VISION-BASED NAVIGATION OF AUTONOMOUS VEHICLE IN ROADWAY ENVIRONMENTS WITH UNEXPECTED HAZARDS

Mhafuzul Islam*
Ph.D. Student
Clemson University
Glenn Department of Civil Engineering
351 Fluor Daniel Engineering Innovation Building, Clemson, SC 29634
Tel: (864) 986-5446, Fax: (864) 656-2670
Email: mdmhafi@clemson.edu

Mashrur Chowdhury, Ph.D., P.E., F. ASCE
Eugene Douglas Mays Endowed Professor of Transportation
Professor of Automotive Engineering, Civil Engineering, and Computer Science
Glenn Department of Civil Engineering, Clemson University
216 Lowry Hall, Clemson, South Carolina 29634
Tel: (864) 656-3313, Fax: (864) 656-2670
E-mail: mac@clemson.edu

Hongda Li
Ph.D. Student
Division of Computer Science
School of Computing, Clemson University
220 McAdams Hall, Clemson, South Carolina 29634
Tel: (864) 986-2018
E-mail: hongdal@clemson.edu

Hongxin Hu, Ph.D.
Assistant Professor
Division of Computer Science
School of Computing, Clemson University
217 McAdams Hall, Clemson, South Carolina 29634
Tel: (864) 656-2847
E-mail: hongxih@clemson.edu

*Corresponding author
Abstract: 209 words + Text: 3673 + References: 1094 + 1 table: 250 = 5226 words
Submission date: November 15, 2018

Submitted to Transportation Research Board 2019
ABSTRACT
Vision-based navigation of modern autonomous vehicles primarily depends on Deep Neural Network (DNN) based systems in which the controller obtains input from sensors/detectors such as cameras, and produces an output such as a steering wheel angle to navigate the vehicle safely in roadway traffic. Typically, these DNN-based systems are trained through supervised and/or transfer learning; however, recent studies show that these systems can be compromised by perturbation or adversarial input features on the trained DNN-based models. Similarly, this perturbation can be introduced into the autonomous vehicle DNN-based system by unexpected roadway hazards such as debris and roadblocks. In this study, we first introduce a roadway hazardous environment (both intentional and unintentional) that can compromise the DNN-based system of an autonomous vehicle, producing an incorrect vehicle navigational output, such as a steering wheel angle, which can cause crashes resulting in fatality and injury. Then, we develop an approach based on object detection and semantic segmentation to mitigate the adverse effect of this hazardous environment, one that helps the autonomous vehicle to navigate safely around such hazards. This study finds the DNN-based model with hazardous object detection and semantic segmentation improves the ability of an autonomous vehicle to avoid potential crashes by 21% compared to the traditional DNN-based autonomous driving system.

*Keywords:* Roadway Hazard, Autonomous Vehicle, Deep Neural Network, Driving Model
INTRODUCTION
According to the 2016 American Automobile Association report (1), 50,658 crashes occurred in the U. S. each year between 2011 and 2014 due to road hazards, resulting in 9,805 injuries and 125 deaths. These hazards, for example, debris, which are considered to be non-fixed and unexpected objects on the travel or driving lane of the roadway and include objects that have fallen from vehicles or have come from construction sites or littering. Given that the autonomous vehicle is considered the future of transportation, its ability to detect small debris or hazards and then navigate safely around them is crucial for avoiding potential crashes. Currently, this navigation task is accomplished through computer vision using Deep Neural Networks (DNN) where an autonomous vehicle perceives objects from sensors, and the software running in the vehicle determines the action to be taken based on their inputs. Several types of sensors, for example, cameras, LIDAR, and Radar are currently available for this perception task, and their cost-effectiveness makes vision-based navigation attractive for autonomous vehicles (2)(3)(4).

The recent development of DNN, in particular, Convolutional Neural Networks (CNN)(5), has improved vision-based navigation for autonomous vehicles significantly. After being trained and tested using a dataset collected by sensors, these CNN models are then deployed in autonomous vehicles to maneuver the vehicle safely. For example, during training, the end-to-end driving model maps a relationship between the driving behavior of humans using roadway images collected from three cameras and the steering wheel angle (6)(7). Thus, the performance of autonomous vehicles primarily depends on the training dataset, meaning if a hazard that the CNN model is not trained on appears on the roadway, the autonomous vehicle driving model may produce an incorrect steering angle and cause a crash. This study addresses the situation where a well-trained driving model may fail due to unexpected hazards that may lead to a crash, and then explores the use of object detection and semantic segmentation (8) for mitigating the navigation problem in this hazardous condition.

