Article

Not peer-reviewed version

Turning Features Detection from Aerial Images: Model Development and Application on Florida’s Public Roadways

Richard Boadu Antwi * , Michael Kimollo , Eren Erman Ozguven , Thobias Sando , Ren Moses , Maxim A. Dulebenets

Posted Date: 1 May 2024

doi: 10.20944/preprints202405.0021.v1

Keywords: Turning lanes; Deep Learning; Roadway Characteristic Index (RCI); Pavement Markings; Machine Learning (ML); Roadway Geometry Features

Article

Turning Features Detection from Aerial Images: Model Development and Application on Florida's Public Roadways

Richard Boadu Antwi [1,*], Michael Kimollo [2], Eren Erman Ozguven [1], Thobias Sando [2], Ren Moses [1] and Maxim A. Dulebenets [1]

[1] Department of Civil and Environmental Engineering, Florida A&M University–Florida State University College of Engineering, Florida State University, 2525 Pottsdamer Street, Tallahassee, FL 32310, USA
[2] School of Engineering, University of North Florida, Jacksonville, FL 32224
* Correspondence: rantwi@fsu.edu

ABSTRACT: Advancements in computer vision are rapidly revolutionizing the way traffic agencies gather roadway geometry data, leading to significant savings in both time and money. Utilizing aerial and satellite imagery for data collection proves to be more cost-effective, more accurate, and safer compared to traditional field observations, considering factors such as equipment cost, crew safety, and data collection efficiency. Consequently, there is a pressing need to develop more efficient methodologies for promptly, safely, and economically acquiring roadway geometry data. While image processing has previously been regarded as a time-consuming and error-prone approach for capturing these data, recent developments in computing power and image recognition techniques have opened up new avenues for accurately detecting and mapping various roadway features from a wide range of imagery data sources. This research introduces a novel approach combining image processing with a YOLO-based methodology to detect turning lane pavement markings from high-resolution aerial images, specifically focusing on Florida's public roadways. Upon comparison with ground truth data from Leon County, Florida, the developed model achieved an average accuracy of 87% at a 25% confidence threshold for detected features. Implementation of the model in Leon County identified approximately 3,026 left turn, 1,210 right turn, and 200 center lane features automatically. This methodology holds paramount significance for transportation agencies in facilitating tasks such as identifying deteriorated markings, comparing turning lane positions with other roadway features like crosswalks, and analyzing intersection-related accidents. The extracted roadway geometry data can also be seamlessly integrated with crash and traffic data, providing crucial insights for policymakers and road users.

Keywords: turning lanes; deep learning; roadway characteristic index (RCI); pavement markings; machine learning (ML); roadway geometry features

Introduction

Advancement of computer vision technology is rapidly empowering traffic agencies to streamline the collection of roadway geometry data, providing savings in both time and resources. Traditionally, image processing has been viewed as a time-consuming and error-prone method for capturing roadway data. However, with the evolution of computational capabilities and image recognition techniques, new possibilities have emerged for accurately detecting and mapping various roadway attributes from diverse imagery sources. A study conducted in 2015 [1] demonstrated that utilizing aerial and satellite imagery for gathering geometry data offered significant advantages over traditional field observations in terms of equipment costs, data accuracy, crew safety, data collection expenses, and time required for collection.

Since the introduction of the Highway Safety Manual (HSM) in 2010, many state departments of transportation (DOTs) have placed a strong emphasis on enhancing safety by aligning with the manual's requirements for geometric data [2]. However, the manual collection of data across extensive roadway networks has presented a considerable challenge for numerous state and local transportation agencies. These agencies have a critical role in maintaining up-to-date roadway data, which is essential for effective planning, maintenance, design, and rehabilitation efforts [3]. Roadway Characteristics Index (RCI) data encompasses a comprehensive inventory of all elements that constitute a roadway, including Highway Performance Monitoring System (HPMS) data, roadway geometry, traffic signals, lane counts, traffic monitoring locations, turning restrictions, intersections, interchanges, rest areas with or without amenities, High Occupancy Vehicle (HOV) lanes, pavement markings, signage, pavement condition, driveways, and bridges. DOTs utilize various methodologies for RCI data collection, ranging from field inventory and satellite imagery to 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 [2].

These methods fall into two main categories: ground observation/field surveying and aerial imagery/photogrammetry. The process of gathering RCI data involves considerable expenses, with each approach requiring specific equipment and time allocation for data collection, thereby impacting overall costs. Ground surveying entails conducting measurements along the roadside using total stations, while field inventory involves traversing the roadway to document current conditions and inputting this information into the inventory database. On the other hand, aerial imagery or photogrammetry entails extracting RCI data from georeferenced images captured by airborne systems such as satellites, aircraft, or drones. Each approach has its own advantages and disadvantages in terms of data acquisition costs, accuracy, quality, constraints on collection, storage requirements, labor intensity, acquisition and processing time, and crew safety. Additionally, pavement markings typically have a lifespan of 0.5 to 3 years [4]. Because of their short lifespan, frequent inspection and maintenance are necessary. However, road inspectors generally adopt a periodic manual inspection approach to assess pavement marking conditions, which is both time-consuming and risky.

Despite the prevalent use of direct field observations by highway agencies and DOTs for gathering roadway data [3], these methods pose numerous challenges, including difficulty, time consumption, risk, and limitations, especially in adverse weather conditions. Therefore, there is a critical need to explore alternative and more efficient approaches for collecting roadway inventory data. Researchers have explored novel and emerging technologies, such as image processing and computer vision methods [2], for this purpose. Over the past decade, satellite and aerial imagery have been extensively utilized for acquiring earth-related information [2]. High-resolution images obtained from aircraft or satellites can quickly provide RCI data upon processing [5–7]. In a previous study, artificial intelligence (AI) was employed to extract school zones from high-resolution aerial images [7]. The increasing availability of such images, coupled with advancements in data extraction techniques, highlights the importance of optimizing their use for efficiently extracting roadway inventory data. While there may be challenges in extracting obscure or small objects, recent advancements in machine learning research have lessened this limitation. Well-trained models can significantly enhance output accuracy. Furthermore, this method entails minimal to no field costs, making it economically advantageous.

RCI data encompass critical features like center, left, or right turning lanes, which are significant to be known to both State DOTs and local transportation agencies in the context of traffic operations and safety. These pavement markings, primarily situated near intersections, indicate permissible turns from each lane, effectively guiding and managing traffic flow. They offer continuous assistance in vehicle positioning, lane navigation, and maintaining roadway alignment, as highlighted in a USDOT study [8]. Many DOTs have expressed keen interest in automated methods for detecting and evaluating these markings [9]. Although turning lane information plays a pivotal role in optimizing roadway efficiency and minimizing accidents, particularly at intersections, a comprehensive geospatial inventory of turning lanes is currently lacking, spanning both local- and state-operated roadways in Florida. Therefore, it is crucial to develop innovative, efficient, and rapid methods for gathering such data. This is of paramount importance for Florida DOT, serving various purposes such as identifying outdated or obscured markings, juxtaposing turning lane positions with other geometric features like crosswalks and school zones, and analyzing accidents occurring in these zones and at intersections.

To the best of the authors' knowledge, there has been no prior research exploring the utilization of computer vision techniques to extract and compile an inventory of turning lane markings using high-resolution aerial images. As such, this study seeks to address this gap by developing automated tools for detecting these crucial roadway features through deep learning-based object detection models. Specifically, the focus will be on creating a You Only Look Once (YOLO)-based artificial intelligence model and framework to identify and extract turning lane markings, including left, right, and center lanes, from high-resolution aerial images. The primary objective is to devise an image processing and object detection methodology tailored to identify and extract these lane features across the State of Florida using high-resolution aerial imagery. This initiative holds significant importance for FDOT and other transportation agencies for various reasons, including infrastructure management, such as identifying aging or obscured markings, comparing turning lane locations with other geometric features like crosswalks and school zones, and analyzing accidents occurring within these zones and at intersections. Additionally, the automated extraction of road geometry data can be seamlessly integrated with crash and traffic data, offering valuable insights to policymakers and roadway users.

Literature Review

Research on using artificial intelligence (AI) techniques to extract Roadway Characteristics Inventory (RCI) data—particularly that pertaining to pavement markings—has significantly increased in the last few years. By using detection models, AI approaches are essential in obtaining information about roadways from such data. For example, a study [7] used high-resolution aerial pictures and YOLO to locate school zones. Although a lot of research has focused on employing LiDAR technology to capture RCI data, this method has drawbacks including expensive equipment, complicated data processing, and long processing periods. For instance, a study [3] used precision navigation and powerful computers to use LiDAR to compile inventories of various roadway elements. Furthermore, in other studies, Mobile Terrestrial Laser Scanning (MTLS) has been utilized to gather data from highway inventories [5]. Furthermore, computer vision, sensor technology, and artificial intelligence (AI) techniques have been combined in recent autonomous driving research to recognize pavement markings for vehicle navigation [10–14].

The majority of previous research has been on conventional approaches, which are linked to longer data gathering times, traffic flow disruptions, and possible dangers to crew members. Conversely, there hasn't been much focus on developing technologies like computer vision, object detection, and deep learning. With the help of these cutting-edge technologies, roadway features may be extracted from high-resolution aerial photos, doing away with the necessity for labor-intensive fieldwork and drastically cutting down on data gathering time. In addition, the data collected using these cutting-edge techniques is readily available, protects crew safety by averting traffic-related risks, and quickly captures roadway characteristics over wide areas. Standardized image collection is not the only criteria to consider when choosing the best object detection model [15]. While the fundamental ideas of many popular models, such as those offered by TensorFlow, are similar, their accuracy, speed, and memory usage vary according to the demands of the particular project.

Utilizing Computer Vision and Deep Learning for Roadway Geometry Feature Extraction

Experts in transportation are beginning to recognize convolutional neural networks (CNN) and recurrent neural networks (RNN) as two new instances of computer vision and deep learning methods [16]. Notable progress has been made in the extraction of roadway geometry data by researchers using CNN and R-CNN. These methods have proven to be effective in quickly identifying, detecting, and mapping roadway geometry features over large regions with no need for human interaction. For example, in [4], photogrammetric data from Google Maps was utilized to automatically identify pavement marking issues on roadways using R-CNN to evaluate pavement conditions. With an average accuracy of 30%, the developed model in this study was trained to detect flaws in the pavement markings, such as incorrect alignment, ghost marking, missing, or fading edges, corners, segments, and cracks.

In one study, Google Street View (GSV) photos taken along interstate segments were used to recognize and categorize traffic signs using computer vision algorithms [17].In order to classify road signs according to their pattern and color information, the researchers used a machine learning technique that paired Histograms of Oriented Gradients (HOG) data with a linear Support Vector Machine (SVM) classifier. 94.63% classification accuracy was attained by them. However, neither local nor non-interstate roads were used to test the methodology. The research by [18] used ground-level photos and a technique known as ROI-wise reverse reweighting network to recognize road markers. Using ROI-wise reverse reweighting and multi-layer pooling operation based on Faster RCNN, the study achieved a 69.95% detection accuracy for road markers in images by highlighting multi-layer features.

In [12], inverse perspective mapped photos were used to estimate roadway geometry and identify lane markings on roadways using CNN. Researchers used CNN in [19] to detect, measure, and extract the geometric properties of hidden cracks in roads using ground-penetrating radar pictures. CNN was used by the authors in [20] to extract roadway features from remotely sensed photos. These methods have been used to detect images for a variety of purposes. For example, driverless cars use them to instantly identify objects and road infrastructure. Using the Caltech-101 and Caltech pedestrian datasets, a novel hybrid local multiple system (LM-CNN-SVM) incorporating convolutional neural network (CNN) and SVMs was tested for item and pedestrian detection [21]. Several CNNs were tasked with learning the properties of local areas and objects after the full image was divided into local regions. Significant characteristics were chosen using principal component analysis, and in order to improve the classifier system's ability to generalize, these features were then fed into several SVMs using empirical and structural risk reduction strategies rather than only using CNN. Moreover, an original CNN architecture and a pre-trained AlexNet were both used, and the SVM output was mixed. Between 89.80% and 92.80% was the accuracy range that the approach showed.

Convolutional neural networks (CNNs) based on deep learning have demonstrated impressive capabilities in object detection recently [22–25]. These algorithms have been greatly enhanced by their progression from region-based CNN (R-CNN) to fast R-CNN [26], and ultimately faster R-CNN [27]. The CNN method basically involves computing CNN features after extracting region proposals as possible object placements. By adding another Region Proposal Network (RPN) to the object detection network, Faster R-CNN improves performance even further. In a study [28], the vehicle detection and tracking skills of machine learning algorithms and deep learning techniques—faster R-CNN and ACF—were thoroughly compared. The results show that quicker R-CNN performs better in this area than ACF. Therefore, choosing the algorithm to perform the analysis requires significant thought. The output of vehicle identification and tracking algorithms may include parameters like speed, volume, or vehicle trajectories, depending on the analysis's goals. Vehicle detection algorithms differ from point tracking techniques in that they can also classify automobiles.

