Improving Pedestrian Safety Using Video Data, Surrogate Safety Measures and Deep Learning

Shile Zhang
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

This Doctoral Dissertation (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
https://stars.library.ucf.edu/etd2020/593
IMPROVING PEDESTRIAN SAFETY USING VIDEO DATA, SURROGATE SAFETY MEASURES AND DEEP LEARNING

By
SHILE ZHANG
B.S., Southeast university, 2016
M.S., University of Central Florida, 2020

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy 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

Spring Term
2021

Major Professor: Mohamed Abdel-Aty
ABSTRACT

The research aims to improve pedestrian safety at signalized intersections using video data, surrogate safety measures and deep learning. Machine learning (including deep learning) models are proposed for predicting pedestrians’ potentially dangerous situations. On the one hand, pedestrians’ red-light violations can expose the pedestrians to motorized traffic and pose potential threats to pedestrian safety. Thus, the prediction of pedestrians’ crossing intention during red-light signals is carried out. The pose estimation technique is used to extract features on pedestrians’ bodies. Machine learning models are used to predict pedestrians’ crossing intention at intersections’ red-light, with video data collected from signalized intersections. Multiple prediction horizons are used. On the other hand, SSMs (Surrogate Safety Measures) can be used to better investigate the mechanisms of crashes proactively compared with crash data. With the SSMs indicators, pedestrians’ near-crash events can be identified. The automated computer vision techniques such as Mask R-CNN (Region-based Convolutional Neural Network) and YOLO (You Only Look Once) are utilized to generate the features of the road users from video data. The interactions between vehicles and pedestrians are analyzed. Based on that, the prediction of pedestrians’ conflicts in time series with deep learning models is carried out at the individual-vehicle level. Besides, two SSMs indicators, PET (Post Encroachment Time) and TTC (Time to Collision), are derived from videos to label pedestrians’ near-crash events. Deep learning model such as LSTM (Long Short-term Memory) is used for
modeling. To make the model more adaptive to a real-time system, the signal timing data ATSPM© (Automated Traffic Signal Performance Measures) can be used. The signal cycles that contain pedestrian phases are labeled with the SSMs indicators derived from videos and then modeled. With the above-mentioned models proposed, the decision makers can determine the possible countermeasures, or the warning strategies for drivers at intersections.
I dedicate this dissertation to my family who have always loved me unconditionally and whose good examples have taught me to work hard for the things I aspire to achieve.
ACKNOWLEDGEMENTS

I wish to thank my committee members who have been very supportive throughout the entire process of this dissertation. A special thanks to my advisor, Dr. Mohamed Abdel-Aty, for his continuous support and patience.

I would like to acknowledge and thank my university for offering various services and giving me opportunities to conduct my research. Special thanks go to the members of our research group, UCF-SST.

My appreciation also goes out to my family and friends for their encouragement and support all through my studies.
TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION ........................................................................................................... 0
1.1 Introduction .......................................................................................................................... 0
1.2 Contributions and Objectives .............................................................................................. 2
1.3 Dissertation Structure .......................................................................................................... 3

CHAPTER 2 LITERATURE REVIEW ............................................................................................. 5
2.1 Pedestrian Safety Analysis Based on Surrogate Safety Measures (SSMs) ......................... 5
  2.1.1 SSMs Indicators .............................................................................................................. 5
  2.1.2 Traffic Conflicts between Pedestrians and Vehicles ....................................................... 7
  2.1.3 Summary ...................................................................................................................... 9
2.2 Computer Vision Applications in Transportation Field ...................................................... 11
  2.2.1 Roadside Videos from Surveillance Cameras ................................................................. 13
  2.2.2 Ego-view Videos from Onboard Cameras ................................................................... 14
  2.2.3 Aerial Videos from UAVs ............................................................................................ 14
  2.2.4 Other Video Types ....................................................................................................... 15
  2.2.5 Summary ...................................................................................................................... 15
2.3 Pedestrians’ Crossing Decisions and Red-Light Violations ............................................... 16
  2.3.1 Pedestrians’ Violation Behaviors at Intersections ......................................................... 16
  2.3.2 Pedestrians’ Crossing Intention Prediction .................................................................. 18
  2.3.3 Summary ...................................................................................................................... 22
2.4 Deep Learning Applications in Transportation Field ......................................................... 22
  2.4.1 Deep Learning Models ................................................................................................. 22
  2.4.2 Applications of Deep Learning Models in Transportation Field .................................. 22
  2.4.3 Human Pose Estimation .............................................................................................. 25
  2.4.4 Limitations of Deep Learning Models .......................................................................... 26
2.5 Summary and Discussion .................................................................................................... 27

CHAPTER 3 PREDICTION OF PEDESTRIANS’ CONFLICTS USING SINGLE SSM INDICATOR .................................................................................................................. 28
## Table of Contents

5.2 SSMs Generation and Threshold Selection .................................................. 66
  5.2.1 Automated Extraction of PET and TTC Values from Videos .................. 66
  5.2.2 Threshold Selection .................................................................................. 69
5.3 Input Variables Overview .............................................................................. 76
5.4 Deep Learning Models .................................................................................. 78
  5.4.1 Support Vector Machine (SVM) ................................................................. 78
  5.4.2 Gated Recurrent Unit (GRU) .................................................................... 78
  5.4.3 Long Short-Term Memory (LSTM) ........................................................... 80
5.5 Experiment Results ....................................................................................... 80
5.6 Summary ........................................................................................................ 81

CHAPTER 6 PREDICTING PEDESTRIANS’ CONFLICTS AT THE SIGNAL CYCLE LEVEL .............................................................................................................................. 82
6.1 Data Collection ............................................................................................... 82
  6.1.1 Variable Extraction .................................................................................... 83
  6.1.2 Data Integration ......................................................................................... 85
6.2 Modeling and Results ..................................................................................... 89
  6.2.1 Re-sampling methods ................................................................................ 89
  6.2.2 Experiments and Results ......................................................................... 90
6.3 Summary ........................................................................................................ 92

CHAPTER 7 CONCLUSION .................................................................................... 93
REFERENCES ....................................................................................................... 95
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The pyramid - interaction between road users as a continuum of events (Hydén, 1987).</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>Applications of deep learning models in transportation (Wang et al., 2019).</td>
<td>23</td>
</tr>
<tr>
<td>3</td>
<td>The studied locations.</td>
<td>29</td>
</tr>
<tr>
<td>4</td>
<td>The definition of conflict area (Lord, 1996).</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>Distribution of small PET values.</td>
<td>31</td>
</tr>
<tr>
<td>6</td>
<td>Object detection and object tracking at the studied area.</td>
<td>32</td>
</tr>
<tr>
<td>7</td>
<td>Trajectories of pedestrians &amp; vehicles at 1st location.</td>
<td>33</td>
</tr>
<tr>
<td>8</td>
<td>Transformation from world coordinates to scaled coordinates.</td>
<td>35</td>
</tr>
<tr>
<td>9</td>
<td>Schematic of LSTM unit (Graves et al., 2013).</td>
<td>37</td>
</tr>
<tr>
<td>10</td>
<td>Model architecture.</td>
<td>38</td>
</tr>
<tr>
<td>11</td>
<td>Pedestrian keypoint detection and transformation.</td>
<td>45</td>
</tr>
<tr>
<td>12</td>
<td>Pedestrian’s red-light crossing intention over time.</td>
<td>49</td>
</tr>
<tr>
<td>13</td>
<td>Procedure of labeling the dependent variable.</td>
<td>50</td>
</tr>
<tr>
<td>14</td>
<td>Shifting the dependent variable (1 sec) ahead.</td>
<td>50</td>
</tr>
<tr>
<td>15</td>
<td>AUC (area under the ROC curve).</td>
<td>55</td>
</tr>
<tr>
<td>16</td>
<td>Variable importance (top 15) from RF model.</td>
<td>58</td>
</tr>
<tr>
<td>17</td>
<td>Automated object detection and tracking models.</td>
<td>64</td>
</tr>
<tr>
<td>18</td>
<td>Trajectories of the road users from videos.</td>
<td>66</td>
</tr>
<tr>
<td>19</td>
<td>Example of generating PET value.</td>
<td>67</td>
</tr>
<tr>
<td>20</td>
<td>TTC values from a pair of pedestrian &amp; vehicle.</td>
<td>68</td>
</tr>
<tr>
<td>21</td>
<td>Validation of automated generated PETs.</td>
<td>69</td>
</tr>
<tr>
<td>22</td>
<td>Parameter stability plots of PET and TTC.</td>
<td>71</td>
</tr>
<tr>
<td>23</td>
<td>Diagnostic plots for GEV and GP models.</td>
<td>76</td>
</tr>
<tr>
<td>24</td>
<td>The architecture of recurrent neural network.</td>
<td>79</td>
</tr>
<tr>
<td>25</td>
<td>Schematic of GRU unit (adapted from Cho et al. (2014)).</td>
<td>80</td>
</tr>
<tr>
<td>26</td>
<td>Example of a crossing pedestrian.</td>
<td>84</td>
</tr>
<tr>
<td>27</td>
<td>An example of ATSPM© record.</td>
<td>84</td>
</tr>
<tr>
<td>28</td>
<td>Modeling workflow.</td>
<td>86</td>
</tr>
<tr>
<td>29</td>
<td>Pearson correlation coefficient.</td>
<td>87</td>
</tr>
<tr>
<td>30</td>
<td>Undersampling and oversampling.</td>
<td>89</td>
</tr>
<tr>
<td>31</td>
<td>Confusion matrix on test data set (XGBT).</td>
<td>92</td>
</tr>
<tr>
<td>32</td>
<td>Variable importance plot (top ten) from (XGBT).</td>
<td>92</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 1 Dissertation structure. .................................................................3
Table 2 Definitions of SSMs indicators (Kathuria and Vedagiri, 2020). ................7
Table 3 Literature on pedestrians’ intention prediction. ..................................21
Table 4 Summary of variable descriptive statistics (data from two intersections). ........36
Table 5 Hyperparameters tuning. ..................................................................39
Table 6 Confusion matrix for binary classification problem. .............................39
Table 7 Internal testing result. ...................................................................40
Table 8 External testing result. ....................................................................42
Table 9 Locations (data collection). ...............................................................45
Table 10 Input variable overview. .................................................................47
Table 11 Classification report for four used models (test data set). .....................56
Table 12 Confusion matrix from RF model (test data set). ...............................57
Table 13 Classification report with different values of prediction horizon. ............59
Table 14 Classification report for general case (test data set) ............................60
Table 15 Details of the processed videos .......................................................63
Table 16 Modeling results for negated PET with different threshold values. ........74
Table 17 Modeling results for negated TTC with different threshold values. ..........75
Table 18 Descriptive statistics of variables. ...................................................77
Table 19 Classification report for three used models. .......................................81
Table 20 Overview of collected videos. .........................................................82
Table 21 Variables from ATSPM© ..............................................................85
Table 22 Input variable overview. .................................................................88
Table 23 Modeling results (twelve models, test data set). .................................91
LIST OF ACRONYMS/ABBREVIATIONS

ADAS: Advanced Driver Assistance System
ATSPM\textsuperscript{©}: Automated Traffic Signal Performance Measures
AUC: area under the ROC curve
CAV: Connected and Automated Vehicle
CCTV: closed-circuit television
CNN: Convolutional Neural Network
CV: Connected Vehicle
DST: Deceleration to Safety Time
FPR: False Positive Rate
GT: Gap time
I2V: Infrastructure-to-Vehicle
LSTM: Long-short term memory
MOT: Multiple Object Tracking
PET: Post Encroachment Time
ROC: Receiver operating characteristics
SMOTE: Synthetic minority over-sampling technique
SSMs: Surrogate Safety Measures
TA: Time to Accident
TCT: Traffic Conflict Techniques
TTC: Time to Collision
UAV: Unmanned Aerial Vehicles
VRU: Vulnerable Road Users
V2I: Vehicle-to-infrastructure
CHAPTER 1 INTRODUCTION

1.1 Introduction

Pedestrians are regarded as Vulnerable Road Users (VRUs). According to the World Health Organization (WHO), 1.35 million fatalities were caused due to road crash annually. Among the total fatalities, 23% were pedestrians’ fatalities (WHO, 2020). Each year, thousands of pedestrians’ deaths are caused by traffic crashes, which constitute 16% of the total road fatalities and injuries in the U.S. (FHWA, 2018). In 2019, 6,590 pedestrian deaths were caused due to traffic crashes, ranked the highest for the last three decades (Governors Highway Safety Association, 2020). Florida has the third highest rate of pedestrian fatality (per 100,000 population) among all the states in the U.S. (GHSA, 2020). Orlando-Kissimmee-Sanford area ranked the fifth among the most dangerous metropolitan areas in the U.S., with 656 pedestrian deaths happened during 2008 and 2017 (DangerousbyDesign, 2021). Researchers have been seeking solutions to better prevent the occurrence of pedestrian-related crashes to reduce fatalities. And Surrogate Safety Measures (SSMs), which are also regarded as traffic conflict techniques (TCTs), can be used to measure crash probability in a proactive way.

Crashes can happen if there is interaction between pedestrian and motorized transportation. Video data have been used in the transportation field for decades. Video data contain rich information from a more microscopic view, which can better help to capture the characteristics of pedestrians and vehicles during their interaction courses. Some deep learning models with
stacked layers, such as LSTM (Long-short term memory) neural network, can be used to better deal with time series data generated from videos.

However, the abrupt movement of pedestrians can leave drivers with no time to take evasive action, thus resulting in potential crashes. It is found that some characteristics of crossing pedestrians, such as red-light violations can lead to more irregularities in pedestrians’ movement, thus causing more dangerous situations. More emphasis should be placed on the prediction of the unexpected behaviors of pedestrians to better improve safety.

Nowadays, the SSMs-based studies show the tendency of combining multiple indicators, which can better capture the dynamic patterns of the pre-crash scenarios. Therefore, with the trajectory data generated from videos, it is better to predict the pedestrians’ safety conditions based on the multiple SSM indicators, since each of them has their features.

The CCTV (closed-circuit television) cameras are widely installed to monitor traffic conditions at intersections, which brings huge potentials for use with relatively low cost. The other data sources from installed infrastructure, such as signal timing data, can also be used to extract variables related to traffic flows. These variables can serve as the input variables to predict the pedestrians’ conflicts at the signal cycle level.

One of the emerging solutions to improve pedestrian safety is the Connected Vehicle (CV) technology. This dissertation proposes models that can be used to warn drivers with I2V (Infrastructure-to-vehicle) communications. The drivers can get more prepared to yield upon receiving warnings.
1.2 Contributions and Objectives

Based on the above discussions, the work in this dissertation aims to make several contributions to improve pedestrian safety in a CV environment. Specifically, this dissertation has three main contributions:

(a): The dissertation advocates models that can predict pedestrian’s safety conditions in time series. The proposed models use trajectory data from videos to predict pedestrians’ unexpected behaviors before crossing (i.e., red-light violations), or potential traffic conflicts during crossing.

(b): The dissertation proposes models to predict pedestrians’ conflicts at the individual vehicle level and the signal cycle level.

(b)-1: Individual vehicle level: the conflicts between each pair of pedestrian and vehicle are predicted.

(b)-2: Signal cycle level: the conflicts between pedestrians and vehicles are aggregated to the signal cycle and predicted.

(c): The dissertation proposes the approach to integrate other data sources, such as ATSPM© (Automated Traffic Signal Performance Measures) with video data.

There are four objectives in this dissertation, corresponding to each chapter:

Objective 1: Predicting pedestrians’ conflicts in time series.

Objective 2: Predicting pedestrians’ crossing intentions during red-light signals in time series.
**Objective 3:** Predicting pedestrians’ conflicts in time series based on two SSMs indicators.

**Objective 4:** Predicting pedestrians’ conflicts at the signal cycle level.

An outline of the related contributions, used video sources, and video processing techniques in each chapter is presented in Table 1.

### Table 1 Dissertation structure.

<table>
<thead>
<tr>
<th>Chapter #</th>
<th>Video sources</th>
<th>Video processing techniques</th>
<th>Related contribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 3</td>
<td>GoPro</td>
<td>YOLO, Deep SORT</td>
<td>(a), (b)-1</td>
</tr>
<tr>
<td>Chapter 4</td>
<td>CCTV</td>
<td>Pose Estimation</td>
<td>(a)</td>
</tr>
<tr>
<td>Chapter 5</td>
<td>CCTV</td>
<td>Mask R-CNN</td>
<td>(a), (b)-1</td>
</tr>
<tr>
<td>Chapter 6</td>
<td>CCTV</td>
<td>YOLO, Deep SORT</td>
<td>(b)-2, (c)</td>
</tr>
</tbody>
</table>

1.3 **Dissertation Structure**

In Chapter 2, a detailed literature review is conducted on previous studies of pedestrian safety based on video data, as well as different video-related applications. However, several research gaps are identified.

Chapter 3 discusses the prediction of pedestrians’ conflicts in time series. An LSTM neural network model is employed to predict pedestrian’s conflicts at the individual vehicle level, with PET indicator.

Chapter 4 discusses the prediction of pedestrians’ crossing intention during red-light signals. Pose estimation techniques are used to generate features on pedestrians’ bodies. Then the models used the generated trajectories to predict the pedestrians’ crossing intentions at intersections’ red-light.

