TẠP CHÍ KHOA HỌC
ĐẠI HỌC QUỐC TẾ SÀI GÒN
Saigon International University Journal of Science

Số 01 - Tháng 10/2025
Website: https://js.siu.edu.vn

URBAN TRAFFIC DATA COLLECTION AND TRAFFIC VIEW APPROACHES

Tran Minh Quang [*]

Abstract: Data quality is crucial in crowd-sourced traffic information systems. While expanding data collection methods is essential, ensuring accuracy and reliability is equally important. Conventional traffic information systems collect data through multiple methods such as web-based forms, speech data, and GPS sensors on mobile devices. This paper enhances these methods by introducing new approaches to traffic data collection from cameras that leverage existing city surveillance infrastructures to provide a continuous stream of real-time traffic data. A novel approach for background scheduling to update traffic status predicted from camera-based data applying AI models is proposed. Then we design necessary components to implement background data fetching scheme to insert traffic camera data into the existing traffic information system which is used to evaluate the proposed approaches in real-world scenarios. The results revealed the effectiveness and the efficiency of the proposed mechanisms showing that they are ready to be applied in real-world applications.

Keywords: Crowd-source; Urban traffic; ITS

1. INTRODUCTION

Data quality is paramount in crowd-sourced traffic information systems (Ha et al., 2020). Modern traffic information systems such as UTraffic (2024) collect data through application forms, speech data, and GPS sensors on mobile devices. Although, these approaches aim at enhancing the coverage of the ITS systems by leveraging mobile user contributions, there still difficulties need to be resolved to reach a suitable incentive model in recruiting mobile users' contribution. Therefore, a mobile crowd-sourcing intelligent transportation system (ITS) such as UTraffic is struggled to maintain a continuous flow of user-generated data. The system often has to resort to the base (background) data instead, which is created based on statistical traffic data (Quang et al., 2022). This work grouped the base velocity of street segments according to time slots, and a traffic status query will refer to the closest slot to the query time. This data was populated using TomTom (2024), and the authoring group managed to make it available to all street segments in Ho Chi Minh City. Obviously, in this approach the real-time traffic information is missed, meanwhile updating this background traffic data required a well-design and huge amount of resources which cannot be conducted frequently. On the other hand, there still are various existing crowd-sourced systems such as surveillance cameras where traffic data can be extracted in real-time.

This paper aims at improving the efficiency and feasibility of the aforementioned methods in real-world applications by introducing new approaches to image-based traffic data collection that leverage existing city surveillance cameras and mobile user contribution. In addition, design and integrating TrafficView function into UTraffic to show and provide real-time traffic conditions from surveillance cameras is a practical requirement. The integration is particularly exciting since it marks a novel approach to UTraffic's data collection scheme. Unlike traditional methods that rely solely on user input, TrafficView opens the pathway to a constant stream of objective traffic data derived from real-time image analysis utilizing AI techniques. This eliminates the potential of user bias or inconsistency often present in crowd-sourced data.

Moreover, a novel approach for background scheduling to update traffic status predicted from camera-based data

[*] Assoc.Prof., University of Technology, Vietnam National University Ho Chi Minh City

applying AI models is devised. This information is not only accurate but is also reliable. By having the means to access and analyze it, the proposed approaches integrated in UTraffic is ready to adapt and extend seamlessly to real-world applications, even should more cameras be installed. The main contributions of this paper can be summarized as follows:
Proposes a novel approach to collect and fuse traffic data from surveillance camera systems which can be extended for collecting traffic data from mobile users' cameras.
Integrate the proposed approaches into a real-world system, namely UTraffic, applied in Ho Chi Minh city, Vietnam. The evaluation results from this system confirm the effectiveness and the efficiency of the proposed mechanisms.
The rest of the paper comes up with a literature review on related work on Sect. 2. The proposed approaches are presented in details in Sect. 3. Sect. 4 describes the implementation and evaluation while Sect. 5 concludes the paper and draws out future work directions.

