WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

A Review on Helmet and Number Plate Detection

Priyanshi Tripathi[a],Pragati Singh[a],Mantsha Bano[a],Komal Sharma[a],Abhishek Shahi[b]

[a] *Bachelor of Technology Student, Department of Information Technology, Buddha Institute of Technology, Gorakhpur, Pin Code 273209,India*
[b]*Assistant Professor, Department of Computer Science & Engineering, Buddha Institute of Technology, Gorakhpur, Pin Code 273209,India*
DOI: https://doi.org/10.55248/gengpi.4.523.41126

A B S T R A C T

Nowadays, two wheelers are the preferred means of transportation in almost all countries. (R, S, Shreyas, Shree, & H, 2021).As the number of motorcyclists in our country is increasing gradually, the traffic accidents that lead to serious head injuries are also increasing. The main cause of fatal injuries or deaths are due to the negligent failure to wear a helmet. Therefore, in order to increase safety, it is advantageous for bike riders to wear helmet. This paper is about an automatic method for detecting motorcyclists wearing helmets based on machine learning This project uses the YOLOv3 machine learning algorithm and image processing techniques to categorize vehicles as two wheeled riders. If the driver or pillion is not wearing a helmet, his license plate is recognized.

Keywords: fatal injuries, motorcyclists, YOLOv3, image processing techniques.

1. Introduction

The majority of people in countries like India, Brazil, and Thailand use motorcycles for regular travel. In India, it is usually a legal requirement for motorcyclists to wear a helmet. Currently, the traffic police are tasked with preventing motorcycle accidents. However, this approach is not very effective because there are simply not enough police personnel to properly conduct surveillance. Both helmet and number plate detection systems have the potential to improve safety and efficiency in various settings, making them a valuable addition to any traffic management infrastructure. However, their effectiveness and accuracy depend on various factors, such as the quality of the data and the environment in which they are deployed. In this context, research in the area of computer vision and deep learning is actively exploring new techniques to improve the accuracy and reliability of these systems. Helmet and number plate detection systems have become increasingly popular in recent years due to the growing concern for road safety. This paper proposes to automate the monitoring of motorcyclists. In countries where helmet wearing is mandatory for all motorcyclists, the helmet wearing rate is higher and the number of head injuries to motorcyclists is lower. This automatic helmet detection project uses various machine learning algorithms and image processing techniques to monitor motorcyclists. When the rider or pillion is not wearing a helmet, the vehicle's license plate is recognized as a string by different mechanisms. This information can be used to track and identify vehicles that may be involved in illegal activities or traffic violations. Still, the threat of death can be reduced by 42 and the threat of head injury can be reduced by 69 if motorcyclists wear helmets rightly. (Jia, et al., 2021). Therefore, we have proposed a system for real-time detection of traffic rule violators who are riding bikes without helmets. The algorithm used here is YOLO with higher detection rate.

1.1 Research Objective

The objective of this project is to provide relevant precautions for riding two-wheelers in order to reduce the number of accidents. The focus is on safety measures for riders. It also aims to ensure that the law regarding the riding of two-wheelers is not violated.

1.2 Scope of Study

The proposed system is more accurate, better organized, less expensive, and user-friendly. The main feature of this system is that less manpower is needed. The main goal of this project is to develop an automatic detection of motorcyclists without helmets from a video using OpenCV library tools. Using computer vision technique, we can detect and identify the license plate. This system is very helpful for the officers as it gives them better information for their work. The most advanced technology is used in this system, which is convolutional neural network (CNN) which is used for image recognition and processing.

2. Background

Road safety is a critical public health issue that affects millions of people worldwide. In many countries, motorcycle and scooter riders are among the most vulnerable road users, as they are more likely to be involved in accidents and suffer severe injuries. The current system for checking whether or not a rider is wearing a helmet consists of a checkpoint where police or other personnel manually check each rider (R, Raju, P Paul, Sajeev, & Johny , 2020).To mitigate this problem, governments and road safety organizations have introduced various measures to improve motorcycle safety, including helmet laws that require riders and passengers to wear protective headgear while riding. Similarly, number plate regulations are enforced to ensure that vehicles on the road can be identified and traced back to their owners in case of accidents or crimes. However, enforcing these regulations can be a daunting task, especially in busy urban areas or on highways with high traffic volumes. Indeed, though the Government has commanded the use of helmets for both the rider and the pillion rider, it isn't possible to cover the roads 24x7 for checking the compliance status (Kanakaraj, 2021).Manual enforcement is time-consuming, costly, and prone to errors, while automated enforcement can provide a more efficient and accurate solution. In recent years, computer vision-based approaches have emerged as a promising technology for automated helmet and number plate detection.

