Analysis of Impacts of Pavement Quality Deterioration on Recurring Traffic Congestion

DISCLAIMER

ACKNOWLEDGEMENT OF SPONSORSHIP AND STAKEHOLDERS

This work was sponsored by a grant from the Southeastern Transportation Research, Innovation, Development, and Education Center (STRIDE).

You can also acknowledge any other significant contributors to this study.

Funding Agreement Number - 69A3551747104 [This number is permanent and required by USDOT on each final report.]

LIST OF AUTHORS

Lead PI:

Muhammad M. Sherif, Ph.D.
Assistant Professor
Department of Civil, Construction, and Environmental Engineering
University of Alabama at Birmingham
msherif@uab.edu
0000-0001-9691-7953

Co-PI:

Virginia P. Sisiopiku, Ph.D.
Professor
Department of Civil, Construction, and Environmental Engineering
University of Alabama at Birmingham
vsisiopi@uab.edu
0000-0003-4262-8990

Additional Researchers:
Afrin Anni, MS
Department of Civil, Construction, and Environmental Engineering
University of Alabama at Birmingham
aanni@uab.edu

Ebisidor J. Agberebi, MS
Department of Civil, Construction, and Environmental Engineering
University of Alabama at Birmingham
ebi1@uab.edu

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The objective of this proposal is to evaluate the impact of pavement deteriorations on traffic congestion. Furthermore, the proposal investigates the feasibility of developing and implementing machine learning models for detecting pavement cracks and deterioration from google street view images. The crack detection results indicate that the machine learning algorithms had on average a mean intersection over union (MIOU) of 71.0% and 63.5% for the training and testing datasets, respectively. Two main corridors in Birmingham, AL, US-31 and I-65 were selected for this project. The Travel Time Index (TTI) was computed and used to quantify traffic congestion. Although the research hypothesis proposed a positive correlation between the two indices, suggesting that higher travel time would be associated with a higher crack ratio, the results obtained were mixed and counterintuitive.

In this project, the developed machine learning models and methodologies could be used for crowd-sensing of pavement quality using on-board cameras of smart vehicles. Implementing the methodologies will assist in prioritization of pavement improvements to improve efficiency and mitigate traffic congestion. The successful implementation of the machine learning methodologies can benefit the department of transportation to reduce labor costs associated with pavement survey through implementing neural networks for processing publicly-sources images. Also, as a return of investment (ROI), the implementation of the proposed framework will allow the agencies to prioritize the resurfacing and repair of road sections that experience heavy-traffic due to high traffic volumes, and percentage of trucks using the segment.

It is recommended to expand the analysis on different types of corridors and vastly assess traffic and pavement data on a regional level (i.e., city and state). Also, it is essential to consider the weather patterns, pavement maintenance data and scheduling, and traffic patterns.

Keywords: Crack Detection, Machine Learning, Traffic Congestion, Neural Networks, Pavement Quality

EXECUTIVE SUMMARY

The US has an extensive road network encompassing 4.2 million miles with 220,000 miles of high-volume corridors. Studies confirm that surface deficiencies of transportation infrastructure cause non-recurrent congestion as they are a contributing factor to 1/3 of all severe traffic crashes in the US. It is argued that poor pavement conditions are also linked to recurrent congestion, when drivers adjust their driving behaviors abruptly to avoid pavement cracks and other infrastructure deficiencies. However, those links are not documented in detail in the literature. Using Google Street View images and machine learning algorithms to assess pavement quality and the National Performance Management Research Data Set (NPMRDS) for congestion monitoring, this project will attempt to quantify the extent to which pavement quality deterioration has an impact on traffic congestion in urban settings. The goal is to provide insight on the links between pavement quality and traffic operations and showcase the feasibility and value of an integrated traffic data and pavement quality management approach for monitoring traffic and infrastructure performance.

The acquired google street view images were processed using machine learning algorithms to quantify the number of pixels that represent a crack for each image representing a pavement segment. A total of four machine learning algorithms were trained and tested to enhance the accuracy and precision in detecting pixels associated with a crack. One of the machine learning algorithms (UABiNet) was developed by the research team to enhance the prediction metrics and reduce the prediction time. A crack index representing a ratio of the pixels of cracks to the total pixels of pavement was calculated for each segment. The crack detection results indicate that the machine learning algorithms had on average a mean intersection over union (MIOU) of 71.0% and 63.5% for the training and testing datasets, respectively. It is worth noting, that the developed machine learning algorithm (UABiNet) outperformed the other algorithms with a MIOU of 80% and 65%, for the training and testing datasets, respectively.

Two main corridors in Birmingham, AL, US-31 and I-65 were selected for this project to assess the impact of pavement quality on traffic congestion. The Travel Time Index (TTI) was computed and used to quantify traffic congestion. Although the research hypothesis proposed a positive correlation between the two indices, suggesting that higher travel time would be associated with a higher crack ratio, the results obtained were mixed and counterintuitive. In the year-by-year analysis, statistically significant findings were observed for 2020 and 2021. However, the correlations were inverse and weak. Similarly, when focusing solely on I-65, statistically weak positive relationships were discovered in 2019, while statistically negative and poor relationships were found in 2021. It is worth noting that the traffic patterns for the years between 2020 and 2022 are heavily altered due to the COVID-19 pandemic and shift towards remote working environments.

1 Introduction

Traffic congestion is a common phenomenon in urban areas. Many definitions have been proposed to describe traffic congestion on roadways in urban areas. However, there is not a universally accepted definition of traffic congestion [1-3]. From the demand-capacity perspective "Congestion may be defined as the state of traffic flow on a transportation facility characterized by high densities and low speeds, relative to some chosen reference state" [4, 5]. From the delay-travel time related perspective "In the transportation realm, congestion usually relates to an excess of vehicles on a portion of roadway at a particular time resulting in speeds that are slower—sometimes much slower—than normal or "free flow" speeds" [6, 7]. As mentioned in GEOTAB 2018 [8], there are two distinct types of traffic congestion, i.e., recurring (expected and often associated with commuting traffic) and non-recurring (unexpected or associated with non-typical events such as traffic crashes, work zones, adverse weather events, and planned special events).

Effective investment decisions for transportation improvements require monitoring of various aspects of transportation system performance, including congestion [9, 10]. Traffic congestion is closely linked to safety, physical condition, environmental quality, economic development, quality of life, and customer satisfaction [11, 12]. Higher levels of congestion have been associated with the degradation of these factors. Therefore, monitoring congestion and transportation infrastructures is crucial in addressing these issues and making informed decisions to improve transportation systems.

This research aims to offer an understanding of the connections between pavement quality and traffic operations, while demonstrating the practicality and benefits of adopting an integrated approach to managing pavement quality and traffic data. This approach can be useful for monitoring the performance of both traffic and infrastructure. The quality of roads, which includes pavement quality, can influence driving decisions such as selecting speed, deceleration, and lane changing, ultimately impacting traffic flow and congestion [13-15]. In some cases, sudden reductions in traffic speed to avoid pavement cracks can cause a ripple effect, leading to further deterioration of traffic conditions, especially during peak hours [16]. Recent surveys conducted on Italian motorways revealed that top-down cracking (TDC) can impact as much as 20% to 30% of the slow traffic lane [17, 18]. To investigate the possible correlation between pavement quality deterioration and recurring traffic congestion, this project focuses on specific roadway segments in the Birmingham, AL metropolitan area as a testbed.

2 Objectives and Motivations

Many studies recognize the impact of driving behavior on quality of traffic operations. Road conditions, including pavement quality, may affect driving decisions related to speed selection, deceleration, and lane changing, thus having an impact on the quality of traffic flow and congestion occurrence. For example, abrupt reduction in traffic speed to evade pavement cracks can have a ripple effect that may contribute to deterioration of traffic conditions, especially in busy times. Using selected roadway segments in the Birmingham, AL metro as a testbed, this project seeks to examine closely the likely association between deterioration of pavement quality and recurring traffic congestion. More specifically, the project has a) assessed the impact of the quality of the roadway surfaces on traffic congestion, and b) showcased an integrated traffic data and pavement quality management approach for monitoring traffic and infrastructure performance. Our case study has demonstrated how traffic performance indices derived from National Performance Management Research Data Set (NPMRDS) can be paired to pavement surface data obtained through Google Street-View Images, and used to explain traffic congestion presence at locations that experience deteriorating pavement conditions. Initially, the Travel Time Index (TTI) values were computed for study roadway segments in the Birmingham region from NPMRDS data. A convolution neural network (CNN) to detect delamination along the same segments efficiently and effectively was developed and trained. Next, a Crack Index to describe the level of pavement deterioration of a road segment as a function of the number of cracks detected in that segment was computed. Finally, a method to assess the effect of pavement deterioration on traffic congestion by comparing pavement PQI values and corresponding TTI values of roadway segments was evaluated. As such, the objective of the proposed research is to establish links between roadway pavement condition and recurrent traffic congestion. The research has two main goals: 1) Understand the impact of roadway surface conditions on traffic congestion, and 2) Develop a framework for autonomously monitoring the cracks and deteriorations of pavements in each network.

3 Literature Review

3.1 Impact of Pavement Quality

The condition of pavements, which are crucial components of transportation infrastructure, is a major issue in the United States [19]. According to the ASCE report card, U.S. roads have been given a poor grade of D due to significant deficiencies. This poor condition of roads leads to substantial costs for American motorists, who incur an estimated $54 billion per year in repair and operating expenses [19, 20]. Based on the 2012 Urban Mobility Report by the Texas Transportation Institute (TTI), congestion in 85 urban areas of the United States resulted in 5.5 billion hours of travel delay and 2.9 billion gallons of excess fuel consumption in 2011 [21, 22]. Congestion related issues amounted to $121 billion in expenses for the public. It is important to note that this estimate does not even include the considerable costs associated with auto emissions by idle vehicles, soaring gasoline prices as well as road surface descent. If these additional factors were considered, the overall costs would be significantly higher [21, 22]. As pavements continue to wear out and degrade over time, it becomes necessary to carry out maintenance and rehabilitation activities to ensure that they can continue to provide a high level of safety and quality of service [23, 24].

Pavement deterioration can be defined as the process by which distress (i.e., defects) develop in the pavement under the combined effects of traffic loading and environmental conditions [25]. Traffic is considered one of the primary factors contributing to pavement deterioration. The magnitude, configuration, and number of load repetitions caused by vehicular traffic in general and heavy vehicles in particular are major factors that influence the performance of pavements [25-27].

There is an argument that poor pavement conditions exacerbate recurring congestion, as drivers may suddenly adjust their driving behaviors to avoid pavement cracks and other deficiencies in the infrastructure. Nevertheless, the literature lacks comprehensive documentation of such connections. Still, it is reorganized that the deterioration of highways' pavement condition has a negative impact on the convenience, cost, and safety of the drivers. Damages on road surfaces can prompt drivers to slow down or maneuver abruptly to avoid potholes and pavement cracks [28, 29]. While minor driving adjustments may have negligible impacts on traffic operations, others can significantly affect network performance, user satisfaction, congestion, safety, and environmental quality as they propagate upstream of the damaged pavement's location [21, 30]. Previous studies, such as the one conducted by Setwyawn et al., (2015) have demonstrated that road damages can cause a substantial 55% decrease in vehicle speeds when comparing very poor roadway conditions to excellent ones [31]. These effects on vehicle speeds have significant implications for network performance as well as motor vehicle emissions and environmental pollution. Therefore, it is crucial to implement congestion monitoring approaches that integrate both traffic and pavement conditions. Such integrated approaches have the potential to effectively detect and lessen traffic congestion in urban areas.

3.2 Machine Learning (ML)

Structural health monitoring (SHM) in its simplest form involves recurrent visual observation, and assessment of structural conditions (cracking, spelling, and deformations). However, these processes require data collection, data processing, and data diagnosis, which historically were very costly, labor intensive, and often unreliable [32]. Present-day research provides the opportunity to develop effective and reliable means of acquiring, managing, integrating, and interpreting SHM performance data to attain the most useful information at the least possible cost. That is implemented by reducing the unreliable human element involved in long-term data monitoring from a variety of sensors installed on the structures [33]. The effective support and inspection systems that reveal problematic performance issues are limited and intermittent inspection procedures must be supplemented with continuous, online, real-time, and automated systems [33]. Civil infrastructure monitoring comprises numerous structures and processes, and machine learning has broken into most of those fields, as researchers and engineers have continually noticed the prospects and innovative technological strength that a deep learning-based approach affords the community [34-36]. That has inspired numerous attempts to incorporate deep learning structural health monitoring in civil infrastructures [37], structural damage detection, and structural condition assessment. Structural damage implementation is essential for the safety of in-service structures. Researchers and industry groups are using deep learning-based SHM as shown in Figure 1, to analyze damage detection on a variety of structures [33], integrating deep ML into the planning, design, construction, maintenance, and management of civil infrastructures [38, 39].

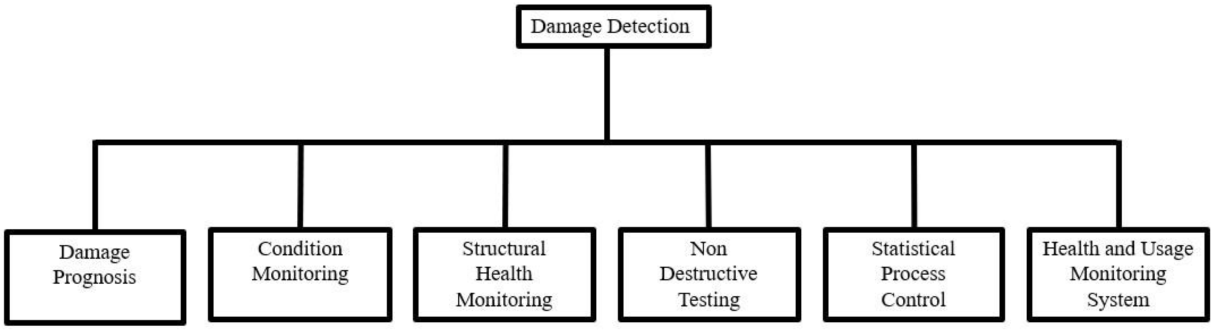


Figure 1. Damage Detection Disciplines[40]

In Bridges for example, a three-level deep learning-based method was proposed by Liang X [36], for inspecting post-disaster bridges, using a visual geometry group (VGG-16) model to detect system-level failure, and using Faster regions with convolutional neural networks (R-CNN) and SegNet to detect component-level and local-level damage, respectively [33], while Kim et al. [41] proposed a unmanned arial vehicle (UAV) and R-CNN-based approach to detect cracks in aged concrete bridges. Khodabandehlou et al. [38] set up an eleven-layer Convolutional Neural Network (CNN) to conduct damage state classification, incorporating acceleration data from shaking table tests of a reinforced concrete bridge model under different loads for validation.

Duan et al. [39] proposed a CNN-based approach to detect bridge damage by acceleration responses. Utilizing a numerical analysis of a tied arch bridge with different damage conditions to generate acceleration responses and applying the acceleration responses and generated Fourier spectra as the dataset. Boa et al. [42] developed an auto-encoder-based network to detect data anomalies. The proposed algorithm used a pre-trained VGG-16 fine-tuned classifier and a pre-trained VGG-16 with a fine-tuned convolution layer and a classifier, using the acceleration data from a cable-stayed bridge for validation. Dung [43] compared three deep learning-based methods based on transfer learning to detect the cracks at the welded joints of gusset plates in bridges.

Structural damage detection in Tunnels and Railways saw an implementation of deep ML in its monitoring. For example, Huang et al. [44] employed an fully convolutional neural network (FCN) based on two-stream approach to semantic segmentation for cracks and leakages in tunnels. Li et al. [45] proposed an image processing and Faster R-CNN-based framework to detect tunnel cracks. Gilbert et al. [46] proposed a CNN-based framework to detect multiple railway damages, with its framework sharing three convolutional layers for material classification, fastener classification, and fastener damage detection. Lin et al. [47] proposed a similarity-based CNN for the inspection of the conditions of railway track fasteners. The similarity of pairs of fastener images was calculated to assess the ability of feature extraction in the pre-training stage. To enlarge the training dataset, a template matching-based classification approach was adopted to select large numbers of fastener images from online railway images. Wei et al. [48] compared the performance of the capacity for the detection of defects in railway fasteners among IPTs, VGG-16, and Faster R-CNN.

Ground movements from soil settling, sinking holes, and earthquakes are the greatest contributors to the wear and tear of civil building infrastructures. That has inspired the implementation of deep learning approaches to building infrastructure. For example, Yeum et al. [49] proposed an AlexNet-based two-stage framework for collapse classification and spalling detection in post-event analysis for concrete buildings. Using a dataset for post-event reconnaissance images built by collecting images after natural disasters. Xu et al. [50] proposed a Faster R-CNN-based model to detect and localize multiple types of seismic damage such as cracks and spelling. Kang and Cha [51] proposed an automatic unmanned aerial vehicle and CNN-based damage detection approach for application in indoor environments, using stationary beacon-based geo-tagging to navigate the UAV and find the damage. Cha et al. [52] proposed a CNN-based method for the detection of structural cracks. Testing images had cracks with different widths, lighting conditions, and noise levels. Yang et al. [53] proposed a VGG 19-based FCN, to detect cracks in different scales. Segmented crack pixels were processed to a single pixel width skeleton for post-evaluation of morphological features including crack topology and length. Wang et al. [54] applied AlexNet and GoogLeNet to detect multiple damages to historic masonry walls, and the sliding window techniques were used to find the damages. Ni et al. [55] proposed a GoogLeNet and ResNet-based method for the detection of cracks, using the Zernike moment operator to process crack images detected for thin crack quantification. Zhang et al.

[56] proposed a SegNet-based context-aware fusion algorithm to detect cracks in images of arbitrary sizes.

Utilizing CrackForest, management dataset, Tomorrows Road Infrastructure Monitoring dataset, and Customized Field Test dataset for the validation [57]. Li et al. [58] proposed a Faster R-CNN-based framework to detect and localize multiple defects in different scenarios. To strengthen the algorithm's ability, multi-scale training, data augmentation, a location block, and negative mining strategies were adopted for the localization of defects. Kim and Cho [59] proposed a Masked R-CNN-based framework for the detection and quantification of concrete cracks. Ye et al. [57] developed an FCN to automatically detect cracks on concrete surfaces, using an online dataset of crack images with pixel-wise labels for training and validation. For steel structures, Gulgec et al. [60] proposed a CNN-based approach to classify damaged and undamaged steel structure components generated by numerical simulations. Zhao et al. [61] proposed a VGG-16-based method to detect the condition of bolt loosening for steel structures. Zhou et al. [62] trained an autoencoder-based network based on stiffness to implement damage identification based on stiffness deterioration.

Structural maintenance is dependent on the structures showing characteristics of long-term wear from their repetitive structural service. Structural maintenance is subdivided into transportation and buildings. Machine learning-induced structural maintenance from building infrastructures has improved in recent years. Oh et al. [63] proposed a CNN-based architecture to predict strain levels of tall buildings under wind loadings, using a dataset with displacements and wind speeds, derived from a wind tunnel test of a model steel structure. Rafiei and Adeli [64] presented an unsupervised learning-based framework for the assessment of local and global conditions of structural systems via collected vibration response data. Lyu et al. [65] proposed a deep belief network-based approach to assess the state/health of offshore platforms, using a model platform fabricated and evaluated to collect the wave force, strain, and acceleration data.

Structural maintenance of transportation infrastructures has seen great advancement as well. For instance, Liang et al. [66] set up a multi-scale SHM system to assess the serviceability of the bridge based on a Hadoop Ecosystem. Li et al. [67] proposed a modified VGG-16 and IPT-based framework to detect multiple parameters of ships coming toward bridges to prevent collision incidents. The modified pre-trained and fine-tuned VGG-16 network coarsely detects and localizes the incoming ships and IPT calculates the ship parameters including width, length, and velocity. Zhang et al. [68] proposed a one-dimensional CNN-based approach to assess the structural state by acceleration signals, using results from an indoor test of a bridge model, an outdoor test of a full-scale bridge model, and a test of an in-service bridge as a dataset.

Machine learning-induced structural maintenance for transportation infrastructures has improved especially with the materials varied between concrete and asphalt. Gopalakrishnan et al. [69] developed a pre-trained VGG-16-based algorithm to detect pavement cracks, using a pre-trained ImageNet VGG-16 architecture. A mixture of asphalt and concrete pavement images was introduced to increase the robustness. Zhang et al. [70] proposed a CNN model called CrackNet to automatically detect pavement cracks on 3D images of asphalt road surfaces for pixel-wise detection. Tong et al. [71] proposed a two-stage CNN-based approach to automatically measure pavement crack lengths. The proposed CNN was pre-trained by images with crack labels and fine-tuned by images with detailed labels of the length of cracks. Zhang et al. [72] proposed CrackNet II, a modified model of CrackNet, for crack identification with greater precision and better recall rates. In comparison, CrackNet II used a deeper architecture with lesser parameters which led to better computing efficiency. Nhat-Duc et al. [73] compared two edge detection methods and a CNN- based approach for the recognition of pavement cracks. Maeda et al. [74] applied MobileNet and Inception to detect multiple road damages. Zhang et al. [75] proposed an RNN-based model called CrackNet-R to detect pavement cracks in 3D images at the pixel level. Alipour et al. [76], proposed CrackPix a deep fully convolutional neural network (FCN) for pixel-level defect detection in concrete infrastructure systems. CrackPix leverages a modified VGG-16 with pertained weights for image classification architectures for dense pixel-wise predictions. Li and Zhao [77], proposed a densely connected and deeply supervised network to improve the accuracy and robustness of existing automated crack detection methods. Using deeply supervised modules to extract more noteworthy features through multi-scale levels. Zhang et al. [78], used pre-classification based on transfer learning to detect and separate both cracks and sealed cracks under the same framework.

