ACCURATE IDENTIFICATION OF TRAFFIC SIGNS USING RADAR AND CAMERA FUSION

An Undergraduate Research Scholars Thesis

by

SHREY SHAH

Submitted to the Undergraduate Research Scholars program at
Texas A&M University
in partial fulfillment of the requirements for the designation as an

UNDERGRADUATE RESEARCH SCHOLAR

Approved by Research Advisor: Dr. Dezhen Song

May 2020

Major: Computer Engineering

TABLE OF CONTENTS

ABSTRACT

Accurate Identification of Traffic Signs Using Radar and Camera Fusion

Shrey Shah
Department of Computer Science and Engineering
Texas A&M University

Research Advisor: Dr. Dezhen Song
Department of Computer Science and Engineering
Texas A&M University

Self-driving cars are no doubt the future of commuting for the world and it is paramount to make them as safe as possible on the road. The paper will cover a start to end process of making a system to easily collect data from the radar and camera and then using algorithms on the data collected to reduce anomalies and detect objects with better accuracy.

Radar and camera both act as a data input for self-driving cars and are extremely important for the safety of both passengers and pedestrians, however, both of these sensors can be easily fooled. The advantage here is that what deceives one of the sensors doesn't always mislead the other one, hence, using the suitable traits of each to overcome the deficiencies of the other will make them more robust and less susceptible to be fooled by such anomalies.

By the end of this paper, the reader will have an in-depth understanding of how data is taken in, manipulated, and converted into results that power a self-driving car.

NOMENCLATURE

ROS	Robotic Operating System
I/O	Input / Output
B/W	Black and White
OCR	Optical Character Recognition
YOLO	You Only Look Once
RCNN	Regional Convolution Neural Network
DPM	Dimension-based Partitioning and Merging clustering
CUDA	Compute Unified Device Architecture
OpenCV	Open-source Computer Vision
HPRC	High Performance Research Computing
TAMU	Texas A&M University

CHAPTER I

INTRODUCTION

Self-driving cars must be able to detect objects, classify them as vehicles, people, traffic, signs, or obstacles and then take appropriate actions all while abiding by traffic rules in all possible environmental conditions. A major part of this is being able to correctly detect and classify objects accurately in a limited time frame. This requires a good calibration of sensors and failsafe methods [1].

Sensors

This thesis will mostly revolve around Radar and Camera as sensors for the self-driving car.

Radar

Radar is a sensor that detects objects' positions using high-frequency electromagnetic waves. In an autonomous vehicle, a radar will be used to detect if an object is present in the environment and if there is one, the distance, heading, and relative radial velocity of the object can be acquired as well. By applying a series of algorithms, we can detect objects and analyzing those objects over time will allow us to classify the object.

Radars are good at detecting the position of objects, but they cannot fully classify an object, which may be necessary to fully decide the next course of action. For example: A radar detects an object at some distance x. Now, this can be a traffic sign or a board for some advertisement or information or it may just be a huge candy poster. A radar will give only so much information to classify it as an object at x distance from the car.

Camera

In order to fill that gap of being able to classify the object self-driving cars typically utilize a camera module.

Cameras are really good at registering colors and patterns. Unlike radars, they can detect objects and the details of the objects very well, but they cannot satisfactorily detect the depth and scale of the objects. For example: A camera can detect that the object that the radar detected earlier, was indeed a traffic sign indicating a detour ahead. Now, using Optical Character Recognition and Natural Language Processing [2], it can say that the detour is 250 meters away from the sign, but it cannot tell the distance from itself to the sign with similar accuracy as that of a radar.

Cameras, unlike radars, are also susceptible to harsh environmental conditions. They can easily be fooled by rain, fog, and lighting conditions, as they focus on optical feedback. Moreover, processing of camera data can take longer times and are not always perfect due to still-developing machine learning algorithms. Hence, they cannot always be relied upon in times of emergency and one might resort to primarily using radar data in certain conditions.

Calibration

Calibration is a significant part of the processing of a self-driving car. It enables algorithms to process data with higher accuracy yielding better predictions and safer self-driving cars. Calibration can be broken down into two major steps: pre-processing and on-board processing. Pre-processing would include calibration done before the car is sent out in the real world. This would involve steps like functioning calibration of a camera, color dynamics of a camera and elements like this which only require a one-time setup. On-board processing means the small calibration tweaks needed when the car starts and while on the trip. These include radar noise-reduction and camera mode switch (sometimes B/W images are needed for OCR) [2]. Now, these calibrations are key to safe self-driving cars.

Radar and Camera Calibration

Radar and Cameras are calibrated separately via intrinsic calibration like camera focus and color dynamics. After they are optimally intrinsically calibrated, the extrinsic calibration of the sensors, the rigid body transformation between them, is computed. This allows one to map the measurements of one sensor in its coordinate frame into the frame of the other.

The calibration can be done via various methods. One such method being the use of MATLAB's camera calibration tool to determine the camera intrinsic calibration and corner reflectors to come up with a rigid body transform of the camera to the radar [7]. Another way is the use of various predeveloped tool kits like Apollo and the extrinsic multi-sensor calibration kit [1] [4] [5] [6].

CHAPTER II

METHODS

There are multiple methods of using radar and camera as a pair. Multiple papers show different ways of using each sensor individually to detect objects as well [3][4][5][6]. The advantage of fusing the sensors here is what deceives one of the sensors doesn't always mislead the other one, hence, using the properties of one to overcome the deficiencies of the other will make them more robust and less susceptible to be fooled by such anomalies.

