Investigate the Effects of V2X Technologies for Automated Vehicles Using Virtual Simulation and Driving Simulator Experiments

- *Exploring the Effects of Visual Environment on Traffic Safety*

SAFETY RESEARCH USING SIMULATION

UNIVERSITY TRANSPORTATION CENTER

Yina Wu, PhD, PI
Research Associate Professor
Department of Civil, Environmental and Construction Engineering
University of Central Florida

Mohamed Abdel-Aty, PhD, Co-PI
Pegasus Professor, Chair
Department of Civil, Environmental and Construction Engineering
University of Central Florida

Qing Cai, PhD
Research Assistant Professor
Department of Civil, Environmental and Construction Engineering
University of Central Florida

Investigate the Effects of V2X Technologies for Automated Vehicles Using Virtual Simulation and Driving Simulator Experiments—Exploring the Effects of Visual Environment on Traffic Safety

Yina Wu, PhD, PI
Research Associate Professor
Department of Civil, Environmental and Construction Engineering
University of Central Florida
https://orcid.org/0000-0001-6516-8144

Qing Cai, PhD
Research Assistant Professor
Department of Civil, Environmental and Construction Engineering
University of Central Florida
https://orcid.org/0000-0003-0822-2268

Mohamed Abdel-Aty, PhD, PE, Co-PI
Pegasus Professor, Chair
Department of Civil, Environmental and Construction Engineering
University of Central Florida
https://orcid.org/0000-0002-4838-1573

A Report on Research Sponsored by

SAFER-SIM University Transportation Center

Federal Grant No: 69A3551747131

August 2021

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. UCF-1-Y4	**2. Government Accession No.**	**3. Recipient's Catalog No.**
4. Title and Subtitle Investigate the Effects of V2X Technologies for Automated Vehicles Using Virtual Simulation and Driving Simulator Experiments—Exploring the Effects of Visual Environment on Traffic Safety		**5. Report Date** August 1, 2021 **6. Performing Organization Code** Enter any/all unique numbers assigned to the performing organization, if applicable.
7. Author(s) Yina Wu, PhD, PI https://orcid.org/0000-0001-6516-8144 Mohamed Abdel-Aty, Ph.D. https://orcid.org/0000-0002-4838-1573 Qing Cai, PhD, https://orcid.org/0000-0003-0822-2268		**8. Performing Organization Report No.** Enter any/all unique alphanumeric report numbers assigned by the performing organization, if applicable.
9. Performing Organization Name and Address University of Central Florida Research & Commercialization 12201 Research Pkwy Ste 501 Orlando, FL 32826 USA		**10. Work Unit No.** **11. Contract or Grant No.** Safety Research Using Simulation (SAFER-SIM) University Transportation Center (Federal Grant #: 69A3551747131)
12. Sponsoring Agency Name and Address Safety Research Using Simulation University Transportation Center Office of the Secretary of Transportation (OST) U.S. Department of Transportation (US DOT)		**13. Type of Report and Period Covered** Final Research Report (August 2020 – August 2021) **14. Sponsoring Agency Code**

15. Supplementary Notes

This project was funded by Safety Research Using Simulation (SAFER-SIM) University Transportation Center, a grant from the U.S. Department of Transportation – Office of the Assistant Secretary for Research and Technology, University Transportation Centers Program.

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by a grant from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. government assumes no liability for the contents or use thereof.

16. Abstract

The visual environment could have effects on the performance of automated vehicles within the V2X technology regarding traffic safety. This research aims to explore the effects of visual environment on traffic safety for the development of virtual simulation and driving simulator experiments. Both the effects on the speeding crashes and the severity of single-vehicle crashes were explored. To obtain the data of drivers' visual environment in the real world, a framework was proposed to obtain the Google street view (GSV) images. Deep neural network and computer vision technologies were applied to obtain the clustering and depth information from the GSV images. To reflect drivers' visual environment in the real world, the coordinate transformation was conducted, and several visual measures were proposed and calculated. Three different tree-based ensemble models (i.e., random forest, adaptive boosting (AdaBoost), and eXtreme Gradient Boosting (XGBoost)) were applied to estimate the number of speeding crashes and the comparison results showed that XGBoost could provide the best data fit. The explainable machine learning method were applied to explore the effects of drivers' visual environment and other features on speeding crashes. The results validated the visual environment data obtained by the proposed method for the speeding crash analysis. It was suggested that the proportion of trees in the drivers' view and the proportion of road length with trees could reduce speeding crashes. In addition, the complexity level of drivers' visual environment was found to increase the crash occurrence.

17. Key Words		18. Distribution Statement	
Highways; Pedestrians and Bicyclists; Safety and Human Factors; Vehicles and Equipment;		No restrictions. This document is available through the SAFER-SIM website, as well as the National Transportation Library	
19. Security Classif. (of this report)	**20. Security Classif. (of this page)**	**21. No. of Pages**	**22. Price**
Unclassified	Unclassified	68	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Table of Contents

List of Figures

List of Tables

Abstract

The visual environment could have effects on the performance of automated vehicles within the V2X technology regarding traffic safety. This research aims to explore the effects of visual environment on traffic safety for the development of virtual simulation and driving simulator experiments. Both the effects on the speeding crashes and the severity of single-vehicle crashes were explored. To obtain the data of drivers' visual environment in the real world, a framework was proposed to obtain the Google street view (GSV) images. Deep neural network and computer vision technologies were applied to obtain the clustering and depth information from the GSV images. To reflect drivers' visual environment in the real world, the coordinate transformation was conducted, and several visual measures were proposed and calculated. Three different tree-based ensemble models (i.e., random forest, adaptive boosting (AdaBoost), and eXtreme Gradient Boosting (XGBoost)) were applied to estimate the number of speeding crashes and the comparison results showed that XGBoost could provide the best data fit. The explainable machine learning method were applied to explore the effects of drivers' visual environment and other features on speeding crashes. The results validated the visual environment data obtained by the proposed method for the speeding crash analysis. It was suggested that the proportion of trees in the drivers' view and the proportion of road length with trees could reduce speeding crashes. In addition, the complexity level of drivers' visual environment was found to increase the crash occurrence.

Besides, crash severity prediction enables traffic operators to expect the severity for reported crashes with unknown severity conditions and allocate first response resources, which would be beneficial for reducing the outcome of severe crashes. This research uses both Google Street View (GSV) image data and crash information data to predict the severity of single-vehicle crashes. Based on crash direction and location, four GSV images of front, left, right, and behind sides, for each crash location, were collected to illustrate the surrounding environment. The computer vision techniques were applied to get the high-dimensional

segmentation information from the GSV images. The four images are augmented into a two-layer 2D data matrix. To use both GSV image and crash information data, the study proposed a concatenated deep learning model by combining a convolutional neural network (CNN) and a deep neural network (DNN) model for the prediction. The data matrixes from GSV images and crash data were utilized simultaneously in the concatenated deep learning model. Three years' single-vehicle crashes (i.e., 2017-2019) were utilized in this study to evaluate the proposed method. The results indicate that integrating high-dimensional and low-dimensional data with the proposed framework could improve prediction performance. Meanwhile, gradient-weighted class activation mapping (Grad-CAM) was employed to explain the deep learning results. The locations in images with the highest contributions to the prediction results could be determined, which would be helpful to identify and interpret contributing factors for severe crashes.

This study provided new insights to obtain detailed information from GSV images for traffic safety analysis. The information from the GSV images could reflect visual environment on roads and could be used for the development of the virtual simulation and driving simulator experiments. Besides, the identified effects at the pixel level could help further explore the effects of visual environment on the performance of automated vehicles with the V2X technology.

1 Introduction

Speeding is one of the major factors in traffic safety. According to the National Highway Traffic Safety Administration (NHSTA), nearly a third of fatal crashes in the United States have been designated as “speeding-related” in the last decade [1]. On urban arterials, the speed limit violation could significantly increase the severity levels of pedestrian and bicycle crashes [2]. A lot of studies have been conducted to examine the contributing factors for the crash occurrence and speeding behavior. The factors include traffic volume, roadway geometric design, land use, socio-demographic characteristics, and weather, etc. For example, Cai, Abdel-Aty [3] developed grouped random parameter models to examine the crash occurrence on segments and intersections considering the roadway attributes and the zonal level effects. Afghari, Haque [4] categorized the speeding behavior into three levels by proportions based on the speed camera data. It was found that high speed limits are highly associated with moderate speed limit violations, compared to minor or major speed limit violations Besides, the study also revealed that a divided median and higher functional class could lead to more major speed limit violations. Besides, the topic of single-vehicle crashes is one of the essential areas of transportation researchers, as single-vehicle crashes tend to be more severe than other types of crashes. In 2018, single-vehicle crashes accounted for 28.7% of total crashes but accounted for 56.8% of fatal crashes in the US [5]. The statistics illustrate the substantial research needs to have a better understanding of single-vehicle crashes and the practice needs to deploy appropriate countermeasures or traffic management strategies to prevent severe single-vehicle crashes. Moreover, previous research indicates that Emergency Medical Services (EMS) could have a significant impact on traffic injury severity [6]. Thus, predicting crash severity could be beneficial for allocating first response resources and minimizing the delays to rescue the victims of the potential severe crashes.

Recently, several studies have focused on the effects of the driving environment on drivers' behavior and safety. For example, Edquist, Rudin-Brown [7] investigated the effects of road environment visual complexity on travel speed and reaction time by conducting a driving simulator study. It suggested that the visual complexity of the roadside environment is an important contributor to driver workload and performance. Based on a survey study, Atombo, Wu [8] revealed the significant effects of the driving environment on speeding and overtaking violations. Marshall, Coppola [9] developed statistical models to study the effects of trees on crash frequency in the urban area. The study indicated that tree density could reduce the crashes. However, to the best of the authors' knowledge, the study about the drivers' visual environment on traffic safety is limited. One possible reason is that it is difficult to obtain the data from drivers' view.

Recently, within the great development of deep learning and computer vision technology, detailed information including object clusters and depth could be obtained from images. In the era of transportation study, computer vision has been applied to count traffic volume and detect traffic speed [10]. Besides, some studies applied detection and tracking algorithms to get vehicles' trajectory and calculate the surrogate safety measures [11, 12]. In these studies, researchers needed to use cameras to collect the video and image first. It might be time-consuming to collect data in a large study area. Recently, some studies conducted to assess street-level urban greenery using Google Street View (GSV) [13, 14]. Through a Google API, users could specify the location, heading, and vertical angle when downloading the image. Hence, it is possible to get a lot of images with drivers' views through GSV images. Recently, Li, Cai [15] proposed a method for predicting and mapping the occurrence of sun glare using GSV images. Google Street View (GSV) data have wide data coverage and low collection cost, which is one of the new data sources that could be utilized to better understand the road environment. In general, there are three types of GSV-related traffic research and practice. The first type of

research and practice aims to obtain data for a certain type of object. For example, some studies focused on detecting, classifying, and mapping traffic signs from GSV images using computer vision techniques [16-19]. The second type of research introduced new variables based on the GSV images, such as the percentage of grass or trees in the images or the studied areas [20]. In 2016, Mooney et al. assessed the environmental contributions for pedestrian injury using GSV data [21]. In this study, different variables, such as sidewalk, road, and crosswalk conditions, were extracted from GSV to represent environmental conditions. Recent advances in deep learning methods, including CNN, provided an opportunity to obtain more information from GSV data by using the image-based data matrix as model inputs. In 2020, Tanprasert et al. collected four GSV images for each spot (i.e., 0, 90, 180, 270 degrees), and developed a fully connected neural network to identify crash-prone locations [22]. Another studies that was conducted by Li et al. developed a CNN model based on GSV images to predict and map the occurrence of sun glare [15]. However, only GSV data are utilized in the deep learning models in this type of studies. Other data, such as traffic data (e.g., Average Annual Daily Traffic (AADT)), roadway data (e.g., speed limit), crash data (e.g., age, gender), were not included in the datasets for modeling. Moreover, in 2021, Bustos et al. conducted research using GSV data to investigate pedestrian safety conditions [23]. In this study, the researchers utilized gradient-weighted class activation mapping to interpret the model results and identify dangerous areas, which addressed one of the common concerns of model explanation for deep learning models.

Many previous studies have been performed to investigate the crash frequency and severity of crashes, including single-vehicle crashes, while statistical models are the primary method used in crash severity analysis. Binary logit, binary probit, bivariate probit, multinomial logit, ordered logit, random parameter logit models are the predominant statistical models for crash severity analysis [24-29]. In 2019, Hou et al. utilized a mixed logit model to analyze single-

vehicle crashes on freeways. The results illustrated that concrete barriers could increase crash severity for senior drivers [30]. Besides the statistical methods, some research employed machine learning methods for crash severity analysis and found that machine learning models have better model performance when compared with statistical models [31-33]. In 2014, Zheng and Huang developed a neural network (NN) model for predicting crash severity and found that the NN model outperforms the ordered logit model [34]. In 2021, Yan et al. developed single-vehicle crash severity prediction models using multiple machine learning techniques and compared feature importance among different models [35]. In recent years, a significant number of studies have been conducted that utilized deep learning methods for traffic safety research, especially for crash risk prediction. However, compared with the impressive progress in crash prediction, limited studies have been conducted that apply deep learning methods for crash severity prediction. Among different deep learning models, Convolutional Neural Networks (CNN) is one of the most prevalent methods, which requires data matrix format as inputs. In 2021, Rahim and Hassan proposed a method that transforms variables into images and conducts crash severity prediction using the CNN-based deep learning method to predict crash severity. Precision and recall are employed in this study to evaluate the model performance. The results indicate that the proposed method could improve the performance of crash severity prediction [36]. Although data matrixes from the generated images is utilized in the deep learning models based on the collected variables, limited information of the roadway environment where the crashes happened is included for prediction. Thus, including high-resolution/dimensional data, such as images from the crash scenarios, could further improve modeling performance and help to identify contributing factors for severe crashes [37].

This study attempts to propose a novel method to obtain drivers' visual environment from GSV images and explore the effects of the visual environment on speeding crashes and the severity of single crashes. To this end, deep learning models were applied to obtain the cluster

and depth information from GSV images. To explore the visual environment on the speed crashes, several indexes were proposed to quantify the visual environment. Then, the effects of the visual environment on speeding crashes were explored by developing both machine learning and statistical models. Besides, the study contributes to exploring the effects of visual environment on the severity of single crashes from three perspectives: 1) applying high-dimensional low-cost data from GSV image for single-vehicle crash severity prediction; 2) proposing a concatenated deep learning model that utilizes both high-dimension GSV data and low-dimension data sources to predict crash severity; and 3) identifying hotspots and interpreting potential contributing factors in road environment using gradient-weighted class activation mapping (GRAD-CAM).

2 Exploring the Effects of Drivers' Visual Environment on Speeding Crashes

2.1 Methodology

The machine learning models based on neural networks and trees are two most popular models in use today [38, 39]. The neural-network-based deep learning models are more appropriate in fields like image recognition, speed recognition, and natural language processing [11, 40, 41]. On the other hand, the tree-based models could have a good balance of accuracy and interpretability, which has made the tree-based models the most popular non-linear models. Hence, this study took the advantage of both approaches by utilizing neural network models to process images to obtain drivers' visual environment and applying tree-based models to analyze crashes in an interpretable approach.

