# Traffic Data Collection and Visualization Tool For Knowledge Discovery Using Google Map

Iftekhar Hossain, Naushin Nower
*Institute of Information Technology, University of Dhaka, Bangladesh*

## ABSTRACT

*Traffic jam is increasingly aggravating in almost every urban area. Traffic forecast, traffic modeling, visualization can help to provide appropriate route and time for traveling and thus provides a significant impact on traffic jam reduction. For traffic forecasting, modeling and visualization, city-wide traffic data collection and analysis are needed, which is still challenging in many aspects. This paper aims to develop a tool for acquiring and processing traffic data from Google Maps that can be used for forecasting, modeling, and visualization. Dhaka city is used as a case study since there is no infrastructure available for traffic data collection. The traffic flow intensity of the road is analyzed to determine the congestion of the road. The flow intensity is used for traffic modeling, visualization, traffic prediction and many more.*

Keywords: Traffic data collection, Traffic data analysis, Prediction, Flow intensity, Image processing

## INTRODUCTION

Traffic jams have become a serious issue in many cities in the world as it affects people's daily life, fuel consumption, time, and air pollution. It has a significant impact on economic productivity and the environment in urban areas. City-wide traffic modeling, visualization, analysis, and prediction can be used to assist transportation planners to make better decisions about improving traffic jams and also help dwellers to make better decisions. Thus, traffic flow management and analysis have become essential for both individuals to better manage daily routes, and for transportation, planners to optimally schedule road infrastructure maintenance and development tasks (Jiber et al., 2018). As a result, it is needed to have traffic data collection and apply effective methods to process and analyze the collected data to extract meaningful information. Traffic conditions can be improved by improving infrastructure also; however, it is expensive, and without proper traffic management, it cannot be a standalone solution to the problem (Hossan and Nower 2020). Proper traffic data analysis and visualization can provide an appropriate guideline to manage road traffic even as it suggests which infrastructure needs improvement. Therefore, the analysis of traffic data to extract the nature of the traffic stream is one of the most important requirements of traffic management systems.

To improve traffic conditions and to manage transportation systems more effectively, various techniques have been adopted to collect traffic data from various sources, such as stationary sensor/camera-based methods (monitoring from traffic sensors or optical devices), air/space-borne methods (monitoring from airplanes or satellites), and floating-car based methods (monitoring from floating/probe cars), etc. However, the above-mentioned methods have a lot of limitations such as they are costly and require

maintenance. Moreover, stationary sensor/camera-based methods can only measure traffic data at fixed positions. To get the traffic information of the whole transportation network, a large number of detectors are needed, as a result, the system can be very expensive. Air/space-borne methods can monitor traffic conditions over large areas, but the data are available only when the air/space-borne detectors are flying over the monitored areas. Thus, the implementation of traffic monitoring systems requires a considerable amount of investment in equipment and qualified personnel to process the data (Lima and Campos 2017). However, many cities in the world suffer a huge traffic jam but there is no traffic monitoring infrastructure such as stationary sensors or detection mechanisms available.

In this context, a geographic information system (GIS) can be utilized which is designed to capture, store, manipulate and manage all types of traffic data (Ciepłuch et al., 2010). It is experimentally shown that the traffic data gathered from the web sources can be considered as a reliable data source that does not require a high level of accuracy (Lima and Campos 2017). Different GIS systems, such as Google map, Bing map, Open Street Maps offer huge amounts of geographic data, for example, street map, aerial/satellite imagery, geocoding, searches, routing functionality, and traffic jam, etc. This traffic information can act as a preferential source of information where monitoring infrastructure is not available. However, traffic information gathered from a web application requires a paid Application Program Interface (API) of these services which is not free of cost.

