

Concept Paper

Obstacle Detection and Safely Navigate the Autonomous Vehicle from Unexpected Obstacles on the Driving Lane

Malik Haris and Jin Hou *

School of Information Science and Technology, Southwest Jiaotong University Xipu campus, West Section, High-tech Zone, Chengdu 611756, Sichuan, China; malikharis@my.swjtu.edu.cn

* Correspondence: jhou@swjtu.edu.cn

Received: 5 July 2020; Accepted: 18 August 2020; Published: 21 August 2020



Abstract: Nowadays, autonomous vehicle is an active research area, especially after the emergence of machine vision tasks with deep learning. In such a visual navigation system for autonomous vehicle, the controller captures images and predicts information so that the autonomous vehicle can safely navigate. In this paper, we first introduced small and medium-sized obstacles that were intentionally or unintentionally left on the road, which can pose hazards for both autonomous and human driving situations. Then, we discuss Markov random field (MRF) model by fusing three potentials (gradient potential, curvature prior potential, and depth variance potential) to segment the obstacles and non-obstacles into the hazardous environment. Since the segment of obstacles is done by MRF model, we can predict the information to safely navigate the autonomous vehicle form hazardous environment on the roadway by DNN model. We found that our proposed method can segment the obstacles accuracy from the blended background road and improve the navigation skills of the autonomous vehicle.

Keywords: roadway hazard; Markov random field; autonomous vehicle; deep learning; image processing; self-driving car

1. Introduction

Global Status Report on Road Safety is released by World Health Organization (WHO, Geneva 27, Switzerland) 2018, in which WHO claims that about 1.35 million people die each year in road traffic accidents [1,2]. Similarly, American Automobile Association (AAA) Foundation released press report in 2016 that 50,658 vehicle roads accidents occurred only in America from the year 2011 to 2014 due to roadway obstacles. Roadway obstacles were the main factor of vehicle crashes and caused 9850 injuries and 125 deaths annually in the United States [3]. Reports indicate that over 90% of crashes are caused by errors of driver [4]. To improve this situation, governments, municipal departments, and car manufacture companies have considered significant investments to support the development of various technologies such as autonomous vehicles and cognitive robots. About 1 billion euros already have been invested by EU agencies on such type of projects [5].

In 2009, autonomous vehicles were developed and tested in four different states in the United States under the supervision of companies such as Waymo (Google, Mountain View, CA, U.S) and Uber with the support of traditional car manufacturers such as Ford and BMW [6]. Since then, this technology has been evolved and currently it is introduced in 33 different states of the United States with its specific regulations. In addition, Victoria Transport Policy Institute quoted that this technology will be widely used after 2040–2050 [7]. Currently, the most advanced features found in autonomous vehicles are Lane Changing (LC) Control [8,9], Adaptive Cruise Control (ACC) [10], Automatic Emergency

Braking System (AEBS) [11], Light Detection and Ranging (LIDAR) [12], Street Sign Recognition [13], Vehicle to Vehicle (V2V) Communication [14], Object or Collision Avoidance System (CAS) [15], etc.

With the continuous development of the highways, roads, city streets, and expressways, challenging problems are increasing to distinguish because the road environment is complex and constantly developing. It will be influenced by small obstacles or debris, shadows, light, water spots, and other factors. Those objects have been fallen from vehicles, construction sites or are littering. Different types of sensors, active sensors (RADAR or LIDAR) to passive sensors (camera), were used to solve this problem. Active sensors such as RADAR or LIDAR offer high precision in measuring distance and speed from point to point but they often suffer from low resolution and high costs. However, in comparison of passive sensors such as camera, its accuracy is crucial for the timely detection of small obstacles and an appropriate response by safety critical moving platforms. Detecting the small obstacle that is displayed in a very small area of the image with all possible shapes and forms is also very challenging problem. Gradient induced by the edges of obstacles can also be caused by paper or soil dust on the road or due to moisture gradient after a rain, mud, or road marking, which can be potential sources of false positives. Figure 2 describes the phenomena very well.

In recent research and development, Convolutional Neural Network (CNN) models are used in autonomous vehicles to navigate safely. For example, during training, the CNN-based end to end driving model maps a relationship between the driving behavior of humans using roadway images collected from stereo camera and the steering wheel angle [16–18], and during testing, the CNN-models predict the steering wheel angle to navigate the autonomous vehicle safely. So that the autonomous vehicle is depended on the training dataset. If the CNN model is not trained on the roadway obstacle than navigation system of autonomous vehicle may generate incorrect information about the steering wheel angle and cause a collision in result. In addition, the number of research studies shows that the autonomous vehicle navigation system may fail to navigate safely due to several reasons, such as Radar sensor failure, camera sensor failure, and software failure [19]. For example, Tesla Model 3 failing to stop for an overturned truck and slamming right into it on highway in Taiwan even ignores the pedestrian with autopilot on.

This study addresses how to improve the robustness of obstacle detection method in a complex environment, by integrating a Markov random field (MRF) for obstacles detection, road segmentation, and CNN model to navigate safely [20]. We segment out the obstacle from the image in the framework of MRF by fuses intensity gradient, curvature cues, and variance in disparity. After analyzing the obstacle from the captured image, CNN-model helps to navigate the autonomous vehicle safely. The main contributions of this study are as follows:

- Pixel label optimization of images as a small obstacle or hindrance on the road detected by using an MRF model.
- Navigating an autonomous vehicle on a roadway from unexpected obstacle.

The remaining part of the research is organized as follows:

- Section 2—Reviews the relevant works carried out and developed in the past few years.
- Section 3—Introduces the method for detecting the physical obstacles or hindrances on the road and predicts the steering wheel angle for AV.
- Section 4—Shows demonstration and simulation.
- Section 5—Discusses the results and its comparison.

2. Related Work

The probability of occupation map is one of the main directions of work for obstacle detection [21]. It is developed through orthogonal projection of the 3D world onto a plane road surface (assuming that the environment structured of the road surface is almost planar). The plane is discretized in cells to form a grid; therefore, the algorithm predicts the probability of occupation of each cell. We conclude

that this method can accurately detect the large obstacle (such as cyclists or cars) by using the stereo vision sensor [22] and can also help to identify the road boundaries by using LIDAR data [23]. However, since the probability function is closely related to the number of measurements in one grid, this method may not be suitable for small obstacle detection by using the stereo vision if the observation is scarce and noisy.

Digital Elevation Map (DEM) [24] is one of the algorithms that tries to detect obstacles relying on the fact that they protrude up from a dominant ground surface. The obstacle detection algorithm proposed in [24] marks DEM cells as road or obstacles, using the density of 3D points as a criterion. It also involves fitting a surface model to the road surface. Oniga and Nedevschi [25] presented a random sample consensus (RANSAC)-based algorithm for road surface estimation and density base classifier for obstacle detection by using the DEM constructed on stereo data, whereas in [26], authors used a similar RANSAC-based methodology for curb detection by using the polynomial fitting in stereo DEM. Although, RANSAC-based algorithm is not suitable for detection of small obstacle because the small obstacle and variation of disparity in the image is often similar to the noise levels and a least square fitting may smooth out the small obstacle along with road surface, RANSAC shows an accurate estimate of the vehicles position relative to the road [27].

Fusion of 3D-LIDAR and high-resolution camera seems to be a reasonable approach for robust curb detection. Traditional range visual fusion methods [28,29] use the detection results from the range data to guide the curb searching in the image, which have the advantage of enlarging the detection spot. However, fusion of 3D-LIDAR and visual data search small obstacle in the visual images is more reliable in enhance the robustness of obstacle and edge detection as compared to traditional range visual fusion methods [30]. It recovers a dense depth image of the scene by propagating depth information from sparse range points to the whole high-resolution visual image [31]. J. Tan and J. Li et al. [32] used the following geometric properties to robustly curb detection such as depth image recovery, curb point detection, curb point linking and curb refinement, and parametrization.

Scene flow-based approach is also used for obstacle detection [33,34], where each point in constructed 3D cloud is tracked temporally and the flow is analyzed to classify the object in the image such as road surface, obstacle, pedestrian, etc. This approach has limited applicability because it just detected the moving obstacles such as moving vehicle, bicycle, or pedestrians. In addition, the decision-making process is based on the flow of nearby points in 3D cloud, which can be too sparse for small obstacles. For obstacle detection in [35], authors used advanced stereo-based algorithms, combining full 3D data with motion-based approach, and yet it focuses only on large obstacle detection such as vehicles.

Hadsell et al. [36] presented the method of vision-based neural network classifier. His idea was to use an online classifier, which is optimized for long-distance prediction and deep stereo analysis to predict the obstacles and to navigate vehicles safely. Ess A. et al. [37] defined the method of combining image-based category detectors with its geometric information received from stereo cameras (such as vehicle or pedestrian detection). However, due to the large differences in its shape, size, and visual look, it will become difficult for vision sensor to train the dataset on small obstacles along the way.

Bernini and Bertozzi et al. [38] proposed an overview of several stereo-based generic obstacle detection such as Stixels algorithm [39] and geometric point clusters [40,41]. The Stixels algorithm distinguishes between a global ground surface model and a set of obstacle segments in rectangular vertical, thus, providing a compact and robust representation of the 3D scene. However, geometric relation between 3D point is used to detect and cluster obstacle point. This method is suitable for detecting medium-sized obstacle over close or medium distance. If the distance increases and the size of the obstacle decreases, then position accuracy and obstacle detection become challenging.

