ISSN 2347- 3657

International Journal of Information Technology & Computer Engineering

International Journal of Information Technology & Computer Engineering

ISSN 2347–3657

Volume 9, Issue 3, 2021

Bike Authentication by Helmet 'Using Faster RCNN Machine Learning'

SUMAYYA FATHIMA1, UDAYINI CHANDANA2

1PG Student, Dept. of ECE, Stanley College Of Engineering &Technology for Women, Hyderabad, Telangana, India, Email:fathimasumayya2710@gmail.com
2Assistant Professor, Dept. of ECE, Stanley College of Engineering &Technology for Women, Hyderabed, Telangana, India, Email: cudayini@stanley.edu.in

Abstract:

Two-wheeled bikes are the most common way to get around in developing countries like India. Even though riding is easy, many accidents happen because riders don't follow the law that says they have to wear helmets. In 2017, at least 98 riders who weren't wearing helmets died every day, according to the Road Accident Report. In 2016, 10,135 people died because they didn't wear a helmet. In 2017, that number rose to 36,000. WHO has said that one reason for the rise in road accidents is that riders don't use safety equipment as they should. The government has many ways to catch people who break the law. But these methods need help from a person, which slows down the system and makes it less reliable. In this paper, the Faster RCNN machine learning algorithm is used to make a system that automatically finds helmets. The camera in the system takes a picture of the rider and sends it to the raspberry pi so it can be processed. To find the helmet, a faster RCNN algorithm is used. If the helmet is not found, the vehicle can only go 40kmph, but it can go as fast as it wants if the helmet is found.

INDEX TERMS: Machine learning, Raspberry Pi, faster RCNN, finding objects, and finding helmets.

1. INTRODUCTION

95 percent of the road accidents that are reported involve two-wheelers, and 25 percent of the deaths are caused by riders who didn't wear a helmet while riding. The government has been taking steps to catch those who break the law, such as putting up video cameras in public places, etc. But the methods we use now are not automatic; they need help from people. There are also problems with how reliable these systems are and how they handle data in real time. Object detection methods add a huge amount of help to make these systems work automatically.

Object detection is a type of Computer Vision technology. It has to do with finding things that belong to a certain class, like faces, cars, buildings, etc. Object detection is used in many areas, like video surveillance, medical diagnosis, archaeology, and so on.
Object detection has two main steps: figuring out where the object is and what kind of thing it is. Object localization tells you where in the picture the object is, while object classification tells you what kind of object it is.
object goes with. Most traditional object detection methods use histograms of gradient and Haar-like features.
These traditional models for detecting objects work in three steps: selecting an informative region, extracting features, and classifying the objects. These are made by hand, and their architecture is not as detailed. When using these traditional models, there are some problems. For example, the sliding window strategy makes the models less accurate, and they are also less accurate at finding complex features of an object.

With the introduction of Regions with Convolutional neural networks (R-CNN), Deep Neural networks algorithms offer a way to solve the problems that traditional object detection methods have. The structure of these algorithms is very deep, and they can find complex features very accurately. Also, expressivity and robust training algorithms make it possible to learn useful representations of objects without having to manually design features.

For detecting helmets, the proposed system uses a Deep neural network – Faster RCNN model. Here, the rider has to type in a password to verify that the bike is his or hers. Once that is done, the helmet detection system starts. If the password doesn't match, the bike stays locked. Once the password matches, the camera attached to the Raspberry Pi takes a picture of the rider when he gets on the bike and sends it to the Raspberry Pi. The image is processed for helmet detection by a faster RCNN algorithm that is run on a Raspberry Pi. If the helmet is found, the motor doesn't have a speed limit (analogous to the engine). If the helmet is not found, the motor can only go as fast as 40kmph.
These are the main goals of the proposal:

• To make a system that can find helmets in real time, even if the object is moving or the lighting changes.

• To keep bike riders safer on the road by limiting the speed of the bike when the rider doesn't have a helmet on.
• To make the vehicle safer by installing a password lock system.