This paper aims to design a traffic flow analysis tool using Google map traffic information without incorporating its API. The designed tool collects traffic information from the Google map images and processes to extract information that can be directly used to analyze, visualize, and predict traffic jams. A methodology is developed to continuously capture Google map images with traffic information, processes it, and finally provides time-to-time traffic intensity of that particular route. That is the output of the developed tool is used to establish a citywide traffic jam modeling. Dhaka city has been chosen as a case study since it has huge traffic jams, a lack of traffic monitoring system and it has no traffic database for visualization. After acquiring Google map traffic jam information, map images were processed to establish Dhaka's traffic jam database. From this database, the flow intensity of a target route, the busiest route, the busiest time, has been identified, traffic jam prediction can be made which helps individuals to make decisions and helps city planners to make better plans for infrastructure. Then, to get the future flow intensities for a target street, a prediction model can be developed from this traffic information also.

## BACKGROUND

With the rapid development of urbanization, collecting and analyzing traffic flow data are of great significance to build congestion-free smart cities. Thus, there are many techniques and methods used to collect traffic flow information. Based on how to collect traffic data, there are mainly two ways to collect traffic data, i) traffic data from infrastructure ii) traffic data from web map services.

There are a huge number of technologies used to collect traffic data from monitoring technologies buried in infrastructure. The most common such technologies are loop detectors, video cameras, active and passive radars, electrical and mechanical sensors, wireless sensor networks, and, more recently, the employment of vehicles and people acting as a sensor (Ciepłuch et al., 2010). Infrastructure devices, e.g., loop detectors, sensors, and video cameras are used to continuously collect traffic data. They are fixed on some important roads and can detect vehicles passing by. From these measurements, the traveling speed of a vehicle and the total number of vehicles passing by (traffic flow) within a period is calculated. Besides these, in many modern cities, a large number of public vehicles are equipped with GPS devices to periodically report their current location, speed, etc. From these data, the road traffic condition can be derived (Liu et al., 2018). All of the above technologies are costly, require maintenance, and the traffic monitoring infrastructure is too expensive to be placed along with the entire

road network in many cities of third world countries. Moreover, traffic data from sensors, loop detectors cameras, etc are not accessible for education and research purposes in most cases.

In the context of limited urban facilities and resources, traffic data from web services could be a potential data source for analysis. Recently, some academic and industrial efforts have been made to mining traffic-related data from web services. There are many ways to get traffic-related data from web services. For example, web map services such as Google Maps, Bing map, etc could be a source to get traffic data. Besides this, social media such as Twitter, Instagram are also becoming popular because of containing self-generated data.

Web map services offer many services, such as positioning and real-time traffic conditions. Among Google Maps, Open Street Map (OSM) and Bing Map are widely used. Google Maps offers satellite imagery, street maps, real-time traffic conditions, and route planning for traveling by foot, car, bicycle, or public transportation. It indicates the traffic condition by four colors, such as blue, yellow, red, and dark red represents fast, slow, congestion, and more serious congestion, respectively. It collects traffic data from the cellular telephone in a crowd-sourcing way. Bing map is also a web mapping service that provides street maps, road views, aerial views, bird's eye views, traffic information, etc. Both Bing and Google Maps are closed source maps and provide quality results (Taylor 2017). However, a comparative study between Google Maps and Bing Maps shows that Google Maps shows more features and labels than Bing Maps for many countries. Besides this, Google maps have higher coverage in many countries in the world compared to Bing maps (Ciepłuch et al., 2010). The traffic data from web map services are mostly not designed for research purposes that means additional efforts are needed to collect these data. There are two ways to access these web map traffic data: one is requesting via API and the other is crawling the websites. The former approach is a paid one, on the other hand, crawling websites is more complicated than the former one. However, it is free and has no rate limits (Zhao et al., 2017).

Recently, social media has become very popular because of the active participation of people. People willingly share their daily activities, life events, etc using their portable devices. From the analysis, it is shown that there are a huge number of posts related to traffic incidents and traffic jams. These texts, posts, and photos can be processed using NLP, sentiment analysis, image processing, etc to extract traffic-related information (Zhao et al., 2017). In the paper [Instagram], the authors showed that the traffic-related data from social sensors (Ribeiro et al., 2014) and web map services are highly correlated and can be used as data sources.