Detection and Recognition

Object detection, a crucial factor of making decisions in a self-driving vehicle, must be both fast and accurate. To achieve this, each of the sensors must produce a result individually according to their strengths. The camera is very good at classifying the objects it captures. The radar is better at measuring object distance and mapping the surroundings. Hence, to get the best results, each of the sensors should only focus on those parts of the process. We must then fuse these results.

Camera

In order to demonstrate an aspect of object recognition with cameras, this paper focuses on detecting traffic signs in the US via the LISA dataset. The LISA Traffic Sign Dataset is a set of videos and annotated frames containing US traffic signs [8]. Now that the dataset is established, a framework to train the model to detect the traffic signs is required. Multiple detection models like RCNN, Fast RCNN, Faster RCNN, DPM, Poselets, YOLO and many more exist that train on datasets in different ways. To select the best one, the goals and the features of the self-driving vehicle, as well as the features of the dataset, must be kept in mind.

After training, the camera module should be fast and fairly accurate. A traffic sign is made to be a big object and usually very distinct in colors compared to its surroundings in order to make it easy to be seen by human eyes. Keeping these factors in mind, YOLO seems to be the best choice. YOLO – You Only Look Once, is a training model that focuses on detecting objects in images in a new way [9]. As the name suggests, it makes the image go through only one convolution neural network and that makes it much faster than any other algorithms in the industry [9]. YOLO is fairly accurate at the given speed for objects that are large enough to detect in the images and are not very crowded by similar objects [9] [10] [11]. This falls perfectly under the restrictions and goals stated earlier.

For training the model, a framework called Darknet was used. Darknet is a framework that trains the YOLO models based on the images using libraries like CUDA and OpenCV. The framework offers a choice of picking the number of classes, filters, subdivisions, and batches to better train the model based on the needs of the consumer. For the classes, the number to be picked is the number of categories of images. In the LISA dataset, there are 47 different signs. So, the classes were set to 47. Filters are a derived value of classes and generally go by the formula (classes + 5) * 5 which gave the value 260 [9]. The subdivisions and batches decide the number of images that are to be processed in parallel. To get a number that suits the needs of this specific research on these, it was decided to try out the following combinations – 1-1, 8-1, 8-8, 64-8.

Each of the former combinations had a preliminary training time of 10 hours on Texas A&M University's High Performance Research Computing (HPRC) center's V100 GPU on the Ada Cluster. After the preliminary training, the output weights were tested on 3 different images.

The first one from the dataset. The second one was also from the dataset but had merged images of 3 signs in it. The third one was from Google Images.

The results of the test were indicative of what subdivision and batch numbers should be used. The 1-1 combination failed to identify the first image and was eliminated. The 8-1 combination identified the first image and the second with a few inaccuracies but couldn't identify the third image. The 8-8 combination was able to identify the first and the second one well but failed on identifying one of the signs in the third image. The 64-8 combination provided a much better result. Although it made a similar error on the third image as the 8-8, it was faster and the pixel marking on the right image structures was much tighter and cleaner than the 8-8 model [Figure 1]. Moreover, the 64-8 model gave a 7 to 8% higher confidence rate on each sign.

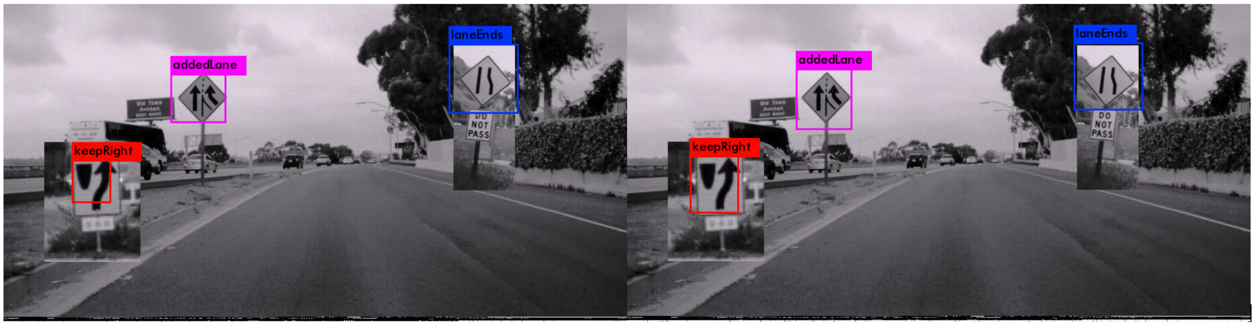

Figure 1a. 8 subdivisions 8 batches rendered output 1b. 64 subdivisions 8 batches rendered output

Radar

With the camera for object recognition in place, the radar is used for object detection. In this scenario, the radar is used for better spatial understating of the sign now that the vehicle can recognize what the sign is. Since the radar gives data based on if there is an object at a place or not based in just 2 dimensions, it is really hard to differentiate between objects with similar shapes and densities. For example, a traffic light pole and a sign pole might look the same in the radar visualization of the area. Moreover, readings from the trees around the signs and other objects near it make it almost undetectable to the radar sometimes.

To avoid this situation from happening and to make the detection better, a few additional characteristics are to be looked at. One of the major ones is curbside visualization. In layman's terms one can add external human knowledge to the radar system's process regarding traffic signs that if the signs are not visualized well, many times, the sign is by a curbside and more towards the end of a curbside. This helps the radar system narrow its search down and find a potential position for the sign or in most cases actually locate the sign. Figure 2 below shows a depiction of how it becomes easier for the system to detect a stop sign based on the curbside.

