M-DAS: A Modular and Cost-Effective Real-Time Driver Alert System

Jeremy Zhang

Abstract

Last year, the tragic death of two local high school students seemed to be the latest in the growing trend of distracted driving among teenage drivers. According to the American Automobile Association, approximately 60% of all teenager related crashes involve a distracted driver. Teenagers are often not as experienced as other drivers on the road and thus are easily distracted by factors both internal and external to the vehicle. Car manufacturers have tried to address this issue through several driver assistance systems that attempt to alert the driver when dangerous driving behavior is exhibited. However, these systems only exist as add on packages to the newest generation of vehicles, are built and mounted physically to the vehicle, expensive to purchase and maintain, and do not achieve the full potential of such a system. For many parents, a driver alert system to keep their teenage child focused and safe on the road is cost prohibitive. They face the dilemma of either purchasing a used car that does not have the newest systems, or a driver assistance package of roughly the same price of the car, but not both at the same time. Purchasing one over the other would either be completely useless or less safe than desired. This project seeks to build an advanced driver alert system so that this dilemma is eliminated completely. To do so, the system must be accurate and robust, affordable, and adaptable to work on any vehicle through a modular design.

The United States Federal Highway Administration reports that lane departure accidents account for more than half of all fatal crashes every year. Therefore, it was paramount to develop a lane departure warning system first. The algorithm for lane detection started with video taken live from the camera, converted the image to grayscale, and used Canny Edge Detection to detect the edges of the lane markings. These detected edges were then grouped together with a linear regression algorithm and drawn on the screen. This algorithm has demonstrated robust lane detection on straight roads in real time with no noticeable latency. This first prototype has also been designed and built with a cost that is one tenth of current systems. While the system still runs on a laptop, this initial prototype is already more modular than systems available on the market today, with swapping between two cars taking no more than 10 minutes.

The second prototype focused on some of the limitations of the first prototype. First, the lane detection algorithm was modified to be more adaptable to different road conditions. This was achieved by changing the grayscale image to an RGB color mask image, as it was realized that lane markings are usually only of a certain color. This step allowed for more accuracy as many potential color distractions on the road were eliminated. The regression algorithm was also changed to be based on parabolas, as it would allow for the accurate detection of curved lanes. Also added in the second prototype was a method to calculate the offset from the center of the lane based on the pixel difference between the center of the car and the calculated center of the lane with a visual and auditory cue when the vehicle is veering into another lane. This system is also very robust and functions with no noticeable latency in lane detection. While these two prototypes have already made significant progress on the previously discussed engineering goals, future prototypes will focus on making the system lighter and smaller, and further reducing the cost of the system. This system would help to fully realize the goals of a driver alert system and make the road safer for all that use it.

Acknowledgement of Major Assistance

The author is thankful for the support of his sponsor, teachers, classmates, and parents. The author would also like to acknowledge the creators of open-source software libraries and is grateful for the work they have done.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Motivation and Purpose

In high school, many students often obtain their driver's license. For many, it is an exciting but anxious experience to drive. Teenagers drivers differ because a wide variety of factors, such as their individual personality, perception of the environment, behavior tendencies, and their knowledge, skill, and experience with operating motor vehicles. (Shope and Bingham, 2008) Even when basic knowledge and skills have been trained, much practice and experience is needed to make teens safer on the road. Due to the young age of teenagers and the general lack of experience because of that young age, novice teen drivers are less able to recognize and detect risks than are more experienced drivers. (Brown and Groeger, 1988; McKnight and McKnight, 2003)

Scholarly journals and literature have published extensive research on the issue. (Bertozzi, 1998; Li et al., 2014; Kim, 2008; etc.) While the extensive literature shows promising results, few of the proposed solutions have actually been implemented for widespread civilian use, most probably due to the novelty of the technology. Most scientific literature on this issue seems to focus on the software aspect of ADAS, while few, if any, mention the hardware aspect. The current literature focuses on high level, novel, and theoretical approaches to ADAS while few state a goal of developing a system for widespread civilian use. The few technologies that have been developed for use have been developed by automobile companies and face many limitations that will be discussed in the next section.

