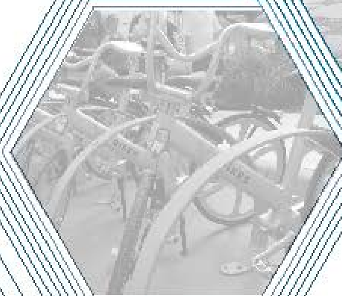


Enhancing Traffic Flow and Driving Safety via Artificial Intelligence

Amir Shahmoradi, Ph.D.
Joshua Osborne
Parvat Sapkota
Christopher Bryant
Niyousha Davachi

FINAL REPORT

Center for Transportation Equity, Decisions & Dollars

ENHANCING TRAFFIC FLOW AND DRIVING SAFETY VIA ARTIFICIAL INTELLIGENCE

FINAL PROJECT REPORT

By:

Amir Shahmoradi, Ph.D.
University of Texas at Arlington

Joshua Osborne
Parvat Sapkota
Christopher Bryant
Niyousha Davachi
University of Texas at Arlington

Sponsorship:
CTEDD

For:
Center for Transportation, Equity, Decisions and Dollars **(CTEDD)**
USDOT University Transportation Center
The University of Texas at Arlington
Woolf Hall, Suite 325
Arlington TX 76019 United States
Phone: 817-272-5138 | Email: c-tedd@uta.edu

In cooperation with United States Department of Transportation's Office of Research, Development, and Technology (RD&T)

Acknowledgments

This work was supported by a grant from the Center for Transportation Equity, Decisions, and Dollars (CTEDD) funded by U.S. Department of Transportation Research and Innovative Technology Administration (OST-R) and housed at The University of Texas at Arlington.

Disclaimer

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
CTEDD 019-23		

4. Title and Subtitle	**5. Report Date**
Enhancing Traffic Flow and Driving Safety via Artificial Intelligence	May 31, 2021
	6. Performing Organization Code
7. Author(s)	**8. Performing Organization Report No.**
Amir Shahmoradi (0000-0002-9720-8937), Joshua Osborne (0000-0003-1321-748X), Parvat Sapkota, Christopher Bryant, Niyousha Davachi	
9. Performing Organization Name and Address	**10. Work Unit No. (TRAIS)**
Department of Physics\ Department of Data Science The University of Texas at Arlington 655 W Mitchell St, Arlington, TX 76010	
	11. Contract or Grant No.
	USDOT - 69A3551747134
12. Sponsoring Organization Name and Address	**13. Type of Report and Period Covered**
Center for Transportation, Equity, Decisions and Dollars (CTEDD) USDOT University Transportation Center The University of Texas at Arlington Woolf Hall, Suite 325 Arlington TX 76019 United States	Final Report
	14. Sponsoring Agency Code

15. Supplementary Notes

Additional project information is available at ctedd.uta.edu. Project performed under a grant from the U.S. Department of Transportation's University Transportation Centers (UTC) Program.

16. Abstract

Motivated by the priorities highlighted by Texas Department of Transportation (TXDOT) and following the guidelines in the recent presidential "Executive Order on Maintaining American Leadership in Artificial Intelligence" in 2019, this proposal aims to utilize the state-of-the-art tools and techniques in the field of Artificial Intelligence and Data Science to automatically identify and report traffic-related anomalies and hazards using live traffic camera footage across major highways and arterial roads in the State of Texas. Examples of such hazards that are the focus of this proposal include major vehicle-wildlife and vehicle-debris encounters (VWEs and VDEs respectively). Our work builds on top of the existing massive body of literature and research at the intersection of Computer Vision and Traffic Engineering. However, to the extent of our knowledge, this proposal is the first attempt to study the development of an automated pipeline for the detection and reporting of VWEs and VDEs using live traffic camera data.

To this aim, we outline and investigate the feasibility of our approach to set up a prototype of a pipeline for real-time collection of data, its reduction, segmentation, analysis, and finally, drawing traffic engineering conclusions and recommendations based on the detected patterns or anomalies in the analysis.

This exploratory investigation aims to provide a comprehensive review and proof-of-concept to pave the way towards the next-step proposal and implementation of a commercial-scale version of such data-analytics pipeline in collaboration with the potential major stakeholders, in particular, within Dallas-Fort Worth Metroplex.

17. Key Words	**18. Distribution Statement**
Traffic Engineering, Traffic Camera, Data Science, Artificial Intelligence, Anomaly Detection	No restrictions. This document is available to the public through the Transport Research International Documentation (TRID) repository.

19. Security Classification (of this report)	20. Security Classification (of this page)	21. No. of Pages	22. Price
Unclassified.	Unclassified.	37	N/A

Table of Contents

List of Figures

Abstract

Motivated by the priorities highlighted by Texas Department of Transportation (TXDOT) and following the guidelines in the recent presidential "Executive Order on Maintaining American Leadership in Artificial Intelligence" in 2019, this proposal aims to utilize the state-of-the-art tools and techniques in the field of Artificial Intelligence and Data Science to automatically identify and report traffic-related anomalies and hazards using live traffic camera footage across major highways and arterial roads in the State of Texas. Examples of such hazards that are the focus of this proposal include major vehicle-wildlife and vehicle-debris encounters (VWEs and VDEs respectively). Our work builds on top of the existing massive body of literature and research at the intersection of Computer Vision and Traffic Engineering. However, to the extent of our knowledge, this proposal is the first attempt to study the development of an automated pipeline for the detection and reporting of VWEs and VDEs using live traffic camera data.

To this aim, we outline and investigate the feasibility of our approach to set up a prototype of a pipeline for real-time collection of data, its reduction, segmentation, analysis, and finally, drawing traffic engineering conclusions and recommendations based on the detected patterns or anomalies in the analysis.

This exploratory investigation aims to provide a comprehensive review and proof-of-concept to pave the way toward a larger-scale proposal and implementation of a commercial-scale version of such data-analytics pipeline in collaboration with the potential major stakeholders, in particular, within Dallas-Fort Worth Metroplex.

Chapter 1: Introduction

One of the major goals within traffic management projects are to enhance the flow of traffic for economic benefits by increasing productivity and reducing traffic congestion. By alleviating traffic, it would be possible to improve upon the impact of traffic congestion on air pollution due to vehicle emission and increase the efficiency of said vehicles (Treiber et al., 2008; Wadud et al., 2016; Fontaras et al., 2017; Pavlovic et al. 2018). The impact of traffic congestion has been investigated thoroughly through the years (Rakha & Ahn, 2004; Greenwood et al., 2007; Treiber et al., 2008; Barth and Boriboonsomsin, 2008). In fact, recent studies have even found that alleviating traffic congestion significantly lowers CO_2 emission (Beevers et al, 2016; Mascia et al., 2017) even up to nearly 20% in the case of southern California (Fiori et al, 2019). Although the number of vehicles is certainly the single most important factor in determining congestion, other factors can also significantly impact the smooth traffic flow of an area such as road work, traffic accidents, and weather conditions (Fontaras et al., 2017).

Traffic accidents are particularly of utmost importance as they are very often preventable. Traffic accidents also lead to economic loss through productivity through traffic congestion, as well as direct financial loss through property damage, medical, and legal expenses (Yang et al., 2013). For example, Sauber-Schatz et al. (2016) find that if the United States had a mortality rate (between vehicles and pedestrians) similar to that of Sweden, nearly 281,000,000 dollars is estimated to be saved in direct medical expenses alone.

In fact, pedestrians account for nearly one-fourth of all road traffic fatalities (see figure 7 of WHO, 2015) and more than one-third of all deaths and injuries worldwide (Peden et al., 2004). Vehicle-pedestrian encounters (**VPEs**) are a leading cause of death for those between 1-54 years old in the US (Sauber-Schatz et al., 2016), but also throughout the world (WHO, 2015; Peden et al., 2004). Therefore, it is not surprising that an extensive body of research has been invested into this topic of avoiding VPEs via computer vision (Dollar et al., 2012; Sivaraman & Trivedi, 2015; Wang & Sng, 2015; Xu et al., 2018). Interestingly, simply putting speed cameras in high incident areas to measure vehicle speed has been found to greatly reduce the number of accidents (Keall et al., 2001; Yang & Kim 2003). This method has also been found to have a high benefit to cost ratio (Gains et al., 2003). The topic of speeding vehicles is particularly important as it endangers everyone and has been estimated to cost the US $40.4 billion annually (Loce et al., 2013).

VPEs are not the only source of vehicle accidents, however; vehicle-wildlife encounters (**VWEs**) are also quite prevalent among vehicle accidents in the US. These encounters often result in vehicle damage and the loss of wildlife (Allen & McCullough, 1976). Based on a comprehensive examination of VWEs delivered to the US Congress by Huijser et al. (2008), The cost attributed to these encounters is estimated to be 8 billion dollars annually (not including animals smaller than deer and not including domesticated animals). Should the encounter result in the death of an endangered species or in the death of a person, the cost could be even higher. Currently, VWEs are already classified as a major threat to the survival of endangered wildlife (Huijser et al., 2008). It should also be noted that this number is likely underrepresenting the true amount of VWEs as this number is derived from carcass counts, the insurance industry, and police-reported crashes.

Before a problem like this can be tackled it will first be necessary to improve precision and collection methods of data for VWEs. Due to the lack of reliable data on VWEs it is often difficult to employ VWE mitigation techniques on roads where it is most needed. Therefore, any work which would be able to set up an automated pipeline to acquire and analyze live video footage could be very valuable and lead to a more consistent and precise source of data.

