Fall 2019

# Computer Vision-Based Traffic Sign Detection and Extraction: A Hybrid Approach Using GIS And Machine Learning

Zihao Wu

COMPUTER VISION-BASED TRAFFIC SIGN DETECTION AND EXTRACTION: A HYBRID APPROACH USING GIS AND MACHINE LEARNING

by

ZIHAO WU

(Under the Direction of Xiaolu Zhou)

## ABSTRACT

Traffic sign detection and positioning have drawn considerable attention because of the recent development of autonomous driving and intelligent transportation systems. In order to detect and pinpoint traffic signs accurately, this research proposes two methods. In the first method, geo-tagged Google Street View images and road networks were utilized to locate traffic signs. In the second method, both traffic sign categories and locations were identified and extracted from the location-based GoPro video. TensorFlow is the machine learning framework used to implement these two methods. To that end, 363 stop signs were detected and mapped accurately using the first method (Google Street View image-based approach). Then 32 traffic signs were recognized and pinpointed using the second method (GoPro video-based approach) for better location accuracy, within 10 meters. The average distance from the observation points to the 32 ground truth references was 7.78 meters. The advantages of these methods were discussed. GoPro video-based approach has higher location accuracy, while Google Street View image-based approach is more accessible in most major cities around the world. The proposed traffic sign detection workflow can thus extract and locate traffic signs in other cities. For further consideration and development of this research, IMU (Inertial Measurement Unit) and SLAM (Simultaneous Localization and Mapping) methods could be integrated to incorporate more data and improve location prediction accuracy.

INDEX WORDS: Geographic Information Systems, Hybrid traffic sign detection approach, Computer vision, Machine learning, TensorFlow, Google Street View, GoPro, GPS

# COMPUTER VISION-BASED TRAFFIC SIGN DETECTION AND EXTRACTION: A HYBRID APPROACH USING GIS AND MACHINE LEARNING

by

ZIHAO WU

B.S., Southwest University, China, 2015

A Thesis Submitted to the Graduate Faculty of Georgia Southern University

in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

STATESBORO, GEORGIA

COMPUTER VISION-BASED TRAFFIC SIGN DETECTION AND EXTRACTION: A HYBRID APPROACH USING GIS AND MACHINE LEARNING

by

ZIHAO WU

| | |
|---|---|
| Major Professor: | Xiaolu Zhou |
| Committee: | Wei Tu |
| | Kai Wang |

Electronic Version Approved:
December 2019

# ACKNOWLEDGMENTS

Thanks to my advisor Dr. Zhou's devotion in the guidance for the past two years. Thanks to Dr. Tu's care and comments. Thanks to Dr. Kai Wang's support for my thesis and defense. Thanks to Dr. Welford's suggestion about the literature review. Thanks to Dr. Yarbrough's seminar.

TABLE OF CONTENTS

# LIST OF TABLES

Page

LIST OF FIGURES

# CHAPTER 1

# INTRODUCTION

*Purpose of the Study*

Traffic signs are designed to regulate traffic flow safely by providing information to both drivers and pedestrians (Gudigar et al. 2016). Traffic signs deliver fundamental instruction on the streets by giving rich road and traffic information. So, detecting traffic signs will help people understand their surroundings better while driving on and walking along these streets. According to police accident reports (Borowsky et al. 2008), failure to obey traffic signs is one of the major causes of road accidents. Thorough and explicit traffic signs play a crucial role in daily road uses, as they can reduce vehicle accidents and pedestrian accidents. Traffic sign detection is also one of the critical areas of concern, given the rise in autonomous driving. Thus, traffic sign detection and management are necessary, indeed significant so, to improve both traffic safety and efficiency (Taylor et al. 2000). Traffic sign detection has been explored by researchers in the Intelligent Transportation System (ITS) in the past few years.

The typical Automatic Traffic Sign Detection and Extraction (ATSDE) system includes components for detection, recognition, and positioning of cars based on computer vision methodologies (Miura et al. 2000) like SIFT (Scale-Invariant Feature Transform), SURF (Speeded Up Robust Feature), and ORB (Oriented FAST and Rotated BRIEF) (Rublee et al. 2011). According to Miura's study (Miura et al. 2000), the subjects and patterns of traffic signs can be found in massive street-view datasets, such as publicly available Google Street View images. These can be processed and analyzed to obtain the geolocation of traffic signs. An increasing number of studies in the transportation area is dealing with street view images, according to Zamir and Shah's study (2010). Street view images allow researchers from different fields (urban planning, GIS, computer vision, and transportation) to capture and collect traffic sign information at street level from a global scale (Anguelov et al. 2010) with easy accessibility. However, traditional manual identification of traffic signs based on these datasets is not feasible due to the extensiveness and variability of these street view images. Automatically establishing and maintaining a traffic sign inventory automatically has thus become an essential task to utilize the existing datasets better and improve the safety and efficiency of the entire transportation system.

With the development of computer vision algorithms and the improvement of both computational and data resources, traffic sign detection has been further explored and developed using the traffic sign database with moving vehicles and cameras over the past few decades (Scott et al. 2011). Gudigar et al. (2016) proposed a traffic sign detection and classification system based on a three-step algorithm, which included color segmentation (Benallal and Meunier 2003), shape recognition (Xu 2009), and a neural network for image recognition from photos (Broggi et al. 2007). By using these algorithms, it is possible to extract useful information from provided street view images. The features compiled from all these images provide road conditions and traffic information. Greenhalgh and Mirmehdi (2012) proposed a traffic sign detection system. This system provides for having maximum stable likelihood regions by offering robustness even with different lighting conditions. The image recognition method used in Greenhalgh and Mirmehdi's study was based on support vector machine (SVM) classifiers, which were refined using the histogram of oriented gradient (HOG) features. Maldonado-Bascon et al. (2007) then developed another automatic road-sign detection and recognition system based on the support vector machine.

However, it remains a challenging task to extract accurate location information from a vast amount of traffic sign images. Most traffic signs need to be automatically digitized with their geospatial related attributes noted (Ford et al.2001). Due to the lack of geolocational attributes, it is time-consuming to coordinate the information and pinpoint traffic signs using traditional labor-intensive tagging processes. By analyzing the structured and unstructured data, Stein et al.'s studies have attempted to extract knowledge about traffic sign categories, road conditions, and traffic sign distributions. Detecting and recognizing traffic sign systems can also be done by mounting a camera on a moving vehicle (Stein et al. 2011). Stein et al.'s work contributed to the intelligent transportation system and auto driving systems. However, these tedious labeling and locating tasks also require a tremendous amount of labor to keep the traffic sign information up to date (Findley et al. 2011). The ongoing fieldwork to locate traffic signs along streets manually also causes safety concerns. Traffic signs recognition speed is slow when using only the traditional image recognition methods, such as Support Vector Machine (SVM), Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), and the Speeded Up Robust Feature (SURF). So, traffic sign collection is more meaningful than having image pixels. Actual traffic sign images come with additional useful information such as location. The traffic sign location information missing challenge is due to the capability and effectiveness of traffic sign detection and location extraction.

Further still, most traffic recognition methods and models are difficult to apply to a broader geographic area, because these models were trained in another particular location, the background and traffic sign content may vary from different regions. It means old existing methods cannot be applied in a different geographic context. Furthermore, using automatic detection to build and maintain traffic sign inventory has not been well illustrated in previous studies, especially those big geospatial data research (Lee and Kang 2015).

In order to address these limitations, this study designed a prototype system for processing a collection of Google Street View images to extract traffic signs. Given the fast development of machine learning techniques and the rapidly growing volume of data, traffic sign extraction and positioning can be accomplished using automated image recognition technology (Balali et al. 2013). According to the issues noted above, this study developed an economical and effective solution for traffic sign detection, positioning, and mapping with high accuracy. The first objective of this research is to detect traffic signs by analyzing the spatial features of images. The knowledge generated by an object detection system can indicate traffic sign contents, show the categories and locations of these traffic signs from the same street view content. The second objective of this research is to automatically extract geospatial information with computer vision using artificial intelligence techniques on a TensorFlow (a machine learning platform). With CUDA (Compute

Unified Device Architecture) parallel computing to accelerate the training and validating the process, such a traffic sign recognition model can achieve high confidence in the testing performance. Results from this research can contribute to both viable and affordable autonomous vehicle delivery systems.

# CHAPTER 2

# RELATED RESEARCH

## *Traffic asset management*

Traffic asset management is defined as a systematic process of maintaining, updating, and rehabilitating traffic assets (roads, bridges, and traffic signs) cost-effectively (McNeil et al. 2000). Traffic signs are managed using several approaches based on the known traffic asset inventory research (Balali and Golparvar-Fard 2015). A traffic inventory system is a valuable solution that has been used in traffic asset management in the past few decades (Vanier 2001). One of the main tasks of traffic asset management is the extraction and maintenance of traffic signs across various assigned categories. The general approach (Balali and Golparvar-Fard 2015) of a traffic sign management system is to use knowledge-based models (Fuchs et al. 2008) to store and update the gathered traffic sign inventory information (Maldonado-Bascon et al. 2008) in a database.

Many state highway agencies in the United States have been trying to develop traffic sign inventories in digital form (Mogelmose 2012). It is thus expected that this kind of project will help to serve and become the basis for evaluating time, labor, and equipment requirements in future sign inventory programs (Eastman 2018). Because of the complexity of the transportation infrastructure, traditional transportation infrastructure management has only focused on manually collecting traffic assets, causing both high labor costs and potential safety issues (Djahel 2014). Regular practices mainly involve tedious manual data collection and analysis (Balali and Golparvar-Fard 2015). For example, Currin's book, *Introduction to traffic engineering: a manual for data collection and analysis* (Currin 2012) introduced a procedure to collect data of roadway and intersections. Wherein multiple observations and human activities are engaged in collecting and recording traffic signs and assets for further traffic data processing. Apart from the costly data collection process, regular road asset monitoring and maintenance can also be expensive (Šelih et al. 2008).