The remainder of the paper is organized as follows. Section 2 explores the existing studies on autonomous vehicle navigation, state-of-art autonomous vehicle driving models, and the limitations of the DNNs that have been used for autonomous driving. Section 3 introduces the method developed in this research for navigating an autonomous vehicle on a roadway with unexpected hazards. We validate our proposed method using three case studies, (1) a model trained using a dataset that includes hazards but without considering them as separate input features; (2) a model trained on a dataset that considers hazards as separate input features using image segmentation and distance measurement sensors; (3) a model trained on a dataset that considers hazards as a separate input features using only image segmentation. In the second and third case studies, we introduce a DNN-based approach to enhance the ability of an autonomous vehicle to navigate safely in a hazardous environment. Section 4 presents the experimental setup employed in this study. Section 5 evaluates all these cases, reporting the results obtained from the experiment, and finally, Section 6 discusses the conclusions and suggests areas for future work.
RELATED WORK

This section reviews the previous research on hazard detection, DNN-based driving models used in an autonomous vehicle, and the techniques for and the importance of object detection and image segmentation in addition to the limitations of using DNN in autonomous vehicles.

DNN-based Autonomous Vehicle Driving Model

DNN-based autonomous vehicle systems are rapidly evolving (7)(9). Not only are software companies such as Waymo (Google) Uber, and Lyft using the DNN-based system for autonomous vehicles but many car companies such as Tesla, Volvo, BMW, and Ford are currently working on DNN-based autonomous driving systems. In such systems, sensors like cameras, LIDAR, and Radar provide input to DNN models, such as Convolutional Neural Networks (CNN)(5) or Recurrent Neural Networks (RNN) (10), which then produce outputs such as steering wheel angle and velocity. For example, Defense Advanced Research Projects Agency (DARPA) Autonomous Vehicle (DAVE-2), an autonomous vehicle architecture developed by NVIDIA, uses a CNN model which takes input from a camera and outputs a steering wheel commands for navigation (7), while Udacity autonomous vehicle driving architectures include both CNN-based (e.g., Autumn) and RNN-based (e.g., Chauffeur using CNN and LSTM) (11). This study used a CNN-based driving model similar to DAVE-2 as it is the fundamental mode of DNN-based autonomous vehicle systems.

DNN-based autonomous driving systems, which are intrinsically software systems, can be error-prone and cause severe consequences if they do not function as intended. Several studies have shown the vulnerabilities of DNN models(12)(13)(14)(15). For example, DNN-based image classification can be exploited by adding a small perturbation to an input image such that the DNN model misclassifies it as another category, a vulnerability recently confirmed by (16), which found that attackers can physically modify objects using a low-cost technique to cause classification errors in DNN-based vision systems. These perturbations can be introduced under widely varying distances, angles, and resolutions. For example, in (16) perturbations caused a DNN to interpret a subtly modified physical stop sign as a speed limit of 45 mph sign. Similarly, the debris or roadblocks on the road can also compromise the autonomous vehicle driving model by producing incorrect steering wheel angles, potentially causing a fatal collision. These limitations prompted this study to evaluate the impact of unexpected hazardous environments on a DNN-based driving model.

Autonomous Vehicle Dataset

Data are an important part of deep learning-based systems, and this study required a dataset that supports 1) end-to-end driving systems (input: image; output: steering wheel angle), 2) Image segmentation, and 3) small hazard detection. To find an appropriate one, we explored various existing datasets used by the autonomous vehicle community. The closest dataset provided by Udacity supports end-to-end and image segmentation, but it does not provide the ground truth for small hazards in the drivable lane (11). KTTI (17) and Cityscape (18) datasets also do not support
small hazard detection as ground truth data. The dataset matching our requirements the closest is the Lost and Found dataset (19), which contains the image as the input, and the yaw rate (angular velocity), not the steering wheel angle required by this study, as an output. Since it did not fully meet our needs, after careful consideration we created our dataset using simulation as described in the experimental setup section.

**DNN-based Object Detection and Segmentation**

Object detection and classification are core components of autonomous driving. By detecting and classifying the objects, the autonomous vehicle controller determines a safe maneuver for both path and route planning. If an autonomous vehicle is not able to detect small unexpected hazards on the road, it will not be able to navigate safely, perhaps resulting in a crash. However, detecting these small objects or hazards is a challenging task. While various sensors such as Radar and LIDAR can be used for accurate distance and velocity measurement, these sensors are relatively costly and exhibit low resolution (19). Considering these limitations, vision-based sensors such as cameras are being used on autonomous vehicles for navigation. With the recent development of DNN models, DNN-based object detection and semantic segmentation can be applied to detect these roadway hazards, making navigation of autonomous vehicles safer.

