International Journal of Knowledge Based Computer Systems
7 (2), December 2019, 14-17
http://www.publishingindia.com/ijkbcs/

Live Helmet Detection System for Detecting Bikers without Helmet

Harsh Nagoriya
Department of Information Technology, Dharmsinh Desai University, Nadiad, Gujarat, India.
Email: harshnagoriya@gmail.com
*Corresponding Author

Abstract: The higher death rate in motorbike accidents is credited to carelessness in wearing a head protector (helmet) by bike riders. Identification of helmetless riders continuously is a necessary task to forestall the event of such accidents. This paper presents an automated framework to distinguish motor bikers without a head protector (helmet) from traffic observation recordings progressively. In this paper, a Single shot multibox detector (SSD) model is applied to the helmet detection problem. This model can utilize just one single CNN system to distinguish the bounding box area of motorbike and rider. When the area is chosen we classify whether the biker is wearing or not wearing a helmet on real-time. Convolutional Neural Network is applied to select motorbikers among the moving objects and recognition of motorbikers without a helmet. Further applying the You only look once (YOLO) model, I recognize the License Plates of motorbikers without a helmet. So I have applied three models in all through the framework, the custom CNN Model, SSD Model and the YOLO model.

Keywords: CNN, Deep learning, Machine learning, OpenCV, YOLO.

I. Introduction

The significance of automated frameworks in traffic control has expanded in the ongoing years. One objective is to improve the use of a traffic flow system. Further, it can be also used to lessen the causes of a road accident. Road traffic accidents are being perceived as a prima issue in developing nations. Since bikes are the most used method of transport, there has been a fast increase in the number of road accidents because of the way that the most of the bikers don't wear helmets and the vast majority of the road accident deaths occur because of head/brain injuries. According to traffic rule, it is required to wear a helmet while riding a bike. But, many bikers overlook the rules and ride their bike without safety measures. The best possible solution is to build up an artificially intelligent framework that can be mechanized to perceive this sort of issue without human cost. To recognize the bikers who don't wear a helmet, we need techniques to detect the bike and rider from the video frames. Further, we need to recognize an area of the biker's head and classify whether the rider is wearing a helmet. A few CNN models have been presented [1] [4] [5], yet the vast majority of these models can be used distinctly to classify or recognize one object from the image, not for multiple objects. For settling this goal, here I used a Deep learning method called Single Shot Multibox Detector [1] [6] [11]. I applied this technique to solve the helmet detection issue from a surveillance video camera utilizing CNN models and SSD method [4] [6]. Moreover, I captured the number plates of the bikers who don't wear helmets applying the YOLO model [10].

II. Proposed System

The objective of the proposed system is to distinguish the riders who don't wear helmets. So as to accomplish this objective, video datasets are gathered from surveillance cameras and the containing boxes containing riders are acquired from every video frame. I have used an SSD Model for detecting riders [1] [6]. After that, I cropped the head part from the stored bounding box pictures. Further, I trained machine on the cropped pictures by making a custom CNN Model [4] [8] [9]. Pictures with and without a helmet are distinguished. At last, the YOLO Model is applied to identify the number plates of riders not wearing helmets [10].

A. Architecture

The system framework starts with getting the recordings from the surveillance cameras. Each video contains numerous video frames. Object Classification is done from the chose area of every image extracted from video frames. The images that don't contain motorbikes are dismissed. Once the images with motorbike are identified. The received images are compared using a scale-invariant feature transform (SIFT) method to cluster the analogous images into groups [2] [5]. The head part of clustered images is cropped and stored one by one. At that point, a custom model is made to train system helmet images. Object classification is done applying the custom model to group the images into classes specifically; with helmet and without a helmet. After that, number plates of the bikers without helmets are identified applying the YOLO model [10].

B. Classification

i. Implementation of Video and Image Gathering Module

In this module, input datasets were gathered from the video surveillance system. After that, object detection of a motorbike is done from the video frames of the gathered dataset. At that point, we crop an area of a bike with a rider and helmet [2]. Further similar images are assembled into cluster lastly the head part is cropped and put away independently in each cluster.

ii. Implementation of Image Training

In this module, helmet pictures are classified into two parts, one for training and another for validation. We test data by making a custom CNN model. The training networks are trained using Python Keras library with Tensorflow Backend and we calculated the accuracy of the trained model [3] [4] [5] [7] [8] [9].

iii. Implementation of Image Detection Module

Once training is finished helmet identification is accomplished from the images with the custom CNN model. The images with or without helmets are stored.

iv. Implementation of Extra Functionality Module

After the motor-bikers without helmets are recognized, the extra framework module catches the number plate of those motor bikers without helmets. The YOLO model is applied for catching number plates. It can run and store predictions on more than each picture in turn [10].

