International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 06 | Jun 2022 www.irjet.net p-ISSN: 2395-0072

Helmet Detection Based on Convolutional Neural Networks

Dr. M. Trupthi[1]

[1]*Chaitanya Bharathi Institute of Technology, Hyderabad*
Assistant Professor, Dept. of Information Technology, TS ,India

---

Abstract - *In the present situation, Riding motorcycle without wearing helmet is a traffic violation which has increased in the number of accidents and deaths in India. The 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 the rider is not wearing a helmet. But this requires a lot of manpower and time. What if a system automatically looks for a traffic violation of not wearing a helmet while riding a motorcycle? If so, would automatically extract the vehicles' license plate number. Recent research has successfully done this work based on CNN, R-CNN, LBP, HoG, HaaR features, etc. But these works are limited concerning 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 a 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 at first level using YOLOv2, helmet at a second level using YOLOv3, License plate at the last level using YOLOv2. Then the car place license number is extracted using OCR (Optical Character Recognition). All these techniques are subjected to predefined conditions and constraints, especially the car place number extraction part. Since this work takes video as its input, the speed of execution is crucial. A holistic system is built for both helmet detection and license plate number extraction.*

***Key Words*: YOLOv3, OCR, OpenCV, Deep learning, Helmet detection**

1. INTRODUCTION

With the advancement of technology, the rapid construction of high-quality roads is now easier. The availability of better road connections leads to an increase in the number of road vehicles required to ensure the safety of road users. Safety rules and regulations must be carefully monitored to reduce road accidents. Road accidents involving two wheels are severely damaged and the chances of survival for the people involved in these accidents are very low. There are several methods used to monitor the rules and safety of road users; some of which are the improved use of Computer Vision.

To avoid accidents of two-wheelers on the road, there must be real-time detection to check that the rider is wearing a helmet or not. Such a type of problem was addressed by developers for the construction sites. Jing Hu et al. 2019 used the YOLOv3 model to process real-time video feed and detect a person wearing a helmet or not at construction sites. The worker is recognized using YOLOv3 and the samples are made. These samples go through the YOLOv3 model to detect if they are wearing helmets or not. The model gives 93.5 mAP and 35fps. Further, Fan Wu et al. proposed an improvement to the YOLOv3 algorithm by using the feature extraction method. They used the feature extraction method as the backbone of the YOLOv3 model. This improvement gave the increased detection rate of 2.44%.

1.1. Motivation

Nowadays, usually many accidents occur. Many of them are due to not following some basic traffic rules. People may ride good but the other person may not. So, we must be safe during traveling from place to place. Mainly Motorcycle riders must wear a helmet. As we know there is a huge impact on image tuning and image detection. There are many ways to detect many things. Starting from OpenCV to YOLO-V5, we have several techniques to perform object detection. So, we came up with the idea of detecting Number Plates who don't wear helmets using Yolo-v4. 1.3. Objective of the Project This project proposes a method to automate the detection of traffic violators and to recognize the riders not wearing helmets using the CNN-based algorithm YOLOv3 which is a fast single-stage object detection algorithm. It is based on darknet architecture and trained on the MS COCO dataset capable of detecting 80 classes of objects which include persons, motorcycles, bicycles, cars, traffic lights, etc. This algorithm is used to find the two-wheeler riders in a frame. People are detected and checked if they are on a two-wheeler. The cropped bounding box of the detection is then passed on to the next phase. This is an image classifier based on the LeNet architecture trained to classify an image to have a helmet or not. If a rider is not wearing the helmet, then the instance is returned and displayed in a different color than the helmet-wearing riders.

1.2. Problem Statement

Efficient and accurate object detection has been an important topic in the advancement of computer vision systems. With the advent of deep learning techniques, the accuracy of object detection has increased drastically. The project aims to incorporate state-of-the-art techniques for object detection to achieve high accuracy with real-time performance. The main aim of Helmet Detection is to extract the number plates who don't wear helmets and charge them a fine.

2. PROPOSED SYSTEM

2.1 YOLO

2.1.1 Grid cells

The Image is divided into S x S grid cells. Each grid cell has n anchor boxes with its center in the grid cell. The value of S can be any integer but the most common grid cell values are 13 x 13 as shown in Fig. 1 and 19 x 19 for larger images with many objects to be detected.

