

Helmet Detection And Number Plate Recognition Using Deep Learning

Angoth Lakshmn¹, M. Madhu Vinay², P. Kalyani³, V. Deepthi⁴, M.Upendra⁵

¹Assistant professor, Dept of CSE-Data science

^{2, 3, 4, 5}Dept of CSE-Data science

^{1, 2, 3, 4, 5} CMR Technical Campus, Kandlakoya, Hyderabad, Telanganna, India.

Abstract- Every country has seen an increase in motorcycle accidents over the years due to social and economic differences as well as regional variations in transportation circumstances. One common mode of transportation for those in the middle class is a motorbike. One of the leading causes of road accidents that result in fatalities nowadays. Among them, motorcycling accidents are common and can cause severe injuries. A motorcycle rider's helmet is one of the most important components of safety. But many people choose to ignore the recommendation to wear a helmet. In current situation, we come across various problems in traffic regulations in India which can be solved with different ideas. Riding motorcycle/mopeds without wearing helmet is a traffic violation which has resulted in increase in number of accidents and deaths in India. Existing system monitors the traffic violations primarily through CCTV recordings, where the traffic police have to look into the frame where the traffic violation is happening, zoom into the license plate in case rider is not wearing helmet. But this requires lot of manpower and time as the traffic violations frequently and the number of people using motorcycles is increasing day-by-day. What if there is a system, which would automatically look for traffic violation of not wearing helmet while riding motorcycle/moped and if so, would automatically extract the vehicle's license plate number. Recent research have successfully done this work based on CNN, R-CNN, LBP, HoG, HaaR features, etc. But these works are limited with respect to efficiency, accuracy or the speed with which object detection and classification is done. In this research work, a Non-Helmet Rider detection system is built which attempts to satisfy the automation of detecting the traffic violation of not wearing helmet and extracting the vehicles' license plate number. The main principle involved is Object Detection using Deep Learning at three levels. The objects detected are person, motorcycle/moped at first level using YOLOv2, helmet at second level using YOLOv3, License plate at the last level using YOLOv2. Then the license plate registration number is extracted using OCR (Optical Character Recognition). All these techniques are subjected to predefined conditions and constraints, especially the license plate number extraction part. Since, this work takes video as its input, the speed of execution is crucial. We have used above said methodologies

to build a holistic system for both helmet detection and license plate number.

Keywords- YOLO (You Look Only Once), OCR, CNN, HoG, Haar Features

I. PROCEDURE FOR PAPER SUBMISSION

Proposed System with Features

In this project we are detecting whether two wheeler rider wearing helmet or not, if he is not wearing helmet then we are extracting number plate of that two wheeler. To extract number plate we have YOLO CNN model with some train and test images and if you want to add some other images then send those images to us so we can include those images in YOLO model with annotation to extract number plate of those new images. You Only Look Once (YOLO) Algorithm You Only Look Once (YOLO) proposes using an end-to-end neural network that makes predictions of bounding boxes and class probabilities all at once. It differs from the approach taken by previous object detection algorithms, which repurposed classifiers to perform detection. Following a fundamentally different approach to object detection, YOLO achieved state-of-the-art results, beating other real-time object detection algorithms by a large margin. While algorithms like Faster RCNN work by detecting possible regions of interest using the Region Proposal Network and then performing recognition on those regions separately, YOLO performs all of its predictions with the help of a single fully connected layer. Methods that use Region Proposal Networks perform multiple iterations for the same image, while YOLO gets away with a single iteration. Several new versions of the same model have been proposed since the initial release of YOLO in 2015, each building on and improving its predecessor. Here's a timeline showcasing YOLO's development in recent years. YOLO v2, also known as YOLO9000, was introduced in 2016 as an improvement over the original YOLO algorithm. It was designed to be faster and more accurate than YOLO and to be able to detect a wider range of object classes. 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.

Advantages

- Fast, good for real-time processing.
- Predictions are made from one single network.
- YOLO is more generalized.

