iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 8 **Issue:** X **Month of publication:** October 2020

DOI: https://doi.org/10.22214/ijraset.2020.31775

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue X Oct 2020- Available at www.ijraset.com

Detection of Motor Bicyclist Violating Traffic Rules using Computational Neural Networks

Boggula Yugnath Reddy[1], Marcin Budka[2]
[1](MSc, Data Science and Artificial Intelligence), [2](Professor), Bournemouth University, Talbot Campus, United Kingdom

Abstract: ***This project is based on automatic detection of motorcyclist who are violating traffic rules. Detection of motorcyclist who are not following traffic rules is quite complicated and laborious. If there are more than one person who are travelling on the same bike considered as violating traffic rules. Once the traffic violator identified, the number plate of such motorcycle is extracted. In this paper we discussed about drawbacks of traditional classifier and their drawbacks and to overcome the drawbacks how our proposed neural network helps. Neural network take decision intelligently as it is a deep learning classifier. In existing there is use of HOG (Histogram Oriented Gradient) features with SVM (Support Vector Machine) classifier while in proposed work we used yolo v3 for detection of traffic violators.***
Keyword: Traffic violation rules, neural network classifier, Yolo V3 algorithm, Deep Learning

I. INTRODUCTION

The aim of this project is to detect motorcyclists who are violating traffic rules. A person riding a motorcycle without helmet considered as traffic violator. More than one person riding on the same bike are also considered as traffic violators. After finding the traffic violator the number plate of his motorbike is extracted. In this project we are folloeing two approaches, traditional classifier based approach and the second is neural network approach.

1) *Neural Network Approach:* (yolo) we have used yolov3 to detect person, motorbike, and helmet. If a person is found riding a motorbike without helmet, then the number plate of the motorbike is extracted using openalpr.
2) *Traditional Classifier Approach:* (HOG+SVM) we have used SVM classifier to detect helmet in image. The HOG features of helmet dataset and non-helmet dataset are extracted and the features are used to train SVM classifier. Then the classifier is tested to detect helmet in the image. Finally, the number plate of the traffic violator is extracted using openalpr.

A. Helmet Detection using HOG+SVM

In this course work i have used YOLO to detect person and motorcycle present in the image. To detect the helmet present in the image i am using HOG features and SVM classifier. The SVM classifier is trained with color histogram features of the helmet and non-helmet dataset. Then the classifier is tested on the random images to detect helmet. If the test image includes persons on motorcycle without helmet, then the number plate of the motorcycle is extracted using openalpr.

B. Load the Training Data

To train the model we need two types of data i.e. positive and negative. Positive dataset includes 174 helmet images. Negative data set contains non-helmet images like roads, walls, trees etc. I have to extract the HOG features from these two datasets and train the SVM classifier. TO read images one by one i have used glob () function and given the path for of the folder where the helmet dataset is located.

I have created an empty array named helmet_images_original and using a loop each image in the helmet dataset is appended into the array. The length of the array returns the total number of helmet images in the dataset. Apeending images to array will be done only after converting their color space to RGB.

The same process is used to read images from nonhelmet images dataset folder.

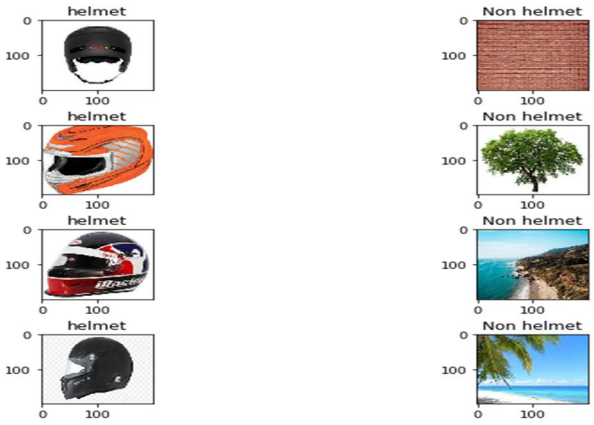
Each image in the dataset is converted into RGB color spce and appended to non_helmet_images_original. The length of this array gives the number of nonhelmet images. In the output i have printed number of helmet and nonhelmet images count. The dimensions of each image in the training dataset should be same.

The extracted HOG features for all the images have to be concatenated. For this to happen all the features dimensions should be same i.e. image dimensions. So i have used images with 200X200 dimensions in the dataset.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 8 Issue X Oct 2020- Available at www.ijraset.com

C. Visualization of Sample images from the Dataset

After rading the helmet and nonhelmet datasets, i have diplayed five randomly choosen images from helmet and nonhelmet categories. And also displayed the shape of helmet and nonhelmet images.