Fig. -1: 13x13 Grid cells

Each anchor box of each grid cell can predict one object i.e., for a model which takes S x S grid cells and n anchor boxes. It can predict S*S*n objects in an image. The anchor boxes predict the objects in the image which have similar ratios as them. For example, we have 2 objects in an image (fig 2) a person and a car. Here the anchor box 1 will predict the person as it is in similar ratio and the anchor box 2 will predict the car in the image.

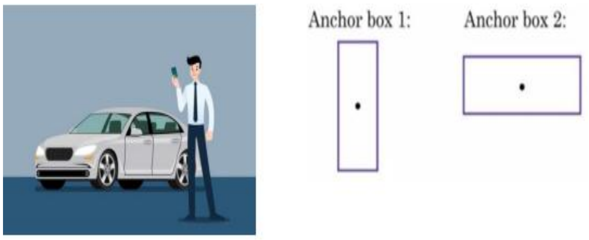

Fig. -2: Anchor Boxes

2.2 Network Output

Each bounding box has 5+ C output values. The 5 values are (x, y, w, h, confidence score). The x, y are the co-ordinates of the center of the bounding box. w, h are the width and height of the bounding boxes and confidence score is the value which tells how confident the model is that there is an object in that bounding box. C is the number of classes. The C values are the conditional class probabilities of the box. It is the probability that the detected object belongs to a specific class. To make the final prediction we use a threshold value. If the class confidence of the bounding box is greater than the threshold then that box is selected. Class confidence = confidence score x conditional class probability In few cases, If the object is large and multiple grid cells predict that the object's center is present in them, we get multiple over lapping bounding boxes. To solve this problem, we use Non-Max Suppression algorithm. We select the bounding box with the highestclass confidence and we check the IoU of that box with the over lapping boxes. We use a threshold and if the IoU is greater than the threshold we remove the Non-Max bounding box in the image.

2.3 YOLO Network Architecture

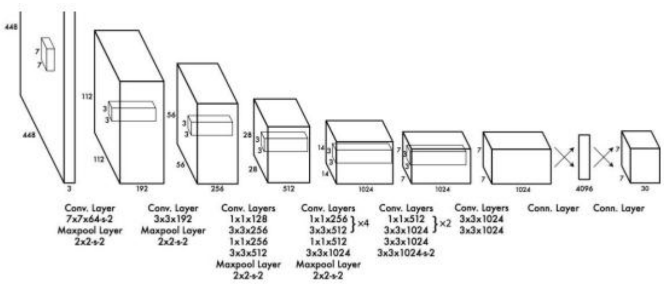

Fig. -3: YOLO Architecture

YOLO uses multiple Convolutional layers followed by fully connected layers. The output of the last layer has (S, S, B*(5+C)) dimensions. The architecture in the above Fig. 3 uses 7 x 7 grid cells and 2 anchor boxes per cell with 10 classes.

2.4 Loss Function

YOLO predicts many bounding boxes per grid cell. To compute the loss for the true positive, we only select the bounding box which is responsible for the object detection. We select the box with the highest IoU with the object. YOLO uses squared error between the predictions and the ground truth to calculate loss. The loss function composes of: The classification loss: This fixes the error in classification of the object. The localization loss: This fixes the position of the bounding box. The confidence loss: Improves the confidence about the detection of an object

Table -1: Layers in YOLOv3 Architecture

Layer description	Filter Size	Activation Function	Output size
Input			(n, 320, 320, 3)
Convolution	3 x 3	Leaky	(n, 320, 320, 32)
Max Pool	2 x 2		(n, 160, 160, 32)
Convolution	3 x 3	Leaky	(n, 160, 160, 64)
Max Pool	2 x 2		(n, 80, 80, 64)
Convolution	3 x 3	Leaky	(n, 80, 80, 128)
Convolution	1 x 1	Leaky	(n, 80, 80, 64)
Convolution	3 x 3	Leaky	(n, 80, 80, 128)
Max Pool	2 x 2		(n, 40, 40, 128)
Convolution	3 x 3	Leaky	(n, 40, 40, 256)
Convolution	1 x 1	Leaky	(n, 40, 40, 128)
Convolution	3 x 3	Leaky	(n, 40, 40, 256)
Max Pool	2 x 2		(n, 20, 20, 256)
Convolution	3 x 3	Leaky	(n, 20, 20, 512)
Convolution	1 x 1	Leaky	(n, 20, 20, 256)
Convolution	3 x 3	Leaky	(n, 20, 20, 512)
Convolution	1 x 1	Leaky	(n, 20, 20, 256)
Convolution	3 x 3	Leaky	(n, 20, 20, 512)
Max Pool	2 x 2		(n, 10, 10, 512)

