Image Processing were used, such as Hough Transform, Morphology, Threshold and Canny Edge Detector to extract characters, as well as Least Squares, Least Mean Squares, Extreme Learning Machine, and Neural Network Multilayer Perceptron to identify the numbers and letters. The system was tested with 700 videos with a resolution of 640 × 480 pixels and AVI format, granting access only when the plate was registered, getting a 98.5% success rate on the tested cases. The movement detection step was linked to the system, becoming faster and more accurate in real time.

Halin, Alfian Abdul, et al (2020) [6] proposed a probabilistic technique to localize license plates regions for cars adhering to the standard set by the Malaysian Road Transport Department. Images of the front/rear-view of cars displaying their license plates are firstly pre-processed, followed by features extraction generated from connected components analysis. These features were then used to train a Naïve Bayes classifier for the final task of license plates localization. Experimental results conducted on 144 images have shown that considering two candidates with the highest posterior probabilities better guarantees license plates regions were properly localized, with a recall of 0.98.

Akhtar, Zuhaib, and Rashid Ali (2020) [7] proposed a number plate recognition method by processing vehicle's rear or front image. After imaged was captured, processing is divided into four steps which are preprocessing, number plate localization, character segmentation and character recognition. Preprocessing enhances the image for further processing, number plate localization extracted the number plate region from the image, character segmentation separates the individual characters from the extracted number plate, and character recognition identified the optical characters by using random forest classification algorithm. Experimental results revealed that the accuracy of this method was 90.9%.

Sathiyabhama, B., et al (2020) [8] proposed a deep learning-based algorithm was proposed for detecting both vehicles and number plates for a reputed company surveillance dataset. The proposed model used a video dataset as an input and the video was segmented into several frames. Using pre-trained weights and labels of the dataset, the vehicles and its number plate were detected by the dark flow toolkit. This tool provided to extract the region of the vehicle with proper annotation. In future work, the proposed model aimed to calculate the speed of the vehicle based on the surrounding area.

Tabrizi, Sahar S., and Nadire Cavus (2019) [9] proposed a LPR systems for Iranian license plates. Increasing the accuracy of the character recognition phase rate and decreasing the training rate were the main advantages of the new Hybrid model. The K-NN was implemented as the first classification method, as it was simple, robust against a noisy data set and effective in large data sets with zero training cost. The confusion problem related to similar characters in the license plates was overcome by using the multiple SVMs classification model. The SVMs were improved the performance of the K-NN in the recognition of similar characters. The SVMs was trained and tested only for the similar characters, thus, the training cost of the SVMs decreased significantly. Comparison results between the current study experimental results of a similar study23revealed that that the presented hybrid KNN-SVM model improved the character recognition rate significantly from 94% to 97.03% for all cases tested.

Singh, Jaskirat, and Bharat Bhushan (2019) [10] proposed a robust technique for License Plate Detection (LPD) in the images using deep neural networks, Pre-process the detected license plate sand perform License Plate Recognition (LPR) using LSTM Tesseract OCR Engine. According to their experimental results, they have successfully achieved robust results with LPD accuracy of 99% and LPR accuracy of 95%just like commercial ANPR systems i.e., Open-ALPR and Plate Recognizer.

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 (2019) [12] proposed a fast algorithm for automatic license plate detection system for the Egyptian license plates that achieved a high detection rate without the need 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, and Alireza Ahmadyfard [14] (2019) 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 neighbourhood signature) method is used. A well-organized database, consisting of car images with different known distances and viewing angels were 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%.

3. PROPOSED SYSTEM

Figure 1 shows the proposed block diagram, which is used to identify the person, bike rider with having helmet. 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 final result is a list of bounding boxes and class probabilities for each person in the input image.

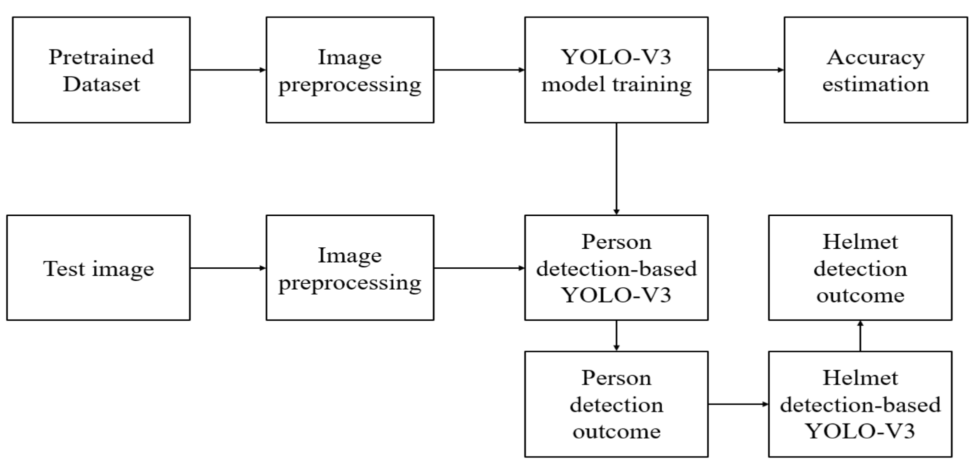

Figure 1. Proposed block diagram.