2.1.1. Machine Learning to Process GSV Images

(1) *GSV image collection*

The GSV panorama is a 360° surrounding image generated from the eight original images captured by multiple cameras by stitching together in sequences. The GSV image could be requested in an HTTP URL form using the GSV image API provided by the Google company[1]. Users can request a static GSV image in customized direction and angle for the locations where GSV is available. Figure 2-1 shows the GSV image requested by the above URL. In this example, the output size of GSV image and latitude and longitude of the location was specified. Besides, the heading indicates the compass heading of the camera which ranges from 0 to 360, pitch specifies the up or down angle of the camera relative to the data collection vehicle, and fov is the horizontal field of view for the image. Previous studies suggested that the horizontal field view is between $50^{o} and\ 60^{o}$[42]. Li et al. [13] used fov of $60^{o}$to collect GSVs, which was

[1] *https://maps.googleapis.com/maps/api/streetview?size=640x400&location= 28.78291, -81.2729& fov=60&heading=0&pitch=0*

adopted in the current study. To get images similar to the drivers' view, the heading was determined based on the road direction and the pitch of 0 was selected.

The current study was conducted on urban arterials. For each segment, one image is collected every 10 meters since images are recorded by the Google cameras every 5-20 meters along roads. In the above URL example, users need to register on Google Maps Platform and purchase to get a valid API key. In this study, a Python script was developed to download the GSV images by automatically using the coordinates.

Figure 2-1 ***An example of GSV collected by the URL***

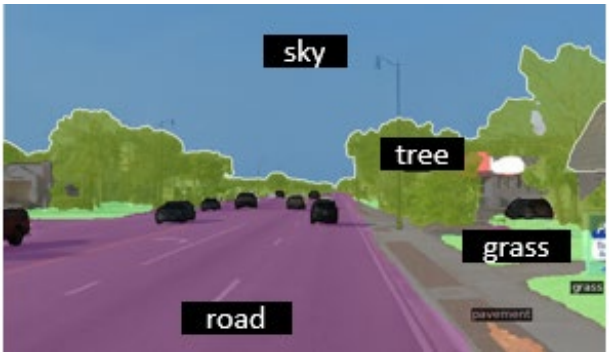
(2) Drivers' visual environment extraction from GSV

Studies about information extraction from images have been growing in the field of computer science. Deep learning has been heavily applied and developed for semantic segmentation from images. In this study, "Detectron2" from Facebook was used to cluster objects. Detectron2, starting with maskrcnn-benchmark [43], is Facebook AI research next generation software system that implements state-of-the-art object detection algorithms by reaching to 34.9 mask average precision [11]. It is also suggested that the model using the Detectron2 framework could reach the state-of-the-art performance for labeling objects in drivers' view [44, 45]. For

example, Syed et al. [44] found that the Detectron2 framework could have a pixel accuracy of around 90% to detect pedestrians in different cloth and offer more stable detection results compared to other detection frameworks with impacts of the pixel area, occlusion rate, and distance. Yu et al. [45] developed models to classify risky driving scenes based on the Detecton2 framework, which could reach 96.4% classification accuracy. As shown in Figure 2-2(a), different objects in the environment such as roads, trees, sky, and buildings in the drivers' view could be labelled from the images. Based on the clustering results, we could know the object type by each pixel in the image. Then, the proportion of pixels by object type in the drivers' view could be calculated, such as the proportion of trees and the proportion of roads. Besides, a measure was suggested to reflect the visual complexity level of drivers' visual environment. The complexity level could be calculated as:

$$\text{complexity level} = \frac{-\sum_k (p_k(\ln(p_k)))}{lnN} \quad (1.1)$$

where k is the category of object, p is the proportion of category k points, N is the number of object categories. Noteworthy, the complexity level has been widely used in previous studies to reflect the land use mix level [46].



(a) Illustration of semantic segmentation results

(b) Illustration of depth information

Figure 2-2 Visualization of image processing

Meanwhile, the depth information could be obtained from the 2D images. Since the GSV image could be treated as a mono camera, a self-supervised monocular depth estimation

method (monodepth2) proposed by Godard et al. [47] was used to obtain the depth information. It was suggested that the depth estimation method could provide an absolute relative error of 0.115 for monocular depth estimation on the KITTI benchmark, achieving state-of-the-art depth estimation. The detection range by this method is from 0 to 80 meters. Figure 2-2(b) illustrates the depth information subtracted from the image in black and white colors.

Through the object clustering and depth estimation, the object types and depth information could be obtained by each pixel (u,v) of the 2D image. In the real world, 3D points could reflect the location (X, Y, Z) of each object. As shown in Figure 2-3, the projection of points in the world coordinate system to the image pixel coordinate including three steps: (1) project points from the world coordinate system to the camera coordinate system; (2) project points from the camera coordinate system to the image coordinate system; (3) project points from the image coordinate system to the pixel coordinate system. The principle axis and principal point (P) connect the camera coordinate system and image coordinate system. To be specific, the principal axis is the line from the camera center perpendicular to the image plane and the principal point is the point where principal axis intersects the image plane. The principle axis is parallel to the road direction and the road surface since the headings of images were the same as the road direction and the pitches of all images were 0. The first projection is related to the extrinsic parameters of cameras including rotation and translation. This projection could be written as: $O_{camera} = [\mathrm{R}|\mathrm{t}] * O_{world}$. $[\mathrm{R}|\mathrm{t}]$ is a 4×4 matrix $\begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}$, and a number 1 is added to the world coordinate (X, Y, Z) to compute the above equation. As the principle axis is parallel to the road direction and the road surface, the world coordinate system and the camera coordinate system could be the same. Then, (X, Y, Z) reflects the location information of a point by assuming that the camera center is the origin. It should be noted that the world and

camera coordinate systems will not be the same if the heading of an image is not the same as the road direction or the pitch is not 0.

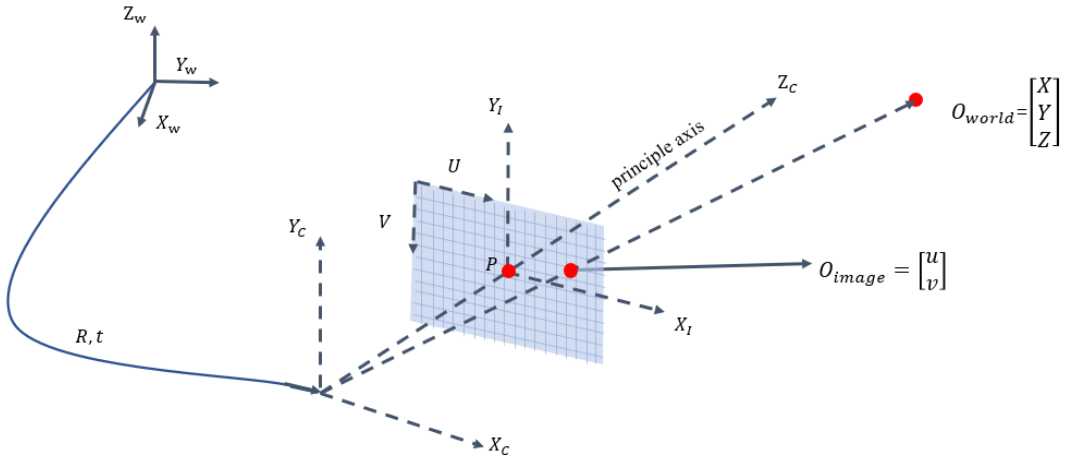
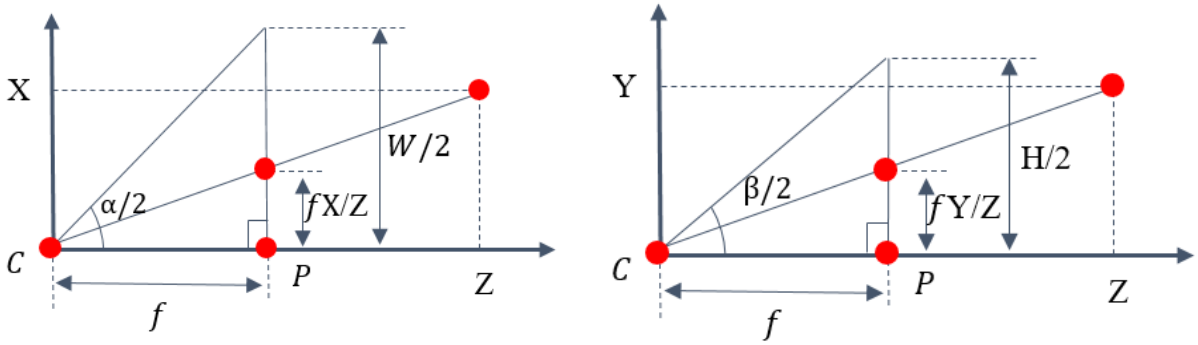


Figure 2-3 Point projection from the world coordinate system to the pixel coordinate system

As shown in Figure 2-4, the Pinhole camera model could be applied to project points from the camera coordinate system to the image coordinate system. The projection on the x and y axis is related to the relation between the focal length of camera f and the depth of point Z. The projection could be expressed as: $O_{image} = KO_{camera}$. K is a 3×3 matrix $\begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix}$, which reflects the intrinsic parameters of the camera. The f could be calculated based on the trigonometry, which is:

$$f = (W/2)/\tan(\alpha/2) \qquad (2\text{-}2)$$

where α is the horizontal field of view and W is horizontal number of pixels of the image. In Figure 2-4(b), the same focal length is applied for the vertical field of view (β) and vertical number of pixels (H) of the image.



(a) Projection on the x axis (b) Projection on the y axis

Figure 2-4 Point projection from the camera coordinate system to the image coordinate system

Finally, the point could be projected to the pixel coordinate system. As shown in Figure 2-3, the origin in the pixel coordinate system is at the top left corner. Hence, the projection should consider the offset of the principle point (u_p, v_P). Then, the K matrix becomes as $\begin{bmatrix} f & 0 & u_p \\ 0 & f & v_p \\ 0 & 0 & 1 \end{bmatrix}$ and the projection from the world coordinate system to the pixel coordinate system could be expressed as:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & u_0 \\ 0 & f & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \qquad (2\text{-}3)$$

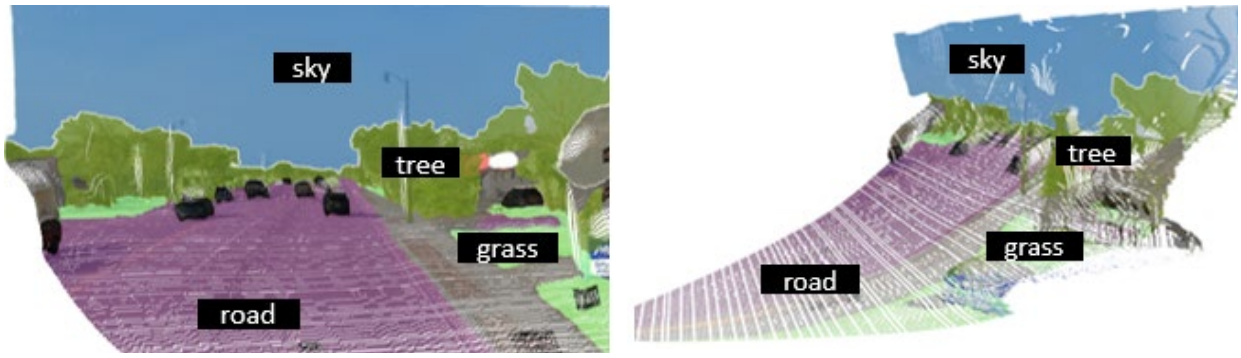
Hence, the (X, Y) in the world coordinate system could be calculated by:

$$X = (u - u_0) * depth[v, u]/f \qquad (2\text{-}4)$$

$$Y = -(v - v_0) * depth[v, u]/f \qquad (2\text{-}5)$$

$$Z = depth[v, u] \qquad (2\text{-}6)$$

Then, by using an inverse projection process, figures with depth information could be transformed into a 3D point cloud. As shown in Figures 5(a) and 5(b), we could know the exact X, Y, Z locations in the real world for specific objects.



(a) 3D point cloud data from driver's view (b) 3D point cloud data from roadside view

Figure 2-5 Illustration of 3D point cloud

As X reflects the horizontal distance and Z reflects the vertical distance, the 3D points could also be projected to the satellite image based on X and Z data. Figure 6 illustrates an example of the satellite image view for the road part based on the semantic segmentation and 3D projection information. It is shown that the transformed data could be along the road in general, which validates the projection method in this study. Due to the large triangulation errors, points far away from the camera are sparser and tend to be wider (yellow lines in Figure 2-6(d)), which is consistent with the previous study [48, 49]. As highlighted in Areas 1 and 2 in Figure 2-5(b) and 2-5(d), the part of the road close to the camera could be cut in the image. Also, the objects on roads such as cars (highlighted in Area 3) and mislabeled objects such as the grass area (highlighted in Areas 4) could block some parts of roads and affect the projection results. In this study, the road width was used to validate the projection accuracy by using 100 images from different segments (Figure 7). The accuracy is calculated by:

$$accuracy = mean(100 * \frac{|w_{obs} - w_{est}|}{w_{obs}}) \qquad (2\text{-}7)$$

where w_{obs} and w_{est} are the observed and estimated road width. It shows that high accuracy (over 90%) could be obtained at the distance from 25 meters to 50 meters. Low accuracy is obtained at the location close to the camera since the view could be cut at the bottom of the image. Meanwhile, the accuracy of locations far away from the camera is also low due to the

sparse effect of cloud points. Hence, the data of the 30-meter distance from 25 meters to 50 meters are used for the following safety analysis. Since the GSV images are requested by 10 meters, the average value of three images is calculated and used.

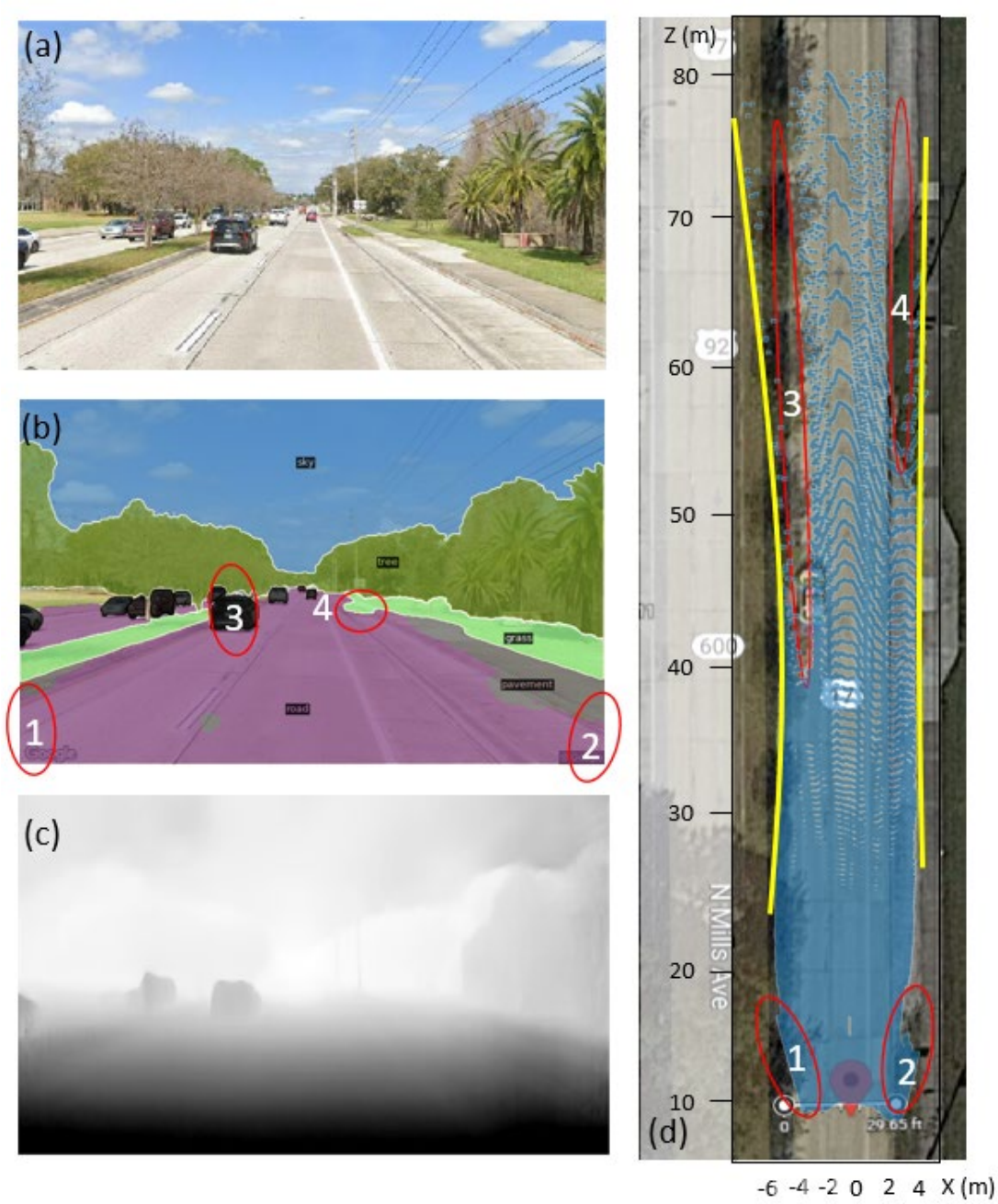


Figure 2-6 Illustration of satellite image view (a: original Google street view; b: sematic segmentation; c: depth estimation; d: projection of satellite image view)

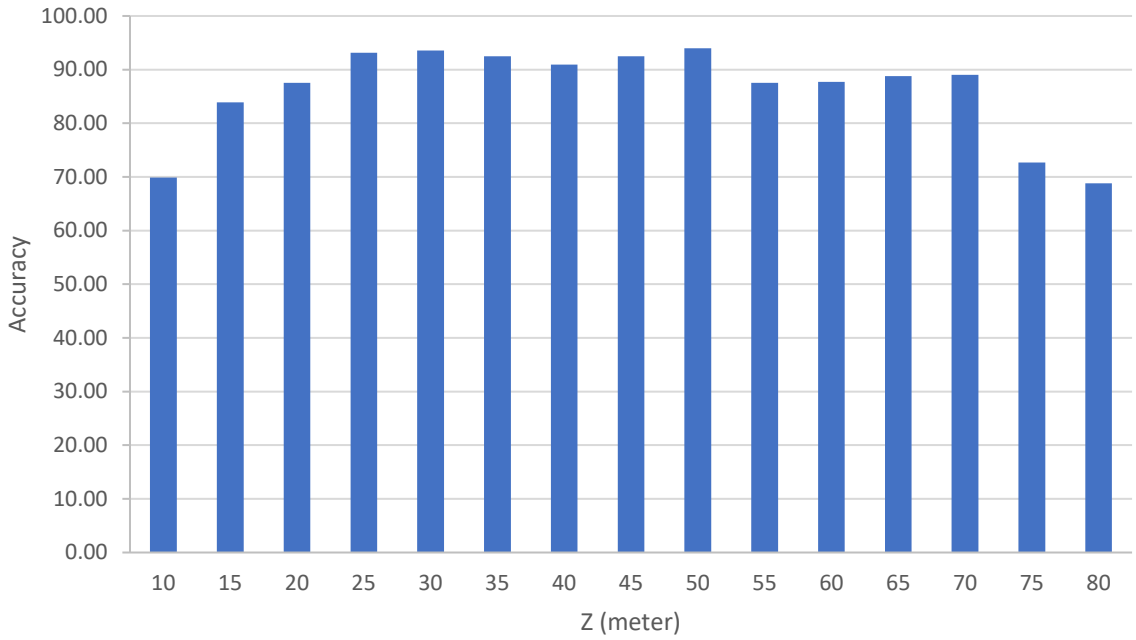


Figure 2-7 Projection accuracy based on road width

Based on the X and segmentation information, the distance of trees away from the edge of roads could be obtained. The trees within 10 meters away from the roads will be used to calculate the proportion of road length with trees. It should be noted that two variables related to trees are calculated, which are the proportion of trees in the drivers' view and the proportion of road length with trees. As shown in Figure 2-8, the proportion of tress in drivers' view reflects the canopy of trees at a certain location while the proportion of road length with trees indicates how many trees the drivers could see along the road.

Figure 2-8 Illustration of tree canopy and road length with trees

Figure 2-9 shows the flowchart of processing GSV images to get the measures related to drivers' visual environment. Four steps were involved which are preparing base map, requesting GSV images, processing images, and calculating measures. At the first step, the direction and coordinates are collected and used as the input parameters to request GSV images. Different computer vision techniques are applied to get the clustering and location information for each pixel in an image. Finally, different measures related to the drivers' visual environment are calculated based on the information at each pixel.

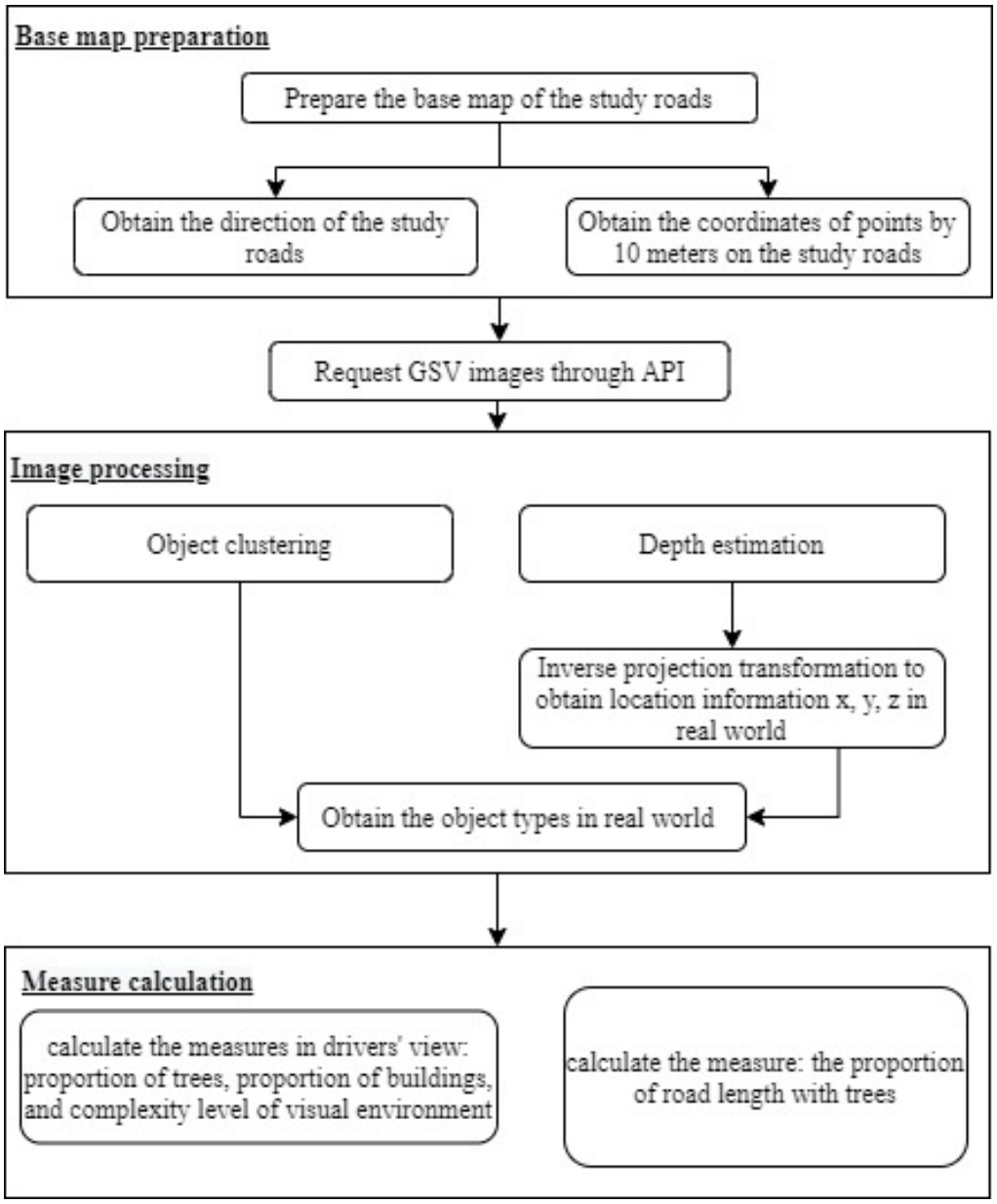


Figure 2-9 Flowchart of GSV image processing based on machine learning

2.1.2. Machine Learning for Crash Analysis

(1) Machine learning to estimate crash counts

In this study, the tree-based ensemble methods were used to estimate the crash counts. The ensemble learner utilizes decision trees as weak learners and generate the expectation of results based on the combined outputs of all learners. Usually, the estimation performance of

the ensemble learners is better than that of a single learner. Compared to other algorithms, the tree-based ensemble algorithms have the following major advantages [50-52]:

- The algorithms are non-parametric and don't assume that the data follow a specific distribution
- The multi-collinearity of features does not affect the accuracy of the model. Features do not need to be removed to decrease the correlations and interactions between them. Hence, the two variables related to trees could be used for the analysis at the same time.
- The algorithms are robust against overfitting since they include multiple weak learners that underfit (high bias) and combine the predictions into a stronger learner.

It should be noted that many studies have developed statistical models such as Poisson or negative binomial models to estimate crash counts [3, 53]. However, the recent studies have revealed that some features show clear non-linear relationships with the crash counts, which challenge the linearity assumption commonly used for the statistical models [54]. Meanwhile, the tree-based machine learning method could better capture the non-linear relationship and achieve better prediction accuracy compared to the statistical models [55].

The bagging and boosting are the two major ensemble methods of the tree-based models. The bagging is a parallel learning process. For each round, a random subset of samples is drawn from the training sample randomly but with the same distribution. These selected samples are then used to grow a decision tree (weak learner). Then, the average prediction value is chosen as the final prediction value. On the other hand, the boosting approach is an algorithm that trains the learners sequentially and assigns the weighting factor to each learner [56]. One bagging method (i.e., random forest) and two boosting methods (i.e., adaptive boosting and extreme gradient boosting) were adopted to estimate the crash counts in this study.

(a) Random forest

The random forest (RF) was developed by Breiman [57] based on the bagging approach. The RF approach involves two randomized procedures before searching for the optimal features and split points. First, a fixed number from the training set is selected randomly. Then, the RF selects random subsamples for each iteration of growing trees. The RF could reduce the overfitting based on the two procedures. The final prediction results of the RF are obtained by averaging the individual results of all learners.

(b) AdaBoost

The adaptive boosting (AdaBoost) was first introduced by Freund and Schapire [58]. Different from the RF, the AdaBoost provides sequential learning of predictors and adjusts weights to each observation based on the errors. Initially, all observations are weighted equally. Then, during the iterative training process, the observations which are incorrectly estimated by the learners will carry more weights. Therefore, the algorithm could adapt and reduce the bias iteratively.

(c) XGBoost

The gradient boosting framework introduced by Friedman [59]. Similar to AdaBoost, gradient boosting sequentially trains predictors and each one corrects its predecessor. However, instead of adjusting the weights for each incorrect estimation at each iteration, Gradient Boosting attempts to fit the new predictor to the residual errors made by the previous predictor. Gradient boosting is generally very slow in implementation due to the sequential modeling training. Extreme Gradient Boosting (XGBoost) is a relatively new algorithm proposed by Chen and Guestrin [60], which is an implementation of gradient boosting decision trees for speed and performance. The XGBoost provides a parallel tree boosting algorithm that could optimize the training process fast and accurately.

(2) Machine learning to interpret effects of features

While machine learning is expected to provide estimation results with high accuracy, it has been a key challenge to interpret the effects of variables on the output. In this study, the interpretability of tree-based ensemble models is explored to understand why a certain prediction is made so as to better suggest countermeasures to enhance transportation safety. The Shapley Additive exPlanations (SHAP) method, proposed by Lundberg and Lee [61], is used to measure the variable importance and interpret the effects. SHAP is a game theoretic approach to explain the output of the prediction model. The goal of SHAP it to explain the prediction for any feature as a sum of contributions from its individual feature values, while the contribution of each feature is allocated based on the marginal contribution [62]. Given a feature value i, the SHAP value could be obtained by:

$$\phi_i = \sum_{S \subseteq F} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} [f_{S\cup\{i\}}(x_{S\cup\{i\}}) - f_S(x_S)] \qquad (2\text{-}8)$$

where $|F|$ is the total number of features, S represents any subset of features that doesn't include the ith feature and $|S|$ is the size of that subset. $f_{S\cup\{i\}}(x_{S\cup\{i\}})$ indicates the model trained with i, and $f_S(x_S)$ is model trained without i. The SHAP value could help interpret the effects of features locally, which could help provide safety improvement strategies for a specific location. Besides, the SHAP value could be used to quantify the global impact of each risk feature by taking the average absolute impact on the model output magnitude: $\sum \frac{|\phi_i|}{n}$ (n is the total number of locations). The global measurement could be used to rank the feature importance and compare the impact among multiple risk factors.

2.2 Data

The data used in this study were collected from urban arterials in Central Florida. The urban arterials of nearly 75 miles were included and around 15,000 GSV images were requested and processed to get the indexes about drivers' visual environment. From the images, the proportion

of trees, the proportion of buildings, and the complexity level of the drivers' view were collected. In addition, the proportion of road length with trees was calculated based on the cluster and depth information. Figures 2-10, 2-11, and 2-12 illustrate the proportion of trees, proportion of buildings, and proportion of roadway length with trees on the study roads. It is shown that the buildings are concentrated in Areas 5 and 6, which is the City of Orlando and beach area. Meanwhile, the roadway segments with the high proportion of trees in drivers' view or proportion of roadway length with trees could be found in different areas.

The speeding crashes were collected from the Florida Department of Transportation (FDOT). In addition to the driving environment data collected from GSV, other exogenous variables were also collected, which included traffic data, roadway information, land use attributes, and socio-demographic for each segment. For traffic data, the Vehicle Miles Travelled (VMT) was obtained by multiplying the Average Annual Daily Traffic (AADT) by the segment length. Besides, the proportion of truck traffic, and daily transit frequency were collected from FDOT. In addition, the probe vehicle data INRIX were collected from RITIS by 5 minutes from 2017 to 2019. Based on each segment, the proportion of INRIX speed over the posted speed limit was calculated, which could indicate the general speeding trend. Seven roadway attributes that could be related to speed management strategies in the Florida Design Manual (FDM) were also identified. They are the indicator of narrow lane, average block length, the existence of median island on crossing, number of parking per mile, presence of road diet, length of the two-way-left-turn lane, and asphalt pavement. Other roadway variables such as lane number, speed limit, median type and width, shoulder type and width were also collected from FDOT. Finally, the land use and socio-demographic variables were also collected in this study. Table 2-1 summarizes the collected variables.

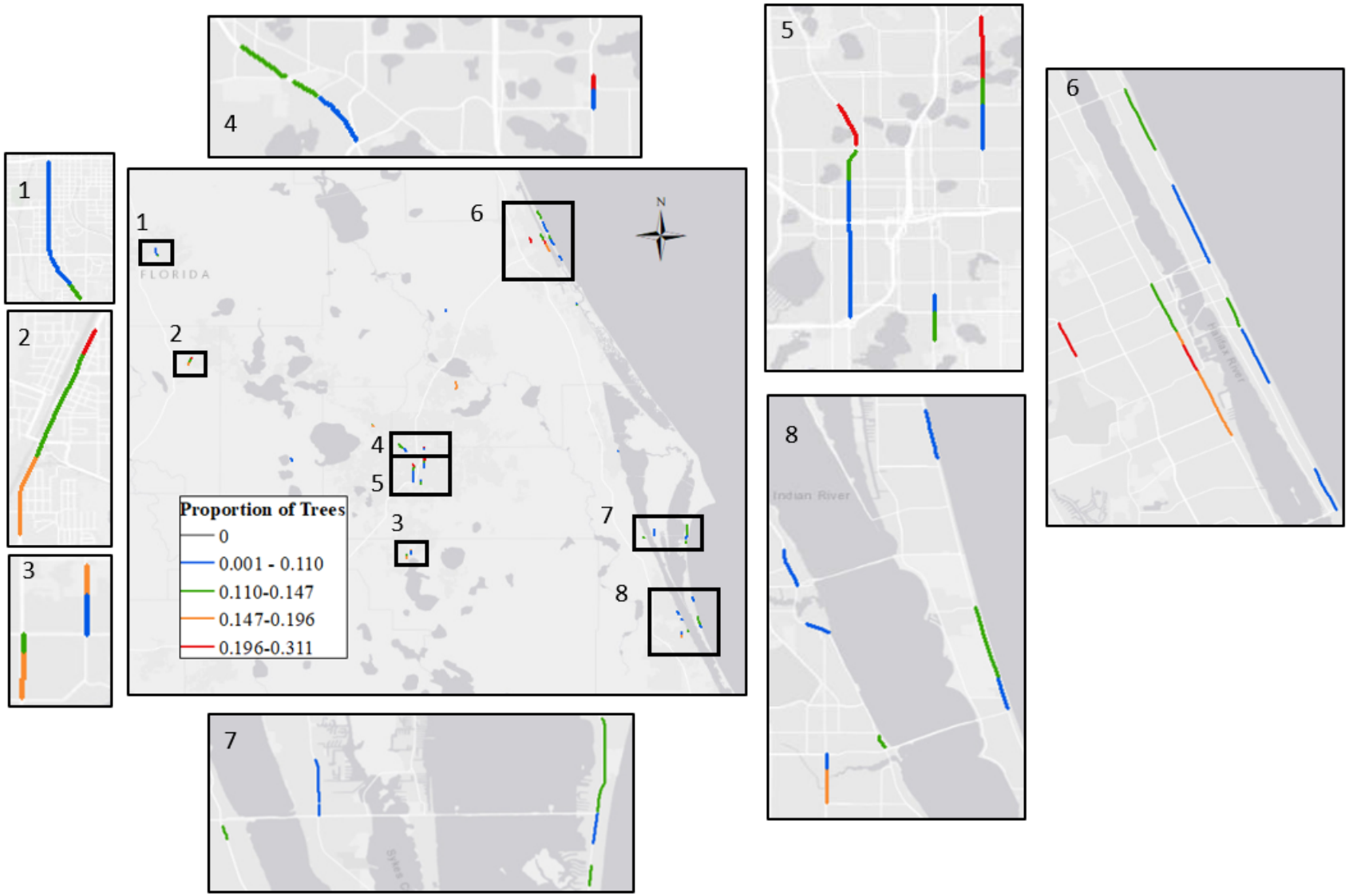


Figure 2-10 Proportion of trees in the drivers' view on urban arterials

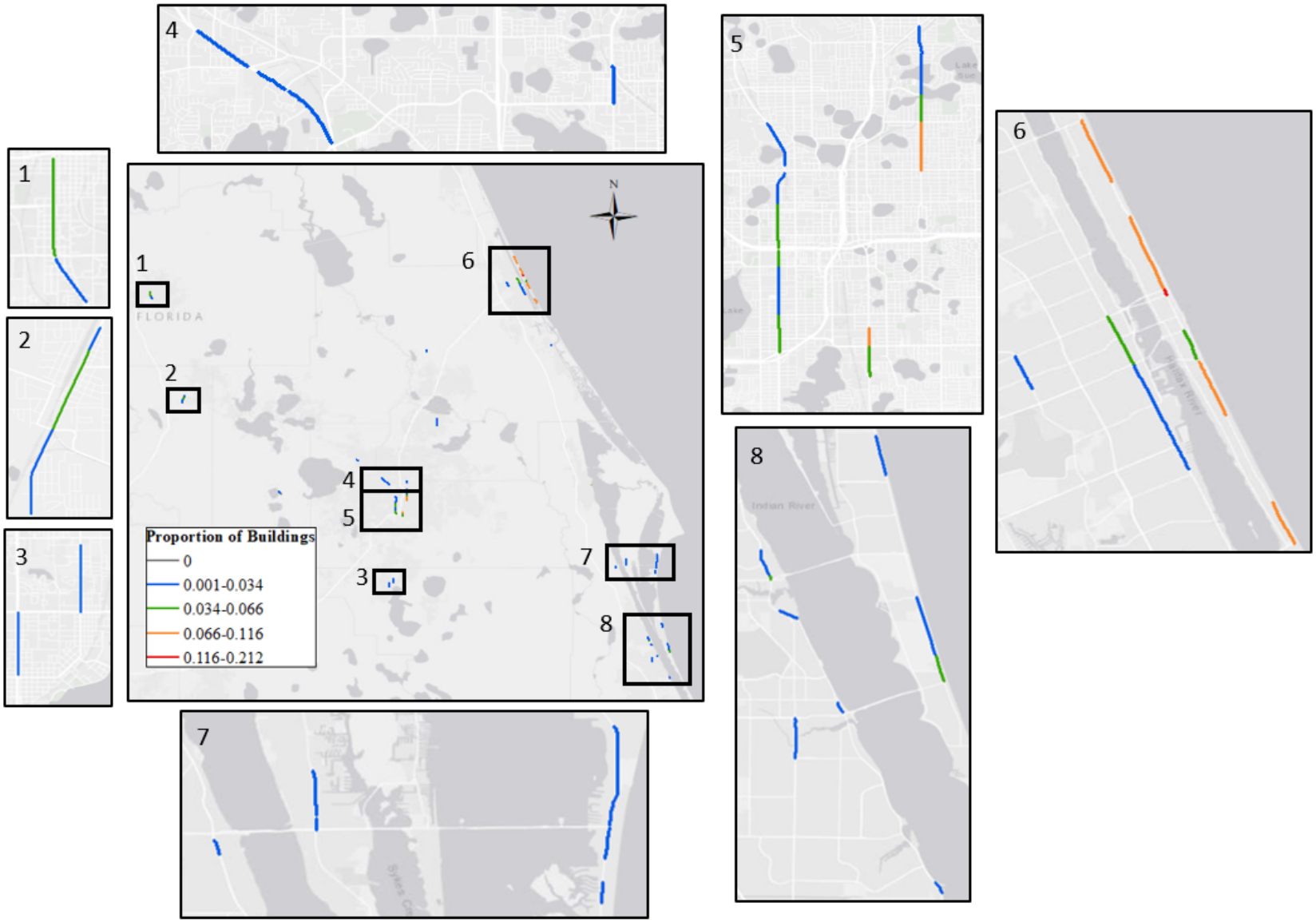


Figure 2-11 Proportion of buildings in the drivers' view on urban arterials

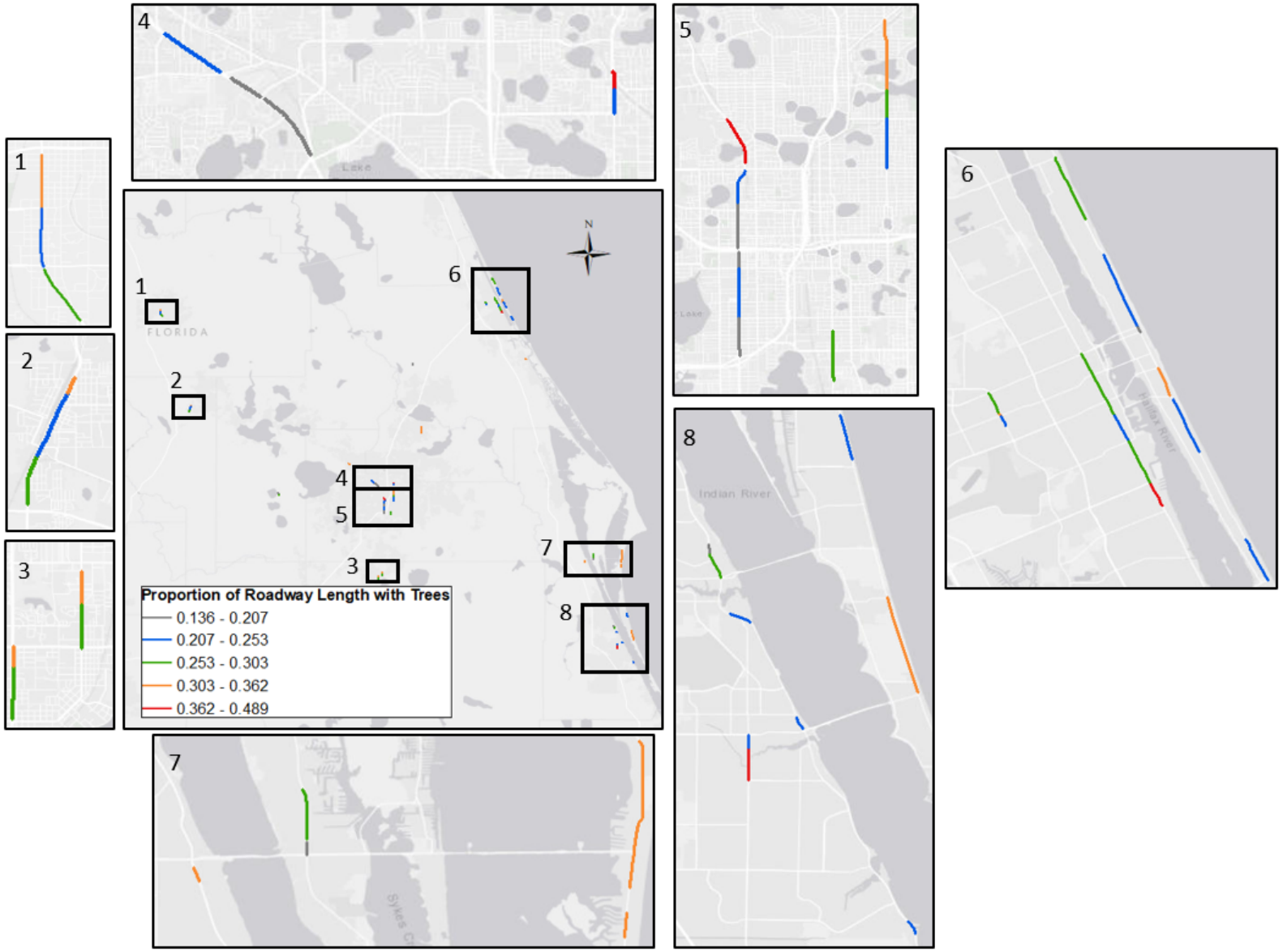


Figure 2-12 Proportion of roadway length with trees in the drivers' view on urban arterials

Table 2-1 Summary of Variables

Variable	mean	S.D.	min.	max.
Speeding crashes	1.24	1.51	0	8
Traffic variables				
Vehicle Miles Travelled (VMT)	14,505	8,920	2,145	40,523
Proportion of truck traffic	6.76	3.31	1.77	16.19
Average daily transit frequency	19.97	18.05	0	72
Proportion of INRIX speed data over the speed limit	0.18	0.14	0	0.78
Drivers' visual environment data from GSV images				
Proportion of tree view	0.13	0.06	0.03	0.31
Proportion of building view	0.03	0.03	0	0.22
Complexity level of driving view	0.73	0.03	0.65	0.79
Proportion of road length with trees	0.28	0.06	0.15	0.48
Roadway variables				
Variables related to FDM speed management strategies				
Indicator of asphalt pavement (1: yes; 0: no)	0.84	0.36	0	1
Indicator of narrow lane (lane width<12 feet) (1: yes; 0:	0.36	0.48	0	1
Average block length (mile)	1.60	3.26	0.06	10.33
Existence of median island on pedestrian crossing (1:	0.07	0.25	0	1
Log (number of parking spot per mile)	0.27	1.12	0	6
Presence of road diet (1: yes; 0: no)	0.44	0.5	0	1
Log (length of two-way-left-turn lane)	0.13	0.23	0	1
Other roadway variables				
Number of lanes	2.10	0.52	1	4
Speed limit (mph)	38.59	5.2	25	55
Pavement condition	4.17	0.8	0	5
Raised median (1: yes; 0: no)	0.44	0.5	0	1
Median Width (feet)	16.53	9.32	0	55.07
Curb, gutter inside shoulder type (1: yes; 0: no)	0.34	0.47	0	1
Width of inside shoulder (feet)	0.97	1.47	0	9
Curb, gutter outside shoulder type (1: yes; 0: no)	0.48	0.5	0	1
Width of outside shoulder (feet)	3.45	1.88	0	10
Proportion of sidewalk length	0.91	0.25	0	1
Sidewalk width (feet)	5.09	1.35	0	10
Proportion of bike lane length	0.12	0.3	0	1
Proportion of bike slot length	0.01	0.05	0	0.64
Number of signalized intersections per mile	3.11	3.15	0	16.98
Number of access per mile	9.52	5.86	0	28.37
Land use and socio-demographic variables				
Proportion of residential land use	0.29	0.38	0	1
Proportion of commercial land use	0.04	0.09	0	0.62
Land use mix	0.04	0.15	0	0.83
Proportion of population below poverty	0.06	0.11	0	0.73
Proportion of zero-vehicle household	0.02	0.04	0	0.22
Proportion of commuters by walking or biking	0.02	0.04	0	0.20

2.3 Results and Discussion

2.3.1 Model Development with K-Fold Cross-Validation

The data was randomly split into training and testing datasets with a ratio of 2:1. In addition, a 5-fold cross-validation was implemented to train the three tree-based ensemble models. The cross-validation is to overcome the overfitting issue and ensure the models' reliability in predicting crash counts in a new dataset. The mean absolute error (MAE), root mean squared error (RMSE), and R^2 were used to assess the model performance. These measures, which could directly reflect the difference between the observations and predictions, have been widely employed for evaluating the model performance in the machine learning studies [37, 56]. A tuning process was applied to determine the best set of parameters for each ensemble model, especially the maximum depth of trees and number of trees to control the overfitting of the models. For AdaBoost, the best prediction result was determined only based on the number of trees, as AdaBoost has no predetermined maximum depth. The cross-validation training results and testing results of the three models are summarized in Table 2-2. It is clearly shown that the boosting methods could gain significantly better performance than the bagging method for estimating crash counts. Besides, by comparing between two boosting methods, it could be found that XGBoost is able to provide significantly more accurate predictions. Hence, the trained XGBoost model will be used in the following analysis of feature effects.

Table 2-2 Summary of results

Model	Maximum depth	Number of trees	MAE		RMSE		R^2	
			Training	Testing	Training	Testing	Training	Testing
Random Forest (RF)	6	50	0.8	0.87	1.12	1.33	0.74	0.68
Adaptive Boosting (AdaBoost)	-	140	0.77	0.81	1.01	1.23	0.81	0.71
eXtreme Gradient Boosting (XGBoost)	6	260	0.41	0.42	0.68	0.74	0.92	0.84

2.3.2 Effects Analysis

Figure 2-13 shows the global feature importance (left bar chart) and local explanation summary plot (right plot). For the local explanation summary, the red color indicates larger value of the explanatory variables. Unsurprisingly, the VMT significantly prevails as the most important feature. Moreover, higher values of this feature result in higher SHAP values, corresponding to more speeding crashes. Meanwhile, other two traffic related variables which are speeding proportion and proportion of truck traffic are relatively important variables. As the speeding proportion reflects the speeding level, higher speeding proportion could lead to more speeding-related crashes. As revealed in many studies, the proportion of heavy traffic is negatively associated to the crash occurrence [3, 53].

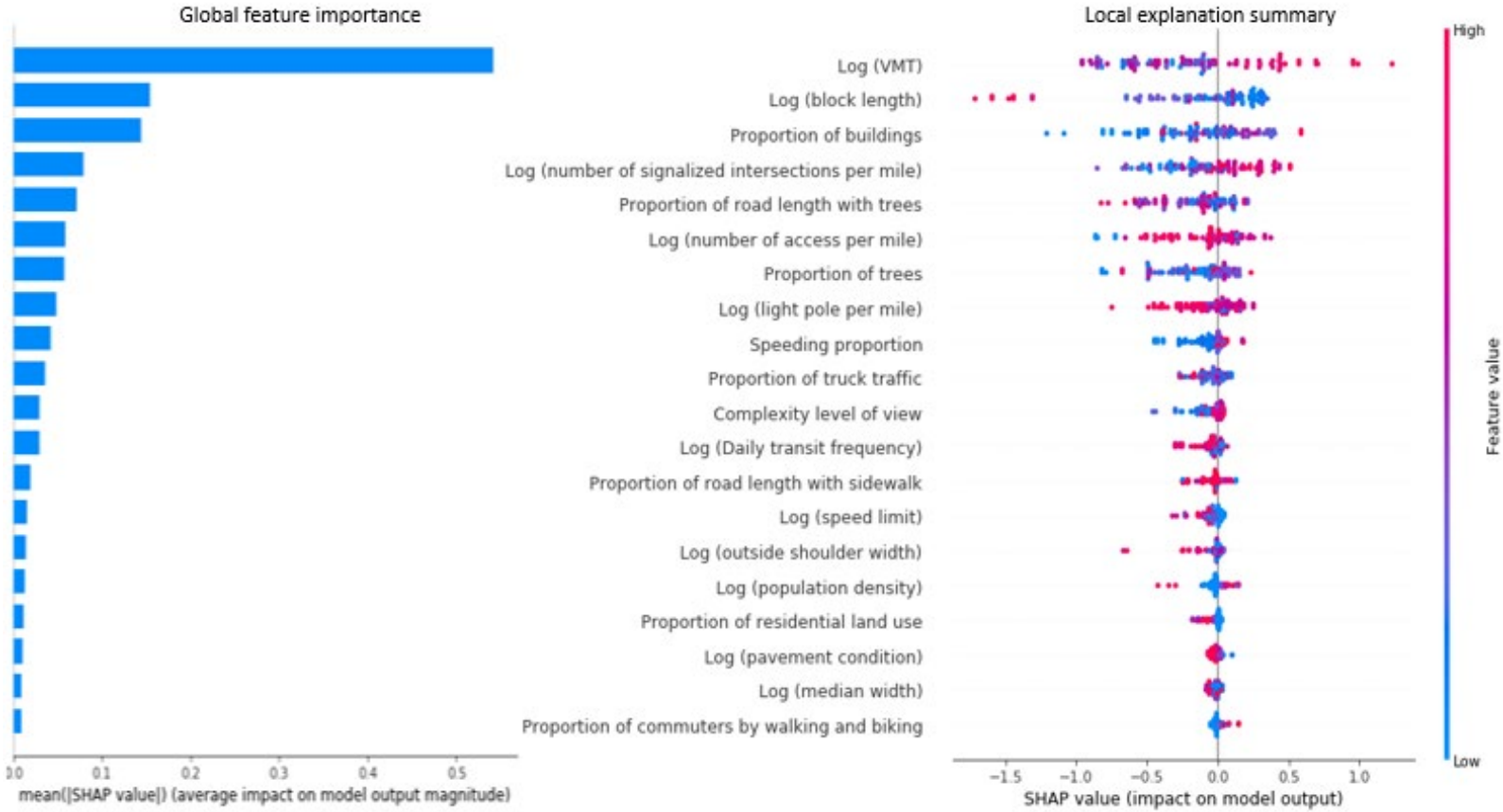


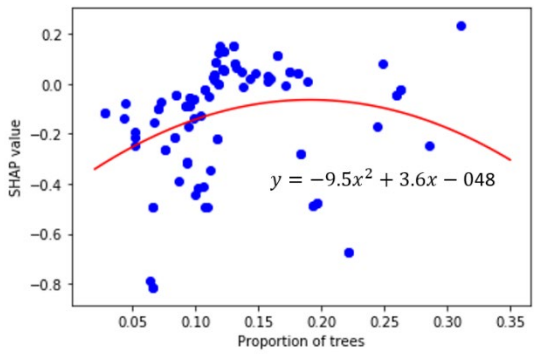
Figure 2-13 Global feature importance and Summary of SHAP value

The four variables related to drivers' view, three variables which are the proportion of buildings, the proportion of road length with trees, and the proportion of trees are among the top ten important variables among all explanatory variables. However, their effects are not clear from the summary plot in Figure 2-13. Hence, the scatter plot of variables related to drivers' visual environment vs SHAP values are presented in Figure 2-14. Polynomial functions with degrees from 1 to 3 were applied to the data and the model with the best fit was plotted in the figures. As shown in Figure 2-14(a), a polynomial function with degree of 2 could provide the best fit for the proportion of trees in drivers' view. As noted above, the proportion of trees is related to trees' canopy. In general, the trees in drivers' view could reduce the speeding crashes as most of SHAP values are negative, which is consistent with the previous study by Marshall et al. [9]. The magnitude of the negative effect decreases when the tree proportion increases from 0 to 0.15, and then the magnitude increases with the increase of the tree proportion. The linear relation between the proportion of road length with trees and crash count is shown in Figure 2-14(b). It suggests that the proportion of road length with trees could reduce the number of speeding crashes. The previous driving simulator study found that drivers tend to decrease their speeds significantly and move toward the centerline of the road when trees are present [63]. Besides, it was found that the crashes could decrease with the increase in the tree density [9]. With the effects of trees on speeding and crash occurrence, it is reasonable to find a negative effect of trees on speeding crashes. Figure 2-14(c) illustrates the effects of the building proportion on the crash counts, and a 3-degree polynomial could provide the best fit. It reveals a negative effect of the building proportion on crash counts when the building proportion is very small. With the increase of the building proportion, its effect on speeding crashes increases and becomes positive when its value is smaller than 0.05. Then, the effect of the building proportion decreases as its value increases from 0.05 to 0.2. In addition to the three variables, the complexity level of drivers' view also has a restively important effect on speeding crashes. Both Figure 2-13 and Figure 2-14(d) clearly show the effect of complexity level. With a linear relation

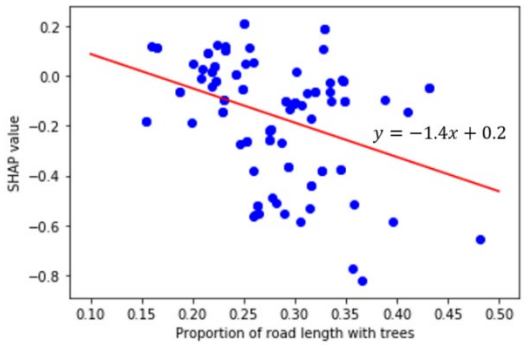
revealed in Figure 2-14(d), it shows that a complex view could lead to more crashes, with some exceptions when the complexity level is around 0.71 or 0.78. The result is reasonable particularly that a previous study suggested that drivers have difficulty in reacting for an emergency when they have to deal with the increased visual complexity [64].

The effects of several other features were also revealed in Figure 2-14. First, it is found that the intersection density could result in more crashes as intersections could increase traffic interaction [3]. Second, it is expected that less speeding crashes could be found on roads with higher speed limit. In addition, Figure 2-14 shows that the wide outside shoulder could reduce the crashes, which is in line with the previous study [65]. Finally, the speeding crashes are less likely to occur in the residential area.

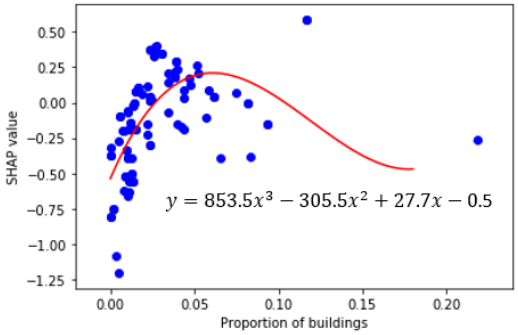
The developed explainable machine learning models could well balance the prediction accuracy and the identification of factors' effects. At the planning level, the developed model could be used to screen the road network and identify the hotspots with high speeding crash risks. For each hotspot, the explainable machine learning model could identify the local effects of visual environment factors and the corresponding engineering solution could be applied to reduce the crashes. For example, more trees could be added along the road to reduce drivers' speeding probabilities, which could reduce the occurrence of speeding crashes. Besides, the warning sign could be added at a road segment to remind drivers not to over speed if the segment's visual complexity level is high.



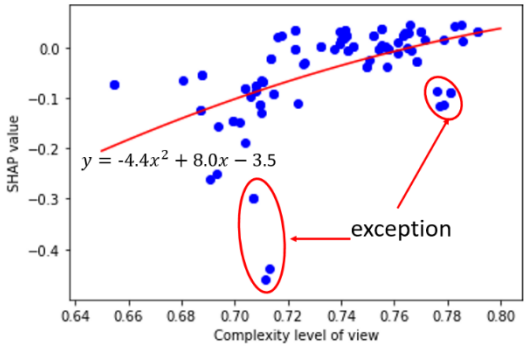
(a) SHAP value vs Proportion of trees in drivers' view



(b) SHAP value vs Proportion of road length with trees



(c) SHAP value vs Proportion of buildings in drivers' view



(d) SHAP value vs complexity level of view

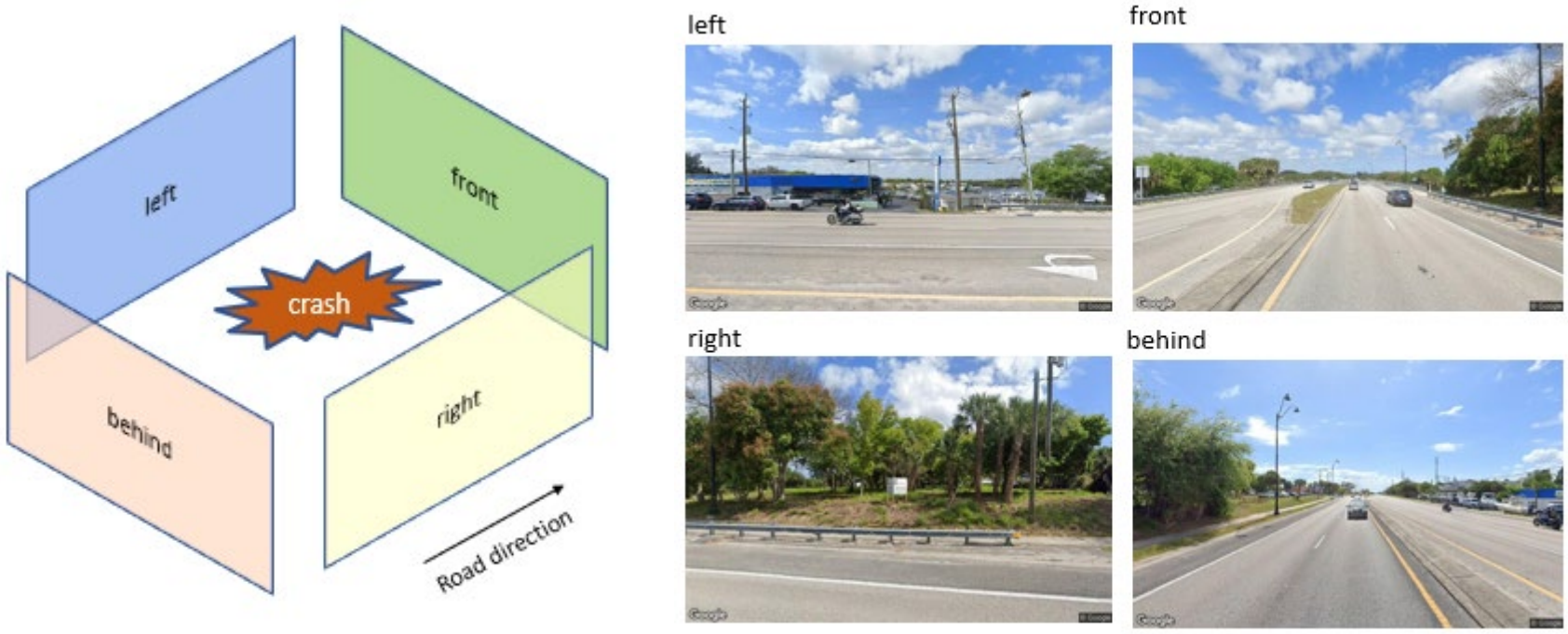
Figure 2-14 Scatter plot of SHAP values vs variables related to drivers' visual environment

3 Exploring the Effects of Drivers' Visual Environment on Single-Vehicle Crash Severity

3.1 Methodology

3.1.1. GSV Images Collection and Processing

The GSV could provide 360° surrounding images generated from eight original images that were captured by multiple cameras and stitched together in a sequence. The GSV images could be downloaded in an HTTP URL format using the GSV API provided by Google. To request a static GSV image, the heading and angle of the view could be customized for a location with coordinate information. To reflect the road environment from the GSV images, four images representing left, front, right, and behind were requested and each image has a field of 90° view. Figure 3-1 illustrates an example that includes the four-side images for a crash location. The direction information based on crash data is used to determine the headings of images. For example, if a crash occurred at northbound of a road that has a bearing (i.e., a clockwise angle from the north direction) of 10°, the headings to request the front, right, behind, and left sides images are 10°, 105°, 195°, and 285°, respectively. In this study, a Python script was prepared to download the GSV images automatically based on the crash coordinates that were obtained from the crash database. The resolution for each image is 400×640.



(a) Illustration of 4 GSV image locations (b) Examples of 4 GSV images

Figure 3-1 Illustration of GSV images for 4 sides

After collecting the GSV data for the crash locations, computer vision techniques are utilized to identify objects from the images. The recent development of deep learning techniques makes it possible to accurately recognize objects in images. In this study, “Detectron2” was applied to detect objects. Detectron2 is a Facebook Artificial Intelligent (AI) research next-generation software system that implements state-of-the-art object detection algorithms, including Mask R-CNN [43]. The previous research has validated that the Detectron2 framework could reach the state-of-the-art performance for detecting objects from street images [44]. As shown in Figure 3-2, different objects in the GSV images such as roads, trees, buildings, fence, and sky, could be identified from the images. Based on the detection results, the object types (e.g., road, sky, tree) for each pixel in the images could be labeled. Meanwhile, as this study attempts to utilize GSV images to analyze the effects of road environment on crash severity, the labels of cars and persons on roads were replaced by road.

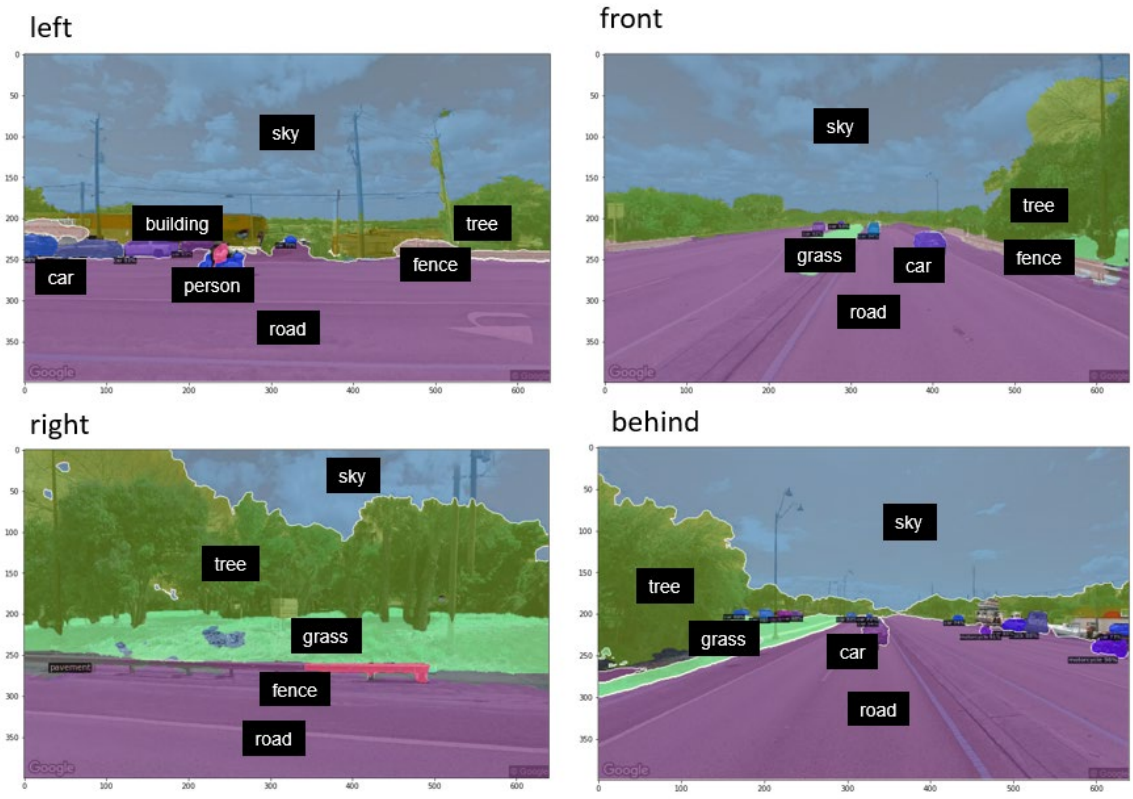
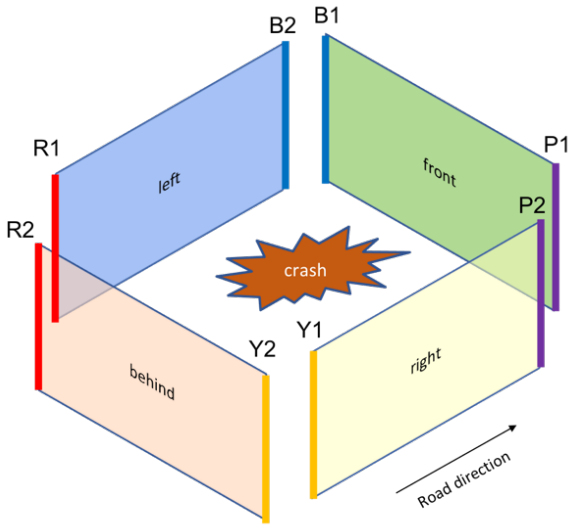


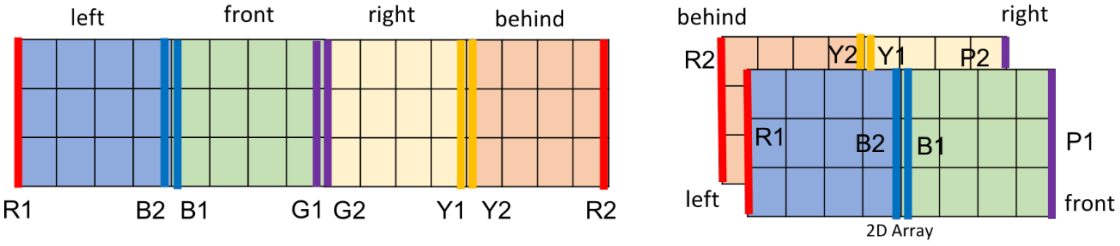
Figure 3-2 Illustration of semantic segmentation results

For each image, a 2D matrix of 400×640 could be generated to represent the object type at each pixel in the image. Thus, four 2D matrixes were obtained for a crash. Figure 3-3 illustrates the data matrix generation process based on GSV images. As GSV images represent the surrounding environment of the crash events, the edges of the images are spatially connected with each other in the real world. In Figure 3-3(a), the edges with the same colors indicate the real-world spatial connection for the two edges, including edges R1&R2, edges B1&B2, edges P1&P2, and edges Y1&Y2. The 4 side GSV images for each crash need to be combined for the prediction process. To combine the data of the 4 side images, one method is to stitch the data at a same layer. Thus, a new matrix of 400×2560 (640×4=2560) could be generated (Figure 3-3(b)). However, as shown in Figure 3-3 (b), the spatial connection with edge R1 and R2 will not exist if this combination method is utilized. To overcome this limitation, the GSV image data of the 4 sides were augmented into two layers of 2D data matrix in this study (2×400×1280) (Figure 3-

3(c)). Moreover, in order to be consistent with the real-world spatial connection, the images of behind and right were flipped before the combination. Thus, the edges with the same colors are spatially connected in the data matrixes.



(a) Illustration of 4 GSV image locations



(b) One layer of the 2D data matrix (c) Two layers of the 2D data matrix

Figure 3-3 Illustration of data matrix for the GSV image

3.1.2. Crash Severity Prediction Model

In this study, the crash severity prediction is based on both high-dimensional (i.e., GSV image data) and low-dimensional data (i.e., crash information). As shown in Figure 3-4, a concatenated neural network architecture is developed to simultaneously process the two databases. Two fully connected layers corresponding to the two databases are concatenated into a layer for the final prediction output. The convolutional neural network (CNN) is adopted for the GSV image data in this study since the data is in 2 dimensions and has 2 layers. The CNN structure is widely used in image recognition and has proven to be a powerful tool to understand the spatial interaction in

the local matrix units. CNN models have been applied in recent research for traffic crash prediction [37, 66, 67]. This method captures the local correlation through convolution layers instead of general matrix products, which in turn reduces the total number of adjusted parameters. By using multiple small filters, the input matrix layers could be projected to multiple convolutional layers. The max-pooling layer is used to reduce the number of adjusted parameters. Besides, batch normalization (BN) and dropout techniques are adopted to avoid overfitting. On the other hand, the dense layer is applied to connect the data of crash information before the concatenation. As noted above, the crash severity levels were divided into two levels: severe and non-severe. Hence, the Softmax activation function is adopted as the last output layer of the prediction model.

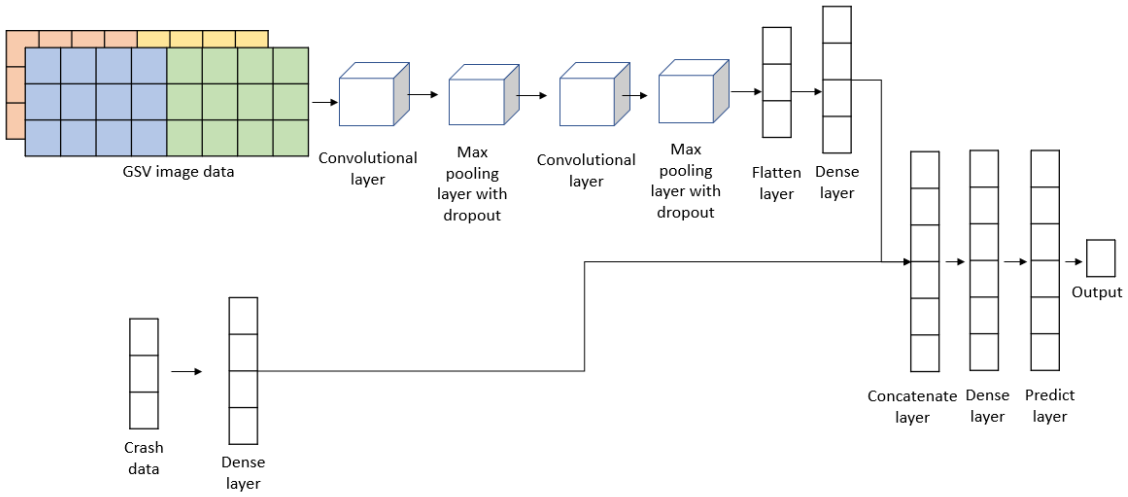


Figure 3-4 Network architecture of the concatenated prediction model

Table 3-1 summarizes the architecture of the concatenated deep learning model in this study. The model comprises two parts, while one is for the parallel layers and the other is for the concatenated layers. The parallel layers include the CNN model for the GSV image data and the NN model for the crash information data. The CNN model has 6 layers, of which 2 layers are convolution layers, 2 max-pooling layers, and 2 fully connected layers. Each convolution layer is followed by a max-pooling layer. The stride =1 and padding =1 were used to ensure that the dimension of the GSV image data matrix decreased gradually. The NN model has 1 layer

connected to the input layer. The CNN model and NN model were estimated at the same time. Then, the 2 output layers from the CNN architecture and NN models are concatenated into 1 layer, which is followed by 2 fully connected layers to get the output. Batch Normalization (BN), ReLU, and dropout techniques were used to train the model and avoid overfitting. With the trained concatenated model, both GSV image data and crash data could be used simultaneously to predict crash severity.

Table 3-1 Architecture for the concatenated model

Architecture for the parallel layers				NN architecture for the concatenated layer		
	#Layer	**Layer type**	**Layer description**	**#Layer**	**Layer type**	**Layer description**
CNN architecture for GSV image data	1	Convolution +BN +ReLU	Filter size=5×5 Stride=1, Padding=1 Output size=4×396×1276	1	Concatenate layer	#Neuron1=256 #Neuron2=256 Output size =512
	2	Max pooling +Dropout (0.2)	Padding=2, stride=2 Output size=4×198×634	2	Fully connected +BN +ReLU +Dropout (0.2)	#Neuron=512 Output size =64
	3	Convolution +BN +ReLU	Filter size=5×5 Stride=1, Padding=1 Output size=8×194×634	3	Fully connected +ReLU +Dropout (0.2) +Softmax	#Neuron=64 Output size=2
	4	Max pooling +Dropout (0.2)	Padding=2, stride=2 Output size=8×97×317	-	-	-
	5	Flatten layer	#Neuron=8×97×317 Output size=245992	-	-	-
	6	Fully connected +ReLU +Dropout (0.2)	#Neuron=245992 Output size=256	-	-	-
NN architecture for crash data	1	Fully connected +BN +ReLU +Dropout (0.2)	#Neuron=42 Output size=256	-	-	-

3.1.3. Interpreting the Deep Learning Model through Activation Mapping

One of the common concerns for NN or CNN algorithms is the models are perceived as “black box” and lack the understanding of their internal functions. In order to overcome the limitation, some efforts have been conducted to explain and interpret machine learning or deep learning models. In 2017, Selvaraju et al. proposed the approach named “gradient-weighted class activation mapping (Grad-CAM),” which interpreted CNN models by highlighting the important regions in the images [68]. In this study, Grad-CAM is used to identify the important regions in images affecting the crash severity. Given input image matrix and the trained deep learning model, Grad-CAM could generate a localization map by using the gradient information of the specific target class (i.e., severe crash) to compute the target class weight of each feature map

of the last convolutional layer of the CNN model before the final classification. Then, the final localization map is synthesized from the sum of these target class weights. The Grad-CAM map for the severe crashes could help to visually identify the patterns and objects learned by the CNN and interpret the reason why the severe crash occurred [23]. Since the images have been fully segmented, the identified objects could be overlapped with the dangerous regions at the pixel level. By analyzing the number of important pixels, the object categories more relevant to determine severe crashes could be identified.

3.2 Crash Data

Single-vehicle crashes that occurred from 2017 to 2019 in Central Florida were collected from the Signal Four Analytics (S4A) database. In total, 6,420 single-vehicle crashes on partially controlled-access roads were collected. The crash severity is categorized into five levels: (1) fatal crash; (2) incapacitating injury crash; (3) non-incapacitating injury crash; (4) possible injury crash; and (5) property damage only (PDO). In this study, the fatal and incapacitating injury crashes were combined as severe crashes, and the rest of the crash severity levels were labeled as non-severe crashes. The frequencies of the severe and non-severe crashes were 1,774 (27.6%) and 4,646 (72.4%), respectively.

The crash data variables were divided into five categories, including driver, crash, environment, roadway, and traffic. The driver, crash, and environment information were collected from the crash database (i.e., S4A). Meanwhile, the information related to roadway and traffic was obtained from the Florida Department of Transportation (FDOT) database based on crash locations. The frequency and percentage descriptions were summarized for the dummy variables (Table 3-2), while the mean and standard deviation were described for the continuous variables in Table 3-3. The inclusion of different variables is based on the previous relevant studies about crash severity [24, 26, 30].

Table 3-2 Statistical description of dummy variables

Category	Variable	Frequency	Percentage
Driver information			
Gender	Male	4157	64.75%
	Female	2263	35.25%
Age	19 years old or younger	579	9.02%
	Between 20 and 24 years old	2590	40.34%
	Between 25 and 54 years old	1799	28.02%
	Between 55 and 64 years old	790	12.31%
	65 years old or older	662	10.31%
Citation	Indicator of citation involvement	1777	27.68%
Driving under influence	Indicator of DUI involvement	455	7.09%
Distraction related	Indicator of distraction	727	11.32%
Crash characteristics			
Location of first harmful event	Off roadway	1534	23.89%
	On roadway	2216	34.52%
	Shoulder	298	4.64%
	Median	147	2.29%
Intersection-related	Indicator of intersection related	728	11.34%
Work zone related	Indicator of work zone related	155	2.41%
Obstruction on roadway	Indicator of obstruction	91	1.42%
Urban area	Indicator of urban area	4025	62.69%
Day of week	Weekday	4590	71.50%
	Weekend	1830	28.50%
Time of day	Peak (6am-10am and 4pm-7pm)	1887	29.39%
	Off-peak (10am to 4pm)	2003	31.20%
	Evening (7pm-12am)	1456	22.68%
	Before dawn (12am-6am)	1074	16.73%
Environmental features			
Light condition	Daylight	3536	55.08%
	Dark with lighting	1970	30.69%
	Dark without lighting	914	14.24%
Adverse weather	Rain	722	11.25%
	Fog	38	0.59%
Wet road surface	Wet	1133	17.65%
Roadway features			
Functional classification	Arterial	4341	67.62%
	Collector	2042	31.81%
	Local road	37	0.58%
Median type	Paved median	1370	21.34%
	Raised separation	733	11.42%
	Vegetation or curb	2334	36.36%
	no median	1983	30.89%

Table 3-3 Statistical description of continuous variables

Category	Mean	Standard deviation
Roadway features		
Median width	17.48	20.58
Pavement condition	3.78	0.76
Speed limit	42.84	9.04
Number of lanes	2.08	0.60
Lane width	11.76	1.04
Traffic features		
AADT	21339.92	15484.97
Percentage of heavy vehicle volume (%)	7.35	5.09

3.3 Results and Discussion

3.3.1. Model Development and Comparison

In this study, a total of 6,420 single-vehicle crashes (1,774 severe crashes and 4,646 non-severe crashes) were collected to validate the proposed prediction framework. Among the crashes, 70% were selected for training the severity prediction and 30% were employed for testing the trained model. The model was tuned based on Pytorch using NVIDIA RTX 2080 Ti 11G GPU. The model hyperparameters are optimized using a grid-search approach and the values applied in the final model are presented in Table 3-4.

Table 3-4 Parameter tuning results

Name	Tested values	Selected values
The 1st CNN convolutional filter	4, 8, 16, 32	4
The 2nd CNN convolutional filter	4, 8, 16, 32	8
Dropout rate	0.2, 0.4, 0.6	0.2
Optimizer	RMSprop, SGD, Adam	Adam
Learning rate	0.01, 0.001, 0.0001	0.01
Epoch number	20, 40, 60, 80, 100	80
Batch size	50, 100, 200, 400	100

To validate the proposed concatenated model, a CNN model using the GSV image only and a NN model using crash data only were developed for model performance comparison. The

following measures are utilized in this study for model performance evaluation, which have been widely used in the previous literature:

$$Precision = \frac{number\ of\ true\ positive}{number\ of\ true\ positive + number\ of\ false\ positve} \quad (3\text{-}1)$$

$$Recall = \frac{number\ of\ true\ positive}{number\ of\ true\ positive + number\ of\ false\ negative} \quad (3\text{-}2)$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (3\text{-}3)$$

Table 3-5 demonstrates the performance of the three models for both training and test datasets. It indicates that the proposed model that utilized both GSV image and crash data achieved the best performance for both training and test datasets. For the proposed model, the recall, precision, and F1-score for the test datasets are 0.707, 0.843, and 0.769, respectively. A NN model was developed only based on the crash information data. The comparison between the NN model and the proposed model illustrates that using GSV images data (i.e., high-dimensional data) could be beneficial to identify severe crash-prone locations, as more information could be obtained from the images. Meanwhile, the comparison between the CNN model and the proposed model indicates the crash data (i.e., low-dimensional data) could provide additional information compared with the GSV image, including the driver, crash characteristics, environment, traffic, and roadway information. Thus, the proposed concatenated model that can incorporate both image data and numerical data (e.g., crash data) can be utilized to better predict the crash severity.

Table 3-5 Model performance

	Model	Data	Recall	Precision	F1-score
Training	NN	crash data only	0.747	0.804	0.775
	CNN	GSV image data	0.856	0.908	0.881
	Proposed model	crash data and GSV image data	0.889	0.910	0.900
Test	NN	crash data only	0.656	0.725	0.689
	CNN	GSV image data	0.667	0.744	0.703
	Proposed model	crash data and GSV image data	0.707	0.843	0.769

3.3.2. Identification of Important Features to Explain the Severe Crashes

Based on the Grad-CAM, the contributing weight at each pixel could be estimated. The weight ranges from 0 to 1, indicating the importance from low to high. Figure 3-5 shows the examples of the important regions' spatial distributions in the images for the predicted severe crashes. The important regions could be utilized to explain the reason that a crash occurring at this location tends to be a severe crash. Based on the pixel information, the feature categories (e.g., road, tree, grass, and pavement) could also be identified in the important regions from the object segmentation outputs. The engineering solutions could be applied to the identified important regions to reduce the likelihood of severe crashes.

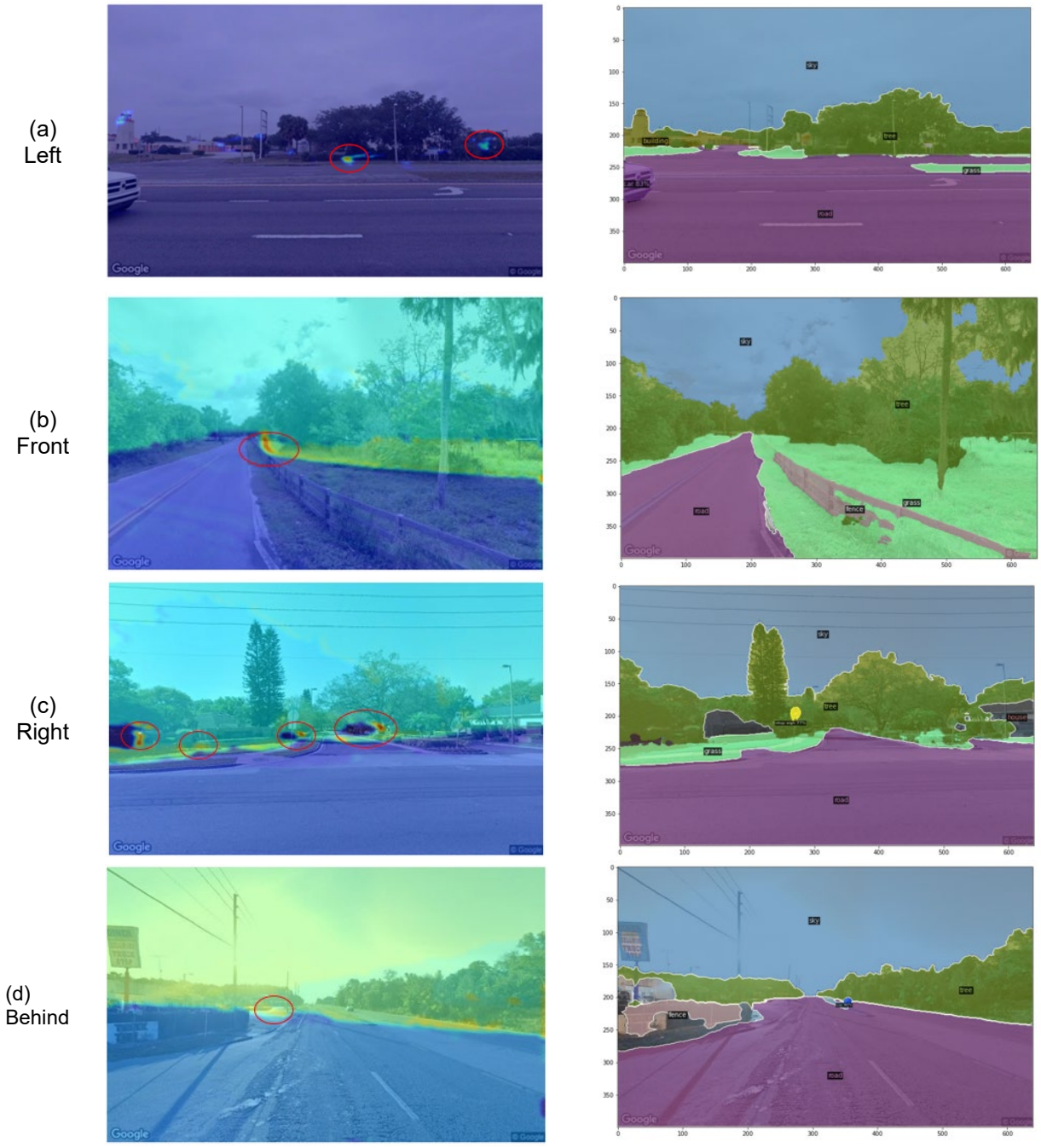


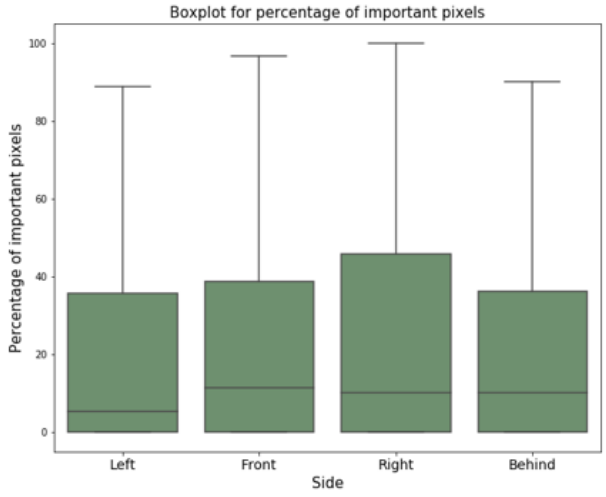
Figure 3-5 Examples of important pixels with object segmentation to explain severe crashes

In this study, the threshold of 0.6 was used to categorize the important pixels, which could be adjusted to identify less or more important pixels. Both the total number of important pixels at each side of the crash location and the total number of important pixels at the crash location could

be obtained from the images. Then, the percentage of important pixels at each side could be estimated by:

$$P_{side,ij} = \frac{Total\ number\ of\ important\ pixels\ at\ side\ i}{Total\ number\ of\ important\ pixels\ at\ location\ j} \times 100 \qquad (3\text{-}4)$$

The ANOVA test was conducted for $P_{side,ij}$ and it was suggested that there are significant differences among the four sides regarding the important pixels leading to severe crashes (F=5.814, P-value<0.001). Turkey's range test was conducted to compare the percentage of important pixels by a pair of sides. As shown in Figure 3-6, significantly more important pixels could be found at the right side, with 4.45%, 2.84%, and 3.44% more pixels compared to the left, front, and behind sides. While there is no significant difference among the left, front, and behind sides, relatively less important pixels were as expected found on the left side.



Tukey's range test

group1	group2	mean of group1 (%)	mean of group2 (%)	Difference (%)	p-value
Left	Front	23.23	24.85	-1.61	0.474
Left	Right	23.23	27.68	-4.45	0.001
Left	Behind	23.23	24.24	-1.01	0.780
Right	Behind	27.68	24.24	3.44	0.011
Right	Front	27.68	24.85	2.84	0.055
Front	Behind	24.85	24.24	0.61	0.900

Figure 3-6 Comparison of important pixels in different sides

At each side, the percentages of important pixels by the object category were also computed and presented in Figure 3-7. At each side, the trees account for more important pixels compared to other object categories. On average, around 30% of the important pixels are identified as trees. It is consistent with the previous studies that a collision with a tree is more likely to be severe [69].

At the front and behind sides, higher percentages of important pixels are found as roads. It is expected since more road views could be observed from those two sides.

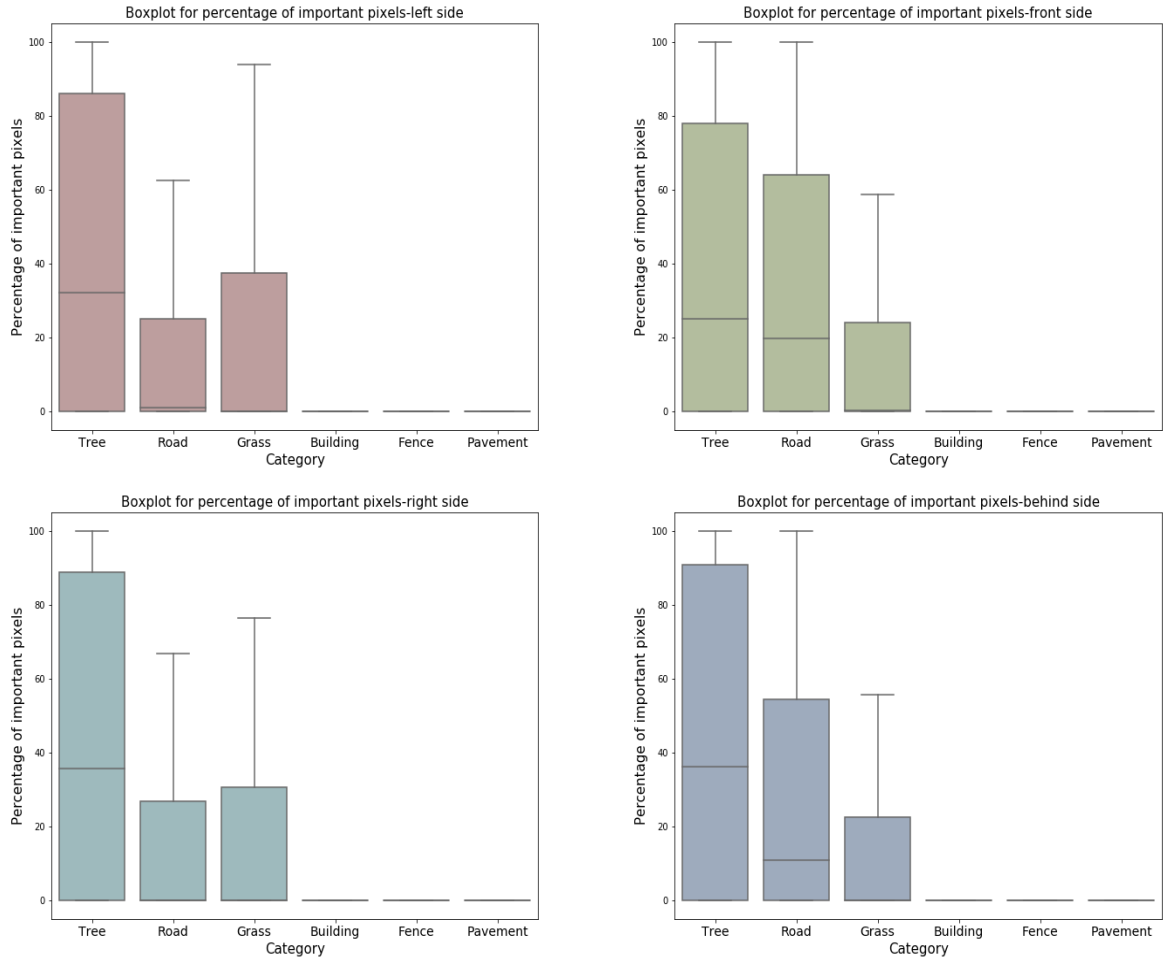


Figure 3-7 Summary of important pixel by object category

4 Conclusions

This study applied different machine learning methods to explore the effects of drivers' visual environment on speeding crashes. Around 15,000 Google Street View (GSV) images of urban arterials were queried through the Google API based on the features of the study roads. The deep neural network model developed by Facebook was used to cluster objects in the images. Based on the clustering results, indexes including the proportion of trees, the proportion of buildings, and the complexity level of the visual environment were calculated by counting the number of pixels of each cluster. Besides, another deep learning method was applied to get distance information from the images. By combining the clustering information, a 3D point cloud data was generated for each GSV image. The proportion of road length with trees was calculated. The information reflects the environment information from the drivers' view and was used to explore its effects on speeding crashes.

