**FLORIDA A&M UNIVERSITY-FLORIDA STATE UNIVERSITY**

**UNIVERSITY OF NORTH FLORIDA**

# Using Computer Vision and Deep Learning Techniques to Extract Roadway Geometry Features from Aerial Images

Sponsor Award No.: BDV30-977-02

A Report Submitted to Florida Department of Transportation, Traffic Engineering and Operations Office

***Task 5: Final Report***

Start Date: October 2023; End Date: January 2024

*FDOT Project Manager*: **Mariano Amicarelli**, State Traffic Studies Engineer, Traffic Engineering and Operations Office

*Principal Investigator:*
**Eren Erman Ozguven, Ph.D.**
Associate Professor
Department of Civil & Environmental Engineering
Florida A&M University-Florida State University
Phone: +1(850) 410-6146
E-mail: eozguven@eng.famu.fsu.edu

*Co-Principal Investigator:*
**Ren Moses, Ph.D., P.E.**
Professor
Department of Civil & Environmental Engineering
Florida A&M University-Florida State University
Phone: +1(850) 410-6191
E-mail: rmoses@eng.famu.fsu.edu

*Co-Principal Investigator:*
**Maxim A. Dulebenets, Ph.D., P.E.**
Associate Professor
Department of Civil & Environmental Engineering
Florida A&M University-Florida State University
Phone: +1(850) 410-6621
E-mail: mdulebenets@eng.famu.fsu.edu

*Co-Principal Investigator:*
**Thobias Sando, Ph.D., P.E., PTOE**
Professor
School of Engineering
University of North Florida
Phone: +1(904) 620-1142
E-mail: tsando@unf.edu

January 2024

## DISCLAIMER

The opinions, findings, and conclusions expressed in this publication are those of the authors and not necessarily those of the State of Florida Department of Transportation.

## METRIC CONVERSION CHART

U.S. UNITS TO METRIC (SI) UNITS
LENGTH

| SYMBOL | WHEN YOU KNOW | MULTIPLY BY | TO FIND | SYMBOL |
|---|---|---|---|---|
| **in** | inches | 25.4 | millimeters | mm |
| **ft** | feet | 0.305 | meters | m |
| **yd** | yards | 0.914 | meters | m |
| **mi** | miles | 1.61 | kilometers | km |

METRIC (SI) UNITS TO U.S. UNITS
LENGTH

| SYMBOL | WHEN YOU KNOW | MULTIPLY BY | TO FIND | SYMBOL |
|---|---|---|---|---|
| **mm** | millimeters | 0.039 | inches | in |
| **m** | meters | 3.28 | feet | ft |
| **m** | meters | 1.09 | yards | yd |
| **km** | kilometers | 0.621 | miles | mi |

## TECHNICAL REPORT DOCUMENTATION PAGE

| | | |
|---|---|---|
| 1. Report No. | 2. Government Accession No. | 3. Recipient's Catalog No. |

| | |
|---|---|
| 4. Title and Subtitle<br>Using Computer Vision and Deep Learning Techniques to Extract Roadway Geometry Features from Aerial Images | 5. Report Date<br>01/31/2024 |
| | 6. Performing Organization Code |
| 7. Author(s)<br>Eren Erman Ozguven, Ren Moses, Maxim A. Dulebenets, Thobias Sando | 8. Performing Organization Report No. |
| 9. Performing Organization Name and Address<br>Florida A&M University-Florida State University<br>2525 Pottsdamer Street<br>Tallahassee, FL 32310-6046, USA | 10. Work Unit No. (TRAIS) |
| | 11. Contract or Grant No.<br>BDV30-977-02 |
| 12. Sponsoring Agency Name and Address<br>Florida Department of Transportation<br>605 Suwannee Street,<br>Tallahassee, FL 32399 | 13. Type of Report and Period Covered<br>Draft Final Report<br>Period Covered: March 2022 – January 2024 |
| | 14. Sponsoring Agency Code |

| |
|---|
| 15. Supplementary Notes |

16. Abstract

A recent project titled "Feasibility Analysis of Real-time Intersection Data Collection and Processing Using Drones" (BDV30-977-29) aimed at providing proof of concept on the use of image processing techniques to extract intersection characteristics. Improving this work, this project proposed to develop a novel methodology to extract different roadway geometry data such as school zone markings and lane configurations (i.e., turning lanes) from high resolution aerial images using computer vision and artificial intelligence techniques. As such, this project developed automated tools to detect these roadway features in Florida using deep learning-based object detection models. This study, with the objective of generating an inventory list of different roadway geometry data for the entire state of Florida, is one of a kind. The overall goal of this project was to develop computer vision tools to extract different roadway geometry data such as school zone markings and lane configurations (i.e., turning lanes) from high resolution aerial images, which can be used by FDOT planners and engineers at various levels of traffic operations and safety analysis. Consistent with this goal, the main objectives of this project were to: (a) examine how traffic data collection can leverage emerging computer vision techniques, in particular, image processing, deep learning, machine learning, and artificial intelligence, to develop statewide roadway inventory lists; (b) design an automated signalized intersection geometric data extraction algorithm based on high-resolution images in order to identify roadway geometry data such as school zone markings and lane configurations (i.e., turning lanes) from high resolution aerial images, and (c) generate a GIS-based inventory list of these roadway geometry features for the entire state of Florida including on and off state highway system roadways. This was an innovative solution that employs computer vision technology to potentially replace traditional manual inventory, which is labor intensive and prone to errors. Meeting these objectives led to appropriate recommendations to Florida DOT in terms of providing automated, deep learning- and image processing-based methodologies in extracting a variety of geometric features from aerial images. The findings could be published in FDOT manuals and reports, as appropriate.

| | |
|---|---|
| 17. Key Word<br>computer vision; image processing; deep learning; school zones; turning lanes | 18. Distribution Statement<br>No restrictions |

| | | | |
|---|---|---|---|
| 19. Security Classif. (of this report)<br>Unclassified | 20. Security Classif. (of this page)<br>Unclassified | 21. No. of Pages<br>85 | 22. Price |

# ACKNOWLEDGEMENTS

This project was sponsored by the Florida Department of Transportation (FDOT). The Principal Investigators would like to thank the FDOT Project Manager Mariano Amicarelli as well as Dibakar Saha and Gail Holley from the FDOT Traffic Engineering and Operations Office and Alan El-Urfali from HNTB for their valuable feedback throughout this task. The research team would like to thank the following devoted and motivated students who played an active and critical role in this task: Richard Antwi, Samuel Takyi, Michael Kimollo, Mehmet Burak Kaya, and Razieh Khayamim. We are very grateful for the time and energy they invested in the task. Richard Antwi, who is the student team leader, has shown exceptional skills in management and analysis.

## EXECUTIVE SUMMARY

A satisfaction survey among the state DOTs according to Jalayer et al. (2015) indicated that the collection of geometry data with aerial and satellite images happened to be more satisfactory than field observations in terms of equipment cost, data accuracy, crew safety, data collection cost, and data collection time. On the other hand, field observations were found to be more satisfactory in terms of data completeness and data reduction time. Also, image processing has been considered a time-consuming and error-prone approach for roadway inventory recording. However, these results have changed in the last six years with the advancement in computing power and the processing methods of imagery data. Rapid advancement in computer vision technology could enable traffic agencies to save money and time in various aspects of data collection. The existing literature suggested that the recent significant improvements in computational power and image pattern recognition algorithms have created new opportunities to detect and map numerous roadway features from various imagery data.

A recent project titled “Feasibility Analysis of Real-time Intersection Data Collection and Processing Using Drones” (BDV30-977-29) aimed at providing proof of concept on the use of image processing techniques to extract intersection characteristics. Improving this work, this project proposed to develop a novel methodology to extract different roadway geometry data such as school zone markings and lane configurations (i.e., turning lanes) from high resolution aerial images using computer vision and artificial intelligence techniques. To this end, there has not been a study on using the computer vision techniques to extract different roadway geometry data and develop a statewide inventory for roadway geometry data such as school zone markings and lane configurations (i.e., turning lanes) from high resolution aerial images and a study related to how this can benefit roadway users such as drivers, pedestrians, and bicyclists. As such, this project developed automated tools to detect these roadway features in Florida using deep learning-based object detection models. This was achieved by running a retrained You Only Look Once (YOLO) artificial intelligent model to look for the introduced pavement marking combinations on the high-resolution aerial images followed by the GIS-based spatial analyses for both on and off state highway system roadways of Florida. This study, with the objective of generating an inventory list of different roadway geometry data for Florida, is one of a kind.

The overall goal of this project was to develop computer vision tools to extract different roadway geometry data such as school zone markings and lane configurations (i.e., turning lanes) from high resolution aerial images, which could be used by FDOT planners and engineers at various levels of traffic operations and safety analysis. Consistent with this goal, the main objectives of this project were to: (a) examine how traffic data collection can leverage emerging computer vision techniques, in particular, image processing, deep learning, machine learning, and artificial intelligence to develop statewide roadway inventory lists; (b) design an automated signalized intersection geometric data extraction algorithm based on high-resolution images in order to identify roadway geometry data such as school zone markings and lane configurations (i.e., turning lanes) from high resolution aerial images, and (c) generate a GIS-based inventory list of these roadway geometry features for the entire state of Florida including on and off state highway system roadways. This was an innovative solution that employed computer vision technology to potentially replace traditional manual inventory, which was labor intensive and prone to errors. Meeting these objectives led to appropriate recommendations to Florida DOT in terms of providing automated, deep learning- and image processing-based methodologies in extracting a variety of geometric features from aerial images.

## TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

# 1. INTRODUCTION

A satisfaction survey among the state DOTs according to Jalayer et al. (2015) indicated that the collection of geometry data with aerial and satellite images happened to be more satisfactory than field observations in terms of equipment cost, data accuracy, crew safety, data collection cost and data collection time. On the other hand, field observations were found to be more satisfactory in terms of data completeness and data reduction time. Also, image processing has been considered a time-consuming and error-prone approach for roadway inventory recording. However, these results have changed in the last six years with the advancement in computing power and the processing methods of imagery data. Rapid advancement in computer vision technology could enable traffic agencies to save money and time in various aspects of data collection. The existing literature suggested that the recent significant improvements in computational power and image pattern recognition algorithms have created new opportunities to detect and map numerous roadway features from various imagery data.

A recent project titled “Feasibility Analysis of Real-time Intersection Data Collection and Processing Using Drones” (BDV30-977-29) aimed at providing proof of concept on the use of image processing techniques to extract intersection characteristics. The project has been completed and proved the potential of using image processing techniques in extracting roadway geometric features. Improving this work, this project proposed to develop a novel methodology to extract different roadway geometry data such as school zone markings and lane configurations (i.e., turning lanes) from high resolution aerial images using computer vision and artificial intelligence techniques.

To this end, there has not been a study on using computer vision techniques to extract different roadway geometry data and develop a statewide inventory for roadway geometry data such as school zone markings and lane configurations (i.e., turning lanes) from high resolution aerial images, and a study related to how this can benefit roadway users such as drivers, pedestrians, and bicyclists. As such, this project developed automated tools to detect these roadway features using deep learning-based object detection models in the State of Florida. This was achieved by running a retrained You Only Look Once (YOLO) artificial intelligence model to look for the introduced pavement marking combinations on high-resolution aerial images followed by GIS-based spatial analyses for both on and off state highway roadways of Florida. This study, with the objective of generating an inventory list of different roadway geometry data for the entire state of Florida, is one of a kind.

## 1.1. Study Objectives

The overall goal of this project was to develop computer vision tools to extract different roadway geometry data such as school zone markings and lane configurations (i.e., turning lanes) from high resolution aerial images, which could be used by FDOT planners and engineers at various levels of traffic operations and safety analysis. Consistent with this goal, the main objectives of this project were to: (a) examine how traffic data collection can leverage emerging computer vision techniques, in particular, image processing, deep learning, machine learning, and artificial intelligence to develop statewide roadway inventory lists; (b) design an automated signalized intersection geometric data extraction algorithm based on high-resolution images in order to identify roadway geometry data such as school zone markings and lane configurations (i.e., turning lanes) from high resolution aerial images, and (c) generate a GIS-based inventory list of these roadway geometry features for the entire state of Florida including on and off state highway system roadways. This was an innovative solution that employed computer vision technology to potentially replace traditional manual inventory, which was labor intensive and prone to errors. Meeting these objectives will lead to appropriate recommendations to Florida DOT in terms of providing automated, deep learning- and image processing-based methodologies in extracting a variety of geometric features from aerial images. The findings could be published in FDOT manuals and reports, as appropriate.

### 1.2. Report Structure

The remainder of this report is structured in the following manner. After describing the research gaps based on the stated objectives in Chapter 1, Chapter 2 describes Task 1 Deliverable of this project, which includes the overview on state DOTs, and the RCI and the MIRE introduced by the Federal Highway Administration, and literature review. Chapter 3 presents Task 2, where study area, materials and methods, and developed object detection models are presented with an evaluation of the machine learning models used in collecting roadway geometric feature and discussion of the results from the study. Chapter 4 proposes guidelines on the statewide utilization of the proposed methodology. Chapter 5 discusses the conclusions and future work.

# 2. TASK 1: CONDUCT A REVIEW OF LITERATURE AND PRACTICE TO IDENTIFY BEST IMPLEMENTATIONS AND STRATEGIES

In this task, the research team developed and evaluated the machine learning models used to detect and extract roadway geometry features such as school zones and turning configurations. Based on the review of literature that has been conducted for the purpose of discovering published information that can help inform, shape, or guide the conduct of this research project, the models were developed and implemented throughout the State of Florida to extract school zones and turn configurations on both on and off state highway system roadways. The main focus of this task was to develop a machine learning model to extract roadway geometry features from high resolution aerial images. For this purpose, machine learning-based object detection models were developed for detecting school zones and turning configurations (e.g., left only, right only, and center lanes) from high-resolution aerial images. Also, the research team outlined various methods and approaches used in developing this model and extracting geometric data at intersections and data needs to facilitate the analysis.

## 2.1. Department of Transportations and Roadway Characteristics Inventory (RCI)

Since the release of the Highway Safety Manual (HSM) in 2010, many state departments of transportation (DOTs) started implementing the manual's safety requirements in their duties with the focus on improving safety (Jalayer et al., 2014). To achieve this, there has been the need to collect vital information about the transportation network made up of thousands of miles of highways, which has been a major challenge for state and local agencies to deal with. The fundamental responsibility for most highway agencies has been the collection of roadway inventory data. Keeping an up-to-date roadway inventory database has been vital for roadway planning, maintenance, designing, and rehabilitation purposes (Shamayleh and Khattak, 2003). According to the literature, recent considerable increases in processing power and development of picture pattern recognition algorithms have created new possibilities for detecting and mapping a variety of roadway features from diverse imagery data.

Roadway characteristics inventory (RCI) data is a collection or data inventory of all features that make up a roadway. Some general RCI data include Highway Performance Monitoring System (HPMS), roadway geometry, traffic signals, Rail Line Facility, number of lanes, Traffic monitoring sites (e.g., TTMS), turning restrictions, intersections, interchanges, number of rest areas with or without facilities, High Occupancy Vehicle (HOV) lanes, pavement markings, road signs, pavement quality, driveways, and bridges. RCI have broad users in addition to the DOT officers. It is used by the public, local governments, mapping companies, engineers, and law enforcement agencies. RCI data is unique from other highway agencies' data since they are not collected only for specific projects, but they are also collected on each roadway. That is, they are specific to the roadway but not the surrounding areas or buildings, and they basically describe the type, use, and physical characteristics of the roadway (Karimi et al., 2000).

There are several ways of collecting RCI data. The methods for collecting roadway feature depends on the characteristic of interest. Therefore, various methods have been used by DOTs to collect RCI data including field inventory, satellite imagery, mobile and airborne Light Detection and Ranging (Lidar), integrated Geographic Information System (GIS)/Global Positioning System (GPS) mapping systems, static terrestrial laser scanning, and photo/video logging (Jalayer et al., 2014).

Table 2-1 describes the common methods used for collecting RCI data in the literature. The methods can be classified into ground observation/field surveying, and aerial imagery or photogrammetry. It is generally costly to collect RCI data. Each of the methods require unique set of equipment and time for data collection which influence the cost of data collection. In field surveying, measurements along the roadside are taken using total stations. Field inventory involves collecting road geometry data by driving along the roadway and taking notes of current conditions which is later fed into the inventory database. For aerial imagery or photogrammetry, RCI data is extracted from georeferenced images taken by airborne systems such as satellites, aircrafts, or drones. Each method has their limitations and advantages related to cost of data acquisition, data accuracy, quantum of data that can be collected, data collection constraints, data storage requirements, labor intensity, data acquisition time, data processing time, and safety of crews. The data formats for RCI data are usually shapefiles, CSV, and Oracle SQL Databases.

**Table 2-1:** Description of RCI Data Collection Methods

| **Method** | **Description** |
|---|---|
| Field Inventory | Conventional GPS survey equipment are used to obtain desired road inventory data |
| Photo/Video Log | Geo-located photos or videos are recorded automatically while driving along the roadway to visually extract information about roadway features. |
| Integrated GPS/GIS mapping | Roadway information is recorded using integrated GPS/GIS field data logger and stored in a GIS database. |
| Aerial photography or Satellite imagery | Roadway information is extracted from high-resolution images taken from aircrafts or satellites. |
| Structure from Motion | Roadway information is extracted from detailed 3D models reconstructed from a set of unstructured photos of the roadway. |
| Static Terrestrial Laser Scanning | Roadway information is extracted from direct 3D precision point clouds acquired from stationary 3D laser scanners. |
| Mobile Mapping Systems (MMS)/Mobile Lidar | A vehicle with a Lidar system installed is used to collect direct 3D precision point clouds information of the roadway out of which RCI data is extracted. |
| Airborne Lidar | Roadway information is extracted from direct 3D precision point clouds obtained from Lidar systems installed on aircrafts. |

The data acquisition method used by most highway agencies and DOTs are from direct field observations. Although accurate, they are tedious, time consuming and subject to limitations by adverse weather conditions (Shamayleh and Khattak, 2003). Aside these, collection of traffic data on sites poses a lot of risk to data collectors. This makes it necessary to explore more efficient ways for collecting data in order to compile the roadway inventory. Lately, new, and emerging technologies have been explored with regards to the collection of roadway inventory data. **Table 2-2** describes the techniques used by various state DOTs to collect roadway characteristic inventory data.

**Table 2-2:** Techniques for Collecting RCI Data (Gong et al., 2012)

| **Data Collection Method** | **Equipment Utilized** | **State Agency Implementation** |
|---|---|---|
| Field Inventory<br>(Khattak et al. 2001) | GPS, distance measuring tools, and laptops | Alaska, Colo., Conn., Del., Hawaii, Ky., La., Mass., Mich., Minn., N.Y., N.D., Okla., Ore., Pa. |
| Photo Log<br>(Jeyapalan 2004; Wang et al. 2010) | Vehicle, GPS, and camera | Ariz., Del., Ga., Mich., Wash., Ohio |
| Video Log<br>(Gunaratne et al., 2003) | Vehicle, GPS, and video cameras | Ala., Idaho, Ind., Md., Nebr., N. J.,<br>Okla., Pa. |
| Integrated GPS/GIS Mapping Systems<br>(Caddell et al., 2009) | GPS and GIS data-logger combination | Ala., N.H., Ohio, S.D., Utah, Wash. |
| Aerial Photography<br>(Veneziano 2001) | Airplane, GPS, and digital camera | Iowa, Nebr. |
| Satellite Imagery<br>(Ravani et al. 2009) | N/A | Iowa |
| Virtual Photo Tourism<br>(Uslu et al. 2011) | Digital Camera | |
| Terrestrial Laser Scanner<br>(Jaselskis et al. 2005) | Terrestrial Laser Scanner | Iowa, Caltrans, Wash. |
| Mobile Mapping Systems<br>(Khattak et al. 2001; Graham 2010; Turner and Comport 2007) | GPS, vehicle, distance measurement indicator, inertia measurement system, digital cameras, and Lidar | FHWA, City of Charlotte, Ind., Wis., Nev., Tex., Tenn., Hawaii |
| Airborne Lidar<br>(Shamayleh and Khattak 2003) | Airplane, GPS, and Lidar | Iowa, Nebr. |

The traditional methods like using survey total stations, photo/video logs, and GPS loggers are accurate; however, they expose crew members to traffic and a lot of other dangers. Data collection time is also too long. It requires less automation and more human interaction. The manual processing of data may lead to blunders affecting data integrity.

Satellite/aerial imagery (e.g., Google map or Bing map images) have been used to collect information about the earth over the past decade (Jalayer et al., 2014). RCI data can be extracted from high resolution images taken from an aircraft or satellite (Gong et al., 2012, Zhou et al., 2013). RCI data for large areas can be collected and processed in a relatively short period of time. The increasing availability of high-resolution images and data extraction techniques brings into perspective the need to efficiently use them to extract roadway inventory data. One limitation of this method is the difficulty to extract small objects that are not clear or visible. However, with the development of more efficient machine learning techniques and algorithms, this limitation has been overcome. With these methodologies, well-trained models can be used to increase the accuracy of the output. This method uses little to no field costs, making it economically preferable.

One of the contemporary methods for collecting roadway inventory is the Light Detection and Ranging (Lidar) technology. Lidar uses a principle like RADAR to collect information (Shamayleh and Khattak, 2003). The information of a surface is obtained by Lidar when the return time of reflected light beams coming from the target surface is recorded. The measured return time of the light beams is used to calculate the distance between the instrument and the target, and consequently used to generate a profile of the surface when many light beams are recorded. The Lidar technology may be attached to an aerial platform or mobile (ground) platform. Mobile Lidar captures geospatial information of an object from a moving vehicle.

To this end, there has not been a study on using the computer vision techniques to extract different roadway geometry data and develop a statewide inventory for roadway geometry data such as school zone markings, lane configurations (i.e., turning lanes lengths, and lane, shoulder and median widths), and sidewalks (i.e., presence or absence of sidewalks) from high resolution aerial images, and a study related to how this can benefit roadway users such as drivers, pedestrians and bicyclists. As such, this project will develop automated tools to detect these roadway features using deep learning-based object detection models in the State of Florida. This will be achieved by running a retrained You Only Look Once (YOLO) artificial intelligent model to look for the introduced pavement marking combinations on the high-resolution aerial images followed by the GIS-based spatial analyses for both on and off state highway system roadways of Florida.

### 2.2. Department of Transportations and the Model Inventory of Roadway Elements (MIRE) System

In 2010, the United States Department of Transportation (USDOT), through the Federal Highway Administration (FHWA, 2018), provided a comprehensive database in other to support the various states Department of Transportation (DOT) to build a well-structured data-driven program. This was a generalized database called the Model Inventory of Roadway Elements (MIRE). This database or inventory makes it possible for all state DOTs to have complete access to a collection of roadway components on all public roadways across the United States (Lefler et al., 2010).

The elements in the MIRE database are categorized into three categories:

i. Roadway segments,
ii. Intersections,
iii. Interchanges.

Many states in the United States do not have the Fundamental Data Elements (FDE) data in their existing roadway inventories. Quality data are the cornerstone for making crucial design, operations, and safety decisions on roadways. The need for safe operation is why the Florida Department of Transportation (FDOT) is in search of more robust, convenient, and technological methodologies to extract these roadway geometric features through aerial images, and computer vision techniques. FDOT is also part of the nationwide MIRE, a project for which various DOTs are required to collect certain roadway geometric features to create its generalized database with its fundamental data elements. MIRE was created by the Federal Highway Administration to provide a recommended approach for the creation of a comprehensive highway and traffic data inventory for efficient safety management. The conversion of MIRE, which is now a list of variables, into a Management Information System (MIS) is a vital step toward acceptance, and implementation of MIRE (USDOT, 2013.).

This project seeks to employ artificial intelligence and aerial imagery (AI2) to extract roadway geometric features. The scope of the project is tuned to building a deep learning algorithm to detect school zones, signalized intersections, and other roadway features through image processing, image classification, and image analysis. This project will enable FDOT to collect roadway geometric features and other elements that are specified in the MIRE database. The method that will be used in this project is also safe and cost-effective since aerial images will be obtained from the Aerial Photography Look-Up System (APLUS) of FDOT (FDOT, 2020) and the technique for detecting these features only requires a deep learning application programming interface (API).

This study covers the methods used by other DOTs for collecting MIRE FDE data and the machine learning techniques that have been employed to detect and classify roadway geometric features from aerial imageries. The study also focuses on how training datasets for the deep learning model were collected and how the features were extracted.

### 2.2.1. MIRE Fundamental Data Elements

The Model Inventory of Roadway Elements as developed by the Federal Highway Authority has 205 roadway geometric features that are to be included in the MIRE database. The fundamental data elements (FDE) account for only 37 of the 205 elements. The FDE is divided into three categories depending on the functional class and surface type. These three types of roads in this system are as follows: non-local paved roads, local paved roads, and dirt roads. All 37 FDE for non-local paved roadways, 9 FDE for paved local roadways, and 5 FDE for unpaved roadways must be available to the state. These data elements also include traffic data elements which are very important to safety management. Under certain conditions, the state may choose not to collect the FDE on gravel or other unpaved roadways. The data elements for highway segments, intersections, and interchanges/ramps are separated from the FDE for non-local paved roadways.

These basic but relevant features are the fundamental data elements that several state DOTs are developing more advance technological ways to extract from satellite imageries and others still use manual data collection methods. The data in the MIRE Management Information

Systems (MIS) must be of good quality in order for it to be most functional (Altobello et al., 2013). The federal highway system has roadway segment data, and there are various elements in the data for all public highways. The MIRE FDEs can be built out using data from the Highway Performance Monitoring System (HPMS). The appropriate tables can be created using the Extract, Transform, and Load (ELT) operations (Experience, 2018). Even though some data exist for traffic counts and crash reporting, intersection data is a big gap in the database. These data elements is utilized on all public roadways to aid in the development of Strategic Highway Safety Plans (SHSPs) and Highway Safety Improvement Plans, as well as to support safety analyses.

Additionally, by combining these features in a data-driven strategy, state DOTs will be able to improve their roadway and traffic data collections and provide a link to crash and safety data via geolocation. MIRE also allows connection to other relevant databases, which can help improve the safety analysis. By employing alternate data sources to broaden the scope of coverage of these datasets, such as speed limits, safety analysis can be expanded (Cook et al., 2021).

## 2.3. Literature Review

This section evaluates the state of the practice for RCIs by state and federal transportation agencies. Published work in literature is presented in order to uncover relevant information that can help inform, shape, or guide to conduct this research project.

### 2.3.1. State-of-the-Practice of RCI Data Collection Methods in Traffic Agencies

RCI data collection methods can be categorized into two: land-based methods and air/space-based methods as seen in **Table 2-3**. Each method uses different techniques for data collection.

**Table 2-3:** Categorization of RCI Data Collection (Gong et al., 2012)

| | **Land** | **Air or Space** |
|---|---|---|
| **GPS** | Field Inventory<br>Integrated GPS/GIS Mapping | |
| **GPS and Imagery** | Photo/Video Log | Satellite Imagery and<br>Aerial Imagery |
| **GPS and Imagery and Lidar** | Static Terrestrial Laser Scanning (Uses stationary 3D laser scanners) and<br>Mobile Lidar (Uses Lidar installed on mobile vehicles) | Airborne Lidar |

A survey was conducted by Jalayer et al. (2015) to find out the methods adopted by various DOTs to collect RCI data. Figure 2-1 is a map showing the states that responded to the survey conducted to assess the various methods used by state DOTs to collect RCI data. Figure 2-2 shows the results of this survey conducted to know the technology used by 50 state DOTs to collect various RCI data. The results received from 30 DOTs showed that over 50% of them relied on the traditional ground methods like integrated GPS/GIS mapping, field inventory, and photo/video logging. Very few of them relied on aerial-based methods. Some methods like laser scanning and Lidar were rarely used.