Step 1. Test Image: Start by obtaining the image to be analyzed. The image can be sourced from a security camera, a photograph, or any other relevant source.

Step 2. Image Preprocessing: Before running YOLOv3 for person and helmet detection, conduct preprocessing on the image to enhance detection accuracy. Common preprocessing steps encompass resizing the image to the required input size for YOLOv3, normalizing pixel values, and handling color channels (e.g., converting from BGR to RGB).

Step 3. Person Detection using YOLOv3: Employ a pre-trained YOLOv3 model for object detection. Pre-trained weights and configurations for YOLOv3 can be sourced from resources such as the Darknet website or other deep learning libraries. Load the YOLOv3 model into the working environment. Input the preprocessed image into the YOLOv3 model. Extract the detection results, including bounding boxes around detected persons, their confidence scores, and class labels (e.g., "person"). Apply filtering to retain only high-confidence bounding boxes on test image.

Step 4. Helmet Detection using YOLOv3: Analogously, use a pre-trained YOLOv3 model specialized in helmet detection. This model should be trained specifically to recognize helmets. Load the helmet detection YOLOv3 model into the working environment. Input the same preprocessed image into the helmet detection YOLOv3 model. Extract the detection results, which encompass bounding boxes around detected helmets, their confidence scores, and class labels (e.g., "helmet").

Step 5. Output and Analysis: Combine the person and helmet detection results to identify individuals wearing helmets. Visualize the results by drawing bounding boxes around persons and helmets in the original image for clarity and interpretation.

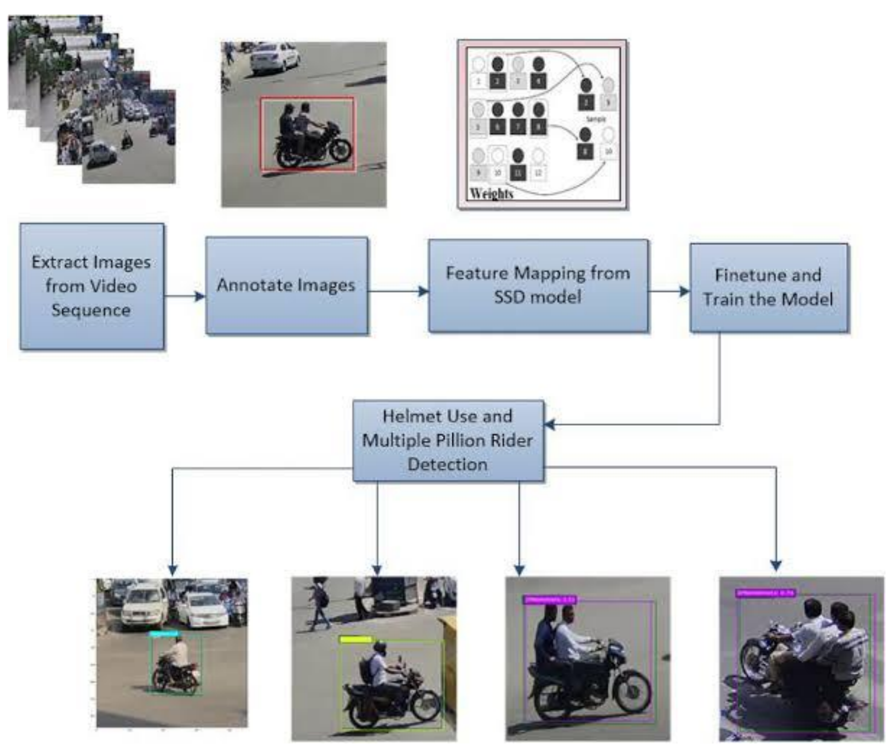

Figure 2. Proposed system architecture of non-helmet rider detection system.

YOLO-V3 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. 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 aforementioned 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.

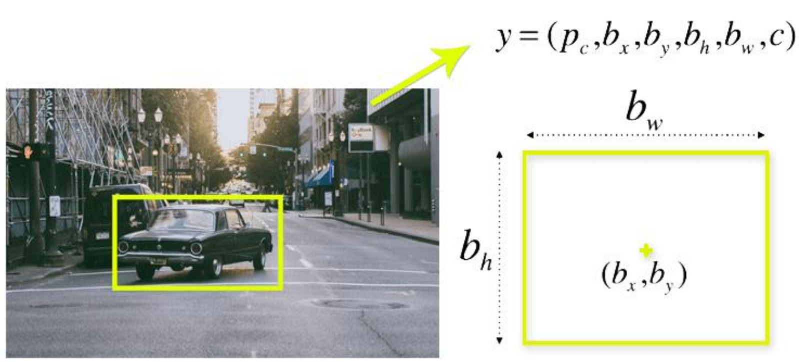

Figure 3. Bounding box regression

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.