Zhou J. and J. Li [42] proposed a solution to detect obstacles using a ground plane estimation algorithm based on homography. This work extends [43] to a smaller obstacle by combining several indicators, such as homography score, super-pixel segmentation, and line segment detector in an MRF framework. However, such indexation based on appearance such as line segment detection and pixel

segmentation fail when they occurred. Therefore, we directly use the lowest level of details available, i.e., image gradients in both stereo depth images and appearance.

Many researchers [44–46] work on the hypothesis of a flat surface assumption, free space, or the ground as a single planar and characterize the obstacles according to their height from the ground. The geometric deviation from the reference plan can be estimated directly from the image data, precalculated point cloud [47], or by extracting from v-disparity histogram model [48]. Instead of relying on the flatness of the road [49], the vertical road profile is modeled as a clothoid [50] and splines [51], which is estimated from the lateral projection of the 3D points. Similarly, free ground profile model was examined using adaptive threshold values in v-disparity domain and multiple filter steps [52].

In addition, we also explore existing datasets that are used in autonomous vehicle community. The dataset is provided by the Udacity (Mountain View, CA, USA) [53], which supports the end-to-end data and image segmentation, but it does not provide the ground truth for the obstacle in the roadway lane. Similarly, KTTI [54] and Cityscape [55] datasets also do not support the obstacle detection as a ground truth data. The dataset which matches our requirements is the Lost and Found dataset [56].

Pinggera P. and Kanzow C. have worked on planar hypothesis testing (PHT) [57,58], fast direct planar hypothesis testing (FPHT), point compatibility (PC) [40,41], Stixels [39,59], and mid-level representation: Cluster-Stixels (CStix) [60–62] for detecting the small obstacle in dataset. In addition, Ramos and Gehrig et al. [63] further extends this work by merging the deep learning with Lost and Found dataset hypothesis results. The people who have worked on Lost and Found dataset [56] did not discuss that how to navigate an autonomous vehicle safely or predict the steering wheel angle for an autonomous vehicle from unexpected obstacle on a roadway.

Since the existing dataset is not able to meet our requirements, we have carefully created our dataset using the steering angle sensor (SAS) and ZED stereo device mounted on an electric vehicle as discuss in the experimental setup section. After that, pixel labelling of small obstacles or hindrance on the road are detected by using MRF model and road segmentation model, which help the CNN-model to navigate the autonomous vehicle on roadways from unexpected obstacle by prediction the steering wheel angle.

3. Method

In this section, we will describe how to develop a safe autonomous driving system in unexpected and dangerous environments. In this section, we discussed three models.

- (a) In the first model, we used various stochastic techniques (such as curvature prepotential, gradient potential, and depth variance potential) to segment obstacles in the image from Markov random field (MRF) frames. These three techniques measure pixel-level images to extract useful information and store it for orientation. In this method, each pixel in the node of interest was distributed in the MRF. Finally, instead of using OR gates, we used AND gates to combine the results of previous techniques.
- (b) In the second model, semantic segmentation technology was used to segment paths and filter outliers and other important obstacles.
- (c) Third model was used to predict the steering wheel angle of the autonomous vehicle. We analyzed the unexpected obstacle on the roadway and determined the threat factor (T_{ff}). This threat factor helped us to ignore that obstacle or consider as accident risk.

The overall pipeline is graphically illustrated in Figure 1. The system input consisted of optical encoders used in applications for angle detection as a SAS in vehicles and two stereo images, which were used to calculate depth using the SGBM algorithm (semi-global block matching) proposed by Hirschmuller [64]. The depth variance and color images were used to calculate three different cues, i.e., image gradient, disparity variance, and depth curvature. These three cues were combined to a unary potential in an MRF with an additional pairwise potential. Then, the standard graph cuts were

used to obtain the obstacle. In addition, deep learning-based semantic segmentation [65] was used to segment the roads and filter out the abnormal values.

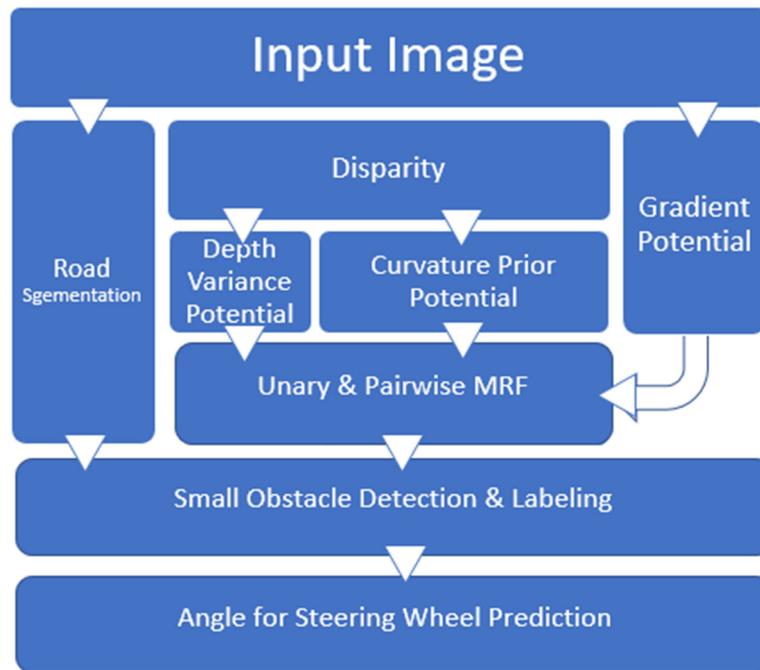


Figure 1. Pipeline of detection of hazard object and prediction of steering wheel angles.

3.1. The Markov Random Field (MRF) Model

During the past decade, various MRF models, inference, and learning methods have been developed to solve many low-, medium-, and high-level vision problems [66]. While inference concern underlying the image and scene structure to solve problem such as image restoration, reconstruction, segmentation, edge detection, 3D vision, and the object labeling in the image [67]. In our research work, we elaborate our query regarding small obstacle segmentation from the road by defining an energy function (such as cost) over an MRF. We associate the image with random process X with the elements X_s , where $s \in S$ represents a position of a pixel in the image. Each pixel position in the image is considered as a node in the MRF and affiliates each node with the unary or pairwise cost [68]. Minimum energy function E is defined in equation as:

$$E(X) = \sum_s E_u(s) + \sum_{(r,s) \in N} E_p(X_r, X_s) \quad (1)$$

where $E_u(s)$ represents the unary term and $E_p(X_r, X_s)$ represents the pairwise term. $X = \{X_1, X_2, \dots, X_n\}$ is set of random variables associated with the set of nodes S that takes label $X_s \in [0, 1]$, which depends on the nature of appearance, either it is road, texture, or the small obstacle on the road. Around each pixel, we calculate the unary term ($E_u(s)$) independently. It is the combination of three cues such as gradient potential ($E_u^g(s)$), depth curvature potential ($E_u^c(s)$), and depth variance potential ($E_u^v(s)$). The unary term ($E_u(s)$) is defined as follows:

$$E_u(s) = E_u^g(s) \cdot E_u^c(s) \cdot E_u^v(s) \quad (2)$$

3.2. Gradient Potential

The potential gradient at the site (i, j) is defined as follows:

$$E_u^g(i, j) = \sqrt{G_x(i, j)^2 + G_y(i, j)^2} \quad (3)$$

Partial derivative ($G_x(i, j)$ and $G_y(i, j)$) is calculated in the horizontal and vertical direction in the original color image [69]. In our work, we use the image gradients rather than edge detectors, because edge detectors mostly perform the thresholding on the feeble gradient response, whereas our work avoid such types of thresholding, and it builds several weak cues as a strong cue.

3.3. Curvature Prior Potential

Curvature can be defined as the reciprocal of the radius of a circle that is tangent to the given curve at a point. Curvature is “1” for a curve and “0” for a straight line. Smaller circles bend more sharply and hence have higher curvature. This feature is very useful for autonomous vehicle in multiple purposes such as free space detection, curb detection, etc. [70,71]. When performing an Eigen analysis near the 3D point under consideration, the algorithm greatly quantifies the changes along the normal surface. Equation of the curvature potential is defined:

$$E_u^c(s) = \frac{\sqrt[3]{\lambda_3}}{\sqrt[3]{\lambda_1} + \sqrt[3]{\lambda_2} + \sqrt[3]{\lambda_3}} \quad (4)$$

where $E_u^c(s)$ is a curvature prior at the corresponding position obtained after reprojection the 3D point in the image space. The $\sqrt[3]{\cdot}$ of the eigenvalue ($\lambda_1, \lambda_2, \lambda_3$) can better distinguish nonplanar surfaces. This technique is little different than the M. Pauly and M. Gross et al. [70] algorithm. The curvature prior feature is stable and robust and stable than tangent plane normal vectors because it does not make a plane surface assumption very clearly.