Obtaining Roadway Geometry Data through LiDAR and Aerial Imagery Techniques

Road markings can be extracted from photos by taking advantage of their high reflectivity, which is a characteristic that they usually display. In [29] researchers developed an algorithm that uses pixel extraction to recognize and locate road markers in pictures taken by cameras installed on moving cars. Initially, a median local threshold (MLT) image filtering technique was used by the system to extract marking pixels. After then, a recognition algorithm was used to identify markings according to their dimensions and forms. For the identified markers, the average true positive rate was 84%.Researchers in [30] extracted highway markers from aerial photographs using a CNN-based semantic segmentation technique. The paper presented a novel method for processing high-resolution images for lane marking extraction that makes use of a discrete wavelet transform (DWT) and a fully convolutional neural network (FCNN). In contrast to earlier techniques, this strategy—

which used a symmetric FCNN—focused on small-sized lane markers and investigated the best places for DWT insertion sites inside the architecture. The accuracy of lane detection could be enhanced by the model's ability to collect high-frequency information with the integration of DWT into the FCNN architecture. Aerial LaneNet architecture consisted of a sequence of pooling, wavelet transform, and convolutional layers. The model's effectiveness was assessed using an aerial lane detection dataset that was made available to the general audience. The study found that washed-out lane markers and shadowed areas presented constraints.

Researchers used the scan line method in [31], which allows them to recover road markings from LiDAR point clouds. The team efficiently structured the data by timestamp-sequentially ordering the LiDAR point clouds and grouping them into scan lines based on the scanner angle. They then isolated seed roadway points, which formed the foundation for determining complete roadway points, by taking advantage of the height difference between trajectory data and the road surface. Using these points, a line was drawn that encircled the scan line's seed points as well as every other point. After that, points that fell within a given range of this line were kept and classified as asphalt points or road markings according to how intense they were. In order to smooth out intensity values and improve data quality by lowering noise, a dynamic window median filter was used. Finally, the research team identified and extracted roadway markers using edge detection algorithms and limitations. Roadway data was gathered by researchers in [3] using two-meter resolution aerial images and six-inch accuracy LiDAR data saved in ASCII comma-delimited text files. Using ArcView, they added height information to LiDAR point shapefiles and then converted the data points into Triangular Irregular Networks (TIN). The analytical process was started by combining the aerial photos and LiDAR boundaries. To find obstructions in the observer's line of sight, ArcView's object obstruction tool was utilized. Tracing the line of sight—visible terrain represented by green line segments, obscured terrain by red line segments—was used to calculate the stopping sight distance. With ArcView's identification and contour capabilities, side slope and contours were determined. The grade was also determined by the researchers by calculating the elevation difference between a segment's two ends. Conversely [32], built a deep learning model for the automated recognition and extraction of road markings using a 3D profile of pavement data obtained by laser scanning. They were able to detect with 90.8% accuracy. As part of the project, a step-shaped operator was designed to identify possible locations for pavement marking edges. The road markings were subsequently extracted using CNN by combining the geometric properties and convolution information of these locations.

In [33] pavement markings were extracted from camera images and pavement conditions were assessed using computer vision techniques. Preprocessing was applied to videos that were taken of cars from the frontal perspective for the research. The preprocessing involved the extraction of image sequences, the Gaussian blur median filter for filtering and smoothing the images, and the inverse perspective transform for correcting the photos. A hybrid detector that relied on color and gradient features was used to find pavement markings. After that, regions with marks were identified using image segmentation and then categorized into edge, dividing, barrier, and continuity lines according to their characteristics. Building on a prior study [34] that used Yolov3 to extract turning lane features from aerial images, this research will use additional image processing techniques along with a more advanced version of yolo to extract turning lane features, including left, right, and center, from extremely high resolution aerial images in Leon County, Florida.

To the authors' knowledge, there has not been any research done on how to make an inventory of turning lane markings using image processing and computer vision techniques. Thus, by creating automated methods for identifying these turning lane markings (e.g., left, right, and center), our research attempts to close this gap. To achieve this, the study will specifically concentrate on developing an artificial intelligence model based on the You Only Look Once (YOLO) methodology.

Study Area

The case study region for turning lanes and configurations detection has been chosen to be Leon County (Figure 1). There are 292,198 people residing in Leon County, which is home to the City of Tallahassee, the capital city of Florida, and it is 668 square miles in total size [35]. It shares boundaries with Georgia's Grady and Thomas counties as well as other Florida counties like Gadsden, Wakulla, Liberty, and Jefferson. The county was taken into consideration because of the variety of its roadway infrastructure development. The models' performance was verified using ground truth data from Leon County, Florida.

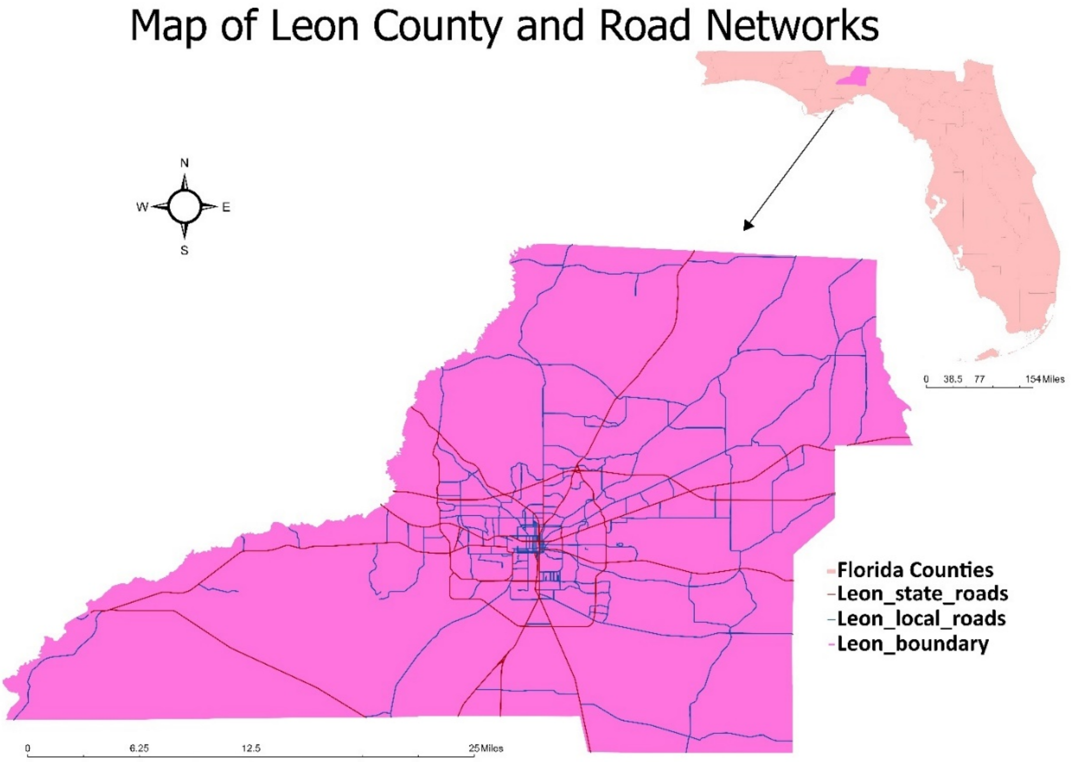


Figure 1. Map of Leon County, Florida with the roadway network.

Materials and Methodology

The selection of the methodology for gathering roadway inventory data is contingent upon various factors including data collection time (e.g., data collection, reduction, and processing), cost (e.g., data collection, and reduction), as well as accuracy, safety, and data storage demands. In this study, our objective was to develop a deep learning object detection model tailored to identify turning lane markings from high-resolution aerial images within Leon County, Florida.

Data Description

The proposed algorithm for detecting turning lanes relies on high-resolution aerial images and recent advancements in computer vision and object detection techniques. The goal is to automate the process of identifying pavement markings for turning lanes, offering a novel solution for transportation agencies. The aerial images used in this study were sourced from the Florida Aerial Photo Look-Up System (APLUS), an aerial imagery database managed by the Surveying and Mapping Office of the Florida Department of Transportation (FDOT). Specifically, images from Leon (2018) and Miami-Dade (2017) Counties, Florida, were utilized. These images have a resolution ranging from 1.5 ft down to 0.25 ft, ensuring sufficient detail for accurate detection of turning lanes. The model's resolution requirement allows for the utilization of any imagery falling within or exceeding this resolution threshold. The majority of the images were in 0.5 ft/pixel resolution, with a size of 5,000 × 5,000 and a 3-band (RGB) image format, although precise resolutions varied by county. Additionally, state, and local roadway shapefiles were obtained from FDOT’s GIS database, providing further data for the detection algorithm. The images are given in MrSID format, which enables GIS projection onto maps, enhancing spatial analysis capabilities. Also, state, and local roadway shapefiles were acquired from FDOT's GIS database, complementing the image data, and enriching the dataset for further analysis.

This study focuses on identifying turning lanes present on roadways managed by counties or cities, as well as those situated on state highway system routes, excluding interstate highways. State highway system routes are termed ON System Roadways or state roadways, while county or city-managed roads are referred to as OFF System Roadways or local roadways, according to FDOT classification. Our proposed approach involves merging all centerlines from the shapefile of county and city-managed roadways. While FDOT's GIS data can provide various geometric data points crucial for mobility and safety assessments, it lacks information on the locations of turning configurations on the local roadways. Therefore, the main objective of this project is to utilize an advanced object identification model to compile an inventory of turning lane markers, specifically left-only, right-only, and center lanes, on both state and local roadways in Leon County, Florida.

Pre-Processing

Due to the volume of data and the complexity of object recognition, preprocessing is essential. This method involves selecting and discarding images that do not intersect with a roadway centerline and mask out pixels outside a buffer zone. In this approach, the number of photos was decreased, and the image masking model excluded objects that were 100 feet away from state and local roadways (Figure 2). Prior to masking the images, the roadway shapefile is buffered to create overlapping polygons, serving as reference for cropping intersecting regions of aerial images. During masking, pixels outside the reference layer's boundary are removed. The resulting cropped images, with fewer pixels, are then mosaiced into a single raster file, facilitating easier handling and analysis.

First, all aerial images from the selected counties are imported into a mosaic dataset using ArcGIS Pro. Geocoded photos are then managed and visualized using mosaic databases, allowing for the intersection with additional vector data to select specific image tiles based on location. Following this, a subset of photos is extracted, comprising images that intersect with roadway centerlines. An automated image masking tool is subsequently developed using ArcGIS Pro’s ModelBuilder interface. This tool systematically processes the photo folder, applying a mask based on a 100-foot buffer around roadway centerlines, and saves the resulting masked images as JPG files for the object detection process. Exported images are further processed for model training and detection.

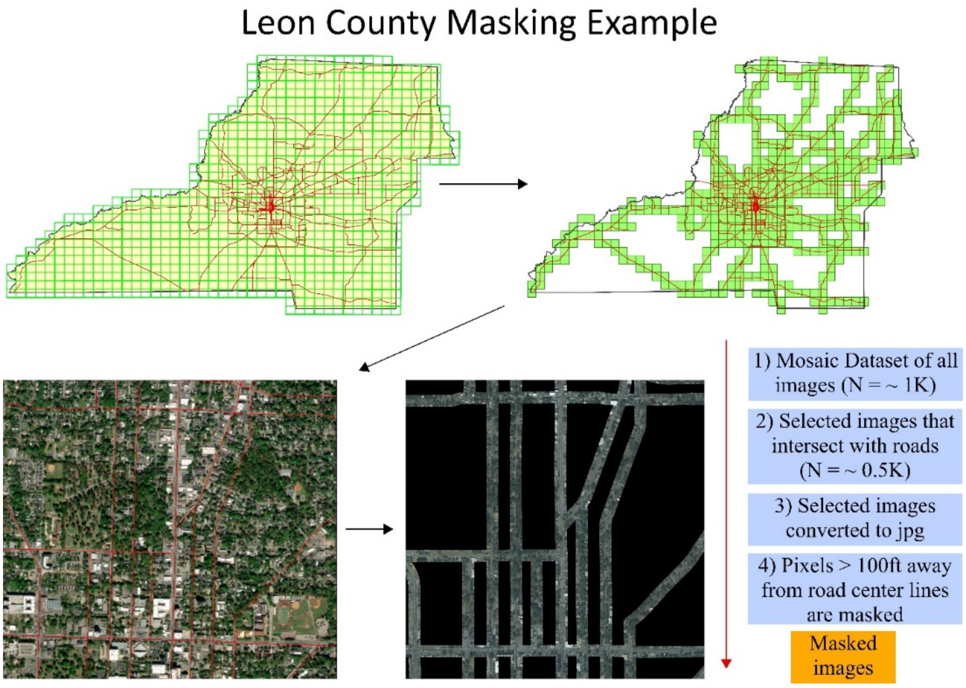

Leon County Masking Example
1) Mosaic Dataset of all images (N = ~ 1K)
2) Selected images that intersect with roads (N = ~ 0.5K)
3) Selected images converted to jpg
4) Pixels > 100ft away from road center lines are masked
Masked images

Figure 2. Preprocessing approach, and automated image masking model.