In the paper (Tostes et al., 2013), Bing map traffic information is extracted to analyze and predict the traffic jam of the Chicago city. However, this work uses paid APIs from Bing map to acquire and analyze data for predicting traffic jams. The paper (Tostes et al., 2013) develops a traffic flow information collection and visualization tool from web map services without using paid APIs. To do this, an image processing method is developed to extract traffic information from the web map services.

## METHODOLOGY

To get the road traffic information for flow analysis, pattern discovery, etc, a tool is developed in this paper to acquire city-wide traffic flow data. It takes a map service as an input, extracts the traffic information from the map, and finally provides a CSV file that contains processed traffic information. The tool starts collecting the snapshots of the traffic flow of a particular route for a particular time (for example three months). The analysis takes place by taking snapshots of Google map's traffic flow of the selected route. The snapshots were captured in an interval of *n* minutes. Once a single snapshot is captured, it is processed using the developed image processing software to extract desired information. Information that is extracted from the snapshots is compiled into a CSV file. The CSV file is

then used for the analysis, flow pattern discovery, and even to predict the possible traffic patterns of a given future time. The whole process is shown in Figure 1.

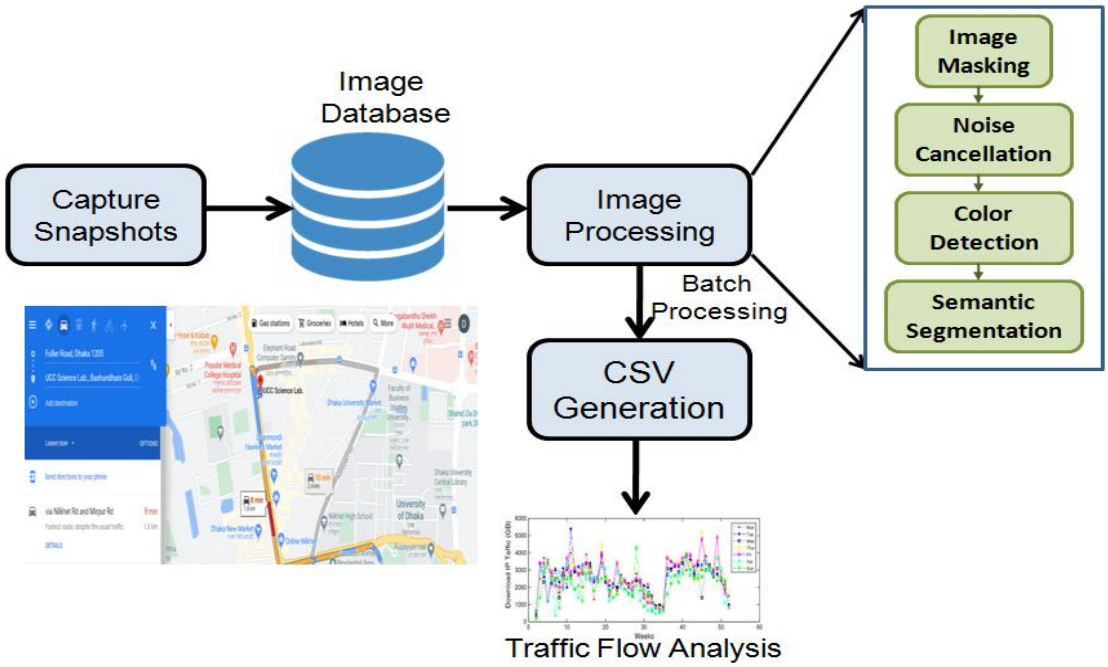



*Figure 1. The traffic flow intensity acquisition process*

### Capture Snapshots

The tool collects the images from the Google map for a selected route. At first, it takes a date, duration, and a route to capture snapshots. Since it uses Google map as a main source of information, it is needed to collect images from Google map for a selected route. Then, it continuously captures a snapshot after the interval of fifteen minutes to observe the traffic flow. It is observed that the traffic intensity of a road does not vary much within fifteen minutes. In this way, the snapshots are captured until the required time is reached and save snapshots in the image database. As it is unfeasible and error-prone to continue the process manually, as a result, Selenium web driver is developed to automate the repeated browser activities. This developed web driver is convenient to use and can easily be used to automate tedious, repeated browser activities. It performs the following steps to capture and saves continuous traffic flow snapshot from the map:

- Open web browser
- Open Google map
- Take the date, duration and route information
- Automatically increase time to *n* minutes
- Take Snapshot
- Save it in the image database

The sample of a collected image is shown in Figure 2.

