Developing a dashboard platform for smart city video survillance

Youssef Yasser Hussein (*Media Engineering Technology*) and **Prof. Mohammed Salem** (*Digital Media Engineering*)

The German university in Cairo

youssef.awadallah@student.guc.edu.eg

Safety and Privacy are being the most important features nowadays that people always seek to have in their neighbourhoods. Smart cities are doing some effort to offer all what residents need so they can offer them the luxurious and easy life that residents are dreaming about.Safety and privacy are being offered nowadays by some of the smart cities but not as the easiest way that can be possible. Therefore, in this work, the task of offering information and a brief about what is going on around the neighbourhood is being tackled by the development of a dashboard. The dashboard, web tool, was developed to show some graphs and charts and summaries for residents, admins and security supervisors which is available 24/7 for the users to get live information about what's going on in their city. Pre-trained algorithms models were used to detect faces, fire detection and crowd count from the cameras recordings all over the city.

Literature Review

Triangulum city dashboard introduced by M. Farmanbar et al. (2020) [1] proposes a cloud-based interactive web-based data visualisation and data analytics toolkit which is supported by big data aggregation tools. The proposed system offers some features like visualisation of the data and real time monitoring of the city and also representing a big amount of dataset on a normal web browser. Moreover, the system is also capable of answering queries and generating graphics from multiple resources. The proposed system passes through three main steps: data layer, application layer and lastly, presentation layer.

Data layer is the process of generating data and collecting them from sensors, actuators and smart meters in the city. However sensors, actuators and smart meters weren't used by this system, the system just collected data from external sources. The system was relying on Triangulum data providers that just return CSV or XML/JSON files from the APIs. Then the submitted data from providers are then extracted from the cloud by the system to go further with the application layer.

The application layer then do some processing on the data collected in data layer to make it ready to be represented and to pass it the representation layer. In the application layer a major data processing functions are made including filtering and sampling, etc. Then the presentation layer is the graphic user interface that is responsible to display the information related to the services.

For the dataset, Bus transport data was collected from Kolumbus. The data set contained the schedules for 186 buses between the Rogaland county bus stop points and 85 routes.

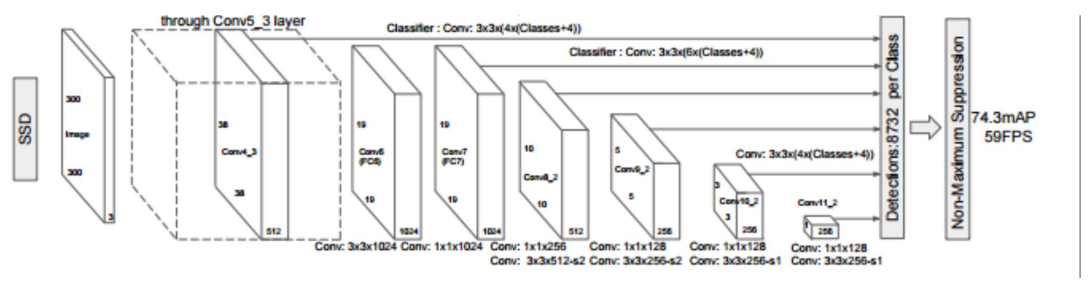
VialisTraffic was used to collect data for real time parking vacancy information located in Eindhoven.

Lastly, the data of energy consumption was collected from LYSE Energy for 56 houses.

In brief, the proposed smart dashboard prototype is robust enough that it can provide various use case scenarios for different smart cities but the limitation is that the proposed system is not using machine learning models to gain the data from video recordings.

Methodology

For face detection pre-trained model HAAR Classifiers were used to detect faces from the video recordings of the security cameras. While, for crowd counting pre-trained model SSD-mobileNet was used to count the pedestrians passing by the region of interest in video recordings and pass the count to the API to be displayed later on the dashboard. Moreover, vehicle counting pre-trained model was also included in this project and the vehicle counting pre-trained model also uses SSD-mobileNet[3] for counting vehicles in the scene of the video recording.