3.4. Depth Variance Potential

If we slide horizontally on the image, we can calculate this potential to detect sudden changes in depth [72]. This is computed by summation of these sudden depth horizontal windows (W_h) in a square window (W_s) and multiplying the figured-out value with corresponding disparity in the pixel, even the obstacles are not available. Equation of depth variance is defined as:

$$D_{var}(i, j) = var\left(\left[D\left(i - \frac{W_h}{2}, j\right) : D\left(i + \frac{W_h}{2}, j\right)\right]\right) \quad (5)$$

Here, the depth map is represented by D , which we obtained with the help of stereo equipment. The final variance potential is defined as follows:

$$E_u^v(i, j) = \left(\sum_{a=-W_s/2}^{W_s/2} \sum_{b=0}^{W_s} D_{var}(i+a, j-b) \right) \cdot d(i, j) \quad (6)$$

Here, parallax value of the specific located pixel (i, j) is represented by $d(i, j)$.

The results of above three potentials are shown in Figure 2. Thus, we can see that the curvature prior potential and depth variance potential are reliable. However, adding some noisy depth values to the stereo will result in false positive values, as shown in Figure 2b,c. To get the more accurate unary potential result, as seen in Figure 2e, weighting the gradient potential in Figure 2d can pacify the problem by using the depth potential.

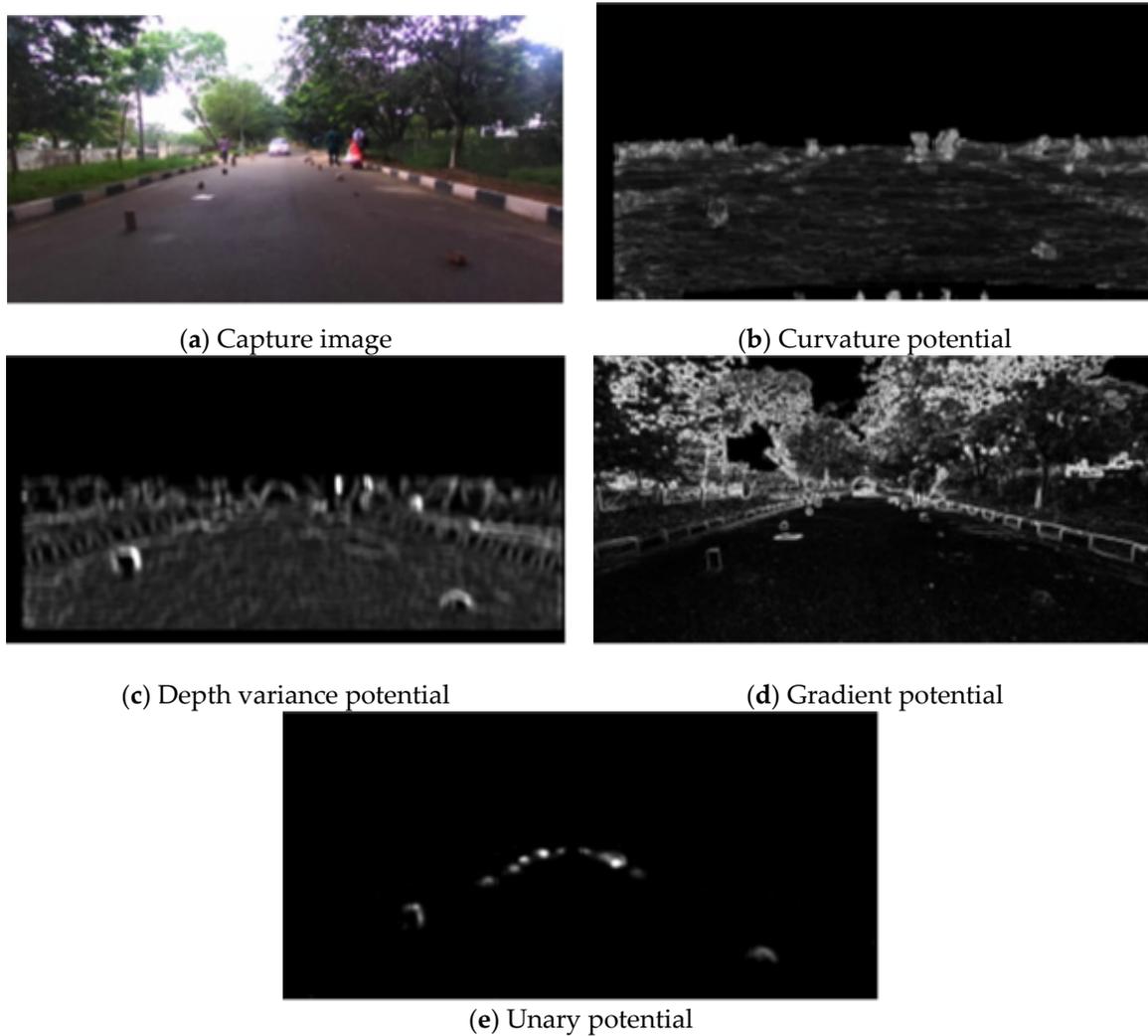


Figure 2. Result extracted from capture image by applying following potentials.

3.5. Pairwise Potential

By using the Potts model, we define the pairwise potential $E_p(X_r, X_s)$:

$$E_p(X_r, X_s) = -J_p \sum_{(r,s)} \delta(X_r, X_s) \quad (7)$$

In the above equation, Kronecker delta is represented by $\delta(X_r, X_s)$; whenever $X_r = X_s$, the value is 1 otherwise 0. The equation of finding the global minima of energy function of the small obstacle that is detected on the road is:

$$X^* = \underset{x}{\operatorname{argmin}} E(X) \quad (8)$$

With the help of graph cut feature, we can find the global minima efficiently, whereas minimum graph cut is found using Boykov Y. and Kolmogorov V. et al. [73] procedure of min-cut/max-flow. Appendix A provides the pseudocode for obstacles detection by using the MRF-model.

3.6. Determination of the Obstacle Threat Value in the Image

In this model, we determine the threat value (T_{tf}) by detecting obstacles in the capture image. Coordinate (x, y) of the obstacle in the capture image is identified once we combine the result of

semantic segmentation model and MRK mode. We can compute the threat value (T_{tf}) by measuring the distance of the obstacles in the image size (*height* (h) and *width* (w)) from the bottom center pixel ($h, \frac{w}{2}$). Finally, the threat value is computed from the following formula:

$$T_{tf} = 1 - \sqrt{\frac{(x-h)^2 + (y-\frac{w}{2})^2}{h^2 + (\frac{w}{2})^2}} \quad (9)$$

To obtain the latitudinal and longitudinal distance of the obstacle from the center line, we subtract the height (h) with half of the width ($\frac{w}{2}$) values from x and y values, respectively. Code related detection of obstacle in the captured image whose pixels size ≥ 50 along with its coordinate specified in Appendix A.

3.7. DNN-based Autonomous Vehicle Driving Model

In our research, we have implemented an autonomous driving model similar to DAVE-2, i.e., an end-to-end autonomous driving model [18]. Here, we train the weights of the network to minimize the average square error between the steering command issue. As shown in Figure 3, the network receives an obstacle detection and labeling image of $400 \times 600 \times 3$ pixels and generates a steering wheel angle as output. This network architecture consists of 11 layers, including 1 lambda layer, 1 normalization layer, 5 convolution layers, and 4 fully connected layers. We use 5×5 kernel and 2×2 stride in the first three convolution layers and 3×3 kernel and 1×1 stride used in last two convolution layers. The whole network contains 7,970,619 trainable parameters.

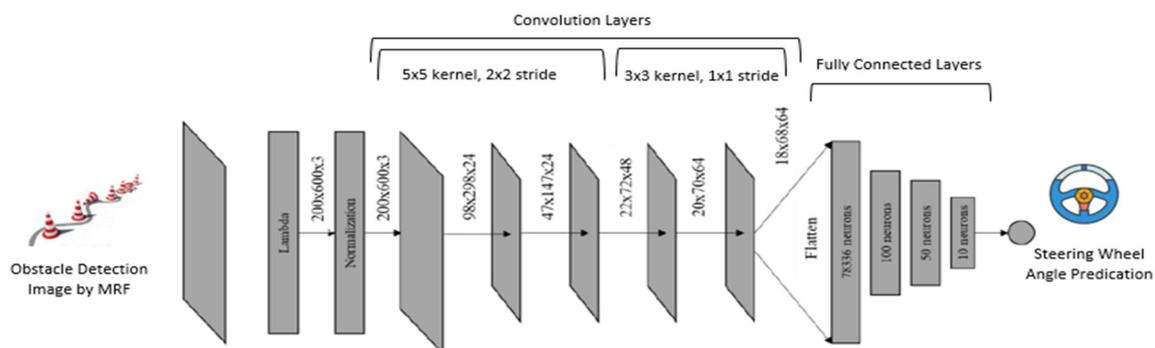


Figure 3. CNN autonomous vehicle driving model used for steering wheel angle prediction.

We train autonomous vehicle models based on the result of obstacle detection by MRF and SAS and then test their performance through prediction of steering wheel angle as describe in Algorithm 1. After training, our well-trained autonomous vehicle model detects high threat value obstacle on the roadway and navigate the autonomous vehicle safely. Appendix A shows the model generation code used for steering wheel angle prediction.