2. RELATED WORK

Urban traffic monitoring and forecasting are important issues in the intelligent transportation systems (ITS) field that attract academic and industrial research in the recent years (A. Essien et al., 2019). In order to provide accurate traffic information and appropriate traffic forecasting, traffic related data must be collected accurately and in-time. Conventional approaches rely on road-side fixed sensor systems such as radio frequency identification (RFID) and camera systems to extract traffic data. However, these systems are limited in terms of coverage, deployment and maintenance cost, especially in developing countries where traffic infrastructures have not been developed well to catch up the development requirement.
The advances of mobile and IoT technologies allow us to utilize crowd-sourcing approaches to traffic data collection from various sources such probe vehicles, mobile users, radio channels, and existing fixed-side systems (Sciberras & Inguanez, 2018; Steenbruggen et al., 2013; University of Maryland Transportation Studies Center, 1997; Bar-Gera, 2027; Mobile Millennium, UC Berkeley, 2024). Studies in CityDrive (Zhao et al., 2014) and GreenDrive (Zhao et al., 2017) introduced mechanisms to collect vehicles' movements via GPS, then estimate volumes and velocities of traffic flows at particular intersections. These studies reveal the feasibility of providing real-time traffic information and smart driving services using crowd-sourced data. However, there are many issues related to GPS errors and the user willingness in utilizing these systems have not yet been thoroughly analyzed.
Beside GPS based approaches, mobile users can report traffic condition via application forms or speech reports on mobile apps (Quang et al., 2022a; Quang et al., 2022b; Quang & Huu, 2021). Numerous studies have focused on collecting, analyzing, and evaluating traffic status from various sources including from mobile users and the surveillance camera systems such as VTIS (Vietnam Traffic Information System) (VTIS, 2024) and UTraffic (2024). These web-based systems, including a mobile app for Android to collect traffic data from traffic channel channels such the VOV (The Voice of Vietnam) on FM91 MHz (VTIS, 2024) and the VOH (The Voice of Ho Chi Minh City) on FM95.6 MHz, 2024). Mobile users and commuters can report traffic condition that they observe by putting the data into the mobile app's form and send to the server or call to the radio station to directly report the traffic information. In order to enhance the voice traffic data processing, the study in Ty & Minh (2023) proposes solution to extract traffic data utilizing ASR (Automatic Speech Recognition) mechanisms.
As a solution to the coverage issues, both the mobile crowd and fixed sensors systems should be utilized for traffic data collection. One of the reliable of traffic data source is the traffic surveillance camera system which is originally deployed for manually monitoring traffic condition. With the development of

artificial intelligence, neural network models are widely used in traffic analysis from surveillance video such as Region-based Convolution Neural Networks (R-CNN) model (Arinaldi et al., 2018) or the single shot based approaches such as SSD (Single Shot MultiBox Detector) (Liu et al., 2016) or YOLO (You Only Look Once) (Redmon et al., 2016) for vehicle detection. From then traffic density can be estimated. However, these approaches have not thoroughly analyzed the difficulties and appropriate solutions for collecting traffic data from a large number of cameras to make the solution available in real-world applications.

In this current work, we will present and analyze techniques to collect camera data, extract traffic data from them and visualize

A them as the map of large urban traffic network. The proposed approaches are also implemented in mobile and web-based systems used for real-field experiments toward real-world applications.

3. PROPOSED METHOD

A. Designs for a background schedule to update traffic status from city surveillance cameras

Firstly, this work's core is extracting traffic data from city's surveillance cameras whose uptime are excellent, and that the information they portray are highly reliable. Here, we need to determine main components and functionalities that should be integrated into the UTraffic (2024). According to our real-field observation and analysis, when inspecting a road segment, users want to obtain two key pieces of information, which are its traffic flow's velocity and the source of the report since UTraffic fuses data from various crowd-sources which may reveal the argues on the source reliabilities.

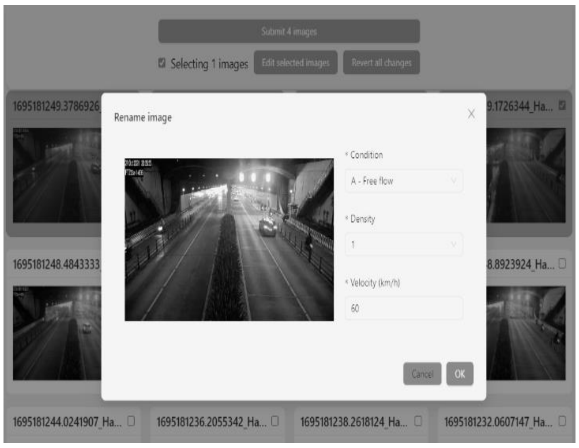
Obviously, from the camera data we can devises AI-based models for predicting traffic related information such as traffic flow velocity, density, and overall condition based on the level of service (LOS) (Tan et al., 2021). The prediction model is briefly described as follows.

Images of traffic are extracted from surveillance cameras which are sent to the server where AI/ML models are implemented to predict traffic condition. The resulted traffic information is sent to commuters for their daily activities such as routing for their travel and to the traffic management for policy and decision making. In addition, to train the AI/ML models, we need to prepared training dataset by having volunteers labeling the images and store them in a database system.