D. HOG

Histogram of Oriented Gradients (HOG) is a feature descriptor used to detect objects in an image. This technique counts occurances of gradient orientation in localized portions of the image. A local object appearance and shape within an image can be described by the distribution of intensity of gradients. First the image is divided into number of cells. Each cell consists a fixed number of pixels. For each pixel within each cell, a histogram of gradients direction is computed. Finally, the descriptor is obtained by concatenating all the histograms. To improve the accuracy, the local histograms can be contrast-normalized by calculating a measure of the intensity across a larger region of the image, called a block, and then using this value to normalize all cells within the block. This normalization results in better invariance to changes in illumination and shadowing.

E. Extracting HOG Features of an Image

I have used a general method from opencv documentation to extract the HOG features of an image. This function takes image, orientation, pixels per cell, and cells per block as input. This function calculates the histogram of gradient for each pixel in the cell and the features in every block are concatenated to return the HOG features. I have tested the Get Features From Hog by extracting HOG features of a randomly chosen imagen.

After loading the image i have converted the image to YUV color space. A YUV image contains only two color components. The third component represents the LUMA component which represents brightness. I have assigned three channels in the image. For each channel in the image i have extracted HOG features by calling the function Get Features From Hog ().

Then i have plotted HOG features of the three channels and the original image. And also printed the feature vector length. Since the input image is of dimensions 200X200, we can extract atmost 40000 features from the image.

F. Extract Features of Dataset

Before extracting the HOG features from an image, i have converted the image into YUV color space.

G. Preprocessing Data

I have imported train_test_split from sklearn module to split the data into training and testing data. 20% of training data is used to test the model.

H. Train SVM classifier

In this course work i have used SVM classifier. The SVM classifier is trained with the extracted HOG features. A support vector machine (SVM) is a supervised machine learning model that uses classification algorithms for two-group classification problems. A support vector machine takes the data and outputs the best hyper line the seperates the features. Anything data that falls to one side of hyper line comes as one class and the data that falls on the other side comes as second class.

I have imported Linear SVC from sklearn. Then the model is fit with the training data. The model results are stored in classifier1 variable. After training the model i have checked the accuracy of the classifier and got accuracy of 1 when tested with 20% of training data kept aside for testing. The next step in this course work is to detect helmet in a test image by using classifier results.

I. Sliding Window Technique to find Objects in an Image

After training linear SVC model with the training data, the model is tested with test images. I have used sliding window technique to detect whether the helmet is present in the image or not. First i have make a copy of the original image. Then draw bounding boxes all over the image. Iterate through all the bounding boxes in the image and draw boxes with given coordinates. The function draw_boxes () gives draws the sliding windows in the image.

J. Detecting Helmet in the test Image

Finally, i have tested the classifier with some test images. After reading the test image, i have drawn windows over the image by using slide window () function. I have drawn windows of dimensions (80, 80), (128,128), (200,200), and (252,252). And also printed the number of windows. By using a generalized assumption that the helmet portion generally lies in the upper portion of the image when the person wearing a helmet is sit on motorcycle, i have processed the upper half of the image only to reduce the processing time. After drawing windows on the image, by using classifier output i have extracted the refined windows. I have plotted the images with windows and refined windows.

K. Apply Heat Map to find the Helmet

I have applied heat map to the image with refined windows. Heat map is a tool to visualize complex statistical data. It is a two dimensional representations of data in which values are colors. Heat map is applied to the refined windows image and the false positives (non-helmet windows) are removed by applying threshold. After removing the false positive windows, windows with helmet are only left. The add heat function adds one to each pixel in the box by iterating through each bounding box.

II. DETECTION OF TRAFFIC VIOLATORS USING YOLOV3

We have used yolo version3 to detect helmet, motorbike and person in an image. Yolo stands for “you only look once". It is a real time object detection system.