To manage and maintain the regular operation road system, departments of transportation need reliable and up-to-date information about the location and condition of road traffic signs (He et al. 2017). Updating traffic sign information during road asset management can be time-consuming (Murphy 2012) as traffic inventory collection involves complicated and repetitive work that requires a lot of personnel and resources (He et al. 2017). In conclusion, because of the limitation of time and budget, along with safety considerations in manual data collection, transportation agencies need a more efficient way to extract and maintain traffic signs.

*Location-based sign detection*

Researchers from different fields have developed several methods or management systems to realize traffic sign extraction (Halfawy 2008). The premise of traffic sign extraction is to search for and detect traffic signs. Some researchers introduced traffic sign inventory systems based on stereo vision and tracking (Wang et al. 2010). Wang's system used multiple sensors of high-resolution cameras to capture Right of Way (ROW) images. The stereo vision technique was employed to realize real-time data acquisition and analysis on vehicles. Wang et al.'s research (2010) used a computer vision technique to achieve an automated traffic sign inventory system while driving the vehicles. However, no coordinates could be extracted in this way, which caused difficulty in pinpointing the actual traffic signs on the map.

Other traffic sign extraction management systems were aiming to acquire spatial information along with traffic sign content. Ford et al.'s research (2006) used a mobile device to capture field data, such as tracking traffic assets and transferred location data into a GIS database assisted by the built-in global positioning systems (GPS) module. It is a good way to acquiring spatial information of traffic assets with GPS information and then convert it to GIS data. They provided a direction for utilizing GPS information for positioning and locating traffic assets. Comparing with a traditional system for managing transportation assets (Sroub and Mackraz 2003), a better solution to acquire geo-tagged traffic assets is to engage image recognition with GPS information.

A GPS driven platform (Ma and Wang 2014) was utilized to consistently acquire an available coordinate reference to collect essential geographical information. Tucker et al. (2009) provided an ideal way to gather traffic asset images with geo-tag by using a vehicle-based image recognition system with accurate coordinates. Wang's methods (2014) and Tucker' s proposed systems (2009) both serve as a prototype, that is similar to the Google Street View vehicle. A Google Street View vehicle has more sensors and stronger functions to use to detect and gather information along all the visited streets. Their solutions overlapped with the Google Street View vehicle solution. However, these solutions are expensive. Also, although they proposed a method for data collection, they have missed offering an efficient way for data processing.

*Traffic sign recognition and machine learning*

Methods have been developed to detect sign recognition, including color segmentation (Crisman and Thorpe 1991), and neural network (Pomerleau 1990). There are serval ways (Chen et al. 2011) to recognize a traffic sign by using feature matching (Ren et al. 2009). Ren proposed a conventional approach to implement the entire recognition process by utilizing feature matching methods (e.g., SIFT or SURF features), wherein the RGB color input images were converted into HSV color space (Ren et al. 2009). These methods were using transformation to detect unique shapes as potential signs, which could be compared to existing reference signs by using feature matching methods. It is classical to recognize traffic signs with traditional image recognition methods. However, due to conventional image recognition hindered by the computation capacity, only a small size image dataset can be processed in a short time. Therefore, the challenge of the fast process on a large dataset remains. Indeed, there is a need to improve image detection efficiency and accuracy with a new methodology. Besides, these traditional traffic sign recognition methods cannot be applied to different geographical contexts and locations. Given such further illustration, sign detection has been a less-studied field during the contemporary period.

Machine learning was defined as a set of methods that can automatically detect patterns in data, and this uncovered pattern can predict future data (Murphy 2012). It provides a solution for a fast process on a large dataset. Recent studies leverage data from multiple sources to strengthen both image detection and image recognition using machine learning. Houben et al. (2013) utilized vision-based vehicles to realize road detection, obstacle detection, and sign recognition. Other researchers also have utilized Convolutional Neural Networks (CNN), a class of deep neural networks in machine learning, to recognize and classify traffic signs. Pierre & Yann (Sermanet and LeCun 2011) applied CNN to learn features at every level and achieved a final accuracy of 98%. With an increasing training network, a new record of 99% accuracy was reached. Besides, Abdi and Meddeb (2017) used deep CNN to realize traffic sign detection, recognition, and augmentation. Their classifications were using Region of Interest (ROI) with linear SVM. They tested the real-time performance on the German Traffic Sign Recognition Benchmark (GTSRB) dataset; both recall and precision were higher than 98.8% in seven different category traffic signs. Other models were also applied in this multi-class classification competition (Stallkamp et al. 2011), such as the Committee of CNN and MLP, IK-SVM, LDA, and 3-NN. Their accuracy ranges from 73.89% to 98.98%, while human performance is 98.81%. It is noticeable that there are a few methods that can outperform humans in recognition accuracy.

Further exploration has approved that machine learning algorithms for traffic sign recognition can also attain the same level of human performance (Stallkamp et al. 2012). Several popular machine learning methods are briefly illustrated here. Multi-column Deep Neural Networks (DNN) for image classification is a fit solution (Ciregan et al. 2012) to deal with handwritten numbers or traffic signs. They focus on combing several deep neural network (Fukushima and Miyake 1982) columns into a Multi-column DNN (MCDNN). In this way, the error rate decreased by 30-40%, and thus their method improved the traffic sign recognition accuracy significantly. Kiran C. G et al.

(2009) used the support vector machine (SVM) to deal with traffic sign detection and pattern recognition. In their research, a linear SVM was applied to improve the performance of segmentation. At the same time, a multi-classifier non-linear support vector machine with edge-related pixels of interest was used for determining traffic sign shape detection and pattern recognition. Their pattern recognition results (Kiran et al. 2009) showed higher accuracy than other research. Shustanov and Yakimov (2017) used CNN to recognize traffic signs in real-time. Yakimov (2015) developed an algorithm for detecting and predicting road traffic signs with vehicle velocity. However, these solutions described above only deal with image recognition and detection without extracting the spatial attributes of the collected traffic signs. In some cities, traffic signs are required to be automatically detected for location information. So, an automatic workflow is also needed for managing, identifying, and positioning the traffic signs on a digital map.

*The hybrid method with an innovative solution*

Fortunately, some issues have been resolved using the traffic sign detection methods mentioned above, such as traffic sign recognition and parallel computing for image processing. However, a more efficient way of detecting and positioning traffic signs is still missing. Besides, extraction with high location accuracy is needed. The low cost of the whole process is also required. To address the remaining gaps, this paper offers a timely and valuable method that leverages the emerging advanced technologies for collecting, detecting, and extracting traffic signs using a hybrid approach. This research used machine learning-based approaches to detect and pinpoint traffic signs from both images and videos. Using these two methods with both Google Street View images and GoPro videos as input resources to recognize traffic signs programmatically, this workflow has much higher accuracy and efficiency in different scenarios. The study features a combination of computer vision with machine learning (i.e., traffic sign detection through image and video procession), geo-localization (i.e., gather geo-tagged photos and videos), and fast processing (i.e., CUDA parallel computing to accelerate the entire process). The whole workflow can extract traffic sign information along streets and also could monitor the shift of road traffic sign database in a period.

# CHAPTER 3

# METHODOLOGY

*A machine learning model principle overview*

Machine learning-based techniques have achieved state-of-the-art performance on traffic sign recognition and classification tasks (Gu et al. 2017). There are many typical models, including KNN, SVM, Backpropagation, CNN, DNN, and so on. Here, I introduced a model that can be fine-tuned for a specific task, like traffic sign recognition. This research utilized the Single Shot Multi-Box Detector (SSD) (Liu et al. 2016) as a feature extractor and used the 2nd version of MobileNets (Howard et al. 2017) as a model. MobileNets is a neural network architecture that uses depth-wise separable convolutions instead of regular convolutions after the first layer. The depth-wise separable convolution is a combination of two different convolution operations: a depth-wise convolution, and a point-wise convolution. A depth-wise convolution performs a convolution on each channel separately instead of combining the input channels (red, green, or blue are three color channels in a pixel) as a regular convolution does. A point-wise convolution is the same as a regular convolution but uses a 1×1 kernel. A regular convolution does both filters and combines them in a single step. Still, the depth-wise separable convolution separates the process into two stages (one step for filtering, and another step for combining). Even though the results of the two approaches are similar (Howard et al. 2017), the depth-wise separable reduces the number of multiplications, making the model faster than regular convolutions. The details of the algorithm can be found in Howard et al. (2017).

Figure 1 illustrates the process of how standard convolutional filters are replaced by two layers, a depth-wise convolution, and pointwise convolution, to build a depth-wise separable filter. Taking an image with three input channels (red, green, blue cubes represent three base color channels, a cube corresponds with one pixel in the image) as an example, convolution operation combines the values of all the input channels. Standard convolution writes a new output pixel with only a single channel (purple cube). This standard convolution consumed (selected 3 × 3 sample image size) × 3 channels = 27 operations. Depth-wise convolution does not combine the input channels (red, green, blue cubes represent three channels), but it performs convolution on each channel separately. For an image with three channels, a depth-wise convolution creates an output image that still remains three channels: red, green, and blue channels. Each channel gets its own set of weights. The purpose of the depth-wise convolution is to filter the input channels. The depth-wise convolution is followed by a pointwise convolution. This pointwise convolution is the same as a regular convolution but with a 1×1 filter. The purpose of this pointwise convolution is to combine the output channels of the depth-wise convolution (red, green, blue cubes represent three channels) to create new features (purple cube). This depth-wise separable convolution consumed (selected 3 × 3 sample image size) + 3 = 12 operations. This figure illustrates why depth-wise separable convolution has a smaller number of multiplications. That is to say, depth-wise separable convolution has fewer weights and will be faster.