Chapter 5 discusses the prediction of pedestrians’ conflicts based on two SSMs indicators,
PET and TTC. The conflicts at individual-vehicle level from CCTV videos are derived automatically from videos.

Chapter 6 discusses the prediction of pedestrians’ conflicts at the signal cycle level. Geometrical design, ATSPM signal timing, and weather data are used for generating input variables. Pedestrians’ conflicts are aggregated at the signal cycle level.

Chapter 7 concluded this dissertation.
CHAPTER 2 LITERATURE REVIEW

2.1 Pedestrian Safety Analysis Based on Surrogate Safety Measures (SSMs)

2.1.1 SSMs Indicators

Traditional traffic safety studies mainly used crash data. However, crash data were not usually complete or accurate, and sometimes failed to reveal the true contributing factors of crashes (Ismail et al., 2009b). Surrogate Safety Measures (SSMs) were proposed and used to measure crash risks (Khosravi et al., 2018; Tarko et al., 2009; Wu et al., 2019b). The SSMs are also regarded as traffic conflict techniques (TCTs). The definition of a traffic conflict is “an observable situation in which two or more road users approach each other in time and space for such an extent that there is a risk of crash if their movements remain unchanged” (Amundsen and Hyden, 1977).

The interaction between two road users, as a simultaneous arrival in a certain limited area, was defined as an “event” in the literature (Hydén (1987)). Figure 1 shows the severity dimension for various kinds of events describing the relations between the events’ severity and their frequency. The kind of events taking the vertical position in the pyramid are accidents, which have the highest severity and low frequency. Other interactions with different severities, defined as traffic conflicts, could be measured with SSMs indicators, such as Time to Accident (TA, Hydén and Linderholm (1984)), Post Encroachment Time (PET, Cooper (1984)), Time to Collision (TTC, Hayward (1972)), etc. The definitions of various indicators are listed in Table 2. It can be found
that the calculations of most indicators require the trajectories of both road users, and
the TTC, GT and DST calculate the continuous values of time during the whole
interaction courses of the pedestrians and vehicles.

![Pyramid diagram](image)

**Figure 1** The pyramid - interaction between road users as a continuum of events (Hydén, 1987).
Table 2 Definitions of SSMs indicators (Kathuria and Vedagiri, 2020).

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Definition</th>
<th>Type of indicator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to Collision (TTC)</td>
<td>The time required for two vehicles to collide if they continue at their present speeds and on the same path (Hayward, 1972)</td>
<td>A continuous value of time</td>
</tr>
<tr>
<td>Post Encroachment Time (PET)</td>
<td>The time difference between the moment an offending vehicle leaves the area of potential crash and the moment of arrival of the conflicting vehicle (Cooper, 1984)</td>
<td>A discrete value of time</td>
</tr>
<tr>
<td>Gap Time (GT)</td>
<td>The time lapse between the first road user leaves the conflict zone and the arrival of the second road user if they continue with the same velocity and trajectory (Archer, 2004)</td>
<td>A continuous value of time</td>
</tr>
<tr>
<td>Deceleration to Safety Time (DST)</td>
<td>Deceleration required for the second road user to reach the conflict zone no earlier than the first user leaves it (Hupfer, 1997)</td>
<td>A continuous measure of time</td>
</tr>
<tr>
<td>Time to Accident (TA)</td>
<td>The time that remains to an accident from the moment that one of the road users starts an evasive action if they had continued with unchanged speeds and directions (Hydén and Linderholm, 1984)</td>
<td>A discrete value of time</td>
</tr>
</tbody>
</table>

2.1.2 Traffic Conflicts between Pedestrians and Vehicles

Vulnerable road users (VRUs) are regarded as road users that do not have a protective shell around them (Cai et al., 2020; Wegman and Aarts, 2006). Those road users include, among others, pedestrians, cyclists and motorcyclists. This proposal is mainly focused on pedestrian safety specifically. Among the various indicators, PET is
regarded as an appropriate indicator to capture the traffic conflicts between pedestrians and vehicles (Ismail et al., 2009b). Given a predetermined threshold, small PET values denote the proximity of crashes (Cooper, 1984; Ismail et al., 2009b; Mizoguchi et al., 2017).

Previous studies investigated the factors contributing to pedestrian-vehicle conflicts. Environmental factors, such as the signal timing and geometric design of the intersections could influence pedestrian-vehicle conflicts (Gårder, 1989; Salamati et al., 2011). Chen et al. (2017) conducted safety evaluation on an intersection and found more severe conflicts (small PET values) outside the crosswalks. Besides, the drivers’ yielding/braking behaviors, pedestrians’ acceptable gaps, and pedestrians’ yielding behaviors were found to be significant factors for pedestrians’ safety conditions at intersections (Jiang et al., 2020; Li et al., 2021b; Tageldin et al., 2017).

The various indicators, as mentioned above, have different features. Taking into considerations of multiple patterns of pedestrian-vehicle conflicts, different indicators should be used under different conditions (Ni et al., 2016). Some studies investigated the possible aggregating of multiples indicators (Ismail et al., 2011). Some studies made possible changes to the existing measurements. For example, Mohamed and Saunier (2013) proposed a probabilistic version of the PET (pPET), based on the motion prediction method, which could measure the probability that two road users attempting evasive action to avoid a crash. Another indicator called p(UEA) was computed using evasive action sampling and initial locations. The study also compared between the two
proposed indicators with TTC and found that the two indicators could be complementary for TTC.

It should be noted that the evasive actions of pedestrians/vehicles could also denote the presences of traffic conflicts (Tarko et al., 2009). The evasive behaviors were regarded as yes/no variable in traditional works, however, the related studies offered more details by measuring them as the category of soft/hard/no yielding behaviors (Van Haperen et al., 2019). Some studies also utilized the evasive action of pedestrians or vehicles as important factors of traffic conflicts. Ni et al. (2016) used evasive actions as well as SSMs indicators to categorize the severities of traffic events, i.e., safe passage, critical event, and conflict. Kathuria and Vedagiri (2020) divided the conflicts into two patterns, non-evasive action involved and evasive action involved conflicts, according to the evasive actions of both parties of road users. It was found that non-evasive involving behaviors can result in critical interactions. An Import Vector Machine (IVM) model was established to classify the severity levels based on the selected indicators. And 1,486 events, i.e., pedestrian-vehicle interactions, were classified as critical events, mild interactions, and safe passage accordingly.

2.1.3 Summary

With the development of big data and emerging techniques in processing the new data, the recent studies of evaluating pedestrian safety based on SSMs offer a few new insights:
2.1.3.1 Behavioral observation studies for pedestrian safety analysis from a more microscopic view

It is gaining more attention to conduct behavioral observation studies for both pedestrians and vehicle drivers. Behavioral observation studies refer to the studies observing the behaviors of road users, in which the observed road users are not informed beforehand of their participation into the research project (Van Haperen et al., 2019). The interactions exist in the traffic events of two road users, and pedestrians are the most vulnerable road user. However, the studies for vehicle drivers have been taking up most of the related works (Van Haperen et al., 2019), compared with pedestrians and other kinds of VRUs.

The studies based on SSMs belong to the behavioral observation studies, which have been widely used (Ismail et al., 2010; Zaki and Sayed, 2014). The recent related studies tend to conduct the research from a more microscopic view, with the integrations of the behaviors such as yielding behaviors, traffic violations (Alver et al., 2021; Ka et al., 2019; Zaki and Sayed, 2014; Zhang et al., 2020c), crossing decisions (Hashimoto et al., 2016; Ka et al., 2019), uses of mobile phones (Pešić et al., 2016; Schwebel et al., 2012), etc.

2.1.3.2 The analysis of traffic conflicts in time series based on trajectory profiles

Recent studies more tend to analyze the pedestrians’ conflicts from a more dynamic perspective, as most of the indicators are generated from the trajectory profiles of the interactions between pedestrians and vehicles. Kathuria and Vedagiri (2020) defined
the pattern as the profile of a continuous SSM indicator with respect to the time. Ni et al. (2016) used observers’ rating-based approaches for understanding traffic conflicts and behaviors through video observations. The continuous profiles were generated, but the categories of severities of pedestrians’ events were labeled with static threshold values. Laureshyn et al. (2010) proposed a framework to utilize the indicators for a continuous description of an interaction during the process.

### 2.1.3.3 The combination of multiple SSMs indicators

A single indicator cannot accurately reflect the actual situation. Thus, the combination of different indicators to complement each other, or new SSM indicators were proposed to better predict pedestrians’ safety conditions (Kathuria and Vedagiri, 2020; Mohamed and Saunier, 2013; Ni et al., 2016).

In summary, the SSMs can be used to better investigate the pedestrian safety conditions from a more microscopic view. It is also necessary to analyze the traffic conflicts in time series, since most of the indicators are continuous values generated from the trajectory profiles of the interactions. And two or more indicators will help to capture the patterns of pedestrian’s dangerous situation more precisely. However, most of the existing studies failed to do so and used the single indicator instead.

### 2.2 Computer Vision Applications in Transportation Field

Traditional research mainly collected data from sources such as Automatic Vehicle Identification (AVI) system, remote traffic microwave sensor (RTMS), and Bluetooth technique. These techniques fail in capturing road user characteristics. Video data,
however, contains rich information such as pedestrians’ and vehicles’ movements and behaviors. In recent years, computer vision techniques have drawn attention from governments, vehicle manufacturers and many researchers. Video data were used in previous works and provided a broader view for traffic safety analysis (Chen et al., 2017; Formosa et al., 2020; Ismail et al., 2010; Ka et al., 2019; Mahmoud et al., 2021b; Sayed et al., 2013; Zhang et al., 2020a; Zhang et al., 2020b).

Among computer vision techniques, Convolutional Neural Networks (CNNs) have been widely used to process videos automatically. For instance, automated object detection models can be used to identify different kinds of objects from images/videos. The object detection models include Faster RCNN (latest version: Mask RCNN) (Girshick, 2015; Ren et al., 2015), YOLO (Redmon and Farhadi, 2018), SSD (Single Shot Multi-box Detector, Liu et al. (2016)), et.al. Some models, such as YOLO and SSD, apply the neural network on the full images, which are usually regarded as one-stage approaches. Other models, such as R-CNN, first crop the whole image into different crops, then apply the neural networks on these crops. These models are usually regarded as two-stage approaches. The two-stage approach can be more computational cost expensive but bring higher accuracy.

Multiple types of video sources were used in the previous studies, such as roadside videos from surveillance cameras (Chen et al., 2010), ego-view videos from onboard cameras, nighttime videos (Cai et al., 2016; Chen et al., 2010), aerial videos from Unmanned Aerial Vehicles (UAVs) (Tang et al., 2017; Zhong et al., 2017), and fused
visions with both camera and LIDAR (Du et al., 2017), etc. This section conducts a review of these different types of videos accordingly.

2.2.1 Roadside Videos from Surveillance Cameras

Traffic surveillance cameras are usually installed at intersections for law enforcement purposes. The surveillance cameras were widely used to conduct tasks such as before-after studies of traffic treatments, analysis of pedestrians’ and cyclists’ crossing safety as well as crossing behaviors (Hediyeh et al., 2014b; Mahmoud et al., 2021a; Zaki et al., 2013).

With the development of CV technologies, more and more video applications could be used to alert drivers of potentially dangerous events of pedestrians. Heng (2008) investigated a traffic signal device that composed of a transmitter, a receiver, and a storage medium, where the transmitter could broadcast signals to the receiver. When a crash happened, the crash impact data after facilitation were recorded to ease responsibility determination. Wolterman (2008) designed a traffic signal system to mitigate crashes caused by drivers’ traffic violations. Another application “SAFESPOT” was composed of hazard warnings and speed alerts. It was implemented at the black spots of the crash on the road networks (Bonnefoi et al., 2007). Rahman et al. (2019) proposed a system using roadside cameras that outperformed DSRC-enabled devices from the perspective of localization accuracy. It could satisfy the latency requirements with the processing speed of 100ms/frame. The system used a camera to detect the pedestrians’ presences. With TTC as the indicator of safety, the drivers could receive
warnings when there were potentially dangerous situations.

2.2.2 Ego-view Videos from Onboard Cameras

The onboard cameras could help to detect the surrounding road objects, such as lane markings and road signs (Maldonado-Bascon et al., 2007). Some studies used onboard cameras to predict the pedestrians’ positions and moving paths for proactive pedestrian protection systems (Møgelmose et al., 2015; Schmidt and Färber, 2009). Formosa et al. (2020) used an onboard camera and other sensors to predict potential traffic conflicts with the former vehicle in real-time. The features extracted from videos, such as distance (with the former vehicle), speed, braking, acceleration, and deceleration, were fed into the deep learning model.

The onboard cameras could be used together with Advanced Driver Assistance System (ADAS) technologies, such as Forward Crash Warning (FCW). And these implementations were found in the literature to significantly reduce crashes (Yue et al., 2018).

Other types of onboard cameras functioned to capture the drivers’ behaviors such as gazes or faces, to conduct distraction analysis (Vignali et al., 2020), as well as drivers’ maneuvers recognitions (Jain et al., 2015).

2.2.3 Aerial Videos from UAVs

The emerging technologies such as UAVs can offer a wider range of view from road segment or intersection (Wu et al., 2020). Compared with traditional roadside cameras, UAVs can only work within limited time durations. However, with a top-down view,
UAVs can solve the problems of occlusions between road users. Ke et al. (2019) extracted traffic flow parameters such as speed, density, and volume from UAV videos. The processing speed could satisfy the real-time requirements.

2.2.4 Other Video Types

With the rapid development of smartphones, smartphones can function well to act as a microcomputer with embedded cameras. For instance, “WalkSafe” (Wang et al., 2012), an application of smartphone was developed for alarming pedestrians. It could recognize vehicles from front-view and back-view. But as the application used the cellphone’s embedded camera, it could not be used if the smartphone was in the pocket or facing the ground. You et al. (2013) developed a new driver safety application on Android phone. Computer vision and machine learning models were used to detect the fatigues and distractions of drivers using the front-facing cameras, and track road conditions using the rear-facing cameras.

2.2.5 Summary

In summary, though video data suffer from some defects, such as easily being affected by external environmental conditions such as low visibilities, they still play important roles in traffic safety analysis. Future studies of video applications should put more emphasis on the integrations of video sensors with other technologies, such as adding the communication part to send warning to pedestrians/drivers under the Connected Vehicle (CV) environment (Hasibur Rahman and Abdel-Aty, 2021).
2.3 Pedestrians’ Crossing Decisions and Red-Light Violations

The safety analysis can be integrated with other pedestrians’ features, as demographical factors and other crossing features play an important role in pedestrians’ safety conditions (Lee and Abdel-Aty, 2005; Yue et al., 2020a). The pedestrians of different demographical features have different crossing behaviors. Females were found to walk slower than males, and elder people were found to walk slower as well (Montufar et al., 2007). Pedestrians’ characteristics affected their crossing behavior, which could increase the irregularities of their motions. This should be taken into consideration when designing warning systems or safety analysis.

Other pedestrians’ features also played an important role in the pedestrians’ crossing behaviors. Zaki and Sayed (2018) proposed a method to identify pedestrians’ grouping behaviors from video data. Pedestrians walking in groups tended to have coupling in their behaviors, such as adjusting their walking directions and walking velocities accordingly. Other studies also investigated this topic (Mazzon et al., 2013; Zhang et al., 2011). The grouping studies could have applications in counting numbers of pedestrians, as well as studying the probabilities of crash avoidances.

2.3.1 Pedestrians’ Violation Behaviors at Intersections

Among all the crashes between vehicles and pedestrians, pedestrians’ sudden movements behaviors can be one of the causes. It was found that pedestrians’ abrupt movements, such as suddenly walking out of road curvature, can make it hard for drivers to take evasive actions, leading to potential crashes (Yue et al. (2020b)). At the
signalized intersections, normally, pedestrians are sequentially separated from vehicles because of traffic signals. However, pedestrians’ violation behaviors especially red-light violations, will expose them to vehicles and cause potential crashes. Pedestrians’ unexpected change of trajectories on the roads can be one of the causal factors (Yue et al., 2020b).

Pedestrians’ violation behavior is also significant for pedestrian-vehicle conflicts at signalized intersections, as violating pedestrians are exposed to motorized traffic without the protection of traffic signals. For example, the pedestrians’ spatial violations were found to be positively correlated with the number of traffic conflicts (Zaki et al., 2013). In addition, pedestrians’ characteristics affected their crossing behaviors, which could increase the irregularities of their motions. For example, pedestrians who walked in groups had lower walking velocities and higher commonality (Hediyeh et al., 2014a). And females were found to walk slower than males (Montufar et al., 2007). More emphasis should be placed on integrating pedestrians’ characteristics into the pedestrian crossing safety.

Some previous studies investigated the relationships between pedestrians’ characteristics with their red-light violations. Behavior models such as the theory of planned behavior (TPB) model (Ajzen, 1991; Evans and Norman, 1998), and some statistical models such as discrete choice models (Brosseau et al., 2013; Hashimoto et al., 2015) were used. Hamed (2001) investigated the factors influencing pedestrians’ waiting time and crossing attempts. It was found that pedestrians’ characteristics, such
as age, gender, number of people in groups, were significant factors. Pedestrians’ volume, pedestrians’ time of arrival, and safety awareness were also significant factors for pedestrians’ red-light violations (Brosseau et al., 2013; Guo et al., 2011; Hamed, 2001). Tiwari et al. (2007) found that as the waiting time increases, pedestrians were more likely to get impatient and violate the traffic signals.

