

Automatic Helmet Detection and Gate Opening using YOLOv3

Vishal Parikh¹, Malaram Kumhar², Nistha Wagh³, Nidhi Sutariya⁴ and Arham Shah⁵

¹⁻⁵Institute of Technology, Nirma University, Ahmedabad, Gujarat, India

Email: {vishalparikh, malaram.kumhar, 17bce136, 17bit058, 18bce351}@nirmauni.ac.in

Abstract—With an increase in the population and improving lifestyle, the demand for transport has increased tremendously. An increase in the number of vehicles increases the risk of accidents. Automatic helmet detection plays an important role in preventing people from breaking the rule of “Helmet Wearing”. Many companies/institutions have, therefore, imposed a rule which says “No entry without helmet”. In this paper, an approach is proposed for automatic gate opening for two-wheeler riders wearing helmets. In this proposed approach two-wheeler riders from surveillance videos are detected and then it determines whether the two-wheeler rider is wearing a helmet or not. Transfer learning has been implemented using Yolo (You only look once) to detect the two-wheeler riders with and without helmets in real-time with the help of surveillance cameras to determine whether the gate needs to be opened or not.

Index Terms— helmet detection, gate opening, surveillance, and real-time helmet detection.

I. INTRODUCTION

One of the most popular mode of transportation in most places of the world is two-wheeler. This involves high risk as people don't take much precautions while riding. Helmets are highly desirable for bike-riders in order to reduce the involved risk. Because of the importance of helmets, governments have made it punishable offense to ride a two-wheeler without one. So, on an institutional as well as industrial level, to spread awareness regarding the importance of wearing a helmet, we propose an approach of automatic helmet detection and gate opening which opens the gate when the bike-rider is detected with helmet and keeps it close if detected without helmet. Existing video surveillance technologies, on the other hand, are passive and rely heavily on human intervention. In general, such systems are impractical because they involve humans, whose performance degrades over time.

A. Motivation

Helmet detection automation is extremely desirable for reliable and robust monitoring, as well as for reducing the quantity of human resources required. A coin has two sides i.e. along with benefits of less human resources stands other challenges in order to be addressed and resolved to get maximum benefits of the automation of human tasks. Following are the some of the challenges encountered when implementation is carried out:

- 1) Real-time Implementation: It's difficult to process a large volume of data in a short length of time. As a result, tasks including segmentation, feature extraction, classification, and tracking are involved, in which a large amount of data must be handled in a short amount of time in order to meet the goal of real-time deployment.
- 2) Direction of Motion: In general, three-dimensional objects have different view from different angles. Consider the appearance of a bike rider from the front and side views.

3) Quality of Video Feed: Generally, CCTV cameras capture low resolution video. Also, conditions such as low light, bad weather further complicates the situation.

Tasks like segmentation, classification, and tracking become considerably more challenging as a result of these limitations. Real-time performance, fine tuning, robustness to abrupt changes, and predictiveness are all desirable characteristics in a surveillance framework.

B. Organization of a Paper

We present a system for automatically detecting two-wheeler riders without helmets utilising feed from current security cameras that operates in real time, keeping these challenges and desirable qualities in mind. The following is how the rest of the paper is organised: Section 2 examines the related work, highlighting its merits and weaknesses. Section 3 outlines the proposed strategy. Section 4 contains all of the experimental specifics, as well as the results and their interpretation. The final section of the article summarises our future plans and concludes with closing remarks.

II. RELATED WORK

Anomaly detection in surveillance videos includes the automatic detection of two-wheeler riders without helmets. Dahiya et al. in [1] shows that real-time detection of bike riders without helmets works in two phases. The first phase involved detecting a bike rider in the video frame, while the second phase involved locating the bike rider's head and determining whether or not the rider was wearing a helmet. For the phase one, feature extraction techniques like Histogram of Oriented Gradients (HOG), Scale Invariant Feature Transform (SIFT) and Local Binary Patterns (LBP) were used out of which HOG was proven to be the most efficient technique. To classify bike-riders vs others, SVM classifiers was used. The same techniques were used in the second phase of detecting the helmet. The detection of bike-riders and violators was shown to be 98.88 percent and 93.80 percent accurate, respectively, in the experiments. The average time to process a frame is 11.58 milliseconds.