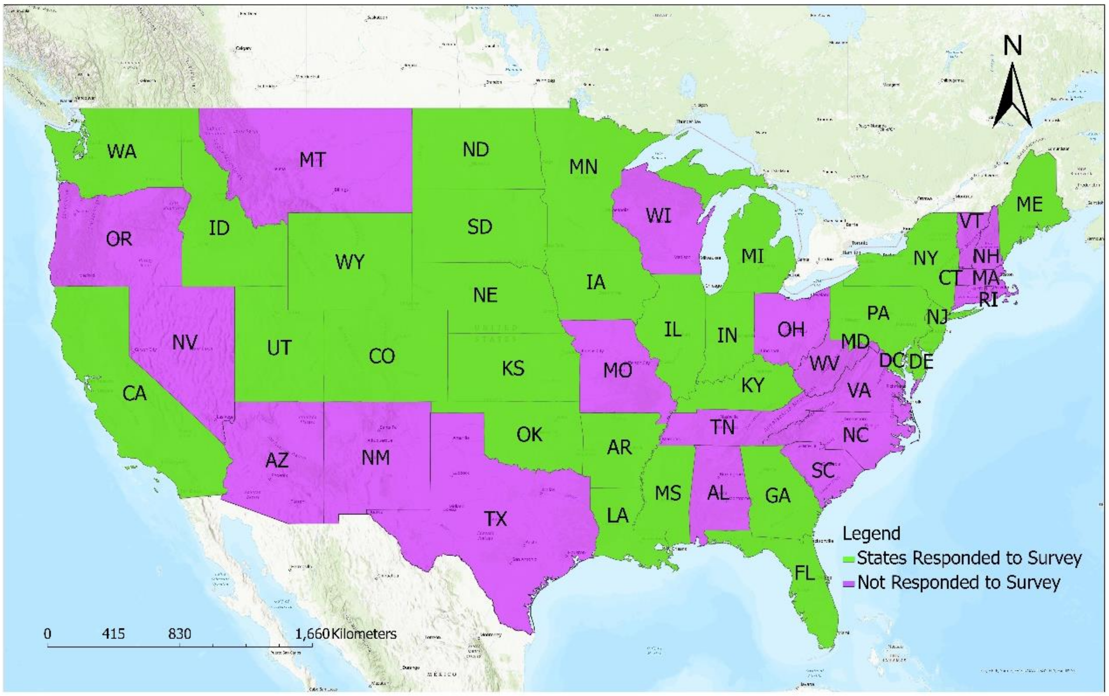


**Figure 2-1:** State DOTs that Responded to the Survey on RCI Methods They Use

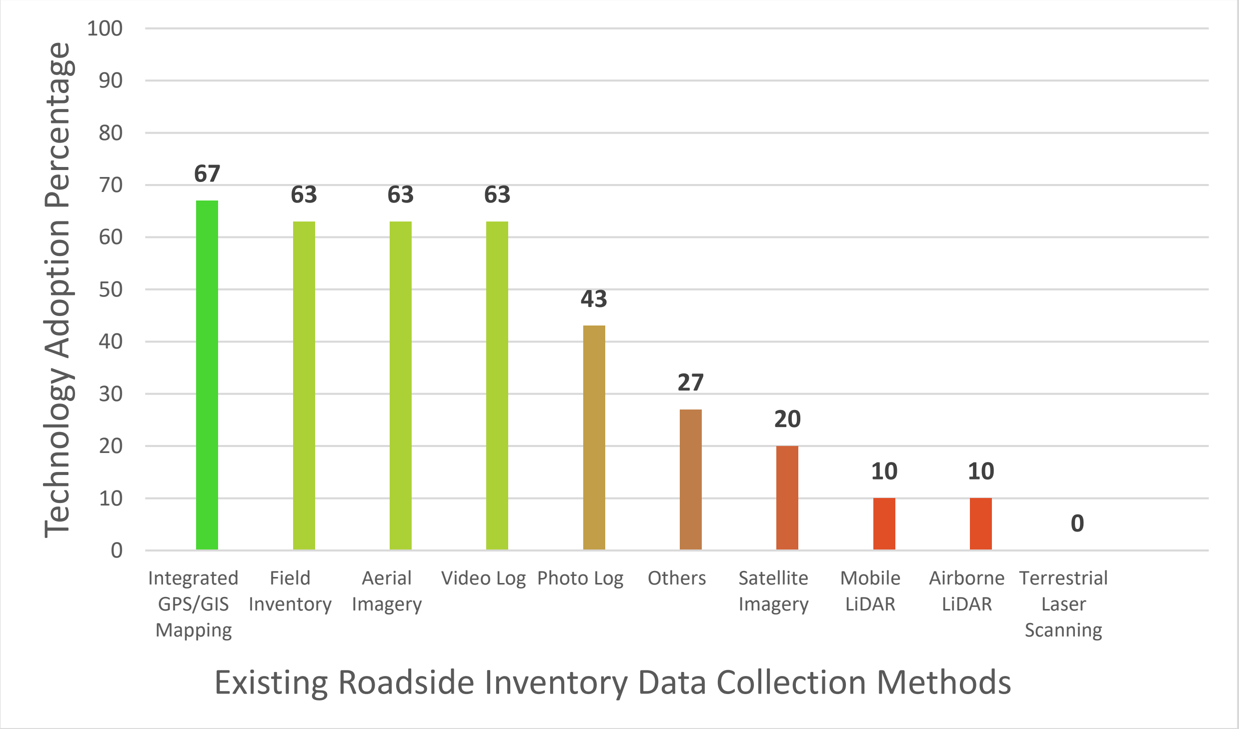


**Figure 2-2:** Percentage of Technology Adoption by 30 DOTs (Jalayer et al., 2015)

Recent studies on RCI data collection methods are shown in Table 2-4. They generally focus more on using Lidar, which is very accurate and produces rich data; however, it has a high cost of equipment and operations, large size of dataset, and long data processing and reduction times. Other studies focus on the traditional methods which has longer data collection times and they interrupt traffic operations and expose crew to traffic and other dangers. Very little focus has been given to emerging technologies such as using deep learning and computer vision techniques to extract roadway features from high resolution aerial images. This method eliminates field work and data collection time. The data for this method is available. It does not expose crew members to any traffic risks and dangers. It is highly accurate and can be used to obtain roadway features in large areas in a short time.

**Table 2-4:** Recent Studies on RCI Data Collection Methods from 2014 to Date

| **Research** | **Method** | **Description** | **Support/Funding** |
|---|---|---|---|
| Jalayer et al. (2014) | Lidar and Aerial Imagery | Evaluated remote sensing methods to collect roadway inventory data. | Illinois DOT |
| Jalayer et al. (2015) | Photo/Video logging | Evaluated photo/video logging methods used to detect highway inventory data. | Illinois DOT |
| Balali and Golparvar-Fard (2015) | Ground/Roadway imagery | Evaluated the use of AI to detect and classify traffic signs | Data from Illinois DOT |
| Guan et al. (2016) | Mobile Lidar | Mobile Lidar was used to collect traffic signals, poles, and roadway pavement. | National Science Foundation of China |
| Yan et al. (2016) | Lidar | A review of roadway collection using Lidar. | Alberta Transportation, Canada. |
| Nagarajan et al. (2016) | Mobile Laser scanner, GPS, and Cameras | Mobile mapping system based on laser scanners and camera was used to collect roadway inventory data (road markings and signs). | Received support from FDOT in collecting field data. |
| He et al. (2017) | Airborne Lidar | Airborne Lidar data together with ArcGIS-based algorithm was used to detect large traffic signs, billboards, bridges, traffic signals. | Utah DOT |
| Teo 2018 | Mobile Lidar | Mobile Lidar was used to extract roadway markings. | Ministry of Interior Taiwan |
| Kargah-Ostadi et al. (2020). | Ground/Roadway imagery | Using AI, traffic signs were detected from roadway imagery obtained from driving along the roadway. | No funding. |

Shamayleh and Khattak (2003) studied the use of Lidar to collect certain roadway inventory such as highway grade, side slope, contours, and stopping and passing sight distance. The Lidar technology uses high speed computing, Global Positioning System (GPS), and laser range finders and precision inertial navigation to collect data (Shamayleh and Khattak, 2003). Lidar systems can be attached to ground or airborne platforms. Lidar systems measure the distance between a target object and the detector (Miotto, 2000) by measuring the time a light pulse takes to reflect to the detector from the target. In this study, the data were collected on computers and used for analysis. The data produced by Lidar systems were used to create digital elevation models (DEM) which is the surface profile of the earth in the form of elevation data. Lidar data and geocoded aerial imagery were merged and analyzed in a GIS platform. The extracted information was compared to ground truth data for validation.

Gong et al. (2012) studied how highway inventory data are extracted using Mobile Terrestrial Laser Scanning (MTLS). Before data collection, the system was calibrated to extract GPS positions of all scanned points during driving. The 3D point clouds, GPS data, and camera data provided from the data collection were post processed to create geo-referenced point clouds and photographs from which roadway features were extracted. Digital elevation models (DEM), digital terrain model (DTM), track geometry, and vectorized contour lines were extracted from the data.

### 2.3.2. Existing Practices of State DOTs for Collecting Roadway Features

Many states have existing inventories that are deficient in FDE data, particularly at intersections although most DOTs have already started working on better data development efforts. Each DOT is required to develop the MIRE fundamental data elements by September 30, 2026. The mapping of safety features maps along roadway networks can help manage and maintain roadway safety infrastructure. The detailed safety feature map can be used by traffic engineers in our various DOTs to identify areas where new safety implementations should be built. The majority of these DOTs acquire data manually by going out into the field to inspect the roadways and visually interpret the street view images taken by transportation officials (Sainju & Jiang, 2020).

The majority of Michigan DOT's available roadway data, including MIRE components, are gathered using the manual data collection from the DOT staff, counties, cities, and other planning organizations (Kwayu et al., 2022). As a result, there have been efforts to use GIS-based technologies and machine learning techniques to automate information extraction from existing databases such as Michigan Imagery Solution and Lidar. The Michigan Department of Transportation has also made attempts to locate and classify intersections by MIRE traffic control type using the State of Michigan Traffic Crash Report (UD-10) crash data, and diagrams (Kwayu et al., 2022). This leads to the roadways being minimally exposed, making majority of Michigan's roadways being exempted from the process.

An Automated Pavement Condition Survey (APCS) is used by Caltrans to collect inventory and condition data for all NHS and SHS pavements. The APCS measures the state of National Highway State and State Highway State pavements every 0.1 miles using high-resolution pictures and lasers. This data collection initiative was started by Caltrans in 2015 (Caltrans, 2018). As part of the preliminary studies and piloting programs for the MIRE project, the California Department of Transportation (CALTRANS) performed a study of state departments of transportation to learn about existing methods for collecting and maintaining roadway asset data in different states (Ravani et al., 2009). The methods used to gather data vary greatly based on the technology available, the type of equipment used, the amount of time required, and the availability of personnel. Integrated mapping systems, airborne LIDAR, aerial photos, and video logging, among other technologies, were used to collect data. The study's findings highlighted the importance of using automated, scalable, and cost-effective approaches to extract desired data from state DOT data sources (Kwayu et al., 2022).

This has proven that artificial intelligent techniques such as deep learning and machine learning should be employed together with other GIS methodologies to extract these roadway geometric features as they provide safer, more convenient, and cost-effective solutions. The deep learning tool embedded in the ArcGIS software makes the entire roadway feature detection model easier and the extraction process more effective and efficient.

New Hampshire DOT (NDHOT) was one of two states chosen by FHWA to run a pilot project to examine the feasibility of collecting, storing, and integrating MIRE data into a MIS. They also combined roadway inventory data with collision and other pertinent data for safety analysis purposes. NHDOT collected data to expand its usage of AASHTOWare Safety Analysis, primarily to assist network screening analysis. They employed the AASHTOWare Safety Analysis to improve the upkeep of their safety data by using automated GIS tools and update the inventory on a yearly basis. This was comparable to the MIRE Fundamental Data Elements component, towards choosing which intersection elements to prioritize in the MIRE MIS intersection data collection operations. The MIRE MIS endeavor was split into two parts. The first part of the project included the design of proprietary GIS toolbar developed to combine the state’s Linear Referencing System (LRS) with the intersection inventory. On the other hand, the second part involved data collectors filling in the intersection elements that were not retrieved from the state’s LRS and double-checked the pre-filled components (Study, n.d.). These were methods that required extra work from the staff as limited data will be collected. This brings the need for these machine learning techniques where algorithms could be built to detect these roadway features from either satellite imageries or drone images automatically.

The more information a state or local agency has for a roadway, the better that agency will be able to deploy resources to identify problem areas, diagnose concerns, recommend appropriate remedies, and assess the effectiveness of those remedies. Other states have various ways of extracting roadway geometric features and other databases, and forms of storage. Table 2-5 indicates these states and their forms of data inventory.

### 2.3.3. Choosing and Object Detection Model

The performance of an algorithm when observing a standardized image collection will not be the sole criterion for selecting the best object detection model. Many considerations have become relevant when choosing a suitable model for a particular target task due to the uniqueness of a project (Campbell et al., 2019). According to this study, most of the leading models, including those provided by TensorFlow, have converged on the same overall concept, making it simple to use any model for custom object detection. As a result, each model's low-level operations will be defined in terms of speed, memory dependence, and accuracy.

**Table 2-5:** Examples of Technologies Used by Other DOTs

| State DOT | Inventory Collection Technique | Storage | Inventory Data |
|---|---|---|---|
| Michigan | Integrated GPS/GIS mapping systems, and field inventory | GIS | Guardrails, pipes, culverts, culvert ends, catch basins, and impact attenuators |
| New Hampshire | Online aerial imagery, and street-level imagery | GIS/Linear Referencing System | Intersection and intersection legs |
| California | Aerial imagery | GIS/Linear Referencing System | Intersection, and interchange ramps |
| Washington | Photo log, integrated GPS/GIS mapping systems | GIS | Cable barriers, concrete barriers, culverts, culvert ends, ditches, drainage inlets, glare screens, guardrails, impact attenuators, miscellaneous fixed objects, pipe ends, pedestals, roadside slope, rock outcroppings, special use barriers, supports, trees, tree groupings, and walls |
| Ohio | Photo log integrated GPS/GIS mapping Systems | GIS | Wetland delineation, and vegetation classification |
| Iowa | Airborne Lidar, and aerial photography | GIS | Landscape, sloped areas, individual counts of trees, side slope, grade, and contour |
| Idaho | Video log | MS Access | Guardrails |
| Tennessee | Tennessee Road Information Management System (TRIMS), and Maintenance Management System (MMS) | Central Database | Traffic signs, guardrails, and pavement markings which are manually collected |
| New Mexico | Photo, Laser Scanner, and Virtual Reality System | Video | Most types of visible highway assets except for light posts, and road detectors |
| Virginia | Web-based asset management system using Google Maps | Google Maps | Cross pipes, and ditches |
| FHWA Baltimore Washington Parkway | Mobile mapping | Point Cloud Software, GIS | Corridors, and signs |

### 2.3.4. Roadway Geometric Feature Extraction and Data Collection Using Computer Vision and Deep Learning

Recently, computer vision and deep learning techniques such as convolutional neural networks and recurrent neural networks have become emerging instruments when it comes to transportation (Habibi Aghdam et al., 2016). These methods have been used in image detection, where autonomous vehicles use them to detect objects on the roadways in real time. Researchers in numerous transportation projects have used aerial images and video data in conjunction with deep learning and computer vision to detect and classify roadway geometry elements (Kwayu et al., 2022).

Computer vision techniques were used to detect and classify traffic signs using Google Street View (GSV) photos of interstate segments in a study (Balali et al., 2016). The researchers employed a machine learning technique that combined the histograms of oriented gradients (HOG) features with a linear support vector machine (SVM) classifier to detect and classify roadway signs based on their pattern and color information. With a classification accuracy of 94.63%, this strategy was successful. The proposed technique, however, was not tested on local roadways or non-interstate routes. Another study used a novel hybrid local multiple systems (LM-CNN-SVM) based on convolutional neural network (CNN) and SVMs for object recognition and pedestrian detection using the Caltech-101 and Caltech pedestrian datasets. This is due to its robust categorization property and powerful feature extraction capabilities (Uçar et al., 2017). The entire image was partitioned into local regions in the proposed system, and several CNNs were used to learn local areas and local object characteristics. Principal component analysis was used to select discriminative characteristics. To improve the generalization ability of the classifier system, the features were then imported into multiple SVMs using both empirical and structural risk minimization instead of a direct CNN. In addition, the output of SVM was fused, and a pre-trained AlexNet and a new CNN architecture were used. The proposed approach had an accuracy range of 89.80% to 92.80%.

Table 2-6 shows the advantages and disadvantages of RCI data collection methods.

**Table 2-6:** Advantages and Disadvantages of RCI Data Collection Methods

| Method | Description | Advantages | Disadvantages | Related studies |
|---|---|---|---|---|
| Field Inventory | Conventional GPS survey equipment is used to collect information in the field. | High data accuracy. Low initial cost. Low data reduction effort. | Tedious. Crew exposed to traffic. Long field data collection time. | Zhou et al. (2013) |
| Photo/Video Log | Automatically recording photos or videos while driving a vehicle along the roadway. | Small roadside objects are visible to identify and record. Less exposure to traffic. Short field data collection time. | Difficulty in recovering accurate geometric information. Inability to measure feature dimensions Time consuming to manually process data. Accuracy depends on user recognizing elements in photos/videos. | Tsai et al., (2009), Wang et al. (2010), Balali et al., (2016), Zhou et al. (2013) |
| Integrated GPS/GIS Mapping | Integrated GPS/GIS field data logger is used to record and store inventory information. | Low initial cost. Low data reduction effort. User friendly, Requires less training. High accuracy. | Tedious. Crew exposed to traffic. Long field data collection time. GPS outage problems due to tree canopy. | Caddell et al., (2009), Ravani et al., (2009) |
| Aerial and Satellite Imagery | High-resolution images taken from aircrafts, drones, or satellites are used to extract road inventory data. | Eliminates field work and data collection time. No exposure of crew to traffic. No traffic disruption. Images are compatible with GPS. | Difficult to identify small objects like traffic signs or signals from images. | Ravani et al., (2009) |
| Terrestrial Laser Scanning | 3D point clouds obtained from stationary 3D laser scanners are used to extract roadway inventory data. | High data accuracy. Can operate in day or night. | Long field data collection time. High cost. Large dataset size. Long data processing and reduction time. | Slattery & Slattery (2010), Vincent and Ecker, (2010), Caltrans, (2018) |
| Mobile Lidar | Lidar instrument attached to a vehicle is used to collect 3D point information out of which roadway inventory data are extracted. | Short data collection time. High accuracy | High cost. Large dataset size. Long data processing and reduction time. | Tang and Zakhor (2011), Vincent and Ecker, (2010), Guan et al., (2016), Teo, (2018) |
| Airborne Lidar | Lidar instrument attached to an aircraft is used to collect 3D point information out of which roadway inventory data are extracted. | No exposure of crew to traffic. Short data collection time. | High cost. Large dataset size. Long data processing and reduction time. | Chow and Hodgson (2009), Vincent and Ecker, (2010) |

### 2.3.5. Extracting Image Features with CNNs

CNN was created primarily to classify images. Its notion of parameter sharing allows a model to learn fewer parameters than a traditional neural network. CNN also has a series of layers (Sainju & Jiang, 2020). In this study, to map and extract features from Google Street View imagery, a model spatial linear pattern using LSTM was presented. An LSTM model was utilized on a sequence of image data retrieved by the CNN model to model the spatial linear structure on a roadway network path. The suggested model's middle component was built around a retrained CNN model for extracting low-dimensional features from individual images. This LSTM is a recurrent neural network (RNN) that uses gating functions to prevent the problems of exploding and disappearing gradients. The structure in Figure 2-3 shows the proposed model used.

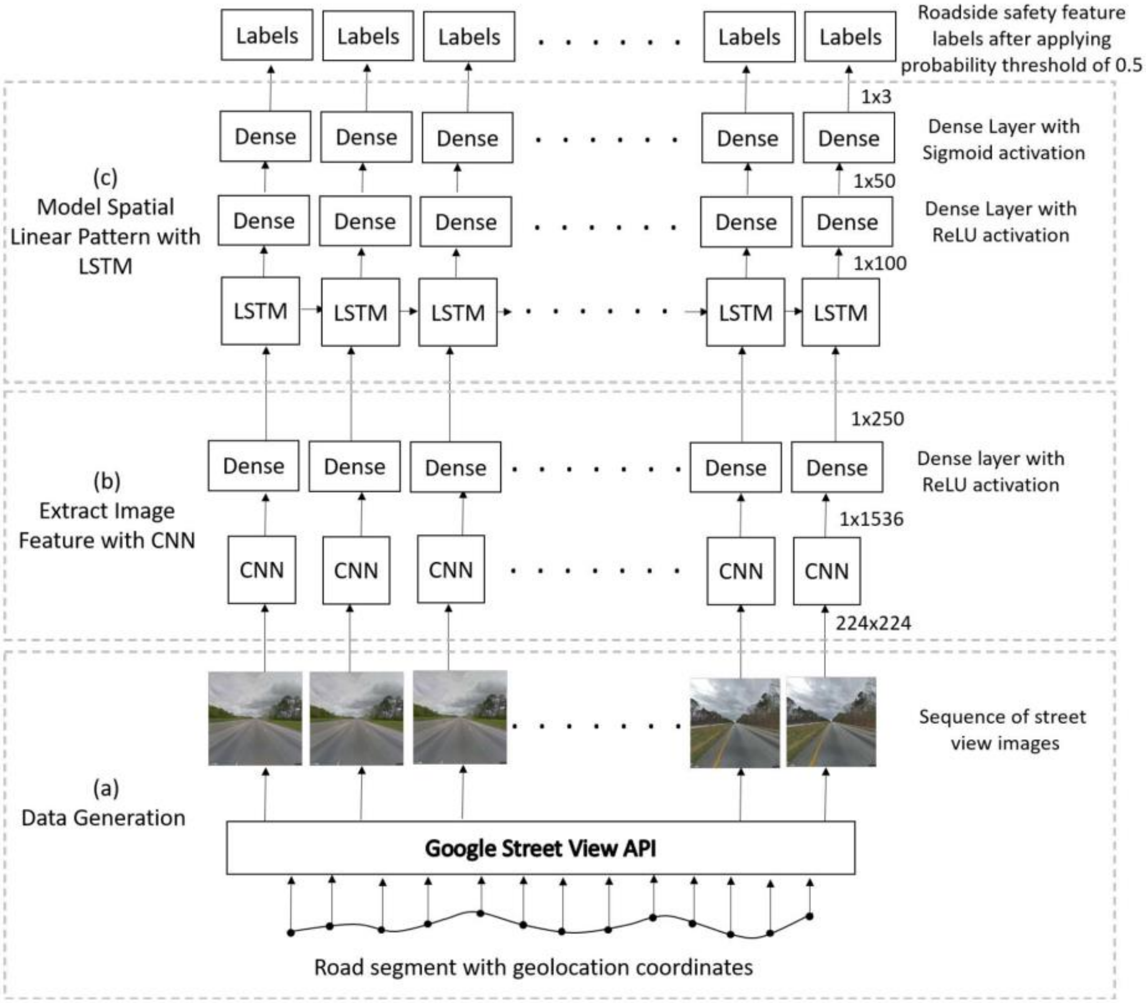

Labels
Roadside safety feature labels after applying probability threshold of 0.5
1x3
Dense
Dense Layer with Sigmoid activation
1x50
Dense Layer with ReLU activation
(c) Model Spatial Linear Pattern with LSTM
1x100
LSTM
1x250
Dense
Dense layer with ReLU activation
(b) Extract Image Feature with CNN
1x1536
CNN
224x224
Sequence of street view images
(a) Data Generation
Google Street View API
Road segment with geolocation coordinates

**Figure 2-3:** Overall Framework of the Deep Learning Model

### 2.3.6. Data Preparation for Model Training and Evaluation

Quality imagery data are needed to train deep learning models to detect the various roadway geometric features. The procedure for the image preparation is shown in Figure 4. The flowchart in Figure 2-4 frames the issue with a focus on school zone identification and image annotation.

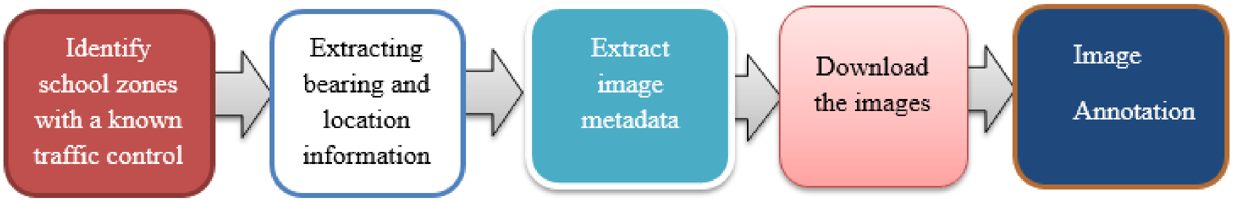

Identify school zones with a known traffic control
Extracting bearing and location information
Extract image metadata
Download the images
Image Annotation

**Figure 2-4:** Example Data Preparation Flowchart

### 2.3.7. Linear Referencing System (LRS)

Using LRS, many of the highway qualities maintained by state DOTs are linked to a physical location on the roadway network. Locations of roadway features like bridges, guardrails, signs, and changes in roadway geometry like the number of lanes and pavement condition are measured using an LRS as a linear distance along a defined route from a given reference point.

A physical feature such as a junction or mile marker signage, a virtual point such as a state or county boundary, and the start point of a state numbered route are all possible choices for the reference points. Almost any roadway feature can be located within the precision of the distance measurement by selecting the reference point and measured distance.

State DOTs have been using linear references for more than 40 years, and it predates the deployment of GIS technology by more than 20 years. However, many State DOTs still save roadway inventory data on mainframe computers and have just lately began converting these databases to a GIS enterprise environment, allowing the data to be linked and shown alongside a State DOT's geospatial roadway network (Spear et al., 2010).

### 2.3.8. Field Cameras/Ground Imagery

Kargah-Ostadi et al. (2020) collected roadway imagery data using a camera mounted on a vehicle. The captured imagery was divided into training and test datasets. These were used to train and test the deep learning object detection model. Convolutional Neural Network (CNN) deep learning model based on MobileNetV2 SSDLite was trained using a pre-processed image data. The trained model created a multiscale feature map for each image made up of cells and a collection of bounding boxes and scores (probabilities) when there was a presence of traffic signs in the boxes. The bounding box with the most overlap and highest score was selected. GPS coordinates and bounding box coordinates assigned to each image from the survey were used to triangulate the local information about the traffic sign.

### 2.3.9. Photo/Video Logging

Jalayer et al. (2015) used photo/video log method to collect roadway inventory data. Geo-tagged digital photos and videos were collected using the Red Hen video mapping system. The collected data were imported into Arc Editor and ArcView, extensions of the ArcGIS software. The digital motion pictures contained GPS locations for the roadways where they were taken. A point feature class was created using the GPS location where each video or picture was taken. Roadside elements were identified from the videos by a user when played.

### 2.3.10. Lidar and Aerial Imagery

In a study by Shamayleh and Khattak (2003), aerial imagery of two meters resolution and Lidar data in ASCII comma delimited text file format with 6 inches accuracy was used to collect the roadway data. Height information was added to Lidar point shapefiles using ArcView and the datapoints were finally converted into Triangular Irregular Networks (TIN). Lidar bounds and aerial images were merged to start the entire analysis. ArcView's object obstruction tool was used to identify objects obstructing the line of sight of an observer. Stopping sight distance was obtained by observing the line of sight drawn. The line of sight was drawn, and green line segments showed visible terrain while red line segments showed obstructed terrain. The side slope and contours were obtained by using the identifying tool and contour tool in ArcView. The grade was calculated by taking a difference of elevation of two ends of a segment. Yan et al. (2016) extracted roadway markings from Lidar point clouds using the scan line method. Using their timestamps, the algorithm orders Lidar point clouds sequentially and organised into scan lines using scanner angle. The height difference between trajectory data and roadway surface was used to extract seed road points, which were later used to extract full road points. A line was fitted through the seed points and all other points along the scan line. Only points that fall within a specified threshold of the line were kept and classified into road marking or asphalt points based on their intensity. To reduce the noise in data, a dynamic window median filter was used to smoothen the intensity values. Edge detection and constraints method was used to extract roadway markings.

### 2.3.11. Object Detection and Tracking

Roadway user detection is the key component of vision-based traffic methods. These can be categorized into point detection, machine learning, and deep learning algorithms. Most of the point detection and tracking algorithms, such as KLT point tracking or blob analysis, are unsupervised algorithms and can be applied rapidly without any training data set. These algorithms represent the early stage of computer vision utilization in transportation as they can be found in vehicle detection and surveillance technologies used in ITS (Mimbela & Klein, 2000). Also, they are easy to cluster with simple algorithms like k-means algorithm (Ke et al., 2015). Speed clustering of the interest points can be used for background subtraction as well (Ke et al., 2017). However, these algorithms are sensitive to the complexity of the image such as shadows, roadway facilities, and adjacent vehicles. Their performance is low in congested traffic conditions.

Most of the supervised vehicle detection algorithms come from the Viola-Jones face recognition algorithm (VJ) with hand crafted features (i.e., Haar, Histogram Oriented Gradients (HOG), or SIFT Local Binary Pattern) (Viola & Jones, 2005). Moreover, Integral Channel Features (ICF) (Dollár et al., 2012) and its improved version Aggregated Channel Features (ACF) (Appel et al., 2014) also originated from the VJ-based object detection algorithms. Basically, the cascade function in these algorithms needs to be trained with positive and negative images. Training data set consists of positive and negative images to emphasize what the interest object is and what is not. For example, in vehicle detection, positive images include vehicle(s) and negative ones include only the roadway with trees or other side features. The larger the size of the training dataset, the higher accuracy on the vehicle detection, similar to face recognition. When an object has all the features according to a running vehicle training set, it will be detected as a vehicle and a bounding box will be drawn around it (Xu et al., 2016). They can be used to detect vehicles in congested traffic flow; however, there is still an overlapping problem. A study (Kim et al., 2019) used a 40% overlapping threshold for ACF-based vehicle detection. If the overlapping area was greater than 40 % of the total detected area, only the vehicle with higher detection score was retained. Haar like features and HOG were also used for pedestrian detection and tracking from UAS-obtained images (Ma et al., 2016).

More recently, deep learning-based convolutional neural networks (CNN) have shown an outstanding performance in object detection (Tang et al., 2017; Vattapparamban et al., 2016; Xie et al., 2018; Xu et al., 2017). These algorithms have been improved with region-based CNN (R-CNN) (Girshick et al., 2012), Fast R-CNN (Girshick, n.d.), and finally with Faster R-CNN (Ren et al., 2017). CNN algorithm basically extracts region proposals as candidate locations for the objects followed by the computation of the CNN features. Faster R-CNN improves performance by using a Region Proposal Network (RPN) in addition to the object detection network. A recent study (Kim et al., 2019) proposed a detailed comparison on the vehicle detection and tracking performances of deep learning (faster R-CNN) and machine learning algorithms (ACF). Their findings indicated that faster R-CNN outperforms ACF. Therefore, the algorithm to run the analysis should be selected carefully. Depending on the purpose of the analysis, the output of the vehicle detection and tracking algorithms can be speed, volume, or vehicle trajectories. Unlike point tracking algorithms, vehicle detection algorithms can classify vehicles as well.

### 2.4. Findings

Findings indicate that, the existing methods provide accurate results; however, they have a lot of setbacks. For example, Lidar and laser scanning methods are costly. Due to the high density of data, it requires considerable number of data processing time. Traditional surveys, integrated GPS/GIS methods, and photo/video logging methods expose crew members to traffic risks. Data collection times are long and data processing has a lot of human interaction. New and cheaper methods for collecting RCI data such as using computer vision and artificial intelligence should be employed to collect data.

A study by Kwayu et al. (2022) appear to be valuable here. In this study, approximately 85% of the total automated photos were randomly selected for model training and 15% for model testing from a scalable deep learning framework where MIRE intersection control types were extracted from panoramic images (Kwayu et al., 2022). It had a 98% recall rate on average. The model losses and learning rate were the most closely examined performance measures. The learning rate is a metric that measures how quickly the model learns from the data. As the number of training steps increases, the model loss should reduce and the learning rate should increase (Kwayu et al., 2022). A total loss of 1.5 was found acceptable in another similar study (Campbell et al., 2019). As a result, when the model had learned from 6,000 training steps, the training process ended. The model was trained for approximately 4 GPU hours in total. In the same study where GIS was used with google street view imagery, the GIS analysis was successful in selecting relevant places that observed 90.80% of the intersection signage within the study area, based on the quantitative analysis. The practicality of using street view photography to detect street signage is determined by several factors. Above all, a geographic area must be covered by Google Street View, and the imagery's currency must be current. Furthermore, urban transportation networks tend to receive more complete coverage than rural transportation networks, resulting in a disparity in the amount of data available to different regions (Campbell et al., 2019).

