Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 175 (2020) 165–172

Procedia Computer Science

www.elsevier.com/locate/procedia

The 17th International Conference on Mobile Systems and Pervasive Computing (MobiSPC)
August 9-12, 2020, Leuven, Belgium

Design and implementation of self-driving car

Mahmoud Fathy[a,*], Nada Ashraf[a], Omar Ismail[a], Sarah Fouad[a], Lobna Shaheen[a], Alaa Hamdy[a]

[a] *Faculty of Computer Science, Misr International University, Cairo, Egypt*

Abstract

Self-driving technology in general is becoming increasingly common and could revolutionize our transportation system. Also, self-driving cars are on their way of being legal, but they still are not trusted enough to be used in real life due to a lack of their safety. In this paper, a self-driving car prototype is proposed which integrates between different technologies including some algorithms which are Road lane detection algorithm, disparity map algorithm to detect the distance between the car and other vehicles, and Anomalies detection using Support Vector Machine classification algorithm as it achieved a very high accuracy using our data set. To test the car prototype, a special road environment was built to fit the car. Using the disparity map algorithm and merging between these algorithms will result in achieving safety and reliability for the self-driving technology.

Keywords: Stereo vision; Self-driving car; Road lane detection; Disparity map; Anomalies detection; Support vector machine;

1. Introduction

A road accident is the worst incident that could ever happen while driving as they happen quite a lot and the majority of them are caused by human error. Self-driving cars are being developed from day to another. It is a creative invention where the car is operated by a computer. It is hard to convince people that having a self-driving car is safe as they cannot trust a machine to keep them safe. A self-driving car is purely analytical that it acts exactly like a smart computer as there are no emotions or distractions involved because computers are faster and smarter to take actions than our minds. As a result, a future full of self-driving cars might be a better one. In this paper, a self-driving car prototype is proposed where new hardware components and methodologies are used in a self-driving car. The system consists of a raspberry pi as the main component that runs the algorithms, cameras attached with the raspberry pi, and a variety of sensors. Also, there is an important component in the system which is the Arduino as it is responsible for the car motors and their motions. Arduino receives signals from the raspberry pi and based on them, it takes the

* Corresponding author. Tel.: +20-115-206-0880;
E-mail address: mahmoud1611910@miuegypt.edu.eg

1877-0509
10.1016/j.procs.2020.07.026

appropriate decision for the car. The car consists of three main sub-systems which are distance measurement using disparity map, anomalies detection, and lane detection. Firstly, stereo vision dual cameras are used to calculate the disparity map so that the car can recognize the distance between itself and any other obstacles in front of it to avoid any chance of collision. The disparity map method is to be applied for the first time in a self-driving car as it calculates the distance and helps the car to be aware of the external environment more accurately than any other sensors used for some of the self-driven car prototypes like ultrasonic. Secondly, one of the dual cameras is also responsible for capturing and detecting the road lanes by applying some filters on the video frames from the camera to enhance them, then, The Hough transform technique is being used as a feature extraction for the lane detection part itself. Thirdly, the anomalies detection is done by using accelerometer and gyroscope sensors. Both sensors are being used to read road data so that, the support vector machine algorithm classifies either if those certain readings from sensors are anomalies or not. Such a usage for the disparity map method and an integration between these techniques leads to great results for the car decisions and awareness.

2. Related Work

2.1. Self-Driving Car Systems

By the innovation of technology and its existence in the manufacture of cars, cars have been developed to be more than a way of transportation. Creating a smart car presents the idea of a self-dependent car that can be safer for the driver. Therefore, self-driving cars made a huge evolution in the computer vision field and deep learning as mentioned in [1]. According to Chishti et al.[2], they used two methodologies to practice the self-driving car. The first methodology is "Supervised learning" where the car was driven on a road to collect more than three thousand data-points, which resulting in 89% accuracy. According to Memon et al.[3], their target is to make a dynamic destination by making one car (following car) follow another car (Target car) that is familiar with its destination by continuously receiving the target's direction and location. So, in this paper vehicles are smart enough to make intelligent decisions in as little time as possible and vehicles can determine their heading distances from other vehicles.