In addition to VPEs and VWEs, there is also the threat of debris on the roadway. Damage to vehicles can drastically vary depending on the object of interest. Fatalities resulting from vehicle-debris encounters (**VDEs**) seem to be low, but even so, the economic loss is quite significant. In a recent report from researchers at Texas A&M, more than 500 accidents within Texas can be attributed directly to VDEs and have been rising (Avelar et al., 2017). There have been attempts to solve this problem typically in the form of Radar or LiDAR technologies and more novel approaches based on Artificial Neural Networks, or a combination of such techniques typically using autonomous vehicles (Creusot & Munawar, 2015; Kinoshita et al., 2020). Many times, the detection of road debris requires drivers to manually report any obstacles on the roadway. This method is not reliable, however, due to the fact that drivers will not be able to identify every object that is on the road. Another issue with this method is the possibility of causing traffic accidents as drivers on their phones, reporting the debris, will be distracted. There has been progress made to address this problem via Basic Safety Messages (BSM) (Concas & Kamrani, 2019).

With the ever-increasing case of VPEs, VWEs, and VDEs as the number of vehicles on roads increases, it is important to find a solution to these issues. This is where computer vision (**CV**) can be helpful. The goal of CV is to automate visual tasks via computers by imitating human vision to some degree. CV requires the gathering, processing and, analysis of footage or videos to make inferences. CV is used in a variety of tasks ranging from reading handwritten postal codes, medical imaging, vehicle safety, etc. (Szeliski, 2010). Particularly in the case of VDEs, VPEs, and VWEs, we are interested in being able to identify and track relevant human, wildlife, car, or unidentified objects on roads. Over the past few years several various automatic generic image segmentation frameworks have been developed within the CV community, such as R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN, YOLO, YOLOv2, YOLOv3, and YOLOv4. These frameworks implement various forms of Convolutional Neural Networks (**CNNs**) for object detection. CNNs are part of the field of deep learning and are most commonly used to inspect images by using multiple hidden layers which are interconnected such as neurons in the brain (hence the name neural network).

The detection of wildlife on roads is virtually untapped despite being fairly similar to that of pedestrian detection. This is despite the fact that wildlife detection is far less complex as animals typically do not congregate en masse as opposed to pedestrians, making their individual detection a much simpler task than that of pedestrians (Xu et al., 2018).

In this work, we will be discussing deep learning techniques used in traffic engineering problems in chapter 2. In chapter 3, we will discuss the various problems in analyzing VWEs and VDEs. In chapter 4 we will discuss object detection frameworks and how they were implemented in this work. Finally, we will discuss the work done as part of this project toward establishing a pipeline for image segmentation and analysis traffic camera footage in chapters 5. Future goals and the roadmap ahead will be discussed in Chapter 6.

Chapter 2: Deep Learning Techniques for Image Segmentation of Traffic Camera Footage

Deep learning is part of a broader family of techniques known as machine learning whose objective is to teach computers to learn from previous experiences using an intricate web of connections known as neural networks. The applications of deep learning go far and wide, from language processing, bioinformatics, speech recognition, etc., for the goal of ultimately classifying an object, shape, or sound. In the case of the work described throughout this report, we are primarily interested in the classification of objects within an image. Therefore, in this section, we will be describing some of the basic techniques that are typically used.

2.1 Convolutional Neural Networks (CNNs)

In biology, neural networks are interweaved neurons that transmit electrical signals throughout the brain relaying information. In a similar fashion, neural networks in computers are an interconnected group of nodes that are fed some examples of the task it is to complete and over time "learns" how to complete these tasks on its own. A depiction of a very simple neural network is given below in **Figure 1**. If the input is a question such as "Do I want to go to the park?" then in the hidden layers we would have three different conditions with differing levels of importance such as "Is the weather good?", "Is someone going with me?", and "Is it late at night?". If we assign some "weight" to each question, say 2, 7, ad 3 respectively, and set a threshold of 5, then depending on how we answer the previous questions, we may or may not go. For example, if we answer yes to the first and last question we will go to the movies, but if we only answer yes to the first question we will instead stay home. Neural networks of course are more complex and typically have many more layers and more Perceptrons per layer such as the case of CNNs.

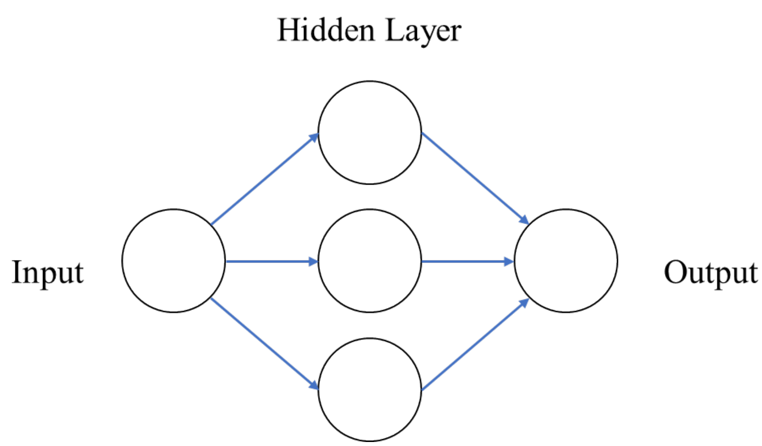


Figure 1: A depiction of a simple neural network. Each circle represents a perceptron, and each arrow represents some sort of information "flowing" between Perceptrons.

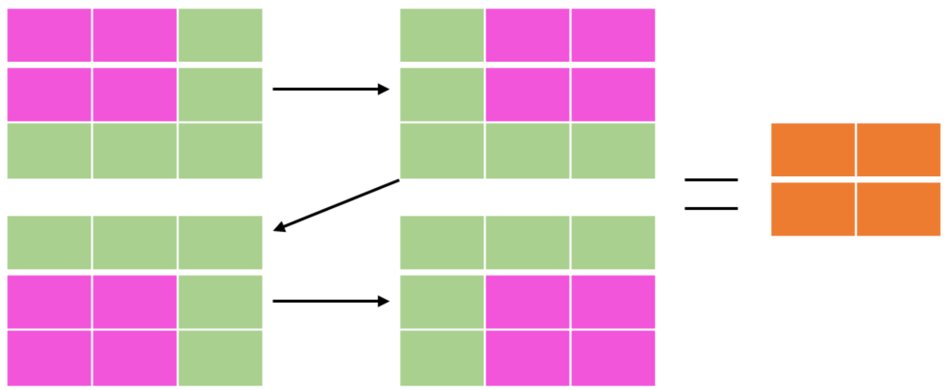


Figure 2: A depiction of how a CNN works. The magenta can be thought of as a filter. As this filter "slides" across the green area we take the dot product of the values from the green area with that of the magenta. Eventually we end up with a convolved feature (the orange)

A CNN is an artificial neural network that is created to specialize in recognizing some kind of pattern and make sense of it, typically within an image. The idea of a CNN is much like the neural network within **Figure 1**, but the hidden layers are what is known as convolutional layers (in addition to its non-convolutional layers). Within each of these convolutional layers, we have a filter. Each filter will have its own tasks such as detecting straight lines, squares, or circles. As the number of hidden layers grows so too does the complexity of our filters which are perhaps now looking for eyes, a nose, etc. So how does this work? We can imagine we have two boxes of values represented by the green box and magenta box in **Figure 2** where the green box is a matrix of values corresponding to our image and the magenta box represents the filter. As we move the filter across the area we take the dot product of the values in the magenta and green box or that is to say, we convolve the layers. We keep doing this until eventually we end up with our set of numbers called our convolved feature. In the case of identifying numbers, this last convolved feature may represent edges on the left side or edges on the right side, etc. depending on the filters we pass through.

2.2 Background subtraction

Another important step in image segmentation via deep learning methods is that of background subtraction which is used for tasks such as counting, detecting, and tracking vehicles. The aim of background subtraction is to essentially determine which objects in an image are static (part of the background) and what objects are moving (part of the foreground). One of the simplest forms of background subtraction is that of frame differencing. This method requires one to look at some frame for time "t", and the frame before at "t-1". We then simply take the absolute value of the difference between the two frames and if this value ends up being greater than some threshold we have set, then we will take it as our foreground. This technique, however, may struggle to track interior pixels for large moving objects such as the top of a white van.

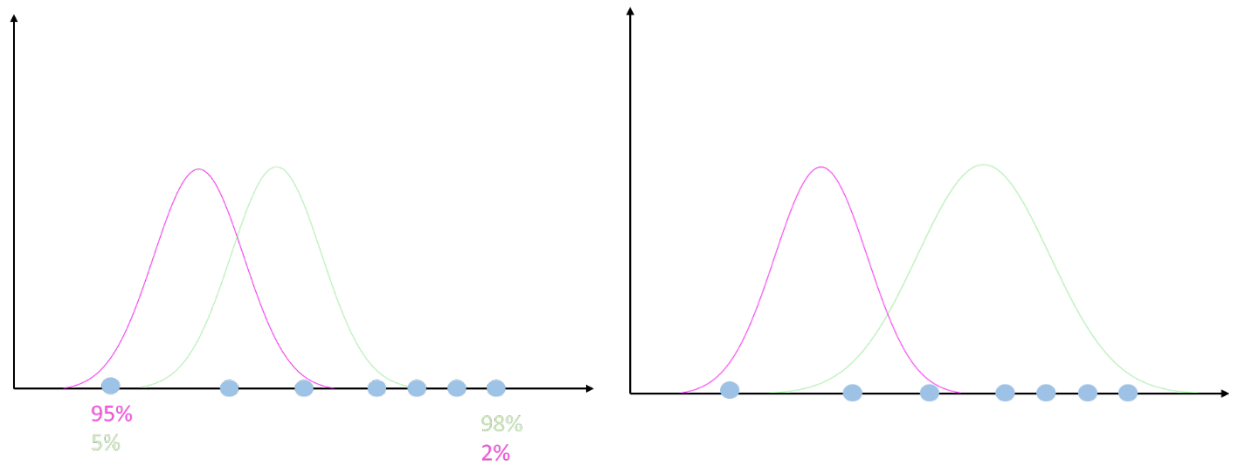


Figure 3: A depiction of how the GMM clusters a set of points probabilistically. The points to the right side of the green distribution are closer to the green distribution with some likelihood (represented as a percentage). Since the points on the right side fall closer to the green distribution, the mean of the green distribution is shifted right and its standard deviation increases.