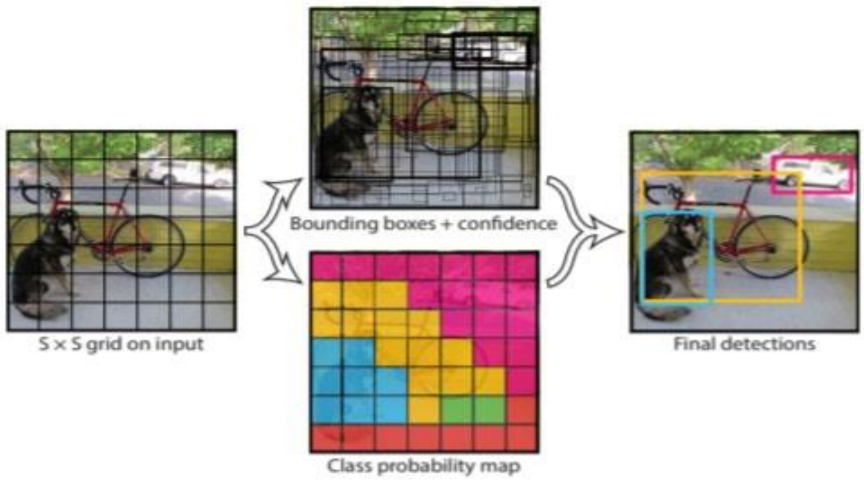

Figure 4. Combination of three modules.

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.

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 this Figure 5, YOLOv3, a popular object detection algorithm, has been applied to an image. The result shows that YOLOv3 has successfully detected a person within the image and has drawn a bounding box around the detected person. Additionally, an accuracy score of 99.69% is associated with this detection. This high accuracy score indicates that the model is extremely confident in its assessment that a person is present in the image, with only a very small margin of error.

Figure 5. Predicted outcome as person detected with bounding box.

Figure 6 presents a different scenario where YOLOv3 has been applied to an image, but this time, it has not detected a helmet. The figure does not provide an accuracy score for this particular detection. This outcome suggests that YOLOv3 was unable to identify a helmet in the image, possibly due to factors such as the absence of a helmet, poor visibility, or challenging angles.

Figure 7 is similar to Figure 3 in that it depicts YOLOv3 successfully detecting a person and drawing a bounding box around the detected individual. The key difference is the accuracy score, which is even higher at 99.89%. This indicates an even greater level of confidence in the model's ability to identify a person in the image with a very low probability of error.

Figure 6. Predicted outcome as helmet not detected.

Figure 7. Predicted outcome as person detected with bounding box.

Figure 8 shows the detection of a helmet by YOLOv3 in an image. The model has identified the presence of a helmet, and the associated accuracy score is 99.00%. This accuracy score signifies a high level of confidence in the model's ability to correctly recognize a helmet in the image, with only a 1% margin of potential error.

Figure 8. Predicted outcome as helmet detected.

Table 1 shows a performance comparison table for different methods used for person and helmet detection, with accuracy percentages reported for each method. Let's break down the information in the table:

- **Person Detection Accuracy (%)**:
  - **Existing SVM**: The existing Support Vector Machine (SVM) method achieves an accuracy of 95.47% for detecting persons.
  - **Existing RFC**: The existing Random Forest Classifier (RFC) method achieves an accuracy of 92.00% for person detection.
  - **Proposed Method**: The proposed method, which is not specified further, achieves a significantly higher accuracy of 99.88% for person detection. This suggests that the proposed method outperforms both existing SVM and RFC methods in accurately detecting persons.
- **Helmet Detection Accuracy (%)**:
  - **Existing SVM**: The existing SVM method achieves an accuracy of 93.13% for helmet detection.
  - **Existing RFC**: The existing RFC method performs better in helmet detection, with an accuracy of 96.48%.
  - **Proposed Method**: The proposed method achieves an accuracy of 99.00% for helmet detection. Similar to person detection, the proposed method outperforms both existing SVM and RFC methods, indicating superior performance in accurately detecting helmets.

Table 10.1. Performance Comparison

Method	Existing SVM	Existing RFC	Proposed Method
Person detection accuracy (%)	95.47	92.00	99.88
Helmet detection accuracy (%)	93.13	96.48	99.00

5. CONCLUSION AND FUTURE SCOPE

In conclusion, the results indicate that the proposed YOLOV3 significantly outperforms the existing SVM and RFC methods in terms of accuracy. It achieves an impressive accuracy of 99.88% for person detection and 99.00% for helmet detection, showcasing its superior capability in accurately identifying persons and helmets in images or video frames. These high accuracy scores suggest that the proposed method holds great promise for applications requiring precise object detection, such as safety and security systems. However, further research and analysis are necessary to fully understand the inner workings and limitations of the proposed method, and it is essential to assess its performance across a broader range of scenarios and datasets to validate its effectiveness.

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] 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.