Detection of Two-Wheeler Riders Without Helmets Using Yolov3 and Image Classifier

T. RADHIKA[1], S. TARUN KUMAR[2], EDE RISHI KIRAN[3], A. RAHUL REDDY[4], C. SAMARA SIMHA REDDY[5]

[1, 2, 3, 4, 5] *Dept. of Information Technology, TKR College of Engineering and Technology, Hyderabad, Telangana*

Abstract— In the current scenario of the traffic regulations and other measures for the safety on roads and work places, the use of safety equipment are essential parts for precautionary measures from the safety point of view. In the present paper, a technique has been developed for the safety measures to maintain the utmost precaution on road traffic or any other places where the helmets are mandatory to use. In this current paper, the algorithms You Only Look Once version 3 (YOLOv3) and LeNet are used for development of a system that can be used for Helmet detection and recognition of two-wheeler riders. The developed technique is able to detect any type of Helmet and it is tested on several cases. In-case, the rider is found not wearing a helmet, the instance is noted and the results are prepared on the basis of data obtained. The model shows an accuracy of 74% and reaches a speed of 1 FPS using CPU.

Indexed Terms— YOLOv3, Helmet detection, Bagging and Boosting method

I. INTRODUCTION

With the advancement in the technology, rapid construction of the quality roads is now easier. The availability of better road connections lead to increase in number of vehicles on the road for which it becomes necessary to ensure safety of road users. The safety rules and regulations to be followed needs to be properly checked in order to reduce road accidents. Road accidents involving two wheelers suffer the maximum damage and chances of survival for the people involved in these accidents are very low. There are several methods that are in use to check the safety rules and regulations to be followed by the road users; some of them is the advanced application of the Computer Vision. Girshick et al. in their paper introduced R-CNN as an algorithm that is simple and scalable and the detection increases mean average precision (mAP) by more than 30% as compared with methods that were previously used. There is an increase in the number of developers using deep learning to detect objects. In 2017, Girshick et al. further improved Fast R-CNN by introducing Faster R-CNN (2017). They proposed a method using Region Proposal Network (RPN) which combines convolutional features with the detection network, providing extremely low cost region proposals. This model overcomes the bottleneck of region proposal computation.

II. RELATED WORKS

Related Work Girshick et al. in their paper introduce R-CNN as an algorithm that is simple and scalable and the detection increases mean average precision (mAP) by more than 30% as compared with methods that were previously used. These were complex systems which typically combined many low-level image features with high-level context. R-CNN yields a significant performance boost as compared to OverFeat. It is also highly effective for vision problems with small datasets. However, the model is expensive to train and the mean average precision has scope for improvement.

The paper “Fast R-CNN” by Ross Girshick proposed using Fast R-CNN for object detection. This was more precise and built on previous models of R-CNN and SPPnet. Feature map is generated from convolution operation once per image. Fast R-CNN combines the precision of R-CNN family algorithms with improved speed. Object detection with it was still slow with up to 5 frames per second and expensive to train.

Girshick et al. further improve upon Fast R-CNN introducing Faster RCNN(2017). They proposed a method using Region Proposal Network (RPN) which combines convolutional features with the detection network, providing extremely low cost region proposals. This model overcomes the bottleneck of region proposal computation. It also improves accuracy up to 73.2% mAP in the PASCAL VOC 2007 dataset. It has a small frame rate on GPU of 5fps which is expensive considering the real time detection. In their paper "SSD: Single Shot MultiBox Detector" authors Wei Liu et al proposed using SSD or Single Shot Detector for object detection which uses a single deep neural network. It is a fast and accurate object detection algorithm which shows good performance in real time scenarios. For small scale objects, SSD performs worse than Faster R-CNN. The detection of small objects is only possible in high resolution images. However, they consist of low-level features such as color patches or edges. It is less informative for classification. It also runs at 22 FSP which is low compared to other algorithms

Redmon et al. in the paper "You Only Look Once: Unified, Real-Time Object Detection" introduces the YOLO algorithm that has proven to be better and faster for detection of objects. YOLO treats detection as a simple regression problem taking an image as input and learning probabilities of the class and coordinates of the bounding boxes. The image processing of their YOLO model is at the rate of 45 fps and of Fast YOLO version is 155 fps which is almost twice as much as other methods and algorithms available. The authors also improved upon the original algorithm in YOLO9000 (YOLOv2) and YOLOv3[8] which has been able to achieve a large number of object detections. It is very fast and makes less background error than R-CNN methods. This method struggles with very small objects and objects in dense spaces. It also works best with sophisticated hardware. The authors G. Chandan et al. have proposed the combination of SSD with MobileNet. The combination of MobileNet and Single Shot Detector (SSD) framework gives a fast and efficient method for object detection. The model performs poorly for small object detection.

