Machine Vision for Safer Self-Driving Cars

Rick Alayza, Trincy Thomas Kozhikkadan, Krystofer Newman

Abstract

Advancement in technologies used in self-driving cars may allow for machine vision to surpass LIDAR as the standard for area mapping. Unfortunate accidents in testing has raised the concern for vehicle safety. Machine vision and neural networks require training but the level of training has an impact on classification performance. The first step to understand machine vision is to understand the convolutional neural network. This network was trained to detect stop signs in random images. The experiment to implement this stop sign detector was successful on several test images.

***Keywords* — Convolutional Neural Networks, LIDAR, Machine Learning, self-driving cars**

I. Introduction

THE research and development in the area of self-driving cars has seen a rapid growth in the recent years. Self-driving technology has promised several benefits like improved road safety since autonomous systems, in contrast to human drivers, have faster reaction times and are fatigue-proof in their functioning, a cooperative approach between vehicles enables to find alternate or shorter routes to avoid traffic congestion to save time and fuel and they will also facilitate mobility for the handicapped and elderly. Many big companies like Waymo and Uber are currently working on the the research and development of self-driving cars. The introduction of fully autonomous vehicles poses a number of concerns regarding the safety and dependability of vehicle operation. Self-driving cars still struggle when faced with variable environments or unpredictable situations. In March 2018, a pedestrian was killed by a self-driving Uber vehicle while crossing a street in Tempe, Arizona and in the same month the driver of a Tesla Model X was killed while the vehicle's autonomous autopilot mode was active[1][2]. There have been other similar incidents where the autonomous test vehicles have failed to function accurately, and this raised doubts over the accuracy and thus, the safety of the self-driving cars. Self-driving cars should be highly accurate systems having zero tolerance for errors to be fully functional and safe for the roads. This research paper discusses some aspects of the existing LIDAR (Light Detection and Ranging) technology and its limitations and how it can be integrated with machine vision and convolutional neural networks to improve the accuracy and safety standards of self-driving cars which is currently a significant challenge faced by the autonomous vehicle industry.

1

II. Current Technology Standards

A. *LIDAR (Light Detection and Ranging)*

Detecting objects reliably is a crucial part of fully autonomous driving, especially for anticipating the actions of other road users. LIDAR (Light Detection and Ranging) sensors have been used for many years in autonomous test vehicles for obstacle detection. LIDAR (Light Detection and Ranging) sensors work by transmitting a pulsed beam of light reflected from a rotating mirror and waiting for the reflected signal[3]. The delay associated with reception of the reflected signal can be used to find the distance of the reflecting object[3]. They can detect objects with much greater resolution than radar or acoustic sensors due to the small wavelengths employed. The LIDAR (Light Detection and Ranging) sensor data is paired with information from optical cameras, radar,GPS and other sensors to create a complete picture[4]. LIDARs (Light Detection and Ranging) can fail to operate accurately in adverse conditions like fog, rain, dust, and snow[5]. LIDAR (Light Detection and Ranging) output can be degraded during situations like stormy weather or changes in lighting[5]. The limitations can be overcome by combining LIDARs (Light Detection and Ranging) with machine vision. Machine vision technology provides automatic image capturing and helps in identifying still or moving objects[6]. Machine vision can provide accurate results in determining lane geometry, drivable road segments, traffic signs, pedestrians and other obstacles that a self-driving car may come across[6].This technology will enable accurate decision making to ensure safer and more reliable autonomous systems. Machine vision is the future of self-driving cars and has the possibility of completely replacing the existing LIDAR (Light Detection and Ranging) technology.

B. Convolutional Neural Network

Processing large quantities of data in real time with variable environments is no trivial task, especially in situations where lives may be at stake. Just one Intel self-driving vehicle can generate and consume around 40 terabytes of data in an eight-hour span. CNNs (Convolutional Neural Networks) make it possible for these vehicles to process this vast amount of data in real time [7]. CNNs are typically used for image recognition and was first introduced to classify digits in handwriting by Bengio, Le Cun, Bottou and Haffner [8]. Today CNNs are used widely in Computer Vision and AI fields including self-driving cars and have evolved rapidly over the last decade. These networks are generally comprised of multiple layers including an input and output layer. The layers may also include pooling layers to combine outputs of neuron clusters into a single entity neuron for the next processing layer [9]. Figure 1 shows a general architecture of a Convolutional Neural Network that may be implemented in a self-driving car. Architectures and implementations can vary for each CNN depending on hardware and targeted classifications.

