International Journal of Recent Technology and Engineering (IJRTE)
ISSN: 2277-3878 (Online), Volume-8 Issue-5, January 2020

Accident Detection and Number Plate Recognition using Image Processing and Machine Learning

Abhishek L, Aravindhan S, T Raghav Kumar

Abstract -The purpose of this project is to detect the accident before it happens along with theextraction the number plate. Different image processing techniques along with morphological operators and Canny Edge Detection are used for image enhancements and object outline detections. With analysis of continuous frames, the relative velocity and the distance from which the leading vehicles are moving could be computed which is further helpful in accident detection and thus prevention too. Histogram of Oriented Gradients (HOG features) are used for feature extraction. Different machine learning classification algorithms like SVM, MLP, and XGBoost are used for classification of the object. Different standard OCR tools like Pytesseract, PyOCR, TesserOCR are used for the retrieval of the vehicle number from the extracted licence plate sub-image.

Index terms –Accident Detection, Vehicle Collision Avoidance, Licence Plate Recognition, Image Processing, Morphological Operators, Canny Edge Detection, Histogram of Oriented Gradients, HOG, Machine Learning, XGBoost, Multilayer Perceptron, MLP, Support Vector Machine, SVM

I. INTRODUCTION

The primary objective of this project is to avoid vehicle collisions and prevent accidents. This is achieved by placing a camera at the front of the car to scan and observe the environment in which the car is further driven. Continuously, images are captured in minute time differences with which analysis could be done and preventive measures could be taken. One of the primary steps involved in this project is detection of a car in the image as without a car at the scene or the image that the front camera captures, there is no possibility of accidents. For the detection of a car in the image, machine learning classifier models are trained with a renowned standard dataset with which the presence or absence of a car in a given image could be determined. The feature descriptors that is helpful for the above learning in Histogram of Oriented Gradients (HOG features) [7]. Different machine learning classifiers ae used for the above prediction namely XGBoost, MLP, and SVM, and their accuracies compared. After the detection of car in the scene, with the variation of pixels from consecutive image frames, the distance of the vehicle from our car and the velocity the vehicle could be determined. With the relative velocity between the vehicle and car, the happening of collision could be determined.

Manuscript received on December 10, 2020.
Revised Manuscript received on December 20, 2020.
Manuscript published on January 30, 2020.
* Correspondence Author
Abhishek L*, School of Computing Science and Engineering, Vellore Institute of Technology Chennai, Chennai (Tamil Nadu), India
Aravindhan S, School of Computing Science and Engineering, Vellore Institute of Technology Chennai, Chennai (Tamil Nadu), India
T Raghav Kumar, School of Computing Science and Engineering, Vellore Institute of Technology Chennai, Chennai (Tamil Nadu), India

If the collision is likely to happen, along with alerting the driver and applying brakes to automated braking system, this project also extracts the licence plate as sub-image and sends either the sub-image to cloud or sends the vehicle number alone as text after retrieving through standard OCR tools like pytesseract, PyOCR,etc.

II. METHIDOLOGY

On the first hand, a camera sensor is placed at the front of the car which is helpful in capturing the images of the vehicles that leads, that is, vehicles that ride in the front of the car. This camera sensor captures images in minute time differences. These images are taken and further processed to identify the presence or absence of the car.

Many feature descriptors come in handy for the same. But Histogram of Oriented Gradients performs very well as they directly help us in understanding the shape of the objects in the image through two values – magnitude and orientation. With help of these HOG featuresretrieved from a dataset of different images of the rear-side of cars, a machine learning classifier is trained and hence can be determined for presence or absence of a vehicle in any given image.

Different ML models [1] are used for the above purpose namely XGBoost, MLP and SVM as said earlier. A standard car dataset [8]taken by Brad Philip and Paul Updike, California Institute of Technology is used for learning.Car images of other standard car datasets that majorly cover the rear-side of the car are also added to this dataset to improve the accuracy of prediction as more number of data points to learn leads to better learning and thus better precision in prediction. Once the presence of the vehicle is found, further monitoring takes place.

Initially when the camera sensor is first placed at the front of the car, a vehicle with average width at the rear is placed at the front a car at a distance ‘d’ and the systems are calibrated with the number of pixels that vehicle’s rear width occupies in the image captured by that camera sensor with focal length ‘f’ when placed at distance ‘d’. This helps us in computing distances between the car and the other vehicles that are travelling ahead of the car.

So, while our car is driven, the camera captures image. And after the presence of car is detected, with the number of pixels that vehicle’s rear width occupies in the image, the distance of the vehicle from car could easily be determined with simple indirect variation and proportion. This computed distance would be accurate if the camera used now and during initial calibration are same. i.e., with same focal length.