Semantic segmentation, a technology that has been widely used in the computer vision area to divide an unknown image into different parts (20), can be applied to an image containing unknown objects. This technology is effective in providing the scenario depicted by an image, allowing the DNN models to capture additional information about the dataset during training. There are three major types of semantic segmentation technologies: region-based semantic segmentation (21)(22), Fully Convolutional Network (FCN)-based semantic segmentation (23)(24)(25) and weakly supervised semantic segmentation (26)(27)(28). The region-based semantic segmentation provides segmentation based on the results of object detection, meaning it can be developed on any CNN model. The FCN-based semantic segmentation functions at a pixel-to-pixel level, meaning it does not require extracting regions of the image and, thus, can be applied to arbitrary sizes of images. The weakly supervised semantic segmentation technology, which was developed to reduce the labeling cost of a large dataset (28), achieves semantic segmentation by exploiting annotated bounding boxes or image-level labels. While recent studies show that segmentation-based navigation can improve navigational performance (29)(30)(31), none considers the navigation of autonomous vehicles in hazardous environments. Thus, by leveraging these DNN-based models, we can detect hazards and then extract their semantic information from images obtained from the camera sensor of an autonomous vehicle. The approach adopted here used an FCN-based model as this network is relatively small yet yields fast results (23). To the best of our knowledge, this is the first work that develops a DNN-based approach for autonomous vehicle driving focusing on unexpected hazardous environments.
METHOD
In this section, we describe our approach for developing a strategy for safer navigation of autonomous vehicles in a hazardous environment. This study used an object detection and segmentation approach to augment the data subsequently used by the autonomous vehicle model to predict the steering wheel angle. This system is based on three models: 1) the DNN-based hazard detection and segmentation model, which detects the hazard, then segments it as a new output image; 2) the hazard analysis and avoidance model, which masks/fuses the data from this new output image with the original input image to make the autonomous vehicle driving model aware of the unexpected roadway hazard. This model then analyzes the hazard, determining if it should be ignored or considered a threat factor (\(T_f\)); 3) the DNN-based autonomous vehicle driving model, which takes the image fused with hazard information and produces the steering wheel angle required to navigate the vehicle safely in this unexpected hazardous environment. Figure 1 below shows the developed driving model framework. We provide the detail description of these three models in the following subsections (as shown in Figure 1).

**FIGURE 1** Framework of an autonomous vehicle driving in an unexpected hazardous environment.

**DNN-based hazard detection and segmentation model**
For hazard detection and image segmentation, this study used a Fully Convolution Network (FCN), which is a DNN-based image object detection and segmentation model (23). Figure 2 shows the structure of the FCN network used in our study. It takes an input of image size 400x600x3 and outputs a segmented image of same size. We used a pre-trained network with a weight of VGGNet (32), a Deep Convolutional Networks for Large-Scale Image Recognition, and then we trained the model with our training dataset to classify the hazard and perform image segmentation.
Hazard analysis and avoidance model
As can be seen in Figure 1, the image captured from the center dashboard camera first goes to the hazard detection and segmentation model, which provides an output of the detected object in addition to a segmented image. Then, this output is combined with the original image in the hazard analysis and avoidance model. In this study, we develop a hazard analysis and avoidance model based on the following equation:

\[ I = (1 - T_f) \times I_{original} + T_f \times I_{segmented} \]

where \( I \) is the image used to predict the autonomous vehicle driving model, \( I_{original} \) is the data from the center camera of the vehicle, \( I_{segmented} \) is the segmented image of \( I_{original} \) containing the hazardous object detected and its segmentation, and \( T_f \) is the threat value of the detected hazardous object or physical-world threat object. This threat value depends on the objects detected: If the object is dangerous to the autonomous vehicle, it will have a high threat factor, while a negligible threat object will have a lower threat factor. Thus, this threat value depends on the longitudinal and latitudinal distance from the autonomous vehicle. To formulate this dependency, we have used two procedures in our experiment: (1) Procedure 1 - threat value determination using Radar data; and (2) Procedure 2 – threat value determination using image segmentation.

**Procedure 1 - threat value determination using Radar data**
According to the first procedure, we measure the longitudinal distance \( l_x \), and latitudinal distance \( l_y \) of hazardous objects from the vehicle using a distance measurement sensor, such as Radar. If the vehicle is moving towards (longitudinal movement) or steering towards (latitudinal movement) the hazard the value of \( l_x \) and \( l_y \) decreases respectively, and hence the hazard poses higher threat of colliding with the vehicle. We consider the hazard as a threat to the vehicle if the hazard is within the \( l_{x,\text{max}} \) longitudinal distance, and \( l_{y,\text{max}} \) latitudinal distance. In our study, we use the Radar data to measure distance, and using the following equations, we measure the threat value.
\[ T = \sqrt{\left(\frac{l_{x,max} - l_x}{l_{x,max}}\right)^2 + \left(\frac{l_{y,max} - l_y}{l_{y,max}}\right)^2} \]

\[ T_f = \begin{cases} 
T - T_{min} & \text{if } l_y \leq l_{y,max} \text{ and } l_x \leq l_{x,max} \\
\frac{T_{max} - T_{min}}{l_{y,max} - l_{y,min}} & \text{if } l_y > l_{y,max} \text{ or } l_x > l_{x,max} 
\end{cases} \]

where, \( T_f \) is the threat value corresponding to the hazardous object, \( l_x \) and \( l_y \) are the longitudinal distance and latitudinal distance of the detected hazard from the vehicle, respectively. \( l_{x,max} \) and \( l_{y,max} \) is the maximum longitudinal distance and maximum latitudinal distance, correspondingly, to consider the hazard as a threat, and \( T \) is the threat value calculated from longitudinal and latitudinal threat effect. The value of \( T \) is normalized using Min-Max normalization technique to obtain a value between 0 to +1 to obtain the final threat value, \( T_f \). (33)(34). In our experiment, we have selected \( l_{x,max} \) is 6000cm as this is the Radar maximum range of finding an object, and \( l_{y,max} \) is 370cm which is the standard lane width of a roadway. We can visualize the relationship between the threat value, and longitudinal and latitudinal distance in Figure 3.