Results from most state DOTs clearly indicated the need for more advanced techniques in data collection, and image processing and classification. For example, New Hampshire DOT uses a Linear Referencing System coupled with the AASHTOWare for data management, but this clearly indicates how most of elements in their database were replicated from other sources and there is still a need to expand their data collection methodology. This is since the data needed for AASHTOWare to make the necessary traffic safety analysis was not efficient.

# 3. TASK 2: GEOMETRIC DATA EXTRACTION

## 3.1. Task Description

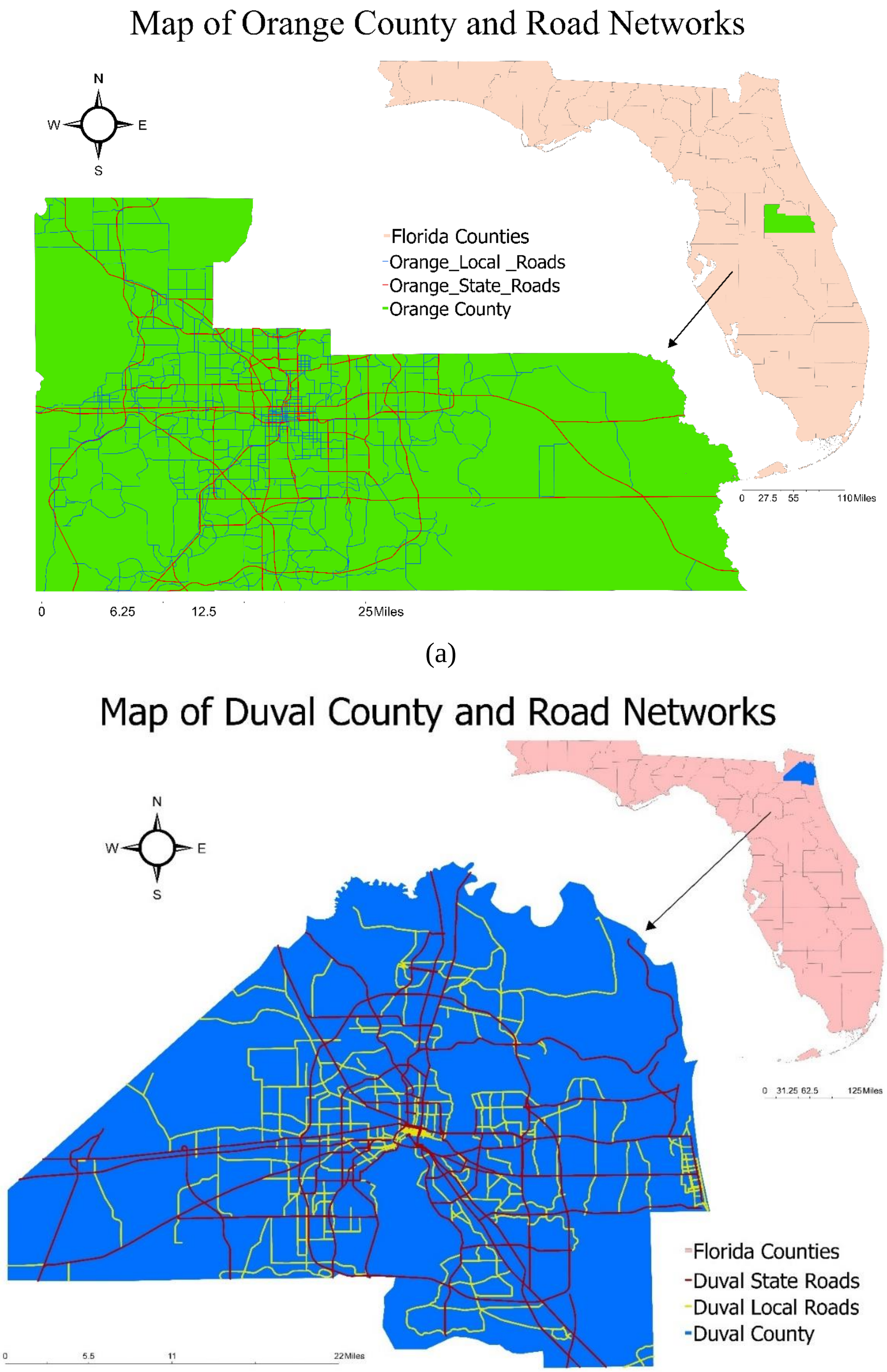
In this task, the research team developed and evaluated the machine learning models used to detect and extract roadway geometry features such as school zones and turning configurations. Based on the review of literature that has been conducted for the purpose of discovering published information that can help inform, shape, or guide the conduct of this research project, the models were developed and implemented throughout the State of Florida to extract school zones and turn configurations on both on and off state highway system roadways. The main focus of this task was to develop a machine learning model to extract roadway geometry features from high resolution aerial images. For this purpose, machine learning-based object detection models were developed for detecting school zones and turning configurations (e.g., left only, right only, and center lanes) from high-resolution aerial images. Also, the research team outlined various methods and approaches used in developing this model and extracting geometric data at intersections and data needs to facilitate the analysis.

## 3.2. Study Area

This section describes the study areas used in this research. Two areas in Florida were studied in this research (a) Orange County, Florida and, and (b) Duval County, Florida (Figure 3-1). They were both considered due to their diversity in roadway infrastructural development. Also, ground truth data collected in Leon County, Florida has been used for validating the models.

Orange County, Florida, the home for the City of Orlando, has been selected as the case study area for detecting school zones. The county has 1,003 square miles of total area and its borders are Brevard County on the east, Seminole and Lake Counties on the north, Lake County on the west, and Osceola County on the south (US Census, 2020). Within the county, land-surface heights are less than 250 feet, and the geography of the region differs significantly from west to east. Light commercial, residential, and open irrigated space such as golf courses and school grounds are the primary land uses in the study area (OCPS, 2022). The estimated population of Orange County is over 1.42 million as of 2020 according to the U.S. Census Bureau (2020). There are 264 schools in the Orange County Public Schools system, serving 199,089 students. 80% of the county's students are from the underrepresented communities. The county has 131 preschools, 170 elementary schools, 89 middle schools, and 56 high schools (OCPS, 2022).

On the other hand, Duval County has been selected as the case study area for detecting turning lanes and configurations. The City of Jacksonville is the county seat for Duval County, which is in Northeast Florida. The county has 918 square miles of total area and shares border with Clay, Baker, Nassau, and St. Johns Counties in Florida (US Census, 2020). According to the U.S. Census Bureau (2020), Duval County has a population of 995,560. Duval County has over 587 miles of local roadways and 503 miles of state roadways.



(a)

(b)

**Figure 3-1:** Map of (a) Orange County and (b) Duval County Florida, with the roadway network

### 3.3. Materials and Methods

The choice of method used to collect roadway inventory data depends on factors such as data collection time (i.e., data collection, reduction, and processing), cost (i.e., data collection, and reduction), and accuracy, safety, and data storage requirements. In this study, our aim was to create a deep learning object detection model to detect school zones and turning lane markings from high resolution aerial images in Orange County and Duval County.

#### 3.3.1. Data Description

Aerial imagery is archived by the Surveying and Mapping Office of the Florida Department of Transportation (FDOT). These well-indexed, georeferenced photos of all 67 Florida counties are preserved across a period of years, with file names that include the three-letter county code, the year the image was taken, and the tile number. The high-resolution aerial images used for the study were obtained from the Florida Aerial Photo Look-Up System (APLUS). This archive is accessible to the public via APLUS, which is run by the FDOT Surveying and Mapping Office (FDOT, 2020). A small number of photos can be downloaded from this platform's website, but massive datasets—covering an entire county or perhaps an entire state—can only be obtained by sending a request and a letter or by supplying an external drive with enough storage capacity. With a combined size of 17.73 GB, the most recent high-resolution pictures (as of December 2019) for the Florida counties of Pinellas, Gulf, Santa Rosa, Hillsborough, Duval, Broward, Orange, Escambia, Miami-Dade, and Leon were retrieved for this study. The aerial images used for model training and detection have a resolution ranging from 1.5 ft down to 0.25 ft. Because our model is based on this type of resolution, any imagery that falls within this resolution or is higher can effectively be utilized to detect school zones using our model. The majority of the photographs were in the 0.5 ft/pixel (0.15 m/pixel) resolution with a size of 10,000 × 10,000 and 3-band (RGB) image format, while the precise resolution varied depending on the county. Additionally, the photos are supplied in MrSID format, which allows for GIS projection on a map.

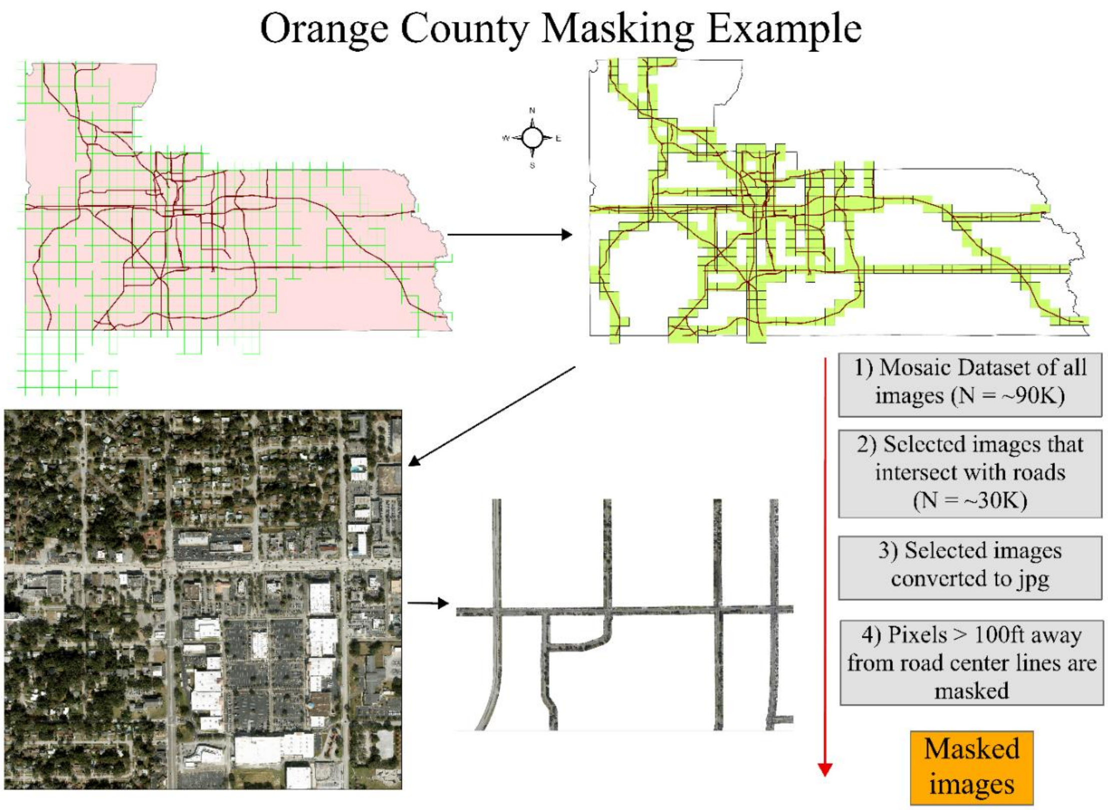
Another source of GIS data is the FDOT Transportation Data and Analytics Office. This information has been divided into four main categories: (a) data on designated roadways, (b) data on roadway features, (c) data on traffic, and (d) data on bicyclists and pedestrians. This study focused on school zones found on county- or city-controlled roads as well as those found on state highway system roads. Interstate roadways were removed from the state roadway, and all centerlines in the county- and city-controlled roadways shapefile were combined. State highway system roads are described in this study as ON System Roadways or state roads while the county- or city-controlled roads are described as OFF System Roadways or local roads according to FDOT classification. It should be noted that the FDOT's GIS data can assist in obtaining several geometric data points necessary for the mobility and safety performance evaluations; however, the dataset lacks information on the positions of school zones on state and local roadways. As such, the main goal of this project was to use an advanced object identification model to compile an inventory of school zone markers and turning lane markers precisely left only, right only, and center lanes on both state and local roadways in Orange and Duval counties, Florida.

#### 3.3.2. Pre-processing

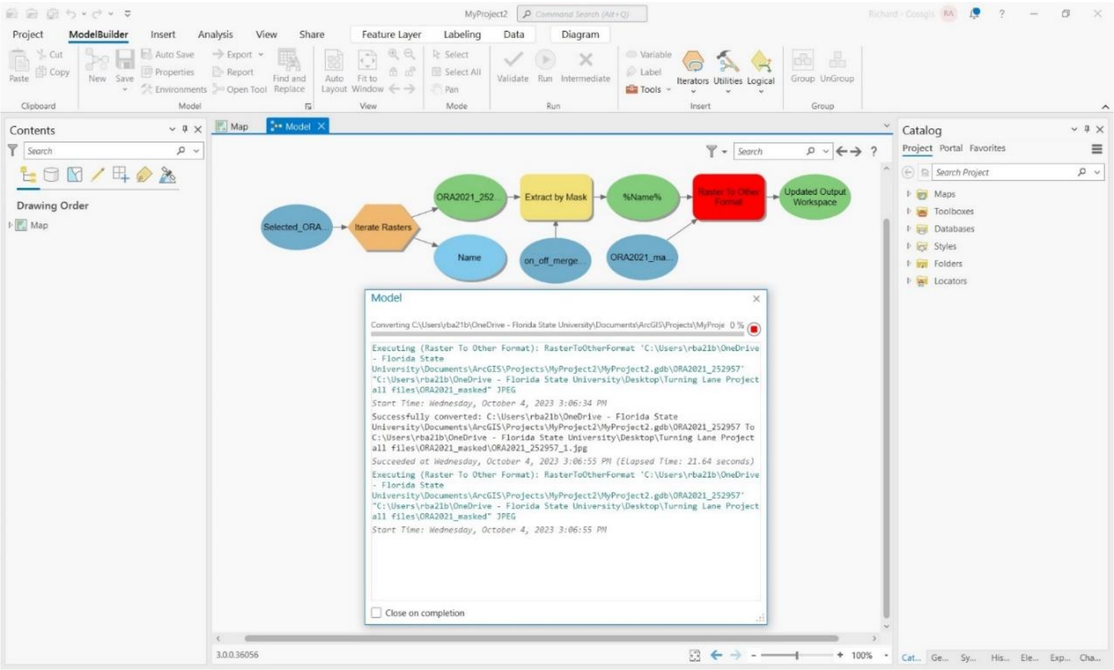
Aerial images were obtained from the Florida Aerial Photo Look-Up System (APLUS). This archive is accessible to the public via APLUS, which is run by the FDOT Surveying and

Mapping Office. After obtaining aerial images, one important task is to mask the images. Due to the amount of data and the difficulty of the object recognition procedure, preprocessing is a necessary step. In general, our method chooses and discards any pictures that do not cross a roadway centerline, and then it masks out any pixels that are not encircled by a buffer zone. In this approach, the number of photos was decreased from 90K to 30K, and the image masking model excluded objects that were 100 feet away from state and local roadways. Before the images were masked, the roadway shapefile which were used, were buffered to form polygons having overlapping boundaries dissolved. Using this layer as a reference, aerial images were iterated, and intersecting regions of the aerial images were cropped. During the masking process, pixels that fell outside the boundary of the reference layer were removed. The cropped images, which consequently had smaller number of pixels, were mosaiced together to form a single raster file. This new file was smaller and was easier to handle in a mosaic format for all forms of raster analysis or data processing.

Figure 3-2 illustrates the preprocessing strategy in detail. All the photos from the chosen counties (n = about 90K) were initially imported into a mosaic dataset using the ArcGIS Pro program. Multiple geocoded photos were managed and shown using mosaic databases. Additionally, mosaic datasets enable the intersection of additional geocoded vector data to pick picture tiles depending on location. For instance, a subset collection of the photographs (n=30K) was created by selecting and extracting individual images that comprise a portion of the roadway centerline. Furthermore, an automatic picture masking process was created using the ArcGIS Pro ModelBuilder interface. As shown in Figure 6b, the tool repeatedly goes through a folder of photographs, applies a mask to each image based on a 100-foot buffer around the centerlines of the road, and then saves the masked images as JPG files for the object recognition method.


Orange County Masking Example
N
W
E
S
1) Mosaic Dataset of all images (N = ~90K)
2) Selected images that intersect with roads (N = ~30K)
3) Selected images converted to jpg
4) Pixels > 100ft away from road center lines are masked
Masked images

(a)



Model

Converting C:\Users\rba21b\OneDrive - Florida State University\Documents\ArcGIS\Projects\MyProje 0 %

```
Executing (Raster To Other Format): RasterToOtherFormat 'C:\Users\rba21b\OneDrive
- Florida State
University\Documents\ArcGIS\Projects\MyProject2\MyProject2.gdb\ORA2021_252957'
"C:\Users\rba21b\OneDrive - Florida State University\Desktop\Turning Lane Project
all files\ORA2021_masked" JPEG
Start Time: Wednesday, October 4, 2023 3:06:34 PM
Successfully converted: C:\Users\rba21b\OneDrive - Florida State
University\Documents\ArcGIS\Projects\MyProject2\MyProject2.gdb\ORA2021_252957 To
C:\Users\rba21b\OneDrive - Florida State University\Desktop\Turning Lane Project
all files\ORA2021_masked\ORA2021_252957_1.jpg
Succeeded at Wednesday, October 4, 2023 3:06:55 PM (Elapsed Time: 21.64 seconds)
Executing (Raster To Other Format): RasterToOtherFormat 'C:\Users\rba21b\OneDrive
- Florida State
University\Documents\ArcGIS\Projects\MyProject2\MyProject2.gdb\ORA2021_252957'
"C:\Users\rba21b\OneDrive - Florida State University\Desktop\Turning Lane Project
all files\ORA2021_masked" JPEG
Start Time: Wednesday, October 4, 2023 3:06:55 PM
```

□ Close on completion

(b)

**Figure 3-2:** (a) Preprocessing approach and (b) implementing automated image masking model used in the fourth step of preprocessing

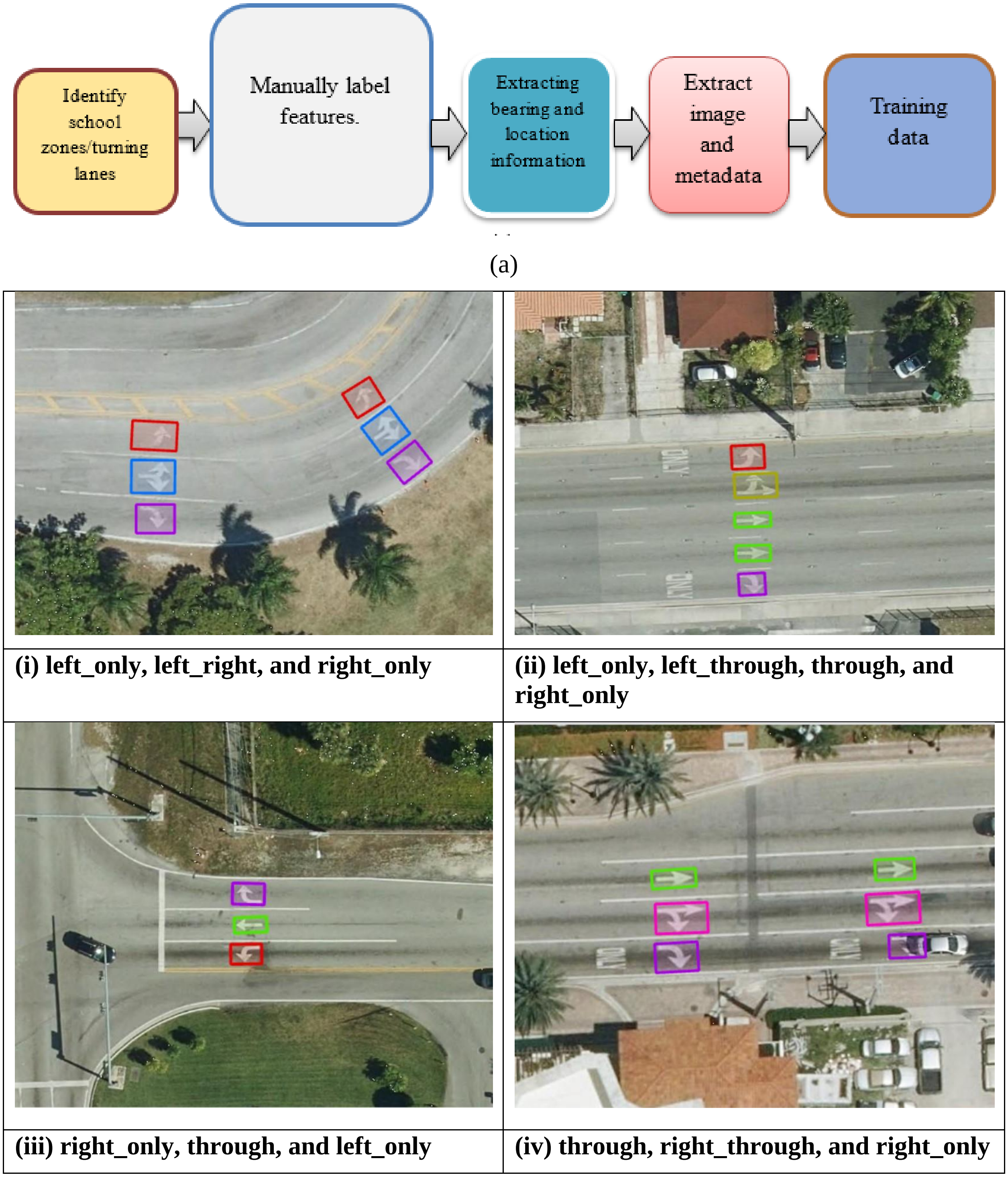
### 3.3.3. Data Preparation for Model Training and Evaluation

#### 3.3.3.1. *School Zone Model*

Quality imagery data are needed to train deep learning models to detect various school zone features. The procedure for the image preparation is shown in Figure 3a which illustrates the process from school zone identification to image or training data extraction. A lot of time and effort were spent to create the training data for the model creation since a greater percentage of the model's performance relies on the amount, quality and diverseness of the training data used for training the model. Similarly, extra time and work was required during the model training process where different parameters were tested, and the optimum was selected to train the model. This is illustrated in the larger box size for "Manually label school zones" and "training data" in Figure 3-3a. The labels used for describing the training data were "schoolzones" and "not_school" which were class 1 and class 0 respectively. The labels were made up of rectangular bounding boxes drawn around the school zone and not school zone markings.

#### 3.3.3.2. *Turning Lane Model*

Two different multi-class models were developed to detect the turning configurations (Figure 3-3b). Firstly, a 12-class object detection model was developed, followed by a 4-class object detection model. Two different forms of training data with different classes were prepared for the study. The first training data which was used to train the first turning lane model had 12 classes which were as follows: "left_only", "right_only", "left_through", "right_through, "through", left_right_through", "bicycle", "center", "left_right", "merge", "u_turn", and "none". These were class 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and 12 respectively. The labels used for describing the training data for the second model were "left_only", "right_only", "center", and "none", which were classes 1, 2, 8, and 12, respectively (Table 3-1). It is important to note that object detection models perform well when trained using clear and distinct features. The observed difference between the left only and right only turning markings is commonly a lateral inversion of the other which is less distinct and therefore its' detection performance by the model is anticipated to be relatively low. To ensure uniqueness, the labels used to train the center lane class contained both left arrows facing each other from different travel directions. A similar idea was employed in labelling all other features to maintain uniqueness in the training features. The metadata of the exported labels were in the pascal visual objects format. The input mosaic data was made up of high-resolution aerial images of the entire State of Florida with tile size of 5000 x 5000 sq ft.


Identify school zones/turning lanes
Manually label features.
Extracting bearing and location information
Extract image and metadata
Training data

(a)

**(i) left_only, left_right, and right_only**

**(ii) left_only, left_through, through, and right_only**

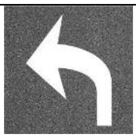
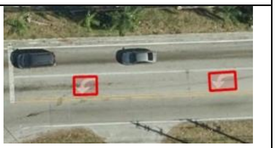
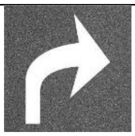
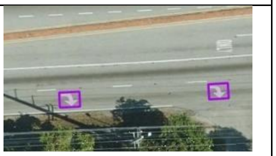
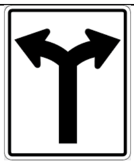
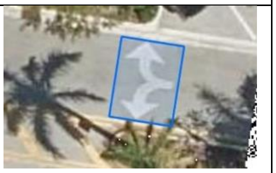
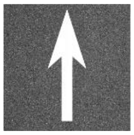
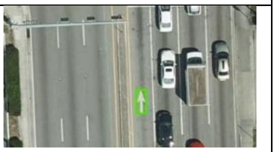
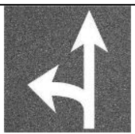
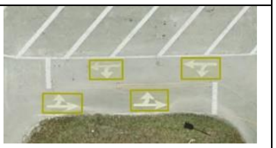
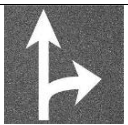
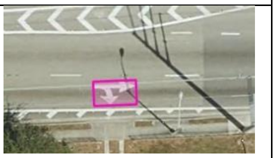
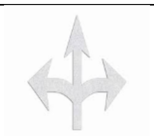
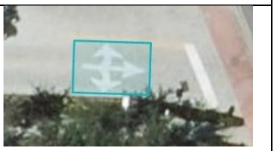
**(iii) right_only, through, and left_only**

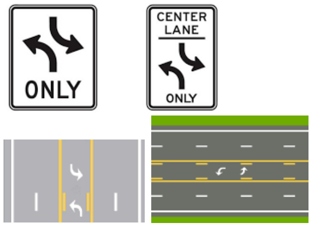
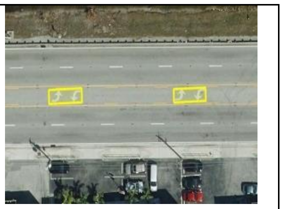
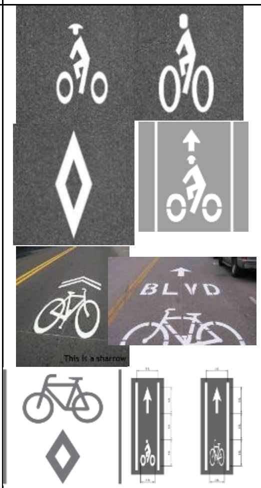
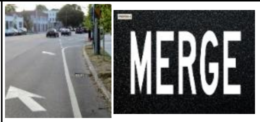
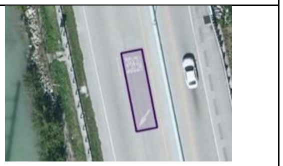
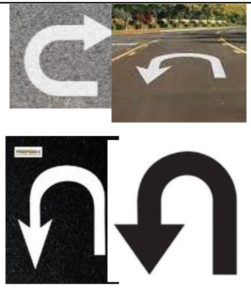
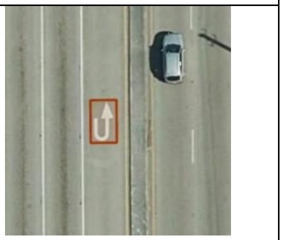
**(iv) through, right_through, and right_only**

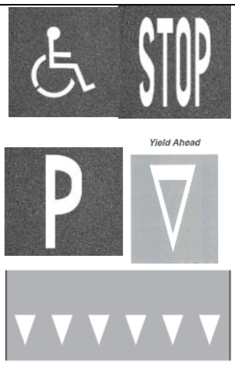
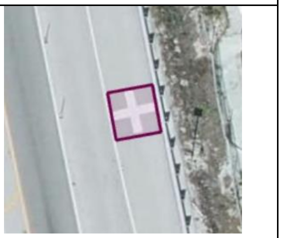
(b)

**Figure 3-3:** (a) Model training data preparation framework and (b) training data examples for turning lane with bounding boxes.

**Table 3-1:** Turning lane model training data description

| ID | Class name | Description | Example/Picture | Training Example |
|---|---|---|---|---|
| 1 | left_only | Left only |  |  |
| 2 | right_only | Right only |  |  |
| 3 | left_right | Left and Right | <br> |  |
| 4 | through | Through |  |  |
| 5 | left_through | Left and Through | <br> |  |
| 6 | right_through | Right and Through | <br> |  |
| 7 | left_right_through | Left Right and Through | <br> |  |

| | | | | |
|---|---|---|---|---|
| 8 | center | Center lane (left turn possible in both direction or two way) |  |  |
| 9 | bicycle | Bicycle lane |  | <br> <br> |
| 10 | merge | Merge lane |  |  |
| 11 | u_turn | U turn |  |  |

| Table 3-1, continued | | | | |
|---|---|---|---|---|
| 12 | none | None<br>1. Yield<br>2. Stop<br>3. Parking<br>4. Speed limits<br>etc |  |  |

### 3.3.4. YOLOv3: School Zone and Turning Lane Detection Model

The neural network model known as You Only Look Once (YOLO) is mostly employed for real-time object identification. When compared to other object identification models like Region-based Convolutional Neural Network (R-CNN) or Faster R-CNN, YOLO's quickness is by far its biggest advantage. In other words, YOLO outperforms R-CNN and Faster R-CNN architectures by 1000 and 100 times, respectively (Redmon and Farhadi, 2018). This is largely because other object identification models classify potential regions first, then identify the item(s) based on the classification probability of those regions, requiring hundreds of network evaluations to be performed on a single picture. On the other hand, the YOLO model may make predictions based on the entire context of the image and look for the entire image using just one network analysis. The initial version of the YOLO object identification method was released in 2016 (Redmon et al., 2016), the second version, YOLOv2, in 2017 (Redmon and Farhadi, 2017), and the third version, YOLOv3, which improved multi-scale predictions, in 2018 (Redmon and Farhadi, 2018). YOLOv4 (Bochkovskiy et al., 2020) and YOLOv5 (Jocher et al., 2020) were published in 2020 and 2022, respectively. YOLOv3 was used in this investigation since our dataset does not have a scalability problem and because it is simple to apply using the available sources (Figure 4). Unlike other detection models, YOLOv3 uses Darknet-53 as its backbone. Darknet-53 is a deeper version of Darknet-19 which was the backbone of YOLOv2. The architecture of the backbone has 53 convolutional layers which improves the model’s accuracy and speed (Tsang, 2019), making the feature extractor perform better and 2x faster than ResNet152 (Redmon and Farhadi, 2018). Figure 3-4 shows images that are produced at the end of the detection layers in YOLOv3, depicting the three-scale detection process. This involves applying a 1x1 detection kernel on feature maps located at three different areas and sizes within the network.