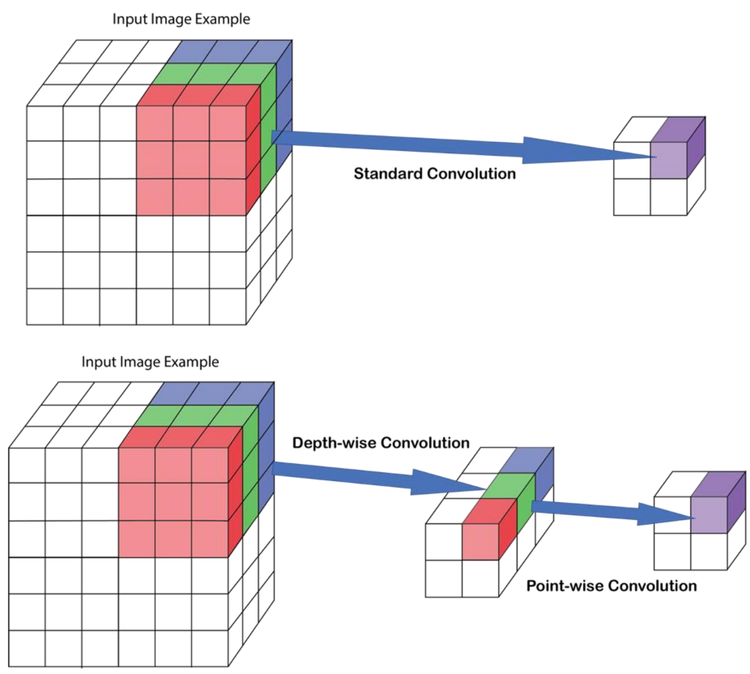



Figure 1 The standard convolution is factorized into a depth-wise convolution and pointwise convolution

***The Google Street View image-based Method***

This section explains how to download images from Google Street View and how to utilize geo-tagged images around intersections for traffic sign recognition. Figure 2 illustrated the workflow of detecting, extracting, locating, and mapping stop signs from Google Street View. More details follow in the next section.

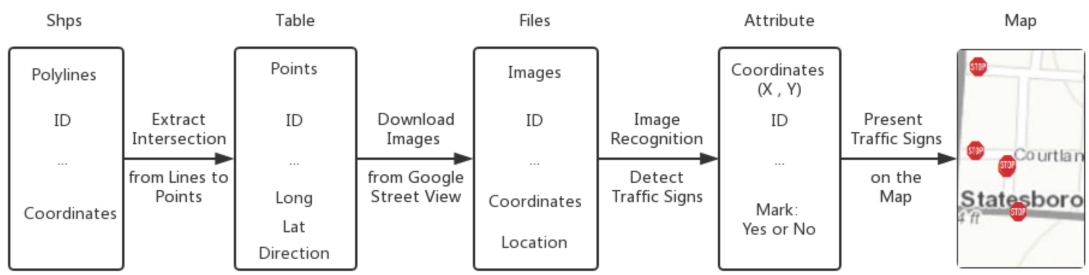



Figure 2 Workflow of the Google Street View method

*Data preparation: Extract intersection and Street View*

For example, stop signs are typically located around road intersections. Road intersections were derived from a road network based on the Topologically Integrated Geographic Encoding and Referencing (TIGER) dataset (Zandbergen et al. 2011). Specifically, I searched all the road intersections in the study area using the intersection operation in GIS. When finding all the intersections, I created intersection buffers with 20 meters to locate observation points on each street in four directions. All these observation points (see Figure 3) were stored in a list $I_s\ \{i_1, i_2, i_3\ ...\}$ to be able to request the images from Google Street View server.


Figure. 3 Observation points $I_s$ *(red dots)* around intersections *(green dots) in a sample study area*

*Stop sign recognition using machine learning*

Google provides many Application Programming Interfaces (APIs). Street View Static API can be used to download Street View images with coordinate information. Longitudes, latitudes, and heading directions were sent through Street View Static API to download the pictures of an observation point in the list $I_s$. Figure 4 illustrates the Geo-tagged images' downloading processes. There are 58,769 recent images downloaded within one year in the study area.

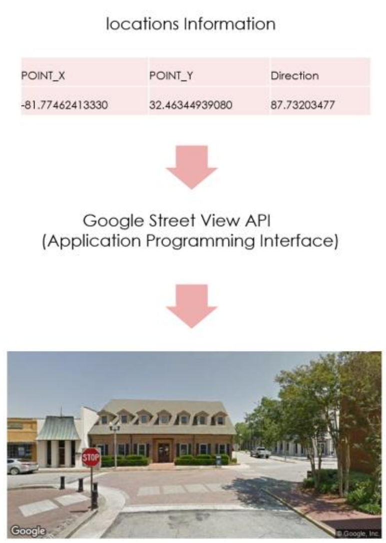



Figure 4 The process of getting geo-tagged images using Google Street View API

After downloading Google Street View images, I used these downloaded geo-tagged images to train the traffic sign recognition model. Initially, there were 496 images selected with traffic signs or assets from the downloaded street view images. They are 136 stop signs, 75 speed limit signs, 188 traffic lights, and 107 fire hydrants. Then, 760 traffic asset records were found and marked down with rectangles among these 496 images. In 760 marked records, randomly selecting 610 records as a training dataset, and the other 150 records as a validation dataset. Additionally, 6250 images were extracted from Google Street View at corresponding 6520 locations around the intersection in the downtown Statesboro area. These images are used as test dataset. After the initial learning process with model evaluation on the validation dataset, all the marked images were put into the TensorFlow Object Detection framework to train a new robust traffic sign detection model. The features of traffic signs were learned, and training parameters (like batch size, initial learning rate, and decay factor) were tuned based on the speed and efficiency during the training process. In this method, the stop sign was chosen as an example. The trained model was then used to detect and locate stop signs on the test dataset, where Google Street View was available. F1-score, recall, and precision (Joshi 2018) were used to evaluate model accuracy. Given a training dataset, $D = \{(x_i, y_i) | x_i \in Rn, y_i \in \{0,1\}\}, i = 1 \ to \ N.$ A positive sample (ground truth is true) is $y_i = 1$, a negative sample is $y_i = 0$. A model $H$ could be built, according to the input sample $x_i$, where there will be a predication $H(x_i)$. Comparing the prediction $H(x_i)$ with ground truth $y_i$, there were thus four situations as follow:

$$H(x_i) = 1, y_i = 1$$

$$H(x_i) = 1, y_i = 0$$

$$H(x_i) = 0, y_i = 1$$

$$H(x_i) = 0, y_i = 0$$

In the first situation, the prediction is true, and the ground truth is true; this situation is called true positive (TP). In the second situation, the prediction is true, but the ground truth is false; this situation is called false positive (FP). In the third situation, the prediction is false, and the ground truth is true; this situation is called false negative (FN). In the last situation, the prediction is false, and the ground truth is false; this situation is called true negative (TN). Every sample would become one of the four situations. It was thus noticeable that

prediction number $N_{pre}$ and the total number $N_{total}$ are:

$$N_{pre} = TP + TN \tag{1}$$

$$N_{total} = TP + TN + FP + FN \tag{2}$$

So, the model accuracy ($Acc$) became:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

Further, the recall, precision, and F1-score are illustrated below.

$$Recall = \frac{TP}{TP + FN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$F1 = \frac{2TP}{2TP + FP + FN} \tag{6}$$

According to these equations, recall represents the model $H$ detection's ability for the positive sample; precision represents the chance that how many percentages this model can distinguish a negative sample capability from a positive sample. The F1-score describes the overall performance of the model prediction. The higher the F1-score, the more robust is the detection model.

### *The GoPro video-based Method*

GoPro (Figure 5) is a versatile action camera with a useful video stabilization function. It can be held by one hand or mounted on a vehicle. It comes with a GPS sensor and an Inertial Measurement Unit (IMU). The camera can be used to record videos and take images. The GPS sensor provides coordinate information. IMU measures speed (both 2D and 3D speed) and accelerator of camera motion.


Figure 5 Experiment setting for the GoPro camera

*Data preparation*

In this method, the Statesboro downtown areas were selected as a study area. GoPro was mounted on the top of a vehicle to capture street views through the camera and collect locations through the GPS sensor. The street view was recorded by GoPro Camera in this experiment. The recorded video set at 60 frames per second. So, videos can be sequentially converted to 60 frames in every 1000 milliseconds. There are 124,896 frames extracted from the recorded street-view videos. The GoPro GPS sensor records coordinate every 55 milliseconds (Table 1) simultaneously. Roughly, 19 frames linked with coordinates per second (see Table 2).