As explained by Simon et al. in [2], their work first deals with the problem of detecting the helmet. Before the segmentation and classification, Pre-processing is used to smooth the input image. Further the problem was splitted in two steps. The first step consisted of segmenting and classifying the vehicle images. After this, the system classified them into motorcycles and non-motorcycle. The process for detecting helmets was the second phase. A hybrid descriptor was used to extract image characteristics, and an SVM classifier was used to classify an image as helmet or non-helmet.

Manoch et al. In [3] used ML approach to detect the helmet. Vishnu et al. In [4] used ADE to classify the moving objects as motorcyclists or non-motorcyclists with the use of CNN. Thereafter, RoI is the upper one-fourth portion and detection of the helmet is done using CNN. Hand engineered feature extraction methods like HOG, SIFT, LBP don't work efficiently on low pixel images. Hence, CNN was used for extraction of discriminative features also. This deals with occlusion, illumination, poor quality of video and varying weather conditions. Features of helmet and head contain less deviation hence may fail to detect helmet correctly. Sparse traffic gives approximately 99% accuracy while in dense traffic, it gives approximately 91.5% in classification of motorcyclists and in classification of helmet, in dense traffic approx. 87% while in sparse traffic 98.6% accuracy is achieved. Multiple-riders are not taken into concern.

Dasgupta et al. in [5], used YOLOv3 to detect the motorcyclists and motorcycles, and co-ordinates of the bounding box covering the motorcycles and humans were noted. Depending on the coordinates of the bounding box of motorcycles, the humans belonging to that motorcycle were found. Upper one-fourth of the human images were taken and CNN was used to detect the helmet. It dealt with helmet detection for multiple riders, high speed vehicles, different shapes, colors, sizes of bikes. The major challenges faced were dense traffic, angle of camera, different shapes, colors, sizes of bikes.

In [6], Raj et al. used a linear SVM for histograms of oriented gradient (HOG) feature vector classification to detect motorcyclists. The expected region containing the head was passed to CNN for classification of helmet/no-helmet.

As mentioned by Boonsirisumpun et al. in [7], four CNN models namely VGG16[11], VGG19[12], Inception V3[13] and MobileNets[14] were used along with SSD and combination of SSD[10] with MobileNets gave highest accuracy. Training and testing data were prepared manually by cropping the images from 50 videos they had as an input dataset. MobileNets (85.19 %) and Inception V3 (84.58%) performed better in the classification phase than VGG16 (78.09%) and VGG19 (78.09%) (79.11%). MobileNets was the most accurate in the detection step, detecting 421 (85.40 %) proper bikers' classes from a total of 493 video pictures. Inception V3 came up second, with only 416 (84.38 %) accurate images spotted.

Parikh et al. in [16] used YOLOv2 for object detection. They detected the human object and processed based on their detection. Parikh et al. in [15] shows the camera placement strategy where the maximum coverage of the objects can be found. Parikh et al. in [17] discusses various keyframe extraction methods based on the detected object.

III. PROPOSED APPROACH

A. Dataset Preparation

Unlike a car which cannot move without fuel, the deep learning models don't work well without a sufficiently labelled dataset. For helmet detection, we need to detect motorcycles and distinguish between motorcycle, person and helmet which needs a good and properly labelled dataset. The universally used dataset 'COCO' unfortunately doesn't have a helmet as a class among 80 of its classes.

Google's Open Images Dataset v4 is the world's largest dataset containing object location annotations, with over 9 million images for 600 item types annotated with image-level labels and object bounding boxes.

Vittorio Mazzia prepared a project named 'OIDv4_ToolKit' in which one can download any number of particular classes from the OIDv4 dataset.

Using this, we downloaded 600 images per class i.e., 600 images containing humans, 600 images containing bikes and 600 images containing biker's helmet and sport's helmet as these were the only two categories present in the particular dataset.