Also, since images are captured in small time intervals, the distance that the vehicle moved in that time interval could be found which directly gives us the average velocity in which the vehicle is moving. With the knowledge of the vehicle's velocity and the velocity of our car, the average relative velocity could be computed.

With knowledge of relative velocities and with determining whether the distance between the vehicle and car reduces less than an acceptable threshold value, the system could identify the collision and hence could warn and alert the driver or apply brakes if the car has automated braking system.

In addition to these, when the system identifies the possibility of collision as high, it analyses the vehicle image and retrieve the licence plate alone as a sub-image. Different image processing techniques like morphological operations [2], Gaussian blurring, and canny edge detection [13] are used for above purpose. This vehicle number is extracted as text from this licence plate sub-image and sent to cloud to identify and track the vehicle or driver in future. Pytesseract, PyOCR are some standard OCR tools that helps in extraction of text.

III. FLOW DIAGRAM

The flow diagram in Fig.1 clearly depicts the entire flow of this project in a simplified manner. It also helps us to understand the flow in better manner as it looks organised.

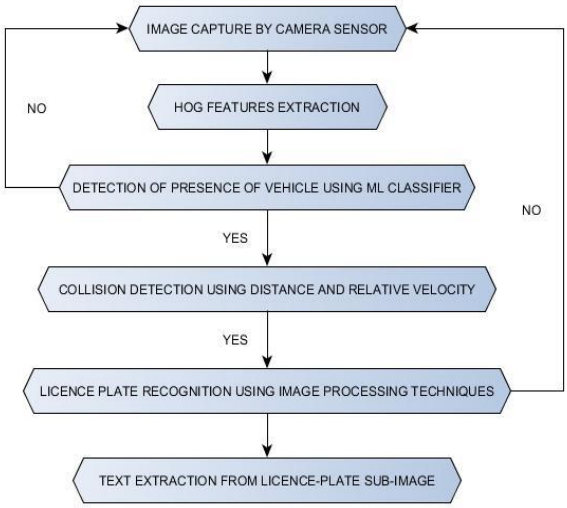

Fig.1. FLOW DIAGRAM

IV. HISTOGRAM OF ORIENTED GRADIENTS (HOG FEATRUES)

Histogram of Oriented Gradients [7] are one of the best feature descriptors in case of object detection and recognition as the extracted features directly describes the shape of the object. Another primary reason for HOG being one of the best feature descriptors is that they are invariant to illumination, shadowing, image rotation and tilting, etc.

For every structuring element, magnitudes along with orientations are determined by these features. These orientations are indicated through angles in degrees which are split into 'n' bins where each bin has 180/n possible values.

In this project, HOG features are computed with 9 bins each having values of 20^o intervals with a window size of 64x64 and a block size of 16x16. The block stride used for above purpose is 8x8 and the final cell size is also 8x8. OpenCV library [3] is used to compute HOG features from the image for the object or vehicle detection.

The vehicle detection [6] is done through these four steps which were explained in simple words earlier - Gradient computation, Orientation binning, Descriptor blocks, Block normalization. For all images in the dataset, the HOG features are computed after converting into any standard set of dimensions and stored as one data-frame with which the ML Classifier is learned. Similarly, for the new image also, HOG features are computed which are used for the detection of presence of vehicle in the image or not.

The Fig.2 shows a plot between an image and the features extracted from it after scaling. This plot helps us to understand the image and the shape of the different objects present in it.

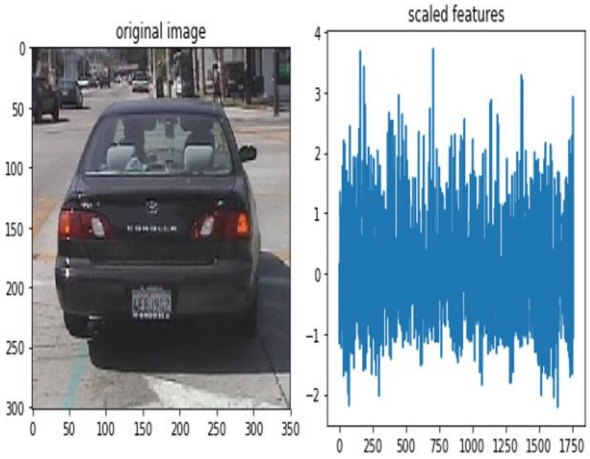

Fig.2. Car Image vs Scaled Features

V. DATASET

Major images were taken from the Car Dataset [5]byBrad Philip and Paul Updike, California Institute ofTechnology. All the images from this dataset contain only the rear part of the car and hence best suitable for this project as the camera sensor would be capturing only the rear part of the vehicle that is riding in the front.