2.5 SSD Architecture

To have more accurate detection, different layers of feature maps are also going through a small 3×3 convolution for object detection as shown in the below Fig. 4

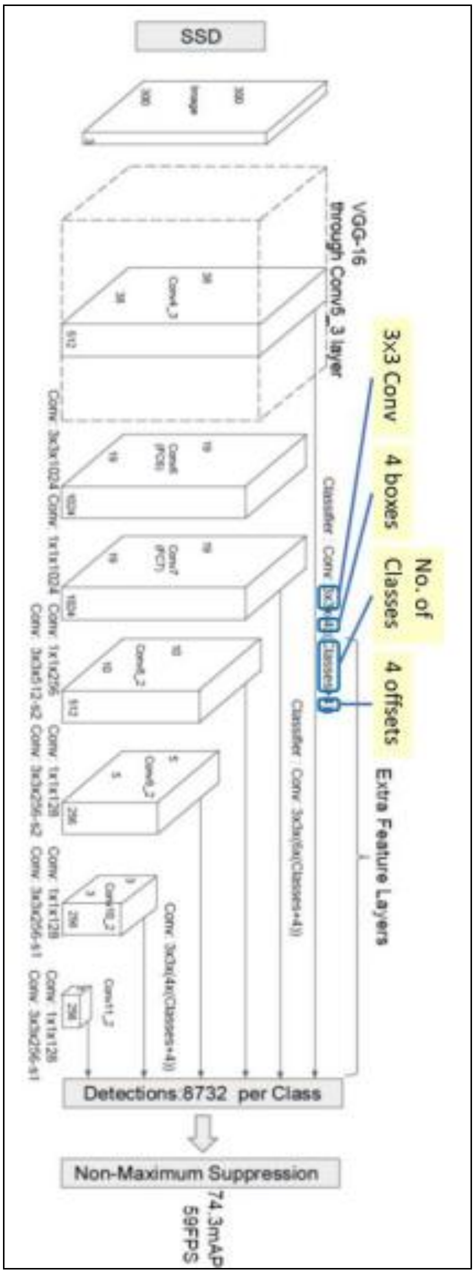

Fig. -4: SSD Architecture

2.6 SSD Loss Function

The loss function is the combination of classification loss and regression loss. The regression loss used here is Smooth-L1 loss, which is the same as Faster RCNN and Fast RCNN.

$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{x,y,w,h\}} \text{smooth}_{L_1}(t_i^u - v_i),$$

in which

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

2.7 Data Augmentation

Now you have your complete training data. There is only one step still missing: data augmentation. It can help algorithm learn the invariance of data. In fact, data augmentation plays an essential role in SSD.

3. IMPLEMENTATION

The work has been developed using YOLOv3 and classified based on LeNet architecture. Fig. 5.1 shows the block diagram for the proposed system. First, we apply the object detection algorithm YOLOv3 to obtain the two-wheeler riders. The bounding boxes obtained contain all the objects detected in the image belonging to the 80 classes of the MSCOCO dataset and it filters only the classes of persons and large vehicles. These bounding boxes are then cropped from the image and forwarded to the image classification algorithm. The image classifier uses the LeNet architecture and is trained to recognize the helmets from non-helmets.

3.1 Detection of Two-Wheeler Rider

The first step is to pre-process the input images. In this, first step the input image is taken through the system from the console with attribute and extracted the image dimensions. In the case of videos, the video frames are taken as an image for pre-processing. Then it is forwarded to the YOLO model for the next process. This is done by creating a blob (Binary Large Object) constructed from the input image. This is followed by non-maxima suppression with subtraction, normalizing, and channel swapping. After the successful creation of the blob, a forward pass-through YOLO model is performed. This is an important step used to remove redundant bounding boxes. This is a result of detecting the same object multiple times with varying confidence levels. Hence for accurate detection, non-maxima suppression is required.