Data Preparation for Model Training and Evaluation

High-quality image data is crucial for training deep learning models to identify various turning lane features. The process of image preparation, depicted in Figure 3, outlines the steps from turning lane identification to the extraction of training data or images. Significant time and effort were dedicated to generating training data for model creation, as a substantial portion of the model's performance hinges on the quantity, quality, and diversity of the training data used. Similarly, additional time and effort were invested in the model training process, involving testing different parameters, and selecting the optimal ones to train the model. This is depicted by the larger box sizes for "Manually label turning lane features" and "training data" in Figure 3.

For the initial investigation, a 4-class object detection model was developed to identify the turning configurations. The training data with varying classes was prepared for the model. The training data prepared for the turning lane model comprised 4 classes: "left_only", "right_only", "center", and "none" denoted as classes 1, 2, 3, and 0, respectively. These labels were represented by rectangular bounding boxes encompassing the turning lane markings (Table 1). The training data initially prepared for the turning lane model was manually labelled 4,687 features on the aerial images of Miami-Dade county using the Deep-Learning Toolbox in ArcGIS Pro. The model had 1,418 features (30.3%) for the "left_only" class, 1,328 features (28.3%) for the "right_only" class, 662 features (14.1%) for the "center" class, and 1,279 features (27.3%) for the "none" class. The "none" class labels for the model were made up of all other visible markings or features except "left_only", "right_only", and "center". Some features which were very common for instance, left only and right only were more prevalent in the training data for the turning lane model while less common features like "center" had lower representation. To address this imbalance and enhance dataset diversity, some of these less common features were duplicated. However, to prevent potential issues like model overfitting and bias resulting from duplication, data augmentation techniques such as rotation were employed. Rotation involved randomly rotating the training data features at various angles to generate additional training data, thereby increasing dataset diversity. This approach enabled the model to recognize objects in different orientations and positions, which was particularly advantageous in this study, given that features like left-only and right-only lanes can appear in multiple travel directions at intersections. A 90 degrees rotation was applied to the training data. Finally, 11,036 exported image chips containing 16,564 features were used to train the model. It is important to mention that a single image chip may contain multiple features.

Each class within the training data was distinctively labeled to ensure clarity. These classes distinctly categorized the turning lane features and non-features identified in the input image by the detector, thereby enhancing the model's detection accuracy. The output data was then sorted into categories such as left only, right only, center, and none, utilizing a class value field. The metadata generated for these labels such as the image chip size, object class, and bounding box dimensions, were stored in an xml file. A function was then created to further convert the metadata into yolov5 format to obtain the transformed bounding box center coordinates, the box width, and height, using the following equations:

$$cx = 0.5 * \frac{i[xmin + xmax]}{w} \tag{1}$$

$$cy = 0.5 * \frac{i[ymin+ymax]}{h} \tag{2}$$

$$bw = \frac{i[xmax - xmin]}{w} \tag{3}$$

$$bh = \frac{i[ymax - ymin]}{h} \tag{4}$$

where i is the bounding box, cx is the x coordinate of the bounding box center, cy is the y coordinate of the bounding box center, bw is width the bounding box, bh is height the bounding box, w is width the image chip, h is height the image chip, $xmax$ is maximum x value of the bounding box, $xmin$ is minimum x value of the bounding box, $ymax$ is maximum y value of the bounding box, and $ymin$ is minimum y value of the bounding box. The normalization of coordinates was performed using the image dimensions to maintain the bounding box positions relative to the input image.

It is worth noting that object detection models perform optimally when trained on clear and distinctive features. Given that left-only and right-only turning markings often exhibit a lateral inversion of one another, rendering them less distinguishable, the model's performance in detecting them is expected to be lower. To ensure uniqueness, the labels for training the center lane class featured both left arrows facing each other from different travel directions. This approach was similarly applied to label all other features to maintain uniqueness in the training features. The input mosaic data comprised high-resolution aerial images covering Miami-Dade County, Florida, with a tile size of 5000x5000 square feet.

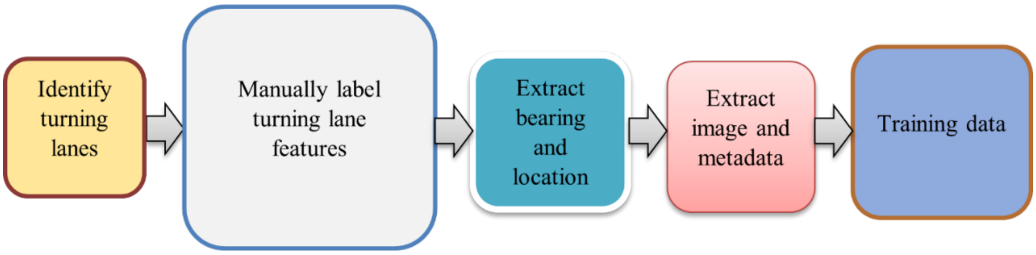
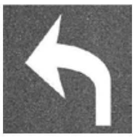
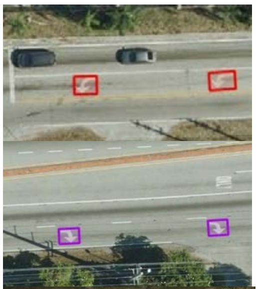
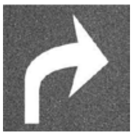
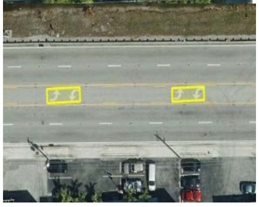

Identify turning lanes
Manually label turning lane features
Extract bearing and location
Extract image and metadata
Training data

Figure 3. Model training data preparation framework.

Table 1. Training data description.

ID	Class name	Description	Example/Picture	Training Example
1	left_only	Left only		
2	right_only	Right only		
3	center	Center lane (left turn possible in both direction or two way)	ONLY CENTER LANE ONLY	
0	none	None: Yield, left through, right through, bicycle, merge, u-turn, Stop, Parking, Speed limits, etc	STOP Yield Ahead P	

YOLOv5 – Turning Lane Detection Model

You Only Look Once (YOLO) is predominantly utilized for real-time object detection. Compared to other models such as R-CNN or Faster R-CNN, YOLO's standout advantage lies in its speed. Specifically, YOLO surpasses R-CNN and Faster R-CNN architectures by a factor of 1000 and 100, respectively [36]. This notable speed advantage stems from the fact that while other models first classify potential regions and then identify items based on the classification probability of those regions, YOLO can predict based on the entire image context and perform the entire image analysis with just one network evaluation. Initially introduced in 2016 [37], YOLO has seen subsequent versions including YOLOv2 [38] and YOLOv3, which introduced improvements in multi-scale predictions. YOLOv4 [39] and YOLOv5 [40] were subsequently released in 2020 and 2022, respectively.

YOLOv5 architecture is made up of the backbone, the neck (PANet), and the head (output). Unlike other detection models, YOLOv5 uses cross-stage partial network (CSP) -Darknet-53 as its backbone, uses PANET as its neck and yolo layer as its head. CSP-Darknet is a better version of Darknet-53 which was the backbone of YOLOv3 with higher accuracy and processing speed [40]. The architecture of the backbone and 6x6 convolution 2d structure improves model's efficiency [41]. Making the feature extractor perform better and faster than other object detection models including the previous versions [42]. Compared to the previous versions, it has a higher speed, higher precision, and a smaller volume. YOLOv5's backbone enhances both the accuracy and speed of the model, performing twice as fast as ResNet152 [36]. Utilizing multiple pooling and convolution, YOLOv5 forms image features at different levels using the CSP and the spatial pyramid pooling (SPP) to extract different-sized features from input images [43]. The architecture has a significant advantage in the area of computational complexity: on the one hand, the BottleneckCSP architecture reduces complexity and improves speed in the inference process; on the other hand, SPP allows to extract features from images in different parts of the input image, creating three-scale feature maps which improve detection quality. Moreover, the neural network neck is a series of layers that are zones between the input and output layers, which is made up of Feature Pyramid Network (FPN) and Path Aggregation Network (PAN) structures, the task of which is to integrate and synthesize image features in various proportions before transferring it to the prediction. FPN enables semantic features of high level to be passed to the lower feature maps and PAN enables localization features of high accuracy to be passed to the higher feature maps, finally moving into the detection phase where targets of different sizes in feature maps are generated in the output head [43]. Figure 4 shows the network architecture of YOLOv5, illustrating the three-scale detection process. This process involves applying a 1x1 detection kernel on feature maps situated at three distinct areas and sizes within the network.

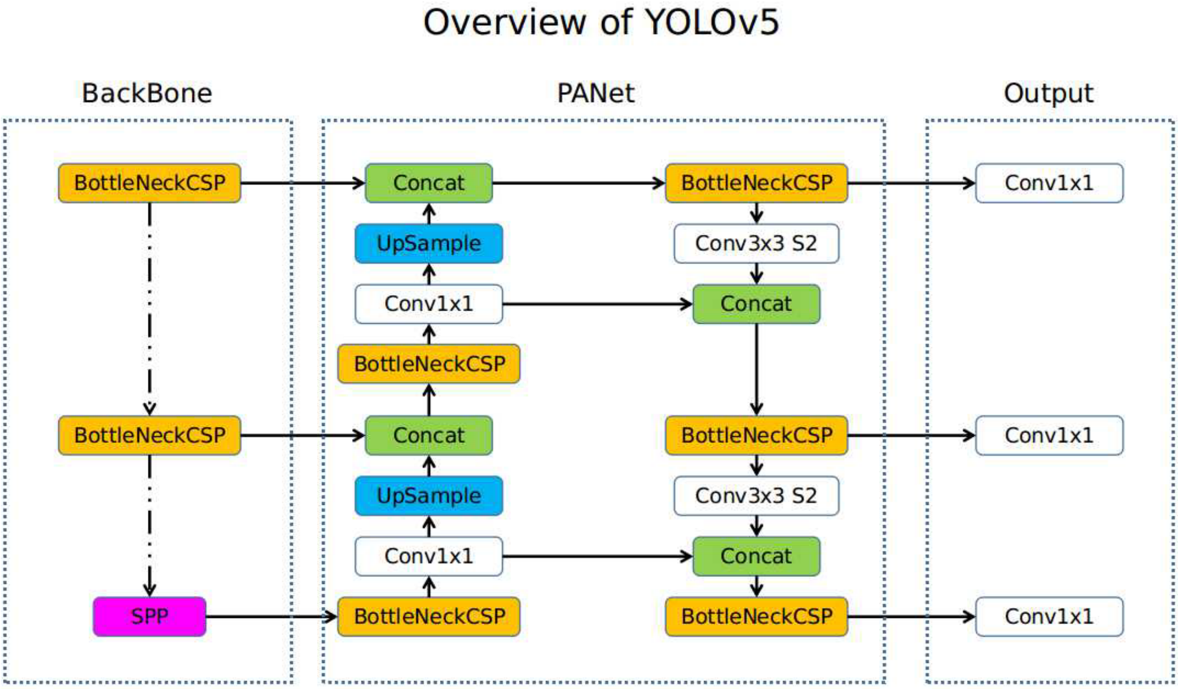


Figure 4. YOLOv5 network architecture, adapted from [40].

Turning Lane Detector

The object detection model's adjustable parameters and hyperparameters encompass various factors such as the learning rate, input image dimensions, number of epochs, batch size, anchor box dimensions and ratios, as well as the percentages of training and testing data. Visualization of the machine learning model's evaluation metrics was presented through graphs depicting precision-recall, validation and training loss, F1-confidence curve, precision confidence, and confusion matrix. Validation loss and mean average precision were computed on the validation dataset, which constituted 20% of the input training dataset. Key parameters influencing object detection performance include the batch size, learning rate, and training epoch. The learning rate dictates the rate at which the model acquires new insights from training data, striking a balance between precision and convergence speed. Optimal learning rates of 3.0e-4 was used to train the model. Batch size refers to the number of training samples processed per iteration, with larger sizes facilitating parallel processing for faster training, albeit requiring more memory. In contrast, smaller batch sizes enhance model performance on new data by increasing randomness during training. Given the multi-class nature and high data complexity of the developed model, a batch size of 64 was adopted to enhance performance. Also, the anchor box represents the size, shape and location of the object being detected. 4 sizes of initial detection anchor boxes were used to train the model. The epoch number denotes the iterations the model undergoes during training, representing the number of times the training dataset is processed through the neural network. The model was trained using 100 epochs. In this study, 80% of the training data was utilized for model training, selected randomly to ensure representativeness. A 20% (10% test-10% validation) split of the training dataset was used for validating and testing the model's performance during training. In other words, 2,202 image chips were used to assess the model's accuracy during the model training process.

