Kennesaw State University
DigitalCommons@Kennesaw State University

Master of Science in Computer Science Theses | Department of Computer Science

Fall 10-10-2019

DEVELOPMENT OF SPATIOTEMPORAL CONGESTION PATTERN OBSERVATION MODEL USING HISTORICAL AND NEAR REAL TIME DATA

Betty Kretlow

Follow this and additional works at: https://digitalcommons.kennesaw.edu/cs_etd

 Part of the Artificial Intelligence and Robotics Commons

Recommended Citation

Kretlow, Betty, "DEVELOPMENT OF SPATIOTEMPORAL CONGESTION PATTERN OBSERVATION MODEL USING HISTORICAL AND NEAR REAL TIME DATA" (2019). *Master of Science in Computer Science Theses*. 27.
https://digitalcommons.kennesaw.edu/cs_etd/27

This Thesis is brought to you for free and open access by the Department of Computer Science at DigitalCommons@Kennesaw State University. It has been accepted for inclusion in Master of Science in Computer Science Theses by an authorized administrator of DigitalCommons@Kennesaw State University. For more information, please contact digitalcommons@kennesaw.edu.

DEVELOPMENT OF SPATIOTEMPORAL CONGESTION PATTERN OBSERVATION MODEL USING HISTORICAL AND NEAR REAL TIME DATA

An Abstract of

A Thesis Presented to

The Faculty of the Computer Science Department

by

Betty Kretlow

In Partial Fulfillment

of Requirements for the Degree

Master of Science

Kennesaw State University

December 2017

ABSTRACT

Traffic congestion is not foreign to major metropolitan areas. Congestion in large cities often is associated with dense land developments and continued economic growth. In general, congestion can be classified into two categories: recurring and nonrecurring. Recurring congestion often occurs at certain parts of highway networks, referred to as bottleneck locations. Nonrecurring congestion, on the other hand, can be caused by different reasons, including work zones, special events, accidents, inclement weather, poor signal timing, etc. The work presented here demonstrates an approach to effectively identifying spatiotemporal patterns of traffic congestion at a network level. The Metro Atlanta highway network was used as a case study. Real time traffic data was acquired from the Georgia Department of Transportation (GDOT) Navigator system. For a qualitative analysis, speed data was categorized into three levels: low, median, and high. Cluster analysis was performed with respect to the categorized speed data in the spatiotemporal domain to identify where and when congestion has occurred and for how long, which indicate the severity of congestion. This qualitative analysis was performed by day of week to identify potential variation in congestion over weekdays and weekend. For a quantitative analysis, actual speed data was used to construct daily spatiotemporal maps to reveal congestion patterns at a more granular level, where congestion is represented as “cloud” in the spatiotemporal domain. Future work will be focusing on

deep learning of congestion patterns using Convolutional Long Short Term Memory (ConvLSTM) networks.

DEVELOPMENT OF SPATIOTEMPORAL CONGESTION PATTERN OBSERVATION MODEL USING HISTORICAL AND NEAR REAL TIME DATA

A Thesis Presented to

The Faculty of the Computer Science Department

by

Betty Kretlow

In Partial Fulfillment

of Requirements for the Degree

Master of Science

Advisor: Chih-Cheng Hung

Kennesaw State University

December 2017

Table of Contents

Table of Figures

Chapter 1 Introduction

Traffic congestion is recognized as a major problem in urban areas today (Ma et al., 2017), (Fouladgar, Parchami, Elmasri, & Ghaderi, 2017) and since most people travel routinely every day, it is a problem that is being investigated routinely. Major planning and detailed statistical work seem to be the best methods to address urban traffic congestion problems. The problem will worsen as more vehicles are added to the traffic network every day. The types of problems that are seen on urban interstates include accidents, merging and exit congestion, stalled vehicles, erratic driving habits, i.e. continually switching lanes or braking too hard, and rubbernecking when some event does occur. The need to identify the onset of congestion is crucial to alleviating the problem. When the beginning of a problem can be identified, congestion flow can be predicted, with enough notice to communicate the congestion to travelers and commuters in order to offer opportunities for a change in route. The Georgia Department of Transportation has a network of cameras that record the actual average speed of vehicles in a specific lane at a snapshot of time, the percent occupancy of each lane, and the number of vehicles in each lane. Using the real data which can be obtained every minute from the Georgia Department of Transportation (GDOT) Navigator website, the onset of a problem can be predicted. The data from the GDOT Navigator system is reliable,

timely and available at every location equipped with a camera. The proposed algorithm will preprocess the data, analyze it, identify the onset of the problem and predict where the traffic congestion will flow. When a problem is developing and the influx of heavy traffic becomes evident from the algorithm, steps can be taken to alleviate the potential bottleneck from vehicles stalled in traffic. Atlanta's system of informational overhead signage can be changed electronically to advise travelers of traffic conditions. When travelers are aware that one area is congested for a short time, they can elect to travel a different route. In addition, traffic inflow can be controlled by the system of on ramp access lights. As traffic is forecast to become congested on one part of a system, controlling future access will possibly alleviate the problem. Care has to be taken though, because traffic can become backed up on the access ramp. This problem clearly has many facets. The algorithm can keep up with all facets of the problem and suggest a solution.

The algorithm also identifies recurring patterns on a temporal and spatial basis, i.e. certain days of the week and certain hours of the day are more congested than others and certain parts of the network tend to have more congestion than other areas. On the planning level, benefits can be obtained from identifying recurring patterns on particular days of the week and certain hours of the day and identifying where congestion patterns are most severe. Funding can be prioritized to tackle the bottleneck locales, perhaps to automatically reroute traffic under certain conditions and to design a new traffic configuration. Communicating the recurring patterns to residents and commuters may

also change commuter behavior, i.e. commuters may choose to change work hours and choose to travel during non-peak traffic hours, and thereby reduce traffic congestion during traditionally peak travel hours. The data from GDOT is being downloaded every five minutes now. This data will be analyzed and stored for future predictions. The learning set from a convolutional neural network will be used to predict future traffic results for five minutes up to one hour. These predictions will be mapped to a GIS (Geographic Information System) map of the traffic system of Atlanta to show where the traffic is congested. It is planned that these maps will be updated every minute and can be accessed by Georgia Department of Transportation for their planning as well as by the general public. It is suggested that travelers can make much better decisions about travel routes as well as travel times if they can see the future possibilities. It is also suggested that the department of transportation can use the results of the mapping to turn on and off ramp access metering for better control of inflow and outflow to the system.

Chapter 2 Literature Review

Traffic prediction of congestion is not new or revolutionary. Understanding congestion patterns is an integral part of the attempt to find local solutions for critical bottleneck sections. Ever since the 90's studies have been carried out to quantify and forecast traffic patterns (Davis, Nihan, & Hamed, 1990), (Dougherty, Kirby, & D., 1993). (An, Yang, Wang, Cui, & Cui, 2016) defines types of congestion as recurrent or non-recurrent with the emphasis on identifying recurrent congestion. Recurrent congestion identification could benefit commuters and city planners. Causes of non-recurrent congestion are identified as accidents, breakdowns, traffic control and it is noted that non-recurrent congestion occurs less frequently than recurrent congestion. Causes of recurrent congestion are identified as occurring at a certain site and at a certain time of the day or certain days of the weeks. Causes of recurrent congestion are also identified as high traffic, not enough capacity for the traffic, signal control is inadequate, infrastructure is not sufficient for the traffic. An (An et al., 2016) uses floating cars, vehicles (taxis, cars, and buses) equipped with GPS data and analyzes the data based on location in a grid that is based on the links in the actual traffic network. An (An et al., 2016) chose the floating car because of lower cost, mobility and coverage as compared to traffic cameras, loop detectors and other types of traffic detectors. The location of the floating car is mapped to the network grid. By mining the speed, grid location, and time of the floating car, an algorithm is derived to determine congestion or non-congestion. Then the

congestion locations are classified as recurring or non-recurring. Next, for recurring congestion, the propagation to other grids is identified. Each grid state is determined, and the method can model real-world conditions and can be applied to networks that have no other means of identifying traffic congestion.

Ruder (Ruder, 2016) explains the various methods of Convolutional Neural Network optimization, specifically gradient descent optimization algorithms, and describes the most commonly used algorithms, some pitfalls, and specific behaviors of algorithms. Gradient descent is one of the most frequently used optimizations and there are different implementations to optimize gradient descent. Most of the optimization algorithms are basically black boxes, and the user may or may not know how they actually work. Gradient descent has three variants, batch, mini batch, and stochastic gradient descent, and multiple algorithms, including Adam, AdaMax, and AdaDelta. The optimizer algorithm should be chosen based on the characteristics of the dataset, and the size of the dataset. Accuracy is a tradeoff for expediency in the choice of optimization algorithms.

Davis (Davis et al., 1990) used an approach based on statistical pattern recognition algorithm to forecast the occurrences of traffic bottlenecks. Many of the current algorithms are not anticipatory, and therefore, act too late to prevent the bottleneck. This approach attempts to anticipate the bottleneck, with time to take an action. For example, once a prediction is made of an impending bottleneck, an action can be taken to govern the traffic entering the highway ahead of the soon-to-occur bottleneck.

The algorithm was tested and yielded between 82 to 96 percent correct in time-series prediction, while for pattern recognition of bottlenecks prediction yielded results between 45-89 percent correct. Since the work was still preliminary, Davis (Davis et al., 1990) claimed that the algorithm needed to be refined to other combination of variables for the forecast to be more effective.

Soon after, Dougherty (Dougherty et al., 1993)published an article that uses the neural network in recognizing and predicting traffic congestion. This work is different than Davis (Davis et al., 1990) because of the use of neuro computing in assisting in short term forecasting. Several other neuro network based prediction of traffic speed and congestion followed suit (Lyons, Hounsell, & Williams, 1996), (Huang & Ran, 2003)). Dougherty (Dougherty et al., 1993) uses the neural network to generalize the patterns of traffic flow, to recognize congestion and forecast short term flows of traffic. Huang (Huang & Ran, 2003) proposes a model for predicting traffic flow under varying conditions. The original research focuses on adverse weather from floods to ice, but the authors noted that the same research could be applied to construction or other adverse incidents that occur. Travelers could use the precise speed prediction for planning short trips. Input of traffic speed for this project comes from DOT sensors under the pavement and the weather conditions from the National Oceanic and Atmospheric Administration website. Huang (Huang & Ran, 2003) states the error mean between the ground truth speed and the predicted speed is 4.7 and the standard deviation is 4.46 which makes the results reasonable. Lyons (Lyons et al., 1996) recognizes that knowledge of the status of

the traffic status comes from an ever increasing use of monitoring systems supports traffic management systems. The challenge is in finding an effective way of using the increasing data. Lyons (Lyons et al., 1996) use the back-propagation technique with the multi-layer perceptron architecture to forecast the onset of congestion.

Zhao (Zhao, Xu, Guo, & Gao, 2016) explores Convolutional Neural Networks as a model to more accurately learn or predict unknown relationships in knowledge graphs, as compared to other traditional models. There are many types of knowledge graphs, and the primary goal of each is to understand the interconnections of entities, i.e., the relationships of the entities. Knowledge graphs are often incomplete, with entities or relationships undocumented, which some learning models do not accommodate and fill in the gaps. Convolutional Neural Networks can predict relationships that may not be detailed sufficiently within the knowledge graph. Numerous models have been devised, to understand the complex patterns of the interconnections of entities within a knowledge graph, however, the Convolutional Neural Network appears to outperform these models.

Zhang (Zhang, Zuo, Zhang, & Chen, 2011) produce a map which shows the traffic congestion of the highway system. Based on floating-car data, they explored how to map the congestion patterns to a map of a city highway system.

Garg (Delhi et al., 2014) advocates vehicle type as the basis for traffic congestion detection. In developing countries, conditions vary greatly on the highway systems and travelers use many type of vehicles, such as motorcycles, bicycles, cars, trucks, and

buses. The congestion for the trucks, buses, and cars is different from the congestion for 2 or 3 wheeled vehicles. The study uses smartphone technology to analyze data for road and traffic monitoring. Different traffic scenarios are used to predict traffic congestion levels. The study involves two locations in Delhi, one interior and one outer area. The study obtains 90 percent accuracy in vehicle classification The study advocates use of different scenarios as wrecks, potholes, traffic lights as well as the type of vehicle, the time of day as features for traffic congestion. This approach predicted more than 80 percent accuracy for congestion areas for each vehicle type.

Different mathematical techniques have also been employed to study traffic congestions as well. Pongpaibool (Pongpaibool, Tangamchit, & Noodwong, 2007) used manually tuned fuzzy logic in evaluating road congestions in Thailand. However, the primary focus of the work is to distinguish between the different levels of road congestion using the adaptive-neuro fuzzy algorithm. The work did not lead to field application or forecasting, but the authors did hint that congestion is a complex process and is not only spatial dependent but also temporally dependent.

Lee (Lee, Hong, Jeong, & Lee, 2014) use a more current data collection technology to study congestion patterns. An Intelligent Transportation System (ITS) was used to collect near-real-time traffic data. It is the opinion of the authors that this method of data collection is much more efficient that the floating car method. The data that was collected was then used to model and predict the decongestion times. An algorithm was written to recognize the congestion pattern spatially and temporally and associate the rate

of change of congestion to similar events and correlate the decongestion times. Lee (Lee et al., 2014) are interested in making predictions about changing traffic congestion and advocate the use of historical patterns compared to current congestion to determine when the congestion will dissipate. Lee (Lee et al., 2014) use spatiotemporal chains to describe congestion for branches of the road. They determine the historical pattern that is most similar to the current pattern to predict the end of the congestion. In their validity testing the best-case error was five percent and the average error was 17 percent.

Another study on congestion pattern was done by Wen (Wen, Sun, & Zhang, 2014). This study was done on a network level. However instead of evaluating individual sections within the network, an index called the Traffic Performance Index (TPI) was used as an aggregated number for congestion measurement. Floating car data were utilized to observe the annual TPI for advance forecasting. The work was focused more on large scale annual events based forecasting and does not provide local level analysis.