Another method of background subtraction that is commonly used due to its accuracy is the mixture of Gaussian distributions (Piccardi 2004) also known as the Gaussian mixture model (**GMM**). This is a method typically used for clustering but is applicable to the field of image segmentation. The main assumption of this method is that a distribution of points is comprised of multiple Gaussian distributions, hence the name. The method can be understood by looking at **Figure 3.** First, we start out by assuming that our set of points from a given traffic camera image is represented by two Gaussian distributions. Then we test to see how close the points fall to each of these distributions and assign a probability to each point that tells us the distribution to which the point most likely belongs to. In this case, we find that the points seem to fall closer to the green distribution and therefore the mean of our distribution increases toward the right and the standard deviation increases to include the wide range of points that are more likely to be part of the green cluster. Meanwhile, the magenta distribution remains almost unchanged due to the fact that not many points fall close to the magenta distribution.

When it comes to using this method for background subtraction, we start by separating our video into three parts: the red, blue, and green channels. We then model each channel as a bimodal Gaussian distribution; one which represents the foreground and the other which represents the background. Once we have our model, we then use the most weighted cluster as our background and the least weighted as the foreground. In videos where objects such as vehicles are not moving for long periods, this methodology may mistake a vehicle as part of the background, but by using longer duration videos this type of error in background detection can be minimized.

Chapter 3: Challenges in Traffic Camera Footage Image Segmentation

3.1 Projection of 2D Images to 3D

One of the most difficult problems in traffic engineering is being able to accurately measure the distance from the traffic camera. The distances of objects seen within camera footage is a necessity if we are to accurately measure how far apart objects are (such as pedestrians and vehicles) as well as accurately measuring the speed at which cars are traveling (Dubská, 2014; Sochor, 2017; Kumar 2018; Sochor, 2018; Lu 2020). The problem of determining the 3D coordinates from a 2D image is known as a perspective-n-point problem. For this reason, many papers have already implemented methods to estimate camera parameters. The parameters of the camera are the intrinsic parameter matrix, as well as the rotation and translation matrix (Hartley, 2004; Sochor, 2018; Wang, 2018). The intrinsic parameter matrix consists of the focal length, the skew (which is normally assumed to be zero), and the principal point (the center point of the cameras view). The camera calibration process is one of the most important as it will influence the accuracy of any distance and speed measurements that are estimated. The way this problem is formulated mathematically is given by $\boldsymbol{P} = \boldsymbol{K}\ [\boldsymbol{R}\ \boldsymbol{T}]$ where P is the projection matrix, K is the matrix of intrinsic parameters, normally written as

$$K = \begin{bmatrix} f_x & s & u_o \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}$$

Where f_x and f_y are the focal lengths, s is the skew, and (u_o, v_o) is the principal point. If we assume a zero skew (Lv, 2006; Krahnstoever, 2006; Wu, 2007; Sochor, 2018; Wang, 2018) then $s = 0$ and the focal lengths will be equal as well: $f = f_x = f_y$. Additionally, R is the rotation matrix, and lastly, T is the translation matrix.

One solution to this problem is simply to first know our camera's parameters and then to measure the real-life distance that the camera's view spans therefore giving us the perspective within a 2D image. This methodology, however, is not feasible on a large scale and so we require a more rigorous approach that allows us to first estimate the camera parameters.

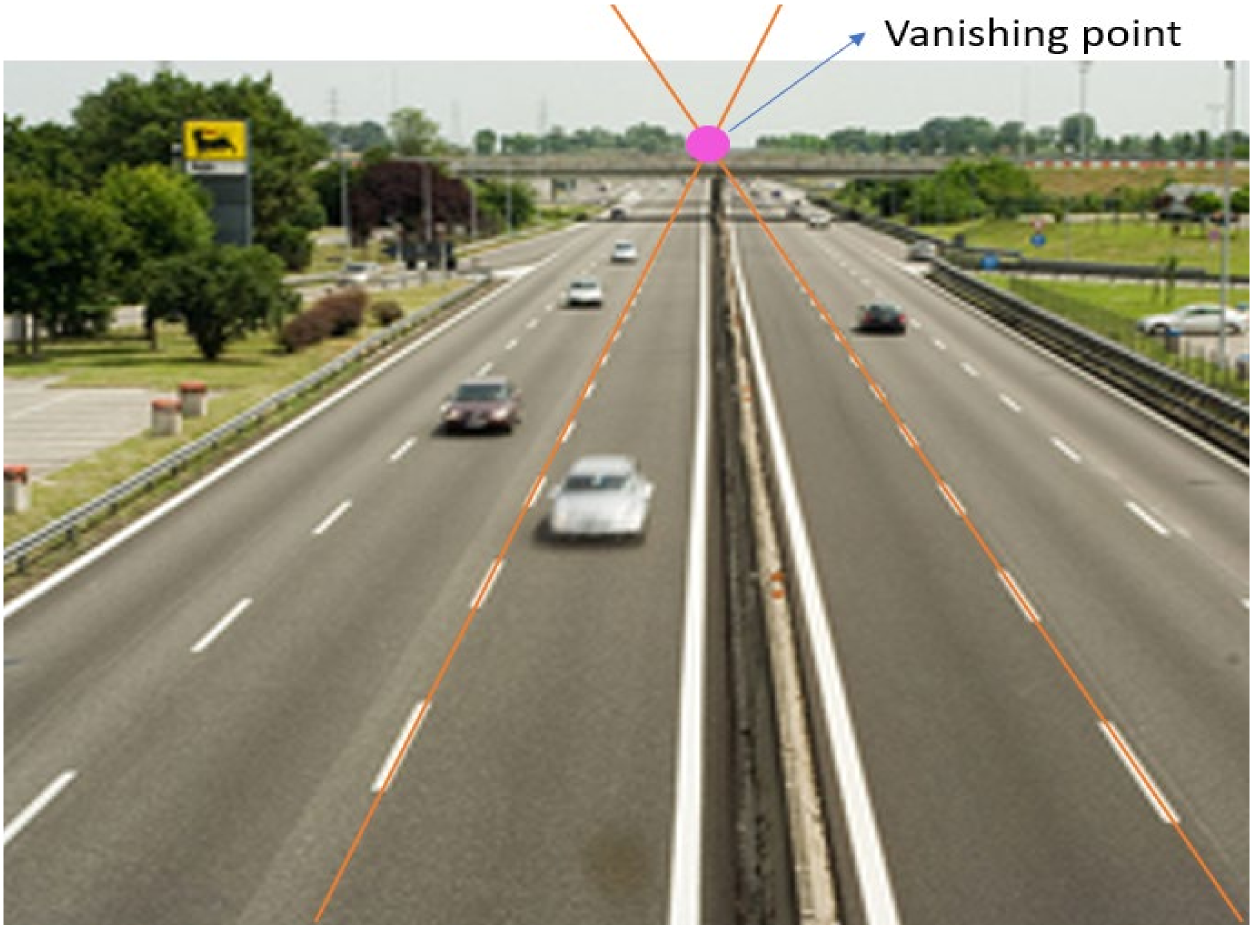


Figure 4: depiction of a vanishing point. The intersection of the two lines is what is known as a vanishing point and gives the perspective of a 2D image, allowing us to then find 3D coordinates.

Various methodologies have already been developed; however, the majority of the existing techniques require some manual measurements that are specific to a particular camera mounted on a specific road. Hence, these techniques are not generic and not automatically applicable to every road. For example, the work of Cathey et al. (2005) requires the use of finding a vanishing point using lane marks. A depiction of a vanishing point is given in **Figure 4**. But it is also necessary to know the width of the lane markings to find the scale factor using their proposed camera calibration technique. Similarly, techniques employed by Grammatikopoulos et al. (2005), He et al. (2007), You et al. (2016), Kumar et al. (2018), Huang (2018), and Tran et al. (2018) each require using lane markings to detect one or more vanishing points in an image. The potential difficulty with this approach is that the lane markings on the roads might not be clearly visible in camera footage due to construction or simply old age of the markings. Additionally, in the case of You et al. (2016) a measure of the height of the mounting point of the camera with respect to the ground is also needed for proper scaling in the camera calibration process. In the case of Kumar et al. (2018), it is also required to know the lane width to determine the proper scale factor.

To circumvent any issues that may arise from using lane markings, other works have been proposed which use vehicle movement to determine the vanishing points within an image (Dailey, 2000; Schoepflin et al., 2003; Zhang, 2008; Dubská et al., 2014; Filipiak et al., 2016).

Nevertheless, there are still limitations to many of the existing proposed methods that make them less feasible for a large-scale implementation. For example, Schoepflin et al. (2003) require that at least one known measurement be input manually for each camera in order to infer the scale of the image, Zhang et al. (2008) require the knowledge of the height at which the camera is placed to infer the intrinsic and extrinsic camera parameters. Similarly, the method of Filipiak et al. (2016) can only be used with ANPR cameras which are cameras placed in such a position that they can only view very small portions of the road in order to enhance the ability of plate recognition.

To our knowledge, the only work that is both fully automated and can be used for any viewpoint is that of Dubská et al. (2014) as described also in Sochor et al. (2018), which relies on acquiring two vanishing points. This methodology will be described in this report as it is needed for accurate speed measurements. This methodology first begins by using a foreground detection model to find any movements within the images. The use of foreground detection is important for any problem which requires an algorithm to detect changes over time in an image. From here, it is possible to create the bounding boxes of cars through the use of a min eigenvalue detector, KLT tracker, and line-to-line Hough transformation (Dubská et al., 2013; Dubská et al., 2014). A depiction of such boxes is given in panel **a)** of **Figure 5**.