The determination of the training and test data split primarily hinged on the size of the training dataset. For datasets exceeding 10,000 samples, a validation size of 20-30% was deemed sufficient to provide randomly sampled data for evaluating the model's performance. A default 50% overlap between the label and detection bounding boxes was considered a valid prediction criterion. Subsequently, recall and precision were computed to assess the true prediction rate among the original labels and all other predictions, respectively. 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 [36]. The calculation methods for the loss function for Yolov5 is the summation of the class loss, object confidence loss, and the location or box loss, which are formulated as follows [40]:

$$Loss = \lambda_1 L_{cls} + \lambda_2 L_{obj} + \lambda_3 L_{loc} \tag{5}$$

$$L_{cls} = \sum_{i=0}^{s^2} \ell_i^{obj} \sum_{j=0}^{B} [(p_i(c) - \widehat{p_i}(c))^2] \tag{6}$$

$$L_{obj} = \sum_{i=0}^{s^2} \sum_{j=0}^{B} \ell_i^{obj} [(c_i - \widehat{c_i})^2] + \lambda_{noobj} \sum_{i=0}^{s^2} \sum_{j=0}^{B} \ell_i^{noobj} [(c_i - \widehat{c_i})^2] \tag{7}$$

$$L_{loc} = 1 - IoU + \frac{\rho^2\,(b, b^{gt})}{c^2} + \alpha v \tag{8}$$

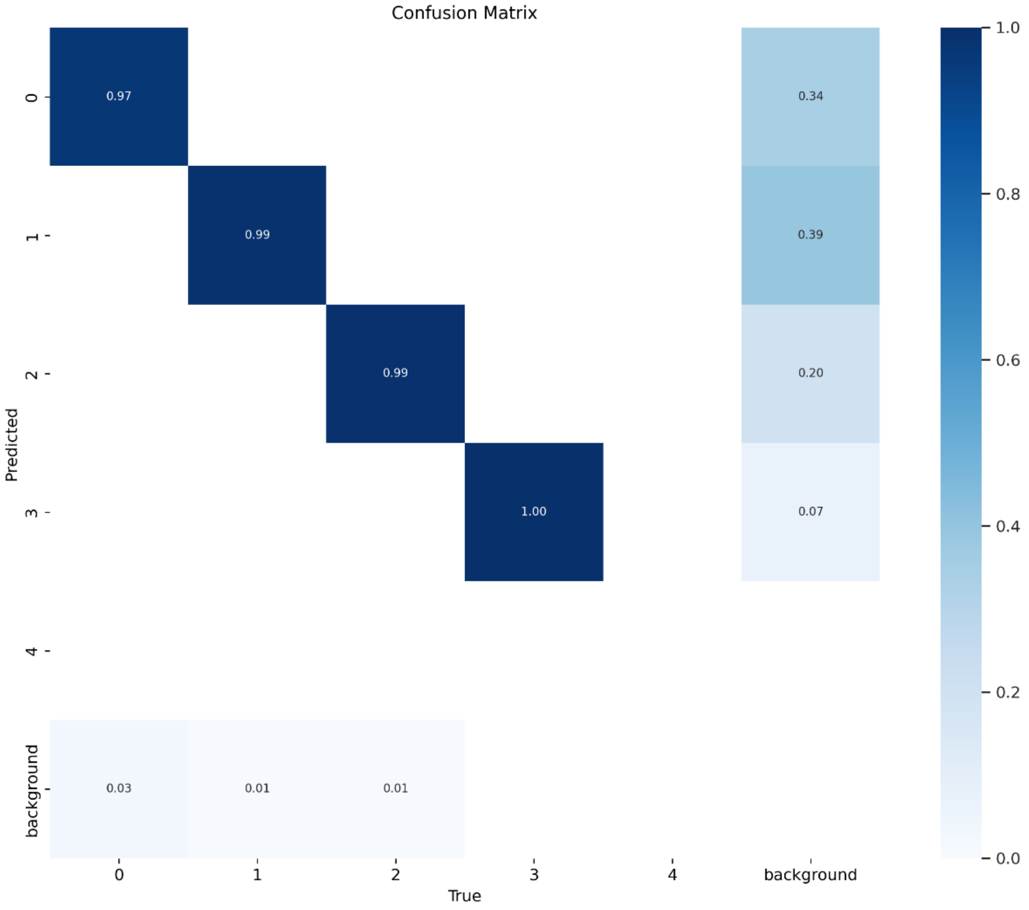
$$IoU = \frac{|b \cap b^{gt}|}{|b \cup b^{gt}|} \tag{9}$$

$$\alpha = \frac{v}{(1 - IoU) + v} \tag{10}$$

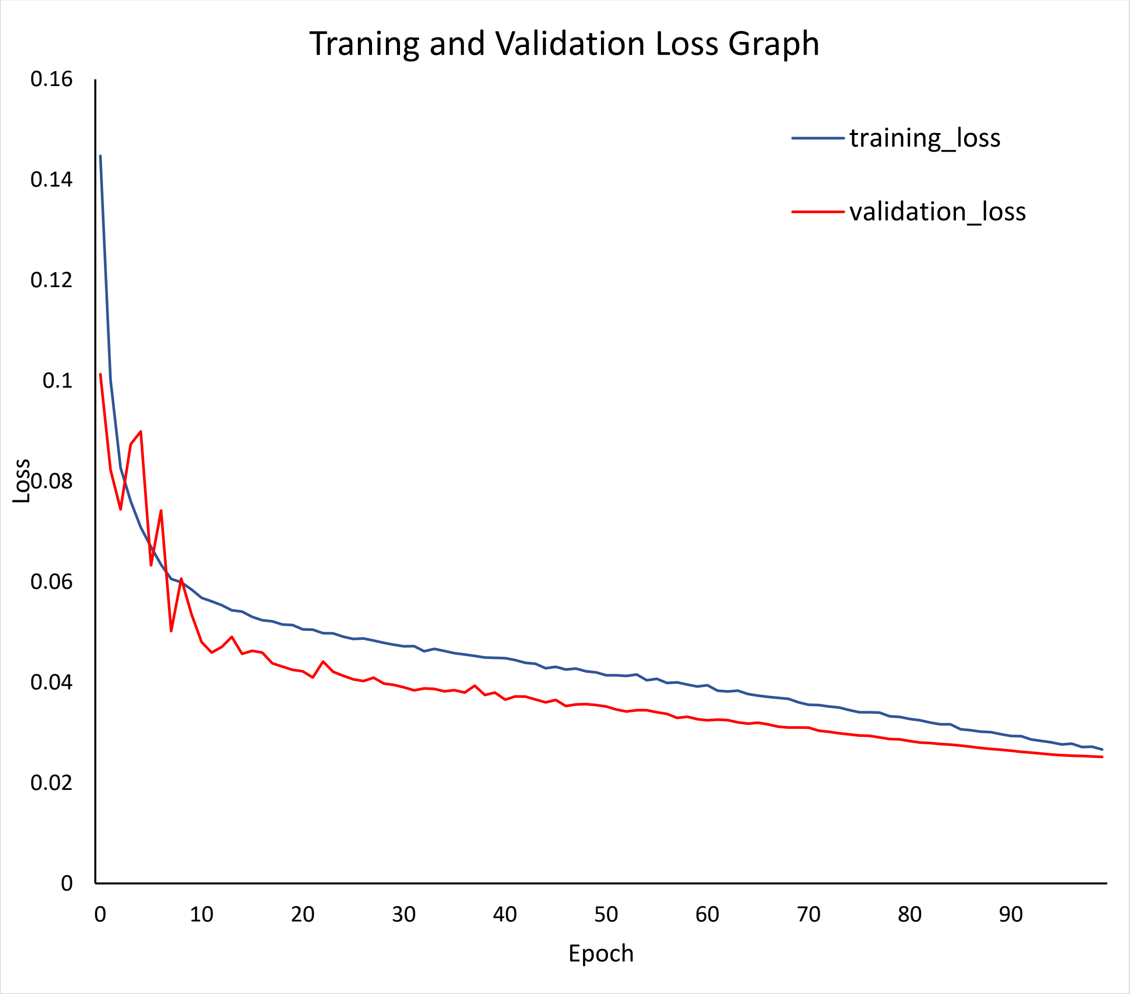
$$v = \frac{4}{\pi^2}\left(arctan\frac{\omega^{gt}}{h^{gt}} - arctan\frac{\omega}{h}\right)^2 \tag{11}$$

where $Loss$ is the overall loss, $L_{cls}$is the classification loss, L_{obj} is the object confidence loss, L_{loc} is the location or box loss, λ_1, λ_2, λ_3 are the weights for the classification loss, object confidence loss, and location or box loss respectively. ℓ_i^{obj} is the probability predictor for the object category, c is the diagonal distance of the smallest closure area that can contain both the predicted box and the real box, c_i is the confidence score, $\widehat{c_i}$ is the intersection of prediction boundary box and basic facts, λ_{noobj} is the weights for the classification error, ℓ_i^{noobj} is the probability predictor of the classification error, IoU is the intersection over union, ρ^2 is the Euclidean distance between the centroids of the prediction box and the real box, b, b^{gt} are the center point of the predicted box and the real box, α is the positive trade-off parameter, v measures the consistency of aspect ratio, and $\frac{\omega^{gt}}{h^{gt}}$ and $\frac{\omega}{h}$ are the aspect ratios of the target and predictions.

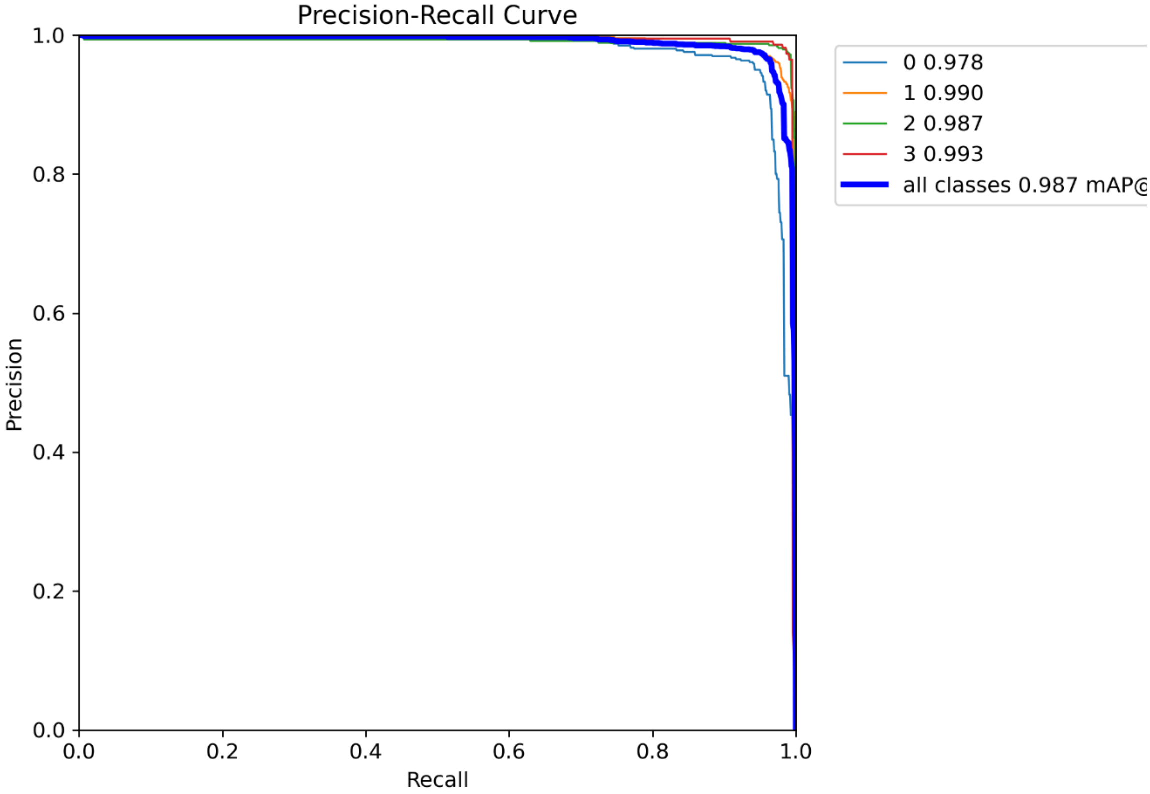
After model training, the performance was evaluated. The confusion matrix describes the true positive rate as well as the misclassified rate. The diagonal values show very high true positive predictions for each class in the confusion matrix (Figure 5a). It can also be observed that the model has extremely low misclassified features with center features showing a 100% correct classification and 0 misclassification. The graph of the train-validation loss for the model is shown in Figure 5b. 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. From the graph, it can be noted that the model had a very low training and validation loss difference with the training loss is slightly higher than validation loss (Figure 5b). This shows that the model is very accurate and performs well on new and existing data however the model performed better on the new dataset. The model has a very good precision-recall curve as shown in Figure 5c. This describes how sensitive the model is in detecting true positives and how well it predicts the positive values. A good classifier has both high precision and high recall across the graph. The developed model has a precision of 0.969, and a recall value of 0.970. The F1 value is a metric calculated from the harmonic mean between the precision and accuracy values. Higher F1 values indicate a better performance. When the F1 value is compared with the confidence thresholds, the optimum threshold can be identified. From the F1-Confidence curve (Figure 5d), the model performs very well between 0.90 and 0.98 F1-score when the confidence threshold is set between 0.05 – 0.8. The optimum threshold of 0.622 returns an F1 value of 0.97 for all classes. Observing the precision-confidence curve, the precision gently increases with higher confidence thresholds. Therefore, the model is very precise even at lower confidence of 0.2 reaching an ideal precision at 0.98 (Figure 5e). The average accuracy of the developed model was 0.987. Therefore, we can conclude that the developed detector performs quite well.


Confusion Matrix
Predicted
True
0
1
2
3
4
background
0.97
0.99
0.99
1.00
0.34
0.39
0.20
0.07
0.03
0.01
0.01
1.0
0.8
0.6
0.4
0.2
0.0

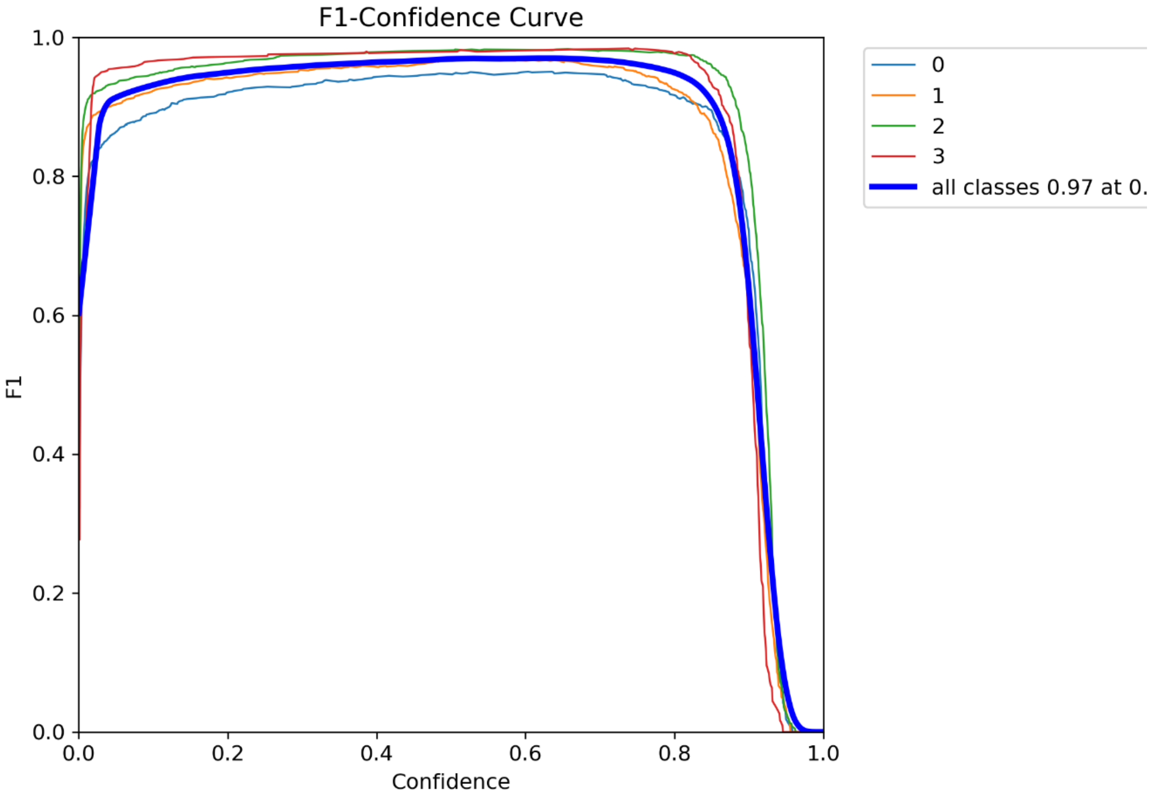
(a)


Traning and Validation Loss Graph
training_loss
validation_loss
Loss
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
10
20
30
40
50
60
70
80
90
Epoch

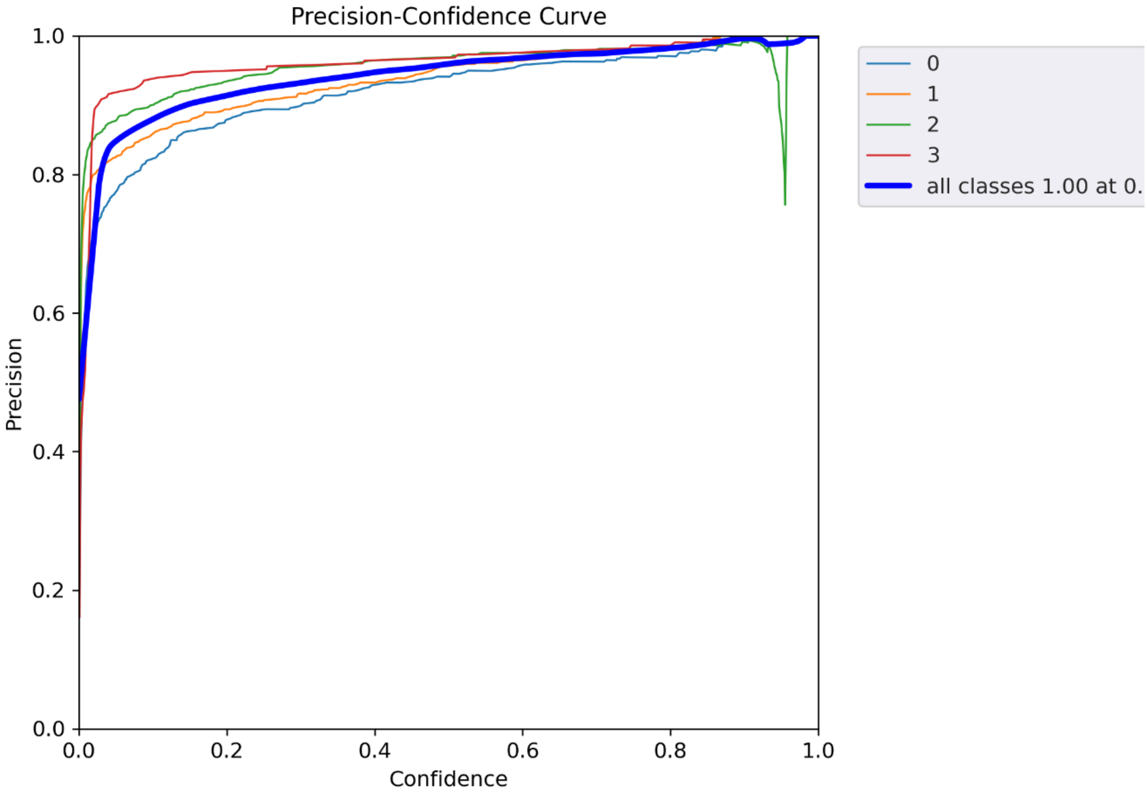
(**b**)


Precision-Recall Curve
0 0.978
1 0.990
2 0.987
3 0.993
all classes 0.987 mAP@
Precision
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Recall

(**c**)



(**d**)



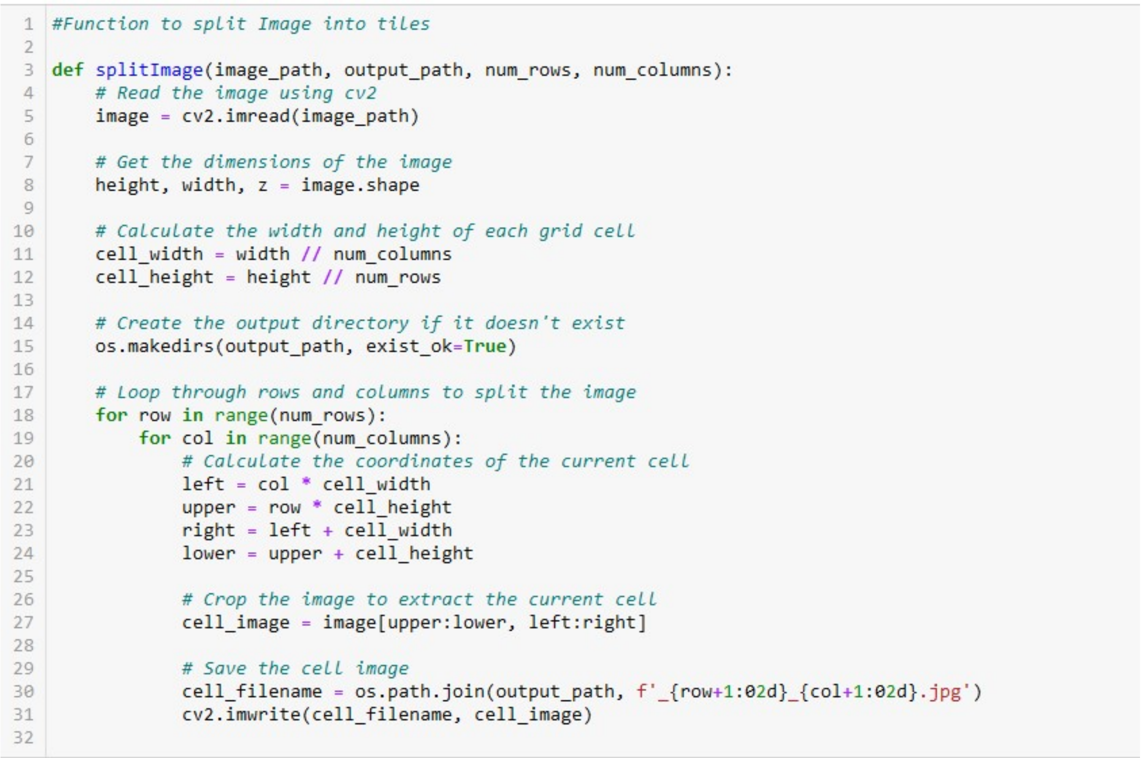
(**e**)

Figure 5. Developed YOLOv3 turning lane model (a) confusion matrix, (b) training and validation loss graph (c) precision-recall curve, (d) F1 confidence curve, and (e) precision-confidence curve.