Car Dataset [8]byBrad Philip and Paul Updikewas taken on the freeways of southern California.Resolution is constant at 320x240 pixels. A total of 1155 images were taken from this dataset.

To this dataset, some images of cars and vehicles from other datasets were also added where the image primarily has the rear side of the vehicles in the image. The other datasets include Automobile Dataset from UCI Repository [10] and Cars Dataset from Stanford[9].

VI. MACHINE LEARNING CLASSIFERS

Different machine learning classifiers [1] were trained with the above dataset and the weights are used to predict the presence of a vehicle in a given new fresh image as said earlier. These include XGBoost, MLP and SVM. These include variations with learning rates, number of estimators, and depth of tree in case of XGBoost, with activation functions in case of MLP and with kernels in case of SVM.

As these help us in identifying the presence of a vehicle at front which is a very necessary element in this project, this module as one of the primary modules of this project.

A. eXtreme Gradient Boosting - XGBoost

XGBoost [14] is one of the machine learning techniques for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It is a systematic solution to combine the predictive power of multiple learners although the final resultant is a single model that produces the aggregated output from different models.

XGBoost classifier gave an accuracy of 95.4% when the learning rate is 0.05 and number of estimators and max-depth are set as 300 and 4 respectively. This is the maximum accuracy obtained from XGBoost learner with different learning rates.

B. Multi-Layer Perceptron - MLP

A simple linear perceptron consists of an input layer and an output layer which are fully connected whereas these multilayer perceptrons have the same input and output layers but may have multiple hidden layers in between them.

Unlike simple perceptrons that cannot classify the dataset that are not linearly separable, MLPs could solve the XOR type problems. MLPs break this restriction and classifies datasets which are not linearly separable. They achieve this by using a more robust and complex architecture to learn the classification models for difficult datasets.

When MLP with ‘relu’ activation function is used without ‘early stopping’ enabled for classification, the accuracy was observed to be 97.1%. MLP generally performs well when the data is very large. Since the dataset used here do not have ample number of records, there is a possibility that the other models could perform even better.

C. Support Vector Machine – SVM

Support vector machine (SVM) is a pattern classification algorithm with nonlinear formulation. SVM maps input datainto a high-dimensional feature space, where it constructs an optimal discriminant hyperplane using a nonlinear kernel function.

SVM performs on the principle that, any dataset on transformation to higher dimensionalities, the dataset will become separable by a hyperplane. SVM generally performs well when the dimension or the attributes of the dataset is very high. And since the HOG features are computed which result in several attributes, SVM should perform very well in this case.

When SVM with ‘rbf’ kernel is used for classification of this dataset, the accuracy observed is 98.3%. This accuracy is the highest accuracy that is observed so far.

VII. FURTHER PROCESSING

Once the presence of vehicle is detected in the captured image, the detected box spots are plotted[11] on the image after applying Gaussian blur [3]. Then a bounding rectangle outline is plotted in such a way that it encompasses the vehicle alone separately from the image.

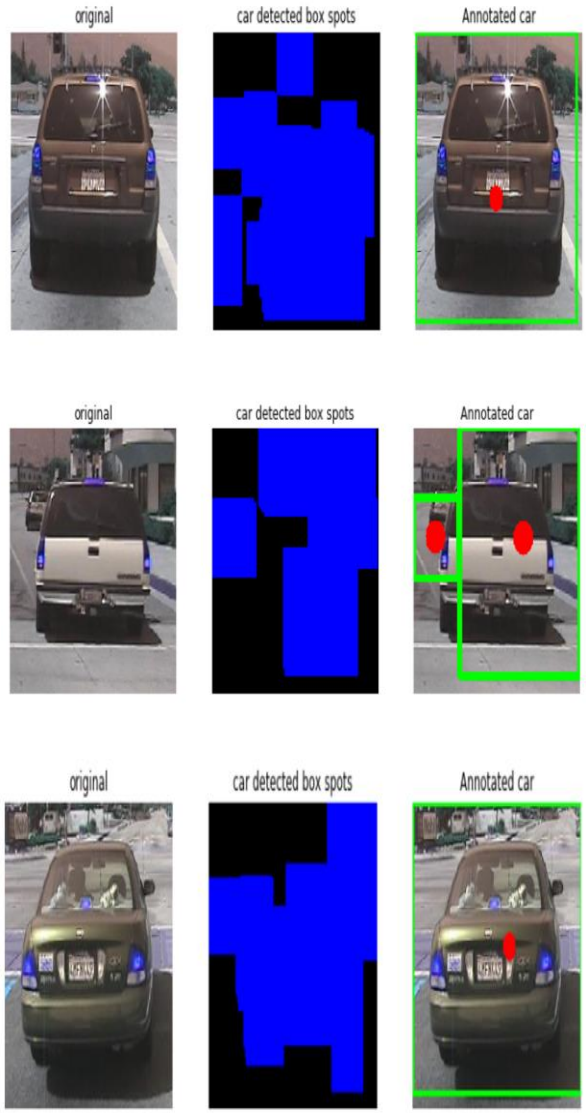

Fig.3. Vehicle Detection

Once the vehicle is detected from the annotated image, with help of the number of pixels that the rear-width of the vehicle occupies in the image and the pre-calibrated distance and number of pixels occupied as told before in this paper, the distance between the vehicle at the front and our car is determined. With help of finding the change in distance after a short interval of time, the relative velocity in which the vehicle is moving could be determined. And with knowledge of this and the speed in which our car moves, the absolute velocity of the vehicle moving at the front could be obtained which further helps in detection of vehicle collision and thus avoidance too.

VIII. COLLISSION DETECTION AND AVOIDANCE

Once we found the velocity of the vehicle at the front, we could easily understand the relative speed patterns as we have the knowledge of the speed in which our car moves. With relative speed that keeps on changes over time, the collision could be predicted before happening which basically satisfies the main objective of this project.

Also, if the distance between the vehicle at front and our car falls below an acceptable threshold value, the system will warn and alert the driver to apply brakes in case of manual braking system. In case of automated braking system, the brakes could be applied on its own when collision possibility is detected. This actually avoids the vehicle-car collision from happening thus preventing the accident. In this way, this project is highly helpful for the self-driving car projects.

IX. LICENCE PLATE RECOGNITION

When the collision possibility in encountered, immediately this project tries to extract the licence plate sub-image from the annotated vehicle image. Retrieval of this number plate helps in identifying the vehicles that met in accident if not prevented so that further police enquiries could be done at ease.

Gaussian Blur is initially applied to annotated vehicle image as image pre-processing for the noise removal and better further processing. Several morphological operators [12] are used to enhance the image. This includes erosion following by dilution so that it would easy for edge detection algorithms to perform well.

Canny Edge Detection [4] is used to identify all the edges on the vehicle so that it will help us in retrieving the licence plate as sub-image more accurately as this algorithm would have plotted a precise boundary over the number plate while detection of edges.

With help of aspect ratios and other shape descriptors, the number plate alone could be easily extracted as sub-image. After the retrieval of the same, different renowned OCR tools are used to extract the vehicle number from the licence plate as text.

Pytesseract, PyOCR, TesserOCR are a few libraries that are helpful in achieving the same. Among these different OCR tools, Pytesseract performed really well as it could extract texts of different font styles and sizes. Pytesseract also extract texts decently when they are rotated by slight angles. That is, Pytesseract takes rotation, tilting, smudging into account every time it tries to extract text.

Once the text is extracted from the licence plate sub-image, the text is sent to the cloud along with the timestamp so that if accident happens, it would help us to identify the culprits. Or the data could also be stored in mini-drives locally itself which could be placed inside our car itself.

If any ambiguity comes while retrieving the text, or if the vehicle number extracted is not in the proper format, the entire licence plate sub-image itself is uploaded in cloud or rather stored locally so that the details of the owner of the vehicle is not missed out. In this way, the vehicle number is obtained in a fool proof manner.

X. RESULT

Usage of HOG features for learning helps in predicting the presence of vehicles. Since the dataset contains vehicles of all types with varying sizes and shapes, the ML classifiers could also determine and predict vehicles of all types and sizes which would act as an advantage in this project. In this way, HOG features highly helps us with detection of vehicles in an image.

Different standard renowned classifiers are used for learning which included **XGBoost, MLP and SVM** with variations in different parameters like learning rate, activation functions, number of estimators, kernels and the accuracies are observed and compared.