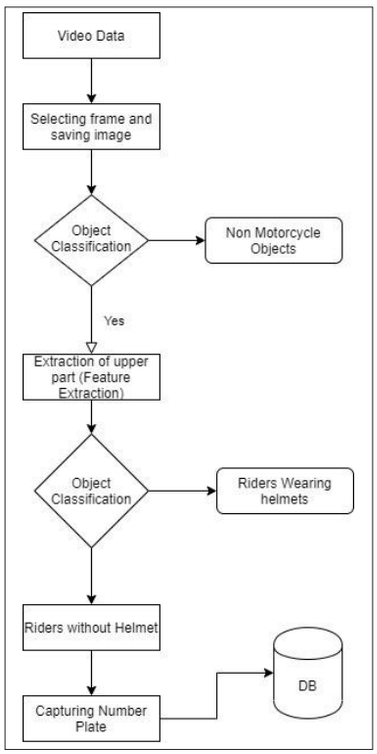

Fig. 1: Architecture and Algorithm Flow

III. Models

A. Single Shot Multibox Detector

SSD is a mainstream algorithm in object detection [1]. It is a deep neural network model to achieve both image segmentation and image classification at the same time in just one execution [6]. An objective of SSD is to find a proper bounding box in every image that ought to be considered to detect an object and then use that area of the bounding box to arrange the type of object. In addition, the best part is that it can detect numerous objects in the image executing only once and is significantly quicker in speed and giving high-accuracy. SSD object detection algorithm makes out of 2 sections: Extract feature maps and apply convolution filters to detect objects. SSD utilizes VGG16 to extract feature maps. At that point, it detects using 3 layers of Con4. SSD is more reliable than YOLO [11].

Fig. 2: Bounding Boxes using SSD Algorithm

B. You Only Look Once

YOLO is a Convolutional Neural Network for real-time object detection. The algorithm executes a single neural network to the whole image, and then divides the image into sectors and predicts bounding boxes and probabilities for each sector [10]. YOLO is similar to FCNN and MaskRCNN [3].

Fig. 3: Extraction of Head Part

IV. Implementation

First thing first, the video frame images are passed to the SSD model to find the bounding boxes of bikers and head/upper part of the bikers. SSD has two components: a backbone model and SSD head [6]. Backbone model usually is a pre-trained image classification network as a feature extractor. This is typically a network like ResNet trained on ImageNet from which the final fully connected classification layer has been removed. I thus left with a deep neural network that is able to extract semantic meaning from the input image while preserving the spatial structure of the image albeit at a lower resolution [11]. After applying this SSD model, the system is trained with manual CNN to classify the system in two categories, with helmet and without helmet and model got trained with accuracy around 96.97%. After classifying bikers without a helmet, such biker's number plate is scanned with YOLO model and data saved into connected DB.

```
[ ] checkpoint = ModelCheckpoint('model-{epoch:03d}.model',monitor='val_loss',verbose=0,save_best_only=True,mode='auto')
    history=model.fit(train_data,train_target,epochs=10,callbacks=[checkpoint],validation_split=0.2)

    Train on 990 samples, validate on 248 samples
    Epoch 1/10
    990/990 [==============================] - 77s 78ms/step - loss: 0.6753 - accuracy: 0.5788 - val_loss: 0.8024 - val_accuracy: 0.5121
    Epoch 2/10
    990/990 [==============================] - 76s 77ms/step - loss: 0.5015 - accuracy: 0.7657 - val_loss: 0.3982 - val_accuracy: 0.8387
    Epoch 3/10
    990/990 [==============================] - 76s 77ms/step - loss: 0.2926 - accuracy: 0.8758 - val_loss: 0.3172 - val_accuracy: 0.8710
    Epoch 4/10
    990/990 [==============================] - 79s 80ms/step - loss: 0.1937 - accuracy: 0.9273 - val_loss: 0.2296 - val_accuracy: 0.9153
    Epoch 5/10
    990/990 [==============================] - 76s 77ms/step - loss: 0.1900 - accuracy: 0.9222 - val_loss: 0.2456 - val_accuracy: 0.8952
    Epoch 6/10
    990/990 [==============================] - 76s 76ms/step - loss: 0.1196 - accuracy: 0.9556 - val_loss: 0.1890 - val_accuracy: 0.9395
    Epoch 7/10
    990/990 [==============================] - 76s 76ms/step - loss: 0.0800 - accuracy: 0.9697 - val_loss: 0.1928 - val_accuracy: 0.9355
    Epoch 8/10
    990/990 [==============================] - 75s 76ms/step - loss: 0.0895 - accuracy: 0.9636 - val_loss: 0.2277 - val_accuracy: 0.9153
    Epoch 9/10
    990/990 [==============================] - 76s 76ms/step - loss: 0.0492 - accuracy: 0.9848 - val_loss: 0.3128 - val_accuracy: 0.8750
    Epoch 10/10
    990/990 [==============================] - 76s 76ms/step - loss: 0.0872 - accuracy: 0.9697 - val_loss: 0.1802 - val_accuracy: 0.9274
```