Figure 1 Honda Sensing Technology
Image Credit: Honda

Few scholarly works exist on other applications of the same image processing and object recognition problem (Jabnoun et al., 2014). This work focuses on another potential application of the same technology. However, the primary focus was walking and daily navigation for the visually impaired, with testing only done on videos and not in real time. Walking speeds are many times slower than vehicle speeds. The response time needed for the former to keep users safe is on a completely different scale from the vehicle problem. The same amount of accuracy is needed, the prototype must be in real time, and the response time must be faster while keeping costs low and making the system convenient.

1.2 Approach

Current technologies are limited by their cost and accessibility due to the lack of interchangeability of the system between vehicles. Because systems cannot be exchanged between cars, current ADAS implementations and designs are highly complex and rely on a wide variety of sensors. These sensors are often mounted in locations around the vehicle that are not easily accessed by mechanics fixing malfunctioning components of the system.

The approach that was used in this project was to focus on simplicity, modularity, and portability. Having a simple design would decrease the overall amount of parts, and in turn reduce complexity of the system. This would make assembly cheaper overall, and make maintaining the system a more trivial task.

Portability and modularity would allow for greater flexibility in the everyday usage of the system. The ability for a user to take the system out of a vehicle and to place it into another vehicle reduces the need for the user to purchase additional systems, resulting in cost savings for the user. Swapping the system would also allow parents to switch the system out between their own vehicle and other vehicles that their teenage children may be driving.

1.3 Engineering Goals

The goal of this project is to develop and build an ADAS system that works on any car, irrespective of the model or age. Because of the broad scope of features that a full ADAS system contains, the scope of the current year's research was limited to focus on one of the many features of a full system. Because lane departures account for the overwhelming majority of road accidents each year, it is important to create a lane departure warning system first, as its impact would be the greatest. To measure the success of the prototype, the following criteria were established.

1.3.1 Performance and Accuracy Goals

Vehicles, when travelling on the road, have the potential to move great distances in short amounts of time. A car has the potential to travel the distance of an entire standard American football field in just five seconds, the average amount of time it takes to take one's eyes off the road, interact with the radio, and look back up. It is, therefore, paramount that the system be able to quickly detect lane departure and issue a warning. For this prototype, a real time detection system would satisfy the goal.

A lane departure warning system for deployment on cars must be able to be accurate in distinguishing lane markings and detecting lane departure as well, as too many false positives may cause the user to turn the system off while too many false negatives may fail to recognize actual lane departures.

1.3.2 Modular and Portable Design

A portable design would allow for greater ease of use and convenience. As mentioned in Section 1.2, current ADAS offerings cannot be taken from one vehicle to another, which poses challenges to users who have more than one vehicle that does not have this technology. This for the modular and portable design goal to be met, the prototype must be able to be swapped between vehicles in under an hour.

1.3.3 Cost

Current implementations of ADAS cost upwards of thousands of dollars. Teenage drivers most likely do not have the budget to purchase such a system. Many other drivers also cite cost as a major factor when it comes to purchasing a new car. Many people are constrained by money when looking for ADAS packages. For this system to be used by everyone, including teenage drivers, the cost of the product must be as low as possible. The ideal prototype should cost one-tenth of current ADAS packages for widespread use.

2. Materials and Methods

2.1 Materials

2.1.1 Software Materials

All of the software materials were open-source or free to use.

- Python 3.6
- Open Source Computer Vision Library (OpenCV) 3.2.0
- Numpy 1.12.1
- JetBrains PyCharm (IDE) Community Edition 2016.3.3

2.1.2 Hardware Materials

- Dell Latitude E6430 running Windows 7 (Laptop)
- Microsoft LifeCam VX-5000 (Webcam)
- 8 inch NuVision Tablet
- YQ-XP058 Universal Dashboard Mount

Figure 2 NuVision Tablet and Mount

2.2 Research Methods

The following sections detail the research methods used in the building of the software and hardware prototypes.