Mapping Turning Lanes

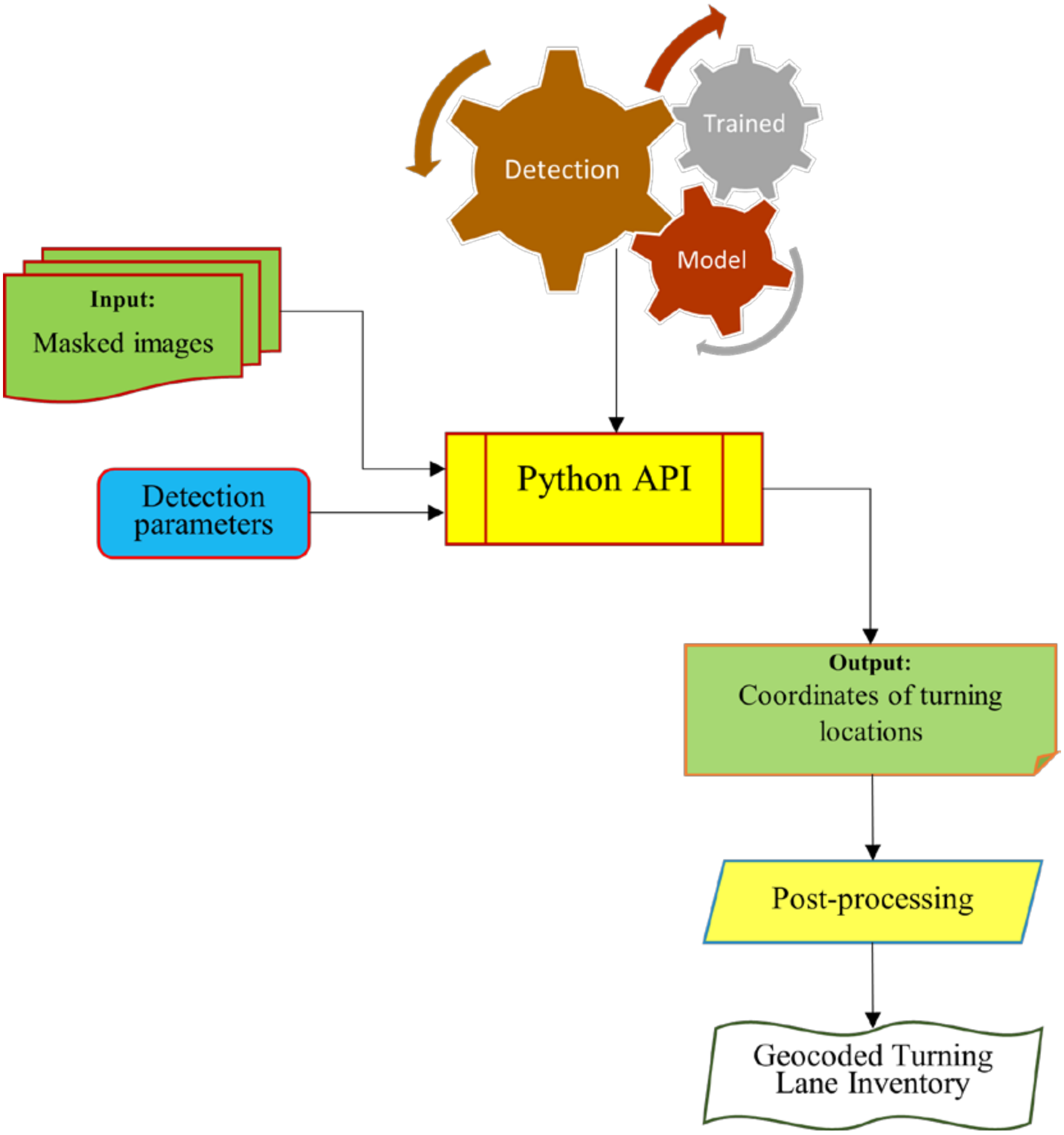
The turning lane detector was initially tested on individual photos. Figure 6c demonstrates how the detector correctly outlines turning lanes with bounding boxes using the detection's confidence score. A threshold of 0.1 was employed to capture all features detected with very low confidence levels. It is important to note that lower detection thresholds generally increase false positives which results in lower precision, increases computational workload as well as detection time, and irrelevant features or noise. On the other hand, lower detection thresholds increase model's sensitivity to detecting 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. More than 10% overlap between two bounding boxes was avoided to minimize duplicate detections. The detector was trained on 512 x 512 sub-images, and a resolution of 0.5 feet per pixel. It should be noted that utilizing huge photos with object detection techniques is impractical since the cost of computing grows rapidly. Therefore, another image processing step was utilized to split detection images into tiles of 512 X 512 or less to carry out tile detections on the input aerial images (Figure 6a). The output detection labels with coordinates were converted into shapefiles and visualized in ArcGIS for further processing and analysis.

The detection and mapping procedure was carried out at the county level since the detector performed very well on single photos. Figure 6b provides a summary of this procedure. The photographs in the folder with a label of "masked images" were first picked out and iterated through the detector. An output file of all the identified turning lanes 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 turning lanes. Note that the model can detect 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 model made some false detections in a few instances. These were outlined and discussed in the results. Figure 6c shows some examples of the detected features.

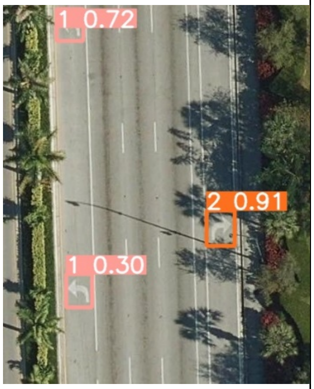
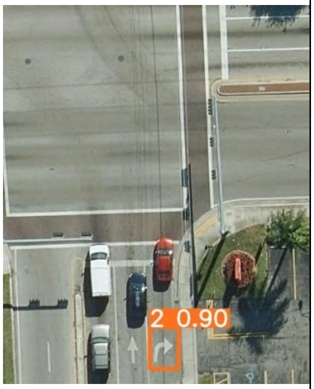
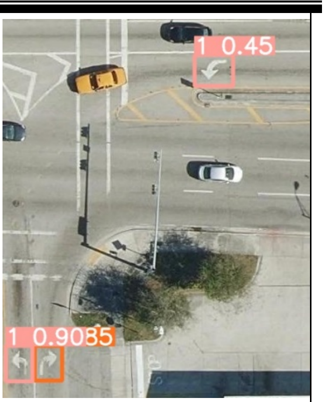
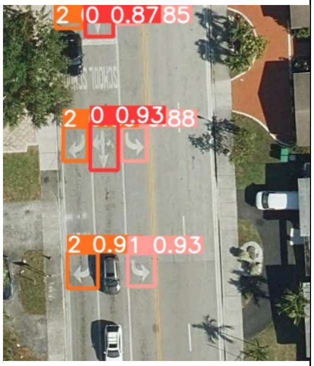
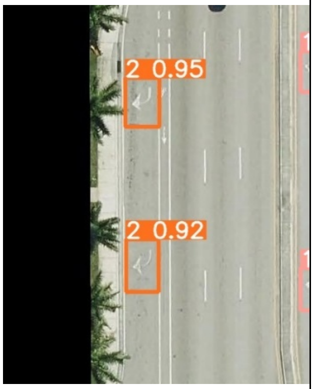
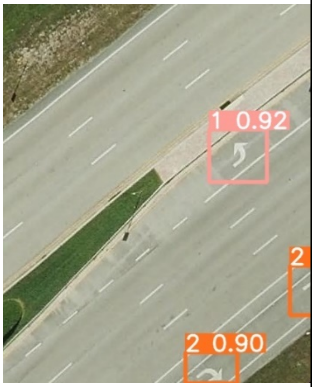
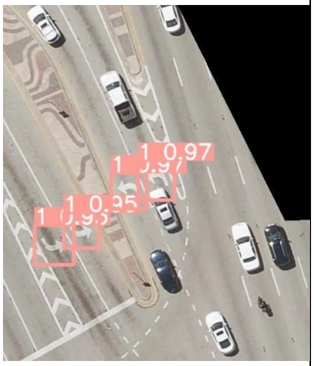
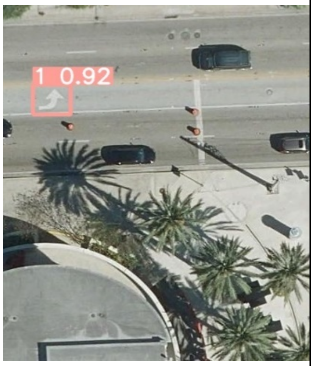
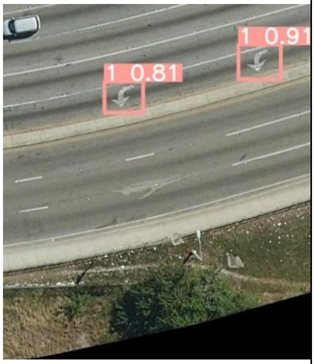
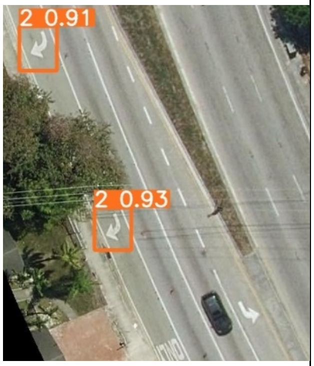
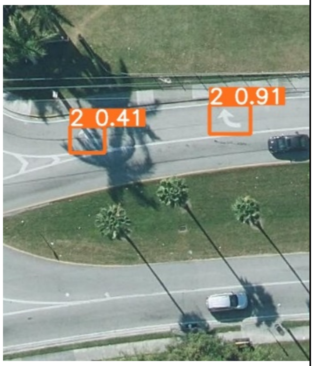
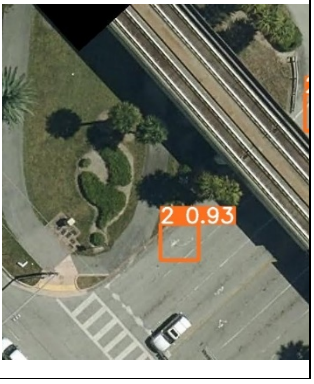


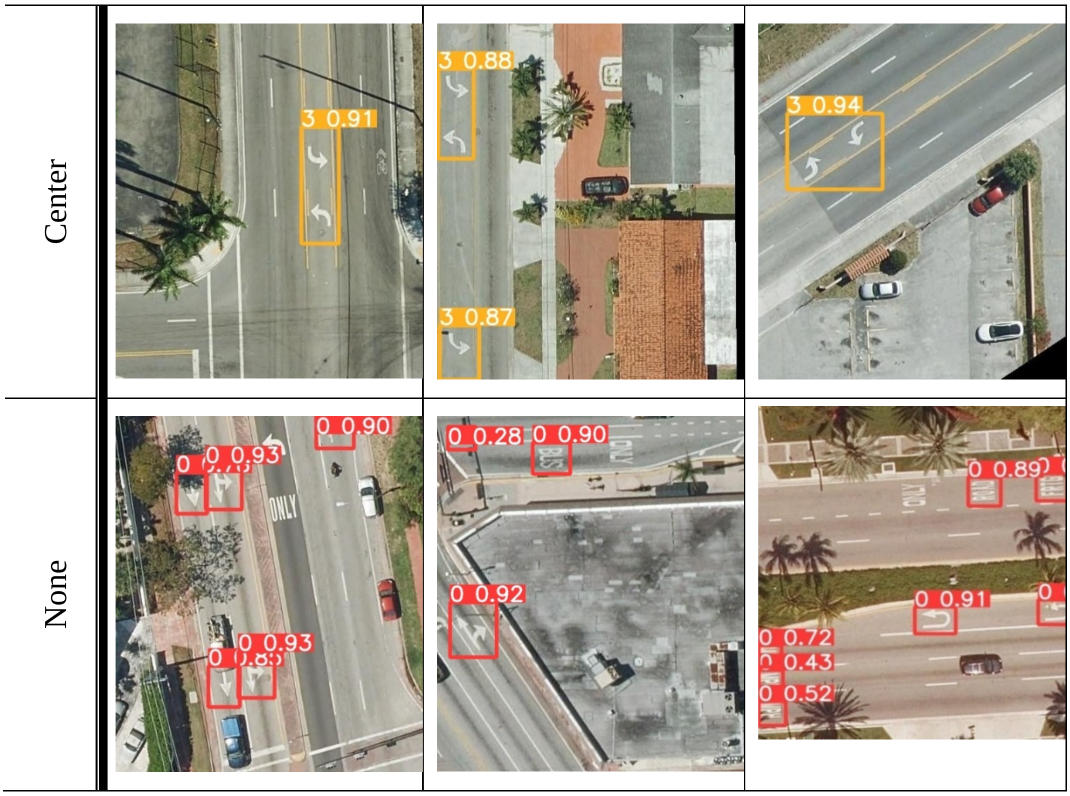
```python
#Function to split Image into tiles

def splitImage(image_path, output_path, num_rows, num_columns):
    # Read the image using cv2
    image = cv2.imread(image_path)

    # Get the dimensions of the image
    height, width, z = image.shape

    # Calculate the width and height of each grid cell
    cell_width = width // num_columns
    cell_height = height // num_rows

    # Create the output directory if it doesn't exist
    os.makedirs(output_path, exist_ok=True)

    # Loop through rows and columns to split the image
    for row in range(num_rows):
        for col in range(num_columns):
            # Calculate the coordinates of the current cell
            left = col * cell_width
            upper = row * cell_height
            right = left + cell_width
            lower = upper + cell_height

            # Crop the image to extract the current cell
            cell_image = image[upper:lower, left:right]

            # Save the cell image
            cell_filename = os.path.join(output_path, f'_{row+1:02d}_{col+1:02d}.jpg')
            cv2.imwrite(cell_filename, cell_image)

```

(a)


Detection
Trained
Model
Input:
Masked images
Python API
Detection parameters
Output:
Coordinates of turning locations
Post-processing
Geocoded Turning Lane Inventory

(b)

Name	Examples of turning lane feature detection from images		
Multiple features			
			
left			
Right			



(c)

Figure 6 (a) Image processing step for tile detection, (b) turning lane detection framework, and (c) turning lane detection polygons and confidence scores on images.

Post-Processing

Finally, after applying the model on the obtained aerial images in Leon County, the total number of observed left, right, and center detections in Leon County was found to be 4,795 using the model. Redundant detections caused by the overlapping distance on image were removed at the post-processing stage . The turning configurations for state and local roadways can also be classified into groups depending on analytical objectives. The filters non-maximum suppression was applied, which kept the detection that overlapped and had the highest confidence level. Therefore, all the detected turning lane marking with over 10% overlap with lower confidence levels were removed. The detected features were converted from polygon shapefiles to point shapefiles for analysis.