In a more recent work, He et al. 10) published a paper to assess traffic congestion based on the speed performance index. Their work was done at the network wide level and is therefore significant for the current work. The speed performance index was used in this work as a measure of congestion. The work uses information from loop detectors in the traffic system to identify road segment congestion. The article did not address the need for an algorithm for network level prediction, but it does show the attempt to understand traffic congestion at a roadway network level. He (He, Yan, Liu, & Ma, 2016) note that many cites have serious traffic congestion problems and traffic management

systems are effective approaches to control congestion. In order to manage the system, the agency has to have an accurate understanding of time and location of congestion. He (He et al., 2016) state that traffic congestion is classified in different ways in different locations and propose that speed performance index be used to measure congestion levels. This index uses the average travel speed compared to maximum speed limit. This paper determines the congestion index to measure the congestion. He (He et al., 2016) analyze the seasonal as well as weekday and weekend days' effects on traffic congestion. This result provides planning data for future traffic management.

Hashemi (Hashemi & Abdelghany, 2015) presented an approach that estimates and predicts traffic conditions with decision support capability for addressing congestion in urban settings. The work focuses on the management strategies where the genetic algorithm was used to evaluate the best congestion managing strategies. Conditions that are defined as congested are not explicitly studied, but rather used as conditions to trigger options for traffic management options and decisions. Hashsemi (Hashemi & Abdelghany, 2015) concludes that deficiencies exist in real time traffic management systems and develops a simulation to illustrate a proactive management system to achieve benefits by keeping the deficiencies under a certain level. Deficiencies listed include accuracy of prediction, time to make decision and managed area coverage. Other factors as demand and travelers' behavior are being investigated in the simulation. Min uses a model with location, time and a relationship to other links in the network. Min (Min, Wynter, & Amemiya, 2007) attempt to predict the traffic flow on the connecting links.

Rempe (Rempe, Huber, & Bogenberger, 2016) attempted to study congestion using the floating car (FC) data and clustering analysis to analyze portion of the congestion within the network at particular time. This strategy was employed on the road network of Munich, Germany with success. Rempe (Rempe et al., 2016) pointed out the spatial and temporal features of their methodology and how congestion is related spatially and temporally. However, Rempe (Rempe et al., 2016) cautioned that the method had only been applied to one location and should be applied to other locations that may have differing bottleneck locations and congestion patterns.

Nguyen (Nguyen, Liu, & Chen, 2017) proposed an algorithm which constructs causality trees from congestion and estimate their propagation probabilities based on temporal and spatial information of the congestion. Their algorithm first identifies the spatial and temporal relationships between congested sections and then constructs a sub-tree algorithm to observe recurrent congestion patterns. The dynamic Bayesian network approach was employed to produce probabilities of congestion given particular patterns. Nguyen (Nguyen et al., 2017) try to determine the location of bottlenecks and flaws in the traffic network designs.

Fouladgar (Fouladgar et al., 2017) used a model where the congestion state of each node is predicted by the congestion states of the neighboring links. Fouladgar (Fouladgar et al., 2017) note that no historical data is required to run this model. Their goal is to provide real-time feedback for traffic flow from each node. They analyze

traffic flows over one day, during rush hours, and in light-traffic hours for a single network point. The advantage from this model is that real-time response is achieved.

Tran (Tran, Bourdev, Fergus, Torresani, & Paluri, 2015) use a 3-dimensional convolutional neural network for spatiotemporal learning to develop a generic video descriptor. They conclude that the model is very simple to use and is efficient and compact. This model could be used if videos of traffic are used in the analysis.

Koesdwiady (Koesdwiady, Soua, & Karray, 2016) cited numerous statistics concerning the severity of future traffic problems. They noted that the number of vehicles on the roads will double by 2050 and that building more roads will not alleviate the problem. They propose Intelligent Transportations Systems in order to develop a smart network for travel. Because adverse weather affects 23 percent of road crashes Koesdwiady (Koesdwiady et al., 2016) proposed a system to combine weather and traffic history and use a Deep Belief Network (using Restricted Bolztmann Machines) to enable better decision about traffic flow. They further propose to update travelers with continuous road information. This update would allow better decisions about route planning, time to begin trip and even which days of the week are good travel days. Their conclusion is that the data based prediction system produces better traffic management strategies.

Ma (Ma et al., 2017) use an image input to a convolutional neural network (CNN). The image is constructed from a time-space matrix with speed data in

kilometers. The x-axis of the matrix is time intervals and the y-axis is spatial intervals. This matrix is converted to a grayscale image for input into the CNN. Ma (Ma et al., 2017) detail the construction of the CNN and describe how they tune the CNN parameters. The model is trained to an actual speed and then is trained to three categories of speed: heavy, moderate and free-flow traffic. With these categories, they obtained accuracy ratings above 92 percent on both traffic networks studied. They conclude that this model can learn the spatiotemporal relations and predict accurate results. Ma, et. al., do not specify the accuracy results when they predict the actual speed.

Vallet (Vallet & Sakamoto, 2015) introduced a multi-label cost function and a prediction method for multi-label classifications in deep convolutional neural networks. They (Vallet & Sakamoto, 2015) used a dataset of animation images and achieved 75.1 percent precision and 66.5 percent accuracy. They (Vallet & Sakamoto, 2015) were able to identify more than one animation figure in an image. This work is interesting because it might be adapted to classifying to a range of speeds, for example, 55 plus or minus 5 miles per hour.

Liu (Liu, Wen, Yu, & Yang, 2016) modified the Softmax loss algorithm to increase feature learning by adjusting the angular margin constraint between the classes. They (Liu et al., 2016) demonstrated a geometric interpretation with a simplified example of two weights. They (Liu et al., 2016) concluded by adjusting the margin to be larger, they could make the decision margin larger and that the Large-Margin Softmax loss had

advantages over the Softmax loss function used in current Convolutional Neural Networks.

This work will propose a framework that collects real time data, identifies congestion location, i.e. bottleneck segments and analyzes network-wide spatial and temporal patterns of congestion. The approach taken in this study is similar to that of Rempe (Rempe et al., 2016) where clustering analysis will be done to segregate the recurring congestion segments, but differs in the type of data, and clustering algorithm and analysis. This work also proposes a model using a convolutional neural network similar to that of Ma (Ma et al., 2017) which will be used for input into a real-time application to predict speed of traffic in a short time period.

This article consists of sections that addresses the data acquisition and processing, followed by the data analytics, results and findings and conclusion. A major part of the work consists of data acquisition and therefore, this section will first be discussed.

Chapter 3 Data Acquisition and Processing

Congestion data are typically collected using GPS-equipped mobile vehicles (An et al., 2016)or floating cars (Xu, Yue, & Li, 2013), (Wen et al., 2014). These methods are reliable, but require major effort and do not provide real-time congestion information. Xu (Xu et al., 2013) use floating car data to identify traffic congestion patterns as a spatiotemporal event. More recently, Zhao (Zhao et al., 2016) use Remote Transportation Microwave Sensors data as input to the analysis system to discover the congestion pattern of each node. They match real-time conditions with the congestion patterns to detect traffic congestion.

It was decided to work with Georgia Department of Transportation to obtain data feeds from the 2,284 cameras aimed at the interstates. This method of data collection is cost effective because it uses existing cameras deployed throughout the major freeway system, and therefore ensures a good network coverage around the metro Atlanta area. The data from Georgia Department of Transportation is downloaded from the Navigator website (https://navigator-c2c.dot.ga.gov/). The files that are used in this study are detectorDataResponse.xml (used to collect current data), detectorInventoryResponse.xml (used for identification of the camera location and characteristics), and eventsUpdate.xml (which lists all events that happen at each location). The detectorDataResponse.xml is downloaded every five minutes to capture current data. This data is pre-processed and

stored in the database. The data from the detectorInventoryResponse.xml is downloaded every week and pre-processed for storage in the database. Although the camera information may not change every week, cameras can be added, lanes can be added and other changes can happen. Data inventory information must be current in order to analyze traffic patterns at those locations. It is planned to compare these events with actual congestion data from the eventsUpdate.xml to see the effect of events on the traffic flow.

The data is continuously updated every minute and therefore is near real time when processing is done immediately following download. Because the data is not stored, the detector data response xml files that contain the data are downloaded every five minutes. This file, the detectorDataRresponse.xml, contains data in the following format: date collected, time collected, detector id (id for camera), detector name, detector Status (0, 1, 2, 3), detector lane number (each camera has from one to 8 lanes), lane vehicle count, lane occupancy, and lane vehicle speed.

There are 2,283 cameras (and one test camera) in operation 24 hours a day, 365 days a year spread over segments of the interstate highways in the Atlanta area. These cameras provided a reliable, continuous way of obtaining data. These downloads do not rely on Global Positioning System (GPS) data or floating car data and are therefore deemed more reliable. Originally the detectorDataResponse.xml file was downloaded every five minutes for 66 days which provided sufficient data to analyze spatially and temporally in order to determine historical congestion areas and times.

Data obtained from the Navigator website have to be spatially related to the segment of roadway where the camera is deployed. The location of the cameras is indicated by a dot in Figure 1. The cameras are located on both sides of the highway, so that both directions can be recorded. To spatially relate the data, an inventory file was used which details the geographical location of the cameras. This file includes information such as detector id, detector name, latitude, longitude, route designator (name of highway), linear reference (mile marker on highway), link direction (direction camera is pointing), cross street name, detector type, approach lane name, lane number and last update time as shown in Figure 2. The camera name is used to link the data to a map of the interstate system using ARCGIS system which is a geographic information system for use with map data.

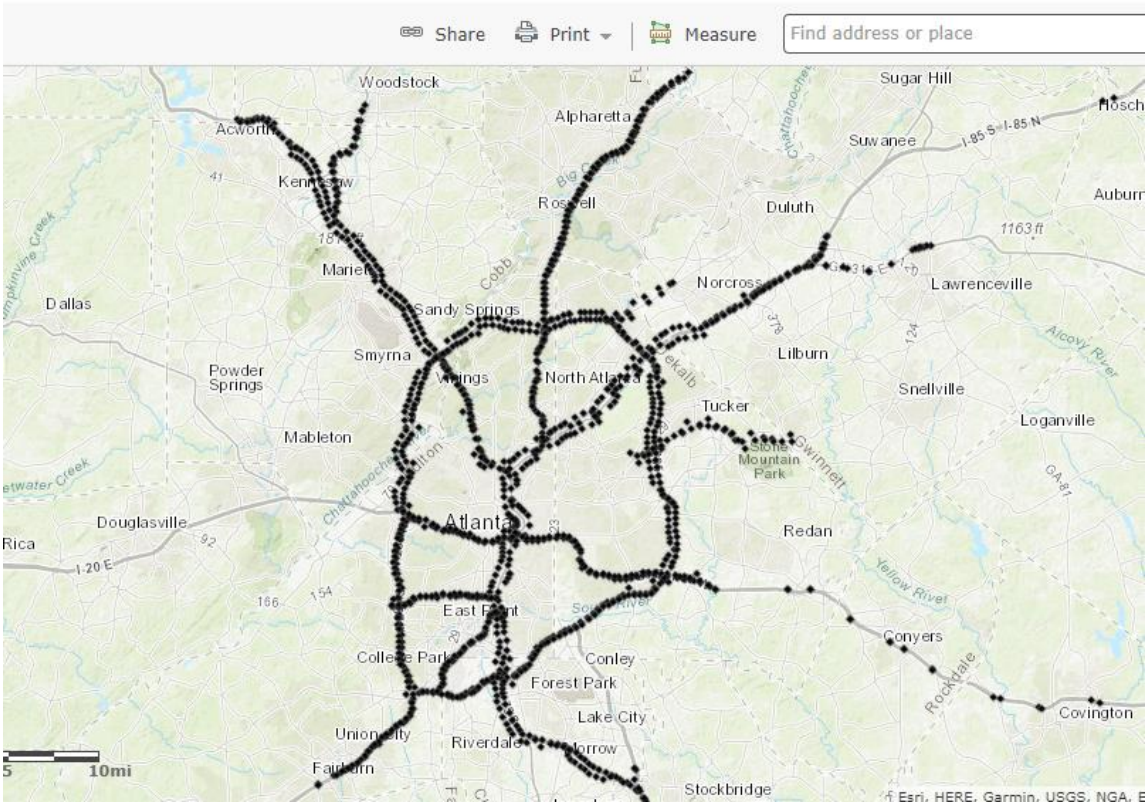

Figure 1: ARCGIS system showing camera locations on Atlanta's network system.

The location of the cameras is indicated by a dot in Figure 1. The cameras are located on both sides of the internet, one recording in one direction and the other camera in the reverse direction.

```
▼<detection-lane xmlns="">
  ▼<detection-lane-item xmlns="">
     <approach-name xmlns="">right_exit_ramp</approach-name>
     <lane-number xmlns="">01000</lane-number>
   </detection-lane-item>
  ▼<detection-lane-item xmlns="">
     <approach-name xmlns="">right_exit_ramp</approach-name>
     <lane-number xmlns="">10000</lane-number>
   </detection-lane-item>
  ▼<detection-lane-item xmlns="">
     <approach-name xmlns="">right_exit_ramp</approach-name>
     <lane-number xmlns="">00100</lane-number>
   </detection-lane-item>
  ▼<detection-lane-item xmlns="">
     <approach-name xmlns="">right_exit_ramp</approach-name>
     <lane-number xmlns="">00010</lane-number>
   </detection-lane-item>
  ▼<detection-lane-item xmlns="">
     <approach-name xmlns="">right_exit_ramp</approach-name>
     <lane-number xmlns="">00001</lane-number>
   </detection-lane-item>
 </detection-lane>
▼<last-update-time xmlns="">
   <date xmlns="">20130214</date>
   <time xmlns="">14400400</time>
   <offset xmlns="">-0500</offset>
 </last-update-time>
</detector>
▼<detector xmlns="">
```

Figure 2: GDOT file for inventory data for each camera.

In Figure 2, data file layout is shown for one of the downloaded file, the inventory file. Note that the lane numbers are in the binary format: in this example, lane numbers are 01000, 10000, 00100, 00010, and 00001 indicating this section of the interstate has five lanes. The approach name indicates the type of lane it is, for example exit ramp or

through lanes. The file also has the cross street name (not shown in the figure) indicating the location of the camera on the interstate. This file is used to determine the total number of lanes on which the camera focuses and is compared to the actual data which may not have all the lanes recorded (if there is no traffic in a lane, the lane is not recorded).

Initially, the downloaded data were parsed into a database, associating the data with its geographical locations. Preprocessing the data is necessary to account for missing data, cameras out of service and added camera locations. The algorithm for dealing with the missing data is as follows: Case 1: Data missing for one of the camera lanes, use average of other lanes' speed. If a camera records no data for a lane, data will be missing for that lane. Case 2: All lanes are present but some data is missing, i.e. have occupancy and vehicle count but missing speed, algorithm uses average of other lanes' speed. Case 3: Missing all camera data (camera is not working, status is 0, 2 or 3) algorithm takes average of data from cameras on either side of the missing camera (based on map GDOT name) or uses the milepost where camera is located to determine which camera to use (if one is close and other is distance algorithm uses values from closer one). If distance < 1320 feet (1/4 mile) algorithm uses closer camera or interpolates the values from the two cameras. The algorithm to account for missing data is one of the most important parts of preprocessing the information. The missing data are tracked so that any future questions could be answered about what is changed to account for missing data.

If the data is not pre-processed correctly, it will not give good results. An example of poorly processed data is shown in the graph for Nov 26, 2016 as shown in Figure 3. Missing speed was changed to speed limit rather than average of other lanes or average of two nearest cameras. This graph shows a column of dark all the same shade indicating that a speed of the 110 kilometers is used for the missing speeds. This picture clearly demonstrates the need for clean data to ensure the algorithm to forecast traffic flow can work properly.

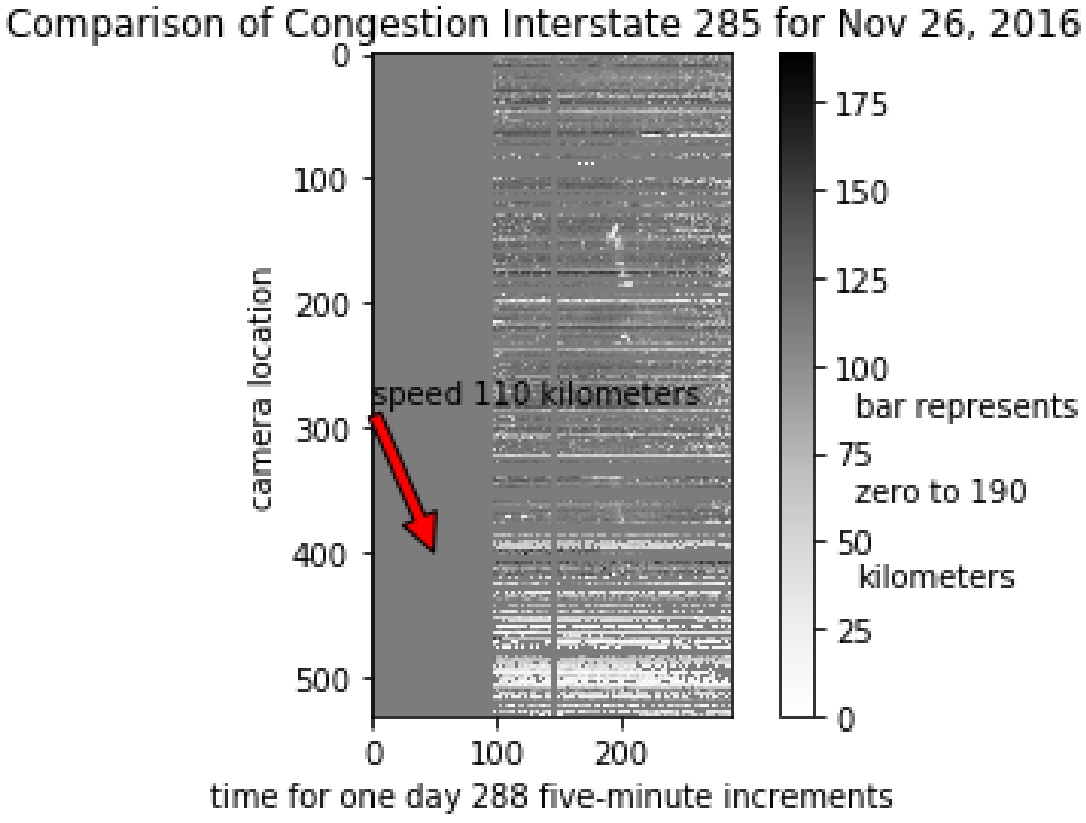

Figure 3: Example of a link that has not been properly processed.

In Figure 3 the y-axis represents the camera location of the 532 cameras on Interstate 285 and the x-axis represents the five-minute increment of the data collection. The pixel at (x, y) is the speed recorded at location x and time y. The bar on the right hand side shows the scale of colors from white (zero kilometers) to black (190 kilometers). Thus areas of lighter shades indicate congestion.

In order to visualize the data that has been collected, statistical properties such as the mean and sample variance were computed for each camera, each lane, and each category (vehicle count, occupancy, and speed) for each time slice (five minutes) for each day. A pseudo-code of the statistical algorithm is provided as follows:

```
Loop:

    For camera id, convert date to day of week
      Loop by day of week
      Loop for number of lanes for camera
      For each lane

        write vehicle count, occupancy, speed
        reshape array to 2-dimenstional array
For range 0 to 7 (number of weekdays)
    Slice array by day of week
```

Slice resulting array >= beginning time and < ending time

Use array to calculate mean and sample variance

Slice resulting array by lane

Write results to database.

The results from the data analysis will be compared to the actual incident file to determine the cause of the traffic slowdowns. The incident file contains all of the incidents that have occurred from planned events to accidents. It is extremely useful to correlate the traffic flow dynamics with what actually happened at the site.

The data incident file is very large and cumulative. The kinds of events described in this file can adversely affect traffic in surrounding areas and are necessary in analyzing traffic patterns. The GDOT data download has these incidents recorded which is an advantage for determining the cause of congestion. If no record of an incident is readily available, there is no way to connect congestion levels with actual events.

Chapter 4 DATA ANALYTICS

The first analysis of the collected data allowed the visualization of the traffic flow. Congestion was observed and defined for average speeds of less than 10, 20, 30, and 40 miles per hours and sample variances of 5, 10, 15, and 20 MPH. The data that corresponds to the average speed ranges and variances were flagged and stored. If average speed <= cutoff and sample variance <= ½ of the cutoff, then a flag was set at that location.

Each flag indicates one congestion event. The frequencies of the flags were determined for each individual camera and time. These flagged data were then written to a file for further processing. For spatial comparisons of locations, the algorithm selects two individual cameras as shown:

```
Loop
    Slice array >= beginning time and < ending time
        Loop (by camera with more lanes)
            If the date and time for the cameras match
                Subtract data from second camera from first camera
                    For each category (vehicle count, occupancy, speed)
```

Plot the data

This algorithm computes the mean and sample variance of the differences for each camera every weekday for each lane. The mean and sample variance across all the lanes for each camera, date, and five-minute interval were also calculated.

In addition to the temporal differences for a camera, spatial differences were computer for selected cameras.

Cameras		timeOne	timeTwo	Lane	Veh count (mean/variance)	
2796	2797	16400000	16450000	100000	0.6101695	10.0005845
2796	2797	16400000	16450000	000010	0.6271186	6.6861485
2796	2797	16400000	16450000	000001	-1.6440678	2.7849211
2796	2797	16400000	16450000	010000	1.2542373	8.8480421
2796	2797	16400000	16450000	001000	-0.3220339	4.3945061
2796	2797	16400000	16450000	000100	0.6440678	5.8883694
2796	2797	16450000	16500000	100000	0.3582090	9.8697422
2796	2797	16450000	16500000	000010	0.0000000	5.5757576
2796	2797	16450000	16500000	000001	-1.5373134	4.3432836
2796	2797	16450000	16500000	010000	1.0746269	8.4640434
2796	2797	16450000	16500000	001000	-0.6119403	8.0895522
2796	2797	16450000	16500000	000100	0.9253731	7.0398010
2796	2797	16500000	16550000	100000	0.3606557	7.8010929
2796	2797	16500000	16550000	000010	0.8032787	5.3939891
2796	2797	16500000	16550000	000001	-1.8688525	5.1491803
2796	2797	16500000	16550000	010000	1.0491803	7.3808743
2796	2797	16500000	16550000	001000	-0.2950820	5.8114754
2796	2797	16500000	16550000	000100	0.3770492	7.3721311
2796	2797	16550000	17000000	100000	0.4482759	10.9885057
2796	2797	16550000	17000000	000010	0.3275862	7.8030853

Figure 4: Spatial difference for two cameras.

These plots of distributions allow visualization of the data that has been collected and lead to further analysis. Since any two cameras can be tested, this gives a good

spatial view and comparison for the camera of interests. In Figure 4, the two cameras selected are close to each other and the small differences in averages of vehicle counts are expected. If the differences were large, an event has caused a stoppage of traffic. Next the temporal differences for one camera was calculated. The temporal difference shows how the mean and variance change over time. Here, the algorithm was written to allow the selection of different time intervals for comparison, i.e. five minute intervals to 12 hour intervals. The mean and sample variance of the difference of two intervals for each of the three categories vehicle count, occupancy, and average speed for each lane were obtained.

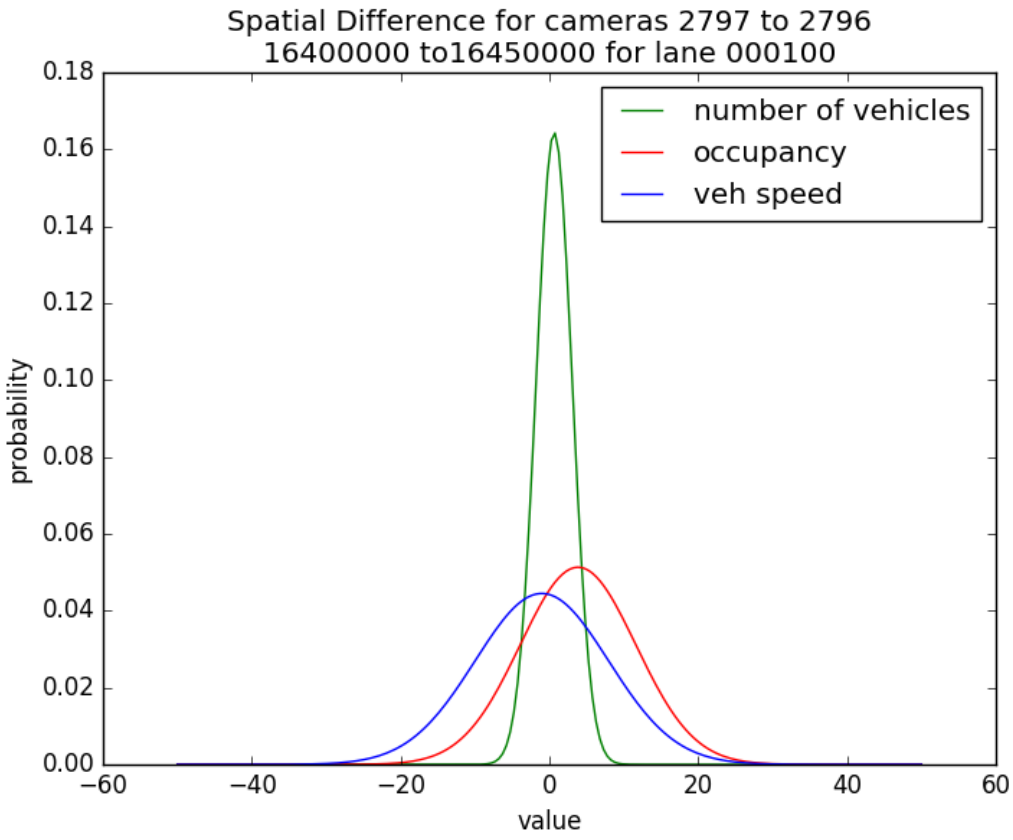

Figure 5: Plot of distributions for time differences for two cameras.

In Figure 5, the number of vehicles for the two cameras chosen were close to the average while the occupancy percentage and speed varied more. In this Figure 5, time is in 24 hour time and lane numbers are binary, for example, 6 lanes are represented as 100000, 010000, 001000, 000100, 000010, and 000001 in the downloaded file.

This data can be analyzed spatially and temporally. In the temporal analysis, data from the same camera were subtracted for two different times, i.e. the beginning time from the data for the ending time for a selected interval. The mean and sample variance

were found by using arrays produced by the subtraction. The program was written to allow selection of the length of time segment to check, for example, one hour, thirty minutes, etc. on a sliding scale beginning with the five-minute period after midnight to the five-minute period before the next midnight. With these layers of time segments, the flow of congestion from one-time period to the next can be seen. Thus the congestion as it flows through the time dimension can be visualized.

For the spatial analysis, cameras may not have the same number of lanes and could not be compared lane by lane. However, statistics across lane data was used for the comparison purposes. For the speed data, the average speed of all the lanes is used; for occupancy data, the average rate is used, and for vehicle count, the sum is used. The selection of any two cameras for the spatial comparison was allowed in the algorithm.

In this study, the severity of a congestion segment was identified by the average speed of the segment. The speed of 20 miles per hour was chosen to indicate severe congestions while the speed of 40 mph indicates moderate congestions. The DBSCAN algorithm (Ester, Kriegel, Sander, & Xu, 1996) is used to perform the clustering analysis. This analysis groups the congestion data to a few different clusters. DBScan does not require subjective predetermination of number of clusters for the analysis and is therefore suitable for this application. The benefit of performing the proposed clustering analysis is that bottlenecks and their immediate influences can be identified based on the cluster numbering assigned to the individual cameras. In addition, these clusters also indicate areas of the highway system with recurrent congestion. These cluster locations were

mapped to GDOT system using ARCGIS. Ester (Ester et al., 1996) concludes that the DBScan algorithm is outperforms CLARANS by a factor of 100 when classifying clusters of arbitrary shapes.

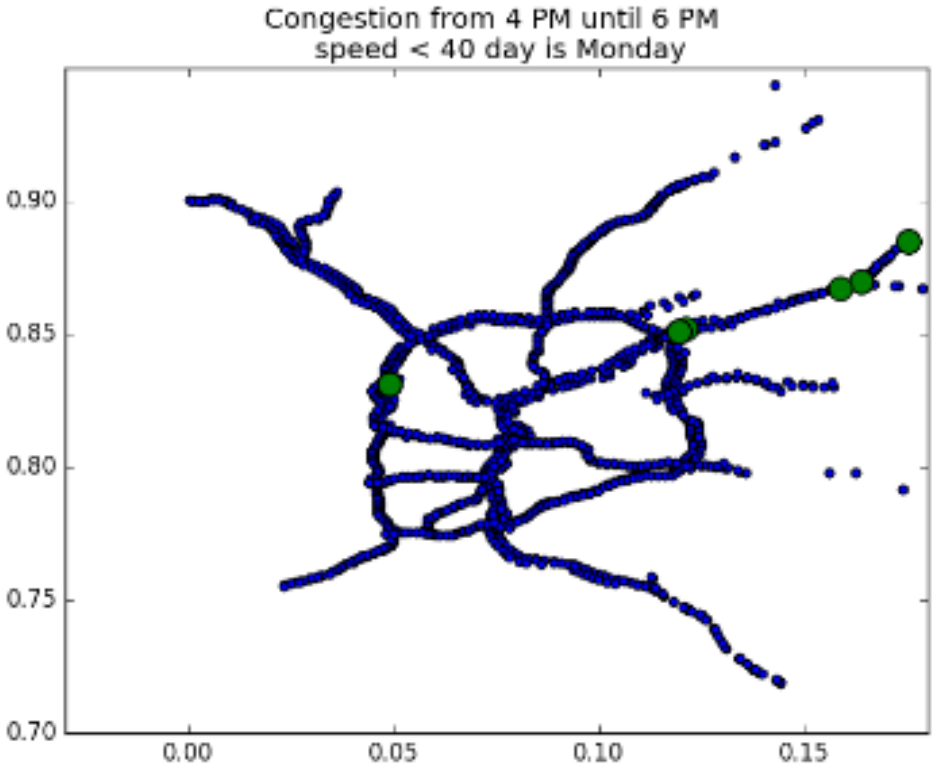

Figure 6: Monday 4:00 PM until 6:00 PM with speeds less than 40 MPH.

The clusters in Figure 6 indicate that congestion forming on interstate 285 and outlying areas to the northeast with one cluster on the west side of Interstate 285. Since the 4:00 until 6:00 PM is normally the time when commuters leave Atlanta the clusters

support the idea that roads leaving the urban area are more congested. This graph is produced for the average of all the Mondays in the study.

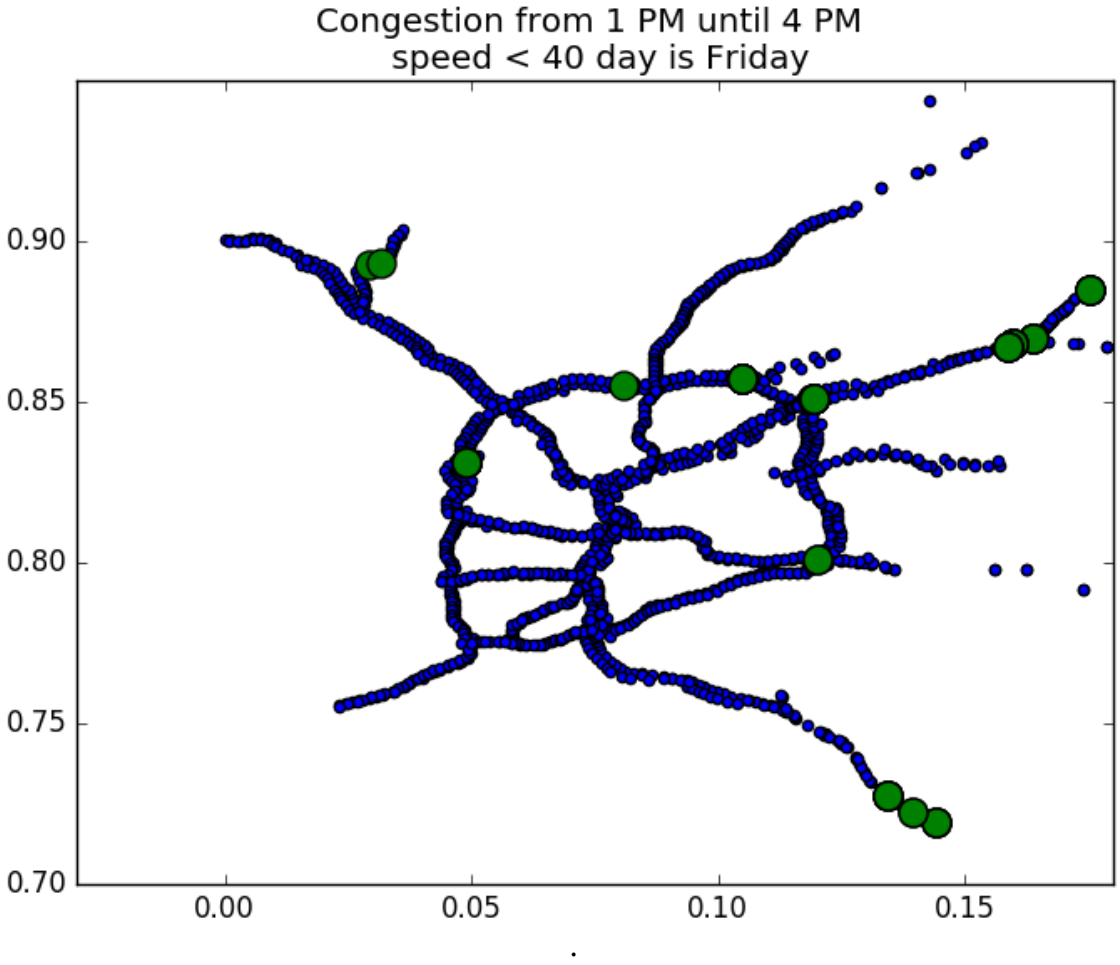

Figure 7: Friday from 1:00 PM until 4:00 PM with speeds less than 40 MPH.

The congestion clusters Friday from 1:00 PM until 4:00 PM indicate that more traffic is leaving the urban area (Figure 7). It is noted that most of the congestion is either on Interstate 285 which is the ring or in outlying areas as Interstate 75 to the

southeast, 575 to the northwest, and 85 to the northeast. Since Friday afternoon is traditionally a time to leave earlier for the weekend, this cluster pattern seems to indicate that pattern.

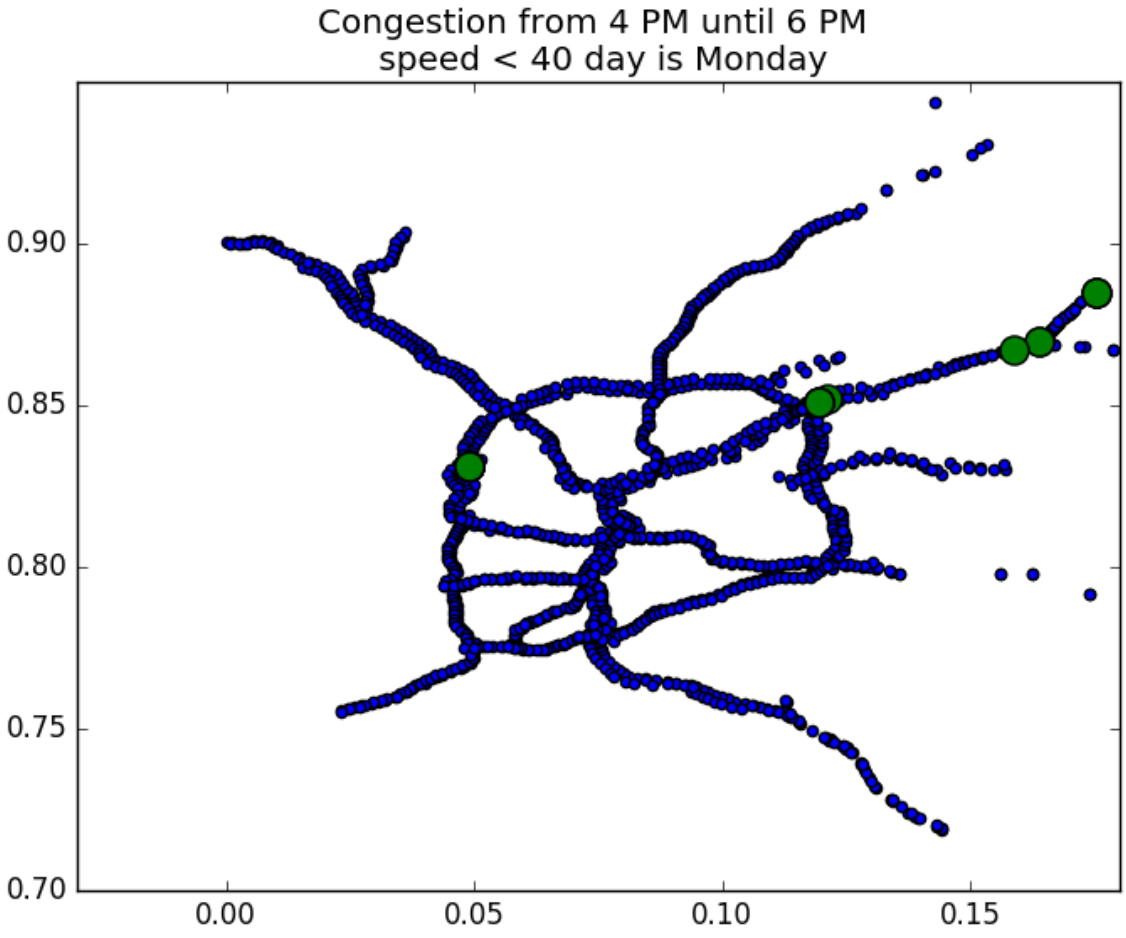

Figure 8: Congestion clusters Friday from 4:00 PM until 6:00 PM with speeds less than 40 MPH.

The congestion patterns on Monday and Friday at the time period of 4:00 PM until 6:00 PM indicate that certain areas tend to be more congested than others and most

of the congestion is on interstate 285 (the perimeter interstate). It is noted also that traffic on the outskirts of Atlanta is heavy (Figure 6 and Figure 8).

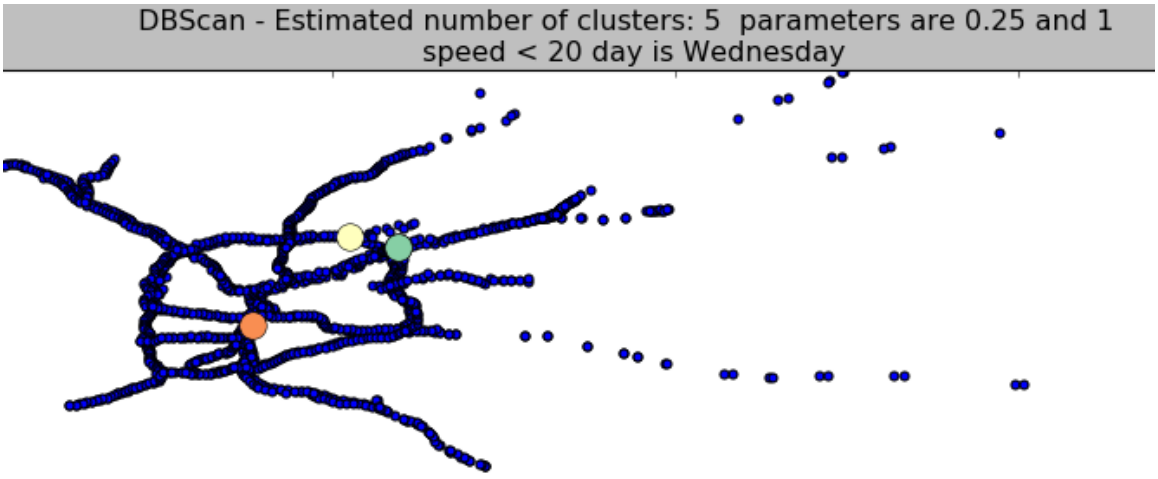

Figure 9: Clusters of congestion Wednesday with speeds less than 20 MPH.

Traffic clusters for the day Wednesday shows congestion inside the city as well as on the perimeter interstate. Wednesday is traditionally a heavy traffic day for commuters, Figure 9 shows one cluster in downtown Atlanta, two on interstate I 85, the ring around Atlanta, one down to the southeast and the last at the coordinates (0, 0) which indicate the camera inventory file did not have a longitude and latitude location but a mile marker post instead. This problem is dealt with by mapping the camera name to the

GIS map with ARCGIS in python. In Figure 9, all of the eight Wednesdays in the downloaded files are used.

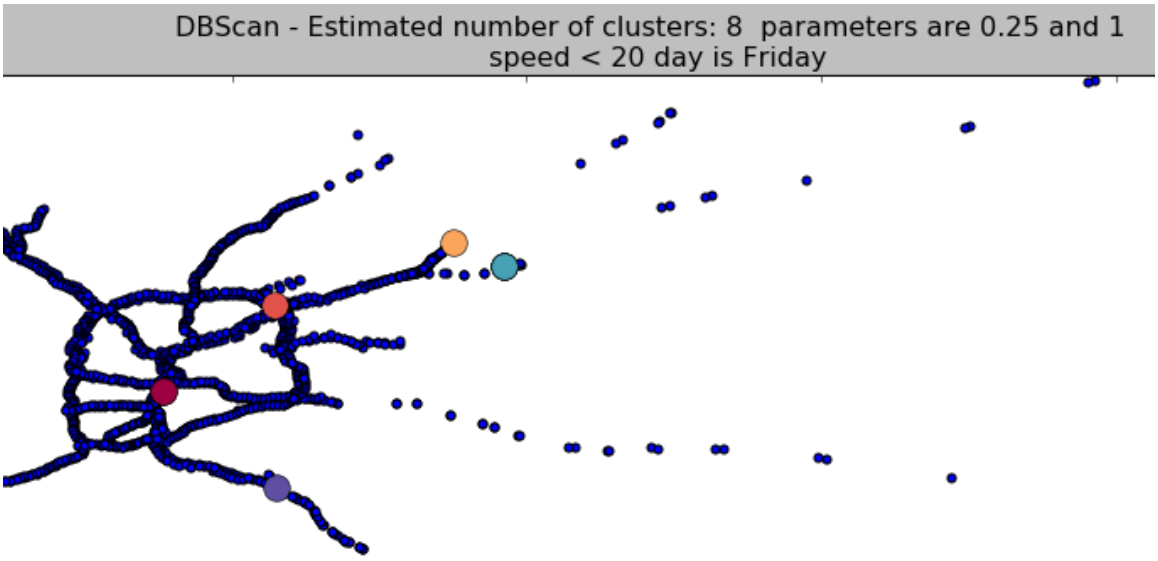

Figure 10: Congestion comparison for Friday from 4:00 PM until 6:00 PM with speed less than 20 MPH.

For the weekday Friday (Figure 10), traffic speeds less than 20 show heavy congestion in the interior of Atlanta as well as on the outskirts. Again the cluster at coordinates (0, 0) which indicate the camera inventory file did not have a longitude and latitude location but a mile marker post instead which is not mapped in the graph.

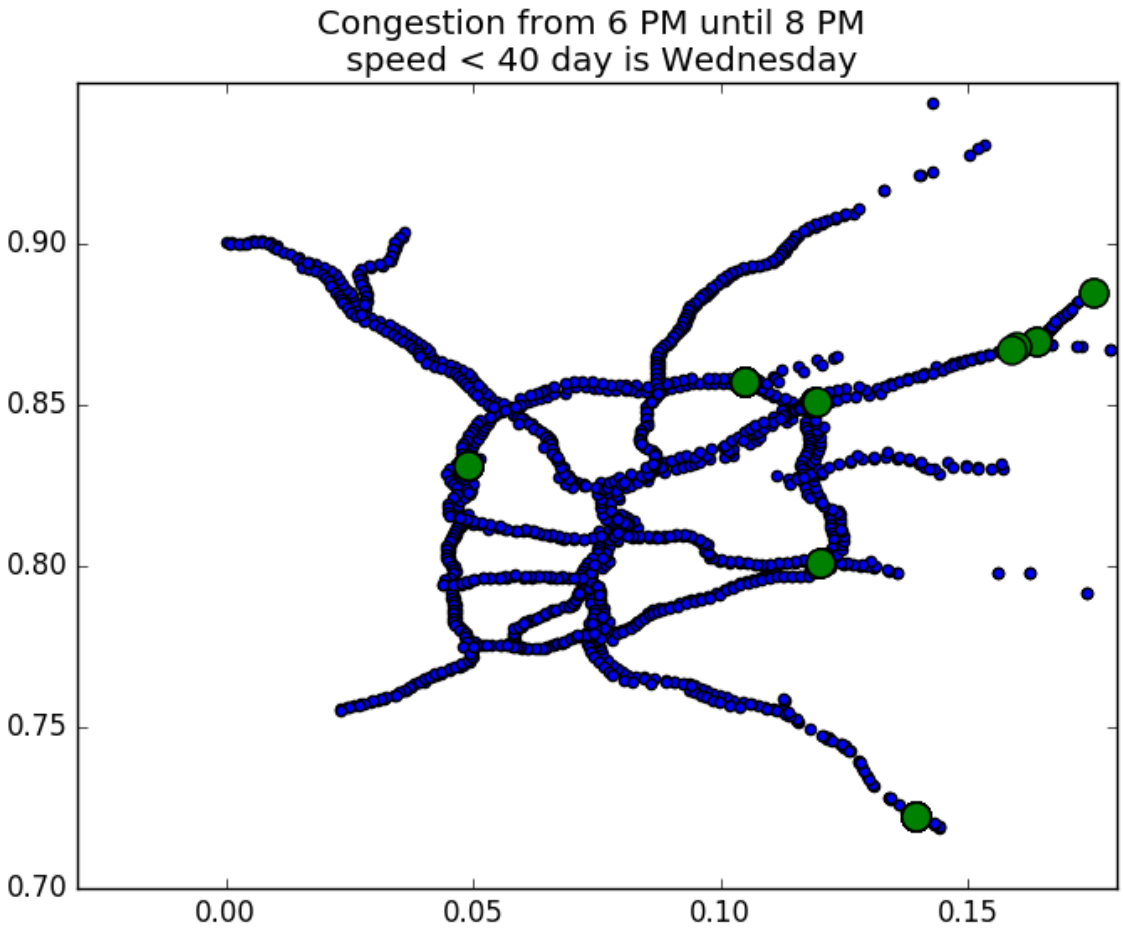

Figure 11: Congestion comparison for Wednesday from 6:00 PM until 8:00 PM with speed less than 40 MPH.

Figure 11 clusters indicate moderate congestion from 6:00 until 8:00 PM for a middle of the week workday. These figures were constructed using the DBSCAN clustering algorithm. This algorithm was chosen to find the number of clusters given the parameters and to discard the noise for data less than the parameters given.

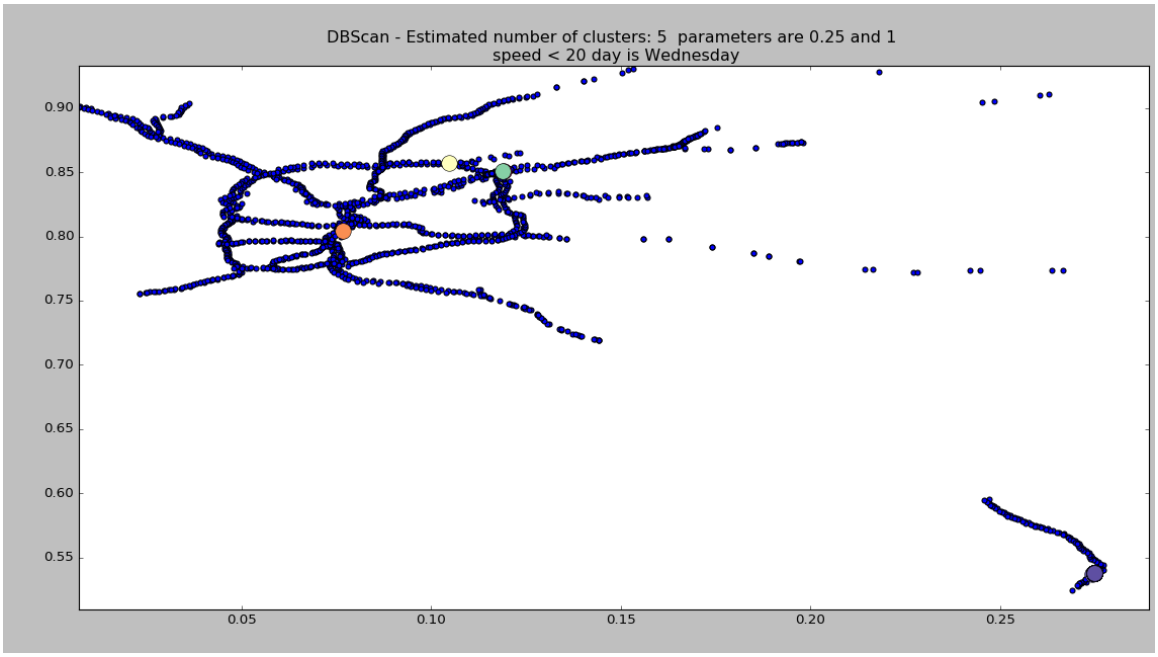

Figure 12: Clusters for Wednesday for speeds less than 20 MPH.

Figure 12 shows the number of clusters for the average of the Wednesdays in the study. This figure indicates congestion in the interior urban area possibly indicating that commuters do not leave as early on Wednesday.

The next matrices indicate the level of congestion on the system. Each location on the matrix represents the level of congestion either heavy (2 which is dark), moderate (1 which is gray) and free-flowing represented as zero (which is white). The matrix was constructed using weekday and time data for the levels of congestion at that time for each camera. The entries in the table are x_{ij} = level of congestion where i is the camera location and j is the time. Then the table is configured to be 48 x 48 in order to show the

congestion levels. This matrix is the first matrix considered for evaluation using a convolutional neural network.

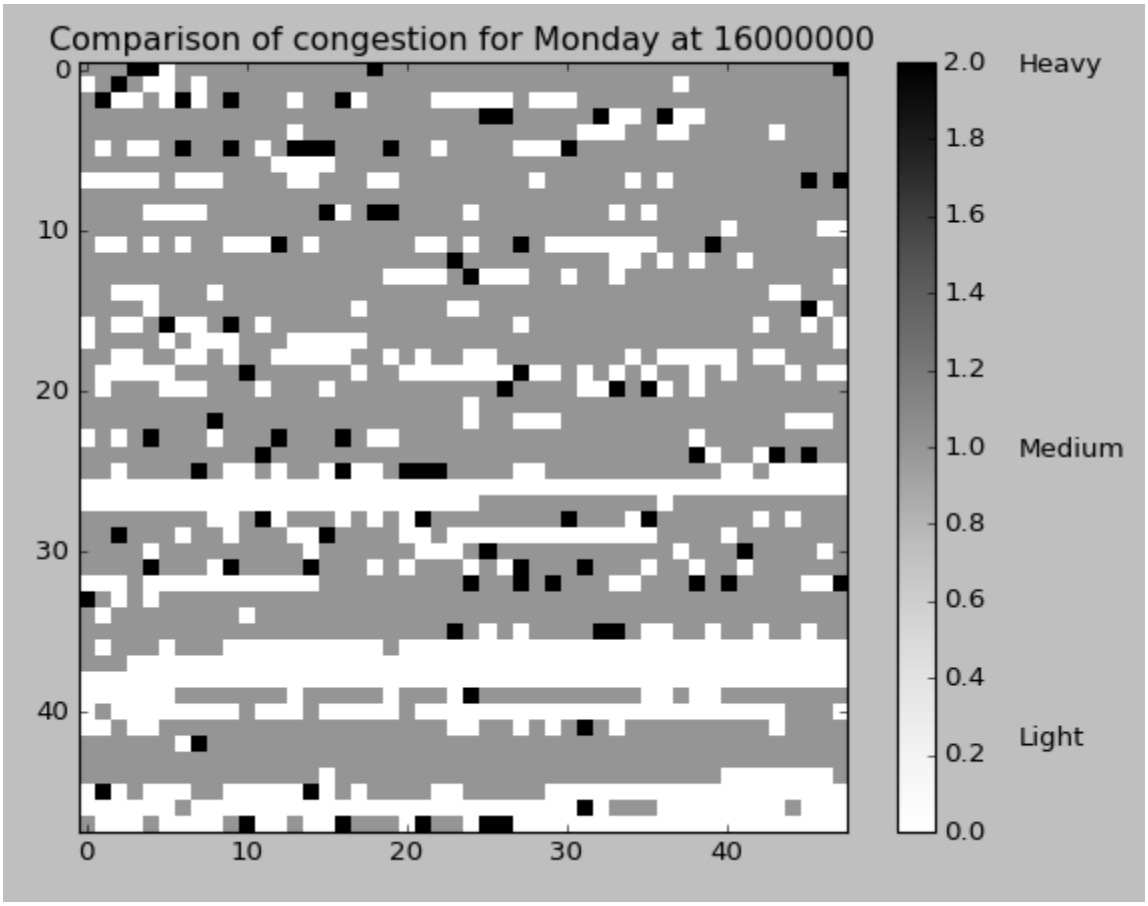

Figure 13: Comparison of congestion for all locations all Mondays at 4:00 PM.

Although locations cannot be directly intuited from Figure 13, the severity of congestion can be. In Figure 13, black indicates heavy congestion, gray moderate congestion and white indicates free flowing traffic. These matrices are built for every five-minute period for each of the days in the database. Then the matrices are built for

every five-minute period for each weekday in the database. Viewing these matrices as an animation plot clearly shows the patterns and severity of congestion in the network for that selection.

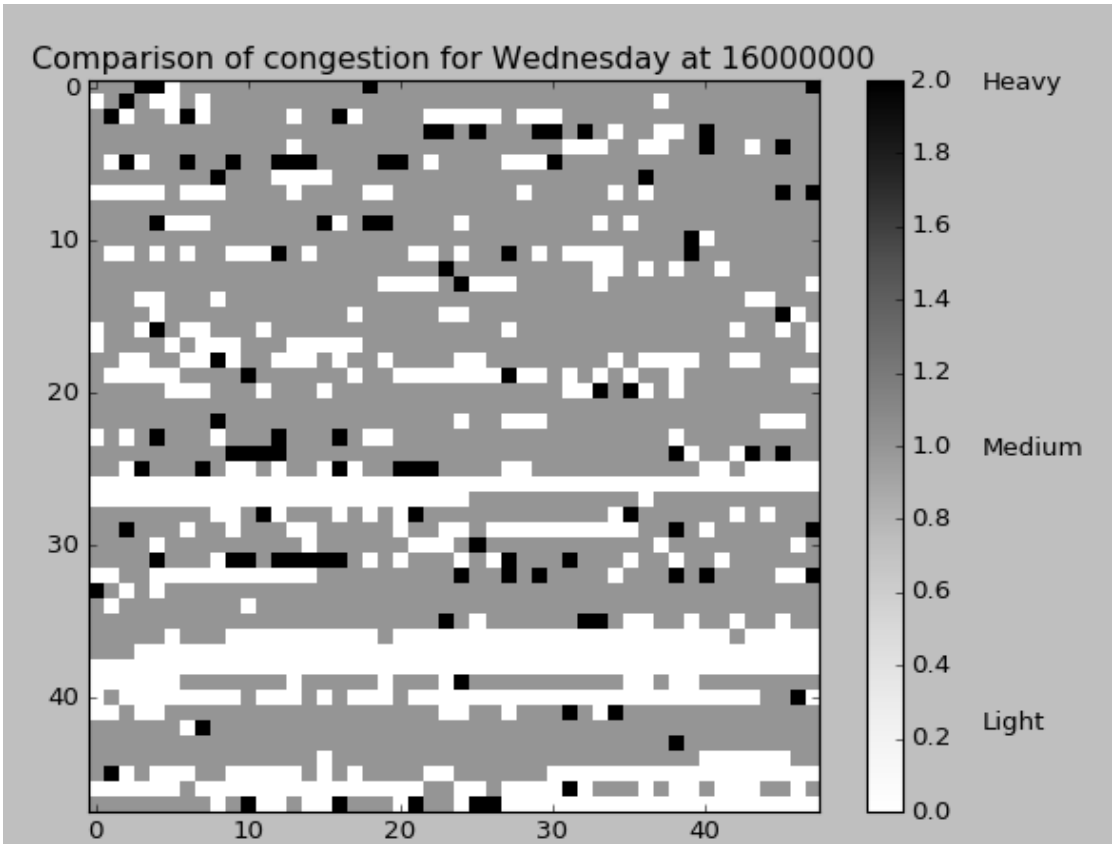

Figure 14: Comparison of congestion for all locations Wednesday at 4:00 PM.

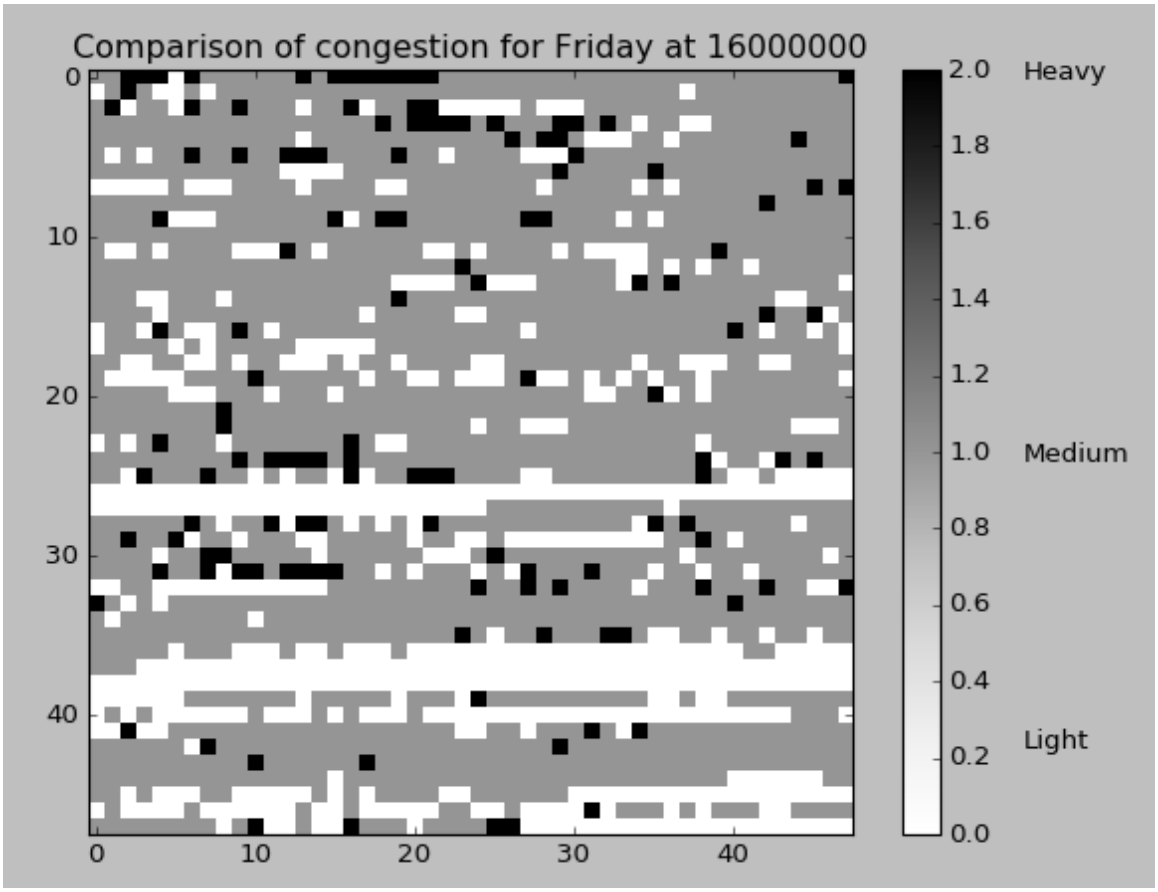

Figure 15: Comparison of congestion for all locations all Fridays at 4:00 PM.

Figure 13, Figure 14, and Figure 15 show the relative congestion for the three weekdays, Monday, Wednesday and Friday at the same time of day for all locations in the network. The matrices use categories of congestion: heavy is dark, moderate is gray and free flowing is white. Each matrix is composed of the average speeds for the total number of Mondays, Wednesdays, or Fridays in the study data.

The second part of this study predicts the average speed within the next timeframe (from a period of 5 minutes up to an hour). For this study, data was downloaded for 60

days which provided sufficient data to analyze spatially and temporally to determine historical congestion areas and times and to provide test data to predict speeds in the near future (up to an hour).

The traffic speeds were averaged for each camera (traffic location determined by longitude and latitude of the camera - obtained from the detectorinventoryresponse.xml file at the same web site) and time of observation. The individual speeds of each lane were averaged to get one speed for each camera at the five-minute snapshot. A matrix with rows as camera location and columns as time periods of one day (from midnight until midnight – giving 288 five-minute time periods) was constructed with x_{ij} = average speed for the lanes. This matrix has dimensions of 2284 x 288. This matrix was separated into 4 matrices for the different segments of the interstate system. For the purpose of this project, the first three sections are used as training data. The fourth section is reserved for testing. An image of this matrix for the date November 28, 2016 for the section consisting of interstate 285 is shown in Figure 16.

The data from the matrix sliced as 12 time periods (one hour) is used for training the Convolutional Neural Network. This slice uses the sequential speed data. The label for the training is sliced from the same matrix at the ith time period in different training episodes. This project uses the 14th time period only. Training was done for 100 epochs. The program uses a stop routine when the training data becomes over fitted.

This project demonstrates the possibilities of prediction with the deep learning network. As the epochs are increased, the accuracy improves. It is theorized that as more hidden layers are added, the accuracy will also improve. One Note: this project does not use a large enough sample to obtain good results from a Convolutional Neural Network. The purpose of this project is to demonstrate the feasibility of predicting speeds using a neural network.

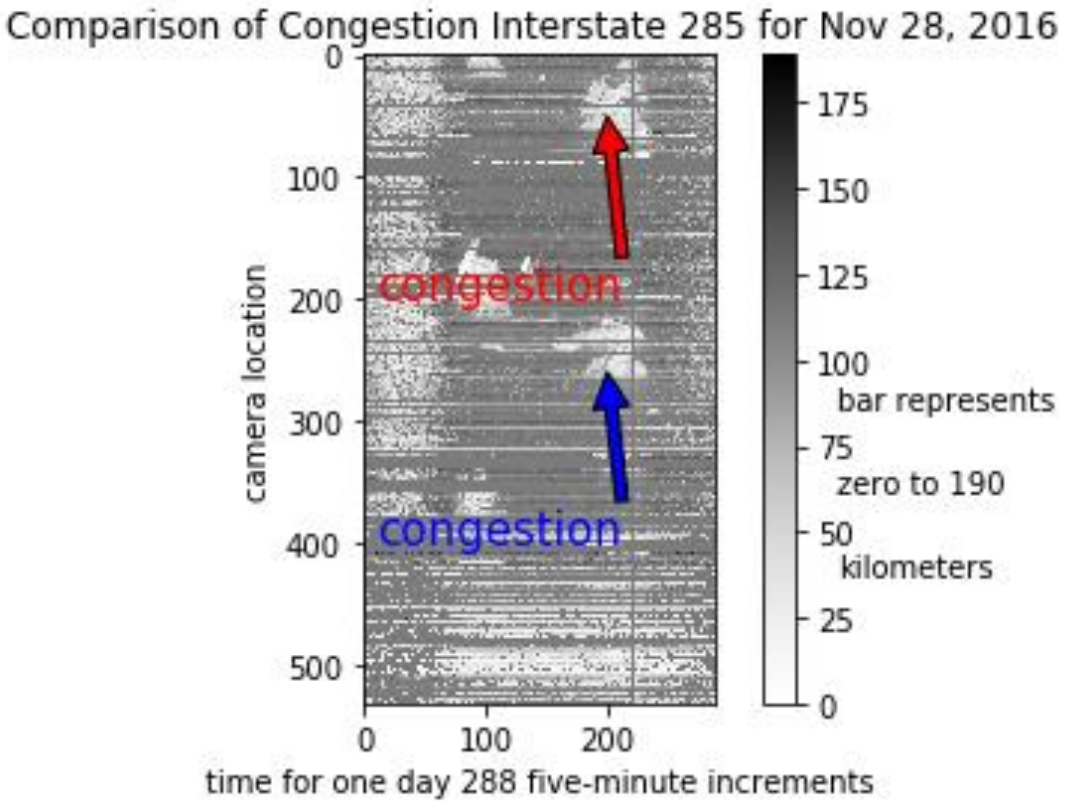

Figure 16: Plot indicating level of congestion for the cameras on Interstate 285 for November 28, 2016.

In Figure 16 congestion is indicated by the white areas marked with arrows, because the speeds are lower. This plot gives a good representation of the congestion patters by camera locations (row) and time (columns).

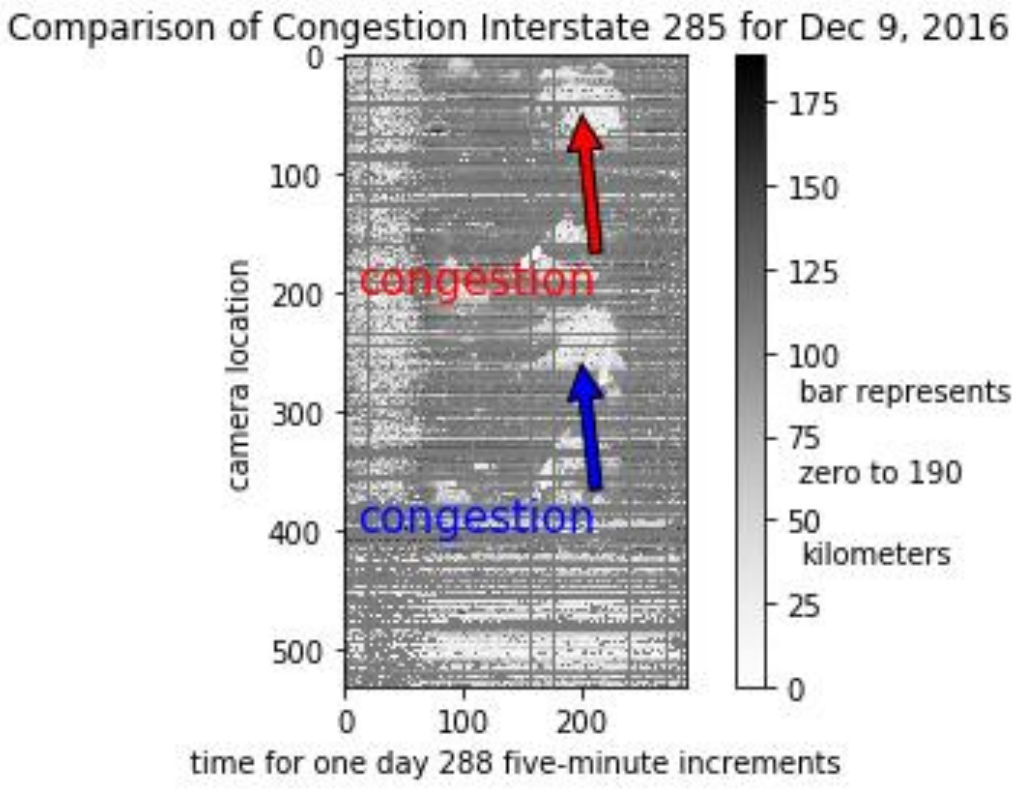

Figure 17: Plots indicating different levels of congestion for the cameras on Interstate 285 for December 9, 2016.

For the plots in Figure 16 and Figure 17, November 28, 2016 was a Monday and December 9, 2016 was a Friday. The lighter areas indicate congestion. Note: speeds are in kilometers. The areas of congestion are in similar locations for the two days indicating

a possible bottleneck location. These pictures of the congestion were used as input to the convolutional neural network.

Keras (Charles, 2013) based on Tensor Flow in Anaconda is used to construct the convolutional neural network. Highway segments one, two and three (representing a period of 62 days' data) are used as training data and segment four (cameras located on Interstate 75 and various other highways) is used as testing data. From two to four convolutional layers along with one fully connected layer are used in training. It is noted that this network trains to an integer speed. Random guessing giving an accuracy of 0.5 percent (from one to 190 kilometers) compared with an accuracy greater than 45 percent gives an 8900 percent increase. If the accuracy is calculated using a range of 10 (i.e. for a prediction of 55, if ground truth is from 50 to 59, prediction is considered correct, the accuracy would improve as it did for Ma (Ma et al., 2017) when categories of predictions were used to determine accuracy.

Actual data from the GDOT website is downloaded every five minutes and processed for use. The last time frame that has been downloaded will be used as input to the CNN using the trained results. Thus the prediction could be used by travelers to choose the best route to take. If the predicted result is very different than the ground truth, an incident is possible and camera monitors should be checked to determine the problem allowing immediate notification of a system problem.

The convolutional neural network has to be fine-tuned to get the correct results. The choice of optimizer, activator for each hidden layer, and loss function is critical to good results. In addition, care has to be taken that the network is not over fitted. Overfitting is the result of obtaining a trained set that closely corresponds to the data used for training and will not be able to test data outside of the trained set accurately. Thus the convolutional neural network must check for overfitting. This study uses the function from (Charles, 2013)) called early stopping. The format is:

early_stopping = Early Stopping (monitor='val_loss', patience=3).

This study uses the value of the loss function to determine when the test data accuracy is not improving indicating that the training function has begun to over fit. Optimizer adadelta based on adam (Adaptive Moment Estimation) (Ruder, 2016) one of the gradient descent optimization algorithms used in Keras (Charles, 2013) gives one of the best results in this training set. Other optimizers tested include: adam. adamax, adagrad, adadelta, stochastic gradient descent (SGD), and nadam. All of these optimizers have advantages and disadvantages. Choice of an optimizer depends on the characteristics of the image or data to be classified. Keras documentation (Charles, 2013) has numerous loss functions including mean squared error, mean absolute error, mean absolute percentage error, mean squared logarithmic error, categorical hinge, categorical cross entropy, binary cross entropy (not suitable for more than two categories, that is labels, for training), and others. It is necessary to evaluate the loss functions to determine which one that performs best on the dataset used for training. Activation for the hidden layers is

chosen to be RELU (rectified linear unit) since it allows for effective training on large datasets. Available activations according to Keras, (Charles, 2013) are: ranh, sigmoid, linear, relu, elu, selu, softplus, softsign, and hard sigmoid. Softmax is chosen as the activation for the fully connected layer because it is a categorical distribution and this study is classifying to categories. Since the convolutional neural network is training to categories, the label data has to be changed to categories using the function to_categorical (Charles, 2013). In addition, different batch sizes and numbers of epochs were used.

This algorithm trains on 229,152 samples and validates on 26,970 samples (one slide of data of the image, one slide is equal to one time slice for example from 6:00 AM until 6:30 AM. This data set is not large enough for training with a convolutional neural network, but it demonstrates the validity of training with it.

Figure 18 shows the accuracy and error plots using adam as optimizer with categorical_crossentropy as loss function, RELU as activation for the hidden layers and softmax as the activator for the fully connected layer.

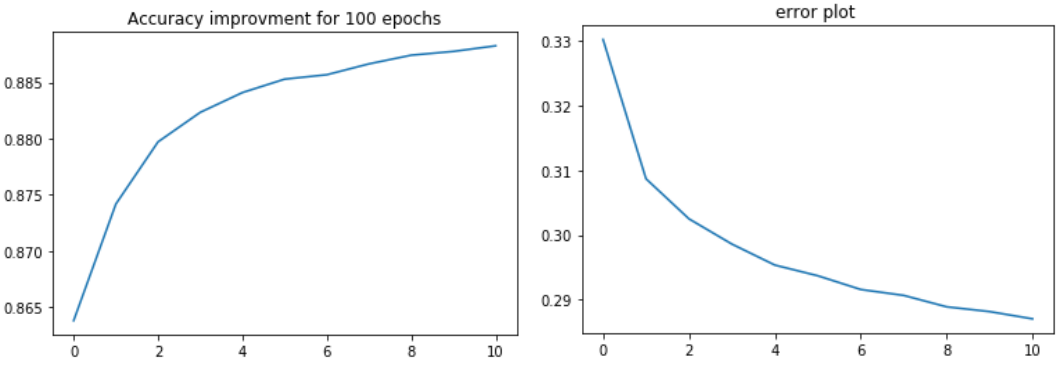

Figure 18 : Accuracy and error plots using adam, categorical_crossentropy, RELU, and Softmax.

Optimizer	Loss function	activation	Batch size	epochs	loss	accuracy
adam	Cat. entropy	relu	128	11	0.2822	88.88%
adam	MSE	relu	128	11	0.0213	89.00%
adadelta	MSE	relu	128	21	0.0215	89.12%
adamax	MSE	relu	128	11	0.0213	89.03%
adagrad	MSE	relu	128	25	0.0270	87.46%
adagrad	Cat. entropy	relu	128	10	0.0215	89.12%
adadelta	Cat. Entropy	relu	128	7	0.2898	87.79%
adamax	Cat. Entropy	relu	128	11	0.2831	89.00%
adamax	Cat. Entropy	relu	32	11	0.2745	88.91%
adamax	MSE	relu	32	15	0.0212	88.91%
adadelta	MSE	relu	32	12	0.0229	88.58%
adam	Cat. Entropy	relu	32	12	0.0214	88.73%

Figure 19: Chart for results of tests with Convolutional Neural Networks.

In the results shown in Figure 19, the number of epochs is controlled by the early stopping routine. All of the epochs were set to stop at 100, but each time the number of epochs is less than that in order to stop before overfitting. The best accuracy is obtained with adadelta as optimizer and mean squared error as loss function and with adagrad as optimizer and catergorical-entropy as a loss function. The second best is obtained with adamax as optimizer and mean squared error as loss function. Both adadelta and adamax are gradient based optimizers that adapt the learning rate to the parameters (Ruder, 2016). All of the above tests were run using training base of one hour (which is 12 data points,

one every five minutes). Another set of tests use 40-minute time frames for training and will train to the point at 10 minutes in the future. The third set of tests use 30-minute time frames from training and will train to the point at 10 minutes in the future. Based on the results in Figure 19, this study used adadelta and mean squared error for the rest of the tests. A good explanation of the function of adadelta is given by Ruder (Ruder, 2016).

Time span	accuracy	error
One hour	89.12%	0.0215
Forty minutes	89.12%	0.0218
30 minutes	88.80%	0.0211
20 minutes	88.87%	0.0217

Figure 20: Results for predictions using different time spans.

It is interesting that the different time spans yield the same approximate results, but it is remembered that the training set is small (Figure 20). All of the above training was done with two sliding bands of data. If 7:00 until 8:00 were selected as the first band, the second band would be 7:05 until 8:05 with the label data being at 8:10 and 8:15. The next test used 3 bands of sliding data from 7:00 until 8:00, 7:05 until 8:05 and 7:10 until 8:10. Results of this 3 band test indicate that adding one sliding band is not

enough data to make a difference in the algorithm. Running the algorithm with four bands did improve the results (Figure 21).

Time span	accuracy	error
40 minutes	89.19%	0.0211
30 minutes	89.16%	0.0210
20 minutes	89.04%	0.0213

Figure 21: Testing with 4 time bands.

Since the previous tests were run on small samples, the next test is run with ten slides. A slide is defined as one-time span, for example 30 minutes of training data. Ten slides would concatenate ten time spans together to us as training data with ten labels concatenated together to use as label data in the training set. Pseudo code for this routine:

X = subset of dataset from start time to start time plus subset size

For j in number of slides

Beg = start time + j times subset size

End = Beg + subset size

Array = subset of dataset from beg until end

This routine allowed the creation of a larger training set. For example, 10 slides will train on 1,191,950 samples, validate on 26970 samples. The results of the larger training set improved (Figure 22).

Number of slides	Time span	accuracy	Error	Training samples	time
1	40 minutes	89.12%	0.0218		
10	40 minutes	91.04%	0.0182	1191950	12 m 48 s
20	40 minutes	91.01%	0.0185	2237710	26m 47s
20	30 minutes	91.32%	0.0179	2237710	20m 39s
40	20 minutes	90.79%	0.0188	4629230	35m 49s

Figure 22: Results of training with larger training sample.

For the next training, the test data set was changed to the set containing Interstate 285. The tests were run starting at 6:00 AM (Figure 23). The next tests are run beginning at 4:00 PM.

slides	Time span	accuracy	Error	samples	time
10	20 minutes	89.21%	0.0221	1131810	16m 56s
20	20 minutes	91.03%	0.0185	2217430	27m 4s
10	30 minutes	90.32%	0.0196	1131810	17m 55s

Figure 23: Results for using the test data set for Interstate 285 beginning at 6:00 PM.

slides	Time span	prediction	accuracy	error	samples	time
10	20	10 minutes	90.33%	0.0196	1131810	13m 33s
10	30	20 minutes	90.05%	0.0205	1131810	13m 25s

Figure 24: Results for rush hour period beginning at 4:00 PM.

The results from the convolutional neural network for the different time periods and different test datasets have comparable results (Figure 23 and Figure 24). Therefore, it is concluded that this method of predicting speed levels in the near future have merit and should be further investigated.

Chapter 5 PRELIMINARY RESULTS AND FINDINGS

Sample preliminary results from the analysis were plotted in Figure 25 and Figure 26. Figure 25 and Figure 26 show the clustering of moderate and severe congestion segments around the Metro Atlanta area for a statistically computed Monday. It should be noted that each camera location shown in the plots may contain more than one individual congestion instance. The plot is simply a stacked up series of points indicating congestion at various occasions and times. Figure 27 shows the qualitative comparisons of the recurrence frequencies for the moderate congestion case.

In Figure 25, results show that clusters for the moderate congestion 40 mph data is more spread out than the 20 mph data based on the distribution of the cameras. The clusters also show that cluster 1 occupying the largest area. The cluster is concentrated in the downtown Atlanta. The size of the cluster provides clues about potential influence or effects of bottlenecks within the cluster. To observe the severity of the congested segment and cluster, the frequencies of the congestion during the data sampling period was plotted. Figure 27 provides a qualitative visual showing the frequency of congestion events at the different locations. In general, recurrent events will have a higher frequency than non-recurrent events. Figure 27 does not show how congestion changes in time, but it does show which locations are most prone to congestion. While not visible, it should also be pointed out that cluster 2, 3 and 4 are at the exact same camera location and

therefore was plotted on top of each other. Implicitly, the DBScan algorithm is creating clusters based on the three dimensions (x coordinates, y coordinates and time). When separate clusters are placed on the same x and y coordinates, this means that the congestion was being clustered in the time dimension and may be an indication of a recurring event that is time dependent.

For this case study, severe congestion happens primarily in one cluster within the metro Atlanta area where the worst recurrent congestion is near the intersection of Interstate 85 with GA 316 and extends to approximately 10 km to the northeast. This cluster is shown in Figure 25, and it is the only cluster around the metro Atlanta area affected by severe congestion. It should also be pointed out that comparing Figure 25 with Figure 26 the largest cluster obtained of the moderate congestion does not necessarily mean the worst congestion will occur within the cluster. In fact, it was shown in Figure 26 that the most severe congestion is nowhere near the downtown Atlanta area.

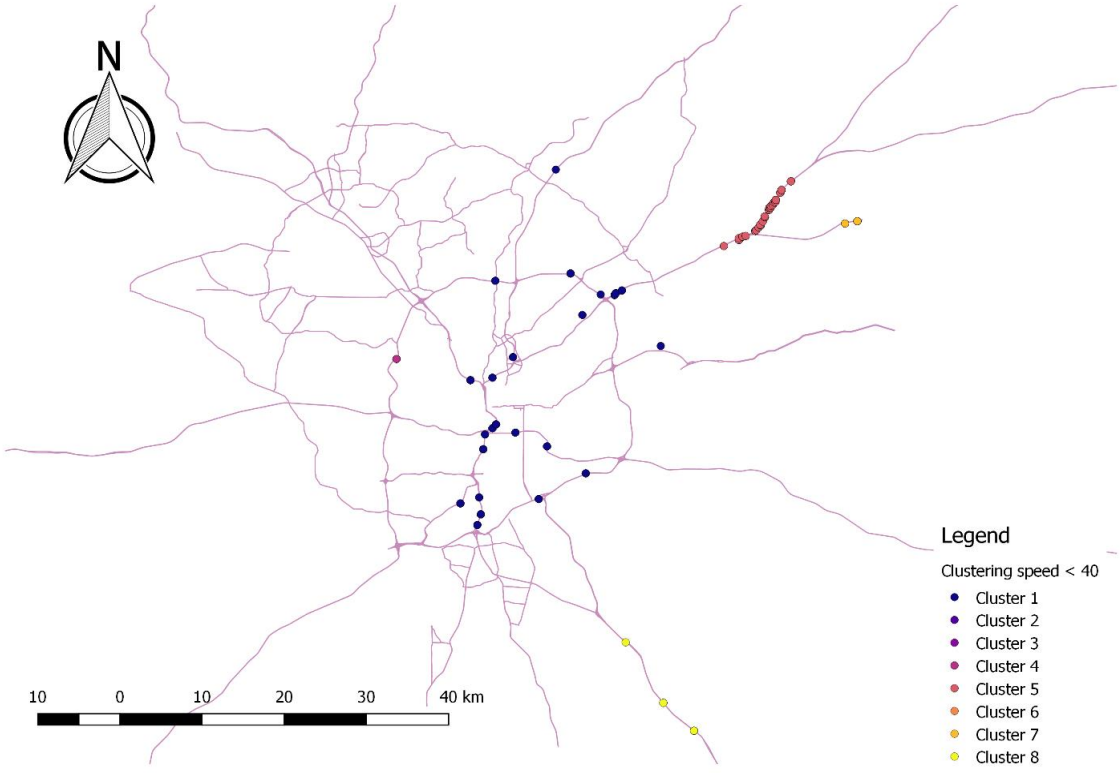

Figure 25: Cluster number assigned using DBScan for moderate congestion (speed<40 mph), for roads within the metro Atlanta area.

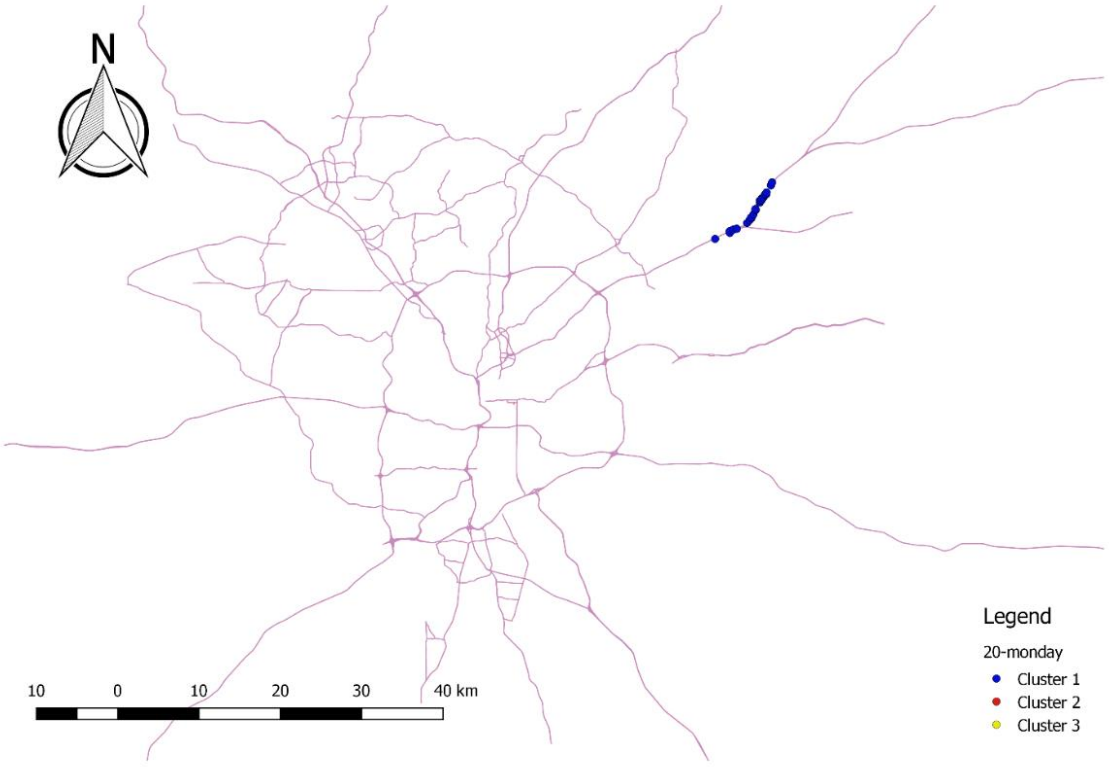

Figure 26: Cluster number assigned using DBScan for severe congestion (speed<20 mph), for roads within the metro Atlanta area.