![FIGURE 3 Heatmap for the threat value based on the longitudinal and latitudinal distance of a hazardous object using Radar sensor data.](image)

**Procedure 2 – threat value determination using image segmentation**

In this procedure, we use the segmented image to calculate the threat value instead of using a Radar sensor. In this way, we can eliminate the use of any sensor data beside camera video feed. After the image segmentation, we get the image coordinates \((x, y)\) of the hazard. As the camera is located at the center of the vehicle, we measure the relative distance of hazardous object in the image of size \((h, w)\), from the bottom center pixel, \(\left(\frac{h}{2}, \frac{w}{2}\right)\) to quantify the threat. We calculated the threat based on the location of the hazard in the segmented image using the following equation:
\[ T_f = 1 - \sqrt{\frac{(x - \frac{h}{2})^2 + (y - \frac{w}{2})^2}{\frac{h^2}{2} + \frac{w^2}{2}}} \]

where, \( T_f \) is the threat value corresponding to the hazardous object located in the segmented image at location \((x, y)\) pixels, where \((x, y)\) is the pixel value closest to the bottom center pixel, \((\frac{h}{2}, \frac{w}{2})\), of the image. The value of \( h \) and \( w \) indicates the height and width of the image. As the camera of the vehicle is located at the center of the vehicle we subtract the \( x \) and \( y \) values from \( \frac{h}{2} \) and \( \frac{w}{2} \) respectively to transform the threat value from the bottom center. We can visualize the threat value in Figure 4 where the threat value decreases as the object moves from the center bottom of the image.

![Heatmap for the threat value based on the location of hazard using pixel value from the segmented image.](image)

FIGURE 4 Heatmap for the threat value based on the location of hazard using pixel value from the segmented image.

**DNN-based autonomous vehicle driving model**

In our study, we implemented a driving model similar to DAVE-2, an end-to-end driving model (7). As shown in Figure 5, the network receives an input image of 200x600x3 pixels and produces a steering wheel angle as an output. This network includes one normalization layer, five convolution layers (Conv2D), and four fully connected (FC) layers. We have used a 5x5 kernel (i.e., filters) and 2x2 stride (i.e., increment of kernel movement) in the first 3 Conv2D layers, and a 1x1 stride and a 3x3 kernel in the last two Conv2D layers. The entire network contains a total of 7,970,619 trainable parameters.
FIGURE 5 CNN-based end-to-end autonomous vehicle driving model used in this study.

We train our developed driving model for autonomous vehicle with hazard detection and image segmentation, and hazard analysis and avoidance model followed by the deployment to test the performance. Based on the approach developed in this study, the trained driving model is aware of hazardous objects on the roadway and produces a steering wheel angle to avoid the potential crash.

EXPERIMENTAL SETUP

In the experimental setup, we describe the data collection method, preparation, and augmentation; and finally we train and validate the DNN-based autonomous vehicle driving model. The steps of our experiment setup are as follows:

Data Collection
For this study, we used the robotics simulation platform Webots (35) to create the roadway environment with hazardous objects and to collect the data including the driving attributes of the camera image, timestamp, location, vehicle speed, and steering wheel angle. The following subsections describe the collection procedure of the dataset.

Roadway Environment Setup
The roadway built in the simulation consisted of two lanes in each direction and 1663 m of roadway with 16 curves (having 45 to 90-degree radius of curvature) and 2 intersections as shown in Figure 6(a). An additional six non-autonomous vehicles were placed randomly on this roadway. The hazardous debris, which included five objects: rocks, wooden boxes, an oil barrels, wooden pallets, and sections of pipe, were created in Webots (35) and placed randomly on the roadway as shown in Figure 6(b).
FIGURE 6 Roadway environment setup for an autonomous vehicle with hazardous objects.

**Autonomous Vehicle Setup**

For collecting the data, the autonomous vehicle was equipped with three dashboard cameras, a front, left and right camera, as seen in Figure 7, and a Radar sensor. The data collected using these cameras were used to train the end-to-end autonomous vehicle driving model. For example, as seen in Figure 8, the images collected by the left and right camera differ from the center camera. After training, the autonomous vehicle used only a single front camera to navigate through the roadway similar to the DAVE-2 system (7). In our developed driving model, we have used a Radar sensor data to calculate the threat value. We have used a Delphi ESR Radar sensor, which is used commercially used in current autonomous vehicles (36). We have used the medium range mode configurations (horizontal field of view of 90 degrees and a maximum range of 6000 cm) of the Radar sensor in our autonomous vehicle (37). We have also equipped the vehicle with three other Radar sensors in three directions (left, right and back side) for monitoring the near-by traffic condition and vehicles. These Radars sensors were also configured in medium range mode.

