

Electronic Theses and Dissertations, 2020-

2022

Development of Active Learning Data Fixing Tool with Visual Analytics to Enhance Traffic Near-miss Diagnosis

Jinyu Pei
University of Central Florida

 Part of the [Civil Engineering Commons](#), and the [Transportation Engineering Commons](#)

Find similar works at: <https://stars.library.ucf.edu/etd2020>

University of Central Florida Libraries <http://library.ucf.edu>

This Masters Thesis (Open Access) is brought to you for free and open access by STARS. It has been accepted for inclusion in Electronic Theses and Dissertations, 2020- by an authorized administrator of STARS. For more information, please contact STARS@ucf.edu.

STARS Citation

Pei, Jinyu, "Development of Active Learning Data Fixing Tool with Visual Analytics to Enhance Traffic Near-miss Diagnosis" (2022). *Electronic Theses and Dissertations, 2020-*. 1268.

<https://stars.library.ucf.edu/etd2020/1268>

DEVELOPMENT OF ACTIVE LEARNING DATA FIXING TOOL WITH VISUAL
ANALYTICS TO ENHANCE TRAFFIC NEAR-MISS DIAGNOSIS

by

JINYU PEI
B.S. University of Central Florida, 2020

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Civil, Environmental and Construction Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Summer Term
2022

Major Professor: Mohamed Abdel-Aty

© 2022 Jinyu Pei

ABSTRACT

This study proposes a software to upgrade the UCF SST's Automated Roadway Conflicts Identification System (ARCIS), a pixel-to-pixel manner automated safety diagnostics and conflict identification system. The system is developed to extract vehicles' trajectories and traffic parameters using unmanned aerial vehicles (UAV) video and utilizing deep learning techniques. A user-friendly tool to improve rapid system development with active-learning, data analysis, and visualization techniques is introduced, which is capable of traffic safety near-miss diagnostics based on the ARCIS output. Multiple approaches are used to enhance the system performance, including video stabilization, object filtering, stitching multiple videos, vehicle detection and tracing. In addition, the active learning technique based on Stream-Based Selective Sampling strategy is adopted for a human-in-the loop label correction that is developed in order to reduce the labeling time and cost. The system outputs 3D maps of vehicle speed, count and surrogate safety measures, which provide insights for traffic safety diagnosis. Ultimately, these functionalities were integrated into a comprehensive system for traffic safety applications. Previous studies only investigated methods for enhancing road traffic safety and traffic network data analysis; this study builds upon the literature but improves upon it with an efficient video processing methodology, a higher quality and accuracy result on traffic trajectory data, and the ability to visualize the data in various formats for traffic analysis.

ACKNOWLEDGMENTS

First of all, my most sincere thanks go to my advisor Dr. Mohamed Abdel-Aty, for the continuous guidance and support of my master's study and research, for sharing his pearls of wisdom with me.

Secondly, I would like to appreciate my committee members: Dr. Zubayer and Dr. Samiul Hasan, for their encouragement and suggestions.

Additionally, I want to acknowledge Ou Zheng, who has provided unlimited ideas and valuable suggestions for the system.

Last but not least, I would like to thank Siyuan Tang, Zihao Zhou, Jiahao Zhu, Youyou Cheng, Zijin Wang and Dunhang Wang for being my best colleague and my best friend. I would also like to thank my lovely cats Tuna for staying with me.

TABLE OF CONTENTS

LIST OF FIGURES	viii
LIST OF TABLES	x
CHAPTER 1 INTRODUCTION	1
1.1 BACKGROUND	1
1.2 MOTIVATIONS AND OBJECTIVES.....	2
1.3 METHODOLOGY	3
1.4 THESIS STRUCTURE	4
CHAPTER 2 LITERATURE REVIEW	5
2.1 COMPUTER VISION	5
2.1.1 Pre-processing.....	6
2.1.2 Detection	8
2.1.3 Tracking	13
2.2 TRAFFIC SAFETY EVALUATION.....	14
2.2.1 Surrogate Safety Measures	15
2.2.2 Conflict Type	16
2.3 ACTIVE LEARNING	17
2.3.1 Pool-Based Method.....	17
2.3.2 Stream-Based Method.....	18
2.4 VISUALIZATION	19
2.4.1 Web-based Visualization	20
2.4.2 Independent software applications Visualization	20

2.5 DATA FIX TOOL	21
CHAPTER 3 ARCIS SYSTEM.....	23
3.1. STABILIZATION.....	23
3.2. FILTERING OBJECT	25
3.3. STITCHING VIDEOS.....	26
3.4. GPS TO PIXEL CONVERTOR	27
3.5 DATA EXTRACTION.....	28
CHAPTER 4 ADDITIONAL SOFTWARE DEVELOPMENT	30
4.1 DATA PROCESSING TOOL.....	31
4.1.1 Filtering Object	32
4.1.2 Stabilization	33
4.1.3 Stitching Videos.....	35
4.1.4 Data Extraction	39
4.1.5 GPS to Pixel Convertor.....	39
4.2 DATA FIX TOOL	41
4.2.1 Start stage.....	42
4.2.2 Video player.....	42
4.2.3 Toolbox	43
4.2.4 Output	44
4.3 ACTIVE LEARNING	44
4.3.1 Data preparation.....	45
4.3.2 Pool Based Sampling	46

4.3.3 Stream-Based Selective Sampling	47
4.3.4 Results	48
4.4 VISUALIZATION TOOL	51
4.4.1 Vehicle Trajectory	52
4.4.2 Surrogate Measures.....	52
4.4.3 Speed and Acceleration.....	53
4.4.4 User-interface.....	53
CHAPTER 5 CASE STUDY	56
CHAPTER 6 CONCLUSION.....	62
REFERENCES	63

LIST OF FIGURES

Figure 1 A flow diagram for structure of stabilization	24
Figure 2 Filtering Object method.....	25
Figure 3 An example of matching key points for stitching	26
Figure 4 An example of transformation using perspective transform matrix	26
Figure 5 An example of converted pixel points for drone view and geographic map	28
Figure 6 An example of detected vehicle with unique ID and bounding box	28
Figure 7 workflow diagram	31
Figure 8 Data processing tool components	32
Figure 9 Filtering object example	33
Figure 10 Stabilization user-interface components.....	34
Figure 11 Colored bounding box surrounding the displayed target object.....	35
Figure 12 Stabilization result	35
Figure 13 Graphic user interface for stitching video	36
Figure 14 Undo button to reverse stitching action	37
Figure 15 Result of an image transformation	38
Figure 16 Before and after results after adjustment.....	38
Figure 17 Main GUI of data extraction.....	39
Figure 18 Main GUI of pixel to GPS convertor	40
Figure 19 GUI of data fix tool	41
Figure 20 Data fix tool functionalities chart	42

Figure 21 Video player control	43
Figure 22 An example of output before fixing the detected vehicle bounding box	44
Figure 23 AP value description based on IoU	45
Figure 24 Pool based sampling flow chart.....	47
Figure 25 Streamed-Based Selective Sampling flow chart.....	48
Figure 26 An example detection made using Detectron2 algorithm.	49
Figure 27 Percentage of the classification accuracy for stream-based and pool-based scenario..	51
Figure 28 An example of a vehicle trajectory.....	52
Figure 29 Data Visualization tool functionalities chart	54
Figure 30 An example of the Safety Analysis tab	55
Figure 31 An example of conflict event	55
Figure 32 An example of result after removed objects that could cause a detection error	57
Figure 33 An example of convert the pixel coordinates to GPS coordinates	58
Figure 34 An example of before and after result using Data Fix Tool	59
Figure 35 An example of data visualizer generates safety measurements for evaluation	59
Figure 36 Vehicles' trajectories on PET and TTC values are less than 1.5 s in a 3D heatmap....	60
Figure 37 Example of conflict type event.....	61

LIST OF TABLES

Table 1 Result of labeled training dataset.....	50
---	----

CHAPTER 1 INTRODUCTION

1.1 Background

Various methods could be utilized to conduct road safety analysis. The traditional crash analysis uses highly aggregated data to evaluate road safety for certain situations. Benefit from the wide employment of traffic infrastructure-based sensors generates big data in real-time that enables researchers to conduct crash likelihood analysis by aggregating it into shorter time intervals. By aggregating the data from nearby traffic detectors and the other infrastructure elements (e.g., weather stations, signals) at certain time intervals, precursors of crash occurrence or surrogate safety measure (SSM) could be identified, which could be used to identify road safety situations and propose corresponding strategies to prevent crashes. Although the deployment of traffic infrastructure sensors brings big data and enables the analysis of real-time crash risk, the rarity of crashes sometimes restricts the power of the statistical model when studying certain locations. Furthermore, another limitation of using highly aggregated data is that it does not include the heterogeneity in traffic and individual vehicle behavior, which makes it difficult to diagnose traffic safety from microscopic perspective. The recent advent in traffic video data and computer vision technics brought unprecedented opportunities to collect vehicle trajectory data and conduct safety evaluation based on conflict analysis. In order to boost the research in the field and provide insights for transportation practitioners, it is of great significance to develop and enhance an automated roadway conflicts identification system that incorporate the whole procedure from video processing to safety diagnosis.

1.2 Motivations and Objectives

Unmanned Aerial Vehicles (UAVs) are becoming increasingly used for data gathering in road traffic monitoring and transportation engineering due to a variety of factors, including mobility, ease of operation, low cost, and wide view range. Increased demand for UAV related safety-critical analysis necessitates the development of a practical UAV based safety system. Based on the UCF-SST Automated Roadway Conflicts Identify System introduced by Wu et al. (2020), this work proposes a system that investigates methods for enhancing road traffic safety analysis by using UAVs. Extraction of video data on recognition and classification of vehicle objects, and analysis of road user activity have all been the focus of this research. However, each function is performed in a separate branch, and it is required to develop a suitable smart learner-based fixing tool in order to expedite on correcting the incorrectly labeled data. In addition, there are needs of designing an efficient tool for the visualization of output traffic safety data. The purpose of this study is to adopt the ARCIS techniques to assure the capacity of all functions and develop additional functions to ensure the accuracy and user end experience. The objectives of this study are

- (1) developing a comprehensive system interface of existing ARCIS Python code
- (2) developing human in the loop data fixing tool
- (3) exploring various strategies of active learning and selecting one strategy for the tasks of increasing accuracy on object detection model
- (4) enabling the software to visualize vehicle trajectory, speed, count, conflict, and other safety indicators.

1.3 Methodology

The additional development that is being conducted, based on the original ARCIS system, can be divided into four separate tasks: Data Processing Tool, Data Fix Tool, Active Learning and Traffic Data Visualization.

The primary functionalities that are included in the Data Processing Tool are stabilization, object filtering, and stitching multiple videos, which assist in improving the performance of videos collected by UAVs. Moreover, the Mask region convolutional neural network (R-CNN) object detection model was implemented into the tool under the Data Extractor tab, whose primary purpose is to detect vehicles in videos and extra every vehicle's specific information in each frame, such as unique ID for each vehicle, pixel coordinate of bounding box. However, while pixel coordinates of vehicles can be visualized on their trajectory in a pixel coordinate, they cannot be visualized on a real-world map. As a result, the Data Processing Tool includes a pixel to GPS coordinates converter, as GPS coordinates can also be used to compute vehicle speed and acceleration.

To ensure the accuracy of ARCIS outputs, enabling the systems to generate related analysis data, a Data Fix Tool has been developed in the software that provides all-in-one access that allows humans to fix bounding boxes from an ARCIS output. Although the Data Fix Tool has accomplished its purpose of appropriately annotating data, an active learning strategy based on the fixed data can contribute to improving detection model accuracy. In this study, Stream-Based Selective Sampling strategies have been applied to an active learning algorithm during model training which to improve the model learns on identifies objects at each time step.

With accurate and reliable trajectory data, the surrogate safety measurement (e.g., TTC/PET) and conflict type was calculated by the system as an indicator of conflict probability at a roadway location for near-miss traffic events. To visualize the measurements for users to understand the results and discover traffic safety problems and patterns, a 3D map based on the MapBox Studio platform has been integrated into the Visualization Tool of this software. The Visualization Tool can illustrate the trajectory of each vehicle and different move speeds, acceleration line chart graphs, and surrogate safety measurements heat map. The software interface is designed to maximize clarity and simplicity while avoiding user interaction complexities with cross platform PyQt5 framework of graphical elements as the development environment.

1.4 Thesis Structure

The remaining thesis sections are organized as follows: Chapter 2 provides a brief overview of previous research; Chapter 3 describes the existing ARCIS system; Chapter 4 discusses additional ARCIS system development; Chapter 5 provides a location-specific case study. Finally, the conclusion is delivered in Chapter 6.

CHAPTER 2 LITERATURE REVIEW

2.1 Computer Vision

Due to a growing number of automobiles and limited infrastructure resources, traffic safety is becoming increasingly problematic. Researchers have incorporated computer vision techniques into traffic monitoring and analysis. (Kastrinaki et al., 2003) note that computer-vision technologies play a significant role in video-based traffic analysis and are widely used for traffic safety diagnostics and autonomous vehicle perception. Various techniques have been utilized to obtain data from video images including feature-based detection & tracking, background subtraction, and optical flow (Rahman et al., 2013; Meng et al., 2017). Utilizing image processing and pattern recognition techniques enables the system to monitor the road, track vehicles, and estimate speed for traffic analysis purposes. In recent years, many studies applied deep learning approaches to extract traffic parameters from videos that could be utilized to overcome the limitations of the traditional approaches under uninterrupted traffic flow conditions or the changes in environments (e.g., shadows, intricate ground conditions) (Kim et al. 2019). Besides the trajectory extraction, vehicle localization for the traffic intersection areas also has a significant influence on road safety diagnostics when calculating surrogate safety measures including TTC and PET. Additionally, real-time video processing is required in numerous other applications, including ultrasound image improvement, traffic monitoring, and camera stabilization.

2.1.1 Pre-processing

2.1.1.1 Camera calibration

Camera calibration, by detecting the camera location and measuring the size of an object in world units, minimizes lens distortion. Techniques such as automated calibration using parallel coordinates and cascaded Hough transform to analyze the trajectories for roadside surveillance cameras (Dubska et al., 2015). Such automated calibration also works with various road settings with either one or multiple lanes, intensity of background obstruction, etc. And boundless viewing angles can also be applied with this technique. By applying automated camera calibration, one must assume that at least one part of vehicle movement is straight.

2.1.1.2 Video stabilization

Sebastiano Battiato, (Dubska et al., 2015; Rahmat-Samii & Topsakal, 2021) states that the purpose of video stabilization is to reduce unwanted shakes and jitters while changing the focus on moving objects or intentionally panning the camera without negatively affecting image quality. Unstable images are usually caused by undesired camera movement and hand jiggling, while unwanted camera position fluctuations result in unstable image sequences. (Wang et al., 2012) describe a method for determining the purposeful motion using an adaptive compensation algorithm to address the issue of unwanted jitter in unmanned aerial vehicles (UAVs). Motion filtering is a crucial stage in the video stabilization process since it retains visual information and removes image jitter when the pre-path is determined to be straight. To ensure the visual quality after processing an over-long image sequence, (Wang et al., 2012; Yang et al., 2009) have proposed a technique for video stabilization based on the particle filtering framework. By extending particle filtering to the estimation and tracking of camera motion parameters in video

sequences, the error variance has been theoretically reduced compared to estimation without particle filtering.

2.1.1.3 Video filtering

To filter an object in a video is to remove unwanted objects, which is a common task in preparing a video for improved detection accuracy of a specific class of object. The proposed computational filtering approach has been studied lately by many researchers. (Kamel et al., 2008) address the problem of removing a moving object from a video sequence using computer vision techniques by filling the empty object area with the image of the background. They utilize SSD-based feature matching to register background images and extract moving objects frame by frame. Another approach is to use noise filtering to eliminate undesired background movements and objects. This creates a barrier to identifying moving items (Agarwal et al., 2016; Kamel et al., 2008). A system has been presented by T. Le et al. in 2019 that enables the user to draw and identify one or more unnecessary objects. Utilizing a CNN-based detector to refine the user-selected segmentation mask is the initial step. The annotation will then propagate throughout the video to remove the object automatically. Object extraction is a technique for detecting subjects that appear in a video scene by suppressing the background. By implementing the usual background-updating approach, (Rahman et al., 2013) applied the second order filter in the gradient direction to extract relevant edges of moving objects, but the approach could not be used in real-time applications or for noise reduction due to the heavy computing.