This work aims at predicting traffic condition using three indicators: the LOS, density and traffic flow velocity. Therefore, each image will be labeled according to these classes. Here, LOS is labeled in A, B, C, ..., F describe as follows:

- A (Free flow) when the velocity is greater of equal to 35km/h
- B (Reasonable free flow) when the velocity is from 30km/h to 35km/h
- C (Stable flow) when the velocity is from 20km/h to 30km/h
- D (Heavy flow Approaching unstable flow) when the velocity is from 12km/h to 20km/h
- E (Lightly congested flow) when the velocity is from 5km/h to 12km/h
- F (Congested flow) when the velocity is from 5km/h to 5km/h

In addition, Density of the traffic flow represented in the image is labeled in 0, 1, 2, 3, 4 ranging from the best to the worst traffic condition, respectively. Meanwhile, traffic flow velocity is estimated in a real number representing the speed (e.g., 20 km/h) of a traffic flow. For example, when the street is free the speed can be 40, 50 or 60 km/h, and the speed at a heavy road could be 15 - 20 km/h. For example, the image in Fig.1 is labeled as: LOS = A, Density = 1 (base on the right lane), and speed = 60km/h.



Rename image

* Condition: A - Free flow

* Density: 1

* Velocity (km/h): 60

Figure 1: Labelling traffic condition based on LOS, velocity and density

As for the traffic prediction, we have applied various deep learning approach dedicated to image processing and evaluate their effectiveness to find out the most appropriate one for real-world application. In this work, we have selected 5 modern models for study which are Selt-attention+YoloBox, EfficientNet, VGG16, GoogleNet/Inception_v1, Resnet. The experiment results of these methods are shown in the evaluation section.

To integrate this functionality seamlessly into UTraffic while maintaining a lightweight background fetch mechanism, we propose utilizing only the velocity model. This choice aligns with the data visualization approach adopted by UTraffic's web client, which primarily focuses on displaying real-time traffic flow velocities. The following are the reasons why focusing on the velocity model is a strategic decision:

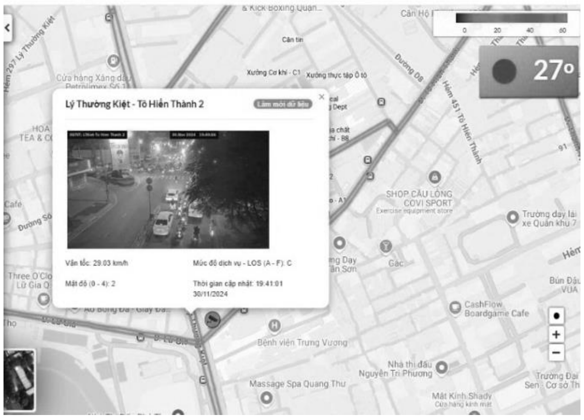
- **Efficiency**: Processing traffic camera images through all three models (velocity, density, and condition) would be computationally expensive for a background schedule. The velocity model alone offers a sufficient data point for our purposes.
- **Data redundancy**: UTraffic's existing system already incorporates preconfigured logic to convert velocity data into corresponding density and condition levels. Therefore, directly retrieving velocity information eliminates the need for redundant calculations within our background fetch process.

B. Inserting traffic camera data into the existing system

A critical aspect of this work involves seamlessly integrating traffic camera data into the existing UTraffic system which analyzes traffic from multiple data sources. It is worth noted that hovering over a traffic status polyline the current traffic flow velocity along with its data source provides more information and reliabilities to users. Therefore, we propose incorporating traffic camera data as an additional data source whose level is the same as the original mobile crowd-sourced data. This approach aligns with the established logic of the UTraffic application, eliminating the need to develop entirely new data processing pathways.

To add the traffic camera status to the tarffic information flow in UTraffic, we applied AI mechanism to predict traffic flow velocity, and then reuse the reporting functionality intended for crowd-sourced data which is already implemented in UTraffic. The original, form-based method receives starting and ending pairs of coordinates so as to determine the correct way of the segment to be updated. Due to most streets being in two ways, this is a necessary approach. However, we do not have such a luxury for a camera image, since:

- The camera has no knowledge of how the lanes it observes is divided. For example, Fig. 2 shows the view at an intersection between in Ho Chi Minh City. The main traffic flow is in two lanes, while a third is waiting near the right corner, presumably for a red light. The model simply counts the vehicles that are within the view and generates a prediction for the velocity accordingly.