Table 1 GPS trajectory points sample in one route

| **Milliseconds** | **Latitude** | **Longitude** | **Altitude(m)** |
|---|---|---|---|
| 0 | 32.42667 | -81.7808 | 43.962 |
| 55 | 32.42667 | -81.7808 | 44.027 |
| 110 | 32.42667 | -81.7808 | 44.072 |
| 165 | 32.42667 | -81.7808 | 44.083 |
| 220 | 32.42667 | -81.7808 | 44.078 |
| 275 | 32.42667 | -81.7808 | 44.085 |
| 330 | 32.42667 | -81.7808 | 44.049 |

Table 2 GPS trajectory linked to corresponding frames with the same timestamp

| Number | Milliseconds | Latitude | Longitude | Seconds | Frames | Frames/Number |
|---|---|---|---|---|---|---|
| 1 | 0 | 32.42667 | -81.7808 | 0 | 0 | 1 |
| 2 | 55 | 32.42667 | -81.7808 | 0.055 | 3.3 | 4 |
| 3 | 110 | 32.42667 | -81.7808 | 0.11 | 6.6 | 7 |
| … | … | … | … | … | … | … |
| 17 | 880 | 32.42667 | -81.7807 | 0.88 | 52.8 | 53 |
| 18 | 935 | 32.42667 | -81.7807 | 0.935 | 56.1 | 57 |
| 19 | 990 | 32.42667 | -81.7807 | 0.99 | 59.4 | 60 |

*Geo-tagging frames and detection of traffic sign*

After recording, all videos were converted to frames for further image recognition. In this approach, using a similar way to build a training dataset and evaluation dataset. There are 994 frames extracted from selected GoPro videos as input, including 200 stop signs, 200 yield signs, 195 pedestrian signs, 200 speed limit signs, 100 one-way signs, and 99 do not enter signs. They were split into two groups. One is a training dataset with 796 images. Here, using the same training process as the Google Street View method, another traffic sign recognition model was trained with marked traffic signs (stop signs, yield signs, pedestrian signs, speed limit signs, one-way signs, and do not enter signs). The difference in this method is that the training dataset size (796 images) is bigger. This training dataset contains more traffic sign categories, such as stop signs, yield signs, pedestrian signs, speed limit signs, one-way signs, and do not enter signs. This newly trained model was used for traffic sign recognition among frames collected by GoPro. After evaluating this trained model with a 198-records validation dataset, those frames with traffic signs would be sorted out during the image recognition process. Using all the geo-tagged frames with a GoPro sensor, a GPS trajectory was plotted, and the detected traffic signs were mapped out. Figure 6 illustrates the entire workflow below.

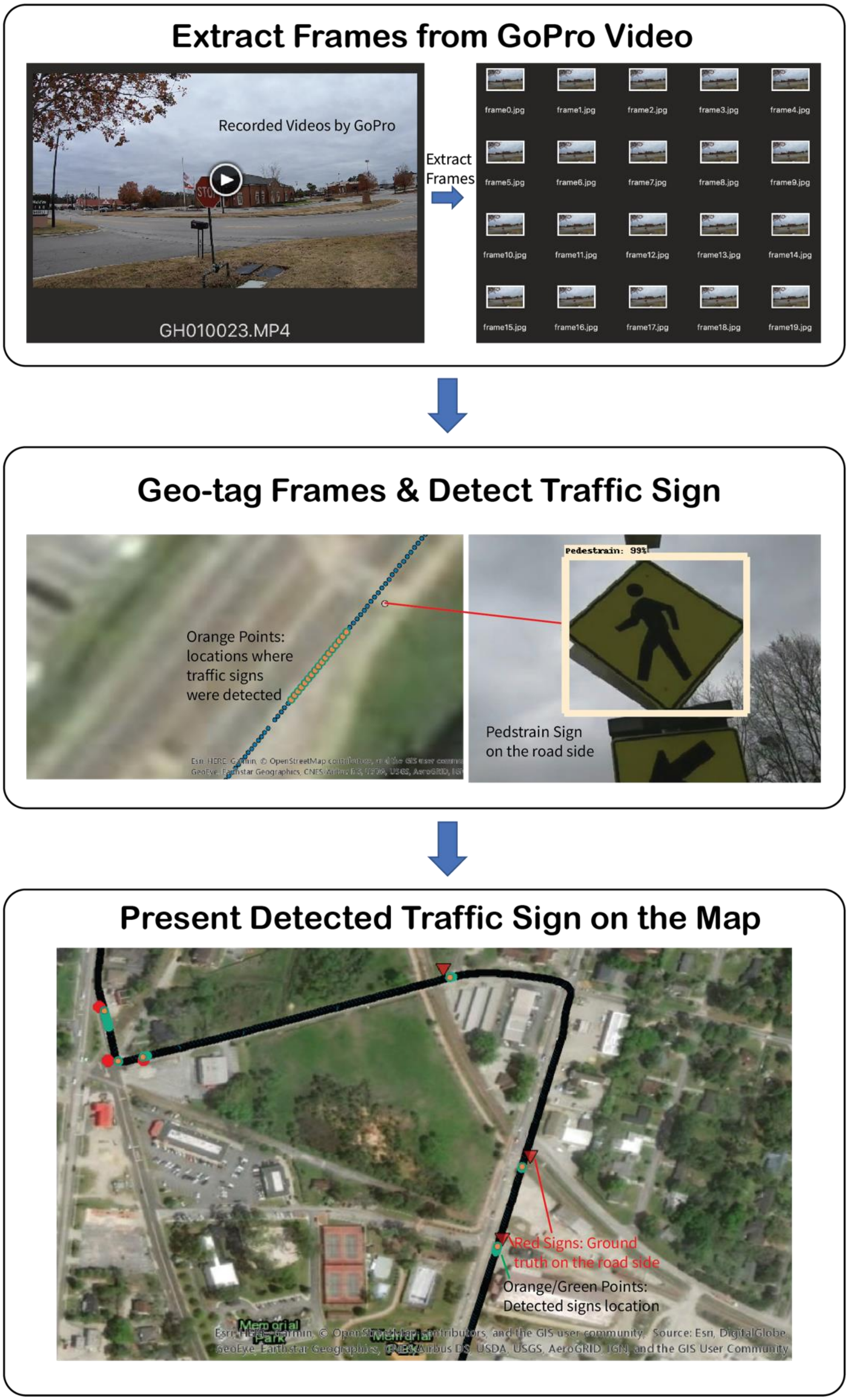



Figure 6 A workflow of the GoPro method

*Accuracy assessment and improvement*

When presenting the detected traffic signs on the map, it is necessary to evaluate the location distance between predictions and ground truth references, the category of traffic signs, and the image detection accuracy of traffic signs. In order to calculate the gap between the detected traffic signs and a ground truth traffic sign, the predictable traffic sign location $D_k$ was taken into account for the distance calculation. A ground truth reference traffic sign $R$ was related to a group of prediction locations $D_k$ $\{D_1, D_2, D_3 \dots D_n\}$. Distance from traffic sign prediction location $D_k$ to the ground reference traffic sign location $R$ was defined as $D_kR$. The average distance $Distance_{Avg}$ describes location accuracy. For example, if a traffic sign was detected at seven locations $\{D_1, D_2, D_3 \dots D_7\}$ around the ground truth traffic sign $R$, the mean center of these seven locations would be taken as the predicted location and the average distance $Distance_{Avg}$ was determined as:

$$Distance_{Avg} = (\,D_1R + D_2R + D_3R + D_4R + D_5R + D_6R + D_7R\,)\ /\ 7.$$

This pattern is illustrated in figure 7. Also, the standard deviation statistics of distance for the different traffic signs and routes were calculated as well.

$$Distance_{Avg} = \sum_{k=1}^{n} \left(\frac{D_kR}{n}\right)$$

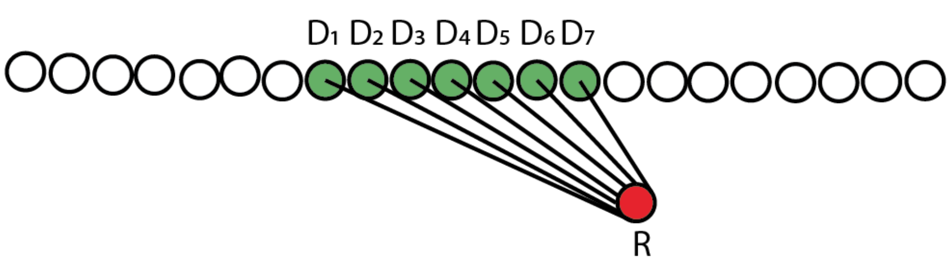



Figure 7 Average distance calculation (Green: detectable traffic sign locations, Red: ground reference)

# CHAPTER 4

# IMPLEMENTATION OF TRAFFIC SIGN DETECTION

*Google Street View of image-based implementation*

Google Street View image-based method was applied to the City of Statesboro, GA, USA, as a study area for testing model usability and accuracy of the proposed solution. A workflow was built for this particular implementation. Python was used for developing the process for downloading geo-tagged images. Longitudes, latitudes, and heading directions were sent to the Google server through Google Street View API to download all the images with their coordinates. The downloaded street view images were associated with certain locations, called Geo-tagged images (Figure 8).


Figure 8 Street view images pop-up map. Green dots: locations of observation points, each pop-up window included a downloaded image from Google Street View at the corresponding location

Object detection API was implemented to train the model (Figure 9) with the TensorFlow framework. Traffic signs were tagged and marked with rectangles on the training dataset images. Every rectangle position and dimension was saved in an XML file. It could be expected to train the model to recognize related traffic sign information by marking down traffic sign features among these images. This particular model was trained on a computer equipped with GPGPU (General-purpose graphics processing units) with Nvidia GTX 1070 Graphic Card. The training process was monitored in the terminal. Also, Tensorboard, a browser-based graph tool, was used to monitor and visualize the training and testing process, providing both graphs and statistics of the training and evaluation process.


Figure 9 The training process of traffic sign recognition

*GoPro video-based implementation*

Five routes of street view (Figure 10) were collected in the downtown area of Statesboro using a car-mounted GoPro, and there were 124,896 frames extracted from all the videos (Figure 11). The entire extracting process contained recording a video with GoPro in 1080p 60 frames per second for street view. These frames were extracted using Python code.