2.1.1.4 Video alignment

Video Alignment is to spatio-temporally align two videos, through temporal correspondence, followed by spatial registration of all the temporally corresponding frames. In several fields of computer vision, such as action recognition and change detection, video alignment is essential. (Purushwalkam et al., 2020) present an alignment method that joins patches in each frame of the first video by matching frames from the second video. They have created cycles in videos of the same action class that track patches within a video, match it to a patch in another video, trace this patch back in time, and finally match it to the original video. This technique has achieved its objective of being more efficient at matching objects in the same state across videos. ((Agarwal et al., 2016; Diego et al., 2013; Kamel et al., 2008; Purushwalkam et al., 2020)) have proposed a novel method for combining spatial and temporal alignment into a single alignment framework. This new approach can handle the alignment of sequences captured at various times by independently moving cameras that follow a similar trajectory. This was accomplished by combining the estimation of spatio-temporal parameters into a conventional pairwise Markov random field (MRF) and limiting the frame correspondence and spatial transformation to the neighborhood. This method outperforms the synchronization accuracy on sequences recorded from vehicles driving along the same track at different times.

2.1.2 Detection

According to (Meng et al., 2017) traffic algorithms remain common among many systems for making analysis of videos which have been presented to the software. This is done through the application of computer vision technologies which support the detection of certain elements within an image to identify an object. These methodologies combine elements like image recognition and

statistical analysis to pinpoint and even link objects within certain images as also appearing on other images (Guo et al., 2021). These techniques are employed in detection of images of vehicles and individuals for analysis to predict their movement. Background difference methods can be applied in order to differentiate the objects and point out specific vehicles using features that can be attributed and identified through the video. The combination of color segmentation, morphological gradient and trajectory analysis can be critical in processing the information presented by video. Traffic flow analysis systems typically collect traditional traffic metrics or traffic incident detection. Faster R-CNN and YOLO are two of the most popular object detection frameworks today for vehicle detection. For example, utilizing temporal information in the design of a building can help detect and track objects faster.

2.1.2.1 One stage detector

A one-stage detector predicts all the bounding boxes with only a single pass through the neural network. This is significantly quicker and more suitable by skipping the region proposal stage and conducting detection over a possible position coordinated value.

YOLO (You Only Look Once) and SSD are the most prevalent examples of one-stage object detectors. Many researchers have proposed a new network structure based on the algorithm of YOLO detectors. (Lan et al., 2018) have proposed the "YOLO-R" network structure to improve the accuracy of pedestrian behavior detecting abilities. The fundamental improvement of their work is to increase from 12 to 16 the number of Passthrough layer connections in the original YOLO network structure. The three Passthrough layers added to YOLO-R can convey the network's fine-grained features to the deep network, allowing it to acquire a deeper understanding of shallow pedestrian feature data. Although object identification systems based on deep learning

have made significant strides in accurately detecting different classes, they are still incapable of handling very small objects, such as those in aerial and satellite video with the object less than 10 pixels in width. Pham M-T et al. (2020) ((Lan et al., 2018; Pham et al., 2020) introduce You Only Look Once (YOLO)-Fine, an improved one-stage deep learning-based detection model based on the structure of YOLO. To achieve the purpose of detecting small objects with both high accuracy and efficiency, two coarse detection levels have been substituted with two fine detection levels to optimize the object search grid so that it can recognize and identify objects smaller than eight pixels per dimension.

SSD is a single-shot detector for multiple objects in image that is substantially more accurate and faster than the previous detectors YOLO (Kolekar & Dalal, n.d.). SSD is designed to accurately recognize a large object by employing low-resolution feature maps of the deep layer, however, the performance accuracy for recognizing a small object is insufficient when applying a high-resolution shallow layer that lacks an amount of high-level semantic information. (Choi et al., 2021) have addressed the issue and proposed an improved SSD using enhanced feature map blocks (SSD-EMB) to focus on the object regions rather than the background and provide additional semantic information without affecting the model's detection speed. Using EMB to adapt the high-resolution feature map of the shallow layer to focus on the object regions and improve the detection performance of small objects. Their efforts were targeted for satellite photo and traffic analysis. The new SSD detector using trident and squeeze and extraction feature fusion (SSD-TSEFFM) has been proposed from (Hwang et al., 2020). Using two modules of TFM and SEFFM to effectively detect small objects, where the TFM module that can scale changes based on the scale diversity and SEFFM is utilized to provide additional semantic information.

2.1.2.2 Two stage detector

Different from the One-stage detector, the two-stage detector divides object detection into two steps: the first step is used to Generate region proposals to extract object regions, while the second step uses convolutional neural networks to classify and refine the object's localization. According to (Ren et al., 2017; Sun et al., 2020)) research focused on comparing the outcomes of the different object detection approaches on SSD and Faster R-CNN. Based on execution time, the accuracy of the detection system, and memory use during execution, SSD has faster processing with less memory usage but lacks precision. Faster R-CNN is the exact opposite of SSD, resulting in more accurate prediction and detection but requiring more time. Typical representatives: Region Based Convolutional Neural Networks (R-CNN), Fast RCNN.

Fast R-CNN is a unified network with a high-precision detection method. (Ren et al., 2017; Sun et al., 2020)) introduce a Region Proposal Network (RPN) that shares convolutional features from image with the Fast R-CNN detection network. They are classifying the proposed regions into object categories using the Fast R-CNN method and have shown more benefits due to its better accuracy in comparison to other detection models. Using Fast RCNN, A (Ullah et al., 2018) improved the performance of pedestrian recognition in infrared images by modifying the method in two ways: one for accuracy by adding an additional convolutional layer to the network, and the other for speed by reducing the number of input channels from three to one. Fast R-CNN may also identify the detection of vehicles in aerial images. In the paper of (Sommer et al., 2017) , the authors systematically analyze how to adapt Fast R-CNN and Faster R-CNN for vehicle detection in aerial imagery. They have proposed relevant changes to the characteristics of aerial images and significantly improved the performance of both detectors on detecting small objects. Adapting the anchor boxes of the RPN and the resolution of the output of the final con-volitional layer used as

a feature map to compensate for the aerial imagery achieves the significant improvements. (Sakla et al., 2017; Sommer et al., 2017))have investigated the fundamental parameters of the faster R-CNN algorithm that impact the capacity to recognize small targets in overhead imagery. To localize small objects in multimodal imagery, they adopt shallower layers of the base architecture. Several modifications have been proposed to increase the detection accuracy of vehicles in aerial imagery, such as adding semantic information or merging features from multiple layers.

2.1.2.3 Anchor free detector

Both YOLO and SSD rely on anchors to refine the final location of detection. Typically, anchor is defined as a grid of image coordinates at all possible locations, with varying scale and aspect ratio. The CenterNet is an example of an anchor-free detector; it is a deep detection architecture without anchors. The main advantage of this structure is that it replaces the traditional NMS (Non-Maximum Suppression) method with a much more elegant method that is natural with the flow of CNNs. Rather than focusing on whether the predictions overlap the object, this approach focuses on the location of their centers to sort them for relevance. It features two customized modules named cascade corner aggregation and center aggregation, which give the central regions a more recognizable appearance by enhancing information collected at the left and right corners respectively. Liu et al. (2021) have used the CenterNet network and enhanced images approach to determine the vehicle's stopping state. They initially enhance the image by improving its contrast and sharpness. The enhanced image was then analyzed using the CenterNet network to detect the wheel position. Centernet uses transposed convolution and can effectively reconstruct image semantic and location information.

2.1.3 Tracking

The single object video tracking task is to locate the same target in all the other frames, which requires the determination of a target's trajectory from a video sequence. The ability to track any object could be beneficial for video analytics and traffic monitoring. Tracking objects is a challenging task due to many factors such as camera motion, low resolution, out of view, and similar objects. There are two types of tracking methods: short-term tracking and long-term tracking. The length of the video sequence is not only distinguished between short-term and long-term tracking, but there are also further distinctions between the two. A longer video sequence increases the probability of an object leaving the field of view or totally disappearing for a period before reappearing. Therefore, tracking algorithms that do not require re-detection approaches for a typical short-term tracking.

2.1.3.1 short term

There are several state-of-the-art approaches to tracking by detection, which make use of discriminatively trained classifiers or regressors to distinguish the target from the background. The Discriminative Correlation Filter (DCF) is showing excellent performance on the standard short-term benchmarks for tracking. These methods use the properties of circular correlation and the Fast Fourier Transform (FFT) to develop a least-squares regression model to predict confidence scores. (Lukezic et al., 2017) have used the concepts of DCF tracking to reflect the quality of the learned filter and it is used as a feature weighting coefficient in localization. (Zhong & Jianbo, 2010) have proposed two approaches for short-term object tracking: the Flock of Trackers (FoT) and the Scale-Adaptive Mean-Shift (ASMS) for multiple trackers and detectors. FoT is a building block in complicated frameworks for tracking multiple objects. The ASMS technique adds

background information into gradient optimization to reduce tracker failures in the presence of background clutter.

2.1.3.2 long term

A long-term tracker focuses primarily on failure recovery and requires detection of target absence and re-detection of the target upon its return. The state-of-the-art learning adaptive discriminative correlation filters (LADCF) tracking algorithm localizes the target in every frame and re-detects it if it disappears by reformulating the appearance learning model. (Ma et al., 2015) present an efficient long-term vision tracking algorithm. The method employs discriminative correlation filters to effectively estimate the translation and scale variations of target objects. The size is determined by scanning the target appearance pyramid exhaustively and the translation is estimated by modeling the temporal context correlation. To avoid internal identity switch, (Sun et al., 2020) present a clustered-based tracklet generating method that takes into account the similarity between any two detections in a tracklet. To learn long-term properties of tracklets for association, create a motion evaluation network (MEN) and an appearance evaluation network (AEN).

2.2 Traffic safety evaluation

The establishment of safety performance functions that relate the number of crashes or crash rate to a series of operational variables (e.g., average annual daily traffic (AADT), average speed) has been the focus of traffic safety research. However, the emergence of vehicle trajectory data has become increasingly popular for safety analysis. Most of the trajectory related research are based on the analysis of conflicts, which largely rely on the calculation of surrogate safety measures.

2.2.1 Surrogate Safety Measures

Surrogate safety measures have been widely adopted for analyzing traffic conflicts. Although there are a variety of surrogate safety measures that may be considered to calculate conflict indicators for safety diagnostics, TTC and PET remain the most common due to their effectiveness and simplicity. Time to collision (TTC) is defined as the time remaining between two vehicles before a collision, based on their speed and trajectory. (Goyani et al., 2021) . Researchers have used TTC as a traffic conflict indicator for video analysis procedure at each signal cycle from the recorded videos to collect rear-end conflicts and various traffic variables including traffic volume, maximum queue length, shock wave speed (Essa & Sayed, 2018) . TTC was selected as the primary indicator because it is considered as an appropriate conflict indicator for recognizing rear-end collisions between vehicles passing in the same lane. For the rear-end collision scenario, TTC refers to the time taken for a collision to occur at the prevailing speeds, distances, and trajectories associated with the driver's vehicle and the closest lead vehicle (van der Horst & van der Horst, 1990).(Kiefer et al., 2006) have used TTC results to develop an alert timing approach for a forward collision warning system intended to assist drivers in avoiding rear-end crashes with the vehicle ahead. In 2019, Guido et al. compared real accident locations and simulated risk areas in an urban road network by applying surrogate safety measures to the SSAM simulation environment to identify potentially risky vehicle interactions. This study uses the TTC and post-encroachment time (PET) values to identify unsafe conflicts under a predetermined threshold, which can provide reliable safety evaluation results for road networks. PET represents a measure of time difference between the instance of encroachment and the arrival of a potential collision object from the vehicle. In contrast to TTC, which only applies in the case of collisions saturation, PET includes "near miss" scenarios that indicate the extent to which the vehicles avoided each other.

Furthermore, PET measurement is more convenient than other indicators because relative speed and distance are not required (Songchitruksa & Tarko, 2006). (Wu et al., 2020) conducted research on automated traffic safety diagnostics solutions that uses deep learning techniques to process traffic videos collected by unmanned aerial vehicles (UAV). Using over 10,000 vehicle samples from UAV videos from various intersections, PETs were collected and measured at the pixel level for each conflict event based on the video vehicle trajectories. Essentially, the PET is determined as the difference in time between the departure of the first vehicle from the pixel and the arrival of the second vehicle at the pixel. The results of its safety diagnostics indicate that the value of PET is a significant indicator for identifying rear-end collisions at the intersection under study.

2.2.2 Conflict Type

Conflict Type is considered to classify the crash type with the highest probability, making it an alternative to crash-based analysis. Crash types including rear-end, sideswipe, angle/left-turn/right-turn, and angle/left-turn/right-turn, as well as crash severity. Wu et al. proposed a methodology to identify conflict type by using traffic intersections video collected from UAV (Wu et al., 2020). They use the angle between the movement directions of two vehicles to determine crash types. First, conflicts can be found by excluding occurrences with a PET value below the threshold. To determine the specific locations and times of identified collisions, the earliest arrival time and pixel of the second vehicle are used as the conflict point and conflict time, respectively. Then, the direction of vehicle movement, the intersecting of pixels (IOP), and vehicle occupancy are then used to determine conflict types (i.e., head on, angle, rear-end, sideswipe). IOP is used to evaluate whether the second car is following the first vehicle, which can be used to establish whether the dispute is a potential rear-end accident or not.

2.3 Active Learning

The objective of active learning is to effectively recognize thousands of labeled occurrences and classify data into their respective categories. Various learning techniques, including semi-supervised learning, weakly supervised learning, active learning, and transfer learning, have been developed to help recognize and classify objects. Face recognition, information extraction, categorization and filtering, and object recognition are among the most common categories requiring active learning. Active learning is the most effective and high-performance classifier for reducing labeling costs by requesting human labeling of only informative occurrences, as (Mizokami, 2018) demonstrates.

2.3.1 Pool-Based Method

The most common form of active learning is the pool-based strategy, which consists of a massive collection of unlabeled data (Mizokami, 2018). The objective of pool-based active learning is to select the most valuable input samples from the pool as test input samples. (Rawat et al., 2022) has demonstrated the effectiveness of pool-based active learning on semantic segmentation, which can identify a classifier for each pixel within an image. They have described the usage of a pool-based active learning scenario that combines a significant amount of unlabeled data with both a union of labeled and unlabeled data sets. Then, repeatedly train their segmentation model using labeled sets. In each training process, the most information of unlabeled data was selected and labeled by humans, then returned to the training model. A query function is to calculate an informativeness score for each image based on predictions made in an unlabeled data collection. This cycle will continue until they achieve a satisfactory performance score.

2.3.2 Stream-Based Method

The presumption of stream-based selective sampling is that unlabeled instances must be either free or inexpensively acquired, which the learner can judge whether the set of instances is usable or not. (Huang et al., 2019) presents a classification model for hand-written alphabet images utilizing stream-based active learning for reinforcement learning. A total of 32460 samples written by 20 separate individuals are evaluated for experimentation. During the training process, 30 images were selected at random without replacement, and stream-based active learner to judiciously determine a continuous decision made to either query or predict the label. Furthermore, (Chen et al., 2012) as a team develop a stream-based learning framework capable of performing active joint class discovery and boundary learning by extending the Query-by-Committee (QBC) algorithm paradigm to discover unknown classes and to deal with multivariate normal likelihoods, which are frequently encountered in vision problems. The proposed framework for active learning minimizes labeling costs compared to passive random labeling and outperforms current state-of-the-art active learning approaches. Stream-based active learning is most often used for time-series data, such as camera video data (Saunier et al., 2004).

Different machine learning algorithms contain different properties and serve unique purposes. While processing certain collections of data, it is better to choose a model with one type of model class, then based on that model's property, observe, and transfer the active learning result onto a separated class with different properties to collect the changes.

2.4 Visualization

Data visualization is transforming data into a graph, chart, or another visual format critical for understanding and analysis. Data presented in graphics make the data studied more engaging and accessible for many stakeholders (Berres et al., 2021; Chen et al., 2012)) Data visualization is a technique for displaying patterns, trends, and relationships between data elements using computer visuals (Healy, 2018). Multiple unique systems for the visualization of traffic safety data at an adaptable scale have been demonstrated for use in urban traffic. In certain studies, the abstract and spatial graphics were evaluated, as well as pedestrian safety and collision issues. Even in dense visualizations, traffic patterns have basic qualities that can be exhibited and analyzed: location, direction, and intensity (Scheepens et al., 2016). Data can be presented in a variety of ways; The use of heat maps allows us to distinguish between the software's dynamic environment and its general environment. It enables the analysis of the system's temporal dimension while also keeping in mind the larger context that is frequently required to comprehend the system (Benomar et al., 2013). For instance, a color-coded heatmap might be used to illustrate hot sections or roadways in a traffic network (Lee et al., 2020). Traffic trajectories, roadmaps in a large-scale region, or traffic flow in a dispersed network are all displayed using line-based visualization approaches (Lee et al., 2020). Another popular visual form is a two-way technique used to construct density maps. Smoothed trajectories are first aggregated into a density field, which is then displayed (Scheepens et al., 2011). The Space-Time-Cube (STC) approach has been extensively investigated for data containing spatio-temporal attributes. The STC provides excellent visual options for studying the relationship between time and space as well as other variables, (Scheepens et al., 2011). However, the majority of platforms now have abstract visualizations that serve as a digital system for traffic;

but even so, the most of representations rely on static data. Studies have revealed a potential strategy for providing an interactive list of regions from which users can make decisions.

2.4.1 Web-based Visualization

The Transportation Injury Mapping System (TIMS) is the most well-known web-based traffic safety tool. It was created by the SafeTREC Research Center at the University of California, Berkeley. The application permits users to access the Statewide Integrated Traffic Records System (SWITRS), which contains crash data for California (Brozen et al., 2021). The tool offers numerous mapping and data-gathering features. Additionally, the TIMS offers collision diagrams and hotspot analysis, safety performance management, and additional GIS tools for various accident types. In addition, users can assist themselves in date range selection choices to receive and manually modify the data. Visual analytics solutions have become highly popular for data analysis and extracting insights from the data to achieve this goal and reduce the impact of crashes. (Zhu et al., 2021) have also introduced a traffic safety tool named Safety Analysis Visualization and Evaluation Tool (SAVE-T), which has the potential to be useful for quickly visualizing past crash data and identifying the circumstances that contributed to the accident. SAVE-T was hosted on a web-based visualization and analytics platform built to operate with New Jersey's crash database. This user-friendly solution boosts engineers' productivity by automating crash and other traffic data analysis and reporting. Many capabilities are added to the tool based on user demands, making it easy to access, save, manage, visualize, and update crash and other relevant traffic data.

2.4.2 Independent software applications Visualization

Numerous safety agencies and researchers have developed computational methods for analyzing road safety data. Several tools are independent software applications, whereas others rely on web-

based mapping APIs to meet their national, state, and local GIS requirements. The examination of highway design using the Highway Geometric Design Consistency Evaluation Software was among the first studies. This menu-driven application evaluates the construction of roadways using preliminary models. AASHTO's Safety Analyst is the most well-known and up-to-date set of tools for traffic safety analysis. A highway safety analyst employs analytical processes to identify and manage cost-effective solutions for enhancing highway safety (Claramunt et al., 2000). The tool automates the six primary elements of the highway safety management process: traffic diagnosis, roadway network screening, countermeasure selection and evaluation, priority ranking, and economic analysis. Several types of research have been conducted on traffic safety visualization, which helps limit the number of incidences on the roads. (Bachechi et al., 2022) propose a monitoring and simulation system for individual vehicle behavior in traffic applications. The simulation output of the traffic model includes data on roadway speeds and traffic flow. Within the Trafair database, data is collected and analyzed to determine urban traffic's temporal and spatial evolution. The speed model index view analysis allows users to visualize the Speed Index (SI) profile (Bachechi et al., 2022). The SI is calculated by dividing the average speed per hour by the speed limit for the road segment to determine if congestion exists.

2.5 Data Fix Tool

Data fixing is an extremely important stage of making sure the quality of a dataset is a hundred percent accurate for the needs of any application. The purpose of Data fixing is to detect whether the dataset contains certain classes of errors and be able to repair them using different algorithms. The usual errors include outliers, duplicates, rule violations, and pattern violations. (Csail et al., 2016) ITK-SNAP is a software that allows users to annotate 3D medical images, sketch anatomical

sections manually, and do image segmentation automatically. The semi-automatic segmentation of multi-modality imaging datasets using information from all accessible modalities simultaneously is supported by this interactive image visualization and segmentation tool. Manual segmentation, image navigation, and automatic segmentation are the software's key features (Yushkevich et al., 2019). LabelIMG is a free and open-source graphical image annotation program that uses Python and QT as its graphical interface. It was created as a demonstration of machine learning training. However, LabelImg only provides a rudimentary image annotation function that does not handle image stream annotation, and there is no auxiliary annotate at all, so sample annotation still takes a long time and costs a lot of money (Hsiao et al., 2019) (Yu et al., 2019). VOTT is a program developed by Microsoft's Commercial Software Engineering (CSE) department for graphical images and video. It solely uses bounding boxes to annotate objects in an image (Aljabri et al., 2022). Supervisory is a web-based program that draws in a completely manual or semi-automatic manner by selecting the necessary region to generate the marking and automatically generating the appropriate form. It was developed by Deep Systems for computer vision development. It can be annotated on a pixel level or using vector graphics(Aljabri et al., 2022).

CHAPTER 3 ARCIS SYSTEM

Effectively extracting the most valuable data from the huge quantity of videos not only ensure the quality of traffic data, but also helps researchers and practitioners save time and cost. The majority of research institutions have investigated the processing of camera footage or CCTV footage for application in traffic surveillance systems. However, the potential of these videos to convey traffic congestion, near-miss incidents, and traffic statistics is limited. Therefore, the analysis of the Automated Road Conflict Identification System (ARCIS) has focused mostly on Unmanned Aerial Vehicle (UAV) data. The research by Wu et al. (2021) on ARCIS has made significant contributions to the discovery of road safety enhancements by utilizing learning algorithms to process UAV videos. The system's primary purpose is to analyze flow conditions at a signalized intersection by processing videos and performing a full analysis of traffic trajectories via automatic detection of moving vehicles and extraction of relevant surrogate safety parameters. The system has provided a systematic and efficient video processing methodology for extracting traffic parameter information from UAVs with accuracy and reliability. The methodology divides the entire procedure into five stages: stabilization, object filtering, video stitching, GPS to pixel converter, and data extraction. Multiple technologies were employed by the system to optimize the processing and analysis of UAV-related traffic video. The majority of the implementation was built with a popular Python image processing package and OpenCV library.

3.1. Stabilization

The processing of traffic flow video obtained with a UAV will inevitably shift and rotate due to the direction of the wind and vibrations influencing circumstances. Captured video will have a translation and rotation problem between frames. This instability can impact the movement of

target objects, resulting in inaccurate extraction vehicles trajectory. As a result, numerous approaches utilizing various software and methods to lessen the effects on camera movements have been introduced to eliminate unintended motion and jitter without affecting the background or the movement of objects. The system employs a stability landmarked with human interaction approach to handle the instability and shakiness of UAV videos. This method ensures stability by tracking the coordinates of stable object locations chosen by humans from start frame, where the object's point remains stationary throughout the video recording.

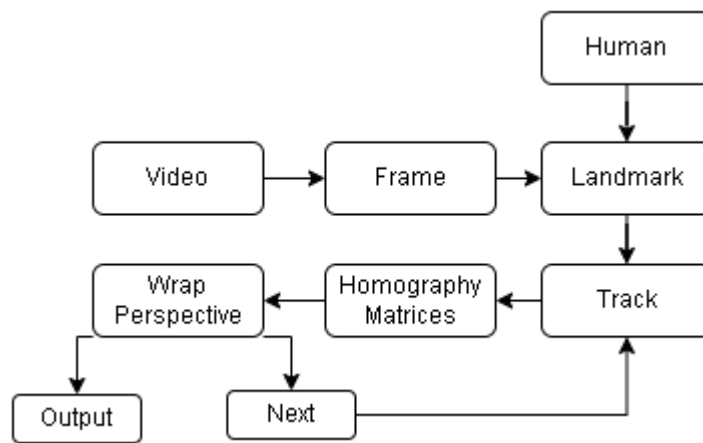


Figure 1 A flow diagram for structure of stabilization

A logical structure is illustrated as Figure 1, with the method using only the first frame of the video to initialize the stable land-marker, which is defined as the object's bounding box. The tracker objects in the algorithm must be initialized with at least four coordinates. The tracker is created by the region of interest of each land-marker. Finally, iteratively update the tracker to obtain a location of the selected landmark in the following frame. This method significantly reduced motion in the final result by tracking various landmarks within the frame and generating a new video without unwanted motion.

3.2. Filtering Object

ARCIS uses object filtering method to increase detection accuracy by removing objects in video that may cause a detection error. This method will remove the visual information inside the zone by using image inpainting techniques (INPAINT-TELEA). The zone is designated using a binary mask and filled with a neighborhood pixel color to inpaint. Image inpainting techniques is a form of image restoration and conservation that can effectively repair removed image regions by propagating information preserved in the surrounding regions. The background subtraction method (Figure 2) has been applied to remove an object with a moving object on top of it. This method can apply a mask to a moving item by subtracting it from its background. The rest frame will use inpaint technologies to produce a new frame by filling in the blanks. Finally, the unwanted object will be removed by reapplying the object's mask to the new frame. By applying these approaches to object filtering, the removed object region will intelligently recreate itself to match the surrounding areas' colors and textures.

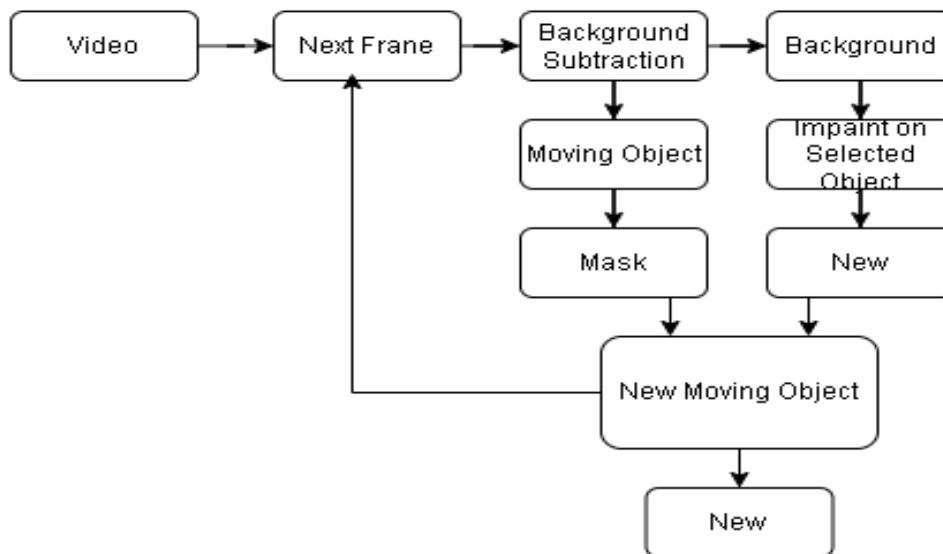


Figure 2 Filtering Object method

3.3. Stitching Videos

Most cities have a 400 feet UAV fly height limit, one UAV can cover about 120 meters of road under this limitation. Video stitching method enables the ability to merge multiple drone videos which can provide the videos to have a wider field of view. The method is preceded by matching the overlapped key point and must be selected as a pair that can match between two images (Figure 3) to transform an image from one perspective to match another.

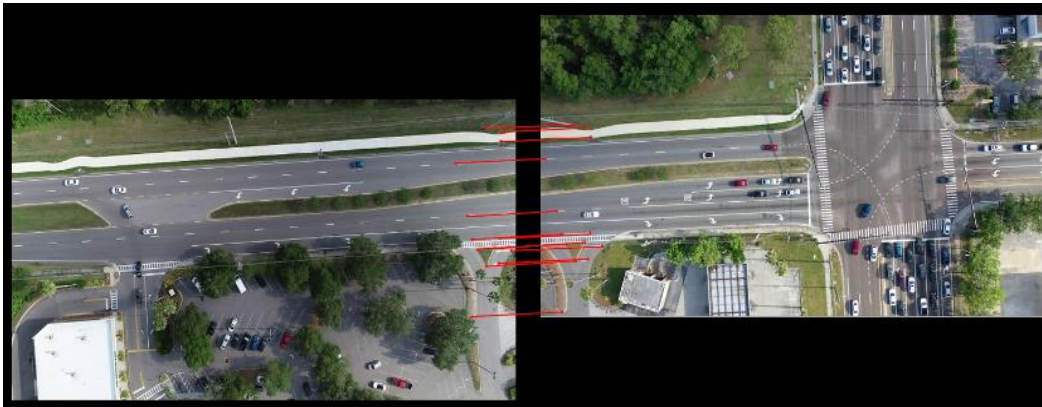


Figure 3 An example of matching key points for stitching

Multiplying the coordinate with the Homograph matrix using a uniform coordinate system is equivalent to transforming an image. Therefore, if there are four pairs of key points that correspond to both images, a homograph transformation matrix can be used to calculate the value of each element. The Figure 4 is the desired result obtained after transformation by using the perspective transform matrix.

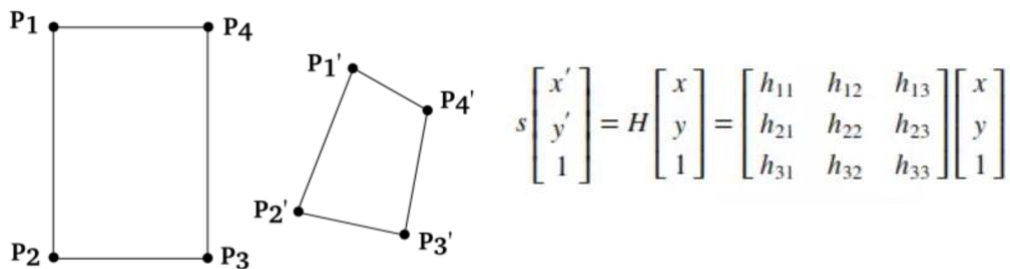


Figure 4 An example of transformation using perspective transform matrix

Therefore, after images have been transformed, a new frame will be generated for stitching two images together with the same image height and the combined image width. In order to accomplish the goal of stitching together videos, each frame from two different videos is applied to this procedure in order to generate a new panoramic video file.

3.4. GPS to Pixel Converter

The converter is allowed image pixel coordinates transfer to GPS coordinates (latitude and longitude). This function is applied by ARCIS to calculate traffic speed, count and statistics; it also has the capability of representing the trajectory of a vehicle in the form of a geographic map rather than an image. The method is based on used perspective transformation between an image and a geographic perspective map by using homograph transformation. A homograph matrix is the matrix to convert the coordinates from image plane to world plane. As described in the video stitching approach, at least four correspondence points must be used to calculate a homograph matrix. In converter method which needs both four-pixel coordinates points and corresponding four physical GPS coordinates. To generate the homograph matrix:

1. Calculating distance between each GPS coordinates
2. Calculating horizontal angle between the direction of a GPS point and another point
3. Localize and output each GPS point.

The homograph matrix will be generated by using output GPS coordinates and pixel coordinates. The converter calculates the angle and distance from each pixel coordinate in a geographic map using the inverse matrix. The converted pixel points will be displayed as in a geographic map shown as Figure 5.



Figure 5 An example of converted pixel points for drone view and geographic map

3.5 Data Extraction

In ARCIS, a data extraction method is employed for vehicle detection, which pinpoints the exact location of vehicles in UAV video. This method has used the Mask Region Convolution Neural Network (Mask R-CNN) algorithm to detect vehicles by providing precise masks for object detection. Rotated bounding rectangles can be obtained from the masks, which provides an alternative method to obtain vehicle sizes and more precise locations. Detected vehicles are tracked by Spatial Reliability Tracking (CSRT) multi-object tracking algorithm. This algorithm also enables the identification of missing vehicles by comparing Intersect of Union (IOU) in stopped vehicles. Figure 6 illustrates an example of the detected vehicle's bounding rectangle box with its unique id for tracking the vehicle in different timestamps.



Figure 6 An example of detected vehicle with unique ID and bounding box

After Mask R-CNN has been applied to each frame of video, an occupancy table for vehicles will be generated. The table stores information about each vehicle in each frame, including its unique id, size in pixels, center pixel points, and bounding box coordinates. By comparing the timestamps of two subsequent vehicles at each pixel, the method may also generate surrogate safety measures such time-to-time collisions (TTC) and post-encroachment time (PET). In addition, conflict types such as rear-end, head-on, and sideswipe might be determined by the process of comparing PET value for traffic safety diagnosis.

CHAPTER 4 ADDITIONAL SOFTWARE DEVELOPMENT

The fundamental structure of the ARCIS system consists of the video processing and data visualizing for traffic safety diagnosis. The backend code for all the components in the Data Processing Tool has been fully implemented in Python based on previous work. Integration of all components into a user-friendly interface is one of the tasks involved in software development. The remaining two tasks focus on the design and development of the Data Fix Tool, which involves an active learning strategy, and the implementation of a visualization tool to illustrate extracted data. The software is defined by its workflow plan (see Figure 7) in terms of its functions and component interactions. The three major components of this software are the Data Processing Tool, Data Fix Tool, and Data Visualization. For the first step, the software will take a raw video as input, which will consist of pre-processing the input video and extracting data from the fixed video. After that, the extracted data will be transmitted to the Data Fix tool for error correction and then convert pixel coordinates to GPS coordinates. Additionally, an active learning strategy is intended to assist humans in labeling and correcting erroneous data in a more efficient manner. The final procedure involves the processing of surrogate safety measurements (TTC/PET/conflict type) and traffic parameter data for visualization.

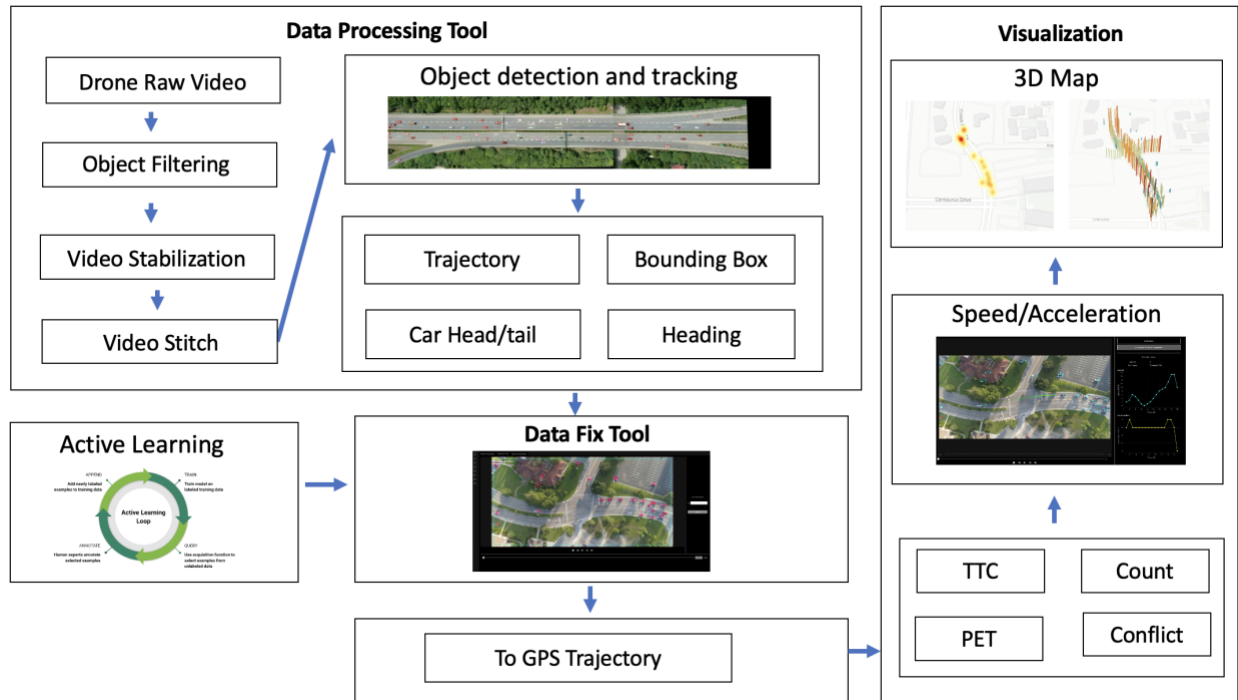


Figure 7 workflow diagram

To ensure that the software can be used easily and effectively by the intended users, the interface must be simple to use and understand without requiring extensive explanations or instructions. Therefore, the software's design will prioritize simplicity to make it as easy as possible for users to operate. User interfaces are implemented via classes, which can then be merged to create UI applications. PyQt5 framework is used to build this GUI because it is compatible with multiple operating systems, such as MacOS, Windows, and Linux, and enables rapid interface development from a simpler base UI component.

4.1 Data Processing Tool

The combination of multiple functionalities and a user-friendly and easy-to-use design were taken into consideration during the development of the Data Processing Tool's user interface. The interface design of the Data Processing Tool has Tabs to organize and allow navigation between

groups of different functionalities. The main components are grouped by the tab in this tool which are the following five components: Filter Object, Stabilization, Stitching, Data Extracting and Converter (Figure 8). It is essential that the user must process videos in the specified order. This tool is considered a pre-processing module that the user must complete prior to extracting and processing data in the subsequent stage.

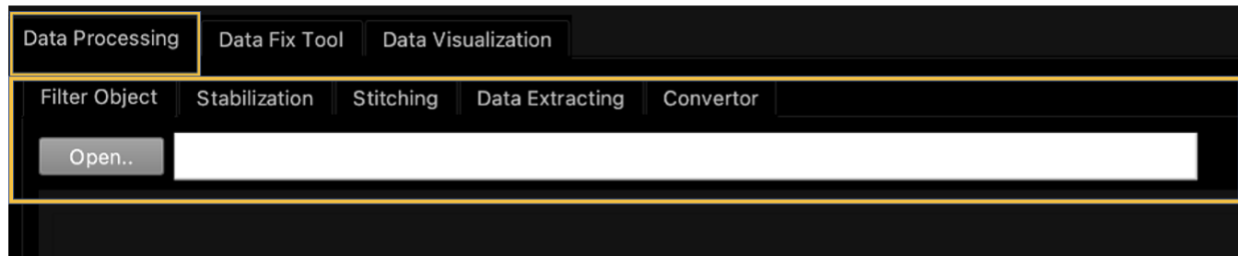


Figure 8 Data processing tool components

4.1.1 Filtering Object

The interface for the object filtering feature allows users to remove multiple unwanted objects such as people, poles, and trees with a few simple steps. This feature employs a method that automatically removes a selected area and instantly analyzes each video frame to generate a new output video file. The operation of the main window including image viewer and control panel on the right side for users to have the option to edit the mask region and save file.

The process begins by loading the video using the 'Open' button; after the first frame has been visualized, the file name will be displayed. The method to remove an object by inpainting the area being surrounded by a region. To create a region mask around the object to be removed, the user needs to simply click around the object to draw a point, and then a polygon is generated based on the points (Figure 9). When the object has been removed from the frame the feature will automatically save current frames as a new file. The graphical user interface (GUI) also has built in a different functionally buttons:

- a. **Add** allow user to draw multiple masks around the object to remove
- b. **Redo** is designed for delete a point
- c. **Delete** to delete a mask
- d. **Start Loading** to remove all masked object to be removed in each frame of the video
- e. **Save/Restart** allow user the stop the program and reset the entire application

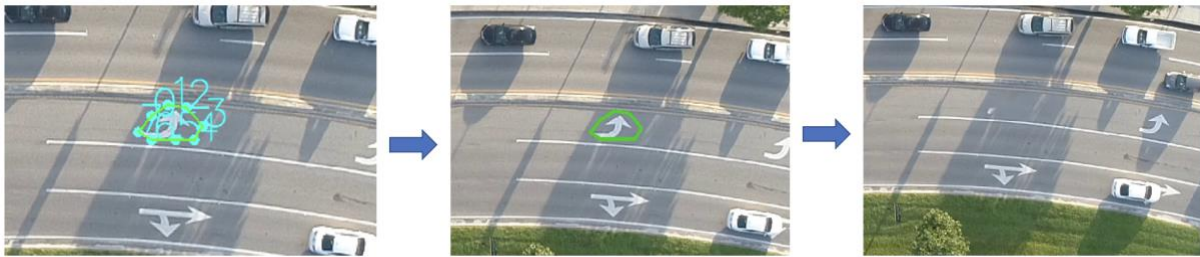


Figure 9 Filtering object example

4.1.2 Stabilization

The stabilization module stabilizes and removes unwanted motion and shaking from UAV video. For stabilization to be processed, at least four steady object coordinates are required throughout the video. The user interface consists of three components (Figure 10): (A) user input panel, (B) view panel and (C) control panel. This component allowed the user to input a raw video and draw or delete regions on the object in the frame. Once the user selects the objects in the video, the program will display the bounding box around each object and store the object's coordinates in the backend. The algorithm used to match coordinates and track the same position from frame to frame. When each frame is displayed in the view panel, the stabled frame is simultaneously saved as a new video file.

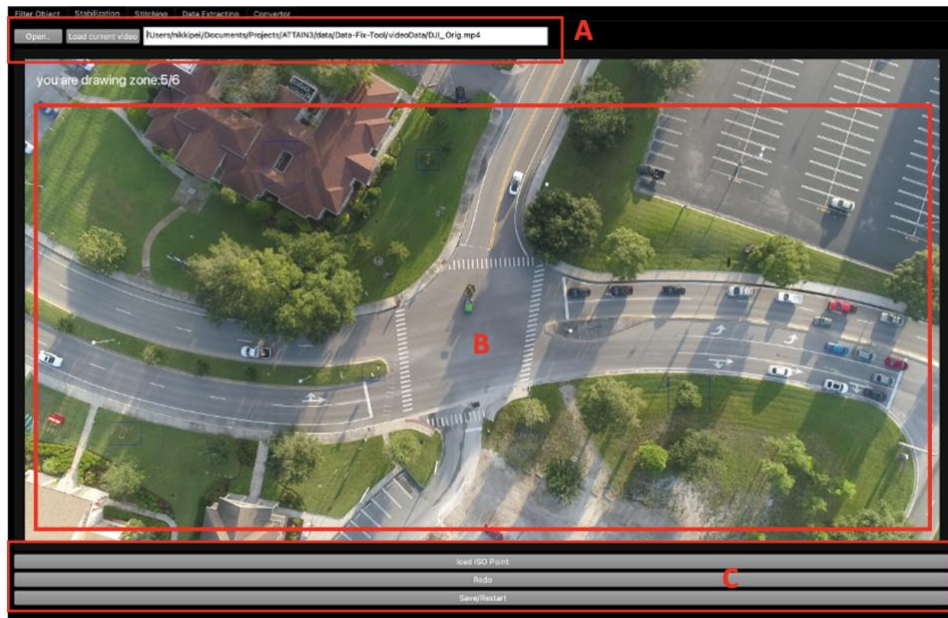


Figure 10 Stabilization user-interface components

To initiate the stabilization process on this program, (A) allows the user to either open a new video file from the local browser by clicking "Open" or load existing videos from the previous function. The GUI will load and extract in order to display the first video frame. In (B), once the image has been successfully loaded for viewing, this view panel has an interactive element based on the mouse event handler. If the user has identified the objects need to be selected, user can start draw a bounding box around the target object:

1. Start by pressing the left mouse button at top-left corner of the object
2. Drag the mouse until the colored area has fully covered
3. Release the mouse
4. Pressing right mouse or "Load ISO Point" button

This approach result will display, as Figure 11, a colored bounding box surrounding the displayed target object. This procedure must be repeated six times in order to obtain the desired level of stability. In addition, the user always has the option to redraw the areas by choosing a new place

on the view panel or to remove the drawn regions by pressing "Redo." When all bounding boxes for each object have been drawn, the module will automatically store the data and execute the program which will process the video from the first frame to the last frame, and the output data will be saved to a file that will be used in the subsequent step process and may be accessed by the user for further analysis. By choosing the "Save/Restart" button in (C)'s control panel, the user has the ability to stop the running program and restart it from the beginning.

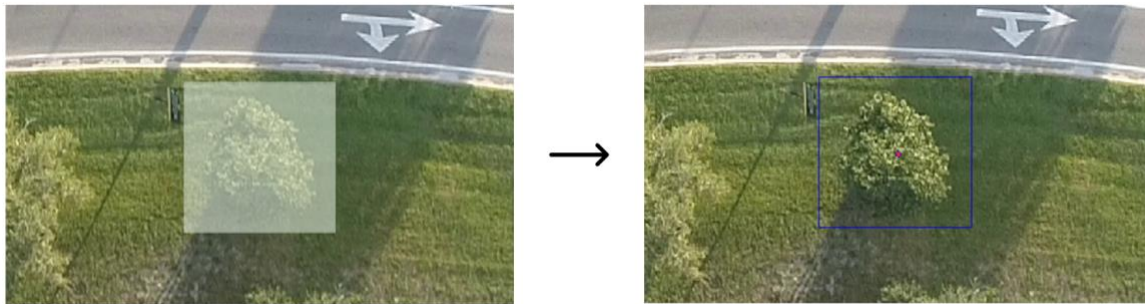


Figure 11 Colored bounding box surrounding the displayed target object

A video stabilization technique based on the point tracking feature, as shown in Figure 12, produced remarkable results in terms of stabilizing high jittery videos that were distorted.

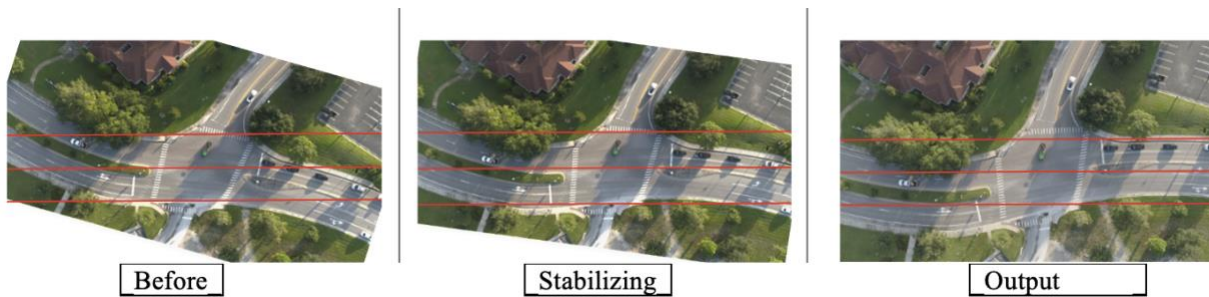


Figure 12 Stabilization result

4.1.3 Stitching Videos

Video stitching refers to the process of combining multiple videos from different drones with overlapping regions, to produce seamless panoramas. The interface was created to provide a user-friendly working environment, enabling the users to follow the steps needed in the stitching

process. Figure 13 illustrates the GUI of this feature. This functionality allows users to stitch together two or three videos and process each frame of the videos. The component in (A) allows the user to either open a new video or load from the previous stage. In order to obtain the optimum match point between two overlapping regions, the GUI requires an additional reference base map image. Two videos will be displayed first to assure the user about the accuracy of the selected videos for further steps. In addition, a match histogram technique (B) is provided to help users improve the contrast of the videos.

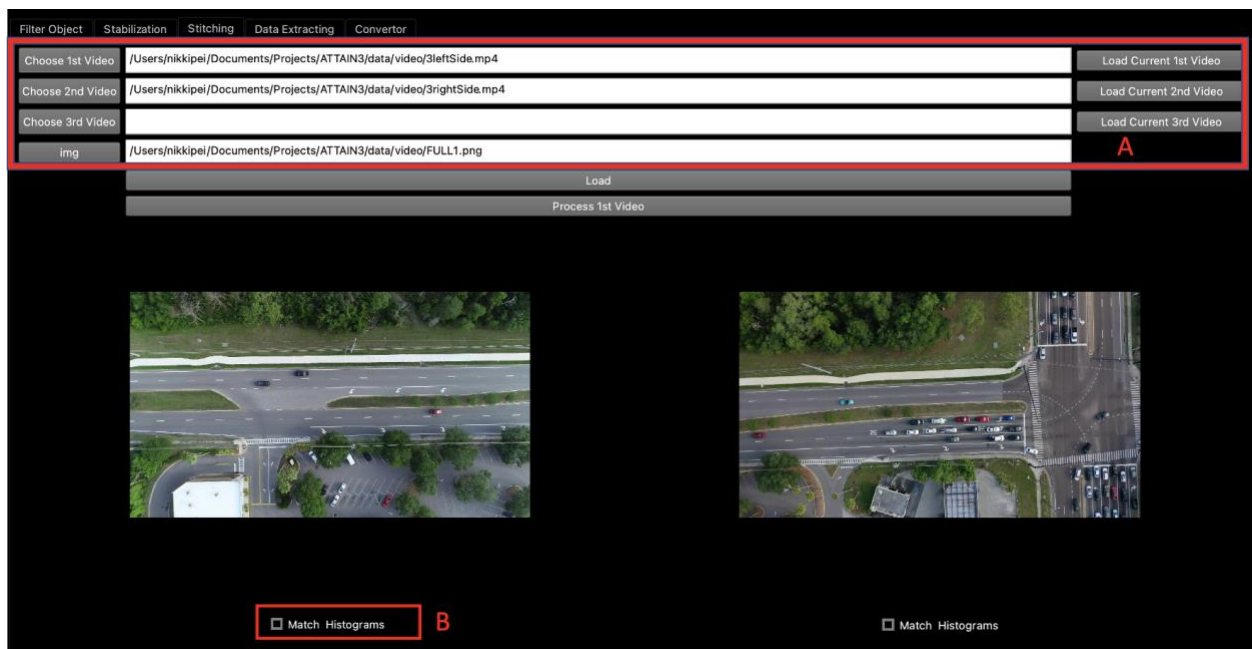


Figure 13 Graphic user interface for stitching video

The "Process 1st Video " button will instruct the user on the next step on drawing the corresponding points between the 1st video and the base map image. As stated previously, the image viewer panel contains a built-in mouse click event; by clicking the left mouse button, a point will be displayed. The most essential factor of image stitching is precisely matching the points from both images. Therefore, a zoom-in/zoom-out button has been made available for the user to specify where the appropriate points are. If the width of the columns exceeds the available display space after

zooming in, horizontal and vertical scroll bars will appear on the viewer panel's sides. A small arrow has been shown in the bottom as Figure 14, indicating an undo button, which a function performs to reverse the action of an earlier action. This option allows the user to remove points that have already been made. Figure 15 is the result of an image transformation after all points have been created. The user always has the option to return to the previous step by clicking the "Back" button in order to make adjustments for a better outcome. This procedure will be repeated for the second video.

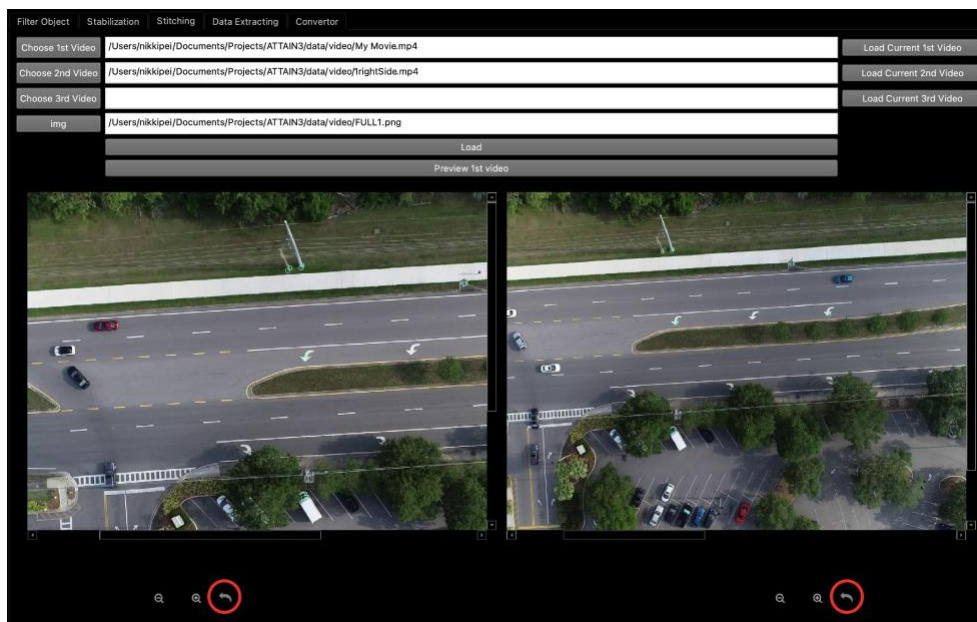


Figure 14 Undo button to reverse stitching action

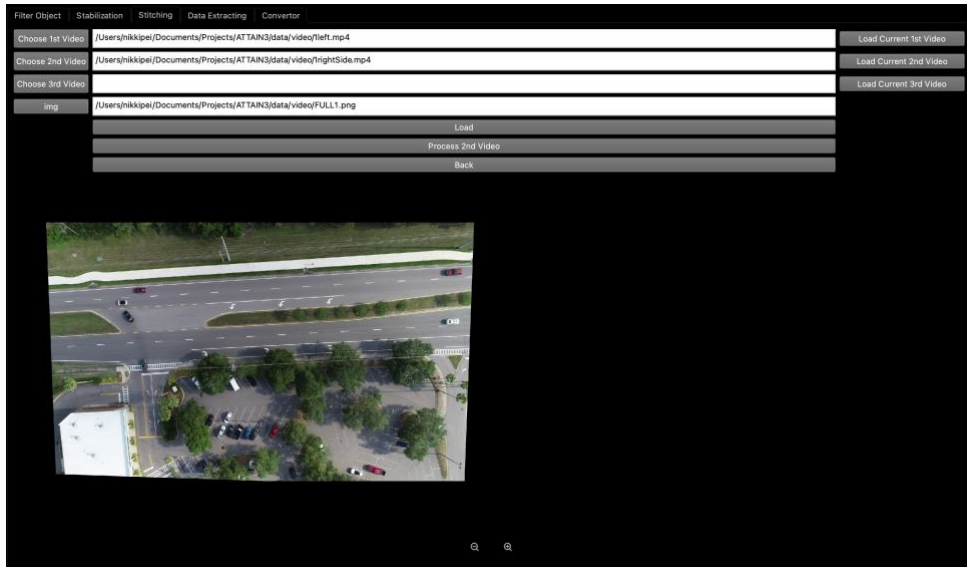


Figure 15 Result of an image transformation

On the last step (Figure 16), which combines the two output images, the user can modify the current frame by clicking the next/previous button (A). The GUI provides a bar (B) for adjusting the distance and area offset between images that overlap. These buttons in (C) allow the user to return to the previous matching point stage again and make more adjustments as necessary for a better output. Figure 16 also displays the before and after results after adjustment, and the user can just click the "Start Video" button to automatically stitch each frame from the two videos together.



Figure 16 Before and after results after adjustment

4.1.4 Data Extraction

Data Extraction enables the user to input a video for vehicle detection and tracking, as well as save the data in a separate table file for further processing. The table file stores every vehicle detail coordinate in each frame. The detection will occur within the inside of area where user needs to click and draw the polyline to generate it. After the user has loaded the video and clicked "start," the program will automatically begin detection and tracking. The "stop" button is intended to completely terminate the program (Figure 17).

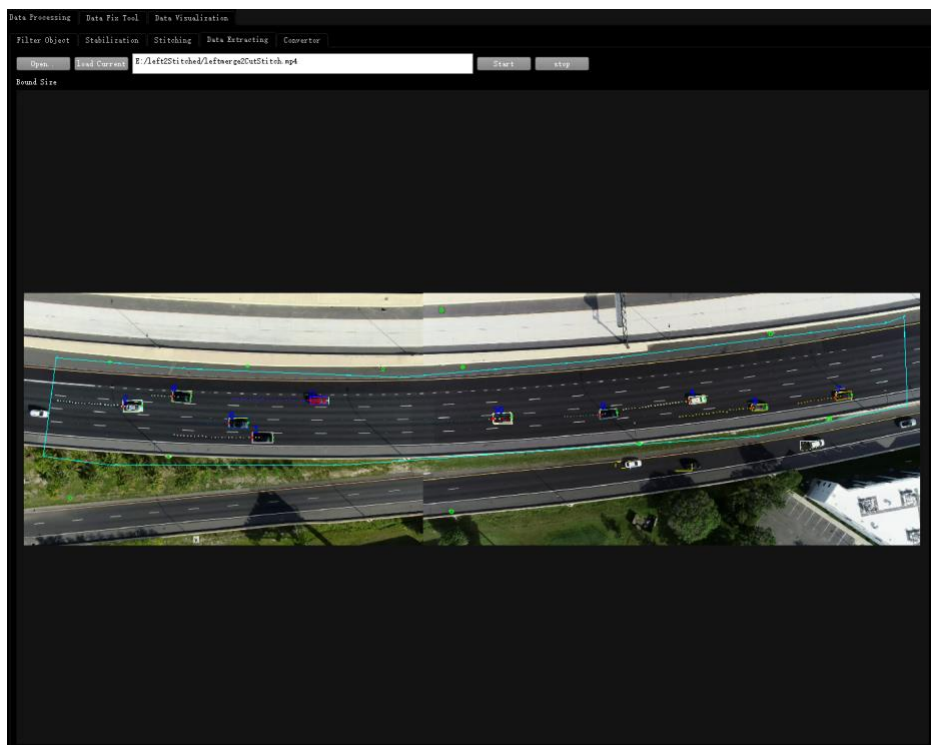


Figure 17 Main GUI of data extraction

4.1.5 GPS to Pixel Converter

This process uses pixel coordinates to GPS coordinates by using user-entered input values. In order to convert all the vehicles in the video, a table containing the pixel coordinates of each vehicle is required, which the user may easily access from the previous data extractor function.

The main window is shown as Figure 18, there are three components designed for user to convert include:

1. **View Panel**, allow user to input correspond GPS coordinates
2. **Edit Panel**, for edit/delete/save input values
3. **Result Panel**, output for converted result

The GUI enables the user to input all GPS coordinates in the view panel by double-clicking on the intersection of the image, a dialog box appears for entry. The input value will be displayed in the edit panel, allowing the user to modify or remove all GPS coordinates. To change pixel coordinates, simply drag the point to a different location. The result panel contains three buttons with distinct functions: "Load/Output" for loading the results of the converter, "Verify on Map" for verifying the output value on a map, and "Convert CSV" to convert all the data to GPS coordinates and append the value to the file.

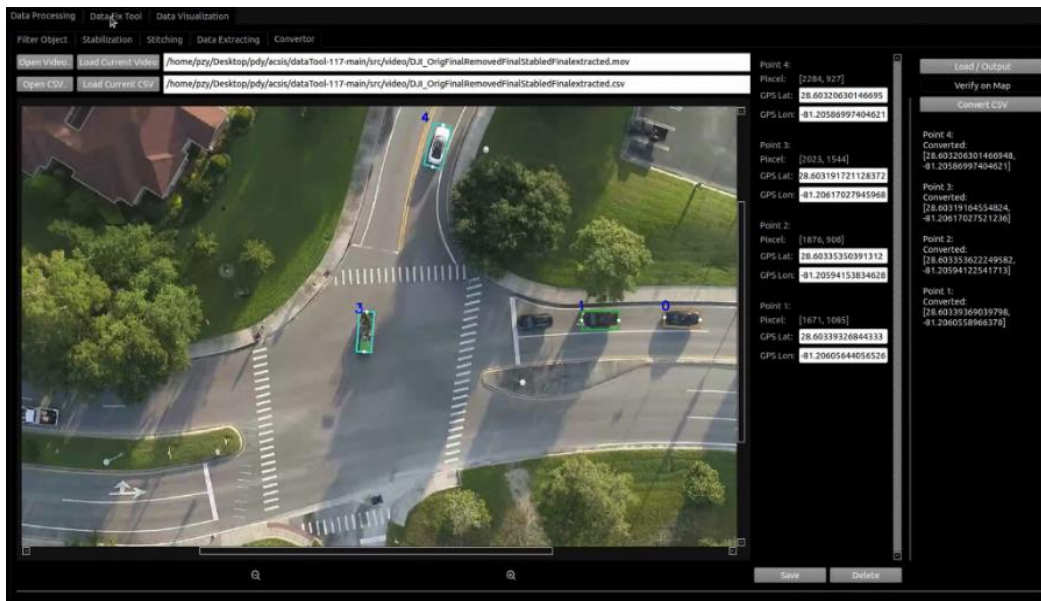


Figure 18 Main GUI of pixel to GPS convertor

4.2 Data Fix Tool

Data Fix Tool is specifically designed to improve ARCIS accuracy, it allows to fix with an error bounding box around vehicle objects, it handles directly with the table file of detected vehicle data. To improve the performance for modifying boxes throughout a sequence smoothly, this tool has been implemented with a variety of capabilities so that it would allow the practice of different kinds of functions such as fixing, labeling, shifting, and rotating the bounding box. Secondly, the system aims to make the fixing process as seamless and user-friendly as possible. The GUI operations are separated into their own distinct areas, making it simple for users to navigate. Figure 19 exhibits the graphical interface of the tool and Figure 20 shows all functionalities of the tool.