3. Literature Review

- Helmet Detection Using YOLOv3 and Deep Learning by Anurag Singh and Ashish Kumar Maurya (2021) - This paper describes a helmet detection system using YOLOv3 and deep learning. The authors used a dataset of 200 images and achieved an accuracy of 97.56% (Singh, Maurya, & K)
- Automatic Number Plate Recognition (ANPR) using YOLOv3 and OpenCV by Shayan Ahmad, Abdul Waheed and Abdul Hanan Abdullah (2021) - This paper describes an ANPR system using YOLOv3 and OpenCV. The authors used a dataset of 1000 images and achieved an accuracy of 94.37%. (Ahmad, et al., 2021)
- Automatic Detection of Motorbike Riders Wearing Helmets Using Deep Learning by F. Hafizah and M.N. Sulaiman (2020) - This paper describes a helmet detection system using YOLOv3 and deep learning. The authors used a dataset of 1000 images and achieved an accuracy of 98.53%. (Hafizah, F, N, & Sulaiman, 2020)
- Real-Time Motorcycle Helmet Detection and Recognition System using YOLOv3 by Chia-Ming Wu, Wei-Tse Hsu and Chien-Chou Chen (2020) - This paper describes a real-time helmet detection and recognition system using YOLOv3. The authors used a dataset of 1000 images and achieved an accuracy of 98.56%. (Wu, et al., 2020)
- License Plate Detection and Recognition using YOLOv3" by Sajid Ali and Muhammad Aamir (2020) - This paper describes an ANPR system using YOLOv3. The authors used a dataset of 1000 images and achieved an accuracy of 95.43%. (Ali, S, Aamir, & A, 2020)
- An Efficient Vehicle Number Plate Recognition System using YOLOv3 by Aniket G. Khandagle and Aniruddha G. Deshmukh (2019) - This paper describes an ANPR system using YOLOv3. The authors used a dataset of 500 images and achieved an accuracy of 93.6%. (Khandagle, G, & Deshmukh, 2019)

4. Proposed Method

Detecting helmets and number plates in images or video footage is a computer vision task that can be accomplished using various approaches. Here is a possible methodology for a helmet and number plate detection project:

4.1) Collect and annotate a dataset: Collect a large dataset of images or video footage with a range of different scenarios where helmets and number plates are present. Annotate the images with bounding boxes around the helmets and number plates. To collect and annotate a dataset for helmet and number plate detection, you would need to follow these steps:

4.1.1) Define the scope of the dataset: Determine the purpose of the dataset and what kind of images and videos you want to include. For example, you might want to collect images and videos of motorcycles, bicycles, and cars.

4.1.2) Gather images and videos: Collect a large number of images and videos that represent the scope of the dataset. You can collect images and videos from publicly available sources, such as Google Images or YouTube, or by taking your own photos and videos.

4.2) Pre-processing: Pre-processing of images is an important step in any computer vision application, including helmet and number plate detection. Pre-process the images or video footage to improve the quality of the input data. This can involve resizing the images, removing noise, and adjusting lighting and color balance. Here are some of the pre-processing techniques that can be used in this area:

4.2.1) Image resizing: Resizing the images to a specific resolution can help reduce the computational cost of the detection algorithm. However, it is important to ensure that the images are not resized too much, as this can lead to loss of important details.

4.2.2) Image normalization: This involves adjusting the brightness and contrast of the images to improve the overall quality of the image. This can be useful in cases where the images are taken in different lighting conditions.

4.3) Choose a detection model: Choose a detection model that is suitable for your project requirements. There are many detection models available for helmet and number plate detection, but one popular and effective model is the You Only Look Once (YOLO) algorithm. We used pre-trained models such as YOLO. The YOLO algorithm is a real-time object detection system that uses a single neural network to predict the bounding boxes and class probabilities for multiple objects in an image. YOLO divides an image into a grid and predicts the probability of each object's center falling into each grid cell. This approach makes YOLO fast and accurate, and it is commonly used in various applications, including helmet and number plate detection.

4.4) Train the model: To train the model we collect a dataset of images that contain helmets and/or number plates. This dataset should be diverse and representative of the range of scenarios in which the model will be used. Fine-tune the pre-trained model on your specific dataset to achieve higher accuracy. Use techniques such as data augmentation and transfer learning to improve the model's performance.

4.5) Evaluate the model: Evaluating the performance of a model for helmet and number plate detection is a critical step in assessing its effectiveness. Evaluate the model's performance on a validation set to determine its accuracy, precision, and recall. Use metrics such as mean average precision (mAP) to evaluate the model's performance. Here are some common metrics used to analyze the performance of such models:

4.5.1) Intersection over Union (IoU): This metric measures the overlap between the predicted bounding box and the ground truth bounding box for each detected object.