4 Methodology

In this study, we used Travel Time (TT) data (2018-2022) from the National Performance Management Research Data Set (NPMRDS) for congestion monitoring of two roadway corridors in the Birmingham area. We also employed machine learning algorithms to assess pavement quality from Google Street View images. The data were used to determine a Travel Time Index (TTI) and a pavement Crack Index and examine their potential correlation.

4.1 Traffic Data

4.1.1 Data Source

Travel Time data were compiled from the National Performance Management Research Data Set (NPMRDS) website for portions of I-65 and US-31 corridors in the Birmingham area. The number of Traffic Message Channels (TMCs) defined by NPMRDS were identified based on the selected study road network/segment. At first, the road corridors shapefile of Alabama was downloaded from the npmrds.ritis.org website, where detailed traffic data such as travel time, length of corridors, speed data were available. From the shapefile, the total number of TMCs of the study road network was found as 62. Among these TMCs, 38 TMCs were for the North and Southbound Road corridors of I-65 lying between Birmingham and Hoover, and 24 TMCs were for the South and Northbound corridors of Montgomery Highway (US-31) lying between Hoover to Vestavia. The detailed information of TMCs displayed in Table 1 and Table 2 are for the I-65 and US-31 in the southbound and northbound direction, respectively. The steps of downloading data from the

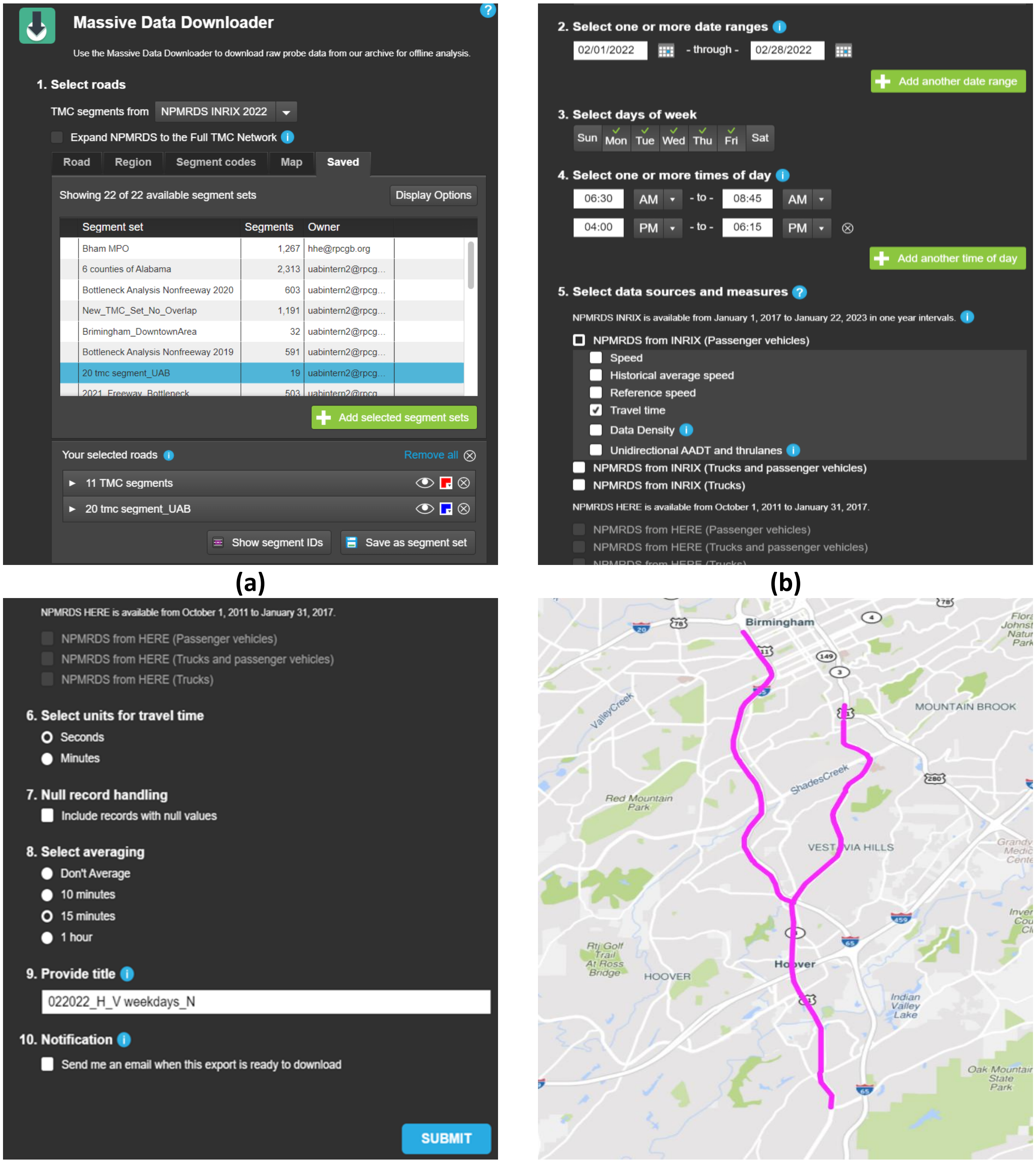
NPMRDS website are illustrated in Figure 2. Furthermore, the steps to process the downloaded Travel Time data into Travel Time Index are shown in Figures 3-6.

Table 1. TMC for the Southbound of I-65 and US-31

TMC	Road	Direction	Length (Miles)	Intersection
101-11461	US-31	SB	1.051536	PATTON CHAPEL RD
101-04381	I-65	SB	0.207012	6TH AVE/EXIT 260
101-11460	US-31	SB	0.576145	I-459
101-04380	I-65	SB	0.232945	3RD AVE/EXIT 260B
101N04375	I-65	SB	0.473173	OXMOOR RD/EXIT 256
101N04374	I-65	SB	0.297113	LAKESHORE DR/EXIT 255
101N04377	I-65	SB	0.308821	AL-149/UNIVERSITY BLVD/EXIT 259
101N04376	I-65	SB	0.361391	GREEN SPRINGS AVE/EXIT 258
101N04379	I-65	SB	0.114376	4TH AVE/EXIT 259B
101N04378	I-65	SB	0.083757	6TH AVE/EXIT 259A
101N04380	I-65	SB	0.145201	3RD AVE/EXIT 260B
101N11460	US-31	SB	0.370157	I-459
101N04381	I-65	SB	0.150305	6TH AVE/EXIT 260
101-04374	I-65	SB	0.496748	LAKESHORE DR/EXIT 255
101-04376	I-65	SB	0.873027	GREEN SPRINGS AVE/EXIT 258
101-04375	I-65	SB	1.022207	OXMOOR RD/EXIT 256
101-04378	I-65	SB	0.083652	6TH AVE/EXIT 259A
101-04377	I-65	SB	0.029036	AL-149/UNIVERSITY BLVD/EXIT 259
101-04379	I-65	SB	0.315	4TH AVE/EXIT 259B

Table 2. TMC FOR THE Northbound OF I-65 AND US-31

TMC	Road	Direction	Length (Miles)	Intersection
101P11461	US-31	NB	0.034315	PATTON CHAPEL RD
101P11462	US-31	NB	0.215653	I-65 (BIRMINGHAM)
101P04381	I-65	NB	0.166591	6TH AVE/EXIT 260
101P04380	I-65	NB	0.145201	3RD AVE/EXIT 260B
101P04375	I-65	NB	0.430823	OXMOOR RD/EXIT 256
101+04380	I-65	NB	0.315891	3RD AVE/EXIT 260B
101+04381	I-65	NB	0.23234	6TH AVE/EXIT 260
101P04379	I-65	NB	0.115262	4TH AVE/EXIT 259B
101+04382	I-65	NB	0.277815	I-59/I-20/EXIT 261
101P04378	I-65	NB	0.082842	6TH AVE/EXIT 259A
101P04377	I-65	NB	0.241238	AL-149/UNIVERSITY BLVD/EXIT 259
101P04376	I-65	NB	0.359093	GREEN SPRINGS AVE/EXIT 258
101+04375	I-65	NB	0.674452	OXMOOR RD/EXIT 256
101+04376	I-65	NB	1.085533	GREEN SPRINGS AVE/EXIT 258
101+04377	I-65	NB	0.859892	AL-149/UNIVERSITY BLVD/EXIT 259
101+04378	I-65	NB	0.034545	6TH AVE/EXIT 259A
101+04379	I-65	NB	0.083031	4TH AVE/EXIT 259B
101+11461	US-31	NB	0.598163	PATTON CHAPEL RD
101+11462	US-31	NB	1.067083	I-65 (BIRMINGHAM)



Massive Data Downloader

Use the Massive Data Downloader to download raw probe data from our archive for offline analysis.

1. Select roads

TMC segments from NPMRDS INRIX 2022

☐ Expand NPMRDS to the Full TMC Network

Showing 22 of 22 available segment sets

Segment set	Segments	Owner
Bham MPO	1,267	hhe@rpcgb.org
6 counties of Alabama	2,313	uabintern2@rpcg...
Bottleneck Analysis Nonfreeway 2020	603	uabintern2@rpcg...
New_TMC_Set_No_Overlap	1,191	uabintern2@rpcg...
Brimingham_DowntownArea	32	uabintern2@rpcg...
Bottleneck Analysis Nonfreeway 2019	591	uabintern2@rpcg...
20 tmc segment_UAB	19	uabintern2@rpcg...
2021_Freeway_Bottleneck	503	uabintern2@rpcg

Your selected roads

- 11 TMC segments
- 20 tmc segment_UAB

(a)

2. Select one or more date ranges

02/01/2022 - through - 02/28/2022

3. Select days of week

Sun ✓Mon ✓Tue ✓Wed ✓Thu ✓Fri Sat

4. Select one or more times of day

06:30 AM - to - 08:45 AM

04:00 PM - to - 06:15 PM

5. Select data sources and measures

NPMRDS INRIX is available from January 1, 2017 to January 22, 2023 in one year intervals.

- ☐ NPMRDS from INRIX (Passenger vehicles)
  - ☐ Speed
  - ☐ Historical average speed
  - ☐ Reference speed
  - ☑ Travel time
  - ☐ Data Density
  - ☐ Unidirectional AADT and thrulanes
- ☐ NPMRDS from INRIX (Trucks and passenger vehicles)
- ☐ NPMRDS from INRIX (Trucks)

NPMRDS HERE is available from October 1, 2011 to January 31, 2017.

- ☐ NPMRDS from HERE (Passenger vehicles)
- ☐ NPMRDS from HERE (Trucks and passenger vehicles)
- ☐ NPMRDS from HERE (Trucks)

(b)

NPMRDS HERE is available from October 1, 2011 to January 31, 2017.

- ☐ NPMRDS from HERE (Passenger vehicles)
- ☐ NPMRDS from HERE (Trucks and passenger vehicles)
- ☐ NPMRDS from HERE (Trucks)

6. Select units for travel time

- ● Seconds
- ○ Minutes

7. Null record handling

- ☐ Include records with null values

8. Select averaging

- ○ Don't Average
- ○ 10 minutes
- ● 15 minutes
- ○ 1 hour

9. Provide title

022022_H_V weekdays_N

10. Notification

- ☐ Send me an email when this export is ready to download

(c)

(d)

Figure 2. Stepwise procedure of selection of travel time data from NPMRDS: (a) Selection of road segments, (b) Selection of time ranges, weekdays and data type, (c) Selection of travel time unit and averaging option, and (d) Visualization of study area in the massive data downloader

The Travel Time data for February 2018, 2019, 2020, 2021 and 2022 were downloaded from the NPMRDS website. Fifteen minutes averaging travel time data for passenger vehicles from 6:30 AM to 8:45 AM (10 data points) of each weekday of February and fifteen minutes averaging travel time data from 4:00 PM to 6:15 PM (10 data points) for the passenger vehicles of each weekday of February were used to determine the morning peak and the evening peak travel times, respectively, for five years.

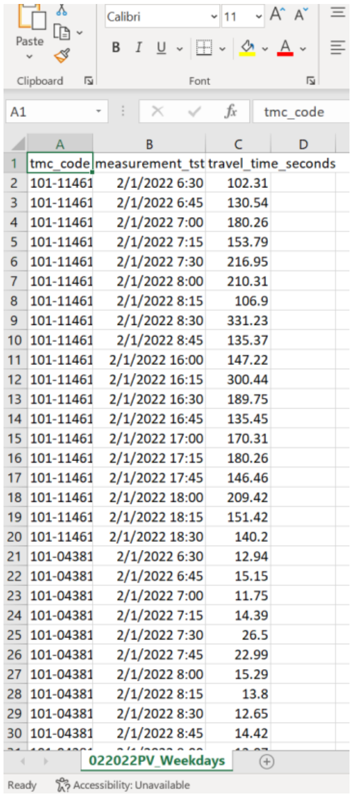
4.1.2 Preparation of the Travel Time Data into Travel Time Index (TTI) for each TMC

The downloaded Travel Time data from the NPMRDS website are shown in Figure 3. First, the travel time data were arranged to determine the 50th percentile travel data for passenger vehicles of each TMC. These data were further used to calculate the 50th percentile TTI of passenger vehicles of each TMC. A stepwise procedure was required to convert the TMC-wise TT into TTI.

- Initially, using Excel pivot tables, TT data were organized for each time slot by 15 minutes interval starting from 6:30 AM to 8:45 AM (10 slots in the morning) and from 4:00 PM to 6:15 PM (10 slots in the evening) of each TMC for each weekday of February as shown in Figure 4.
- After getting the 50th percentile monthly Travel Time for each TMC time slots (20 slots per TMC), the Excel table was modified as per the format displayed in Figure 5.
- The Travel Time Index (TTI) of a road segment is defined as the ratio of the actual average travel time to the time required to travel the same distance at free-flow speeds (FFS), as shown in Equation 1;

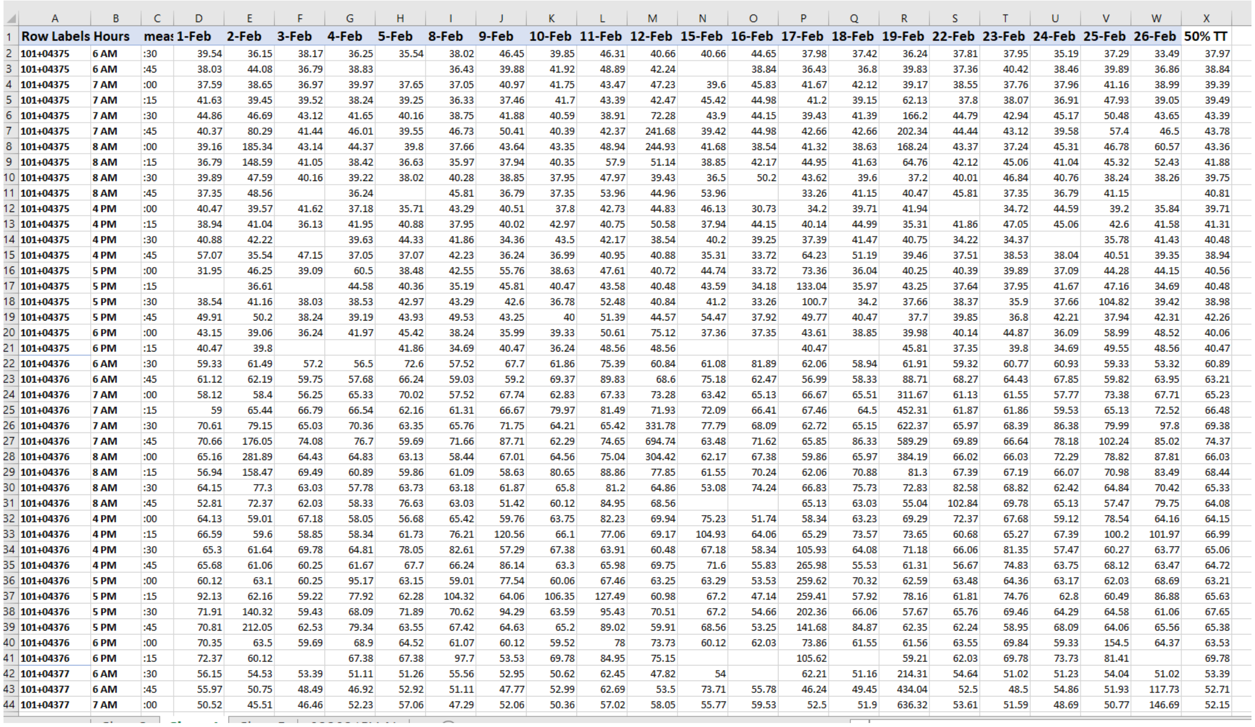
$$Travel\ Time\ Index\ (TTI) = \frac{Average\ Travel\ Time}{Travel\ Time\ Based\ on\ Free\ Flow\ Speed} \tag{1}$$

- The Travel Time Index (TTI) is a measure of congestion as well as reliability of roadway segments. A TTI value greater than 1.0 is indicative of congestion presence.
- Thus, to calculate the average travel time of the 62 TMCs, the road length and FFS data were obtained using ‘Alabama shapefiles’ from NPMRDS in ArcGIS and added to the TT Excel files in ArcGIS. A sample is shown in Figure 5.
- At the time of accumulating monthly 50th percentile TMC-wise travel times for each of the 20-time slots with Road Corridor Name, Length of TMCs, Direction, Intersection point and FFS, some blank / missing data were found in the database (Figure 4). To address this issue, the value of zero was used in lieu of blank data.



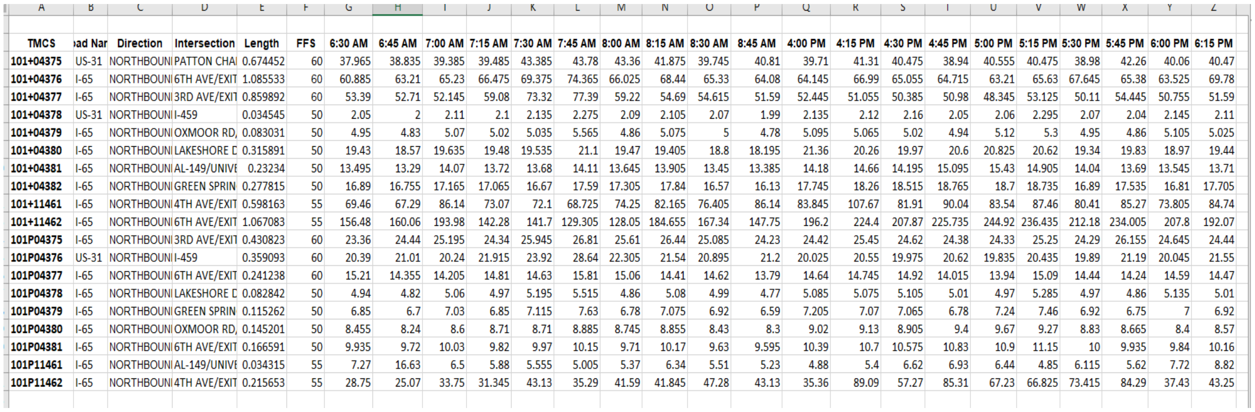
	A	B	C	D
1	tmc_code	measurement_tst	travel_time_seconds	
2	101-11461	2/1/2022 6:30	102.31	
3	101-11461	2/1/2022 6:45	130.54	
4	101-11461	2/1/2022 7:00	180.26	
5	101-11461	2/1/2022 7:15	153.79	
6	101-11461	2/1/2022 7:30	216.95	
7	101-11461	2/1/2022 8:00	210.31	
8	101-11461	2/1/2022 8:15	106.9	
9	101-11461	2/1/2022 8:30	331.23	
10	101-11461	2/1/2022 8:45	135.37	
11	101-11461	2/1/2022 16:00	147.22	
12	101-11461	2/1/2022 16:15	300.44	
13	101-11461	2/1/2022 16:30	189.75	
14	101-11461	2/1/2022 16:45	135.45	
15	101-11461	2/1/2022 17:00	170.31	
16	101-11461	2/1/2022 17:15	180.26	
17	101-11461	2/1/2022 17:45	146.46	
18	101-11461	2/1/2022 18:00	209.42	
19	101-11461	2/1/2022 18:15	151.42	
20	101-11461	2/1/2022 18:30	140.2	
21	101-04381	2/1/2022 6:30	12.94	
22	101-04381	2/1/2022 6:45	15.15	
23	101-04381	2/1/2022 7:00	11.75	
24	101-04381	2/1/2022 7:15	14.39	
25	101-04381	2/1/2022 7:30	26.5	
26	101-04381	2/1/2022 7:45	22.99	
27	101-04381	2/1/2022 8:00	15.29	
28	101-04381	2/1/2022 8:15	13.8	
29	101-04381	2/1/2022 8:30	12.65	
30	101-04381	2/1/2022 8:45	14.42	

Figure 3. NPMRDS Travel Time Data in Seconds



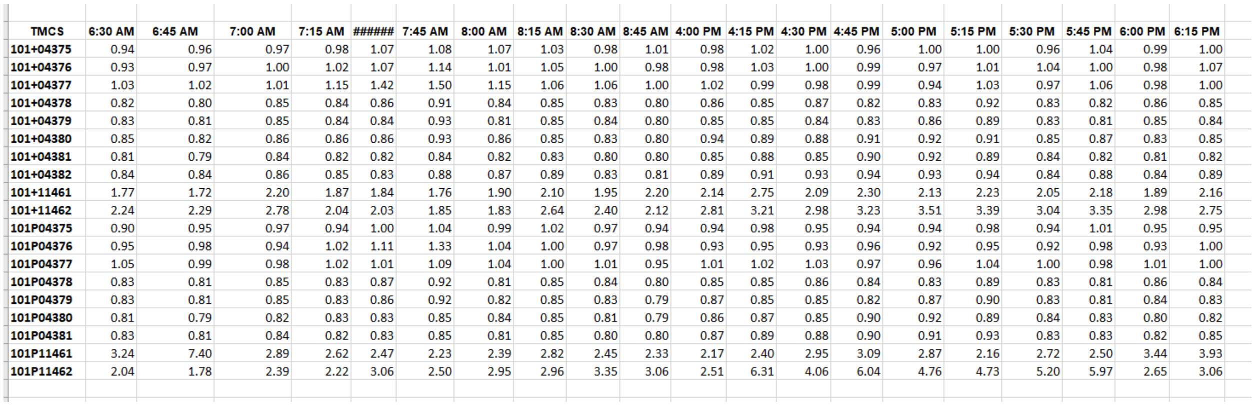
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
1	Row Labels	Hours	mea:	1-Feb	2-Feb	3-Feb	4-Feb	5-Feb	8-Feb	9-Feb	10-Feb	11-Feb	12-Feb	15-Feb	16-Feb	17-Feb	18-Feb	19-Feb	22-Feb	23-Feb	24-Feb	25-Feb	26-Feb	50% TT
2	101+04375	6 AM	:30	39.54	36.15	38.17	36.25	35.54	38.02	46.45	39.85	46.31	40.66	40.66	44.65	37.98	37.42	36.24	37.81	37.95	35.19	37.29	33.49	37.97
3	101+04375	6 AM	:45	38.03	44.08	36.79	38.83		36.43	39.88	41.92	48.89	42.24		38.84	36.43	36.8	39.83	37.36	40.42	38.46	39.89	36.86	38.84
4	101+04375	7 AM	:00	37.59	38.65	36.97	39.97	37.65	37.05	40.97	41.75	43.47	47.23	39.6	45.83	41.67	42.12	39.17	38.55	37.76	37.96	41.16	38.99	39.39
5	101+04375	7 AM	:15	41.63	39.45	39.52	38.24	39.25	36.33	37.46	41.7	43.39	42.47	45.42	44.98	41.2	39.15	62.13	37.8	38.07	36.91	47.93	39.05	39.49
6	101+04375	7 AM	:30	44.86	46.69	43.12	41.65	40.16	38.75	41.88	40.59	38.91	72.28	43.9	44.15	39.43	41.39	166.2	44.79	42.94	45.17	50.48	43.65	43.39
7	101+04375	7 AM	:45	40.37	80.29	41.44	46.01	39.55	46.73	50.41	40.39	42.37	241.68	39.42	44.98	42.66	42.66	202.34	44.44	43.12	39.58	57.4	46.5	43.78
8	101+04375	8 AM	:00	39.16	185.34	43.14	44.37	39.8	37.66	43.64	43.35	48.94	244.93	41.68	38.54	41.32	38.63	168.24	43.37	37.24	45.31	46.78	60.57	43.36
9	101+04375	8 AM	:15	36.79	148.59	41.05	38.42	36.63	35.97	37.94	40.35	57.9	51.14	38.85	42.17	44.95	41.63	64.76	42.12	45.06	41.04	45.32	52.43	41.88
10	101+04375	8 AM	:30	39.89	47.59	40.16	39.22	38.02	40.28	38.85	37.95	47.97	39.43	36.5	50.2	43.62	39.6	37.2	40.01	46.84	40.76	38.24	38.26	39.75
11	101+04375	8 AM	:45	37.35	48.56		36.24		45.81	36.79	37.35	53.96	44.96	53.96		33.26	41.15	40.47	45.81	37.35	36.79	41.15		40.81
12	101+04375	4 PM	:00	40.47	39.57	41.62	37.18	35.71	43.29	40.51	37.8	42.73	44.83	46.13	30.73	34.2	39.71	41.94		34.72	44.59	39.2	35.84	39.71
13	101+04375	4 PM	:15	38.94	41.04	36.13	41.95	40.88	37.95	40.02	42.97	40.75	50.58	37.94	44.15	40.14	44.99	35.31	41.86	47.05	45.06	42.6	41.58	41.31
14	101+04375	4 PM	:30	40.88	42.22		39.63	44.33	41.86	34.36	43.5	42.17	38.54	40.2	39.25	37.39	41.47	40.75	34.22	34.37		35.78	41.43	40.48
15	101+04375	4 PM	:45	57.07	35.54	47.15	37.05	37.07	42.23	36.24	36.99	40.95	40.88	35.31	33.72	64.23	51.19	39.46	37.51	38.53	38.04	40.51	39.35	38.94
16	101+04375	5 PM	:00	31.95	46.25	39.09	60.5	38.48	42.55	55.76	38.63	47.61	40.72	44.74	33.72	73.36	36.04	40.25	40.39	39.89	37.09	44.28	44.15	40.56
17	101+04375	5 PM	:15		36.61		44.58	40.36	35.19	45.81	40.47	43.58	40.48	43.59	34.18	133.04	35.97	43.25	37.64	37.95	41.67	47.16	34.69	40.48
18	101+04375	5 PM	:30	38.54	41.16	38.03	38.53	42.97	43.29	42.6	36.78	52.48	40.84	41.2	33.26	100.7	34.2	37.66	38.37	35.9	37.66	104.82	39.42	38.98
19	101+04375	5 PM	:45	49.91	50.2	38.24	39.19	43.93	49.53	43.25	40	51.39	44.57	54.47	37.92	49.77	40.47	37.7	39.85	36.8	42.21	37.94	42.31	42.26
20	101+04375	6 PM	:00	43.15	39.06	36.24	41.97	45.42	38.24	35.99	39.33	50.61	75.12	37.36	37.35	43.61	38.85	39.98	40.14	44.87	36.09	58.99	48.52	40.06
21	101+04375	6 PM	:15	40.47	39.8			41.86	34.69	40.47	36.24	48.56	48.56			40.47		45.81	37.35	39.8	34.69	49.55	48.56	40.47
22	101+04376	6 AM	:30	59.33	61.49	57.2	56.5	72.6	57.52	67.7	61.86	75.39	60.84	61.08	81.89	62.06	58.94	61.91	59.32	60.77	60.93	59.33	53.32	60.89
23	101+04376	6 AM	:45	61.12	62.19	59.75	57.68	66.24	59.03	59.2	69.37	89.83	68.6	75.18	62.47	56.99	58.33	88.71	68.27	64.43	67.85	59.82	63.95	63.21
24	101+04376	7 AM	:00	58.12	58.4	56.25	65.33	70.02	57.52	67.74	62.83	67.33	73.28	63.42	65.13	66.67	65.51	311.67	61.13	61.55	57.77	73.38	67.71	65.23
25	101+04376	7 AM	:15	59	65.44	66.79	66.54	62.16	61.31	66.67	79.97	81.49	71.93	72.09	66.41	67.46	64.5	452.31	61.87	61.86	59.53	65.13	72.52	66.48
26	101+04376	7 AM	:30	70.61	79.15	65.03	70.36	63.35	65.76	71.75	64.21	65.42	331.78	77.79	68.09	62.72	65.15	622.37	65.97	68.39	86.38	79.99	97.8	69.38
27	101+04376	7 AM	:45	70.66	176.05	74.08	76.7	59.69	71.66	87.71	62.29	74.65	694.74	63.48	71.62	65.85	86.33	589.29	69.89	66.64	78.18	102.24	85.02	74.37
28	101+04376	8 AM	:00	65.16	281.89	64.43	64.83	63.13	58.44	67.01	64.56	75.04	304.42	62.17	67.38	59.86	65.97	384.19	66.02	66.03	72.29	78.82	87.81	66.03
29	101+04376	8 AM	:15	56.94	158.47	69.49	60.89	59.86	61.09	58.63	80.65	88.86	77.85	61.55	70.24	62.06	70.88	81.3	67.39	67.19	66.07	70.98	83.49	68.44
30	101+04376	8 AM	:30	64.15	77.3	63.03	57.78	63.73	63.18	61.87	65.8	81.2	64.86	53.08	74.24	66.83	75.73	72.83	82.58	68.82	62.42	64.84	70.42	65.33
31	101+04376	8 AM	:45	52.81	72.37	62.03	58.33	76.63	63.03	51.42	60.12	84.95	68.56			65.13	63.03	55.04	102.84	69.78	65.13	57.47	79.75	64.08
32	101+04376	4 PM	:00	64.13	59.01	67.18	58.05	56.68	65.42	59.76	63.75	82.23	69.94	75.23	51.74	58.34	63.23	69.29	72.37	67.68	59.12	78.54	64.16	64.15
33	101+04376	4 PM	:15	66.59	59.6	58.85	58.34	61.73	76.21	120.56	66.1	77.06	69.17	104.93	64.06	65.29	73.57	73.65	60.68	65.27	67.39	100.2	101.97	66.99
34	101+04376	4 PM	:30	65.3	61.64	69.78	64.81	78.05	82.61	57.29	67.38	63.91	60.48	67.18	58.34	105.93	64.08	71.18	66.06	81.35	57.47	60.27	63.77	65.06
35	101+04376	4 PM	:45	65.68	61.06	60.25	61.67	67.7	66.24	86.14	63.3	65.98	69.75	71.6	55.83	265.98	55.53	61.31	56.67	74.83	63.75	68.12	63.47	64.72
36	101+04376	5 PM	:00	60.12	63.1	60.25	95.17	63.15	59.01	77.54	60.06	67.46	63.25	63.29	53.53	259.62	70.32	62.59	63.48	64.36	63.17	62.03	68.69	63.21
37	101+04376	5 PM	:15	92.13	62.16	59.22	77.92	62.28	104.32	64.06	106.35	127.49	60.98	67.2	47.14	259.41	57.92	78.16	61.81	74.76	62.8	60.49	86.88	65.63
38	101+04376	5 PM	:30	71.91	140.32	59.43	68.09	71.89	70.62	94.29	63.59	95.43	70.51	67.2	54.66	202.36	66.06	57.67	65.76	69.46	64.29	64.58	61.06	67.65
39	101+04376	5 PM	:45	70.81	212.05	62.53	79.34	63.55	67.42	64.63	65.2	89.02	59.91	68.56	53.25	141.68	84.87	62.35	62.24	58.95	68.09	64.06	65.56	65.38
40	101+04376	6 PM	:00	70.35	63.5	59.69	68.9	64.52	61.07	60.12	59.52	78	73.73	60.12	62.03	73.86	61.55	61.56	63.55	69.84	59.33	154.5	64.37	63.53
41	101+04376	6 PM	:15	72.37	60.12		67.38	67.38	97.7	53.53	69.78	84.95	75.15			105.62		59.21	62.03	69.78	73.73	81.41		69.78
42	101+04377	6 AM	:30	56.15	54.53	53.39	51.11	51.26	55.56	52.95	50.62	62.45	47.82	54		62.21	51.16	214.31	54.64	51.02	51.23	54.04	51.02	53.39
43	101+04377	6 AM	:45	55.97	50.75	48.49	46.92	52.92	51.11	47.77	52.99	62.69	53.5	73.71	55.78	46.24	49.45	434.04	52.5	48.5	54.86	51.93	117.73	52.71
44	101+04377	7 AM	:00	50.52	45.51	46.46	52.23	57.06	47.29	52.06	50.36	57.02	58.05	55.77	59.53	52.5	51.9	636.32	53.61	51.59	48.69	50.77	146.69	52.15

Figure 4. Sample of Data Calculation of 50% Travel Time



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
TMCS	›ad Nar	Direction	Intersection	Length	FFS	6:30 AM	6:45 AM	7:00 AM	7:15 AM	7:30 AM	7:45 AM	8:00 AM	8:15 AM	8:30 AM	8:45 AM	4:00 PM	4:15 PM	4:30 PM	4:45 PM	5:00 PM	5:15 PM	5:30 PM	5:45 PM	6:00 PM	6:15 PM
101+04375	US-31	NORTHBOUNI	PATTON CHAI	0.674452	60	37.965	38.835	39.385	39.485	43.385	43.78	43.36	41.875	39.745	40.81	39.71	41.31	40.475	38.94	40.555	40.475	38.98	42.26	40.06	40.47
101+04376	I-65	NORTHBOUNI	6TH AVE/EXIT	1.085533	60	60.885	63.21	65.23	66.475	69.375	74.365	66.025	68.44	65.33	64.08	64.145	66.99	65.055	64.715	63.21	65.63	67.645	65.38	63.525	69.78
101+04377	I-65	NORTHBOUNI	3RD AVE/EXIT	0.859892	60	53.39	52.71	52.145	59.08	73.32	77.39	59.22	54.69	54.615	51.59	52.445	51.055	50.385	50.98	48.345	53.125	50.11	54.445	50.755	51.59
101+04378	US-31	NORTHBOUNI	I-459	0.034545	50	2.05	2	2.11	2.1	2.135	2.275	2.09	2.105	2.07	1.99	2.135	2.12	2.16	2.05	2.06	2.295	2.07	2.04	2.145	2.11
101+04379	I-65	NORTHBOUNI	OXMOOR RD,	0.083031	50	4.95	4.83	5.07	5.02	5.035	5.565	4.86	5.075	5	4.78	5.095	5.065	5.02	4.94	5.12	5.3	4.95	4.86	5.105	5.025
101+04380	I-65	NORTHBOUNI	LAKESHORE D	0.315891	50	19.43	18.57	19.635	19.48	19.535	21.1	19.47	19.405	18.8	18.195	21.36	20.26	19.97	20.6	20.825	20.62	19.34	19.83	18.97	19.44
101+04381	I-65	NORTHBOUNI	AL-149/UNIVE	0.23234	50	13.495	13.29	14.07	13.72	13.68	14.11	13.645	13.905	13.45	13.385	14.18	14.66	14.195	15.095	15.43	14.905	14.04	13.69	13.545	13.71
101+04382	I-65	NORTHBOUNI	GREEN SPRIN	0.277815	50	16.89	16.755	17.165	17.065	16.67	17.59	17.305	17.84	16.57	16.13	17.745	18.26	18.515	18.765	18.7	18.735	16.89	17.535	16.81	17.705
101+11461	I-65	NORTHBOUNI	4TH AVE/EXIT	0.598163	55	69.46	67.29	86.14	73.07	72.1	68.725	74.25	82.165	76.405	86.14	83.845	107.67	81.91	90.04	83.54	87.46	80.41	85.27	73.805	84.74
101+11462	I-65	NORTHBOUNI	6TH AVE/EXIT	1.067083	55	156.48	160.06	193.98	142.28	141.7	129.305	128.05	184.655	167.34	147.75	196.2	224.4	207.87	225.735	244.92	236.435	212.18	234.005	207.8	192.07
101P04375	I-65	NORTHBOUNI	3RD AVE/EXIT	0.430823	60	23.36	24.44	25.195	24.34	25.945	26.81	25.61	26.44	25.085	24.23	24.42	25.45	24.62	24.38	24.33	25.25	24.29	26.155	24.645	24.44
101P04376	US-31	NORTHBOUNI	I-459	0.359093	60	20.39	21.01	20.24	21.915	23.92	28.64	22.305	21.54	20.895	21.2	20.025	20.55	19.975	20.62	19.835	20.435	19.89	21.19	20.045	21.55
101P04377	I-65	NORTHBOUNI	6TH AVE/EXIT	0.241238	60	15.21	14.355	14.205	14.81	14.63	15.81	15.06	14.41	14.62	13.79	14.64	14.745	14.92	14.015	13.94	15.09	14.44	14.24	14.59	14.47
101P04378	I-65	NORTHBOUNI	LAKESHORE D	0.082842	50	4.94	4.82	5.06	4.97	5.195	5.515	4.86	5.08	4.99	4.77	5.085	5.075	5.105	5.01	4.97	5.285	4.97	4.86	5.135	5.01
101P04379	I-65	NORTHBOUNI	GREEN SPRIN	0.115262	50	6.85	6.7	7.03	6.85	7.115	7.63	6.78	7.075	6.92	6.59	7.205	7.07	7.065	6.78	7.24	7.46	6.92	6.75	7	6.92
101P04380	I-65	NORTHBOUNI	OXMOOR RD,	0.145201	50	8.455	8.24	8.6	8.71	8.71	8.885	8.745	8.855	8.43	8.3	9.02	9.13	8.905	9.4	9.67	9.27	8.83	8.665	8.4	8.57
101P04381	I-65	NORTHBOUNI	6TH AVE/EXIT	0.166591	50	9.935	9.72	10.03	9.82	9.97	10.15	9.71	10.17	9.63	9.595	10.39	10.7	10.575	10.83	10.9	11.15	10	9.935	9.84	10.16
101P11461	I-65	NORTHBOUNI	AL-149/UNIVE	0.034315	55	7.27	16.63	6.5	5.88	5.555	5.005	5.37	6.34	5.51	5.23	4.88	5.4	6.62	6.93	6.44	4.85	6.115	5.62	7.72	8.82
101P11462	I-65	NORTHBOUNI	4TH AVE/EXIT	0.215653	55	28.75	25.07	33.75	31.345	43.13	35.29	41.59	41.845	47.28	43.13	35.36	89.09	57.27	85.31	67.23	66.825	73.415	84.29	37.43	43.25

Figure 5. Calculation of Length and Free Flow Speed (FFS) for 20 Timeslots



TMCS	6:30 AM	6:45 AM	7:00 AM	7:15 AM	######	7:45 AM	8:00 AM	8:15 AM	8:30 AM	8:45 AM	4:00 PM	4:15 PM	4:30 PM	4:45 PM	5:00 PM	5:15 PM	5:30 PM	5:45 PM	6:00 PM	6:15 PM
101+04375	0.94	0.96	0.97	0.98	1.07	1.08	1.07	1.03	0.98	1.01	0.98	1.02	1.00	0.96	1.00	1.00	0.96	1.04	0.99	1.00
101+04376	0.93	0.97	1.00	1.02	1.07	1.14	1.01	1.05	1.00	0.98	0.98	1.03	1.00	0.99	0.97	1.01	1.04	1.00	0.98	1.07
101+04377	1.03	1.02	1.01	1.15	1.42	1.50	1.15	1.06	1.06	1.00	1.02	0.99	0.98	0.99	0.94	1.03	0.97	1.06	0.98	1.00
101+04378	0.82	0.80	0.85	0.84	0.86	0.91	0.84	0.85	0.83	0.80	0.86	0.85	0.87	0.82	0.83	0.92	0.83	0.82	0.86	0.85
101+04379	0.83	0.81	0.85	0.84	0.84	0.93	0.81	0.85	0.84	0.80	0.85	0.85	0.84	0.83	0.86	0.89	0.83	0.81	0.85	0.84
101+04380	0.85	0.82	0.86	0.86	0.86	0.93	0.86	0.85	0.83	0.80	0.94	0.89	0.88	0.91	0.92	0.91	0.85	0.87	0.83	0.85
101+04381	0.81	0.79	0.84	0.82	0.82	0.84	0.82	0.83	0.80	0.80	0.85	0.88	0.85	0.90	0.92	0.89	0.84	0.82	0.81	0.82
101+04382	0.84	0.84	0.86	0.85	0.83	0.88	0.87	0.89	0.83	0.81	0.89	0.91	0.93	0.94	0.93	0.94	0.84	0.88	0.84	0.89
101+11461	1.77	1.72	2.20	1.87	1.84	1.76	1.90	2.10	1.95	2.20	2.14	2.75	2.09	2.30	2.13	2.23	2.05	2.18	1.89	2.16
101+11462	2.24	2.29	2.78	2.04	2.03	1.85	1.83	2.64	2.40	2.12	2.81	3.21	2.98	3.23	3.51	3.39	3.04	3.35	2.98	2.75
101P04375	0.90	0.95	0.97	0.94	1.00	1.04	0.99	1.02	0.97	0.94	0.94	0.98	0.95	0.94	0.94	0.98	0.94	1.01	0.95	0.95
101P04376	0.95	0.98	0.94	1.02	1.11	1.33	1.04	1.00	0.97	0.98	0.93	0.95	0.93	0.96	0.92	0.95	0.92	0.98	0.93	1.00
101P04377	1.05	0.99	0.98	1.02	1.01	1.09	1.04	1.00	1.01	0.95	1.01	1.02	1.03	0.97	0.96	1.04	1.00	0.98	1.01	1.00
101P04378	0.83	0.81	0.85	0.83	0.87	0.92	0.81	0.85	0.84	0.80	0.85	0.85	0.86	0.84	0.83	0.89	0.83	0.81	0.86	0.84
101P04379	0.83	0.81	0.85	0.83	0.86	0.92	0.82	0.85	0.83	0.79	0.87	0.85	0.85	0.82	0.87	0.90	0.83	0.81	0.84	0.83
101P04380	0.81	0.79	0.82	0.83	0.83	0.85	0.84	0.85	0.81	0.79	0.86	0.87	0.85	0.90	0.92	0.89	0.84	0.83	0.80	0.82
101P04381	0.83	0.81	0.84	0.82	0.83	0.85	0.81	0.85	0.80	0.80	0.87	0.89	0.88	0.90	0.91	0.93	0.83	0.83	0.82	0.85
101P11461	3.24	7.40	2.89	2.62	2.47	2.23	2.39	2.82	2.45	2.33	2.17	2.40	2.95	3.09	2.87	2.16	2.72	2.50	3.44	3.93
101P11462	2.04	1.78	2.39	2.22	3.06	2.50	2.95	2.96	3.35	3.06	2.51	6.31	4.06	6.04	4.76	4.73	5.20	5.97	2.65	3.06

Figure 6. Data Representing Travel Time Index (TTI) for 20 Timeslots

4.1.3 Map Visualization of TTI

Using the TTI data for the study corridors, the morning peak and evening peak TTI maps for the month of February were prepared in ArcGIS. Prior to preparing the maps, the maximum TTI values out of 10 time slots of both morning peak hours and evening peak hours were prepared. In the map, the TTI values were categorized in 4 divisions (from low to high) which helps to identify the high congested area (i.e., those with high TTI value) and low congested area (i.e., those with low TTI). Using the pavement Crack Ratio, maps with cracks had also been prepared in the same way.

4.2 Pavement Data

This research utilized three different datasets for model training: 1) UAB Green Space, 2) Crack500, and 3) I-65 and US-31. The Crack500 was implemented to train our model, as it's mostly used and recognized in the industry/research community, to compare the built architectures and evaluate the model's performance effectively to known industry matrices. For the UAB Green Space, I-65 and US 31 panoramic images extracted from Google Street View were used. The panoramic images were pre-processed to be used as an input for the various machine learning algorithms tested and developed.

4.2.1 External Open-Source Dataset: Crack500

The GitHub repository was used to access open source “Crack500” dataset needed for model training and comparison purposes. GitHub is an internet hosting service for software development and version control using Git. It provides a distributed version control of Git, bug tracking, software feature requests, task management, and continuous integration.

The GitHub downloaded images consist of cracks from asphalt and concrete surfaces [79-81] with varying textures, lighting, and crack sizes as shown in Figure 7a. The images are annotated with the crack represented as white and the background/non-cracked images represented as black as shown in Figure 7b.


(a) **(b)**

Figure 7. GitHub Crack500 Images: (a) Ground Truth, (b) Cracked Images [82, 83]

The total images downloaded were 1552, of which 776 were Red, Green, Blue [RGB] ground truth images and the other 776 were crack annotated images. The images were sorted in labeled folders and fed to the image-to-array algorithm. Once pre-processed, the images underwent the same post-processing procedure. To keep with uniformity, TensorFlow/Kera’s augmentation parameters [flipping and rotating] were used to enhance the robustness of the dataset, which doubled the 776 filtered ground truth images and the 776 annotated crack images to a total of 3104 images. Finally, the fed images are split into training [60% (1862 images) of the dataset allocated to teach the model], validation [15 % (466 images) of the training dataset is used to evaluate the model’s learning process simultaneously as during training], and testing [25 % (776 images) to test the model on what it has learned with new data].

4.2.2 Google Street View Image Acquisition

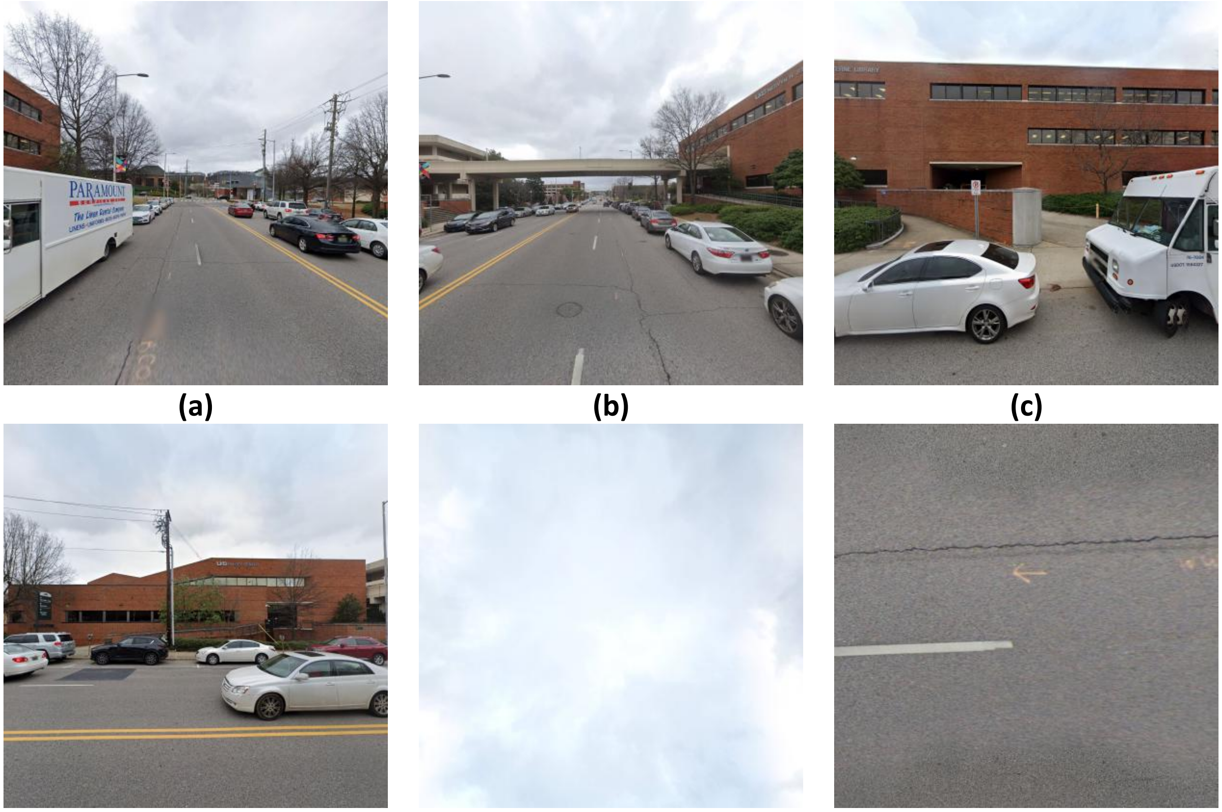
Google Street View (GSV), amongst other functions is an image dataset source that can be accessed by all, generates and embeds diverse images and roadways metadata without compromising resolution. The images are derived from a GSV car with a unique setup as shown in Figure 8a. To maintain consistent image capture conditions with the exemption of light [day/night] roadway conditions [wet/dry], and roadway material [asphalt/concrete]. The research focus area is mapped out as shown by the highlighted region in Figure 8b. The designated images were downloaded through a third-party software 'Street View Download 360' designed by Thomas Orlita as shown in Figure 8b. The downloaded images from Google Street View by default are retrieved in a panoramic format as shown in Figure 8c. The total panoramic images downloaded were 2009 images, these images were converted to 2-dimensional images using a CubeMap algorithm.

(a) (b)

(c)

Figure 8. (a) Google Street View image retrieval car/equipment, (b) Street View Download Interface, and (C) Google Street View Panoramic image

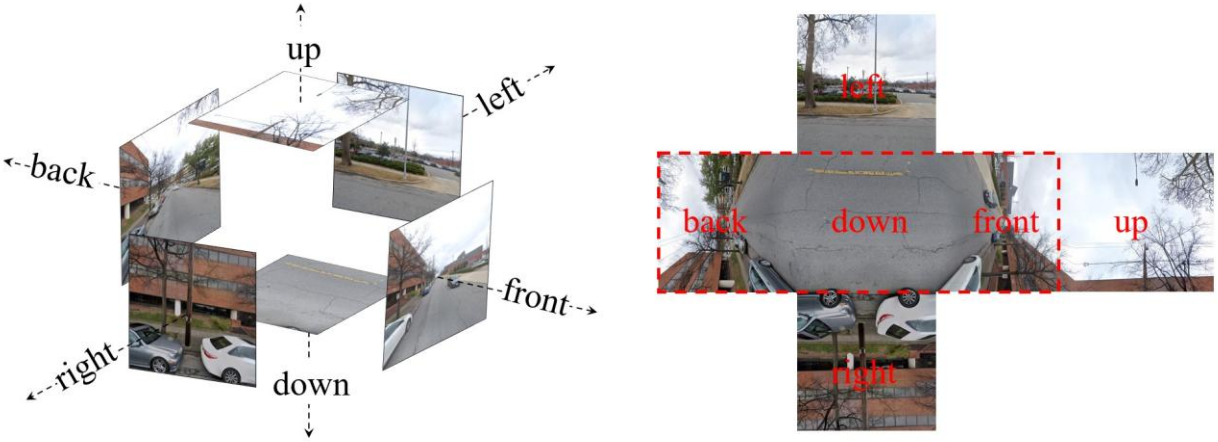
CubeMap returns six, 2-dimensional images, each view represents the orientation/direction of the Google vehicle as shown in Figure 9. A visual representation of its application is shown in Figure 10, resulting in a new total dataset of 12054 [2009 images by 6 CubeMap sides] images.



(a) **(b)** **(c)**

(d) **(e)** **(f)**

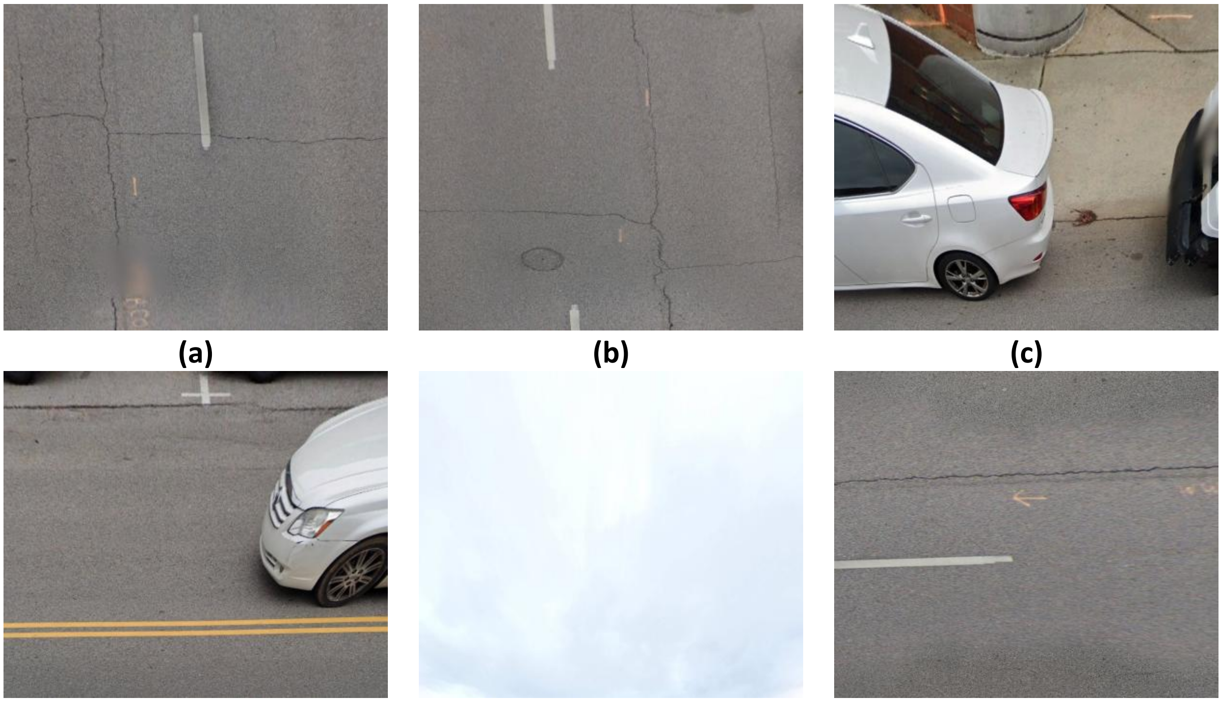
Figure 9. CubeMap output of panoramic image: (a) Back of GSV car, (b) Front of GSV car, (c) Right of GSV car, (d) Left of GSV car, (e) Top of GSV car, (f) Bottom of GSV car.



(a) **(b)**

Figure 10. CubeMap output: (a) Visual CubeMap representation, (b) Visual representation of attached CubeMap images by orientation

Those images were filtered (to remove debris, sidewalks, and buildings). The remaining unfiltered images were processed using an image management algorithm to transform, reshape, and correct the perspective as shown in Figure 11. The GPS metadata was recoded into the output of the image management algorithm to aid the restitching of the images. The output images were sorted for unwanted images and stored in labeled [Path 0-5] folders, where 0-5 represents (a-f) a specific view/orientation of the panoramic image. Therefore, using the principles of relatability, two hundred images were selected from the total based on the 6th/bottom image exhibiting roadway crack.

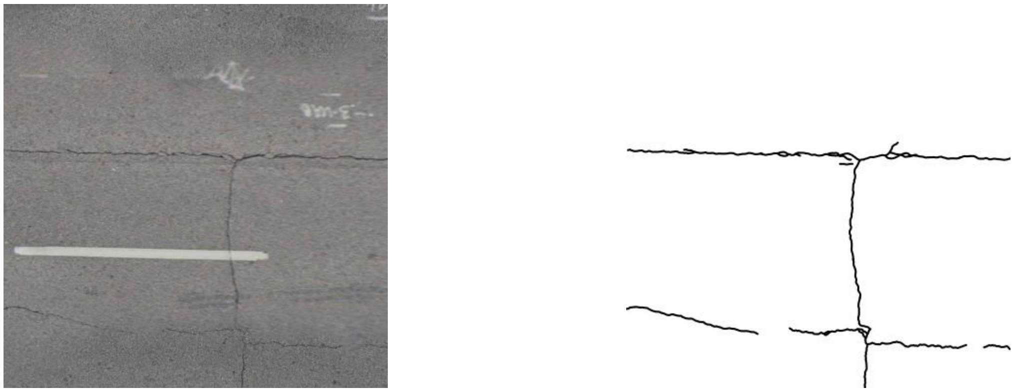


(a) **(b)** **(c)**

(d) **(e)** **(f)**

Figure 11. Image Management output of panoramic image: (a) Back of GSV car, (b) Front of GSV car, (c) Right of GSV car, (d) Left of GSV car, (e) Top of GSV car, (f) Bottom of GSV car.

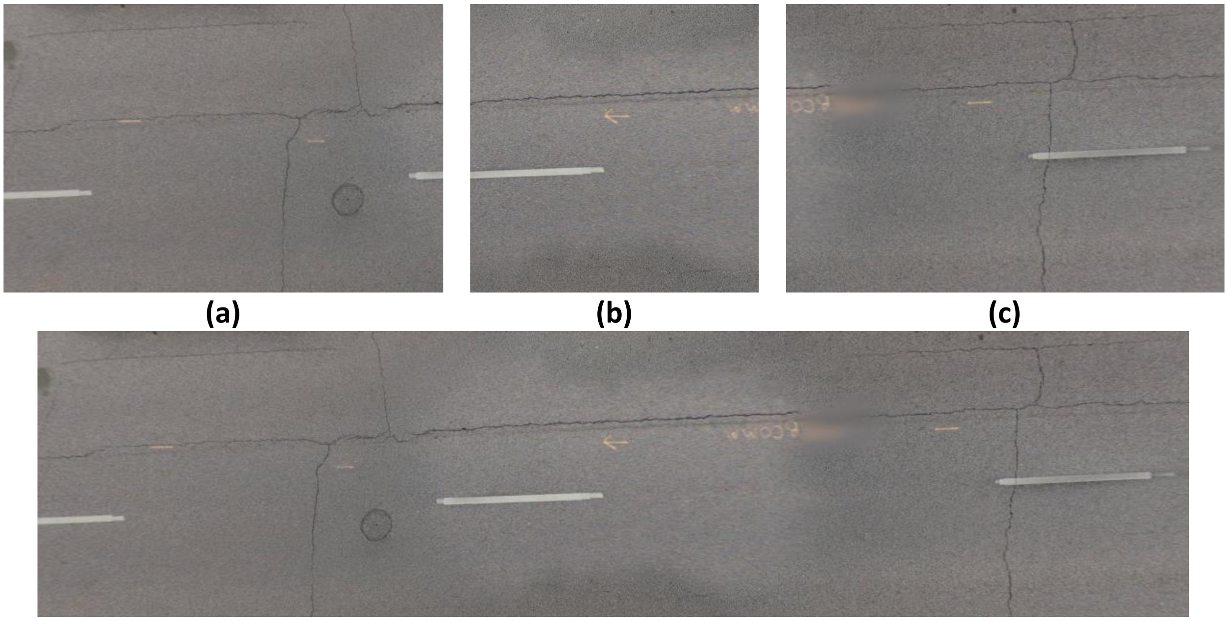
The stored 200 filtered images were annotated, using a third-party software ‘paint.net’ designed by Microsoft [82, 83]. Paint.net was used to create a masked version of the “Cracked images” as shown in Figure 12a, of the original RGB image “Ground Truth” as shown in Figure 12b in black and white, with white representing the background and black representing the cracks on the roadway.



(a) **(b)**

Figure 12. Visual representation of machine learning model Input Image; (a) Ground Truth, (b) Annotated Crack Image

Figure 13 represents image management including (a) rotated front Image, (b) untransformed bottom image, (c) rotated back Image, and (d) concatenated image of the CubeMap outputs. The images were saved in a different folder while keeping their original labeling and then fed into an image-to-array algorithm. Finally, each CubeMap output image was horizontally concatenated in the order and saved from the management algorithm as shown in Figure 13.



(a) **(b)** **(c)**

(d)

Figure 13. Image management: (a) rotated front Image, (b) Rotated back Image, (c) Untransformed bottom image, and (d) Concatenated image of the CubeMap outputs

The pre-processed 2048 pixels by 2048 pixels images were fed to the image-to-array algorithm. The algorithm resizes the images to 160 pixels by 160 pixels, converts them to a string, and saves it to memory. The images were resized for computational memory processing reasons and fed to a machine learning model [an FCN with deep dense convolutional layers] for training. TensorFlow/Kera's augmentation parameters [flipping and rotating] were used to enhance the robustness of the dataset, which doubled the 200 filtered ground truth images and the 200 annotated crack images to a total of 800 images. This is to address the issue of crack/non-crack unbalanced dataset [76]. Finally, the fed images are split into training [60% (600 images) of the dataset allocated to teach the model], validation [15% (120 images) of the dataset is used to evaluate the model's learning process simultaneously during training], and testing [25% (200 images) to test the model on what it has learned with new data].

4.2.3 Image Stitching

This section describes the efforts conducted to reattach (stitch) the GSV panoramic images that were converted to 2D by the CubeMap algorithm, back to a landscape image, highlighting only the roadway surface (i.e., for any designated area without the noise from buildings, plants, and other obstructions). The Google Street View images downloaded were accompanied by a .json file. The file holds important unstructured metadata for the images and was exported to Excel. This is to extract and organize the data in a structured manner chronologically using the GPS latitude (LAT) and longitude (LONG) coordinates. The processed data is saved as a .csv file. The file is fed into a data tracking algorithm appendix, and a GPS stitching algorithm. The data tracking algorithm converts the image pixels into GPS coordinates, which are superimposed onto a street map [OSM] as shown in Figure 14.

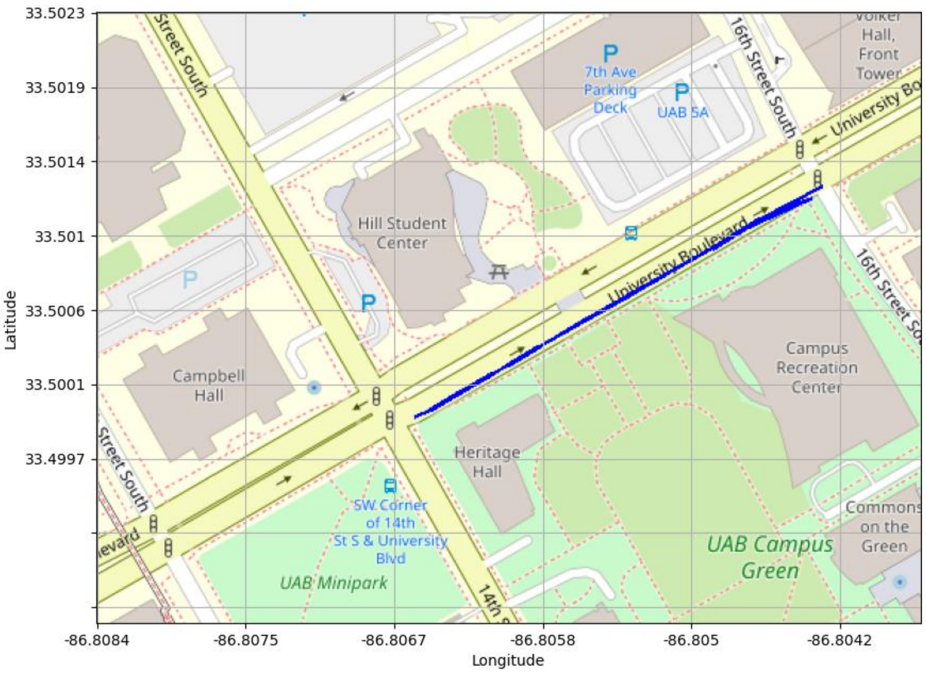


Figure 14. GPS Tracking of images in the dataset

Some images cannot be stitched due to the panoramic image lacking details of the intersection. Once the view is corrected through the image management algorithm, the corrected images lack pavement details to be used for the image stitching. Another limitation stems from the disjoint connection between two images from the driver switching lanes.

4.3 Crack Detection

4.3.1 General Description

Neural networks and more specifically, artificial neural networks (ANNs), are the backbone of machine/deep learning models as they imitate the human brain through a set of algorithms [84, 85]. At a basic level, a neural network is made up of four main components: input variables, weights, bias or threshold, and an output shown in Figure 15. An algebraic/ mathematical expression of those components is shown in Equation 2.

$$\sum_{i=1}^{m} w_i x_i + bias = w_1 x_1 + w_2 x_2 + w_3 x_3 \qquad (2)$$

Perceptron is used to define the features in the input data. Perceptron primarily leverages sigmoid neurons and represents values from negative infinity to positive infinity [16, 33]. Therefore, if the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Else, no data is transmitted to the next layer of the network. This process is constantly repeated for any decision made because of the numerous hidden layers [86]. Deep learning algorithms require each hidden layer to have a specific activation function, which facilitates the efficient transfer of information from the previous layer into the next one. Afterwards the outputs from the hidden layers are generated and used as inputs to calculate the final output of the neural network. This process is like linear regression but distinctively different because unlike linear regression a change in a single neural network weight imparts the total outcome [as the output of a changed weight layer, becomes the input of the next layer thus causing a domino effect].

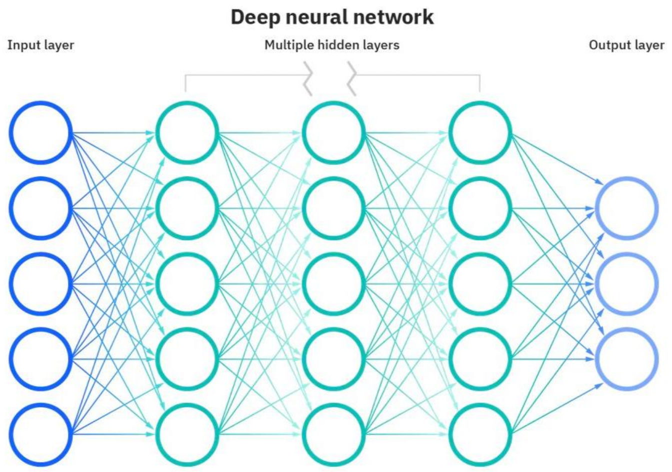


Figure 15. Neural Network Feed Forward Neurons Movement Error! Reference source not found.

Most deep neural networks are feedforward as shown in Figure 15 (i.e., the neuron's flow in one direction only, from input to output). However, backpropagation, when employed, causes the neurons to move in the opposite direction from output to input while calculating and attributing the error associated with each neuron. Neural networks can be classified into distinct types, based on their intended purposes [40]. Data usually is fed into these models to train them, and they are the foundation for computer vision, natural language processing, and other neural networks.

Convolutional neural networks (CNNs) are similar to feedforward networks, but they're usually utilized for image recognition, pattern recognition, and/or computer vision, the algorithm utilizes principles from linear algebra, particularly matrix multiplication, to identify patterns within an image, and finally recurrent neural networks (RNNs) are identified by their feedback loops, the algorithm primarily leverages using time-series data to make predictions about future outcomes, such as stock market predictions or sales forecasting [76].

4.3.2 Built Models

4.3.2.1 U-Net

Ronneberger et al. [79] proposed an elegant fully convolutional network architecture by Long et al. [80], by modifying, and extending the architecture operation such that training images generate precise segmentations. The network and training strategy relies on the strong use of data augmentation to maximize the available annotated samples more efficiently. While Long's approach was to contract the network by successive layers, substituting pooling operators for up-sampling operators to increase the resolution of the output; however, Ronneberger's modifications included many feature channels in the up-sampling, which allow the network to propagate context information to higher-resolution layers. The architecture as shown in Figure 16, consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. Drop-out layers were applied to the end of the contracting path, to perform further implicit data augmentation. Its shape yields a U-shaped architecture, hence the name U-Net.

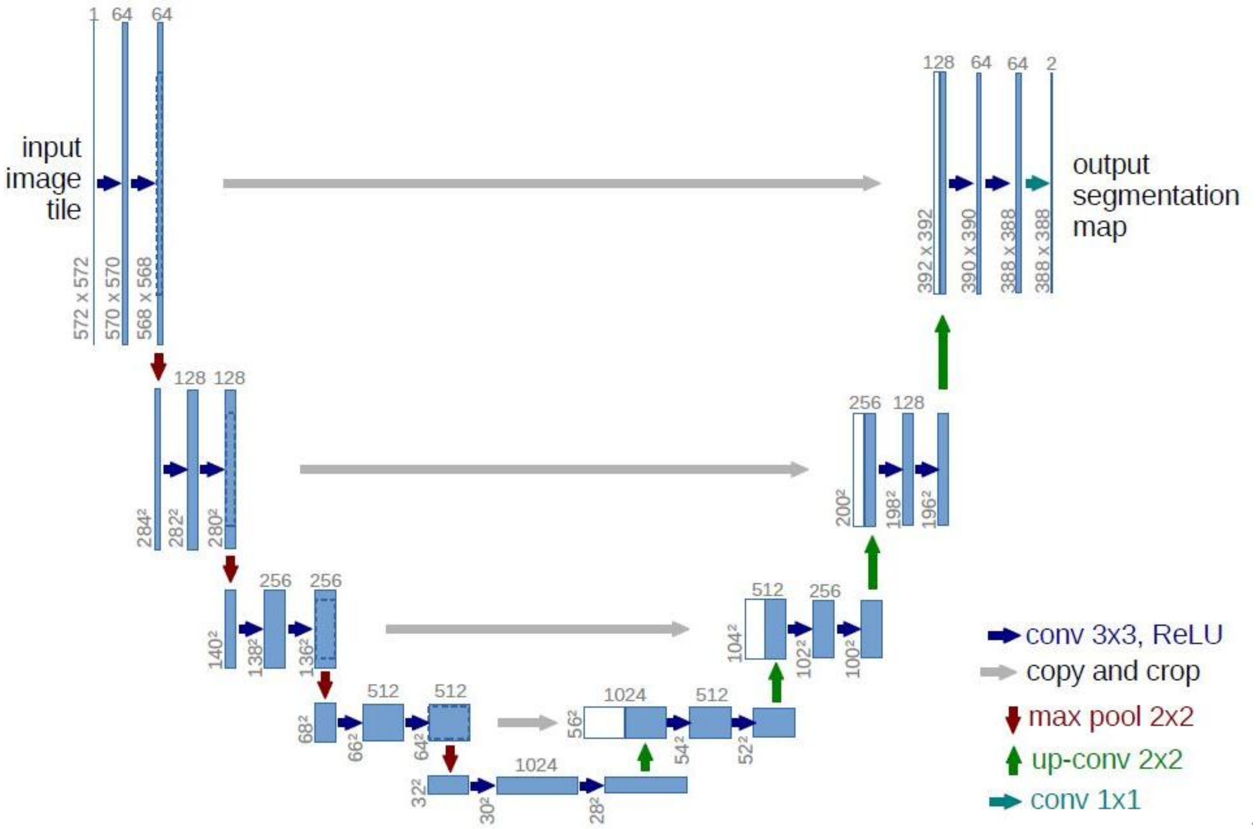


Figure 16. U-Net Architecture[79]

4.3.2.2 CrackPix

Proposed CrackPix, is a deep fully convolutional neural network (FCN) for pixel-level defect detection in concrete infrastructure systems. CrackPix leverages a modified VGG-16 proposed for the 1,000-class Image-Net challenge [81], which was pertained for image classification and dense predictions by transforming their fully connected layers into convolutional filters. The proposed architecture as shown in Figure 17, was built with four parallel NVIDIA Tesla P100 GPU nodes with 100GB Ram. Upsampling [2x, 4x, 8x, 16x, 32x], and resizing the prediction heatmap to the size of the input images. CrackPix can work on images of arbitrary size and provide pixel-level detection of the cracks. Image data used to develop the model was sourced from a wide variety of concrete surfaces that include bridges, buildings walls, slabs, and sidewalks at various distances and carefully annotated at the pixel level. All 5 unsampled [2x, 4x, 8x, 16x, 32x] versions were modeled; however, the 2x was the most precise in its prediction.

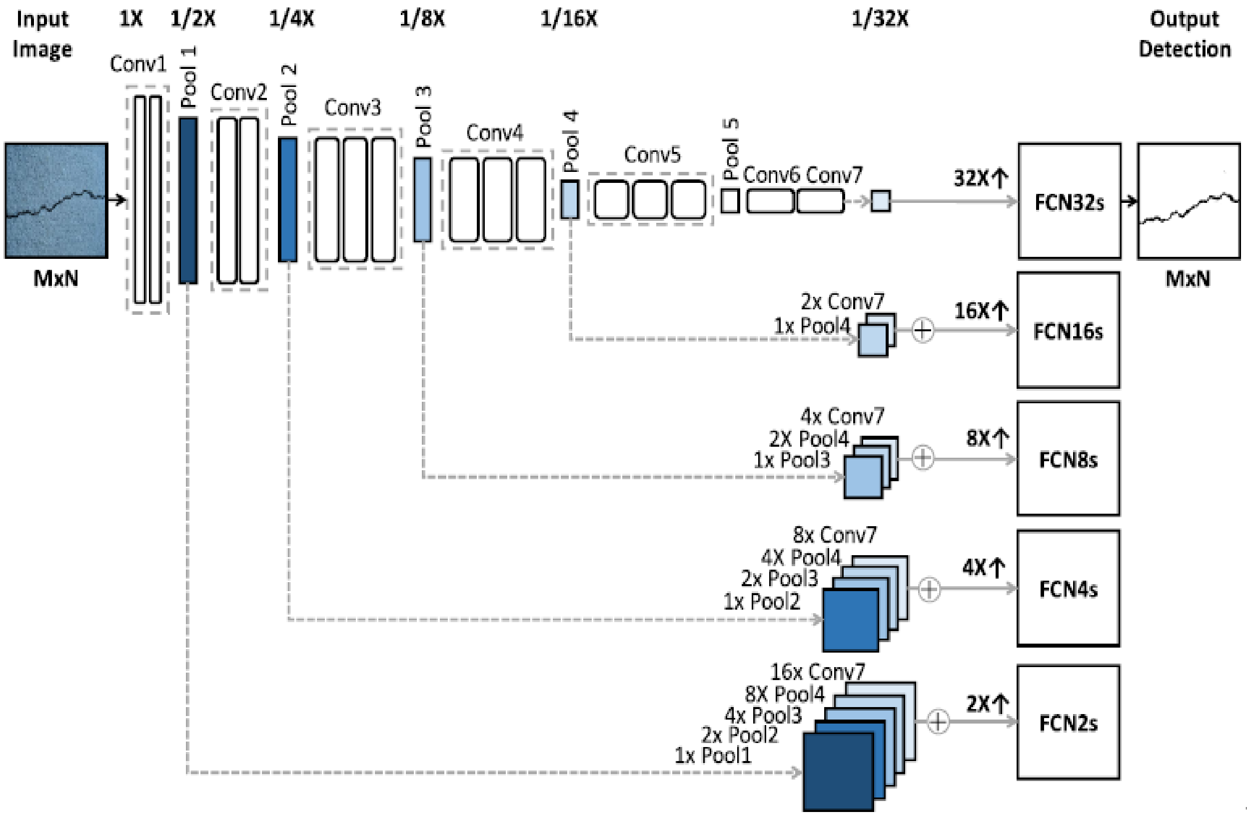


Figure 17. CrackPix Architecture[76]

4.3.2.3 DenseNet

Li et al. [87] proposed a densely connected and deeply supervised network as shown in Figure 18, to improve the accuracy and robustness of existing automated crack detection methods. In this approach, the densely connected layers are first applied to enhance the propagation and reuse of crack features, followed by the deeply supervised modules designed to make the network extract more noteworthy features through multi-scale levels, a deconvolution layer to resize the feature map to the original size with a cross-entropy loss function and lastly, a fusion model to fuse the image feature and semantic features from different levels. The proposed method evaluated three open-source datasets. Based on the similarity between crack detection and edge detection, Optimal dataset scale (ODS) and Optimal image scale (OIS) thresholding were applied to evaluate the model's performance. A portion of Crack500 images was used to train the proposed method, and the rest for test data, in addition to AEL and Cracktree200 datasets. To increase the robustness of the model image enhancement methods rotation and clipping are used to enhance the training data, with stochastic gradient descent (SGD) momentum adopted for network parameters optimization. The model was implemented and trained with a pyTorch framework and evaluated on a 64GB Ram, 11GB GeForce GTX 1080 Ti and 8700 CPU @ 3.2GHz.

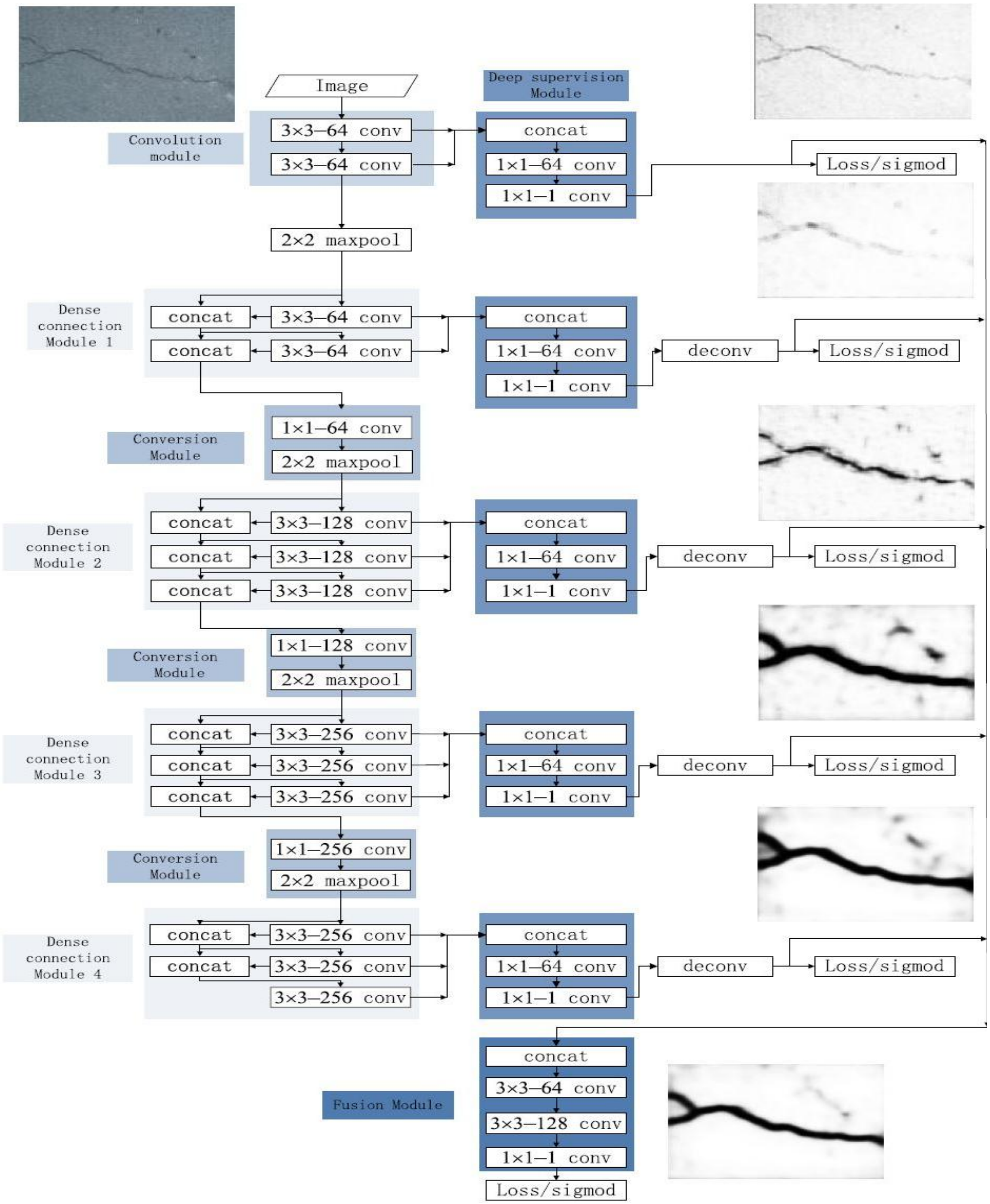


Figure 18. DenseNet Architecture[87]

4.3.2.4 UABiNet

Image classification based on convolutional neural networks requires a few sequentially organized convolutional and max/ average pooling layers, to extract features from any input [Image], with the fully connected layers performing the image classification, by designating one of the few categories to each image binary detection, in our case cracked or non-crack. UabiNet machine learning model uniquely combines the strength of three high-performing models, a fully connected network, a dense and deeply supervised CNN, and the U-Net symmetric contracting / expanding ability, which enables precise localization, to create an efficient deep dense fully connected network. The VGG-19 architecture was prioritized as opposed to the VGG-16 adopted by Alipour et al. [79], to make the model network deeper. UabiNet utilizes Alipour’s approach with the exemption of the max-pooling layer being replaced with average pooling. The arrangement for the architecture utilizes [87], with the exemption of the fusion module and U-Net’s contracting and expanding approaches with the exemption of the dropout layers.

The network as shown in Figure 19, is composed of convolutional, dense connection, conversion, and deconvolution layers. The network takes an RGB roadway image and outputs a crack prediction map with the same size as the input as shown in Figure 19. After the input, the multiscale feature maps are extracted by the convolution and dense connection modules, that are connected by the conversion modules which compresses the dense features from the previous modules to alleviate feature redundancy [87]. All filters were initialized from pre-trained VGG19 weights to transfer its feature extraction knowledge, except the upsampling layer, which was initialized to bilinear upsampling and then learned [76]. During training, the loss function of the feature maps generated by the model is calculated. The pooling operations reduce the sizes of the feature maps’ spatial resolution, therefore deconvolution/ transposed convolution is used to restore the feature map to the original image size before obtaining the final crack prediction map [76].

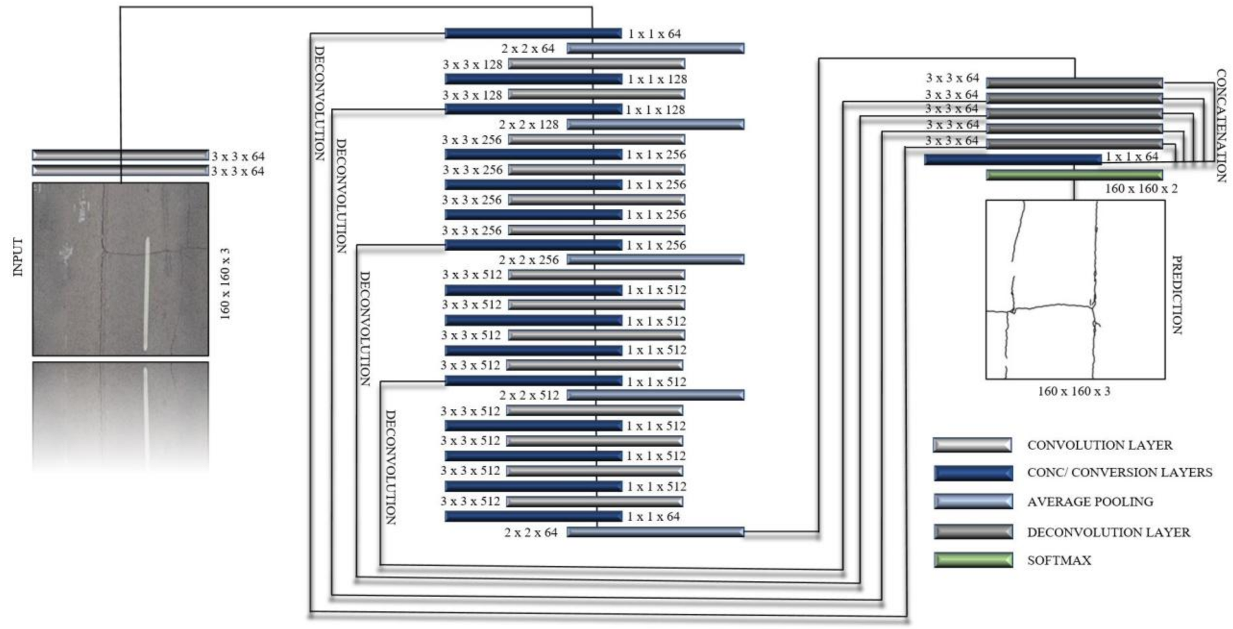


Figure 19. UABiNet Architecture

Using the VGG-19 blocks for dense modules, the dense connection module in each layer applies the concatenation of feature maps produced from all previous layers as one of the inputs for all preceding layers as shown in Figure 20.

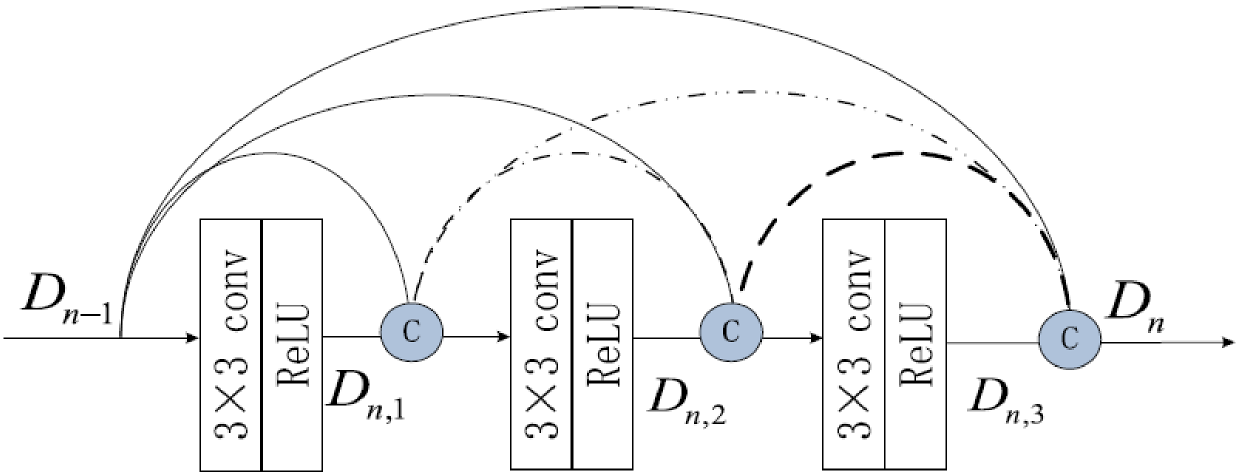


Figure 20. Dense Block Connection Mechanism[87]

The mathematical representation of the Dense Module in VGG-19 blocks is given in Equation 3.

$$Dn, l = Hl ([Dn,l, Dn;l, \ldots Dn, l-1]) \qquad (3)$$

where Dn,1 and Dn are input and output of the n-th dense connection module, and the output of the layers in the module are Dn,l [the output of the ith layer in the dense connection module n], while [Dn-l, Dn,l…, Dn,l-1] refers to the concatenation of feature maps from all layers 1,…,l-1. And the nonlinear transformation Hi (□) is a composite function of 3 x 3 convolutions. Incorporating dense connections of features in different layers, the modules can alleviate the gradient vanishing problem. The conversion modules connect the dense connection modules, by using a 1 x 1 convolution layer alongside a 2 x 2 average-pooling layer to facilitate calculation and to fuse features of various levels from the dense connection module.

Afterwards a devolution block receives the input and concatenates the features from previous dense modules in channel dimension using expanding channel depths like the Unet [64, 128, 256, 512]. During training, after each forward pass of data through the network, the per-pixel loss (error) between the predictions and ground truth is calculated using a Binary Focal loss function [119] as shown in Equation 4.

$$FL(pt) = -\alpha t(1 - pt)\gamma \log(pt) \qquad (4)$$

Images of cracked surfaces are usually dominated by non-crack pixels, causing a pixel classification issue that skews the prediction towards non-crack pixels. To balance resolve that, binary focal loss was implemented. The Focal Loss is designed to address object detection scenario with an extreme imbalance between foreground and background classes during the training [88].