Algorithm 1 Pseudocode for Predicting the Steering Angle

```

#Nvidia Model
Lambda: Output shape: 400 × 600 × 3
Image normalization to avoid saturation and make gradients work better.
#2D Convolution Neural Network for handle features.
Convolution1: 5 × 5, filter: 24, strides: 2 × 2, activation: ELU
Convolution2: 5 × 5, filter: 24, strides: 2 × 2, activation: ELU
Convolution3: 5 × 5, filter: 48, strides: 2 × 2, activation: ELU
Convolution4: 3 × 3, filter: 64, strides: 1 × 1, activation: ELU
Convolution5: 3 × 3, filter: 64, strides: 1 × 1, activation: ELU
#Dropout avoids overfitting
Drop out (0.5)
#Fully Connected Layer for predicting the steering angle.
Fully connected 1: neurons: 100, activation: ELU
Fully connected 2: neurons: 50, activation: ELU
Fully connected 3: neurons: 10, activation: ELU
Fully connected 4: neurons: 1 (output)
model.compile(Adam(lr = 0.0001), loss='mse')
return model

```

4. Dataset

Since little attention has been paid to finding small obstacles in the past, no reasonable dataset is available for public use. Therefore, we created a new small dataset for obstacle detection. The dataset contains 5626 stereo images. These images are obtained from *ZED stereo device*, which were installed on electric vehicles in various environments. ZED camera can create the real-time point cloud at a framerate of 30 fps with the resolution of $2 \times (1280 \times 720)$, and we extracted them from different video sequences of electric vehicle view, while running on road. Specification of the ZED camera is presented in Table 1.

Table 1. Spec/parameters of ZED stereo device.

Camera	ZED M
Frame per second	30 fps
Resolution	$2 \times (1280 \times 720)$
RGB sensor type	1/3" 4MP CMOS
Field of view	Max. 90° (H) \times 60° (V) \times 100° (D)
Exposure time	Set exposure to 50% of camera framerate
Focal length	2.8 mm (0.11")—f/2.0
Interface	USB 3.0 Type-C port

Optical encoders or separate arrays of LEDs and photodiodes are used as steering angle sensor (SAS) in vehicle angle sensing applications [74,75]. Accurate and reliable steering angle detection is the main challenge of modern vehicles. This data is used not only for electronic power steering (EPS) and electronic stability control (ECS) but also for controlling adaptive headlights, lane keeping assist system, or other advanced driver assistance systems (ADAS). Optical steering angle sensor consists of an LED and a photodiode array or a CMOS line sensor. The sensor output is a digital square wave signal whose frequency depends on the speed of the turning wheel. Thus, the sensor can help determine the actual angle of rotation (ground truth value), which allows us to compare it with our

proposed model. There are some mathematical tools that can adjust the steering angle values between different vehicles according to the formula of Equation (10):

$$R = \frac{s}{\sqrt{2 - 2 \cos\left(\frac{2a}{n}\right)}} \quad (10)$$

where “R” is the radius of curvature, “a” is the steering wheel angle, “n” is the steering ratio, and “s” is the steering wheel frame. The onboard computer is NVIDIA Jetson TX2. In the training phase, the captured images from ZED stereo cameras associated with steering angle are sent to this main computer and saved frame by frame as a custom image object. Later, the training can be accomplished offline using a more powerful computer than the onboard computer.

According to the steering angle sensor, we have many zero steering values and only a few left and right steering values. In fact, we mostly go straight, if we put all of them equal, then it will go left and right frequently. We cutoff 4402 images with zero values, which help us to balance the dataset, and the remaining 1224 stereo images are used to train the model as shown in Figure 4.

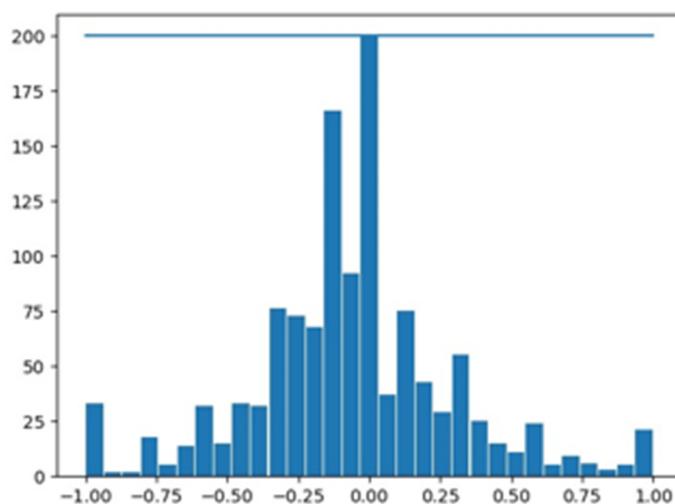


Figure 4. Balancing of the dataset shown in histogram graph.

Many obstacles can be found in this dataset, such as bricks, dirt, newspapers, animals, and pedestrians, and even, we can see the nature of the road to provide driving assistance and more information. Naturally, there are many obstacles on the road, while we have deliberately placed a few obstacles to further complicate our dataset. There are total of 52 obstacles found in this dataset with variable height (8–45 cm). In addition, we also create the ground truth by manually driving the electric vehicle on the roadways. In order to obtain satisfactory driving model performance, it is necessary to train the model on multiple training dataset. Increasing the size of existing data through affine transformation [76], specifically by random change in contrast or brightness, horizontal flipping of the image, and random rotation. After development of autonomous vehicle driving model, we use the augmented dataset for training. The pseudocode for augmented dataset is included in Appendix A. This dataset is divided into two parts, 80% for training and 20% for validation, as shown in Table 2.

Table 2. Dataset distribution for training and validation.

	Proposed Method	Dataset Size after Expansion
Size of dataset	1224	2448
Size of training dataset	980	1958
Size of validation dataset	244	490
Size of testing dataset (contain obstacle)	52	104

5. Results and Comparisons

We can use the following quantitative indicators to analyze the performance of the model: root mean square error (RMSE) and mean absolute error (MAE), as well as qualitative indicators through visualization: appearance challenges, distance and size challenges, clustering and shape challenges, and prediction of steering wheel angle challenges.

5.1. Quantitative Results

5.1.1. Model Performance

Through our method and the measurement of RMSE and MAE, quantitative comparisons can be made to successfully identifying the obstacles. By comparing the RMSE and MAE values, we can predict the angle of the steering wheel. If the number of pixels of the obstacle is greater than the 50 pixels marked by the MRF model, the obstacle can be identified. If the pixel size of the segmented obstacle is higher than 50 pixels, it will be marked as a false alarm or high threat value and label it on the autonomous vehicle lane.

After successfully detecting the obstacles on the roadway by using the MRF model, we extracted the following two errors, root mean square error (RMSE) and mean absolute error (MAE), by which we can compare ground truth value with predicting steering wheel angle. For extracting these errors, we used following equations:

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (G_i - P_i)^2} \quad (11)$$

$$MAE = \frac{1}{N} \sum_{i=1}^N (|G_i - P_i|) \quad (12)$$

Total number of images in the dataset is represented by “ N ”, ground truth represented by “ G_i ”, and the steering wheel angle represented by “ P_i ” for the image i^{th} of the dataset.

We find out the RMSE and MAE values in different weather conditions and different times of the day and compare it with the ground truth value of the electric vehicle. In Figure 5, RMSE and MAE values are low in the following situation: daytime, shadow, and street light, while the error values are high in the following situations: night and rainy weather. The RMSE and MAE values are less than the value indicating that the predicted steering wheel angle closely follows the ground truth values.

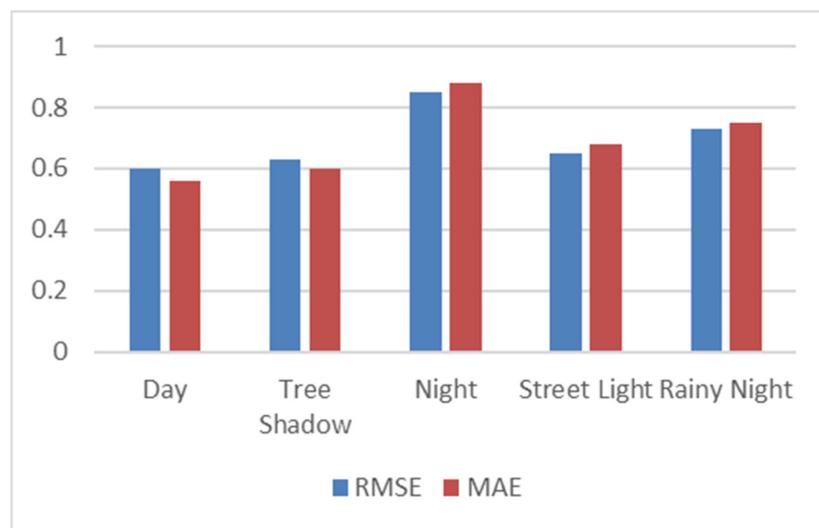


Figure 5. Measurement of root mean square error (RMSE) and mean absolute error (MAE) in different situation.