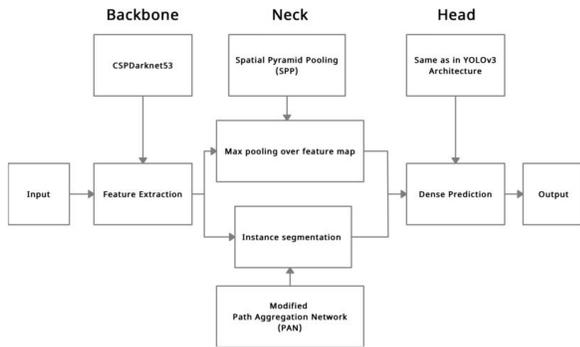


Figure Block diagram of Proposed System

Input And Output Design Input Design The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dial go to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

III. MODULES AND THEIR FUNCTIONALITIES

Data Preprocessing After collecting datasets from various resources. Dataset must be pre-processing before training to the model. The data pre-processing can be done by various stages, begins with reading the collected dataset the process continues to data cleaning. In data cleaning the datasets contain some redundant attributes, those attributes are not considering for phishing detection. So, we have to drop

unwanted attributes and datasets containing some missing values we need to drop these missing values in order to get better accuracy.

- Getting the dataset
- Importing libraries
- Importing datasets
- Splitting dataset in to training and test set
- Train the classifier
- Test the classifier
- Evaluate

b. Splitting the Dataset into the Training set and Test Set In machine learning data pre-processing, we divide our dataset into a training set and test set. This is one of the crucial steps of data pre-processing as by doing this, we can enhance the performance of our machine learning model. Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models. If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as: 11

1. **Training Set:** A subset of dataset to train the machine learning model, and we already know the output.
2. **Test Set:** A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.
3. **Train the Classifier:** Training the classifier with the training data by specifying the value of k . Use $k=3$ for binary classification, i.e., two labels classification. If used $k=1$ then it is simply a nearest neighbor classifier.
4. **Test the Classifier:** Testing the classifier with the testing data.
5. **Evaluate:** Evaluating the classifier using confusion matrix and its evaluation metrics i.e., accuracy, precision, recall, etc.

Functional Requirements

It provides the users a clear statement of the functions required for the system in order to solve the project information problem it contains a complete set of requirements for the applications. Functional requirements may involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. A requirement is condition that the application must meet for the customer to find the application satisfactory. A requirement has the following characteristics:

- It provides a benefit to the origination.
- It describes the capabilities the application must provide in business terms.
- It does not describe how the application provides that capability.
- It is stated in unambiguous words. Its meaning is clear and understandable.
- It is verifiable.

Non-Functional Requirements

Specifies the quality attribute of a software system. They judge the software system based on Responsiveness, Usability, Security, Portability and other non-functional standards that are critical to the success of the software system. Career recommendation non-functional requirements, like interests he has, how hours he can work likewise, with today’s IT projects, to determine non-functional requirements, like availability, the approach requires that the designer 1st determine the scope: does the whole solution or only part of it need to be architected 12 to meet minimum levels

This is done through 4 steps:

- Identify the critical areas of solutions
- Identify the critical components within each critical area.
- Determine each components availability and risk.
- Model worst-case failure scenarios.

IV. SYSTEM ARCHITECTURE

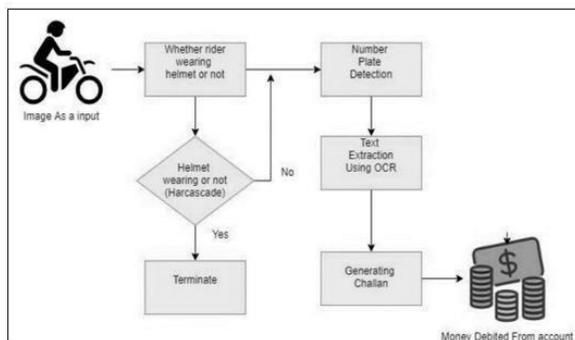


Figure: System Architecture

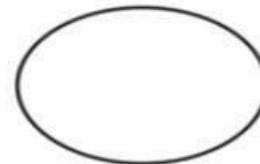
Here we have taken image of the bike racer as an input then it checks whether racer it wearing helmet or not. It checks the helmet wearing or not through the Haar cascade algorithm if yes then it terminate the process, and if not then it check the number plate through the text recognition using OCR technique then after the text recognition and number plate detection it generate the challan. receipt and then it debited the money from the bike owner. Dataflow Diagram. A Data Flow Diagram (DFD) is a traditional visual

representation of the information flows. within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It may be used as a communication tool between a system analyst and any person who plays a part in the order that acts as a starting point for re designing a system. The DFD is also called as a data flow graph or bubble chart. The Basic Notation used to create a DFD’s are as follows:

A. Dataflow: Data move in a specific direction from an origin to a destination.



B. Process: People, procedures, or devices that use or produce (Transform)Data. The physical component is not identified.



C. Source: External sources or destination of data, which may be People, programs.



D. Data Store: Here data are stored or reference by a process in the System

REFERENCES

- [1] Jaiswal R. C. and Fiza Shaikh, “ Augmented Reality based Car Manual System ”, Journal of Emerging Technologies and Innovative Research (JETIR), Open Access, Peer Reviewed and refereed Journal, Indexed in Google Scholar, Microsoft Academic, CiteSeerX, Thomson Reuters, Mendeley : reference manager, ISSN-2349-5162, Impact Factor:7.95, Volume Issue 5 pp. c326-c332, May 2023.
- [2] Tejveer Pratap, “ Multiparametric Monitoring of Vital Signs in Clinical and Home Settings for Patients ”, Journal of Emerging Technologies and Innovative Research (JETIR), Open Access, Peer Reviewed and refereed Journal, Indexed in Google Scholar, Microsoft Academic, CiteSeerX, Thomson Reuters, Mendeley : reference manager, ISSN-2349-5162, Impact Factor:7.95, Volume 9, Issue 5 pp. a701-a705, May 2022.
- [3] Shreyas Nazare, “ IoT Based Home Automation System”, Journal of Emerging Technologies and Innovative Research (JETIR), Open Access, Peer Reviewed and

- refereed Journal, ISSN-2349- 5162, Impact Factor:7.95, Volume 8, Issue 11 pp. 151-153, November 2022.
- [4] XINHUA JIANG “A Study of Low-resolution Safety Helmet Image Recognition Combining Statistical Features with Artificial Neural Network”.ISSN: 1473-804x, May 2021.
- [5] Maharsh Desai, Shubham Khandelwal, Lokneesh Singh, Prof. Shilpa Gite, “Automatic Helmet Detection on Public Roads”, International Journal of Engineering Trends and Technology (IJETT), Volume 35 Number 5- March 2021, ISSN: 2231-5381, September 2021
- [6] Thepnimit Marayatr, Pinit Kumhom, “Motorcyclist’s Helmet Wearing Detection Using Image Processing”, Advanced Materials Research Vol 931- 932,pp. 588-592,June 2020.
- [7] Romuere Silva, “Helmet Detection on Motorcyclists Using Image Descriptors and Classifiers”, 27th SIBGRAPI Conference on Graphics, Patterns and Images.IEEE, 2020.
- [8] Sharvari Doifode, “Virtual Assistant”, Journal of Emerging Technologies and Innovative Research (JETIR), Open Access, Peer Reviewed and refereed Journal, ISSN-2349-5162, Impact Factor:5.87, Volume 7, Issue 10, pp. 3527-3532, June 2020
- [9] Pravalika Kunda and Jaydeep Bhoite, “Home Renovation Using Augmented Reality”, Journal of Emerging Technologies and Innovative Research (JETIR), Open Access, Peer Reviewed and refereed Journal, ISSN-2349-5162, Impact Factor:5.87, Volume 7, Issue 6, pp. 682-686, October 2020.
- [10] Akshat and Kaushik, “Automated Attendance Monitoring system using discriminative Local Binary Histograms and PostgreSQL”, Journal of Emerging Technologies and Innovative Research (JETIR), Open Access, Journal, ISSN-2349-5162, Impact Factor:5.87, Volume 7, Issue 11, pp.M., et al. (2018). The Malicious Use of