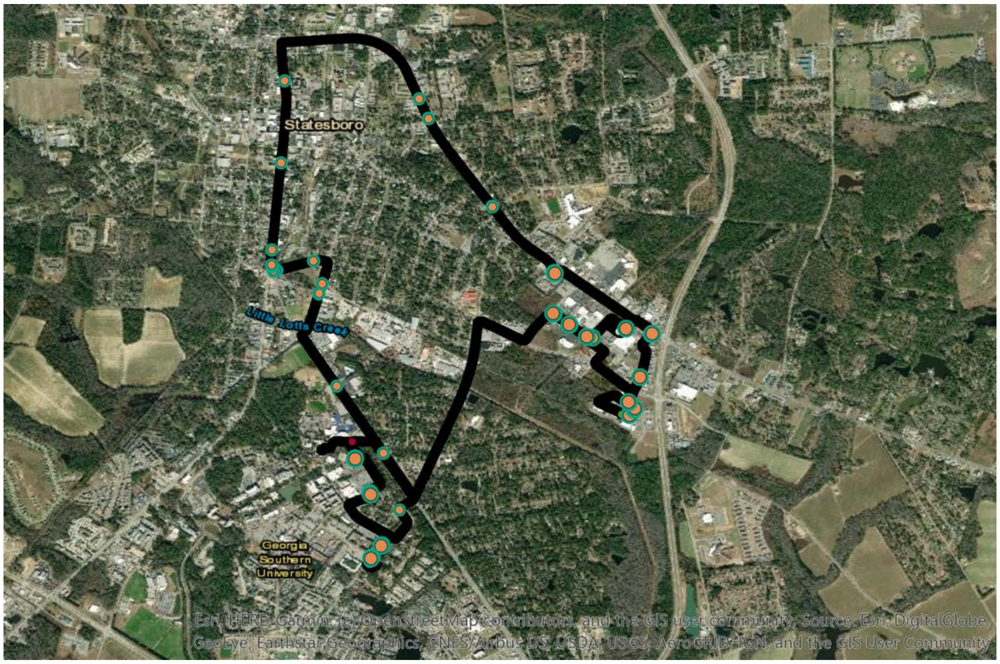
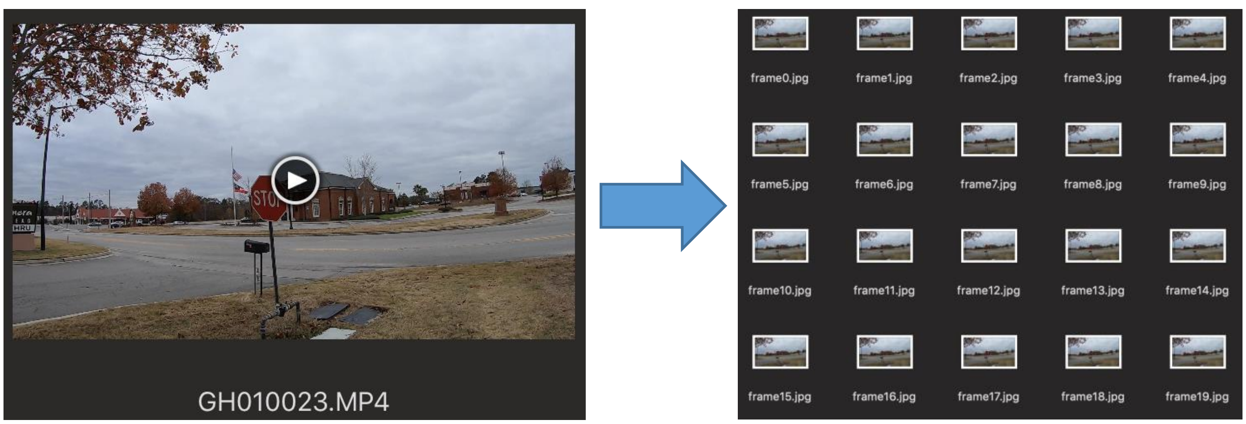



Figure 10 Routes (black lines) in the study area

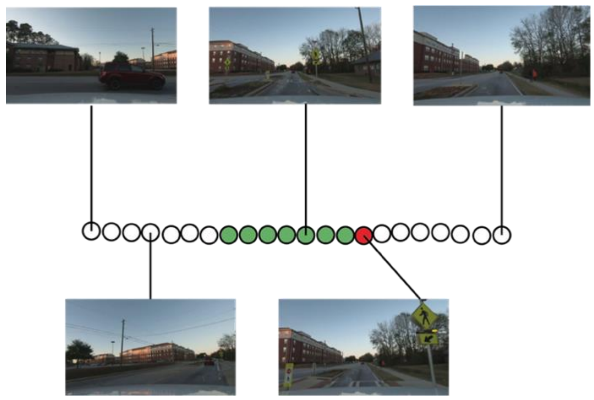



(a) Record the street view video

(b) Extract frames from the video

Figure 11 Process of converting video to frames (60 fps)

The extracted frames from GoPro videos were selected to train a new image recognition model. The GoPro-based method used this new-trained image recognition model to realize traffic sign detection and extraction. Every traffic sign was marked with a rectangle in the training dataset. Traffic sign rectangles position and dimension were saved in XML files. I chose 994 images taken by GoPro consisting of a training and validation dataset with six sign categories, including 200 stop signs, 200 yield signs, 195 pedestrian signs, 200 speed limit signs, 100 one-way signs, and 99 do not enter signs. When the traffic sign recognition model was trained and ready to use, I linked frames to the same timestamp GPS coordinates (Figure 12a). These frames with assigned coordinates can then be mapped out as point features. For example, one pedestrian sign on the ground could be related to multiple frames. (see Figure 12b).



(a) GoPro GPS records linked to frames (Green: traffic sign detected; Red: ground truth; the mid-top picture: a pedestrian sign was detected; images point to hollow points without traffic signs)


(b) Locations of geo-tagged frames (Red: ground truth, Green: traffic signs detected)

Figure 12 The relationship between locations of GoPro frames with detectable traffic signs and locations of ground truth traffic sign

# CHAPTER 5

# RESULTS

*Google Street View of image-based results*

The trained model was applied to traffic sign recognition in Statesboro, GA, USA. The stop signs were successfully detected. Also, I tried the same method to identify the following: : traffic light and speed limit sign (Figure 13). That is to say, this method can be applied to other traffic signs as well. In this result, the stop sign detection result was illustrated as an example. Among geo-tagged images, this model detected and extracted stop signs around intersections (Figure 14). All the detected stop signs with coordinates are visualized in the digital map (Figure 15).


Figure 13 Types of detected traffic signs


Figure 14 Detected stop signs in different background and lighting situations

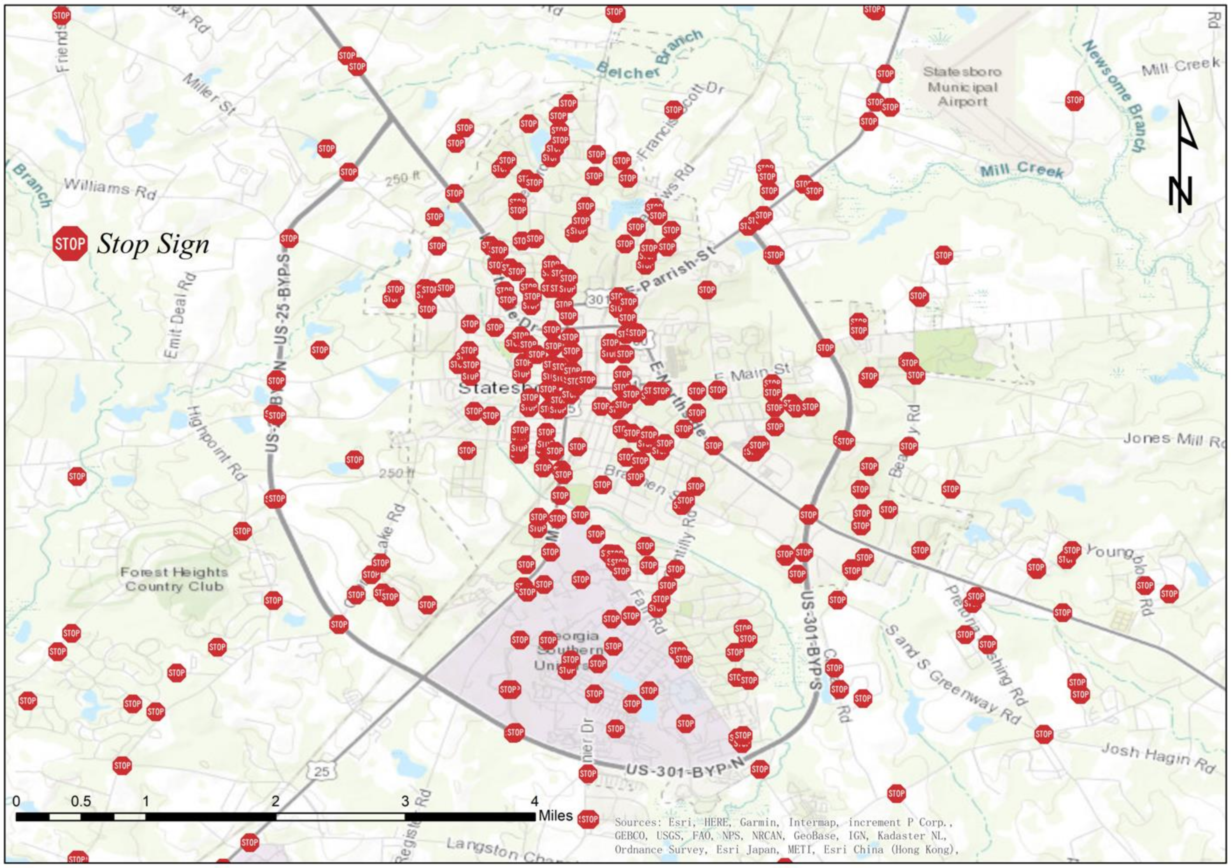



Figure 15 The automatically detected stop signs overlaid on the Statesboro street map

*Model evaluation and accuracy improvement*

After training and validating the traffic sign recognition model, it was necessary to assess the overall accuracy of the model. Iteration also called training step; every iteration will update parameters of the neural network. With the number of training steps increasing, the loss value decreased from 18 to 0.6. Loss value means how well this model worked on training and validation; the lower the loss, the better a model. In this model, the producer accuracy was 93.90%; the user accuracy was 95.85%; thus, the overall accuracy of this model reached 99.60%. The F1 score for stop sign image recognition was 94.86%. The recall detection ability for this model was 93.9%, the precision of this model to distinguish non-stop sign image capability from a stop sign was 95.85%.