5.1.2. Path Follow

The Frenet Coordinate System [77] is used to analyze the situation avoiding the obstacles on the roadway of autonomous vehicle. We did not follow the Frenet Coordinate System for performance evaluation, but compared the ground truth value and RMSE value with time steps from 72,000 to 82,000 ms (represented on x -axis) and latitude (represented on y -axis). It can help us to understand that how autonomous vehicle closely follow the ground truth trajectory data and avoid the obstacle on the road as shown in Figure 6. As we can see in Figure 6, in the following situation: daytime, shadow, and under street light, the autonomous driving model closely follows the ground truth values while difference between the night and rainy weather is high. When camera performance declines at night or in rainy weather, it is a challenge to identify obstacles in such situations. Appendix A mentions code related to extracting the ground truth and predicting RMSE values of steering wheel for comparison.

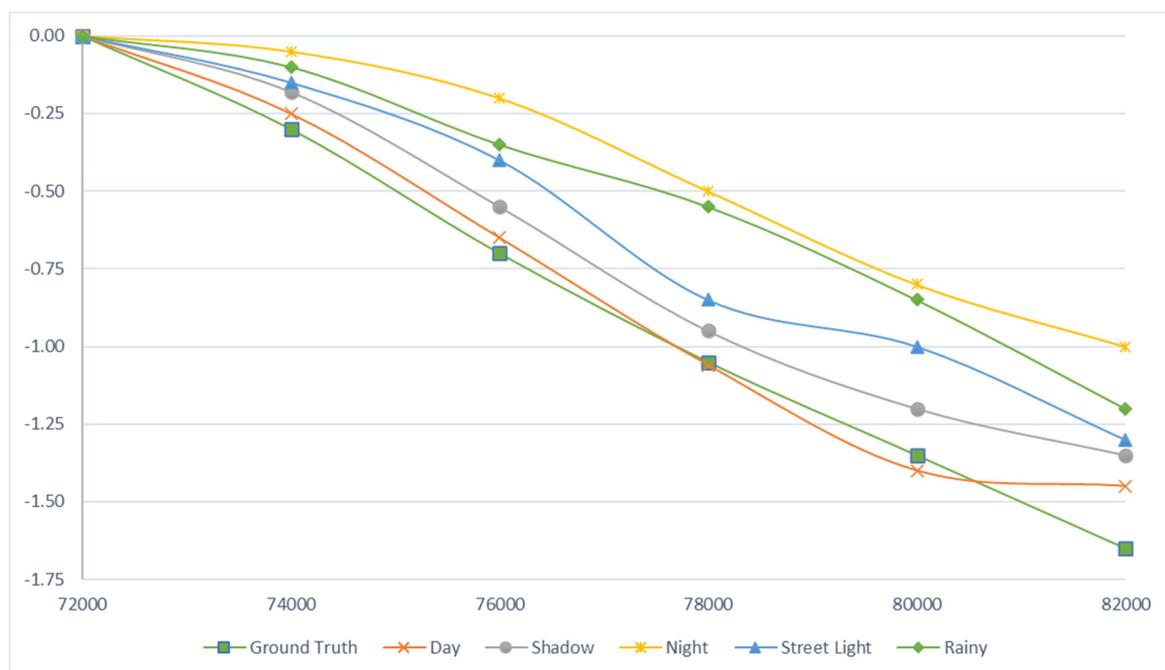


Figure 6. Comparison of the trajectory of the ground truth value with the RMSE value in different situation.

5.2. Qualitative Results

Here, we will examine methods implemented for proposed dataset. We categorize them based on challenges related to appearance, distance and size, cluttering and shape, and steering wheel angle prediction.

5.2.1. Appearance Challenges

The appearance is challenging to see in two ways. First, the deceptive texture by dirt or mud on the road due to rain or flat object (such as newspaper and cardboard box) is very different from other objects (such as rock and brick) that may mislead autonomous vehicle. Second, the appearance of obstacles seems to be its background. It is very challenging to detect obstacle due to weak intensity gradient. However, the method we propose can solve the above problems and robustly extract obstacles. As you can see in Figure 7d, we can accurately identify obstacle even if the road is wet. Likewise, it can avoid the objects such as mud and shadow in Figure 7c. It also helps to identify obstacles that look like road (obstacle 4 in Figure 7d). Our method can also take into account local changes in the road structure and even applies to cement floors (Figure 7a,c).



Figure 7. Obstacles in the captured image successfully identified by Markov random field (MRF) model.

5.2.2. Distance and Size Challenges

As we all know, stereo depth is unreliable after the range of 10–12 m, which makes obstacle detection challenging at long distance. Detecting the small obstacles from distance, large changes in noise, or light reflection can cause serious problem. Our model solves these two problems well. As shown in Figure 7e, label 1 shows that our method has detected a 9 cm height obstacle at a distance of 8 m. Similarly, in Figure 7a, an obstacle of 7 cm height is detected at a distance of 7 m.

5.2.3. Cluttering and Shape Challenges

Obstacle appear on the road in random shapes. Therefore, it makes sense to only allow certain shapes to use for modeled. However, our proposed algorithm has advantages in this regard because it does not require any assumptions about the shape or appearance of obstacles and always performs reliable detection. We can be seen rocks and tires in Figure 7b,c, respectively.

5.2.4. Prediction of Steering Wheel Angle

The rule-based classification method is used to divide all obstacles into two categories: (a) the ones to be ignored (the pixel size of obstacle is less than 50) and (b) which causes of vehicle to collide (the pixel size of the obstacle is greater than 50). We use MRF model and segmented image to calculate

the threat value using equation 9 as discuss in methodology section. After pixel segmentation of obstacles, the coordinates of the risky obstacles in the image are obtained.

After collecting the data, we prepare the image dataset for end-to-end driving model training by normalizing and resizing of the images [78,79]. As shown in Figure 8, the steering wheel angle is normalized between the -1.0 to $+1.0$, where positive values represent right rotation and a negative value represents the left rotation of the steering wheel. We normalize the values by using the liner transformation Equation (13):

$$\theta_{normalized} = -1.0 + \max\left(0, \min\left(1.0, \frac{\theta_{raw} - \theta_{min}}{\theta_{max} - \theta_{min}}\right)\right) \quad (13)$$

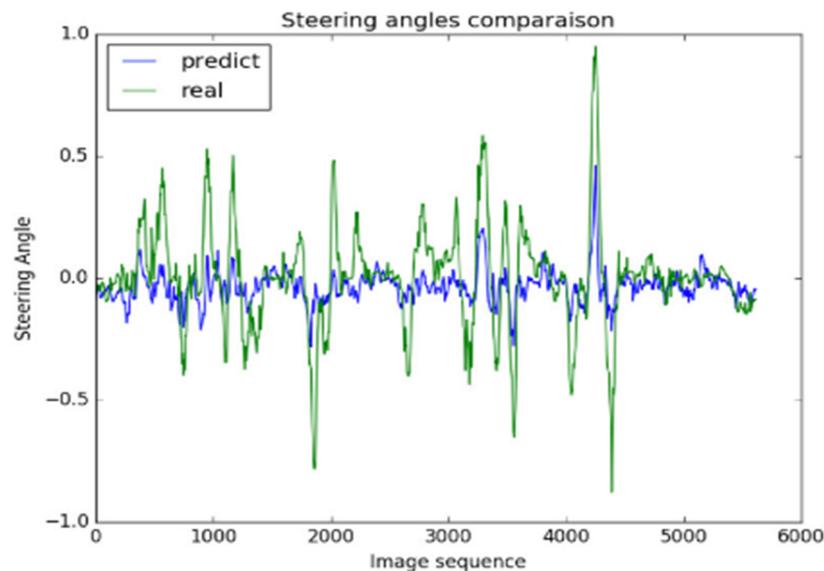


Figure 8. Comparison of ground truth steering wheel angle (real) with predicting steering wheel angle by model (predict).

Among them, $\theta_{normalized}$ is the normalized steering wheel angle between -1.0 and $+1.0$, θ_{raw} is steering wheel angle in radians, and θ_{min} and θ_{max} are the minimum and maximum steering wheel angle, respectively.

In the Figure 8, we have plotted the “steering angle” and “image sequence” on the y-axis and x-axis, respectively. Figure 8 shows that how closely proposed model follows the ground truth values. Using our model, we conducted a statistical significance test between the ground truth values and prediction values; we obtained a 91.5% confidence interval.

The accuracy of the prediction can be qualitatively represented by observing the direction of the driving angle of an autonomous vehicle as shown in Figure 9. As shown in Figure 9, when there are obstacles on the road, our proposed model can predict that the angle of the steering wheel will be closer to the ground truth angle. In Figure 9, the green line represents the ground truth angle and red line represents the expected angle of the steering wheel of our proposed model.

