Applying Machine Learning and Google Street View to Explore Effects of Drivers’ Visual Environment on Traffic Safety

Qing Cai
Department of Civil, Environmental and Construction Engineering, University of Central Florida, Orlando, Florida 32816, USA
Email: qingcai@knights.ucf.edu

Mohamed Abdel-Aty
Department of Civil, Environmental and Construction Engineering, University of Central Florida, Orlando, Florida 32816, USA
Email: M.Aty@ucf.edu

Ou Zheng*
Department of Civil, Environmental and Construction Engineering, University of Central Florida, Orlando, Florida 32816, USA
Email: ouzheng1993@knights.ucf.edu

Yina Wu
Department of Civil, Environmental and Construction Engineering, University of Central Florida, Orlando, Florida 32816, USA
Email: jessicawyn@knights.ucf.edu

* Corresponding Author
Abstract

This study aims to explore the effects of drivers’ visual environment on speeding crashes by using different machine learning techniques. 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. This study provided new insights to obtain the detailed information from GSV images for traffic safety analysis. The findings based on the explainable machine learning could also provide road planners and engineers clear suggestions to select appropriate countermeasures to enhance traffic safety.

Keyword: drivers’ visual environment, google street view, coordinate transformation, speeding crashes, deep learning, explainable machine learning, computer vision
1. Introduction

Speeding is one of the major factors impacting 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 (Heidi and Hrista, 2018). On urban arterials, the speed limit violation could significantly increase the severity levels of pedestrian and bicycle crashes (Kim et al., 2008). 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, weather, etc. For example, Cai et al. (2018) developed grouped random parameter models to examine the crash occurrence on segments and intersections considering the roadway attributes such as speed limit and the zonal level effects. Afghari et al. (2018) 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.

Recently, several studies have focused on the effects of the driving environment on drivers’ behavior and safety. For example, Edquist et al. (2012) 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 et al. (2016) revealed the significant effects of the driving environment on speeding and overtaking violations. Marshall et al. (2018) developed statistical models to study the effects of trees on crash frequency in the urban area. The study indicated that tree density could reduce crashes. However, 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 the drivers’ view.

Within the rapid 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 studies, computer vision has been applied to count traffic volume and detect traffic speed (Wan et al., 2014). Besides, some studies applied detection and tracking algorithms to get vehicles’ trajectory and calculate the surrogate safety measures (Wu et al., 2020; Xie et al., 2019). 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. In the recent years, Google Street View (GSV) images have been used to analyze the relationship between the environment and traffic safety. For example, Mooney et al. (2016) used GSV images to assess environmental contributions to the frequency of pedestrian crashes. It was found that traffic islands, visual advertising, bus stops, and crosswalk infrastructures are significantly associated with the counts of pedestrian crashes. Kita and Kidziński (2019) manually labeled data about the conditions of the house and neighborhood from GSV images and developed a Generalized Linear Model (GLM) model to reveal the correlations between these factors and the risk of that house’s residents getting involved in a car accident. Recently, machine learning techniques were also used to explore traffic safety based on GSV images. Li et al. (2019) developed deep learning algorithms to estimate and map the
occurrence of sun glare for drivers using GSV images. The study also estimated the time windows of sun glare by calculating the sun positions and the relative angles between drivers and the sun for different locations. Tanprasert et al. (2020) developed a distance-aware pixel accumulation to extract information about objects surrounding the spots on roads in the street view images. The extracted characteristics were used to train fully connected neural networks to identify black spots on roads. While the machine learning methods could reach a high accuracy in traffic safety analysis, the relation between the characteristics related to drivers’ view extracted from GSV images and safety is unclear. Meanwhile, to the best of the authors’ knowledge, several factors such as tree density and driving environment complexity which reflect drivers’ visual environment have not been explored by using GSV images. Hence, the applications of using GSV images to enhance traffic safety might be limited.