The F1-score was 94.96%, which means the overall performance for this model, and its prediction was robust. The confusion matrix of the 6250-images test dataset is shown in Table 3.

Table 3 Confusion matrix for the test dataset

| **Model test results** / **Reference data** | **Stop sign detected (Prediction: YES)** | **Stop sign undetected (Prediction: NO)** | **Total** |
|---|---|---|---|
| There is a stop sign (Ground truth: YES) | 231 (TP) | 15 (FN) | 246 |
| There is no stop sign (Ground truth: NO) | 10 (FP) | 5994 (TN) | 6004 |
| Total | 241 | 60 | 6250 |
| Accuracy Assessment | Recall: 93.90% | Precision: 95.85% | F1 score: 94.86% |

*The GoPro video-based results and accuracy assessment*

Totally, there were 680 video frames detected with traffic signs among selected routes in Statesboro. These geo-tagged frames illustrated the spatial distribution pattern of traffic signs, which are shown on the digital map below (Figure 16). It is noticeable that all the detected traffic signs were around the ground truth traffic signs.

As mentioned above, the distance between the traffic sign detected locations and ground truth reference points can be calculated. Taking Route B as an example, there were five ground truth reference points and 104 frame locations with traffic signs. And the average distance from the detected points to the ground truth locations was around 4.8 meters. I also summarized the distance between detectable traffic sign locations and ground truth traffic sign locations for the selected four routes. The average distance and standard deviation statistical table are listed below (Table 4). Overall, the detection performance was robust, and the prediction results were accurate. According to the analysis, I mapped out the detected traffic signs for five routes in Statesboro. In the GoPro video-based method, there were 32 traffic signs (stop sign, yield sign, pedestrian sign, and speed limit sign) detected and overlaid on the street map (Figure 16).

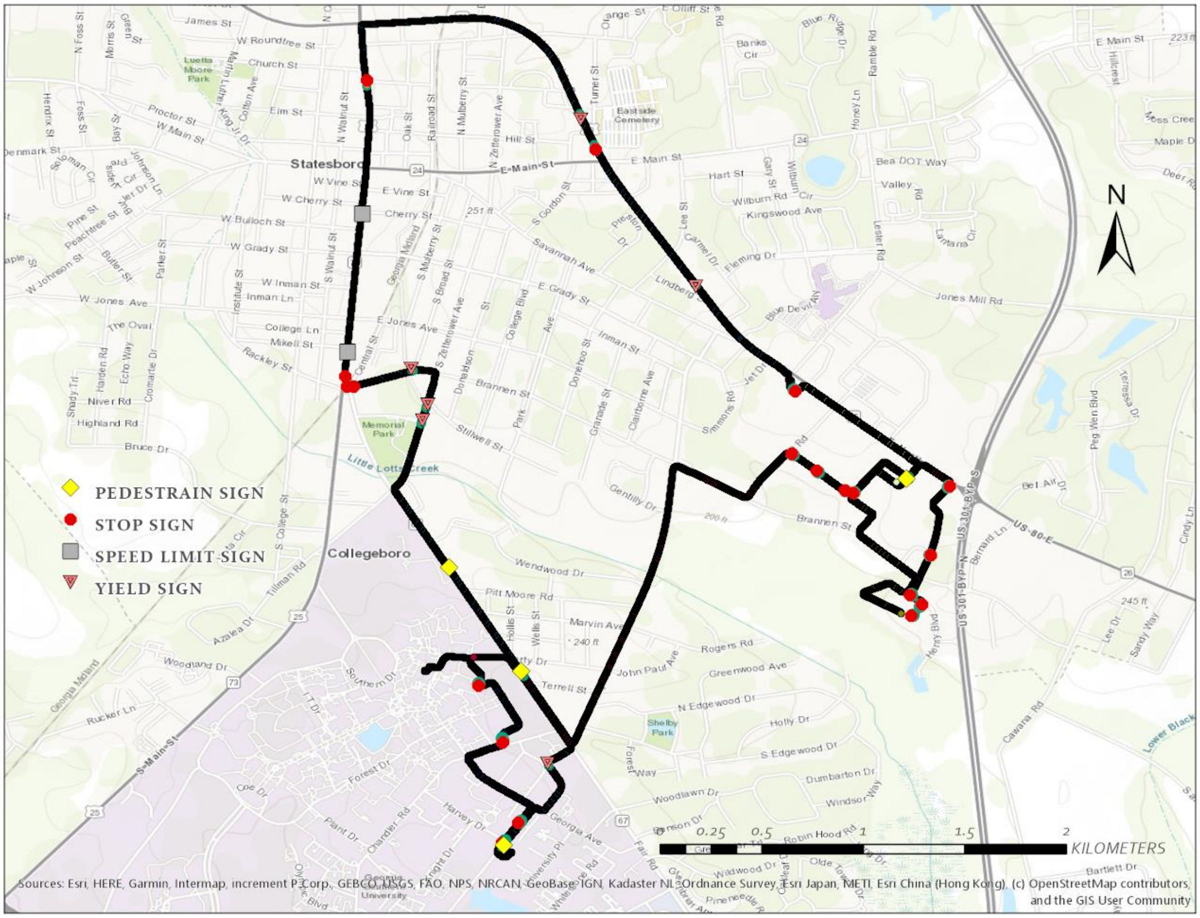



Figure 16 Detected traffic signs using GoPro method in Statesboro, GA

There are four selected routes listed in Table 4. Detected traffic sign number refers to the total number of frames detected with a traffic sign along each route. Reference points mean the number of traffic signs on the ground in this route. Model detection user accuracy refers to the percentage of traffic signs recognized by the trained model in one route. Average distance and median distance represent traffic sign positioning and location accuracy. Average distance means the mathematical average distance between detected traffic sign locations to ground truth. Average distance calculation refers to chapter 3. Median distance is the median number among all the distance numbers between detected traffic sign locations to ground truth. The average distance and median distance were calculated under NAD83 (North American Datum of 1983) projection in ArcGIS.

Standard deviation indicates the consistency of the prediction location. A lower number of standard deviation means location prediction is more stable.

Table 4 Detectable traffic sign locations accuracy assessment

| **Route No.: Traffic No.** | **Detected traffic signs** | **Reference points** | **Model Detection User accuracy** | **Avg. Distance (meters)** | **Med. Distance (meters)** | **Std. Dev** |
|---|---|---|---|---|---|---|
| Route A: | 206 | 16 | 80% | 9.08 | 6.99 | 5.00 |
| Route B: | 104 | 5 | 92% | 4.84 | 4.03 | 1.45 |
| Route C: | 94 | 2 | 99% | 10.45 | 9.18 | 3.28 |
| Route D: | 276 | 9 | 92% | 6.98 | 7.52 | 2.04 |
| Overall | 680 | 32 | 88% | 7.78 | | |

# CHAPTER 6

# DISCUSSION AND CONCLUSION

*Summary of this Research*

This research analyzed the traffic signs in the city of Statesboro and illustrated the workflow of traffic sign recognition and selected types of road traffic sign extraction. In this research, traffic sign detection and positioning workflow were developed to collect and extract traffic signs from Google Street View images and GoPro videos. This research provided a new approach to build and update the traffic sign database efficiently. Specifically, it is valuable and helpful for governments to find damaged traffic signs and rebuild them after a natural disaster. Otherwise, it would be labor and time-intensive to engage personnel to check them one by one. This research applied artificial intelligence and geographic information techniques to detect and locate traffic signs based on images and videos programmatically. Besides, the traffic signs extraction processes were also accelerated by leveraging big data and parallel computing technology.

This research used two data sources, one was Google Street View images, and the other was GoPro videos. Google Street View is available in many cities that provide worldwide area images. They are easy to access and convenient to download so that they can be used to all the areas where Google Street Views are available. Therefore, the traffic sign recognition service proposed in this study can be applied in a wider geographic area.

However, the proposed approach is related to the volume of downloadable street view images for individual use. For example, a personal user is allowed to download only 25,000 images per day. However, Google can unlock this limitation for transportation departments and related authorities.

Google Street View is not available in some locations, and some street views are currently out of date. Therefore, GoPro video was chosen as a second source. When using GoPro to collect data, users can select and control the locations and time of data collection according to their individual needs. However, small dataset size is a limitation in the GoPro video-based method. But the prototype that is proposed in this research can be used to provide an accessible and cost-friendly solution. If there was cooperation between a city transportation department and its police department, multiple GoPro's could be mounted on the police cars, which will provide a large volume of street view videos without additional costs.

This research illustrated a clear solution for locating and mapping traffic signs. Using GoPro realizes this is an accessible and reliable solution. In this method, I created an economic traffic sign detection and mapping system. It can pinpoint traffic sign locations programmatically. Then it is available to visualize traffic signs on the map by using the data generated from the system.