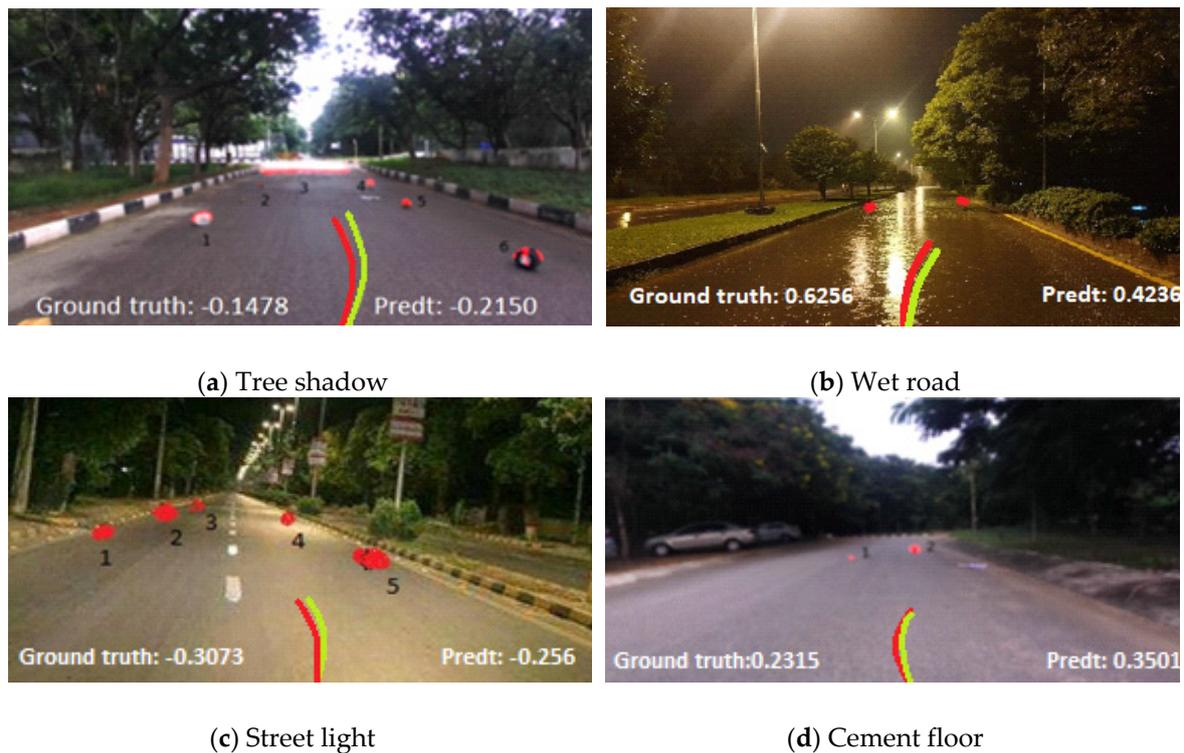


Figure 9. Comparison of ground truth angle with proposed model angle with random dataset images in different environment.

6. Conclusions

This research addresses how the robustness of obstacle detection method in a complex environment, by integrating a Markov random field (MRF) for obstacle detection, which is left intentionally or unintentionally on roadway, road segmentation, and CNN-model to navigate the autonomous vehicle safely. Our designed model predicts the steering wheel angle after the high threat value found of obstacle on the roadway of autonomous vehicle for both manual and self-driving vehicle. We conclude on the basis of analysis that our approach improves the safety of autonomous vehicle or human driving vehicle in terms of risk or collision with obstacles on the road by 35% as compared to vision-based navigation system rather than those who do not have the capability of detecting the high threat value obstacle on the road from long distances.

Author Contributions: In this research activity, M.H. participated in the creation of datasets, model generation, simulation and analysis of results. J.H. reviews and monitors to improve the overall work and analysis. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

```
def load_image(data_dir, 2 x ZED stereo image_file):
    """
```

Load images from a file

Crop the image (removing the sky at the top and the car front at the bottom)

Resize the image to the input shape used by the network model

Convert the image from RGB to YUV (This is what the NVIDIA model does.)

Randomly flip the image left <-> right, and adjust the steering angle.

Randomly adjust brightness of the image.

```

    Randomly shift the image virtually and horizontally (translation).
    Generate an augmented image and adjust steering angle.
    (The steering angle is associated with the center image.)
    """
def MRF(augmented_images):
    """
    Disparity of augmented images
    Depth variance potential result 1.
    Curvature prior potential result 2
    Gradient potential result 3.
    Combine all required results 1, 2 and 3 by AND Gate
    """
def batch_generator(data_dir, image_paths, steering_angles, batch_size, is_training):
    """
    Generate training image give image paths and associated steering angles
    """
    #Stereo Disparity Map Code.
    # -----
    def stereo_disparity_map(rectified_pair):
    #image location of an object seen by the left and right image of a stereo pair
    global disparity_max
    global disparity_min
    ZEDLeft = rectified_pair[0]
    ZEDRight = rectified_pair[1]
    disparity = sbm.compute(ZEDLeft, ZEDRight)
    local_max = disparity.max()
    local_min = disparity.min()
    if (dm_colors_autotune):
    disparity_max = max(local_max, disparity_max)
    disparity_min = min(local_min, disparity_min)
    local_max = disparity_max
    local_min = disparity_min
    print(disparity_max, disparity_min)
    disparity_grayscale = (disparity-local_min)*(65535.0/(local_max-local_min))
    #disparity_grayscale = (disparity+208)*(65535.0/1000.0)
    # test for jumping colors prevention
    disparity_fixtype = cv2.convertScaleAbs(disparity_grayscale, alpha=(255.0/65535.0))
    disparity_color = cv2.applyColorMap(disparity_fixtype, cv2.COLORMAP_JET)
    cv2.imshow("Image", disparity_color)
    key = cv2.waitKey(1) & 0xFF
    if key == ord("q"):
    quit();
    return disparity_color
    #Calculate the signed curvature of a 2D curve at each point using interpolating splines.
    # -----
    def curvature(x, y=None, error=0.1):
    Parameters
    x,y: numpy.array(dtype=float) shape (n_points, )
    or
    y=None and

```

```

x is a numpy.array(dtype=complex) shape (n_points, )
#In the second case the curve is represented as a np.array of complex numbers.
error: float
#The admissible error when interpolating the splines
Returns
#Detect the obstacle along with coordinate on the roadway whose pixel >= 50
# -----
contours, _ = cv2.findContours(mask, cv2.RETR_TREE, cv2.CHAIN_APPROX_NONE)
for contour in contours:
    area = cv2.contourArea(contour)
    print(area)
    if area > 50:
        #Red Color Spot of Obstacle Detection
        cv2.drawContours(frame, contour, -1, (0, 0, 255), 5)
        x, y, w, h = cv2.boundingRect(contour)
        cv2.rectangle(frame, (x, y), (x + w, y + h), (0, 255, 0), 2)
        center = (x, y)
        print(center)
#Model with 5 CNNs and 4 FCLs.
# -----
def create_nvidia_model1():
    model = Sequential()
    model.add(Convolution2D(24, 5, 5, subsample=(2, 2), border_mode="same", input_shape=(row,
col, ch)))
    model.add(Activation('relu'))
    model.add(Convolution2D(24, 5, 5, subsample=(2, 2), border_mode="same"))
    model.add(Activation('relu'))
    model.add(Convolution2D(48, 5, 5, subsample=(2, 2), border_mode="same"))
    model.add(Activation('relu'))
    model.add(Convolution2D(64, 3, 3, border_mode="same"))
    model.add(Activation('relu'))
    model.add(Convolution2D(64, 3, 3, border_mode="same"))
    model.add(Flatten())
    # Added a Dropout layer to help reduce overfitting.
    model.add(Dropout(0.5))
    model.add(Activation('relu'))
    model.add(Dense(100))
    model.add(Activation('relu'))
    model.add(Dense(50))
    model.add(Activation('relu'))
    model.add(Dense(10))
    model.add(Activation('relu'))
    model.add(Dense(1))
    model.compile(optimizer="adam", loss="mse")
    print('Model is created and compiled..')
    return model
# Creates plots of steering angles by consecutive timestamps
# -----
def plotFeatures(data):
# Plot all the time sections of steering data

```

```

j = 0 # Rotating Color
k = 0 # Rotating Marker
jj = 0 # Subplot Number
timebreak = [0] # Store indices of timebreaks
start = 0
c = ['r','b','g','k','m','y','c']
marker = [',','o','x','+','*','s','d']
for i in range(1,data.shape[0]):
if data[i,0] != data[i-1,0] and data[i,0] != (data[i-1,0] + 1):
timebreak.append(int(data[i-1,0]))
if jj < 70:
jj = jj + 1
print(jj)
plt.subplot(4,1,jj)
plt.plot(data[start:i-1,0],data[start:i-1,1],c=c[j],marker=marker[k])
start = i
j = j + 1
if jj == 69:
plt.subplot(7,10,jj+1)
plt.plot(data[start:-1,0],data[start:-1,1],c=c[j],marker=marker[k])
if j == 6:
j = 0
k = 0 #k = k + 1
if k == 7:
k = 0
for i in range (1,5):
plt.subplot(4,1,i)
plt.xlabel('TimeStamp')
plt.ylabel('Steering Angle')
plt.grid(True)
plt.suptitle('Consecutive Timestamp Steering')
plt.subplots_adjust(left=0.05,bottom=0.05,right=0.95,top=0.95,wspace=0.4,hspace=0.25)
fig = plt.gcf()
fig.set_size_inches(30, 15)
fig.savefig('Steering.png', dpi=200)
# Calculate RMSE
# -----
print "Calculating RMSE"
test = pd.read_csv("submissions/groundtruth_evaluation.csv")
predictions = pd.read_csv("submissions/final_predictions.csv")
t = test['steering_angle']
p = predictions['steering_angle']
length = predictions.shape[0]
print "Predicted angles: " + str(length)
sq = 0
mse = 0
for j in range(length):
sqd = ((p[j] - t[j])**2)
sq = sq + sqd
print(sq)