2.2. Lane Detection

Lane detection is an important technique that is needed for building a smart driving car. It is crucial to control the car movement according to the road lane that the car is on. [4] research work discusses the implementation of an algorithm combined with different techniques. Hough Transformation for feature extraction, SVM for machine learning along with line, edge, and region selection. They came up with a percentage of 90%. Also, they introduce the problem of different lighting affecting the system. On the other hand, Satzoda et al.[5] improved the Hough transform algorithm in lane detection and introduced the hierarchical additive Hough transform (HAHT) to achieve an accuracy of 99%. The usage of YOLO and CNN algorithms were introduced by [6], the system was able to detect road lanes, objects and provide suggestions to the driver. However, the system was highly recommended on highways only and not on urban roads. Moreover, Daigavane et al.[7] proposed a lane detection technique using Canny edges and Ant colony algorithm along with Hough transform, the system was able to be applied on painted and straight roads, but they pointed out that ant colony algorithm requires more research for such a domain.

2.3. Stereo Vision

Stereo vision cameras capture from two vantage points. They capture two images, one for the image itself and the other for depth map. The stereo vision approach was introduced in many fields other than robotics as human-computer interaction. Oniga et al.[8] had two phases while using stereo vision cameras. The first phase was to detect obstacles. The second phase was to improve the output result. In this system, they have used individual pixels that can be affected by the low quality of images and also consume more time. According to [9], they have implemented a method that detects easy and complex obstacles in the U disparity map on long-distance. They also pointed out that users of such a system would make a great difference in the field of smart cars as they have reached high accuracy with their method. Moreover, in research [10], they have discussed the effect of computation cost, aggregation cost, and disparity map algorithm from the stereo vision on the system. They suggested that in the case of unneeded high-resolution images, it is preferred to use normal cameras for not consuming much time.

2.4. Anomalies Detection

Anomalies are double-edged weapons as they may disturb drivers, cause accidents, and sometimes they can not be seen due to faded road signs. Anomalies classification is different from one system to another. In the research work [11], Anomalies can be easily detected from mobile accelerometer readings with the usage of the SVM classification algorithm as it achieved 90% accuracy. Moreover, the embedded sensor (Accelerometer) was used in [12] to detect the potholes on multi-lane and classify the data using the SVM classifier. Also, they mentioned that their accuracy increases when the number of vehicles they test on increases. Ready-made algorithms are not the only algorithms used in anomalies classification. Simple algorithms can reach an acceptable accuracy as well. In the research work [13], they used the Accelerometer in their mobile device and their own implemented classifier algorithm to classify potholes and bumps. They obtained an accuracy of 85.6%. In the research [14], their goal was to make an automated system without the interaction of humans. Accelerometer was used at the beginning along with a microphone from an Android smartphone to recognize the sound of potholes. The used algorithms in this system are z-diff, z-thresh, STDEV, and g-zero. This system accuracy was above 90%.

3. Implementation methodologies

3.1. Road Lane Detection

There are some methods and filters that are being used in order to detect the road lanes accurately. One of these methods used is canny edge detection as it includes different mathematical methods that identify the points that are used to detect edges in the captured video frames. The road lane detection technique is applied through some steps which are, applying the Gaussian filter to smooth the video frame as well as removing the noise, finding the intensity gradients of the image, getting rid of illusory response to edge detection, applying a threshold to determine possible edges. Then finally track detected edges and connect them. Filters such as grayscale and blurred are also being applied on the captured frames to smooth them in order to make it easier to apply some of the methods as well. In this stage, a Hough transform is used. It is a feature extraction technique used in image analysis techniques, computer vision field, and digital image processing as mentioned in [15]. The purpose of the whole methodology is to find the identification of lines in the digital image captured and draw a virtual path upon these lines to be the path that should be taken by the car.

$$r = xcos\theta + ysin\theta \tag{1}$$

Hough transform is defined by the equation 1 where r is the distance from the origin to the closest point on a straight line and is the angle between the x-axis and the line that is connecting the origin with that closest point. After lane detection is done and the lane is detected successfully. Mask lines are drawn upon the real lane to recognize the path that should be taken. Then, the raspberry pi sends an output of three signals which are right, left, and forward and decide to move according to the signal of the threshold value and angle theta detected.

3.2. Disparity Map

Disparity map is more like when eyes capture two different views of a three-dimensional object. Retinal images are fused in the brain in a way that their disparities are transformed into depth intuition, creating a three-dimensional representation of the object in the observer's mind. That is why The normal case of stereo vision arranges two cameras horizontally that is pointing in the same direction within a distance between them. This arrangement results in two different perspectives. Then, algorithms are applied to match corresponding points and store the depth information in a disparity map. To calculate the disparity map, the car had to go through two main phases pre-processing phase and processing phase.