Once they have found the bounding boxes of the vehicles, it is then possible able to create a grid representing the perspective of the image and find the two vanishing points corresponding to the "x" and "y" coordinates of this plane (**Figure 5. Panel b**). From this point on, the act of measuring the speed is quite simple as we now know the true distance the vehicles are traveling (with some error since camera calibration will not be perfect) and we also know the length of time for any video recorded.

(a) (b)

Figure 5: a) a schematic of the bounding box created by the method of Dubská et al. (2014) b) Schematic of how the bounding box can be used to find create a grid in order to find 2 vanishing points.

3.2 Wildlife Interaction With Vehicles

As the number of vehicles on the road increases and as the road is constructed through various natural habitats, the number of vehicle-wildlife collisions (VWCs) has also increased, making VWCs a major safety problem (Hughes et al., 1996). In order to reduce such incidents, various measures have been implemented. First of such methods is the construction of overpasses and underpasses (Dodd et al., 2004; Tissier et al., 2016). For this, the distribution patterns of the VWCs have to be initially determined (Puglisi et al., 1974; Krisp and Durot, 2007; Ramp et al., 2005, 2006; Mountrakis and Gunson, 2009). Based on this information, the location for the construction of such mitigation measures is then implemented. These overpasses and underpasses have been relatively effective in reducing such VWCs (Clevenger and Waltho, 2000; Foster and Humphrey, 1995; Dodd et al., 2007; Mata et al., 2008).

However, the construction of such features i.e. wildlife-crossing structures is not feasible in most areas. Another method that is commonly used is to put up traffic signs so as to alert the driver about potential wildlife encounter ahead. However, the efficacy of this method has been questioned in some studies (Pojar et al, 1975). For example, motorists respond to the signs by reducing vehicle speed, however, the amount of speed reduction is too small to be of practical importance. Traffic cameras have been also extensively used for wildlife crossing detection (Ford et al., 2009), although, during the nighttime or when the sunlight is directly facing the camera, this method might not work properly. This limitation of traditional cameras while detecting the presence of wildlife can be mitigated by the use of thermal cameras (Zhou et al., 2012, Christiansen et al., 2014). One of the algorithms that can be used to detect wildlife using thermal cameras is by using "counter-based pattern recognition" (Zhou et al., 2012). In addition to thermal cameras, sensors have been also utilized in determining the presence of wildlife near roadways. Typically, two types of sensors are used: "area cover sensors" or "break-the beam sensors" (Huijser et al., 2006). Area cover sensors detect the presence of large animals within the working range of the sensor either by "only receiving signals" or by sending "a signal over an area" and measuring "its reflection." (Huijser et al., 2006). On the other hand, break-the-beam sensors consist of a transmitter, which sends out laser or microwave signals, and a receiver. It then detects the presence of wildlife when their body interferes with this continuous stream of signals (Huijser et.al., 2006). Regardless of the sensor type, warning signs get activated as soon as animals are detected. This will then alert the drivers to reduce speed and watch for animal encounters, thus playing an effective role in reducing VWCs (Huijser et al., 2006).

Figure 7: Example of the use of the sensor for mitigating VWCs. (Photo: Robert Weinholzer, MNDOT)

Recently, state-of-the-art technologies have also been implemented to mitigate VWCs. One such technology is via light detection and ranging (LiDAR) technology. For instance, LiDAR has been used where the data processing procedure included background filtering, object clustering, and object classification (Chen et.al., 2019). The method is reportedly able to detect deer as far as approximately 30m from the installation location of the LiDAR (Chen et.al., 2019). Another recent technology that is being used is artificial intelligence (AI). Specifically, AI is used in conjunction with other systems for wildlife monitoring. For instance, Gonzalez et al. (2016) use AI in order to process images from thermal cameras for monitoring wildlife. In the field of artificial intelligence, the use of deep learning is becoming more prominent for wildlife detection. Deep learning is specifically used for object detection in images taken via cameras. There are different types of deep-learning-based object detection techniques and tools such as Faster R-CNN, YOLO, which are capable of identifying animals for which the models have been trained. For instance, Schneider et al. (2018) show a comparison of two such deep-learning-based object detection algorithms: R-CNN and YOLO using images taken from camera traps. Bonneau et al. (2020) use a time-lapse camera along with YOLO to detect wildlife. In another example, Sato et al. (2021) used YOLO which provided different signals corresponding to different types of wildlife detected, thus showing the potential to embed it as an alert system near roadways to reduce VWCs.

3.3 Debris Detection

Vehicle-debris encounters (VDEs), although uncommon, are costly. There have been numerous attempts at solving this problem. Nevertheless, several outstanding problems still remain to be solved, such as the effective detection of small objects on roads and the detection of objects in poor environmental conditions (Loce et al., 2013). Another challenge in detecting road anomalies is due to the poor quality of the conventional datasets used for object detection (Creusot & Munawar, 2015; Chan et al., 2021). This section provides an overview of the relevant studies on this topic.

Creusot & Munawar (2015) discuss a methodology for detecting any object that is not part of the road by introducing a new methodology that they name as "patch appearance reconstruction". This methodology for debris detection on roads uses a compressive Restricted Boltzmann Machine (**RBM**) trained exclusively to reconstruct the road which is an Autoencoding Neural Network which the authors have trained specifically for the task of road reconstruction. Their proposed methodology is in response to the lack of studies attempting to detect anomalous objects at very high speeds which is a difficult task. Their use of an RBM is mainly due to the quick execution of their online pipeline without any additional stress put on the hardware, making it the ideal choice when attempting to find anomalies in real-time. The process can be summarized in three steps: 1) the preprocessing stage, 2) offline training, and lastly 3) the online pipeline. First, in the preprocessing stage, they find a video of the road and manually choose a portion of the video to view. From here they then perform their offline training: this part requires feeding the image to the RBM and retrieving the weights and saving it for the online pipeline. They then define a reconstruction vector and attempt to determine the parameters (weight matrix, hidden/visible biases vectors) which minimize the sum of the squared residuals of the reconstruction to identify the best-fit parameters. For the online pipeline, they feed the normalized image into the RBM with the weights calculated in the offline mode to create the reconstructed image. Finally, they calculate the difference between the reconstructed and true image to find any anomalies.

In a more recent work by Lis et al. (2020), the authors propose a method of detecting anomalies by inpainting patches of an image and then comparing the original patch to the inpainted version via a discrepancy network in order to detect differences i.e., anomalies. The previous sentence essentially summarizes the two-step approach that the authors suggest. The authors begin their method by first using an inpainter from Yu et al. (2019) that is trained on a scene recognition dataset named Places2 which works to make the inpainted patches look as realistic as possible so that the road may be later reconstructed. However, the authors also note that unless the object is entirely enclosed within a patch, the anomalous object will also be recreated in the inpainted version which is an unwanted outcome. To solve this issue, the authors suggest using patches that have an overlap of 0.7 to increase the possibility that an object will be enclosed by at least one patch. Next, the authors describe their discrepancy network whose purpose is to differentiate between any differences between the original image and any artifacts left behind by the inpainting process, however, this network must first be trained. For their work, the authors choose the synthetic dataset of Cityscapes (a common dataset used for assessing how well a program performs at a given task for an urban scene). Once the discrepancy network has been trained it is then fed into a Visual Geometry Group (**VGG**) CNN model for extracting features. Once the features are extracted they are then passed to an upconvolutional pyramid to create a heatmap of where anomalies exist in the image. After testing this methodology using the Fishyscapes benchmark, it was found to be the best performing in average precision, but poorly in their false positive rate (**FPR**) except for in one case. The authors explain that this poor score in FPR is due to road boundary detection algorithms and that this score will increase over time as these algorithms improve.

One drawback of two aforementioned methods is that they are both implemented for autonomous vehicles and have not been tested on mounted cameras at far distances such as those used in live traffic footage. The subject of detecting obstacles from live traffic footage also seems to be a field left largely unstudied. Additionally, the work of Lis et al. (2020) also criticizes the method used by Creusot & Munawar (2015) for being limited to roads that are not considered “textured”. For these reasons, we have decided to construct our own methodology to be tested and is described in chapter 5 of this report.

Chapter 4: How to Detect and Track Objects?

For any problem in which we are trying to classify an object such as a pedestrian, animal, or vehicle we need some type of architecture that will be able to classify such an object using deep learning techniques such as CNNs. The idea of object detection truly took off after the success of region proposal method given by R-CNN (Girshick et al., 2014). The initial implementations were very slow. However, over a short time span, new significantly improved algorithms and implementations emerged. For the proposes of this project, one could create a new dedicated CNN model. Given the extensive set of available tools and pretrained models and the variety of options for use such as R-CNN, Fast R-CNN, Faster R-CNN, and YOLOv4, we have tested some of the best implementations for suitability for usage in this work. We are specifically interested in a pretrained algorithm that can perform real-time image segmentation and object detection. This narrows down our choices to specifically two implementations: Faster R-CNN and YOLOv4. Additionally, we want to be able to track the object through time, so we also use YOLOv4-deepsort found on the GitHub page associated with the account name "theAIGuysCode".