Lý Thường Kiệt - Tô Hiến Thành 2

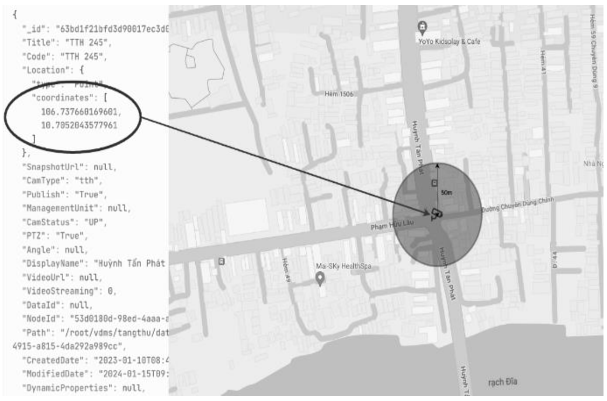
Vận tốc: 29.03 km/h
Mức độ dịch vụ - LOS (A - F): C
Mật độ (0 - 4): 2
Thời gian cập nhật: 19:41:01 30/11/2024

Figure 2: A camera enables the view to an entire street and has no knowledge of the radius it covers or the lanes in view.

- The camera only covers a small field of view, and this value differs depending on the individual camera. Moreover, an angle value exists for some cameras as we inspect the data, but largely vacant otherwise.

This angle could be the value formed by the meridian going through the center of the camera with the line beginning from its center of view. The value itself could be valuable in pinpointing the segments whose status will be updated by the camera's prediction outputs. However, the number of cameras missing this field is accounting for 33% (177 cameras) of the total cameras in the system. Manually labeling this field is an option, but given the time limit, we decided against it.

To make the integration possible, we accept the cameras' location themselves as the sole coordinates for the report instead. This also allows us to utilize an existing functionality of the system, which is finding nearby segments of a given pair of coordinates. The velocity data extracted from the camera image using the AI model is then used to update the traffic status for each of the identified segments. This approach ensures that the real-time traffic flow information captured by the camera is effectively reflected in UTraffic. The reason for us to accept a maximum of 100 segments is because there are many street segments levels that we need to account for. Having a lower number can accidentally leave out a set of street segments of different levels, which can affect the status when viewing the map on different zoom levels. Figure 3 clarifies the method.



```
{
  "_id": "63bd1f21bfd3d90017ec3d6
  "Title": "TTH 245",
  "Code": "TTH 245",
  "Location": {
    "type": "Point",
    "coordinates": [
      106.737660169601,
      10.7052043577961
    ]
  },
  "SnapshotUrl": null,
  "CamType": "tth",
  "Publish": "True",
  "ManagementUnit": null,
  "CamStatus": "UP",
  "PTZ": "True",
  "Angle": null,
  "DisplayName": "Huỳnh Tấn Phát
  "VideoUrl": null,
  "VideoStreaming": 0,
  "DataId": null,
  "NodeId": "53d0180d-98ed-4aaa-a
  "Path": "/root/vdms/tangthu/dat
4915-a815-4da292a989cc",
  "CreatedDate": "2023-01-10T08:4
  "ModifiedDate": "2024-01-15T09:
  "DynamicProperties": null,
```

Figure 3: With the camera's location field, we can determine the segments within a 50-meter radius circle whose center is the camera. The status extracted from the camera will be applied to the segments within this circle.

C. Camera traffic data update in large cities

Our consistent goal is to expand the status mapping in a large scale aiming at applying to the whole Ho Chi Minh City area. To extend the operation to all the cameras in the city, we perform a query to the camera collection in the database for those that are active resulting in an array of cameras. Then, their IDs are extracted, and we process them linearly to make calls to the computing server responsible for predicting the traffic flow velocities. We generate a traffic status report as soon as a velocity value is available, and then we move on to the next. A try-catch block is placed on this to prevent cases where the operation might fail, or if there is no segment to update. Therefore, between two consecutive cameras, the operation for the previous one is guaranteed to terminate before the latter, avoiding cases where a failed update could invalidate all the pending cameras. This also alleviates memory pressure, because the data is written as soon as it is generated without being stored in memory.

Our initial approach to fetching traffic camera data relied on a 5-minute cron job where the backend application calls the update function and performs the mapping. However, considering the extensive camera network in Ho Chi Minh City, this approach presented scalability limitations.

a) Challenges of fixed-interval fetching

With iterative camera processing, assuming a 1-second processing time per camera, fetching data for all cameras would require approximately 403 seconds (total cameras times the processing time per camera). This exceeds the 5-minute cron job window, resulting in an incomplete data update cycle.