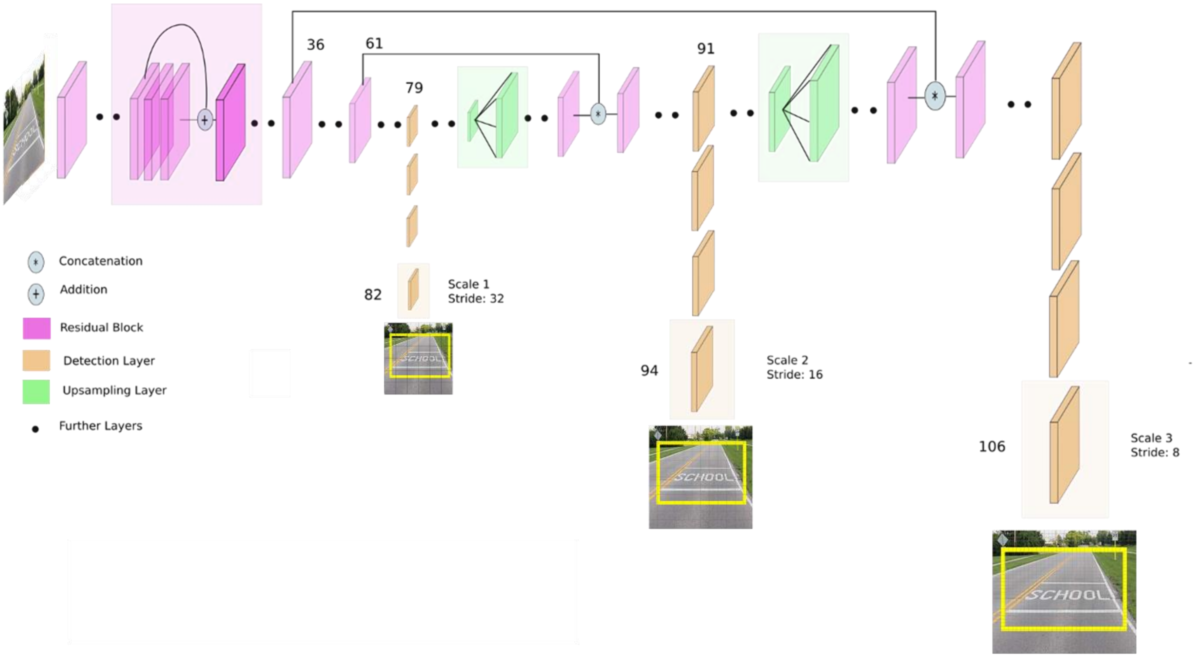


**Figure 3-4:** YOLOv3 network architecture school zone example adapted from (Kathuria, 2018)

### 3.3.4.1. *School Zone Detector*

The school zone model was trained with manually labelled 13,208 school zone and not-school-zone classes on the aerial images using the Deep-Learning Toolbox in ArcGIS Pro. The school zone class was made up of 7,056 features, which is 53.4% of the training data, whereas the not-school-zone class contained 6,152 features which is the remaining 46.6% of the training data. The not-school-zone labels were made up of randomly selected roadway markings that describe other features than school zones including railroad crossings, stop, slow, only, and speed limit markings. The school zone class was labeled as 1 while the not-school-zone class was labeled as 0 in the training data. The two classes clearly distinguished between the school zone features and not-school zone features observed in an input image by the detector. This clear classification improves the model's detection performance. Based on these classes, the output data was categorized into school zones and not-school zone using a class value field. The training data consists of the school zones and not-school-zones from Miami-Dade (8,096), Pinellas (4,060), Escambia (80), and Leon (972) counties. For a fair model performance, these counties were selected due to the variations in image resolutions, demographics, and roadway infrastructure development for the training. These counties selected also had varying school zone markings on the roadways.

Note that demographics provide valuable statistical information about human population in terms of age, income, sex, education, race, and others. These variables, which can also be used to describe the needs of a community, consequently, provide vital information on the highly populated communities as well as communities that have a substantial number of schools to serve their residents. Information about underserved neighborhoods with limited infrastructural development and less schools available can serve as a valuable resource in obtaining training data for the model since a large chunk of school zone markings with varying characteristics were needed to efficiently train the model. Demographical data pointed us to specific areas with a lot of school zone markings and vice versa.

The school zone model's tunable parameters and hyperparameters include the learning rate, input image size, number of epochs, batch size, anchor box size and ratios, and training and test data percentages. The ML model evaluation metrics were illustrated using the validation and training loss graph (Figure 5a). The validation loss and mean average precision were computed on the validation set, which is made up of 30% of the input training dataset. The object detection parameters that have a high impact on the object detection are the batch size, the learning rate, and the training epoch. The learning rate represents the rate at which the model learns new information about the training data by overwriting the existing information with newly acquired information during the training process. Choosing an ideal learning rate strikes a compromise between precision and convergence speed. A model can be trained effectively, with excellent convergence characteristics, and with the greatest level of accuracy by using an optimal learning rate. Therefore, an optimum learning rate of 0.0003 was used to train the model. The batch size describes the number or bundle of training samples or images selected and processed for training each iteration. The selection of batch size depended on factors like the size of dataset, model's complexity, and the available computer hardware's resources. For instance, with a larger batch size, more data can be processed in parallel, and the training process is faster. However, this requires more computer memory. Smaller batch size, on the other hand, increases randomness in training data selection during the training process. This enhances the model's performance on new data and therefore improves predictions. Since the developed school zone model has two classes with less data complexity, a smaller batch size of 16 was used in the training process to improve model's performance. Also, the anchor box represents the size, shape and location of the object being detected. 9 anchor boxes were used to train the model. The epoch number is the number of iterations the model will be trained. It describes the number of times the training dataset is passed forward and backward through the neural network once. The percentage of training data actually used to train the model was 70%. This dataset is randomly chosen to train the model.

A 30% split of the training dataset was used to assess the detection model's performance. In other words, 3,962 randomly selected school zone and non-school zone photos were used to assess the model's accuracy rather than for training. The size of the training dataset was mainly considered when determining the training and test data split. With a training dataset > 10000, a 20%-30% validation size provides enough randomly sampled data to test the model's performance. As a valid prediction, a default 50% overlap of the label and detection bounding boxes was accepted. Following that, recall and precision were determined as the true prediction rate among the original labels and all other predictions, respectively. In YOLOv3, the likelihood of an input belonging to a certain label is calculated using individual logistic classifiers instead of the SoftMax function which was used in previous versions (Redmon et al., 2016; Redmon and Farhadi, 2017). During the model training, a binary cross-entropy loss was used for each label to calculate the classification loss, rather than using mean square error. Therefore, logistic regression was used to predict both object confidence and class predictions. This approach reduces the complexity of computations involved and improves the model's performance (Redmon and Farhadi, 2018). The graph of the train-validation loss is shown in Figure 3-5a. The validation loss examined how well the model fits new data, whereas the training loss evaluated how well the model fits training data.  A high loss indicates that the product of the model has errors while a low loss value shows there are fewer errors present in the model. Since the accuracy of the developed school zone model was 0.88, we can conclude that the developed detector performs quite well.

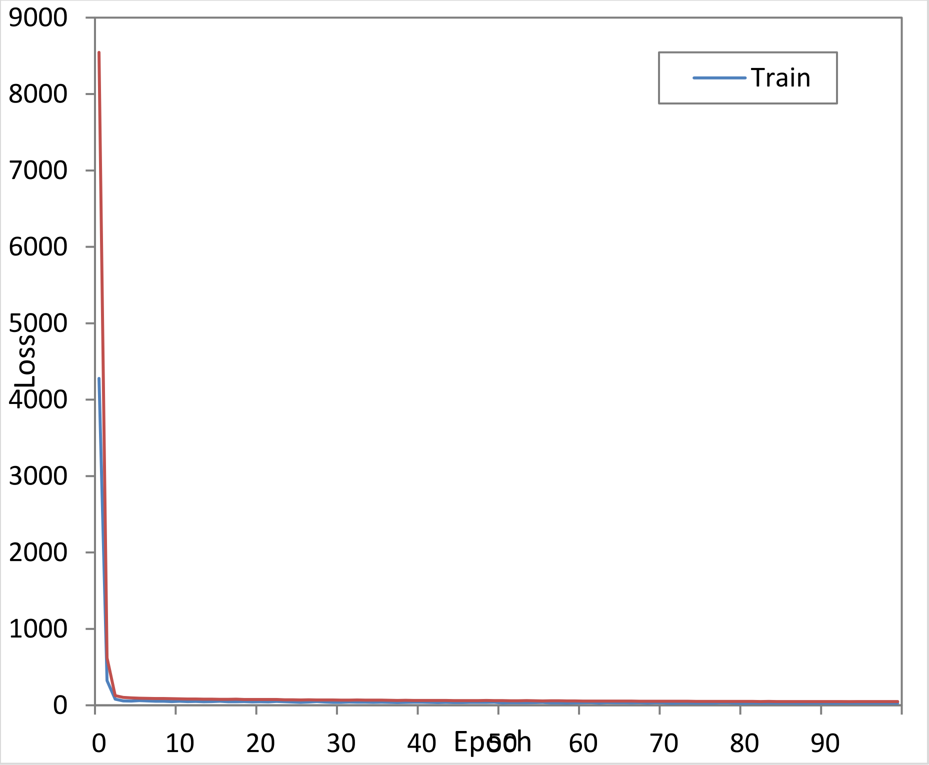
#### 3.3.4.2. *Turning Lane Detector*

The 12-class and 4-class turning lane models were trained with manually labelled 23,669 and 8,241 features, respectively, on the aerial images of Miami-Dade, Gulf, Santa Rosa, Hillsborough, and Broward counties using the Deep-Learning Toolbox in ArcGIS Pro. For the 12-class model, the “left_only” class had 3,021 features (12.3% of training data), “right_only” class had 2,179 features (8.9%), “left_through” class had 1,627 features (6.6%), “right_through” class had 1,583 features (6.4%), “through” class had 2,233 features (9.1%), “left_right_through” class had 920 features (3.7%), “bicycle” class had 3,230 features (13.1%), “center” class had 3,043 features (12.4%), “left_right” class had 159 features (0.6%), “merge” class had 2,262 features (9.2%), “u_turn” class had 632 features (2.6%), and “none” class had 2,780 features (11.3%). On the other hand, the 4-class model had 2,178 features (26.4%) for the “left_only” class, 2,060 features (25%) for the “right_only” class, 1,884 features (22.9%) for the “center” class, and 2,119 features (25.7%) for the “none” class. The “none” class labels for the 12-class model were made up of randomly selected roadway markings that describe all other features than the turning lane markings of interest including railroad crossings, stop, slow, only, and speed limit markings. On the other hand, the “none” class labels for the 4-class model were made up of all other visible markings or features except “left_only”, “right_only”, and “center”. Some features which were very common such as “left only”, “right only”, and “bicycle” had higher presence in the training data for the turning lane model with 12 classes. However, rare features such as, “left_right”, “u_turn”, and “left_right_through” had lower presence in the training data set.

Some of these features were duplicated to increase their proportions in the training data set. To avoid model overfitting, bias and limited model generalization as a result of duplication, a data augmentation technique named rotation was utilized to increase dataset diversity and quantity without solely resorting to duplication. Data rotation method was used to randomly rotate the training data features at desired angle producing extra training data in various rotations. This helps the model detect objects in various positions and orientations. This was very useful since the study features such as left only and right only lanes can be observed in all travel directions at each intersection. A 90 degrees rotation was applied to the training data. Finally, 468,176 exported image chips containing 567,876 features were used to train the 12-class model and 144,224 image chips containing 185,560 features were used to train the 4-class model. Please note that a single image chip may contain more than 1 feature.

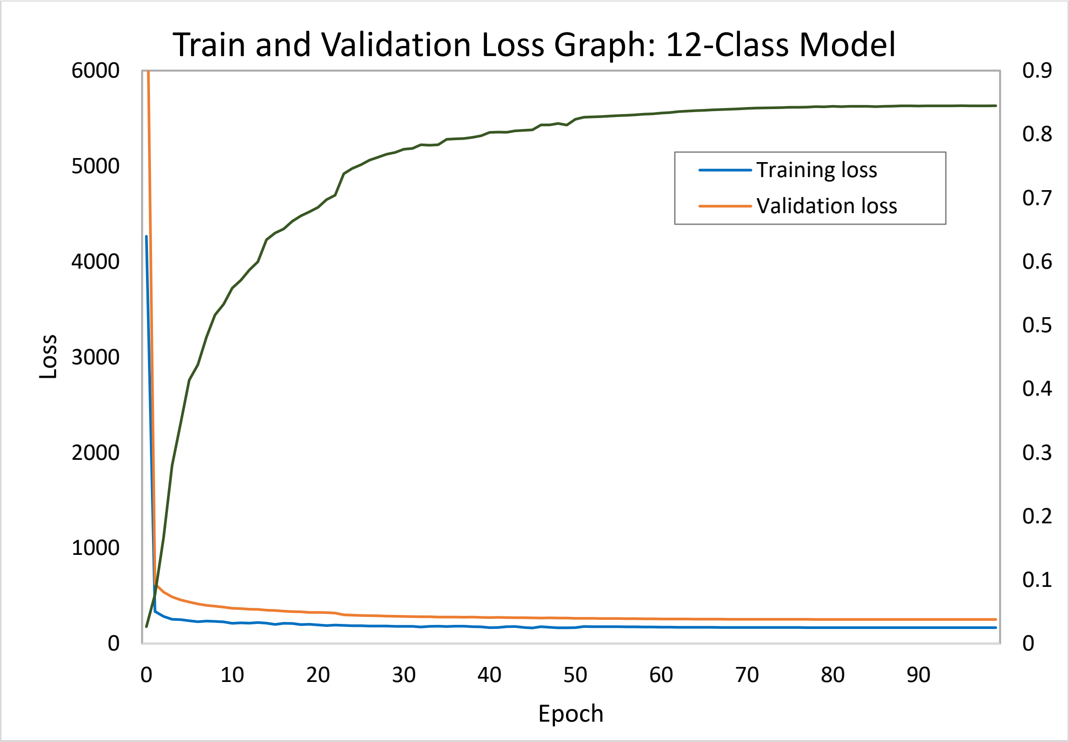
Each class was distinctively labeled in the training data. The classes clearly distinguished between the turning lane features, and “none” features observed in an input image by the detector. This clear classification improves the model’s detection performance. Based on these classes, the output data was categorized into either “left only”, “right only”, “center”, and “none” using a class value field. The training data for the 12-class model consists of turning lane features from Miami-Dade (169,592), Hillsborough (286,584), Santa Rosa (4,708), and Gulf (8,556) counties. That of the 4-class model consists of Broward (6,712), Miami-Dade (112,552), Hillsborough (53,136), Santa Rosa (5,324), and Gulf (7,836) counties. For a fair model performance, these counties were selected due to the variations in image resolutions and roadway infrastructure development for the training. These counties selected also had varying turning lane markings on the roadways.

The ML model evaluation metrics were illustrated using the average precision, validation, and training loss graph (Figure 3-5b and Figure 3-5c). The validation loss and mean average precision were computed on the validation set, which is made up of 30% of the input training dataset. An optimum learning rate of 1.096e-06 and 3.311e-06 was used to train the 12-class model and 4-class model, respectively. Since the developed turning lane models were multi-class models with high data complexity, a higher batch size of 64 was used in the training process to improve model's performance. The percentage of training data was 70%. A 30% split of the training dataset (140,453 and 43,268) image chips were used to assess the 12-class and 4-class models' accuracy, respectively. Similar to the school zone model, a default 50% overlap of the label and detection bounding boxes was accepted as true detection. The graphs of the train-validation loss for 12-class and 4-class turning lane models are shown in Figure 3-5b and Figure 3-5c. From the graphs, it can be noted that the 4-class model had a relatively higher training and validation loss difference with the training loss much lower than validation loss (Figure 3-5c). However, there was very little difference between the training and validation loss values obtained from the 12-class model (Figure 3-5b). This shows that the developed turning lane models fit equally with new data; however, the 4-class model fits better with the training data. Also, the 4-class model has less errors compared to the 12-class model. The average accuracy of the developed 12-class model was 0.84 whereas the 4-class model average accuracy was 0.85. Therefore, we can conclude that the developed turning lane detector performs quite well.

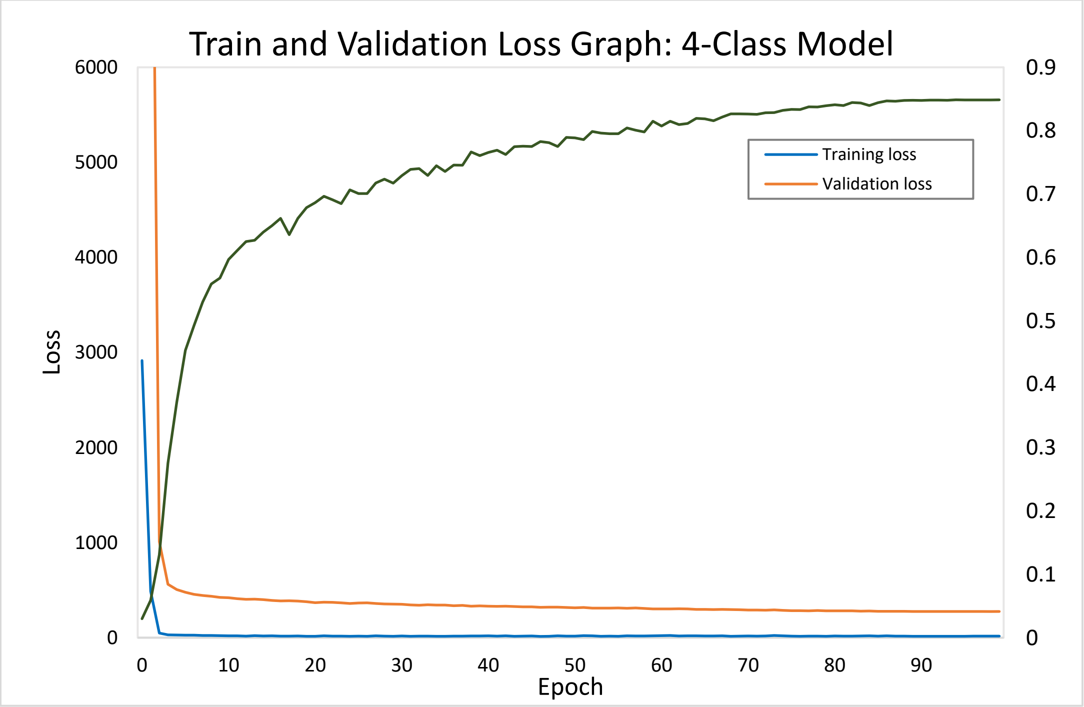


(a)

**Figure 3-5:** Developed YOLOv3 model train and validation loss graph: (a) school zone model, (b) 12-class turning lane model, and (c) 4-class turning lane model



(b)



(c)

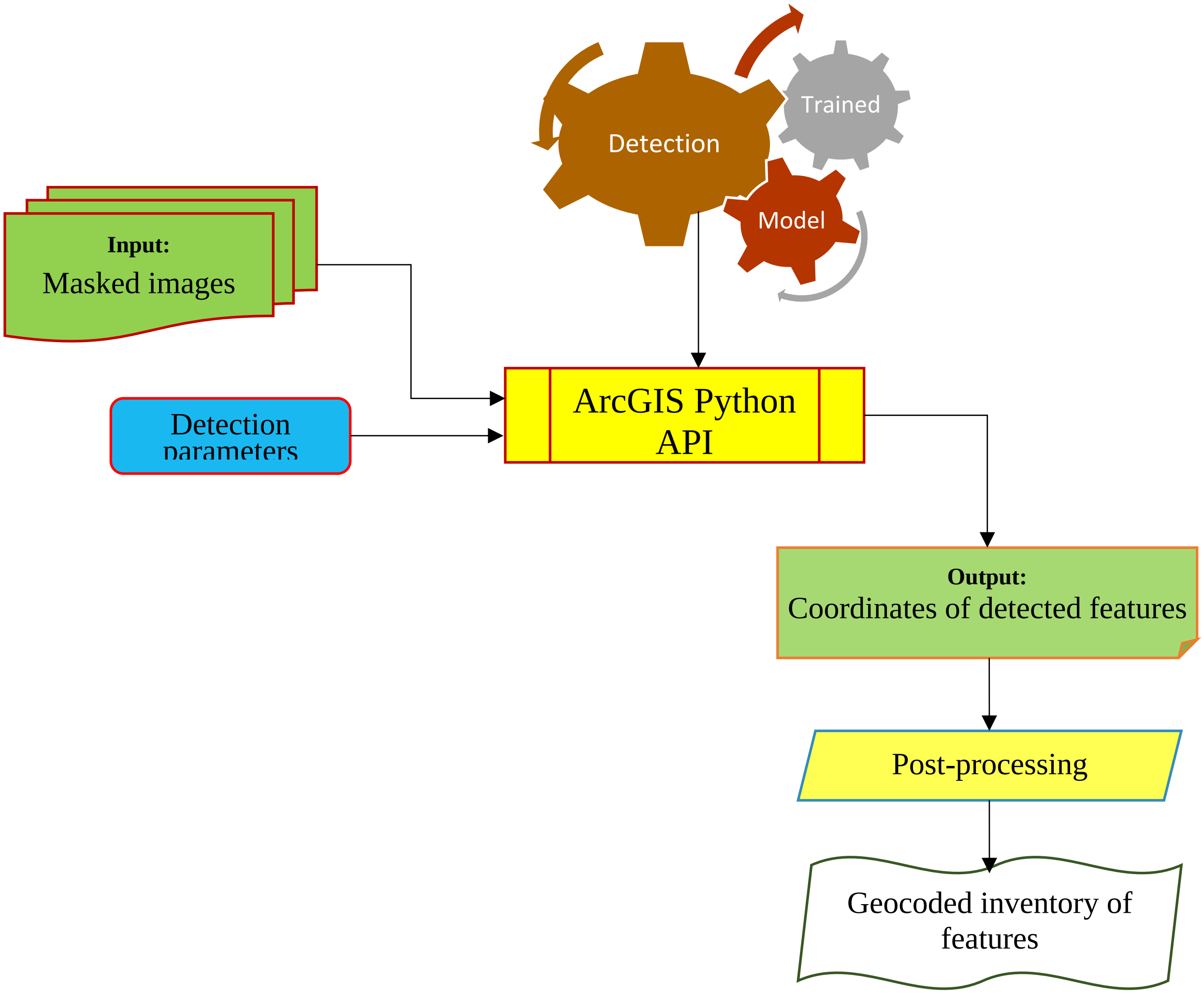
**Figure 3-5:** Developed YOLOv3 model train and validation loss graph: (a) school zone model, (b) 12-class turning lane model, and (c) 4-class turning lane model

### 3.3.5. Mapping School Zones and Turning Lanes

The school zone and turning lane detectors were initially tested on individual photos. Figure 3-6 demonstrates how the detectors correctly outline school zones and turning lanes with bounding boxes using the detection's confidence score.

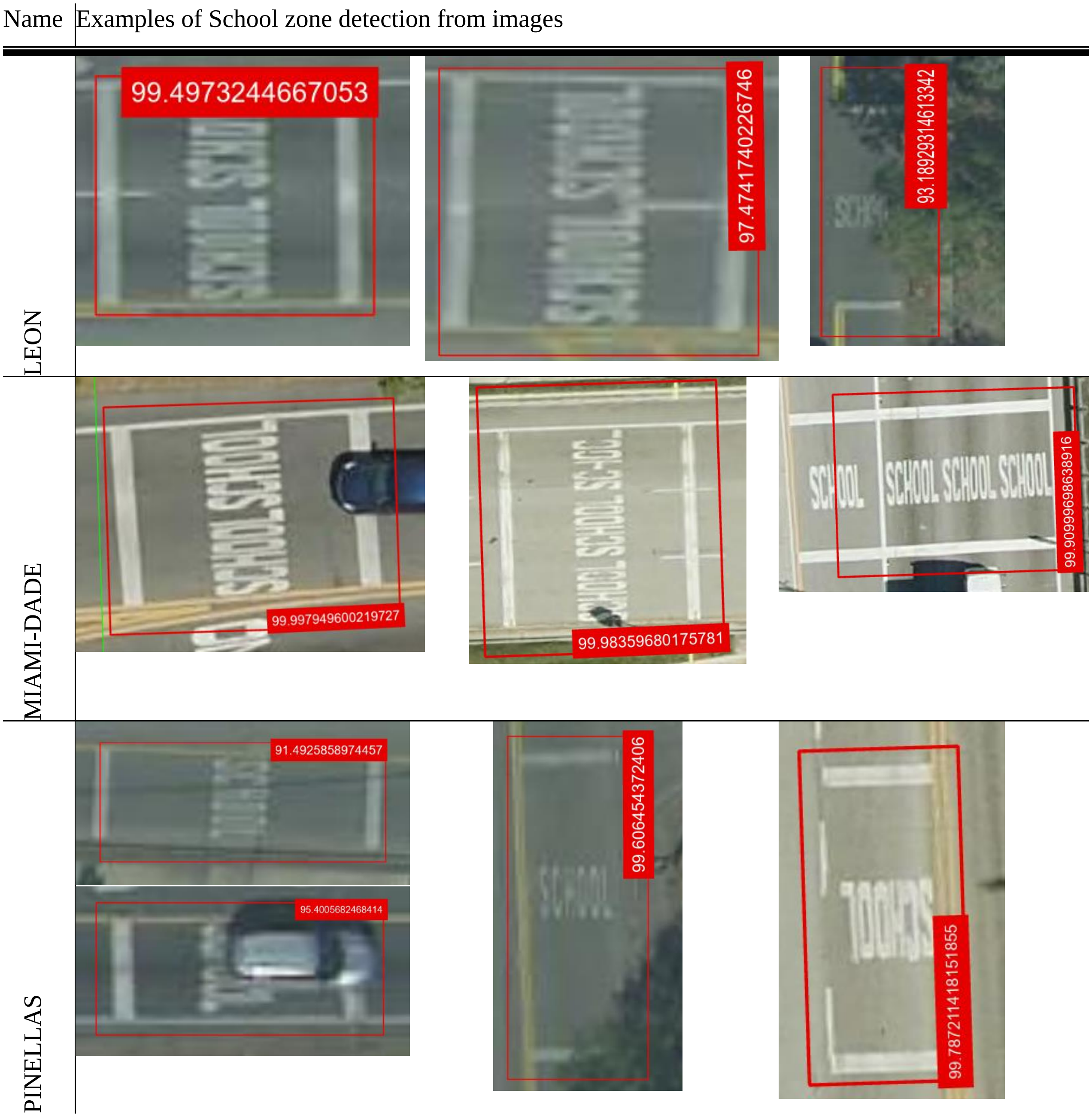
For the school zone detector, a threshold of 0.1 was employed to lessen false positives. However, due to the complexity of multiclass models, a threshold of 0.05 was employed in turning lane detection to capture all features detected with very low confidence levels. It is important to note that lower detection thresholds generally increase false positives, which results in lower precision, increases computational workload, which tends to increase detection time, and increases detection of irrelevant features or noise. However, lower detection thresholds also increase model’s sensitivity to detect faint or partially visible features, while increasing recall. With higher recall, the model is more likely to identify and detect all instances of the target object class and reduce the chances of missing any objects. For all models, more than 10% overlap between two bounding boxes was avoided to minimize duplication detections. To lessen information loss from the margins of the detection pictures, a padding parameter of 56 was added to the boundary pixels on the outside of the image. The detector was trained on 256 x 256 sub-images with a stride of 128 by 128 pixels and a resolution of 0.5 feet per pixel. Because the stride is half of the size of the image chip, this leads to a 50% overlap between any two image chips in sequence. The image was rescaled and processed with black pixels to have the same attributes if the provided image had a different size or resolution. It should be noted that utilizing huge photos with object detection techniques is impractical since the cost of computing grows rapidly with size.

The mapping procedure was carried out at the county level because the detector performs very well on single photos. Figure 3-6a provides a summary of this procedure. The photographs in the folder labeled "masked images" were first selected and iterated through the detector. An output file of all the identified school zones in that county was created once all photos had been sent to the detector. Confidence scores were included in the output file. This file was used to map school zones. Note that the model can detect school zone and turning lane markings from images with a resolution ranging from 1.5 ft down to 0.25 ft or higher. However, the model has not been tried on any images with resolution lower than the ones provided by the Florida APLUS system. From the observations, the models made some false detections in a few instances. These were outlined and discussed in the results. Figure 3-6c shows some examples of the detected features, and some observed false detections or misclassification.