1) *How yolo Different from other Approaches:* The yolo approach is completely different from Traditional classifier approaches. Classifier approaches apply the model to an image at multiple locations and scales. The high scoring areas in the image are considered as detection. Whereas yolo is applied on the total image at a time. This neural network divides the image into regions and predicts the bounding boxes and probabilities for each region. The bounding boxes are weighted by the predicted probabilities.
2) *How yolo Different from other Approaches:* The yolo approach is completely different from Traditional classifier approaches. Classifier approaches apply the model to an image at multiple locations and scales. The high scoring areas in the image are considered as detection. Where as yolo is applied on the total image at a time. This neural network divides the image into regions and predicts the bounding boxes and probabilities for each region. The bounding boxes are weighted by the predicted probabilities.
3) *YOLO v3 Architecture:* The above figure shows the architecture of yolo v3. Yolo v3 uses the variant of dark net, which originally has 53 layers network trained on image net. Later 53 more layers are stacked on it, giving 106 layer fully convolutional underlying architecture for yolo v3.
4) *Working Principle of yolo v3:* YOLO v3 makes prediction at three scales, which are precisely given by down sampling the dimensions of the input image by 32, 16 and 8 respectively. The first detection is made by 82nd layer. For the first 81 layers the image is down sampled by the network, the 81st layer has a stride of 32. If we have an image of 416 x 416, the resultant feature map would be of size 13 x 13. One detection is made here using the 1 x 1 detection kernel, giving us a detection feature map of 13 x 13 x 255. Then, the feature map from layer 79 is subjected to a few convolutional layers before being up sampled by 2x to dimensions of 26 x 26. This feature map is then depth concatenated with the feature map from layer 61. Then the combined feature a map is again subjected a few 1 x 1 convolutional layer to fuse the features from the earlier layer (61). Then, the second detection is made by the 94th layer, yielding a detection feature map of 26 x 26 x 255. A similar procedure is followed again, where the feature map from layer 91 is subjected to few convolutional layers before being depth concatenated with a feature map from layer 36. Like before, a few 1 x 1 convolutional layers follow to fuse the information from the previous layer (36). We make the final of the 3 at 106th layer, yielding feature map of size 52 x 52 x 255. Detections at different layers help address the issue of detecting small objects. The upsampled layers concatenated with the previous layers help preserve the fine grained features which help in detecting small objects. The 13 x 13 layer is responsible for detecting large objects, whereas the 52 x 52 layer detects the smaller objects, with the 26 x 26 layer detecting medium objects.

5) *Creating Custom coco Weights for Helmet Detection:* We can train yolo network by using pretrained coco weights for object detection. I this project we have to detect helmet in ana image. There are no pretrained coco weights for helmet object readily available to use. So we have created custom coco weights for detecting the objects. To train yolo we need all of the COCO data and labels for images. The script scripts/get_coco_dataset.sh will get the COCO weights. Then we can go to darknet directory and will change the configuration file.
6) *Test Dataset:* The main objective of this project is to find traffic violators. A single person riding a motorcycle wearing a helmet is considered as a traffic rule follower. Persion riding without helmet, double or triple riding are considered as traffic violators. By considering all these cases i have created test images incluing all the above test cases. In test images we have 10 images.

A. Data Augmentation / Labelling

Transfer learning to train the yolo model and generate the yolo weights transfer learning was applied from the existing yolo model. For data augmentation / Labelling following steps were applied:

1) *Step 1:* With the help of an open source tool called "labelimg" available in the open source git repo - "https://github.com/tzutalin/ labelimg", the tool was installed and selected images were loaded in the tool
2) *Step 2:* Boundary boxes were manually drawn and labelled
3) *Step 3:* Once the Images was labelled, the image was saved in the yolo format which is .txt file format with the sample data as given below 0 0.401765 0.664448 0.588831 0.541474 1 0.430700 0.266300 0.175637 0.117091 2 0.388455 0.492284 0.446470 0.593750 0 - Class, rest are the boundary boxe coordinates Classes chosen for the labelling are given below 0 - Motorcycle 1 - Helmet 2 - Person
4) *Step 4:* The directory should have both the image and respective YOLO format files. as given below n3.jpg n3.txt
5) *Step 5:* Download the darknet pretrained model and yolov3.weights! Git clone https://github.com/pjreddie/darknet %cd darknet! Make! Curl -O https://pjreddie.com/media/files/yolov3.weights

B. Open ALPR

The number plate of the traffic violator is extracted by using open ALPR. Open ALPR is an open source Automatic License Plate Recognition library written in C++ with bindings in C#, Java, Node.js, Go, and Python. The library analyzes images and video streams to identify license plates. The output is the text representation of any license plate characters recognized in the processed image.

III. RESULTS

HOG+SVM: HOG+SVM detected helmet object in the image effectively for some images. Since there was a potential difficulty with the dataset, we have trained SVM with 313 images of helmets and non-helmets. So the models detecting some objects as helmets, which are not helmet. But it is detecting the helmet part accurately. We need more number of images in the dataset to train SVM model. As SVM is two group classification algorithms, we can detect only one object at a time. YOLO V3: we have got accurates results with yolo compared to HOG+SVM. YOLO model is able to detect multiple objects at a time and it has detected very small objects in the image also. We have detected number of persons on the bike, helmet, and motorbike accurately. And we have extracted the number plate of the traffic violators accurately.