Based on the above discussion, pedestrians’ features play an important role in pedestrians’ violation behaviors, thus influencing their waiting time and crossing intentions. Most studies used statistical models or behavioral models to investigate the influences of geometric factors and demographical factors.

2.3.2  Pedestrians’ Crossing Intention Prediction

Pedestrians’ crossing intention prediction was typically conducted in the same context with trajectory prediction. Among all the sensors, Wi-Fi and Bluetooth were usually used for indoor localization, while camera and LiDAR were used more in the road environment (Ellis et al., 2009; Keller and Gavrila, 2014). The related studies are summarized in Table 3. It can be found that most studies used cameras to predict pedestrians’ crossing intention or trajectories (Ellis et al., 2009; Ka et al., 2019; Keller and Gavrila, 2014). From the perspective of modeling methods, three types of methods were mainly used in the literature, including parametric models such as Kalman Filter (KF) and Gaussian Process Dynamical Models (GPDMs), machine learning models such as Support Vector Machine (SVM), and deep learning models such as long short-term memory (LSTM) (Goldhammer et al., 2013; Keller and Gavrila, 2014; Rehder and
Kloeden, 2015; Schulz and Stiefelhagen, 2015a). From the perspective of the predicting objectives, the output data were trajectories or crossing/non-crossing intentions (Hariyono and Jo, 2015; Mínguez et al., 2018; Rasouli et al., 2019; Saleh et al., 2018). It was found that pedestrians could change their motions abruptly, or could stop at any time (Yue et al., 2020b). Quintero et al. (2015) used GPDMs and naïve-Bayes classifiers to predict pedestrians’ trajectories and crossing intentions. However, trajectories of more than four seconds were used to predict the next second. The prediction horizon up to 2.5 sec were regarded as short-term prediction. The research gap was to find a robust way with less previous moving profiles as input (Ridel et al., 2018). Besides, as a more critical case, pedestrians’ crossing intention at red-light signals was not emphasized.

It should be noted that it’s usually complicated to define pedestrians’ crossing intention. Most of the traditional studies defined pedestrians’ crossing intention as binary categories, “crossing” or “not crossing”. To better define crossing intentions, some studies classified the pedestrians’ intention into several categories such as walking, standing, starting, stopping, etc. (Hariyono and Jo, 2015; Mínguez et al., 2018; Rasouli et al., 2019; Saleh et al., 2018; Schneemann and Heinemann, 2016). Hariyono and Jo (2015) used observers’ ratings to label the levels of pedestrians’ intention. In most of the cases, the pedestrians were labeled with certain categories such as 0 (“not crossing”) or 1 (“crossing”). Other categories between 0 and 1 were caused by some of the pedestrians’ behaviors, such as turning heads to watch for vehicles. Rasouli et al. (2018)
collected a data set labeling pedestrians’ behaviors across various countries under different lighting conditions. Most of the behavioral patterns found are the sequences of “standing, looking, and crossing”, or “moving, looking, and crossing”. 
<table>
<thead>
<tr>
<th>Title</th>
<th>Sensor</th>
<th>Method</th>
<th>Objective/output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bonnin et al. (2014)</td>
<td>Camera</td>
<td>Context model tree</td>
<td>Crossing intention (crossing/not crossing)</td>
</tr>
<tr>
<td>Kooij et al. (2014)</td>
<td>Camera</td>
<td>Neural network</td>
<td>Trajectory</td>
</tr>
<tr>
<td>Ferguson et al. (2015)</td>
<td>Lidar</td>
<td>Gaussian process mixture model</td>
<td>Crossing intention (crossing/not crossing), trajectory</td>
</tr>
<tr>
<td>Völz et al. (2015)</td>
<td>Lidar</td>
<td>Machine learning</td>
<td>Crossing intention (crossing/not crossing)</td>
</tr>
<tr>
<td>Goldhammer et al. (2015)</td>
<td>Camera</td>
<td>Neural network</td>
<td>Trajectory</td>
</tr>
<tr>
<td>Quintero et al. (2015)</td>
<td>Camera</td>
<td>Gaussian Process Dynamical Models (GPDM)</td>
<td>Crossing intention (crossing/not crossing), trajectory</td>
</tr>
<tr>
<td>Hashimoto et al. (2015)</td>
<td>Camera</td>
<td>Dynamic Bayesian Network (DBN)</td>
<td>Crossing intention (crossing/not crossing)</td>
</tr>
<tr>
<td>Bock et al. (2017)</td>
<td>Camera</td>
<td>Neural network</td>
<td>Trajectory</td>
</tr>
<tr>
<td>Rehder et al. (2018)</td>
<td>Camera</td>
<td>Neural network (LSTM)</td>
<td>Trajectory, goal prediction</td>
</tr>
<tr>
<td>Saleh et al. (2018)</td>
<td>Camera</td>
<td>Neural network (LSTM)</td>
<td>Behavior (bending in, starting, crossing, stopping)</td>
</tr>
<tr>
<td>Mínguez et al. (2019)</td>
<td>Camera</td>
<td>Gaussian process dynamical models (GPDMs)</td>
<td>Behavior (walking, standing, starting, stopping)</td>
</tr>
<tr>
<td>Abughaliuch and Alawneh (2020)</td>
<td>Camera</td>
<td>Neural network</td>
<td>Pedestrians’ moving direction, distance to vehicle</td>
</tr>
<tr>
<td>Goldhammer et al. (2020)</td>
<td>Camera</td>
<td>KF, machine learning</td>
<td>Trajectory, behavior (waiting, starting, moving, stopping)</td>
</tr>
</tbody>
</table>
2.3.3 Summary

In summary, there is few studies using sequential trajectory data as input to predict pedestrians’ red-light violations. Compared with the general case of crossing/not crossing problem, pedestrians’ crossing at red-light can be more critical. It is better to incorporate the pedestrians’ red-light violations with crossing intention prediction.

2.4 Deep Learning Applications in Transportation Field

2.4.1 Deep Learning Models

With the development of deep learning, it could be used to better solve transportation problems. Deep learning is a kind of machine learning technique that uses multiple layers to extract higher level features from the data. The input data can be images, text, or sound. The varieties of deep learning models include deep neural networks (DNNs), recurrent neural networks (RNNs), convolutional neural networks (CNNs), and deep reinforcement learning (DQN). Deep learning models outperform other models from the perspective of accuracy.

2.4.2 Applications of Deep Learning Models in Transportation Field

In recent years, deep learning has shown great advantages with applications involving in four kinds of tasks, including computer vision, time series prediction, classification and optimization (Islam and Abdel-Aty, 2021b; Wang et al., 2019). Figure 2 offers a summary of these typical applications using different kind of deep learning models. It can be found that different models have their own strengths. The DNNs were used in both classification problems as well as time series prediction problems. And the
CNNs had advantage in solving problems involving sequential data, and high dimensional data such as image data (Cai et al., 2019). RNNs also showed advantage in applying on time series data (Al-Hussaini et al., 2017). The DQN could be used to handle optimization problems, such as traffic signal control (Gong et al., 2020).

Figure 2 Applications of deep learning models in transportation (Wang et al., 2019).

The time series tasks included typical predictions of traffic flow parameters, such as speed prediction (Jia et al., 2016), travel time prediction (Gang et al., 2015; Siripanpornchana et al., 2016), traffic volume prediction (Akiyama and Inokuchi, 2014; Xiaojian and Quan, 2009), imbalanced data oversampling (Islam et al., 2021; Islam and Islam, 2017), and driving behavior prediction (Li et al., 2021a). Other applications included abnormal event detection using the estimation of people’s density (Sindagi and Patel, 2018), and using DQN to improve traffic signal control (Genders and Razavi, 2016).

Other implementations are listed as below. Formosa et al. (2020) derived an
indicator Time-To-Crash (TTC) from videos. Fang et al. (2017) used a DNN model to predict transportation modes. The transportation modes included activities such as still, walk, run, bike, motorcycle, car, bus, metro, train, and HSR (high speed rail). The proposed DNN was composed of stacking three layers of shallow structures to better handle the sequential data generated from smartphone sensors. Li et al. (2020a) used a recurrent neural network to predict drivers’ turning movements at intersections.

However, traditional deep learning neural networks were less effective at capturing the relationships in sequential data for future predictions. Thus, recurrent neural network (RNN) was proposed to mitigate this defect by feeding back the output from a time window to the next time window in the same layer (Li, 2020; Li et al., 2020b). But RNNs suffer from the problem of vanishing gradient, i.e., they had difficulty to connect the information when the time span between input and output units was long (Bengio et al., 1994; Pascanu et al., 2013). A particular implementation of the Recurrent Neural Network was LSTM (Long Short-Term Memory) neural network model (Hochreiter and Schmidhuber, 1997), which could capture long-term dependencies of time series data. In transportation fields, LSTM neural networks were used to predict vehicle travel time or traffic speed on highway links as well as urban arterials (Altché and Fortelle, 2017; Ma et al., 2015; Yanjie et al., 2016). They were also used for driving behavior classification and real-time crash risk prediction (Li et al., 2020c; Saleh et al., 2017; Yuan et al., 2019). Through these implementations, LSTM models proved their good performances on sequential traffic data. LSTM neural network brings the possibility to
better predict pedestrians’ movements such as trajectory predictions (Manh and Alaghband, 2018; Xue et al., 2018). Besides, Alahi et al. (2016) used an LSTM neural network to predict pedestrians’ movements based on their interaction between each other in crowded spaces.

2.4.3 Human Pose Estimation

Traditional studies learned pedestrians’ trajectories for predicting future states. However, it was found that merely trajectories of pedestrians and vehicles were not sufficient. Body languages such as leg movements or turning of body were indispensable among all the factors used for predicting pedestrians’ crossing intention. And there were controversial conclusions about whether pedestrians’ gaze or head orientations were important (Fang and López, 2019; Ghori et al., 2018; Schulz and Stiefelhagen, 2015b).

The development of pose estimation (keypoint detection) could better help recognize pedestrians’ states (Fang and López, 2018). The pose estimation technique were used to detect the key points on human body (Pavllo et al., 2019). Pavllo et al. (2019) first applied a convolutional neural network on keypoint data generated from video. Luvizon et al. (2018) used pose estimation to conduct activity recognition. Elings (2018) used videos to recognize drivers that were distracted by phones while driving. Face detection, hand detection, as well as pose estimation of the upper body were used. Moreover, pose estimation offered a robust and effective way to estimate pedestrians’ crossing intention. Ghori et al. (2018) used a long short-term memory (LSTM) model
to predict pedestrians’ and bicyclists’ crossing intention. A Bayesian inference function was used to predict the probabilities of five categories of behaviors (crossing, stopping, starting, etc.). Konrad et al. (2018) used a sequence of poses to extract variables such as lengths, angles, rotation rates, and linear accelerations formed by pedestrians’ joints. The kinematic variables of pedestrians were found to be reliable and accurate enough compared with an inertial measurement unit (IMU).

### 2.4.4 Limitations of Deep Learning Models

The limitations of neural networks are:

1. The deep learning models usually require large computing resources. GPU can provide hundreds of parallel computing units at the same time, which can satisfy the needs for neural networks. But it is much more expensive than CPU;

2. Deep learning models also require the amount of data (Wang et al., 2019);

3. Deep learning models usually lack interpretabilities. The importance of features need to be revealed using extra tools, such as visualization tools of active mechanism (Vaswani et al., 2017; Wang et al., 2019; Zhang and Zhu, 2018).

The special procedures needed for applying neural networks are:

1. Parameter tuning is necessary step as the users need to select the proper hyperparameters to reach the ideal performance. Failing to well tune the model will result in a bad model (Wang et al., 2019);

2. Avoiding overfitting problem. As there are too many weights to be trained, the models are easily to be overfitted. The users need to take special notice of overfitting,
or it will mitigate the performance of the model on the test data set. To resolve this problem, the deep learning model can be trained by using dropout layer, or regularization with L1 or L2 norm (Srivastava et al., 2014).

In summary, the deep learning models outperformed other models from the perspective of accuracy, while showing the lack of ability to interpret the model, such as important or significant features. More studies can be carried out to apply deep learning models in transportation field to solve the problems while maintaining the interpretations of model and input variables.

2.5 Summary and Discussion

Based on the review of the previous studies, it can be found that the SSMs are widely used in pedestrians’ safety-related studies. Since most of the SSMs indicators are continuous values through the pedestrians’ interactions with vehicles, video data will help to analyze the interactions from a microscopic view. However, most of the existing studies used the static values of the indicators, without considering the dynamic patterns lying in the trajectory data.

This research is aimed to better predict pedestrians’ safety conditions, with deep learning models applied to trajectory data. The pedestrians’ red-light violations are also taken into consideration.

Moreover, each of the SSMs indicators has its own feature. The combinations of multiple SSM indicators will complement each other and offer new sights to better predict pedestrians’ safety conditions.
CHAPTER 3 PREDICTION OF PEDESTRIANS’ CONFLICTS USING SINGLE SSM INDICATOR

This chapter proposed a deep learning model to predict pedestrians’ safety condition at intersections using PET as the indicator of safety. Pedestrians’ characteristics affected their crossing behavior, which could increase the irregularities of their motions. Thus, more emphasis should be placed on integrating pedestrians’ characteristics into the pedestrian crossing safety.

With emerging new technologies, road transportation system is changing rapidly. The techniques in computer science field such as the automated computer vision methods and deep learning models, i.e. neural networks, are getting more and more popular. These technologies have the potentials to reduce crashes by processing information of the surrounding traffic conditions. And the idea of Connected Vehicle (CV) has been proposed for decades to better improve the communications between pedestrians and vehicles. CV technologies enable vehicles to receive information from infrastructures though Vehicle-to-infrastructure (V2I) communications. It is possible to integrate the developed models with the infrastructures to improve transportation safety.

3.1 Data Collection

To analyze pedestrians’ crossing behavior at intersections, video data from two intersections were collected using GoPro HERO 7. The first intersection is a three-lag intersection (Figure 3 (a)). It was used to offer original training data set for the proposed
model. Another intersection shown in Figure 3 (b), which had different geometric designs than the first intersection, was used to conduct external test. Video data for both crosswalks (marked in shadow areas) were collected during daytime and nighttime.

![Image](image.png)

**Figure 3** The studied locations.

### 3.1.1 Evaluation of the Crosswalk Safety at the Studied Site

Post Encroachment Time (PET) was used as an indicator of safety (Allen et al., 1978; Tarko et al., 2009). It was defined as the time difference between the moment when the first road user left the potential crash point and the moment when the second user reached it. This indicator was typically used for denoting pedestrian safety in previous studies (Ismail et al., 2009a). As shown in (1), $t_2$ and $t_1$ were the moments when the vehicle/pedestrian left/reached the same area accordingly. And the absolute value of the difference was the PET value. PET threshold was set to be 6s according to the literature (Radwan et al., 2016). If the PET value during a pedestrian-vehicle interaction was smaller than 6 sec, it was a critical event for the pedestrian. The definition of conflict area found in the literature is shown in Figure 4, the conflict area
consisted of the painted lines of the crosswalk and the paths of the vehicles (i.e. the width of the vehicle) (Ismail et al., 2009c; Lord, 1996).

![Figure 4](image)

**Figure 4 The definition of conflict area (Lord, 1996).**

\[ PET = |t_2 - t_1| \]  \hspace{1cm} (1)

Pedestrian-vehicle conflicts were manually collected to assure analysis accuracy. At the first crosswalk, four-hour videos of 334 pedestrians were recorded. There were 69 traffic conflicts. Among them, 40 happened during daytime, and 29 happened during nighttime. At the second crosswalk, two-hour videos of 254 pedestrians were collected. There were 62 traffic conflicts, 48 happened during daytime and 14 happened during nighttime. The details can be found in Figure 5. For both locations, there were more dangerous situations for pedestrians during daytime, with the recording time equally distributed for daytime and nighttime.
3.1.2 Video Processing

To generate the trajectories of pedestrians and vehicles, computer vision techniques including object detection, object tracking, and perspective transformation were used.

3.1.2.1 Object Detection

YOLO (You only look once) is a real-time object detection model first proposed by Redmon et al. (2016). It could apply a single neural network to the full image, dividing different areas (anchor boxes) and classifying the objects in these areas at the same time. This characteristic made it more efficient to use, compared with two-stage models such as R-CNN (Girshick, 2015; Ren et al., 2015). YOLOv3 model (Redmon and Farhadi, 2018) improved the original model by using multi-scale images, data augmentation, and batch normalization during the training procedure. YOLOv3 proved to be effective on the COCO dataset (Lin et al., 2014), a standardized large-scale data set for evaluating...
the performance of object detection algorithms. YOLOv3 has been used to detect road users from traffic video data in previous studies (Jana et al., 2018; Lin and Sun, 2018).

