Autonomous Driving System With Real Time Testing

Yasha Manjunatha
dept. of ECE,
Stevens Institute of Technology
Hoboken, New Jersey
ymanjuna@stevens.edu

Hong Man
Assistant Professor
dept. of ECE,
Stevens Institute of Technology
Hoboken, New Jersey
hman@stevens.edu

***Abstract*—The goal of autonomous driving is to be as close to human perfection as possible. We can do miracles, especially when it comes to incorporating humanity into autonomous driving if we are able to quantify crucial aspects of driving and the driver's responses to the labels attached to them. With technology like lane detection, car detection, and classification of road signs, we can leverage this problem in this project to construct a data frame of human driving behavior that can be used as training data for a model that can imitate human behavior. This endeavor aims to demonstrate this potential.**

***Index Terms*—HOG Features, Hough Line Transform, VGG16, Transfer Learning, Dataframes, Human Behaviour, Lane Detection, Car Detection, Feature Engineering, SVM with Grid Search.**

I. Introduction

The question of whether humans can choose between different features, make moral judgments, lessen the dimensionality of images, and focus primarily on key factors is one that is often raised. As an illustration, let's consider a straightforward situation where a pedestrian crosses a street. Because of the way our minds are wired, we are aware that there will be a pedestrian crossing sign in front of us on the right as we approach it. As a result, we slow down and stop when we see a pedestrian cross the street. However, it is clear how expertly our minds turn the entire image into a vector. The vector, in this case, is (Road-Crossing Sign: Yes, Which Road Sign: Pedestrian Crossing, Pedestrian there: Yes, Output Label: Stop).

However, it is possible to detect the features mentioned above, particularly with the development of computer vision tools. These include Machine Learning models capable of identifying cars, pedestrian crossings, traffic signals, and road signs, as well as analyzing pedestrian behavior based on available data. By using these technologies, our project aims to create a structured data frame where we can label the actual driver behavior. This data frame can then be used to train a neural network model to mimic the behavior of a human driver. The subsequent technologies in computer vision and machine learning were employed to generate the data frame in this project.

A. Car Detection Using HOG Features and SVM

In this section of our research paper, we employed the KITTI car data set to identify cars in a frame. To perform car detection, we utilized HOG Feature extraction, a feature selection technique that computes a histogram for each region of the image based on the orientation and gradient of its pixel values. The name "Histogram of Oriented Gradients" reflects this approach. Implementing HOG with the OpenCV library is a straightforward process.

We employed [1] to create an SVM classifier that utilizes grid search to identify cars and trucks in successive iterations and enclose them in a bounding box.

B. Lane Detection using Hough transform

The Hough transform [2] is a method commonly employed in image analysis, digital image processing, and computer vision to extract features. Its objective is to identify instances of imperfect objects within a specific class of shapes using a voting approach. The voting process occurs in parameter space, and the algorithm generates an accumulator space to obtain local maxima as object candidates.

To identify lanes in an image frame, we utilized the Hough transform. This output will constitute one of the columns in the self-driving data frame.

C. Road Signs Classification

We used a Kaggle dataset (reference [3]) to classify common road signs into four different classes using the VGG16 transfer learning model. Additionally, we attempted to place bounding boxes around the identified road signs by utilizing the provided dataset annotations.

D. Creating a self-driving dataframe

Using the aforementioned technologies, we generated a data frame that can be utilized to train a machine-learning model. This model can then predict decisions at different points in time, similar to how humans do.

II. Related Work

Inspired by the Histograms of oriented gradients (HOG) approach for human detection, as described in reference [1], we developed a similar algorithm to detect cars in the test dataset. We also incorporated a bounding box around the identified cars. Our algorithm utilized binary data to determine whether a car was present or not, and this information was incorporated into a corresponding data frame for each frame extracted from the test data. In a similar manner as inspired by reference [2] that used Hough transform, we applied it to detect lanes using the Canny Edge detector as described in reference [4]. Specifically, we developed a lane detection algorithm that could identify yellow and white lines. We chose to use the conventional slope-intercept form instead of the more advanced angle-radius form described in reference [5]. Our lane detection algorithm was incorporated into our self-driving data frame. Using the capabilities of the VGG16 model as described in reference [8], we were able to classify the images and add bounding boxes around objects in the dataset referenced in [3]. This information was then added as a significant column to our autonomous driving dataframe.

III. Dataset

To detect cars in our study, we utilized the KITTI car dataset as referenced in [6]. We employed the dataset referenced in [3] to detect road signs. In order to test our algorithm, we used a section of a YouTube video as described in [8].

IV. Problem Statement and Solution

Despite the availability of abundant data sources and highly accurate computer vision tools, fully autonomous driving remains a distant goal. This is because the human element of driving is crucial, and the human brain's ability to make intuitive decisions is unmatched. The auto industry faces the challenging task of replicating human intuition and decision-making ability in autonomous vehicles, particularly in complex or unpredictable situations. Finding ways to solve this challenge is a major focus for the industry.

Consider a scenario where a car needs to stop at a signal. Vectorizing this scenario and making a decision seems simple: Vector=(Road-sign= 'Stop', Car Speed=20mph, output: Apply Break). However, this scenario becomes more complex when a speeding car is approaching from behind, and there is a possibility of a crash if the car stops suddenly. The decision vector becomes complicated for the machine, and if we add another car crossing the intersection, the situation becomes even more challenging. The vector in this case would be Vector=(Road-sign= 'Stop', Car-Speed='20'mph,Rear car speed='50'mph, rear car velocity differential='30'mph, speed differential= '30'mph, rear crash probability=0.6, front crash probability=0.5, output: '?'). Humans rely on intuition to make decisions in such situations, but machines lack this ability. However, if we collect human responses to such scenarios and train a machine-learning model to make decisions in the most humane way possible, we can incorporate human intuition into the machine.

It can be observed that we possess adequate computer vision technology to anticipate different situations, however, we fall short in our decision-making ability. By gathering the necessary data and optimizing the decisions based on humane responses, we can develop an autonomous self-driving system. The generation of our data frame is a small but important step towards this goal.

V. Experiments and Evaluation

A. Car Detection

We developed an algorithm for car detection in images, using [1] as a source of inspiration and the [2] dataset. Our tests showed that the algorithm successfully identified the majority of cars and trucks in the test data.

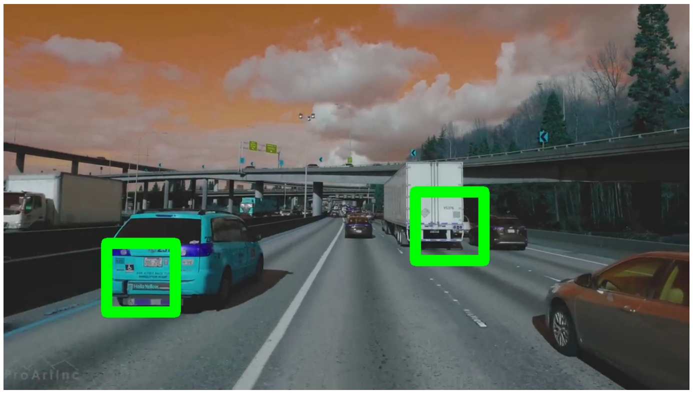

Fig. 1: Car detection by SVM classifier and HOG feature

Even though the HoG features proved effective in identifying cars, our dataset primarily consists of car images captured from the rear. Therefore, we encountered difficulty detecting cars that were only partially visible from the left or right.

Fig. 2: Truck detected from back

Fig. 3: Truck not detected in right

To address this issue, one potential solution would be to augment the training dataset with more images of cars captured from side angles. This could help the algorithm better recognize and detect cars that are only partially visible from the left or right.

B. Lane Detection

To detect lanes in our project, we employed a combination of Canny edge detection and Hough transform methods. In order to accurately detect both yellow and white lanes, we fine-tuned the threshold parameters. We discovered that the lane markings in our test data were not completely white, so we utilized an RGB color range of (130,130,130) to (255,255,255) for detecting white lanes. Similarly, we used the color range of (130,130,0) to (255,255,0) to detect yellow lanes.

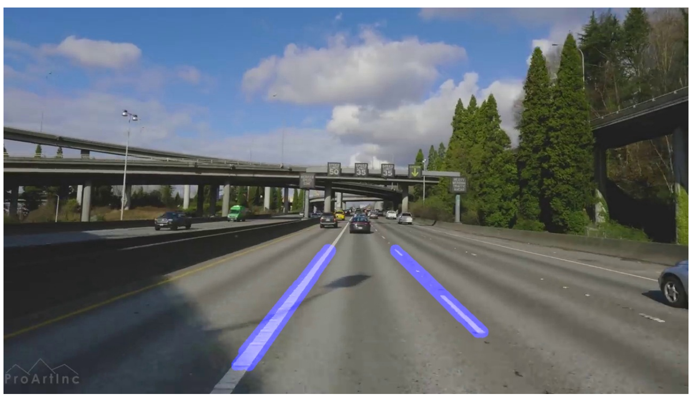

Fig. 4: Lane Detection

However, similar to the car detection scenario, we encountered two issues in this case as well. Firstly, our algorithm is not able to identify lanes in areas with insufficient light, despite lowering the white band threshold. Secondly, in areas with intense sunlight, the central white section of the road appears too bright, resulting in false positive errors.

Fig. 5: False Negatives

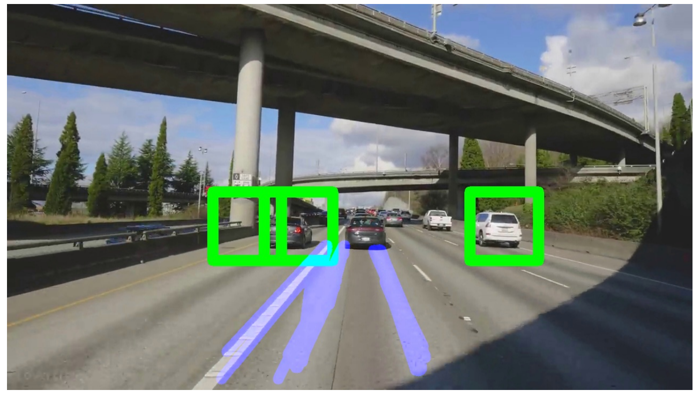

Fig. 6: False Positives

Yes, color selection threshold was chosen that represented a compromise between the two issues mentioned earlier.

C. Road Sign Classification

In this study, we utilized a dataset from [3] comprising 877 images of road signs along with corresponding annotations, which could be categorized into four classes: Traffic Light, Stop, Speed Limit, and Crosswalk. We employed a transfer learning model, specifically the VGG 16, to train our data. Additionally, we used the same model to create a bounding box around the road signs and predict both the class and the bounding box. Nonetheless, the primary challenge we faced was the limited training dataset and the absence of prominent road signs in our dataset, except for traffic lights. To assess the model's performance, we randomly selected images from Google and validated them using our trained model.

Fig. 7: Testing model for randomly selected image

D. Self Driving DataFrame Creation

To construct a dataset for self-driving, we utilized five columns. The initial column comprises frame names, the second column contains cars identified in the frame using HOG and SVM, and the third column includes binary data indicating whether lanes were detected. Since we lacked road signs apart from traffic lights, we generated a random number between 0 to 3. The final column is the most crucial as it serves as the label for our model, consisting of four categories: Accelerate, Break, Left, and Right. These actions need to be recorded based on the driver's response to the situation. This data frame can be used to train a machine-learning model.

	Image_path	No_Cars	lanes	Road_sign	Output
0	frame0.jpg	0	1	0	0
1	frame1.jpg	0	1	3	1
2	frame10.jpg	2	1	3	2
3	frame100.jpg	1	1	1	1
4	frame1000.jpg	2	1	2	3
...	...	...	...	...	...
8987	frame995.jpg	1	1	1	3
8988	frame996.jpg	1	1	1	1
8989	frame997.jpg	1	1	3	0
8990	frame998.jpg	0	1	0	3
8991	frame999.jpg	1	1	2	1

Fig. 8: Snap of dataframe created.

VI. Applications and Future scope

- The primary significance of the newly created dataframe is that, with the aid of modern technology and sensors, a dataframe that includes more output classes can serve as a valuable source of information for decision-making in autonomous cars.
- Just like a collaborative recommender system, a driver's behavior can be classified and utilized by various entities such as insurance companies, car sales teams, and others. In this case, the sparsity of data is lower compared to a movie recommender system.
- Using this dataframe along with recommender system techniques and machine learning models can be employed in autonomous cars to categorize them and customize their design based on the specific behavior and preferences of the customers.

VII. Conclusion

The autonomous driving technology currently lacks human intuition. However, if we can incorporate this intuition into the machines, it can revolutionize the way cars are designed and operate on the roads. Our project is an initial step towards achieving this goal.

References

[1] Dalal,N., Triggs, B.(2005). Histograms of oriented gradients for human detection. In 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

[2] Duda, R. O., Hart, P. E. (1972). "Use of the Hough transformation to detect lines and curves in pictures". Communications of the ACM, 15(1), 11-15

[3] Make ML. Road Signs Dataset. Retrieved from https://makeml.app/datasets/road-signs.

[4] J. Canny, "A Computational Approach to Edge Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, no. 6, pp. 679-698, Nov. 1986, doi: 10.1109/TPAMI.1986.4767851.

[5] Duda, R. O., Hart, P. E. (1972). Use of the Hough transformation to detect lines and curves in pictures. Communications of the ACM, 15(1), 11-15. doi: 10.1145/361237.361242.

[6] Geiger, A., Lenz, P., Stiller, C., Urtasun, R. (2013). Vision meets Robotics: The KITTI Dataset. The International Journal of Robotics Research, 32(11), 1231-1237. doi: 10.1177/0278364913491297.

[7] Google(2021).Colaboratory.https://research.google.com/colaboratory/faq.html

[8] 4K Urban Life.(Oct 12, 2018).4K Seattle Streets - Car Driving Relax Video - Washington State, USA.https://www.youtube.com/watch?v=fkps18H3SXYab$_{c}hannel = 4KUrbanLife$.