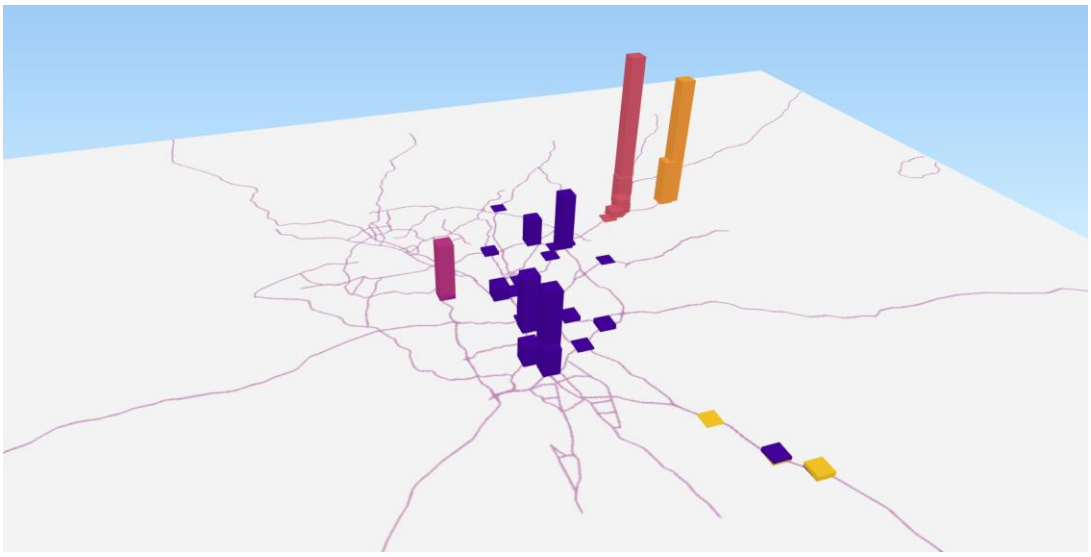

Figure 27: Plot of Clusters indicating sum of congestion events throughout the sampling period. The plotted data is an indicator of temporal severity of congestion clusters.

In the second part of this study, the convolutional neural network algorithm by (Charles, 2013) is employed to predict the traffic speeds for a short time frame, that is, from five minutes up to an hour. The network is successfully trained and it is planned that the training algorithm will be uploaded to the cloud and used for a method to predict normal congestion.

Chapter 6 CONCLUSION

The research showed the framework for a general spatiotemporal identification of traffic congestion model. The preliminary results shown does indicate the ability of the algorithm to observe congestion locations as well as the spatial influence of the congestion. In addition, statistical temporal data was displayed which shows the severity of each recurring congestion location and cluster.

From the Metro Atlanta case study, congestion can be related spatially by observing the distribution of the points based on its geographical location on the plotted map. The largest cluster happens to be in downtown Atlanta and its immediate surrounding areas; however, this does not necessarily mean that the largest cluster is the worst recurring congestion event. For our case study, severe congestion happens primarily in one cluster within the metro Atlanta area where the worst recurrent congestion is near the intersection of I-85 with GA 316 and extends to approximately 10 kilometers to the northeast.

Another point to summarize is that congestion is related temporally at individual locations in instances that different cluster numbers are assign to the same cameras. This indicates that the same camera is repeatedly flagged for congestion at different times.

The possibility of immediate notification of a traffic problem as well as the ability to allow travelers to get predicted speeds for the next hour is a big step forward to managing a traffic system and planning for a trip.

The results of the convolutional neural network tests that the possibility exists to build a website application that can show predicted speeds for the near future. Accuracy levels of above 90 percent for relatively small samples indicate that much larger samples will produce higher accuracy levels.

Chapter 7 FUTURE STUDY

This study is currently still ongoing and will move towards real-time identification of potential bottlenecks and perhaps evolve to an integrated traffic dynamics system for advance planning purposes. Data is continuing to be collected so that the analysis data can be expanded. It is thought that a larger database up to one year's collection of data will allow further exploration for the prediction and analysis of congestion. There are many ways to fine-tune the convolutional neural network programmatically as well as many ways to compare the data. For example, comparing time periods of every day, comparing time periods of a weekend, comparing time periods of a weekday, comparing time periods of a weekday. Results from the algorithms ConvLSTM NN (Convolutional Long Short Term Memory neural network) and K-NN (nearest neighbor) will be compared with Convolutional Neural Network.

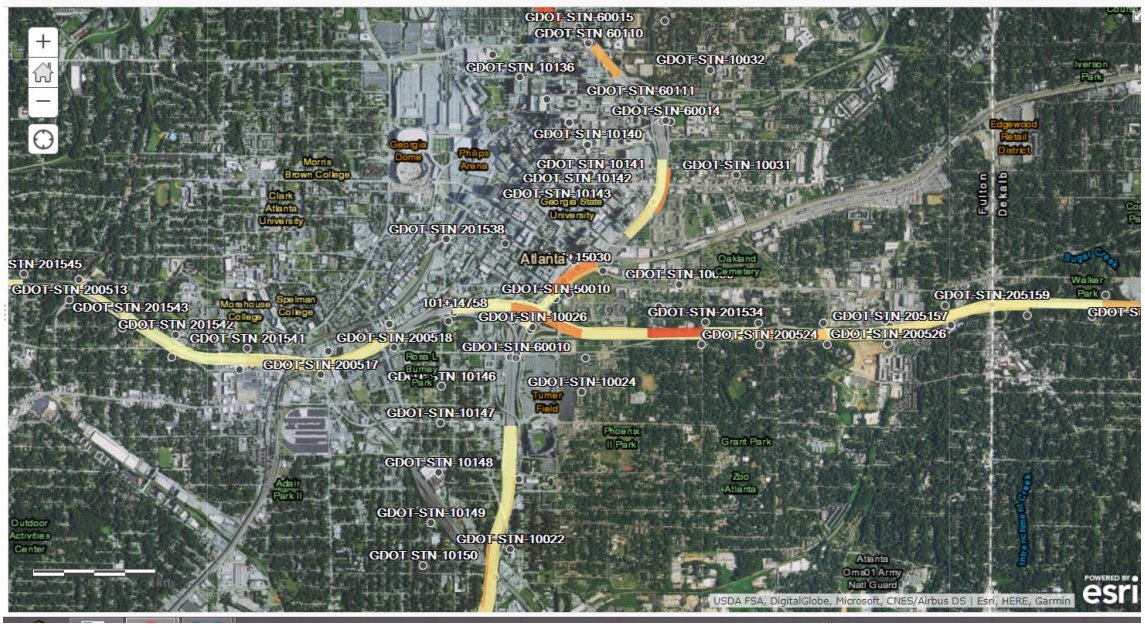

Figure 28: Planned GIS representation of predicted congestion areas in metro Atlanta.

In Figure 28, camera names have been overlaid on the GIS map of Atlanta with congestion indicated as red and orange. This figure is the desired outcome of the future study for this project. It is planned that the map will be refreshed every minute with the latest predicted speeds from the convolutional neural network.

After the training set is built, an app (a piece of software, a program that can run through a web browser in the cloud, on a smart phone, on a computer) will be built to process the latest update from the GDOT navigator website, run the processed data with the training set and post the results to the ARCGIS site so that anyone can view it.

Travelers could access the website and find the predicted speed for the next five minutes, ten minutes, up to 30 minutes to determine the route they want to take. This

ability might revolutionize the way commuters plan their travel. It also might improve the traffic flow in the metro area.

ACKNOWLEDGEMENTS

The author would like to thank Mr. Mark Demidovich and GDOT for providing assistance and access to data for this project. The author also gives special thanks to Dr. Chih-Cheng Hung, Dr. Jidong Yang, Dr. Tien Yee, and Dr. M. A. Karim for their encouragement, support, and help. Dr. Dan Lo also heartily supported the study and the author and thanks go to him. All of the author's instructors in the Computer Science Department are responsible for any success that the author enjoys. Sincere appreciation to all of them.

The author depends heavily on the support and encouragement of her spouse, Milledge L. Bonham and could not succeed with him. The author is indebted to her daughter, M. C. Price, who spent many hours of proof reading.

REFERENCES

An, S., Yang, H., Wang, J., Cui, N., & Cui, J. (2016). Mining urban recurrent congestion evolution patterns from GPS-equipped vehicle mobility data. *Information Sciences*, *373*, 515–526. https://doi.org/10.1016/j.ins.2016.06.033

Charles, P. W. D. (2013). Project Title. *GitHub Repository*.

Davis, G., Nihan, N., & Hamed, M. (1990). Adaptive Forecasting of Freeway Traffic Congestion. *Transportation Research*, *1287*, 29–33. Retrieved from http://onlinepubs.trb.org/Onlinepubs/trr/1990/1287/1287-004.pdf

Delhi, N., Committee, T., Leduc, G., Liu, Y., Feng, X., Wang, Q., … Zheng, Y. (2014). A novel approach for vehicle specific road / traffic congestion Student Name : Shilpa Garg. *Proceedings of the 3rd ACM Symposium on Computing for Development - ACM DEV '13*, *49*(4), 286–305. https://doi.org/10.1145/2743025

Dougherty, M. S., Kirby, H. R., & D., B. R. (1993). The Use of Neural Networks to Recognize and Predict Freeway Traffic Congestion. *Traffic Engineering & Control*, *34*, 6.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 226–231. https://doi.org/10.1.1.71.1980

Fouladgar, M., Parchami, M., Elmasri, R., & Ghaderi, A. (2017). Scalable Deep Traffic Flow Neural Networks for Urban Traffic Congestion Prediction, (March). https://doi.org/10.1109/IJCNN.2017.7966128

Hashemi, H., & Abdelghany, K. (2015). Real-Time Traffic Network State Prediction for Proactive Traffic Management. *Transportation Research Record: Journal of the Transportation Research Board*, *2491*(February), 22–31. https://doi.org/10.3141/2491-03

He, F., Yan, X., Liu, Y., & Ma, L. (2016). A Traffic Congestion Assessment Method for Urban Road Networks Based on Speed Performance Index. *Procedia Engineering*, *137*, 425–433. https://doi.org/10.1016/j.proeng.2016.01.277

Huang, S.-H., & Ran, B. (2003). An Application of Neural Network on Traffic Speed Prediction Under Adverse Weather Condition. *82nd Annual Meeting of the Transportation Research Board, Washington, DC*, (June 1995), 1–21. Retrieved from http://www.researchgate.net/profile/Bin_Ran/publication/265318230_An_Application_of_Neural_Network_on_Traffic_Speed_Prediction_Under_Adverse_Weather_Condition/links/54999dbe0cf22a83139625a2.pdf%5Cnhttp://www.ltrc.lsu.edu/TRB_82/TRB2003-000915

Koesdwiady, A., Soua, R., & Karray, F. (2016). Improving Traffic Flow Prediction with Weather

Information in Connected Cars: A Deep Learning Approach. *IEEE Transactions on Vehicular Technology*, *65*(12), 9508–9517. https://doi.org/10.1109/TVT.2016.2585575

Lee, K., Hong, B., Jeong, D., & Lee, J. (2014). Congestion pattern model for predicting short-term traffic decongestion times. *2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014*, (January 2015), 2828–2833. https://doi.org/10.1109/ITSC.2014.6958143

Liu, W., Wen, Y., Yu, Z., & Yang, M. (2016). Large-Margin Softmax Loss for Convolutional Neural Networks. *Proceedings of the 24th European Transport Forum, Traffic Management and Road Safety, Brunel University, London, UK, 2nd - 6th September, 1996.* Retrieved from http://arxiv.org/abs/1612.02295

Lyons, G. D., Hounsell, N. B., & Williams, B. (1996). Urban Traffic Management; the Viability of Short Term Congestion Forecasting Using Artificial Neural Networks. *Proceedings of the 24th European Transport Forum, Traffic Management and Road Safety, Brunel University, London, UK, 2nd - 6th September, 1996.*

Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., & Wang, Y. (2017). Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors (Switzerland)*, *17*(4). https://doi.org/10.3390/s17040818

Min, W., Wynter, L., & Amemiya, Y. (2007). Road Traffic Prediction with Spatio-Temporal Correlations Road Traffic Prediction with Spatio-Temporal Correlations. *IBM Research Report, 24275*, RC24275 (W0706-018) June. Retrieved from http://domino.watson.ibm.com/library/CyberDig.nsf/home

Nguyen, H., Liu, W., & Chen, F. (2017). Discovering Congestion Propagation Patterns in Spatio-Temporal Traffic Data. *IEEE Transactions on Big Data*, *3*(2), 169–180. https://doi.org/10.1109/TBDATA.2016.2587669

Pongpaibool, P., Tangamchit, P., & Noodwong, K. (2007). Evaluation of Road Traffic Congestion Using Fuzzy Techniques. In *TENCON 2007 - 2007 IEEE Region 10 Conference*.

Rempe, F., Huber, G., & Bogenberger, K. (2016). Spatio-Temporal Congestion Patterns in Urban Traffic Networks. *Transportation Research Procedia*, *15*, 513–524. https://doi.org/10.1016/j.trpro.2016.06.043

Ruder, S. (2016). An overview of gradient descent optimization algorithms, 1–14. https://doi.org/10.1111/j.0006-341X.1999.00591.x

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3D convolutional networks. *Proceedings of the IEEE International Conference on Computer Vision, 2015 International Conference on Computer Vision, ICCV 2015*, 4489–4497. https://doi.org/10.1109/ICCV.2015.510

Vallet, A., & Sakamoto, H. (2015). A Multi-Label Convolutional Neural Network for Automatic

Image Annotation. *Journal of Information Processing*, *23*(6), 767–775. https://doi.org/10.2197/ipsjjip.23.767

Wen, H., Sun, J., & Zhang, X. (2014). Study on Traffic Congestion Patterns of Large City in China Taking Beijing as an Example. *Procedia - Social and Behavioral Sciences*, *138*(0), 482–491. https://doi.org/10.1016/j.sbspro.2014.07.227

Xu, L., Yue, Y., & Li, Q. (2013). Identifying Urban Traffic Congestion Pattern from Historical Floating Car Data. *Procedia - Social and Behavioral Sciences*, *96*, 2084–2095. https://doi.org/10.1016/j.sbspro.2013.08.235

Zhang, Y. C., Zuo, X. Q., Zhang, L. T., & Chen, Z. T. (2011). Traffic congestion detection based on GPS floating-car data. *Procedia Engineering*, *15*, 5541–5546. https://doi.org/10.1016/j.proeng.2011.08.1028

Zhao, J. D., Xu, F. F., Guo, Y. J., & Gao, Y. (2016). Traffic congestion detection based on pattern matching and correlation analysis. *Advances in Transportation Studies an International Journal Section A*, *40*, 2016. https://doi.org/10.4399/97888548970073