ML Classifiers	Accuracies (%)
XGBoost	95.4
MLP	97.1
SVM	98.3

Table 1. ML Classifiers vs Accuracies

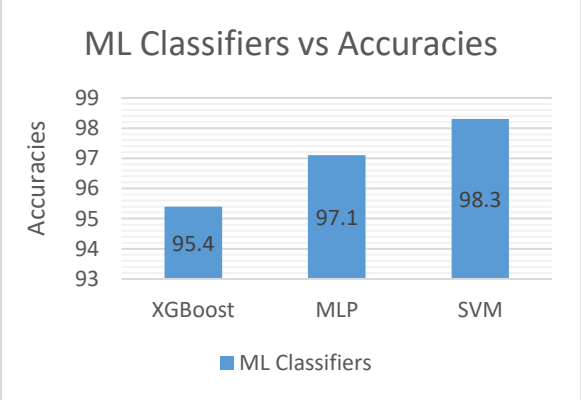

Fig.4. ML Classifiers vs Accuracies

The accuracy for **XGBoost** was observed to be **95.4%** whereas the MultilayerPerceptron (**MLP**) helped in achieving a higher accuracy compared to the XGBoost which is **97.1%.** Although this is a decent accuracy, Support Vector Machines (**SVM**) resulted in the highest accuracy of **98.3%** which is above par.

After the car is detected in the image, the distances and relative velocities are determined with which the vehicle collisions are manipulated. If the possibility of collision is encountered, the driver is warned and the brakes are applied automatically.

Then the process of retrieval of licence plate is initiated at once. This is done with the help of different morphological operators and edge detection algorithms after several image pre-processing techniques.

The vehicle number is successfully extracted as text from the licence plate image through established OCR tools like Pytesseract and PyOCR. This is further stored either locally or sent to the cloud along with timestamp for any time retrieval in future in case discrepancies and enquiries.

XI. CONCLUSION

Histogram of Oriented Gradients is preferred over the other feature descriptors as it helps in getting better accuracies from the ML classifiers as the features directly describe the shape of the objects in the image through magnitude and orientation. Learning the features that describe the shape of an object helps in predicting the objects that resembles with the shapes of the objects that are already learned. That is, this helps us in achieving the object detection module more precisely.

This entire project is thus helpful in accident detection and avoidance. Different image processing techniques along with various machine learning classifiers are used to achieve the above. This project also acts as the one of the highly helpful modules for the projects and researches on self-driving cars. The above proves that this is one the best projects that the society needs in the present or in near future.

REFERENCES

1. scikit-learn, Scikit-learn: Machine Learning in Python, Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J. and Passos A., Cournapeau D, Brucher M, Perrot M, Duchesnay E.
Journal of Machine Learning Research
Volume 12 pages 2825--2830, year=2011
2. Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu and the scikit-image contributors. scikit-image: Image processing in Python. PeerJ 2:e453 (2014) https://doi.org/10.7717/peerj.453
3. opencv_library: The OpenCV Library, Bradski G., Dr. Dobb's Journal of Software Tools, 2000
4. J. Canny, "A computational approach to edge detection", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 8, issue-6, pp. 679-698, Nov. 1986
5. Oliveira, Miguel & Santos, Vitor. (2008). Automatic Detection of Cars in Real Roads using Haar-like Features.
6. "Vehicle Detection and Counting Using Hog Feature Extraction for Traffic Signal Control System" - Wahengbam Suman Singh, SapamJitu Singh, International Journal of Innovations & Advancement in Computer Science, vol, 6, issue-3, pp.83-89, March 2017
7. Histogram of Oriented Gradients, by Satya Mallick, December 6, 2016
https://www.learnopencv.com/histogram-of-oriented-gradients
8. Understanding of the dataset
https://www.researchgate.net/figure/Car-dataset-taken-by-Brad-Philip-and-Paul-Updike-California-Institute-of-Technology-It_fig5_267863282
9. 3D Object Representations for Fine-Grained Categorization - Jonathan Krause, Michael Stark, Jia Deng, Li Fei-Fei, 4th IEEE Workshop on 3D Representation and Recognition, at ICCV 2013 (3dRR-13). Sydney, Australia. Dec. 8, 2013. https://ai.stanford.edu/~jkrause/cars/car_dataset.html
10. Automobile Dataset from UCI Repository
https://archive.ics.uci.edu/ml/datasets/automobile
11. Matplotlib: A 2D graphics environment, Hunter et al., Computing In Science & Engineering, Vol-9, Number-3, pp. 90-95, 2007.
12. Morphological Image Processing, , Chapter 11, Digital Image Processing: A Practical Introduction Using Java, by Nick Efford, Pearson Education, 2000
https://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.htm
13. Canny Edge Detection Step by Step in Python – Computer Vision, by SofianeSahir, Jan 25 https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123
14. An end-to-end Guide to Understand the Math behind XGBoost, by RamyaBhaskarSundaram – Data Scientist, Noah Data, September 6, 2018https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/