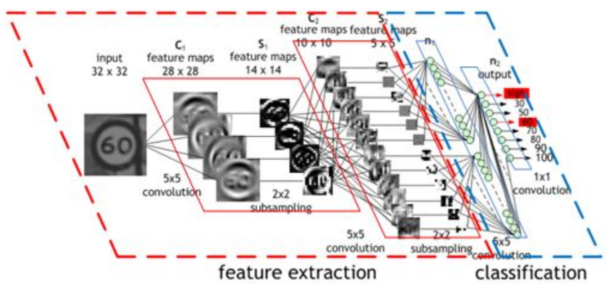

Figure 1 Example CNN Image: Udacity

In this Figure the CNN is attempting to classify an image of a street sign with the number ‘60’.

III. POSSIBLE REPLACEMENTS TO LIDAR

Although it seems that almost every large technology company involved with autonomous vehicles uses LIDAR systems, few companies believe there may be better solutions to reach full autonomy. Due to the popularity of LIDAR, other technologies may be slower to evolve in the industry. Tesla is one major company who equips their self-driving vehicles with cameras, radar and ultrasonic sensors as an alternative to LIDAR [10]. Uber also uses cameras and radar systems in addition to LIDAR, however both companies have had fatal accidents and faced public backlash for unsafe operation or malfunctions in the vehicles as mentioned before so it is unclear which is more effective [11]. Figure 2 shows a recent analysis by Navigant of rankings among different companies in the self-driving car industry [12]. All companies listed in the Leaders section in the analysis of Figure 2 all use LIDAR, cameras, and sensors in their autonomous systems. This analysis also shows Tesla last in execution and just ahead of Apple in Strategy of deploying their vehicles which shows that LIDAR technologies may have a large impact in autonomy moving forward. With these technologies evolving as fast as they do, it may be quite some time before the public sees an emerging replacement or event supplement that could replace the LIDAR systems due to popularity, efficiency, and safety regulations of current systems.

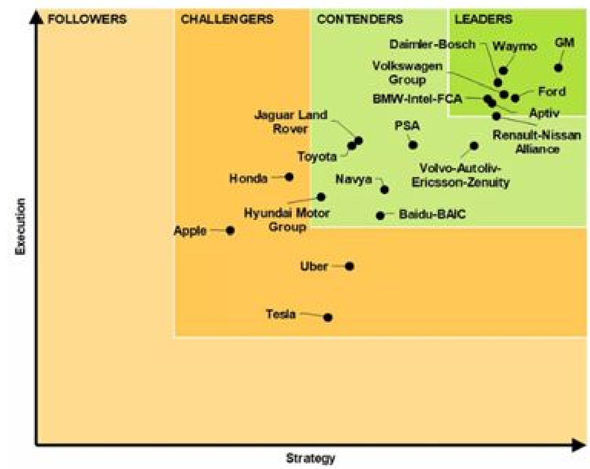

Figure 2 Analysis of Market Image: Naviga

Other companies such as Intel are using onboard map data in addition to LIDAR to provide the cameras textural data for the surrounding environment [13]. Figure 3 shows just how many different systems a self driving car can have.

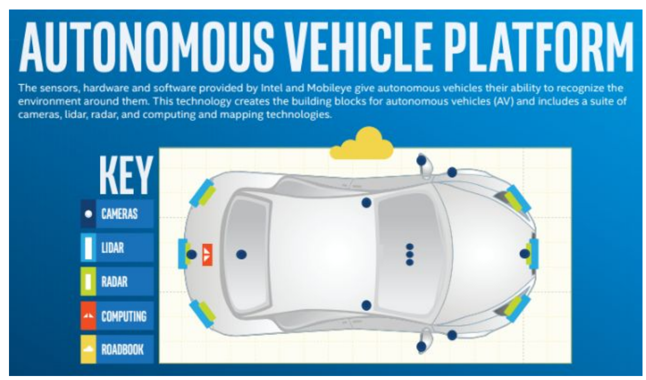

Figure 3 Vehicle Platform Image: Intel

IV. EXPERIMENT

The experiment consisted of implementing a convolutional neural network that was trained to detect objects. This experiment was part of tutorial created by MathWorks® for MATLAB™ [19]. The tutorial “Object Detection Using Deep Learning” was designed to create a neural network from a database of 50,000 images. This section will explain the experiment with regards to setup, training, and results.

A. Setup

This section will provide an abbreviated setup process for the neural network. More specific instructions are listed in the associated website [19]. The first step was to install the appropriate toolboxes MATLAB would need to create and train the convolution neural network. Neural Network, Statistics and Machine Learning, Computer Vision System, and Image Processing Toolboxes were required for the experiment. The Parallel Computing Toolbox was option since this toolbox only lowers the processing time during training. Along with the toolboxes, a dataset of 50,000 images as loaded into the workspace. In older version of MATLAB, this dataset was downloaded from a repository but comes bundled with Image Processing Toolbox in the current version. The sample dataset consisted of 10 categories. Figure 4 was a sample of the dataset generated using the supplied MATLAB commands in the tutorial. It is important to note that the images in Figure 4 were relatively small and had a low resolution. This was intentional because larger images at a higher resolution would significantly increase training times and be beyond the scope of the tutorial. The next step was to create the convolutional neural network (CNN). The Neural Network Toolbox simplifies this process since the CNN is a layered network. This experiment used 15 layers. Image input layer, 2D convolution layer, and classification output layer were some layer examples. The commands listed in the tutorial can be repeated with downsampling to create a deeper network. This downsampling may impact training if important or useful information is discarded. This specific experiment did not repeat the layering commands. A single execution was used to create the CNN.