Results and Discussions

The model's performance was assessed during the model training using precision, recall, and F1 scores. Afterwards, Leon County was utilized as the county where ground truth data were collected and the developed model's performances were assessed with respect to completeness, correctness, quality, and F1 score. The GT dataset of turning lane features on Leon County roadways were used as a proof of concept. After detection of left, right and center features, a total of 4,795 detections were observed, where 3,101 were classified as left, 1,478 were classified as right and 216 were classified as center lanes features. After visual inspection, a total of 5,515 visible turning lane markings for left, right, and center lane features were collected as GT dataset using the masked photographs as the background. Note that each turning lane may exhibit a range of 1-6 features, such as left-only or right-only markings. Consequently, a single lane may contain multiple consecutive turning features aligned in a straight line and each lane may consist of several features that collectively define that lane. However, to perform a robust analysis, the assessment was done by directly comparing individual turning features on the aerial images. Figure 7 shows the GT dataset and detected turning lane markings in Leon County. Although some of the turning lane markings, especially the left only marking, were missed by the model, the overall performance of the model was reliable. This is mainly because of various reasons such as occlusions, faded markings, shadows, poor image resolution, and variety of the pavement marking design.

As noted, the suggested model has identified turning lane markers with a minimum confidence score of 10%. For the purposes of this case study, the model identified turning lane markers (M) on the local roads in Leon County were retrieved. On the GT, a similar location-based selection methodology was used. The suggested model's performance was assessed by examining the points that were discovered within the polygons and vice versa at various confidence levels of 90%, 75%, 50%, 25%, and 10%, respectively (Table 2). A confusion matrix was used to visualize the model's performance against the GT dataset of Leon County (Figure 8a).

This study's major goal was to assess the accuracy and performance of the proposed model's predictions and contrast them with a ground truth dataset. Separate evaluation analysis has been performed using each "left_only", "right_only", and "center" detections of the developed model. The accuracy and performance on turning lane model were assessed after testing the model's correctness (precision) and completeness (recall) using a complete ground truth dataset and measuring the F1-score. The F1-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.

Finally, the suggested model's performance was assessed using the criteria of completeness (recall), correctness (precision), quality (intersection over union), and F1. The model's performance was also visualized using a circus plot [44] in Figure 8b. These criteria were initially utilized in [45] and [46] for the goal of highway extraction, and they are now often used for performance evaluation of the related models [47,48]. The following selection criteria are necessary to determine the performance evaluation metrics:

i. GT: Number of GT turning lane polygon,
ii. M: Number of Model detected turning lane points
iii. False Negative (FN): # of GT turning lane polygon without M turning lane point,
iv. False Positive (FP): # of M turning lane points not found within GT turning lane polygon,
v. True Positive (TP): # of M turning lane points within GT turning lane polygon,

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)

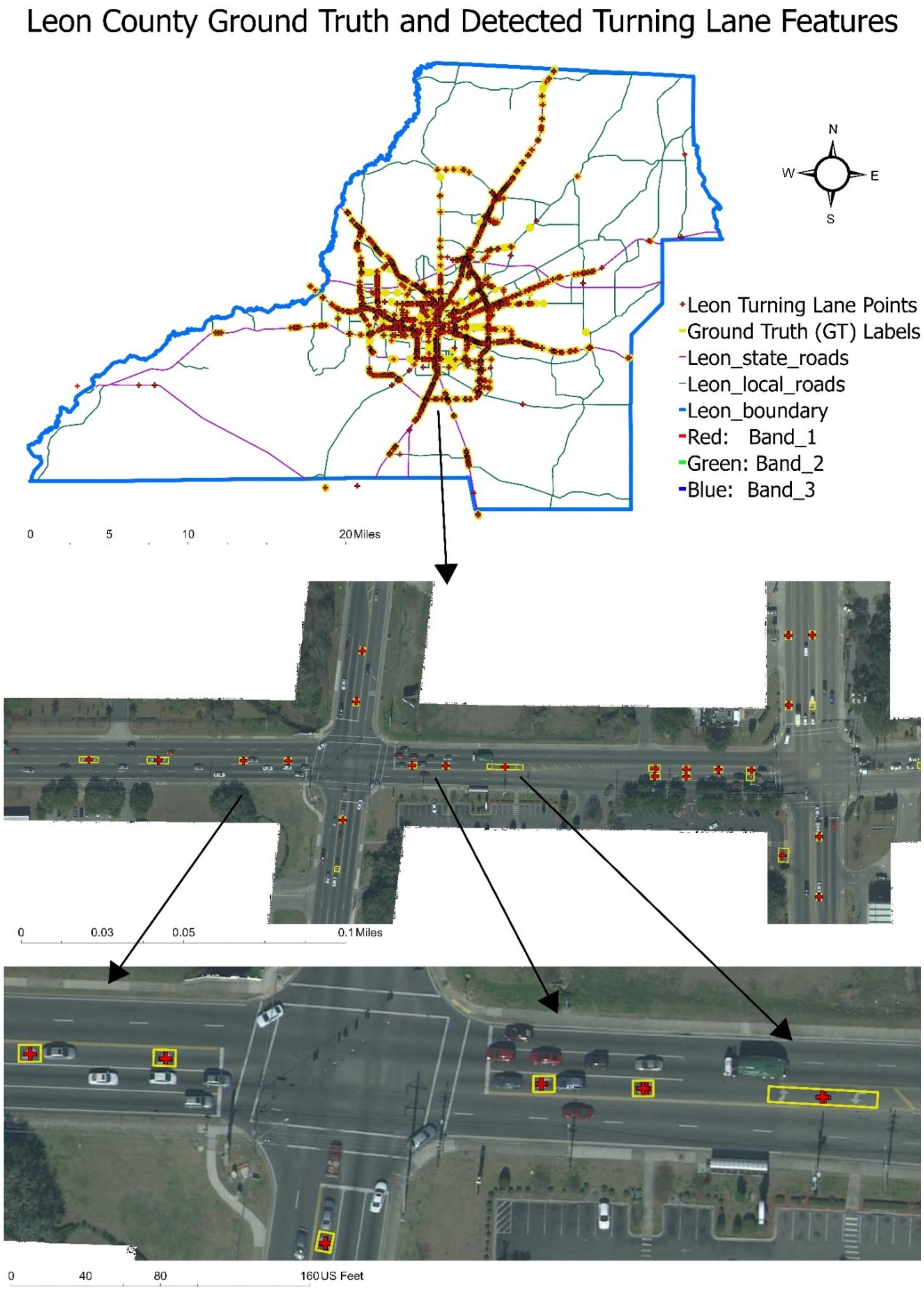
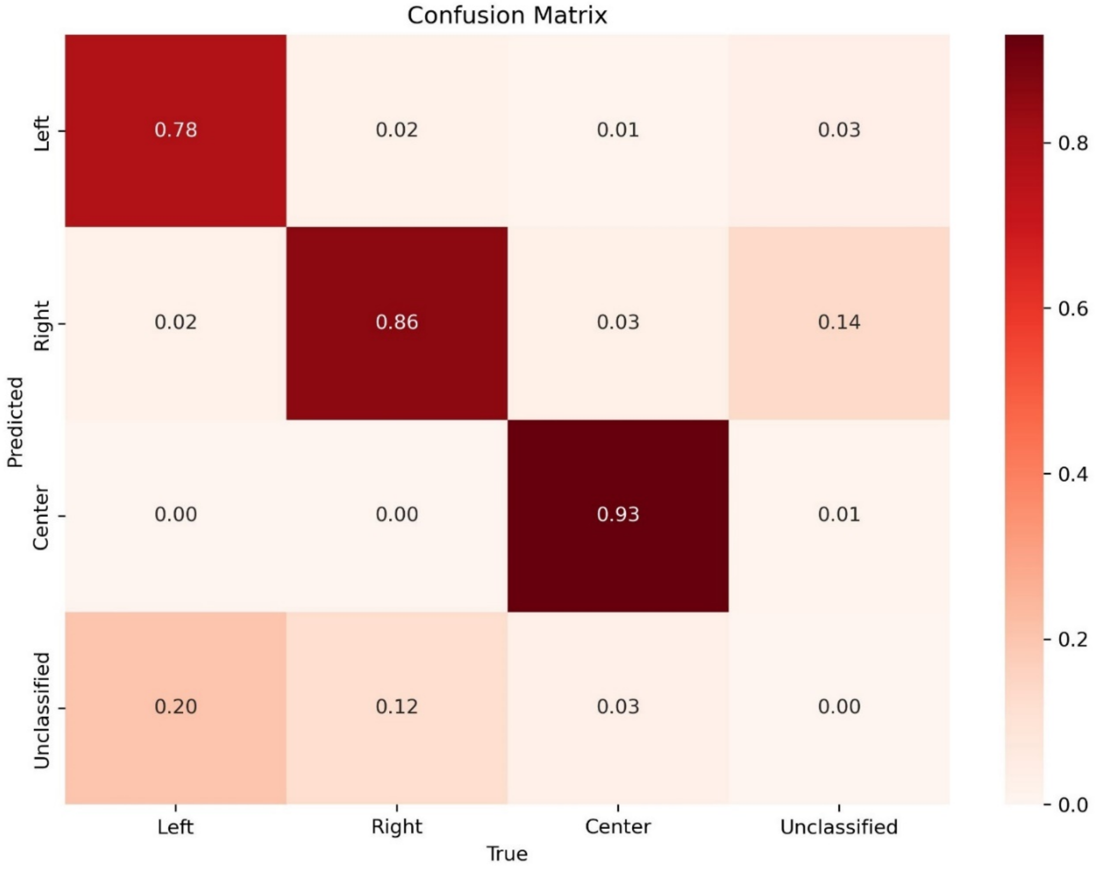


Figure 7. Manually labeled Ground Truth turning lane markings (GT) and detected turning lane markings in Leon County, Florida.

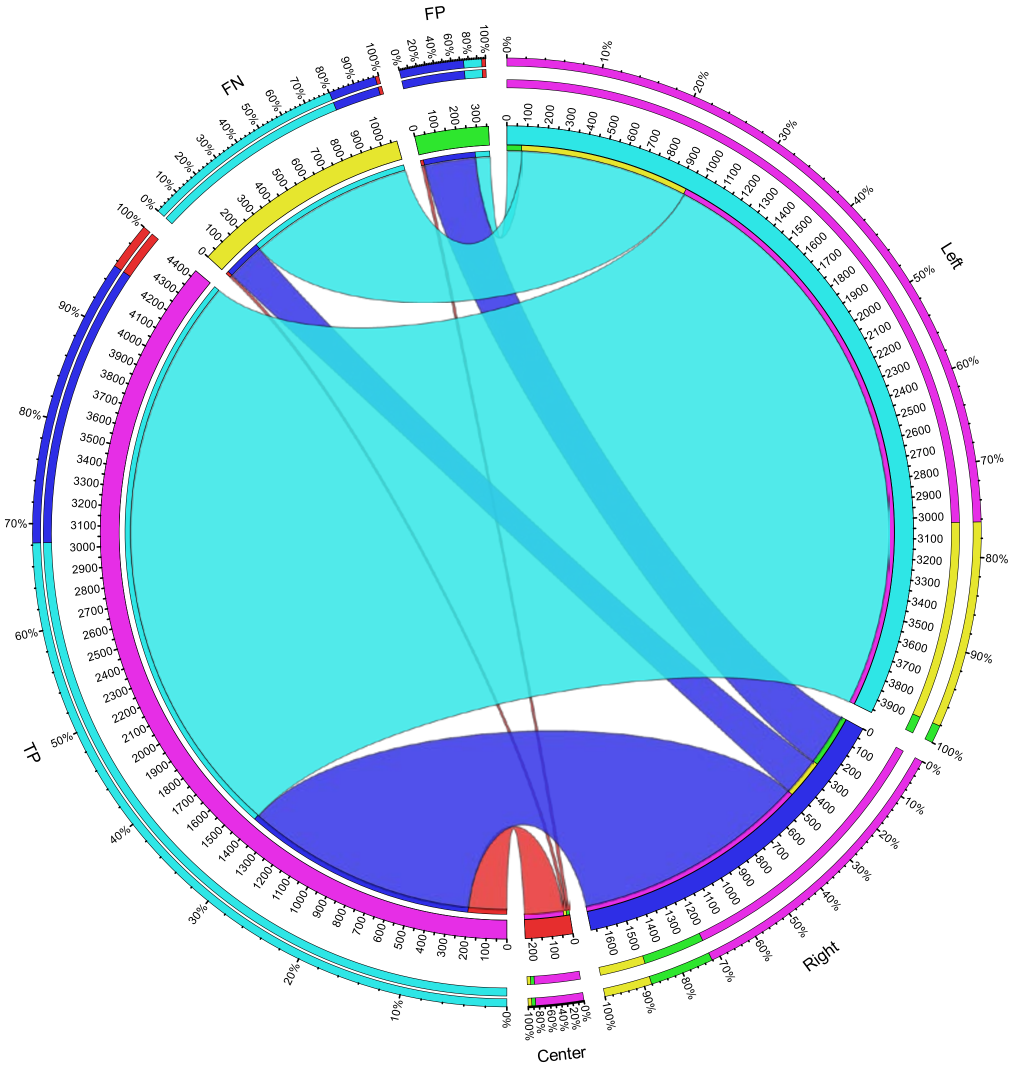
Based on the findings, we observe that this automated turning lane detection and mapping model can averagely detect and map 57% of the turning lanes with 99% precision at 75% confidence level and an F1-score of 71%. At a lower confidence level of 25%, it can detect ~80% of the turning lanes with 96% precision and an F1-score of 87% . At 75% confidence level, the average quality of the model's detection is 58% whereas the average quality of the model is also 77% at 25% 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 turning lane 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, zero or 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 and the total number of detections. It can also be observed that the poor distinctiveness of the detection features 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 lower detection confidence levels. 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)

doi:10.20944/preprints202405.0021.v1



(b)

Figure 8. (a) Confusion matrix of predicted versus GT (true) turning features in Leon County, Florida, (b) Visualization (circus plot) of performance evaluation metrics between the ground truth (GT) and predictions made by the YOLOv5 based 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 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 the plot, about 72% of right only detections are true positives while about 18% are false negatives. Also, over 76% of left_only are true positives.

The detected turning lanes were classified under different confidence levels. The final list is shown in Table 2. The extracted roadway geometry data can be integrated with crash and traffic data especially at intersections 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 turning lane locations with other geometric features like crosswalks, school zones, and analyzing the crashes occurring around the turning lanes at intersections

Table 2. Model performance evaluations.

Leon County Ground Truth Comparison Analysis								
Turning Lane Model								
left_only: GT = 3891								
Confidence (%)	**M**	**TP**	**FP**	**FN**	**Completeness (%)**	**Correctness (%)**	**Quality (%)**	**F1-score (%)**
90	251	251	0	3640	6.45	100.00	6.45	12.12
75	1392	1383	9	2508	35.54	99.35	35.46	52.36
50	2146	2125	21	1766	54.61	99.02	54.32	70.40
25	2683	2648	35	1243	68.05	98.70	67.45	80.56
10	3101	3026	75	865	77.77	97.58	76.30	86.56
right_only: GT = 1409								
Confidence (%)	**M**	**TP**	**FP**	**FN**	**Completeness (%)**	**Correctness (%)**	**Quality (%)**	**F1-score (%)**
90	197	196	1	1213	13.91	99.49	13.90	24.41
75	887	871	16	538	61.82	98.20	61.12	75.87
50	1108	1066	42	343	75.66	96.21	73.47	84.70
25	1262	1153	109	256	81.83	91.36	75.96	86.33
10	1478	1210	268	199	85.88	81.87	72.15	83.82
center: GT = 215								
Confidence (%)	**M**	**TP**	**FP**	**FN**	**Completeness (%)**	**Correctness (%)**	**Quality (%)**	**F1-score (%)**
90	13	13	0	202	6.05	100.00	6.05	11.40
75	162	160	2	55	74.42	98.77	73.73	84.88
50	191	188	3	27	87.44	98.43	86.24	92.61
25	200	195	5	20	90.70	97.50	88.64	93.98
10	216	200	16	15	93.02	92.59	86.58	92.81

Conclusions and Future Work

This study investigates the utilization of computer vision tools for roadway geometry extraction focusing on Florida turning lanes as a proof of concept. This is a creative approach that uses computer vision technology to possibly replace labor- and error-intensive traditional manual inventory. The created system can extract recognizable turning 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 will 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 with other geometric features like crosswalks, school zones, and analyzing crashes that take place close to these locations.

However, the study also identifies notable limitations and offers recommendations for future research. Challenges arise from aerial images of roadways obstructed by tree canopies, limiting the identification of turning lane markings. Moving forward, the developed model can be integrated into roadway geometry inventory datasets, such as those used in the Highway Safety Manual (HSM) and the Model Inventory of Roadway Elements (MIRE), to identify and rectify outdated or missing lane markings. Future research endeavors will also focus on refining and expanding the capabilities of the model to detect and extract additional roadway geometric features. Additionally, there are plans to integrate the extracted left-only, right-only, and center lanes with crash data, traffic data, and demographic information for a more comprehensive analysis.

Author Contributions: The following authors confirm contribution to the paper with regards to Study conception and design: Richard B. Antwi, Kimollo Michael, Eren Erman Ozguven, Ren Moses, Maxim A. Dulebenets, and Thobias Sando; Data collection: Richard Boadu Antwi, Kimollo Michael, Eren Erman Ozguven; Analysis and interpretation of results; Manuscript preparation: Richard B. Antwi and Eren Erman Ozguven. All authors reviewed the results and approved the final version of the manuscript.

Acknowledgements: This study was sponsored by the State of Florida Department of Transportation (DOT) grant BED30-977-02. The contents of this paper and discussion represent the authors' opinions and do not reflect the official views of the Florida Department of Transportation.

References

1. 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.
2. Jalayer, M., Zhou, H., Gong, J., Hu, S., & Grinter, M. (2014). A comprehensive assessment of highway inventory data collection methods. In *Journal of the Transportation Research Forum* (Vol. 53, No. 1424-2016-117955, pp. 73-92).
3. Shamayleh, H., & Khattak, A. (2003). Utilization of LiDAR technology for highway inventory. In *Proceedings of the 2003 Mid-Continent Transportation Research Symposium, Ames, Iowa*.
4. Alzraiee, H., Leal Ruiz, A., & Sprotte, R. (2021). Detecting of pavement marking defects using faster R-CNN. *Journal of Performance of Constructed Facilities*, *35*(4), 04021035.
5. 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).
6. 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*.
7. Antwi, R. B., Takyi, S., Karaer, A., Ozguven, E. E., Moses, R., Dulebenets, M. A., & Sando, T. (2023). Detecting School Zones on Florida's Public Roadways Using Aerial Images and Artificial Intelligence (AI2). *Transportation Research Record*, 03611981231185771.
8. Carlson, P. J., Park, E. S., & Andersen, C. K. (2009). Benefits of pavement markings: A renewed perspective based on recent and ongoing research. *Transportation research record*, *2107*(1), 59-68.
9. Cho, Y., Kabassi, K., Pyeon, J. H., Choi, K., Wang, C., & Norton, T. (2013). Effectiveness study of methods for removing temporary pavement markings in roadway construction zones. *Journal of construction engineering and management*, *139*(3), 257-266.
10. Cheng, W., Luo, H., Yang, W., Yu, L., & Li, W. (2020). Structure-aware network for lane marker extraction with dynamic vision sensor. *arXiv preprint arXiv:2008.06204*.
11. Lee, S., Kim, J., Shin Yoon, J., Shin, S., Bailo, O., Kim, N., ... & So Kweon, I. (2017). Vpgnet: Vanishing point guided network for lane and road marking detection and recognition. In *Proceedings of the IEEE international conference on computer vision* (pp. 1947-1955).
12. Li, J., Mei, X., Prokhorov, D., & Tao, D. (2016). Deep neural network for structural prediction and lane detection in traffic scene. *IEEE transactions on neural networks and learning systems*, *28*(3), 690-703.
13. He, B., Ai, R., Yan, Y., & Lang, X. (2016). Accurate and robust lane detection based on dual-view convolutional neutral network. In *2016 IEEE intelligent vehicles symposium (IV)* (pp. 1041-1046). IEEE.
14. Huval, B., Wang, T., Tandon, S., Kiske, J., Song, W., Pazhayampallil, J., ... & Ng, A. Y. (2015). An empirical evaluation of deep learning on highway driving. *arXiv preprint arXiv:1504.01716*.
15. 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*, 101350.
16. Aghdam, H. H., Heravi, E. J., & Puig, D. (2016). A practical approach for detection and classification of traffic signs using convolutional neural networks. *Robotics and autonomous systems*, *84*, 97-112.
17. Balali, V., Ashouri Rad, A., & Golparvar-Fard, M. (2015). Detection, classification, and mapping of US traffic signs using google street view images for roadway inventory management. *Visualization in Engineering*, *3*, 1-18.
18. Zhang, X., Yuan, Y., & Wang, Q. (2018). ROI-wise Reverse Reweighting Network for Road Marking Detection. In *BMVC* (p. 219).

19. Tong, Z., Gao, J., & Zhang, H. (2017). Recognition, location, measurement, and 3D reconstruction of concealed cracks using convolutional neural networks. *Construction and Building Materials*, *146*, 775-787.
20. Panboonyuen, T., Jitkajornwanich, K., Lawawirojwong, S., Srestasathiern, P., & Vateekul, P. (2017). Road segmentation of remotely-sensed images using deep convolutional neural networks with landscape metrics and conditional random fields. *Remote Sensing*, *9*(7), 680.
21. Uçar, A., Demir, Y., & Güzeliş, C. (2017). Object recognition and detection with deep learning for autonomous driving applications. *Simulation*, *93*(9), 759-769.
22. 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*, *17*(2), 336.
23. Vattapparamban, E., Güvenç, I., Yurekli, A. I., Akkaya, K., & Uluağaç, S. (2016). Drones for smart cities: Issues in cybersecurity, privacy, and public safety. In *2016 international wireless communications and mobile computing conference (IWCMC)* (pp. 216-221). IEEE.
24. Xie, X., Yang, W., Cao, G., Yang, J., Zhao, Z., Chen, S., ... & Shi, G. (2018, September). Real-time vehicle detection from UAV imagery. In *2018 IEEE Fourth International Conference on Multimedia Big Data (BigMM)* (pp. 1-5). IEEE.
25. 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*.
26. Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440-1448).
27. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*.
28. 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*.
29. Foucher, P., Sebsadji, Y., Tarel, J. P., Charbonnier, P., & Nicolle, P. (2011, October). Detection and recognition of urban road markings using images. In *2011 14th international IEEE conference on intelligent transportation systems (ITSC)* (pp. 1747-1752). IEEE.
30. Azimi, S. M., Fischer, P., Körner, M., & Reinartz, P. (2018). Aerial LaneNet: Lane-marking semantic segmentation in aerial imagery using wavelet-enhanced cost-sensitive symmetric fully convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, *57*(5), 2920-2938.
31. 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.
32. Zhang, D., Xu, X., Lin, H., Gui, R., Cao, M., & He, L. (2019). Automatic road-marking detection and measurement from laser-scanning 3D profile data. *Automation in Construction*, *108*, 102957.
33. Xu, S., Wang, J., Wu, P., Shou, W., Wang, X., & Chen, M. (2021). Vision-based pavement marking detection and condition assessment—A case study. *Applied Sciences*, *11*(7), 3152.
34. Antwi, R. B., Takyi, S., Kimollo, M., Karaer, A., Ozguven, E. E., Moses, R., Dulebenets, M. A., & Sando, T. (2024). Computer Vision-Based Model for Detecting Turning Lane Features on Florida's Public Roadways from Aerial Images. *Transportation Planning and Technology*, 1-20.
35. United States Census Bureau (US Census). (2020). Population Estimates. https://www.census.gov/quickfacts/leoncountyflorida. Accessed June. 20, 2023.
36. Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.
37. 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).
38. Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263-7271).
39. Bochkovskiy, A., Wang, C. Y., & Liao, H. Y. M. (2020). Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*.
40. Jocher, G. (2020). YOLOv5 by ultralytics. *Released date*, 5-29. Https://Github. Com/Ultralytics/Yolov5. 2020.
41. Wang, C. Y., Liao, H. Y. M., Wu, Y. H., Chen, P. Y., Hsieh, J. W., & Yeh, I. H. (2020). CSPNet: A new backbone that can enhance learning capability of CNN. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 390-391).
42. Liu, K., Tang, H., He, S., Yu, Q., Xiong, Y., & Wang, N. (2021, January). Performance validation of YOLO variants for object detection. In *Proceedings of the 2021 International Conference on bioinformatics and intelligent computing* (pp. 239-243).
43. Horvat, M., & Gledec, G. (2022, September). A comparative study of YOLOv5 models performance for image localization and classification. In *33rd Central European Conference on Information and Intelligent Systems (CECIIS)* (p. 349).

44. 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.
45. Wiedemann, C., Heipke, C., Mayer, H., & Jamet, O. (1998). Empirical evaluation of automatically extracted road axes. *Empirical evaluation techniques in computer vision*, *12*, 172-187.
46. Wiedemann, C., & Ebner, H. (2000). Automatic completion and evaluation of road networks. *International Archives of Photogrammetry and Remote Sensing*, *33*(B3/2; PART 3), 979-986.
47. 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*.
48. 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.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.