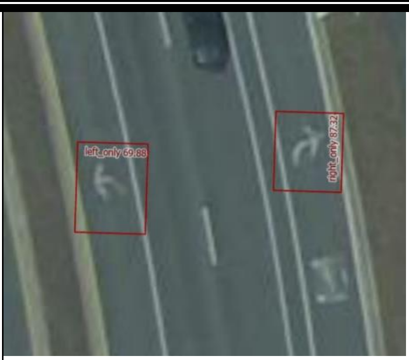
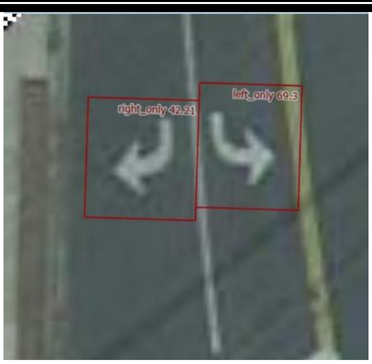
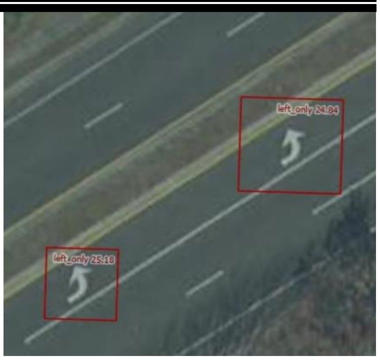
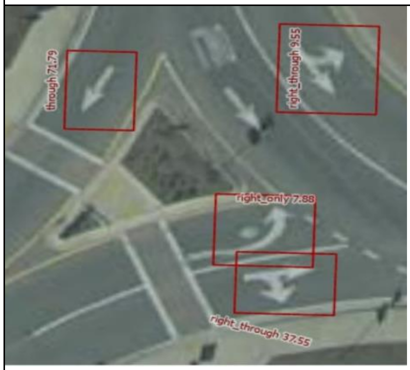
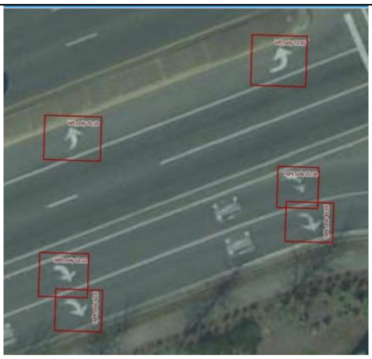
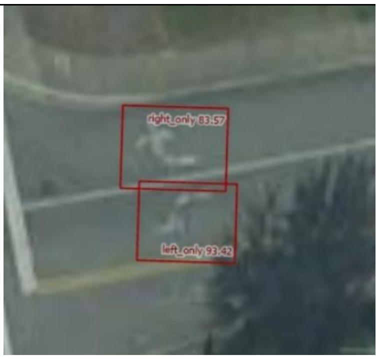
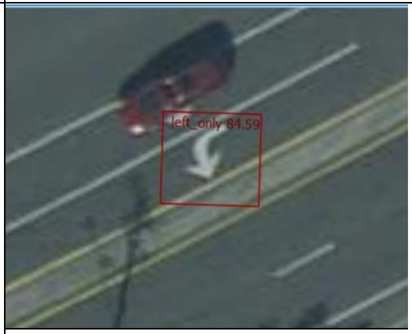
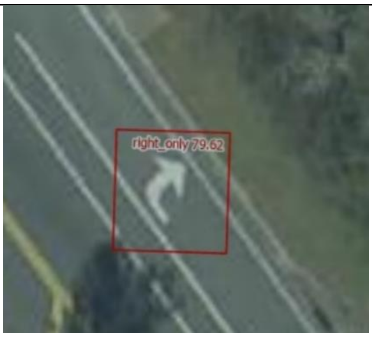
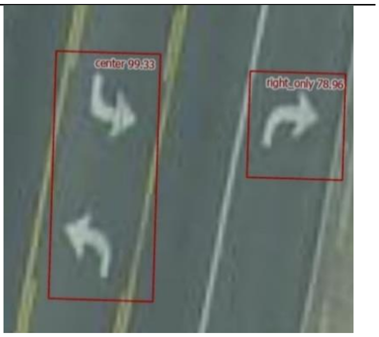
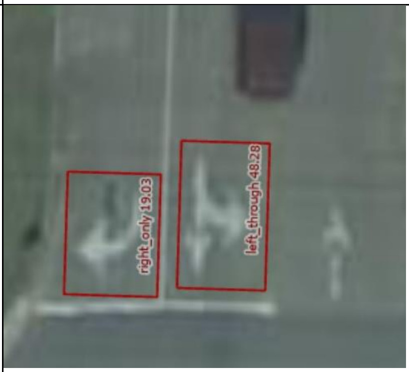
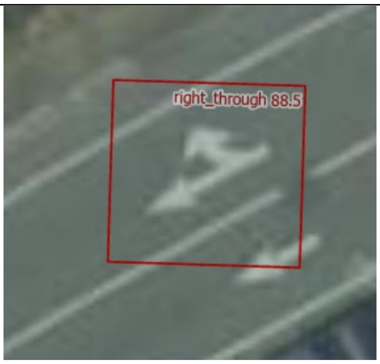
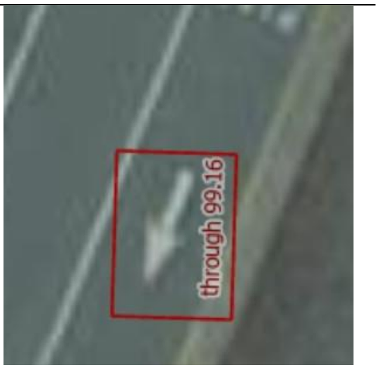
(a)

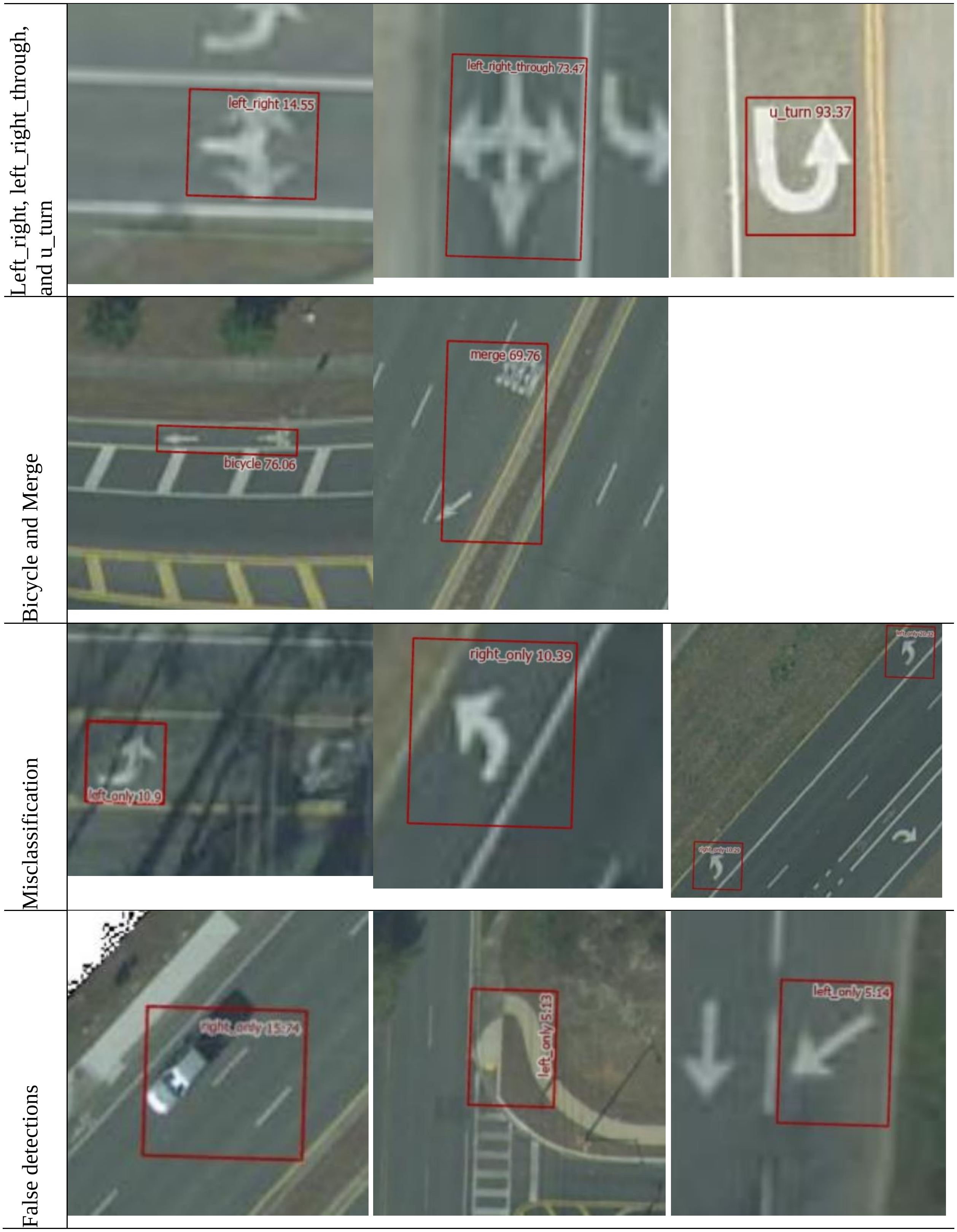
**Figure 3-6:** (a) School zones and turning lane detection framework, (b) school zone detection polygons and confidence scores on images from Leon, Miami-Dade, and Pinellas counties, and (c) turning lane detection polygons and confidence scores on images from Leon County



(b)

**Figure 3-6:** (a) School zones and turning lane detection framework, (b) school zone detection polygons and confidence scores on images from Leon, Miami-Dade, and Pinellas counties, and (c) turning lane detection polygons and confidence scores on images from Leon County

| Name | Examples of turning lane feature detection from images | | |
|---|---|---|---|
| |  |  |  |
| Multiple features |  through 71.79<br>right_through 9.55<br>right_only 7.88<br>right_through 37.55 |  |  |
| Left, Right, and Center |  |  |  |
| Left_through, right_through, and through |  |  |  |


Left_right, left_right_through, and u_turn
left_right 14.55
left_right_through 73.47
u_turn 93.37
Bicycle and Merge
bicycle 76.06
merge 69.76
Misclassification
left_only 10.9
right_only 10.39
left_only 20.32
right_only 10.29
False detections
right_only 15.74
left_only 5.13
left_only 5.14

(c)

**Figure 3-6: (**a) School zones and turning lane detection framework, (b) school zone detection polygons and confidence scores on images from Leon, Miami-Dade, and Pinellas counties, and (c) turning lane detection polygons and confidence scores on images from Leon County

### 3.3.6. Post-processing

The redundant detections caused by the overlapping distance on sliding images are removed at the post-processing stage. For various analytical objectives, school zones and turning lanes found on state and local roadways can also be divided into categories. The total number of observed school zone markings in Leon County was 61. After passing the turning lane model on the obtained aerial images in Leon County, the total number of observed left, right, and center detections in Leon County from the 12-class model and 4-class model were 2,736 and 3,165, respectively. Non maximum suppression was used to filter the detected school zones by selecting and keeping school zone detections that overlap and have the highest confidence level. Detected school zones with more than a 20% overlap and lower confidence levels were removed whereas detected turning lane markings with more than a 10% overlap and lower confidence levels were also removed. Detected features were then converted from polygon shapefiles into point shapefiles for subsequent analysis.

## 3.4. Results

### 3.4.1. Overall Performance Evaluation Using the Ground Truth Data

#### 3.4.1.1. *Experimental Design*

Leon County was utilized as the county where ground truth data were collected and the developed models' performances are assessed together with their accuracies and completeness, and the results are contrasted with training data. A full, manually positioned Ground Truth (GT) school zone and turning lane datasets were created in Leon County as proof of concept. After visual inspection, a total of 51 and 2,566 visible school zone and turning lane markings for left, right, and center lanes were collected using the masked photographs as the background, respectively. Please be aware that whereas a single school zone marking is observed at a time, each turning lane may exhibit a range of 1-6 features, such as left-only or right-only indicators. Consequently, a single turning lane could encompass several consecutive features that collectively define that lane. In cases where an intersection contains multiple turning lanes aligned in parallel, each lane may consist of several features. Within this framework, any feature that is inaccurately classified amidst correctly classified features within the same lane is deemed a false positive or a misclassified turning lane. For instance, in a left-only lane with four features, if three of them are correctly identified as left, and the fourth is misclassified as right or center, the entire lane is considered a false positive or misclassified detection. Figure 3-7a and Figure 3-7b show the GT dataset and detected school zone and turning lane markings in Leon County, respectively. Although some of the turning lane markings, especially the left only marking, were missed by the model, the overall performance of the turning lane model was reliable. Also, a few of the school zone markings were also missed by the model. This is mainly because of various reasons such as occlusions, faded markings, shadows, poor image resolution, and the variety in pavement marking design.

As noted, the suggested turning lane model has identified turning lane markers with a minimum confidence score of 5% while the school zone model detected school zones at a minimum 55%. For the purposes of this case study, the model identified turning lane or school zone markers (M) on both state and local roadways in Leon County, which were then retrieved. On the GT, a similar location-based selection methodology was used. While the school zone model's performance was assessed at various confidence levels of 0.55, 0.6, 0.7, 0.8, and 0.9, the suggested turning lane model's performance was assessed by examining the points that were discovered within the polygons and vice versa at various confidence levels of 75%, 50%, 25%, 10%, and 5% because the GT and M are points and polygons, respectively (Table 3-2). The performances of the developed models were assessed.

The school zone and turning lane models' performances were assessed using the criteria of completeness (precision), correctness (recall), and quality (intersection over union) and visualized using a graph and a circus plot (Krzywinski et al., 2009) respectively in Figures 8a, 8b and 8c. These criteria were initially utilized in (Wiedemann et al., 1998) and (Wiedemann and Ebner, 2000) for the goal of highway extraction, and they are now often used for performance evaluation of the related models (Sun et al., 2019; Dai et al., 2020). The following selection criteria are necessary to determine the performance evaluation metrics using the turning lane example:

i. False Negative (FN): # of GT turning lane points not found within M turning lane polygon,
ii. False Positive (FP): # of M turning lane polygons with no GT turning lane point,
iii. True Positive (TP): # of M turning lane polygons with GT turning lane point,
iv. GT: Number of GT turning lane points, and
v. M: Number of Model detected turning lane markings

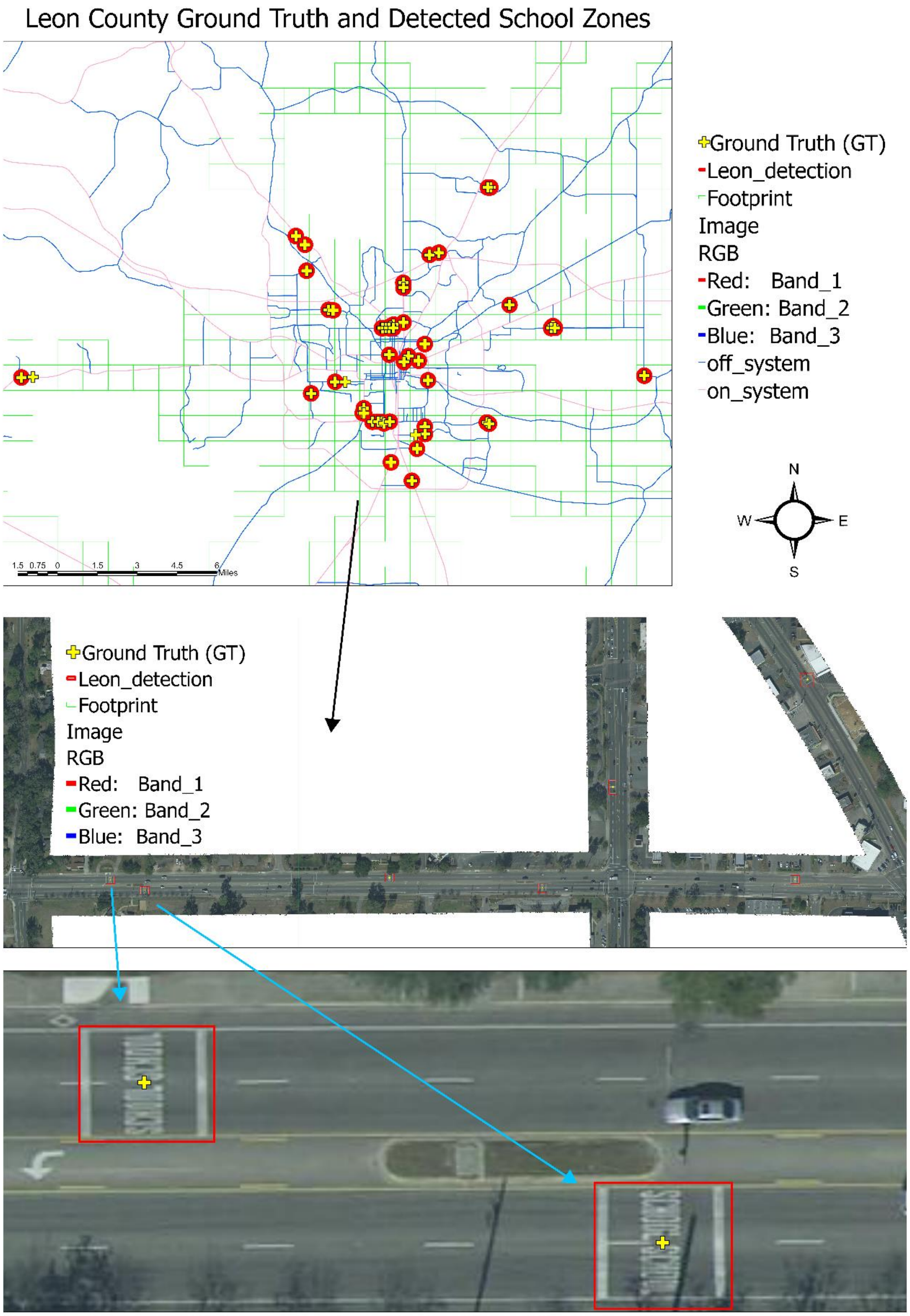
**Performance evaluation metrics:**

Completeness $=\frac{GT-FN}{GT} * 100\%$, true detection rate among GT turning lane (recall)

Correctness $=\frac{M-FP}{M} * 100\%$, True detection rate among M turning lane (precision)

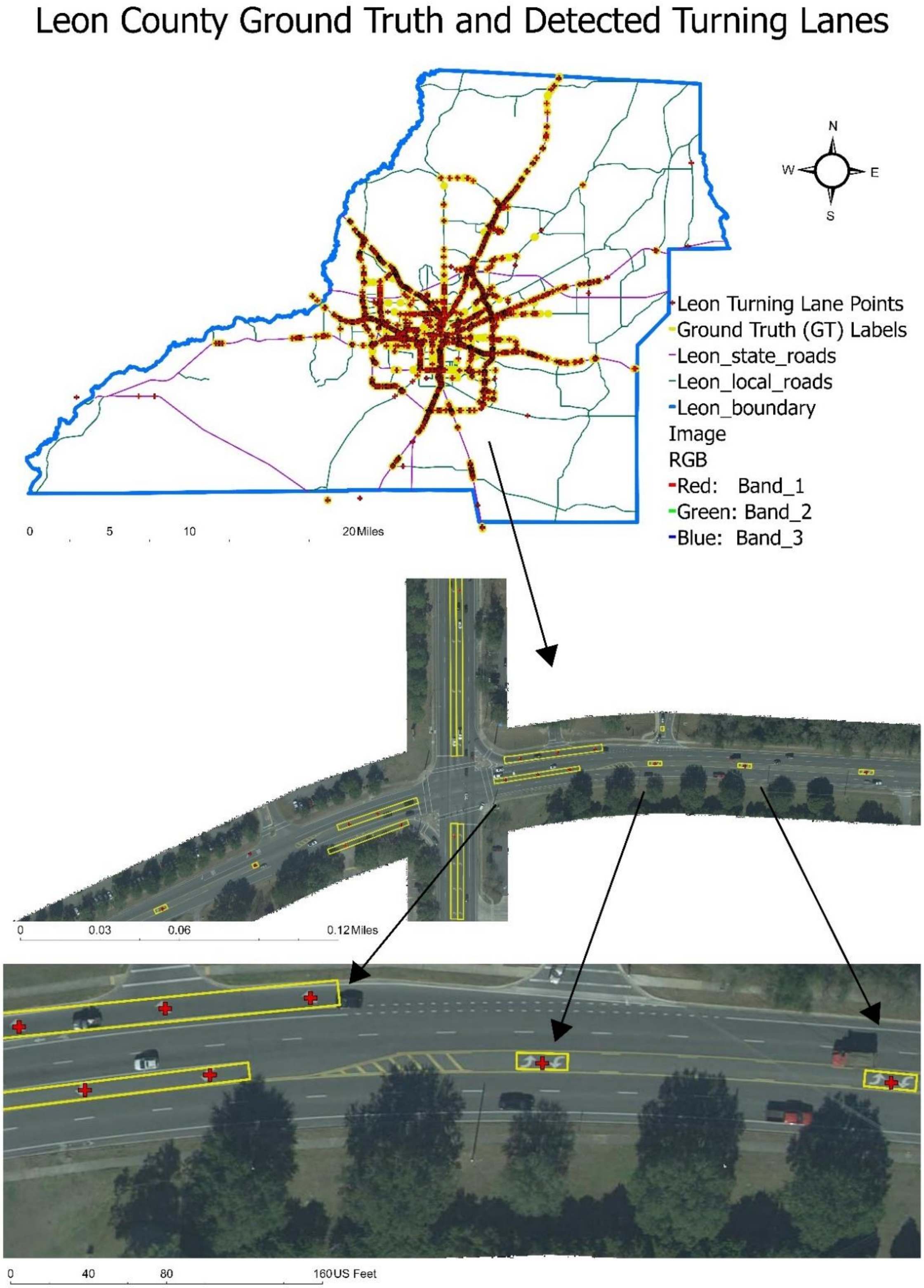
Quality $=\frac{GT-FN}{GT+FP} * 100\%$, True detection among M turning lane plus the undetected GT turning lane (Intersection over Union: IoU)

Here, our major goal is to assess the accuracy and performance of the proposed model's predictions and contrast them with a ground truth dataset. Separate evaluation analysis will be performed using "school zone", "left_only", "right_only", and "center" detections of the developed models. The accuracy and performance of models will be assessed after testing the models' correctness (precision) and completeness (recall) using a complete ground truth dataset and measuring the f-1 score. The f-1 score calculates the harmonic mean using the precision and recall values. It is the appropriate evaluation metric when dealing with imbalanced datasets. It is crucial in object detection tasks where missing actual objects is more detrimental than incorrectly classifying background regions as objects. The f1-scores of the developed turning lane models were compared and visualized using a circus plot in Figure 3-9.



(a)

**Figure 3-7**: Leon County manually labeled Ground Truth (GT) and detected (a) school zone markings and (b) turning lane markings.



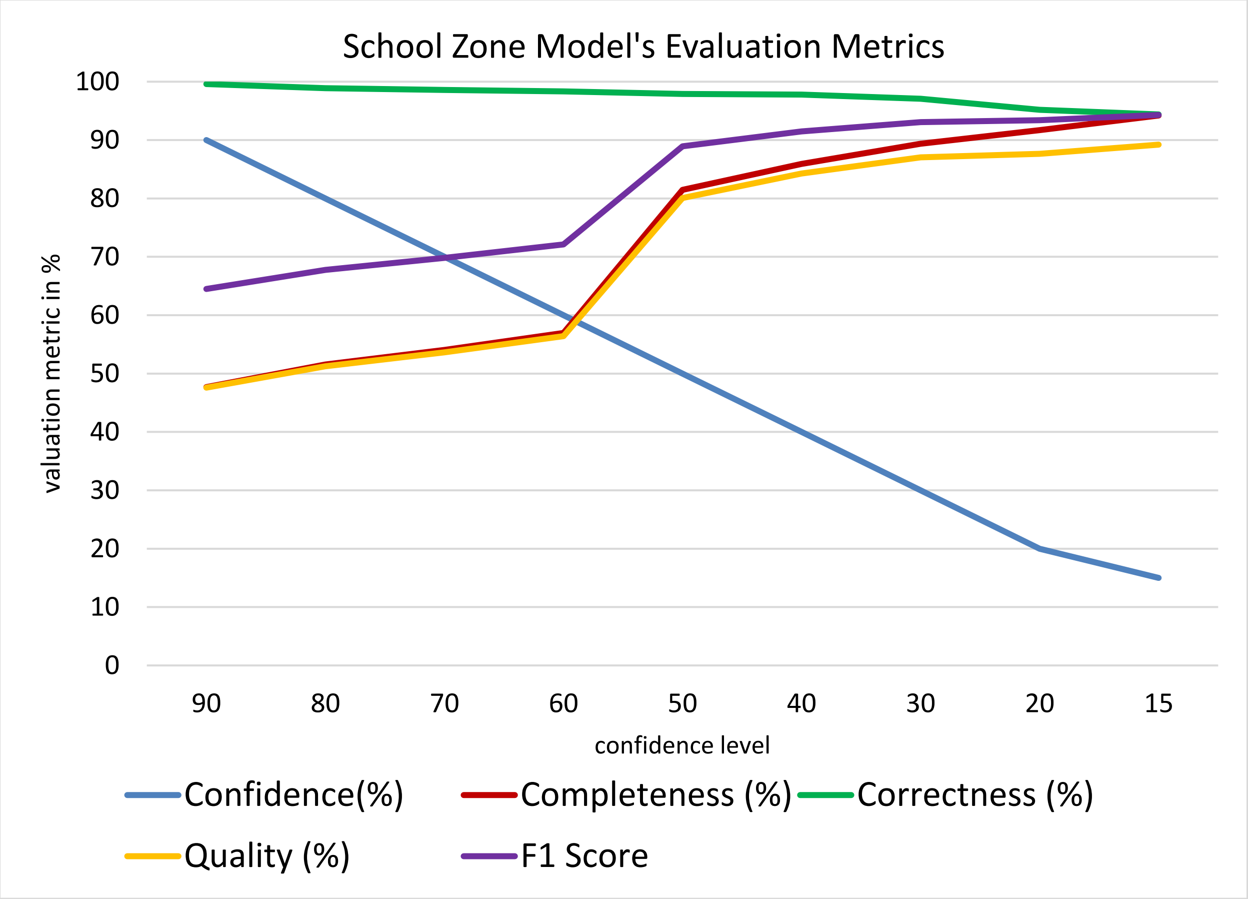
(b)

**Figure 3-7:** Leon County manually labeled Ground Truth (GT) and detected (a) school zone markings and (b) turning lane markings.

Based on the findings, we observe that this automated school zone detection and mapping model can detect and map 94% of the school zones with 100% precision at 90% confidence level. At a lower confidence level of 55%, it can detect 96% with 80% precision. At 90% confidence level, the quality of the model is 94% whereas the quality of the model is 77.8% at 55% confidence level. Meanwhile, the 12-class turning lane detection and mapping model can, on average, detect and map 23% of the turning lanes with 94% precision at 75% confidence level. At a lower confidence level of 5%, it can detect 73% of the turning lanes with 96% precision. At 75% confidence level, the average quality of the model's detection is 16% whereas the average quality of the model is 59% at 5% confidence level. On average, the 4-class turning lane model recorded 42% of detections with 99% precision at 75% confidence level. At a lower 5% confidence level, 80% of detections were observed at 97% precision. The average quality of the 4-class model detections at 75% confidence level is 28% and 68% at 5% confidence level.

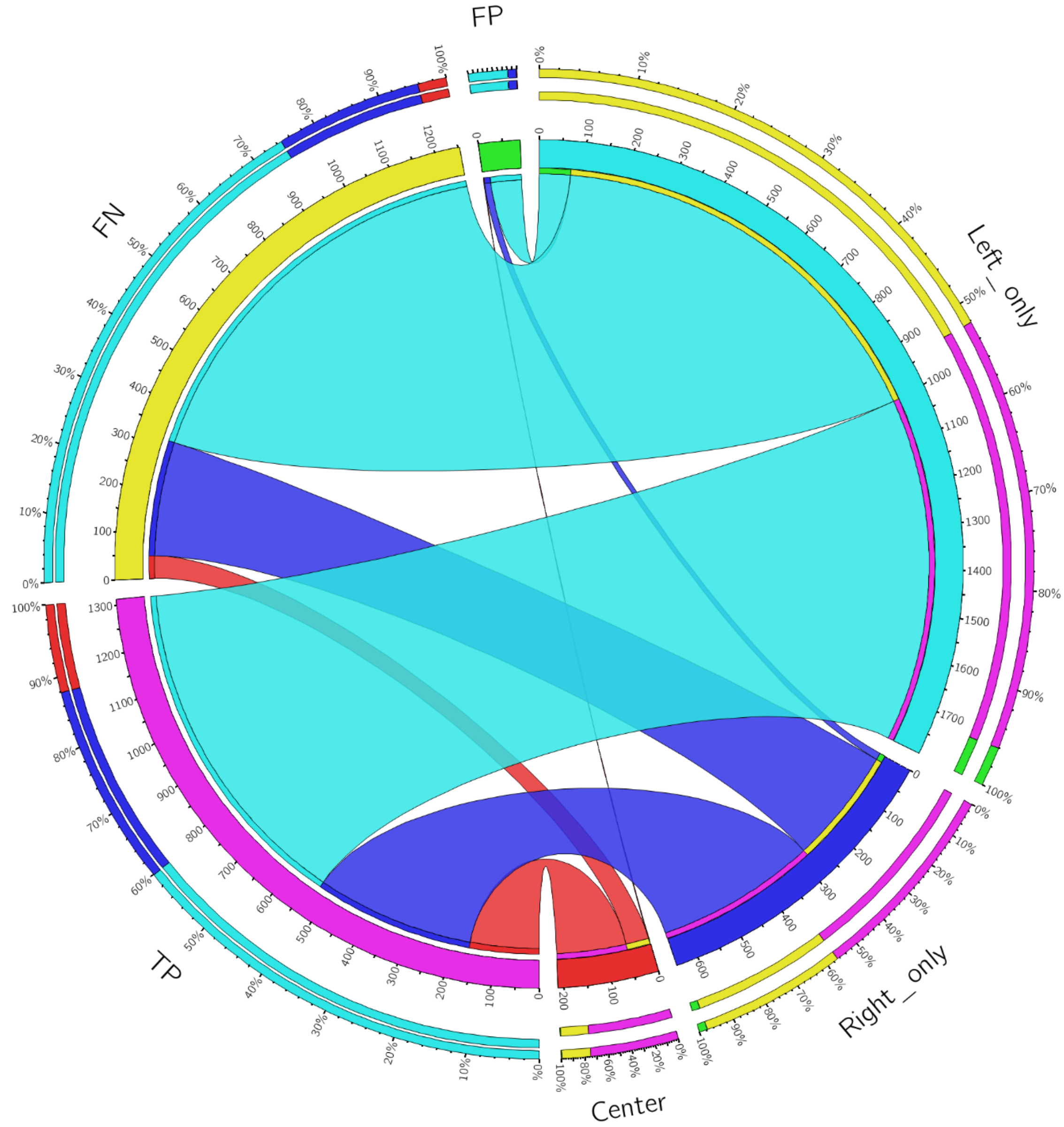
Higher accuracy was achieved at low confidence levels since there is a higher recall and more room is given to increase the number of detections. That is, from the observations, detected roadway geometry markings that had occlusions from vehicle or trees, shadows, and faded markings generally had lower confidence levels. Therefore, reducing the confidence level threshold adds these detected features to the total number of detections. The new detections allowed into the pool for valuation relatively includes more true positives, less false positives, and less or zero false negatives. With the increase in the number of true positives as confidence decreases, the accuracy of the model increases since the accuracy is described based on the relationship between the number of true positives, total number of detections, the high-resolution aerial images, and the false negatives. It can be observed that the 4-class turning lane detection model generally recorded higher accuracies than the 12-class model. This is due to the less complexity or lower number of classes of the 4-class model which increases model's performance.