An enhancement to YOLOv3 was proposed by Kim et al that focuses on detection of vehicles at various scales using spatial pyramid pooling. They introduced two layers for prediction and before every prediction layer inserted SPP networks. This made the system more robust to various sizes of vehicles. It gives mAP of 84.96 on the DETRAC dataset which shows a much higher accuracy than other vehicle detection approaches. The additional prediction layers result in a decrease in frames processed per second (6-7 fps) as compared to YOLOv3.

Katyal et al approaches the problem of vehicle detection in foggy conditions by pre-processing the image and generates a saliency map using regional covariance. Object detection is then performed using YOLO algorithm on the image. Dehaze algorithm is used to pre-process the image and improve the quality. The system was able to recognise objects in foggy conditions which is not possible with traditional YOLO. It required hardware such as fog sensors.

Guanqing Li in their paper focuses on the real time implementation of YOLO. They present a method which focuses on sample enhancement and transfer learning. They are able to show the generalization ability of sample enhancements and transfer learning. They achieved 87.4% detection rate on 6 different targets. It is accurate and rapid. Recall rate is enhanced at a great extent providing better results. Migration learning is a necessity to gain better results, while training if the sample size is less than 50 then has very poor detection.

M. H. Putra in their paper presents a system for detection of real time persons and cars which can be used in Intelligent Car Systems. This is also called Advanced Driver Assistance System (ADAS). They have used YOLO which they have modified to use 7 CNN layers. The paper is able to establish that the reduced complexity YOLO which has high detection accuracy and is capable of real time application and thus is suitable for ADAS use. Reducing the layers results in reduced complexity, this may lead to reduced accuracy but by applying larger grid size the desired results are achieved with good accuracy and speed. It is successfully able to detect small classes too. Although the speed of detection is good yet accuracy is relatively low. The system is only successful if the frame size is large (11x11).

III. METHODOLOGY

The proposed method is a two stage detector is based on YOLOv3 and image classifier. The objective of this model is to bypass the requirement of sophisticated hardware such as GPUs and enable seamless integration with pre-installed systems. It can be used with video streams from existing surveillance systems. The output is displayed in real-time video stream and is also stored for reference later. This two stage model uses YOLOv3 in the first stage to detect two wheeler riders in traffic followed by a classifier to determine if the rider is wearing a helmet. YOLO is a deep learning based object detection algorithm which has shown promising results in real time applications. Here, the pre-trained model provided by the developers of the algorithm has been used which has been trained using the Titan X GPU on the MSCOCO dataset. This model has been used to detect the persons in the input frames. In YOLO, the input image is divided into a grid and predicts bounding boxes for each grid. It uses the darknet53 architecture for feature extraction. YOLOv3 predicts bounding boxes using anchor boxes which act as dimension clusters. Detection is also made across three scales and three boxes are predicted at each scale which is forwarded to the next layers. Thus it performs better with smaller objects as compared to its predecessors. The proposed model is able to classify small scale low resolution images obtained from the first stage. The classifier labels the detection containing helmets and these are displayed in green coloured bounding boxes whereas the detections without helmets are displayed in red coloured bounding boxes. Thus the model is able to detect the riders and classify whether the rider is wearing helmet or not.

IV. IMPLEMENATION

- Video and Image Gathering

Our input datasets were collected from the video surveillance system of Loei Rajabhat University in Loei province, Thailand. A camera we chose to start an experiment is the camera at the front gate of the university. We collected 50 videos of a vehicle passing through the gate, each video is 5 minutes long then the total of all videos length is 250 minutes. After that, we manually classify the image of a biker wearing a helmet and no helmet from the video data. Then we crop an area of a motorcycle with a biker and helmet into one image dataset call "Biker_with_helmet" and the area of a motorcycle with a biker who wears no helmet into another dataset call "Biker_with_no_helmet". The total input image we have in "Biker_with_helmet" dataset is 336 images and for "Biker_with_no_helmet" we have 157 images. The total of them is 493 images.

- Image classification

Experiment After gathering 493 images for our training dataset, we split our images into two groups, one for training data and another for test data to use in classification experiment. This experiment we test them with four CNN models for image classification (VGG16, VGG19, Inception V3, and MobileNets). For the evaluation, we used 10-fold cross validation experiment which we set a number of test data for 10% of the total image. The training networks are trained using Python TensorFlow library, and then we calculate the accuracy and choose two good models to use in image detection step.