4.1 Faster R-CNN

The Faster R-CNN method begins by defining a region proposal network (**RPN**) by dividing the image into multiple proposal areas where an object could possibly be found. This works by creating what is known as an anchor/bounding box which are simply boxes of different sizes, such as a wide rectangle or a tall and wide rectangle, etc. The reason for this is to attempt to detect an object of any size located within an image. Then a method is defined for deciding if an object will be detected or not using an intersection over union (**IOU**). This method looks at the bounding boxes, one being a prediction and the other being the true bounding box. If the percentage of overlap is above some minimum threshold, an object is detected, otherwise, it is not detected. Finally, the bounding box with the greatest percentage will be considered the true bounding box and therefore, part of the foreground. Any area where there is no object is considered part of the background. This part of object detection will be done using CNNs, giving an output of features. From here, any object within the foreground class then continues on to the next layer known as the region of interest (**ROI**) pooling. We feed different sizes of feature maps extracted from the previous step into the ROI to reduce any feature maps to the same size matrices, essentially what is being done here is that we are putting this ROI into a Fast-RCNN with pooling, fully-connected layers, and lastly a softmax layer regressor for the bounding box.

4.2 YOLOv4

The Yolov4 algorithm begins by an architecture breakdown i.e. the backbone, neck, and head (Bochkovskiy et al, 2020). The backbone refers to the type of network which we want to feed our data into, such as a VGG network, Darknet, etc. in order to extract features from an image. The neck section tries to increase the ability to discriminate between different features using tools such as a feature pyramid network (**FPN**). Lastly, the head is the section that handles the prediction and in the case of YOLOv4 uses a one-stage process, otherwise known as a dense prediction. Let us next talk about the "Bag of Freebies" which is a method which only changes the training strategy or increases the training cost which gives us an improvement in the detection of objects essentially for free without any extra work. Next, Bochkovskiy et al. (2020) discuss the architecture selection criterion. The objective of this step is to find a balance between the input image size, the number of convolutional layers, and the number of filters for feature extraction. Also, we want a very good "neck" to aggregate features from the backbones. The architecture of YOLOv4 is as follows: the backbone is CSPDarknet53, the neck is SPP+PANet, and lastly, the head is YOLOv3. Beginning in the backbone, YOLOv4 uses what is referred to as a cross-stage partial (CSP) connection to remove duplicate gradient flow information that is present with DenseNet and therefore the accuracy and speed of YOLOv4 is increased. The spatial pyramid pooling (**SPP**) method allows changing of the input size using different sized max pool layers. Overall, given the speed to accuracy ratio, YOLOv4 seems to outperform nearly every method on Volta and Pascal GPU architectures, however, falls short somewhat using the Maxwell architecture in the number of frames per second, but outperforms other architectures in accuracy.

4.3 Tracking With DeepSORT

DeepSORT is a powerful method that expands upon Simple Online Real-time tracking (SORT) and allows for the tracking of multiple objects. For this section, we will be using the methodology as described in the paper of Kumar et al. (2021). The authors first start by splitting their problem into three parts: object detection, tracking of objects, and lastly zonal counting. For object detection, we are using the YOLOv4 algorithm which comes with pre-trained weights and whose algorithm is discussed in section 4.2. For the tracking of objects, a crucial component is that of Kalman filters which is a filter comprised of a distribution for sensor confidence and for motion confidence. When an object is in full view, more weight will be put on the sensor confidence and when the view of an object is obstructed, we put more weight on the motion confidence. To better understand the DeepSORT algorithm, we begin by estimating the position of an object given its previous position, in this step we care only for the spatial information. We then extract a feature vector or appearance descriptor which describes features detected within an image by comparing the current frame to the previous frame. This feature vector is a trained CNN that extracts features such that features for differing objects appear to be far apart but features for the same object are close together. This is done by using what is known as the Mahalanobis distance (a non-Euclidean distance measurement). Then, any new detection from incoming frames is connected to the previous tracks using the feature vector and the estimated position based on the last frame assuming that the confidence level is above the threshold. The last section, zonal counting, is unrelated to DeepSORT and therefore is not discussed here.

Chapter 5: Establishing A Workflow for Automated Analysis of Traffic Camera Footage

5.1 Data Collection for the Analysis of Wildlife Vehicle Encounters

We focused on deer-vehicle collision sites. Accordingly, we searched the available online databases of traffic camera footage for sites with the highest number of deer-vehicle collisions (DVCs). Based on the research by StateFarm[1] we found out that West Virginia had the highest number of deer-vehicle collisions. Consequently, we identified the website 511 (http://wv511.org) in West Virginia, which provides live traffic camera footages. This website provides many traffic camera footages in West Virginia, many of which do not necessarily contain high rates of and live cases of DVC. Since the analysis of all videos were nearly impossible, we focused our attention on the locations where there was already a high likelihood of seeing deer-vehicle encounters.

Our extensive searches led us to a report by Nichols et al. (2014), submitted to the Department of Transportation of West Virginia. This report summarizes data from 2008-2012 on crashes related to deer and carcass. Nichols et al. (2014) categorize the frequency of DVCs according to some county and provide the location of such collisions on the maps. We used this information about the locations of incidents to identify potential live traffic camera footages for subsequent collection and analysis. One major challenge, however, turned out to be the far distance of the traffic camera from the exact location where most DVCs are reported. Hence, we performed a secondary comprehensive search among all available traffic camera footages within the State of West Virgnia and finally identified a traffic camera site which was very close to a reported DVC site. Unfortunately, the latter camera site was also far from ideal as it only captured a certain part of the collision site. We monitored and collected the candidate camera for extended periods of time throughout this project. However, all of our search efforts for detecting deers or other animals in the collected frames were unsuccessful. As an alternative, we collected some pre-recorded camera footages publicly available from other countries, specifically, Japan, that we subsequently used for testing.

Our method to mitigate Vehicle-Wildlife Collisions (VWCs) is to analyze and keep a record of certain live traffic camera footages using YOLO where the animals were detected. Such recorded video segments can be collected over time to create a map of spatial presence of animals on roads which could be then used to identify potential animal corridors and take the necessary VWC mitigation steps in those areas. The lack of continuous reliable stream of data from potential sites of VWCs, significantly limited our ability in successfully implementing and testing the proposed pipeline. However, we were able to test the fundamental concepts of the proposed pipeline via pre-recorded footages. An example of animal detection in one of our pre-recorded videos is given in Figure 8. Although the algorithm is capable of identifying the object as an animal, it fails to recognize the animal (deer) correctly. The fact that the algorithm labels a deer as a cow in the given

---

[1] AASHTO Journal. 2018. *State Farm Survey: Deer-Vehicle Collisions Dropping, But Costs Rising*. [online] Available at: <https://aashtojournal.org/2018/10/12/state-farm-survey-deer-vehicle-collisions-dropping-but-costs-rising/> [Accessed 21 October 2020].

frame highlights the importance of continuous training of Deep Learning models with more and more labeled data to improve their accuracies.

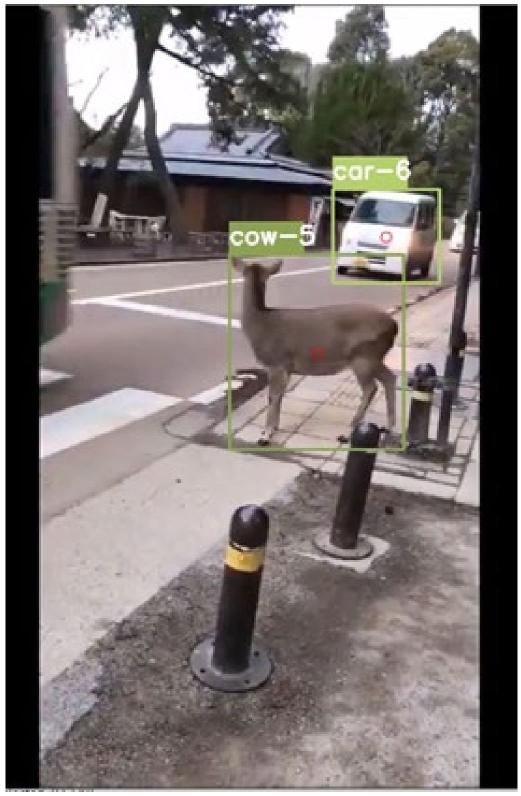


Figure 8: An example of animal detection using YOLO

The workflow setup created to achieve the goal is illustrated in the following chart[2].

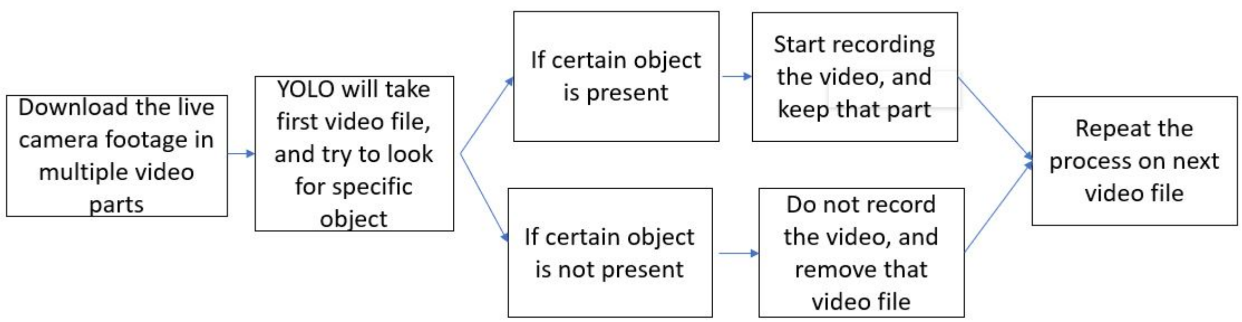


Figure 9: Flowchart for the overall working process of the algorithms developed in this project.