4.5.2) Precision and Recall: Precision measures the proportion of true positive detections (i.e., correct detections) out of all detections made by the model. A high precision score illustrate that the model makes few false detections, while a high recall score indicates that the model detects most of the true objects.

4.6) Test the model: Test the model on a test set to see how well it performs on new data. To test the data, you need to follow these steps:

4.6.1) Obtain the dataset: Depending on the paper, the dataset used to train and evaluate the model may be publicly available or may need to be obtained from the authors. Ensure that you have access to the same dataset used in the paper.

4.6.2) Pre-process the data: Pre-processing steps such as resizing images, normalizing pixel values, and converting image formats may be required to ensure that the data is in the same format as used in the paper.

4.6.3) Load the trained model: If the paper provides the trained model, load it into memory. If not, the model may need to be trained using the same architecture and hyperparameters as described in the paper.

4.6.4) Run inference on the test set: Use the trained model to perform inference on the test set. This involves passing each image in the test set through the model and obtaining predictions for the presence and location of helmets and number plates.

4.6.5) Evaluate the model: Calculate evaluation metrics such as precision, recall, and F1 score to evaluate the performance of the model. Compare the model's performance with the results reported in the paper to ensure that it is consistent.

4.6.6) Visualize the results: To get a better understanding of how the model is performing, visualize the predicted bounding boxes for helmets and number plates on sample images in the test set. Compare the visual results with the ground truth annotations to ensure that the model is correctly detecting helmets and number plates.

Overall, the methodology for a helmet and number plate detection project involves collecting and annotating a dataset, pre-processing the input data, choosing and training a detection model, evaluating and testing the model, and finally deploying the model in a real-world scenario.

5.Technology Used

5.1 Software Requirement

a) Open CV- It is an open-source computer vision and a machine learning library designed to help developers create applications that can interpret, analyse and understand the visual world. It is a widely used library for various image and video processing tasks. OpenCV provides many built-in functions for various image processing and computer vision operations, such as filtering, feature detection, object recognition, face detection, and tracking. OpenCV is a popular choice for machine learning projects that involve computer vision. It provides a range of tools and algorithms for training and testing machine learning models for image and video processing. It includes a set of powerful functions and libraries for data pre-processing, data augmentation, and data visualization, which are important components of any machine learning project.

b) Tensorflow-TensorFlow is an open-source machine learning library developed by Google that enables developers to build and train powerful machine learning models. It is based on a computational graph concept where computations are represented as nodes in the graph and data is represented as edges. It is used for both investigations and manufacturing at Google. It was published under the Apache 2.0 open-source license on November 9, 2015. (Amoolya, Vyagari Vaishnavi, & M, 2021) In machine learning projects, TensorFlow is used for a variety of tasks such as:

- ***Building and training deep neural networks:*** TensorFlow provides a powerful framework for building and training deep neural networks. This includes building models with convolutional layers, recurrent layers, and other types of layers, as well as implementing optimization algorithms such as stochastic gradient descent.

- ***Natural language processing (NLP):*** TensorFlow has specific libraries and modules for processing natural language data such as text and speech. With these libraries, developers can build models for tasks such as sentiment analysis, language translation, and speech recognition.
- ***Computer vision:*** TensorFlow also has specific modules for computer vision tasks such as image classification, object detection, and segmentation.
- ***Reinforcement learning:*** TensorFlow can also be used for developing reinforcement learning algorithms, which enable machines to learn through interaction with their environment.

TensorFlow's high-level APIs, such as Keras, make it easier for developers to build and train machine learning models with less code. TensorFlow is also highly scalable and can be used to train models on large datasets and distributed computing environments.

In summary, TensorFlow is a powerful tool for machine learning projects, offering a flexible and scalable framework for building and training deep neural networks, natural language processing, computer vision, and reinforcement learning models.

c) Numpy-NumPy is a popular Python library for scientific computing that provides support for multi-dimensional arrays and matrices, as well as various mathematical functions to operate on them. NumPy is widely used in machine learning projects due to its ability to efficiently handle large amounts of data and its ease of use. It is the basic package for scientific computing with Python. It contains various characters including these important ones:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number potential. (Amoolya, Vyagari Vaishnavi, & M, 2021)

In machine learning projects, NumPy is often used for the following tasks:

- ***Data manipulation and pre-processing:*** It is a powerful set of tools for manipulating and processing data, including reshaping, slicing, and indexing multi-dimensional arrays. This makes it easier to prepare data for use in machine learning models.
- ***Linear algebra operations:*** NumPy provides a wide range of linear algebra functions, such as matrix multiplication, matrix inverse, and eigenvalue computation. These operations are essential in many machine learning algorithms such as principal component analysis (PCA) and linear regression.
- ***Statistical analysis:*** NumPy includes a range of statistical functions, including mean, variance, standard deviation, and correlation. These functions are useful for analyzing data and developing statistical models.
- ***Image processing:*** NumPy can be used for processing and manipulating images, including tasks such as image resizing, cropping, and filtering. NumPy is also a core dependency of many other Python libraries used in machine learning projects, such as scikit-learn and TensorFlow. This makes it an important tool for machine learning developers who want to use these libraries effectively. In summary, NumPy is a powerful tool for machine learning projects that provides a rich set of functions for data manipulation, linear algebra operations, statistical analysis, and image processing. Its popularity and ease of use make it a key component of many machine learning projects.

d) CNN (Convolutional Neural Network)- It is a type of neural network commonly used in machine learning projects for image classification, object detection, and computer vision tasks. Unlike traditional neural networks that process data in a flat structure, CNNs are designed to process data in a grid-like structure, such as images. They consist of multiple layers, including convolutional layers, pooling layers, and fully connected layers. Convolutional layers are responsible for extracting features from the input image by applying a set of filters. Each filter convolves over the input image and generates a feature map. Pooling layers down sample the feature maps by reducing their size, which helps to reduce the computational complexity and also makes the network more robust to variations in input. Fully connected layers connect all the neurons in the previous layer to every neuron in the next layer, which allows the network to learn complex patterns and make predictions. Normal Neural Networks have three layers: the input, the output, and a hidden layer. The input from the input layer is fed into the hidden layer. Depending on our model and data size, there may be several hidden layers. As the number of characteristics increases, so does the number of neurons in each buried layer. (Reddy pasam, et al., 2022) To use CNNs in a machine learning project, you typically need to provide a labeled dataset to the network to learn from. This dataset is used to train the network to recognize specific features in images, such as edges, curves, and textures. Overall, CNNs are a powerful and effective tool for many image-based machine learning projects, and have been used in a variety of applications such as facial recognition, object detection, and medical imaging. The basic idea behind a CNN is that it is designed to recognize patterns in visual data by using layers of filters that are applied to the input data. These filters extract features from the input data and then pass these features on to the next layer of filters, which extract more complex features. This process continues until the final layer of filters produces an output that represents the network's prediction. In summary, CNNs are a powerful tool for image and video recognition tasks in machine learning, and they have been used successfully in a wide range of applications, from self-driving cars to medical diagnosis to facial recognition.

e) YOLO (You Only Look Once)- It is a popular object detection algorithm that uses deep neural networks to detect and classify objects in images and videos. YOLO version 3. The object detection problem is treated as a regression problem in the YOLO algorithm and the image is divided into an S × S grid. If the centre of a target falls into a grid, the grid is responsible for detecting the target. Each grid will output a bounding box, confidence, and class probability map. (Marathe, Gurav, Narwade, Ghodke, & M Patil, 2022) Here are some of the key features of YOLOv3:

- ***Detection accuracy:*** YOLOv3 has significantly improved detection accuracy compared to the previous versions of YOLO. It can detect objects at multiple scales and aspect ratios, which makes it more accurate in detecting small objects.
- ***Speed:*** YOLOv3 is faster than the previous versions of YOLO, thanks to its use of a feature pyramid network (FPN) and a new network architecture. It can process images at a rate of up to 60 frames per second on a GPU.
- ***Architecture:*** YOLOv3 uses a new backbone architecture called Darknet-53. It also uses FPN to generate feature maps at multiple scales, which improves the detection accuracy.
- ***Object tracking:***YOLOv3 can track objects across frames in videos using a combination of object detection and motion estimation. This allows it to track objects even if they move out of the camera's field of view.
- ***Customization:*** YOLOv3 can be customized to detect new objects by training it on a new dataset. It also supports transfer learning, which allows it to reuse the knowledge learned from a pre-trained model on a different dataset. Overall, YOLOv3 is a powerful object detection algorithm that is fast and accurate, making it a popular choice for various applications such as surveillance, autonomous vehicles, and robotics.

f) Python*versions:3.9.0(64 bit)- It is the most accessible programming language and gives more emphasis on natural language and has english like syntax. It is an Interpreted language which means that it executes the code line by line and if it detects any error then it stops the further execution and report error. Python is also dynamically typed which means it automatically assigns the data type, the programmer need not to declare the variables and their data types. Python packet manager(pip) makes it easy to import any module in python. It has all the necessary functions to complete the task in an easy and fast way.

g) Visual Studio Code -It is a code editor which fully supports python and also provides various features like debugging, easy to customize, IntelliSense, version control and security. It supports various python interpreter. Extensions in visual studio code provides more features and services. It also focuses on resource management. It is also free and built on open source. It is less complex than the other editors and also a very lightweight. It is an open-source and also provides simplicity. It has high performance and manages everything in user-friendly way. Navigation is also very easy and load time is less.