- Image Detection

Experiment In this step, we use all 50 videos that we collected to do image detection experiment using SSD technique combine with two CNN models we chose from the previous step. All videos will be tested and calculated the accuracy of the biker with helmet and no helmet detection in the video. We also count a number of undetected motorcyclists to be an error.

- Result Interpretation

The last step, we compare the performance from two previous steps and make the conclusion. The accuracy of the experiments will show the performance of each technique in terms of image classification and image detection.

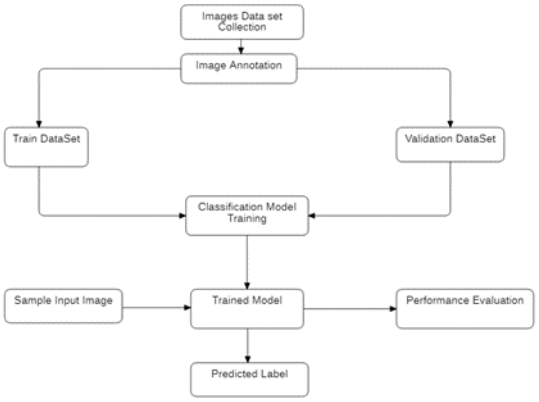
- Working Process

The work has been developed using YOLOv3 and classified based on LeNet architecture. In the Fig. 1 shows the block diagram for the proposed system. First we apply 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. This is discussed in detail as follows 4.1Detection Of Two-Wheeler Rider The first step is to pre-process of 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 case of videos, the video frames are taken as image for pre-processing. Then it is forwarded to the YOLO model for 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 a successful creation of 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 an accurate detection, non-maxima suppression is required. This suppresses weak overlapping bounding boxes. 4.2Localisation of Person YOLOv3 algorithm detects all objects in the MSCOCO dataset. However, the model requires only the person's 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 centre 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. 4.3Helmet 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 non-helmet 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.

V. SYSTEM ARCHITECTURE



- Improving predictions with ensemble methods

Data scientists mostly create and train one or several dozen models to be able to choose the optimal model among well-performing ones. Models usually show different levels of accuracy as they make different errors on new data points. There are ways to improve analytic results. Model ensemble techniques allow for achieving a more precise forecast by using multiple top performing models and combining their results. The accuracy is usually calculated with mean and median outputs of all models in the ensemble. Mean is a total of votes divided by their number. Median represents a middle score for votes rearranged in order of size.

The common ensemble methods are stacking, bagging, and boosting.

Stacking. Also known as stacked generalization, this approach suggests developing a meta-model or higher-level learner by combining multiple base models. Stacking is usually used to combine models of different types, unlike bagging and boosting. The goal of this technique is to reduce generalization error.

Bagging (bootstrap aggregating). This is a sequential model ensembling method. First, a training dataset is split into subsets. Then models are trained on each of these subsets. After this, predictions are combined using mean or majority voting. Bagging helps reduce the variance error and avoid model overfitting.

Boosting. According to this technique, the work is divided into two steps. A data scientist first uses

subsets of an original dataset to develop several averagely performing models and then combines them to increase their performance using majority vote. Each model is trained on a subset received from the performance of the previous model and concentrates on misclassified records. A model that most precisely predicts outcome values in test data can be deployed.

VI. RESULTS ANALYSIS

Figure 1: Screen shot

Result The model works to automate traffic surveillance and successfully detects two-wheeler riders wearing helmet and not wearing helmets. The model uses pre-trained YOLOv3 model which has 57.9 mAPand combined with the classifier which has an accuracy of 95.2%. To evaluate the performance of the resultant model, confusion matrix method was used. It is a table which considers actual and predicted values for the results. The final output of the model is labeled using four parameters. These are True positive (TP), True negative (TN), False positive (FP), False negative(FN). The accuracy is the calculated using the formula Accuracy = TN+TP Total

Using the above formula, the accuracy obtained is 74.4%. the model successfully detect two-wheeler riders. In this case, of the five persons detected, it correctly classifies the two riders wearing helmet using distinguishing color mark as Green and two riders not wearing helmet color mark as Red. The rider in the middle is a false positive and he has a helmet kept on the motorcycle. Such type of cases is included in the result calculation and accuracy is obtained after all this consideration and it is 74.4%.