LITERATURE SURVEY

Athuljith MK and his team have made a "Intelligent System for Helmet Detection Using Raspberry Pi" to figure out how to set up an automatic system that would make helmets mandatory. This system makes sure that a motorcyclist is always wearing a helmet by taking a picture of the rider's helmet with a Pi Camera and confirming the presence of an object with the Haar cascading technique. The main goal of the project was to cut down on motorcycle deaths on the road. Intelligent System for Helmet Detection was made to keep riders safe on the road by giving them no choice. The vehicle's engine is run by a single-channel relay that only closes when the rider's helmet is found. The Haar features extraction model was used to find out if a helmet was on. But there are many situations where these assumptions aren't true because of changes in the view angle or lighting.

A small team at UC Berkeley led by Professor Jitendra Malik and made up of Ross Girshick, Jeff Donahue, and Trevor Darrel showed that a CNN is much better at detecting objects on PASCAL VOC than systems that use simpler HoG-like features [5]. R-CNN is the name of the model they came up with because it uses both CNN and region-based detection.
But R-CNN had some problems. For example, it needs a forward pass of the CNN (AlexNet) for every region proposal and every image, and it has to train three different models separately, which makes it very hard to train the pipeline.
Ross Girshick, who came up with R-CNN for the first time, solved both of these problems, which led to the second algorithm, Fast R-CNN[4]. ROI pooling was suggested as a way to fix the problem with the forward pass of CNN for each region. Then, the things in each area are put together (usually using max pooling). So we only have to look at the original picture once. Fast R-second CNN's idea is to train the CNN, the classifier, and the bounding box regressor all at the same time in a single model.

The first step in figuring out where things are is to make a bunch of possible "bounding boxes" or "regions of interest" to check. In Fast R-CNN, these proposals were made with a method called "Selective Search," which was found to be the slowest part of the whole process. Shaoqing Ren, Kaiming

International Journal of Information Technology & Computer Engineering
ISSN 2347–3657
Volume 9, Issue 3, 2021

He, Ross Girshick, and Jian Sun, all of Microsoft Research, came up with an architecture called Faster R-CNN that makes the region proposal step almost free. Comparing the accuracy and speed of different RCNN networks [1] shows that Faster RCNN is more accurate than other RCNN algorithms.

PROPOSED MODEL

The proposed system improves road safety by ensuring control over the speed of the vehicle, especially the one driven by a rider with no helmet.

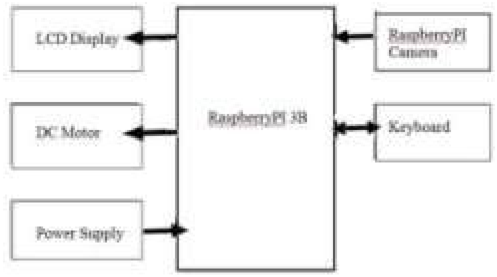


Fig. 3.1 Block Diagram of Helmet Detection System

The Helmet Detection system consists of the following interfaces-

- Raspberry Pi microcontroller is being used as the main controlling unit of the system.
- Raspberry Pi camera captures the screenshot of the rider and sends the data to the microcontroller for processing.
- KEYBOARD used in the system provides a user interface allowing the rider to enter the password and sends the data to the processor.
- LCD is used to display the warning commands if a helmet is not detected and also used for signing in into the system.
- DC MOTOR here acts as analogous to the bike engine.
- Power Supply Unit provides sufficient power for the proper working of the system.

When the rider mounts the bike, the open circuit under the seat TURNS ON the microcontroller, which thereby TURNS ON the pi camera and password module. Raspberry Pi has a password lock system. So, the rider has to log in using a password. The camera module takes the snapshot of the rider and sends the image to the microcontroller.

The rider has to enter the password through a keyboard. If password matches, Faster RCNN object detection algorithm checks for the presence of helmet. If the password does not match, no processing takes place and the motor is locked.

If the helmet is detected, the microcontroller turns on the DC motor, which is analogous to the engine of the bike, without applying any speed limit. If the helmet is not detected the LCD displays the warning message and turns on the motor with speed limit of 40kmph.

2. PRINCIPLE OF OPERATION

The goal is to make a helmet detection system that can run on a Raspberry Pi and uses the Faster RCNN object detection method.

R-CNN uses selective search to pull out a bunch of boxes from the image and then checks to see if any of these boxes have an object in them. In this case, the first step is to pull the regions out of the image. Next, CNN is used to pull features out of each region. R-CNN is slow because it has to go through a lot of steps to work.