Figure 4: Image Dataset Sample

B. Training

Once the neural network was created, it was trained. The training consisted of the CNN scanning the image, formulating associations, and attempting classifications. Those attempted classifications were then compared to the classifications in the dataset to calculate the CNN accuracy. This training required 40 epochs, or scan the entire dataset 40 times. Along with the epochs, the training required iterations which was scanning each image several times. The tutorial offered two options: to load training data or to execute the training. Loading the training data was not as accurate in object detection but required less processing time. Executed training averaged around 45-50 minutes. The training was separated into two different runs. The first training run was performed on the loaded dataset. Figure 5 was the output of the first training. The accuracy of the CNN versus the epoch. This training performed 15,600 iterations and lasted 55 minutes. The average accuracy achieved was 85%. This percentage can be increased with a deeper network though more images or repeated trainings. To correctly ensure that the CNN was created/trained correctly, a validation was executed. Figure 6 was a sample output of the first layer in the CNN. The first layer filter samples have started to acknowledge edges.

The next training would now focus on the object to be detected. In this instance, the object was a stop sign. The dataset consisted of 41 images. This is not ideal but sufficient for the scope of this tutorial. This training also decreased the classification list from 10 to 2, stop sign or background. Figure 7 was a sample of the training dataset. The region of interest (ROI) was highlighted to train the network to detect stop signs. That ROI was changed this neural network to a R-CNN, regions with convolutional neural network. A training procedure was executed in a similar fashion as the initial training. Unfortunately, a visual representation of the training accuracy was not generated due to the re-classification in MATLAB.

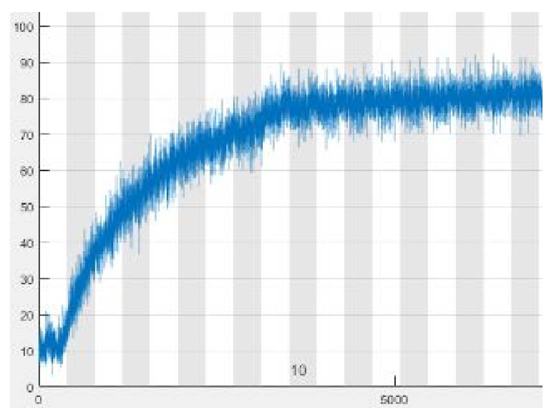

Figure 5: Accuracy (%) vs. Iteration Training Graph

Figure 6: First Training Validation Sample

Figure 7: Stop Sign Training – Region of Interest (ROI)

C. Results

The final step was to test the accuracy of the R-CNN against an image that was not part of the dataset. Several images were downloaded and tested. Out of those, 5 images were selected and displayed in Figure 8. Starting from the top left corner of Figure 8, this image was considered a standard image of a stop sign. The sign was easy to read and recognize at first glance. The detection process placed a yellow box to represent the ROI with an identifier and a percent confidence in decimal format. Image 1 returned a confidence of 99.99997%. This was to be expected. Image 2 was the stop sign covered by a tree branch. The R-CNN was not able to detect a stop sign and only outputted a null set which is the image without the ROI highlighted. Image 3 (middle) was used to determine if the R-CNN could detect multiple stop signs. From the results, it was concluded that the R-CNN was programmed to detect the first stop sign and generate an output. This could be corrected in later revisions. Image 4 was a stop sign that was at an off angle to the picture. The R-CNN detector was able to identify a stop sign in the image with a confidence of 100%. To better understand and display the boundaries of the R-CNN stop sign detector, more images of partially concealed signs were tested. Image 5 was one example. The CNN detector was able to recognize the stop sign with a confidence of 82.9%. The difference between the two concealed representations was that the letters are more apparent in the Image 5. The unexpected result was the region of interest (ROI) did not highlight the whole sign. More analysis of the program will be required to fully explain this result. As for the scope of this project, the R-CNN stop sign detector functioned correctly and had satisfactory results for simple tasks.