While updates are written as generated, the loop terminates at the 5-minute mark, creating a "dead zone" for unprocessed cameras at the end of the queue, as shown in Fig. 4. Restarting the loop before those cameras are visited perpetuates this issue. Extending the cron job interval to 10 minutes would not solve the problem either. Assuming a 2-second processing time per camera, the total processing time would become 806 seconds, again exceeding the update window.

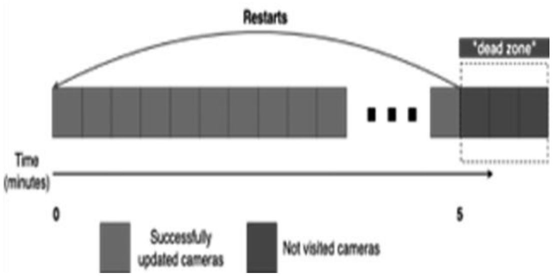


Figure 4: The iterative update of traffic status based on camera and a strict time window reveals a "dead zone" when the loop resets before the cameras in this zone can be visited for traffic data extraction.

b) A self-calling function for improve efficiency

To address these limitations, we re-investigated UTraffic's reporting logic and found that each segment status report has a 30-minute validity period (Quang et al., 2022). When a client requests traffic status updates via the API, the system checks for the validity of traffic status reports that are present for the segments included in the response. If these reports are expired, the system retrieves data from the base source instead. While the system does update its cache every 5 minutes, this action does not invalidate the report yet as it only scans whether new reports are submitted to UTraffic. We do not need to worry that the traffic camera data reports will expire after this 5-minute window, since their validity are not determined by it. As a result, camera data can persist for up to 30 minutes like any other report submitted by mobile users. This characteristic allows us to adopt a new strategy in which the background update function will call itself once it finishes. This is akin to recursion (Causey., 2006), where a function is applied within its own definition.

c) Schedule the traffic camera data background fetching

Rush hours in HCMC set the basis for our schedule, rush hours for road traffic are from 6:00 to 8:00 in the morning and from 16:00 to 19:00 in the evening, daily (Ha et al., 2020). Therefore, we set up the background fetch function to run as follows:

- **Rush hours (6:00 – 8:00 and 16:00 – 19:00):** The function restarts immediately after a round trip, with no delay between consecutive cameras.
- **Night (23:00 - 6:00):** The function is disabled.
- Other times: The function restarts immediately, but with a 5000ms delay between consecutive cameras.

The schedule is controlled by the function itself with the help of a new cron job. At 6:00 daily, the cron job will start its execution. Then, after the function has completed a round trip, it checks whether the current time is still between the rush hour blocks. If yes, then the function restarts itself or does so with the introduction of the middle delay otherwise. If the time is between the nighttime block, the fetch will skip the recall part completely, and stops its execution. When it gets to 6:00 again the next day, the whole process restarts similarly.

To this point, the automated data acquisition method is complete. Next, we move on to how the study on cameras power user-submitted images as a new source of community data.

D. User-submitted images as crowd-source data

TrafficView introduces exciting possibilities for enhancing UTraffic's data collection through user-submitted images. This functionality allows users to capture real-time traffic conditions directly from their phones and upload them to the system, complementing existing traffic report methods via forms, speech recognition, and GPS sensors. We need to assess the current method for gather crowd-sourced data and then compare that with user-submitted images. Traditionally, reports for traffic are done with a form submission, as seen in Fig. 5 with the following fields:

- **Starting and ending coordinates:** The form accepts two pairs of coordinates in order to determine the correct segment as well as the way traffic is meant to be reported. Since the average segment length is 100 meters (Quang et al., 2022), the maximum distance allowed between two coordinates is also 100 meters.
- **Velocity:** The average velocity that the user identifies for the reported location.

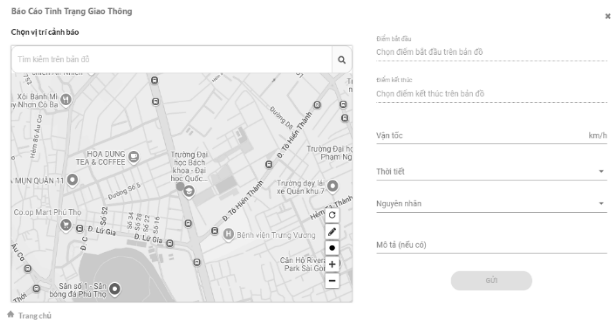


Figure 5: A traffic status report form where mobile user must specify a starting location (1), an ending location (2), and the velocity (3).