In brief, a python script downloads and stores a 10 seconds video camera footage from the traffic camera website. The downloaded stream is subsequently passed to the YOLO algorithm for potential object detection. If any object is detected, the program will start recording and generating an output video file containing the image segmentation analysis. If no object is detected, the program will discard the downloaded footage and no output analysis file is generated. The process

[2] https://github.com/cdslaborg/traffic

is then repeated with the next 10 seconds of downloaded video and continues for as long as desired. In summary, if a certain object is detected within a particular 10 seconds video file, the python script generates two video files: one is the original 10 seconds file, and the other file, which will contain the labeled objects in the video as the output of the analysis. However, if no object is detected, then there will be no files since there is no labeled object in the image and the original video file is also deleted.

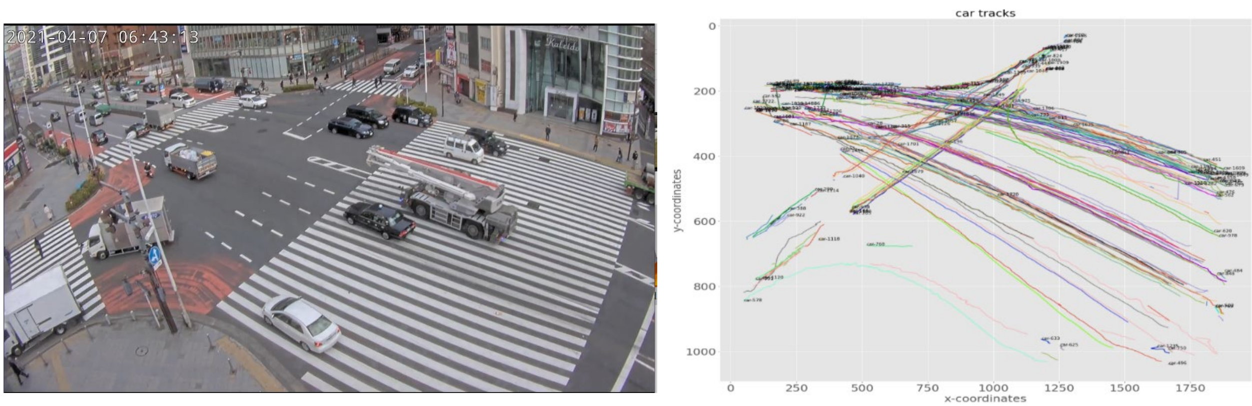
5.2 A Pipeline for Detection of Road Debris and Anomalous Driving Patterns in Roads

In this section, we briefly describe the approach we adopted for detecting debris in roads, and more generally, for detecting anomalous patterns of driving in highways and streets. Road debris detection is an extremely challenging task given the existing camera technologies and image segmentation algorithms.

The difficulty in road debris detection stems from the lack of a particular shape for road debris, making them extremely difficult to detect for the existing Deep Learning algorithm. As such, our method of road debris detection relies on indirect methods of searching for anomalous vehicle behavior in roads. Here, anomalous behavior is defined with respect to the behavior of the majority of vehicles in the same road in the past. For example, if a debris suddenly blocks a lane in a highway, most vehicles will try to avoid the debris by changing lanes or significantly reducing their speed. This sudden change in vehicle behavior does not necessarily imply the presence of road debris, however, it can be a clear indication of the presence of an anomaly in the road, for example, an intoxicated driver. Therefore, the above approach is generic, and it can aid automatic detection of strange behaviors and deviations from formal routes of driving in roads.

The essence of the idea that we pursue here is that by continuously collecting traffic camera footage, we can identify the normal routes of transportation in camera footages such that any vehicle tracks that deviates from the normal routes (taken by the majority of vehicles) triggers an alert for a possible anomalous behavior on the road. Such anomalies could be due to drivers suddenly changing lanes because of the presence of road debris, a crash, or simply imply the presence of a careless driver.

(a)



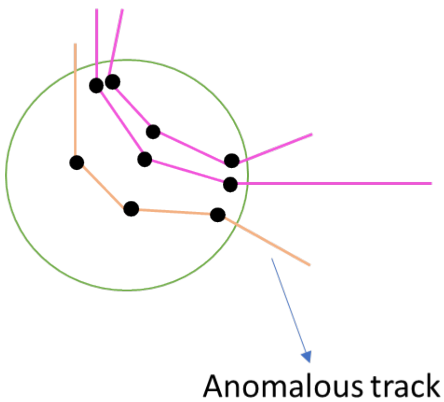
(b)

Figure 10: a) An example of vehicle track detection performed on a road in Seattle on the left. The corresponding tracks of cars are plotted using the generated CSV file. b) An image of live traffic footage from Tokyo shown on the left on which we performed our analysis. On the right we have the tracks inferred from the segmentation of the traffic camera footage.

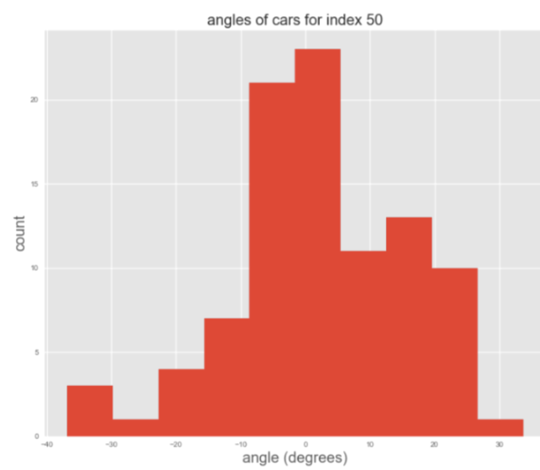
To achieve the goals, a script was written to automatically parse the camera footage contents from multiple city websites across the United States which held the live videos. The script automatically clicks on "show video" button in the websites and then extracts the link to the camera by inspecting the page and looking for relevant links. Then, we import the VLC media into python to download the live camera footages to the local system, first checking if the video is playing and the link is not dead. If the live feed is dead, then the script moves on to the next live traffic feed. After recording the live traffic videos, we then reduce the camera footages to coordinates of vehicles in the frames of videos using the yolo-deepsort algorithm. The reduced data is significantly smaller in size than the original camera footages and is in comma-separated format (CSV). It summarizes the movements of all objects in a video stream into a set of adjacent points each set of which represents the trace of a single object in the footage. These objects could be pedestrians, cars, trucks or other types of vehicles or animals. Once the objects are classified, a particular category, for example, vehicles can be selected for further studies.

As an example of this process, we can look at **Figure 10**. In panel (a) we are looking at publicly available live traffic feed from Seattle, Washington on the left. We record this camera footage for a set amount of time, save it, and then analyze the video via yolo-deepsort algorithm. An output file is then generated from this analysis, giving the coordinates for each pedestrian, car, and truck in the video. From this we can then plot the tracks inferred from the camera footage as illustrated on the right plot of **Figure 10,** panel (a). Once this summary output file containing the object tracks is created, there is no longer a need to keep the original video and it is deleted to save storage space. **Figure 10,** panel (b), illustrated the same process applied to a different live camera footage from Tokyo, Japan.

A major challenge arises at this point in the analysis: First, the temporal analysis of the footage in search of anomalies relies on measuring both temporal and spatial deviations form normality. Examples of such deviations are sudden speed reduction or sudden change of lane, respectively. Both of these require a measure of scale in the frames and the vanishing point in the camera footages. This information is essential since a significant deviation that happens far from the camera would not be detectible, whereas a small deviation from the normality would be flagged as an anomalous pattern of driving or motion.



(a)



(b)

*Figure 11: **a)** A schematic illustration of two paths (magenta) which are labeled as normal paths that most cars take due to their similarity and proximity, versus an anomalous path (orange track). As seen, the two magenta lines tend to follow a very similar path, only slightly deviating from one another, thus forming very similar angles formed by the adjacent points in each track. Conversely, the orange track deviates largely from the others, which could be an indication of the presence of an anomaly in the road, such as road debris. **b)** An illustration of the angles formed by adjacent points in each track in an example camera footage. Any point which lies three standard deviation outside of this distribution is labeled as a potentially anomalous behavior.*

For the purposes of spatial analyses, specifically, the perspective and vanishing point can be resolved if we instead focus on anomalies in the angles of motions instead of anomalies in object displacements. This idea is illustrated in **Figure 11**. First, we look at 3 points on a certain path and calculate the angle formed by the two vectors that connect the three points. We then draw a circle with a radius that encompasses all three points, similar to what is seen in panel (a) of **Figure 11**.

When at least 2 points of another trajectory fall within an already existing circle, we include it as part of the distribution of angles for that neighborhood, otherwise, we draw a new circle to encompass these new points. Now, for each neighborhood (corresponding to one circle) we will have a distribution of angles with some mean and standard deviation for each circle drawn (as seen in panel b of **Figure 11**). From here we then determine the angles formed by the points of new car trajectories traveling through each circle. Next, we determine if the newly calculated angles fall within three standard deviations of the distributions for the given circle it has passed through. At this point, if the angle is over three standard deviations, we would then classify this as an anomaly, otherwise, it would be considered a normal movement (a depiction of a normal and anomalous track is shown in **Figure 11** panel a). If multiple anomalies happen within the same area, within a small time period then this would then potentially indicate the presence of some obstacle or debris in the road which vehicles are trying to avoid.

Nevertheless, challenges remain regarding accurate measurement of anomalies in vehicle speeds. Such measurements require the identification of the unique vanising points for each traffic camera separately. Although there exists a body of work on this topic, the existing methods have severe limitations that would make a commercial implementation of this anomaly detection methodology rather challenging.

5.3 Data Management

Another potential challenge towards commercialization of the use of Deep Learning techniques for analyzing traffic camera footage is the setup of an automated pipeline for fast real-time data acquisition and reduction from cameras. This project heavily relies on the use of big data to infer insights into anomalous patterns in traffic. Although the input data can be on the order of Megabytes per second, it can be rapidly analyzed and subsequently discarded or reduced after the analysis and stored in a device for future reference.