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 the GSV images. Computer vision technology has been applied to process GSV images automatically instead of manually. Based on the computer vision technology, different information such as street-level morphology, urban feature composition, and urban greenery could be extracted from the GSV images (Gong et al., 2018; Li et al., 2015; Middel et al., 2019; Richards and Edwards, 2017). Li et al. (2015) assessed the street-level urban greenery with the field of view (fov) as 60 degrees by using GSV images. The Red, Green, Blue (RGB) bands were detected from the images, and the difference among different bands was calculated to determine the area of green vegetation. Gong et al. (2018) used the PSPNet model, which is a deep Convolutional Neural Network (CNN) model for the semantic segmentation of GSV images. Several view factors, including sky, tree, and building view factors of street canyons, were quantified by using the photographic method. Middel et al. (2019) derive street-level morphology and urban feature composition as experienced by a pedestrian from GSV images. This study used the Caffe deep learning framework to segment GSV images into six classes: sky, trees, buildings, impervious surfaces, pervious surfaces, and non-permanent objects. While the previous studies could extract accurate information at each pixel from images by using computer vision methods, these studies focused on the feature segmentation in GSV images without considering depth information of features. In a GSV image, features should have different distances away from the camera, reflecting objects at different locations drivers pass by along the road. Hence, depth information should be considered to reflect drivers’ visual environment when driving along roads in the real world.

This study contributes by proposing a novel method to obtain drivers’ visual environment from GSV images and explore the effects of the visual environment on speeding crashes. To this end, deep learning models were applied to obtain the cluster and depth information from GSV images. The coordinate transformation was conducted to quantify drivers’ visual environment in the real world. Then, the effects of the visual environment on speeding crashes were explored by developing explainable machine learning models. This paper is organized as follows: the method to process GSV images and obtain the indexes for the visual environment is introduced in Section 2. The method about exploring the effects of the visual environment was also included in this section. Section 3 describes the data used for the analysis including the data related to the visual
environment. Section 4 presents the analysis results and discussions. Finally, the last section concludes the findings of this paper.

2. Methodology

The machine learning models based on neural networks and trees are two most popular models in use today (Cai et al., 2020; Rahman et al., 2019). The neural-network-based deep learning models are more appropriate in fields like image recognition, speed recognition, and natural language processing (Wu et al., 2020; Zhang et al., 2018; Zheng and Wu, 2019). 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 Machine learning to process GSV images

2.1.1 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. Users can request a static GSV image in customized direction and angle for the locations where GSV is available. An example of requesting a GSV static image is shown below:

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

Figure 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° and 60° (Yang et al., 2009). Li et al. (2015) used fov of 60° to collect GSVs, which was 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.
2.1.2 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 (He et al., 2017), 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 (Wu et al., 2019). 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 (Syed et al., 2020; Yu et al., 2020). For example, Syed et al. (2020) 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. (2020) developed models to classify risky driving scenes based on the Detecton2 framework, which could reach 96.4% classification accuracy. As shown in Figure 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:

\[
\text{complexity level} = \frac{-\sum_k (p_k \ln (p_k))}{\ln N}
\]

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 (Bhowmik et al., 2019).
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. (2019) 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(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 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: \( O_{\text{camera}} = [R|t] \ast O_{\text{world}} \).

\[ [R|t] \] is a 4 \times 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}, \text{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.\]
As shown in Figure 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:

\[
O_{image} = K \cdot O_{camera}
\]

\( K \) is a \( 3 \times 3 \) 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:

\[
f = (W/2)/\tan (\alpha/2)
\]

(2)

where \( \alpha \) is the horizontal field of view and \( W \) is horizontal number of pixels of the image. In Figure 4(b), the same focal length is applied for the vertical field of view (\( \beta \)) and vertical number of pixels (\( H \)) of the image.

Finally, the point could be projected to the pixel coordinate system. As shown in Figure 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 \[
\begin{bmatrix}
 f & 0 & u_p \\
 0 & f & v_p \\
 0 & 0 & 1
\end{bmatrix}
\] and the projection from the world coordinate system to the pixel coordinate system could be expressed as:

\[
\begin{bmatrix}
 u \\
 v \\
 1
\end{bmatrix} = \begin{bmatrix}
 f & 0 & u_p \\
 0 & f & v_p \\
 0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
 X \\
 Y \\
 Z \\
 1
\end{bmatrix}
\]

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

\[
X = (u - u_0) \ast \frac{\text{depth}(v, u)}{f} \\
Y = -(v - v_0) \ast \frac{\text{depth}(v, u)}{f} \\
Z = \text{depth}(v, u)
\]

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.