## Image processing

This is the most challenging part of the designed tool. It takes the snapshot collected from the previous steps and as no defined measurement can be made from the Google map data, it is needed to do a lot of

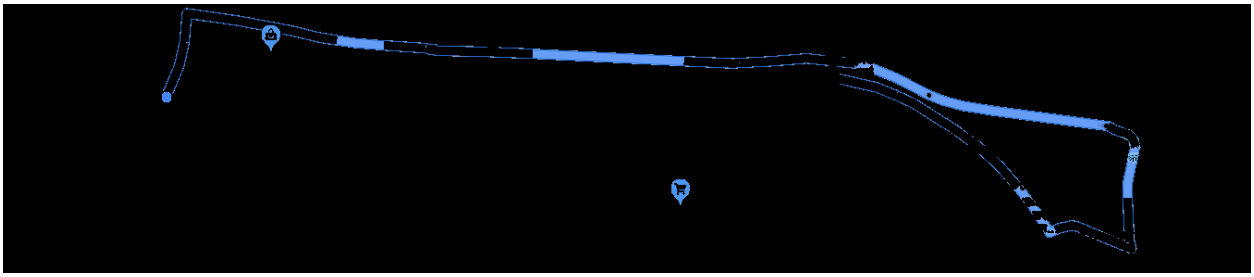
preprocessing work to extract some meaningful information from it. The main goal is to extract meaningful text data from the collected snapshots. This step takes the traffic flow snapshot as an input, processes it to remove noise and unnecessary information. To process the collected snapshots, the following steps are used.

- Image Masking
- Noise Cancellation
- Color detection using RGB value
- CSV Generation

*Figure 2. The sample of a collected image*

## Image Masking

As it is mentioned earlier, screenshots taken from Google maps provide little information that cannot be used directly for applying pattern discovery or in a machine learning model. As a result, the snapshots are needed to be processed. The intensity of traffic flow in the Google map is represented by the four colors, blue, yellow, red, and dark red. The selected route with traffic intensity color is needed to be extracted and others are considered as noise. To do this, the RGB image is converted to the HSV, since in HSV, it is easier to represent a color than RGB color-space. It is needed to create a mask to extract blue, yellow, red, and dark red colors from the map that contains traffic information. For blue color detection, a binary mask is created to retain the blue color with the black background. In this way, the other colors are extracted using corresponding masks. That is, in this step the image is masked in such a way that it contains the information of the selected route only. A sample of intermediate image for blue color with the mask and yellow color with mask is shown in Figure 3.



*(a)*

*(b)*

*Figure 3. The sample of intermediate image (a) blue and (b) yellow color with mask*

## Noise Cancellation

The image generated from the previous step contains some noises. For example, the logo of restaurants, hospitals, and others will cause some problems to extract data as these are irrelevant for processing as shown in Figure 4. Manual, fine-tuning of RGB value filtering was enough to obtain a satisfactory level noise-free image. This process includes nothing but manual tweaking of RGB values to eliminate irrelevant information as much as possible which is shown in Figure 5.

## Color detection using RGB value

In this step, the intermediate images (image contains the only blue part, yellow part, etc ) are merged. A sample image after merging is shown below:

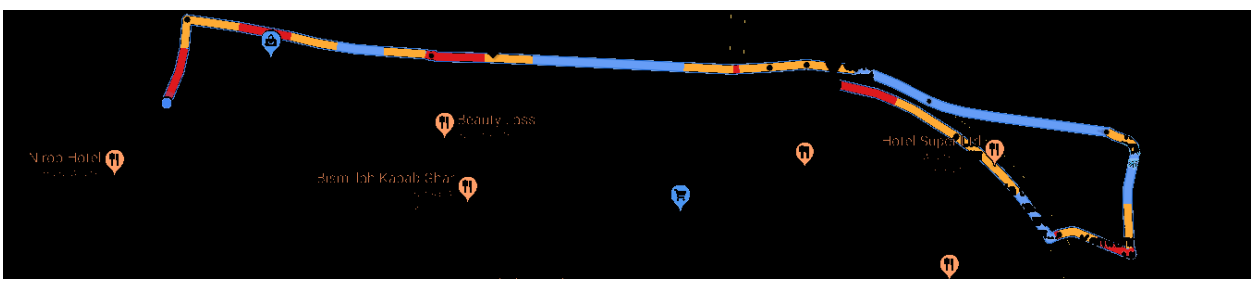


*Figure 4. The snapshot containing noise*

After all types of noise removal the final image looks like the following:

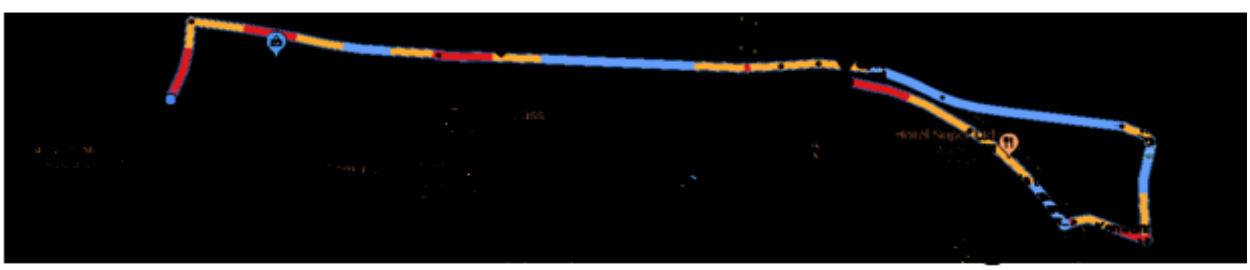


*Figure 5. The snapshot after removing noise*

The Pseudocode for color detection is given below.

| |
|---|
| *n* = number of images in device |

```
for i= 0 → n
        image ← read from local disk
        hsv ← convertColor(image)
        lower_yellow = [10,160,240]
        upper_yellow = [80,250,255]
        mask = part of hsv that lies within the range
        intermediate_yellow = mask

       lower_blue = [100,90,150])
       upper_blue = [145,255,255]
        mask = part of hsv that lies within the range
        intermediate_blue = mask

       lower_yellow = [150,150,150]
        upper_yellow = [255,255,255]
        mask = part of hsv that lies within the range
        intermediate_red = mask

output = overlap intermediate color images
```

## CSV Generation

Once the images with color information are saved into a database they are then batch processed to generate CSV files that can be applied for analysis/prediction. The color information is extracted from the image of the selected route. From the image file, the start and endpoint of the route are selected. Then, the part that contains the route is segmented to assign numeric values of the yellow, blue, red, and dark red color. In this research, the whole route (four hundred) is divided into ten segments where each contains forty pixels and then each segment is labeled according to the dominance of the color they contain. From the analysis, it is observed that the lower size segment provides more accurate color extraction. However, it takes more time to extract color from the image for a smaller segment size. It was observed that a satisfactory level of accuracy was achieved by setting each segment as forty pixels. The CSV file contains numeric values of each segment with time, day, and holiday information. The sample of CSV file is shown in the Table 1.