2.2.1 Software Prototype Development

The development of the software prototype was done through an iterative design. Flaws were fixed from one prototype to the next and new features were added over time.

2.2.1.1 Canny Edge Detection

The initial design of the software prototype focused on simplicity of design. A reliable method of edge detection as outlined by Canny (1986) was the primary step in finding lane markings on the road. There were many other steps in addition to Canny Edge Detection in the creation of the first real time prototype. Each step was executed in the precise order listed below for each incoming video frame from a camera.

1. Apply a color mask. Lane markings on the road are most commonly either white or yellow. This step converted all of the pixels in the frame that were not yellow or white to black.
2. Grayscale the image. Grayscaling the image reduces the variety of colors, and makes the contrast between lane markings and the normally dark colored road more apparent for Canny Edge Detection.
3. Apply a Gaussian Blur, also known as a Gaussian Smoothing. This, as the name suggests, applies a blur to the image, which filters out even more of the noise in the image frame. This step is important because edge detection algorithms are easily affected by image noise, and serves to reduce the amount of false detections.
4. Apply Canny Edge Detection to the image. This step detects the lane lines because of the contrast to the road it provides.
5. Use Probabilistic Hough Transform (Hough, 1962; Duda and Hart, 1972; Kiryati et al., 1991) to detect lines in the image frame. This step ensures that many small line segments of detected edges will be present for a regression fit.
6. Mask a trapezoid area where the lanes are most likely to be. This reduces even more distracters because it masks out edges of nearby vehicles, roadside signs, etc. This reduces the amount of false detections as linear regression is prone to noise and distractions.
7. Use linear regression to fit all of the Hough Lines into two lines, one for each lane. To ensure the accuracy of the lines, one line will always have a positive slope and the other line will always have a negative slope.
8. Overlay the regression lines on the original image and output the image to the screen.

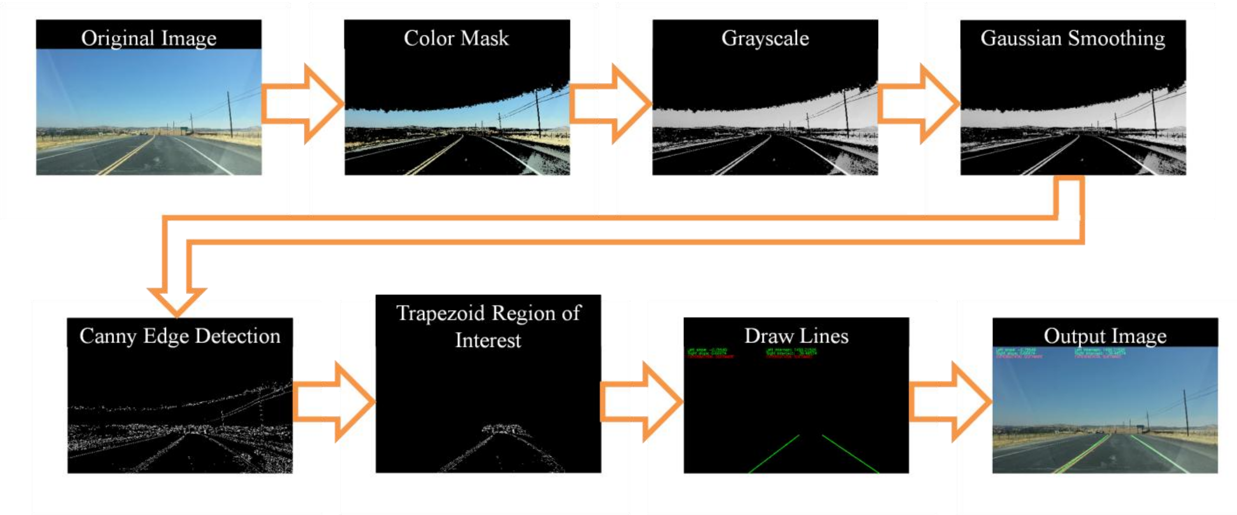

Figure 3 Illustration of Canny Edge Detection Processing Steps

2.2.1.2 Multi-Window Scanning

The creation of the second prototype was driven by many of the limitations of the first software prototype. This prototype allowed the overall product to reach the engineering goals that were previously defined.