The model was also trained using the stochastic gradient descent (SGD) algorithm, which minimizes the loss function by iteratively updating the weights on small random sets of training data. In each epoch, the gradients of loss concerning the weights are calculated and a learning rate of the gradient is added to the weights from the previous step. It incorporates a contracting path and an expansive path, with repeated application of padded two 3 x 3 convolutions, each followed by a ReLU, a 2x2 average pooling operation with a stride of 2 for down-sampling, and batch normalization. After each average pooling step, the feature channels are doubled (i.e., 64, 128, 256, 512), and the expansive path up-samples the feature map followed by a concatenation. In total the network has 16 convolutional layers. The training was performed for 90 epochs with a batch size of 3 on 160 x 160 input images. A learning rate of 10^{-3} was used for training of the model. Implementation and training of the UabiNet models were conducted on Jupyter Notebook within Cheaha [UAB super-computer], utilizing the Anaconda environment. Jupyter Notebook is a Python package with tensor computations.

4.3.3 Crack Index

Once the model is completely post-processed, it is trained and evaluated on its training and testing dataset, saved, and further evaluated specifically on the US Highway dataset, to detect the highway crack and to determine the correlation, if any, between the roadway conditions and traffic data. The input images for the testing are fed into an image management algorithm, which converts the RGB images of the highway into grayscale shown in Figure 21a.

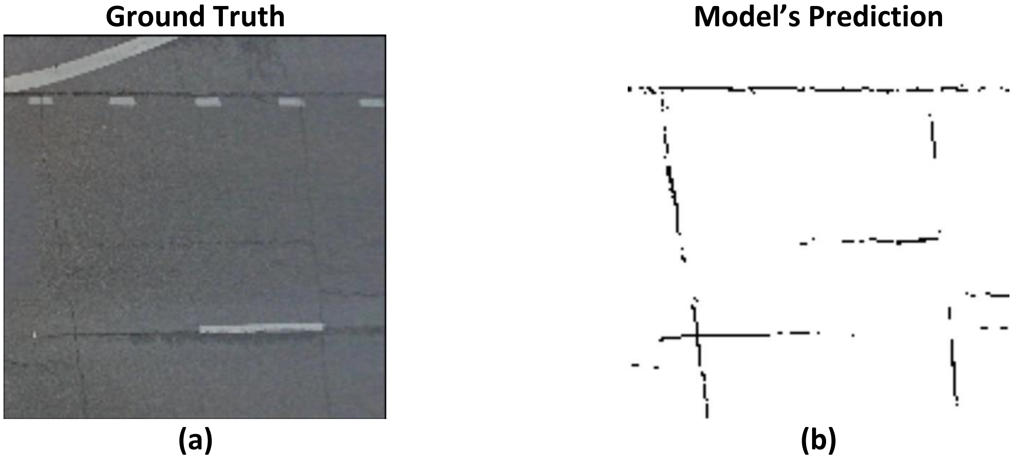


Figure 21. Highway RGB to Grayscale Image and Crack Detection

The borderless output [965 x 966] from the image management algorithm is then saved chronologically and fed into a crack count algorithm. The algorithm counts the white and black pixels, determines if it is cracked or uncracked, and exports the information to Excel for the crack index to be determined. Using Excel, the crack index is calculated on a pixel level, using Equation 5.

$$Crack\ Index = \frac{Total\ Crack\ Pixels\ per\ TMC}{Total\ Pixels\ Per\ TMC} \tag{5}$$

4.4 Statistical Analysis

To determine potential correlation, the two variables (TTI and Crack Index) were normalized prior to performing the t-tests. The purpose of normalization is to convert variables into a similar scale in order to improve the performance and training stability of the dataset. One of the four normalization techniques named 'scale to range' was used. The data were converted from their natural range to a standard range, usually 0 to 1. The formula used for normalization of the TTI and Crack Index is represented by Equation 6:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{6}$$

where x' is the normalized value, x is the raw value, x_{min} is minimum value of raw data, and x_{max} is maximum value of raw data. Once the data were normalized, the maximum and minimum values from the 5 years datasets for both variables were taken for performing Pearson's correlation and t-tests. At first, a Pearson correlation coefficient was used to identify the linear association between the two variables (TTI and Crack Index). In general, a positive correlation means that as one variable increases, the other one increases too and vice versa. Also, a negative correlation means that when one variable increases the other one decreases and vice versa. For the analysis, Microsoft Excel's function was used. To determine if the correlation coefficient was statistically significant, a t-test had been performed, which involved calculating a t-score and a corresponding p-value. The formula to calculate the t-score is given by Equation 7:

$$t = \frac{r\sqrt{n-2}}{1-r^2} \tag{7}$$

where t is the t-test score, r is the correlation co-efficient, and n is the number of observations (i.e., sample size). In the t-test, two types of hypotheses are defined, namely the null hypothesis and the alternative hypothesis. The null hypothesis specifies that there is no significant correlation between the two groups by default. A t-value that is greater than the critical value (t_c), the null hypothesis is rejected. If the t-value is less than the critical value (t_c), the null hypothesis should be accepted, and the alternative hypothesis will be rejected. The confidence interval (i.e., level of significance) should have to be fixed for determining the critical value. the level of significance (i.e., p-value) was considered as 0.10. Using the t-table chart, the critical value (t_c) was identified. The p-value is calculated as the corresponding two-sided p-value for the t-distribution with n-2 degrees of freedom. The t-test analysis was applied for both maximum and minimum normalized values of TTI and Crack Ratio as well as for a year-by-year basis.

5 Results and Discussions

5.1 Pavement Cracks

5.1.1 UNet

This model was built with a binary focal loss function. The dataset was divided into training [60%: 600 images], validation [15%: 120 images], and testing [25%: 200 images]. In each iteration/epoch, the training set was used to train the model. The model was evaluated on the validation set to report its performance. Each layer is batch normalized, a Rectified Linear Unit Function [RELU] activation is added to the fully connected layers and a SOFMAX activation was adapted on the last fully connected layer for binary classification of crack versus non-crack. The model had a total of 1,947,970 parameters [1,944,546 trainable and 3,424 nontrainable]. Using the training dataset, the model trained for 94 epochs, showing the convergence of the weights earlier due to less trainable parameters. The model completed each epoch for training and validation under 1 minute, and the changes in the loss function and the crack MIOU during training are shown in Figure 22. The model's prediction after training with the UAB dataset was evaluated on the training and testing ability as shown in Figure 23.

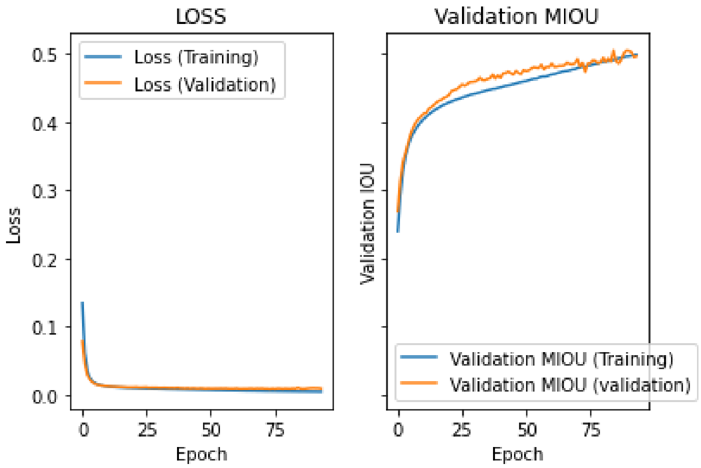


Figure 22. UNet Left [Epoch against Loss] and Right [Epoch against MIOU]

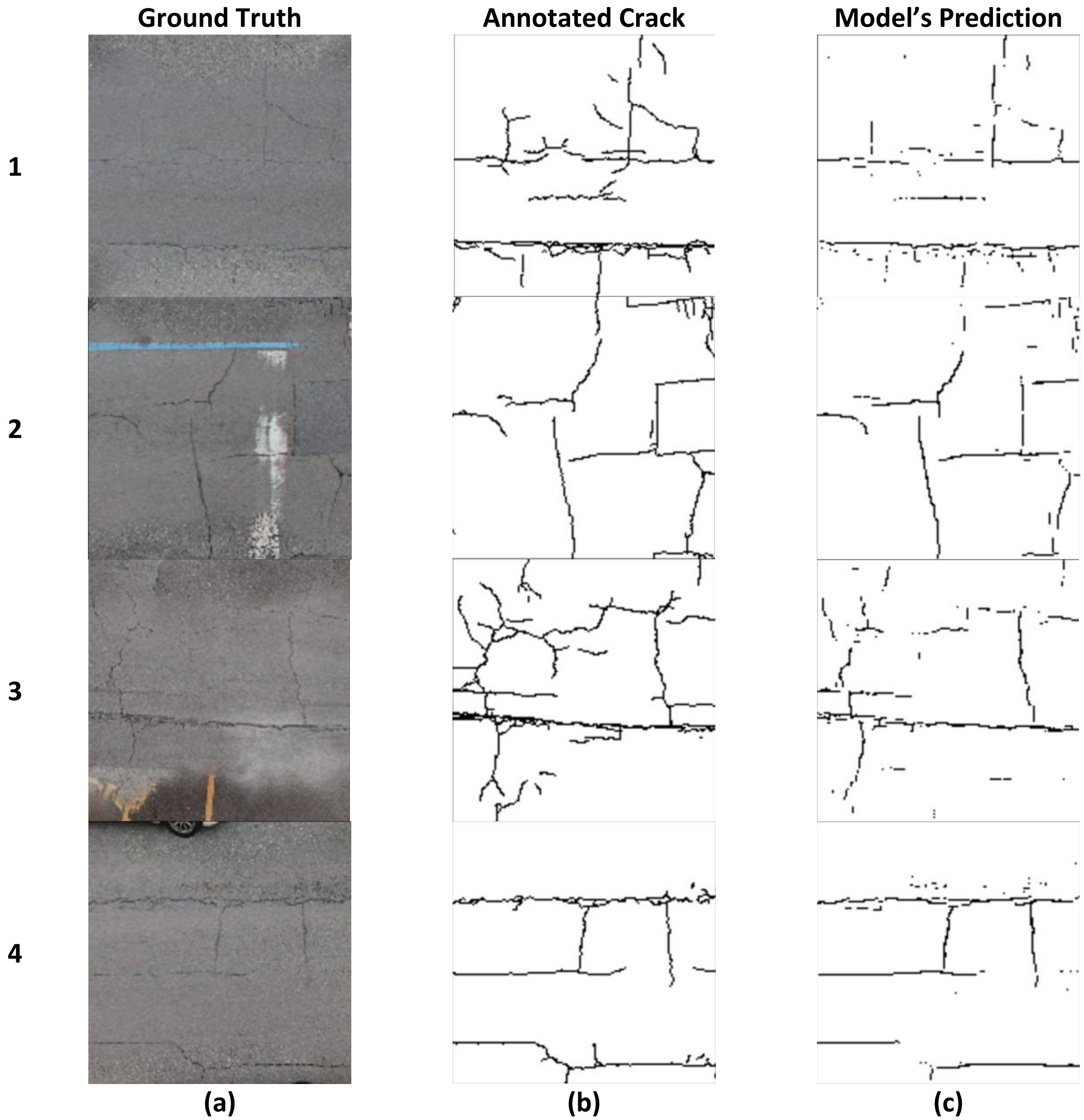


Figure 23. UAB data comparison of U-Net architecture prediction between training data and testing data (a) Ground Truth, (b) Annotated Image, (c) Training Prediction

5.1.2 CrackPix

CrackPix utilized TensorFlow/Kera's library to retrieve the VGG-16 model with its pertained ImageNet weights. The model is composed of a total of 19 layers including the input layer. The model input was a 160 x 160 x 3 RGB image. The dataset was divided into training [60%: 600 images], validation [15%: 120 images], and testing [25%: 200 images]. In each iteration/epoch, the model was evaluated on the validation set to report its performance. Each layer is batch normalized, a RELU activation is added to the fully connected layers. A SOFMAX activation was adapted on the last fully connected layer for binary classification of crack versus non-crack. Using the training dataset, the model was trained for 96 epochs. The model completed each epoch for training and validation in under 3 minutes. The model had a total of 137,191,882 parameters [137,164,614 trainable and 27,268 nontrainable]. Figure 24 presents the variations in the loss function and crack IOU during training for the fully connected network with a 2x up-sampling. The models were evaluated after training, on both the training and testing dataset as shown in Figure 25.

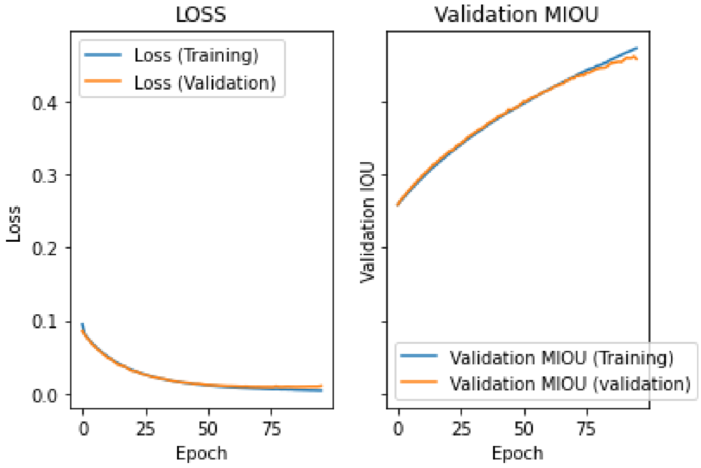


Figure 24. CrackPix Left [Epoch against Loss] and Right [Epoch against MIOU]

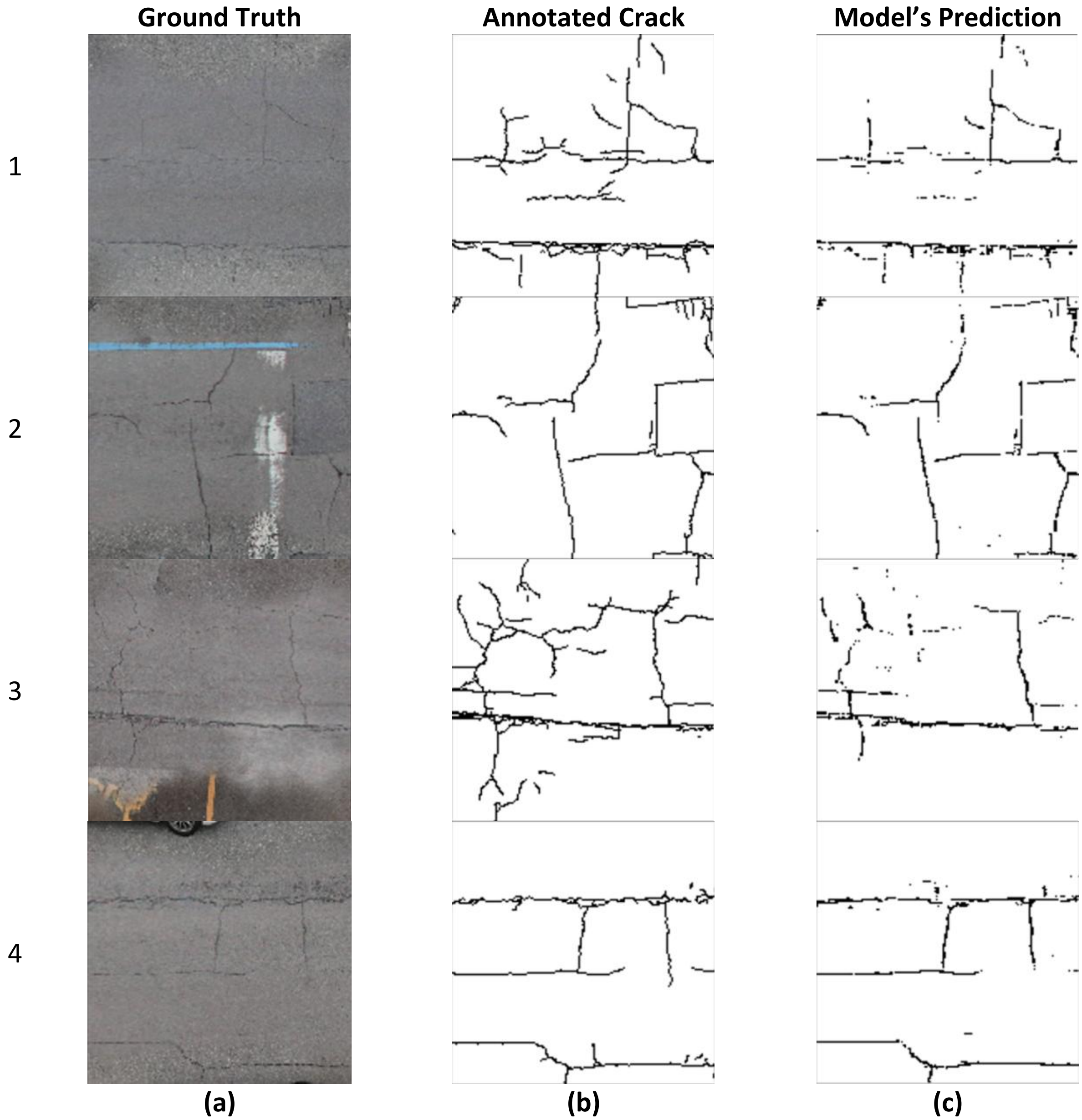


Figure 25. UAB data comparison of CrackPix architecture prediction between training data and testing data (a) Ground Truth, (b) Annotated Image, (c) Training Prediction

5.1.3 DenseNet

This model applied UAB dataset [160 x 160 x 3 RGB image] as input. This model had a binary focal loss function. The model input was a 160 x 160 x 3 RGB image. The dataset was divided into training [60%: 600 images], validation [15%: 120 images], and testing [25%: 200 images]. In each iteration/epoch, the training set was used and then evaluated on the validation set to report its performance. Each layer is batch normalized, a RELU activation is added to the fully connected layers. A SIGMOID activation was adapted on the last fully connected layer for binary classification of crack versus non-crack. The model had a total of 3,998,098 parameters [3,993,472 trainable and 4,626 nontrainable]. Using the training dataset, the model trained for 37 iterations/epochs, showing the convergence of the weights as shown in Figure 26. The model took approximately 1 minute and 5 seconds to complete. The model was evaluated after training, on both the training and testing dataset as shown in Figure 27.

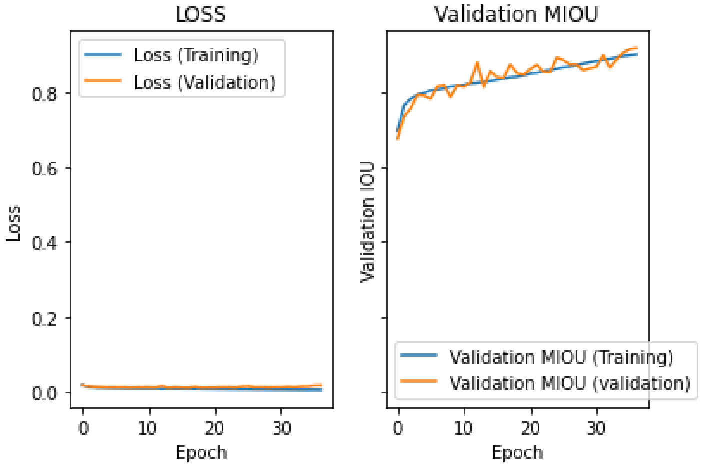


Figure 26. DenseNet Left [Epoch against Loss] and Right [Epoch against MIOU]

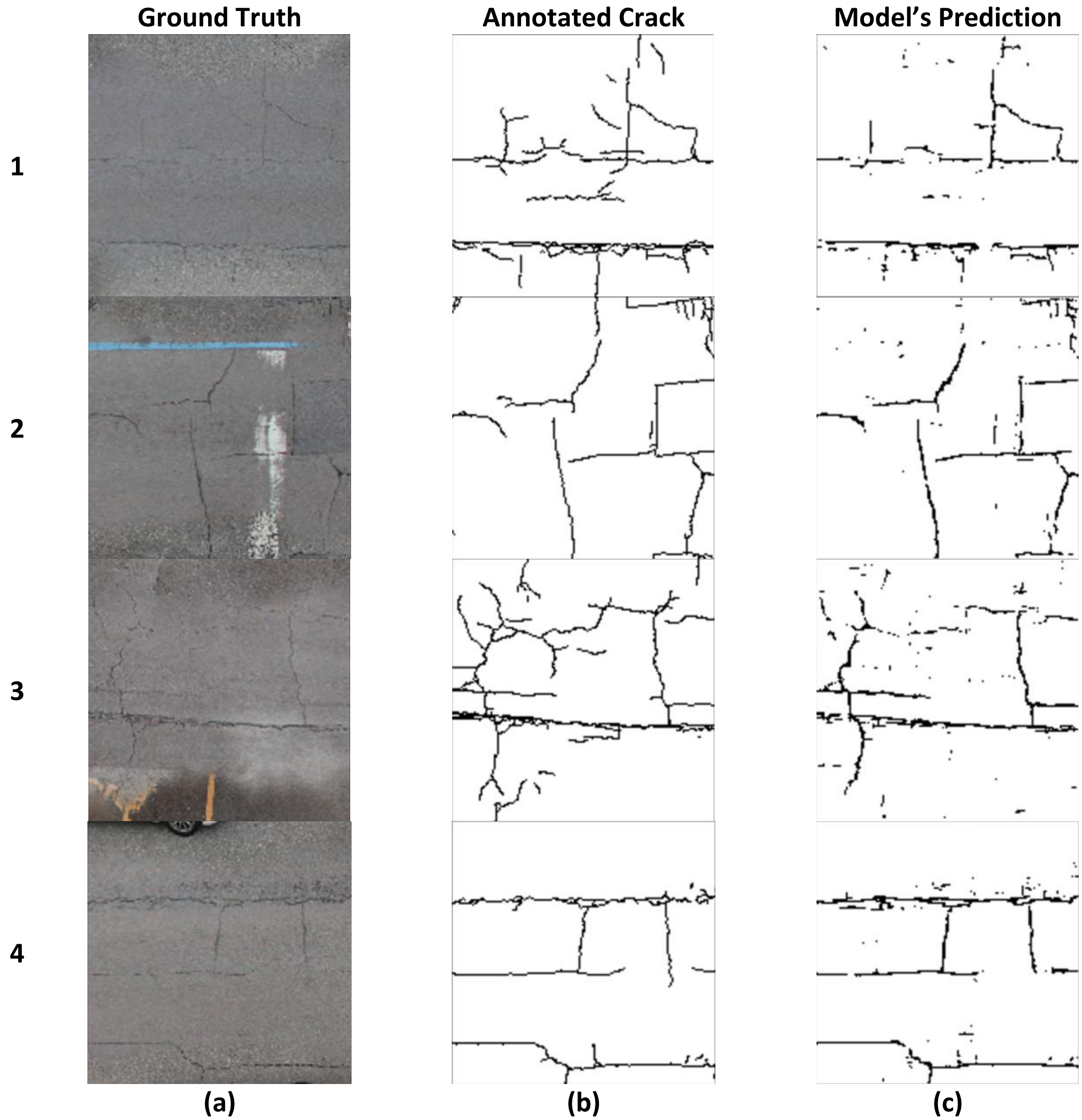


Figure 27. UAB data comparison of DenseNet architecture prediction between training data and testing data (a) Ground Truth, (b) Annotated Image, (c) Training Prediction

5.1.4 UABiNet

The model's input was reduced from 1440 x 2560 to 160 x 160 RGB image. The dataset was divided into training [60%: 600 images], validation [15%: 120 images], and testing [25%: 200 images]. The model was trained for 46 iterations/epochs, showing the convergence of the weights earlier as shown in Figure 50. However, it converged later than the other models. The model completed each epoch for training and validation in under 20 minutes. The model was evaluated after training, on both the training and testing dataset as shown in Figure 51.