```

```

mse = sq/length
print(mse)
rmse = np.sqrt(mse)
print("model evaluated RMSE:", rmse)

```

References

1. Tran, N. *Global Status Report on Road Safety*; World Health Organization: Geneva, Switzerland, 2018; pp. 5–11.
2. Jeppsson, H.; Östling, M.; Lubbe, N. Real life safety benefits of increasing brake deceleration in car-to-pedestrian accidents: Simulation of Vacuum Emergency Braking. *Accid. Anal. Prev.* **2018**, *111*, 311–320. [[CrossRef](#)] [[PubMed](#)]
3. Tefft, B.C. The Prevalence of Motor Vehicle Crashes Involving Road Debris, United States, 2011–2014. *Age* **2016**, *20*, 10–1.
4. American Association of State Highway and Transportation Officials (AASHTO). *Highway Safety Manual*; AASHTO: Washington, DC, USA, 2010.
5. Andreopoulos, A.; Tsotsos, J.K. 50 Years of object recognition: Directions forward. *Comput. Vis. Image Underst.* **2013**, *117*, 827–891. [[CrossRef](#)]
6. Poczter, S.L.; Jankovic, L.M. The Google car: Driving toward a better future? *J. Bus. Case Stud.* **2013**, *10*, 7–14. [[CrossRef](#)]
7. Xu, X.; Fan, C.K. Autonomous vehicles, risk perceptions and insurance demand: An individual survey in China. *Transp. Res. Part A Policy Pract.* **2019**, *124*, 549–556. [[CrossRef](#)]
8. Guo, Y.; Xu, H.; Zhang, Y.; Yao, D. Integrated Variable Speed Limits and Lane-Changing Control for Freeway Lane-Drop Bottlenecks. *IEEE Access* **2020**, *8*, 54710–54721. [[CrossRef](#)]
9. Mehmood, A.; Liaquat, M.; Bhatti, A.I.; Rasool, E. Trajectory Planning and Control for Lane-Change of Autonomous Vehicle. In Proceedings of the 2019 5th International Conference on Control, Automation and Robotics (ICCAR), Beijing, China, 19–22 April 2019. [[CrossRef](#)]
10. Ko, W.; Chang, D.E. Cooperative Adaptive Cruise Control Using Turn Signal for Smooth and Safe Cut-In. In Proceedings of the 2018 18th International Conference on Control, Automation and Systems (ICCAS), Yongpyong Resort, Seoul, Korea, 17–20 October 2018; IEEE: Piscataway, NJ, USA, 2018.
11. Katare, D.; El-Sharkawy, M. Embedded System Enabled Vehicle Collision Detection: An ANN Classifier. In Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019. [[CrossRef](#)]
12. Liu, J.; Sun, Q.; Fan, Z.; Jia, Y. TOF Lidar Development in Autonomous Vehicle. In Proceedings of the 2018 IEEE 3rd Optoelectronics Global Conference (OGC), Shenzhen, China, 4–7 September 2018. [[CrossRef](#)]
13. Islam, K.T.; Wijewickrema, S.; Raj, R.G.; O’Leary, S. Street Sign Recognition Using Histogram of Oriented Gradients and Artificial Neural Networks. *J. Imaging* **2019**, *5*, 44. [[CrossRef](#)]
14. Bulumulle, G.; Boloni, L. Reducing Side-Sweep Accidents with Vehicle-to-Vehicle Communication. *J. Sens. Actuator Netw.* **2016**, *5*, 19. [[CrossRef](#)]
15. Lai, Y.-K.; Ho, C.-Y.; Huang, Y.-H.; Huang, C.-W.; Kuo, Y.-X.; Chung, Y.-C. Intelligent Vehicle Collision-Avoidance System with Deep Learning. In Proceedings of the 2018 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Chengdu, China, 26–30 October 2018. [[CrossRef](#)]
16. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*; Touretzky, D.S., Mozer, M.C., Hasselmo, M.E., Eds.; Mit Press: Nevada, NV, USA, 2012.
17. Yang, Z.; Zhang, Y.; Yu, J.; Cai, J.; Luo, J. End-to-end Multi-Modal Multi-Task Vehicle Control for Self-Driving Cars with Visual Perceptions. In Proceedings of the 2018 24th International Conference on Pattern Recognition (ICPR), Beijing, China, 20–24 August 2018.
18. Bojarski, M.; Del Testa, D.; Jacket, L.; Firner, B.; Flepp, B.; Muller, U.; Zieba, K. End to end learning for self-driving cars. *arXiv* **2016**, arXiv:1604.07316.
19. Bhavsar, P.; Das, P.; Paugh, M.; Dey, K.; Chowdhury, M. Risk Analysis of Autonomous Vehicles in Mixed Traffic Streams. *Transp. Res. Rec. J. Transp. Res. Board* **2017**, *2625*, 51–61. [[CrossRef](#)]
20. Geng, L.G.L.; Sun, J.; Xiao, Z.; Zhang, F.; Wu, J. Combining CNN and MRF for road detection. *Comput. Electr. Eng.* **2018**, *70*, 895–903. [[CrossRef](#)]

21. Elfes, A. Using occupancy grids for mobile robot perception and navigation. *Computer* **1989**, *22*, 46–57. [[CrossRef](#)]
22. Li, Y.; Ruichek, Y. Building variable resolution occupancy grid map from stereoscopic system—A quadtree based approach. In Proceedings of the 2013 IEEE Intelligent Vehicles Symposium (IV), Gold Coast, Australia, 23–26 June 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 744–749.
23. Homm, F.; Kaempchen, N.; Ota, J.; Burschka, D. Efficient occupancy grid computation on the GPU with lidar and radar for road boundary detection. In Proceedings of the 2010 IEEE Intelligent Vehicles Symposium (IV), La Jolla, CA, USA, 21–24 June 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 1006–1013.
24. Zang, A.; Chen, X.; Trajcevski, G. High Definition Digital Elevation Model System vision paper. In Proceedings of the 29th International Conference on Scientific and Statistical Database Management, Chicago, IL, USA, 27–29 June 2017; pp. 1–6.
25. Oniga, F.; Nedevschi, S. Processing Dense Stereo Data Using Elevation Maps: Road Surface, Traffic Isle, and Obstacle Detection. *IEEE Trans. Veh. Technol.* **2009**, *59*, 1172–1182. [[CrossRef](#)]
26. Oniga, F.; Nedevschi, S. Polynomial curb detection based on dense stereovision for driving assistance. In Proceedings of the 13th International IEEE Conference on Intelligent Transportation Systems, Funchal, Madeira Island, Portugal, 19–22 September 2010; pp. 1110–1115.
27. Du, X.; Tan, K.K.; Htet, K.K.K. Vision-based lane line detection for autonomous vehicle navigation and guidance. In Proceedings of the 2015 10th Asian Control Conference (ASCC), Sutera Harbour Resort, Sabah, Malaysia, 31 May–3 June 2015.
28. Aufrere, R.; Mertz, C.; Thorpe, C. Multiple sensor fusion for detecting location of curbs, walls, and barriers. In Proceedings of the IEEE IV2003 Intelligent Vehicles Symposium, Columbus, OH, USA, 9–11 June 2003; pp. 126–131.
29. Michalke, T.; Kastner, R.; Fritsch, J.; Goerick, C. A self-adaptive approach for curbstone/roadside detection based on human-like signal processing and multi-sensor fusion. In Proceedings of the 2010 IEEE Intelligent Vehicles Symposium (IV), San Diego, CA, USA, 21–24 June 2010; pp. 307–312.
30. Hata, A.Y.; Osório, F.S.; Wolf, D.F. Robust curb detection and vehicle localization in urban environments. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 1257–1262.
31. Dolson, J.; Baek, J.; Plagemann, C.; Thrun, S. Upsampling range data in dynamic environments. In Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 1141–1148.
32. Jun, T.; Jian, L.; Xiangjing, A.; Hangen, H. Robust Curb Detection with Fusion of 3D-Lidar and Camera Data. *J. Sens.* **2014**, *14*, 9046–9073. [[CrossRef](#)]
33. Rabe, C.; Franke, U.; Gehrig, S. Fast detection of moving objects in complex scenarios. In Proceedings of the 2007 IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, 13–15 June 2007; IEEE: Piscataway, NJ, USA, 2007; pp. 398–403.
34. Lenz, P.; Ziegler, J.; Geiger, A.; Röser, M. Sparse scene flow segmentation for moving object detection in urban environments. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; IEEE: Piscataway, NJ, USA, 2011; pp. 926–932.
35. Broggi, A.; Cattani, S.; Patander, M.; Sabbatelli, M.; Zani, P. A full-3D voxel-based dynamic obstacle detection for urban scenario using stereo vision. In Proceedings of the 16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013), The Hague, The Netherlands, 6–9 October 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 71–76.
36. Hadsell, R.; Sermanet, P.; Ben, J.; Erkan, A.; Scoffier, M.; Kavukcuoglu, K.; Muller, U.; LeCun, Y. Learning long-range vision for autonomous off-road driving. *J. Field Robot.* **2009**, *26*, 120–144. [[CrossRef](#)]
37. Ess, A.; Schindler, K.; Leibe, B.; Van Gool, L. Object Detection and Tracking for Autonomous Navigation in Dynamic Environments. *Int. J. Robot. Res.* **2010**, *29*, 1707–1725. [[CrossRef](#)]
38. Bernini, N.; Bertozzi, M.; Castangia, L.; Patander, M.; Sabbatelli, M. Real-time obstacle detection using stereo vision for autonomous ground vehicles: A survey. In Proceedings of the 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), Qingdao, China, 8–11 October 2014.
39. Pfeiffer, D.; Franke, U. Towards a Global Optimal Multi-Layer Stixel Representation of Dense 3D Data. In Proceedings of the BMVC, Dundee, UK, 29 August–2 September 2011.