Overall, the complexity of the software increased, which in turn led to higher accuracy when detecting lane boundaries. More features were added in this prototype as well. The major difference between the two prototypes is that this one has the ability to detect curvatures in lane boundaries. As a result, the processing flow was changed significantly to address these limitations. Each step of the new processing flow was performed in the exact order listed below for each video frame coming from either a prerecorded video or a camera mounted to the vehicle.

1. Calibrate the camera with 20 checkerboard images taken from various orientations on a white background. Although the camera used during experimentation had minimal to no distortion, this step increases the accuracy of detection for all cameras. In the event that a user uses a wide angle camera with fish-eye distortion, the algorithm would still be able to adjust for the camera distortion.

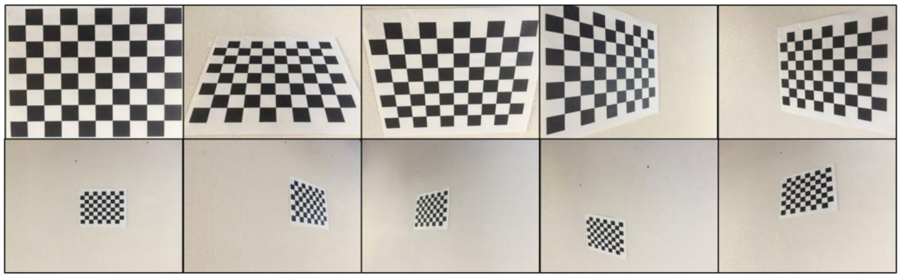

Figure 4 Selected Camera Calibration Images

2. Apply a color mask to mask out colors that are not of lane lines. This step was from the previous algorithm, because it was determined to be highly effective reducing distractions and in turn increasing overall accuracy.
3. Perform a perspective transform. This step warps the image to allow a “bird’s eye view” from a perspective above the road looking down. This step allows for greater accuracy in polynomial fitting that occurs later.
4. Calculate a histogram for the amount of white pixels against the black background. The peaks of the histograms indicate where potential lane lines will be, due to the high number of white pixels.
5. Divide up the image into left and right and enclose the area around each histogram peak.
6. Create nine search windows based on the location of the histogram peak. This step essentially uses many smaller search windows to find the lane lines. These search windows reduce the amount of total area to be searched and also reduces distractions that are not part of lane boundaries.