Fig. 4: Training Model using Custom CNN

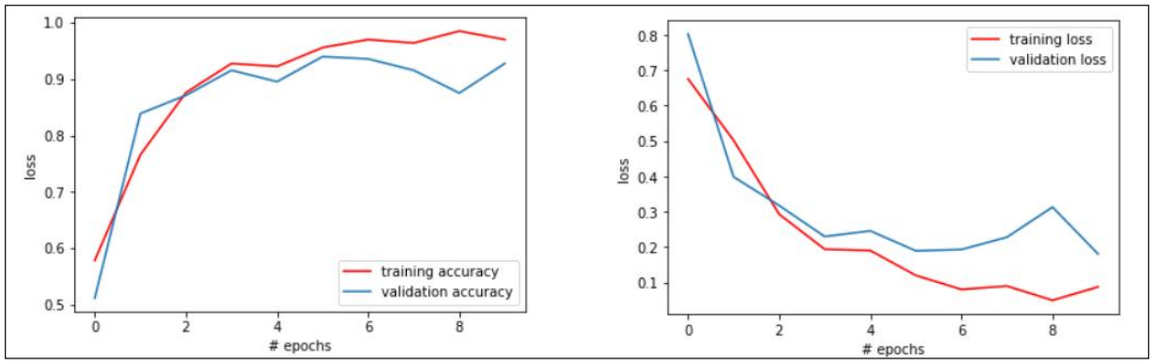

Fig. 5: Accuracy Plot of the CNN System

Fig. 6: Extraction of Number Plate

V. Significance and Enhancements

This framework can be introduced in major traffic crossing points and main roads to mechanize the task of traffic rule management. If improved, it additionally robotizes the task of penalty form creation in the event of rule infringement.

VI. Conclusion

In my venture, deep learning technology is applied to comprehend the issues of motorbikers not wearing helmets. Single Shot Multibox Detector is a mainstream algorithm in object detection. It does both the image segmentation and image

classification simultaneously. SSD method can distinguish the pictures by executing just one single runtime. It requires no other picture pre-processing algorithms. Convolutional Neural Network is the best-fit algorithms that we can apply to the objectives of image detection and classification. Further, YOLO models can successfully identify the number plates of those motor-bikers who are not wearing helmets.

Acknowledgement

To all the peers and mentors who helped at the time of practical implementation.

References

[1] N. Boonsirisumpun, W. Puarungroj, and P. Wairotchanaphuttha, "Automatic detector for bikers with no helmet using deep learning," *2018 22nd International Computer Science and Engineering Conference (ICSEC),* Chiang Mai, Thailand, pp. 1-4, 2018. Available: https://doi.org/10.1109/ICSEC.2018.871277

[2] N. Dalal, and B. Triggs, "Histograms of oriented gradients for human detection," In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 886-893, 2005.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," In *Advances in Neural Information Processing Systems*, pp. 91-99, 2015.

[4] L. Shine, and C. V. Jiji, "Automated detection of helmet on motorcyclists from traffic surveillance videos: A comparative analysis using hand crafted features and CNN," *Multimedia Tools and Applications*, vol. 79, pp. 14179-14199, 2020. Available: https://doi.org/10.1007/s11042-020-08627-w

[5] C. Vishnu, D. Singh, C. K. Mohan, and S. Babu, "Detection of motorcyclists without helmet in videos using convolutional neural network," *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, pp. 3036-3041, 2017. Available: https://doi.org/10.1109/IJCNN.2017.7966233

[6] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," In *Proc. ECCV*, Amsterdam, The Netherlands, pp. 21-37, 2016.

[7] H. Jeremy, Lesson 9: Deep Learning Part 2 2018 - Multi-object Detection. Available: https://docs.fast.ai/vision.models.unet.html#Dynamic-U-Net

[8] M. D. Zeiler, and R. Fergus, "Visualizing and understanding convolutional networks", In *European Conference on Computer Vision*, pp. 818-833, Springer, Cham, 2014.

[9] D. H. T. Hien, "A guide to receptive field arithmetic for convolutional neural networks". Available: https://medium.com/mlreview/a-guide-to-receptive-field-arithmetic-for-convolutional-neural-networks-e0f514068807

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2015. Available: arXiv:1506.02640

[11] Webpage, "How single-shot detector (SSD) works?," ArcGIS API for Python. Available: https://developers.arcgis.com/python/guide/how-ssd-works/