40. Manduchi, R.; Castaño, A.; Talukder, A.; Matthies, L. Obstacle Detection and Terrain Classification for Autonomous Off-Road Navigation. *Auton. Robot.* **2005**, *18*, 81–102. [[CrossRef](#)]
41. Broggi, A.; Buzzoni, M.; Felisa, M.; Zani, P. Stereo Obstacle Detection in Challenging Environments: The VIAC Experience, IV. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; IEEE: Piscataway, NJ, USA, 2015.
42. Zhou, J.; Li, B. Robust Ground Plane Detection with Normalized Homography in Monocular Sequences from a Robot Platform. In Proceedings of the 2006 International Conference on Image Processing, Atlanta, GA, USA, 8–11 October 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 3017–3020.
43. Kumar, S.; Karthik, M.S.; Krishna, K.M.; Kumar, S. Markov Random Field based small obstacle discovery over images. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; IEEE: Piscataway, NJ, USA, 2014; pp. 494–500.
44. Luong, Q.-T.; Weber, J.; Koller, D.; Malik, J. An integrated stereo-based approach to automatic vehicle guidance. In Proceedings of the Proceedings of IEEE International Conference on Computer Vision, Boston, MA, USA, 20–23 June 1995.
45. Williamson, T.A. A high-performance Stereo Vision System for Obstacle Detection. Ph.D. Thesis, Robotics Institute Carnegie Mellon University, Pittsburg, PA, USA, September 1998.
46. Hancock, J.A. *High-Speed Obstacle Detection for Automated Highway Applications*; Carnegie Mellon University: Pittsburg, CA, USA, May 1997.
47. Labayrade, R.; Aubert, D.; Tarel, J.-P. Real time obstacle detection in stereovision on non-flat road geometry through "v-disparity" representation. In Proceedings of the Intelligent Vehicle Symposium 2002 IEEE IVS-02, Versailles, France, 18–20 June 2002; IEEE: Piscataway, NJ, USA, 2002.
48. Goldbeck, J.; Huertgen, B. Lane detection and tracking by video sensors. In Proceedings of the 199 IEEE/IEEE/JSAI International Conference on Intelligent Transportation Systems (Cat. No.99TH8383), Tokyo, Japan, 5–8 October 1999.
49. Aufrère, R.; Chapuis, R.; Chausse, F. *A Model-Driven Approach for Real-Time Road Recognition, Machine Vision and Applications 13*; Springer: Aubière Cedex, France, November 2001. [[CrossRef](#)]
50. Aufrère, R.; Chapuis, R.; Chausse, F. A fast and robust vision-based road following algorithm. In Proceedings of the IEEE Intelligent Vehicles Symposium 2000, Dearborn, MI, USA, 3–5 October 2000; IEEE: Piscataway, NJ, USA, 2000; pp. 192–197.
51. Takahashi, A.; Ninomiya, Y. Model-based lane recognition. In Proceedings of the IEEE Intelligent Vehicles Symposium 1996, Tokyo, Japan, 19–20 September 1996; pp. 201–206.
52. Trucco, E.; Verri, A. *Introductory Techniques for 3-D Computer Vision*; Prentice-Hall: Upper Saddle River, NJ, USA, 1998.
53. Udacity Self Driving Car. Available online: <https://github.com/udacity/self-driving-car> (accessed on 20 October 2019).
54. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [[CrossRef](#)]
55. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The Cityscapes Dataset for Semantic Urban Scene Understanding. 2016. In Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, USA, 27–30 June 2016. [[CrossRef](#)]
56. Pinggera, P.; Ramos, S.; Gehrig, S.; Franke, U.; Rother, C.; Mester, R. Lost and Found: Detecting small road hazards for self-driving vehicles. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016.
57. Pinggera, P.; Franke, U.; Mester, R. High-performance long-range obstacle detection using stereo vision. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015.
58. Kanzow, C.; Yamashita, N.; Fukushima, M. Levenberg–Marquardt methods with strong local convergence properties for solving nonlinear equations with convex constraints. *J. Comput. Appl. Math.* **2004**, *172*, 375–397. [[CrossRef](#)]
59. Wedel, A.; Badino, H.; Rabe, C.; Loose, H.; Franke, U.; Cremers, D. B-spline modeling of road surfaces with an application to free space estimation. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 572–583. [[CrossRef](#)]
60. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the KDD, Portland, OR, USA, 4 September 1996.

61. Guttman, A. R-trees: A Dynamic Index Structure for Spatial Searching. In Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data, New York, NY, USA, 18–21 June 1984; pp. 47–57.
62. Leutenegger, S.T.; Lopez, M.A.; Edgington, J. STR: A Simple and Efficient Algorithm for R-tree Packing. In Proceedings of the 13th International Conference on Data Engineering, Birmingham, UK, 7–11 April 1997.
63. Ramos, S.; Gehrig, S.; Pinggera, P.; Franke, U.; Rother, C. Detecting Unexpected Obstacles for Self-driving Cars: Fusing Deep Learning and Geometric Modeling. *arXiv* **2016**, arXiv:1612.06573.
64. Hirschmuller, H. Stereo Processing by Semiglobal Matching and Mutual Information. *IEEE Trans. Pattern Anal. Mach. Intell.* **2007**, *30*, 328–341. [[CrossRef](#)] [[PubMed](#)]
65. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. *arXiv* **2016**, arXiv:1511.07122.
66. Wang, C.; Komodakis, N.; Paragios, N. Markov Random Field modeling, inference & learning in computer vision & image understanding: A survey. *Comput. Vis. Image Underst.* **2013**, *117*, 1610–1627. [[CrossRef](#)]
67. Li, S.Z. *Markov Random Field Modeling in Image Analysis*; Springer Science and Business Media: New York, NY, USA, 2001. [[CrossRef](#)]
68. Savchynskyy, B. Discrete Graphical Models—An Optimization Perspective. *Found. Trends@Comput. Graph. Vis.* **2019**, *11*, 160–429. [[CrossRef](#)]
69. Freno, A.; Trentin, E. *Markov Random Fields*; Springer: Berlin/Heidelberg, Germany, 2011. [[CrossRef](#)]
70. M Pauly, M.; Gross, M.; Kobbelt, L.P. Efficient simplification of point-sampled surfaces. In Proceedings of the IEEE Visualization 2002 VIS, Boston, MA, USA, 27 October–1 November 2002; pp. 163–170.
71. Fernandez, C.; Fernández-Llorca, D.; Stiller, C.; Sotelo, M.Á. Curvature-based curb detection method in urban environments using stereo and laser. In Proceedings of the 2015 IEEE Intelligent Vehicles Symposium (IV), Seoul, Korea, 28 June–1 July 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 579–584.
72. Shabaninia, E.; Naghsh-Nilchi, A.R.; Kasaei, S. High-order Markov random field for single depth image super-resolution. *IET Comput. Vis.* **2017**, *11*, 683–690. [[CrossRef](#)]
73. Boykov, Y.; Kolmogorov, V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 1124–1137. [[CrossRef](#)]
74. Sugiyama, Y.; Matsui, Y.; Toyoda, H.; Mukozaka, N.; Ichori, A.; Abe, T.; Takabe, M.; Mizuno, S. A 3.2kHz, 13-bit Optical Absolute Rotary Encoder with a CMOS Profile Sensor. *IEEE Sens. J.* **2008**, *8*, 1430–1436. [[CrossRef](#)]
75. Kemal Alkin Gunbay, Mert Arıkan, Mehmet Turkan. Autonomous Cars: Vision based Steering Wheel Angle Estimation. *arXiv* **2019**, arXiv:1901.10747.
76. Tian, Y.; Pei, K.; Jana, S.; Ray, B. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In Proceedings of the 40th International Conference on Software Engineering, Gothenburg, Sweden, 27 May–3 June 2018. [[CrossRef](#)]
77. Houenou, A.; Bonnifait, P.; Cherfaoui, V.; Yao, W. Vehicle Trajectory Prediction Based on Motion Model and Maneuver Recognition. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3–7 November 2013; pp. 4363–4369. [[CrossRef](#)]
78. Suarez-Alvarez, M.M.; Pham, D.T.; Prostov, M.Y.; Prostov, Y.I. Statistical approach to normalization of feature vectors and clustering of mixed datasets. *Proc. R. Soc. A Math. Phys. Eng. Sci.* **2012**, *468*, 2630–2651. [[CrossRef](#)]
79. About Feature Scaling and Normalization. Available online: https://sebastianraschka.com/Articles/2014_about_feature_scaling.html (accessed on 16 November 2018).