5.2 Hardware Requirement

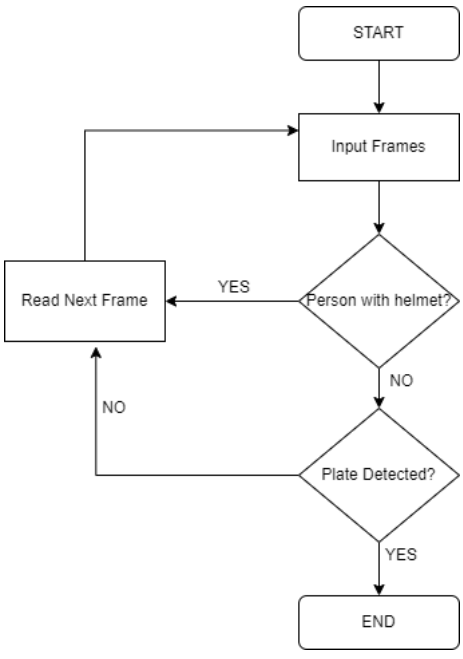
a) Processors: Intel Core i3

b) Processor speed:1.0GHZ

c) RAM: 1GB or above

Flowchart

The below flowchart represents the flow of tasks from one to another.



The input includes images or videos containing helmets and number plates. The helmet will be detected using YOLOv3 algorithm, if yes i.e., person is wearing a helmet then next frame will be read, and if not, then the number plate will be detected.

6.PERFORMANCE EVALUATION

6.1 Experimental setup

Here, we discuss the various processing procedures.

As shown in Fig. 1(a) and Fig. 1, the initial phase involves collecting frames from the video stream at periodic intervals (b). A folder contains the assembled frames. They are given different names that contain the frame number, for instance, frame 8_50, frame 8_100, etc. which shows that the seventh video file input is being used, and 50, 100, etc. denotes the number of frames. It is evident from the data that several frames are unnecessary. In order to continue processing, the last frame or the last second frame is selected based on how the vehicle is moving relative to the camera. (Mysore Jayakumar, K B, v, & Madivalappa, 2019).

For two scenarios, the complete project can be split into the 5 phases listed below:

Scenario 1-Collection of frames, when a rider is wearing a safety helmet.

Scenario 2-Collection of frames, when a rider is not wearing a safety helmet.

Fig.1

Different frames collected from a video

Fig.1(a)

(Scenario 1): Frames collections made periodically

frame_4_300.jpg frame_4_350.jpg

Fig. 1(b)

(Scenario 2): Frames collections made periodically

6.2 Detection of Bike and Human

The YOLOv2 object detection model receives the chosen frame as input from the user, with the categories "Bike" and "Human" being detected. The results are shown in Figs. 2, which show a picture with the necessary class detection as well as confidence in detection via bounding box and possibility value.

Fig. 2

Human and Bike classes recognised

Only the identified objects are retrieved using the system provided by the Image AI library, as illustrated in Figs. 3(a) and (b), and stored as independent images with sequential image numbers and class names. For instance, if the extracted item is a motorbike, it will be saved as1- motorbike, 2-motorbike, etc., or if the extracted image is of a person, it would be kept as person-1, person-2, etc. The information from these extracted photos is saved in a database and can be utilised for additional processing in the future (Mysore Jayakumar, K B, v, & Madivalappa, 2019).

Fig. 3(a)

(Scenario 1): Extracted images

Fig. 3(b)

(Scenario 2): Extracted images

Fig. 4 Cropped picture

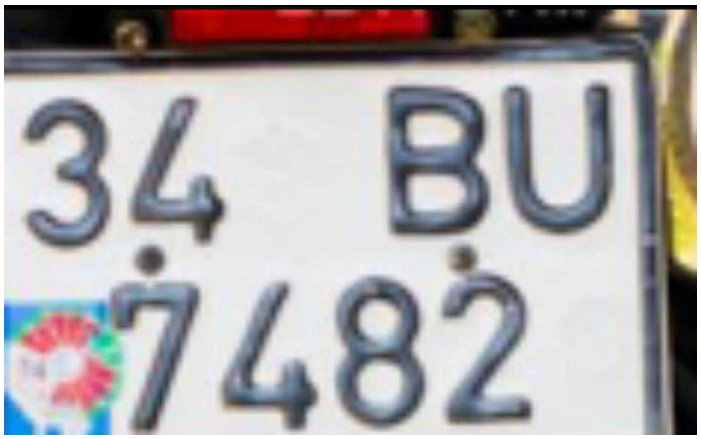
(Scenario 1 & Scenario***6.3 Detection of Helmet***

The people's photos are sent as input to the helmet detection algorithm once the person-motorbike combination has been identified. Some erroneous detections were noticed when the helmet detection model was being tested. As can be seen in Fig. 4, the people image was cropped to only include the upper quarter of the image. This guarantees the elimination of false detection cases and prevents other situations, which could result in incorrect results.