Fast-RCNN doesn't send CNN the image's extracted regions. Instead, it sends CNN the whole image, which makes the Region of Interest (RoI). It uses a single model to extract features, sort them into

different classes, and return the bounding boxes. Fast-selective RCNN's search makes it slow and not very useful when used on bigger datasets. By replacing selective search with the Region Proposal Network, Faster R-CNN fixes the problem with selective search (RPN). First, ConvNet is used to pull out the feature maps from the input image. Then, these maps are sent through an RPN, which gives back object proposals. The maps are then put into groups, and the bounding boxes are guessed.
This plan is put into action in three steps. They're
• Making a custom dataset that can tell if someone is wearing a helmet or not.
• Getting the model ready.
• Making a hardware set with the object-finding algorithm Faster RCNN and testing the model on a new image.
A large number of images are needed to train a neural network on the custom dataset so that it can spot a helmet. In this case, the goal is to find a "person wearing a helmet." Images of the "person wearing the helmet" target are gathered. So that the prediction is correct, Pi camera takes 140 pictures of a person wearing the helmet. The train set has pictures of the target that move in different ways, like tilting, getting brighter or darker, etc.

The next step, after getting the pictures, is to give them names. In this project, labeling software is used to put labels on the target. In order to train a model, you need a fast CPU with a lot of memory. Jupyter Notebook can be used to train the model on Google Colab. As of October 13, 2018, Google Colab has one NVIDIA Tesla K80 GPU with 12GB of memory that can be used for up to 12 hours straight. In this project, the model is trained using the transfer learning method on a Windows 7 LENOVO computer with a 75GB CPU.

The custom dataset is used to train the model, and the weights are changed and saved. With these weights, predictions can be made about the new images. For the detection algorithm to work, these weights should be sent to the Raspberry Pi.

3. **RESULTS**

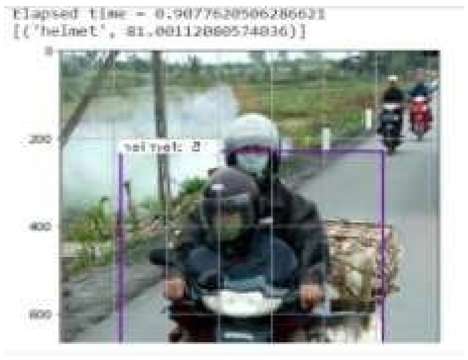
A dataset of "person wearing a helmet" is prepared with 140 images. Among these 140 images, 120 images are moved into the train_images directory, 5 are moved into the test_images directory and the remaining 15 are moved into a validation_images directory.

Some Modifications made on original paper are-

• The input image size used the original paper is 600. But in the project size is reduced to 300 for faster processing.
• The scale of the anchor is changed from (128, 154, 308) to (64, 128, 256).
• The ratio of the anchors is changed from (2:3, 3:2, 2:2) to (1:1, 2:3, 3:2).
• The number of classes to be detected is changed to 1("helmet").

ISSN 2347–3657

Volume 9, Issue 3, 2021



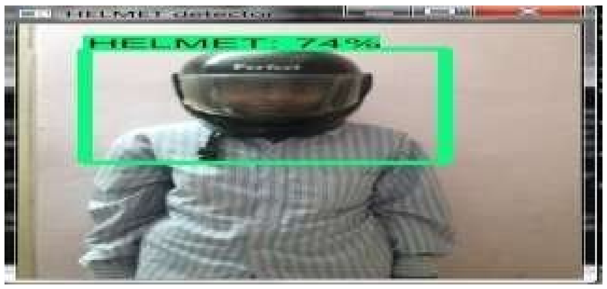
```
Elapsed time = 0.9077620506286621
[('helmet', 81.00112880574836)]
```

Fig 5.1 Result Showing Prediction on Test Image

The above image shows the bounding boxes generated on the test image present in the test images folder, along with their probability of detection. The Person wearing Helmet is detected with 81 percent accuracy. The time taken to detect the objects in this image is 0.9 sec. This test case is generated on Windows.

Here are the test cases that were made on Windows using real-time detection..