For the goals of this project, we realized that the storage and computational capacity of a regular modern desktop computer was sufficient for prototyping and testing. However, for future larger scale commercial implementations of such techniques, the use of a large, dedicated filesystem for safe long-term storage of such data would be essential. The infrastructure for such activities already exists, for example, at Supercomputing centers like the Texas Advanced Computing Center. Partnership between cities and supercomputing centers could be established for setting up automated pipelines that collect data directly from traffic engineering centers, reduce it, analyze it, and store the results securely for long-term usage and inference.

Chapter 6: Discussion and Conclusion

Efficient detection and prediction of the locations of Vehicle-Wildlife Encounters (VWEs), Vehicle-Debris Encounters (VDEs) as well as Vehicle-Pedestrian Encounters (VPEs) have long been considered as challenging unresolved problems in traffic engineering. Such encounters are costly as they are frequently followed by an incident. Identifying the root causes of these encounters and resolving them could reduce the frequency of such events and the associated costs.

This project explored the idea of using traffic camera footages for automatic detection of such close encounters via artificial intelligence in the hope that the potential underlying causes could be identified more efficiently, thus reducing the environmental, societal, and financial costs associated with such incidents.

In this work, we explored the existing literature on the approaches to VWE, VDE, and VPE detection. Common techniques used in deep learning for image analysis such as the basics of CNNs and background subtraction were also studied and detailed. Challenges for efficient accurate image segmentation of traffic camera footages were also identified and discussed.

A major challenge in image segmentation of traffic camera footage is the determination of the 3D perspective from 2D camera images. Although there is a vast amount of research on this topic, the majority of the existing techniques require some level of manual input to the algorithms, such as the camera height, lane width spacing, width of lane markings, etc. The feasibility of these methodologies, therefore, dramatically decreases with the scale of projects that deploy such techniques since each camera would have to have its own uniquely tuned image segmentation algorithm. Furthermore, there is no guarantee for the lack of a need for subsequent tuning of the algorithms as the camera settings and locations can change over time.

Similarly, there seemed to also exist a significant number of studies on the topic of road debris detection. The majority of these methods, however, use Radar or LiDAR technologies on autonomous vehicles. Consequently, these techniques are not applicable to the detection of road debris in traffic footage collected by live static cameras.

A viable approach to automatic real-time road debris and anomaly detection is the state-of-the-art object detection algorithms. Based on tests that we have performed as part of this project, we found that algorithms such as YOLOv4 offer a fast method of object detection on roads without sacrificing precision to a great degree. Such algorithms (pre-trained with related labeled traffic images) could be combined with other algorithms (e.g., DeepSORT) to reconstruct the tracks of different moving objects in traffic camera recordings, thus reducing the entire visual data to sets of points that require orders of magnitude less storage space without sacrificing any accuracy or precision in inferences or detection of anomalies on roads. This resolves the problem of data storage one would envisage for this project, even at a larger-scale commercial implementation.

Two challenges remain outstanding with the use of pre-trained algorithms for automatic analysis of traffic camera footage: 1) automatic determination of perspective from 2D images, which is essential for automated detection temporal and spatial anomalies in traffic (such as sudden change of speed or change of lane among a single or a group of cars, 2) improving the accuracy of inferences, which requires continuous training of the models with newly labeled traffic data.

Additionally, several other issues must be resolved to ensure the scalability of the proposed methodologies and ideas in this project on a larger scale. Most importantly, traffic cameras on the road frequently vibrate. These vibrations could be due to wind or due to passing of heavy machinery from nearby. This is particularly the case for cameras mounted on bridges. Although the existing algorithms are capable of accurately detecting objects within individual frames even in the presence of significant vibrations, such movements present a challenge for anomaly detection discussed in this project as these techniques require registration and matching of numerous snapshots of the same location monitored by a single camera footage. Rapid significant vibrations can easily change the background landscape viewing angle of camera, potentially making image registration impossible. Further studies are required to quantify the extent of the impact of camera vibrations on the utilities of the methodologies discussed in this project.

Another major issue that remains to be solved is the analysis of camera footage collected at nighttime. However, we consider this problem more of a technological, as opposed to algorithmic, issue which could be potentially resolved with the arrival of better technologies for nighttime surveillance at scale on roads and highways.

In conclusion, we set up a method for retrieving live traffic camera footage, from different areas throughout the country that seemed to have the highest-quality camera recordings. Ideally, we aimed for camera footages from with Dallas-Fort Worth Metroplex. However, our initial searches for high-quality camera footages from DFW, comparable to those found from other parts of the country, were fruitless. Our preliminary studies and experimentations exhibit the feasibility of the proposed approaches to automatic detection of traffic anomalies via traffic camera footages, although several outstanding issues (discussed in the above paragraphs) remain to be resolved.

References

Allen, Ross E., and Dale R. McCullough. "Deer-car accidents in southern Michigan." The Journal of Wildlife Management (1976): 317-325.

Avelar, Raul Eduardo, et al. Develop metrics of tire debris on Texas highways: technical report. No. FHWA/TX-16/0-6860-1. Texas A&M Transportation Institute, 2017.

Barth, Matthew, and Kanok Boriboonsomsin. "Real-world carbon dioxide impacts of traffic congestion." Transportation Research Record 2058.1 (2008): 163-171.

Beevers, Sean David, et al. "Traffic management strategies for emissions reduction: recent experience in London." Energy and emission control technologies 4 (2016): 27-39.

Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." arXiv preprint arXiv:2004.10934 (2020).

Bonneau, Mathieu, et al. "Outdoor animal tracking combining neural network and time-lapse cameras." *Computers and Electronics in Agriculture* 168 (2020): 105150.

Cathey, F. W., and D. J. Dailey. "A novel technique to dynamically measure vehicle speed using uncalibrated roadway cameras." IEEE Proceedings. Intelligent Vehicles Symposium, 2005.. IEEE, 2005.

Chan, Robin, et al. "SegmentMeIfYouCan: A Benchmark for Anomaly Segmentation." arXiv preprint arXiv:2104.14812 (2021).

Chen, Jingrong, et al. "Deer crossing road detection with roadside LiDAR sensor." *Ieee Access* 7 (2019): 65944-65954.

Christiansen, Peter, et al. "Automated detection and recognition of wildlife using thermal cameras." *Sensors* 14.8 (2014): 13778-13793.

Clevenger, Anthony P., and Nigel Waltho. "Factors influencing the effectiveness of wildlife underpasses in Banff National Park, Alberta, Canada." *Conservation Biology* 14.1 (2000): 47-56.

Concas, Sisinnio, and Mohsen Kamrani. "Development of a Real-Time Roadway Debris Hazard Spotting Tool Using Connected Vehicle Data to Enhance Roadway Safety and System Efficiency." (2019).

Creusot, Clement, and Asim Munawar. "Real-time small obstacle detection on highways using compressive RBM road reconstruction." 2015 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2015.

Dailey, Daniel J., Fritz W. Cathey, and Suree Pumrin. "An algorithm to estimate mean traffic speed using uncalibrated cameras." IEEE Transactions on Intelligent Transportation Systems 1.2 (2000): 98-107.

Dollar, Piotr, et al. "Pedestrian detection: An evaluation of the state of the art." IEEE transactions on pattern analysis and machine intelligence 34.4 (2011): 743-761.

Dodd, Norris L., et al. "Video surveillance to assess highway underpass use by elk in Arizona." *The Journal of Wildlife Management* 71.2 (2007): 637-645.

Dodd Jr, C. Kenneth, William J. Barichivich, and Lora L. Smith. "Effectiveness of a barrier wall and culverts in reducing wildlife mortality on a heavily traveled highway in Florida." *Biological Conservation* 118.5 (2004): 619-631.

Dubská, Markéta, and Adam Herout. "Real Projective Plane Mapping for Detection of Orthogonal Vanishing Points." BMVC. 2013.

Dubská, Markéta, Adam Herout, and Jakub Sochor. "Automatic Camera Calibration for Traffic Understanding." BMVC. Vol. 4. No. 6. 2014.

Filipiak, Patryk, Bartlomiej Golenko, and Cezary Dolega. "NSGA-II based auto-calibration of automatic number plate recognition camera for vehicle speed measurement." European Conference on the Applications of Evolutionary Computation. Springer, Cham, 2016.

Fiori, Chiara, et al. "The effect of electrified mobility on the relationship between traffic conditions and energy consumption." Transportation Research Part D: Transport and Environment 67 (2019): 275-290.

Fontaras, Georgios, Nikiforos-Georgios Zacharof, and Biagio Ciuffo. "Fuel consumption and CO2 emissions from passenger cars in Europe–Laboratory versus real-world emissions." Progress in Energy and Combustion Science 60 (2017): 97-131.

Ford, Adam T., Anthony P. Clevenger, and Andrew Bennett. "Comparison of methods of monitoring wildlife crossing-structures on highways." *The Journal of Wildlife Management* 73.7 (2009): 1213-1222.

Foster, Melissa L., and Stephen R. Humphrey. "Use of highway underpasses by Florida panthers and other wildlife." *Wildlife Society Bulletin* (1995): 95-100.

Gains, Adrian, et al. "A cost recovery system for speed and red-light cameras-two year pilot evaluation." (2003).

Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.

Gonzalez, Luis F., et al. "Unmanned aerial vehicles (UAVs) and artificial intelligence revolutionizing wildlife monitoring and conservation." *Sensors* 16.1 (2016): 97.

Grammatikopoulos, Lazaros, George Karras, and Elli Petsa. "Automatic estimation of vehicle speed from uncalibrated video sequences." Proceedings of International Symposium on Modern Technologies, Education and Profeesional Practice in Geodesy and Related Fields. 2005.

Greenwood, I. D., R. C. Dunn, and R. R. Raine. "Estimating the effects of traffic congestion on fuel consumption and vehicle emissions based on acceleration noise." Journal of Transportation Engineering 133.2 (2007): 96-104.