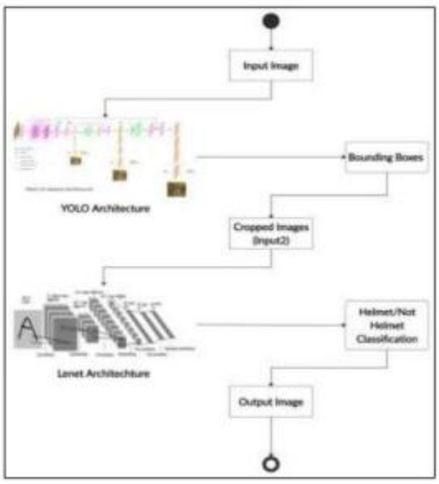

Fig. -5: Block Diagram of the Architecture for Proposed System

3.2 Localisation of Person

YOLOv3 algorithm detects all objects in the MS COCO dataset. However, the model requires only the persons riding two-wheelers. For this, large vehicles such as cars, buses, and trucks are detected and stored. The model then filters all the persons detected. It then ensures the persons are not in these large vehicles by checking the center coordinates of the person with the bounding box of the large vehicle. The cropped images of the persons found are then forwarded to the image classifier.

3.3 Helmet Vs Non-Helmet Classification

The obtained image contains the two-wheeler rider. The next step is to find, the rider is wearing a helmet or not. This is done using an image classifier trained to classify helmets and nonhelmet objects. The classifier uses the LeNet architecture. This architecture performs well for low resolution and small images as it was originally meant to classify handwritten letters. This model trained to classify helmets showed an accuracy of 95.2%. The input for the classifier is the cropped image. It detects the presence or absence of a helmet and returns the label found. If the rider is not wearing a helmet the bounding box for the rider in the image or video is shown in a different color.

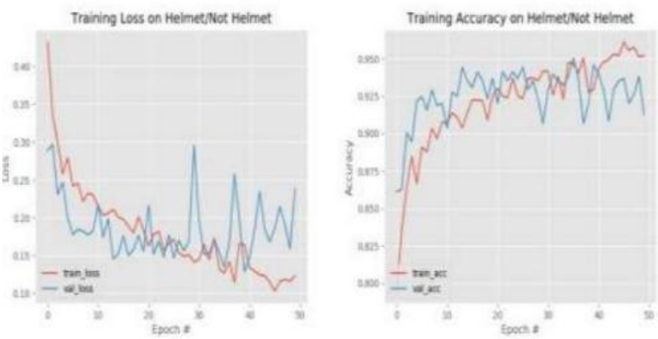

Fig. -6: The Average Loss and Accuracy Obtained During Training of Image Classifier

3.4 Detection of Registration plates

To perform this, we trained a set of 1000 images of registration plates. The loss was good at the end (0.01). But the problem as Number plates is very small in the big frame the recognition of these plates is not that accurate and here, we come with the drawback of the project. We were unable to detect most of the number plates. This work will be continued.