Three tree-based ensemble models (i.e., random forest, adaptive boosting (AdaBoost), and eXtreme Gradient Boosting (XGBoost)) were applied to estimate the number of speeding crashes. The comparison results suggested that the XGBoost could provide the best fit. The explainable machine learning method was used to explore the effects of information extracted from GSV images on the speed crashes. Other factors including traffic volume, speeding proportions, road attributes, land use, and socio-demographic characteristics were also examined. The result revealed that features related to drivers' visual environment are very important contributing factors for speeding crashes on urban arterials. It was suggested that the proportion of trees in drivers' view and the proportion of road length with trees could reduce the speeding crashes. On the other hand, the complexity level of the visual environment could lead to more speeding crashes. The results validated that more insight could be obtained by using deep learning algorithms to extract detailed information from GSV images. Besides, other significant factors for speeding crashes

were also revealed, such as traffic explore, speed limit, outside shoulder width, and intersection density.

In addition to the speeding crash analysis, the effects of visual environment on crash severity have been also explored. In this study, 25,680 GSV images were extracted at the locations where 6,420 single-vehicle crashes that happened on partially controlled-access roadways. For each crash location, 4 GSV images were collected (i.e., front, left, right, behind) with the consideration of crash direction. The 4 GSV images were augmented based on the proposed method in this study to reflect the real-world spatial relationship and transformed into 2 layers data matrixes with two dimensions. The high-dimensional data from GSV images were then utilized to predict severe crashes. A concatenated deep learning model was proposed that can use both high-dimensional and low-dimensional data as inputs simultaneously. For the development of the crash severity prediction model, crash-related data were collected and utilized in the prediction model as low-dimensional data. A CNN model using the GSV image data only and a NN model using the crash data only were also developed to compare with the proposed deep learning model. The evaluation results show that the proposed model can improve crash severity prediction accuracy. Meanwhile, the results also confirm the benefits of using GSV data for traffic safety research and the assumption that using high-dimensional data (e.g., images) could further boost the prediction performance, as it usually contains more information especially for the roadway environment. Moreover, the modeling results are interpreted and visualized using GRAD-CAM, which highlights the locations that have influenced the most about the classification decisions (i.e., severe, non-severe). This method is expected to provide road safety insights by identifying potential contributing factors for severe crashes. It was revealed that information from the images of the right side could contribute more to the occurrence of severe crashes. Moreover, features identified as trees and roads were found to contribute more to severe crashes. Noteworthy, the computer vision model used to segment GSV images has general object types. It could gain more insights

from the proposed analysis method if the computer vision model could be augmented and improved to cluster detailed object types such as medians, road shoulders, light poles, and other fixed objects.

In summary, this project explored the effects of drivers' visual environment on speeding crashes and the severity of crash severity. The detection results of GSV images could help generate the visual environments for the development of virtual simulation and driving simulator experiments. Besides, the effects could be identified at the pixel level, which could help design the test scenarios of automated vehicles and the V2X technology.

References

1. Sung, J., et al., *Traffic safety facts* 2017, National Highway Traffic Safety Administration.
2. Kim, J., et al., *Age and pedestrian injury severity in motor-vehicle crashes: A heteroskedastic logit analysis.* Accident Analysis & Prevention, 2008. **40**(5): p. 1695-1702.
3. Cai, Q., et al., *Developing a grouped random parameters multivariate spatial model to explore zonal effects for segment and intersection crash modeling.* Analytic methods in accident research, 2018. **19**: p. 1-15.
4. Afghari, A.P., M. Haque, and S. Washington, *Applying fractional split model to examine the effects of roadway geometric and traffic characteristics on speeding behavior.* Traffic injury prevention, 2018. **19**(8): p. 860-866.
5. NHTSA, *Traffic Safety Facts 2018*. 2020.
6. Lee, J., et al., *Effects of emergency medical services times on traffic injury severity: A random effects ordered probit approach.* Traffic injury prevention, 2018. **19**(6): p. 577-581.
7. Edquist, J., C. Rudin-Brown, and M.G. Lenné, *The effects of on-street parking and road environment visual complexity on travel speed and reaction time.* Accident Analysis & Prevention, 2012. **45**: p. 759-765.
8. Atombo, C., et al., *Investigating the motivational factors influencing drivers intentions to unsafe driving behaviours: Speeding and overtaking violations.* Transportation research part F: traffic psychology and behaviour, 2016. **43**: p. 104-121.
9. Marshall, W., N. Coppola, and Y. Golombek, *Urban clear zones, street trees, and road safety.* Research in Transportation Business & Management, 2018. **29**: p. 136-143.
10. Wan, Y., Y. Huang, and B. Buckles, *Camera calibration and vehicle tracking: Highway traffic video analytics.* Transportation Research Part C: Emerging Technologies, 2014. **44**: p. 202-213.

11. Wu, Y., et al., *Automated Safety Diagnosis Based on Unmanned Aerial Vehicle Video and Deep Learning Algorithm.* Transportation Research Record, 2020: p. 0361198120925808.
12. Xie, K., et al., *Mining automatically extracted vehicle trajectory data for proactive safety analytics.* Transportation research part C: emerging technologies, 2019. **106**: p. 61-72.
13. Li, X., et al., *Assessing street-level urban greenery using Google Street View and a modified green view index.* Urban Forestry & Urban Greening, 2015. **14**(3): p. 675-685.
14. Richards, D.R. and P.J. Edwards, *Quantifying street tree regulating ecosystem services using Google Street View.* Ecological Indicators, 2017. **77**: p. 31-40.
15. Li, X., et al., *A novel method for predicting and mapping the occurrence of sun glare using Google Street View.* Transportation research part C: emerging technologies, 2019. **106**: p. 132-144.
16. Campbell, A., A. Both, and Q. Sun, *Detecting and mapping traffic signs from Google Street View images using deep learning and GIS.* Computers, Environment and Urban Systems, 2019. **77**: p. 101350.
17. Balali, V., A. Rad, and M. Golparvar-Fard, *Detection, classification, and mapping of US traffic signs using google street view images for roadway inventory management.* Visualization in Engineering, 2015. **3**(1): p. 1-18.
18. Balali, V., E. Depwe, and M. Golparvar-Fard. *Multi-class traffic sign detection and classification using google street view images*. in *Transportation Research Board 94th Annual Meeting, Transportation Research Board, Washington, DC*. 2015.
19. Yan, W., A. Shaker, and S. Easa, *Potential accuracy of traffic signs' positions extracted from Google Street View.* IEEE Transactions on Intelligent Transportation Systems, 2013. **14**(2): p. 1011-1016.
20. Hankey, S., et al., *Predicting bicycling and walking traffic using street view imagery and destination data.* Transportation research part D: transport and environment, 2021. **90**: p. 102651.

21. Mooney, S., et al., *Use of Google Street View to assess environmental contributions to pedestrian injury.* American journal of public health, 2016. **106**(3): p. 462-469.
22. Tanprasert, T., et al., *Recognizing Traffic Black Spots From Street View Images Using Environment-Aware Image Processing and Neural Network.* IEEE Access, 2020. **8**: p. 121469-121478.
23. Bustos, C., et al., *Explainable, automated urban interventions to improve pedestrian and vehicle safety.* Transportation research part C: emerging technologies, 2021. **125**: p. 103018.
24. Li, Z., et al., *Investigation of driver injury severities in rural single-vehicle crashes under rain conditions using mixed logit and latent class models.* Accident Analysis & Prevention, 2019. **124**: p. 219-229.
25. Schneider, W., P.T. Savolainen, and K. Zimmerman, *Driver injury severity resulting from single-vehicle crashes along horizontal curves on rural two-lane highways.* Transportation Research Record, 2009. **2102**(1): p. 85-92.
26. Wu, Q., et al., *Analysis of driver injury severity in single-vehicle crashes on rural and urban roadways.* Accident Analysis & Prevention, 2016. **94**: p. 35-45.
27. Kim, J., et al., *Driver-injury severity in single-vehicle crashes in California: a mixed logit analysis of heterogeneity due to age and gender.* Accident Analysis & Prevention, 2013. **50**: p. 1073-1081.
28. Xie, Y., K. Zhao, and N. Huynh, *Analysis of driver injury severity in rural single-vehicle crashes.* Accident Analysis & Prevention, 2012. **47**: p. 36-44.
29. Zhou, M. and H. Chin, *Factors affecting the injury severity of out-of-control single-vehicle crashes in Singapore.* Accident Analysis & Prevention, 2019. **124**: p. 104-112.
30. Hou, Q., et al., *Examination of driver injury severity in freeway single-vehicle crashes using a mixed logit model with heterogeneity-in-means.* Physica A: Statistical Mechanics and its Applications, 2019. **531**: p. 121760.

31. Li, Z., et al., *Using support vector machine models for crash injury severity analysis.* Accident Analysis & Prevention, 2012. **45**: p. 478-486.
32. Ahmadi, A., et al., *Crash severity analysis of rear-end crashes in California using statistical and machine learning classification methods.* Journal of Transportation Safety & Security, 2020. **12**(4): p. 522-546.
33. Iranitalab, A. and A. Khattak, *Comparison of four statistical and machine learning methods for crash severity prediction.* Accident Analysis & Prevention, 2017. **108**: p. 27-36.
34. Zeng, Q. and H. Huang, *A stable and optimized neural network model for crash injury severity prediction.* Accident Analysis & Prevention, 2014. **73**: p. 351-358.
35. Yan, X., et al., *Single-vehicle crash severity outcome prediction and determinant extraction using tree-based and other non-parametric models.* Accident Analysis & Prevention, 2021. **153**: p. 106034.
36. Rahim, M. and H. Hassan, *A deep learning based traffic crash severity prediction framework.* Accident Analysis & Prevention, 2021. **154**: p. 106090.
37. Cai, Q., et al., *Applying a deep learning approach for transportation safety planning by using high-resolution transportation and land use data.* Transportation research part A: policy and practice, 2019. **127**: p. 71-85.
38. Cai, Q., et al., *Real-time crash prediction on expressways using deep generative models.* Transportation research part C: emerging technologies, 2020. **117**: p. 102697.
39. Rahman, M., et al., *Applying machine learning approaches to analyze the vulnerable road-users' crashes at statewide traffic analysis zones.* Journal of safety research, 2019. **70**: p. 275-288.
40. Zhang, Z., et al., *A deep learning approach for detecting traffic accidents from social media data.* Transportation research part C: emerging technologies, 2018. **86**: p. 580-596.
41. Zheng, O. and Y. Wu, *OpenCV C++ Implementation of Drone View Car Tracker*. 2019.

42. Yang, J., et al., *Can you see green? Assessing the visibility of urban forests in cities.* Landscape and Urban Planning, 2009. **91**(2): p. 97-104.
43. He, K., et al. *Mask r-cnn*. in *Proceedings of the IEEE international conference on computer vision*. 2017.
44. Syed Sha, Q., O. Grau, and K. Hagn. *DNN Analysis through Synthetic Data Variation*. in *Computer Science in Cars Symposium*. 2020.
45. Yu, S., et al., *Scene-graph augmented data-driven risk assessment of autonomous vehicle decisions.* IEEE Transactions on Intelligent Transportation Systems, 2021.
46. Bhowmik, T., S. Yasmin, and N. Eluru, *A multilevel generalized ordered probit fractional split model for analyzing vehicle speed.* Analytic methods in accident research, 2019. **21**: p. 13-31.
47. Godard, C., et al. *Digging into self-supervised monocular depth estimation*. in *Proceedings of the IEEE international conference on computer vision*. 2019.
48. Chen, S., et al., *3d point cloud processing and learning for autonomous driving: Impacting map creation, localization, and perception.* IEEE Signal Processing Magazine, 2020. **38**(1): p. 68-86.
49. Saxena, A., M. Sun, and A.Y. Ng, *Make3d: Learning 3d scene structure from a single still image.* IEEE transactions on pattern analysis and machine intelligence, 2008. **31**(5): p. 824-840.
50. Dietterich, T.G. *Ensemble methods in machine learning*. in *International workshop on multiple classifier systems*. 2000. Springer.
51. Li, B., L. Peng, and B. Ramadass, *Accurate and efficient processor performance prediction via regression tree based modeling.* Journal of Systems Architecture, 2009. **55**(10-12): p. 457-467.

52. Nitze, I., U. Schulthess, and H. Asche, *Comparison of machine learning algorithms random forest, artificial neural network and support vector machine to maximum likelihood for supervised crop type classification.* Proc. of the 4th GEOBIA, 2012. **35**.
53. Lee, J., M. Abdel-Aty, and Q. Cai, *Intersection crash prediction modeling with macro-level data from various geographic units.* Accident Analysis & Prevention, 2017. **102**: p. 213-226.
54. Ding, C., P. Chen, and J. Jiao, *Non-linear effects of the built environment on automobile-involved pedestrian crash frequency: a machine learning approach.* Accident Analysis & Prevention, 2018. **112**: p. 116-126.
55. Abdel-Aty, M. and K. Haleem, *Analyzing angle crashes at unsignalized intersections using machine learning techniques.* Accident Analysis & Prevention, 2011. **43**(1): p. 461-470.
56. Zhang, X., S. Waller, and P. Jiang, *An ensemble machine learning-based modeling framework for analysis of traffic crash frequency.* Computer-Aided Civil and Infrastructure Engineering, 2020. **35**(3): p. 258-276.
57. Breiman, L., *Random forests.* Machine learning, 2001. **45**(1): p. 5-32.
58. Freund, Y. and R.E. Schapire, *A decision-theoretic generalization of on-line learning and an application to boosting.* Journal of computer and system sciences, 1997. **55**(1): p. 119-139.
59. Friedman, J.H., *Stochastic gradient boosting.* Computational statistics & data analysis, 2002. **38**(4): p. 367-378.
60. Chen, T. and C. Guestrin. *Xgboost: A scalable tree boosting system*. in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016.
61. Lundberg, S. and S. Lee. *A unified approach to interpreting model predictions*. in *Advances in neural information processing systems*. 2017.
62. Parsa, A., et al., *Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis.* Accident Analysis & Prevention, 2020. **136**: p. 105405.

63. Calvi, A., *Does Roadside Vegetation Affect Driving Performance?: Driving Simulator Study on the Effects of Trees on Drivers' Speed and Lateral Position.* Transportation Research Record, 2015. **2518**(1): p. 1-8.
64. Lee, S., Y. Kim, and Y. Ji, *Effects of visual complexity of in-vehicle information display: Age-related differences in visual search task in the driving context.* Applied ergonomics, 2019. **81**: p. 102888.
65. Noland, R. and L. Oh, *The effect of infrastructure and demographic change on traffic-related fatalities and crashes: a case study of Illinois county-level data.* Accident Analysis & Prevention, 2004. **36**(4): p. 525-532.
66. Bao, J., P. Liu, and S. Ukkusuri, *A spatiotemporal deep learning approach for citywide short-term crash risk prediction with multi-source data.* Accident Analysis & Prevention, 2019. **122**: p. 239-254.
67. Huang, T., S. Wang, and A. Sharma, *Highway crash detection and risk estimation using deep learning.* Accident Analysis & Prevention, 2020. **135**: p. 105392.
68. Selvaraju, R., et al. *Grad-cam: Visual explanations from deep networks via gradient-based localization*. in *Proceedings of the IEEE international conference on computer vision*. 2017.
69. Fitzpatrick, C., *The effect of roadside elements on driver behavior and run-off-the-road crash severity*. 2013, University of Massachusetts Amherst.