Hartley, Richard, and Andrew Zisserman. Multiple View Geometry in Computer Vision. Cambridge University Press, 2004.

He, Xiao Chen, and Nelson HC Yung. "A novel algorithm for estimating vehicle speed from two consecutive images." 2007 IEEE Workshop on Applications of Computer Vision (WACV'07). IEEE, 2007.

Huang, Tingting. "Traffic speed estimation from surveillance video data." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2018.

Hughes, Warren E., A. Reza Saremi, and Jeffrey F. Paniati. "Vehicle-animal crashes: an increasing safety problem." *ITE journal* 66 (1996): 24-29.

Huijser, M., et al. "Wildlife-vehicle collision reduction study: report to congress (No. FHWA-HRT-08-034)." Washington, DC: US Department of Transportation (2008).

Huijser, Marcel P., Patrick T. McGowen, and Whisper Camel. "Animal vehicle crash mitigation using advanced technology phase I: review, design, and implementation." (2006).

Keall, Michael D., Lynley J. Povey, and William J. Frith. "The relative effectiveness of a hidden versus a visible speed camera programme." Accident Analysis & Prevention 33.2 (2001): 277-284.

Kinoshita, Yoshito, et al. "A Generalized Bayesian Approach for Localizing Static Natural Obstacles on Unpaved Roads." 2020 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). IEEE, 2020.

Krahnstoever, Nils, and Paulo RS Mendonça. "Autocalibration from tracks of walking people." in Proc. British Machine Vision Conference (BMVC). 2006.

Krisp, Jukka Matthias, and Sara Durot. "Segmentation of lines based on point densities—An optimisation of wildlife warning sign placement in southern Finland." *Accident Analysis & Prevention* 39.1 (2007): 38-46.

Kumar, Amit, et al. "A semi-automatic 2D solution for vehicle speed estimation from monocular videos." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2018.

Lis, Krzysztof, et al. "Detecting Road Obstacles by Erasing Them." arXiv preprint arXiv:2012.13633 (2020).

Loce, Robert P., et al. "Computer vision in roadway transportation systems: a survey." Journal of Electronic Imaging 22.4 (2013): 041121.

Lu, Shengnan, Yuping Wang, and Huansheng Song. "A high accurate vehicle speed estimation method." Soft Computing 24.2 (2020): 1283-1291.

Lv, Fengjun, Tao Zhao, and Ramakant Nevatia. "Camera calibration from video of a walking human." IEEE transactions on pattern analysis and machine intelligence 28.9 (2006): 1513-1518.

Mascia, Margherita, et al. "Impact of traffic management on black carbon emissions: a microsimulation study." Networks and Spatial Economics 17.1 (2017): 269-291.

Mata, C., et al. "Are motorway wildlife passages worth building? Vertebrate use of road-crossing structures on a Spanish motorway." *Journal of Environmental Management* 88.3 (2008): 407-415.

Mountrakis, Giorgos, and Kari Gunson. "Multi-scale spatiotemporal analyses of moose–vehicle collisions: a case study in northern Vermont." *International Journal of Geographical Information Science* 23.11 (2009): 1389-1412.

Nichols, A. P., et al. "Evaluation of deer-vehicle collision rates in West Virginia and a review of available mitigation techniques." (2014).

Pavlovic, Jelica, et al. "Dealing with the gap between type-approval and in-use light duty vehicles fuel consumption and CO2 emissions: present situation and future perspective." Transportation Research Record 2672.2 (2018): 23-32.

Peden, Margie, et al. World report on road traffic injury prevention. World Health Organization, 2004.

Piccardi, Massimo. "Background subtraction techniques: a review." 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No. 04CH37583). Vol. 4. IEEE, 2004.

Pojar, Thomas M., et al. "Effectiveness of a lighted, animated deer crossing sign." *The Journal of Wildlife Management* (1975): 87-91.

Rakha, Hesham, and Kyoungho Ahn. "Integration modeling framework for estimating mobile source emissions." Journal of transportation engineering 130.2 (2004): 183-193.

Ramp, Daniel, et al. "Modelling of wildlife fatality hotspots along the snowy mountain highway in New South Wales, Australia." *Biological conservation* 126.4 (2005): 474-490.

Sato, Denis, Adroaldo José Zanella, and Ernane Xavier Costa. "Computational classification of animals for a highway detection system." *Brazilian Journal of Veterinary Research and Animal Science* 58 (2021): e174951-e174951.

Sauber-Schatz, Erin K., et al. "Vital signs: motor vehicle injury prevention—United States and 19 comparison countries." Morbidity and Mortality Weekly Report 65.26 (2016): 672-677.

Schneider, Stefan, Graham W. Taylor, and Stefan Kremer. "Deep learning object detection methods for ecological camera trap data." *2018 15th Conference on computer and robot vision (CRV)*. IEEE, 2018.

Schoepflin, Todd N., and Daniel J. Dailey. "Dynamic camera calibration of roadside traffic management cameras for vehicle speed estimation." IEEE Transactions on Intelligent Transportation Systems 4.2 (2003): 90-98.

Shi, Honghui. "Geometry-aware traffic flow analysis by detection and tracking." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2018.

Sivaraman, Sayanan, and Mohan M. Trivedi. "A review of recent developments in vision-based vehicle detection." 2013 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2013.

Sochor, Jakub, Roman Juránek, and Adam Herout. "Traffic surveillance camera calibration by 3d model bounding box alignment for accurate vehicle speed measurement." Computer Vision and Image Understanding 161 (2017): 87-98.

Sochor, Jakub, et al. "Comprehensive data set for automatic single camera visual speed measurement." IEEE Transactions on Intelligent Transportation Systems 20.5 (2018): 1633-1643.

Szeliski, Richard. Computer vision: algorithms and applications. Springer Science & Business Media, 2010.

theAIGuysCode. “yolov4-Deepsort.” GitHub, github.com/theAIGuysCode/yolov4-deepsort.

Tissier, Mathilde L., et al. "An anti-predation device to facilitate and secure the crossing of small mammals in motorway wildlife underpasses.(I) Lab tests of basic design features." *Ecological Engineering* 95 (2016): 738-742. Puglisi, M.J., Lindzey, J.S., Bellis, E.D., 1974. Factors associated with highway mortality of white-tailed deer. Journal of Wildlife Management 38, 799e807.

Tran, Minh-Triet, et al. "Traffic flow analysis with multiple adaptive vehicle detectors and velocity estimation with landmark-based scanlines." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2018.

Treiber, Martin, Arne Kesting, and Christian Thiemann. "How much does traffic congestion increase fuel consumption and emissions? Applying a fuel consumption model to the NGSIM trajectory data." 87th Annual Meeting of the Transportation Research Board, Washington, DC. Vol. 71. 2008.

Wadud, Zia, Don MacKenzie, and Paul Leiby. "Help or hindrance? The travel, energy and carbon impacts of highly automated vehicles." Transportation Research Part A: Policy and Practice 86 (2016): 1-18.

Wang, Li, and Dennis Sng. "Deep learning algorithms with applications to video analytics for a smart city: A survey." arXiv preprint arXiv:1512.03131 (2015).

Wang, Na, et al. "Self-calibration of traffic surveillance cameras based on moving vehicle appearance and 3-d vehicle modeling." 2018 25th IEEE international conference on image processing (ICIP). IEEE, 2018.

World Health Organization. Global status report on road safety 2015. World Health Organization, 2015.

Wu, Qi, Te-Chin Shao, and Tsuhan Chen. "Robust self-calibration from single image using RANSAC." International Symposium on Visual Computing. Springer, Berlin, Heidelberg, 2007.

Xu, Weijia, et al. "Automated pedestrian safety analysis using data from traffic monitoring cameras." Proceedings of the 1st ACM/EIGSCC Symposium on Smart Cities and Communities. 2018.

Yang, Bong-Min, and Jinhyun Kim. "Road traffic accidents and policy interventions in Korea." Injury control and safety promotion 10.1-2 (2003): 89-94.

Yang, Shu, Sijia Lu, and Yao-Jan Wu. "Gis-based economic cost estimation of traffic accidents in St. Louis, Missouri." Procedia-Social and Behavioral Sciences 96 (2013): 2907-2915.

You, Xinhua, and Yuan Zheng. "An accurate and practical calibration method for roadside camera using two vanishing points." Neurocomputing 204 (2016): 222-230.

Yu, Jiahui, et al. "Free-form image inpainting with gated convolution." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.

Zhang, Zhaoxiang, et al. "Practical camera auto-calibration based on object appearance and motion for traffic scene visual surveillance." 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, 2008.

Zhou, Debao, Jingzhou Wang, and Shufang Wang. "Countour based HOG deer detection in thermal images for traffic safety." *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV)*. The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), 2012.

CENTER FOR TRANSPORTATION EQUITY, DECISIONS & DOLLARS

The **Center for Transportation, Equity, Decisions and Dollars (CTEDD)** is a USDOT University Transportation Center, leading **transportation policy research** that aids in **decision making** and **improves economic development** through **more efficient, and cost-effective use of existing transportation systems,** and offers **better access to jobs and opportunities.** We are leading a larger consortium of universities focused on **providing outreach** and **research to policy makers**, through **innovative methods** and **educating future leaders of the transportation field.**

UNIVERSITY OF TEXAS ARLINGTON

CAL POLY SAN LUIS OBISPO

Georgia Tech

USF UNIVERSITY OF SOUTH FLORIDA

WISCONSIN UNIVERSITY OF WISCONSIN–MADISON

C-TEDD@UTA.EDU CTEDD.UTA.EDU @UTACTEDD @C__TEDD @CENTER FOR TRANSPORTATION, EQUITY, DECISIONS AND DOLLARS 817.272.5138