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 6(d)), which is consistent with the previous study (Chen et al., 2020; Saxena et al., 2007). As highlighted in Areas 1 and 2 in Figure 5(b) and 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:

\[
\text{accuracy} = \text{mean}(100 \ast \frac{|w_{obs} - w_{est}|}{w_{obs}})
\]
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 6 Illustration of satellite image view (a: original Google street view; b: semantic segmentation; c: depth estimation; d: projection of satellite image view)](image)
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 8, the proportion of trees 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 8 Illustration of tree canopy and road length with trees

Figure 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.
2.2 Machine learning for crash analysis

2.2.1 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 (Dietterich, 2000; Li et al., 2009; Nitze et al., 2012):
• 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.

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 (Zhang et al., 2020). 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.

(1) Random forest
The random forest (RF) was developed by Breiman (2001) 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.

(2) AdaBoost
The adaptive boosting (AdaBoost) was first introduced by Freund and Schapire (1997). 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.

(3) XGBoost
The gradient boosting framework introduced by Friedman (2002). 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 (2016), 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.
2.2.2 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 (2017), 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 is 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 (Parsa et al., 2020). Given a feature value $i$, the SHAP value could be obtained by:

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

where $|F|$ is the total number of features, $S$ represents any subset of features that doesn’t include the $i^{th}$ 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.