Such a form is simple, user intuitive and effective in gathering data. On top of the velocity, they can report extra details including the cause of the traffic congestion, the weather condition, and other description if they wish to. However, a form requires manual input and may not be convenient for users to interact with while they are on the move, potentially disrupting traffic themselves.

To make it easier for crowd-sourced data to be gathered, a background fetch using GPS sensors was developed for Android devices (Quang et al., 2022). This feature also included a sleep/wake-up mechanism, breakthrough helping in limiting sensor polling and save energy. The disadvantage of this method is the difficulty of implementation on iOS devices. iOS places a strict limit on background execution, which renders the feature not working on iPhone devices.

With user-submitted images, we would like to not only introduce a novel method of crowd-sourced data but also resolve the shortcomings of the existing methods as follows:

- **Streamlined contribution:** Users simply submit a photo taken directly within the application, minimizing steps compared to manual form filling.
- **Reduced user ambiguity:** The existing methods require the user to specify the velocity values themselves, whose drawback is that the same traffic flow can be identified differently between two separate users. This leads to discrepancies between the actual situation and the report, and consequently, lessen the submission's quality.
- **Simplified processing:** Taking a photo and submitting in requires less processing compared to the audio counterpart (Utraffic, 2024; Dolby, 2024). In a traffic congestion environment, for example, the background noise can interfere with the report and require more processing effort to identify correct words. On the other hand, an image is free from such a noise, and the current system can process imagery data right away.

Nonetheless, image submission depends largely on external factors, such as the angle of the situation, lighting conditions, and field-of-view. To solve this issue, the image processing model should be a prime target of improvement in later UTraffic's development effort.

While the current system discourages irrelevant submissions by disabling camera roll access, further filtering mechanisms are still required to address potential misuse. User-submitted traffic reports via images have a temporary visibility window of 15 minutes. After this period, the status of affected segments reverts to the data obtained from primary sources.

Similar to city traffic cameras, user-submitted images are processed to determine the location and nearby traffic segments. To ensure accurate location data, access to location services is required. This step was done in the on-boarding phase where the app asked the user for certain permissions. Then, users simply capture a photo, which is encoded in Base64 format and sent to the server for analysis. A user may submit an image that are too distant from a segment, for instance. In normal circumstances, this results in a rejected report, but for camera images, we allow the markers representing every report to be visible on the map anyway. Overall, user-submitted images, along with the automated schedule, have given UTraffic a lot more flexibility in its effort to gather data to the system.

4. EVALUATION

The proposed system has been deployed for real-world application in HCMC traffic estimation and management. Real-field data generated from it are used to evaluate the performance of the proposed approaches. In order to confirm the efficiency of the proposed approaches the response time of main functionalities, the required memory (RAM), the system's fault tolerance with large numbers of concurrent requests under limited cloud resources, and the energy consumption have been evaluated.

Table 1: Execution time measurements for UTraffic iOS

Event	**Execution time (ms)**
Lauch the app	444
Map view's first display	292
Display 200 segments status	340
Display 500 segments status	420
Routing with a 2km distance	290
Routing with a 4km distance	385
Draw 10 cameras markers	135
Drawing camera image with predictions in modal bottom sheet	690

Firstly, we measured the execution time of main UTraffic iOS features in Table 1. The table reveals that most of the main functionalities' responses in less than a half of second, while the longest delay time is for drawing camera image with predictions in modal bottom sheet which takes 690ms which is short enough.

In addition to the response time, the memory (RAM) usage of the mobile application was evaluated to confirm its runtime efficiency as shown Table 2. Here, the most RAM consumption is to draw 30 camera makers on the traffic map which accounted for 223MB, while the application averages at 185 MB under continuous execution.

Table 2: Peak RAM usage recorded for various Utraffic iOS tasks

Event/Task	Memory usage (MB)
Device idle	135
Display 200 segments status	148
Display 200 segments status	170
Routing with a 2km distance (from 2 to 4 intersections)	165
Routing with a 4km distance (from 3 to 6 intersections)	173
Drawing 10 cameras markers on the map	160
Drawing 30 cameras markers on the map	**223**

In addition, RAM consumed by UTraffic is less than the average value from off-the-shelf mobile apps as it consumes 85MB and 182MB in the background and the active modes, which are much smaller than RAM consumed (on average 250MB) by off-the-shelf mobile apps.

Energy consumption - one of the most important indicator for sustainable computing - of Utraffic mobile app was also evaluated. This indicator does not only show how the application drains the battery (a limited resource) but also the heat of the device while operating. We consulted the device's battery usage levels to compare it with other running applications. In this evaluation Utraffic has worked on active mode for 45 minutes and 22 minutes in background mode taking 1 hour and 7 minutes in total, but the system was not able to account any battery percentage that UTraffic consumes on the iOS device in a 24-hour cycle. Meanwhile, other applications such as App Store, Messenger accounted 1% and 2% battery usage, respectively. It means that UTraffic is efficient in term of battery usage compared to off-the-shelf mobile applications. The system fault tolerance was also evaluated under a large number of concurrent requests to the server using MonkeyRuner (2024). The results revealed that the proposed mobile app (UTraffic) passed 50,000 concurrent requests satisfying our design purpose.

The experimental results from the real-world deployed system presented above have strongly confirmed the feasibility, effectiveness and the efficiency of the proposed approaches, which are ready for practical applications.

5. CONCLUSION

This paper proposed a novel approach to urban traffic data collection where surveillance cameras are utilized as important source for traffic information. Various approaches have been devised to fuse the camera data efficiently which in turn is processed by AI mechanisms to predict traffic flow velocity. The proposed approaches have been implemented in mobile and web-based systems for real-world applications. The experimental results strongly confirmed the efficiency of the proposed approaches. In the future, we will conduct further research on encouraging mobile users to contribute real traffic data to complement with the camera data for traffic analysis.

ACKNOWLEDGEMENTS
This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grand number DS2024-20-01

REFERENCES

Arinaldi, A., Pradana, J. A., & Gurusinga, A. A. (2018). Detection and classification of vehicles for traffic video analytics. *Procedia Computer Science, 144*, 259–268. https://doi.org/10.1016/j.procs.2018.10.528

Bar-Gera, H. (2007). Evaluation of a cellular phone-based system for measurements of traffic speeds and travel times: A case study from Israel. *Transportation Research Part C: Emerging Technologies, 15*(6), 380–391. https://doi.org/10.1016/j.trc.2007.03.003

Causey, R. L. (2006). *Logic, sets, and recursion.* Jones and Bartlett Publishers.

Dolby. (2024). Introduction to media APIs. *https://docs.dolby.io/media-apis/docs/* (truy cập 2025)

Essien, A., Petrounias, I., Sampaio, P., & Sampaio, S. (2019). Improving urban traffic speed prediction using data source fusion and deep learning. *In Proceedings of the IEEE International Conference on Big Data and Smart Computing (BigComp)* (pp. 1–8). IEEE. https://doi.org/10.1109/BIGCOMP.2019.8679236

Ha, M. T., Hoang-Nam, P. N., Long, N. X., & Quang, T. M. (2020). Mining urban traffic condition from crowd-sourced data. *SN Computer Science, 1*, Article 225. https://doi.org/10.1007/s42979-020-00234-7

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., & Berg, A. C. (2016). SSD: Single Shot MultiBox Detector. In E. Leibe, B. Schiele, J. M. Klein, & K. M. R. (Eds.), *Computer Vision – ECCV 2016* (pp. 21–37). Springer.

Minh, Q. T., & Huu, P. N. (2021). Crowd sourced data organization and analytics for urban traffic estimation. *In Proceedings of the IEEE International Conference on Data Analytics for Business and Industry (ICDABI)* (pp. 90–94). IEEE. https://doi.org/10.1109/ICDABI53623.2021.9655848

Mobile Millennium. (2024). Mobile Millennium project. *University of California, Berkeley. http://traffic.berkeley.edu/project*

MonkeyRunner. (2024). MonkeyRunner. *Android Developers*. *https://developer.android.com/studio/test/monkeyrunner*

PyTorch. (2024). PyTorch. *https://pytorch.org/* (truy cập 2025)

Quang, T. M., Phat, N. H., & Tsuchiya, T. (2022a). Mobile crowd-sourced data fusion and urban traffic estimation. *Journal of Mobile Multimedia, 18*(4), 1035–1062. https://doi.org/10.13052/jmm1550-4646.1848

Quang, T. M., Tan, D. P., Hoang, H. N. L., & Nhat, M. N. (2022b). Effective traffic routing for urban transportation capacity and safety enhancement. *IATSS Research, 46*(4), 574–585. https://doi.org/10.1016/j.iatssr.2022.10.001

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 779–788). IEEE. https://doi.org/10.1109/CVPR.2016.91