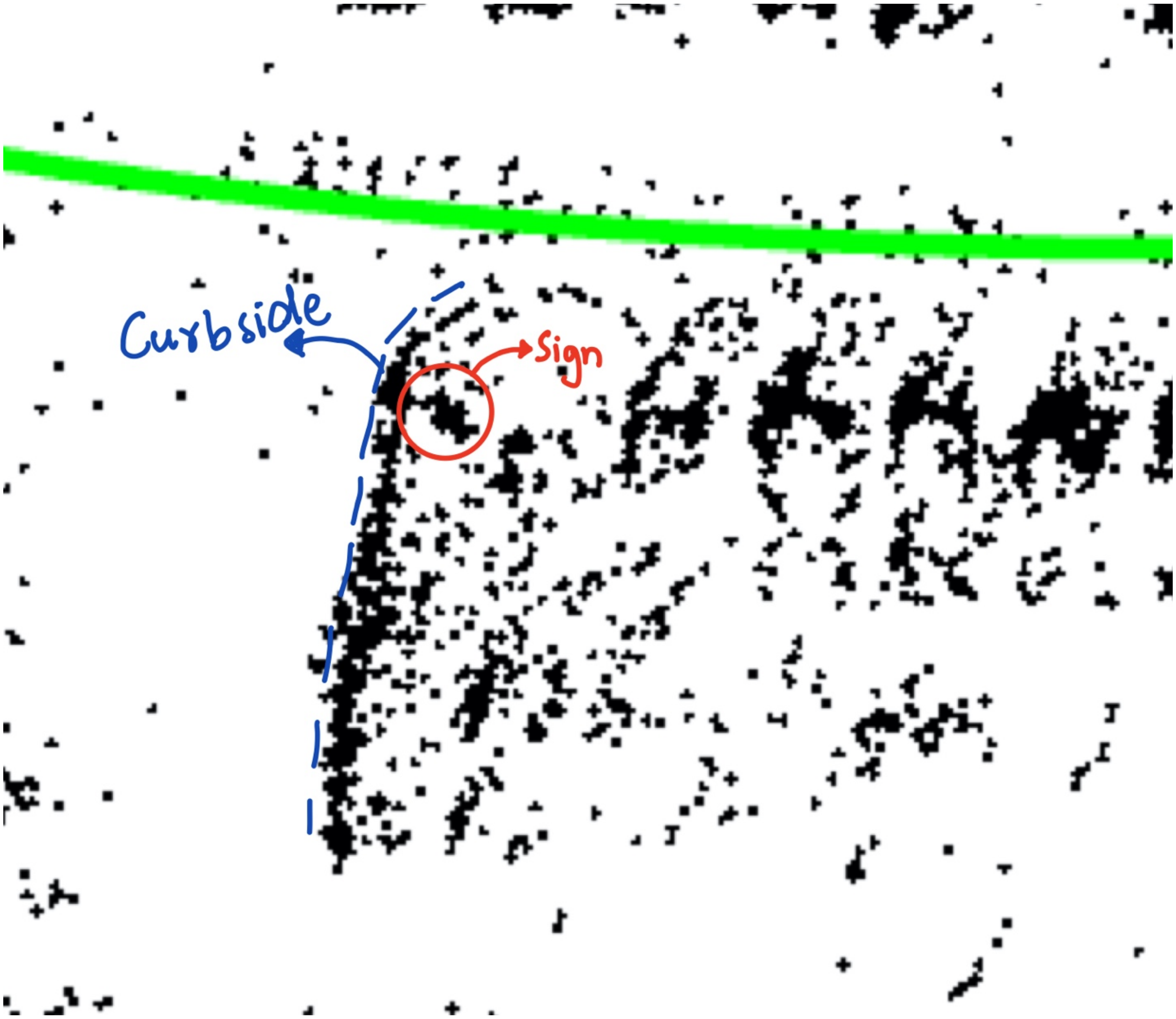

Figure 2. Radar map capture depicting curbside and a traffic sign

Another useful tactic is based on High Definition (HD) maps. This method is used by multiple autonomous vehicle testing today due to its simplicity and effectiveness[12]. In this method, the autonomous vehicle is preloaded with GPS data about traffic signs on the road to create a prediction before trying to detect and verify the sign to get its accurate position. This method works the fastest in conditions where there is a pre-installed and collated permanent traffic sign stored in the GPS. However, the use of makeshift signs on roads is very common in cases like construction or accidents or national emergency situations. These cannot be mapped as they are variable and short-lived. However, these are probably the most important signs to detect as their main purpose is to avoid accidents in places that are already chaotic. Hence, these cannot be ignored.

Radar Camera Fusion

Radar and Camera are two very powerful sensors as mentioned above, however, both present some serious flaws like not being able to rectify the spatial location of the objects in the camera's case and not being able to recognize the object detected in the radar's case. However, with the fusion of the two, the recognition and detection model looks promising.

The way we went about developing the fusion is by combining the strengths of each and aligning them in a way to reduce the time taken while increasing the accuracy. Initially, the radar and camera have a rigid body transformation structure between them. What this means is that there is a formula that transforms measurements between the radar's and the camera's different coordinate systems that allow them to communicate with each other about the position of the object based on their positions relative to each other.