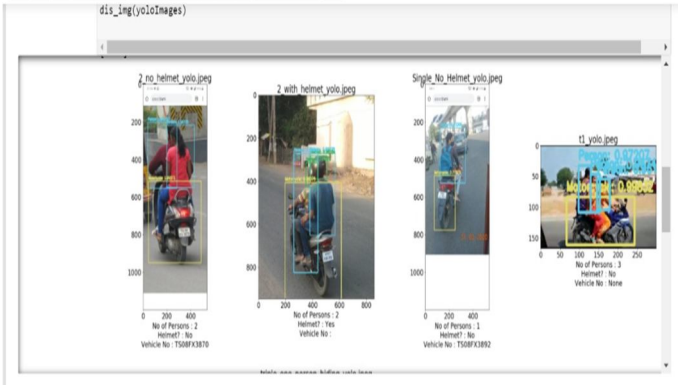


Fig.3.1 some results obtained by software analysis

IV. CONCLUSION

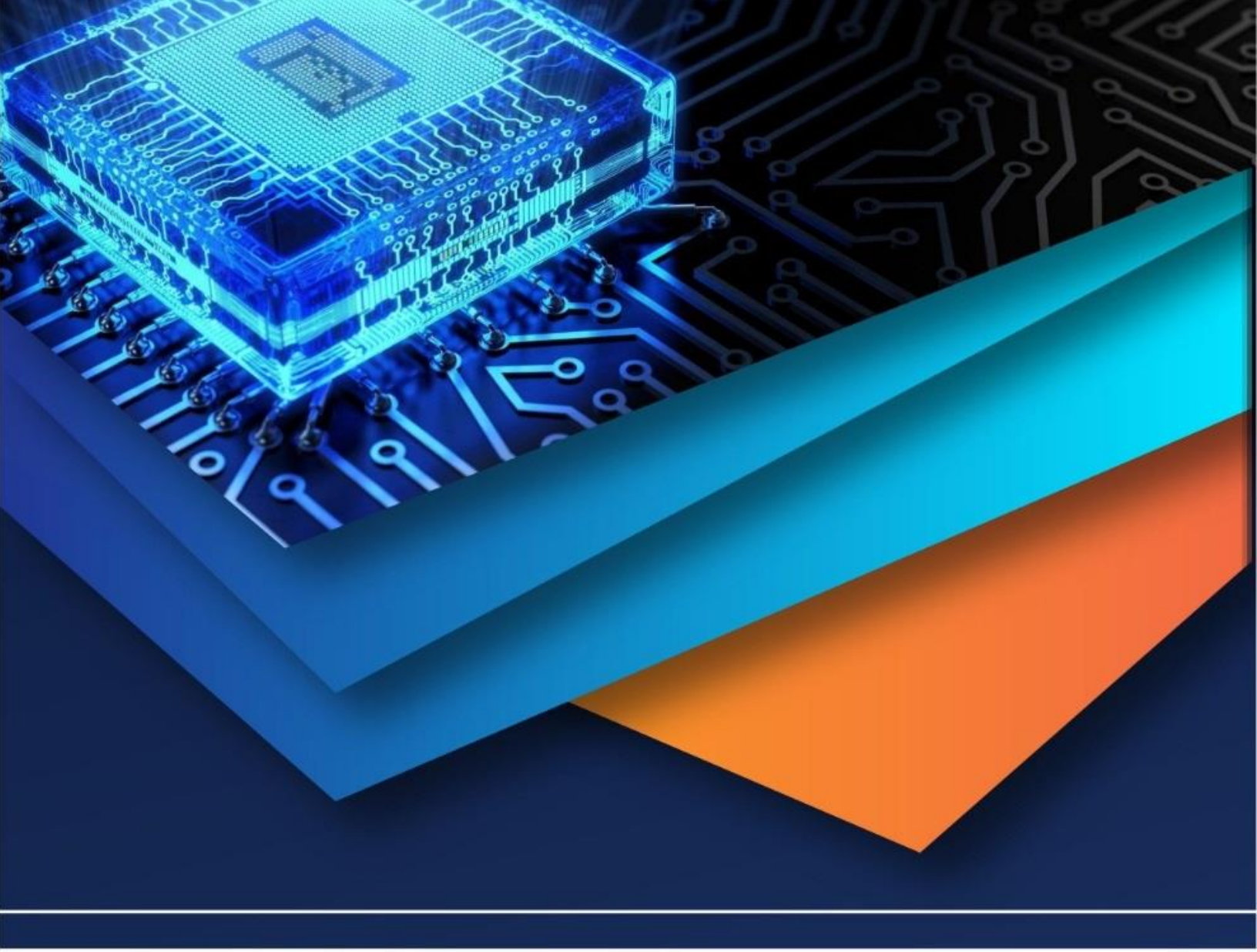
In conclusion yolo detected objects in the image accurately. As it is a convolutional neural network the performance of yolo is far above HOG+SVM. Open ALPR performed accurately to detect lisence plate of the vehicle.

REFERENCES

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

[2] V. Kakani, D. Gandhi and S. Jani, "Improved OCR based automatic vehicle number plate recognition using features trained neural network," 2017 8th International Conference on Computing, and Networking Technologies (ICCCNT), Delhi, 2017, pp. 1-6. C. Vishnu, D. Singh,

[3] 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, 2017, pp. 3036-3041.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY