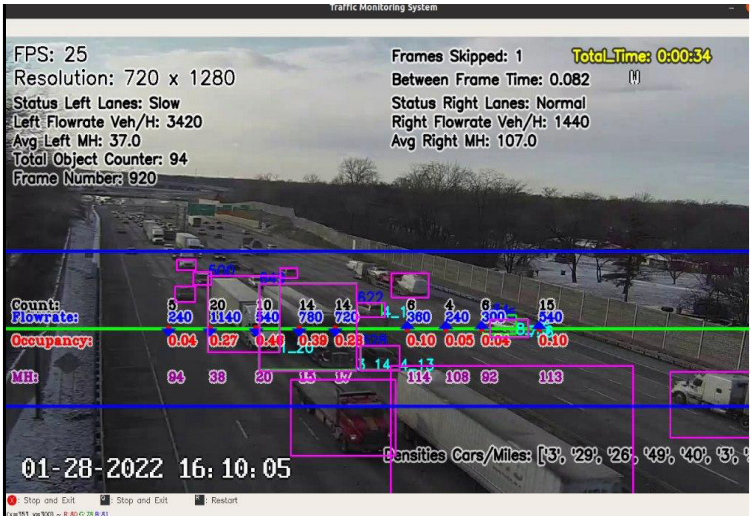
Integration of Lane-Specific Traffic Data Generated from Real-Time CCTV Videos into INDOT's Traffic Management System



Stanley Chien, Lauren Christopher, Yaobin Chen, Mei Qiu, Wei Lin

SPR-4638 • Report Number: FHWA/IN/JTRP-2022/25 • DOI: 10.5703/1288284317400

RECOMMENDED CITATION

Chien, S., Christopher, L., Chen, Y., Qiu, M., & Lin, W. (2022). *Integration of lane-specific traffic data generated from real-time CCTV videos into INDOT's traffic management system* (Joint Transportation Research Program Publication No. FHWA/IN/JTRP-2022/25). West Lafayette, IN: Purdue University. https://doi.org/10.5703/1288284317400

AUTHORS

Stanley Chien, PhD
Elmore Associate Professor of Electrical and Computer Engineering
School of Electrical and Computer Engineering
Indiana University-Purdue University Indianapolis
(317) 274-2760
schien@iupui.edu
Corresponding Author

Lauren Christopher, PhD
Associate Professor of Electrical and Computer Engineering
School of Electrical and Computer Engineering
Indiana University-Purdue University Indianapolis

Yaobin Chen, PhD
Director of TASI
Chancellor's Professor of Electrical and Computer Engineering
School of Electrical and Computer Engineering
Indiana University-Purdue University Indianapolis

Mei Qiu
Graduate Research Assistant
School of Electrical and Computer Engineering
Indiana University-Purdue University Indianapolis

Wei Lin
Graduate Research Assistant
School of Electrical and Computer Engineering
Indiana University-Purdue University Indianapolis

JOINT TRANSPORTATION RESEARCH PROGRAM

The Joint Transportation Research Program serves as a vehicle for INDOT collaboration with higher education institutions and industry in Indiana to facilitate innovation that results in continuous improvement in the planning, design, construction, operation, management and economic efficiency of the Indiana transportation infrastructure. https://engineering.purdue.edu/JTRP/index_html

Published reports of the Joint Transportation Research Program are available at http://docs.lib.purdue.edu/jtrp/.

NOTICE

ACKNOWLEDGMENTS

This work was supported by the Joint Transportation Research Program (JTRP), administered by the Indiana Department of Transportation (INDOT) and Purdue University. The authors would like to thank all members of the INDOT Study Advisory Committee (SAC), including Tim Wells, Project Advisor (PA), Ed Cox, the Business Owner (BO), and Jim Sturdevant, for their guidance and advice throughout the project period. Special thanks go to Dr. Darcy Bullock, Director of JTRP, and his staff at Purdue University for their administrative support and service. The authors also thank SAC members for their technical support and constructive suggestions.

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. FHWA/IN/JTRP-2022/25	**2. Government Accession No.**	**3. Recipient's Catalog No.**
4. Title and Subtitle Integration of Lane-Specific Traffic Data Generated from Real-Time CCTV Videos into INDOT's Traffic Management System		**5. Report Date** September 2022 **6. Performing Organization Code**
7. Author(s) Stanley Chien, Lauren Christopher, Yaobin Chen, Mei Qiu, and Wei Lin		**8. Performing Organization Report No.** FHWA/IN/JTRP-2022/25
9. Performing Organization Name and Address Joint Transportation Research Program Hall for Discovery and Learning Research (DLR), Suite 204 207 S. Martin Jischke Drive West Lafayette, IN 47907		**10. Work Unit No.** **11. Contract or Grant No.** SPR-4638
12. Sponsoring Agency Name and Address Indiana Department of Transportation (SPR) State Office Building 100 North Senate Avenue Indianapolis, IN 46204		**13. Type of Report and Period Covered** Final Report **14. Sponsoring Agency Code**

15. Supplementary Notes

Conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration.

16. Abstract

The Indiana Department of Transportation (INDOT) uses about 600 digital cameras along populated Indiana highways in order to monitor highway traffic conditions. The videos from these cameras are currently observed by human operators looking for traffic conditions and incidents. However, it is time-consuming for the operators to scan through all video data from all the cameras in real-time. The main objective of this research was to develop an automatic and real-time system and implement the system at INDOT to monitor traffic conditions and detect incidents automatically. The Transportation and Autonomous Systems Institute (TASI) of the Purdue School of Engineering and Technology at Indiana University-Purdue University Indianapolis (IUPUI) and the INDOT Traffic Management Center have worked together to research and develop a system that monitors the traffic conditions based on the INDOT CCTV video feeds. The proposed system performs traffic flow estimation, incident detection, and the classification of vehicles involved in an incident. The goal was to develop a system and prepare for future implementation.

The research team designed the new system, including the hardware and software components, the currently existing INDOT CCTV system, the database structure for traffic data extracted from the videos, and a user-friendly web-based server for identifying individual lanes on the highway and showing vehicle flowrates of each lane automatically. The preliminary prototype of some system components was implemented in the 2018–2019 JTRP projects, which provided the feasibility and structure of the automatic traffic status extraction from the video feeds. The 2019– 2021 JTRP project focused on developing and improving many features' functionality and computation speed to make the program run in real-time. The specific work in this 2021–2022 JTRP project is to improve the system further and implement it on INDOT's premises. The system has the following features: vehicle-detection, road boundary detection, lane detection, vehicle count and flowrate detection, traffic condition detection, database development, web-based graphical user interface (GUI), and a hardware specification study. The research team has installed the system on one computer in INDOT for daily road traffic monitoring operations.

17. Key Words lane detection, traffic status monitoring	**18. Distribution Statement** No restrictions. This document is available through the National Technical Information Service, Springfield, VA 22161.

19. Security Classif. (of this report) Unclassified	**20. Security Classif. (of this page)** Unclassified	**21. No. of Pages** 27	**22. Price**

EXECUTIVE SUMMARY

Introduction

The Indiana Department of Transportation (INDOT) uses about 600 digital cameras along highways in populated areas of Indiana to monitor highway traffic conditions. The videos from these cameras are currently observed by human operators looking for traffic conditions and incidents. However, it is time-consuming for the operators to scan through all the video data from all the cameras in real-time. The main objective of this research was to develop and implement an automatic and real-time system at INDOT to monitor traffic conditions and detect incidents automatically.

Findings

The Transportation and Autonomous Systems Institute (TASI) of the Purdue School of Engineering and Technology at Indiana University-Purdue University Indianapolis (IUPUI) and the Traffic Management Center of INDOT have worked together to conduct research and develop a system that monitors traffic conditions based on the INDOT CCTV video feeds. The proposed system performs traffic flow estimation, incident detection, and classification of vehicles involved in an incident. The goal was to develop a system and prepare for future implementation.

The research team designed the new system, including the hardware and software components, the currently existing INDOT CCTV system, the database structure for traffic data extracted from the videos, and a user-friendly web-based server for showing the incident locations automatically.

Implementation

The preliminary prototype of some system components was implemented in the 2018–2019 JTRP projects, which provided the feasibility and structure of the automatic traffic status extraction from the video feeds. The 2019–2021 JTRP project focused on developing and improving the functionality and computation speed of features to make the program run in real-time. The specific work in this 2021–2022 JTRP project is to improve the system further and implement it on INDOT's premises. The system implemented at INDOT has the following features.

1. *Vehicle-Detection*
   Automatic detection of vehicles is an essential part of this project. The artificial intelligence object detection method, YOLOv4, has been used and further improved for vehicle detection in various lighting and traffic conditions. As a result, vehicle detection performance in an automatically selected region of interest can reach over 90% accuracy.

2. *Road Boundary Detection*
   Since the aiming direction of each camera can change, it is impossible to specify, a priori, the road and lane locations on the video image. Therefore, the team used the tracking information of detected moving vehicles to determine road boundaries. This way only the vehicles detected on the road are considered for traffic condition checking. The knowledge of road location helps to eliminate vehicle detection errors.

3. *Lane Detection*
   Since the aiming direction of each camera can change, the lane locations on the video image must also adapt to the camera angle. The lanes are statistically determined by tracing the vehicle motion at the reference lines on the road, since the majority of vehicles remain in the same lane for their duration in the region(s) of interest.

4. *Vehicle Count and Flowrate Detection*
   There are horizontal region(s) on the video frames where the vehicle can be detected most accurately. The vehicles in each lane are counted and the frames have timestamps. Each lane's flow rate (cars/hour) is derived based on vehicle counting and the camera frame timestamps.

5. *Traffic Condition Detection*
   The traffic flow status is derived from the availability of the real-time flow rate on each detected lane. The traffic flow status is categorized as normal, slow, and congested. The conditions for each camera-observed road segment are reported to the central database and displayed on the webpage for traffic operators.

6. *Database Development*
   This project uses a database as a central place to gather and distribute the information generated from all camera videos and the incident detection results derived from sensory information. In addition, the database also provides information for the user interface. The database has been successfully implemented in MySQL.

7. *Web-Based Graphical User Interface (GUI)*
   A web-based graphical user interface (GUI) was developed with input and suggestions from INDOT. The GUI automatically reads data from the database and displays information on the output webpage. The GUI displays four types of information—(1) the location of all installed cameras on the Google Map, (2) all traffic incident locations (shown in red color) on the Google Map, (3) the real-time video of the focused incident location, and (4) all traffic information at the focused incident location. In addition, this GUI supports the selection of a focused traffic incident location among all incident locations.

8. *Hardware Specification Study*
   In addition to developing and testing on Linux PCs with powerful NVidia GPUs, the research team also tailored the system to enable it to run on Linux PC with relatively low-end GPU and CPU (e.g., a $1,500 PC is sufficient to process one camera data in real-time).

The research team has installed the system on one computer at INDOT for daily road traffic monitoring operations.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Indiana Department of Transportation (INDOT) uses about 600 digital cameras along highways in populated areas in Indiana to monitor highway traffic conditions. The videos from these cameras are currently observed by human operators looking for traffic conditions and incidents. However, it is time-consuming for the operators to scan through all video data from all the cameras in real-time. Therefore, the main objective of this research was to develop an automatic and real-time system and implement the system in INDOT to monitor traffic conditions and detect incidents automatically.

The Transportation and Autonomous Systems Institute (TASI) of the Purdue School of Engineering and Technology at Indiana University-Purdue University Indianapolis (IUPUI) and the Traffic Management Center of INDOT have worked together to conduct this research to develop a system that monitors the traffic conditions based on the INDOT CCTV video feeds. The proposed system performs traffic flow estimation, incident detection, and classification of vehicles involved in an incident. The goal was to develop a system and prepare for future implementation.

The research team designed the system that includes the hardware and software components, and using the currently existing INDOT CCTV system, creating the database structure for traffic data extracted from the videos, and providing a user-friendly web-based server for automatically showing the incident locations.

The preliminary prototype of some system components was implemented in the 2018–2019 JTRP projects, which provided the feasibility and structure of the automatic traffic status extraction from the video feeds (Qiu et al., 2021). The 2019–2021 JTRP project focused on developing and improving many features' functionality and computation speed to make the program run in real-time. The specific work in this 2021–2022 JTRP project is to improve the system further and implement it in INDOT.

2. SYSTEM ARCHITECTURE AND DEVELOPMENT

In the 2018–2019 JTRP project *Development of Automated Incident Detection System Using Existing ATMS CCTV*, we developed the system structure. The design was derived from the knowledge that INDOT currently uses hundreds of cameras all over the state of Indiana. Figure 2.1 shows the overall system structure for the implemented system. The entire system had three subsystems. The first subsystem is the image processing portion of the implementation running on the field computers. The second subsystem is the webserver and database server implemented on the central computer. Finally, the third subsystem is the graphical user interface and the web server.

In the 2019–2021 JTRP project *Road Condition Detection and Classification from Existing CCTV Feed*, the system was further developed to make it run real-time and cover more environments and lighting conditions.

In the 2021–2022 JTRP project, *Integration of the Lane-Specific Traffic Data Generated from Real-Time CCTV Videos into INDOT's Traffic Management System*, we further improved the system operability in unsatisfactory camera angles, very low flow rates, and high congested situations. The system had been implemented in INDOT computer for daily operations.

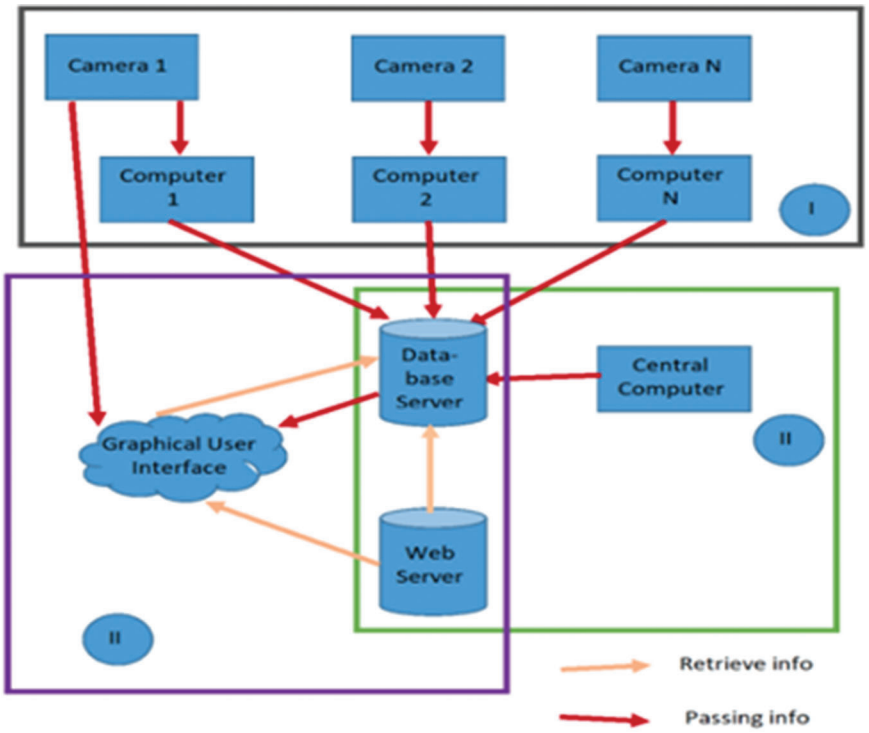


Figure 2.1 Overall system structure.

3. FIELD COMPUTER OPERATION

The incident detection for each camera is a program on a field computer. The input to a computer is a stream of video frames from the cameras. INDOT provides the IP address of over 500 cameras on the road. The control of the camera's pan, tilt, and zoom is manual and not implemented for computer control in the INDOT system.

Since the cameras in the INDOT system have been purchased over many years in multiple batches, the frame per second rate and camera video resolution will vary significantly. The flow rate varies from 6 to 60 fps (frames per second). The resolution varies from 640×480 to 2048×1536. The frame rate setup of the cameras is affected by the network speed at the installation location and method. Therefore, the software was developed to adapt to all possible frame rates and resolutions of camera videos.

The outputs of each computer are real-time road condition data abstracted from video streams, which includes the following.

1. The number of lanes and lane locations
2. The traffic direction of each lane
3. The vehicle flowrate in each lane
4. The traffic conditions (jam, slow, normal)

This output is inserted into a database every 30 sec.

The structure of the incident detection program is depicted in Figure 3.1. The program has two main parts, one is a one-time environment learning, and the other is repeated incident detection.

3.1 The Environment Learning

Environment learning aims to find lanes, lane boundaries, lane directions, and the region where the vehicles can be detected and traced accurately. The environment needs to be automatically relearned after detecting the changes in the camera viewing angle or zoom level. The advantage of this learning method is that it can detect the lanes for any camera image that contains the road. However, relearning the road with a new camera direction takes time. It is assumed that the camera viewing direction and zoom level remain constant most times unless the operator changes it. When the operator changes the camera direction, it means that the operator is looking for the traffic condition, so the automated traffic condition tracking by the computer is not essential.

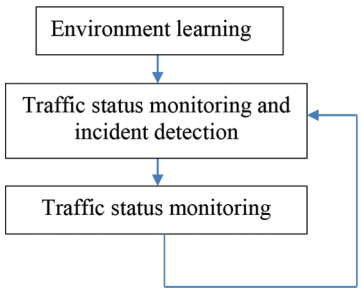


Figure 3.1 The structure of the incident detection program.

The environment learning part has the following modules.

1. Vehicle detection (in the whole frame)
2. Road boundary detection
3. Multiple regions of interest (ROI) determination and reference line detection for each ROI
4. Lane center detection at the reference line in each ROI
5. Lane detection based on vehicle clustering
6. Lane direction detection and lane boundaries determination
7. Traffic status tracking for each lane (no car, few cars, many cars, jam)
8. Camera viewing angle change detection
9. Camera direction detection
10. Database interface

3.1.1 Vehicle Detection

Vehicle detection is the basis for lane detection. It is assumed that most cars move within a highway lane.

3.1.1.1 Introduction to YOLOv4. We explored deep learning algorithms to detect and classify vehicles in video frames. One deep learning algorithm, You Only Look Once (YOLO, https://pjreddie.com/darknet/yolo/), matches what we need. YOLO can be trained to recognize hundreds of types of objects (such as humans, cars, trucks, trains, motorcycles, etc.) in a video frame and directly generates bounding boxes around recognized objects in real time. In this INDOT project, we implemented YOLOv4 to achieve high accuracy in various vehicle detection in real-time with the help of graphic processing units (GPUs). The architecture of YOLOv4 has three main parts, as shown in Figure 3.2, including the backbone, the neck, and the head. The backbone is the architecture to extract features. The neck block aggregates features from different layers to enhance accuracy. The head block also localizes the boxes and does the classification. Compared with others object detectors, YOLOv4 has achieved both higher accuracy and speed.

YOLOv4 was built on Darknet, an open-source neural network framework written in C and CUDA. It is fast, easy to install, and supports CPU and GPU computation. YOLOv4 can also be implemented in the framework of TensorFlow and Pytorch. We built YOLOv4 using Darknet on the Ubuntu system (Ubuntu 20.04.2 LTS).

3.1.1.2 Vehicle detection using YOLOv4. Figure 3.3 shows an example frame of vehicle detection using YOLOv4. YOLOv4 generated the box. The number associated with each box was the vehicle type and the confidence percentage of the vehicle recognition. Figure 3.4 shows another example of YOLOv4-based vehicle detection in lit night conditions. Even with low lighting in the evening, YOLOv4 still achieves high detection accuracy. Some regions with more lighting have much better detection than the darker part. So, in the next section, we define the ROI (regions of interest) and perform the critical counting of vehicles in the ROIs rather than in the whole frame.

3.1.1.3 Vehicle detection accuracy evaluation. The vehicle detection performance was checked at an automatically determined horizontal line by YOLOv4 with 21 videos under different lighting conditions. The result was as follows.

- Day time detection rate is high (>95%).
- The detection rate during the daytime decreased to the 15% range when the images were not stable (vibrating).
- During the daytime, the detection rate decreases if the camera angle is oblique.
- Evening time (before it becomes totally dark) detection rate was 60%–98%.
- In dark-lit conditions, the detection rate was in the 40%–70% range. Thus, in the dark scene, YOLOv4 is not much better than YOLOv3.
- The detection rate was extremely low in the unlit dark condition.

3.1.1.4 The computation speed of YOLOv4 per frame. Table 3.1 shows the execution time of Yolov4 on the

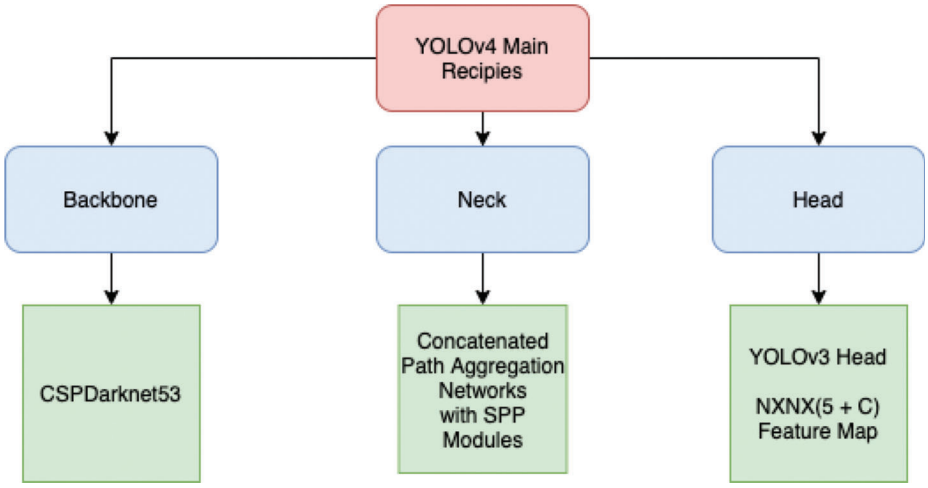


Figure 3.2 YOLOv4 network architecture.

Figure 3.3 Results of YOLOv4 detection (daylight).

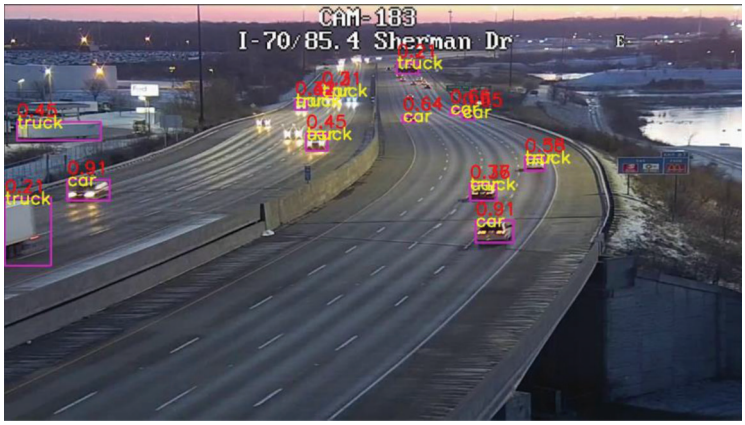


Figure 3.4 Detection results at nighttime (lit by light poles).

INDOT video frame in four computers. Based on that, we can conclude the following.

- GPU is necessary.
- CPU's computation ability is more important for this INDOT project.
- For processing one video stream, a powerful GPU is not essential.

3.1.1.5 The limitations of YOLOv4-based vehicle detection. Since the accuracy of YOLOv4 is essential to the applicability of this system, we studied several issues.

Accuracy improvement—there are still many situations where YOLOv4 does not work well, such as the following.

- Inconsistent vehicle detection in a sequence of video frames.
- Poor vehicle detection in the unlit dark road—since the vehicles cannot be seen on the video images and only the light halo can be seen, YOLOv4 cannot classify the light halo as vehicles.
- Camera lens covered with raindrops.
- False detection—a traffic sign on the road detected as a truck.

Computation cost improvement: although we can achieve real-time operation, more efficient computation is desirable to reduce computing power requirements by using lower-cost computers.

3.1.1.6 Transfer learning improves YOLOv4 vehicle detection under harsh weather and complex road situations. To reduce the YOLOv4 missing detection and false detection under challenging conditions such as harsh weather, low lighting, or low visibility, we used a transfer learning method to improve YOLOv4's detection performance. In the first step, we tested videos with the pre-trained YOLOv4 model and picked out the frames with bad detection results, such as missing vehicles or false classifications. Then we made a dataset of 382 image frames with 3,098 cars and 766 trucks. Compared with the pre-trained YOLOv4, we improved vehicle detection accuracy by 20% on our data (see Table 3.2). The detection accuracy of cars and trucks is both improved. Creating the testing dataset is the same as the training dataset. Table 3.2 shows the testing accuracy and testing time.

TABLE 3.1
The execution time of YOLOv4 based on Darknet for the INDOT video frame on four computers

YOLOv4 Implementation	Model Size	GPU	CPU	YOLOv4 Time Per Frame
Darknet	608 × 608	GeForce RTX 2060 Memory: 15.6 G	Intel@Xeon(R)ES1650 v3@3.50 GHz × 12	≈0.039s
Darknet	608 × 608	NVIDIA Corporation GP102 [GeForce GTX 1080 Ti] Memory: 15.6 G	Intel@Xeon(R)ES1650 v3@3.50 GHz × 12	≈0.031s
Darknet	608 × 608	NVIDIA Corporation Quadro RTX 4000; Memory: 31 G	Intel@Xeon(R) W-2223 CPU@3.60 GHz × 8	≈0.038s
Darknet	608 × 608	NVIDIA Corporation TU104GL [Quadro RTX 5000]; Memory: 31 G	Intel® Core™ i7-9800X CPU @ 3.80 GHz × 16	≈00.35s
Darknet	608 × 608	Intel Xeon(R) W-1250 CPU @ 3.30 GHz × 12 /remmina.mo	Quadro P620/PCIe/SSE2 / NVIDIA Corporation GP107GLM [Quadro P620]	≈0.2s
Darknet	512 × 512	Intel Xeon(R) W-1250 CPU @ 3.30 GHz × 12 /remmina.mo	Quadro P620/PCIe/SSE2 / NVIDIA Corporation GP107GLM [Quadro P620]	≈0.103s
Darknet	416 × 416	Intel Xeon(R) W-1250 CPU @ 3.30 GHz × 12 /remmina.mo	Quadro P620/PCIe/SSE2 / NVIDIA Corporation GP107GLM [Quadro P620]	≈0.086s
Darknet	320 × 320	Intel Xeon(R) W-1250 CPU @ 3.30 GHz × 12 /remmina.mo	Quadro P620/PCIe/SSE2 / NVIDIA Corporation GP107GLM [Quadro P620]	≈0.058s

TABLE 3.2
The testing accuracy of the transfer learning trained model on 50 testing images

Model	Car-AP0.5	Truck-AP0.5	All f1-score
Pre-Trained	0.724	0.605	0.61
After Transfer	0.887	0.707	0.83

Figure 3.5 Multiple separated road segments detection.

3.1.2 Road Boundary Detection

The goal of this project is to monitor road conditions. When there is no entry and exit, all highway vehicles must pass on the highway. Therefore, road traffic conditions can be checked at the easiest object detection point on the image. YOLOv4 detects the vehicles on the input images and associates each vehicle with its position and confidence score. We accumulated N vehicle detections, and then decide the location of the road based on accumulations of the detected vehicle positions on the images. Figure 3.5 shows the road segments: the white pixels represent where the road is.

3.1.3 Reference Line and Regions of Interest (ROI) Determination in Each Road Segment

We find the separate ROIs inside each road segment to improve lane detection. First, we get the relationship of the y-pixel position, total vehicles, and average confidence score inside for each road segment. In Figure 3.6, the blue curve enclosures show three separate road segments. In each road segment, a green rectangle represents the ROI of this road segment. The yellow line in the middle of each ROI represents the baseline of each ROI where vehicles will be counted. The left and right ROI is defined as the left most and right most positions of all the horizontal lines with a y pixel value in the range of the associated ROI.

Our ROI find algorithm sets all the parameters automatically. The ROI detection accuracy is above 86%, with good performance even in very challenging situations, such as during the night or in snowy weather, highly congested traffic, and complex road configurations, as shown in Figure 3.7.

3.1.4 Lane Center at Reference Line

Lane center detection is based on vehicle detection and the assumption that most vehicles are driven within

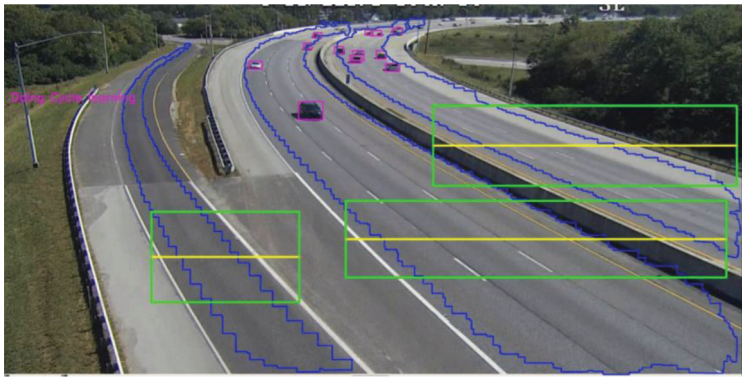


Figure 3.6 An example of the reference line and region of interest.

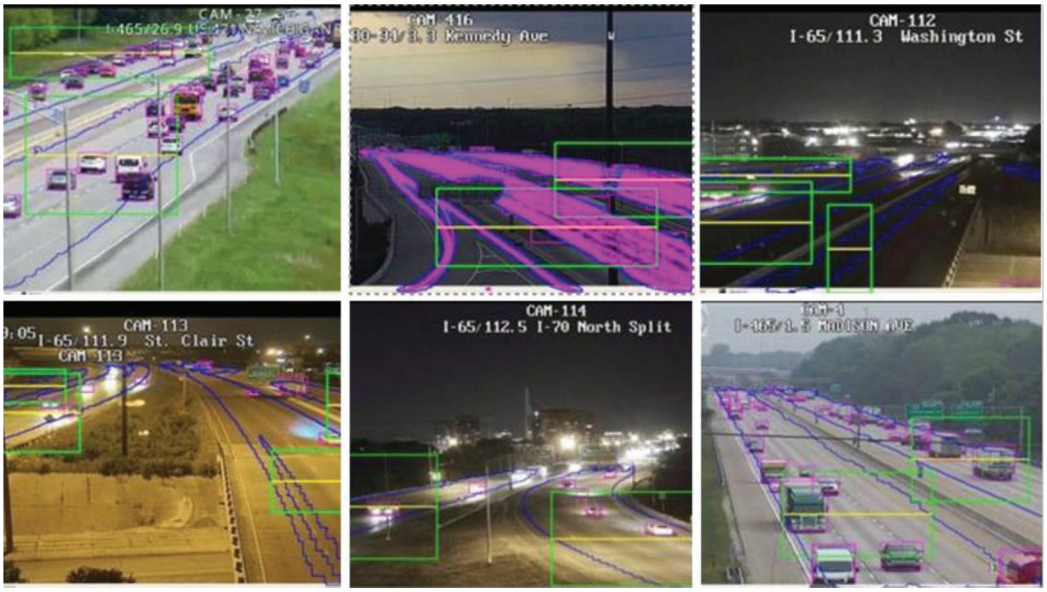


Figure 3.7 ROI learning under various situations.

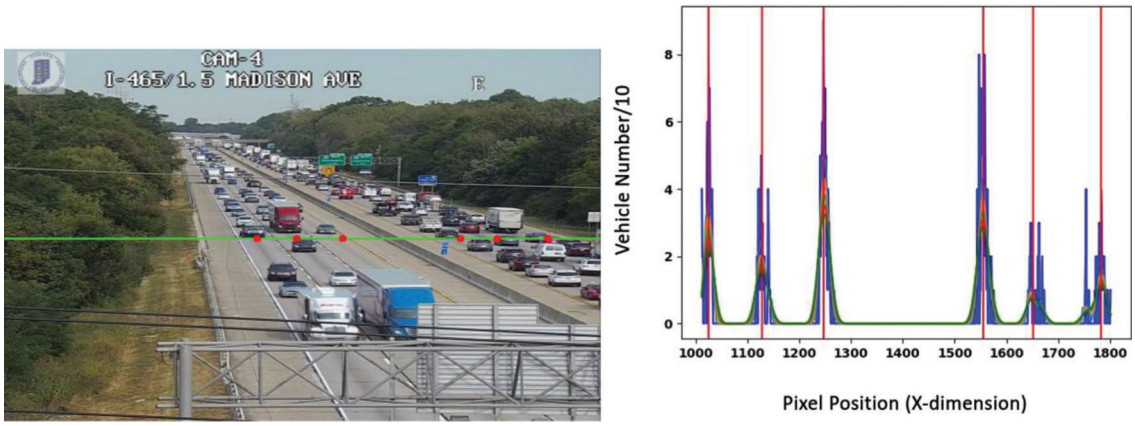


Figure 3.8 Vehicles pass the reference line (right) and corresponding histogram (left).

the lanes. The approach tracks the center location of the vehicles passing a given horizontal line in the image and counts the number of vehicles passing different pixels on the line. The number of vehicles passing the horizontal line is accumulated to a histogram. Assuming that most vehicles are often driven within their lanes, the plot's peaks indicate each lane's center. Figure 3.8 shows a camera view of the road and

generated histogram. Using mean filters to smooth out the close-by local peaks, a lane center is found based on the peaks of the filtered histogram.

In a single ROI, we have proved that this method can achieve good performance under poor weather and complex road conditions, as shown in Figure 3.9.

In the multiple ROI approach, we use the same method to find lane centers at each baseline in each ROI, as shown in Figure 3.10.

3.1.5 Lane Detection Based on Vehicle Clustering

The next step is to find the lanes associated with the detected lane centers on the reference line on the image. Since the lane on the video frame has angles and may not even be straight, the lane location in the ROI needs to be determined to know which vehicle is in which lane. Assuming that most vehicles are driven within a lane most of the time, starting at the lane centers detected in Section 3.1.4, we cluster all adjacent vehicle centers with the vehicle center at the detected lane center at the reference line in ROI. The result is that each cluster contains all vehicles in one lane. Then we use a second-order curve fitting function to find a curve function representing the lane's center line. In Figure 3.11 light blue lane fitting curves are the second-order curves learned by the clustered data in each lane.

3.1.6 Lane Direction Detection and Lane Boundaries in ROI

The lane direction detection is based on the vehicle's motion direction. If the frame rate is not too low (too low means less than ten frames per second), the exact vehicle can be seen at the close-by location of the reference line in several consecutive video images. We can determine the vehicle's moving direction by tracking the vehicle positions on consecutive images. If most vehicles in the same lane at the reference line move in the same direction, that is the lane direction. To count the vehicles for each lane, each YOLOv4

Figure 3.9 Lane center detection in adverse conditions in a single ROI.

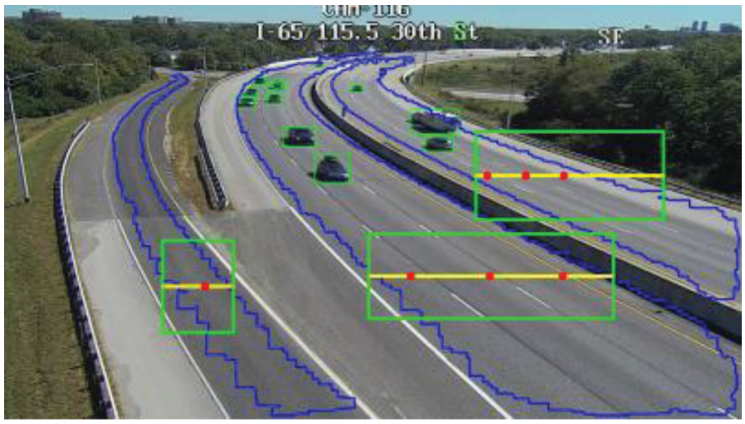


Figure 3.10 Lane center detection in multiple ROIs.

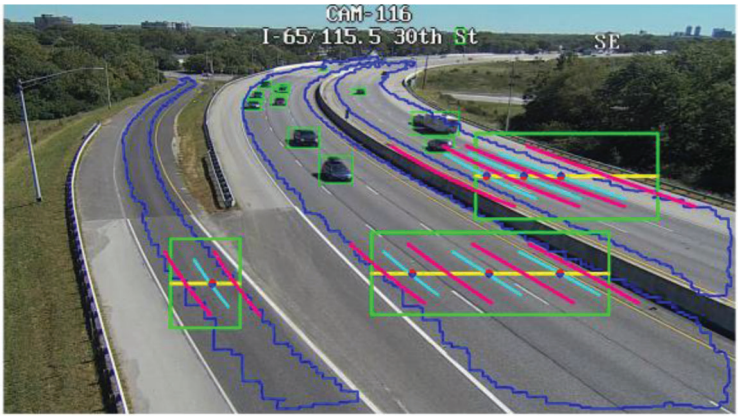


Figure 3.11 Lane learning in each ROI.

detected vehicle needs is assigned to the lane from the boundary determined using the lane centers, starting from the previously determined lane center curve. The boundary between the adjacent lanes is the midpoint between the lane centers of the adjacent lanes in the same direction. If a lane is an edge lane, the lane boundary to the roadside is defined as the average car width from the lane center. The pink lines in Figure 3.11 show the lane boundaries of all detected lanes.

3.1.7 Cycle Learning

This step will keep the environment learning adapting to changes until sufficient vehicles are detected in each lane to ensure the lane finding results are robust and trusted. This process of cycled learning is key when too few vehicles are on the road. Each learning cycle is approximately 1 minute, which can be shorter or longer depending on conditions.

Four traffic states that are useful for determining the learning quality. We define four states—*M*, *L*, *J*, and *N*. *M* means many vehicles seen on a lane so that we can learn lanes well; *L* means a low number of vehicles on a lane, so we need longer time to accumulate vehicles to learn the lane; *N* means no vehicles were seen, so lane learning is not possible; *J* means vehicles are jammed on a lane, so lane direction cannot be determined since there is no vehicle motion information. The status of each lane is set to *N* as the default at the start of the learning cycles. *N* means no vehicles, or less than ten vehicles passed in the time accumulated. *L* means less than 200 vehicles but greater than ten vehicles passed. *M* status is at least 200 vehicles passing the lane center at the reference line. *J* status is determined when many vehicles are in the lane (detected through occupancy) within ROI but have no motion or are very slow-moving. If the traffic status is *M* in both moving directions, or the traffic status is *M* in one direction but and the traffic status in the other direction is *N* for at least five learning cycles, learning is sufficient and stops. Otherwise, learning cycles continue.

3.1.8 Camera Viewing Angle Change Detection

Since the traffic controllers operate the cameras, they change the cameras' viewing angles and zoom levels unpredictably. As the camera viewing angle changed, the learned lane information obsoletes, and the environment needs to be relearned. Therefore, this module detects the camera angle change based on the image background changes. The idea is to check the image difference for the areas not on the road. There are some solutions available for detecting a camera angle change. However, the problem with most of them is that they require heavy computation. We developed a hash function module to overcome these problems and form a reliable method that can work very effectively in this project.

There are multiple benefits of using hashing, but the most important ones include the following.

- The image hash will not change if the aspect ratio of our input image changes (as we disregard the aspect ratio).
- Adjusting the brightness or contrast will not change our hash value, or will only vary it slightly, ensuring that the hashes remain intact.

The description of the ImageHash algorithm is shown in Figure 3.12. The algorithm works straightforwardly while taking the input frame from the camera. To detect camera angle change, we need a background that does not change much, which means the part of the image does not contain the road. After cropping the required area, we convert the RGB channel image into the grayscale image. Discarding color enables us to hash the image faster as we only need to examine only one channel and match the same images even with slightly modified color spaces (because color details have been removed). Hashing is *extremely fast;* it takes less than one millisecond to run on a PC.

3.1.9 Camera Direction Detection

The environment learning can only determine the road and lanes on the frames fed by the camera. We still need to know where the camera is facing to tell which

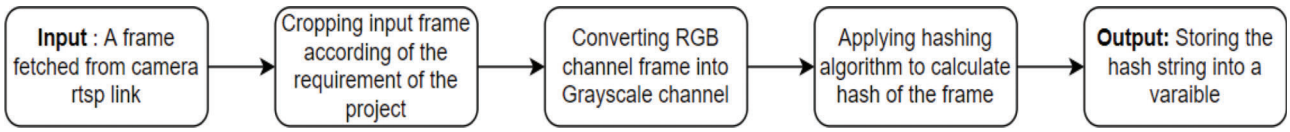


Figure 3.12 Block diagram of the image hash method.

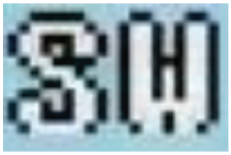


Figure 3.13 Cropped image.

```
Detected output :  SW
Corrected Output :  SW
```

Figure 3.14 Output of OCR.

incidents occur on which side of the road. Tesseract OCR (optical character recognition) is used to recognize and read an image's text containing the direction information overlayed on the image. The text always appears at the top right area right after the intersection name ends, the image is cropped and focused just on direction text. Figure 3.13 shows the cropped region.

The image is then binarized and smoothed using the image processing library. It is then fed to the tesseract OCR method that reads the string from the image. This results in output, as shown in Figure 3.14. The accuracy of the text detection with the known text location is about 98% in over three hundred test cases.

3.1.10 Database Interface

At the end of the environment learning, the following information is sent to the database.

- The camera ID
- The direction that the camera is facing
- Baseline found by the program
- The number of lanes detected
- Timestamp
- For each lane
  - The lane ID
  - Lane annotation
  - Location of the lane center is also stored

3.2 Traffic Status Monitoring and Incident Detection

Traffic status monitoring is conducted in real-time in an infinite loop after environment learning. It has the following modules.

1. Video rate-computer speed synchronization
2. Vehicle detection
3. Filter for unrelated vehicles
4. Vehicle tracking
5. ID mapping
6. Vehicle count
7. Calculate vehicle flow rate
8. Calculate vehicle occupancy on each lane in the ROI
9. Traffic status and incidents detection
10. Camera direction change detection
11. Equipment error detection and reporting
12. Database interface
13. *System diagnostics*

They are explained in the following subsections.

3.2.1 Video Rate-Computer Speed Synchronization

There are many different types of cameras on the INDOT network. The frames per second and frame resolution of these cameras vary significantly. These factors affect vehicle tracking, which is essential for incident detection. To support the real-time vehicle detection and tracking, it is desirable to process a frame and skip from 0 to N frames. This module observes the computer processing time for each frame and the incoming image frames per second and determines how many frames should be skipped for each frame processed, so there will be no backlogged frames. Since the processing time for each frame is affected by the vehicles captured the frames, the number of frames skipped fluctuates. The incident detection program generates good results if the computer can process about 10 frames or above per second.

3.2.2 Vehicle Detection

This vehicle detection is the same as the one described in Section 3.1.1.

3.2.3 Filter Out Unrelated Vehicles

YOLOv4 detects vehicles on the whole frame. It has higher detection errors on objects that are not in the ROI (as we chose the ROI for highest detection performance). These errors cause vehicle counting and vehicle tracking, and lane mapping errors. Therefore, we filter out all detected vehicles not in any ROI and any that are outside the road boundary.

3.2.4 Vehicle Tracking

This step matches the vehicles detected in the current frame to those detected in previous frames. In unfavorable lighting conditions, a vehicle may be detected in one frame and not in the past and future frames, which causes the vehicle lose track. Any tracked

ID becomes invalid when it is not tracked again in more than 30 consecutive frames. The Kalman filter predicts the potential position of all the previous tracked vehicles in the current frame. The Cascade matching method is used to match the new detection and the predictions from Kalman Filter. IOU and Mahalanobis distance matrixes are used to set the threshold in the matching process. The Hungarian assignment method uses the shortest distance for the data association/matching of the detected and predicted vehicle locations.

3.2.4.1 Lane vehicle ID mapping. The ID mapping step determines which lane a new tracked vehicle belongs to. When a vehicle is first tracked in the region of interest, it is assigned a lane vehicle ID, depending on if its bounding box center is within which lanes boundaries. A lane ID indicates that the vehicle belongs to a specific lane. When a tracked vehicle already has a lane ID obtained in previous frames, it will not be assigned with a new lane ID. As the vehicle goes through the ROI, it may change lanes. To prevent a vehicle be counted in multiple lanes, the lane vehicle ID does not change even if the vehicle changes the lanes.

If there is no entry or exit, all vehicles on the road must pass a specific location. Therefore, we only count the vehicles at the reference line we learned in the environment learning, where the vehicle detection and tracking are most accurate.

3.2.5 Vehicle Count

The lane vehicle ID of the first vehicle detected in that lane is assigned 1 at the beginning of the program execution. Each time a newly tracked car is mapped, the lane ID increases by 1 and is modular with the maximum integer. So, the lane vehicle ID is essentially the vehicle count over time.

3.2.6 Calculate Vehicle Flow Rate

Vehicle count is not meaningful for traffic management since it depends on when the count started. Vehicle count per time unit and flow rate are suitable measures of the traffic status. In this module, the flow rate uses the unit of passing vehicles per hour. The flow rate in each lane is estimated based on the lane vehicle ID increment over the last 60 seconds. The vehicle count per hour is updated every 1 minute. The time interval for the flowrate calculation cannot be too long since it will not show the short-term traffic status changes. On the other hand, the time interval for the flow rate calculation cannot be too short since it will fluctuate too much to understand its meaning.

3.2.7 Calculate Vehicle Occupancy on Each Lane in the ROI

Since we cannot correctly figure out the ratio of pixel to meters in various unfixed camera angles, the vehicle speed cannot be detected accurately. Therefore, we use another criterion: occupancy. The occupancy definition is the same as in the environment learning part. Figure 3.15 shows an example of the traffic status report, vehicle counting, flow rate estimation, and occupancy checking at each lane and each road direction in the production status. The current count is the black outlined text above the baseline in every lane, the flowrate is the accumulated data for each lane (per hour) in the blue text above the baseline, and the occupancy is the red text below the baseline which is a measure of the current car(s) in the lane which are occupying the lane.

3.2.8 Traffic Status and Incidents Detection

At present, the traffic status and incidents to be detected are traffic jams. We use occupancy to determine the traffic conditions. The traffic flowrate status is in four categories, high, normal, low, and jam based on the flowrate values. Since a low flowrate can be due to no vehicle or too many vehicles on the road, the estimated vehicle speed, or the recent average number of vehicles in the ROI can be used to separate these two situations. The traffic flowrate status and

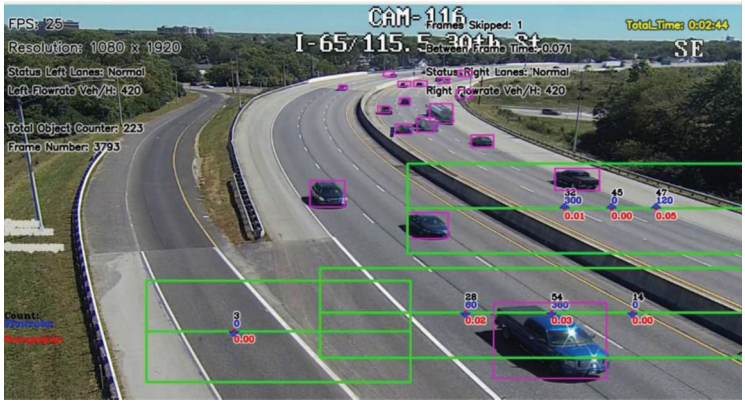


Figure 3.15 Traffic monitoring output with count, flow rate, and occupancy in each lane.

incidents are updated periodically every time the vehicle flowrate is updated and submitted to the database. This traffic flowrate status is then used to alert the traffic operators on the webpage. The thresholds to decide high, normal, and low and jam status will be decided by INDOT.

3.2.9 Camera Direction Change Detection

The method used here is the same as that described in Section 3.1.8.

3.2.10 Equipment Error Detection and Reporting

As the program will be used for any of the 500+ cameras in Indiana, it is essential to automatically detect equipment errors, such as camera malfunction, network disruption, camera not facing any road, etc. Therefore, this module constantly monitors the availability of the video input and image change pattern to decide equipment errors. Then, the error is reported to the database that notifies the traffic operators.

3.2.11 Database Interface

This module reports all lane-based information, such as flow rates, traffic status, and incidents with time-stamps, to the database every 30 seconds. The data are formatted as the SQL query based on the known database table structure. Two data types are sent to the database—traffic and incidence data. For each data, we have two tables to store the information, one for the current traffic status, and the other stores a history of the traffic status.

- Traffic Data
  - This data includes the traffic flow rate (for each lane), speed, density, sum traffic flow rate of each side of the road, the average speed of each side of the time, and the time that the data was generated and sent to the database.
- Incident Data
  - The incident detection module calculates incident data. The result is sent to the database as a text that is either *no vehicle*, *slow*, *normal*, or *jam*. In addition, the creation time of the data is also sent to the database.

3.2.12 System Diagnostics

Since the environmental conditions for the input videos from hundreds of cameras change significantly, many situations can cause the system not to work as expected. Therefore, we have implemented a system diagnostic module that can selectively output the intermediate results of any implemented module. In addition, this diagnostic module helps to pinpoint the reason for the unsatisfactory output. For example, when the vehicles are standstill jammed, this system diagnostic module indicates that the lane direction learning module does not work due to the lack of motion information.

4. DATABASE DEVELOPMENT

The automatic traffic incidents detection needs information of the locations of the cameras, the adjacency graph of all cameras in the road network, the number of lanes of each road, traffic directions of each lane, traffic flow rates for each road/lane, past incidents, and so on. This project uses a database as a central place to gather the information generated from camera images, the incident detection results derived from sensory information in the database and provide user interface information. Figure 4.1 shows the multiple ROI database diagram. Some tables describe the property of the cameras and their installation information. Some tables store the real-time traffic flow information generated by the image processing computers on the field. Some tables store the information of the incident generated from the traffic information. The difference between the two databases is the design of the baseline. In the single ROI version, we only have one baseline since only one ROI exists. The baseline is stored as an attribute in the cam_facing_road table. In the multiple ROI version, we have a baseline for each ROI. Therefore, the baseline is a table instead of an attribute in the cam_facing_road table. The new baseline table is related to the cam_facing_road table and the lane_details table. This means that the database could hold several baselines for each camera facing the road, and for each of the baselines, it could have several lanes related to the baseline. The database has been implemented in MySQL https://www.mysql.com/. It can convert it to PostgreSQL https://www.postgresql.org/ if needed. This database has been hosted on a TSI computer on the INDOT network and kept running for months.

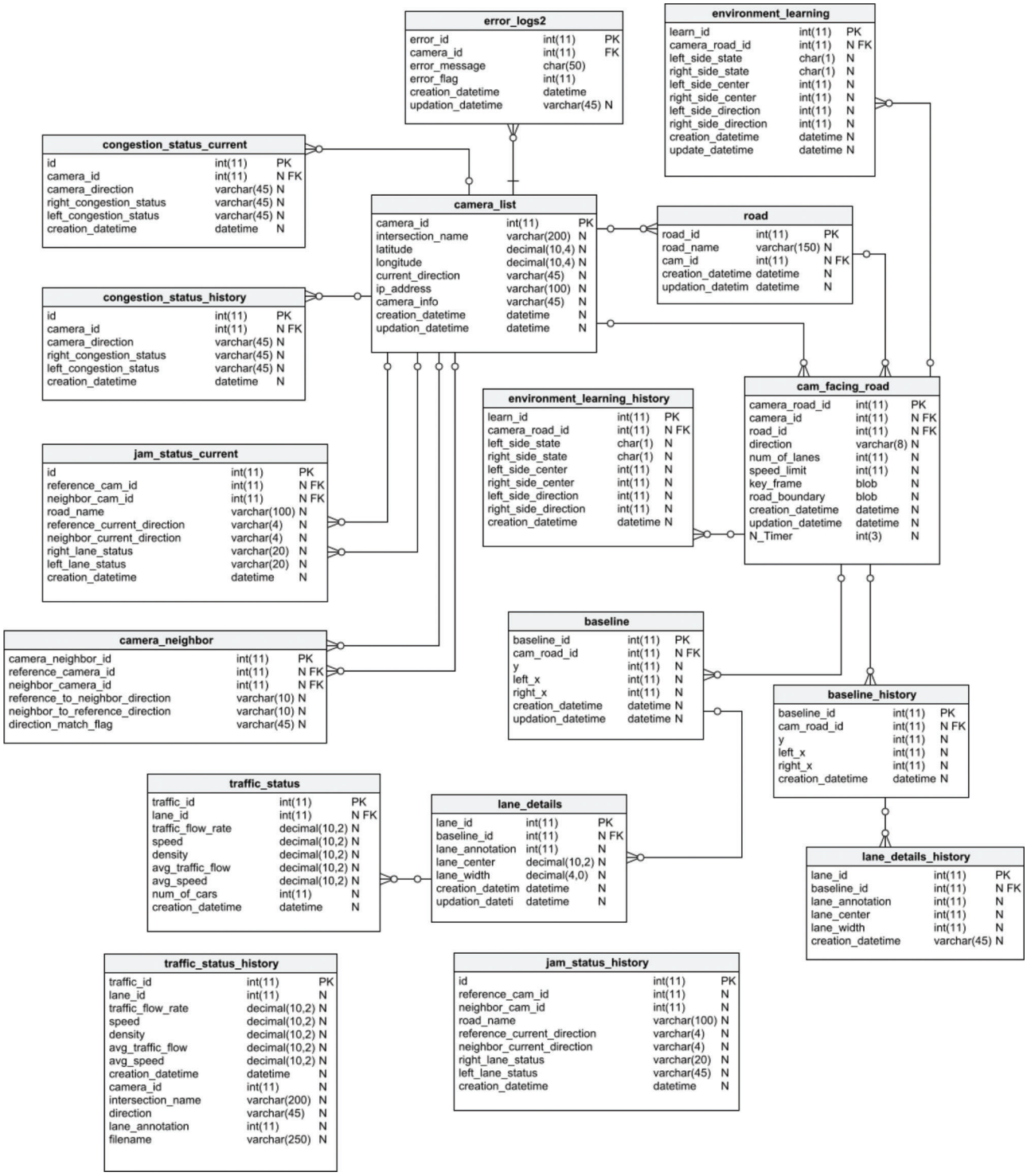


Figure 4.1 ER-diagram of the designed database (multiple ROI).

5. WEB-BASED GRAPHICAL USER INTERFACE

A web-based graphical user interface (GUI) was developed with the requirement input from INDOT. The web-based GUI reads data from the database described in Section 4 and generates the user-interested information in an easy-to-read output display. Figure 5.1 shows a screenshot of the road traffic monitored by camera ID 50 by the field computer.

The environment was learned by the traffic status monitoring and incident detection program on one computer, and the result is also displayed in Figure 5.2. The result data was sent to the database and reflected

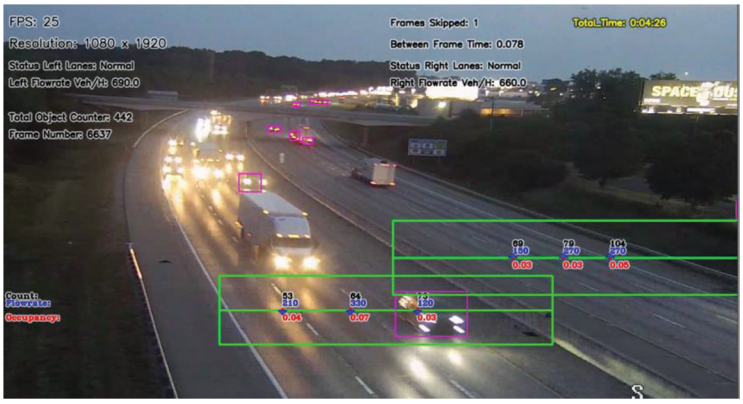


Figure 5.1 The environment was learned by the traffic status monitoring and incident detection program on the field computer.

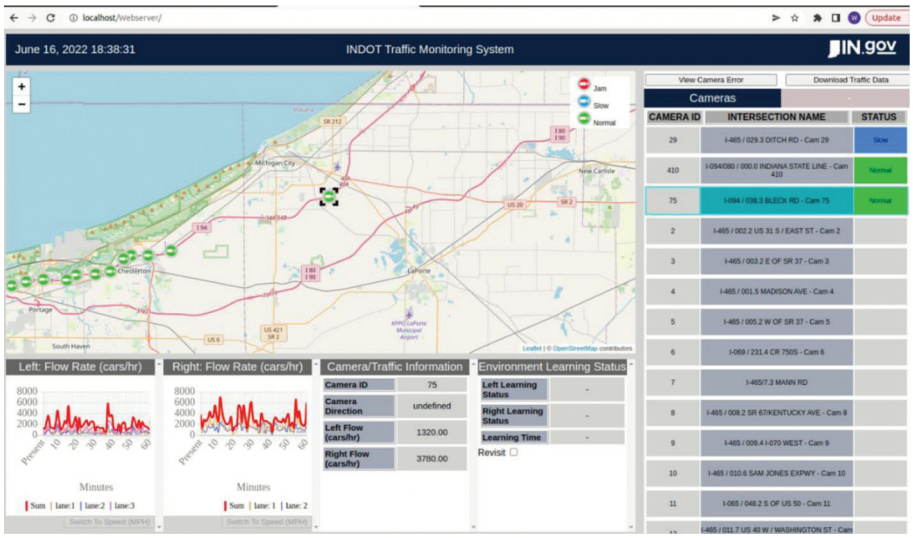


Figure 5.2 Webpage display when a camera is selected. The information displayed is from the information the field computer submitted to the database.

on the webpage (Figure 5.2). Figure 5.2 has four major components. The upper left shows a Google Map with camera locations. The right side shows a table of all cameras. In this table, the jammed roads are shown in red and at the top of the list, followed by slow roads (in blue), normal roads (in green), and roads without new data in the past hour (no color). The cameras in the learning stage are also shown at the top of the list (in bluish green). There is also text describing the status of the road in the color cell at the right column in Figure 5.2. The lower left graphs show the flow rates on the road's left and right sides in the past hour. Finally, the middle bottom shows the road status information observed by the currently focused camera. This window supports the following operations.

When the user selects a specific camera on the table on the right of the screen, the webpage displays the traffic status, and a zoomed-in map with a specific camera selected and highlights the table row to show that the camera is selected. The camera's live feed will also be shown if the Django Server is activated; otherwise, a message will be shown that Django Server is not activated to display the video. If there are values associated with the camera in the past hour, the live feed will display the overlay values from the database (Figure 5.3); otherwise, it will display a text in the live feed that "Database values not available because tracking is not active on this camera" (Figure 5.5). Charts and tables at the bottom of the screen will also display values if they are available (Figure 5.2). Figure 5.4 shows the multiple ROI overlay on the live video.

Clicking on the download report next to the view camera error allows download report with the specified DateTime. When a user enters a DateTime, it will also display the cameras with values in the range of the datetime user specified to assist the user in selecting the camera (Figure 5.6). After the user enters all required fields (camera ID, DateTime), the download button will be enabled. When the user clicks the download button, a report and camera images will be downloaded. Each image is an image of the program when the counting of the program starts. This is included because the camera

Figure 5.3 The traffic status overlays on the live video from that camera.

Figure 5.4 The multiple ROI version of traffic status overlay on the live video.

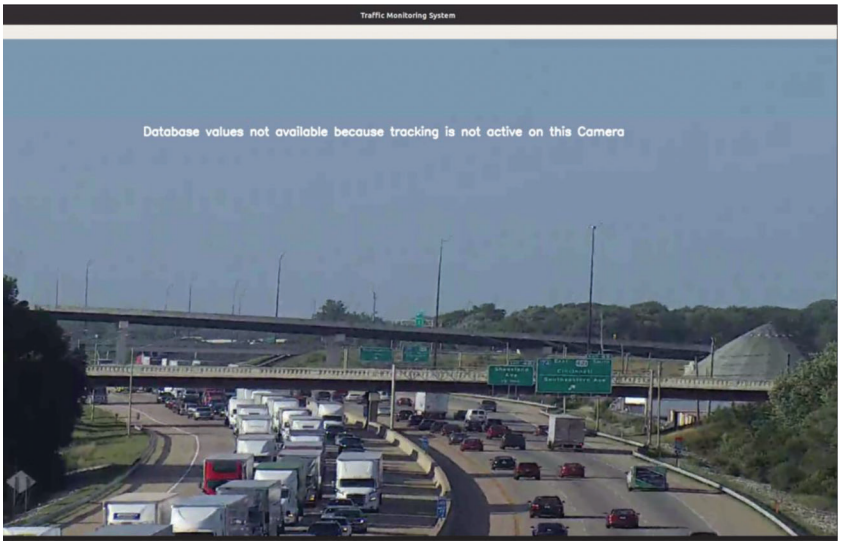


Figure 5.5 Display of camera when there is no value in the database.

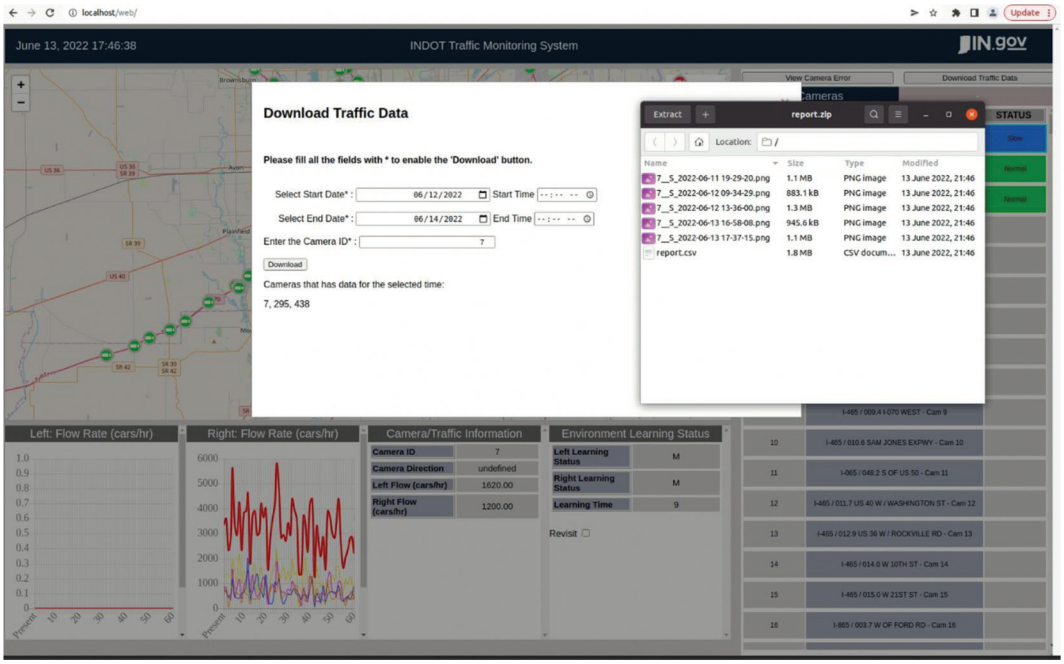


Figure 5.6 Downloading report.

angle could change on this camera, and each program run might be different. The csv report and the images will be packed as a zip file. Figure 5.7 shows an example of the csv report.

Clicking on the *view camera error* tab in the upper right (above the *camera* tab), the table in Figure 5.8 will pop up. The first column is the camera ID, and the second is the associated error message. In the *view camera error* window, there is also a run connection check button under camera diagnostics, this button will send a message to a server PC to run a connection check for all the cameras and report it to the database. When this check is completed, the webpage will display a message for completing the check, and the camera diagnostics table is updated.

In the bottom half of the webpage, where the tables for the camera environment learning status are located, there is a checkbox to use the *revisit* function. This function will mark the selected camera as *revisit* rather than the camera status. This functionality allows the operator to mark the cameras that they might want to check again.

As described above, the downloading feature requires the images to be downloaded along with the csv report in a zip file. To save all images into a server, a Python socket server was developed to receive images from the field computer and save it on the server computer. Next, we need to extract the requested images from the server to download the images. To accomplish this, a Python flask server is developed to handle downloading the images upon request from the webserver. In addition to the image downloading, the flask server handles the *camera connection check* function described above.

	A	B	C	D	E	F	G
1	camera_id	direction	lane_id	lane_annotation	traffic_flow_rate(cars/hr)	datetime	filename
2	50	S	1311	-3	150	2022-07-16 21:34:14	50__S_2022-07-16 21-29-44.png
3	50	S	1312	-2	270	2022-07-16 21:34:14	50__S_2022-07-16 21-29-44.png
4	50	S	1313	-1	270	2022-07-16 21:34:14	50__S_2022-07-16 21-29-44.png
5	50	S	1308	1	210	2022-07-16 21:34:14	50__S_2022-07-16 21-29-44.png
6	50	S	1309	2	330	2022-07-16 21:34:14	50__S_2022-07-16 21-29-44.png
7	50	S	1310	3	120	2022-07-16 21:34:14	50__S_2022-07-16 21-29-44.png
8	50	S	1311	-3	210	2022-07-16 21:33:46	50__S_2022-07-16 21-29-44.png
9	50	S	1312	-2	240	2022-07-16 21:33:46	50__S_2022-07-16 21-29-44.png
10	50	S	1313	-1	660	2022-07-16 21:33:46	50__S_2022-07-16 21-29-44.png
11	50	S	1308	1	90	2022-07-16 21:33:46	50__S_2022-07-16 21-29-44.png
12	50	S	1309	2	270	2022-07-16 21:33:46	50__S_2022-07-16 21-29-44.png
13	50	S	1310	3	150	2022-07-16 21:33:46	50__S_2022-07-16 21-29-44.png
14	50	S	1311	-3	270	2022-07-16 21:33:15	50__S_2022-07-16 21-29-44.png
15	50	S	1312	-2	420	2022-07-16 21:33:15	50__S_2022-07-16 21-29-44.png
16	50	S	1313	-1	570	2022-07-16 21:33:15	50__S_2022-07-16 21-29-44.png
17	50	S	1308	1	90	2022-07-16 21:33:15	50__S_2022-07-16 21-29-44.png
18	50	S	1309	2	240	2022-07-16 21:33:15	50__S_2022-07-16 21-29-44.png
19	50	S	1310	3	360	2022-07-16 21:33:15	50__S_2022-07-16 21-29-44.png
20	50	S	1311	-3	150	2022-07-16 21:32:46	50__S_2022-07-16 21-29-44.png
21	50	S	1312	-2	240	2022-07-16 21:32:46	50__S_2022-07-16 21-29-44.png
22	50	S	1313	-1	300	2022-07-16 21:32:46	50__S_2022-07-16 21-29-44.png
23	50	S	1308	1	210	2022-07-16 21:32:46	50__S_2022-07-16 21-29-44.png
24	50	S	1309	2	150	2022-07-16 21:32:46	50__S_2022-07-16 21-29-44.png
25	50	S	1310	3	270	2022-07-16 21:32:46	50__S_2022-07-16 21-29-44.png
26	50	S	1311	-3	420	2022-07-16 21:32:15	50__S_2022-07-16 21-29-44.png
27	50	S	1312	-2	360	2022-07-16 21:32:15	50__S_2022-07-16 21-29-44.png
28	50	S	1313	-1	540	2022-07-16 21:32:15	50__S_2022-07-16 21-29-44.png
29	50	S	1308	1	150	2022-07-16 21:32:15	50__S_2022-07-16 21-29-44.png
30	50	S	1309	2	300	2022-07-16 21:32:15	50__S_2022-07-16 21-29-44.png
31	50	S	1310	3	180	2022-07-16 21:32:15	50__S_2022-07-16 21-29-44.png
32	50	S	1311	-3	240	2022-07-16 21:31:44	50__S_2022-07-16 21-29-44.png
33	50	S	1312	-2	390	2022-07-16 21:31:44	50__S_2022-07-16 21-29-44.png
34	50	S	1313	-1	210	2022-07-16 21:31:44	50__S_2022-07-16 21-29-44.png
35	50	S	1308	1	360	2022-07-16 21:31:44	50__S_2022-07-16 21-29-44.png
36	50	S	1309	2	240	2022-07-16 21:31:44	50__S_2022-07-16 21-29-44.png
37	50	S	1310	3	210	2022-07-16 21:31:44	50__S_2022-07-16 21-29-44.png
38	50	S	1311	-3	270	2022-07-16 21:31:16	50__S_2022-07-16 21-29-44.png
39	50	S	1312	-2	60	2022-07-16 21:31:16	50__S_2022-07-16 21-29-44.png
40	50	S	1313	-1	300	2022-07-16 21:31:16	50__S_2022-07-16 21-29-44.png
41	50	S	1308	1	180	2022-07-16 21:31:16	50__S_2022-07-16 21-29-44.png
42	50	S	1309	2	120	2022-07-16 21:31:16	50__S_2022-07-16 21-29-44.png
43	50	S	1310	3	480	2022-07-16 21:31:16	50__S_2022-07-16 21-29-44.png
44	50	S	1311	-3	210	2022-07-16 21:30:45	50__S_2022-07-16 21-29-44.png
45	50	S	1312	-2	180	2022-07-16 21:30:45	50__S_2022-07-16 21-29-44.png
46	50	S	1313	-1	90	2022-07-16 21:30:45	50__S_2022-07-16 21-29-44.png
47	50	S	1308	1	90	2022-07-16 21:30:45	50__S_2022-07-16 21-29-44.png
48	50	S	1309	2	180	2022-07-16 21:30:45	50__S_2022-07-16 21-29-44.png
49	50	S	1310	3	270	2022-07-16 21:30:45	50__S_2022-07-16 21-29-44.png

Figure 5.7 Example report table of one camera during a time period.

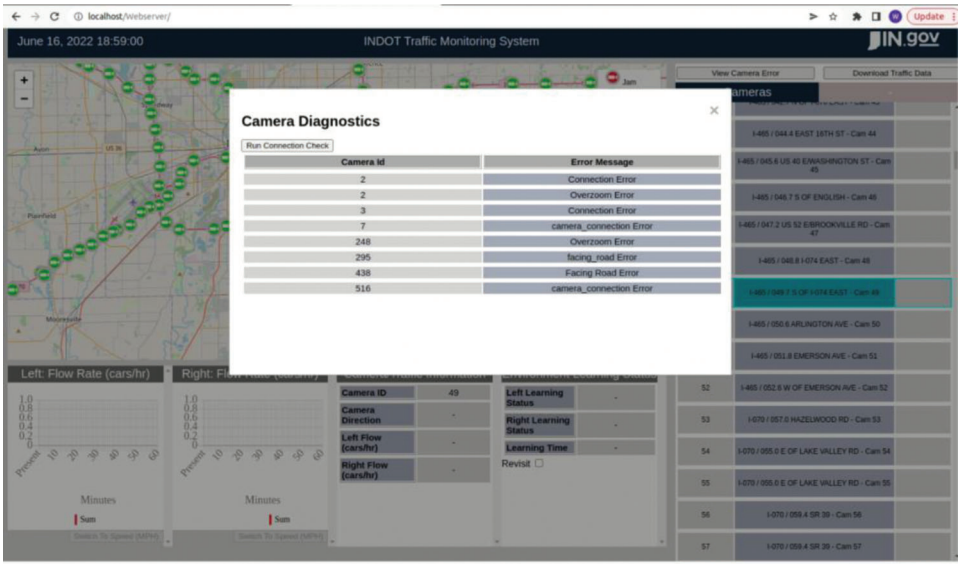


Figure 5.8 Webpage showing the status of all cameras.

6. TEST CASES

The developed traffic status monitoring system is under production tests. Two example test cases are illustrated here.

6.1 Test Case 1

The first test case was at the ramp from I-065 / 055.4 SR 11. Figure 6.1 shows the ramp location on the Google Map and Google Street View (https://goo.gl/maps/9W2N6LAMPJBTGYXv7). Figure 6.2 is a screenshot of the traffic monitoring output.