The entire workflow discussed here can be utilized by related departments and technology companies. Traffic sign detection, extraction, positioning, and mapping using GIS, GPS, computer vision, and machine learning can be utilized by local authorities to monitor, maintain and update traffic sign inventory effectively and economically. This method can be widely used for road traffic sign maintenance to improve efficiency, reduce costs, and deliver a smart traffic sign inventory in cities. Further still, this research provides a way to pinpoint traffic signs with high location accuracy, which can also contribute to the autonomous vehicle driving systems.

*Comparison of the Different Approaches*

There are two different approaches applied in this research. These two approaches share some commonalities, but they also have individual differences. Below, three main aspects of difference in the discussion: Traffic sign image classification, location prediction accuracy, implementation convenience and method accessibility.

Firstly, both the Google Street View-based method and the GoPro video-based method used computer visualization with image recognition technology. This research utilizes object detection based on image recognition technology. However, the training images come from two different sources, Google Street View images and GoPro Video frames. The Google Street View method uses stop signs and the others as a training dataset downloaded from Google Street View. The GoPro video method uses six different categories of traffic signs. As a result, when training just one category from Google Street View images, like stop signs, a higher image recognition accuracy than the GoPro Video-based method is achieved. GoPro Video-based method selected six different traffic signs to apply image recognition, which took longer to get decent accuracy.

Secondly, these two approaches have different traffic sign location accuracy. The Google Street View image-based method has a prediction buffer distance of 20 meters. In contrast, the GoPro Video-based method has a higher location accuracy of 7.7 meters, improved by driving an onboard GoPro camera with GPS. Multiple video frames also achieved the prediction with higher location accuracy.

Thirdly, implementation convenience and method accessibility are also different. It is convenient to download Google Street View images online. And these images are available in most large cities in the United States. In comparison, it takes much more time to collect street view using GoPro video-based approach for the same study area. Also, there is a download volume limitation hindered by Google Street View API. Besides, Google Street View is not up to date in some areas. However, the GoPro video method can be applied to everywhere there is a road, even though the GoPro videos collection may be limited by certain conditions such as weather.

*Further Research*

Some other methods and algorithms are available to be used to detect and pinpoint traffic signs. For example, SLAM (Simultaneous localization and mapping) can measure the distance from the viewpoint to the object, which produces an accurate scale from the recorded frame location to the ground truth location. It can be used to pinpoint traffic signs by measuring key points between two frames (Figure 17). While looking and snapping on the same key points (Mur-Artal et al. 2015), the change of view position and the movement of the camera in rotation and its transmission dimension can be calculated. Vision SLAM with IMU (Inertial Measurement Unit) can improve the detection accuracy in the horizontal distance (Tang et al. 2015). The system error radius of prediction for a candidate likelihood area can be narrowed down from 7 meters to 2 meters.

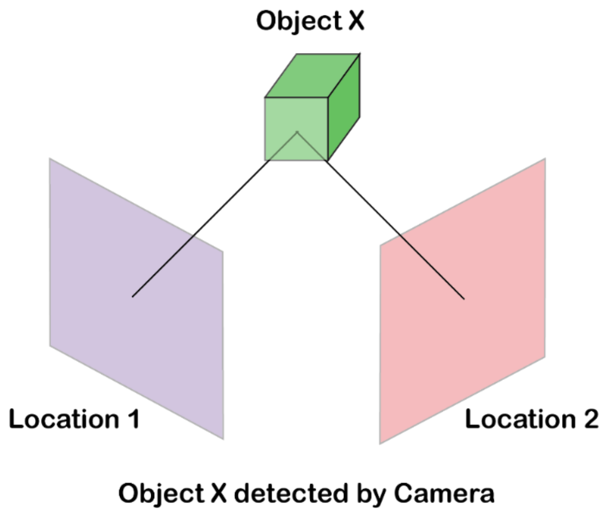



Figure 17 Triangularization to evaluate the location of object X

Structure from Motion (SfM) can be used to estimate camera pose and also help rebuild 3D construction (Carrivick et al. 2016). Due to GoPro coming with a single camera, it is appropriate to choose mono-camera vision SLAM to realize 3D reconstruction. Besides, GoPro equipped with IMU can measure acceleration and orientation and angular velocity in a moving situation. IMU measurement won't change too much in a stable movement, which is called the IMU draft issue (Carrivick et al. 2016). Given the camera can provide image and visual information, I can take this advantage to solve the IMU draft issue in slow and stable movement situations. Combining SLAM with IMU can offer a positive solution for 3D reconstruction. With 3D reconstruction, a computer will understand the real world with scale. In a word, it is possible to know the distance from point A to point B. After 3D reconstruction, I can use deep learning to extract the frame and outline of traffic signs from the mesh generated by cloud points. So, it is possible to detect traffic sign locations in this way. With higher accuracy achieved by using the SLAM method in spatial scale, it will be possible to predict and pinpoint traffic signs within a minimal buffer area, the radius of which could then be controlled within one-meter accuracy.

Augmented Reality (AR) (Todeschini et al. 2019) technology can also be integrated into future work. It can generate a 3D traffic sign model in addition to locating their positions, thereby delivering better visualization. This new workflow (Figure 18) and its expected technical progress can be applied to crewless delivery vehicles as well as other inertial navigation platforms.

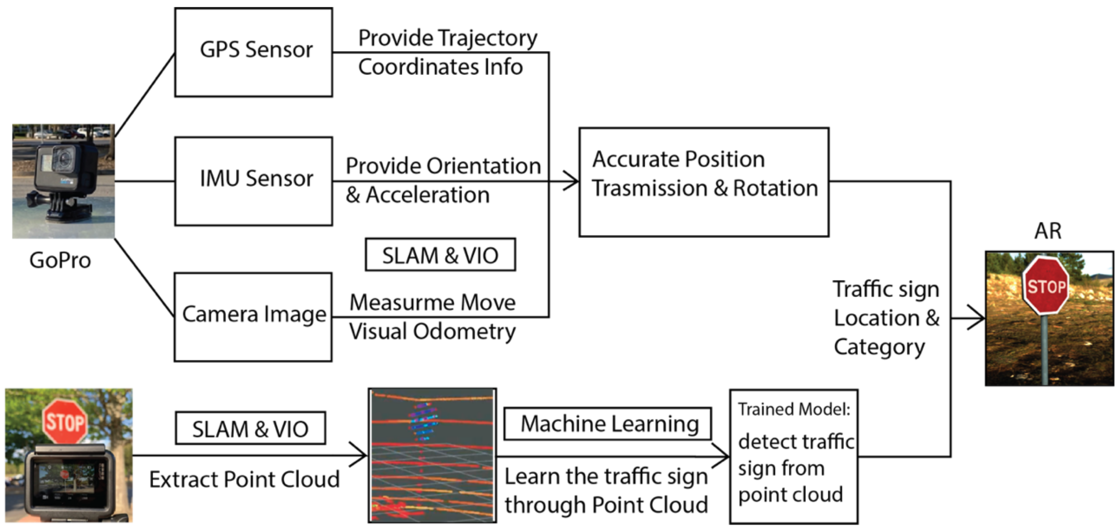



Figure 18 Traffic sign detection and extraction with SLAM

*Acknowledgment*

This thesis researched traffic signs in cities with Google Street View or good road conditions. Two approaches can be applied separately at the same. However, two traffic sign recognition models have to be trained with different image resources and datasets. There is a trade-off between the detection location accuracy and the accessible area of traffic sign detection, based on these two different approaches. This research was funded by a research grant of the Graduate Student Organization from Georgia Southern University and granted by the AAG Applied Geography Specialty Group Awards Committee with AAG Annual Meeting 2019 Project Development Award and Travel Award.

# REFERENCES

Abdi, L. & A. Meddeb. 2017. Deep learning traffic sign detection, recognition and augmentation. In *Proceedings of the Symposium on Applied Computing*, 131-136. ACM.

Anguelov, D., C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent & J. Weaver (2010) Google street view: Capturing the world at street level. *Computer*, 43**,** 32-38.

Balali, V. & M. Golparvar-Fard (2015) Evaluation of multiclass traffic sign detection and classification methods for US roadway asset inventory management. *Journal of Computing in Civil Engineering*, 30**,** 04015022.

Balali, V., M. Golparvar-Fard & J. M. de la Garza. 2013. Video-based highway asset recognition and 3D localization. In *Computing in Civil Engineering (2013)*, 379-386.

Benallal, M. & J. Meunier. 2003. Real-time color segmentation of road signs. In *CCECE 2003-Canadian Conference on Electrical and Computer Engineering. Toward a Caring and Humane Technology (Cat. No. 03CH37436)*, 1823-1826. IEEE.

Borowsky, A., D. Shinar & Y. Parmet (2008) Sign location, sign recognition, and driver expectancies. *Transportation research part F: traffic psychology and behaviour*, 11**,** 459-465.

Broggi, A., P. Cerri, P. Medici, P. P. Porta & G. Ghisio. 2007. Real time road signs recognition. In *2007 IEEE Intelligent Vehicles Symposium*, 981-986. IEEE.

Carrivick, J. L., M. W. Smith & D. J. Quincey. 2016. *Structure from Motion in the Geosciences*. John Wiley & Sons.

Chen, L., Q. Li, M. Li & Q. Mao. 2011. Traffic sign detection and recognition for intelligent vehicle. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, 908-913. IEEE.

Ciregan, D., U. Meier & J. Schmidhuber. 2012. Multi-column deep neural networks for image classification. In *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*, 3642-3649. IEEE.