3. 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 10-12 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) (Florida Department of Transportation, 2020) 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 1 summarizes the collected variables.
Figure 10 Proportion of trees in the drivers’ view on urban arterials
Figure 11 Proportion of buildings in the drivers’ view on urban arterials
Figure 12 Proportion of roadway length with trees in the drivers’ view on urban arterials
<table>
<thead>
<tr>
<th>Variable</th>
<th>mean</th>
<th>S.D.</th>
<th>min.</th>
<th>max.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speeding crashes</td>
<td>1.24</td>
<td>1.51</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td><strong>Traffic variables</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vehicle Miles Travelled (VMT)</td>
<td>14.505</td>
<td>8.920</td>
<td>2.145</td>
<td>40.523</td>
</tr>
<tr>
<td>Proportion of truck traffic</td>
<td>6.76</td>
<td>3.31</td>
<td>1.77</td>
<td>16.19</td>
</tr>
<tr>
<td>Average daily transit frequency</td>
<td>19.97</td>
<td>18.05</td>
<td>0</td>
<td>72</td>
</tr>
<tr>
<td>Proportion of INRIX speed data over the speed limit</td>
<td>0.18</td>
<td>0.14</td>
<td>0</td>
<td>0.78</td>
</tr>
<tr>
<td><strong>Drivers' visual environment data from GSV images</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proportion of tree view</td>
<td>0.13</td>
<td>0.06</td>
<td>0.03</td>
<td>0.31</td>
</tr>
<tr>
<td>Proportion of building view</td>
<td>0.03</td>
<td>0.03</td>
<td>0</td>
<td>0.22</td>
</tr>
<tr>
<td>Complexity level of driving view</td>
<td>0.73</td>
<td>0.03</td>
<td>0.65</td>
<td>0.22</td>
</tr>
<tr>
<td>Proportion of road length with trees</td>
<td>0.28</td>
<td>0.06</td>
<td>0.15</td>
<td>0.48</td>
</tr>
<tr>
<td><strong>Roadway variables</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variables related to FDM speed management strategies</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Indicator of asphalt pavement (1: yes; 0: no)</td>
<td>0.84</td>
<td>0.36</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Indicator of narrow lane (lane width&lt;12 feet) (1: yes; 0: no)</td>
<td>0.36</td>
<td>0.48</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Average block length (mile)</td>
<td>1.60</td>
<td>3.26</td>
<td>0.06</td>
<td>10.33</td>
</tr>
<tr>
<td>Existence of median island on pedestrian crossing (1: yes; 0: no)</td>
<td>0.07</td>
<td>0.25</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Log (number of parking spot per mile)</td>
<td>0.27</td>
<td>1.12</td>
<td>0</td>
<td>6</td>
</tr>
<tr>
<td>Presence of road diet (1: yes; 0: no)</td>
<td>0.44</td>
<td>0.5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Log (length of two-way-left-turn lane)</td>
<td>0.13</td>
<td>0.23</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Other roadway variables</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of lanes</td>
<td>2.10</td>
<td>0.52</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Speed limit (mph)</td>
<td>38.59</td>
<td>5.2</td>
<td>25</td>
<td>55</td>
</tr>
<tr>
<td>Pavement condition</td>
<td>4.17</td>
<td>0.8</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>Raised median (1: yes; 0: no)</td>
<td>0.44</td>
<td>0.5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Median Width (feet)</td>
<td>16.53</td>
<td>9.32</td>
<td>0</td>
<td>55.07</td>
</tr>
<tr>
<td>Curb, gutter inside shoulder type (1: yes; 0: no)</td>
<td>0.34</td>
<td>0.47</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Width of inside shoulder (feet)</td>
<td>0.97</td>
<td>1.47</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>Curb, gutter outside shoulder type (1: yes; 0: no)</td>
<td>0.48</td>
<td>0.5</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Width of outside shoulder (feet)</td>
<td>3.45</td>
<td>1.88</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Proportion of sidewalk length</td>
<td>0.91</td>
<td>0.25</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Sidewalk width (feet)</td>
<td>5.09</td>
<td>1.35</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Proportion of bike lane length</td>
<td>0.12</td>
<td>0.3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Proportion of bike slot length</td>
<td>0.01</td>
<td>0.05</td>
<td>0</td>
<td>0.64</td>
</tr>
<tr>
<td>Number of signalized intersections per mile</td>
<td>3.11</td>
<td>3.15</td>
<td>0</td>
<td>16.98</td>
</tr>
<tr>
<td>Number of access per mile</td>
<td>9.52</td>
<td>5.86</td>
<td>0</td>
<td>28.37</td>
</tr>
<tr>
<td><strong>Land use and socio-demographic variables</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proportion of residential land use</td>
<td>0.29</td>
<td>0.38</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Proportion of commercial land use</td>
<td>0.04</td>
<td>0.09</td>
<td>0</td>
<td>0.62</td>
</tr>
<tr>
<td>Land use mix</td>
<td>0.04</td>
<td>0.15</td>
<td>0</td>
<td>0.83</td>
</tr>
<tr>
<td>Proportion of population below poverty</td>
<td>0.06</td>
<td>0.11</td>
<td>0</td>
<td>0.73</td>
</tr>
<tr>
<td>Proportion of zero-vehicle household</td>
<td>0.02</td>
<td>0.04</td>
<td>0</td>
<td>0.22</td>
</tr>
<tr>
<td>Proportion of commuters by walking or biking</td>
<td>0.02</td>
<td>0.04</td>
<td>0</td>
<td>0.20</td>
</tr>
</tbody>
</table>
4. Results and discussion

4.1 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 (Cai et al., 2019; Zhang et al., 2020). 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 (Pereek, 2020). The cross-validation training results and testing results of the three models are summarized in Table 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>
<thead>
<tr>
<th>Model</th>
<th>Maximum depth</th>
<th>Number of trees</th>
<th>MAE</th>
<th>RMSE</th>
<th>$R^2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Training</td>
<td>Testing</td>
<td>Training</td>
</tr>
<tr>
<td>Random Forest (RF)</td>
<td>6</td>
<td>50</td>
<td>0.8</td>
<td>0.87</td>
<td>1.12</td>
</tr>
<tr>
<td>Adaptive Boosting (AdaBoost)</td>
<td>-</td>
<td>140</td>
<td>0.77</td>
<td>0.81</td>
<td>1.01</td>
</tr>
<tr>
<td>eXtreme Gradient Boosting (XGBoost)</td>
<td>6</td>
<td>260</td>
<td>0.41</td>
<td>0.42</td>
<td>0.68</td>
</tr>
</tbody>
</table>

4.2 Effects analysis

Figure 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 (Cai et al., 2018; Lee et al., 2017).