Once the radar and camera have their rigid body transformation handled, the system works in an interdependent manner for the most part. The camera looks for traffic signs as the vehicle is moving. Meanwhile, the radar detects the sign as a set of points. These can easily be hidden by bushes or trees or such other structures as mentioned earlier. By detecting the sign via the camera first, we verify if there is a sign and then ask the radar to locate it in the vicinity of where the camera found the sign.

As soon as a sign is noticed, the camera sends the heading angle and the size of the bounding box to the radar. The heading angle is the angle from the horizontal center of the frame, calculated via the resolution of the camera, to the center of the sign in this case. The bounding box is a virtual image region generated by the detection and recognition framework, YOLO, in which the camera believes that the sign resides. In the case of the image below, θ and x will be sent to the radar system [Figure 3] [Equation 1].

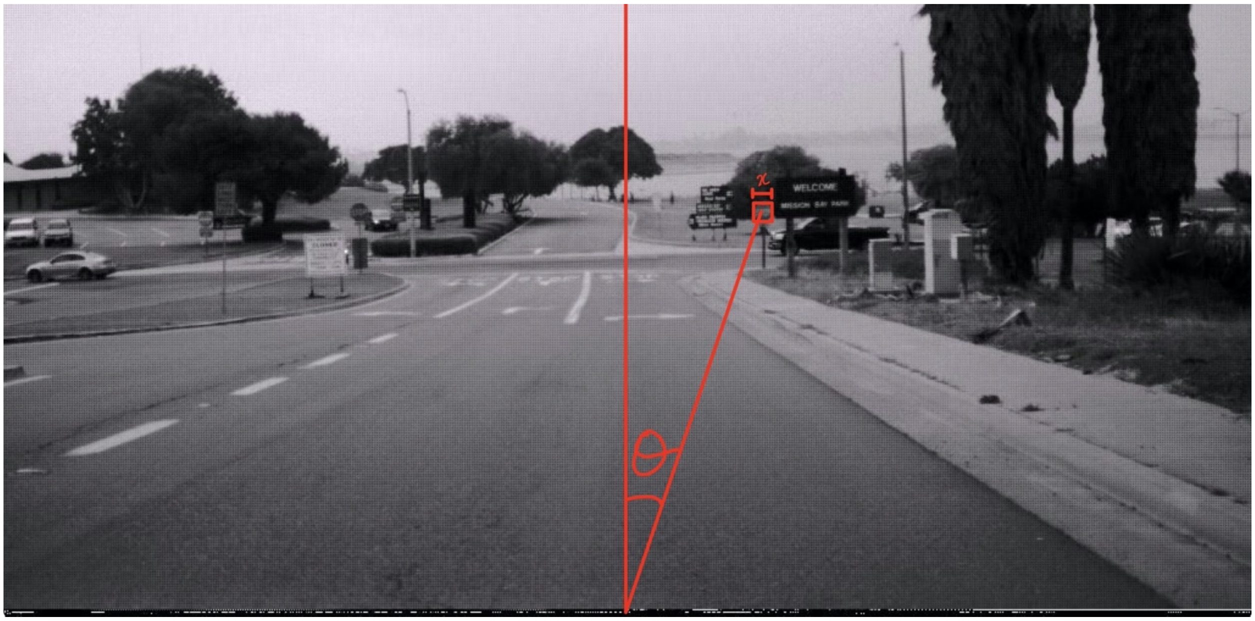

Figure 3. Image of the traffic sign with the data marking visualization from the camera module

$$\theta = 90^\circ - \mathrm{TAN}^{-1}\frac{y_B}{x_B - x_F} \tag{1}$$

Equation 1. Equation to calculate θ**.** B: Center of Bounding Box. F: Center of frame

These values relate to the positional data that the camera can give to the radar. The θ and the x values seem to increase as the car approaches the sign. The θ value describes the heading angle of the sign around which the radar should try to search for the sign. The x value describes the relative position of the sign to the vehicle as it signifies the perspective view of the camera. The x value will have a distance formula for it to give the radar a relative distance range for the sign. These θ and x values increase as the vehicle gets closer. This can be seen in Figure 4.