6.2 Test Case 2

The second test case was to monitor the traffic status in the road construction segment around I-94 105 mile marker. INDOT was interested in the traffic jam conditions daily between 1–7 pm and the ratio between the passenger cars and the trucks. Any of the three consecutive cameras with ID 519, 520 and 522, could be used.

- Camera 175 I-70/105.1 E of SR 9 (IP: 10.7.109.21)
- Camera 176 I-70/107.1 Greenfield Rest Area (IP: 10.7.109.22)
- Camera 177 I-70/107.1 Greenfield Rest Area (IP: 10.7.109.23)
- Camera 519 I-094 / 027.4 E OF SR 49 (IP: 10.6.21.10)
- Camera 520 I-094 / 027.4 E OF SR 49 (IP: 10.6.21.11)
- Camera 522 I-094 / 028.9 TRUCK SCALES (IP: 6.21.13)

Figure 6.3 shows the camera views from cameras 519, 520, and 522, respectively, from top to bottom. The daily test lasted over 5 weeks. Figure 6.4 shows the lane detection in the output of the environment learning. Figure 6.5 is a screenshot of the traffic monitoring. This test also revealed some situations that were not considered before—(a) when the traffic is jammed standstill, the road boundary detection and the lane direction detection in the environment learning part did not work well since they rely on the vehicle motion. (b) There are many trucks on this road. During very slow traffic or traffic jams, adjacent vehicles are close to each other, which causes occlusion of small vehicles and causes vehicle detection errors, and

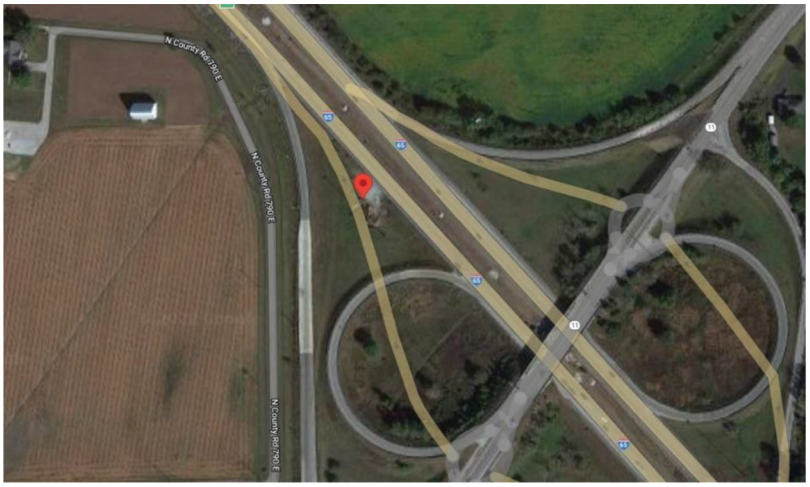


Figure 6.1 The location of the ramp on Google Map and its street view.

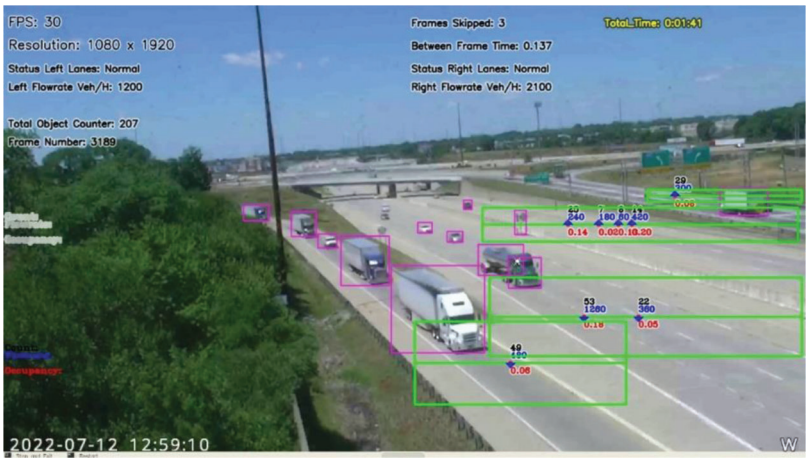


Figure 6.2 A screenshot of the traffic monitoring output (INDOT personnel, personal communication, July 12, 2022).

Figure 6.3 The camera views from cameras 519, 520, and 522 (from top to bottom, respectively).

tracking issues. We chose one of the cameras whose corresponding road segment does not have a standstill traffic jam. (c) During the traffic jam, vehicles were moving on the road shoulders occasionally. Since the road construction caused slow traffic was a constant interest by INDOT traffic operators, the traffic operators changed the camera aiming angle several times a day and sometimes do 180 degrees pan angle change. This demonstrated the need for automatic camera angle change detection and environment relearning. The vehicle counting accuracy per lane is about 90%. The flowrate is at the same accuracy level. Due to the lack of distance reference, the vehicle speed detection was inaccurate.

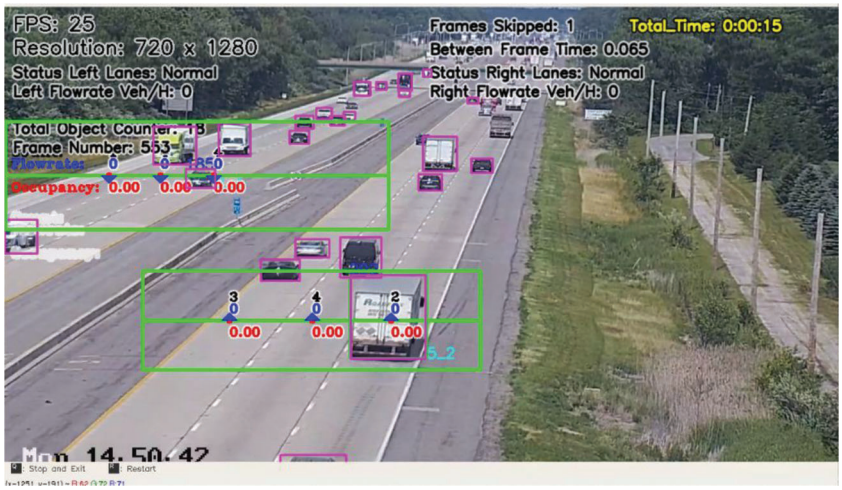


Figure 6.4 The output of the environment learning.

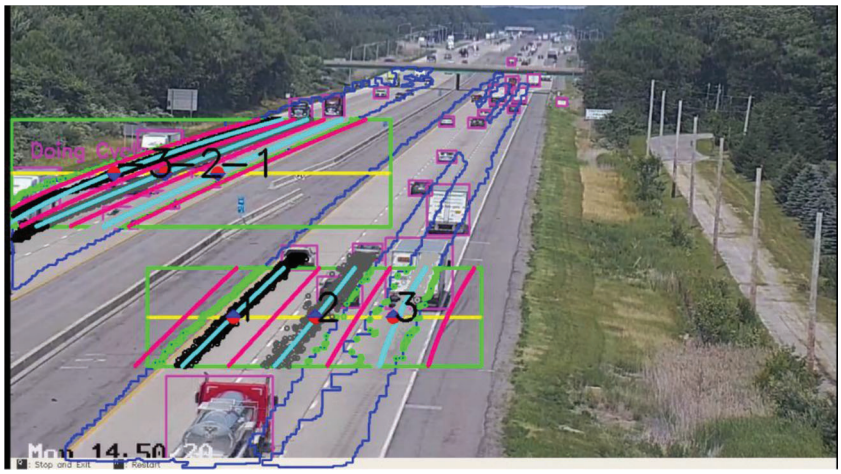


Figure 6.5 A screenshot of the traffic monitoring.

7. CONCLUSIONS AND FUTURE WORK

In this project, we have developed a traffic monitoring system for monitoring traffic conditions using the INDOT CCTV video feeds automatically in real time. More specifically, an AI-based deep learning algorithm, YOLOv4, was used for detecting vehicles on the video frames with transfer learning improvement. This vehicle information automatically learned the lane locations on the video frames using the multiple regions of interest approach. We also developed tracking and mapping algorithms to identify all vehicles to specific lanes. We used this information to count the vehicles and monitor each lane's real-time flow rate changes. The traffic jam can be automatically detected using the detected flow rate and vehicle density information. The system has been successfully tested during daytime, dawn, dusk, and on well-lit roads at night. However, the system does not function well in totally dark conditions as the vehicle headlights generated hallow make vehicle detection difficult. A database was designed as the central place to gather and distribute the information generated from all camera videos. The traffic conditions generated from each camera video feed can be uploaded to the database in real time. The webpage was developed to extract the information from the database and display the immediate past and current traffic status observed from each camera. The system is installed in INDOT.

REFERENCES

Qiu, M., Chien, S., Mulay, A. A., Christopher, L., Ding, X., Chen, Y., Sturdevant, J., & Cox, E. (2021, September). *Intelligent highway lane center identification from surveillance camera video*. Intelligent Transportation Systems Conference, Indianapolis, Indiana.

About the Joint Transportation Research Program (JTRP)

On March 11, 1937, the Indiana Legislature passed an act which authorized the Indiana State Highway Commission to cooperate with and assist Purdue University in developing the best methods of improving and maintaining the highways of the state and the respective counties thereof. That collaborative effort was called the Joint Highway Research Project (JHRP). In 1997 the collaborative venture was renamed as the Joint Transportation Research Program (JTRP) to reflect the state and national efforts to integrate the management and operation of various transportation modes.

The first studies of JHRP were concerned with Test Road No. 1—evaluation of the weathering characteristics of stabilized materials. After World War II, the JHRP program grew substantially and was regularly producing technical reports. Over 1,600 technical reports are now available, published as part of the JHRP and subsequently JTRP collaborative venture between Purdue University and what is now the Indiana Department of Transportation.

Free online access to all reports is provided through a unique collaboration between JTRP and Purdue Libraries. These are available at http://docs.lib.purdue.edu/jtrp.

Further information about JTRP and its current research program is available at http://www.purdue.edu/jtrp.

About This Report

An open access version of this publication is available online. See the URL in the citation below.

Chien, S., Christopher, L., Chen, Y., Qiu M., & Lin, W. (2022). *Integration of lane-specific traffic data generated from real-time CCTV videos into INDOT's traffic management system* (Joint Transportation Research Program Publication No. FHWA/IN/JTRP-2022/25). West Lafayette, IN: Purdue University. https://doi.org/10.5703/1288284317400