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 13. Hence, the scatter plot of variables related to drivers’ visual environment vs SHAP values are presented in Figure 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 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. (2018). 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 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 (Calvi, 2015). Besides, it was found that the crashes could decrease with the increase in the tree density (Marshall et al., 2018). With the effects of trees on speeding and crash occurrence, it is reasonable to find a negative effect of trees on speeding crashes. Figure 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 resitively important effect on speeding crashes. Both Figure 13 and Figure 14(d) clearly show the effect of complexity level. With a linear relation revealed in Figure 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 (Lee et al., 2019).

The effects of several other features were also revealed in Figure 14. First, it is found that the intersection density could result in more crashes as intersections could increase traffic interaction (Cai et al., 2018). Second, it is expected that less speeding crashes could be found on roads with higher speed limit. In addition, Figure 14 shows that the wide outside shoulder could reduce the crashes, which is in line with the previous study (Noland and Oh, 2004). 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.
Figure 13 Global feature importance and Summary of SHAP value
25

(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 14 Scatter plot of SHAP values vs variables related to drivers’ visual environment

5. 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, the significant streetscape factors for speeding crashes were also revealed, such as speed limit, outside shoulder width, and intersection density. With the decrease of the speed limit and outside shoulder width, the speeding crashes could increase. In addition, the increase of intersection density could lead to more speeding crashes. Appropriate countermeasures could be proposed by considering both street view variables and the streetscape factors. For example, more trees could be added along the road with the low speed limit and narrow outer shoulder to reduce the speeding crashes. If the intersection density is high, it is recommended to reduce the visual complexity level by removing unnecessary signs and billboards along roads.

This paper contributed to proposing a new method to obtain valuable information from GSV images by using deep learning algorithms. The information from drivers’ visual view was obtained and used to analyze speeding crashes. Besides, the explainable machine learning method was used to analyze the effects of different features on speeding crashes to ensure a good balance of accuracy and interpretability. The modeling results confirmed the importance of information obtained from GSV images. In this paper, aggregated information from GSV was used for the speeding crash analysis. It would be interesting to use pixel-level data for traffic safety and driving environment analysis. It should be also noted that this study constructed the drivers’ visual environment in the real world by the depth detection of features in GSV images and the coordinate transformation. The accuracy of transformed locations is affected by the distance of objects from the camera and temporary objects (e.g., cars) on the road. Also, the semantic segmentation could also introduce errors in the detection results. The accuracy of detection results could influence the crash prediction results and identified effects of drivers’ visual environment. Given the fact that the GSV images are collected continuously along the road, Simultaneous Localization And Mapping (SLAM) could be used (Bresson et al., 2017) to improve the accuracy by capturing the connectivity of the environment in GSV images at different locations along the road. Further, this study conducted the semantic segmentation and 3D coordinate transformation separately. It is expected the accuracy of feature detection could be improved by conducting 3D semantic segmentation based on the cloud points after the transformation.

Acknowledgement

The research discussed in this paper was funded by the Florida Department of Transportation (FDOT). The authors also acknowledge INRIX for providing the speed data. This paper and its contents, including conclusions and results, are solely those of the authors; they do not represent opinions or policies of the funding agency.
AUTHOR CONTRIBUTIONS

The authors confirm contributions to the paper as follows: model development: Qing Cai, Mohamed Abdel-Aty, Ou Zheng; data collection and processing: Ou Zheng, Qing Cai; analysis and result interpretation: Qing Cai, Yina Wu; draft manuscript: Qing Cai, Mohamed Abdel-Aty, Yina Wu. All authors have reviewed the results and approved the final version of the manuscript.

Reference