The summarized model performance evaluation can be observed in Table 3-2 and visualized in **Error! Reference source not found.**. It can also be observed that the poor distinctiveness of the detection features of the turning lane model greatly affected detection performance. As stated earlier, the observed difference between the left only and right only turning markings, which is just a lateral inversion of the other marking, made it less unique and therefore resulted in a low detection performance. When the left turn is flipped horizontally, it becomes a right turn and vice versa. On the other hand, center lane which was trained using a relatively distinct shape recorded better detection results than the left only and right only lanes.



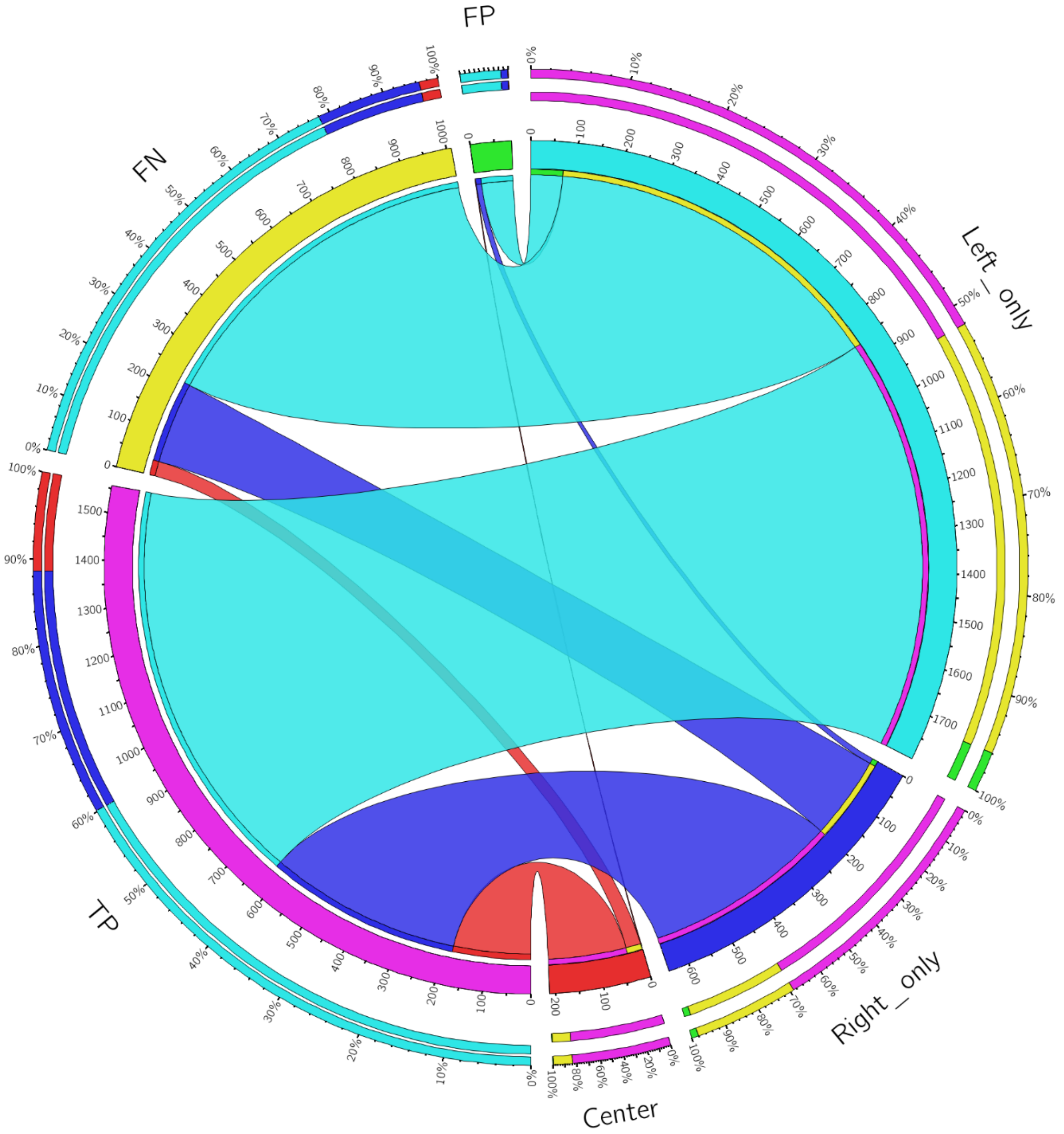
(a)

**Figure 3-8**: (a) Visualization of school zone model and visualization (circus plot) of performance evaluation metrics between the ground truth (GT) and predictions made by the YOLOv3-based (b) 12-class turning lane model and (c) 4-class turning lane model for detecting left_only (turquoise), right_only (blue), and center (red). The circus plot also shows the distribution of the true positives (magenta), false negatives (yellow), and false positives (green). The links between the classes show the number of true positives (correctly classified), false negatives (unclassified), and false positives (misclassified) in each class; the thickness of the links describe**s** their percentages. The size of the radii of the inner segments depicts the total value of the fields in ascending order. The outer concentric bars depict the percentages of the values in descending order. From 8c (4-class model), about 70% of right-only detections are true positives while only 56% were true positives in 8b (12-class model).



(b)

**Figure 3-8:** (a) Visualization of school zone model and visualization (circus plot) of performance evaluation metrics between the ground truth (GT) and predictions made by the YOLOv3-based (b) 12-class turning lane model and (c) 4-class turning lane model for detecting left_only (turquoise), right_only (blue), and center (red). The circus plot also shows the distribution of the true positives (magenta), false negatives (yellow), and false positives (green). The links between the classes show the number of true positives (correctly classified), false negatives (unclassified), and false positives (misclassified) in each class; the thickness of the links describe**s** their percentages. The size of the radii of the inner segments depicts the total value of the fields in ascending order. The outer concentric bars depict the percentages of the values in descending order. From 8c (4-class model), about 70% of right only detections are true positives while only 56% were true positives in 8b (12-class model).



(c)

**Figure 3-8:** (a) Visualization of school zone model and visualization (circus plot) of performance evaluation metrics between the ground truth (GT) and predictions made by the YOLOv3 based (b) 12-class turning lane model and (c) 4-class turning lane model for detecting left_only (turquoise), right_only (blue), and center (red). The circus plot also shows the distribution of the true positives (magenta), false negatives (yellow), and false positives (green). The links between the classes show the number of true positives (correctly classified), false negatives (unclassified), and false positives (misclassified) in each class; the thickness of the links describes their percentages. The size of the radii of the inner segments depicts the total value of the fields in ascending order. The outer concentric bars depict the percentages of the values in descending order. From 8c (4-class model), about 70% of right only detections are true positives while only 56% were true positives in 8b (12-class model).

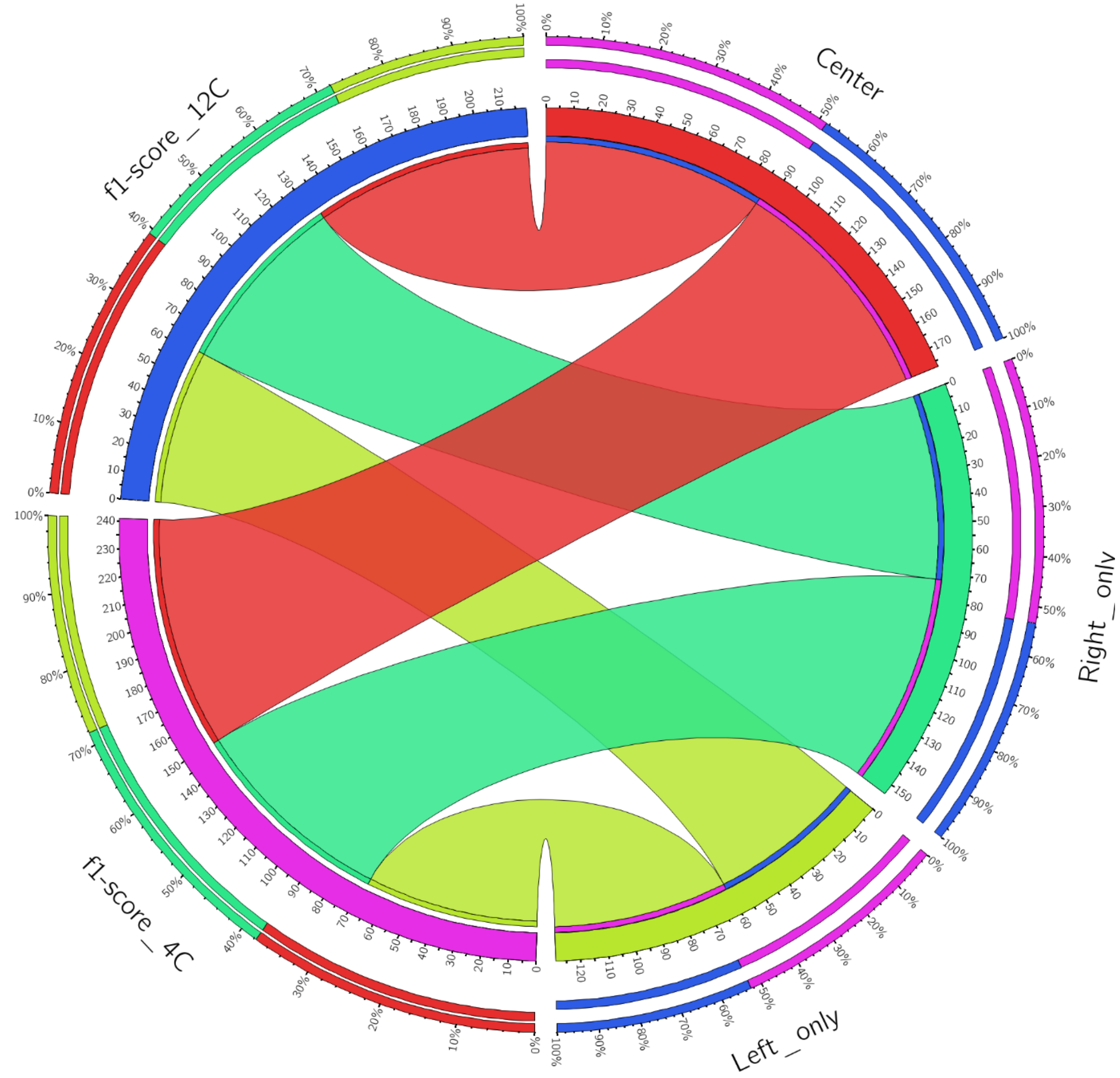
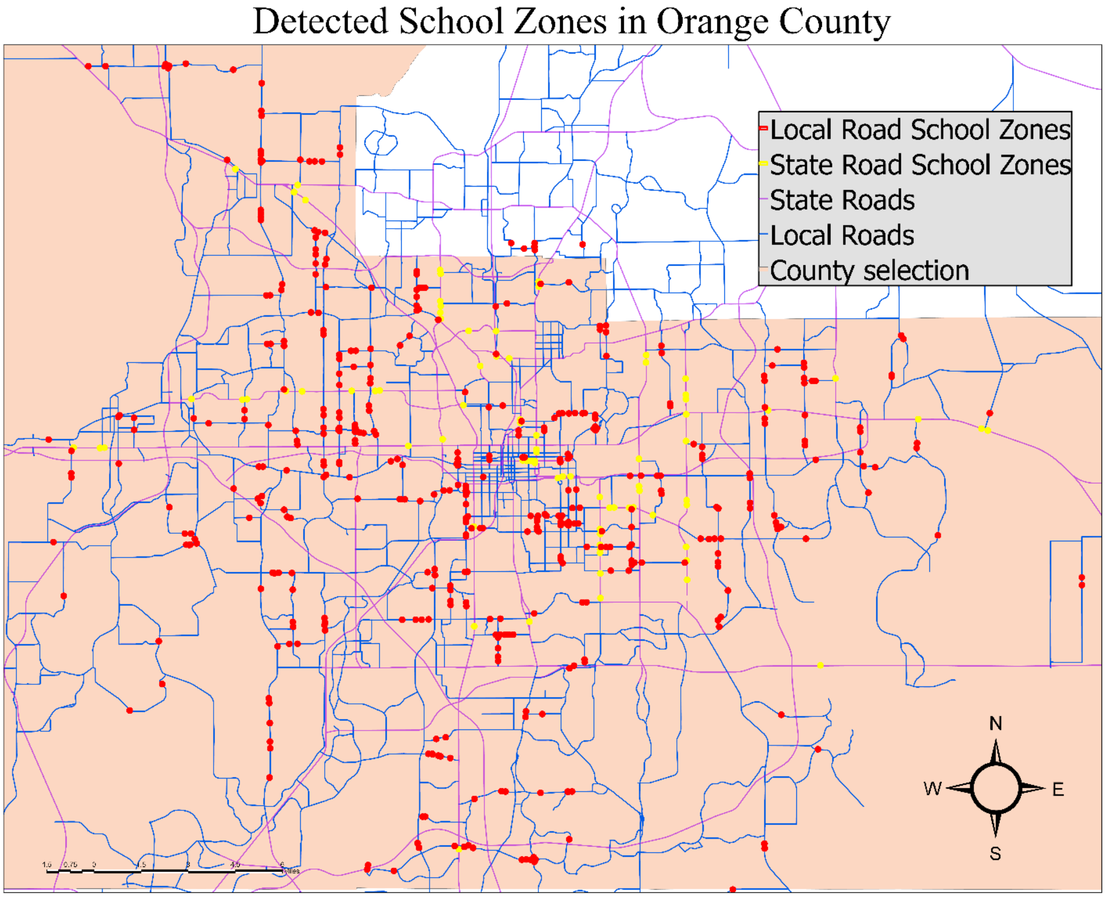


**Figure 3-9:** Visualization (circus plot) of f1-score comparison between the 12-class and 4-class YOLOv3-based turning lane model for detecting left_only (green-yellow), right_only (shamrock), and center (red). The circus plot also shows the distribution of the 4-class model's f1-score (magenta) and the 12-class model's f1-score (blue). The links between the classes show the f1-scores in each class; the thickness of the links describes their percentages. The size of the radii of the inner segments depicts the total value of the fields in ascending order. The outer concentric bars depict the percentages of the values in descending order. The f1-score of the 4-class model is more than 50% in all classes, indicating a better performance than the 12-class model. Also, the total value of the 4-class model's f1-score is ~240 which is higher than the 12-class model which is ~215.

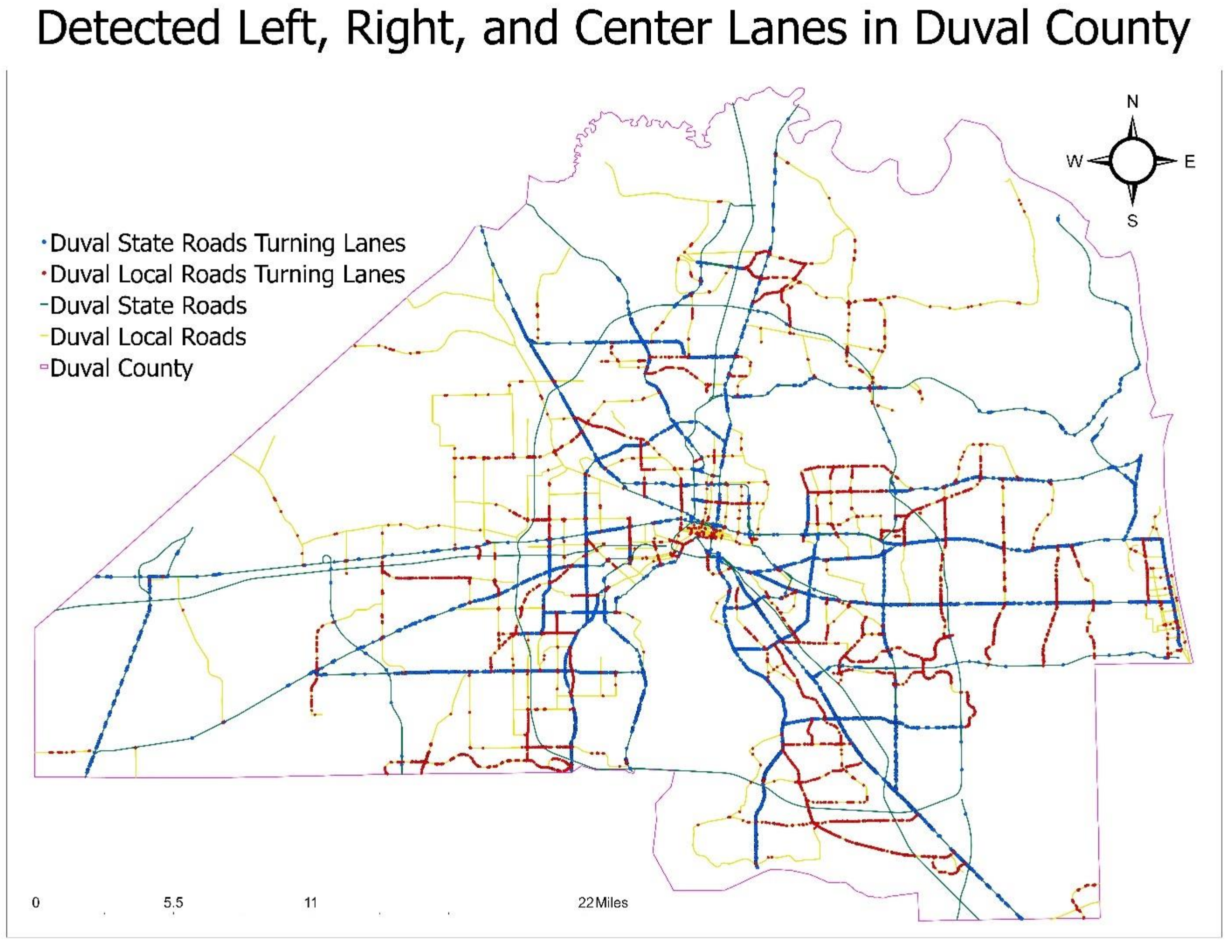
### 3.4.2. Detecting School Zones in Orange County and Turning Lanes in Duval County

Additionally, the model was used to detect school zones and turning lanes in Orange and Duval Counties respectively. The detected features were classified under different confidence levels. The final list is shown in Table 3-2. The detected school zone and turning lanes features have been visualized on Figure 3-10a and Figure 3-10b, respectively. The extracted road geometry data can be integrated with crash and traffic data to advise policy makers and roadway users. That is, they can be used for a variety of purposes such as identifying those markings that are old and invisible, comparing the school zone or turning lane locations with other geometric features like crosswalks, and analysing the crashes occurring around the zones or intersections.



(a)

**Figure 3-10**: Model-detected (a) school zones in Orange County and (b) turning lanes in Duval County.


Detected Left, Right, and Center Lanes in Duval County
N
W
E
S
Duval State Roads Turning Lanes
Duval Local Roads Turning Lanes
Duval State Roads
Duval Local Roads
Duval County
0
5.5
11
22 Miles

(b)

**Figure 3-10:** Model-detected (a) school zones in Orange County and (b) turning lanes in Duval County.

**Table 3-2:** Model performance evaluations and detected features

| Leon County Ground Truth Comparison Analysis | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| **4-class turning lane model** | | | | | | | | |
| **left_only: GT = 1723** | | | | | | | | |
| **Confidence (%)** | **M** | **TP** | **FP** | **FN** | **Completeness (%)** | **Correctness (%)** | **Quality (%)** | **F1-score (%)** |
| 75 | 283 | 275 | 8 | 1448 | 15.96 | 97.17 | 15.89 | 27.42 |
| 50 | 457 | 436 | 21 | 1287 | 25.30 | 95.40 | 25.00 | 40.00 |
| 25 | 666 | 626 | 40 | 1097 | 36.33 | 93.99 | 35.51 | 52.41 |
| 10 | 821 | 764 | 57 | 959 | 44.34 | 93.06 | 42.92 | 60.06 |
| 5 | 1008 | 935 | 73 | 788 | 54.27 | 92.76 | 52.06 | 68.47 |
| **right_only: GT = 632** | | | | | | | | |
| **Confidence (%)** | **M** | **TP** | **FP** | **FN** | **Completeness (%)** | **Correctness (%)** | **Quality (%)** | **F1-score (%)** |
| 75 | 129 | 127 | 2 | 505 | 20.09 | 98.45 | 20.03 | 33.38 |

Table 3-2, continued

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 50 | 216 | 211 | 5 | 421 | 33.39 | 97.69 | 33.12 | 49.76 |
| 25 | 341 | 334 | 7 | 298 | 52.85 | 97.95 | 52.27 | 68.65 |
| 10 | 400 | 390 | 10 | 242 | 61.71 | 97.50 | 60.75 | 75.58 |
| 5 | 456 | 444 | 12 | 188 | 70.25 | 97.37 | 68.94 | 81.62 |

| center: GT = 211 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Confidence (%) | M | TP | FP | FN | Completeness (%) | Correctness (%) | Quality (%) | F1-score (%) |
| 75 | 104 | 104 | 0 | 107 | 49.29 | 100.00 | 49.29 | 66.03 |
| 50 | 133 | 133 | 0 | 78 | 63.03 | 100.00 | 63.03 | 77.33 |
| 25 | 157 | 157 | 0 | 54 | 74.41 | 100.00 | 74.41 | 85.33 |
| 10 | 166 | 166 | 0 | 45 | 78.67 | 100.00 | 78.67 | 88.06 |
| 5 | 178 | 177 | 1 | 34 | 83.89 | 99.44 | 83.49 | 91.00 |

**12-class turning lane model**

| left_only: GT = 1723 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Confidence (%) | M | TP | FP | FN | Completeness (%) | Correctness (%) | Quality (%) | F1-score (%) |
| 75 | 11 | 10 | 1 | 1713 | 0.58 | 90.91 | 0.58 | 1.15 |
| 50 | 62 | 57 | 5 | 1666 | 3.31 | 91.94 | 3.30 | 6.39 |
| 25 | 271 | 255 | 16 | 1468 | 14.80 | 94.10 | 14.66 | 25.58 |
| 10 | 553 | 513 | 40 | 1210 | 29.77 | 92.77 | 29.10 | 45.08 |
| 5 | 858 | 787 | 71 | 936 | 45.68 | 91.72 | 43.87 | 60.98 |

| right_only: GT = 632 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Confidence (%) | M | TP | FP | FN | Completeness (%) | Correctness (%) | Quality (%) | F1-score (%) |
| 75 | 11 | 10 | 1 | 622 | 1.58 | 90.91 | 1.58 | 3.11 |
| 50 | 62 | 60 | 2 | 572 | 9.49 | 96.77 | 9.46 | 17.29 |
| 25 | 182 | 176 | 6 | 456 | 27.85 | 96.70 | 27.59 | 43.24 |
| 10 | 304 | 291 | 13 | 341 | 46.04 | 95.72 | 45.12 | 62.18 |
| 5 | 382 | 367 | 15 | 265 | 58.07 | 96.07 | 56.72 | 72.39 |

| center: GT = 211 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Confidence (%) | M | TP | FP | FN | Completeness (%) | Correctness (%) | Quality (%) | F1-score (%) |
| 75 | 98 | 98 | 0 | 113 | 46.45 | 100.00 | 46.45 | 63.43 |
| 50 | 129 | 129 | 0 | 82 | 61.14 | 100.00 | 61.14 | 75.88 |
| 25 | 148 | 148 | 0 | 63 | 70.14 | 100.00 | 70.14 | 82.45 |
| 10 | 155 | 154 | 1 | 57 | 72.99 | 99.35 | 72.64 | 84.15 |
| 5 | 160 | 159 | 1 | 52 | 75.36 | 99.38 | 75.00 | 85.71 |

**School zone model**

| GT = 51 | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Confidence (%) | M | TP | FP | FN | Completeness (%) | Correctness (%) | Quality (%) | F1-score (%) |
| 90 | 48 | 48 | 0 | 3 | 94.12 | 100.00 | 94.12 | 96.97 |
| 80 | 52 | 48 | 4 | 3 | 94.12 | 92.31 | 87.27 | 93.20 |
| 70 | 55 | 48 | 7 | 3 | 94.12 | 87.27 | 82.76 | 90.57 |
| 60 | 59 | 48 | 11 | 3 | 94.12 | 81.36 | 77.42 | 87.27 |
| 55 | 61 | 49 | 12 | 2 | 96.08 | 80.33 | 77.78 | 87.50 |

**Table 3-2, continued**

**Detection of Right, Left, and Center Markings in Duval County**

**Left**

| Confidence Level (%) | Detected left only features | | |
|---|---|---|---|
| | State Roads | Local Roads | Total |
| 75 | 1238 | 971 | 2209 |
| 50 | 2167 | 1722 | 3889 |
| 25 | 3607 | 2819 | 6426 |
| 10 | 4315 | 3346 | 7661 |
| 5 | 4969 | 3768 | 8737 |

**Right**

| Confidence Level (%) | Detected right only features | | |
|---|---|---|---|
| | State Roads | Local Roads | Total |
| 75 | 459 | 311 | 770 |
| 50 | 822 | 607 | 1429 |
| 25 | 1381 | 1066 | 2447 |
| 10 | 1661 | 1347 | 3008 |
| 5 | 1946 | 1631 | 3577 |

**Center**

| Confidence Level (%) | Detected Center lane features | | |
|---|---|---|---|
| | State Roads | Local Roads | Total |
| 75 | 317 | 406 | 723 |
| 50 | 430 | 533 | 963 |
| 25 | 478 | 627 | 1105 |
| 10 | 522 | 706 | 1228 |
| 5 | 634 | 886 | 1520 |

**Detection of School Zone Markings in Orange County**

| Confidence Level (%) | Detected School Zone Features | | |
|---|---|---|---|
| | State Roads | Local Roads | Total |
| 90 | 79 | 410 | 489 |
| 80 | 80 | 426 | 506 |
| 70 | 83 | 442 | 525 |
| 60 | 85 | 451 | 536 |
| 55 | 88 | 481 | 569 |

# 4. TASK 3: PROPOSE GUIDELINES ON THE STATEWIDE UTILIZATION OF THE METHODOLOGY

## 4.1. Task Description

In this task, the research team described the approach employed to develop the machine learning models used to detect and extract roadway geometry features such as school zones and turning configurations. Based on the review of literature that has been conducted for the purpose of discovering published information that can help inform, shape, or guide the conduct of this research project, the models were developed and implemented on the entire State of Florida to extract school zones and turn configurations on both on and off state highway system roadways. The main focus of this task was to provide the step-by-step method used to develop the machine learning model to extract roadway geometry features from high resolution aerial images. For this purpose, the following steps were described: (1) preprocessing steps, (2) training data creation method, (3) machine learning-based object detection model development, and (4) post-processing steps used for detecting school zones and turning configurations (e.g., left-only, right-only, and center lanes) from high-resolution aerial images. Also, the research team outlined various methods and approaches used in developing this model and extracting geometric data at intersections and data needs to facilitate the analysis.

## 4.2. Materials

The choice of method used to collect roadway inventory data depends on factors such as data collection time (i.e., data collection, reduction, and processing), cost (i.e., data collection, and reduction), and accuracy, safety, and data storage requirements. In previous tasks of this project, our aim was to create a deep learning object detection model to detect school zone and turning lane markings from high resolution aerial images in Orange County and Duval County.

### 4.2.1. System Configuration and Software Setup

To ensure the successful implementation of the developed object detection model, there are several system and software setup requirements that need to be addressed. The software used was ESRI’s ArcGIS Pro version 3.0. The default version does not contain the deep learning libraries; therefore, they need to be installed separately after installing the ArcGIS Pro software. ArcGIS pro has a python notebook interface where lines for codes can be used to implement model creation and image processing. Here, the advantage is that the interface allows users to directly import geoprocessing tools and utilize them for various analytical purposes. In other words, each line of code can be visualized in a geoprocessing tool format for easy implementation and usage, which would be very helpful for FDOT users. Similarly, the visualized geoprocessing tool in extension can also be used to generate lines of codes which can be implemented in other python environments (i.e., outside ArcGIS Pro software). It is important to check software access options to ensure that the user has access to the ArcGIS Image Analyst tools and ArcGIS API for python.

Several deep learning libraries for the current ArcGIS Pro version should be downloaded and installed. The deep learning libraries downloaded for ArcGIS Pro v3.0 is the Deep Learning Libraries Installer for ArcGIS Server 3.0 or equivalent (https://github.com/Esri/deep-learning-frameworks/blob/master/README.md?rmedium=links_esri_com_b_d&rsource=https%3A%2F%2Flinks.esri.com%2Fdeep-learning-framework-install). Most packages included in the Deep

Learning Libraries installer are designed to function well on a wide range of machine configurations. For instance, PyTorch has the capability to utilize a Graphics Processing Unit (GPU) for accelerated computation, but it switches to Central Processing Unit (CPU) processing when a GPU is absent. Nonetheless, it is worth noting that GPU-based computation offers a substantial speed boost, and certain packages like TensorFlow in this distribution exclusively require a compatible GPU to operate effectively. To enable GPU processing, one will need Compute Unified Device Architecture (CUDA)—a fundamental necessity for modern GPU-accelerated deep learning tools. NVIDIA drivers are required and a 8GB or more of dedicated memory is recommended.

## 4.3. Implementation of Methods

### 4.3.1. Pre-processing

As described in previous tasks, aerial images were obtained from the Florida Aerial Photo Look-Up System (APLUS). This archive is accessible to the public via APLUS, which is run by the FDOT Surveying and Mapping Office. After obtaining aerial images, one important task is to mask the images. Due to the amount of data and the difficulty of the object recognition procedure, preprocessing is a necessary step. In general, our method chooses and discards any images that do not cross a roadway centerline, and then it masks out any pixels that are not encircled by a buffer zone. This study focused on the roadway geometry data located on the state highway system roadways (ON System Roads) as well as those located on the county- or city-controlled roadways (OFF System Roads). For this purpose, first, interstates were excluded from the data set all together after merging both the ON System Roads shapefile and OFF System Roads shapefile to combine all centerlines. Before the images were masked, the roadway shapefile which were used, were buffered 100 ft. to form polygons having overlapping boundaries dissolved. Using this layer as a reference, aerial images were iterated, and intersecting regions of the aerial images were cropped. During the masking process, pixels that fell outside the boundary of the reference layer were removed. The cropped images, which consequently had smaller number of pixels, were mosaiced together to form a single raster file. In this approach, the number of photos was decreased from 90K to 30K, and the image masking model excluded objects that were 100 feet away from state and local roadways. This new file was smaller and easier to handle in a mosaic format for all forms of raster analysis or data processing.