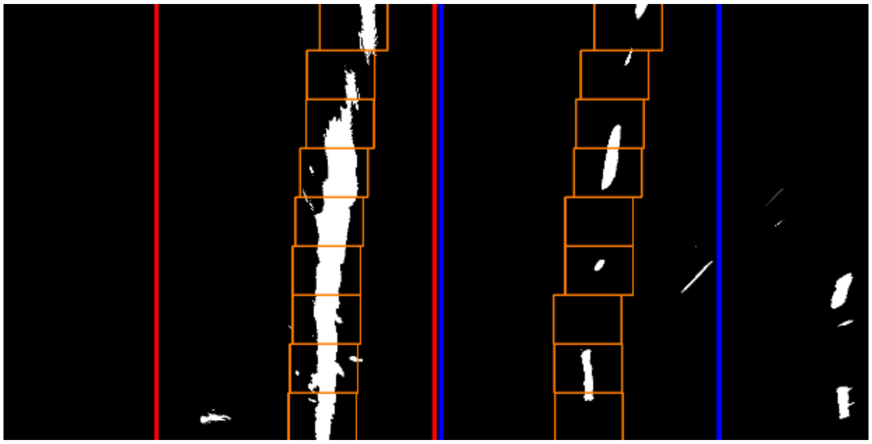

Figure 5 Searching Windows

7. Use polynomial regression to fit a second degree polynomial based on the detected pixels for each side. This is the main difference between the previous software prototype. A quadratic regression fit would allow the curvatures in the road to be represented by parabolas, which is a significant improvement over representing them with linear lines.
8. Perform a perspective transform again. This time, the perspective transform warps the image back to the original form.
9. Overlay the fitted quadratic lines onto the original image, and shade in the region between them to represent the lane.
10. Calculate the distance of the vehicle from the center of the lane. If the deviation from the center of the lane is greater than a preset value, a visual warning is displayed on the image.

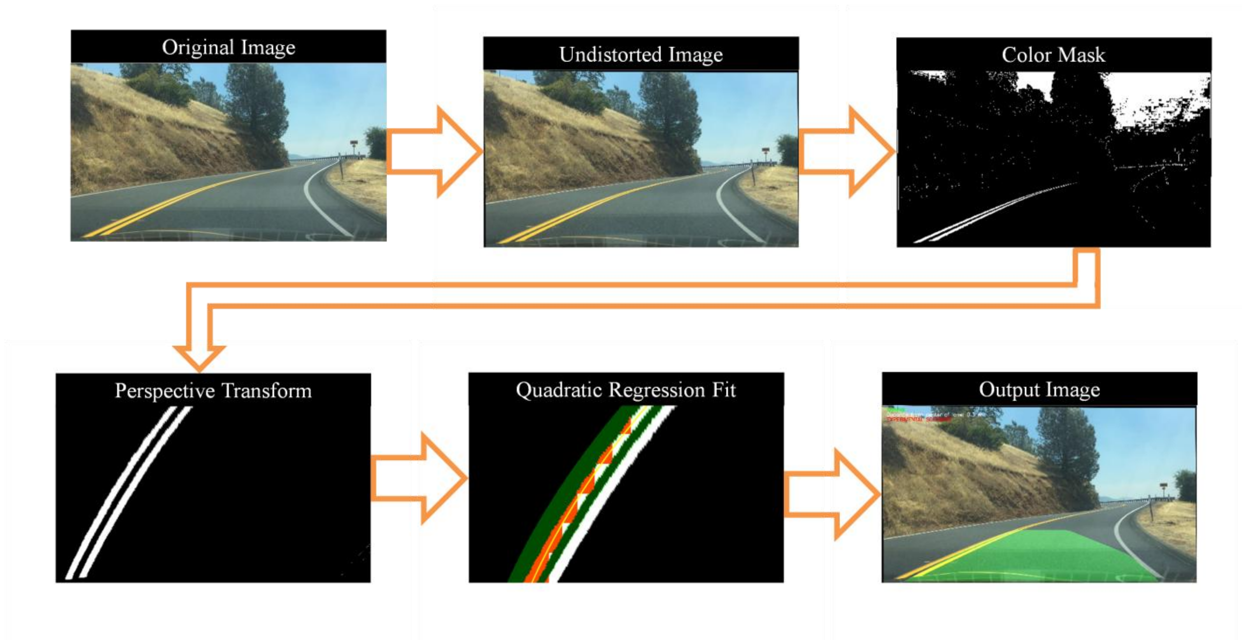

Figure 6 Illustration of Multi-Window Scanning Processing Steps

2.2.2 Hardware Prototype Development

During the experimentation phase with the lane detection algorithms mentioned in the previous section, it was observed that processing real-time video requires massive amounts of computing power. Therefore, open source maker boards such as the Arduino and Raspberry Pi or toys such as Lego Mindstorms were not considered for real life implementations as they did not have the computing power required, and were difficult to scale into a future commercial implementation of ADAS that would satisfy engineering goals.