FIGURE 7 Camera placements in the autonomous vehicle (left, center, and right cameras).
FIGURE 8 Example of images collected by the three cameras in the autonomous vehicle (left, center, and right camera images (from left to right))

Data Preparation
After collecting the data, we prepared the image dataset for training the end-to-end driving model by normalizing and resizing. As shown in Figure 9, the steering wheel angle output, was normalized between the values of -0.5 and +0.5, where positive value indicate the steering to the right and negative value represents steering to the left using linear transformation following this equation:

\[
\theta_{\text{normalized}} = -0.5 + \max(0, \min(1.0, \frac{\theta_{\text{raw}} - \theta_{\text{min}}}{\theta_{\text{max}} - \theta_{\text{min}}}))
\]

Where, \(\theta_{\text{normalized}}\) is the normalized steering angle between 0 and 1, \(\theta_{\text{raw}}\) is the actual steering wheel angle (in degree radians) measured from the vehicle, \(\theta_{\text{max}}\), and \(\theta_{\text{min}}\) are the maximum and minimum steering wheel angle, respectively. We also normalized the input images for training, which is necessary to improve the DNN model performance (38). The Red, Green, and Blue (RGB) channel values of the input images were normalized between the values of -1.0 and +1.0, and their top 200 pixels was cropped using a Lambda layer (as shown Figure 10) as top portion of the image is not necessary to predict the steering wheel angle, and doing so does not impact the steering wheel angle output of the driving model. For all data collected, we used an online image annotation tool, LabelMe (39), for labeling the hazardous object and image segmentation. Using this tool, we have created the ground truth for training the image segmentation model on detecting and segmenting the hazards in the image dataset.
**Data Augmentation**
To obtain satisfactory performance from the driving model, it is necessary to train the model on multiple training datasets. Using the techniques of data augmentation, we created additional data from the existing data through affine transformation (40), specifically random rotation, random brightness change, and horizontal flipping of the images, to double the size of the dataset as shown in Table 1. In our original dataset, we have collected 1390 images, and we doubled it using data augmentation (as presented in Table 1). Among these 2780 images, 2222 images were used for training where 468 images contained hazards. The remaining 558 images were used for validation, and 104 images were used from second simulation for evaluating or testing the driving model performance.
### TABLE 1 Dataset description

| Dataset                                      | Collected Dataset Size | Dataset Size After Data Augmentation |
|----------------------------------------------|------------------------|--------------------------------------|
| Total dataset size                          | 1390                   | 2780                                 |
| Total training dataset size                  | 1111                   | 2222                                 |
| Total hazard dataset size in the training dataset | 234                    | 468                                  |
| Total validation dataset size                | 279                    | 558                                  |
| Total testing dataset size (all containing hazard) | 52                     | 104                                  |

#### Model Training and Validation

After the development of the end-to-end driving model, we trained it using the augmented dataset. This dataset was divided into two, 80% in a training set (2222 images as per Table 1) and the remaining 20% in a validation set (558 images as per Table 1). Then we augmented the data from the three cameras to create another dataset of clear roadway images and images with roadway hazards. We then trained two models:

- **Case 1**: A model trained on a dataset that includes hazards but without considering them as a separate input feature.
- **Case 2**: A model trained on a dataset including hazards considered as separate input features using distance measurement sensors and segmentation (Procedure 1 - threat value determination using Radar data).
- **Case 3**: A model trained on a dataset including hazards considered as separate input features using only image segmentation (Procedure 2 – threat value determination using image segmentation).

For the training of DNN models, we have used Adam optimizer that can change the learning rate dynamically. The mean square error based loss function, droprate of 0.5 in the last four FC layers, and L2 regularization were used to reduce overfitting and under-fitting and to minimize training error. We used model checkpoints to stop the training when the validation loss is not decreasing over the time. Figure 11(a) shows the performance of the model training for Case 1 where the training was stopped after 14 epochs because the model did not exhibit much improvement after 11 epochs. We observed no overfitting or under-fitting during the training. In Case 2, the model stopped training after 16 epochs (as shown in Figure 11(b)), and in Case 3, the model stopped the training around 15 epochs (as shown in Figure 11(c))
FIGURE 11 Training and validation of end-to-end driving model using the training and validation dataset for each case scenario.

ANALYSIS RESULTS

After training and validating the model, we evaluated it using the test dataset of 104 images presented in Table 1. We created this dataset from a second simulation where all debris were in the middle of the road, measuring the predicted steering wheel angle for each image. In this second simulation, we created the ground truth by manually driving the vehicle on the roadway. Then we deployed the trained DNN model for Case 1, Case 2, and Case 3. We then analyzed the performance of the model for each cases using quantitative measures (i.e., root mean square error (RMSE) and the mean absolute error (MAE)) and qualitatively (i.e., visualization).

Quantitative Results of Model Performance

The quantitative results included the RMSE and the MAE, measured by comparing the predicted steering wheel angle with the actual steering wheel angle/ground truth. We defined the RMSE and MAE as follows:

(a) Case 1
(b) Case 2
(c) Case 3
\[ RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (G_i - P_i)^2} \]

\[ MAE = \frac{1}{N} \sum_{i=1}^{N} |G_i - P_i| \]

where \( N \) is the total number of frames in the testing dataset and \( G_i \) and \( P_i \) are the ground truth and predicted steering wheel angle, respectively, for the \( i^{th} \) frame of the testing dataset. As shown in Figure 12, both the RMSE and MAE are higher for Case 1 than Case 2 and Case 3. A lower RMSE and MAE indicate that the predicted steering wheel output is close to the actual steering wheel angle, meaning the navigation of the autonomous vehicle reduced the probability of a crash, i.e., indicating that it is the safer option between two cases.