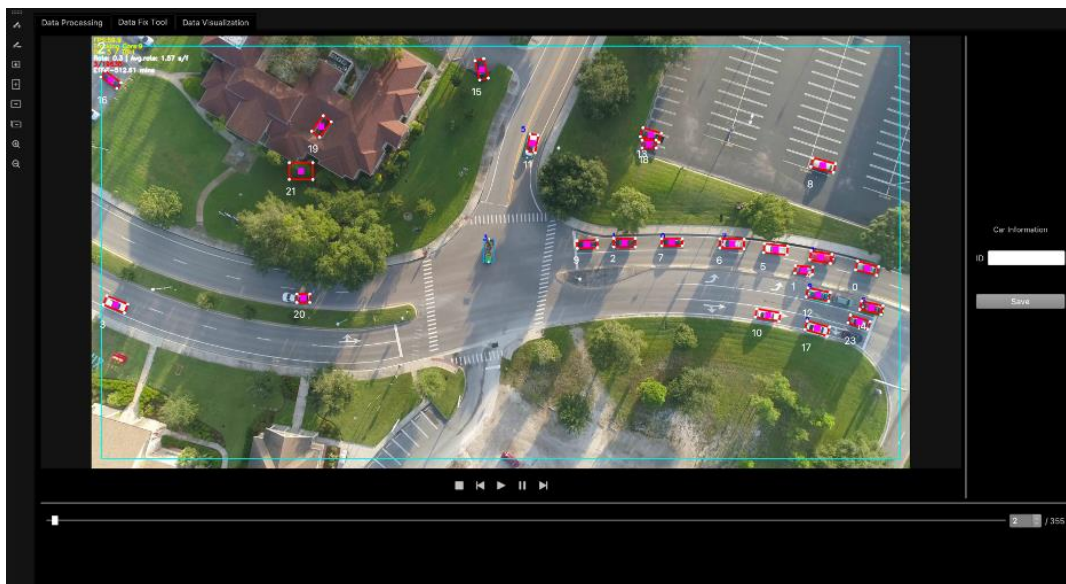


Figure 19 GUI of data fix tool

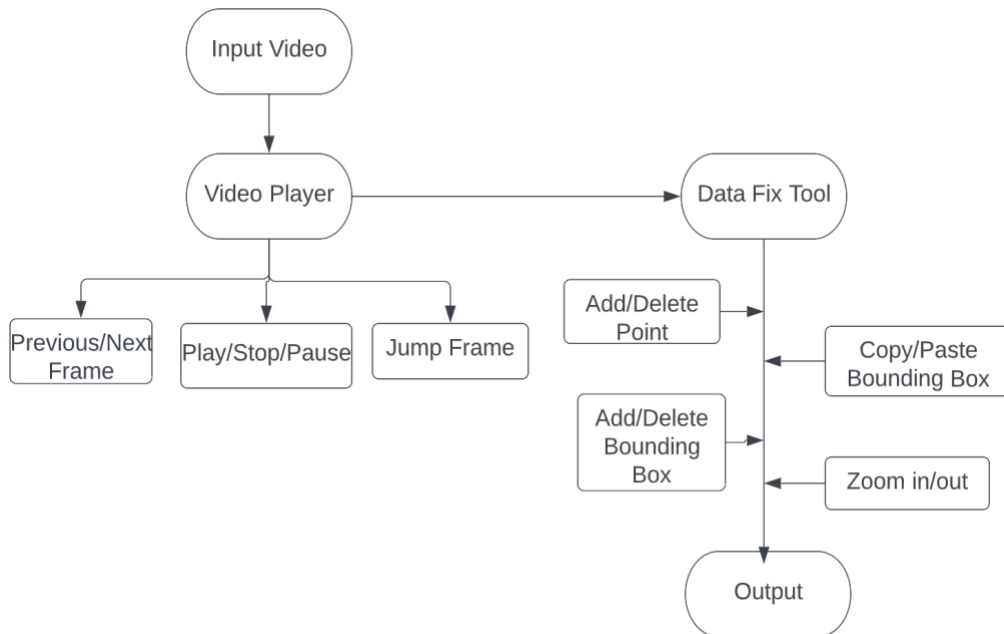


Figure 20 Data fix tool functionalities chart

4.2.1 Start stage

To load the video and a table file for this Tool is similar to GPS to Pixel Converter by either opening a new file or can be loaded from the backend. Fixing data in a video is equivalent to fixing in a series of frames. The view panel is designed to load the first frame and the data from the table file that corresponds to the bounding box of each vehicle. Each bounding box contains four white color spots connected by a red line, as well as a vehicle identification number shown on its side.

4.2.2 Video player

Controls for the video player give the user availability to navigate the video to different frames to fix the bounding box. The tool provides seven different functionalities buttons, each with a simple recognizable icon that allow user easy to access as follows (Figure 21):

- a. Play all frames in video
- b. Pause video to display current frame

- c. Stop video back to the first frame
- f. Next frame
- e. Previous frame
- f. Drag the scroll bar jump to different frame
- g. Enter frame number in the slot to certain frame



Figure 21 Video player control

4.2.3 Toolbox

The Data Fix Tool provides the functionality of fixing and labeling a bounding box by clicking the white point and it will get highlighted. Delete or add a point to the bounding box. To change the location of the highlighted point, the user needs to click a different location. The toolbox on the left side of the main window allows the user to choose from a different type of functions in order to precisely adjust the bounding box. In addition, the tool provides easy access to all functionalities using the keyboard. The user can delete the bounding box in the event of an error detection or add a missing bounding box for the vehicle by using the arrow keys on the keyboard. It may be difficult for a user to label lost vehicle tracking frames individually. As a solution, a copy-and-paste ability has been added to the tool, which allows the user to copy the first frame of a vehicle's bounding box and then paste it into the last frame before the vehicle disappears. Each vehicle has its unique ID throughout the video, while adjusting on deleting or adding a new bounding box, this may have the potential to change the vehicle's id. A slot of box has been created for modifying the vehicle id by inputting a new id; this will change the selected vehicle's id in all frames.

4.2.4 Output

Figure 22 demonstrates an example of output before and after fixing the detected vehicle bounding box. This Data Fix Tool incorporates a variety of functions to enhance data quality and productivity.

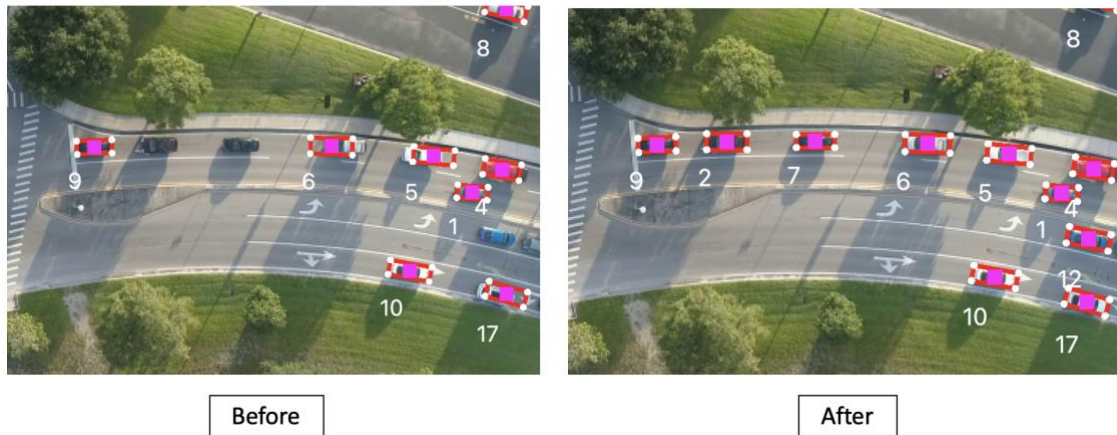


Figure 22 An example of output before fixing the detected vehicle bounding box

4.3 Active Learning

However, fixing, and labeling data is usually a time-consuming and error-prone process. The results of an experiment indicate that a user must spend at least six hours correcting 1,000 frames of data. Due to this inefficiency, the following section introduces a human-in-the-loop (HITL) on active learning method to increase detection data accuracy and reduce human effort. Human-In-The-Loop often refers to a system in which an active learning machine can include selected human inputs or labels into its training model process, creating a 'loop' that the model continuously learns and improves its capabilities. The classification and labeling of a big dataset is a monumental task for object detection. Even if a most lightweight detection model will need hundreds of thousand human labeled training data, the effort and cost are still substantial. Therefore, to reduce the training dataset labeling cost, in the field of machine learning, Active learning can proactively identify subsets of training data and continuously deliver labeling requests for filtered data to

humans. In this study, compared two approaches use Pool Based Sampling and Stream-Based Selective Sampling strategies. The purpose of these two strategies is to increase Average Precision (AP) values based on an IoU (Intersection over Union) threshold while detecting objects in a train dataset. As described in Figure 23, IoU is used to determine if a detection is valid or not by ranges from 0 to 1, with 0 indicating no overlap and 1 meaning complete overlap between ground-truth and predicted box. In this instance, the MaskRCNN model is used to build vehicle detection, under each vehicle's object detected, MaskRCNN always produces the estimation of its class, size, position, and bounding box that completely encompasses an object.

	MAP	AP50	AP75	APmedium
$\text{IoU} = \frac{\text{area of overlap}}{\text{area of union}}$	Mean value of AP50 and AP75	AP at $\text{IoU} = .50$	AP at $\text{IoU} = .75$	AP for medium objects $32^2 < \text{area} < 96^2$

Figure 23 AP value description based on IoU

4.3.1 Data preparation

Accurately labeling a small set of data is necessary for active learning, which requires specific information to identify the dataset. To validate the proposed active learning framework and get better performance on detect vehicle in accuracy, data collection was conducted at different intersection:

- On Feb 7th, 2022, from 11:30 AM to 12:50 AM at a typical 4-leg intersection of SR 436 and Oxford Road. A DJI Phantom 4 UAV was utilized to collect the data, and the video was captured by an optical camera with 1920 × 1080 resolution.

- On May 17th, 2022, from 17:30 PM to 17:50 PM at a typical 4-leg intersection at the University of Central Florida (UCF). A DJI Phantom 4 UAV was utilized to collect the data, and the video was captured by an optical camera with 1920×1080 resolution.

A total of 10,000 frames containing the most information have been extracted from both. A sixty percent of all frames are assigned to the train set, thirty percent to the validation set, and the remaining ten percent to the test sets. While the train set and validation set are partitioned randomly at the beginning of each experiment, the test set remains consistent throughout all reported experiments. The validation set in both the Pool-Based Sampling and Stream-Based Selective Sampling strategies requires a JSON file containing human-perfectly labeled frames and includes diverse information on each vehicle's pixel coordinates and identified title of frames; this will play a significant role in evaluating the result file.

4.3.2 Pool Based Sampling

The pool-based sampling consists of small sets of labeled and extensive unlabeled data. In this method, the modular design selected allowed the difference of parameters that surrounds the active learning process. This experiment is using detectron2 as a segmentation model with a R-CNN backbone. A dataset consisting of a JSON file and a random selection of 2,000 frames that have been perfectly labeled by humans has been used to train an initial segmentation model. A query strategy $Q(M, V)$ will be implemented at this step (Figure 24) to improve performance by applying the current model and a new video as input. Using the prediction probabilities from each trained frame, the query function will provide an informativeness score where the average confidence score is more than 0.90. This enables more accurate vehicle detection by continuously transmitting the most useful data to the model. After the function evaluated the performance based on the

model's prediction, 200 frames with the highest information score were sent to the unlabeled dataset. In this stage, a new collection of 500 perfectly labeled frames from an unlabeled dataset is randomly selected and sent to the model for retraining. This active learning cycle will be repeated three times, and each cycle will result in an IoU-based AP value.

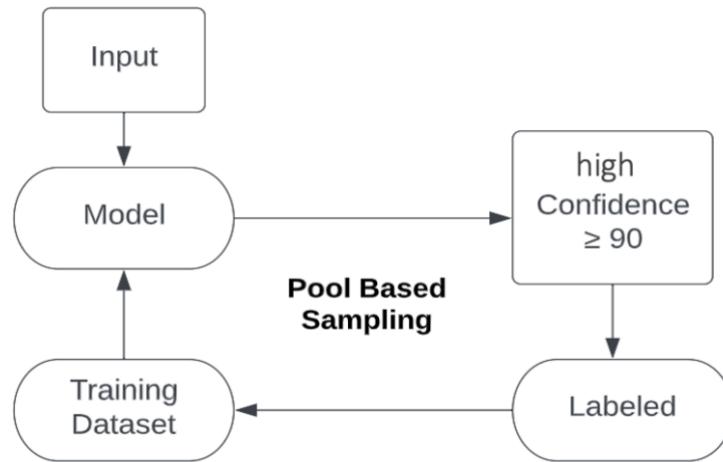


Figure 24 Pool based sampling flow chart

4.3.3 Stream-Based Selective Sampling

This method utilizes the collection of images that have been accurately labeled by humans as ground truth data (Figure 25). Using similar techniques and steps for model training in pool-based sampling the stream-based selective sampling method will result in a video that contains a .csv file including all the information on each vehicle for each frame. This data will comprise the frame number, coordinates of each vehicle object's bounding box, and its unique identifier. A human will be required to manually fix the labeled bounding box that has been detected on the video, which can be done with the Data Fix Tool. A labeled dataset of 2,000 frames was used to train an initial segmentation model. After applying a new input drone video, the next step will be the removal of errors using a filtered query function. In this function, the query performs as a decision maker, determining whether the frame will be transmitted to a human to label or not. This decision

will be made by first ranking the lowest expected confidence probabilities, then selecting frames with low confidence ≤ 50 or no detection and sending them to the Data Fix Tool. To improve the overall detection performance of the model, fixed frames and data will be returned to the folder that contains the original training dataset.

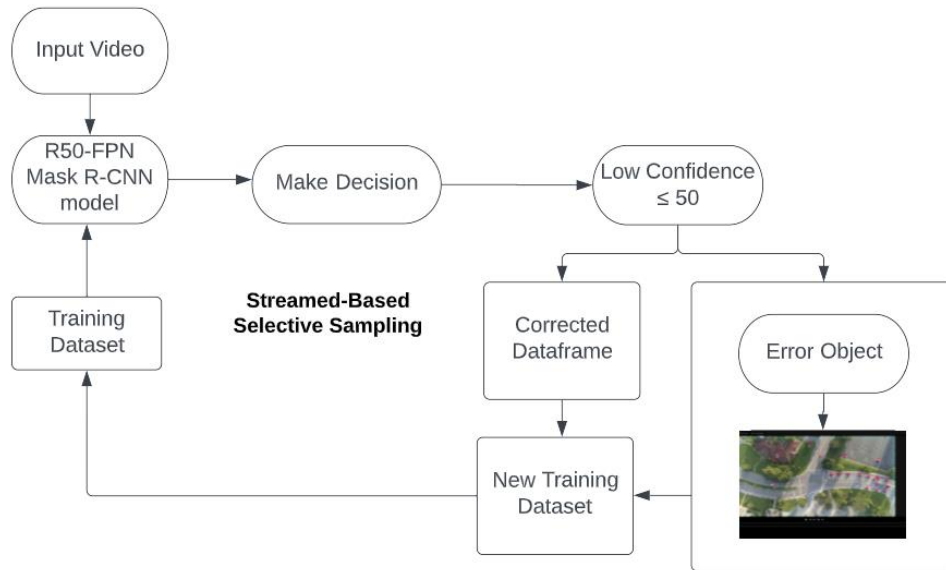


Figure 25 Streamed-Based Selective Sampling flow chart

4.3.4 Results

The training dataset substantially influences the quality of a trained classification model. To examine the performance of training outcomes, a 10,000 sample with most information vehicle frames were used. Figure 26 shows an example of detection resulting in an image, which includes a bounding box (red rectangle), segmentation mask, class of vehicle, and a confidence score of 86 percent.

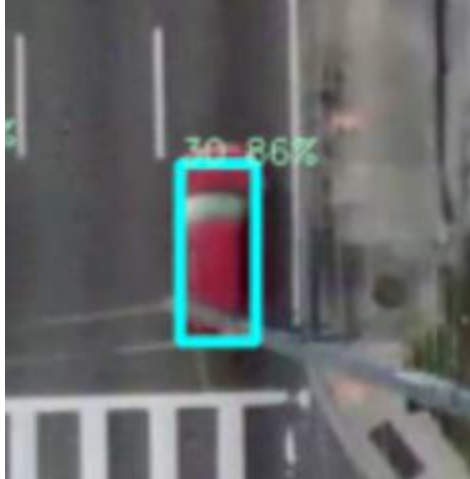


Figure 26 An example detection made using Detectron2 algorithm.

Two experiment methods are designed to keep the labeled training dataset as balanced as possible. To achieve this task, 500 frames as the number for input on each training round. According to the results table, MAP represents the mean of AP50 and AP75, whereas APmedium reflects the size of the detected item. Because of the widespread use of drone recordings, every vehicle on the road will seem to be identical, and as a result, the terms APsmall and APlarge will be rendered meaningless. Therefore, the value of APmedium will always be about equivalent to that of MAP. The MAP value in both methods for the initial model is 61.94%, after 3 times the active learning cycle repeated, the value in method 1 has increased 15%, whereas method 2 increased 23%. Both values reflect a substantial increase. As shown in Chart 1, the MAP has grown with each training cycle, which demonstrates that both procedures have improved their training model. In each loop of Pool-based Sampling scenario, most informative data samples are selected from the input of unlabeled data samples based on informativeness measure that confidence score $\geq 90\%$. These data samples improve the model around average 8% accuracy from the initial training dataset. Since each iteration needs select 500 frame samples, unlabeled data must be randomly selected

from a pool dataset if the model lacks the required quantity in order to achieve the same input every time. As another result from Stream-based scenario, which involves human labeled data in each loop that confidence score $\leq 50\%$. The stream-based selective technique may acquire a higher AP score than the first method. The reason it could be used is because this approach uses perfectly fixed labels from humans as input sample, which makes it more accurate for models to train data than randomly selected training datasets.