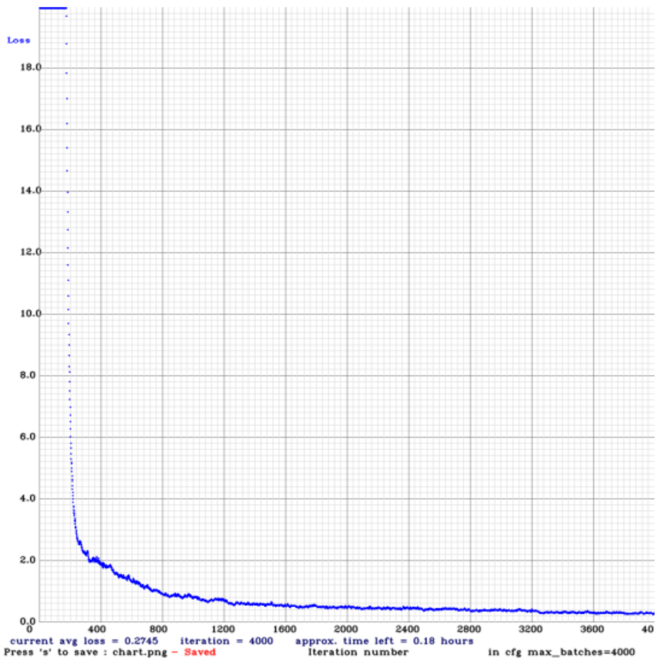

Fig. -7: Loss chart of helmet plate detection

4. RESULTS

The output of custom YOLO is the data of the bounding boxes and the one hot encoded array of class labels (assailant and hostage). The bounding box data obtained from the model is used to draw the boxes on the images and display the probability scores using Opencv3.

Fig. -8: Testing image of helmet-1

Fig. -9: Testing image of helmet-1

In order to calculate the map scores, we used 70 testing images and map score came out to be 48%.

4.1 Detection of Number Plates

It has not totally failed but it was predicting for some test images and some of the working and non-working test cases:

Fig. -10: Testing for Registration plate-1

5. CONCLUSION AND FUTURE SCOPE

We first started the whole thing by using a CNN classifier which just classifies them. As expected, that gave a very bad result and we can't use videos in that type of detection. Then next we tried it using MobileNetSSD that was giving a partially good result but it was very slow and not a GPU based. Finally, when we started with YOLO, we came to know that it is much faster than many of the detectors. Initially, the helmets were detectable like they will be in a good size in the frame so they were performing good. But as mentioned there is a slight problem with YOLO while detecting number plates.

5.1 Incorrect Predictions

There were some incorrect predictions for helmet too which was resolved totally by setting threshold to 0.7. Helmets were good but the problem is totally with registration plates as they are very small to detect in a large frame.

5.2 Accuracy

SSD has better performance than YOLO in all 3 cases i.e., small, medium and large images. But when only few labels are available yolo performed much faster.

5.3 Future Scope

Training the object detection algorithm to work on Vehicle registration plates and Helmets. The model can further be trained by using Pose estimation which helps us in taking care of a case where the person is actually riding a bike or not. We can also get the total number plate characters into a text form by using OCR (Optical Character Recognition) which is the electronic or mechanical conversion of images of typed, handwritten or printed text into machine-encoded text, whether from a scanned document, a photo of a document, a scene-photo or from subtitle text superimposed on an image.

REFERENCES

[1] Joseph Redmon; Ali Farhadi, "YOLO9000: Better, Faster, Stronger", 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

[2] Joseph Redmon, Ali Farhadi," YOLOv3: An Incremental Improvement",2018

[3] https://towardsdatascience.com/yolo-you-only-look-once

[4] IRJET-V8I4235.pdf - Mask and Helmet Detection in Two-Wheelers using YOLOv3 and Canny Edge Detection

[5] A. Adam, E. Rivlin, I. Shimshoni, and D. Reinitz, "Robust real-time unusual event detection using multiple fixedlocation monitors," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, no. 3, pp. 555–560, March 2008.

[6] C.-Y. Wen, S.-H. Chiu, J.-J. Liaw, and C.-P. Lu, "The safety helmet detection for atm's surveillance system via the modified hough transform," in IEEE 37th Annual International Carnahan Conference on Security Technology., 2003, pp. 364–369.

[7] C.-C. Chiu, M.-Y. Ku, and H.-T. Chen, "Motorcycle detection and tracking system with occlusion segmentation," in WIAMIS '07, USA, 2007

[8] A. Hirota, N. H. Tiep, L. Van Khanh, and N. Oka, Classifying Helmeted and Non-helmeted Motorcyclists. Cham: Springer International Publishing, 2017, pp. 81–86.

[9] Z. Y. Wang, Design and Implementation of Detection System of Warning Helmets Based on Intelligent Video Surveillance, Beijing University of Posts and Telecommunications, Beijing, China, 2018.

[10] A. H. M. Rubaiyat, T. T. Toma, M. Kalantari-Khandani et al., "Automatic detection of helmet uses for construction safety," in Proceedings of the 2016 IEEE ACM International Conference on Web Intelligence Workshops (WIW), ACM, Omaha, NE, USA, October 2016