3.1.2.2 Object Tracking

To follow the movements of multiple road users that appeared in the scene, the Deep SORT model (Wojke and Bewley, 2018; Wojke et al., 2017) was used. The model assigned unique tracker IDs to each pedestrian and each vehicle recognized by the detection model and followed their movements. As shown in Figure 6, the blue bounding boxes were generated from the YOLOv3 model, and the white bounding boxes were from the Deep SORT model. The green numbers were the tracker IDs for pedestrians, and the white numbers were the tracker IDs for vehicles.

![Figure 6 Object detection and object tracking at the studied area.](image)

Deep SORT was evaluated on the MOT16 Challenge benchmark (Milan et al., 2016), a standardized benchmark for evaluating the performance of different Multiple Object Tracking algorithms. Deep SORT outperformed previous models from the...
perspectives of MOTA score (Multi-object tracking accuracy), and reducing FN (false negatives), etc. (Wojke et al., 2017). The Deep SORT had a few applications in transportation field (Arvind et al., 2019; Hou et al., 2019).

From this step, location information of pedestrians and vehicles was generated at a frequency of 10Hz (environment: NVIDIA GTX 1080Ti 11G GPU). The trajectories of road users (pedestrians and vehicles) were generated from videos. Figure 7 shows a sample of the trajectories from 1st intersection. Different trackers are marked with different colors.

![Sample trajectories from 1st intersection](image)

**Figure 7 Trajectories of pedestrians & vehicles at 1st location.**

### 3.1.2.3 Perspective Transformation

The purpose of the perspective transformation was to create a mapping from the image plane to the world plane. To calibrate the camera, a set of correspondences between 3D world points and 2D image points needed to be established. Hence, the corners of the studied area in the image were used and related to the actual 3D coordinates in the world coordinate system associated to the selected area.

A homograph matrix $h$ was used to transform the coordinates extracted from videos to world coordinates. To generate matrix $h$, Equation (2) was used, and $h$ was composed
of nine values from $h_1$ to $h_9$. Points from image plane, $(u_i, v_i)$, and world plane $(X_i, Y_i)$, formed matrix $A$. The Singular Value Decomposition (SVD) was used to solve the formula with the constraint of $h_9 = 1$ (Naphade et al., 2019; Španhel et al., 2019; Tang et al., 2019). After generating matrix $h$, the image coordinates could be converted to the world coordinates through the inverse matrix of $h$, thus generating the correct world coordinates of road users.

$$A \cdot h = 0$$  \hspace{1cm} (2)

Another transformation procedure was used to create a generic coordinate system for different intersections. As shown in (3), the matrix $M$ is used to convert world coordinates $(X_i, Y_i)$ to the scaled coordinates $(X_{i, \text{scale}}, Y_{i, \text{scale}})$ based on the length and width of the crosswalk. Matrix $M$ could be calculated using the four-points perspective transformation method offered by OpenCV packages (Szeliski, 2010). And the four points were on the four corners of the boundary at this intersection, as shown in Figure 8. Basically, the camera covers the areas within the four points. Take the top-left point for example, the world coordinate of this point is $(X_1, Y_1) = (-81.1993496, 28.5963094)$, and the scaled coordinate is $(X_{1, \text{scale}}, Y_{1, \text{scale}}) = (0, 58.94)$, as the length of the rectangle area is 58.94 ft. Though this step, the world coordinates were converted to the scaled coordinates, and then normalized to feed into the model. This ensured that
the model can be further implemented to more intersections with different geometric designs.

\[
\begin{pmatrix}
X_{i,\text{scale}} \\
Y_{i,\text{scale}} \\
1
\end{pmatrix} = \begin{pmatrix}
X_i \\
Y_i \\
1
\end{pmatrix} M
\]  

\( (3) \)

Figure 8 Transformation from world coordinates to scaled coordinates\(^*\).

Note: the world coordinates are from Google Maps (2020).

3.1.3 Variables Description

From the above procedures, variables obtained from videos are listed in Table 4. The independent variables were composed of pedestrians’ features, such as gender (male/female), pedestrian coordinates \( (X^{\text{ped}}_i, Y^{\text{ped}}_i) \) walking directions (towards/away from camera), whether the pedestrians crossed during the red light (yes/no), as well as vehicle coordinates \( (X^{\text{veh}}_j, X^{\text{veh}}_j) \). The variables except for pedestrians’ genders and pedestrians crossed during red light were automatically generated from videos. The pedestrians’ and vehicles’ coordinates were preprocessed to feed into the model, as mentioned above. The dependent variables in this study were if the pedestrians had conflicts with vehicles (“yes” or “no”), denoting by the PET values. If the pedestrian had a PET value smaller than the threshold (6s), then the
dependent variable was labeled as “1”.

For predicting purpose, also to implement the system prototype under the Connected Vehicle environment, the dependent variables were shifted ahead by $\theta$ units of time. Considering drivers’ reaction time, $\theta$ was taken as 2 sec (Rahman et al., 2021). The trajectories of pedestrians and vehicles were extracted before reaching the conflict zones 2 sec ahead.

### Table 4 Summary of variable descriptive statistics (data from two intersections).

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Distribution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genders</td>
<td>Male/female</td>
<td>(“male” =389, “female” =199)</td>
</tr>
<tr>
<td>Crossed during red light</td>
<td>Yes/no</td>
<td>(“yes” = 182, “no” = 406)</td>
</tr>
<tr>
<td>Walking directions</td>
<td>Towards/away</td>
<td>(“towards” =305, “away” = 283)</td>
</tr>
<tr>
<td>Pedestrian locations $X_i^{ped}$</td>
<td>Coordinates</td>
<td>(0, 1)</td>
</tr>
<tr>
<td>Pedestrian location $Y_i^{ped}$</td>
<td>Coordinates</td>
<td>(0, 1)</td>
</tr>
<tr>
<td>Vehicle locations $X_j^{veh}$</td>
<td>Coordinates</td>
<td>(0, 1)</td>
</tr>
<tr>
<td>Vehicle locations $Y_j^{veh}$</td>
<td>Coordinates</td>
<td>(0, 1)</td>
</tr>
<tr>
<td>Traffic conflicts*</td>
<td>Yes/no</td>
<td>(“1” =131, “0” = 457)</td>
</tr>
</tbody>
</table>

Note: features of genders and crossed during red light were manually labeled. * marked was the dependent variable.

### 3.2 Methodologies

As trajectory data are time series data, an LSTM neural network (Hochreiter and Schmidhuber, 1997) model can better capture the temporal relationships lying in the data. LSTM neural network is an advanced Recurrent Neural Network (RNN). As Recurrent Neural Networks are less effective to learn long-term dependency from time-series data (Graves et al., 2013), the LSTM neural network is proposed to solve this problem.

Given the time window $t$, a single unit from the LSTM neural network is composed of an input gate $i_t$, a forget gate $f_t$, an output gate $o_t$, as shown in Figure
9. These three gates control information flows in each unit of the neural network. The $i_t$, $f_t$, and $o_t$ are calculated using weight matrices $W$, the input sequence $x_t$, and the last layer output $h_{t-1}$. $c_t$ is the cell activation vector formed by two elementwise products of the vectors. The input sequence $x_t$ is computed from (4) to (8) to generate the hidden layer output $h_t$, which is a vector of probabilities. And the output sequence $y_t$ for the neural network is calculated iteratively from hidden layer output $h_t$, as shown in (9).

![Figure 9 Schematic of LSTM unit (Graves et al., 2013).](image)

\[
i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ic}c_{t-1} + b_i) \quad (4)
\]
\[
f_t = \sigma(W_{fx}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \quad (5)
\]
\[
o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \quad (6)
\]
\[
c_t = f_t \odot c_{t-1} + i_t \odot o(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (7)
\]
\[
h_t = o_t \odot \phi(c_t) \quad (8)
\]
\[
y_t = W_{hy}h_t + b_y \quad (9)
\]

$\sigma$: logistics sigmoid function
$\odot$: elementwise product of the vectors
$\phi$: activation function tanh

The model architecture used in this study is shown in Figure 10. The model contains two stacked LSTM layers and a dense layer. Features from four time slices are
fed into the model as the input of the next layer. The output neuron denotes the classification result. Sigmoid function is the activation function. Adam (Kingma and Ba, 2014) is the optimization function. The model is implemented in Kera’s framework (Chollet, 2015).

![Model Architecture Diagram]

**Figure 10 Model architecture.**

### 3.3 Experiment and Results

The proposed model is first trained and tested using data from one intersection (Figure 3 (a)). There are totally 566,085 records in the data set after slicing and stacking the features from different time slices. The dependent variable is whether a pedestrian has a traffic conflict (PET values are smaller than 6 sec), denoted by “1” and “0”. Eighty percent of the data are used as training data set, and 20% of the data are used as test data set.
The proposed LSTM model is well trained before getting overfitted. As shown in Table 5, the batch size is selected as 1000, the learning rate is 0.005, and the unit number in the LSTM layer is 64. The epoch number for training process is 28.

**Table 5 Hyperparameters tuning.**

<table>
<thead>
<tr>
<th>Hyperparameter</th>
<th>Tuning range</th>
<th>Selected value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Batch size</td>
<td>100, 500, 1000, 5000</td>
<td>1000</td>
</tr>
<tr>
<td>Learning rate</td>
<td>0.001, 0.005, 0.01</td>
<td>0.005</td>
</tr>
<tr>
<td>LSTM unit number</td>
<td>32, 64, 128</td>
<td>64</td>
</tr>
</tbody>
</table>

### 3.3.1 Evaluating Metrics

The diagram for metrics calculation is shown in Table 6. TP (true positive) means the number of actual positive samples that are correctly classified. FP (false positive) means the number of actual negative samples that are wrongly classified. FN (false negative) is the number of actual negative samples that are wrongly classified. TN (true negative) is the number of actual negative samples that are correctly classified.

**Table 6 Confusion matrix for binary classification problem.**

<table>
<thead>
<tr>
<th>Classified value</th>
<th>Actual value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>TP (true positive)</td>
</tr>
<tr>
<td>Negative</td>
<td>FN (false negative)</td>
</tr>
</tbody>
</table>

Using these four values, the equations of calculating the metrics are listed from (10)-(13). Precision, also called positive predictive value (PPV), is the ratio of actual positive samples to the classified positive samples. Recall, also called sensitivity, is the proportion of the actual positive examples that are correctly classified. F1 score is an integrated metric taking into consideration of both precision value and recall value.
Accuracy is defined as the ratio of the correctly classified samples in the whole data set, taking into consideration of both positive samples and negative samples.

\[
Precision = \frac{TP}{TP + FP} \quad (10)
\]

\[
Recall = \frac{TP}{TP + FN} \quad (11)
\]

\[
F1 \text{ score} = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (12)
\]

\[
Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (13)
\]

### 3.3.2 Experiment Results

The experiment results of the proposed LSTM model based on internal testing at the first intersection are listed in Table 7. The model achieves an accuracy of 0.886 on the training data set. On the test data set, the model achieves the precision of 0.864, showing it can identify most of the traffic conflicts successfully. And the recall is 0.880, the F1 score is 0.872. The model achieves an overall accuracy of 0.884 on the test data set at the same intersection.

<table>
<thead>
<tr>
<th>Training set</th>
<th>Test set</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>Precision</td>
</tr>
<tr>
<td>0.886</td>
<td>0.864</td>
</tr>
</tbody>
</table>

### 3.3.3 Experiment Results (External Experiments)

The data set collected at the second intersection (shown in Figure 3(b)) is regarded as external data set, which contains 254 pedestrians and 62 traffic conflicts in total. The objective of the external experiments is to prove that the LSTM model can be implemented at different locations.
The idea is to further train the model by gradually including more external data in the training data set, while keeping the original training data from the 1st intersection. As shown in Table 8, the first column shows the ratios of external data in the training data set, and the second column shows the experiment results on the external test set. When there are no external data in the training data set, the previous model achieves the precision rate of 0.515, the recall rate of 0.537, and the overall accuracy of 0.533. With more external data used in the training process, the model’s performance at the external location gets improved as well. When the external data take up 10% of the training data set, the model achieves the precision of 0.744 and the accuracy of 0.809. When the external data take up 30% of the training data set, the model achieves the accuracy of 0.849 (around 0.85). If the external data continue to increase, as until 50% of the training data set, the model achieves performances that are similar to the original intersection, from the perspective of accuracy. Models are well trained before getting overfitted. And each model achieves the accuracy of around 0.884 on the test data set of the first intersection. In other words, the model’s performance doesn’t get worse at the original intersection with the external data increasing in the training data set.
Table 8 External testing result.

<table>
<thead>
<tr>
<th>Ratio of external data in the training data set</th>
<th>Prediction result on the test set (external data)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
</tr>
<tr>
<td>0%</td>
<td>0.515</td>
</tr>
<tr>
<td>10%</td>
<td>0.744</td>
</tr>
<tr>
<td>20%</td>
<td>0.774</td>
</tr>
<tr>
<td>30%</td>
<td>0.842</td>
</tr>
<tr>
<td>40%</td>
<td>0.851</td>
</tr>
<tr>
<td>50%</td>
<td>0.856</td>
</tr>
</tbody>
</table>

The experiment results indicate that when there are two intersections, the ratio of 30% of external data will improve the model’s accuracy to around 0.85 on the test data set of the new location. This indicates that the model can be further trained and implemented with more external data, to be implemented at different intersections.

3.4 Summary

In this chapter, an LSTM neural network model is employed to predict whether the pedestrians would face dangerous situations, denoting by small PET values. Based on detection and tracking techniques in computer vision, the characteristics of pedestrians and vehicles are fed into the model. The proposed model achieves the accuracy of 0.884 at one signalized intersection. The external test indicates that including 30% new data significantly improves the model performance at a different location to an ideal accuracy (around 0.85). The results imply that the characteristics during pedestrian-vehicle interaction processes will reflect the potentially dangerous situations of pedestrians. Moreover, the model can be further trained and implemented at different locations with smaller size of new data set required.
This chapter predicts the pedestrians’ conflicts in time series before the pedestrians and vehicles reach the conflict zones. And different geometric designs of intersections are taken into considerations by transforming the location coordinates. More research can be further conducted to implement the model in the field experiments with Connected Vehicles’ (CV) technologies, to better warn drivers.
CHAPTER 4 PREDICTING PEDESTRIANS’ RED-LIGHT CROSSING

INTENTIONS USING POSE ESTIMATION

In this chapter, four machine learning models are used to predict pedestrians’ crossing intention at intersections’ red-light. With pose estimation (keypoint detection) technique, the CCTV videos collected from three signalized intersections are used to extract pedestrians’ variables, such as the angles between ankle and knee, and elbow and shoulder, etc. Different prediction horizons are also taken into consideration.

4.1 Data Collection

The videos used in this study are from three signalized intersections located in Seminole County, Florida. All the videos are collected using CCTV (closed-circuit television) cameras during 8:00-19:00 on sunny workdays in October and November 2019. All the intersections are four-lane by two-lane intersections to ensure the performance of the pose estimation model. The detailed information is listed in Table 9. A total of 150-hours of videos are processed with 182 pedestrians collected as valid samples. The pedestrians’ trajectories before crossing (in waiting zone) are extracted. Another data source is ATSPM© (Florida Department of Transportation, 2020) (Automated Traffic Signal Performance Measures) signal timing data to label pedestrians who cross at the red-light.
Table 9 Locations (data collection).

<table>
<thead>
<tr>
<th>Intersection</th>
<th>Road width (major/minor)</th>
<th>Vehicle volume (daily, major/minor road)</th>
<th>Vehicle approach speed (major road)</th>
</tr>
</thead>
<tbody>
<tr>
<td>US 17-92@3rd St</td>
<td>61 ft/20 ft</td>
<td>24023/9519</td>
<td>28 mph</td>
</tr>
<tr>
<td>US 17-92@13th St</td>
<td>62 ft/36 ft</td>
<td>24251/4769</td>
<td>27 mph</td>
</tr>
<tr>
<td>SR 46@Park Dr</td>
<td>63 ft/39 ft</td>
<td>9959/5864</td>
<td>32 mph</td>
</tr>
</tbody>
</table>

(a) Keypoint detection (CMU-Perceptual-Computing-Lab)  
(b) Egocentric coordinates

Figure 11 Pedestrian keypoint detection and transformation.

4.1.1 Video Processing

The main objective of pose estimation is to derive a representation of the pedestrian’s skeleton from each frame of video. Eighteen key points on the human body are generated, as shown in Figure 11 (a), which mainly include nose, eyes, shoulders, elbows, wrists, hips, knees, ankles, etc. (CMU-Perceptual-Computing-Lab; Ou Zheng, 2019). The object tracking model (Zhou et al., 2020), is used to follow the movement of each road user, and generate the trajectory of the pedestrian. The blue number on the top left corner of the bounding box is the pedestrian’s tracking ID number. The number on the bottom right corner is the confidence level of pose estimation model.

As shown in Figure 11 (b), the coordinates of keypoint are first normalized and converted to the egocentric coordinates on the pedestrian’s body. The origin of this
egocentric coordinate system is located at the middle point between left hip and right hip, with three orthogonal axes. Previous studies have found that variables from human bodies such as angles formed by joints were related to their acceleration (Konrad et al., 2018). Thus, some of the angles generated between joints (such as angle between left wrist and left elbow $\alpha_{67}$, angle $\alpha_{90}$ between right ankle and right knee, etc) are extracted as input variables for further modeling. Facial variables such as angle between nose and eye are also extracted as they can potentially reflect head orientation.

The other step is perspective transformation, to convert pedestrians’ coordinates to the world coordinates. The procedures are illustrated in chapter 3.1.2. The walking speed of the pedestrian is calculated using (14).

$$\text{Walking speed} = \frac{\text{Haversine}((X_{t_1}, Y_{t_1}), (X_{t_2}, Y_{t_2}))}{(t_2 - t_1)} \quad (14)$$

4.1.2 Input Variables Overview

Using pose estimation, the angles between some of the key joints are generated. Besides, pedestrians’ walking directions, waiting time (time that the pedestrian has reached the waiting zone), walking speed, and whether pedestrian has pushed the pushbutton (to activate pedestrian signal phase), are also used as input variables. Some external variables are also included. The hourly temperature data are from National Oceanic Atmospheric Administration (NOAA). Total vehicle volume and right-turn vehicle volume at the current signal cycle, and green time of the vehicle signal phase on pedestrian’s conflicting direction are extracted from ATSPM©. An overview of all input variables is listed in Table 10.
Table 10 Input variable overview.

<table>
<thead>
<tr>
<th>Description</th>
<th>Mean</th>
<th>Standard deviation</th>
<th>Minimum</th>
<th>Maximum</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Walking direction</td>
<td>3.154</td>
<td>1.659</td>
<td>0.079</td>
<td>6.259</td>
<td>Rad</td>
</tr>
<tr>
<td>Walking speed</td>
<td>1.418</td>
<td>1.042</td>
<td>0.026</td>
<td>3.091</td>
<td>Ft/s</td>
</tr>
<tr>
<td>Pushing button</td>
<td>0.725</td>
<td>0.446</td>
<td>0</td>
<td>1.000</td>
<td>-</td>
</tr>
<tr>
<td>Waiting time</td>
<td>10.485</td>
<td>10.001</td>
<td>0.5</td>
<td>55.667</td>
<td>Sec</td>
</tr>
<tr>
<td>Angle $\alpha$ (ear &amp; eye, left)</td>
<td>1.362</td>
<td>0.792</td>
<td>7.5e-04</td>
<td>3.135</td>
<td>Rad</td>
</tr>
<tr>
<td>Angle $\alpha$ (ear &amp; eye, right)</td>
<td>0.521</td>
<td>0.270</td>
<td>2e-04</td>
<td>3.141</td>
<td>Rad</td>
</tr>
<tr>
<td>Angle $\alpha$ (nose &amp; eye, left)</td>
<td>1.374</td>
<td>0.870</td>
<td>3e-04</td>
<td>3.134</td>
<td>Rad</td>
</tr>
<tr>
<td>Angle $\alpha$ (nose &amp; eye, right)</td>
<td>0.397</td>
<td>0.214</td>
<td>9.70e-05</td>
<td>3.091</td>
<td>Rad</td>
</tr>
<tr>
<td>Angle $\alpha$ (elbow &amp; shoulder, left)</td>
<td>0.583</td>
<td>0.340</td>
<td>1e-03</td>
<td>3.116</td>
<td>Rad</td>
</tr>
<tr>
<td>Angle $\alpha$ (elbow &amp; shoulder, right)</td>
<td>0.692</td>
<td>0.458</td>
<td>1e-03</td>
<td>3.140</td>
<td>Rad</td>
</tr>
<tr>
<td>Angle $\alpha$ (wrist &amp; elbow, left)</td>
<td>0.649</td>
<td>0.406</td>
<td>4e-04</td>
<td>3.139</td>
<td>Rad</td>
</tr>
<tr>
<td>Angle $\alpha$ (wrist &amp; elbow, right)</td>
<td>0.629</td>
<td>0.487</td>
<td>1e-04</td>
<td>3.129</td>
<td>Rad</td>
</tr>
<tr>
<td>Angle $\alpha$ (ankle &amp; knee, left)</td>
<td>0.801</td>
<td>0.592</td>
<td>8e-04</td>
<td>3.132</td>
<td>Rad</td>
</tr>
<tr>
<td>Angle $\alpha$ (ankle &amp; knee, right)</td>
<td>0.756</td>
<td>0.576</td>
<td>7.49e-05</td>
<td>3.122</td>
<td>Rad</td>
</tr>
<tr>
<td>Vehicle volume (current cycle)</td>
<td>72.000</td>
<td>16.102</td>
<td>20.000</td>
<td>120.000</td>
<td>-</td>
</tr>
<tr>
<td>Vehicle green time (current cycle)</td>
<td>45.208</td>
<td>21.575</td>
<td>0.007</td>
<td>84.256</td>
<td>Sec</td>
</tr>
<tr>
<td>Vehicle counts (right-turn, current cycle)</td>
<td>5.000</td>
<td>3.060</td>
<td>0</td>
<td>11.000</td>
<td>-</td>
</tr>
<tr>
<td>Temperature</td>
<td>83.847</td>
<td>3.337</td>
<td>68.000</td>
<td>89.000</td>
<td>Fahrenheit</td>
</tr>
</tbody>
</table>

4.1.3 Pedestrians’ Crossing Intention Labeling

Previous studies found that pedestrians’ red-light intention increased when waiting time increased (Guo et al., 2011; Keegan and O’Mahony, 2003). Thus, the last
few moments are an important research target when a pedestrian approaches the road, stops at the curb (waiting zone), and finally starts crossing at red-light signal phase. Figure 12 (a) shows a sequence of video frames. The time-to-cross has been previously used in the related work (Ghori et al., 2018; Schneider and Gavrila, 2013) as the time difference between each frame and the frame when the pedestrian starts crossing. Time-to-cross equals zero means that the pedestrian starts to cross. As the time-to-cross gets closer to zero (shown in Figure 12 (b)), the pedestrian behaves more and more impatiently while looking around and watching for approaching traffic. Meanwhile, his crossing intention becomes clearer over time.

On average, the time intervals pedestrians spent on observing the surrounding environment are between 1 sec and 2 sec, which are around 1.32 sec for adults and 1.45 sec for the elderly and children (Rasouli et al., 2018). This time interval is important for the decision-making of the crossing/not-crossing behavior. So, the last 1 sec to 2 sec before crossing can be important for the prediction of pedestrians’ red-light crossing behavior.
The dependent variable is pedestrians’ crossing intention in this study. The labeling procedure is shown in Figure 13. The CCTV videos are first processed using pose estimation and object tracking techniques. The frame rate of CCTV videos is 30 frames per second (fps). The samples in every 0.5 s are then smoothed and aggregated into one sample to remove noise. The samples in the waiting zones are labeled with three classes, standing, walking normally (for pedestrians who cross during pedestrian signals), and walking at red-light (for those who cross at red-light). Basically, the first class is from the video frames when the pedestrians stand still, and the other two classes are from the video frames when the pedestrians start to cross (last 1 sec 2 sec before time-to-cross=0). The labels are validated through manual checks to ensure accuracy.
For prediction purpose, we suppose the driver will yield to pedestrians after capturing the pedestrians’ crossing intentions after the reaction time 1 sec (Obeid et al., 2017; Wilson et al., 1997). In this case, vehicles travel at 20 mph will have a stopping distance of 40 ft. The dependent variable is shifted 1 sec ahead of time (Figure 14). This is regarded as the prediction horizon. The generated data set is later split into training and test data set for further modeling.

4.2 Machine Learning Models

4.2.1 Support Vector Machine (SVM)

As this is a classification problem, the classifier from Support Vector Machine
(SVM) is used (Boser et al., 1996). SVM is a supervised learning algorithm which is widely used. Given a data set $D$ in the form of $\{x_i, y_i\}_{i=1}^{N}$ where $x_i \in R_d$ are the samples, and the $y_i$ is the class label, SVM maps the feature vector $x_i$ to a $N$-dimensional space, with $N$ as the number of features of the samples. For a two-class classification problem, SVM finds the hyperplane (decision boundary) to distinctly separate the two classes of samples. And the distance between the two classes, is regarded as margin distance. SVM uses a loss function to maximize the margin distance, which is to solve:

$$ J = \min \frac{1}{2} w^T w + C \sum_{i=1}^{N} \varepsilon_i, $$

s.t. $y_i(w^T K_i(x_i) + b) \geq 1 - \varepsilon_i$ \hspace{1cm} (15)

With $w$ is weight vector, $C$ is cost coefficient, and $\varepsilon_i$ is a slack variable for the non-separable data. And $K_i$ is the kernel function to transform data to the feature space. Kernel functions can be linear function, or radial basis function (rbf), etc.

The hyperparameters used in the SVM mainly include:

(1) C: a regularization parameter, a small C value denote a large margin.

(2) kernel function: the kernel function $K_i$ used in the loss function.

(3) gamma: kernel coefficient.

### 4.2.2 Random Forest (RF)

A Random Forest model (RF) usually consists of a few decision tree models (Breiman et al., 1984). A decision tree model starts at the root node and split the data on the features that result in the largest information gain (IG). This partition process is
repeated iteratively until the child node has values all belong to the same class (Breiman et al., 1984). A tree model is built on all the samples and all the features. Thus, the decision tree can get overfitted easily, and not robust enough.

Nevertheless, RF constructs many decision trees (weak learners) and uses a subset of features as well as a subset of samples to build trees (Breiman, 2001). Each tree is independently constructed using a subset of the original data set and gets trained. This is regarded as a bootstrap aggregating way by resampling training data with replacement. The advantage of the random forest is that it can partially eliminate the correlations between different trees if there are some influential variables in the data set. After each tree generates the classification result, RF uses majority voting to decide the final output.

The hyperparameters of RF mainly include:

(1) the number of trees: the number of decision trees the algorithm builds before taking the maximum voting.

(2) the minimum number of samples: the minimum number of samples required to split an internal node.

(3) the minimum number of leaves: the minimum number of samples at each leaf node.

(4) the maximum depth of trees: the maximum number of levels in trees.

4.2.3 Gradient Boosting (GBM)

Gradient Boosting model (GBM) is a machine learning method that also utilizes
many decision trees (weak learners) to generate the results. For GBM, at each iteration, a new tree is added. The subsequent trees will give extra weights to the samples that are incorrectly classified by the prior tree, and weighted voting is used to generate the final classification result based on the all the trees (Friedman, 2001).

The hyperparameters of GBM are similar to RF, which include:

1. learning rate: the weight of every new tree adds to the model.
2. the number of trees: the number of trees the algorithm builds.
3. the minimum number of samples: the minimum number of samples required to split an internal node of the tree.
4. the minimum number of leaves: the minimum number of samples at each leaf node of the tree.
5. subsample rate: subsample rate of the training instances at each iteration.
6. maximum depth of trees: the maximum number of levels in the trees. If there is no maximum depth defined, the nodes of the trees will be expanded until all leaves are pure (the node only contains samples from one class).

4.2.4 Extreme Gradient Boosting (XGBoost)

Under the Gradient Boosting framework, Extreme Gradient Boosting (XGBoost) is a scalable tree boosting model that is efficient and effective (Chen and Guestrin, 2016). When the number of features is very large, there can be numerous new trees to build with numerous selections of feature subset. Instead of building all the trees, XGBoost estimates the distributions of features across all data points in a leaf to reduce
the search space for building new trees. Thus, XGBoost is used as an efficient model without weakening the performance of GBM.

The hyperparameters of XGBoost model mainly include:

1. **learning rate**: the weight of every new tree adds to the model.
2. **the number of trees**: the number of trees the algorithm builds before taking the maximum voting.
3. **maximum depth of trees**: the maximum number of levels in trees.
4. **the minimum number of samples**: the minimum number of samples required to split an internal node of the tree.
5. **the minimum number of leaves**: the minimum number of samples at each leaf node of the tree.
6. **gamma**: minimum loss reduction required to make a further partition on a leaf node of the tree.
7. **subsample rate**: subsample rate of the training instances at each iteration.
8. **lambda**: regularization term, a larger lambda value will make model more conservative.

### 4.3 Experiment and Results

Four machine learning models, Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting (GBM), and Extreme Gradient Boosting (XGBoost), are established to predict pedestrians’ red-light crossing intentions. The models’ hyperparameters are tuned to reach the best performance.
The evaluating metrics, such as precision, recall, F1 score, and accuracy were illustrated in the Chapter 3.3. The AUC (Area under the ROC curve) value is a comprehensive metric to evaluate the performance of the imbalanced data set (Zweig and Campbell, 1993). AUC measures the two-dimensional area underneath the entire ROC curve from (0,0) to (1,1), as shown in Figure 15. The curve (dotted line) is composed by different pairs of TP Rate (true positive rate, or recall) and FP Rate (false positive rate, 1 minus specificity), under different threshold values (decision boundaries) of a classification problem. The solid line shows the null model (by random selection), with AUC value as 0.5.

![ROC Curve](image)

**Figure 15 AUC (area under the ROC curve).**

### 4.3.1 Experiment Results

Among all the 182 pedestrians collected from CCTV data, 61 pedestrians started to cross the road during the red-light signals. With the sampling time window at 0.5 sec, there are 2,375 data samples collected, with the number of samples between the three classes is 1,725: 407: 243. Eighty percent of the samples are used as the training data
set, and twenty percent of the samples are used as the test data set. Synthetic Minority Over-Sampling Technique (SMOTE) is used to balance the numbers of samples in three classes in the training data set, to make all three categories balanced (Chawla et al., 2002; Islam and Abdel-Aty, 2021a). SMOTE is a popular over-sampling method, which can generate new minority class records by interpolating between several minority class examples that lie together. It should be noted that SMOTE is only applied to the training data set, while the test data set still uses the original records.

In this study, the dependent variable is divided into three classes, standing, walking (normal), and walking (red-light). The last class is the most critical class. So, the model’s performance of this class should be put more emphasis on. Meanwhile, the average value of metrics over three classes, which is usually called macro average value, is also calculated. The modeling results of the four models are listed in Table 11.

**Table 11 Classification report for four used models (test data set).**

<table>
<thead>
<tr>
<th>Model</th>
<th>Class</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 score</th>
<th>Accuracy</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>Walking (red-light)</td>
<td>0.714</td>
<td>0.400</td>
<td>0.513</td>
<td>0.838</td>
<td>0.668</td>
</tr>
<tr>
<td></td>
<td>Macro average</td>
<td>0.797</td>
<td>0.562</td>
<td>0.659</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RF</td>
<td>Walking (red-light)</td>
<td>0.800</td>
<td>0.757</td>
<td>0.778</td>
<td>0.920</td>
<td>0.849</td>
</tr>
<tr>
<td></td>
<td>Macro average</td>
<td>0.859</td>
<td>0.818</td>
<td>0.837</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GBM</td>
<td>Walking (red-light)</td>
<td>0.800</td>
<td>0.673</td>
<td>0.731</td>
<td>0.903</td>
<td>0.818</td>
</tr>
<tr>
<td></td>
<td>Macro average</td>
<td>0.846</td>
<td>0.751</td>
<td>0.796</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XGBT</td>
<td>Walking (red-light)</td>
<td>0.818</td>
<td>0.667</td>
<td>0.735</td>
<td>0.912</td>
<td>0.836</td>
</tr>
<tr>
<td></td>
<td>Macro average</td>
<td>0.862</td>
<td>0.777</td>
<td>0.817</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: the best model is marked in **bold**, macro average is average value of the metrics
over three classes.

With the prediction horizon as 1 sec, the best model is determined to be RF (Random Forest). The recall value for “walking (red-light)” class is 0.757, which means the model can recognize 75.7% of the samples (video frames) in which the pedestrians start walking at red-light. Meanwhile, the precision rate is 0.8. It also achieves the best performance over three classes compared with the other models. Overall, RF achieved an accuracy of 0.920 and an AUC value of 0.849 over the test data set.

Confusion matrix is usually used to check the overall performance of the model, and identify the specific errors affecting each class. The confusion matrix of the RF model on the test data set is shown in Table 12. Most of the samples in each class are classified correctly, denoting the model has a good performance.

Table 12 Confusion matrix from RF model (test data set).

<table>
<thead>
<tr>
<th>Actual class</th>
<th>Predicted class</th>
<th>Standing (normal)</th>
<th>Walking (normal)</th>
<th>Walking (red-light)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standing</td>
<td>362</td>
<td>7</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Walking (normal)</td>
<td>15</td>
<td>46</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Walking (red-light)</td>
<td>7</td>
<td>2</td>
<td>28</td>
<td></td>
</tr>
</tbody>
</table>

Note: numbers of correctly classified samples are marked in **bold**.

The variable importance plot with the top fifteen important variables is shown in Figure 16. It can be found that walking speed, waiting time, green time (vehicle signal phase), pushing button behavior play important roles for predicting pedestrians’ red-light crossing intention. Besides, the angles between knee and ankle (on the left side) also play an important role. Facial variables are also found to be important, such
as the angle between left ear and left eye. This may be related to head orientation. As the data are collected in Florida, where extremely hot weather is usually present at noon, the temperature is also an influencing factor.

Given higher speed limits, there is a need to use longer prediction horizon to build the model (Wu et al., 2019a). Thus, the other values, 2 sec, 3 sec, and 4 sec, are also taken into consideration. The experiment results are shown in Table 13. When the prediction horizon is 2 sec, the model still maintains an AUC value of 0.841. With the prediction horizon increases to up to 4 sec, the sample size keeps shrinking. So, the model’s performance on “walking (red-light)” class (the most minority class) gets worse, resulting in low values of the evaluating metrics. Besides, the macro average values of evaluating metrics show an overall tendency of decreasing. For an imbalanced data set, the AUC can better reflect model performance than accuracy. The AUC shows a steady decreasing tendency with the prediction horizon increases.

![Figure 16 Variable importance (top 15) from RF model.](image)
### Table 13 Classification report with different values of prediction horizon.

<table>
<thead>
<tr>
<th>Prediction horizon</th>
<th>Class</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 score</th>
<th>Accuracy</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 sec</td>
<td>Walking (red-light)</td>
<td>0.800</td>
<td>0.757</td>
<td>0.778</td>
<td>0.920</td>
<td>0.849</td>
</tr>
<tr>
<td></td>
<td>Macro average</td>
<td>0.859</td>
<td>0.818</td>
<td>0.837</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 sec</td>
<td>Walking (red-light)</td>
<td>0.838</td>
<td>0.623</td>
<td>0.715</td>
<td>0.920</td>
<td>0.841</td>
</tr>
<tr>
<td></td>
<td>Macro average</td>
<td>0.813</td>
<td>0.797</td>
<td>0.797</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 sec</td>
<td>Walking (red-light)</td>
<td>0.750</td>
<td>0.316</td>
<td>0.444</td>
<td>0.889</td>
<td>0.719</td>
</tr>
<tr>
<td></td>
<td>Macro average</td>
<td>0.795</td>
<td>0.619</td>
<td>0.674</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4 sec</td>
<td>Walking (red-light)</td>
<td>0.625</td>
<td>0.417</td>
<td>0.500</td>
<td>0.925</td>
<td>0.688</td>
</tr>
<tr>
<td></td>
<td>Macro average</td>
<td>0.769</td>
<td>0.715</td>
<td>0.735</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: **macro average** is the average value of the metrics calculated over three classes.

#### 4.3.2 Experiment Results (General Case)

For comparison, a more general case is also established. Without including the information of the red-light signals, pedestrians’ crossing intentions are divided into “standing” and “walking”. As shown in Table 14, the experiment results on the test data set is shown as below. It can be found that the best model achieves the recall value of 0.817 on the “walking” class, and an AUC value of 0.889.
Table 14 Classification report for general case (test data set)

<table>
<thead>
<tr>
<th>Model</th>
<th>Class</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 score</th>
<th>Accuracy</th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>Walking</td>
<td>0.717</td>
<td>0.589</td>
<td>0.647</td>
<td>0.862</td>
<td>0.763</td>
</tr>
<tr>
<td></td>
<td>Macro average</td>
<td>0.805</td>
<td>0.763</td>
<td>0.781</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RF</td>
<td>Walking</td>
<td>0.883</td>
<td>0.817</td>
<td>0.849</td>
<td>0.924</td>
<td>0.889</td>
</tr>
<tr>
<td></td>
<td>Macro average</td>
<td>0.909</td>
<td>0.855</td>
<td>0.878</td>
<td></td>
<td></td>
</tr>
<tr>
<td>GBM</td>
<td>Walking</td>
<td>0.869</td>
<td>0.736</td>
<td>0.797</td>
<td>0.921</td>
<td>0.886</td>
</tr>
<tr>
<td></td>
<td>Macro average</td>
<td>0.900</td>
<td>0.853</td>
<td>0.874</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XGBT</td>
<td>Walking</td>
<td>0.797</td>
<td>0.708</td>
<td>0.750</td>
<td>0.900</td>
<td>0.830</td>
</tr>
<tr>
<td></td>
<td>Macro average</td>
<td>0.861</td>
<td>0.830</td>
<td>0.844</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: bold marked is the best model; macro average: average value of the metrics calculated over three classes.

4.4 Summary

This chapter uses video data to predict pedestrians’ red-light crossing intentions at the signalized intersections with pose estimation and various machine learning models. The highlights of the study mainly include:

(1) The pose estimation technique is used to capture the variables of the pedestrians’ bodies, such as angles formed by some of the key joints (wrist, elbows, etc.) and facial landmarks (nose, eyes, and ears) over time.

(2) Upon labeling the dependent variable, pedestrians’ red-light crossing intention, the study takes into consideration both mobility (standing or walking) and pedestrians’ red-light crossings.

(3) Four machine learning models are used to predict the pedestrians’ red-light crossing intentions with multiple prediction horizons. The best model achieves an AUC
value of 0.849.

Through the established models, there are a few points to be marked on pedestrians’ crossing intention prediction. The walking speed is the top important variable to reflect pedestrians’ crossing intentions. The other variables such as button pushing and waiting time may be related to red-light violations. The leg movement denoting by the angle between knee and ankle is an important variable. Compared with the body part, the facial landmarks also reveal early signs of starting walking, which can be related to head orientation.

With respect to different prediction horizons, though the evaluating metrics on walking (red-light) class fluctuate, the model still shows a fairly good performance over all target classes. Overall, the AUC value decreases as prediction horizon increases. When the prediction horizon is 2 sec the model’s performance is still good, with the recall value as 0.623 (on “walking (red-light)” class and AUC value as 0.841.

A more generic model with the dependent variable labeled as “standing”/“walking” is also established for comparison. It can be found that the model’s performance gets improved, with the AUC value as 0.889. If the signal timing data is not available, then this model can be used instead for warning approaching vehicles, especially the right-turn vehicles.

The limitation of this study is that data from merely three intersections are used with similar geometric design (four-lane by two-lane intersections). But as the CCTV cameras are installed at different locations with different angles, the model successfully
deals with the heterogeneity of the generated data set. The study sheds light on the application of pose estimation for studying pedestrian safety. The variables automatically generated from pose estimation can better reflect pedestrians’ red-light crossing intention than dynamic variables such as position and speed (Zhang, 2020; Zhang et al., 2020c). Future work can be conducted to implement the proposed model in field test.
CHAPTER 5 PREDICTING PEDESTRIANS’ CONFLICTS USING MULTIPLE SSMS INDICATORS

In this chapter, the pedestrians’ conflicts are predicted based on the variables extracted from the CCTV video data. Two SSMS indicators, PET and TTC, are derived from videos. Pedestrians’ conflicts are further labeled and modeled at the individual-vehicle level (each pair of pedestrian and vehicle).

5.1 Variable Extraction

The details of the processed videos are shown in Table 15. For these two intersections, videos were collected during October 2019 through November 2019, from 8 am to 6 pm on five weekdays at each intersection.

<table>
<thead>
<tr>
<th>Studied location</th>
<th>Cross section</th>
<th>Video length</th>
<th>Daily pedestrian volume</th>
</tr>
</thead>
<tbody>
<tr>
<td>US17-92@13th St</td>
<td>Four-lane by two-lane</td>
<td>50-hours</td>
<td>130</td>
</tr>
<tr>
<td>US17-92@3rd St</td>
<td>Four-lane by two-lane</td>
<td>50 hours</td>
<td>159</td>
</tr>
</tbody>
</table>

The procedures taken to extract pedestrians’ variables from video data include object detection, object tracking, and perspective transformation.

5.1.1 Object Detection and Tracking Models

The object detection model is used to classify different kinds of road users, i.e., human beings and vehicles. The detection model used in this study is Mask R-CNN (Region-based Convolutional Neural Network), the state-of-the-art automated object
detection model (He et al., 2017). The Mask R-CNN model is adapted from Faster-RCNN (Ren et al., 2015). The detection model can classify different kinds of objects in a frame and generate a segmentation mask. It first scans the whole image and estimates the areas that are likely to contain an object, then classifies the objects in each crop of these areas. This mechanism ensures the good performance of the detection model, especially on small objects that are usually hard to detect.

The object tracking model is used to take the initial sets of the detection model and track the movements of each road user. The tracking method used is CSRT (DCF-CSR, Discriminative Correlation Filter with Channel and Spatial Reliability) tracker from OpenCV (Open Sources Computer Vision) library (Bradski, 2019).

Figure 17 shows a snapshot of the output videos. The bounding box is generating from the object detection model, with the classification result (pedestrian or vehicle). The object tracking model is used to follow the movements of the road users, and trackers’ ID numbers are displayed at the top left corner of the bounding box. The moving trajectories of the road users, with the latest ten movements) are also plotted.

![Figure 17 Automated object detection and tracking models.](image-url)
5.1.2 Perspective Transformation

The illustration of perspective transformation was presented in chapter 3.1.2. This step is used to map the coordinates generated from videos to the world coordinates (GPS coordinates in decimal degrees). The trajectories of the pedestrians and vehicles generated at the two studied intersections are plotted on Google Maps (Figure 18). The first intersection is located on US17-92 and 13th ST, and the second intersection is located on US17-92 and 3rd St, as illustrated in Table 15. Different road users are marked with different colors.
5.2 SSMs Generation and Threshold Selection

5.2.1 Automated Extraction of PET and TTC Values from Videos

With respect to how to calculate Post Encroachment Time (PET) from the video, Figure 19 shows an example. The color bar on the right denotes the frame number during this interaction between this pair of tracker with ID 3 and tracker with ID 9. The frame rate of the video is 30 Hz (30 frames per second). First, loop over the two
sequences of the trajectories (pedestrian and vehicle) to obtain the closest point (conflict point). Secondly, the frame numbers of these two trackers when approaching this conflict point are determined. Lastly, the time difference \( \frac{(577 - 500)}{30} = 2.57 \text{ s} \) are generated as PET value.

Figure 19 Example of generating PET value.

TTC is defined as the time required for two road users to collide if they continue at their present speeds and on the same paths (Hayward, 1972). The equation of deriving TTC is shown in (16). Since the GPS coordinates are in decimal degrees, it will generate some inaccuracies when calculating distances between different objects due to the curvature of the Earth. To eliminate this effect, the coordinates in GPS are further transformed into a plane coordinate grid system UTM (Universal Transverse Mercator) system (zone:17). The coordinates of pedestrian, vehicle, and their conflict point are \( (x_{ped}, y_{ped}) \), \( (x_{veh}, y_{veh}) \), and \( (x_{conflict}, y_{conflict}) \). The speed of the road user is defined as the distance that traveled between the frame number \( N - m \) and \( N \). Each video frame takes \( \frac{1}{30} \) second ((17)).
\[ \text{TTC} = \frac{\text{haversine distance}\left( (x_{\text{ped}}, y_{\text{ped}}), (x_{\text{conflict}}, y_{\text{conflict}}) \right)}{\text{speed}_{\text{ped}}} - \frac{\text{haversine distance}\left( (x_{\text{veh}}, y_{\text{veh}}), (x_{\text{conflict}}, y_{\text{conflict}}) \right)}{\text{speed}_{\text{veh}}} \]  

(16)

\[ \text{Speed} = \frac{\text{haversine distance}\left( (x_N, y_N), (x_N - m, y_N - m)\right)}{m \times \left( \frac{1}{20} \right)} \]  

(17)

As shown in Figure 20, the TTC values are continuously calculated over the whole process of the interaction, and the minimum of the TTC are determined to be 1.4s to represent the severity of the interaction between pedestrian and vehicle.

![Figure 20 TTC values from a pair of pedestrian & vehicle.](image)

The automated way of measuring PET is validated using 30 pairs of pedestrians and vehicles collected from over 20 intersections. The manually measured PET values can serve as ground truth values. A linear model (automated measured PET = manually measured PET) is built as shown in Figure 21. It can be found that the model has a good fit with R-squared value as 0.9759. So, the automated method of generating PET can achieve reasonable accuracy. However, the TTC values are hard to obtain manually. Thus, automated TTC values are not validated.
5.2.2 Threshold Selection

Traditional studies usually use single threshold values for TTC and PET. However, locations with different geometric designs or speed limits can have influence on the selection of the threshold values of TTC and PET. Upon selecting the threshold values to identify pedestrians’ critical situations, some studies investigated different threshold values of the SSMs indicators (Borsos et al., 2020; Mahmud et al., 2017). Among them, the Extreme Value Theory (EVT) was used. The EVT model could be employed to model the stochastic behaviors of extremely large or small values. As illustrated in Chapter 2, the traffic conflicts are “near” crashes, i.e., the boundary of PET/TTC to differentiate the conflicts and the crashes is zero. With the distributions of traffic conflict, the best fitted models can denote the correlations between the conflict and the frequency of actual crashes.

EVT offers two approaches to model extreme events, the block maxima (BM)
approach using Generalized Extreme Value (GEV) distribution, and the Peak over Threshold (POT) approach using Generalized Pareto (GP) distribution. As the typical EVT model is used to estimate the maxima of extreme values, here the minima of negated values of PET and TTC during the pedestrian-vehicle interactions are considered. GEV models and GP models are used for comparison.

5.2.2.1 Block Maxima (BM) Approach

For BM approach, the observations are aggregated into fixed intervals (block). So, every interaction between a pair of pedestrian and vehicle can be a block, and the extremes values of PET or TTC from each interaction are extracted from every interaction. Assume \( X \) is a variable with a certain probability distribution. And \( x_1, x_2, ..., x_n \) are independent random observations from \( X \). Let \( M = \max (x_1, x_2, ..., x_n) \). When \( n \to \infty \), the \( M \) will converge to a General Extreme Value (GEV) distribution that best illustrates the probabilities of those occurrences of extreme values. The standard GEV distribution is as follows:

\[
G(x) = \exp \left\{ - \left[ 1 + \varepsilon \left( \frac{x - \mu}{\sigma} \right) \right]^{-1/\varepsilon} \right\} \tag{18}
\]

With \(-\infty < \mu < \infty, \sigma > 0 \) and \(-\infty < \varepsilon < \infty \), three parameters are regarded as location parameter (\( \mu \)), the scale parameter (\( \sigma \)), and the shape parameter (\( \varepsilon \)). With the threshold value of \( \mu \), scale parameter \( \sigma_\mu > 0 \) (depending on threshold \( \mu \)), and shape parameter \(-\infty < \varepsilon < \infty \). When the shape parameter \( \varepsilon \) is equal to 0, the GEV tends to a Gumbel distribution; when \( \varepsilon > 0 \), GEV tends to the Frechet distribution; when \( \varepsilon < 0 \), GEV tends to a Weibull distribution.
5.2.2.2 Peak over Threshold (POT) Approach

For POT approach, an event is identified as an extreme case if it exceeds a predefined threshold $\mu$. With $\sigma > 0$ and $-\infty < \varepsilon < \infty$, the threshold excesses $(x - \mu)$ will converge to a GP distribution:

$$G(x) = 1 - \left[1 + \left(\frac{\varepsilon \cdot (x - \mu)}{\sigma}\right)\right]^{-1/\varepsilon} \tag{19}$$

For POT approach, the threshold values need to be predefined. And two parameters, scale parameter $\sigma$ and shape parameter $\varepsilon$ are to be estimated. The parameter stability plots can be used to determine the threshold values. Figure 22 (a) shows the parameter stability plots of negated PET for reparametrized scale ($\sigma^* = \sigma - \varepsilon \mu$) and shape ($\varepsilon$). A threshold value of about -7 or -5 seems appropriate, as $\sigma^*$ and $\varepsilon$ parameters seem to be stable within the ranges near -7 and -5, as the parameters are independent with the threshold value. For negated TTC, as shown in Figure 22 (b), the parameters show steady tendencies within the range near -3.

(a) GP model for PET  \hspace{1cm} (b) GP model for TTC

Figure 22 Parameter stability plots of PET and TTC.

In this study, the PET or TTC values for each interaction between a pair of
pedestrian and vehicle can be the extreme values to model. The threshold values of negated PET is selected within the range (-7, -5), such as -7, -6, -5. The threshold values of negated TTC are selected from -5, -4, and -3. Both GEV and GP models are established. For GEV model, three parameters, $\mu$, $\sigma$, and $\varepsilon$, are to be estimated, while for GP model, two parameters, $\sigma$, and $\varepsilon$ are estimated. Suppose $x$ is negated PET or negated TTC generated from pedestrian-vehicle interactions, the crash occurrence can be regarded as extreme cases, with $x (-PET) = 0$ or $x (-TTC) = 0$. The crash probability can be regarded as $Pr(x (-PET) > 0)$ or $Pr(x (-TTC) > 0)$, which can be calculated using the respective model. The Kolmogorov–Smirnov test is used with the null hypothesis that the true distribution of the samples is drawn from the hypothesized distribution. If the $p$-value is greater than 0.05, we cannot reject the null hypothesis. The modeling results are shown in Table 16 and Table 17. All statistical analysis is done using “ExtRemes” and “evd” packages from R (Gilleland and Katz, 2016; Team, 2013). With the AIC and BIC as evaluating metrics, four models are selected, as marked in bold. It should be noted that GEV model with $\text{threshold}_{PET} = -5$ is not selected, as the estimated standard error for shape parameter $\varepsilon$ is too large, which means the model is not well fitted.

The diagnostic plots, including the quantile plots and (kernel) probability density plots, for the four selected models are shown in Figure 23. The quantile plots compare between the quantiles of empirical data and quantiles of the fitted distribution, if the points are close to good linearity, the model has a good fit. The density plots
compare between and the histogram of the empirical data and the probability density function of the fitted model. For negated PET, the GEV model (Figure 23 (a)) has a better fit than the GP model (Figure 23 (c)), as the blue dotted line (fitted distribution) almost covers the histogram of the empirical data (black line). For negated TTC value, the GP model (Figure 23(d)) has a better fit than GEV model (Figure 23 (b)). Thus, the threshold value for PET is selected as 6 sec, and the threshold value for TTC is selected as 3 sec.
Table 16 Modeling results for negated PET with different threshold values.