B. Model Training

We used pre-trained weights of YOLO using transfer learning for training the model. A stored network that was previously trained on a large dataset, generally on a large-scale image-classification task, is known as a pre-trained model. Transfer learning for image classification is based on the idea that if a model is trained on a large and general enough dataset, it may successfully serve as a generic model of the visual world. The ability to use these learnt feature maps instead of starting from scratch by training a large model on a large dataset is a significant benefit.

C. Helmet and Two-wheeler Detection

We have used YOLOv3 [9] for motorcycle detection as well as helmet detection using laptop cameras.

YOLO v3 is based on a Darknet variation with a 53-layer network trained on Imagenet. For detection, 53 more layers are put on top of it, giving YOLO v3 a 106-layer fully convolutional underlying architecture.

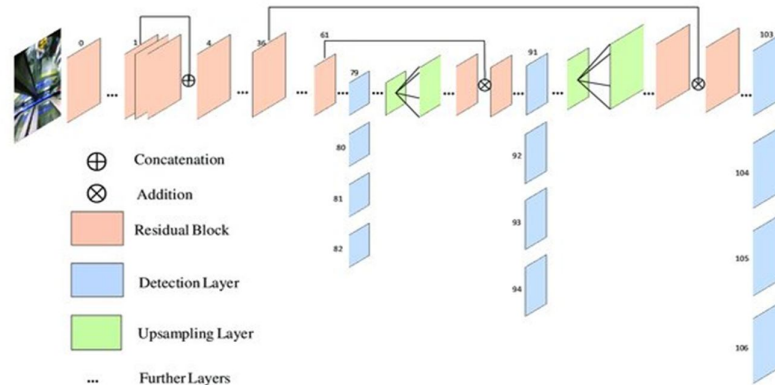


Fig. 1. YOLOv3 Network Architecture (Source: <https://ieeexplore.ieee.org/document/9026893>)

The most notable feature of version 3 is that it detects at three distinct scales. Application of 1×1 detection kernels to feature maps of three different sizes at three distinct locations in the network is used to identify the network. Large things are detected by the 13×13 layer, whereas tiny objects are detected by the 52×52 layer, and medium objects are detected by the 26×26 layer. At three distinct scales, YOLOv3 predicts boxes. The number of expected boxes for the same image of 416×416 is 10,647. This means that YOLOv3 predicts 10 times as many boxes than YOLOv2 [8]. In YOLOv3, logistic regression is used to determine object confidence and class predictions. Therefore, YOLOv3 is one of the most popular real-time object detectors in Computer Vision.

Various algorithms were implemented for object detection but found it to be one of the best algorithms in case of surveillance applications. Good results are obtained for two-wheelers and person but were not getting good results for helmets. Therefore, we came up with the idea of cropping the image to make the helmet cover more number pixels.



Fig. 1. (a) The original image containing detection as well as cropping box (b) the cropped image

The Fig. 2 shows the cropping of the captured frame. The black box shown is the detection box; i.e., the box where the bike is detected and the yellow box is the cropping box where the area of the box with the person is cropped for detecting the helmet. The original image is cropped in such a way that the left and right bounding edges are same as that of motorcycle and for top we divided the portion from top to top of bounding box of bike into half and for bottom we divided the portion from top of bounding box of bike to bottom of bounding box of bike into half. On the basis of our experience, we have seen detectors detecting bikes in two ways: either it detects only the bike or it detects a bike with a person as a bike. To avoid the mistake, we took half of upper portion as in case the detector detects only the bike, there remains high chance of person's head getting into our cropped image and in case of bike detected as bike plus person we took half of bottom portion such that it avoids tyres and get the image zoomed as much as possible.

To deal with multiple motorcycles in a frame, we decided to detect the helmet of the rider riding the motorcycle having the highest confidence score to determine whether to open the gate or not.