To implement the image masking process, a new blank geoprocessing model was developed where a folder variable was created and connected to the path of the masked images folder. Then, a raster iterator tool was imported to iterate through the raster images. A name string was used for the iteration and the output raster was extracted using the extract by mask tool. The "Name" (blue) is a derived output from the raster iteration tool which is a string containing the name of the raster file. While the "%Name%" (green) is the output raster from extract by mask tool containing the cell values extracted from the input raster. The enclosed percent signs (%) were used to perform inline variable substitution by substituting the output variable names derived from the extract by mask tool with the original names used for raster iteration. This method maintains the original names of the input raster images used for the masking process. The buffered roadway shapefile was used as a mask feature and the extraction area was set inside. Finally, "raster to other format tool" was used to convert and export the output of the extraction in the format of a jpeg file into a folder. This masking process was automated using the ArcGIS Pro ModelBuilder.

Figure 3-2a illustrates the preprocessing strategy in detail. All the photos from the chosen counties (N is approximately 90K) were initially imported into a mosaic dataset using the ArcGIS Pro program. Multiple geocoded photos were managed and shown using mosaic databases. Additionally, mosaic datasets enable the intersection of additional geocoded vector data to pick picture tiles depending on location. For instance, a subset collection of the images (n=30K) was created by selecting and extracting individual images that comprise a portion of the roadway centerline. This automated masking process is shown in Figure 3-2b, the tool repeatedly goes through a folder of photographs, applies a mask to each image based on a 100-foot buffer around the centerlines of the road, and then saves the masked images as JPG files for the object recognition method.

To implement image masking in ArcGIS Pro, the following steps are required where they are shown for Orange County specifically:

**(a) Preprocessing - Selecting Images for Masking (Figure 3-2a)**

- Start **ArcGIS Pro**
- In the upper right corner, click **Sign in** and type your username and password. Click **Sign in.**
- On the start page, start a new project by clicking on **Map** or open existing project by clicking on any existing project or **Open another project** in the **Recent Projects** tab.
- After clicking on **Map**, a pop up appears, create a **Name** (avoid leaving any spaces in the name) for the project, set project folder **Location** and click **OK**.
- Go to the **Geoprocessing** group in the **Analysis** tab and click on **Tools** to open the Geoprocessing tools search box. In the search box, type ‘Create Mosaic Dataset’.
- Click on the **Create Mosaic Dataset** tool in the search results to open it.
- On the **Parameters** tab, navigate to and select the **Output Location** or the path to your geodatabase where the mosaic dataset will be stored by clicking on the folder icon. Create a name for the mosaic file in the **Mosaic Dataset Name** field and input the coordinate system of the mosaic dataset. Note that the coordinate system can be set by easily dragging and dropping an image file from the raster images to be used to create the mosaic dataset into the map view. Afterwards, click the drop down arrow in the **Coordinate System** field to select the added image as the source for the coordinate system. Maintain all other default fields and click **Run**.
- Again, in the **Geoprocessing** tools search box, type ‘Add Rasters’. Click on the **Add Rasters to Mosaic Dataset** tool to open it.
- In the **Parameters** tab, navigate to and select the created mosaic dataset or use the drop-down arrow to select the created mosaic dataset to fill the **Mosaic Dataset**

field. Change the **Input Data** from 'File' to 'Folder' and use the folder icon with a plus sign to browse to the folder containing the raster images and select it.

- Keep all other fields as default and click **Run**. The created mosaic dataset contains all the raster images in the folder and displays it as a single raster file.
- Now, go back to the **Geoprocessing** tools search box, type 'Select by location'. Click on the **Select Layer By Location** tool to open it.
- In the **Parameters** tab, add the 'Footprint' of the created mosaic dataset in the **Input Features** field by clicking on the drop-down arrow and selecting 'mosaic_name\Footprint'. Set the spatial relationship to be evaluated to 'Intersect' in the  **Relationship field**. Add the feature layer that will be used to select the input features based on their relationship to the selecting features in the **Selecting Features** field in this case 'on_off_merge_no_freeways.shp'. This layer was used to select the images that intersect the roadways of interest. Maintain the **Selection Type** field as 'New selection', keep all other fields as default and click **Run**.
- After the tool completes the selection, go back to the **Geoprocessing** tools search box, type 'Export mosaic dataset items'. Click on the **Export Mosaic Dataset Items** tool to open it.
- On the **Parameters** tab, add the created mosaic dataset in the **Mosaic Dataset** field by clicking on the drop-down arrow and selecting. Indicate the output folder of the exported mosaic dataset items in the **Output Folder** field by clicking on the folder icon and navigating to an existing or created output folder. Leave all other fields as default and click **Run**.
- The selected images are exported into a folder for masking.

**(b) Preprocessing – Creating Image Masking Model (Figure 3-2b)**

- To buffer, go to the **Geoprocessing** tool search box and search 'buffer'. Click on the **Buffer** tool and add the roadway layer in the **Input Features** field. Create the **Output Feature Class** name in the field and indicate the output directory. Set the distance value in the **Distance** field, set the desired units, and maintain all other fields as default. Click **Run** to create the buffer layer for masking.
- On the ribbon above the map view, click on the **Analysis** tab. In the **Geoprocessing** group, click on the **ModelBuilder** tool. A new model interface opens and **ModelBuilder** tab is added to the ribbon.
- On the **Insert** group of the **ModelBuilder** tab, click on **Iterators** to expand the options and select **Iterate Rasters** to load the tool in the model interface.
- In the model interface, double click on the **Iterate Raster** tool to open. A pop-up window will show. In the **Parameters** tab, navigate to the folder containing raster images to be masked (images that intersect the roadways) by clicking on the folder icon next to the **Workspace** field. Leave the **Wildcard** field empty and click on the drop-down arrow in the **Raster Format** field and select the extension of the images in this case was 'JPG'. Leave all other parameters as default and

click **OK**. Notice the tool loads the parameters and becomes active (turns from gray to color).

- Go back to the **Geoprocessing** group in the **Analysis** tab and click on **Tools** to open the **Geoprocessing** tools search box. In the search box, type ‘Extract by Mask’.
- Click and drag the **Extract by Mask** tool onto the model interface.
- Double click the **Extract by Mask** tool to open. A pop-up window appears where the tool’s parameters can be added. In the **Parameters** tab, indicate the **Input raster** by clicking on the drop-down arrow in the field to select the available **Model Variable** (the output from the **Iterate Rasters** tools) in this case was ‘ORA2021_252955.JPG’. Add the **Input raster or feature mask data** by navigating to the folder containing the masking layer to select and click **OK**, in this case ‘on_off_merge_no_freeways_buffer.shp’. Remember, this layer was buffered a 100 ft before utilizing in masking. Input In the **Output raster** field, type ‘%Name%’ as the output name, leave the **Extraction Area** as ‘Inside’, maintain all other parameters as default and click **OK**.
- Once again, go back to the **Geoprocessing** tools search box and type ‘Raster to other format’.
- Click and drag the **Raster To Other Format** tool onto the model interface.
- Double click the **Raster To Other Format** tool to open. A pop up window appears where the tool’s parameters can be added. In the **Parameters** tab, indicate the **Input Rasters** by clicking on the drop-down arrow in the field to select the **Model Variable** (the output from the **Extract by Mask** tool) which is ‘%Name%’. For **Output Workspace**, navigate to the directory where your output folder for the masked files is to select and click **OK**. Change the **Output raster format** to ‘JPEG’. Maintain all other default fields and click **OK**.
- Click on **Auto Layout** in the **View** group of the **ModelBuilder** tab to automatically arrange the elements in the model.
- Before running the model, verify that all data elements and parameters are valid by clicking on **Validate** in the **Run** group of the **ModelBuilder** tab. Correct all errors if any.
- Run the model by clicking on **Run** in the **Run** group of the **ModelBuilder** tab.

## 4.4. Data Preparation for Model Training

### 4.4.1. Creating Training Samples

Quality imagery data are needed to train deep learning models to detect various school zone features. The procedure for the image preparation is shown in Figure 4-1a which illustrates the process starting with school zone identification towards image or training data extraction. A lot of time and effort were spent to create the training data for the model creation since a greater percentage of the model’s performance relies on the amount, quality, and diverseness of the training data used for training the model. Similarly, extra time and work were required during the model training process where different parameters were tested, and the optimum one was selected to train the model. Two different forms of training data with different classes were prepared for the study. The labels used for describing the training data for the turning lane model were “left_only”, “right_only”, “center”, and “none”, which were classes 1, 2, 8, and 12, respectively. On the other hand, the labels used for describing the training data for school zone model were “schoolzones” and “not_school”, which were class 1 and class 0, respectively. It is important to note that object detection models perform well when trained using clear and distinct features. Some of the training features were duplicated to increase their proportions in the training data set. To avoid model overfitting, bias, and limited model generalization as a result of duplication, a data augmentation technique named rotation was utilized to increase dataset diversity and quantity without solely resorting to duplication. Data rotation method was used to randomly rotate the training data features at desired angle producing extra training data in various rotations. This helps the model detect objects in various positions and orientations. This was very useful since the study features such as left only and right only lanes can be observed in all travel directions at each intersection. A 90 degrees rotation was applied to the training data. The metadata of the exported labels were in the pascal visual objects format. The input mosaic data was made up of high-resolution aerial images of the entire State of Florida with tile size of 5,000 x 5,000 sq ft.

To create training samples in ArcGIS Pro, the following steps are required:

**(a) Creating training classes (Figure 4-1a)**

- Start **ArcGIS Pro** and open the recent project.
- On the **View** tab, go to the **Windows** group and click on the **Contents** to open the **Contents Pane** if Contents Pane is not already open.
- Following the previous steps, create a mosaic dataset out of the masked images.
- In the **Contents Pane**, select the mosaic data set (make sure it is highlighted), and go to the **Imagery** tab. In the **Image Classification** group, click on the **Classification Tools** to expand the tool options. Select **Label Objects for Deep Learning** to open tool.
- Using the mouse, move to one end of the **Image Classification** window to expand the **Image Classification** window in order to make visible all the available buttons in the **Image Classification** toolbar.
- On the toolbars in the **Layer** tab, click on the plus symbol to add new class to the classification schema.
- In the open window, create a name for the class in the **Name** field and a value for the class in the **Value** field. Please note, all name and value entries should be unique. Click on the drop-down arrow to select preferred color for the class. Optionally, you can add an alias and a description to the class. Repeat the process to add all classes to the schema. Make sure the **New Schema** field is highlighted when creating classes.

- Save the classification schema by clicking on the **Save** button. Create the schema name in the **Name** field, add a brief description in the **Description** field, add the output folder location in the **Output Location** field and click **Save**.
- Begin bounding box creation.

**(b) Creating bounding box (Figure 4-1b)**

- On the **Image Classification** toolbar, select the **Rectangle** button (for rectangular bounding box).
- Click on the class that is about to be labeled in the schema field to ensure that the class is highlighted.
- Use the mouse to zoom into the features on the images.
- Click and draw the bounding box around the feature in the image. Repeat the process for all features in the image. Remember to click and highlight the required class when labelling any feature within that class. Note that labelling can be done using the entire mosaic layer view or individual footprints view. This can be achieved by switching from **Layer** tab to **Image Collection** tab.
- To delete a wrongly classified feature or wrongly drawn bounding box, go to the table in the **Labeled Objects** field found at the lower part of the **Image Classification** window. Click on the wrong label to select and highlight. Click on the 'X' icon or the **Delete** button on the **Labeled Objects** tab to remove the selected label. It is important to save training samples into a feature class to avoid losing labelled data.
- To save training samples, click on the **Save** button on the **Labeled Objects** tab. Navigate to any output file Geodatabase. Create a name for the training samples and click **Save**. Remember to click on the **Save** button each time the samples are updated to avoid any data losses.

**(c) Exporting training samples (Figure 4-1c)**

- Training samples can be directly exported using the **Export Training Data** tab in the **Image Classification** window. Otherwise, go to the Geoprocessing tool search box and type 'Export training data for deep learning'. Click on the **Export Training Data For Deep Learning** tool in the search results to open it.
- On the **Parameters** tab, add the created mosaic dataset in the **Input Raster** field by clicking on the drop-down arrow and selecting the mosaic dataset or using the browse button to navigate to the folder containing the images. Indicate the output folder where the exported training dataset image chips and metadata will be stored in the **Output Folder** field by clicking on the folder icon and navigating to an existing or created output folder. In the **Input Feature Class** or **Classified Raster Or Table** field, add the saved training sample layer by navigating to the geodatabase where the file was stored and selecting it. Set the **Class Value Field** to 'Classvalue'. To set the **Image Format** field, which is the output raster format, click on the drop-down arrow in the field and select 'TIFF format'. The rotation

angle value in the **Rotation Angle** field can be set to 90 and **Metadata Format** can be set to ‘PASCAL Visual Object Classes’ which is compatible with YOLO models. Leave all other fields as default and click **Run**.

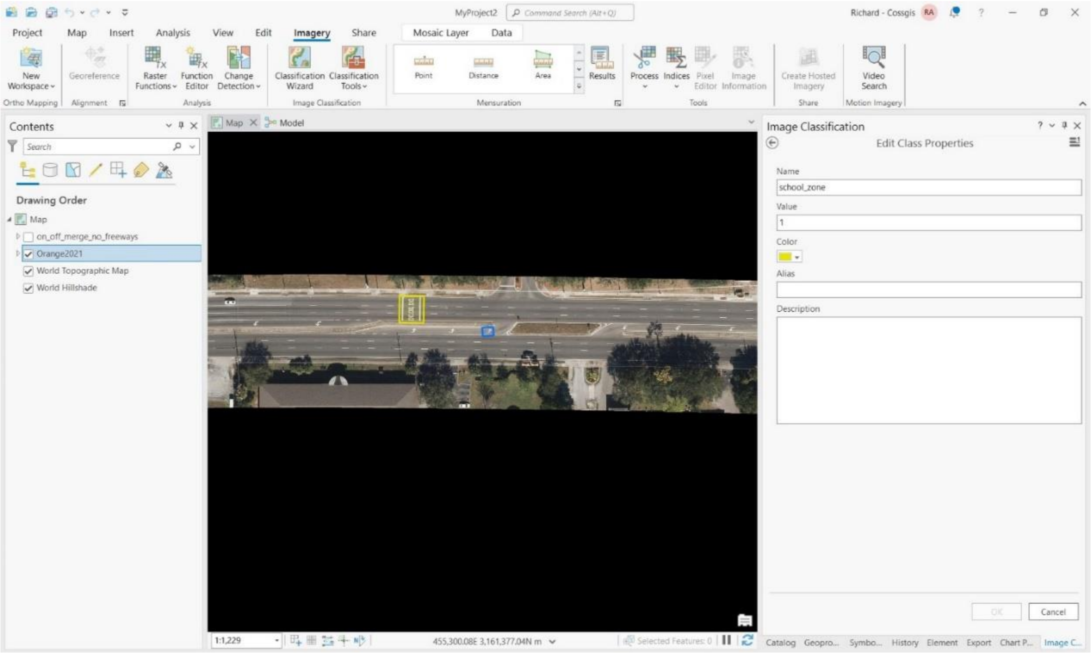


Image Classification

Edit Class Properties

Name

school_zone

Value

1

Color

Alias

Description

(a)

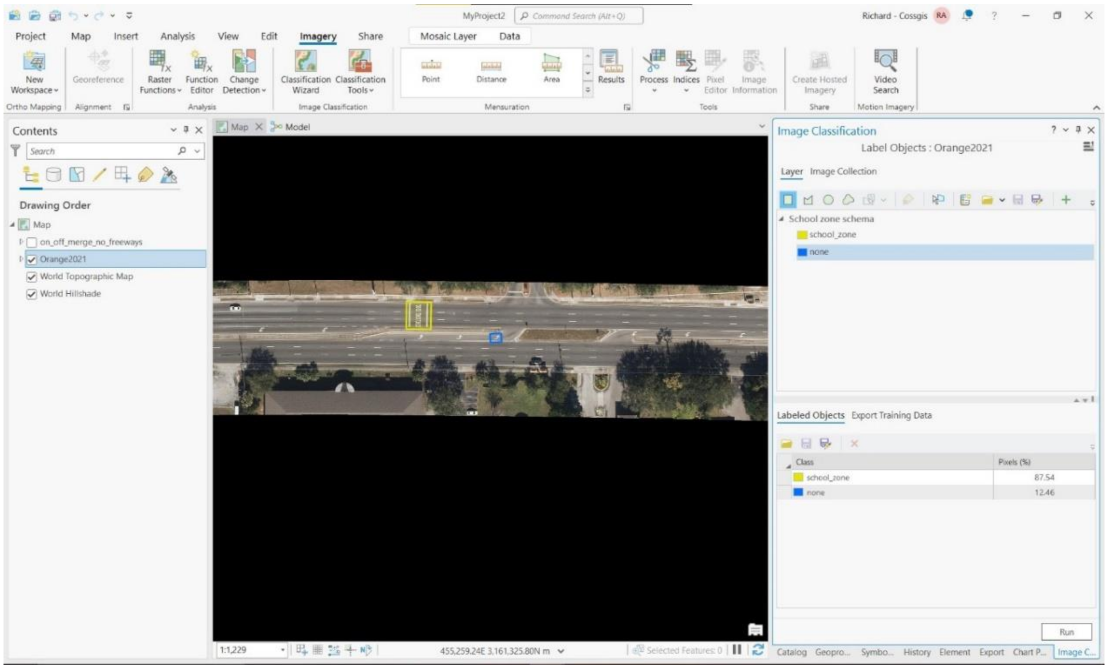


Image Classification

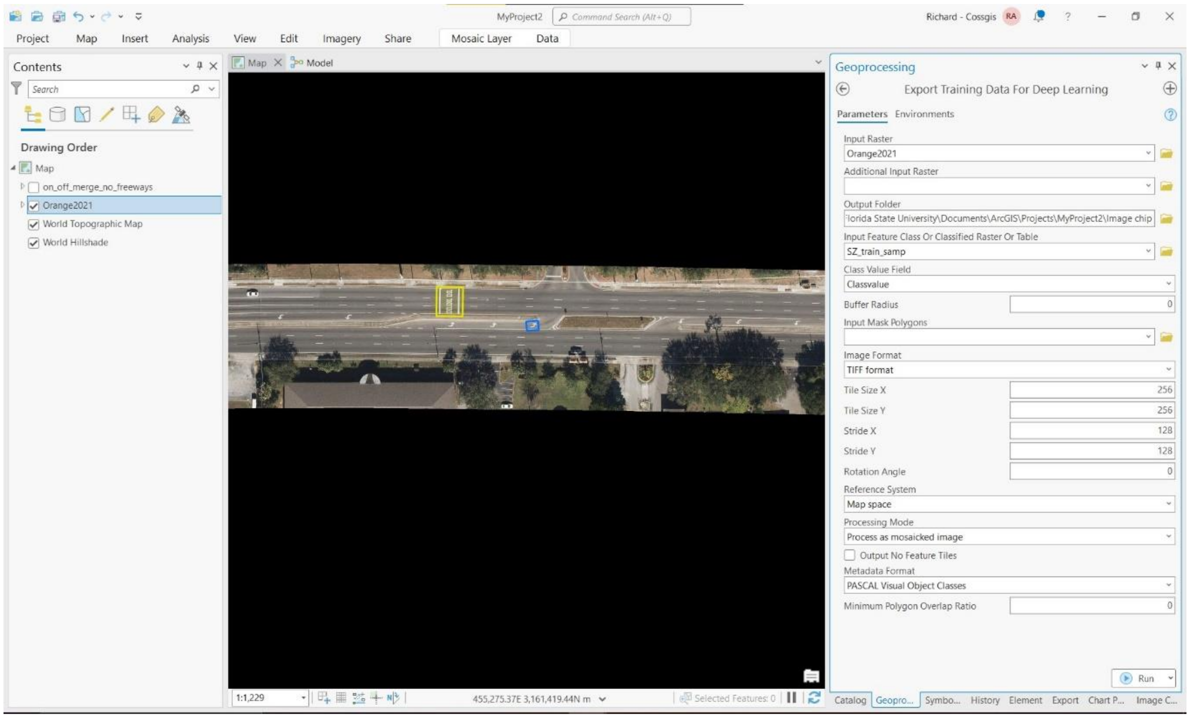
Label Objects : Orange2021

School zone schema
- school_zone
- none

| Class | Pixels (%) |
|---|---|
| school_zone | 87.54 |
| none | 12.46 |

(b)

**Figure 4-1**: (a) Creating training classes, (b) creating bounding boxes, and (c) exporting training data



Map
☐ on_off_merge_no_freeways
☑ Orange2021
☑ World Topographic Map
☑ World Hillshade

1:1,229
455,275.37E 3,161,419.44N m
Selected Features: 0

Geoprocessing
Export Training Data For Deep Learning
Parameters Environments

Input Raster: Orange2021
Additional Input Raster:
Output Folder: 'lorida State University\Documents\ArcGIS\Projects\MyProject2\Image chip
Input Feature Class Or Classified Raster Or Table: SZ_train_samp
Class Value Field: Classvalue
Buffer Radius: 0
Input Mask Polygons:
Image Format: TIFF format
Tile Size X: 256
Tile Size Y: 256
Stride X: 128
Stride Y: 128
Rotation Angle: 0
Reference System: Map space
Processing Mode: Process as mosaicked image
☐ Output No Feature Tiles
Metadata Format: PASCAL Visual Object Classes
Minimum Polygon Overlap Ratio: 0

(c)

**Figure 4-1:** (a) Creating training classes, (b) creating bounding boxes, and (c) exporting training data

## 4.5. Training YOLOv3 Detection Model

The neural network model known as You Only Look Once (YOLO) is mostly employed for real-time object identification. When compared to other object identification models like Region-based Convolutional Neural Network (R-CNN) or Faster R-CNN, YOLO's quickness is by far its biggest advantage. In other words, YOLO outperforms R-CNN and Faster R-CNN architectures by 1000 and 100 times, respectively (Redmon and Farhadi, 2018). This is largely because other object identification models classify potential regions first, then identify the item(s) based on the classification probability of those regions, requiring hundreds of network evaluations to be performed on a single picture. On the other hand, the YOLO model may make predictions based on the entire context of the image and look for the entire image using just one network analysis. The initial version of the YOLO object identification method was released in 2016 (Redmon et al., 2016), the second version (YOLOv2) (Redmon and Farhadi, 2017), and the third version (YOLOv3) (which improved multi-scale predictions), respectively (Redmon and Farhadi, 2018). YOLOv4 (Bochkovskiy et al., 2020) and YOLOv5 (Jocher et al., 2020) were published in 2020 and 2022, respectively. YOLOv3 was used in this investigation since our dataset does not have a scalability problem and because it is simple to apply using the available sources. Unlike other detection models, YOLOv3 uses Darknet-53 as its backbone. Darknet-53 is a deeper version of Darknet-19 which was the backbone of YOLOv2. The architecture of the backbone has 53 convolutional layers which improves model's accuracy and speed (Tsang, 2019). Making the feature extractor perform better and 2x faster than ResNet152 (Redmon and Farhadi, 2018). Figure 4 shows images that are produced at the end of the detection layers in

YOLOv3 undergoes a three-scale detection process which involves applying a 1x1 detection kernel on feature maps located at three different areas and sizes within the network. The object detection model was trained with manually labelled features on the aerial images using the Deep-Learning Toolbox in ArcGIS Pro consisting of all classes. The classes were clearly distinguished between the features observed in an input image by the detector. This clear classification improves the model's detection performance. Based on these classes, the output data were categorized into either school zones, not-school zone, or left or right turn lanes using a class value field. For a fair model performance, the counties selected for training data creation were ones having variations in image resolutions and roadway infrastructure development. These counties selected also had varying markings on the roadways.

The model's tunable parameters and hyperparameters include the learning rate, input image size, number of epochs, batch size, anchor box size and ratios, and training and test data percentages. The validation loss and mean average precision were computed on the validation set, which is made up of 30% of the input training dataset. The object detection parameters that have a high impact on the object detection are the batch size, the learning rate, and the training epoch. The learning rate represents the rate at which the model learns new information about the training data by overwriting the existing information with newly acquired information during the training process. Choosing an ideal learning rate strikes a compromise between precision and convergence speed. A model can be trained effectively, with excellent convergence characteristics, and with the greatest level of accuracy by using an optimal learning rate. The batch size describes the number or bundle of training samples or images selected and processed for training each iteration. The selection of batch size depended on factors like the size of dataset, the model's complexity, and the available computer hardware's resources. For instance, with a larger batch size, more data was processed in parallel, and the training process was faster. However, this requires more computer memory. Smaller batch size, on the other hand, increases randomness in training data selection during the training process. This enhances the model's performance on new data and therefore improves predictions. Also, the anchor box represents the size, shape, and location of the object being detected. The epoch number was the number of iterations the model would be trained. It describes the number of times the training dataset was passed forward and backward through the neural network once. The percentage of training data actually used to train the model was 70%. This dataset was randomly chosen to train the model.

A 30% split of the training dataset was used to assess the detection model's performance. The size of the training dataset was mainly considered when determining the training and test data split. With a training dataset >10,000, a 20%-30% validation size provides enough randomly sampled data to test the model's performance. As a valid prediction, a default 50% overlap of the label and detection bounding boxes was accepted. Following that, recall and precision were determined as the true prediction rate among the original labels and all other predictions. In YOLOv3, the likelihood of an input belonging to a certain label was calculated using individual logistic classifiers instead of the SoftMax function which was used in previous versions (Redmon et al., 2016; Redmon and Farhadi, 2017). During the model training, a binary cross-entropy loss was used for each label to calculate the classification loss, rather than using mean square error. Therefore, logistic regression was used to predict both object confidence and class predictions. This approach reduces the complexity of computations involved and improves the model's performance (Redmon and Farhadi, 2018). The validation loss examined how well the model fits new data, whereas the training loss evaluated how well the model fits training data.  A high loss indicates that the product of the model has errors while a low loss value shows there are fewer errors present in the model.

To train object detection model in ArcGIS Pro:

**(a) Training object detection model using tabs (Figure 4-2)**

- Start **ArcGIS Pro** and open the recent project.
- Go to the **Geoprocessing** group in the **Analysis** tab and click on **Tools** to open the Geoprocessing tools search box. In the search box, type 'Train deep learning model'.
- Click on the **Train Deep Learning Model** tool in the search results to open it.
- In the **Parameters** tab, navigate to the folder containing the image chips and add it in the **Input Training Data** field by using the browse folder button. You can add multiple input training data folders if you have your training data grouped in several folders. Indicate the output folder where the trained model will be stored in the **Output Model** field by clicking on the folder icon and navigating to an existing or created output folder. Type the number of epochs value into the **Max Epochs** field for instance '100'.
- In the **Model Type** field of the **Model Parameters** tab, select 'YOLOv3 (Object detection)'.
- Input the batch size in the **Batch Size** field for instance '16'.
- In the **Model Arguments** field, maintain the default values and add another field. In the **Name** column, type 'monitor' and 'average_precision' in the corresponding **Value** column.
- Change the backbone model to 'DarkNet-53' to correspond to YOLOv3.
- Set the **Validation** field value to '30%' and uncheck **Stop when model stops improving** checkbox.
- In the Environments tab, set the **Processor Type** to GPU using the drop down arrow if your computer has a GPU, otherwise leave it blank.
- Leave all other fields as default and click **Run**.

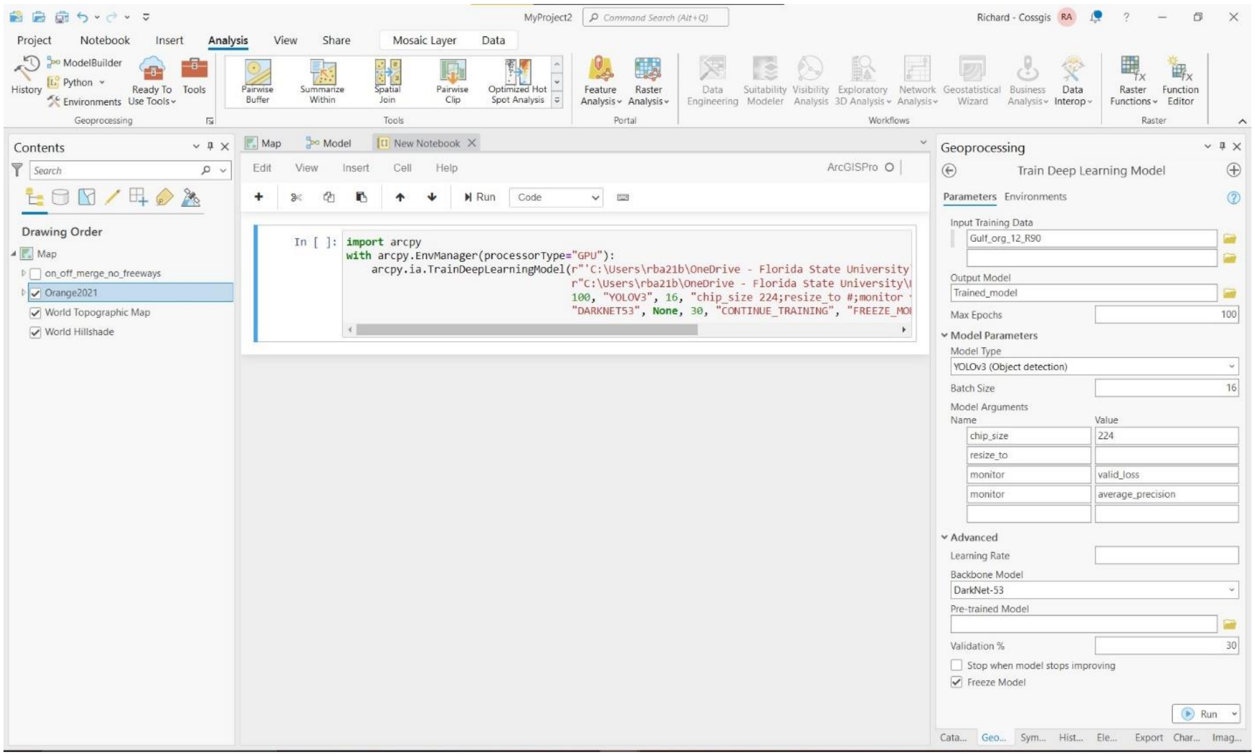
**(b) Training object detection model using python notebook**

- Open python repository (jupyter, google colab, etc) or use ArcGIS Pro python notebook.
- To open the python notebook, go to the **Geoprocessing** group in the **Analysis** tab and click on **Python** to open the python notebook to run script.
- Remember to install arcpy module before running the script in any environment outside ArcGIS Pro.
- Run the following script:

```
input_f = r'''C:\Users\rba21b\OneDrive – Florida State
University\Desktop\Turning_lane\Gulf_org_12_R90'''
```

```
output_f = r"C:\Users\rba21b\OneDrive - Florida State
University\Documents\ArcGIS\Projects\MyProject2\Trained_model"
epochs = 100
model = "YOLOV3"
batch_size = 16
lr = None
backbone = "DARKNET53"
pretrained_mod = None
val = 30
stop = "CONTINUE_TRAINING"
freeze = "FREEZE_MODEL")
import arcpy
with   arcpy.EnvManager(processorType="GPU"):
arcpy.ia.TrainDeepLearningModel(input_f, output_f, epochs, model, batch_size,
"chip_size 224;resize_to #;monitor valid_loss;monitor average_precision", lr, backbone,
pretrained_mod, val, stop, freeze)
```

- Replace folder directories into the folders you are working in before running script.



```
In [ ]: import arcpy
        with arcpy.EnvManager(processorType="GPU"):
            arcpy.ia.TrainDeepLearningModel(r"'C:\Users\rba21b\OneDrive - Florida State University
                                            r"C:\Users\rba21b\OneDrive - Florida State University\
                                            100, "YOLOV3", 16, "chip_size 224;resize_to #;monitor
                                            "DARKNET53", None, 30, "CONTINUE_TRAINING", "FREEZE_MO
```

Geoprocessing

Train Deep Learning Model

Parameters Environments

Input Training Data: Gulf_org_12_R90

Output Model: Trained_model

Max Epochs: 100

Model Parameters

Model Type: YOLOv3 (Object detection)

Batch Size: 16

Model Arguments

| Name | Value |
|---|---|
| chip_size | 224 |
| resize_to | |
| monitor | valid_loss |
| monitor | average_precision |

Advanced

Learning Rate:

Backbone Model: DarkNet-53

Pre-trained Model:

Validation %: 30

□ Stop when model stops improving

☑ Freeze Model

**Figure 4-2:** Training YOLO v3 object detection model

## 4.6. Detecting School Zones and Turning Lanes

After model training, the detectors were initially tested on individual photos and later applied to the entire State of Florida on a county level. For the school zone detector, a threshold of 0.1 was employed to lessen false positives. However, due to the complexity of multi class models, a threshold of 0.05 was employed in turning lane detection to capture all features detected with very low confidence levels. It was important to note that lower detection thresholds generally increase false positives which results in lower precision, increases computational workload which increases detection time and increases detection of irrelevant features or noise. However, lower detection thresholds also increase model's sensitivity to detect faint or partially visible features, while increasing recall. With higher recall, the model was more likely to identify and detect all instances of the target object class and reduce the chances of missing any objects. For all models, more than 10% overlap between two bounding boxes was avoided to minimize duplicate detections. To lessen information loss from the margins of the detection pictures, a padding parameter of 56 was added to the boundary pixels on the outside of the image. The detector was trained on 256 x 256 sub-images with a stride of 128 by 128 pixels, yielding a 50% overlap with the following image chip, and a resolution of 0.5 feet per pixel. The image was rescaled and processed with black pixels to have the same attributes if the provided image had a different size or resolution. It should be noted that utilizing huge photos with object detection techniques is impractical since the cost of computing grows rapidly.