<table>
<thead>
<tr>
<th>Model</th>
<th>Threshold (negated PET)</th>
<th>$-7$</th>
<th>$-6$</th>
<th>$-5$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sample size</td>
<td>73</td>
<td>67</td>
<td>34</td>
</tr>
<tr>
<td>GEV</td>
<td>Location parameter $\mu$</td>
<td>-4.540</td>
<td>-4.319</td>
<td>-3.792</td>
</tr>
<tr>
<td></td>
<td>(Standard error)</td>
<td>(0.182)</td>
<td>(0.168)</td>
<td>(0.138)</td>
</tr>
<tr>
<td></td>
<td>Scale parameter $\sigma$</td>
<td>1.3644</td>
<td>1.190</td>
<td>0.840</td>
</tr>
<tr>
<td></td>
<td>(Standard error)</td>
<td>(0.133)</td>
<td>(0.124)</td>
<td>(0.104)</td>
</tr>
<tr>
<td></td>
<td>Shape parameter $\varepsilon$</td>
<td>-0.233</td>
<td>-0.171</td>
<td>-0.007</td>
</tr>
<tr>
<td></td>
<td>(Standard error)</td>
<td>(0.097)</td>
<td>(0.110)</td>
<td>(0.139)</td>
</tr>
<tr>
<td></td>
<td>Probability of crash ($\text{PET} \geq 0$)</td>
<td>0.0017</td>
<td>0.0034</td>
<td>0.0101</td>
</tr>
<tr>
<td></td>
<td>AIC (Akaike information criterion)</td>
<td>263.425</td>
<td>228.786</td>
<td>152.340</td>
</tr>
<tr>
<td></td>
<td>BIC (Bayesian information criterion)</td>
<td>270.296</td>
<td>235.400</td>
<td>158.194</td>
</tr>
<tr>
<td></td>
<td>Negative Log-Likelihood Value</td>
<td>128.713</td>
<td>111.392</td>
<td>73.170</td>
</tr>
<tr>
<td></td>
<td>Kolmogorov-Smirnov test p-value</td>
<td>0.996</td>
<td>0.992</td>
<td>0.998</td>
</tr>
<tr>
<td>GP</td>
<td>Scale parameter $\sigma$</td>
<td>5.186</td>
<td>3.656</td>
<td>2.741</td>
</tr>
<tr>
<td></td>
<td>(Standard error)</td>
<td>(0.299)</td>
<td>(0.490)</td>
<td>(0.440)</td>
</tr>
<tr>
<td></td>
<td>Shape parameter $\varepsilon$</td>
<td>-0.828</td>
<td>-0.688</td>
<td>-0.628</td>
</tr>
<tr>
<td></td>
<td>(Standard error)</td>
<td>(0.0452)</td>
<td>(0.101)</td>
<td>(0.119)</td>
</tr>
<tr>
<td></td>
<td>Probability of crash ($\text{PET} \geq 0$)</td>
<td>0.076</td>
<td>0.051</td>
<td>0.047</td>
</tr>
<tr>
<td></td>
<td>AIC (Akaike information criterion)</td>
<td>269.362</td>
<td>219.566</td>
<td>147.529</td>
</tr>
<tr>
<td></td>
<td>BIC (Bayesian information criterion)</td>
<td>273.943</td>
<td>223.976</td>
<td>151.431</td>
</tr>
<tr>
<td></td>
<td>Negative Log-Likelihood Value</td>
<td>132.681</td>
<td>107.783</td>
<td>71.764</td>
</tr>
<tr>
<td></td>
<td>Kolmogorov-Smirnov test p-value</td>
<td>0.500</td>
<td>0.726</td>
<td>0.734</td>
</tr>
</tbody>
</table>
### Table 17 Modeling results for negated TTC with different threshold values.

<table>
<thead>
<tr>
<th>Model</th>
<th>Threshold (negated TTC)</th>
<th>(-5)</th>
<th>(-4)</th>
<th>(-3)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sample size</td>
<td>110</td>
<td>96</td>
<td>63</td>
</tr>
<tr>
<td>GEV</td>
<td>Location parameter (\mu)</td>
<td>-3.016</td>
<td>-2.723</td>
<td>-1.986</td>
</tr>
<tr>
<td></td>
<td>(Standard error)</td>
<td>(-0.136)</td>
<td>(0.126)</td>
<td>(0.105)</td>
</tr>
<tr>
<td></td>
<td>Scale parameter (\sigma)</td>
<td>1.231</td>
<td>1.038</td>
<td>0.713</td>
</tr>
<tr>
<td></td>
<td>(Standard error)</td>
<td>(0.105)</td>
<td>(0.099)</td>
<td>(0.080)</td>
</tr>
<tr>
<td></td>
<td>Shape parameter (\varepsilon)</td>
<td>-0.341</td>
<td>-0.287</td>
<td>-0.271</td>
</tr>
<tr>
<td></td>
<td>(Standard error)</td>
<td>(0.093)</td>
<td>(0.112)</td>
<td>(0.125)</td>
</tr>
<tr>
<td></td>
<td>Probability of crash ((-m_{TTC} \geq 0))</td>
<td>0.010</td>
<td>0.0076</td>
<td>0.0056</td>
</tr>
<tr>
<td></td>
<td>AIC (Akaike information criterion)</td>
<td>358.857</td>
<td>287.411</td>
<td>144.132</td>
</tr>
<tr>
<td></td>
<td>BIC (Bayesian information criterion)</td>
<td>366.959</td>
<td>295.105</td>
<td>150.562</td>
</tr>
<tr>
<td></td>
<td>Negative Log-Likelihood Value</td>
<td>176.429</td>
<td>140.706</td>
<td>69.066</td>
</tr>
<tr>
<td></td>
<td>Kolmogorov-Smirnov test p-value</td>
<td>0.754</td>
<td>0.677</td>
<td>0.836</td>
</tr>
<tr>
<td>GP</td>
<td>Scale parameter (\sigma)</td>
<td>4.694</td>
<td>3.073</td>
<td>2.475</td>
</tr>
<tr>
<td></td>
<td>(Standard error)</td>
<td>(2e-8)</td>
<td>(0.288)</td>
<td>(0.059)</td>
</tr>
<tr>
<td></td>
<td>Shape parameter (\varepsilon)</td>
<td>-0.994</td>
<td>-0.820</td>
<td>-0.906</td>
</tr>
<tr>
<td></td>
<td>(Standard error)</td>
<td>(2e-8)</td>
<td>(0.079)</td>
<td>(0.018)</td>
</tr>
<tr>
<td></td>
<td>Probability of crash ((-m_{TTC} \geq 0))</td>
<td>0.057</td>
<td>0.037</td>
<td>0.077</td>
</tr>
<tr>
<td></td>
<td>AIC (Akaike information criterion)</td>
<td>345.640</td>
<td>254.126</td>
<td>130.052</td>
</tr>
<tr>
<td></td>
<td>BIC (Bayesian information criterion)</td>
<td>351.040</td>
<td>259.254</td>
<td>134.339</td>
</tr>
<tr>
<td></td>
<td>Negative Log-Likelihood Value</td>
<td>170.820</td>
<td>125.063</td>
<td>63.026</td>
</tr>
<tr>
<td></td>
<td>Kolmogorov-Smirnov test p-value</td>
<td>0.641</td>
<td>0.993</td>
<td>0.940</td>
</tr>
</tbody>
</table>
Figure 23 Diagnostic plots for GEV and GP models.

5.3 Input Variables Overview

The generating rate of the trajectories from videos is 30 records per second. The input variables are the mobility features of the interacting pedestrians and vehicles, such
as the traveling courses, speed, distance between the road user and the conflict point.

The feature vectors from the trajectory data are used as input for the prediction of the pedestrians’ near-crash events. As mentioned above, the threshold values of PET and TTC are 6 sec and 3 sec, respectively. So, the categories of the dependent variables are defined accordingly: when PET value is smaller than 6 sec, and TTC value is smaller than 3 sec, the interaction is defined as “serious conflict”; when one of the indicators is smaller than the threshold value, the interaction is defined as “slight conflict”; when neither of the two indicators is smaller than the threshold values, the case is regarded as “safe”. The descriptive statistics of the variables used are listed in Table 18.

**Table 18 Descriptive statistics of variables.**

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Range</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course (ped)</td>
<td>The traveling directions</td>
<td>(10.95, 347.31)</td>
<td>Degree</td>
</tr>
<tr>
<td>Course (veh)</td>
<td>The traveling directions</td>
<td>(0.19, 359.99)</td>
<td>Degree</td>
</tr>
<tr>
<td>Speed (ped)</td>
<td>The traveling speed</td>
<td>(0, 6.39)</td>
<td>Ft/s</td>
</tr>
<tr>
<td>Speed (veh)</td>
<td>The traveling speed</td>
<td>(0, 50.83)</td>
<td>Ft/s</td>
</tr>
<tr>
<td>Distance to conflict point (ped)</td>
<td>The distance between the pedestrian and conflict point</td>
<td>(0, 154.73)</td>
<td>Ft</td>
</tr>
<tr>
<td>Distance to conflict point (veh)</td>
<td>The distance between the vehicle and conflict point</td>
<td>(0, 220.34)</td>
<td>Ft</td>
</tr>
<tr>
<td>Near-crash events*</td>
<td>Category of events (Safe; Slight conflict; Serious conflict)</td>
<td>(safe: 16799; slight: 5179; serious: 3280)</td>
<td>-</td>
</tr>
</tbody>
</table>

Note: *marked is the response variable.

The pedestrians’ and vehicles’ features before reaching the conflict points 2 sec ahead are generated to feed into the model. There are 25,258 records in the data set, with a ratio of 5.1:1.6:1 between the targeted classes “safe”, “slight conflict”, and “serious conflict”. Eighty percent of the data are used as the training data set, and twenty
percent of the data is used as the test data set. An over-sampling method SMOTE is used on the training data set to increase the number of records for the two minority classes, i.e., “slight conflict”, and “serious conflict”.

5.4 Deep Learning Models

Two deep learning models, LSTM and GRU, and one machine learning Support Vector Machine (SVM) are used. The tuning procedures of the models are shown below.

5.4.1 Support Vector Machine (SVM)

The illustration of SVM model can be referred to previous chapter.

5.4.2 Gated Recurrent Unit (GRU)

As trajectory data are in time series, recurrent neural networks can be used to better handle sequential data (Figure 24). Different from traditional neural networks, the output of the recurrent neural network from the current time slice is the input of the next time slice. Suppose $x_1, x_2, x_3$ are the input vectors, $h_1, h_2, h_3$ are the hidden state vectors, and $y_1, y_2, y_3$ are the output vectors. At the time slice $t$, the hidden state vectors $h_t$ are computed by the input vector $x_t$, the previous hidden state vector $h_{t-1}$, and the weights $w_t^x$ and $w_t^h$. Thus, the output is produced by joining hidden layer vectors together with input from previous time slices. That’s why the recurrent neural network can memorize the sequential information lying in the time series data.
However, when dealing with data in a long sequence, recurrent neural network models can suffer from vanishing gradient problems, thus mitigating the ability of the model to learn long-term information (Pascanu et al.). The Gated Recurrent Unit (GRU) is proposed to solve the long-term dependency problem of recurrent neural networks (Cho et al., 2014). The GRU model consists of two gates, reset gate, and update gate, as shown in Figure 25. The update gate controls the previous information that will be carried over to the current layer, while the reset gate decides the amount of information to forget. The equation of the update gate $z_t$ and the reset gate $r_t$ are shown in (20) and (21), respectively. The input vector at time $t$ is given by $x_t$. When $x_t$ is fed into the network unit, it is multiplied by the weight $W_z$. And $h_{t-1}$ generated from the last hidden layer is multiplied by its weight $U_z$. The two results are added together and a sigmoid function $\sigma$ is used as the activation function to generate a probabilistic value between 0 and 1. So the values of update gate $z_t$ and the reset gate $r_t$ are generated. The weight vectors $W_z, U_z, W_r, U_r$ are learned through the training process. $\tilde{h}_t$ is the memory unit that can store the relevant information using the reset gate $r_t$. $\sigma$ is the sigmoid function and $\otimes$ is the element-wise product function of two vectors. $W_c$ and
$U_r$ are weight matrices that are learned through the process. The hidden layer output $h_t$ at time $t$ is calculated by (23). These equations are iteratively computed from the first time slice to the last time slice, and finally generate the output of GRU.

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \quad (20)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \quad (21)$$

$$\tilde{h}_t = tanh(W_c x_t + r_t \otimes (U h_{t-1})) \quad (22)$$

$$h_t = (1 - z_t) \otimes h_{t-1} + z_t \otimes \tilde{h}_t \quad (23)$$

5.4.3 Long Short-Term Memory (LSTM)

The LSTM model was illustrated in chapter 3.2.

5.5 Experiment Results

The evaluating metrics, such as recall, false alarm rate (FAR), accuracy, and AUC are illustrated in the previous chapters. Except for FAR, other metrics have been illustrated in previous chapters.

$$FAR = \frac{FP}{FP + TN} \quad (24)$$

In this study, the dependent variable is divided into three classes, safe, slight conflict, and serious conflict. The last class is the most critical class. So, the model’s performance of this class should be put more emphasis on. Meanwhile, the average
value of metrics over three classes, which is usually called macro average value is also calculated. The modeling results of the three models are listed in Table 19.

Table 19 Classification report for three used models.

<table>
<thead>
<tr>
<th>Model</th>
<th>Training set</th>
<th>Test set</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUC</td>
<td>Recall</td>
<td>FAR</td>
<td>Accuracy (with cross-validation)</td>
<td>AUC</td>
</tr>
<tr>
<td>GRU</td>
<td>Average</td>
<td>0.876</td>
<td>0.856</td>
<td>0.138</td>
<td>0.878</td>
</tr>
<tr>
<td></td>
<td>Serious</td>
<td>0.810</td>
<td>0.189</td>
<td></td>
<td></td>
</tr>
<tr>
<td>LSTM</td>
<td>Average</td>
<td>0.868</td>
<td>0.818</td>
<td>0.168</td>
<td>0.861</td>
</tr>
<tr>
<td></td>
<td>Serious</td>
<td>0.813</td>
<td>0.186</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td>Average</td>
<td>0.785</td>
<td>0.560</td>
<td>0.252</td>
<td>0.580</td>
</tr>
<tr>
<td></td>
<td>Serious</td>
<td>0.515</td>
<td>0.304</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: the bold marked is the best model.

5.6 Summary

In this chapter, GRU is used to predict the pedestrians’ conflicts at signalized intersections. With PET and TTC indicators generated from videos, Extreme Value Theory is used to select the best threshold values. The near-crash events of pedestrians are classified into three severity categories. With the sequential data generated from pedestrians’ and vehicles’ trajectories, the GRU model reaches an AUC value of 0.865 on the test data set. The proposed model can be used to warn drivers of the potentially dangerous situations involving pedestrians if to be implemented in the CV environment.
CHAPTER 6 PREDICTING PEDESTRIANS’ CONFLICTS AT THE SIGNAL CYCLE LEVEL

In this chapter, pedestrians’ conflicts are extracted at the signal cycle level from CCTV videos. Multiple data sources, CCTV videos, ATSPM®, and geometrical variables are used. The variables, such as the temperature, visibility, signal cycle length, vehicle counts at current cycle, and median presence, etc., are extracted. Compared with last chapter, this proposed model mostly uses variables extracted from the infrastructure (ATSPM®, weather stations, etc.), instead of CCTV videos. Thus, the computation cost is much lower to be implemented in a real-time system.

6.1 Data Collection

CCTV videos from 25 intersections located in Orlando, FL are collected. The numbers of intersections of different types, the collected video lengths, and the pedestrian volume (on sidewalks and crosswalks) are shown in Table 20.

Table 20 Overview of collected videos.