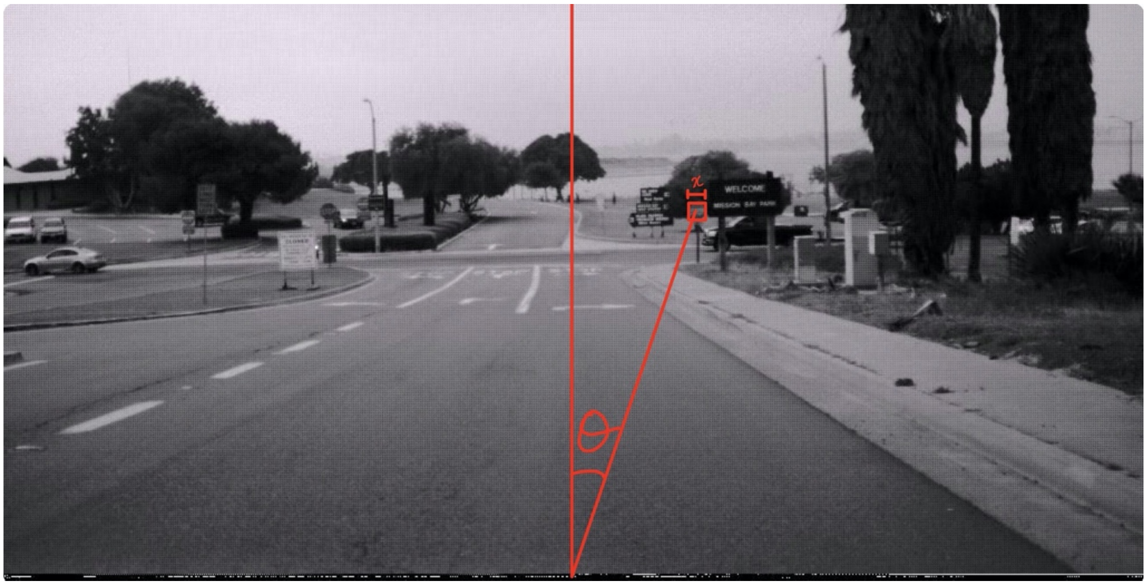

Figure 4a. Image of the camera view of the sign far away

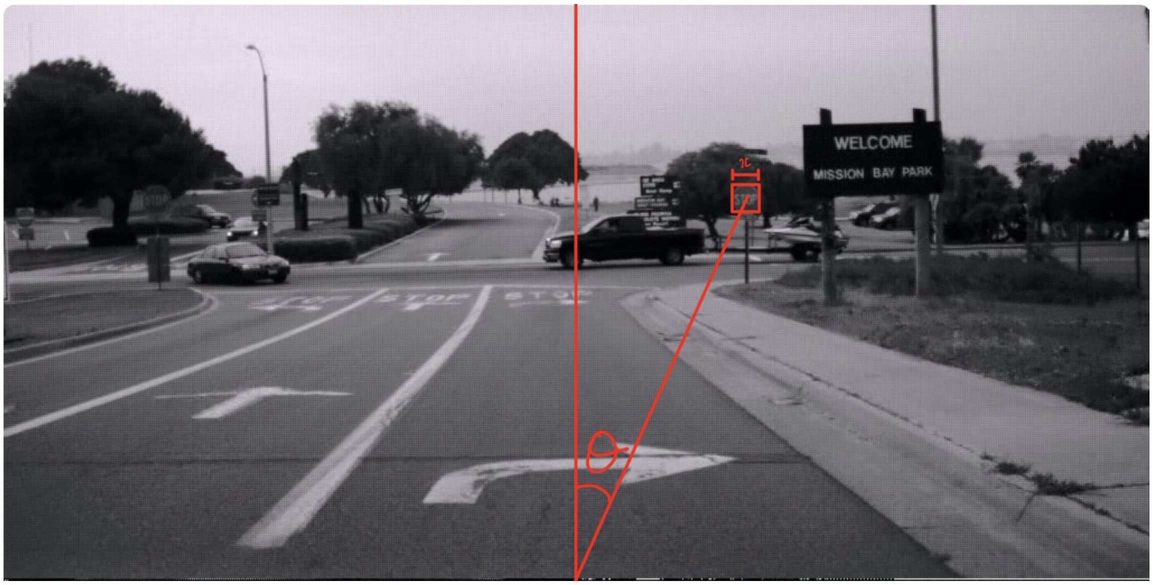

Figure 4b. Image of the camera view of the sign closer than the one in Figure 4a

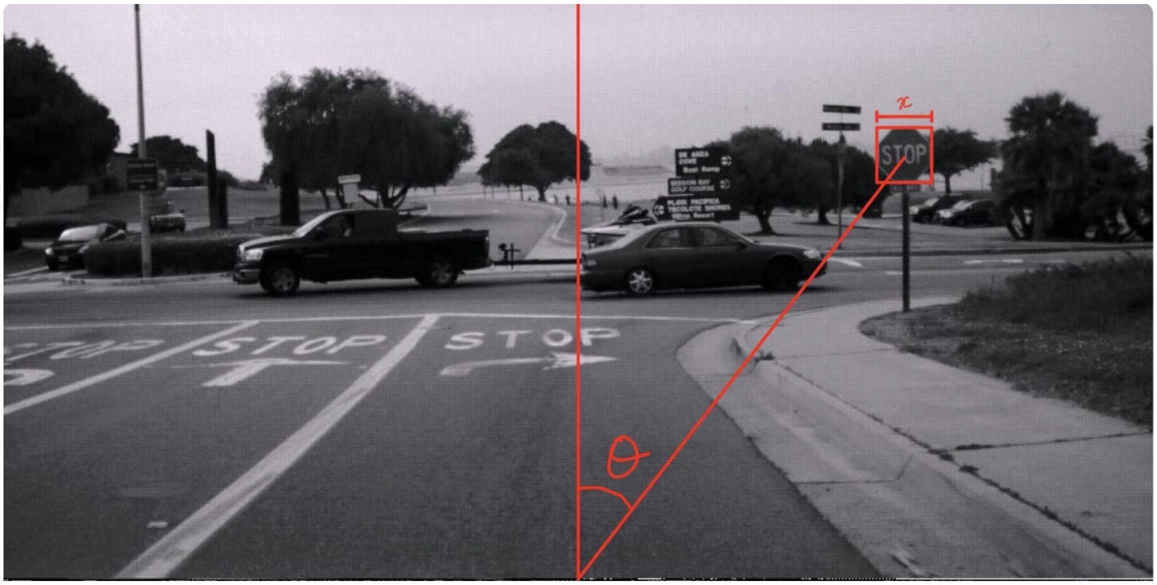

Figure 4c. Image of the camera view of the sign closer than the one in Figure 4b

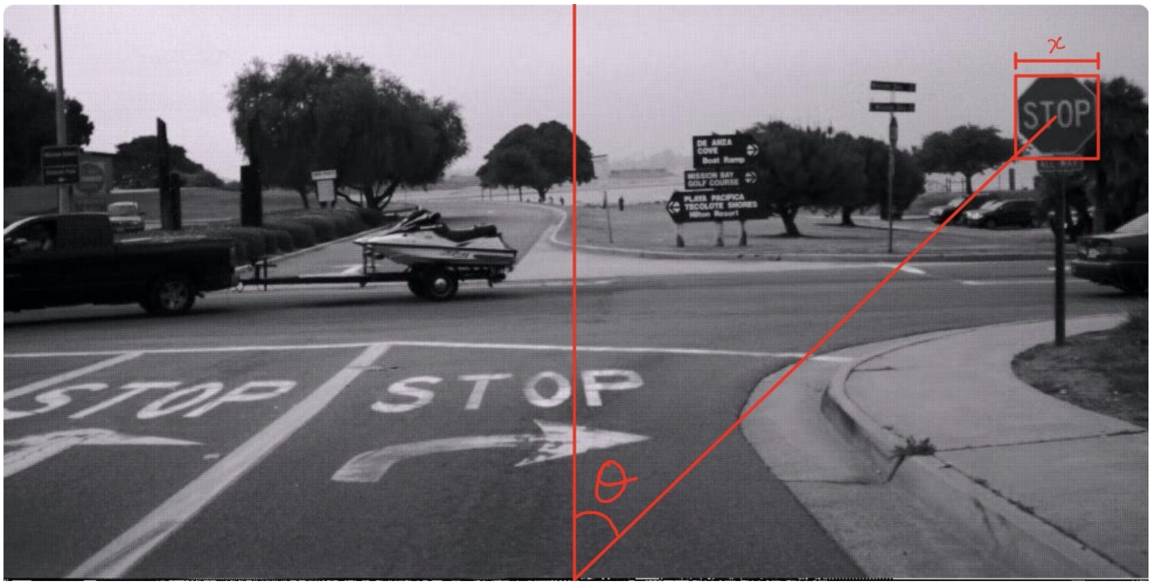

Figure 4d. Image of the camera view of the sign closer than the one in Figure 4c

Figure 4. Images from top to bottom of the camera view of the sign signifying the increase in *x* and θ values

However, this is not always the case. Multiple times while driving, the sign is on the right side of the road but due to the curvature of the road, the sign appears on the left of the center of the frame. This is also taken care of using the θ value. The camera with sending just the *x* value and a θ value greater than 90° signifies that the sign is on the left side [Figure 5]. Sending these two parameters to the radar gives it a small area to search for the sign rather than the whole sensing region [Figure 6].