Table 1 Result of labeled training dataset

Method 1	MAP	AP50	AP75	APmedium
1 st Training (2000 frames)	61.94	68.557	55.323	61.94
2 nd Training (add 500 frames)	70.927	75.310	66.544	70.927
3 rd Training (add 500 frames)	76.992	81.219	72.765	76.992
Method 2	MAP	AP50	AP75	APmedium
1 st Training (2000 frames)	61.94	68.557	55.323	61.94
2 nd Training (add 500 frames)	73.933	78.384	69.483	73.933
3 rd Training (add 500 frames)	84.884	87.529	82.240	84.884

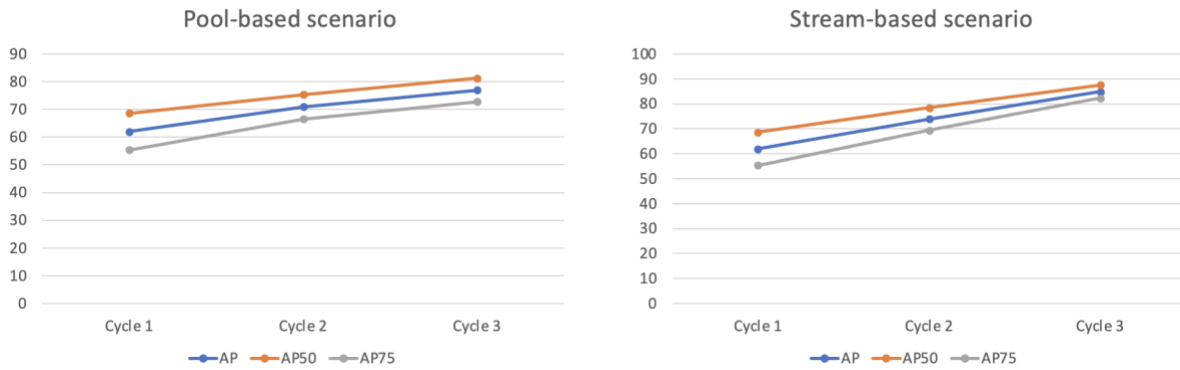


Figure 27 Percentage of the classification accuracy for stream-based and pool-based scenario

A limitation of this study is the variety of drone videos with perfectly labeled data that were utilized as input; the test results increased considerable value in each training because the input data was 2000 frames only. If the dataset was large enough, the results could be lower than the current results. Ultimately, increasing the default number of training iterations for Detectron2 could produce more accurate results. On the other hand, the Detectron2 helps to train a dataset that correctly recognizes vehicles from drone videos, generates AP results for each training model, and consistently achieves improved output.

4.4 Visualization Tool

The Data Visualization Tool designed for ARCIS is to extract the most valuable insights and visualize the data in different formats for traffic analysis. The table file from the previous implemented process result can be analyzed and reviewed to identify intersection with safety issues or with potential for future safety issues. Using vehicle trajectories, the safety issue can be identified by calculating PET, TTC value, and conflict type events. In addition, the average speed and acceleration of each vehicle can be determined using its GPS coordinates.

4.4.1 Vehicle Trajectory

Trajectory data can be used to describe safety situations for a certain time and spatial range. The trajectory of each vehicle can be determined from the result table, which contains the duration of each vehicle in frames. To estimate the duration of a single vehicle, simply subtract the last frame before it disappeared from the first frame using the vehicle's ID to identify. Given the fact that the software extracts 30 frames per second from the video, the following equation can be used to compute the total amount of time that a vehicle spends traveling through an intersection.



Figure 28 An example of a vehicle trajectory

4.4.2 Surrogate Measures

TTC is the shortest time before a collision was observed during the conflict. To calculate the TTC values that may occur during the observed intersection using the current position, speed, and future trajectory of each vehicle based on the result table. The value is determined by the timestamp between two vehicles, considering their future trajectories to estimate the minimum possible occurrence time. The PET values were calculated between when the first vehicle last occupied a pixel and the time when the second vehicle subsequently arrived at the same pixels and one value was returned for each conflict corresponding to the pixel level conflict position. Conflict type is the event of a side-wipe, head-on, or angle movement to identify potential conflicts that were obtained based on the PET values with a certain threshold.

4.4.3 Speed and Acceleration

Speed is a measurement of how fast a vehicle is traveling in a certain time; in a general equation, speed equals the distance traveled divided by the time. To compute the speed of a vehicle in a video by evaluating its departure and arrival distance and time based on its GPS coordinates and the total number of frames from the result table. The acceleration of a vehicle is the rate at which its velocity changes over time, which can be computed by determining the velocities at two consecutive locations in a total number of frames and the distance traveled by utilizing GPS coordinates.

4.4.4 User-interface

The visualization tool will automatically calculate all the values in the previously saved result table file. The tool has displayed all the measurement value into two components with Overview tab and Safety Analysis tab, allowing users to comprehend data quickly and maximize operational efficiency and productivity. To develop a user-friendly and interactive interface that is easy to access, all elements must be present in the right proportions, and certain specific errors must be avoided to build a meaningful visualization. The tool's interactive data visualization enables data exploration through the manipulation of a visual representation of the data. The measurements of trajectories, speeds, and accelerations for all the vehicles in a video are displayed on the Overview tab, which is utilized to perform an overview analysis of an intersection. To help visualize result data in a more detailed and useful way, it comes in different forms such as line graph, information box, and moving path in image. A video player with controls has been implemented to play and pause the video and allow the user to select a specific vehicle to closely examine it. The user may simply click the center point in each vehicle to visualize the future trajectory path, and on the right

panel (Figure 28), a line graph with the selected vehicle's speed and accelerations also be provided. The reason a line graph with distance on the x-axis and time on the y-axis is used to illustrate a vehicle's speed and accelerations is that it can assist the user in determining the changing speed at different distances.

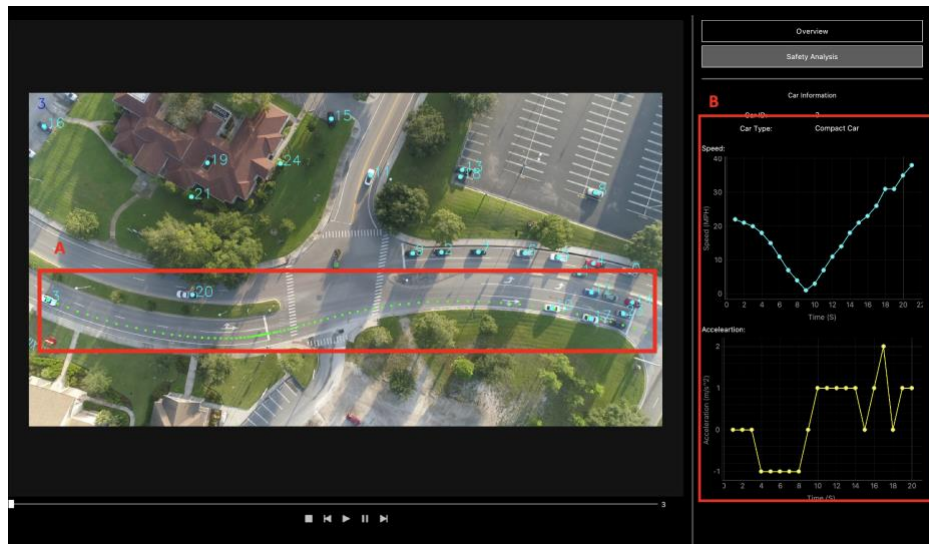


Figure 29 Data Visualization tool functionalities chart

Using a 3D map on the MapBox platform is another optimal method for illustrating the speed of every vehicle. Using a 3D map visualization technique, a static two-dimensional graph can be transformed into a dynamic visual display allowing the user to view more information. An example shows in Figure 29 of the Safety Analysis tab, the speed of each vehicle is demonstrated in a hexagon shape and speed value in its height level. The interface allows users to modify the presented results by selecting a value from a drop-down menu to either filter by vehicle id or by time speed. This interactive feature gives users more accurate indicators in extracting the data they need for analysis. A heatmap in 3D form uses color to represent value for PET/TTC measurement and the darker colors are perceived as being higher change to cause a conflict. Conflict type has been using markers. To demonstrate a conflict event, a marker is placed in the area where a

collision may occur, and a table on the side provides the user with additional information about the event, such as the ids of the two vehicles involved, the time, and the conflict type (Figure 30). Color-coded systems are utilized, and the reason it was selected was because of their ability to make better visualization of the volume of locations that are taking place within the dataset. This made it easier for the users to be directed to the information of value to them. The large volume of data that traffic contains can display the data in a much more generalized view to numeric values.

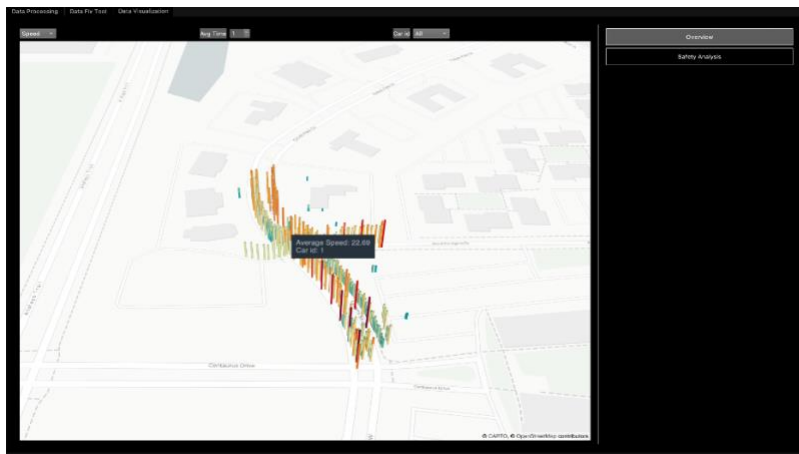


Figure 30 An example of the Safety Analysis tab



TTC

PET

Conflict Type

Figure 31 An example of conflict event

CHAPTER 5 CASE STUDY

With all the tasks outlined in the previous chapters, the software-friendly user interface for ARCIS has been established. Continuous testing is frequently considered while delivering dependable and high-quality software to the end user. A case study has been presented in this chapter which has followed the work-flow diagram (Figure 7) to demonstrate this software. A raw video from UAV has been used for the test data as input, where the video is a typical 4-leg intersection of SR 436 and Oxford Road.

Before all vehicles can be detected in a video, the video must be preprocessed, which includes object filtering and stabilization. Since this raw video consists of just a single file, no stitching is required at this stage. Figure 31 illustrates the outcome after the video has removed objects that could cause a detection error, as well as the stabilization results after selecting six stationary objects as reference points.



Figure 32 An example of result after removed objects that could cause a detection error

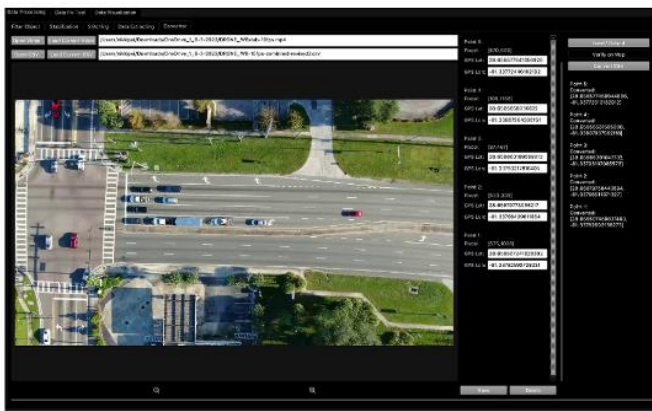
The next step is to use the data extorter interface to detect all vehicles in videos and generate output data for each vehicle that includes trajectory, head, tail, and bounding box. The output table file will be sent to the interface of a convertor that allows all vehicles to convert the pixel coordinates to GPS coordinates (Figure 32).



Before detection



After detection



Convert from Pixel to GPS

A	B	C	D	U	V
FrameNo	carID	carCenterX	carCenterY	lat	lon
0	0	227	282	28.6588479	-81.337816
0	22	429	1101	28.6585079	-81.337913
0	21	405	407	28.6587988	-81.337742
0	14	207	861	28.6586618	-81.337956
0	11	1273	746	28.6584481	-81.337418
0	9	1488	754	28.6583931	-81.337318
0	8	990	750	28.6585156	-81.337553
0	10	363	879	28.6586161	-81.337886
0	6	2099	899	28.6582468	-81.337017
0	5	762	679	28.6586031	-81.337642
0	4	755	674	28.6586205	-81.337629
0	3	948	616	28.6585811	-81.337539
0	2	796	550	28.6586452	-81.337594
0	1	968	547	28.6586045	-81.337512
1	0	733	744	28.658581	-81.337674
1	9	1487	754	28.6583939	-81.337819
1	22	433	1087	28.6585128	-81.337908
1	21	405	381	28.6588085	-81.337736
1	14	209	879	28.6586539	-81.337696
1	11	1275	746	28.6584481	-81.337417
1	10	367	860	28.6586229	-81.337879
1	8	992	750	28.6585155	-81.337552
1	3	948	616	28.6585811	-81.337539
1	6	2085	700	28.658271	-81.337024
1	5	763	674	28.6586025	-81.337641
1	4	755	610	28.6586305	-81.337629
1	2	796	550	28.6586452	-81.337594
1	1	968	547	28.6586045	-81.337512
1	0	236	401	28.6588425	-81.337821
1	7	731	745	28.6585811	-81.337675

Output table result

Figure 33 An example of convert the pixel coordinates to GPS coordinates

Automated detection may contain errors, such as the detection of a missing vehicle or the wrong object. Consequently, detected data must be fixed for a high-quality and reliable dataset prior to analysis. Figure 33 have shown the before and after result using Data Fix Tool to correct the erroneously detected bounding box caused by shadow and to fix the bounding box to accurately cover the vehicle.



Figure 34 An example of before and after result using Data Fix Tool

The data visualizer will present fixed data and generate safety measurements for evaluation; Figure 34 is an illustration of a vehicle id of 22 trajectory that is attempting to turn to the right. The graph on the side of the main window indicates that the vehicle accelerates as it turns onto the main road. In addition, the speed of a vehicle can also be displayed on a 3D map for a much broader perspective.



Figure 35 An example of data visualizer generates safety measurements for evaluation

To analyze the safety conditions for the intersection in the video can be viewing the surrogate safety measures such as TTC and PET. Figure 35 demonstrates a comparison of the trajectories

of vehicles based on PET and TTC values with different thresholds in seconds. The higher the threshold on time, the greater the conflict event count. A conflict type is presented in Figure 36 to highlight that a head-on collision involving vehicle id 56 has the potential to occur.



Figure 36 Vehicles' trajectories on PET and TTC values are less than 1.5 s in a 3D heatmap

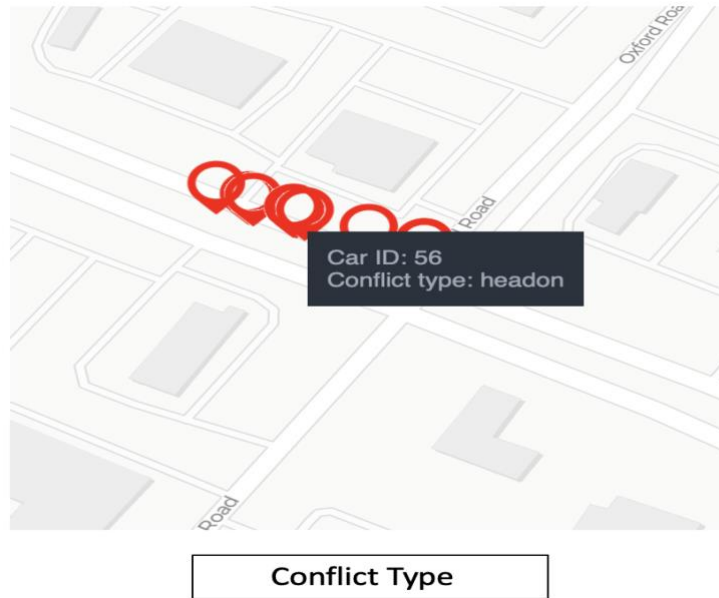


Figure 37 Example of conflict type event

As a result, the test case using a raw video of a 4-leg intersection has successfully generated the output data and video for traffic safety analysis. This software's testing is time-consuming for longer videos and large datasets, as the preprocessing and detection stages need frame-by-frame video processing. Future studies should focus on combining these stages into a single frame process to increase processing speed.