Crisman, J. D. & C. E. Thorpe. 1991. Unscarf-a color vision system for the detection of unstructured roads. In *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*, 2496-2501. IEEE.

Currin, T. R. 2012. *Introduction to traffic engineering: a manual for data collection and analysis*. Cengage Learning.

Djahel, S., Doolan, R., Muntean, G. M., & Murphy, J. (2014). A communications-oriented perspective on traffic management systems for smart cities: Challenges and innovative approaches. IEEE Communications Surveys & Tutorials, 17(1), 125-151.

Eastman, C. M. (2018). Building product models: computer environments, supporting design and construction. CRC press.

Findley, D. J., C. M. Cunningham & J. E. Hummer (2011) Comparison of mobile and manual data collection for roadway components. *Transportation Research Part C: Emerging Technologies,* 19**,** 521-540.

Ford, T. E., J. A. Yotka & R. J. Turek Jr. 2001. Spatial asset management system that time-tags and combines captured speech data and captured location data using a predifed reference grammar with a semantic relationship structure. Google Patents.

---. 2006. Spatial asset management system and method. Google Patents.

Fuchs, S., S. Rass, B. Lamprecht & K. Kyamakya. 2008. A model for ontology-based scene description for context-aware driver assistance systems. In *Proceedings of the 1st international conference on Ambient media and systems*, 5. ICST (Institute for Computer Sciences, Social-Informatics and ….

Fukushima, K. & S. Miyake. 1982. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In *Competition and cooperation in neural nets*, 267-285. Springer.

Gu, T., B. Dolan-Gavitt & S. Garg (2017) Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*.

Gudigar, A., S. Chokkadi & U. Raghavendra (2016) A review on automatic detection and recognition of traffic sign. *Multimedia Tools and Applications,* 75**,** 333-364.

Halfawy, M. R. (2008) Integration of municipal infrastructure asset management processes: challenges and solutions. *Journal of Computing in Civil Engineering,* 22**,** 216-229.

He, Y., Z. Song & Z. Liu (2017) Updating highway asset inventory using airborne LiDAR. *Measurement,* 104**,** 132-141.

Houben, S., J. Stallkamp, J. Salmen, M. Schlipsing & C. Igel. 2013. Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *The 2013 international joint conference on neural networks (IJCNN)*, 1-8. IEEE.

Howard, A. G., M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto & H. Adam (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Joshi, R. (2018) Accuracy, Precision, Recall & F1 Score: Interpretation of Performance Measures. *URL: https://blog. exsilio. com/all/accuracy-precision-recall-f1-score-interpretation-ofperformance-measures/(visited on 09/30/2018)*.

Kiran, C., L. V. Prabhu & K. Rajeev. 2009. Traffic sign detection and pattern recognition using support vector machine. In *Advances in Pattern Recognition, 2009. ICAPR'09. Seventh International Conference on*, 87-90. IEEE.

Lee, J.-G. & M. Kang (2015) Geospatial big data: challenges and opportunities. *Big Data Research,* 2**,** 74-81.

Liu, W., D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu & A. C. Berg. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*, 21-37. Springer.

Ma, X. & Y. Wang (2014) Development of a data-driven platform for transit performance measures using smart card and GPS data. *Journal of Transportation Engineering,* 140**,** 04014063.

Maldonado-Bascón, S., S. Lafuente-Arroyo, P. Gil-Jimenez, H. Gómez-Moreno & F. López-Ferreras (2007) Road-sign detection and recognition based on support vector machines. *IEEE transactions on intelligent transportation systems,* 8**,** 264-278.

Maldonado-Bascon, S., S. Lafuente-Arroyo, P. Siegmann, H. Gomez-Moreno & F. Acevedo-Rodriguez. 2008. Traffic sign recognition system for inventory purposes. In *2008 IEEE Intelligent Vehicles Symposium*, 590-595. IEEE.

McNeil, S., M. Tischer & A. DeBlasio (2000) Asset management: What is the fuss? *Transportation Research Record: Journal of the Transportation Research Board***,** 21-25.

Miura, J., T. Kanda & Y. Shirai. 2000. An active vision system for real-time traffic sign recognition. In *ITSC2000. 2000 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 00TH8493)*, 52-57. IEEE.

Mogelmose, A., Trivedi, M. M., & Moeslund, T. B. (2012). Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey. IEEE Transactions on Intelligent Transportation Systems, 13(4), 1484-1497.

Mur-Artal, R., J. M. M. Montiel & J. D. Tardos (2015) ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE transactions on robotics,* 31**,** 1147-1163.

Murphy, K. P. 2012. *Machine learning: a probabilistic perspective*. MIT press.

Pomerleau, D. A. 1990. Neural network based autonomous navigation. In *Vision and Navigation*, 83-93. Springer.

Rasdorf, W., J. E. Hummer, E. A. Harris & W. E. Sitzabee (2009) IT issues for the management of high-quantity, low-cost assets. *Journal of Computing in Civil Engineering,* 23**,** 91-99.

Ren, F., J. Huang, R. Jiang & R. Klette. 2009. General traffic sign recognition by feature matching. In *Image and Vision Computing New Zealand, 2009. IVCNZ'09. 24th International Conference*, 409-414. IEEE.

Rublee, E., V. Rabaud, K. Konolige & G. R. Bradski. 2011. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, 2. Citeseer.

Scott, P., S. Cooper, P. Ingram, H. Chalmers & S. Miller. 2011. Road Asset Management Systems. In *Eighth International Conference on Managing Pavement AssetsFugroFederal Highway AdministrationIntervial ChileCAF-Banco de Desarrollo de America Latina*.

Šelih, J., A. Kne, A. Srdić & M. Žura (2008) Multiple‐criteria decision support system in highway infrastructure management. *Transport,* 23**,** 299-305.

Sermanet, P. & Y. LeCun. 2011. Traffic sign recognition with multi-scale convolutional networks. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, 2809-2813. IEEE.

Shustanov, A. & P. Yakimov (2017) CNN design for real-time traffic sign recognition. *Procedia Engineering,* 201**,** 718-725.

Sroub, B. & J. Mackraz. 2003. System and method for managing transportation assets. Google Patents.

Stallkamp, J., M. Schlipsing, J. Salmen & C. Igel. 2011. The German traffic sign recognition benchmark: a multi-class classification competition. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, 1453-1460. IEEE.

--- (2012) Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural networks,* 32**,** 323-332.

Stein, G. P., O. Shachar, Y. Taieb & U. Wolfovitz. 2011. Detecting and recognizing traffic signs. Google Patents.

Tang, J., Y. Chen, A. Kukko, H. Kaartinen, A. Jaakkola, E. Khoramshahi, T. Hakala, J. Hyyppä, M. Holopainen & H. Hyyppä (2015) SLAM-aided stem mapping for forest inventory with small-footprint mobile LiDAR. *Forests,* 6**,** 4588-4606.

Taylor, B., A. Bergan, N. Lindgren & C. B. P. D. P. Eng (2000) The importance of commercial vehicle weight enformcement in safety and road asset management. *Annual Review pp,* 234**,** 237.

Todeschini, E., S. Venkatesha & D. Anderson. 2019. Augmented reality asset locator. Google Patents.

Tucker, L. D., J. Lepper, D. E. Colby, P. Tucker & T. Y. Sawyer Jr. 2009. Precision GPS driven utility asset management and utility damage prevention system and method. Google Patents.

Vanier, D. D. (2001) Why industry needs asset management tools. *Journal of computing in civil engineering*, 15**,** 35-43.

Wang, K. C., Z. Hou & W. Gong (2010) Automated road sign inventory system based on stereo vision and tracking. *Computer ‐ Aided Civil and Infrastructure Engineering*, 25**,** 468-477.

Wolshon, B. 2003. Louisiana traffic sign inventory and management system. Louisiana State University. Department of Civil and Environmental Engieering.

Xu, S. (2009) Robust traffic sign shape recognition using geometric matching. *IET Intelligent Transport Systems*, 3**,** 10-18.

Yakimov, P. Y. (2015) Tracking traffic signs in video sequences based on a vehicle velocity. *Computer Optics*, 39**,** 795-800.

Zamir, A. R. & M. Shah. 2010. Accurate image localization based on google maps street view. In *European Conference on Computer Vision*, 255-268. Springer.

Zandbergen, P. A., D. A. Ignizio & K. E. Lenzer (2011) Positional accuracy of TIGER 2000 and 2009 road networks. *Transactions in GIS,* 15**,** 495-519.

# APPENDIX A

## ABBREVIATIONS

IMU: Inertial Measurement Unit

SLAM: Simultaneous Localization and Mapping

ITS: Intelligent Transportation System

ATSDE: Automatic Traffic Sign Detection and Extraction

SIFT: Scale-Invariant Feature Transform

SURF: Speeded Up Robust Feature

ORB: Oriented FAST and Rotated BRIEF

GIS: Geographic Information System

GPS: Global Positioning System

CUDA: Compute Unified Device Architecture

RGB: Red, Green, Blue

HSV: Hue, Saturation, Value

CNN: Convolutional Neural Networks

ROI: Region of Interest

GTSRB: German Traffic Sign Recognition Benchmark

MLP: Multilayer Perceptron

SVM: Support Vector Machines

IK-SVM: Intersection Kernel Support Vector Machines

DNN: Deep Neural Networks

MCDNN: Multi-column Deep Neural Networks

SSD: Single Shot Multi-Box Detector

TIGER: Topologically Integrated Geographic Encoding and Referencing

API: Application Programming Interface

GPGPU: General-Purpose Graphics Processing Units

NAD83: North American Datum of 1983

SFM: Structure from Motion

AR: Augmented Reality