D. Gate Opening

IoT components required for the implementation are; an ultrasonic sensor, Arduino Uno. The Arduino Uno is a microcontroller board that uses the ATmega328P microcontroller. It comes with everything that is needed to get started with the microcontroller. To turn it on, simply connect it to a computer through USB or power it with an AC-to-DC adapter or battery. Despite the fact that Arduino comes with its own programming language, it may also be used with Python or another high-level programming language. In fact, Python and platforms like Arduino work well together, especially for applications that need sensor and other physical device integration.

IV. EXPERIMENTAL RESULTS

Experimental results of our model were quite good with an accuracy of over 87% approximately.

The model was trained and tested on a GPU computer having Intel core i5 and 7th generation with graphics card of NVIDIA GEFORCE GTX. CPU size was noted to be 16GB RAM. Further model details are, python language was used with YOLOv3 as a computer vision library and neural network framework with tensorflow and keras and other libraries.

For training, around 150 epochs were carried out which took time of around 8hrs on GPU-enabled platform. For helmet detection, we tested our model on the roadway so that we can know whether we have good results and so we tested from 3 profiles that are - right side way, left side way and front. Initial testing took 1-hour approximately where various test cases were kept in mind such as brightness of the day, number of vehicles, helmet wearers, type of vehicles, etc. Most of the motorcycles were identified by the model with a precision of 0.8 approximately.

Our model was quite successful in identifying various types of helmets with good precision. Due to the cropping algorithm, results are good in detection of person, helmet and motorcycle at the same time. As cropping made images zoomed which ultimately made the helmet cover a greater number of pixels making the detector easier to detect the helmet. Dataset was created by taking images from the OIdv4 as mentioned earlier and was trained on pre-trained weights of YOLOv3.

Thereafter, we needed to open the gate whenever the rider had worn a helmet which introduced IoT in this project. We required IoT components such as; an ultrasonic sensor, Arduino Uno and a webcam.

We have used an ultrasonic sensor on the gate so that when a vehicle passes through the range of the sensor, the camera starts and starts to detect images. If the bike is found, that particular frame is extracted and cropped. Then the helmet is checked in the cropped part of the frame and if the helmet is found, the gate opens, else it

remains closed. This algorithm is trained only on motorbikes but it still works on other two-wheelers to a great extent. Following is the flowchart of working of our proposed model:

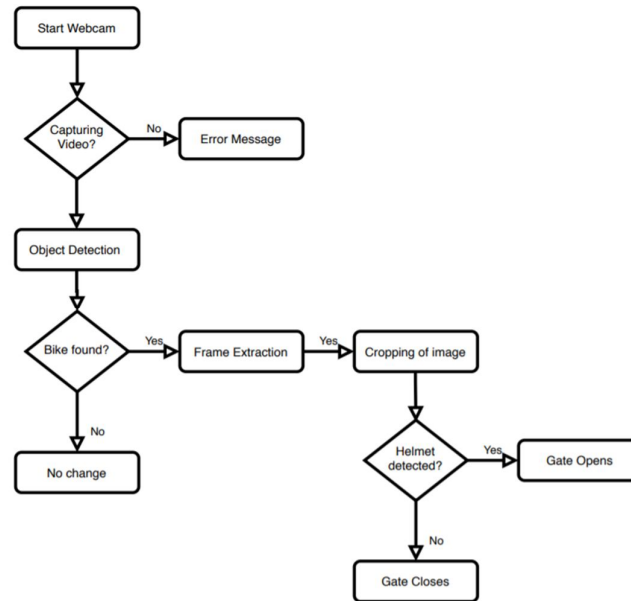


Fig. 2. Working of the proposed model

We performed a series of experiments to find the fitness of the model and in order to improve it. This was carried out by changing the number of training epochs, modification in algorithm and changes in some of the hyper parameters. As mentioned, this model is using the YOLOv3 framework which is considered for giving good results in object detection.

Following are the experimental results for helmet/no helmet classification:

Stage 1: We started from a basic model and then increased the model complexity by implementing a cropping facility to detect helmets efficiently and also changed some of the hyper-parameters. Input size of the image was taken as 416*416.