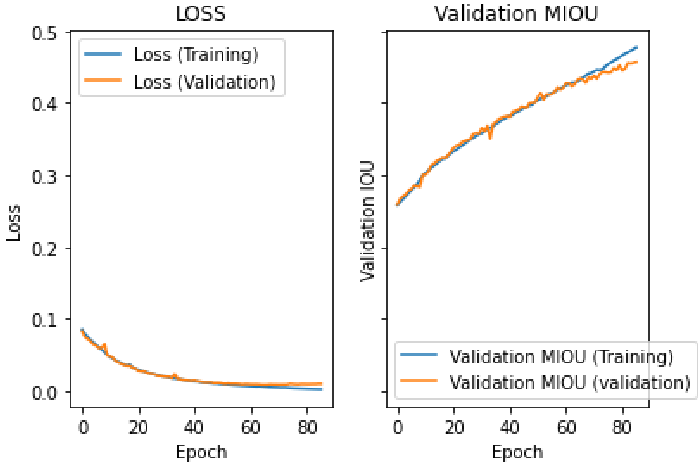


Figure 28. UABiNet Left [Epoch against Loss] and Right [Epoch against MIOU]

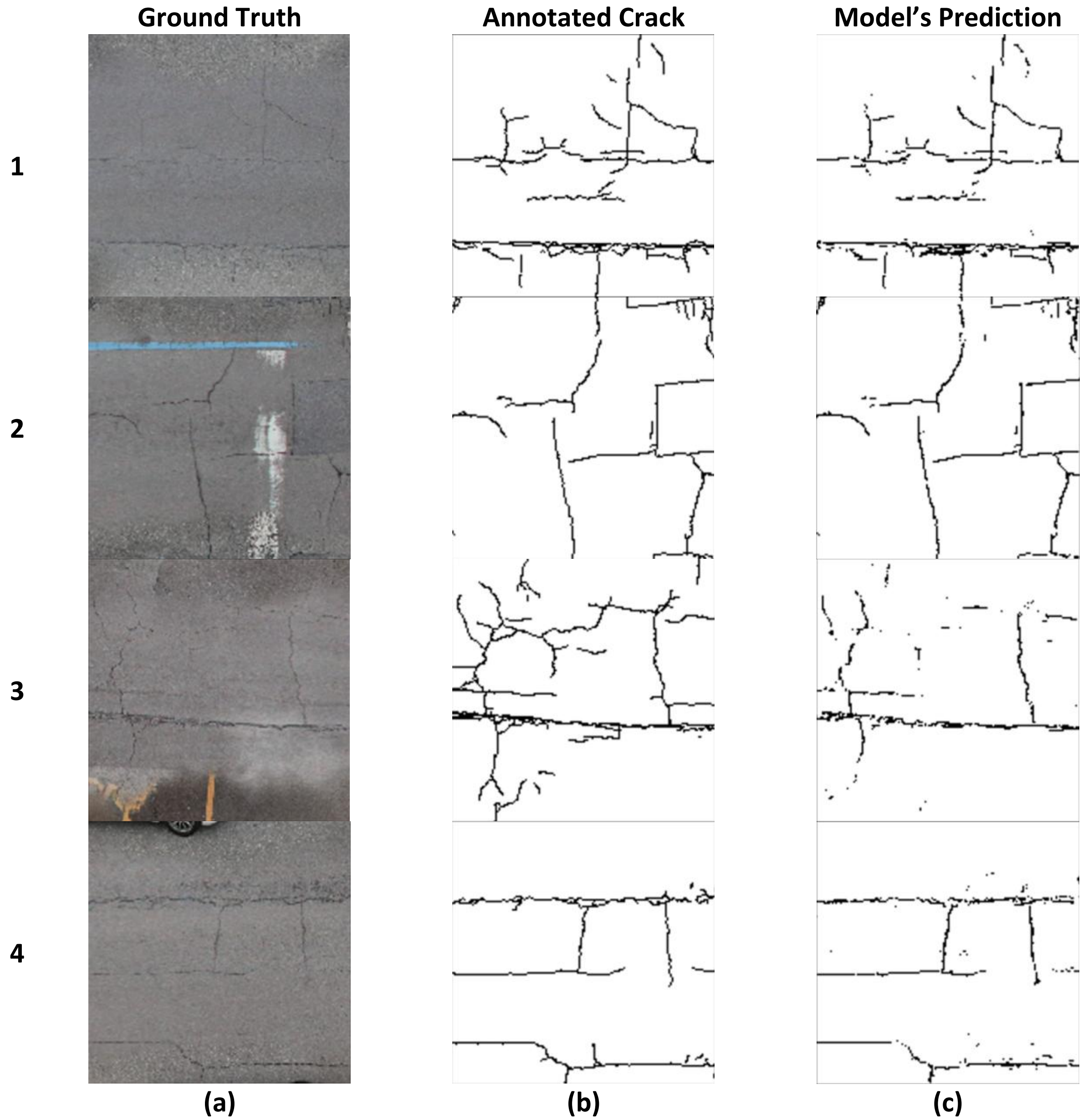


Figure 29. UAB data comparison of UabiNet architecture prediction between training data and testing data (a) Ground Truth, (b) Annotated Image, (c) Training Prediction

5.1.5 Comparison of Models

The model's performance was evaluated on both the training and testing. The models were evaluated based on the ratio of correct crack predictions to total crack pixels [Recall], the ratio of correct crack precisions to all crack predictions [Precision], the harmonic means of precision and recall [F1 Score]. Lastly, the ratio of the intersection of the ground truth and the predictions to their union the intersection over union [MIOU] was computed. Table 3 and Table 4 summarize the performance metrics comparison of all the models on the training and testing datasets. As it can be seen, UABiNet exhibits superior performance in all performance metrics considered, and this is the preferred model among the four models considered in this study.

Table 3. Performance Metric on Training Dataset

Metric	UNET	CrackPix	DenseNET	UABiNet
Recall	0.7486	0.7882	N/A	0.8301
F1 Score	0.5759	0.5975	N/A	0.7610
Precision	0.4680	0.4810	N/A	0.7026
MIOU	0.6962	0.7074	0.6582	0.8033

Table 4. Performance Metric on Testing Dataset

Metric	UNET	CrackPix	DenseNET	UABiNet
Recall	0.6716	0.6287	N/A	0.6706
F1 Score	0.4684	0.4145	N/A	0.4801
Precision	0.3596	0.3092	N/A	0.3739
MIOU	0.6459	0.6233	0.6266	0.6510

5.2 Data Correlation

Table 5 encompasses all the Traffic Message Channel (TMCs) along I-65 and US-31 that were considered in this study and was used to examine potential correlations between TTI and Crack Index. The analysis involved a sample size of 57, with 39 TMCs for I-65 and 18 TMCs related to US-31. The t-tests were conducted over a span of five years, from 2018 to 2022, and presented on a yearly basis. The correlation between TTI and Crack Index was found to be negative, indicating a weak and counter-intuitive statistical relationship. The year-by-year analysis showed that the t-test value for the year 2021 exceeded the critical t-value, signifying their statistical significance at a 90% confidence level.

Table 5. Correlation and t-test values of all 57 TMCs

Statistic	**2018**	**2019**	**2020**	**2021**	**2022**
r	-0.1437	-0.0493	-0.2096	-0.2438	-0.0454
n	57	57	57	57	57
t-test score	-1.077	-0.366	-1.589	-1.864	-0.337
p-value	0.286	0.715	0.117	0.067	0.742
Results	Not significant	Not significant	Not significant	Significant	Not significant

Note: **Critical t (tc) 90% Confidence Interval = 1.67**

Table 6 presents the analysis of the Traffic Message Channels (TMCs) specifically for I-65, exploring their correlation and t-test results. The study utilized a sample size of 39 TMCs and comprised a five-year period, from 2018 to 2022, with the findings organized on an annual basis. The correlation between TTI and Crack Index was generally negative, indicating a weak statistical relationship, except for the year 2019 when a positive correlation was observed, although statistically weak. Furthermore, the t-test values for both 2019 and 2021 surpassed the critical t-value, demonstrating their statistical significance at a 90% confidence level.

Table 6. Correlation and t-test values of I-65 (39 TMCs)

Statistic	**2018**	**2019**	**2020**	**2021**	**2022**
r	-0.267	0.278	-0.265	-0.268	-0.053
n	39	39	39	39	39
t-test score	-1.685	1.761	-1.673	-1.693	-0.322
p-value	0.1004	0.087	0.103	0.098	0.749
Results	Not significant	Significant	Not significant	Significant	Not significant

Critical t (tc) 90% Confidence Interval = 1.684

Table 7 presents the correlation and t-test outcomes specifically for US-31. The investigation involved a sample size of 18 and comprised a five-year period, from 2018 to 2022, with the findings organized on an annual basis. The correlation between TTI and Crack Ratio exhibited a predominantly negative trend, suggesting a weak statistical relationship. Moreover, the t-test values for all the years were found to be lower than the critical t-value except for the year 2018, indicating their statistical insignificance at a 90% confidence level.

Table 7. Correlation and t-test values of US-31 (18 TMCs)

Statistic	2018	2019	2020	2021	2022
r	0.4983	-0.2903	0.0614	-0.1537	0.0091
n	18	18	18	18	18
t-test score	2.299	-1.213	0.246	-0.622	0.036
p-value	0.0353	0.243	0.808	0.543	0.972
Results	Significant	Not significant	Not significant	Not significant	Not significant

critical t (tc) 90% Confidence Interval = 1.746

Table 8 considers the relationship between TTI and Crack Index for all TMCs in two different viewpoints: one considers the maximum TTI, and maximum Crack Index observed over the course of five years for each TMC, while the other focuses on the minimum TTI and minimum Crack Index within the same five-year period. The findings reveal a negative and weak correlation among the 57 observations. Additionally, the t-test values for both perspectives were found to be lower than the critical t-value, signifying their statistical insignificance at a 90% confidence level.

Table 8. Correlation and t-test values for all TMCs considering Max TTI and Max Crack Ratio

Statistic	**Max TTI and Max Crack Ratio**	**Max TTI and Max Crack Ratio**
r	-0.126698	-0.1631
n	57	57
t-test value	-0.947248	-1.2262
p value	0.347	0.225
Result	Not Significant	Not Significant

critical t (tc) at 90% confidence level = 1.67

6 Conclusions and Recommendations

This research successfully demonstrated the use of machine learning algorithms to assess pavement quality from Google Street View images. The Travel Time Index (TTI) was used to quantify traffic congestion and a pavement Crack Index was developed to quantify pavement deterioration. Attempts were also made to establish correlations between those two indices, thus examining the possible impact of cracked pavements on travel time in an urban setting.

Although the research hypothesis proposed a positive correlation between the two indices, suggesting that higher travel time would be associated with a higher crack ratio, the results obtained were mixed and counterintuitive. In the year-by-year analysis, statistically significant findings were observed for 2020 and 2021 (Table 5); however, the correlations were inverse and weak. Similarly, when focusing solely on I-65, statistically weak positive relationships were discovered in 2019, while statistically negative and poor relationships were found in 2021 (Table 6). Tables 7 and 8 presented statistically insignificant results, which are challenging to explain.

One possible explanation for these outcomes could be the omission of other congestion-related factors, such as changes in traffic volume, changes in traffic composition, impacts from crashes, and impacts from weather conditions and other environmental factors. Another factor to consider is that I-65 and US-31 are separate facilities with different characteristics and potentially different maintenance or physical management approaches, making the results difficult to interpret. Moreover, the sample size remains relatively small, with only 57 samples total, which limits the conclusiveness of the findings. To reach more reliable conclusions, it would be beneficial to augment the dataset with additional data points, allowing for more robust statistical analyses. Another limitation in this research is the lack of data regarding the maintenance locations and timelines that were conducted during the five-year period of the analysis (i.e., 2018 to 2022). Also, it is worth noting that the traffic patterns for the years between 2020 and 2022 are heavily altered due to the COVID-19 pandemic and shift towards remote working environments. Therefore, it is recommended to expand the analysis on different types of corridors and vastly assess traffic and pavement data on a regional level (i.e., city and state). Also, it is essential to consider the weather patterns, pavement maintenance data and scheduling, and traffic patterns. Furthermore, analyzing the data monthly rather than on a yearly basis can provide further insight. For future investigations, it is recommended to assess the potential correlation of TTI values and alternative pavement cracking indices such as Present Serviceability Index (PSI), and International Roughness Index (IRI). PSI and IRI are representative of segments with high volumes and poor pavement conditions. Therefore, the consideration of such indices might provide a better understanding of the pavement condition and traffic congestion. Moreover, the congestion should be analyzed in reference to the shape of flow curves and pavement quality.

To mitigate congestion and improve network performance, it is important for departments of transportation to prioritize locations in urgent need of pavement improvements. By implementing methodologies for pavement inspection, such as the use of images collected by vehicles with mounted cameras, drones, and traffic closed-circuit television (CCTV) cameras, timely interventions for deteriorated pavement can be made to reduce the likelihood of congestion occurrence. Overall, monitoring congestion that accounts for pavement conditions and their impact on traffic performance is highly desirable as it can identify actions that are appropriate and beneficial to combat congestion and improve network performance.

7 References

[1] Downs A. Still stuck in traffic: coping with peak-hour traffic congestion. Brookings Institution Press; 2005 Jun 22.

[2] Kukadapwar SR, Parbat DK. Modeling of traffic congestion on urban road network using fuzzy inference system. American Journal of Engineering Research. 2015;4(12):143-8.

[3] Das AK, Saw K, Katti BK. Traffic congestion modelling under mixed traffic conditions through fuzzy logic approach: an Indian case study of arterial road. 12th international on TPMDC, IIT Bombay, India. 2016 Dec:19-21.

[4] Bovy, P.H.L. and Salomon, I. (2002) Congestion in Europe: measurements, patterns and policies, in E. Stern, I. Salomon and P.H.L. Bovy (eds.), Travel Behaviour: spatial patterns, congestion and modelling, Cheltenham: Edward Elgar.

[5] Aftabuzzaman M. Measuring traffic congestion-a critical review. In30th Australasian transport research forum 2007 Sep (pp. 1-16). London, UK: ETM GROUP.

[6] Systematics C. Traffic congestion and reliability: Trends and advanced strategies for congestion mitigation. United States. Federal Highway Administration; 2005 Sep 1.

[7] FHWA. (2020) Traffic Congestion and Reliability: Trends and Advanced Strategies for Congestion Mitigation, Federal Highway Administration- US Department of Transportation, https://ops.fhwa.dot.gov/congestion_report/chapter2.htm

[8] GEOTAB. (2018) what causes traffic congestion?

[9] Handy S. Regional transportation planning in the US: An examination of changes in technical aspects of the planning process in response to changing goals. Transport Policy. 2008 Mar 1;15(2):113-26.

[10] Pratt RH, Lomax TJ. Performance measures for multimodal transportation systems. Transportation Research Record. 1996 Jan;1518(1):85-93.

[11] Karlaftis M, Kepaptsoglou K. Performance measurement in the road sector: a cross-country review of experience. International Transport Forum Discussion Paper.

[12] Cambridge Systematics. A guidebook for performance-based transportation planning. Transportation Research Board; 2000.

[13] Mannering FL, Washburn SS. Principles of highway engineering and traffic analysis. John Wiley & Sons; 2020 Jul 8.

[14] Nesamani KS, Chu L, McNally MG, Jayakrishnan R. Estimation of vehicular emissions by capturing traffic variations. Atmospheric Environment. 2007 May 1;41(14):2996-3008.

[15] Ahmed KI. Modeling drivers' acceleration and lane changing behavior (Doctoral dissertation, Massachusetts Institute of Technology).

[16] Llopis-Castelló D, García-Segura T, Montalbán-Domingo L, Sanz-Benlloch A, Pellicer E. Influence of pavement structure, traffic, and weather on urban flexible pavement deterioration. Sustainability. 2020 Nov 21;12(22):9717.

[17] Canestrari F, Ingrassia LP. A review of top-down cracking in asphalt pavements: Causes, models, experimental tools and future challenges. Journal of Traffic and Transportation Engineering (English Edition). 2020 Oct 1;7(5):541-72.

[18] Ingrassia LP, Spinelli P, Paoloni G, Canestrari F. Top-down cracking in Italian motorway pavements: A case study. Case Studies in Construction Materials. 2020 Dec 1;13:e00442.

[19] Zhang H, Lepech MD, Keoleian GA, Qian S, Li VC. Dynamic life-cycle modeling of pavement overlay systems: Capturing the impacts of users, construction, and roadway deterioration. Journal of Infrastructure Systems. 2010 Dec;16(4):299-309.
[20] ASCE. Road to infrastructure renewal—A voter's guide.
[21] Hale D, Jagannathan R, Xyntarakis M, Su P, Jiang X, Ma J, Hu J, Krause C. Traffic bottlenecks: Identification and solutions. United States. Federal Highway Administration. Office of Operations Research and Development; 2016 Nov 1.
[22] Lomax T, Schrank D, Eisele B. TTI's 2012 Urban Mobility Report. Texas A&M Transportation Institute. 2012.
[23] Huang YH. Pavement analysis and design. Upper Saddle River, NJ: Pearson Prentice Hall; 2004.
[24] Zaumanis M, Mallick RB. Review of very high-content reclaimed asphalt use in plant-produced pavements: state of the art. International Journal of Pavement Engineering. 2015 Jan 2;16(1):39-55.
[25] Adlinge SS, Gupta AK. Pavement deterioration and its causes. International Journal of Innovative Research and Development. 2013 Apr 1;2(4):437-50.
[26] Salama HK, Chatti K, Lyles RW. Effect of heavy multiple axle trucks on flexible pavement damage using in-service pavement performance data. Journal of transportation engineering. 2006 Oct;132(10):763-70.
[27] Chatti, K., Salama, H., & El Mohtar, C. (2004, March). Effect of heavy trucks with large axle groups on asphalt pavement damage. In Proc., 8th Int. Symp. on Heavy Vehicle Weights and Dimensions.
[28] Burgart S. Gap Trap: A pothole detection and reporting system utilizing mobile devices.
[29] Ben-Edigbe J. Influence of pavement distress on capacity loss and their implications for PCE.
[30] Dora C, Phillips MA, Phillips M, editors. Transport, environment and health. WHO Regional Office Europe; 2000.
[31] Setyawan A, Kusdiantoro I. The effect of pavement condition on vehicle speeds and motor vehicles emissions. Procedia Engineering. 2015 Jan 1;125:424-30.
[32] Brownjohn JM. Structural health monitoring of civil infrastructure. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences. 2007 Feb 15;365(1851):589-622.
[33] Ye XW, Jin T, Yun CB. A review on deep learning-based structural health monitoring of civil infrastructures. Smart Struct Syst. 2019 Nov 1;24(5):567-85.
[34] Onat O, Gul M. Application of artificial neural networks to the prediction of out-of-plane response of infill walls subjected to shake table. Smart Structures and Systems. 2018 Apr 1;21(4):521-35.
[35] Salehi H, Burgueño R. Emerging artificial intelligence methods in structural engineering. Engineering structures. 2018 Sep 15;171:170-89.
[36] Liang X. Image-based post-disaster inspection of reinforced concrete bridge systems using deep learning with Bayesian optimization. Computer-Aided Civil and Infrastructure Engineering. 2019 May;34(5):415-30.
[37] Vodrahalli K, Bhowmik AK. 3D computer vision based on machine learning with deep neural networks: A review. Journal of the Society for Information Display. 2017 Nov;25(11):676-94.

[38] Khodabandehlou H, Pekcan G, Fadali MS. Vibration-based structural condition assessment using convolution neural networks. Structural Control and Health Monitoring. 2019 Feb;26(2):e2308.