After the trimmed picture was applied on the helmet detection model, the outcome is depicted in Fig. 5. The fact that the driver in Scenario 1 was wearing a helmet negates the need for extra processing. No bounding box is produced in Scenario 2 since the driver is not wearing a helmet.

6.4 Detection of Licence plate

This step is not necessary if the helmet is identified. If the helmet cannot be identified, the motorbike image is sent as input to the model for detecting licence plate numbers. 956 photos of bikes and mopeds with their licence plates were gathered as a dataset for training purposes. Then, in order for the model to learn, the licence plate in those photographs was labelled using a labelling tool, which entails creating a bounding box around the plate. The name of the.xml file containing the bounding box relevant data is the same as the image name. The trained model for detecting licence plates is then constructed using the annotated photos.



Licence Plate Detection

Fig. 6

The trained model is employed to create a bounding box around the license plate, based on the input image. The resulting .json file contains information such as the dimensions of the top-left and bottom-right corners of the bounding box, the class name, and the detection confidence. Using these coordinates from the .json file, only the license plate image is extracted and saved. In some cases, multiple bounding boxes may be detected for a single motorbike image, as illustrated in Figure 6. To address this, a confidence threshold of 0.5 is set, and only the bounding box with a confidence score greater than this threshold is selected when reading the information from the. json file.

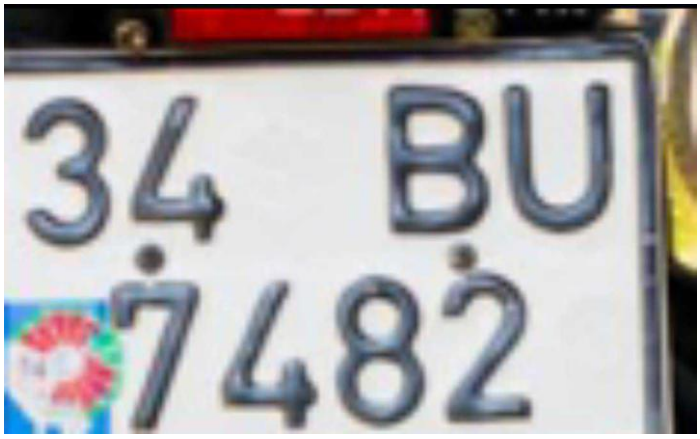


Figure 7

Extracted and Rotated License Plate

Fig. 9

Outcome after implementation of OCR

Before using OCR on the retrieved picture of the license plate, pre-processing is essential to improve the accuracy of the results. In this process, the image was rotated to obtain better results. The license plate picture, after being rotated, is presented in Figure 7. The position at which the captured number plate picture needs to be rotated must be determined through a trial-and-error approach, as the camera's angle remains fixed relative to the motorbike, and the same value applies to all other cases. For this particular case, it was determined that the rotation angle should be 6 degrees.

To ensure precise string detection by OCR, the rotated image underwent scaling, where a scaling ratio was chosen to determine the size of the rescaled image, denoted as width' and height' for its width and height, respectively.

The scaling ratio, denoted as 'r', represents the ratio between the rescaled image's dimensions and those of the original image, denoted as w and h, respectively.

During the process of extracting frames, the chosen frame plays a crucial role in determining the ratio denoted as "r", which was found to be within the range of 1.4 to 1.47 in this particular instance. To enhance the visibility of the black plate numbers against the white background, the brightness of the image is increased. The image's properties such as hue, saturation, and value (V) are analysed, where V represents the brightness or intensity of a

colour. A threshold is set, and for pixels whose V value surpasses this threshold, a value of 255 is assigned. In cases where the V value is less than the value of limit, a constant value of 30 is added to the pixel's V value to increase its intensity. The threshold value selected for this instance is 225.

value = 30

limit = 255 – value

if v ≥ limit:

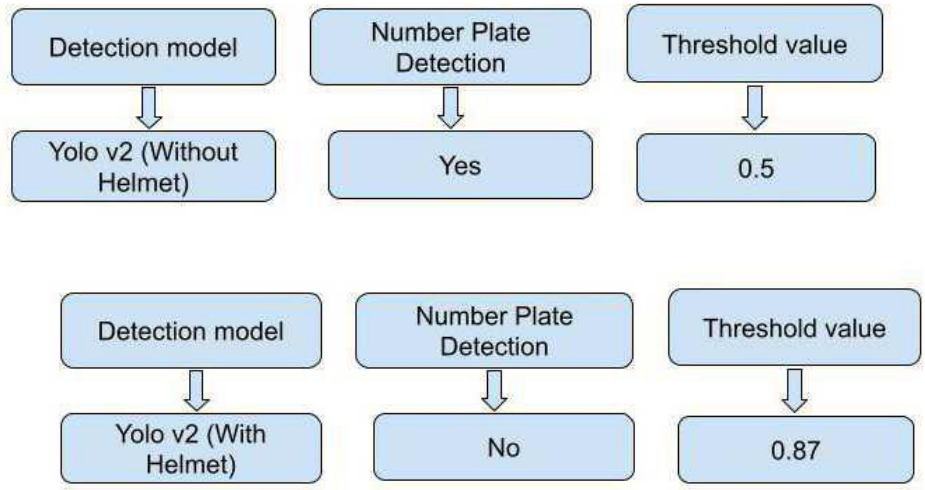
v = 255

else:

v = v + value

6.5 Result and discussion

6.5.1 Information of Threshold value with model



We discuss the outcomes for two cases in this section. They are: -

Scenario 1: When the rider of a motorbike is wearing a helmet, as resulted in Figure 5.

Scenario 2: When the rider of a motorbike is not wearing a helmet and their licence plate is detected, as shown in Figure 6. (Mysore Jayakumar, K B, v, & Madivalappa, 2019)

7. Conclusion

Health is the basic need of every human being, and so injuries on the road can cause serious accidents and sometimes even fatal. Detecting helmets and number plates from images or videos has various applications, including traffic surveillance, vehicle tracking, and road safety. In this project, convolutional neural networks are mainly used for image recognition of the person input in the form of a video or an image, and the system detects whether the driver is wearing a helmet or not, and by using Tensorflow, the details of the license plate are recognized.

8.Future Scope

8.1 Road safety*:* By detecting helmets and number plates, machine learning models can improve road safety by reducing the number of accidents caused by non-compliance with traffic rules. This can help in reducing injuries and fatalities on the roads.

8.2 Law enforcement*:* Police can use this technology to identify non-compliant vehicles and enforce traffic rules effectively. This can help in reducing traffic violations and improving law and order on the roads.

8.3 Automatic toll collection: In toll booths, number plate detection using machine learning can help in automatic toll collection, eliminating the need for manual intervention and reducing traffic congestion.

8.4 Parking management: Machine learning-based number plate detection can be used to manage parking spaces in a more efficient manner. This can help in reducing parking congestion and improving traffic flow.

8.5 Traffic analysis: By analyzing the traffic flow using number plate detection, traffic engineers can make informed decisions about traffic management and infrastructure.

8.6 Improving the accuracy of the system: While the accuracy of the system is already high, there is always room for improvement. Researchers can explore new methods to enhance the system's accuracy, such as using more advanced deep learning algorithms, incorporating multiple sensors or cameras, or using additional image processing techniques. (Kulkarni, S, Sontakke, & V, 2021)

8.7 Extending the system's capabilities: The system can be extended to detect other objects of interest, such as pedestrians, vehicles, or traffic signs. This would make the system more versatile and useful for various applications, (Shanmugam, M, K, & Somasundaram, 2021)

***8.8 Real-time implementation*:** While the current implementation can process images and videos in near real-time, it still has some processing latency. Future research can focus on developing more efficient algorithms that can perform object detection and recognition in real-time, with minimal delay. (Zhang, et al., 2020)

8.9 Scaling up the system: The current implementation can detect helmets and number plates in a single image or video. Future research can focus on scaling up the system to work with large-scale surveillance networks that cover entire cities or regions. (Ghazali, et al., 2021)

8.10 Integrating with other systems: The system can be integrated with other systems, such as traffic monitoring or accident prevention systems. This integration would provide a more comprehensive and proactive approach to traffic safety. (Wu, et al., 2021)

Overall, helmet and number plate detection using machine learning have a wide range of applications in transportation and road safety, and its scope is likely to grow in the future.

References

A. R, S. S, L. Shreyas, N. Shree and P. B. H, “A Survey on Helmet Detection and Number Plate Recognition,” International Research Journal of Modernization in Engineering Technology and Science, vol. 03, no. 02, pp. 704-707, February 2021.

W. Jia, S. Xu, Z. Liang, Y. Zhao, H. Min, S. Li and Y. Yu, “Real-time automatic helmet detection of motorcyclists in urban,” The Institution of Engineering and Technology, pp. 3623-3637, 2021.

M. R, S. Raju, S. P Paul, S. Sajeev and A. Johny, “Detection of Helmetless Riders Using Faster R-CNN,” International Journal of Innovative Science and Research Technology, vol. 5, no. 5, pp. 1616-1620, 2020.