Helmets were detected but with low precision and sometimes it failed to detect.

Stage 2: We changed the hyper parameters - batch size and learning rate which made the results better with added cropping facility. We tried various hyper parameter values with learning rates of 0.01 and 0.01. Also changed batch size from 16 to 32. So, we tested the model in a medium light area, and it was noticed that the helmet could be detected with a precision score of 94%. These were good results as compared to other models. We tried testing of black and grey helmet for the initial phase.

Stage 3: The last phase testing included testing of various helmets with other objects on a busy road and various profiles or angles were considered. Due to good training, we could observe perfect detection of helmets. We add the functionality of detecting the helmet of the only person riding the motorcycle having the highest confidence score. In this phase, we took into consideration, left, right, back and front view in real time video surveillance via laptop webcam. As due to lockdown, we lacked devices, we used laptop cameras instead of CCTV cameras. Results were good and are shown in the images below.

Testcase1: The below Fig. 4(a) shows the successful detection of the bike and helmet from the left profile angle.

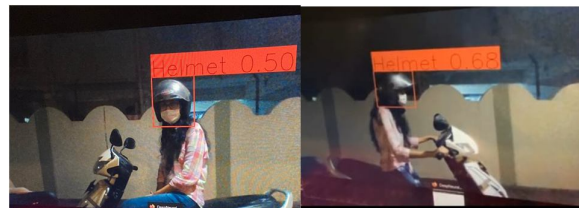


Fig. 4 (a) The detection from left profile and (b) from the right profile.

Testcase2: The above Fig. 4(b) shows the successful detection of the bike and helmet from the right profile angle.

Testcase3: The below Fig. 5(a) shows the successful detection of the bike and helmet from the front profile angle.



Fig. 5 (a) The detection from left profile and (b) from slightly right profile.

Testcase4: The above Fig. 5(b) shows the successful detection of the bike and helmet from the slightly right profile angle.

Testcase5: The below Fig. 6(a) shows the successful detection of the bike and helmet from the slightly right profile angle on a crowded road.

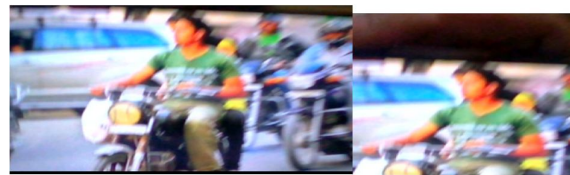


Fig. 6 (a) The detection from slightly left profile and (b) the cropped image.

Testcase6: The above Fig. 6(b) shows the successfully cropped image of the person without helmet.

Testcase7: The below Fig. 7 shows the successful detection of the bike and helmet from the front profile angle.

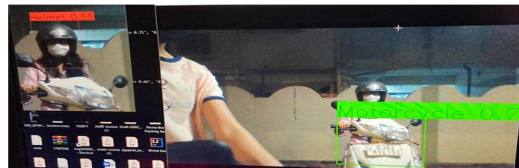


Fig. 7 Detection from the front profile

V. CHALLENGES

Our aim was to overcome the limitations faced while using YOLO v3 for object detection. Some of the limitations using YOLO v3 include medium-level accuracy, structure and size of helmet, shadows, light, orientation of the vehicles and many more. We were able to overcome many of these but still there are some problems.

1. As in the darkness, the maximum number of pixels get distorted, the detector fails to detect efficiently.
2. Secondly, the model faces difficulty in recognizing helmet sometimes as it is difficult to distinguish between the features of helmet and human head.
3. As the industry is progressing rapidly, the designs of helmets and motorcycles are evolving over the time. Due to insufficient experience with these new designs, the projects fail to detect them occasionally.
4. As our dataset is provided by Google, it doesn't include many images of vehicles like Activa which is most preferred two-wheeler in India. Therefore, our model occasionally fails to detect two-wheeler.

VI. CONCLUSION AND FUTURE WORK

Helmet Detection is one of the most important measure for the safety of the bikers. Our model performs pretty well in case of helmet detection which enables it to efficiently open the gate when the rider is wearing the helmet. We used YOLOv3 as it works better in surveillance applications and our results prove that.