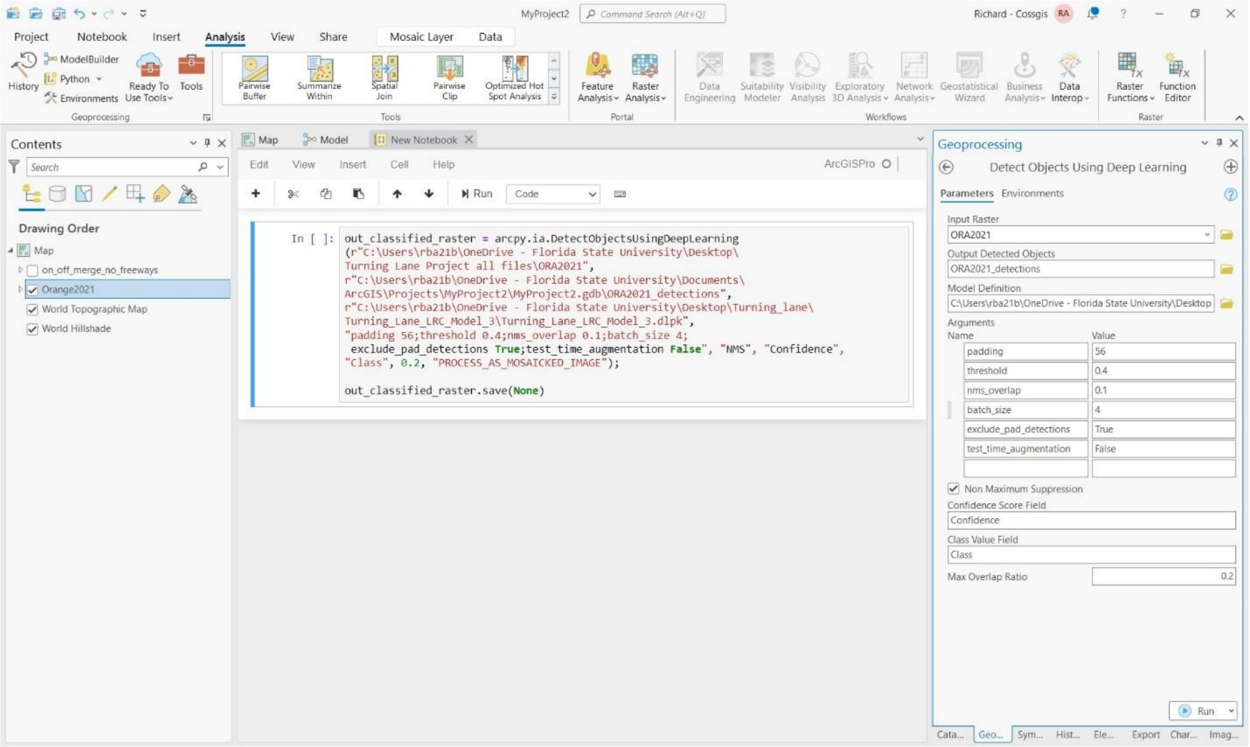
The mapping procedure was carried out at the county level because the detector performs very well on single photos. The detection images were first selected and iterated through the detector. An output file of all the identified school zones in that county was created once all photos had been sent to the detector. Confidence scores were included in the output file. This file was used to map school zones. Note that the model can detect school zone and turning lane markings from images with a resolution ranging from 1.5 ft down to 0.25 ft or higher. However, the model has not been tried on any images with resolution lower than the ones provided by the Florida APLUS system.

To perform detections, the following steps are required:

**(a) Performing object detection (Figure 4-3)**

- Start **ArcGIS Pro** and open the recent project.
- Go to the **Geoprocessing** group in the **Analysis** tab and click on **Tools** to open the Geoprocessing tools search box. In the search box, type 'Detect objects using deep learning'.
- Click on the **Detect Objects Using Deep Learning** tool in the search results to open it.
- On the **Parameters** tab, navigate to the folder containing the images and add it in the **Input Raster** field by using the browse folder button. Rename the output folder where the trained model will be stored in the **Output Detected Objects** field by clicking on the folder icon and navigating to an existing or created output folder.
- Use the browse folder button next to the **Model Definition** field to navigate to the folder containing the trained model files and add the '.dlpk' file.
- Set the required model **Arguments Name** and **Value**.
- Check the **Non Maximum Suppression** box. Ensure that the **Confidence Score Field** has 'Confidence' and the **Class Value Field** has 'Class'. **Set Max Overlap Ratio** value to 0.2.

- On the **Environments** tab, set **Processor Type** to ‘GPU’.
- Leave all other fields as default and click **Run**.



```
In [ ]: out_classified_raster = arcpy.ia.DetectObjectsUsingDeepLearning
        (r"C:\Users\rba21b\OneDrive - Florida State University\Desktop\
        Turning Lane Project all files\ORA2021",
        r"C:\Users\rba21b\OneDrive - Florida State University\Documents\
        ArcGIS\Projects\MyProject2\MyProject2.gdb\ORA2021_detections",
        r"C:\Users\rba21b\OneDrive - Florida State University\Desktop\Turning_lane\
        Turning_Lane_LRC_Model_3\Turning_Lane_LRC_Model_3.dlpk",
        "padding 56;threshold 0.4;nms_overlap 0.1;batch_size 4;
         exclude_pad_detections True;test_time_augmentation False", "NMS", "Confidence",
        "Class", 0.2, "PROCESS_AS_MOSAICKED_IMAGE");

        out_classified_raster.save(None)
```

Geoprocessing

Detect Objects Using Deep Learning

Parameters Environments

Input Raster: ORA2021

Output Detected Objects: ORA2021_detections

Model Definition: C:\Users\rba21b\OneDrive - Florida State University\Desktop

Arguments

| Name | Value |
|---|---|
| padding | 56 |
| threshold | 0.4 |
| nms_overlap | 0.1 |
| batch_size | 4 |
| exclude_pad_detections | True |
| test_time_augmentation | False |
| | |

☑ Non Maximum Suppression

Confidence Score Field: Confidence

Class Value Field: Class

Max Overlap Ratio: 0.2

**Figure 4-3:** Detecting features from aerial imagery

## 4.7. Post-processing

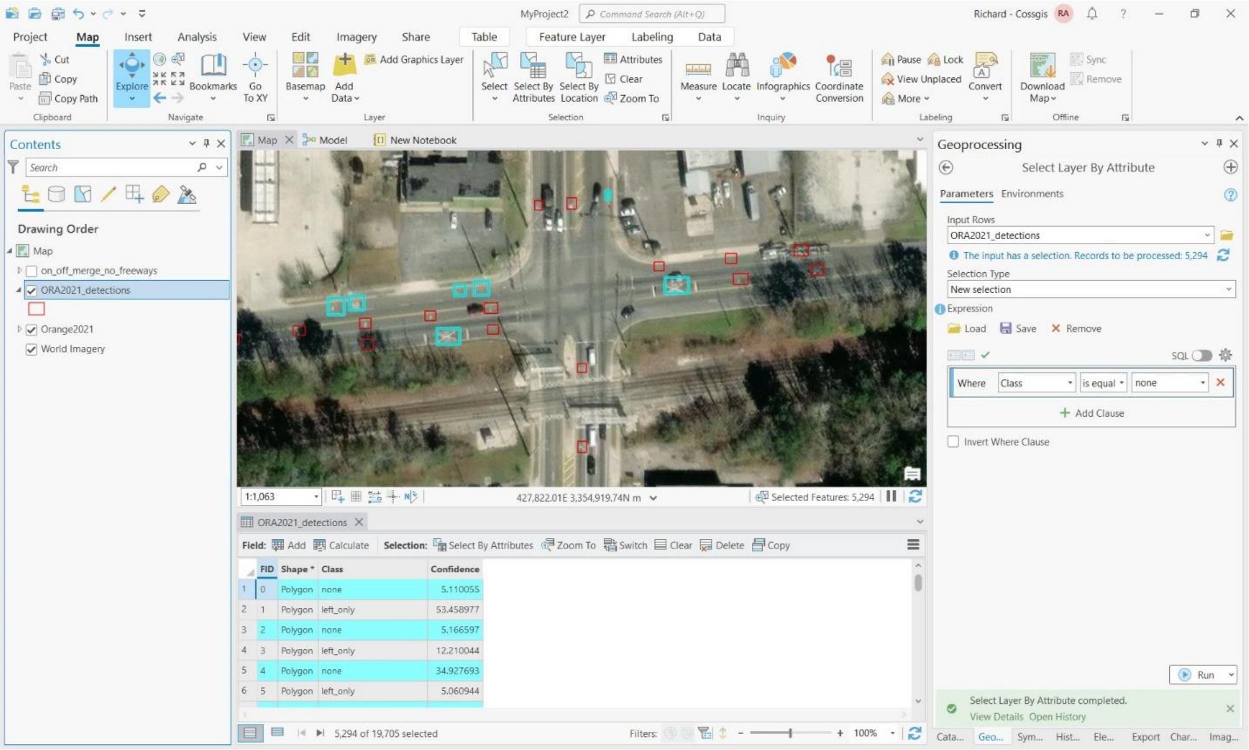
The redundant detections caused by the overlapping distance on sliding images are removed at the post-processing stage (Figure 4-4). For various analytical objectives, school zones and turning lanes found on state and local roadways can also be divided into categories. Non maximum suppression was used to filter the detected school zones by selecting and keeping school zone detections that overlap and have the highest confidence level. Detected features were converted from polygon shapefiles into point shapefiles for subsequent analysis.

To perform post-processing in ArcGIS Pro, the following steps are required:

### (a) Filtering detections

- Go to the **Geoprocessing** group in the **Analysis** tab and click on **Tools** to open the Geoprocessing tools search box. In the search box, type ‘Select layer by attribute’.
- Click on the **Select Layer By Attribute’** tool in the search results to open it.
- In the **Input Rows** field of the **Parameters** tab, use the browse folder button navigate to the folder containing the output shapefile from the detections to select the ‘.SHP’ file and click **OK**.
- Selection **Type** field should contain ‘New Selection’.

- Use the SQL expression in the **Expression** field to select and remove the ‘none’ class from the detections. In the **Where** field, select ‘Class’, select ‘is equal to’ in the next field and type ‘none’. Click on **Run**. If you must filter by confidence level, click on **Add Clause** button, and include the filter boundaries.
- Export selected features by going to the **Geoprocessing** tools search box, type ‘Export features’. Click on the **Export Features** tool to open it.
- On the **Parameters** tab, add the detections shapefile in the **Input Features** field by clicking on the drop-down arrow and selecting the detections shapefile. Indicate the output folder where the exported feature class will be stored in the **Output Feature Class** field by clicking on the folder icon and navigating to an existing or created output folder or renaming the generated output folder name.
- Optionally, you can select which fields in the attribute table to keep in the **Field Map** section, and sort them using the **Sort** field in the **Sort** section.
- Leave all other fields as default and click **Run**.



Select Layer By Attribute

Input Rows: ORA2021_detections

The input has a selection. Records to be processed: 5,294

Selection Type: New selection

Expression

Where Class is equal none

Add Clause

☐ Invert Where Clause

Run

Select Layer By Attribute completed.

View Details Open History

| | FID | Shape * | Class | Confidence |
|---|---|---|---|---|
| 1 | 0 | Polygon | none | 5.110055 |
| 2 | 1 | Polygon | left_only | 53.458977 |
| 3 | 2 | Polygon | none | 5.166597 |
| 4 | 3 | Polygon | left_only | 12.210044 |
| 5 | 4 | Polygon | none | 34.927693 |
| 6 | 5 | Polygon | left_only | 5.060944 |

5,294 of 19,705 selected

**Figure 4-4: Post-processing**

# 5. CONCLUSIONS

This study investigated the utilization of computer vision tools for roadway geometry extraction focusing on Florida school zones and turning lanes as proof of concept. This has been a creative approach that used computer vision technology to possibly replace labor- and error-intensive traditional manual inventory. The created system could extract recognizable turning lane and school zone marks from images using high quality images. By removing the requirement for a human inventory procedure and improving highway geometry data quality by removing mistakes from manual data entry, the findings would assist stakeholders in saving money. The benefits of such roadway data extraction from imagery for transportation agencies are numerous and include identifying markings that are outdated and invisible, comparing the locations of turning lanes or school zones with other geometric features like crosswalks, and analyzing, crashes that take place close to these locations.

Based on the results of this assignment, there are several significant constraints and recommendations for the future. Aerial images of roadways with tree canopies and occlusion limit the identification of markings. The developed model can be used to collect roadway geometry data such as school zones and turning lanes precisely left, right and center which are part of the roadway geometry inventory data usable in HSM and Model Inventory of Roadway Elements (MIRE). Turning lane and school zone markings that are old or have missing inscriptions can be identified and repainted. In future work, this model will be improved and extended to detect and extract other roadway geometric features particularly turning lanes such as right and through, left and through, and through lanes. Also, future works will include integration of extracted school zone, left only, right only, and center lanes with crash data, traffic data, and demographics for a more detailed analysis.

## REFERENCES

Altobello, M., Thurber, M., Lefler, N., McGee, H. W., & DeLucia, B. H. (2013). *Development of a Structure for a MIRE Management Information System* (No. FHWA-SA-13-007). United States. Federal Highway Administration. Office of Safety.

USDOT (2013). *Gap Analysis Report: FHWA Highway Safety Program*. Retrieved November 5, 2020, from https://safety.fhwa.dot.gov/rsdp/downloads/tsp_gap_analysis_rpt.pdf

Appel, R., Belongie, S., Perona, P., & Doll, P. (2014). Fast Feature Pyramids for Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *36*, 1532–1545. https://doi.org/10.1109/TPAMI.2014.2300479

Balali, V., Ashouri Rad, A., & Golparvar-Fard, M. (2016). Detection, classification, and mapping of U.S. traffic signs using google street view images for roadway inventory management. *Visualization in Engineering*, *3*(1), 1–18. https://doi.org/10.1186/s40327-015-0027-1

Balali, V., & Golparvar-Fard, M. (2015). Recognition and 3D localization of traffic signs via image-based point cloud models. In *Computing in Civil Engineering 2015* (pp. 206-214).

Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*. https://doi.org/10.48550/arXiv.2004.10934

Caddell, R., Hammond, P., & Reinmuth, S. (2009). Roadside features inventory program. *Washington state department of transportation*.

California Transportation Commission (Caltrans). (2018). *California Transportation Asset Management Plan - Fiscal Years 2017/18-2026/27*. 126. https://dot.ca.gov/-/media/dot-media/programs/asset-management/documents/20190726-am-finalcaliforniatamp-a11y.pdf

Campbell, A., Both, A., & Sun, Q. C. (2019). Detecting and mapping traffic signs from Google Street View images using deep learning and GIS. *Computers, Environment and Urban Systems*, *77*, article 101350.

Chow, T. E., & Hodgson, M. E. (2009). Effects of lidar post-spacing and DEM resolution to mean slope estimation. *International Journal of Geographical Information Science*, *23*(10), 1277-1295.

Cook, R., Harris, M., & Barber, D. (2021). Project deliverables. *Management Consulting Projects*, 51–58. https://doi.org/10.4324/9781003174516-6

Dai, J., Wang, Y., Li, W., & Zuo, Y. (2020). Automatic method for extraction of complex road intersection points from high-resolution remote sensing images based on fuzzy inference. *IEEE Access*, *8*, 39212-39224.

Dollár, P., Wojek, C., Schiele, B., & Perona, P. (2012). Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *34*(4), 743–761. https://doi.org/10.1109/TPAMI.2011.155

Experience, T. V. (2018). Assembling MIRE Data The Vermont Experience.

FDOT Surveying and Mapping Office. (2020). Aerial Photography Look-Up System (APLUS). Retrieved February 17, 2022, from https://fdotewp1.dot.state.fl.us/AerialPhotoLookUpSystem/

FHWA. (2018). *Use of Unmanned Aerial Systems ( UAS ) by State DOTs February 27 , 2018 Peer Exchange* (FHWA-HIF-18-060).

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448).

Girshick, R., Donahue, J., Darrell, T., Berkeley, U. C., & Malik, J. (2012). (r-cnn) Rich feature hierarchies for accurate object detection and semantic segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2–9. https://doi.org/10.1109/CVPR.2014.81

Graham, L. (2010). Mobile mapping systems overview. *Photogrammetric engineering and remote sensing*, *76*(3), 222-228.

Gong, J., Zhou, H., Gordon, C., & Jalayer, M. (2012). Mobile terrestrial laser scanning for highway inventory data collection. In *Computing in Civil Engineering (2012)* (pp. 545-552).

Guan, H., Li, J., Cao, S., & Yu, Y. (2016). Use of mobile LiDAR in road information inventory: A review. *International Journal of Image and Data Fusion*, *7*(3), 219-242.

Gunaratne, M., Mraz, A., & Sokolic, I. (2003). *Study of the feasibility of video logging with pavement condition evaluation* (No. Report No. BC-965,).

Habibi Aghdam, H., Jahani Heravi, E., & Puig, D. (2016). A practical approach for detection and classification of traffic signs using Convolutional Neural Networks. *Robotics and Autonomous Systems*, *84*, 97–112. https://doi.org/10.1016/j.robot.2016.07.003

Jalayer, M., Zhou, H., Gong, J., Hu, S., & Grinter, M. (2014). A comprehensive assessment of highway inventory data collection methods. *Journal of the Transportation Research Forum.* Vol. 53, No. 1424-2016-117955, pp. 73-92.

Jalayer, M., Hu, S., Zhou, H., & Turochy, R. E. (2015). Evaluation of Geo-Tagged Photo and Video Logging Methods to Collect Geospatial Highway Inventory Data. *Papers in Applied Geography*, *1*(1), 50-58.

Jaselskis, E. J., Gao, Z., & Walters, R. C. (2005). Improving transportation projects using laser scanning. *Journal of construction engineering and management*, *131*(3), 377-384.

Jeyapalan, K. (2004). Mobile digital cameras for as-built surveys of roadside features. *Photogrammetric Engineering & Remote Sensing*, *70*(3), 301-312.

Jocher, G., Nishimura, K., Mineeva, T., & Vilariño, R. (2020). yolov5. Code repository. *Code repository*. Available online: https. *github. com/ultralytics/yolov5 (accessed on 1 October 2022).*

Kargah-Ostadi, Nima, Ammar Waqar, and Adil Hanif. "Automated real-time roadway asset inventory using artificial intelligence." *Transportation Research Record* 2674, no. 11 (2020): 220-234.

Karimi, H. A., Khattak, A. J., & Hummer, J. E. (2000). Evaluation of mobile mapping systems for roadway data collection. *Journal of computing in civil engineering*, *14*(3), 168-173.

Kathuria, A. (2018). What’s new in YOLO v3. *Towards data science*, *23*.

Ke, R., Kim, S., Li, Z., & Wang, Y. (2015). Motion-vector clustering for traffic speed detection from UAV video. *2015 IEEE 1st International Smart Cities Conference, ISC2 2015*, 1–5. https://doi.org/10.1109/ISC2.2015.7366230

Ke, R., Li, Z., Kim, S., Ash, J., Cui, Z., & Wang, Y. (2017). Real-Time Bidirectional Traffic Flow Parameter Estimation from Aerial Videos. *IEEE Transactions on Intelligent Transportation Systems*, *18*(4), 890–901. https://doi.org/10.1109/TITS.2016.2595526

Khattak, A. J., Hummer, J. E., & Karimi, H. A. (2001). Evaluation of roadway infrastructure data collection technologies. *Public Works Management & Policy*, *6*(1), 18-31.

Kim, E.-J., Park, H.-C., Ham, S.-W., Kho, S.-Y., & Kim, D.-K. (2019). Extracting Vehicle Trajectories Using Unmanned Aerial Vehicles in Congested Traffic Conditions. *Journal of Advanced Transportation*, *2019*, 1–16. https://doi.org/10.1155/2019/9060797

Krzywinski, M., Schein, J., Birol, I., Connors, J., Gascoyne, R., Horsman, D., ... & Marra, M. A. (2009). Circos: An Information Aesthetic for Comparative Genomics. *Genome Research*, *19*(9), 1639-1645.

Kwayu, K. M., Toth, M., & Himmelein, A. (2022). A Scalable Deep Learning Framework for Extracting Model Inventory of Roadway Element Intersection Control Types From Panoramic Images. *Transportation Research Record*, *2676*(5), 630-642.

Lefler, N., Council, F., Harkey, D., Carter, D., McGee, H., & Daul, M. (2010). *Model inventory of roadway elements-MIRE, version 1.0* (No. FHWA-SA-10-018).

Ma, Y., Wu, X., Yu, G., Xu, Y., & Wang, Y. (2016). Pedestrian detection and tracking from low-resolution unmanned aerial vehicle thermal imagery. *Sensors*, *16*(4), 446.

Mimbela, L. E. Y., & Klein, L. A. (2000). Summary of vehicle detection and surveillance technologies used in intelligent transportation systems.

Miotto, A. (2000). A better image. *Civil Engineering*, *70*(1), 42.

Nagarajan, S., Sairam, N., & Antoniou, C. (2016). Integration of images and laser scanning data for automated 3D road feature extraction. *Surveying and Land Information Science*, *75*(2), 49-63.

Orange County Public Schools (OCPS). (2022). https://www.ocps.net/. Accessed Jul. 28, 2022.

United States Census Bureau (US Census). (2020). *Population Estimates*. https://www.census.gov/quickfacts/duvalcountyflorida. Accessed June. 16, 2022.

United States Census Bureau (US Census). (2022). *Population Estimates*. https://www.census.gov/quickfacts/orangecountyflorida. Accessed July. 28, 2022.

Ravani, B., Darter, M., Hiremagalur, J., Lasky, T. A., & Tabib, S. (2009). Inventory and Assessing Conditions of Roadside Features Statewide Final Report. *Academy of Management Review*, 127. file:///var/zpanel/hostdata/zadmin/public_html/arsip_repositoryartikel_net/mendeley/Academy of Management Review/2004/Biggart - University of California at Davis.pdf

Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263-7271).

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

Ren, S., He, K., Girshick, R., & Sun, J. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(6), 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Sainju, A. M., & Jiang, Z. (2020). Mapping Road Safety Features from Streetview Imagery. *ACM/IMS Transactions on Data Science*, *1*(3), 1–20. https://doi.org/10.1145/3362069

Shamayleh, H., & Khattak, A. (2003, August). Utilization of LiDAR technology for highway inventory. In *Proceedings of the 2003 Mid-Continent Transportation Research Symposium, Ames, Iowa*.

Slattery, D. K., & Slattery, K. T. (2010). *Evaluation of 3-D laser scanning for construction application*. Rantoul, IL: Illinois Center for Transportation.

Spear, B., Vandervalk, A., Snyder, D., & Systematics, C. (2010). *Roadway geometry and inventory trade study for Intellidrive applications* (No. FHWA-HRT-10-073). United States. Federal Highway Administration.

Sun, K., Zhang, J., & Zhang, Y. (2019). Roads and intersections extraction from high-resolution remote sensing imagery based on tensor voting under big data environment. *Wireless Communications and Mobile Computing*, *2019*.

Tang, J., & Zakhor, A. (2011). 3D object detection from roadside data using laser scanners. In *Three-Dimensional Imaging, Interaction, and Measurement* (Vol. 7864, pp. 300-317). SPIE.

Tang, T., Zhou, S., Deng, Z., Zou, H., & Lei, L. (2017). Vehicle detection in aerial images based on region convolutional neural networks and hard negative example mining. *Sensors (Switzerland)*, *17*(2). https://doi.org/10.3390/s17020336

Teo, T. A. (2018, June). The extraction of urban road inventory from mobile lidar system. In *IOP Conference Series: Earth and Environmental Science* (Vol. 169, No. 1, p. 012022). IOP Publishing.

Tsai, Y., Kim, P., & Wang, Z. (2009). Generalized traffic sign detection model for developing a sign inventory. *Journal of Computing in Civil Engineering*, *23*(5), 266-276.

Tsang, S. H. (2019). Review: Review: Yolov3—You Only Look Once (Object Detection). *medium. com. https://towardsdatascience. com review-yolov3-you-only-look-once-object-detection-eab75d7a1ba6 (accessed June 25, 2022).*

Turner, B., & Comport, L. (2007). Automatic collection of safety related road and roadside data. In *Proceedings of the Australasian road safety research, policing and education conference* (Vol. 11). Monash University.

Uçar, A., Demir, Y., & Güzeliş, C. (2017). Object recognition and detection with deep learning for autonomous driving applications. *Simulation*, *93*(9), 759–769. https://doi.org/10.1177/0037549717709932

Uslu, B., Golparvar-Fard, M., & de la Garza, J. M. (2011). Image-based 3D reconstruction and recognition for enhanced highway condition assessment. In *Computing in Civil Engineering (2011)* (pp. 67-76).

Vattapparamban, E., Güvenç, I., Yurekli, A. I., Akkaya, K., & Uluağaç, S. (2016). Drones for smart cities: Issues in cybersecurity, privacy, and public safety. *2016 International Wireless Communications and Mobile Computing Conference, IWCMC 2016*, 216–221. https://doi.org/10.1109/IWCMC.2016.7577060

Veneziano, D. (2001). Roadway intersection inventory and remote sensing. *ISU Center for Transportation Research and Education, ed*.

Vincent, R. A., & Ecker, M. (2010). *Light detection and ranging (LiDAR) technology evaluation* (No. OR11-007). Missouri. Dept. of Transportation.

Viola, P., & Jones, M. (2005). *Rapid object detection using a boosted cascade of simple features*. *July 2014*, I-511-I–518. https://doi.org/10.1109/cvpr.2001.990517

Wang, K. C., Hou, Z., & Gong, W. (2010). Automated road sign inventory system based on stereo vision and tracking. *Computer-Aided Civil and Infrastructure Engineering*, *25*(6), 468-477.

Wiedemann, C., and H. Ebner. *International Archives of Photogrammetry and Remote Sensing*. 2000.

Wiedemann, C., C. Heipke, H. Mayer, and O. Jamet. *EMPIRICAL EVALUATION OF AUTOMATICALLY EXTRACTED ROAD AXES*. 1998

Xie, X., Yang, W., Cao, G., Yang, J., Zhao, Z., Chen, S., Liao, Q., & Shi, G. (2018). Real-Time Vehicle Detection from UAV Imagery. *2018 IEEE 4th International Conference on Multimedia Big Data, BigMM 2018*, 1–5. https://doi.org/10.1109/BigMM.2018.8499466

Xu, Y., Yu, G., Wang, Y., Wu, X., & Ma, Y. (2016). A hybrid vehicle detection method based on viola-jones and HOG + SVM from UAV images. *Sensors (Switzerland)*, *16*(8). https://doi.org/10.3390/s16081325

Xu, Y., Yu, G., Wang, Y., Wu, X., & Ma, Y. (2017). Car detection from low-altitude UAV imagery with the faster R-CNN. *Journal of Advanced Transportation*, *2017*. https://doi.org/10.1155/2017/2823617

Yan, L., Liu, H., Tan, J., Li, Z., Xie, H., & Chen, C. (2016). Scan line based road marking extraction from mobile LiDAR point clouds. *Sensors*, *16*(6), 903.

Zhou, H., Jalayer, M., Gong, J., Hu, S., & Grinter, M. (2013). Investigation of methods and approaches for collecting and recording highway inventory data. *FHWA-ICT-13-022*.