CHAPTER 6 CONCLUSION

In this thesis, a user-friendly interface designed for the ARCIS system is introduced, and the effort on ARCIS additional development is demonstrated. With the integration ARCIS drone video processing function, a human-in-the-loop tool for correcting data, and traffic safety visualization, the software provides a convenient platform for end user to perform conflict identification and safety diagnostics based on UAV videos.

The framework of UAV video processing is presented, which includes video stabilization, video stitching, pixel to GPS converter, and data extraction. Afterwards, in order to reduce the labeling time and cost, the human-in-the loop data fix tool adopted the active learning technique. Analysis for the different active learning strategy was conduct in the study, method based on Stream-Based Selective Sampling strategy out preformed others. Besides, visualization method of traffic safety data is discussed, and the operation guide of the user interface is explained. Finally, A case study was conducted at a typical 4 leg signalized intersection at Orlando. The case study has validated the feasibility of investigating safety situations from UAV videos based on different traffic parameters and surrogate safety measures (i.e., PET) output from ARCIS.

There are few limitations. First, only UAV video can be processed for vehicle detection. Second, the computational efficiency of the system still needs to be improved. Last, more advanced analysis of traffic safety (e.g., cause of conflict, hot spot identification) is expected to be included. In the future, the software might be extended to process any types of video data in addition to drone video data. This software can be viewed as another steppingstone in the process of diagnosing road safety risks.

REFERENCES

1. Agarwal, A., Gupta, S., & Singh, D. K. (2016). Review of optical flow technique for moving object detection. In *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*. <https://doi.org/10.1109/ic3i.2016.7917999>
2. Aljabri, M., AlAmir, M., AlGhamdi, M., Abdel-Mottaleb, M., & Collado-Mesa, F. (2022). Towards a better understanding of annotation tools for medical imaging: a survey. *Multimedia Tools and Applications*, 1–35.
3. A new framework for the integration, analysis and visualization of urban traffic data within geographic information systems. (2000). *Transportation Research Part C: Emerging Technologies*, 8(1-6), 167–184.
4. Bachechi, C., Po, L., & Rollo, F. (2022). Big Data Analytics and Visualization in Traffic Monitoring. In *Big Data Research* (Vol. 27, p. 100292). <https://doi.org/10.1016/j.bdr.2021.100292>
5. Benomar, O., Sahraoui, H., & Poulin, P. (2013). Visualizing software dynamicities with heat maps. In *2013 First IEEE Working Conference on Software Visualization (VISSOFT)*. <https://doi.org/10.1109/vissoft.2013.6650524>
6. Berres, A. S., Xu, H., Tennille, S. A., Severino, J., Ravulaparthi, S., & Sanyal, J. (2021). Explorative Visualization for Traffic Safety using Adaptive Study Areas. In *Transportation Research Record: Journal of the Transportation Research Board* (Vol. 2675, Issue 6, pp. 51–69). <https://doi.org/10.1177/0361198120981065>
7. Brozen, M., Rios, N., Cardenas, I., Ekman, A. Y., Bressette, B., University of California, Los Angeles. Lewis Center for Regional Policy Studies, & Pacific Southwest Region 9 UTC, University of Southern California. (2021). *Intersectional Transportation Trends in LA County* (No. PSR-19-60-TO-029). United States. Dept. of Transportation. Office of the Assistant Secretary for Research and Technology. <https://rosap.ntl.bts.gov/view/dot/60611>

8. Chen, Y., Mani, S., & Xu, H. (2012). Applying active learning to assertion classification of concepts in clinical text. *Journal of Biomedical Informatics*, 45(2), 265–272.
9. Choi, H.-T., Lee, H.-J., Kang, H., Yu, S., & Park, H.-H. (2021). SSD-EMB: An Improved SSD Using Enhanced Feature Map Block for Object Detection. *Sensors*, 21(8).
<https://doi.org/10.3390/s21082842>
10. Csail, Z. A. M., Xu Chu University of Waterloo, Dong Deng Tsinghua University, Csail, R. C. F., Ihab F. Ilyas University of Waterloo, Mourad Ouzzani Qatar Computing Research Institute, HBKU, Paolo Papotti Arizona State University, Csail, M. S. M., & Nan Tang Qatar Computing Research Institute, HBKU. (2016). Detecting data errors. *Proceedings of the VLDB Endowment International Conference on Very Large Data Bases*. <https://doi.org/10.14778/2994509.2994518>
11. Diego, F., Serrat, J., & Lopez, A. M. (2013). Joint Spatio-Temporal Alignment of Sequences. In *IEEE Transactions on Multimedia* (Vol. 15, Issue 6, pp. 1377–1387).
<https://doi.org/10.1109/tmm.2013.2247390>
12. Dubska, M., Herout, A., Juranek, R., & Sochor, J. (2015). Fully Automatic Roadside Camera Calibration for Traffic Surveillance. In *IEEE Transactions on Intelligent Transportation Systems* (Vol. 16, Issue 3, pp. 1162–1171). <https://doi.org/10.1109/tits.2014.2352854>
13. Essa, M., & Sayed, T. (2018). Traffic conflict models to evaluate the safety of signalized intersections at the cycle level. In *Transportation Research Part C: Emerging Technologies* (Vol. 89, pp. 289–302). <https://doi.org/10.1016/j.trc.2018.02.014>
14. Goyani, J., Paul, A. B., Gore, N., Arkatkar, S., & Joshi, G. (2021). Investigation of Crossing Conflicts by Vehicle Type at Unsignalized T-Intersections under Varying Roadway and Traffic Conditions in India. In *Journal of Transportation Engineering, Part A: Systems* (Vol. 147, Issue 2, p. 05020011). <https://doi.org/10.1061/jtepbs.0000479>
15. Guo, B. H. W., Zou, Y., Fang, Y., Goh, Y. M., & Zou, P. X. W. (2021). Computer vision technologies for safety science and management in construction: A critical review and future research directions. In *Safety Science* (Vol. 135, p. 105130).

<https://doi.org/10.1016/j.ssci.2020.105130>

16. Healy, K. (2018). *Data Visualization: A Practical Introduction*. Princeton University Press.
17. Huang, H., Huang, J., Feng, Y., Zhang, J., Liu, Z., Wang, Q., & Chen, L. (2019). On the improvement of reinforcement active learning with the involvement of cross entropy to address one-shot learning problem. In *PLOS ONE* (Vol. 14, Issue 6, p. e0217408).
<https://doi.org/10.1371/journal.pone.0217408>
18. Hwang, Y.-J., Lee, J.-G., Moon, U.-C., & Park, H.-H. (2020). SSD-TSEFFM: New SSD Using Trident Feature and Squeeze and Extraction Feature Fusion. *Sensors*, 20(13).
<https://doi.org/10.3390/s20133630>
19. Kamel, S., Ebrahimnezhad, H., & Ebrahimi, A. (2008). Moving object removal in video sequence and background restoration using kalman filter. In *2008 International Symposium on Telecommunications*. <https://doi.org/10.1109/istel.2008.4651368>
20. Kastrinaki, V., Zervakis, M., & Kalaitzakis, K. (2003). A survey of video processing techniques for traffic applications. In *Image and Vision Computing* (Vol. 21, Issue 4, pp. 359–381).
[https://doi.org/10.1016/s0262-8856\(03\)00004-0](https://doi.org/10.1016/s0262-8856(03)00004-0)
21. Kiefer, R. J., Flannagan, C. A., & Jerome, C. J. (2006). Time-to-collision judgments under realistic driving conditions. *Human Factors*, 48(2), 334–345.
22. Kolekar, A., & Dalal, V. (n.d.). Barcode Detection and Classification using SSD (Single Shot Multibox Detector) Deep Learning Algorithm. In *SSRN Electronic Journal*.
<https://doi.org/10.2139/ssrn.3568499>
23. Lan, W., Dang, J., Wang, Y., & Wang, S. (2018). Pedestrian Detection Based on YOLO Network Model. In *2018 IEEE International Conference on Mechatronics and Automation (ICMA)*.
<https://doi.org/10.1109/icma.2018.8484698>
24. Lee, C., Kim, Y., Jin, S., Kim, D., Maciejewski, R., Ebert, D., & Ko, S. (2020). A Visual Analytics System for Exploring, Monitoring, and Forecasting Road Traffic Congestion. *IEEE Transactions on Visualization and Computer Graphics*, 26(11), 3133–3146.

25. Lewis, D. D., & Gale, W. A. (1994). A Sequential Algorithm for Training Text Classifiers. In *SIGIR '94* (pp. 3–12). https://doi.org/10.1007/978-1-4471-2099-5_1
26. Liu, Y., & Yan, J. (2021). Research on Detection Algorithm of Wheel Position based on CenterNet. In *Journal of Physics: Conference Series* (Vol. 1802, Issue 3, p. 032126). <https://doi.org/10.1088/1742-6596/1802/3/032126>
27. Lukezic, A., Vojir, T., Zajc, L. C., Matas, J., & Kristan, M. (2017). Discriminative Correlation Filter with Channel and Spatial Reliability. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. <https://doi.org/10.1109/cvpr.2017.515>
28. Ma, C., Yang, X., Zhang, C., & Yang, M.-H. (2015). Long-term correlation tracking. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. <https://doi.org/10.1109/cvpr.2015.7299177>
29. Meng, L. I., Han, W. X., & Ke, S. H. I. (2017). Traffic Conflict Identification Technology of Vehicle Intersection Based on Vehicle Video Trajectory Extraction. In *Procedia Computer Science* (Vol. 109, pp. 963–968). <https://doi.org/10.1016/j.procs.2017.05.454>
30. Mizokami, S. (2018). Deep Active Learning from the Perspective of Active Learning Theory. In *Deep Active Learning* (pp. 79–91). https://doi.org/10.1007/978-981-10-5660-4_5
31. Pham, M.-T., Courtrai, L., Friguet, C., Lefèvre, S., & Baussard, A. (2020). YOLO-Fine: One-Stage Detector of Small Objects Under Various Backgrounds in Remote Sensing Images. In *Remote Sensing* (Vol. 12, Issue 15, p. 2501). <https://doi.org/10.3390/rs12152501>
32. Purushwalkam, S., Ye, T., Gupta, S., & Gupta, A. (2020). Aligning Videos in Space and Time. In *Computer Vision – ECCV 2020* (pp. 262–278). https://doi.org/10.1007/978-3-030-58574-7_16
33. Rahman, F. Y. A., Hussain, A., Zaki, W. M. D., Zaman, H. B., & Tahir, N. M. (2013). Enhancement of Background Subtraction Techniques Using a Second Derivative in Gradient Direction Filter. In *Journal of Electrical and Computer Engineering* (Vol. 2013, pp. 1–12). <https://doi.org/10.1155/2013/598708>
34. Rahmat-Samii, Y., & Topsakal, E. (2021). *Antenna and Sensor Technologies in Modern Medical*

Applications. John Wiley & Sons.

35. Rawat, S., Chandra, A. L., Desai, S. V., Balasubramanian, V. N., Ninomiya, S., & Guo, W. (2022). How Useful Is Image-Based Active Learning for Plant Organ Segmentation? *Plant Phenomics* (Washington, D.C.), 2022, 9795275.
36. 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.
37. Sakla, W., Konjevod, G., & Nathan Mundhenk, T. (2017). Deep Multi-modal Vehicle Detection in Aerial ISR Imagery. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. <https://doi.org/10.1109/wacv.2017.107>
38. Scheepens, R., Hurter, C., van de Wetering, H., & van Wijk, J. J. (2016). Visualization, Selection, and Analysis of Traffic Flows. *IEEE Transactions on Visualization and Computer Graphics*, 22(1), 379–388.
39. Scheepens, R., Willems, N., van de Wetering, H., & van Wijk, J. J. (2011). Interactive visualization of multivariate trajectory data with density maps. In *2011 IEEE Pacific Visualization Symposium*. <https://doi.org/10.1109/pacificvis.2011.5742384>
40. Sommer, L. W., Schuchert, T., & Beyerer, J. (2017). Fast Deep Vehicle Detection in Aerial Images. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. <https://doi.org/10.1109/wacv.2017.41>
41. Songchitruksa, P., & Tarko, A. P. (2006). Practical Method for Estimating Frequency of Right-Angle Collisions at Traffic Signals. In *Transportation Research Record: Journal of the Transportation Research Board* (Vol. 1953, Issue 1, pp. 89–97). <https://doi.org/10.1177/0361198106195300111>
42. Sun, C., Zhan, W., She, J., & Zhang, Y. (2020). Corrigendum to “Object Detection from the Video Taken by Drone via Convolutional Neural Networks.” In *Mathematical Problems in Engineering* (Vol. 2020, pp. 1–1). <https://doi.org/10.1155/2020/4806359>

43. Ullah, A., Xie, H., Farooq, M. O., & Sun, Z. (2018). Pedestrian Detection in Infrared Images Using Fast RCNN. In *2018 Eighth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. <https://doi.org/10.1109/ipta.2018.8608121>
44. van der Horst, A. R. A., & van der Horst, A. R. (1990). *A Time-based Analysis of Road User Behaviour in Normal and Critical Encounters*.
45. Wang, L., Zhao, H., Guo, S., Mai, Y., & Liu, S. (2012). The adaptive compensation algorithm for small UAV image stabilization. In *2012 IEEE International Geoscience and Remote Sensing Symposium*. <https://doi.org/10.1109/igarss.2012.6350400>
46. Wu, Y., Abdel-Aty, M., Zheng, O., Cai, Q., & Zhang, S. (2020). Automated Safety Diagnosis Based on Unmanned Aerial Vehicle Video and Deep Learning Algorithm. In *Transportation Research Record: Journal of the Transportation Research Board* (Vol. 2674, Issue 8, pp. 350–359). <https://doi.org/10.1177/0361198120925808>
47. Yang, J., Schonfeld, D., & Mohamed, M. (2009). Robust Video Stabilization Based on Particle Filter Tracking of Projected Camera Motion. In *IEEE Transactions on Circuits and Systems for Video Technology* (Vol. 19, Issue 7, pp. 945–954). <https://doi.org/10.1109/tcsvt.2009.2020252>
48. Yu, C.-W., Chen, Y.-L., Lee, K.-F., Chen, C.-H., & Hsiao, C.-Y. (2019). Efficient Intelligent Automatic Image Annotation Method based on Machine Learning Techniques. In *2019 IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW)*. <https://doi.org/10.1109/icce-tw46550.2019.8991727>
49. Yushkevich, P. A., Pashchinskiy, A., Oguz, I., Mohan, S., Schmitt, J. E., Stein, J. M., Zukić, D., Vicory, J., McCormick, M., Yushkevich, N., Schwartz, N., Gao, Y., & Gerig, G. (2019). User-Guided Segmentation of Multi-modality Medical Imaging Datasets with ITK-SNAP. *Neuroinformatics*, *17*(1), 83–102.
50. Zhong, J., & Jianbo, S. U. (2010). A Real-time Moving Object Tracking System Based on Visual Prediction. In *ROBOT* (Vol. 32, Issue 4, pp. 516–521). <https://doi.org/10.3724/sp.j.1218.2010.00516>

51. Zhu, Y., Demiroglu, S., Ozbay, K., Xie, K., Yang, H., & Sha, D. (2021). SAVE-T: Safety Analysis Visualization and Evaluation Tool. *Journal of Advanced Transportation*, 2021. <https://doi.org/10.1155/2021/5545117>