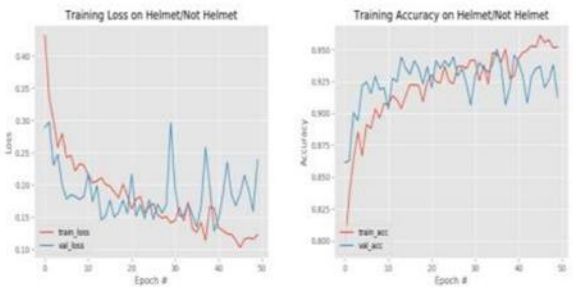


Figure 1:accuracy

The Optimized YOLOv3 gives an accuracy of 93.5% with 35 fps. The YOLODense backbone method gives an accuracy of 98.78% with 21 fps. However these models focus on helmet detection at construction sites. KNN with Feature Extraction gives an accuracy of 74%. The developed method gives an accuracy of 74% with 1 fps. It is a two stage method where in the first stage YOLOv3 detects person and in the second stage the classifier states the presence or absence of the helmet. This method achieves the goal of detecting riders with and without helmets in traffic surveillance video. In the present paper, the developed method does not use GPU which makes it more suitable for current development of helmet detection in real time traffic condition

VII. CONCLUSION

In this paper the target of the developed model is to perform well for the detection of bike riders and classification to detect the presence of the helmets in real time image and video. In the set target, the developed model reached up-to an accuracy of 74% and a speed of 1fps without GPU. In this, a minor drawback of the current developed model is that, it captures the image of all persons coming in the frame rather than bike riders, this is affecting the accuracy of the designed method because it should be applicable on the bike riders only. This implies that if a rider is not wearing a helmet but carrying it, a false positive is returned as the location of the helmet is not checked with respect to the person. A possible solution for this error is to use a detection algorithm in the second layer as well, which may compromise the speed and may require more sophisticated hardware

REFERENCES

[1] Ross Girshick, Jeff Donahue, Trevor Darrell, JitendraMalik(2014), “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation.” The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 580-587.

[2] Girshick, R.B. (2015). “Fast R-CNN.” 2015 IEEE International Conference on Computer Vision (ICCV), 1440-1448.

[3] Liu, Wei &Anguelov, Dragomir&Erhan, Dumitru&Szegedy, Christian & Reed, Scott & Fu, ChengYang & C. Berg, Alexander. (2016). SSD: Single Shot MultiBox Detector. 9905. 21-37. 10.1007/978- 3-319-46448-0_2.

[4] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 779-788.

[5] Redmon, Joseph and Ali Farhadi. “YOLO9000: Better, Faster, Stronger.” 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016): 6517-6525.

[6] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017.

[7] Z. Cai and N. Vasconcelos, "Cascade R-CNN: Delving Into High Quality Object Detection," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, 2018, pp. 6154-6162.

[8] Redmon, Joseph and Ali Farhadi. “YOLOv3: An Incremental Improvement.” ArXivabs/1804.02767 (2018).

[9] G. Chandan, A. Jain, H. Jain and Mohana, "Real Time Object Detection and Tracking Using Deep Learning and OpenCV," 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), Coimbatore, 2018, pp. 1305-1308.

[10] K. Kim, P. Kim, Y. Chung and D. Choi, "Performance Enhancement of YOLOv3 by Adding Prediction Layers with Spatial Pyramid Pooling for Vehicle Detection," 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 2018, pp. 1-6.

[11] S. Katyal, S. Kumar, R. Sakhuja and S. Gupta, "Object Detection in Foggy Conditions by Fusion of Saliency Map and YOLO," 2018 12th International Conference on Sensing Technology (ICST), Limerick, 2018, pp. 154-159.

[12] G. Li, Z. Song and Q. Fu, "A New Method of Image Detection for Small Datasets under the Framework of YOLO Network," 2018 IEEE 3rd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, 2018, pp. 1031-1035.

[13] Waranusast, R., Bundon, N., Timtong, V., Tangnoi, C., and Pattanathaburt, P. (2013). “Machine vision techniques for motorcycle safety helmet detection. 35–40.

[14] Hu, J., Gao, X.,Wu, H., and Gao, S. (2019). “Detection of workers without the helments in videos based on yolo v3.” 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), IEEE. 1–4.

[15] F. Wu, G. Jin, M. Gao, Z. HE and Y. Yang, "Helmet Detection Based On Improved YOLO V3 Deep Model," 2019 IEEE 16th International Conference on Networking, Sensing and Control (ICNSC), Banff, AB, Canada, 2019, pp. 363-368.

[16] M. H. Putra, Z. M. Yussof, K. C. Lim, S. I. Salim, “Convolutional Neural Network for Person and Car Detection using YOLO Framework”, Journal of Telecommunication, Electronic and Computer Engineering (JTEC).

[17] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998. doi: 10.1109/5.726791.