![Error measurement on the testing dataset](image)

**FIGURE 12 Error measurement on the testing dataset.**

We measured the prediction accuracy and improvement of Case 2 and Case 3, over Case 1. By comparing \( RMSE \) of Case 2 and Case 3 with \( RMSE_{case1} \), we calculate the prediction improvement based on the equation below:

\[
Percentage of improvement = \left( \frac{|RMSE - RMSE_{case1}|}{RMSE_{case1}} \right) \times 100 \%
\]

Based on our study, we found a 21.25% improvement in the prediction accuracy of Case 2 over Case 1, and 18.25% improvement in the prediction accuracy of Case 3 over Case 1 suggesting that both Case 2 and Case 3 improved the safety of the navigation by avoiding the hazards on the roadway.

**Qualitative Results for Driving Direction**

Figure 13 shows the qualitative results from the study conducted here. To obtain the qualitative measurement, we transformed the steering wheel angle (-0.5 to +0.5) into a driving direction angle.
In Webots, the steering wheel angle follows the Ackermann geometry, representing a linear relationship between steering wheel angle vs. driving direction \((35)(44)\). The prediction accuracy can be presented qualitatively by observing the driving direction angle or angle of movement of the autonomous vehicle. For example, Figure 13 shows that the continuous steering wheel output of data from the time step of 62000 milliseconds (ms) to 72000ms window for ground truth, Case 1, Case 2, and Case 3. A hazard is present in the roadway, and the driving model is producing the output for maneuvering the autonomous vehicle. According to Figure 13, the autonomous vehicle is moving towards the left for each case. In Case 1, at time step 66000ms, the predicted driving direction is +5.2 degrees toward the right, causing the car to move closer to the hazard, represented here as a box. However, in Case 2 and Case 3, the predicted driving direction is +11.7 degrees and +9.28 degrees respectively, which is a value closer to the ground truth than in Case 1. Overall, the qualitative results indicate better accuracy prediction for Case 2 and Case 3 than for Case 1.

![Figure 13 Qualitative results of ground truth, Case 1, Case 2, and Case 3 of driving direction.](image-url)
Quantitative Results for Driving Direction
Following the Frenet coordinate system, we have performed quantitative analysis of hazard avoidance. In a Frenet coordinate system, the longitudinal movement, and latitudinal movement are represented in x-axis and y-axis, respectively (45). In our analysis, we have plotted the time step in the x-axis and latitudinal movement in the y-axis (Figure 14). We analyzed the trajectory of the autonomous vehicle and calculated the RMSE for each case compared to the ground truth. For example, we have taken a segment from time step 64000ms to 72000ms to present the hazard avoidance result from our experiment. Both qualitative and quantitative results are presented in Figure 13 and Figure 14, respectively. In Figure 14, in Case 2, the model is producing the output, which is closer to the ground truth, compared to Case 1. In both Case 1 and Case 2, the vehicle was able to avoid the hazard (Figure 14), but in the Case 1, the vehicle maneuvered closer to the hazard. In Case 1, the RMSE value was 0.52, whereas in the Case 2 and Case 3, the vehicle’s trajectory is far from the hazard, having an RMSE value of 0.07 and 0.23, respectively. As in Case 2, and Case 3, the trajectory was closer to the ground truth, we conclude that the maneuver occurred in Case 2 and Case 2 safer from Case 1. We performed a statistical significance test (t-test) between the ground truth, and Case 1, Case 2, and Case 3 at a 95% confidence interval. We found that Case 1 is significantly different from the ground truth at a 95% confidence interval. However, Case 2, which uses Radar, and Case 3, which only uses the segmented image, are not significantly different from the ground truth. Thus, we can achieve the same level of performance without using any additional object detection sensor, such as Radar, and using image segmentation.

FIGURE 14 Trajectory of the autonomous vehicle for ground truth, Case 1, Case 2, and Case 3.
CONCLUSIONS
Detecting unexpected hazards on a roadway is a crucial task for the safe operation of an autonomous vehicle. In this work, we developed and evaluated a DNN-based driving system for autonomous vehicles for unexpected hazardous roadway conditions. First, we detected the hazard, and then using semantic segmentation, we extracted the hazard information and performed data fusion to improve the safety of the navigation of an autonomous vehicle. This study makes the following contributions to the current body of research: 1) we evaluated the effect of the hazardous roadway environment on the DNN of an autonomous vehicle; 2) we developed a DNN-based driving model for autonomous driving that can address an unexpected hazardous roadway condition and navigate the autonomous vehicle safely through this environment. More specifically, we explored the field of object detection and semantic segmentation based deep learning models to address this dangerous navigation problem; 3) we contributed a new dataset that can be used by the autonomous vehicle community to improve the driving model in unexpected hazardous roadway environments. Based on this dataset, our method improved the safety of the autonomous vehicle by 21% in terms of avoiding hazards, compared to autonomous vehicles without hazard detection and segmentation as input features in their vision-based navigation models. Future work will include fusing the temporal and spatial information into the DNN-based model, potentially further improving the safety of autonomous vehicles operating in an unexpected hazardous roadway environment.