Sciberras, R., & Inguanez, F. (2018). Road traffic flow estimation via public IP cameras. *In Proceedings of the IEEE 8th International Conference on Consumer Electronics – Berlin* (pp. 1–5). IEEE. https://doi.org/10.1109/ICCE-Berlin.2018.8576207

Steenbruggen, J., Borzacchiello, M. T., Nijkamp, P., & Scholten, H. (2013). Mobile phone data from GSM networks for traffic parameter and urban spatial pattern assessment: A review of applications and opportunities. *GeoJournal, 78*, 217–232. https://doi.org/10.1007/s10708-012-9447-2

Tan, H. M., Nguyen, H. N. P., Phat, N. H., & Quang, T. M. (2021). Traffic condition estimation based on historical data analysis. *In Proceedings of the 8th International Conference on Communications and Electronics (ICCE)* (pp. 256–261). IEEE. https://doi.org/10.1109/ICCE52940.2021.9502447

The Voice of Ho Chi Minh City (VoH). (2024). Urban traffic channel FM 95.6 MHz. *http://voh.com.vn* (truy cập 2025)

TomTom. (2024). Introduction TrafficAPI. *TomTom Developer Portal*. *https://developer.tomtom.com/traffic-api/documentation/product-information/introduction*

Ty, N. T., & Minh, Q. T. (2023). Research and develop solutions to traffic data collection based on voice techniques. *In Intelligence of Things: Technologies and Applications, Lecture Notes on Data Engineering and Communications Technologies* (Vol. 187, pp. 70–80). Springer.

University of Maryland Transportation Studies Center. (1997). Final evaluation report for the CAPITAL-ITS operational test and demonstration program. *University of Maryland.*

UTraffic. (2024a). Urban traffic information system. *https://bktraffic.com/utraffic/* (truy cập 2025)

UTraffic. (2024b). Voice data contribution. *https://bktraffic.com/home/collect-data* (truy cập 2025)

VTIS. (2024). VTIS homepage. *https://vtis.vn/*

Zhao, Y., Li, S., Hu, S., Su, L., Yao, S., Shao, H., Wang, H., & Abdelzaher, T. (2017). Greendrive: A smartphone-based intelligent speed adaptation system with real-time traffic signal prediction. *In Proceedings of the 8th ACM/IEEE International Conference on Cyber-Physical Systems* (pp. 229–238). IEEE.

Zhao, Y., Zhang, Y., Yu, T., Liu, T., Wang, X., Tian, X., & Liu, X. (2014). City-drive: A map-generating and speed-optimizing driving system. *In Proceedings of the IEEE INFOCOM* (pp. 1986–1994). IEEE.

URBAN TRAFFIC DATA COLLECTION AND TRAFFIC VIEW APPROACHES

Trần Minh Quang[*]

Tóm tắt: Chất lượng dữ liệu đóng vai trò thiết yếu trong các hệ thống thông tin giao thông dựa trên nguồn dữ liệu cộng đồng. Bên cạnh việc mở rộng các phương pháp thu thập dữ liệu, việc đảm bảo tính chính xác và độ tin cậy của dữ liệu là điều không thể thiếu. Các hệ thống thông tin giao thông truyền thống thu thập dữ liệu thông qua nhiều phương thức như biểu mẫu trực tuyến, dữ liệu giọng nói và cảm biến GPS trên thiết bị di động. Nghiên cứu này nâng cao các phương thức đó bằng cách giới thiệu các phương pháp mới thu thập dữ liệu giao thông từ camera, tận dụng hạ tầng giám sát đô thị hiện có để cung cấp dòng dữ liệu giao thông thời gian thực liên tục. Một phương pháp mới về lập lịch nền nhằm cập nhật trạng thái giao thông được dự đoán từ dữ liệu camera dựa trên mô hình trí tuệ nhân tạo (AI) được đề xuất. Tiếp theo, chúng tôi thiết kế các thành phần cần thiết để triển khai cơ chế thu thập dữ liệu nền, nhằm tích hợp dữ liệu từ camera giao thông vào hệ thống thông tin giao thông hiện có, qua đó đánh giá các phương pháp đề xuất trong các kịch bản thực tế. Kết quả cho thấy hiệu quả và tính hiệu suất của các cơ chế đề xuất, chứng minh rằng chúng sẵn sàng được áp dụng trong các ứng dụng thực tế.

Từ khóa: crowd-source; urban traffic; ITS

[*] PGS., Trường Đại học Bách khoa, Đại học Quốc gia Thành phố Hồ Chí Minh