3.2.1. Pre-processing

Collecting Images For calibration: Cameras are different and it's impossible to align them perfectly to create perfect depth maps. Thus, a software calibration method is used by two cameras that take multiple photos of an object. In our case, we used a printed chessboard. The calibration method will then analyze these photos by using edge detection to find the intersection points in the chessboard image to create a common view area of the two cameras.

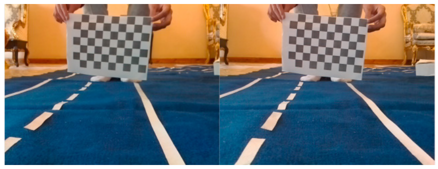
Fig. 1. Pair of a chessboard image for calibration.

Calibration: After collecting images and saving it, calibration then takes a place. Calibration works on detecting the common cameras view area by Loading all pairs capture and saved from the previous process and calculating the correction matrices. It first tries to find a chessboard on the photo. Then, it detects the intersection points of the black and white boxes in the chessboard to find the matching points in each camera and then build calibrated images. This calibration step refers to the quality of the calibration which results in either a good or a bad disparity map output.

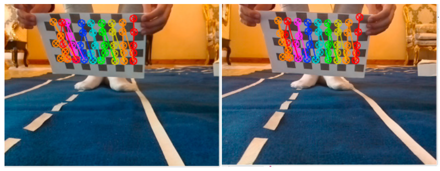
Fig. 2. Calibration quality.

Depth Map Tuning: It concatenates previous steps by loading the images that were saved along with the calibration results. Then, it presents the depth map and an interface for fine-tuning. In depth-map tuning, video frames are gathered from each camera after applying calibration results on it so that, it will be easy to set the appropriate disparity map parameters according to the results of the calibration which helps the disparity map to be more accurate.

3.2.2. Processing

In this phase, the real-time disparity map calculation takes place. The car simply moves while capturing live video frames and calculating the disparity map for each frame. Then, based on the output of the disparity map, the car motors take actions as if it should either slow down, keep on moving or stop. Real-time disparity map differentiates between distances by different colors on the live captured video frames using equation 2.

$$disparity = \frac{BF}{Z} \tag{2}$$

The disparity map is the distance between points in an image corresponding to the scene point three dimensions and their camera center. B is the baseline (distance between the two cameras), F is the focal length of the cameras, and Z is the perpendicular distance on the baseline. The depth of a point in a scene is inversely proportional to the difference in distance of identical image points and their camera centers. With this information, the depth of every pixel in a video frame is derived.

3.3. Anomalies Detection

There are two sensors required to read and collect the data of the car direction and rotation axes. These sensors are accelerometer and gyroscope. The accelerometer is a sensor that simply measures the acceleration force exerted upon it. The gyroscope is a sensor that is a bit more advanced than Accelerometer as it can measure the orientation and rotational velocity of itself. Therefore, we are using the MPU-6050 unit which is a small hardware unit that is composed of these two sensors. To accurately detect road anomalies, there are two main phases that the car had to go through. Pre-processing phase and processing phase.

3.3.1. Pre-Processing

In this phase, filtration and collection of the data from the sensors take a place. Firstly, the data of the Accelerometer and Gyroscope that consists of X, Y, Z for each of them was read from the raspberry pi as the MPU unit is directly connected to it with DuPont wires. Then, we used two filters:

- The low-pass filter to help by reducing the noise of sensor readings as it passes signals with lower frequency. Also, according to [16], they mentioned some benefits for low-pass filters like attenuation of high-frequency noise which will help us during the collection and the prediction of the data.
- Kalman filter which consists of mathematical equations that take the sensor data and estimate more accurate data. Kalman filter combines between the estimated states and the noisy measurements which is why it is optimal. Also, it is used to eliminate some types of noise that affect the readings of the sensors like momentum and different sensibilities for the Gyroscope and Accelerometer axes data. The data is simply passed through a low-pass filter and Kalman filter to reduce the noises exerted on it and to boost the sensor data.