The choosing of materials for hardware prototype development also focused on using as many commonly found or preexisting parts to be used in prototypes. This placed a special emphasis on the use of laptops, tablets, and smartphones, because many potential users would most likely also possess these items.

2.2.2.1 Laptop and Webcam

The initial prototype that was considered was a laptop. Using a laptop had many benefits, chiefly that laptops have the required processing power to achieve real-time lane detection. The laptop was then connected to a webcam mounted on the dashboard of the vehicle with a piece of tape, firmly securing it in place. The initial prototype performed well on all aspects of the engineering goals. This prototype, when running the Multi-Window Scanning algorithm was very accurate in detecting lane boundaries. Because of the massive amounts of processing power that the laptop was able to provide, lane boundary detection and issuing warnings were done almost instantly. The cost goal was met, because the total cost of the prototype was well under

$300. The portability goal was also met, because the system could be removed from one vehicle and placed in another vehicle within the matter of minutes.

The overall laptop prototype was limited in one way. Although the prototype made great strides in making ADAS smaller and moveable, the size and weight of the laptop posed challenges while driving. The laptop computer running the software must be either secured to the front passenger seat of a car or physically held onto by the passenger sitting in the front passenger seat. This prototype was not very realistic for many users. Many users simply do not carry a laptop computer with them at all times.

2.2.2.2 Tablet Integrated System

The second prototype focused on improving upon the limitations of the first prototype. This prototype used a dashboard mount to hold a tablet. This prototype was designed with a tablet, but can work with any model of phone or tablet as long as an integrated camera is present.

Figure 7 Tablet Integrated System Setup

This second prototype was able to make substantial increases for all of the engineering goals previously mentioned. A tablet is smaller than a laptop, and consequently cost less money to obtain for this study. Moreover, it is more common and practical for people to have either a smartphone or tablet on them instead of a laptop computer.

2.2.3 Prototype Testing

Initial prototype testing was to be done with pre-recorded videos that were processed later. This was to allow more time to fix any problems that were encountered while not needing to drive at the same time.

Figure 8 New Priest Grade, California Created Using Google Earth

After it was established that the prototype was able to perform at a reasonably accurate level, real-time testing was done on a real vehicle. Testing was done on local roads and highways. These roads, during normal hours and light traffic were considered to represent typical road conditions, as long, straight sections of highway with few cars is a common occurrence in Texas. Challenging tests designed to find the limits of the prototypes were conducted as well. The majority of the challenging tests were done on the New Priest Grade in California, because the road was narrow and consisted of many sharp turns and blind corners while driving up a grade of 3%.

3. Results

The following shows the results of testing the software prototypes on typical roads, challenging roads, and in real time.

3.1 Typical Roads (Highway 190, Texas)

Figure 9 Canny Edge Detection on Typical Roads

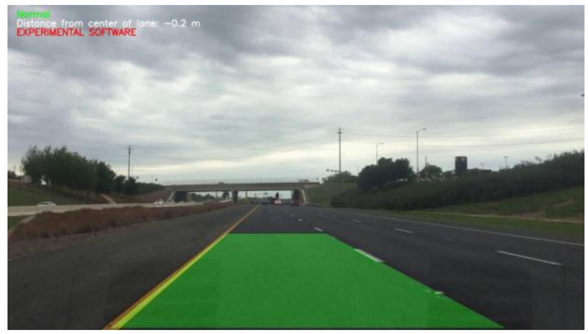

Figure 10 Multi-Window Scanning on Typical Roads

The pictures seen in Figure # represent the performance of the prototypes on typical roads during the day. The green lines on the left represent the lanes as detected by Canny Edge Detection and the green shaded region show the lane as detected by Multi-Window Scanning.