There still remains challenges which needs to be faced and resolved and install this project in a real-life application such as toll area, institution gate entry etc. Our future work will be based on the mentions. It will be the installation of this into a proper real-life system and development of other systems such as number-plate classification, and seat-belt classification will be carried out. Apart from increasing the accuracy, we would also try to develop a project which can be fitted into the motorbike so that it wouldn't start in absence of a helmet. This would definitely increase the cost of the vehicle but safety wouldn't be compromised.

REFERENCES

- [1] Dahiya, K., Singh, D., & Mohan, C. K. (2016, July). Automatic detection of bike-riders without helmet using surveillance videos in real-time. In 2016 International Joint Conference on Neural Networks (IJCNN) (pp. 3046-3051). IEEE.
- [2] Stemy Simon, Divya Kumaran A.K. DETECTION OF MOTORCYCLISTS WITHOUT HELMET AND FINEPAYMENT USING OpenCV. In 2018 National Conference on Advanced Innovations in VLSI, Embedded Systems, Signal Processing & Communications
- [3] Manocha, D., Purkayastha, A., Chachra, Y., Rastogi, N., & Goel, V. (2019, March). Helmet Detection Using ML & IoT. In 2019 International Conference on Signal Processing and Communication (ICSC) (pp. 355-358). IEEE.
- [4] Vishnu, C., Singh, D., Mohan, C. K., & Babu, S. (2017, May). Detection of motorcyclists without helmet in videos using convolutional neural network. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 3036-3041). IEEE.
- [5] Dasgupta, M., Bandyopadhyay, O., & Chatterji, S. (2019, December). Automated helmet detection for multiple motorcycle riders using cnn. In 2019 IEEE Conference on Information and Communication Technology (pp. 1-4). IEEE.
- [6] Raj, K. D., Chairat, A., Tintong, V., Dailey, M. N., & Ekpanyapong, M. (2018, January). Helmet violation processing using deep learning. In 2018 International Workshop on Advanced Image Technology (IWAIT) (pp. 1-4). IEEE.
- [7] Boonsirisumpun, N., Puarungroj, W., & Wairotchanaphuttha, P. (2018, November). Automatic detector for bikers with no helmet using deep learning. In 2018 22nd International Computer Science and Engineering Conference (ICSEC) (pp. 1-4). IEEE.
- [8] Redmon, J., & Farhadi, A. (2018). Yolo3: An incremental improvement. arXiv preprint arXiv:1804.02767.
- [9] Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).
- [10] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016, October). Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.
- [11] Theckedath, D., & Sedamkar, R. R. (2020). Detecting affect states using VGG16, ResNet50 and SE-ResNet50 networks. SN Computer Science, 1(2), 1-7.
- [12] Carvalho, T., De Rezende, E. R., Alves, M. T., Balieiro, F. K., & Sovat, R. B. (2017, December). Exposing computer generated images by eye's region classification via transfer learning of VGG19 CNN. In 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 866-870). IEEE.
- [13] Xia, X., Xu, C., & Nan, B. (2017, June). Inception-v3 for flower classification. In 2017 2nd International Conference on Image, Vision and Computing (ICIVC) (pp. 783-787). IEEE.
- [14] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
- [15] Parikh, V., Sharma, P., Shah, V., & Ukani, V. (2018, February). Optimal Camera Placement for Multimodal Video Summarization. In International Conference on Futuristic Trends in Network and Communication Technologies (pp. 123-134). Springer, Singapore.
- [16] Parikh, Vishal, Priyanka Sharma. Intelligent Video Analytic Based Framework for Multi-view Video Summarization. In: International Journal Of Computing and Digital System.
- [17] Vishal Parikh, Jay Mehta, Saumyaa Shah, Priyanka Sharma Comparative Analysis of Key-frame Extraction Techniques for Video Summarization. In Recent Advances in Computer Science and Communications (pp. 2753-2763), Volume 14, Bentham Science.