*Table 1.The Sample of CSV file*

| Day | Time | Is Holiday | Segment 1 | Segment 2 | …….. | Segment 10 |
|---|---|---|---|---|---|---|
| Sat | 11.00 | Yes | 1 | 2 | | 1 |
| Sun | 15.45 | No | 2 | 2 | | 2 |
| Mon | 20.00 | No | 2 | 3 | | 1 |
| Tue | 21.30 | No | 2 | 1 | | 2 |

Color Codes: 1- Blue, 2-Orange, 3-Red, 4-Dark Red

Pseudo code for CSV generation is given below.

```
/* discards the part of image that contains no info*/
```

```
findInit(width, height)
      start = 0
      for i = 0 → width
            for j = 0 → height
                  If mask_red or mask_blue or mask_yellow is nonzero
                      start = 0
                      break
     return start

findInit(width, height)
      finish = 0
      for i = width → 0
            for j = height → 0
                  If mask_red or mask_blue or mask_yellow is nonzero
                      finish = 0
                      break
     return start
start = findInit(width,height)
finish = findFinish(width,height)

Colored_pixels = []
red, green, blue = 0
define red_mask, green_mask, yellow_mask empirically

for i=start → finish
          for j = i → i + 40  /* segment of 40 pixels */
                  for k = 0 → width
                              if mask_red is nonzero red ← red+1
                              else if mask_green is nonzero green ← green+1
                              else if mask_blue is nonzero blue ← blue+1
          write in file max(red, green, yellow)
```

## TRAFFIC FLOW ANALYSIS AND VISUALIZATION USING THE DESIGNED TOOL

Using the developed tool, the traffic information of Dhaka city is collected and visualized. Dhaka is a highly congested city and it suffers huge traffic jams and it does not have a traffic data repository. Thus Dhaka is chosen as a case study. The developed tool is used to collect the traffic information of the selected route. To do this, traffic flow data for three months (1st August 2019- 30 October 2019) is collected and processed using the developed tool. The tool takes a specific route, date, time duration as input and provides the intensity score of the specific route within a given period. As mentioned earlier, the whole road is divided into segments, and intensity is assigned to all these segments individually. Then the intensity score of all the segments is aggregated to find the intensity score of the route in a selected time. Using these three month's traffic data, traffic intensity of different times and days of the week is analyzed.

### Comparison between Weekend and Weekdays

The traffic intensity pattern changes at different days and times of the week, which needs to be investigated for better planning. In Bangladesh, Friday and Saturday are Weekends and Government offices and educational institutes are remained closed on these two days. Also, for weekdays two groups are made, one contains the first weekday (Sunday) and last weekday (Thursday) and the rest of the weekdays (Monday to Wednesday) fall into the second group. The aim is to analyze the effect of the different days on traffic patterns. In Figure 6, the traffic intensity of the different days in August 2019 is depicted. It is shown that from midnight (23:00 am) to the early morning (6:00 am), the traffic intensity of all days is the same. However, traffic intensity varies greatly between weekdays and weekends between the time 6:00 am to 21:00 am. On weekends traffic intensity is much lower than that of weekdays as shown in Figure 6.

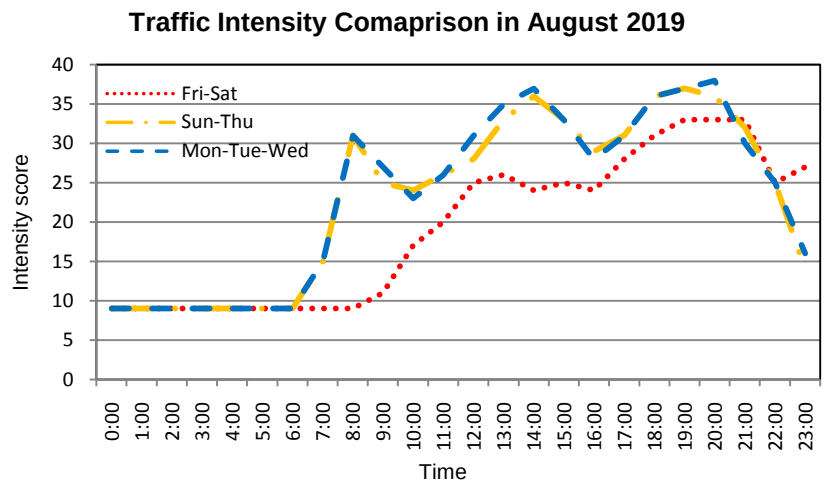



*Figure 6. The effect of traffic intensity on different days of the week (August 2019)*