ACKNOWLEDGMENTS
This paper is based on research supported by the USDOT Center for Connected Multimodal Mobility (Tier 1 University Transportation Center) Grant. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the USDOT Center for Connected Multimodal Mobility (C2M2). The U.S. Government assumes no liability for the contents or use thereof. The authors would like to thank Aniqa Chowdhury for editing the paper.

AUTHOR CONTRIBUTION STATEMENT
The authors confirm contribution to the paper as follows: study conception and design: M. Islam, M. Chowdhury, H. Li, and H. Hu; Data collection: M. Islam; Analysis and interpretation of results: M. Islam, M. Chowdhury, H. Li; draft manuscript preparation: M. Islam, M. Chowdhury, H. Li, and H. Hu. All authors reviewed the results and approved the final version of the manuscript.

REFERENCES
1. American Automobile Association. The Prevalence of Motor Vehicle Crashes Involving Road Debris, United States, 2011 – 2014. Biochemistry, Vol. 31, No. 24, 1992, pp. 5507–5514. https://doi.org/10.1021/bi00139a012.
2. Bertozzi, M., A. Broggi, and A. Fascioli. Vision-Based Intelligent Vehicles: State of the Art and Perspectives. Robotics and Autonomous Systems, Vol. 32, No. 1, 2000, pp. 1–16. https://doi.org/10.1016/S0921-8890(99)00125-6.
3. Dagan, E., O. Mano, G. P. Stein, and A. Shashua. Forward Collision Warning with a Single
4. Tatarek, T., J. Kronenberger, and U. Handmann. Functionality, Advantages and Limits of the Tesla Autopilot.

5. Krizhevsky, A., I. Sutskever, and G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, 2012, pp. 1–9. https://doi.org/http://dx.doi.org/10.1016/j.procyt.2014.09.007.

6. Yang, Z., Y. Zhang, J. Yu, J. Cai, and J. Luo. End-to-End Multi-Modal Multi-Task Vehicle Control for Self-Driving Cars with Visual Perception. 2018.

7. Bojarski, M., D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba. End to End Learning for Self-Driving Cars. 2016, pp. 1–9.

8. Yao, J., S. Fidler, and R. Urtasun. Describing the Scene as a Whole: Joint Object Detection, Scene Classification and Semantic Segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012, pp. 702–709. https://doi.org/10.1109/CVPR.2012.6247739.

9. Pomerleau, D. A. Alvinn: An Autonomous Land Vehicle in a Neural Network. *Advances in Neural Information Processing Systems 1*, 1989, pp. 305–313.

10. Beaufays, F., H. Sak, and A. Senior. Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling Has. *Interspeech*, No. September, 2014, pp. 338–342. https://doi.org/arXiv:1402.1128.

11. Udacity Self Driving Car. https://github.com/udacity/self-driving-car.

12. Carlini, N., and D. Wagner. Towards Evaluating the Robustness of Neural Networks. *Proceedings - IEEE Symposium on Security and Privacy*, 2017, pp. 39–57. https://doi.org/10.1109/SP.2017.49.

13. Papernot, N., P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks. *Proceedings - 2016 IEEE Symposium on Security and Privacy, SP 2016*, 2016, pp. 582–597. https://doi.org/10.1109/SP.2016.41.

14. Athalye, A., N. Carlini, and D. Wagner. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. 2018.

15. Papernot, N., P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The Limitations of Deep Learning in Adversarial Settings. *Proceedings - 2016 IEEE European Symposium on Security and Privacy, EURO S and P 2016*, 2016, pp. 372–387. https://doi.org/10.1109/EuroSP.2016.36.

16. Eykholt, K., I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song. Robust Physical-World Attacks on Deep Learning Models. 2017. https://doi.org/10.1109/CVPR.2018.00175.

17. Geiger, A., P. Lenz, C. Stiller, and R. Urtasun. Vision Meets Robotics: The KITTI Dataset. *International Journal of Robotics Research*, Vol. 32, No. 11, 2013, pp. 1231–1237. https://doi.org/10.1177/0278364913491297.

18. Cordts, M., M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, B. Benenson, U. Franke, S. Roth, and B. Schiele. The Cityscapes Dataset for Semantic Urban Scene Understanding. 2016. https://doi.org/10.1109/CVPR.2016.350.

19. Pinggera, P., S. Ramos, S. Gehrig, U. Franke, C. Rother, and R. Mester. Lost and Found: Detecting Small Road Hazards for Self-Driving Vehicles. *IEEE International Conference*
20. Guo, Y., Y. Liu, T. Georgiou, and M. S. Lew. A Review of Semantic Segmentation Using Deep Neural Networks. *International Journal of Multimedia Information Retrieval*, Vol. 7, No. 2, 2018, pp. 87–93. https://doi.org/10.1007/s13735-017-0141-z.