3.3.2. Processing

After pre-processing, the classification takes a place where the data is obtained and classified using a support vector machine(SVM) algorithm with Radial Basis Function(RBF) kernel method. It was indicated in the research [17] that the support vector machine algorithm achieved the highest accuracy compared with other classification algorithms in most of the data sets used. The decision of using the RBF kernel method on SVM was taken according to [18], as they compared between SVM kernel methods on five different data sets. They showed that RBF SVM has achieved the highest accuracy compared with Linear SVM, Polynomial SVM, Sigmoid SVM. Also, after trying these kernels on our data set, it came out with the expected results that give the highest advantage of using RBF SVM for our data set as it has obtained 98.6% accuracy. SVM simply creates a hyperplane or a line that separates perfectly between our data into two classes. Anomaly class and normal road class. Using the RBF kernel method allows the algorithm to expand the number of dimensions that separate each record of the data into infinite dimensions if needed. Increasing this number of dimensions simply depends on $\frac{1}{2\sigma^2}$ value in equation 3 which equals the gamma value that the RBF kernel uses to increase or decrease the number of dimensions between the hyper plane points that separates between data. Also, $-\|x_i - x_j\|^2$ represents the squared euclidean distance between features X_i and X_j in the data set.

$$K(x_i, x_j) = e^{\frac{-\|x_i - x_j\|^2}{2\sigma^2}} \tag{3}$$

4. Experiments

4.1. Experiment Environment

‘ In our self-driving car environment, the track size has a length of four meters and a width of two meters. The car hardware platform has a length of twenty-one centimeters, a width of seventeen centimeters, and a height of twelve centimeters. The track is divided into two lanes to train and test the swerving of the car across each lane. Also, we used white tape for drawing these lines and curves as shown in figure 3. Moreover, the curves angle is approximately forty-five degrees. We use the oval-shaped lines in the track as they have more curves and will have a great impact on the experiments. The bump height is approximately four centimeters as displayed in figures 4, 5.

Fig. 3. Oval-shaped lined track.

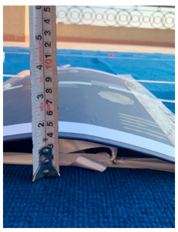

Fig. 4. Bump's length.

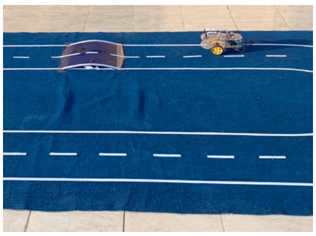

Fig. 5. Road bump.

4.2. Hardware Environment

In this section, an overview of our hardware platform is presented as shown in figure 6. The car consists of two motors, L298N H-Bridge which is a motor driver that allows a full control of the two car motors at the same time, and

Arduino UNO that simulate the car movement. The main component of the system is the raspberry pi that is connected to the MPU unit, Wi-Fi dongle, and stereo vision cameras. By all of these hardware components connected together, the system is able to accurately recognize its whole surrounding environment. As shown in figure 7, The two motors are being controlled by L298N motor driver based on the received signals from Arduino that is powered by an external nine-volt battery, and Arduino sends these signals based on the serials received from the raspberry pi. These serials are being transmitted on a serial path that is connected between Arduino and raspberry pi. Moreover, the algorithms take their input either as video frames or data readings from the cameras and sensors connected with the raspberry pi. The SD card is also a part of stereo pi as it has the Raspbian operating system of the raspberry pi.

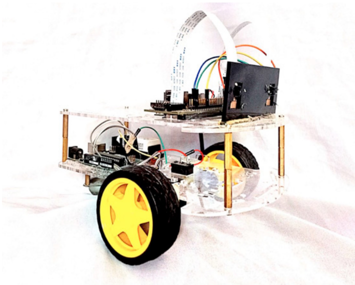

Fig. 6. Car Hardware Platform.

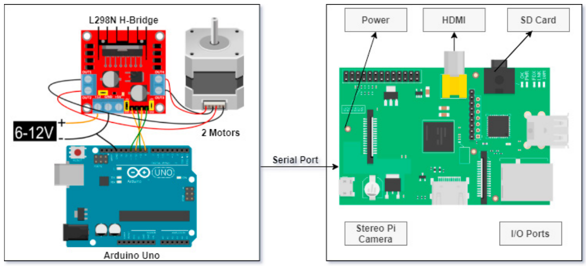

Fig. 7. Hardware Implementation.

4.3. Dataset

To gather our data set, the car was manually driven on our customized road to collect the data from the sensors using Bluetooth with the help of mobile to collect the data of road anomalies. The data was gathered based on eighty-six trails as our environment made it easier to collect the data without any difficulties. Collected data contains three thousand records of Accelerometer data merged with Gyroscope data as it was mentioned by Fazeen et al.[19] that when they used multiple axis classifications of their sensor data, they achieved higher accuracy.