Among the two groups of weekdays, it is observed that traffic intensity is a little bit higher in Monday to Wednesday group compared to the Sunday, Thursday group. This is because of the socio-economical impact of people's life in Dhaka city. Many people choose Sunday or Thursday to take leave from the office to get three consecutive leave. As a result, Monday to Wednesday group has greater traffic intensity compared to others. A similar pattern of traffic intensity is also observed in the month of September 2019 and October 2019 as shown in Figure 7 and Figure 8.

## Traffic flow prediction

Traffic prediction is the task of forecasting real-time traffic information based on historical traffic data, such as traffic flow, average traffic speed, and traffic incidents. There are many methods to forecast traffic information, among them *K-NN* is a simple data-driven heuristic approach for forecasting. This method makes use of a database to search for data that are similar to the current data and then makes a forecast based on the found data. These found data are called the nearest neighbors of the current data. Three criteria are widely used to evaluate the performance of the traffic forecast model. They are mean absolute error (MAE), mean square error (MSE), and mean relative error (MRE). The definitions of them are given below

$$\text{MAE}=\frac{1}{n}\sum_{i=1}^{n}|\widehat{d_i}-d_i|$$
$$\text{MSE}=\frac{1}{n}\sum_{i=1}^{n}(\widehat{d_i}-d_i)^2$$
$$\text{MRE}=\frac{1}{n}\sum_{i=1}^{n}(|\frac{\widehat{d_i}-d_i}{d_i}|)$$

where, $\hat{d}_i$ is the predicted traffic data and $d_i$ is the collected traffic data.

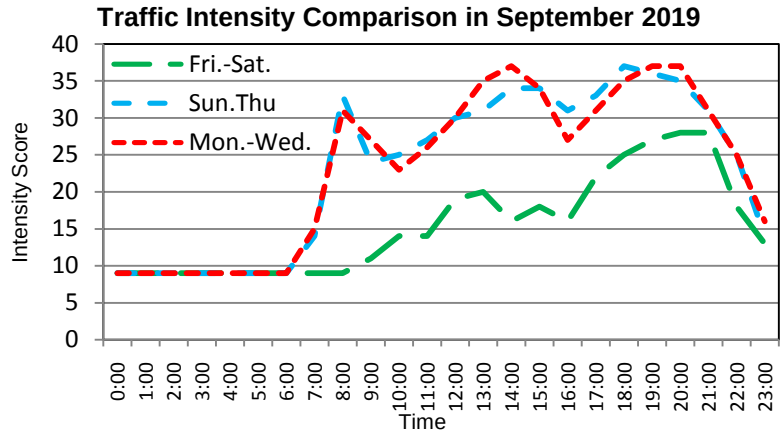



*Figure 7. The effect of traffic intensity on different days of the week (September 2019)*

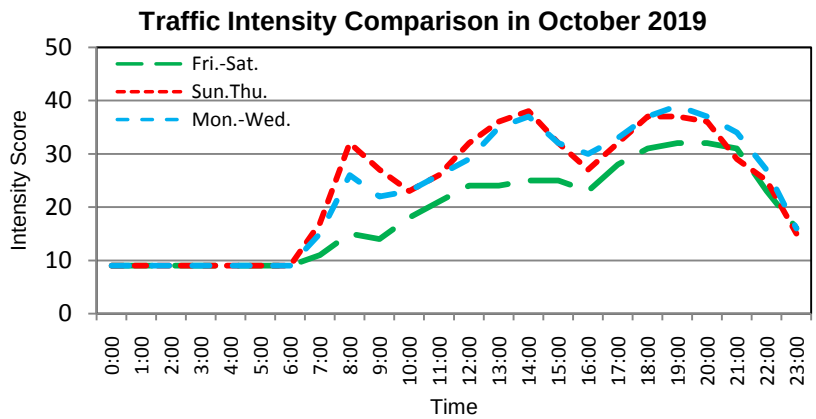



*Figure 8. The effect of traffic intensity on different days of the week (October 2019)*