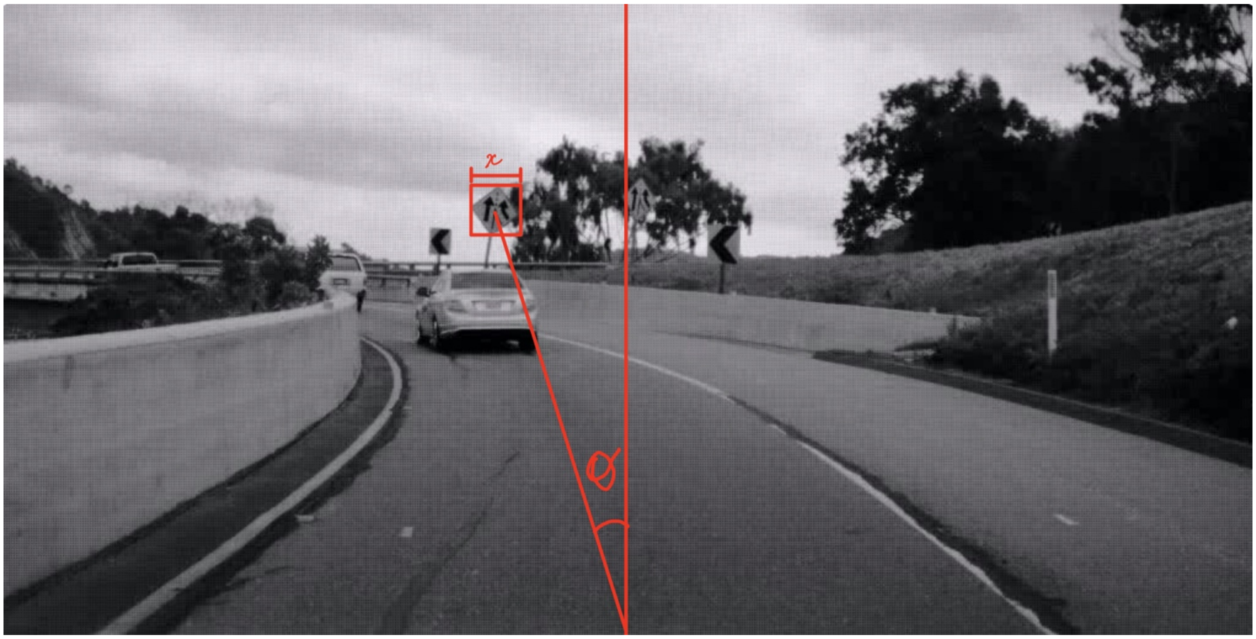

Figure 5. Right-left deflection of the θ value

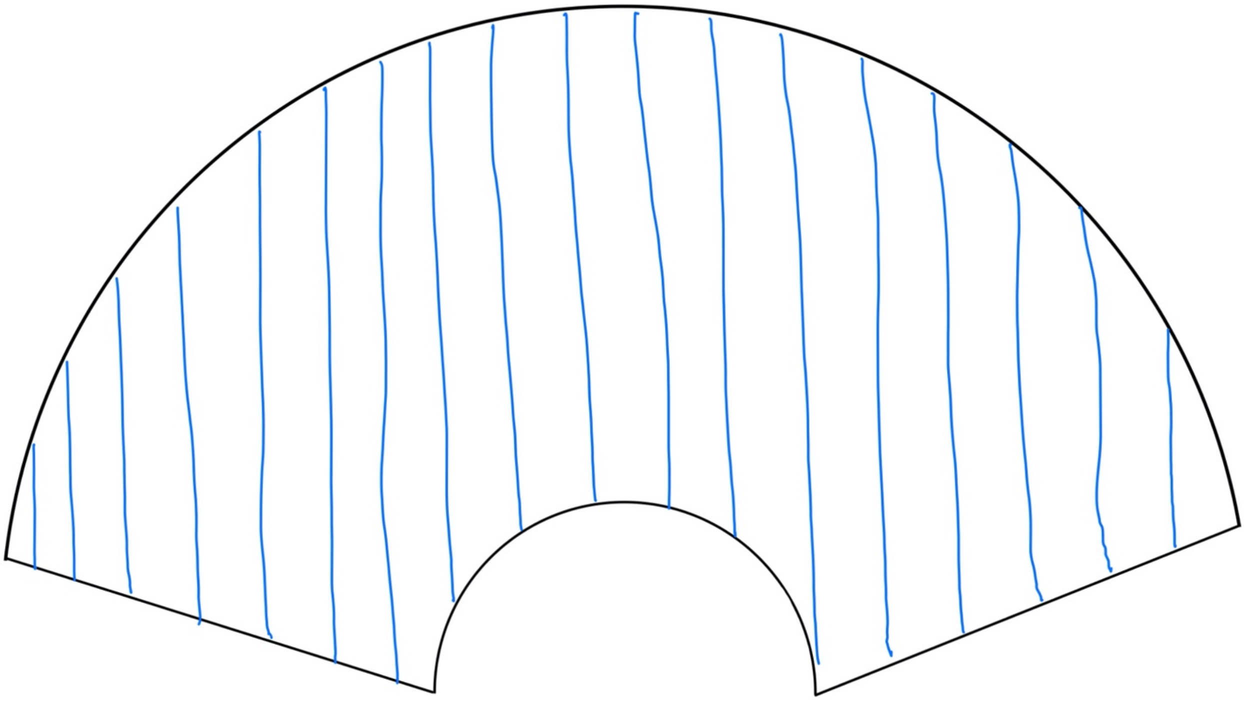

Figure 6a. Radar detection range example without camera parameters (not to scale)

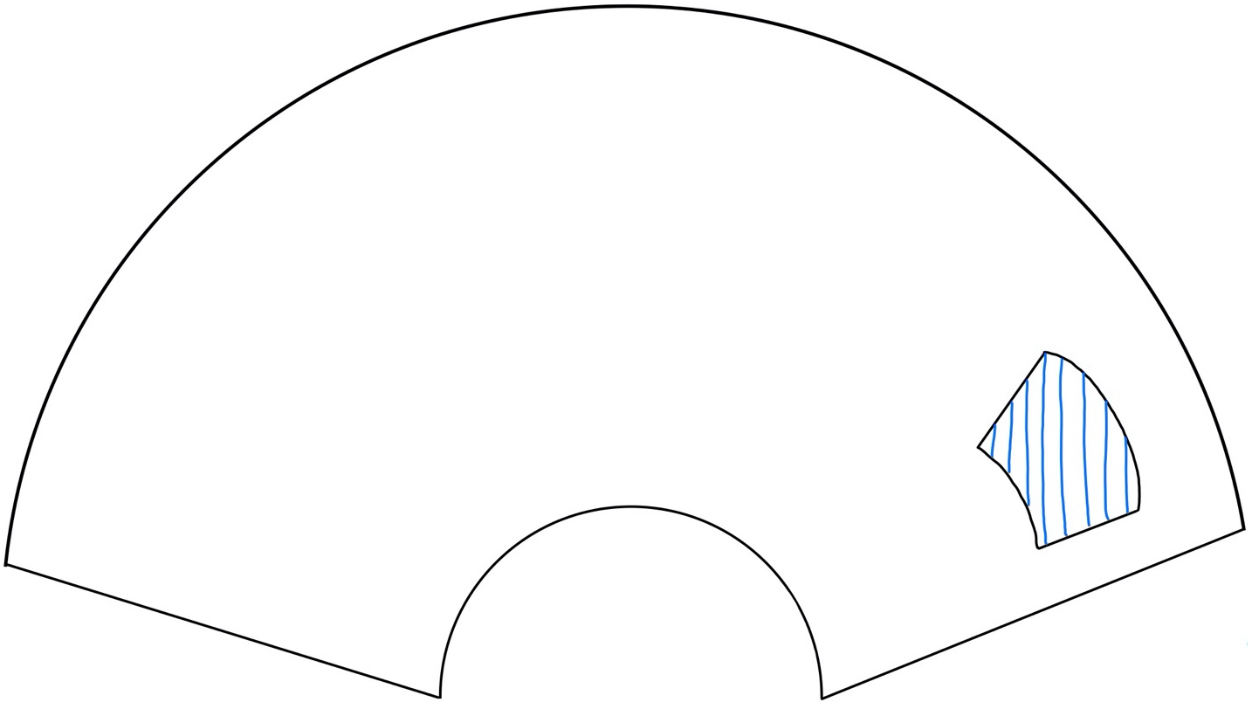

Figure 6b. Radar detection range example with camera parameters (not to scale)

Figure 6. Radar detection range examples with and without camera parameters (not to scale)

As depicted in the image above, with the correct use of the camera, the radar can be helped a lot by narrowing the field of search making it faster and more accurate. This tactic works even better with the other tactics as they enhance this search with further data about the sign that help the system narrow its search to locate the sign and recognize it.