Figure 8: CNN Stop Sign Detection Results

V. Conclusion

As more and more companies compete to produce a self-driving car, vehicle safety has become the key concern. The majority of companies rely on LIDAR, an existing and tested technology, to map a car's surroundings. The minority of those competing have selected cameras and machine vision to map a car's surroundings. Machine vision utilizes optics and a neural network created by machine learning to classify the incoming images. That neural network is constructed through training. The more specific and longer the training, the more accurate the network becomes at identifying and classifying. Machine vision is relatively new compared to LIDAR but shows great potential. The convolutional neural network, just one example of neural networks was specifically designed for image classification. Current development in autonomous driving utilize both LIDAR and machine vision. The LIDAR would map all the objects and the machine vision would be tasked with generating classifications. LIDAR has a distinct advantage over machine vision and that is the ability to determine distance. Currently, machine vision is advancing to the point of being able to determine object distance in lab experiments. When that becomes tested and true, machine vision could replace LIDAR in the technology that helps a self-driving car to see and drive safely.

VI. References

[1] Troy Griggs and Daisuke Wakabayashi. How a Self-Driving Uber Killed a Pedestrian in Arizona.Retrieved from https://www.nytimes.com/interactive/2018/03/20/us/self-driving-uber-pedestrian-killed.html

[2] Tesla Model X sped up in Autopilot mode seconds before fatal crash, according to NTSB. Retrieved from https://techcrunch.com/story/tesla-model-x-fatal-crash-investigation/

[3] Suliman Gargoum and Karim El-Basyouny. Automated Extraction of Road Features using LiDAR Data: A Review of LiDAR applications in Transportation .In 2017 4th International Conference on Transportation Information and Safety (ICTIS)

[4] Tim Menke. Self-driving Cars: The technology, risks and possibilities.Retrieved from http://sitn.hms.harvard.edu/flash/2017/self-driving-cars-technology-risks-possibilities

[5] Tom Simonite. MIT Technology Review: Self-Driving Cars' Spinning-Laser Problem. Retrieved from https://www.technologyreview.com/s/603885/autonomous-cars-lidar-sensors

[6] Benjamin Ranft and Christoph Stiller.The Role of Machine Vision for Intelligent Vehicles.IEEE Transactions on Intelligent Vehicles.Volume: 1 Issue: 1

[7] For Self-Driving Cars, There's Big Meaning Behind One Big Number: 4 Terabytes. (2017, April 14). Retrieved August 30, 2018, from https://newsroom.intel.com/editorials/self-driving-cars-big-meaning-behind-one-number-4-terabytes

[8] Lecun, Y., Haffner, P., Bottou, L., & Bengio, Y. (1999). Object Recognition with Gradient-Based Learning. Shape, Contour and Grouping in Computer Vision Lecture Notes in Computer Science, 319-345. doi:10.1007/3-540-46805-6_19

[9] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60(6). doi:10.1145/3065386

[10] Hawkins, A. J. (2018, February 07). Elon Musk still doesn't think LIDAR is necessary for fully driverless cars. Retrieved from https://www.theverge.com/2018/2/7/16988628/elon-musk-lidar-self-driving-car-tesla

[11] Wakabayashi, D. (2018, March 19). Self-Driving Uber Car Kills Pedestrian in Arizona, Where Robots Roam. Retrieved from https://www.nytimes.com/2018/03/19/technology/uber-driverless-fatality.html

[12] Navigant Research Leaderboard: Automated Driving Vehicles. (2018, January 14). Retrieved from https://www.navigantresearch.com/reports/navigant-research-leaderboard-automated-driving-vehicles

[13] R. (2018, May 30). Mapbox partners Microsoft, Intel to provide self-driving car maps. Retrieved from https://www.businessinsider.com/r-mapbox-partners-microsoft-intel-to-provide-self-driving-car-maps-2018-5

[14] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.

[15] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces (Translation Journals style)," *IEEE Transl. J. Magn.Jpn.*, vol. 2, Aug. 1987, pp. 740–741 [*Dig. 9th Annu. Conf. Magnetics* Japan, 1982, p. 301].

[16] M. Young, *The Techincal Writers Handbook.* Mill Valley, CA: University Science, 1989.

[17] J. U. Duncombe, "Infrared navigation—Part I: An assessment of feasibility (Periodical style)," *IEEE Trans. Electron Devices*, vol. ED-11, pp. 34–39, Jan. 1959.

[18] S. Chen, B. Mulgrew, and P. M. Grant, "A clustering technique for digital communications channel equalization using radial basis function networks," *IEEE Trans. Neural Networks*, vol. 4, pp. 570–578, Jul. 1993.

[19] Mathworks, "Object Detection Using Deep Learning," MATLAB, 2018. [Online]. Available: https://www.mathworks.com/help/vision/examples/object-detection-using-deep-learning.html#mw_4dc10f01-e0dc-4970-8e4a-28817f13198e. [Accessed 22 August 2018].