21. Caesar, H., J. Uijlings, and V. Ferrari. Region-Based Semantic Segmentation with End-to-End Training. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 9905 LNCS, 2016, pp. 381–397. https://doi.org/10.1007/978-3-319-46448-0_23.

22. Girshick, R., J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587. https://doi.org/10.1109/CVPR.2014.81.

23. Long, J., E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 07–12 June, 2015, pp. 3431–3440. https://doi.org/10.1109/CVPR.2015.7298965.

24. Eigen, D., and R. Fergus. Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture. 2015. https://doi.org/10.1109/ICCV.2015.304.

25. Liu, Y., Y. Guo, and M. S. Lew. On the Exploration of Convolutional Fusion Networks for Visual Recognition. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Vol. 10132 LNCS, No. 1, 2017, pp. 277–289. https://doi.org/10.1007/978-3-319-51811-4_23.

26. Dai, J., K. He, and J. Sun. BoxSup: Exploiting Bounding Boxes to Supervise Convolutional Networks for Semantic Segmentation. *Proceedings of the IEEE International Conference on Computer Vision*, Vol. 2015 Inter, 2015, pp. 1635–1643. https://doi.org/10.1109/ICCV.2015.191.

27. Papandreou, G., L.-C. Chen, K. P. Murphy, and A. L. Yuille. Weakly-and Semi-Supervised Learning of a Deep Convolutional Network for Semantic Image Segmentation. *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1742–1750. https://doi.org/10.1109/ICCV.2015.203.

28. Khoreva, A., R. Benenson, J. Hosang, M. Hein, and B. Schiele. Simple Does It: Weakly Supervised Instance and Semantic Segmentation. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Vol. 2017–Janua, 2017, pp. 1665–1674. https://doi.org/10.1109/CVPR.2017.181.

29. Eraqi, H. M., M. N. Moustafa, and J. Honer. End-to-End Deep Learning for Steering Autonomous Vehicles Considering Temporal Dependencies. No. Nips, 2017, pp. 1–8.

30. Teichmann, M., M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun. MultiNet: Real-Time Joint Semantic Reasoning for Autonomous Driving. 2016. https://doi.org/10.1109/CVPR.2012.6248074.

31. Siam, M., S. Elkerdawy, M. Jagersand, and S. Yogamani. Deep Semantic Segmentation for Automated Driving: Taxonomy, Roadmap and Challenges. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, Vol. 2018–March, 2018, pp. 1–8. https://doi.org/10.1109/ITSC.2017.8317714.

32. Simonyan, K., and A. Zisserman. Very Deep Convolutional Networks for Large-Scale
33. Suarez-Alvarez, M. M., D. T. Pham, M. Y. Prostov, and Y. I. Prostov. Statistical Approach to Normalization of Feature Vectors and Clustering of Mixed Datasets. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, Vol. 468, No. 2145, 2012, pp. 2630–2651. https://doi.org/10.1098/rspa.2011.0704.

34. About Feature Scaling and Normalization. https://sebastianraschka.com/Articles/2014_about_feature_scaling.html. Accessed Nov. 16, 2018.

35. Cyberbotics Ltd. Webots: Robot Simulator. http://www.cyberbotics.com. Accessed Nov. 15, 2018.

36. AutonomouStuff. Delphi Electronically Scanning Radar. 1–2. https://autonomoustuff.com/product/delphi-esr-2-5-24v/. Accessed Nov. 14, 2018.

37. Webots Documentation: Radar Sensors. https://cyberbotics.com/doc/guide/radar-sensors#delphi-esr. Accessed Nov. 14, 2018.

38. Zha, S., F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov. Exploiting Image-Trained CNN Architectures for Unconstrained Video Classification. 2015. https://doi.org/10.5244/C.29.60.

39. Russell, B. C., A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: A Database and Web-Based Tool for Image Annotation. *International Journal of Computer Vision*, Vol. 77, No. 1–3, 2008, pp. 157–173. https://doi.org/10.1007/s11263-007-0090-8.

40. Tian, Y., K. Pei, S. Jana, and B. Ray. DeepTest: Automated Testing of Deep-Neural-Network-Driven Autonomous Cars. 2017. https://doi.org/10.1145/3180155.3180220.

41. Konur, O. ADAM: A METHOD FOR STOCHASTIC OPTIMIZATION. *ICLR*, Vol. 5, No. 2, 2015, pp. 365–380. https://doi.org/10.1063/1.4902458.

42. Baldi, P., and P. Sadowski. Understanding Dropout. *Nips*, No. 1, 2013, pp. 1–9. https://doi.org/10.1007/978-3-642-33269-2_28.

43. Keras. https://keras.io/callbacks/.

44. Mitchell, W. C., A. Staniforth, and I. Scott. Analysis of Ackermann Steering Geometry. *SAE Technical Papers U6*, No. 724, 2006. https://doi.org/10.4271/2006-01-3638.

45. Houenou, A., P. Bonnifait, V. Cherfaoui, and W. Yao. Vehicle Trajectory Prediction Based on Motion Model and Maneuver Recognition. *IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 4363–4369. https://doi.org/10.1109/IROS.2013.6696982.