The data set consists of GX, GY, GZ which are the data of the Gyroscope and AX, AY, AZ which are the data of the Accelerometer. Also, the data set has two classes. The first class with label "0" is the data that is classified as a normal road, and the second class with label "1" is for the data that is classified as road anomalies.

4.4. Experiment 1 anomalies classification

This experiment was done using four classification algorithms that are the most algorithms being used in the detection of road anomalies. As shown in table 1, We have used two data sets to test the algorithms on them.
Our environment data-set: A data set that we have collected on our road environment shown in figure 3. The SVM achieves the highest accuracy reaching 98.6%, followed by the decision tree, KNN, and finally, Naive Bayes.
Real-life data-set: A data set that we have collected on real streets. This data set is made of more than twenty-seven thousand records. The SVM achieves the highest accuracy as well by reaching 82.0%.

Table 1. Anomalies detection accuracy with different Classifications Algorithm on our data set and a real data set.

Classification Algorithm	NaiveBayes (t)	SVM (t)	KNN (t)	DecisionTree (t)
Our Data Set Accuracy	95.8%	98.6%	96.5%	97.9%
Real Data Set Accuracy	65.0%	82.0%	76.7%	78.1%

4.5. Experiment 2 lane detection

Two experiments were done on lane detection. The first experiment was done by using a mask, threshold, edge detection, local API, and some complex calculations. This method had some drawbacks that were challenging to us:

- It was not detecting the road lane accurately.
- It was slow in the recognition operation which led to deviation in the lanes.

Therefore, the second experiment succeeded as it is done by using blurry and gray scale filters, canny edge detection, and then Hough transform as a feature extraction. This method has some advantages:

- It is more accurate.
- It is faster in video frames recognition.
- It detects the right road lane as shown in figure 8. So, the car takes the appropriate decision and sends it to the Arduino to give the signal to the motors if the car should continue moving straight, move right or left.

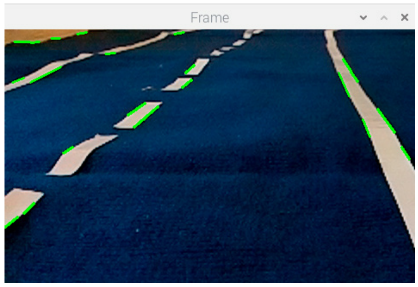

Fig. 8. Lane Detection.

4.6. Experiment 3 Disparity Map

In the begging, we tended to use the ultrasonic sensor to measure the distance between the car and any other obstacle, but it generates ultrasonic waves to the obstacle and waits to receive these waves back to measure the distance so, it takes time to measure the distance based on the distance of the obstacle. Besides, it did not work efficiently measuring long distances. So, we changed the ultrasonic with stereo-vision cameras using the disparity map algorithm. This experiment was done in our environment to test the disparity map technique and the distance between the car and an obstacle in front of it. The car was placed on the road with an object in front of it. The object was replaced with different distances to measure if the color of the object in the disparity map frames will change according to the object's distance. The results showed the following:

- If the object is from twenty-five to fifty centimeters away from the car, its color will be red.
- If the distance between the car and the object increased from fifty to a hundred centimeters, the object's color will be orange.
- If the distance is more than a hundred centimeters, the object's color will be yellow.
- If the distance is more than a hundred fifty centimeters, the object's color tends to be Blue.

This is illustrated in figure 9 (c) that shows the results of the disparity map's output.

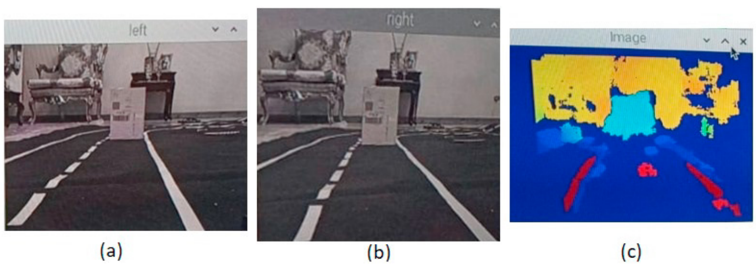

Fig. 9. (a) Left camera video frame. (b) Right camera video frame. (c) Disparity map output video frame.