[39] Duan Y, Chen Q, Zhang H, Yun CB, Wu S, Zhu Q. CNN-based damage identification method of tied-arch bridge using spatial-spectral information. Smart Struct. Syst. 2019 May 1;23(5):507-20.

[40] Flah M, Nunez I, Ben Chaabene W, Nehdi ML. Machine learning algorithms in civil structural health monitoring: a systematic review. Archives of computational methods in engineering. 2021 Jun;28:2621-43.

[41] Kim B, Cho S. Automated vision-based detection of cracks on concrete surfaces using a deep learning technique. Sensors. 2018 Oct 14;18(10):3452.

[42] Bao Y, Tang Z, Li H, Zhang Y. Computer vision and deep learning–based data anomaly detection method for structural health monitoring. Structural Health Monitoring. 2019 Mar;18(2):401-21.

[43] Dung CV. Autonomous concrete crack detection using deep fully convolutional neural network. Automation in Construction. 2019 Mar 1;99:52-8.

[44] Huang HW, Li QT, Zhang DM. Deep learning based image recognition for crack and leakage defects of metro shield tunnel. Tunnelling and underground space technology. 2018 Jul 1;77:166-76.

[45] Li S, Zhao X, Zhou G. Automatic pixel-level multiple damage detection of concrete structure using fully convolutional network. Computer-Aided Civil and Infrastructure Engineering. 2019 Jul;34(7):616-34.

[46] Gibert X, Patel VM, Chellappa R. Deep multitask learning for railway track inspection. IEEE transactions on intelligent transportation systems. 2016 Jul 7;18(1):153-64.

[47] Lin YZ, Nie ZH, Ma HW. Structural damage detection with automatic feature-extraction through deep learning. Computer-Aided Civil and Infrastructure Engineering. 2017 Dec;32(12):1025-46.

[48] Wei X, Yang Z, Liu Y, Wei D, Jia L, Li Y. Railway track fastener defect detection based on image processing and deep learning techniques: A comparative study. Engineering Applications of Artificial Intelligence. 2019 Apr 1;80:66-81.

[49] Yeum CM, Dyke SJ, Ramirez J. Visual data classification in post-event building reconnaissance. Engineering Structures. 2018 Jan 15;155:16-24.

[50] Xu Y, Wei S, Bao Y, Li H. Automatic seismic damage identification of reinforced concrete columns from images by a region-based deep convolutional neural network. Structural Control and Health Monitoring. 2019 Mar;26(3):e2313.

[51] Kang D, Cha YJ. Autonomous UAVs for structural health monitoring using deep learning and an ultrasonic beacon system with geo-tagging. Computer-Aided Civil and Infrastructure Engineering. 2018 Oct;33(10):885-902.

[52] Cha YJ, Choi W, Büyüköztürk O. Deep learning-based crack damage detection using convolutional neural networks. Computer-Aided Civil and Infrastructure Engineering. 2017 May;32(5):361-78.

[53] Yang X, Li H, Yu Y, Luo X, Huang T, Yang X. Automatic pixel-level crack detection and measurement using fully convolutional network. Computer-Aided Civil and Infrastructure Engineering. 2018 Dec;33(12):1090-109.

[54] Wang N, Zhao Q, Li S, Zhao X, Zhao P. Damage classification for masonry historic structures using convolutional neural networks based on still images. Computer-Aided Civil and Infrastructure Engineering. 2018 Dec;33(12):1073-89.

[55] Ni F, Zhang J, Chen Z. Zernike-moment measurement of thin-crack width in images enabled by dual-scale deep learning. Computer-Aided Civil and Infrastructure Engineering. 2019 May;34(5):367-84.

[56] Zhang X, Rajan D, Story B. Concrete crack detection using context-aware deep semantic segmentation network. Computer-Aided Civil and Infrastructure Engineering. 2019 Nov;34(11):951-71.

[57] Ye XW, Jin T, Chen PY. Structural crack detection using deep learning–based fully convolutional networks. advances in structural engineering. 2019 Dec;22(16):3412-9.

[58] Li R, Yuan Y, Zhang W, Yuan Y. Unified vision-based methodology for simultaneous concrete defect detection and geolocalization. Computer-Aided Civil and Infrastructure Engineering. 2018 Jul;33(7):527-44.

[59] Kim B, Cho S. Image-based concrete crack assessment using mask and region-based convolutional neural network. Structural Control and Health Monitoring. 2019 Aug;26(8):e2381.

[60] Gulgec NS, Takáč M, Pakzad SN. Convolutional neural network approach for robust structural damage detection and localization. Journal of computing in civil engineering. 2019 May 1;33(3):04019005.

[61] Zhao X, Zhang Y, Wang N. Bolt loosening angle detection technology using deep learning. Structural Control and Health Monitoring. 2019 Jan;26(1):e2292.

[62] Zhou C, Chase JG, Rodgers GW. Degradation evaluation of lateral story stiffness using HLA-based deep learning networks. Advanced Engineering Informatics. 2019 Jan 1;39:259-68.

[63] Oh BK, Glisic B, Kim Y, Park HS. Convolutional neural network-based wind-induced response estimation model for tall buildings. Computer-Aided Civil and Infrastructure Engineering. 2019 Oct;34(10):843-58.

[64] Rafiei MH, Adeli H. A novel unsupervised deep learning model for global and local health condition assessment of structures. Engineering Structures. 2018 Feb 1;156:598-607.

[65] Lyu T, Xu C, Chen G, Li Q, Zhao T, Zhao Y. Health state inversion of Jack-up structure based on feature learning of damage information. Engineering Structures. 2019 May 1;186:131-45.

[66] Liang Y, Wu D, Liu G, Li Y, Gao C, Ma ZJ, Wu W. Big data-enabled multiscale serviceability analysis for aging bridges☆. Digital Communications and Networks. 2016 Aug 1;2(3):97-107.

[67] Li S, Guo Y, Xu Y, Li Z. Real-time geometry identification of moving ships by computer vision techniques in bridge area. Smart Structures and Systems. 2019;23(4):359-71.

[68] Zhang Y, Miyamori Y, Mikami S, Saito T. Vibration-based structural state identification by a 1-dimensional convolutional neural network. Computer-Aided Civil and Infrastructure Engineering. 2019 Sep;34(9):822-39.

[69] Gopalakrishnan K, Khaitan SK, Choudhary A, Agrawal A. Deep convolutional neural networks with transfer learning for computer vision-based data-driven pavement distress detection. Construction and building materials. 2017 Dec 30;157:322-30.

[70] Zhang A, Wang KC, Li B, Yang E, Dai X, Peng Y, Fei Y, Liu Y, Li JQ, Chen C. Automated pixel-level pavement crack detection on 3D asphalt surfaces using a deep-learning network. Computer-Aided Civil and Infrastructure Engineering. 2017 Oct;32(10):805-19.

[71] Tong Z, Gao J, Han Z, Wang Z. Recognition of asphalt pavement crack length using deep convolutional neural networks. Road Materials and Pavement Design. 2018 Aug 18;19(6):1334-49.

[72] Zhang A, Wang KC, Fei Y, Liu Y, Tao S, Chen C, Li JQ, Li B. Deep learning–based fully automated pavement crack detection on 3D asphalt surfaces with an improved CrackNet. Journal of Computing in Civil Engineering. 2018 Sep 1;32(5):04018041.

[73] Nhat-Duc H, Nguyen QL, Tran VD. Automatic recognition of asphalt pavement cracks using metaheuristic optimized edge detection algorithms and convolution neural network. Automation in Construction. 2018 Oct 1;94:203-13.

[74] Maeda H, Sekimoto Y, Seto T, Kashiyama T, Omata H. Road damage detection and classification using deep neural networks with smartphone images. Computer-Aided Civil and Infrastructure Engineering. 2018 Dec;33(12):1127-41.

[75] Zhang A, Wang KC, Fei Y, Liu Y, Chen C, Yang G, Li JQ, Yang E, Qiu S. Automated pixel-level pavement crack detection on 3D asphalt surfaces with a recurrent neural network. Computer-Aided Civil and Infrastructure Engineering. 2019 Mar;34(3):213-29.

[76] Alipour M, Harris DK, Miller GR. Robust pixel-level crack detection using deep fully convolutional neural networks. Journal of Computing in Civil Engineering. 2019 Nov 1;33(6):04019040.

[77] Li S, Zhao X. Image-based concrete crack detection using convolutional neural network and exhaustive search technique. Advances in Civil Engineering. 2019 Apr 30;2019.

[78] Zhang K, Cheng HD, Zhang B. Unified approach to pavement crack and sealed crack detection using preclassification based on transfer learning. Journal of Computing in Civil Engineering. 2018 Mar 1;32(2):04018001.

[79] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. InMedical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18 2015 (pp. 234-241). Springer International Publishing.

[80] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.

[81] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. 2014 Sep 4.

[82] Zhang L, Yang F, Zhang YD, Zhu YJ. Road crack detection using deep convolutional neural network. In2016 IEEE international conference on image processing (ICIP) 2016 Sep 25 (pp. 3708-3712). IEEE.

[83] Yang F, Zhang L, Yu S, Prokhorov D, Mei X, Ling H. Feature pyramid and hierarchical boosting network for pavement crack detection. IEEE Transactions on Intelligent Transportation Systems. 2019 Apr 22;21(4):1525-35.

[84] Topics in Machine Learning

[85] Ideas Made to Matter

[86] Tyagi AK, Abraham A, editors. Recurrent Neural Networks: Concepts and Applications.
[87] Li H, Zong J, Nie J, Wu Z, Han H. Pavement crack detection algorithm based on densely connected and deeply supervised network. IEEE Access. 2021 Jan 11;9:11835-42.
[88] Lin TY, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. InProceedings of the IEEE international conference on computer vision 2017 (pp. 2980-2988).

Appendix A. Travel Time Index

Table 9. TTI Data for I65

TMCS	2018	2019	2020	2021	2022
101N04377	4.27	5.27	2.53	1.28	1.72
101N04380	2.54	3.24	2.97	1.35	1.94
101N04379	4.70	5.26	4.09	1.71	3.06
101N04376	2.55	3.43	4.85	2.35	4.20
101-04375	1.84	2.83	6.76	2.25	3.94
101-04378	4.56	7.48	5.52	1.65	3.67
101N04375	2.11	2.55	5.48	1.10	1.81
101N04374	1.71	1.53	3.89	1.30	1.86
101-04374	1.83	2.66	2.92	2.60	2.51
101-04377	4.79	5.25	2.81	2.44	2.61
101N04378	4.75	5.10	1.46	1.20	1.64
101N04381	1.19	1.37	2.66	1.27	1.88
101-04376	2.87	4.04	3.20	1.42	2.60
101-04380	1.84	2.54	4.97	2.51	3.86
101-04379	4.58	6.75	5.10	2.34	4.18
101-04381	1.45	2.63	4.46	2.06	3.91
101+04375	1.65	4.57	2.00	1.08	1.49
101+04376	1.54	4.25	1.63	1.14	1.50
101+04377	1.75	4.21	1.63	1.50	1.67
101+04379	1.66	3.51	2.06	0.93	0.95
101+04380	1.53	3.01	1.98	0.94	1.12
101+04381	1.46	2.37	1.54	0.92	1.00
101+04382	1.27	2.14	1.40	0.94	1.00
101P04375	1.43	3.29	1.23	1.04	1.39
101P04376	1.80	4.10	1.15	1.33	1.58
101P04377	1.70	3.48	1.61	1.09	1.15
101P04378	1.65	2.88	1.69	0.92	0.94
101P04379	1.74	2.41	1.37	0.92	0.99
101P04380	1.47	2.02	1.35	0.92	1.03
101P04381	1.34	1.95	1.20	0.93	1.01
101+04378	1.56	4.48	2.32	0.92	0.95
101+04373	1.86	2.55	2.30	1.23	1.999
101+04374	1.58	1.88	1.77	1.15	1.521
101-04372	1.16	1.20	1.16	1.18	1.24
101-04373	1.44	1.66	1.62	1.16	1.49
101N04372	1.07	0.97	0.93	0.97	1.06
101N04373	1.20	1.28	1.26	1.06	1.22
101P04373	1.62	2.09	1.85	1.22	1.573
101P04374	1.49	1.68	1.46	1.06	1.553

Table 10. Normalized TTI Data for I65

TMCS	2018	2019	2020	2021	2022
101N04377	0.86	0.66	0.27	0.21	0.24
101N04380	0.39	0.35	0.35	0.26	0.31
101N04379	0.98	0.66	0.54	0.47	0.65
101N04376	0.40	0.38	0.67	0.85	1.00
101-04375	0.21	0.29	1.00	0.79	0.92
101-04378	0.94	1.00	0.79	0.43	0.84
101N04375	0.28	0.24	0.78	0.10	0.27
101N04374	0.17	0.09	0.51	0.23	0.28
101-04374	0.21	0.26	0.34	1.00	0.48
101-04377	1.00	0.66	0.32	0.90	0.51
101N04378	0.99	0.63	0.09	0.16	0.21
101N04381	0.03	0.06	0.30	0.21	0.29
101-04376	0.49	0.47	0.39	0.30	0.51
101-04380	0.21	0.24	0.69	0.94	0.90
101-04379	0.94	0.89	0.72	0.84	0.99
101-04381	0.10	0.26	0.61	0.68	0.91
101+04375	0.16	0.55	0.18	0.10	0.17
101+04376	0.13	0.50	0.12	0.13	0.17
101+04377	0.18	0.50	0.12	0.34	0.22
101+04379	0.16	0.39	0.19	0.01	0.00
101+04380	0.12	0.31	0.18	0.01	0.06
101+04381	0.11	0.22	0.11	0.00	0.02
101+04382	0.05	0.18	0.08	0.01	0.02
101P04375	0.10	0.36	0.05	0.07	0.14
101P04376	0.20	0.48	0.04	0.24	0.20
101P04377	0.17	0.39	0.12	0.10	0.06
101P04378	0.16	0.29	0.13	0.00	0.00
101P04379	0.18	0.22	0.08	0.00	0.02
101P04380	0.11	0.16	0.07	0.00	0.03
101P04381	0.07	0.15	0.05	0.01	0.02
101+04378	0.13	0.54	0.24	0.00	0.00
101+04373	0.21	0.24	0.24	0.19	0.32
101+04374	0.14	0.14	0.14	0.14	0.18
101-04372	0.02	0.04	0.04	0.15	0.09
101-04373	0.10	0.11	0.12	0.14	0.17
101N04372	0.00	0.00	0.00	0.03	0.04
101N04373	0.04	0.05	0.06	0.09	0.08
101P04373	0.15	0.17	0.16	0.18	0.19

Table 11. TTI Data for US31

TMCS	2018	2019	2020	2021	2022
101-11461	3.19	2.85	4.16	1.12	2.23
101N11460	3.13	10.01	3.52	1.10	1.64
101-11460	3.41	7.56	2.25	4.06	2.39
101+11461	3.26	4.80	2.76	2.75	2.37
101+11462	2.80	5.42	3.13	3.51	3.45
101P11461	4.42	9.10	2.95	7.40	5.50
101P11462	5.22	11.44	4.05	6.31	5.61

Table 12. Normalized TTI Data for US31

TMCS	2018	2019	2020	2021	2022
101-11461	0.16	0.00	1.00	0.00	0.15
101N11460	0.14	0.83	0.66	0.00	0.00
101-11460	0.25	0.55	0.00	0.47	0.19
101+11461	0.19	0.23	0.27	0.26	0.19
101+11462	0.00	0.30	0.46	0.38	0.46
101P11461	0.67	0.73	0.37	1.00	0.97
101P11462	1.00	1.00	0.94	0.83	1.00

Appendix B. Pavement Quality

Table 13. Pavement Cracking Index for I65

TMCS	2018	2019	2020	2021	2022
101N04377	0.00679	0.00271	0.00267	0.00452	0.00303
101N04380	0.00607	0.03173	0.00309	0.00279	0.01977
101N04379	0.00356	0.00312	0.00335	0.00301	0.00372
101N04376	0.00691	0.00352	0.00515	0.00425	0.00328
101-04375	0.00788	0.00693	0.00199	0.00459	0.00576
101-04378	0.00184	0.01646	0.00362	0.00236	0.00223
101N04375	0.00789	0.00600	0.00528	0.00276	0.00358
101N04374	0.00620	0.00454	0.00373	0.00197	0.00663
101-04374	0.00679	0.00592	0.00373	0.00184	0.00520
101-04377	0.00300	0.00000	0	0	0
101N04378	0.00180	0.02229	0.00356	0.00381	0.00332
101N04381	0.00350	0.00350	0.00373	0.00124	0.00079
101-04376	0.00641	0.00346	0.00477	0.00184	0.00449
101-04380	0.00000	0.00000	0.00000	0.00000	0
101-04379	0.00157	0.00111	0.00038	0.00183	0.00077
101-04381	0.00088	0.00264	0.00356	0.00097	0.00091
101+04375	0.00728	0.00839	0.00365	0.00236	0.00284
101+04376	0.00472	0.00672	0.0036233	0.00516	0.00310
101+04377	0.00614	0.00629	0.00444	0.00351	0.00260
101+04379	0.00082	0.00112	0.0013294	0.00134	0.00078
101+04380	0.00218	0.00216	0.00068	0.00157	0.00057
101+04381	0.00000	0.00588	0.0076355	0.00547	0.00383
101+04382	0.01047	0.00973	0.00404	0.00075	0.00068
101P04375	0.00759	0.00201	0.00339	0.00478	0.00314
101P04376	0.00831	0.00387	0.00450	0.00426	0.00297
101P04377	0.00430	0.00315	0.00280	0.00492	0.00234
101P04378	0.00297	0.00352	0.00368	0.00496	0.00170
101P04379	0.00442	0.00432	0.00587	0.00593	0.00204
101P04380	0.00560	0.00531	0.00309	0.00102	0.00110
101P04381	0.00591	0.00671	0.00339	0.00075	0.00127
101+04378	0.00000	0.00000	0.00000	0.00000	0.00000
101+04373	0.00475	0.00000	0.00414	0.00674	0.00390
101+04374	0.00731	0.00000	0.00427	0.00447	0.00507
101-04372	0.00515	0.00000	0.00209	0.00345	0.00361
101-04373	0.00540	0.00115	0.00208	0.00265	0.00482
101N04372	0.00839	0.00000	0.00628	0.00491	0.00491
101N04373	0.00732	0.00000	0.00696	0.00443	0.00568
101P04373	0.01661	0.00000	0.01322	0.01004	0.04099
101P04374	0.01019	0.00274	0.01047	0.01146	0.00528

Table 14. Normalized Pavement Crack Index for I65

TMCS	2018	2019	2020	2021	2022
101N04377	0.41	0.09	0.20	0.39	0.07
101N04380	0.37	1.00	0.23	0.24	0.48
101N04379	0.21	0.10	0.25	0.26	0.09
101N04376	0.42	0.11	0.39	0.37	0.08
101-04375	0.47	0.22	0.15	0.40	0.14
101-04378	0.11	0.52	0.27	0.21	0.05
101N04375	0.48	0.19	0.40	0.24	0.09
101N04374	0.37	0.14	0.28	0.17	0.16
101-04374	0.41	0.19	0.28	0.16	0.13
101-04377	0.18	0.00	0.00	0.00	0.00
101N04378	0.11	0.70	0.27	0.33	0.08
101N04381	0.21	0.11	0.28	0.11	0.02
101-04376	0.39	0.11	0.36	0.16	0.11
101-04380	0.00	0.00	0.00	0.00	0.00
101-04379	0.09	0.04	0.03	0.16	0.02
101-04381	0.05	0.08	0.27	0.08	0.02
101+04375	0.44	0.26	0.28	0.21	0.07
101+04376	0.28	0.21	0.27	0.45	0.08
101+04377	0.37	0.20	0.34	0.31	0.06
101+04379	0.05	0.04	0.10	0.12	0.02
101+04380	0.13	0.07	0.05	0.14	0.01
101+04381	0.00	0.19	0.58	0.48	0.09
101+04382	0.63	0.31	0.31	0.07	0.02
101P04375	0.46	0.06	0.26	0.42	0.08
101P04376	0.50	0.12	0.34	0.37	0.07
101P04377	0.26	0.10	0.21	0.43	0.06
101P04378	0.18	0.11	0.28	0.43	0.04
101P04379	0.27	0.14	0.44	0.52	0.05
101P04380	0.34	0.17	0.23	0.09	0.03
101P04381	0.36	0.21	0.26	0.07	0.03
101+04378	0.00	0.00	0.00	0.00	0.00
101+04373	0.29	0.00	0.31	0.59	0.10
101+04374	0.44	0.00	0.32	0.39	0.12
101-04372	0.31	0.00	0.16	0.30	0.09
101-04373	0.32	0.04	0.16	0.23	0.12
101N04372	0.51	0.00	0.47	0.43	0.12
101N04373	0.44	0.00	0.53	0.39	0.14
101P04373	1.00	0.00	1.00	0.88	1.00
101P04374	0.61	0.09	0.79	1.00	0.13

Table 15. Pavement Crack Index for US31

TMCS	2018	2019	2020	2021	2022
101-11461	3.19	2.85	4.16	1.12	2.23
101N11460	3.13	10.01	3.52	1.10	1.64
101-11460	3.41	7.56	2.25	4.06	2.39
101+11461	3.26	4.80	2.76	2.75	2.37
101+11462	2.80	5.42	3.13	3.51	3.45
101P11461	4.42	9.10	2.95	7.40	5.50
101P11462	5.22	11.44	4.05	6.31	5.61

Table 16. Normalized Pavement Crack Index for US31

TMCS	2018	2019	2020	2021	2022
101-11461	3.19	2.85	4.16	1.12	2.23
101N11460	3.13	10.01	3.52	1.10	1.64
101-11460	3.41	7.56	2.25	4.06	2.39
101+11461	3.26	4.80	2.76	2.75	2.37
101+11462	2.80	5.42	3.13	3.51	3.45
101P11461	4.42	9.10	2.95	7.40	5.50
101P11462	5.22	11.44	4.05	6.31	5.61