CHAPTER III

CONCLUSION

Autonomous vehicles are the future of the automobile industry. They not only will aid in reducing accidents but also alleviate the stress of driving for hours from one place to the other. In this paper, we look at how to get this one step closer to reality. This paper focused on using the power of two sensors, Radar and Camera, to detect road signs.

The radar is a good spatial detection sensor with really accurate and fast detection but with limited recognition capability. The camera, on the other hand, is fast and accurate at recognition but lacks spatial location. When fused together, not only can they reduce the time taken to analyze road signs, but also be more accurate. We looked at how using YOLO in the camera reduced times by up to a third of that taken by standard CNN algorithms while improving accuracy by iterating through the number subdivisions and batches while training it in the supercomputers of HPRC center in TAMU.

With this, we also looked at how the radar is good at detecting traffic signs and how it can be improved via tactics like curbside recognition, HD mapping, and Radar-Camera fusion. Although HD mapping and curbside detection aided in the detection of traffic signs, they did not help as much as the Radar-Camera fusion.

Radar-Camera fusion is a method when the camera sends data about the traffic sign gathered based on the size of the bounding box and its heading angle. This gives the radar preliminary data about the position and approximate distance of the sign. This narrows the radar's search perimeter to get the exact location of the time and hence, makes it more accurate and faster. This when combined with the other tactics like curbside visualization and HD mapping, gives us a robust method of detection and recognition.

Future Steps

In the upcoming period, testing and improving would be the main goal, however, due to the unforeseen events surrounding the coronavirus disease, COVID-19, in Spring 2020, complete data was unavailable at the time of publication for this URS thesis. Other steps would be the addition of probabilistic localization of the sign, or one look-ahead on the radar-camera fusion, similar to the $H\ \infty$ method for maintaining car distance, as it will help make the process even more accurate and faster [13]. Another step would be to add an extra CNN layer to improvise the accuracy of the YOLO recognition model.

REFERENCES

[1] Schöller, C., Schnettler, M., Krämmer, A., Hinz, G., Bakovic, M., Güzet, M., & Knoll, A. (2019). Targetless Rotational Auto-Calibration of Radar and Camera for Intelligent Transportation Systems. *ArXiv, abs/1904.08743*.

[2] Tabernik, D., & Skocaj, D. (2019). Deep Learning for Large-Scale Traffic-Sign detection and Recognition. *ArXiv, abs/1904.00649*.

[3] H. Cho, Y. Seo, B. V. K. V. Kumar and R. R. Rajkumar, "A multi-sensor fusion system for moving object detection and tracking in urban driving environments," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, 2014, pp. 1836-1843.

[4] Zhengping Ji and D. Prokhorov, "Radar-vision fusion for object classification," *2008 11th International Conference on Information Fusion*, Cologne, 2008, pp. 1-7.

[5] J. Domhof, K. Julian F. P. and K. Dariu M., "An Extrinsic Calibration Tool for Radar, Camera and Lidar," *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, 2019, pp. 8107-8113.

[6] Lekic, Vladimir, and Zdenka Babic. "Automotive Radar and Camera Fusion Using Generative Adversarial Networks." *Computer Vision and Image Understanding*, vol. 184, 2019, pp. 1–8., doi:10.1016/j.cviu.2019.04.002.

[7] H. Ha, M. Perdoch, H. Alismail, I. S. Kweon, and Y. Sheikh, "Deltille Grids for Geometric Camera Calibration," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Venice, 2017, pp. 5354–5362, doi: 10.1109/ICCV.2017.571.

[8] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, "Vision-Based Traffic Sign Detection and Analysis for Intelligent Driver Assistance Systems: Perspectives and Survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 13, no. 4, pp. 1484–1497, Dec. 2012, doi: 10.1109/TITS.2012.2209421.

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and*

Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[10] C. Tung *et al.*, “Large-Scale Object Detection of Images from Network Cameras in Variable Ambient Lighting Conditions,” *ArXiv181211901 Cs*, Dec. 2018.

[11] L. Jiao *et al.*, “A Survey of Deep Learning-based Object Detection,” *IEEE Access*, vol. 7, pp. 128837–128868, 2019, doi: 10.1109/ACCESS.2019.2939201.

[12] Y. Meneroux, A. Guilcher, G. Saint Pierre, M. Hamed, S. Mustiere, and O. Orfila, “Traffic signal detection from in-vehicle GPS speed profiles using functional data analysis and machine learning,” *Int. J. Data Sci. Anal.*, Oct. 2019, doi: 10.1007/s41060-019-00197-x.

[13] F. Roselli *et al.*, “H∞ control with look-ahead for lane keeping in autonomous vehicles,” in *2017 IEEE Conference on Control Technology and Applications (CCTA)*, Aug. 2017, pp. 2220–2225, doi: 10.1109/CCTA.2017.8062781.