3.2 Challenging Road Conditions (New Priest Grade, California)

Figure 11 Canny Edge Detection on Challenging Roads

Figure 12 Multi-Window Scanning on Challenging Roads

The pictures seen in Figure # above represent the performance of the prototypes on challenging roads during the day. The two different time periods were chosen because they featured tight turns and blind corners, which was initially hypothesized to be difficult for the algorithms to detect. The green lines on the left represent the lanes as detected by Canny Edge Detection and the green shaded region show the lane as detected by Multi-Window Scanning.

3.3 Real Time Testing

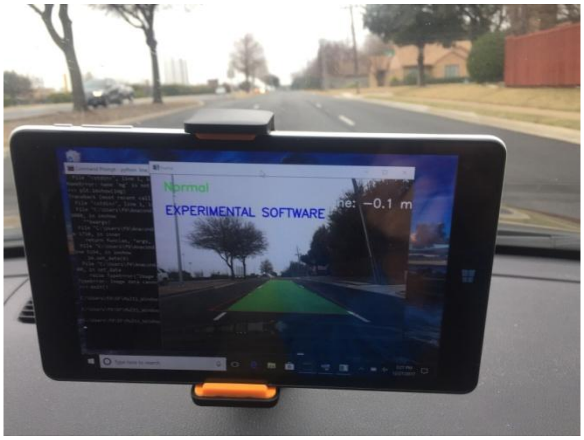

Figure 13 Real Time Testing

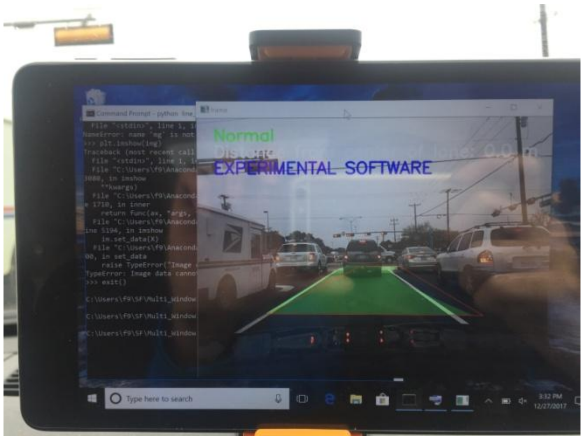

Figure 14 Real Time Testing in Heavy Traffic

The pictures seen in Figure # represents the final iteration of both the software and hardware prototype working in real time. The pictures above were taken during real-time testing on local roads. The picture seen in Figure # represents the final iteration of both the software and hardware prototypes working in real time under heavy traffic on a local road. It can be observed that there are many vehicles surrounding the target vehicle. The green shaded region represents the lane as detected by the final software prototype, and a red trapezoid shown on the screen represents the search area for lanes.

4. Discussion and Conclusions

4.1 Performance and Accuracy

The overall performance, measured in Hz, was approximately 3 to 5 Hz. This means that on average, the delay – if one was needed – would come at 250 milliseconds after the lane departure has occurred. In that time period, the vehicle would have travelled 447 centimeters at 60km/h or 695 centimeters at 100km/h.

The average interstate roadway in the United States is 3.7 meters wide, and an automobile is 1.8 meters wide. This leaves almost 2 meters of space in the lane, 1 meter on each side. Realizing that there was a very minimal delay in the detection of lane departure, the prototype was modified to issue a warning when the vehicle was more than 0.3 meters from the center of the lane. That way, the driver of the vehicle would have time to correct for potential lane departure. If the driver did not correct for lane departure, another warning was issued when the vehicle completely departed the lane.

From the pictures seen in the results section, both of the algorithms performed well in detecting lane lines on typical roads. However, on challenging roads, Canny Edge Detection did not perform as well as Multi-Window Scanning, as evidenced by the erratic nature of the detected lane boundaries. This was because the Canny Edge Detection algorithm was not designed to detect curvatures in road boundaries. This suggests that Multi-Window Scanning more applicable to a wider variety of road scenarios.

A real time software prototype was developed with accurate results, which demonstrates that the first engineering goal has been met.