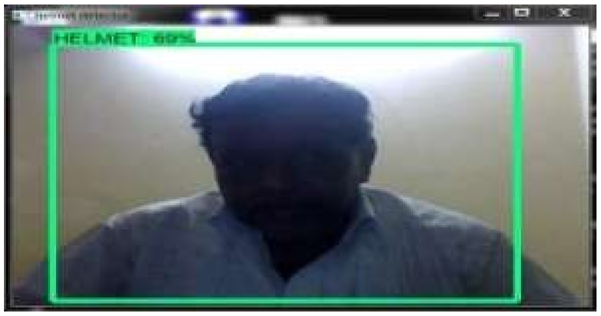
Case a: Person in standing position with HELMET



The person in the test case shown below [Fig. 5.2] is wearing a helmet, but he is standing up. So, this is not what should happen. This test case got a score of 74 percent for being able to be found.

Fig. 5.2: A person standing up with a helmet on

Case b: Person in a sitting position, not wearing HELMET



ISSN 2347–3657

Volume 9, Issue 3, 2021

picks up on the helmet, but the test case is not sitting like the rider. So, the above test case has a detection score of 78%.

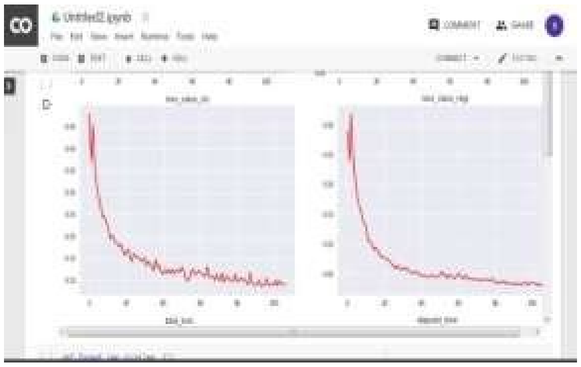
Fig.5.4: A person sitting down with a helmet on

Case d: Person in rider position, wearing HELMET

In the test case shown below [Fig. 5.5], the person is riding the bike while wearing the helmet. Here, the test case is sitting in a way that looks like a rider, and he is also wearing a helmet. This test case has a helmet detection score of 91%.

Fig.5.5 Person Riding a Bike and wearing Helmet

4. CONCLUSION



The loss functions for both the classifier layer and the regression layer are shown in Fig. 6.1. From the graph, we can see that the loss for the first set of images is high, but it goes down as the network learns.

Graph of Loss in Classifier and Regressor (Fig. 6.1)

In this project, the loss functions for the classifier layer are below 0.10, and the loss functions for the regression layer are below 0.05.

Based on the results of the experiment, it was clear that the size of the model needs to be kept in check in order for it to work on hardware devices. When putting object detection on mobile devices, it is important to choose the right algorithm and a pre-trained model that works well with it.

The size of the data used to train the model has a big impact on how accurate the model is. Choose the data so that it shows all of the different ways the object can change. The speed of the model on the CPU might be different from the speed on the Raspberry Pi. The main thing that limits how a Neural Network can be used on hardware is the size of the models and the speed of the device they are running on.

REFERENCES

1. Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu, "Object Detection with Deep Learning: A Review," IEEE Transactions on Neural Networks and Learning Systems (Early Access), DOI: 10.1109/TNNLS.2018.2876865, Page.no: 1–21, March 2017.

2. Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Transactions on Pattern Analysis & Machine Intelligence, Volume 39, Issue 6, Pages 1137-1149, June 2017.

ISSN 2347–3657

Volume 9, Issue 3, 2021

3. Athuljith MK, Biren Patel, Sourabh Pardeshi, Vivien Rajguru, and Nitin, "Intelligent System for Helmet Detection Using Raspberry Pi," International Journal for Scientific Research & Development (IJSART), Volume no. 3, Issue 6, ISSN [ONLINE]: 2395-1052, June 2017.

4. "Fast R-CNN," by Ross Girshick, IEEE International Conference on Computer Vision (ICCV),

Electronic ISSN: 2380-7504, February 2016, DOI: 10.1109/ICCV.2015.169.

5. Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," The IEEE Conference on Computer Vision and Pattern Recognition (CVPR),

ISSN: 1063-, DOI: 10.1109/CVPR.2014.81

6919, September, 2014.