It is well known that MAE and MSE are more sensitive to raw data while MRE is considered as a suitable criterion for evaluating traffic forecasting models (Zhao et al., 2017) The lower the value, the better the model performance for all these parameters. The Figure 9 shows the comparison of the predicted and actual output of the K-NN algorithm on the month of September and October 2019. The traffic data of the month of August is used for traning. Table 2 shows the result of MRE, MSE, and MAE of the *K-NN.*

*Table 2. Forecast performance of the K-NN in terms of MRE, MSE and MAE*

| | **Mean Relative Error (MRE)** | | **Mean Square Error (MSE)** | | **Mean Absolute Error (MAE)** | |
|---|---|---|---|---|---|---|
| Days/ Year | September | October | September | October | September | October |
| Sun. Thu. | 12.28 | 18.06 | 9.75 | 35.5 | 2.75 | 4.5 |
| Fri.-Sat. | 14.64 | 19.54 | 17.42 | 30.5 | 3.25 | 4.08 |

| | | | | | | |
|---|---|---|---|---|---|---|
| Mon.- Wed. | 14.90 | 13.06 | 17.5 | 17.04 | 3.33 | 2.79 |

## CONCLUSION

This paper develops a tool that collects and processes traffic flow information from any GIS map service. The tool takes the map image of a road with traffic information as an input and produces flow intensity as output which can be used for analysis, visualization, prediction, and many other applications. Dhaka city's traffic information is collected and processed using the developed tool. After processing traffic information, one month's data is used for learning and then *the K-NN* algorithm is applied to predict the traffic information of the next two months. Any other learning algorithm, visualization approach can be applied to extract further knowledge from the data.

*Figure 9. Comparison of traffic forecast on the month of September and October 2019*

## REFERENCES


Ciepłuch, B., Jacob, R., Mooney, P., & Winstanley, A. C. (2010, July). Comparison of the accuracy of OpenStreetMap for Ireland with Google Maps and Bing Maps. In *Proceedings of the Ninth International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences 20-23rd July 2010* (p. 337). University of Leicester.

Hossan, S., & Nower, N. (2020). Fog-based dynamic traffic light control system for improving public transport. *Public Transport*, *12*(2), 431-454.

Jiber, M., Lamouik, I., Ali, Y., & Sabri, M. A. (2018, April). Traffic flow prediction using neural network. In *2018 International Conference on Intelligent Systems and Computer Vision (ISCV)* (pp. 1-4). IEEE.

Lima, A. M., & Campos, J. (2017). How reliable is the traffic information gathered from Web Map Services?. In *GEOINFO* (pp. 234-245).

Liu, Z., Li, Z., Wu, K., & Li, M. (2018). Urban traffic prediction from mobility data using deep learning. *IEEE Network*, *32*(4), 40-46.

Ribeiro, A. I. J. T., Silva, T. H., Duarte-Figueiredo, F., & Loureiro, A. A. (2014, September). Studying traffic conditions by analyzing foursquare and instagram data. In *Proceedings of the 11th ACM symposium on Performance evaluation of wireless ad hoc, sensor, & ubiquitous networks* (pp. 17-24).

Taylor, K. (2017). A Comparison of Feature and Placename Density Between Google Maps and OpenStreetMap. *University of Nebraska at. Omaha*.

Tostes, A. I. J., de LP Duarte-Figueiredo, F., Assunção, R., Salles, J., & Loureiro, A. A. (2013, August). From data to knowledge: city-wide traffic flows analysis and prediction using bing maps. In *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing* (pp. 1-8).

Zhao, Z., Chen, W., Wu, X., Chen, P. C., & Liu, J. (2017). LSTM network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, *11*(2), 68-75.