<table>
<thead>
<tr>
<th>Intersection type (cross section)</th>
<th>Intersection number</th>
<th>Video length</th>
<th>Pedestrian volume (total)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Six-lane by six -lane</td>
<td>2</td>
<td>18 hours</td>
<td>78</td>
</tr>
<tr>
<td>Six-lane by two-lane</td>
<td>2</td>
<td>18 hours</td>
<td>34</td>
</tr>
<tr>
<td>Four-lane by four -lane</td>
<td>10</td>
<td>135 hours</td>
<td>750</td>
</tr>
<tr>
<td>Four-lane by two-lane</td>
<td>6</td>
<td>45 hours</td>
<td>250</td>
</tr>
<tr>
<td>Four-lane by three-lane</td>
<td>1</td>
<td>9 hours</td>
<td>9</td>
</tr>
<tr>
<td>Three-lane by three-lane</td>
<td>2</td>
<td>18 hours</td>
<td>52</td>
</tr>
<tr>
<td>Two-lane by two-lane</td>
<td>2</td>
<td>18 hours</td>
<td>40</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>25</strong></td>
<td><strong>261 hours</strong></td>
<td><strong>1213</strong></td>
</tr>
</tbody>
</table>
6.1.1 Variable Extraction

Figure 26 shows a signalized intersection. The major road is in north-south direction, and the minor road is in east-bound direction. Suppose a pedestrian wants to cross the major road, the pedestrian traveling on the designated crosswalk can have conflicts with the left-turn vehicles from east direction (WL), and the right-turn vehicles from south direction (NR). The vehicular volumes with different turning movements can be extracted from ATSPM©. ATSPM© is one kind of high-resolution event-based data source. As shown in Figure 27, there are four columns in the ATSPM© data, “signal ID”, “timestamp”, “EventCode”, and “EventParam”. For example, “EventCode” equals 2 denotes the vehicles’ northbound through movement on the major road (NT), then the green time for NT can be calculated as the time interval between “EventParam” turns from 1 to 7. On the other hand, the column “EventParam” can also denote the ID number of installed loop detector located on each lane. Suppose the detector ID is 3 for the right most lane from northbound direction (NR), the count of right-turn vehicles of the current signal cycle is the count number of “EventCode” turns from 82 to 81. Table 21 shows the calculations of the variables from ATSPM©.
Figure 26 Example of a crossing pedestrian.

Figure 27 An example of ATSPM© record.
<table>
<thead>
<tr>
<th>Variables</th>
<th>Description</th>
<th>Used EventCode</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pedestrian signal phase</td>
<td>Pedestrian signal phase exists in the cycle.</td>
<td>21: pedestrian phase on</td>
<td>-</td>
</tr>
<tr>
<td>Cycle length</td>
<td>Time difference between the cycle starts and ends.</td>
<td>1: phase begin green</td>
<td>Cycle length = timestamp_1,2 − timestamp_1,1</td>
</tr>
<tr>
<td>Green time (NT)</td>
<td>Time difference between the green light starts and ends (yellow time starts).</td>
<td>1: phase on 8: phase begin yellow</td>
<td>Green time = timestamp_8,2 − timestamp_8,1</td>
</tr>
<tr>
<td>Yellow time (NT)</td>
<td>Time difference between the phase starts yellow and ends (red time starts).</td>
<td>8: phase begin yellow 10: phase begin red</td>
<td>Yellow time = timestamp_10,2 − timestamp_10,1</td>
</tr>
<tr>
<td>Red time (NT)</td>
<td>Time difference between the phase starts red and ends (green time starts).</td>
<td>10: phase begin red 1: phase begin green</td>
<td>Red time = timestamp_10,2 − timestamp_1,1</td>
</tr>
<tr>
<td>Vehicle counts (NR, during green light)</td>
<td>Vehicle counting during green light between detector on and off for detector 3.</td>
<td>1: phase begin green 8: phase begin yellow 81: detector off 82: detector on</td>
<td>(\text{Vehicle count} = \sum_{timestamp_{i,j}} EventCode_{82-81})</td>
</tr>
</tbody>
</table>

Note: Timestamp\(_{i,j}\) (i: EventCode, j: EventParam)

### 6.1.2 Data Integration

Two SSMs indicators, PET and TTC are derived from CCTV videos. The illustrations of generating indicators are presented in the Chapter 5.2. The threshold values of the indicators need to be determined. Using the procedures illustrated in the last chapter, TTC threshold is determined to be 4 sec and PET threshold is determined to be 4.5 sec.
Geometrical factors, such as the presence of hard median, and if the crosswalk is located on major/minor road are also used. Other variables include temperature, AADT (Average Annual Daily Traffic) on the major and minor road, whether a bus stop is located within the 200 ft of the intersection, hour in the day, speed limit on the major road and minor road, etc. For prediction purpose, the variables from the last prior cycle are used to model the safety condition of the current cycle. In total, 944 signal cycles are collected. Most of them contain pedestrian phases. However, the others don’t have pedestrian signal phases, but pedestrians are detected from videos, which means the pedestrians cross during red-light signals. An overview of the modeling workflow is shown in Figure 28.

Figure 28 Modeling workflow.

With the threshold values of PET and TTC determined, the pedestrians’ signal cycles are labeled as “conflict” or “no conflict”. Most signal cycles do not contain pedestrian-vehicle conflicts. The plot of the variable correlations (Pearson’s correlation coefficients) are shown in Figure 29. It can be found that the median presence has high
correlations with two variables, speed limit on minor road (coefficient equals -0.77) and the presence of bus stop (coefficient equals -0.65). The daily exposure of pedestrian trips also has high correlation with hourly exposure (coefficient equals 0.76). The illustration of the variables is shown in Table 22.

![Pearson correlation coefficient](image_url)
Table 22 Input variable overview.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Range</th>
<th>Unit</th>
</tr>
</thead>
<tbody>
<tr>
<td>AADT_MAJOR</td>
<td>AADT of major road</td>
<td>(5,611, 56,333)</td>
<td>Veh</td>
</tr>
<tr>
<td>AADT_MINOR</td>
<td>AADT of minor road</td>
<td>(5,200, 37,833)</td>
<td>Veh</td>
</tr>
<tr>
<td>Median</td>
<td>Whether the crosswalk has a median</td>
<td>(0, 1)</td>
<td>-</td>
</tr>
<tr>
<td>Settings</td>
<td>The crosswalk is on major/minor road</td>
<td>(0, 1)</td>
<td>-</td>
</tr>
<tr>
<td>Bus_stop</td>
<td>Whether a bus stop is located within 200 ft of intersection</td>
<td>(0, 1)</td>
<td>-</td>
</tr>
<tr>
<td>SPEED_MAX_MAJOR</td>
<td>Speed limit of the major road</td>
<td>(35, 55)</td>
<td>Mph</td>
</tr>
<tr>
<td>SPEED_MAX_MINOR</td>
<td>Speed limit of the minor road</td>
<td>(35, 55)</td>
<td>Mph</td>
</tr>
<tr>
<td>Hour</td>
<td>Hour in the day</td>
<td>(7, 18)</td>
<td>-</td>
</tr>
<tr>
<td>Exposure (pedestrian trips)</td>
<td>Hourly pedestrian phases provided (ATSPM©)</td>
<td>(0, 29)</td>
<td>-</td>
</tr>
<tr>
<td>Exposure_daily (pedestrian trips)</td>
<td>Daily pedestrian phases provided (ATSPM©)</td>
<td>(0, 216)</td>
<td>-</td>
</tr>
<tr>
<td>Cycle_length</td>
<td>Signal cycle length</td>
<td>(59, 239)</td>
<td>-</td>
</tr>
<tr>
<td>Redlight</td>
<td>Whether the pedestrian crosses during red light</td>
<td>(0, 1)</td>
<td>-</td>
</tr>
<tr>
<td>Temperature</td>
<td>Temperature</td>
<td>(42, 80)</td>
<td>Fahrenheit</td>
</tr>
<tr>
<td>Count_major</td>
<td>Vehicle count on major road (NT+ST)</td>
<td>(9, 411)</td>
<td>Veh</td>
</tr>
<tr>
<td>Count_minor</td>
<td>Vehicle count on minor road (ET+WT)</td>
<td>(0, 128)</td>
<td>Veh</td>
</tr>
<tr>
<td>Count_left_green/yellow/red</td>
<td>Vehicle count (NL movement, arriving during green/yellow/red light)</td>
<td>(1, 11)</td>
<td>Veh</td>
</tr>
<tr>
<td>Count_right_green/yellow/red</td>
<td>Vehicle count (NR movement, arriving during green/yellow/red light)</td>
<td>(0, 10)</td>
<td>Veh</td>
</tr>
<tr>
<td>Gr_major</td>
<td>Green signal time (vehicle) on major road</td>
<td>(27, 178)</td>
<td>Sec</td>
</tr>
<tr>
<td>Gr_minor</td>
<td>Green signal time (vehicle) on minor road</td>
<td>(7, 86)</td>
<td>Sec</td>
</tr>
<tr>
<td>Conflict*</td>
<td>Conflict exists or not (PET &lt; 4.5 sec or TTC &lt; 4 sec)</td>
<td>(&quot;conflict&quot; : 92; “no conflict”: 852)</td>
<td></td>
</tr>
</tbody>
</table>
6.2 Modeling and Results

Four machine learning models, Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting, and Extreme Gradient Boosting (XGBoost) are used. The details of the models are illustrated in the Chapter 4.2.

6.2.1 Re-sampling methods

The generated data set contains 944 samples (signal cycles), and the ratio between conflict and no conflict class is around 1:9. The usual approaches of dealing with imbalanced data set is to apply re-sampling strategies to obtain a more balanced data distribution. As shown in Figure 30, the undersampling strategy randomly takes samples from the majority class, decreases the size of the majority class, and makes two classes more balanced. And the oversampling strategy randomly adds new samples from the minority class and makes the two classes more balanced. Both strategies can be repeated until the training dataset achieves desired distribution, such as the sample sizes of two classes equal each other.

![Figure 30 Undersampling and oversampling.](image)

The data set is split into training data set (70%, 660 samples) and test set (30%,
284 samples). Originally, the training data set has 591 samples in “no conflict” class and 69 samples in “conflict” class. In this work, three methods are compared to balance the two classes, all methods are from “imblearn” and “scikit-learn” packages in Python:

(1) Random undersampling, randomly selecting samples from the majority class with bootstrap. The ratio between “conflict” and “no conflict” class becomes 69:138 (1:2) on training data set.

(2) Random oversampling, randomly selecting minority samples with replacement and adding them into the training data set. Normal bootstrap is generated without perturbation. The ratio between “conflict” and “no conflict” class becomes 591: 591 (1:1) on training data set.

(3) Borderline-SMOTE oversampling (Nguyen et al., 2011), generating new minority samples near the decision borderline so as to help establish boundary between the two classes. The ratio between “conflict” and “no conflict” class becomes 591: 591 (1:1) on training data set.

6.2.2 Experiments and Results

Oversampling or undersampling methods are used on the training dataset. The test data set still contain the original data without using any method. The evaluation matrix include recall, FAR (false alarm rate), accuracy, and AUC.

Based on the results, among the three resampling methods, random oversampling achieves the best performance. XGBT model with random oversampling achieves the AUC value as 0.838 and recall value of 0.826 on the test data set. The
confusion matrix on the test data set is shown in Figure 31. Nineteen samples from the conflict class are correctly classified (83%), while four samples are wrongly classified (17%). The variable importance plot is shown in Figure 32. It can be found that pedestrians’ red-light crossing is the most important variable. And the vehicle count on the major road, hour in the day, count of right-turn vehicles arriving during yellow light, and green time on the minor road are also playing important roles.

<table>
<thead>
<tr>
<th>Metric/model</th>
<th>Resampling</th>
<th>SVM</th>
<th>RF</th>
<th>GBM</th>
<th>XGBT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Recall</td>
<td>Random under</td>
<td>0.826</td>
<td>0.782</td>
<td>0.783</td>
<td>0.782</td>
</tr>
<tr>
<td></td>
<td>Random over</td>
<td>0.826</td>
<td>0.826</td>
<td>0.826</td>
<td>0.826</td>
</tr>
<tr>
<td></td>
<td>Borderline-SMOTE</td>
<td>0.786</td>
<td>0.750</td>
<td>0.750</td>
<td>0.821</td>
</tr>
<tr>
<td>FAR</td>
<td>Random under</td>
<td>0.184</td>
<td>0.215</td>
<td>0.238</td>
<td>0.233</td>
</tr>
<tr>
<td></td>
<td>Random over</td>
<td>0.195</td>
<td>0.169</td>
<td>0.142</td>
<td>0.149</td>
</tr>
<tr>
<td></td>
<td>Borderline-SMOTE</td>
<td>0.222</td>
<td>0.254</td>
<td>0.262</td>
<td>0.176</td>
</tr>
<tr>
<td>Accuracy</td>
<td>Random under</td>
<td>0.817</td>
<td>0.785</td>
<td>0.764</td>
<td>0.767</td>
</tr>
<tr>
<td></td>
<td>Random over</td>
<td>0.806</td>
<td>0.831</td>
<td>0.856</td>
<td>0.849</td>
</tr>
<tr>
<td></td>
<td>Borderline-SMOTE</td>
<td>0.778</td>
<td>0.746</td>
<td>0.739</td>
<td>0.824</td>
</tr>
<tr>
<td>AUC</td>
<td>Random under</td>
<td>0.821</td>
<td>0.784</td>
<td>0.773</td>
<td>0.774</td>
</tr>
<tr>
<td></td>
<td>Random over</td>
<td>0.815</td>
<td>0.829</td>
<td>0.842</td>
<td>0.838</td>
</tr>
<tr>
<td></td>
<td>Borderline-SMOTE</td>
<td>0.782</td>
<td>0.748</td>
<td>0.744</td>
<td>0.823</td>
</tr>
</tbody>
</table>
6.3 Summary

In this chapter, an approach of predicting pedestrians’ conflicts at the signal cycle level is proposed with the variables fed from ATSPM© and weather data. The model has the potential to be extended and implemented in an I2V system to provide pre-warnings to the drivers nearby, thus better preventing pedestrian crashes. Compared with the last chapter (predicting 2-5 sec ahead), the model can predict one signal cycle ahead, which can be last 2-3 min.
CHAPTER 7 CONCLUSION

This dissertation aims to improve pedestrian safety using video data, surrogate safety measures (SSMs), and deep learning models. With the P2I technologies, the proposed models can be used to identify pedestrians’ situations and send warnings to the drivers. The conclusions of this dissertation are summarized as below:

In Chapter 2, a literature review is conducted from three perspectives, pedestrian safety analysis using SSMs, computer vision applications, and deep learning models used in the transportation field. Several research gaps are identified.

In chapter 3, an LSTM neural network model is employed to predict pedestrians’ conflicts in time series. An SSM indicator, PET is manually labeled. Conflict at the individual vehicle level are further predicted.

In chapter 4, pedestrians’ red-light crossing intentions are predicted in time series. Pose estimation is used to process videos.

In chapter 5, two SSMs indicators, PET and TTC are used to label pedestrians’ conflicts. The conflicts are further predicted at the individual vehicle level.

In chapter 6, pedestrians’ conflicts are aggregated and predicted at the signal cycle level. The input variables used are generated from ATSPM© and weather data, which will reduce computational cost compared with variables from videos.

In conclusion, this dissertation uses automated video processing techniques, such as YOLO, Mask R-CNN to process videos. The generated data set are pedestrians’
trajectories. Furthermore, the models are proposed to predict pedestrian conflicts, from the individual vehicle level and signal cycle level. Other data source such as ATSPM© are used to integrate with video data. The proposed models have the potential to be extended and implemented in infrastructure-to-pedestrian (I2V) systems to provide pre-warnings to the drivers nearby, thus better preventing pedestrian crashes. The decision makers can determine the possible countermeasures at intersections.

The limitations of the dissertation are summarized as below. First, the developed models are mainly used for signalized intersections. However, the pedestrians’ fatalities at the road segments are also. Second, the developed models are all conflict-based. However, the relationship between traffic conflicts and crashes needs to be further investigated. For variables generated from ATSPM©, the effect of and adaptive signal control and signal coordination are not taken into consideration.

Further research may consider implementing the developed models to the real-time application. And the transferability of the model needs to be investigated using data from more intersections.
REFERENCES


CMU-Perceptual-Computing-Lab, Openpose.


Hediyeh, H., Sayed, T., Zaki, M.H., Ismail, K., 2014a. Automated analysis of pedestrian crossing speed behavior at scramble-phase signalized intersections using computer
vision techniques. *International journal of sustainable transportation* 8(5), 382-397.


Islam, Z., Abdel-Aty, M., 2021b. Sensor-Based Transportation Mode Recognition


Anticipating Maneuvers via Learning Temporal Driving Models.


Li, P., Abdel-Aty, M., Cai, Q., Yuan, C., 2020b. The application of novel connected vehicles emulated data on real-time crash potential prediction for arterials. Accident Analysis & Prevention 144, 105658.


Li, P., Abdel-Aty, M., Yuan, J., 2020c. Real-time crash risk prediction on arterials based on LSTM-CNN. Accident Analysis & Prevention 135, 105371.

Li, P., Abdel-Aty, M., Yuan, J., 2021b. Using bus critical driving events as surrogate safety measures for pedestrian and bicycle crashes based on GPS trajectory data. Accident Analysis & Prevention 150, 105924.


108


Mohamed, M.G., Saunier, N., 2013. Motion prediction methods for surrogate safety


Pascanu, R., Mikolov, T., Bengio, Y., 2013. On the difficulty of training recurrent neural


Record 2264(1), 148-155.


Schulz, A.T., Stiefelhagen, R., 2015b. Pedestrian intention recognition using latent-


Van Haperen, W., Riaz, M.S., Daniels, S., Saunier, N., Brijs, T., Wets, G., 2019. Observing the observation of (vulnerable) road user behaviour and traffic safety: A scoping review. Accident Analysis & Prevention 123, 211-221.


Wu, Y., Abdel-Aty, M., Zheng, O., Cai, Q., Yue, L., 2019b. Developing a Crash Warning System for the Bike Lane Area at Intersections with Connected Vehicle Technology. 

*Transportation Research Record*, 0361198119840617.


You, C.-W., Lane, N., Chen, F., Wang, R., Chen, Z., Bao, T., Montes de Oca, M., Cheng,