5. Conclusion And Future work

To summarize this paper, The system aims to improve the safety of self-driving technology using new techniques. It is mainly about implementing a self-driving car that can make its own decisions accurately. The system is considered to be a prototype car that contains sensors and cameras to perceive the surrounding environment. The methodologies used in the system are lane detection using image filtering methods along with Hough transform feature extraction technique, anomalies detection using SVM classification algorithm with radial basis function, and distance measurement using disparity map. The result of lane detection is that the car moves accurately in its path according to the signals that are being sent to the motors from the road lane detection algorithm. For anomalies detection, the car is able to detect the road anomalies with an accuracy of 98.6%. Finally, for the results of the distance measurement, the car is able to make the right decisions either moving forward, slowing down, or stopping based on the disparity map algorithm output. The car was tested in an indoor and outdoor environment and its performance was very good. During the development process, we reached our goal which was to apply the idea on a small car prototype. So, we aim that our future work will consider applying this system in a real car.

References

[1] Chaocheng Li, Jun Wang, Xiaonian Wang, and Yihuan Zhang. A model based path planning algorithm for self-driving cars in dynamic environment. pages 1123–1128, 11 2015.

[2] Syed Ali Chishti, Sana Riaz, Muhammad Zaib, and Mohammad Nauman. Self-driving cars using cnn and q-learning. pages 1–7, 11 2018.

[3] Qudsia Memon, Muzamil Ahmed, Shahzeb Ali, Azam Rafique, and Wajiha Shah. Self-driving and driver relaxing vehicle. 11 2016.

[4] Mochamad Aziz, Ary Prihatmanto, and Hilwadi Hindersah. Implementation of lane detection algorithm for self-driving car on toll road cipularang using python language. pages 144–148, 10 2017.

[5] Ravi Satzoda, Suchitra Sathyanarayana, and Thambipillai Srikanthan. Hierarchical additive hough transform for lane detection. *Embedded Systems Letters, IEEE*, 2:23 – 26, 07 2010.

[6] Brilian Nugraha, Shun-Feng Su, and Fahmizal. Towards self-driving car using convolutional neural network and road lane detector. pages 65–69, 10 2017.

[7] Manoj Daigavane and Preeti Bajaj. Road lane detection with improved canny edges using ant colony optimization. pages 76 – 80, 12 2010.

[8] Florin Oniga, Ervin Sarkozi, and Sergiu Nedevschi. Fast obstacle detection using u-disparity maps with stereo vision. pages 203–207, 09 2015.

[9] Yihan Sun, Libo Zhang, Jiaxu Leng, Tiejian Luo, and Yanjun Wu. *An Obstacle Detection Method Based on Binocular Stereovision*, pages 571–580. 05 2018.

[10] Elena Bebeselea-Sterp, Raluca Brad, and Remus Brad. A comparative study of stereovision algorithms. *International Journal of Advanced Computer Science and Applications*, 8, 01 2017.

[11] Fatjon Seraj, Berend Jan van der Zwaag, Arta Dilo, Tamara Luarasi, and Paul Havinga. Roads: A road pavement monitoring system for anomaly detection using smart phones. page 128–146, 01 2016.

[12] Andrew Fox, B. Kumar, Jinzhu Chen, and Fan Bai. Multi-lane pothole detection from crowdsourced undersampled vehicle sensor data. *IEEE Transactions on Mobile Computing*, PP:1–1, 04 2017.

[13] Mohamed Fazeen, Brandon Gozick, Ram Dantu, Moiz Bhukhiya, and Marta C. Gonzalez. Safe driving using mobile phones. *Intelligent Transportation Systems, IEEE Transactions on*, 13:1462–1468, 09 2012.

[14] Artis Mednis, Girts Strazdins, Reinholds Zviedris, Georgijs Kanonirs, and Leo Selavo. Real time pothole detection using android smartphones with accelerometers. pages 1 – 6, 06 2011.

[15] Manoj Daigavane and Preeti Bajaj. Road lane detection with improved canny edges using ant colony optimization. pages 76 – 80, 12 2010.

[16] Richard Gregg, Sophia Zhou, James Lindauer, Eric Helfenbein, and Karen Giuliano. What is inside the electrocardiograph? volume 41, pages 8–14, 02 2008.

[17] Durgesh Srivastava and L. Bhambhu. Data classification using support vector machine. *Journal of Theoretical and Applied Information Technology*, 12:1–7, 02 2010.

[18] S. Chidambaram and K. Srinivasagan. Performance evaluation of support vector machine classification approaches in data mining. volume 22, 01 2019.

[19] Mohamed Fazeen, Brandon Gozick, Ram Dantu, Moiz Bhukhiya, and Marta C. Gonzalez. Safe driving using mobile phones. volume 13, pages 1462–1468, 09 2012.