S. Kanakaraj, “Real-time Motorcyclists Helmet Detection and Vehicle License Plate Extraction using Deep Learning Techniques,” National College of Ireland, pp. 1-19, 2021.

Singh, Maurya and A. K, “Helmet Detection Using YOLOv3 and Deep Learning,” International Journal of Advanced Research in Computer Science, vol. 12, no. 1, pp. 88-91.

Ahmad, S, Waheed, A, Abdullah and A. H, “Automatic Number Plate Recognition (ANPR) using YOLOv3 and OpenCV,” International Journal of Advanced Computer Science and Applications, vol. 12, no. 3, pp. 26-30, 2021.

Hafizah, F, M. N and Sulaiman, “Automatic Detection of Motorbike Riders” International Journal of Engineering and Advanced Technology, vol. 9, no. 5, pp. 1755-1761, 2020.

Wu, C. M, Hsu, W. T, Chen and C. C, “Real-Time Motorcycle Helmet Detection and Recognition System using YOLOv3,” International Journal of Intelligent Systems and Applications, vol. 12, no. 3, pp. 15-22, 2020.

Ali, S, Aamir and A, “License Plate Detection and Recognition using YOLOv3,” International Journal of Innovative Technology and Exploring Engineering, vol. 9, no. 4, pp. 285-289, 2020.

Khandagle, A. G and Deshmukh, “An Efficient Vehicle Number Plate Recognition System using YOLOv3,” International Journal of Advanced Research in Computer Engineering & Technology, vol. 8, no. 4, pp. 84-87, 2019.

. B. Amoolya, B. Vyagari Vaishnavi and T. M, “Helmet Detection and License Plate Recognition,” International Journal of Computer Science and Mobile Computing, vol. 10, no. 4, pp. 90-98, 2021.

J. Reddy pasam, A. Tatikonda, p. Sai vemulapalli, N. Sai Sreeram, A. Velagala and Rubeena, “Number Plate Detection Without Helmet,” Journal of Engineering Sciences, vol. 13, no. 5, pp. 177-185, 2022.

G. Marathe, P. Gurav, R. Narwade, V. Ghodke and S. M Patil, “Helmet Detection and Number Plate Recognition using Machine Learning,” IJIRT, vol. 8, no. 12, pp. 1085-1088, 2022.

P. Mysore Jayakumar, T. K B, V. v and M. Madivalappa, “Detection of Non-Helmet Riders and Extraction of License Plate Number using YOLOv2 and OCR Method,” International Journal of Innovative Technology and Exploring Engineering, vol. 9, no. 2, pp. 2278-3075, December 2019.

Kulkarni, P. S, Sontakke and S. V, "Real-time helmet detection and recognition using deep learning techniques," Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 8, pp. 8775-8791, 2021.

Shanmugam, M, K and Somasundaram, "Automatic vehicle number plate recognition using deep learning," vol. 183, pp. 109-122, 2021.

Zhang, W, Chen, X, Cheng and Y, "A survey on deep learning-based object detection.," Neurocomputing, vol. 399, pp. 286-304, 2020.

Ghazali, R, Saim, N. F, N. S and Jufri, "Real-time vehicle detection and recognition system using YOLOv3 for smart city applications," Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 1, pp. 767-780, 2021.

Wu, Y, Chen, M, Li, G, Wang and Zhang, "A comprehensive survey on the application of deep learning in intelligent transportation systems," IEEE Access, vol. 9, pp. 32812-32837, 2021.

K. Dahiya, D. Singh and C. K. Mohan, "Automatic detection of bike riders without helmet using surveillance videos in real-time," Int. Joint Conf. Neural Networks, pp. 3046-3051, July 2016.

D. Singh, C. Vishnu and C. K. Mohan, "Visual big data analytics for traffic monitoring in smart city," IEEE Conf Machine Learning and Application, pp. 18-20, 2016.

C. Behera, R. Ravi, L. Sanjeev and D. T, "A comprehensive study of motorcycle fatalities in south Delhi," Journal of Indian Academy of Forensic Medicine, vol. 31, pp. 6-10, 2009.

. R. R.V.e Silva, K. R.T. Aires and R. de M. S. Veras, "Detection of helmets on motorcyclists.".

L. Allamki, M. Panchakshari, A. Sateesha and K. S Pratheek, "Helmet Detection using Machine Learning and Automatic License Plate Recognition".

. F. Wilhelm Sieberta, and H. Lin, "Detecting motorcycle helmet use with deep learning".

R. Roy, S. Kumar, P. Dumbhare and M. Barde, "Helmet Detection and Number Plate Recognition using Machine Learning," International Research Journal of Engineering and Technology, vol. 8, no. 5, pp. 3239-3243, 2021.