4.2 Modularity and Portability

The modularity and portability goal was met because the system could be taken out of a vehicle and placed into another vehicle. This demonstrates that the modularity and portability goal was met because of this mobility. However, this engineering goals was exceeded, because the prototype can be removed from a vehicle and placed into another one in a matter of just a few minutes.

4.3 Cost Analysis

Component/Part	Cost
Dell Latitude E6430 Laptop	$250
Microsoft VX-8000 Webcam	$20
Software	$0
Total Cost	$270

Table 1 Laptop and Webcam Prototype Cost Analysis

A cost analysis was performed on the first prototype, with the exact cost breakdown as shown below. The cost of the laptop accounts for 93% of the total prototype cost. However, it is very common for households nowadays to have laptops. In that case, the existing laptop could be used as a substitution for purchasing a new laptop. This would significantly drive down the cost of the first hardware prototype.

Component/Part	Cost
8 inch NuVision Tablet	$60
YQ-XP058 Universal Windshield Mount	$8
Software	$0
Total Cost	$68

Table 2 Tablet Integrated System Cost Analysis

A cost analysis of the second hardware prototype was performed as well, with the exact cost breakdown as shown below. Again, the cost of the tablet accounts for the overwhelming majority of the overall prototype cost. However, because the design of the prototype focused on using materials that were commonly found, the overall cost could be reduced to just $8 for a mount, or completely free if one possess a tablet, or smartphone, and a mount.

The engineering goal of cost was met, but also exceeded because the prototype cost one-tenth of the current average price of ADAS systems.

4.4 Error Analysis

Mounting of the camera in the vehicle may have not been perfect. The algorithm calculated the center of the lane and the deviation from that center under the assumption that the camera is mounted perfectly in the center of the vehicle. However, the calculation of deviation from lane is built with a buffer distance, and any slight difference in mounting would not compromise the safety of the system.

5. Future Work and Applications

This prototype changes the paradigm in advanced driver assistance systems by providing a modular and cost-effective implementation. It provides users with greater flexibility and allows for more widespread use of driver alert systems. This, in turn, would prevent more accidents, and help keep people focused on the road.

Although the prototype was designed as a passive assistance system, this system can also aid in the development of a fully autonomous driving system in a self-driving car that could one day make travel easier for those with disabilities that cannot drive.

It was observed previously that ADAS has many different components and aspects. Future research could also focus on implementing a system for vehicle proximity warnings without the use of expensive sensors, or even traffic signal and road signage detection with the current physical architecture.

6. References

Bertozzi, M., & Broggi, A. (1998). GOLD: A parallel real-time stereo vision system for generic obstacle and lane detection. *IEEE transactions on image processing*, *7*(1), 62-81.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6), 679-698.

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.

Jabnoun, H., Benzarti, F., & Amiri, H. (2014, November). Object recognition for blind people based on features extraction. In *Image Processing, Applications and Systems Conference (IPAS), 2014 First International* (pp. 1-6). IEEE.

Kim, Z. (2008). Robust lane detection and tracking in challenging scenarios. *IEEE Transactions on Intelligent Transportation Systems*, *9*(1), 16-26.

Kiryati, N., Eldar, Y., & Bruckstein, A. M. (1991). A probabilistic Hough transform. *Pattern recognition*, *24*(4), 303-316.

Li, Q., Chen, L., Li, M., Shaw, S. L., & Nuchter, A. (2014). A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios. *IEEE Transactions on Vehicular Technology*, *63*(2), 540-555.

McKnight, A. J., & McKnight, A. S. (2003). Young novice drivers: careless or clueless?. Accident Analysis & Prevention, 35(6), 921-925.

Shope, J. T., & Bingham, C. R. (2008). Teen driving: motor-vehicle crashes and factors that contribute. American Journal of Preventive Medicine, 35(3), S261-S271.

VC, H. P. (1962). *U.S. Patent No. 3,069,654*. Washington, DC: U.S. Patent and Trademark Office.