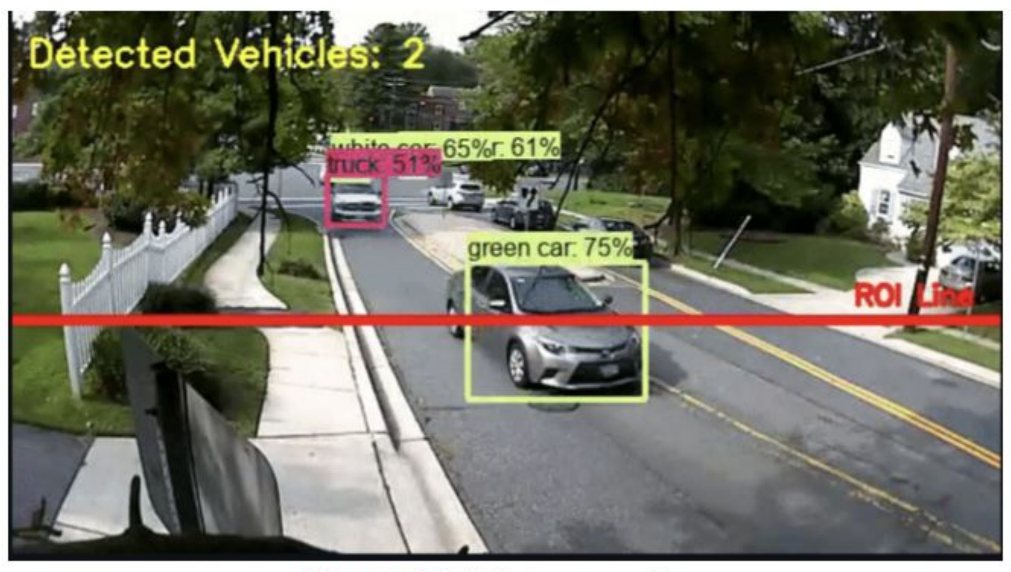
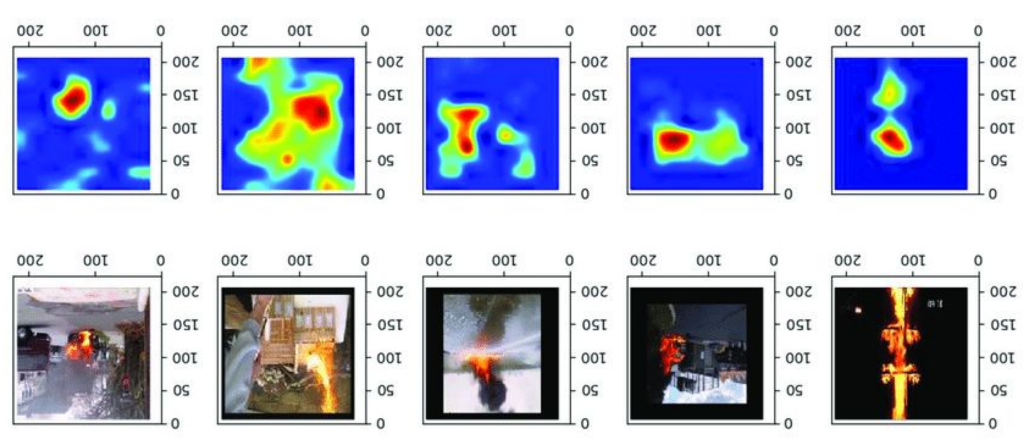
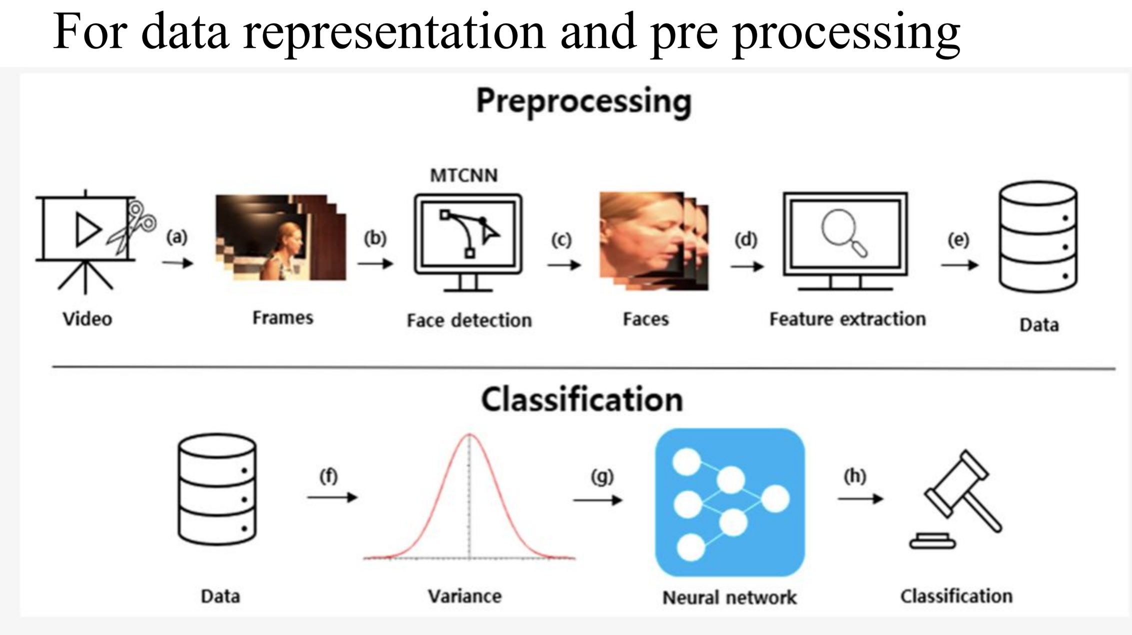


Figure 2: Vehicles counting

Lastly, the fire detection pre-trained model is a YOLOv3 model and it get fed the cameras recording and automatically detect if there is any fire in the scene. FireNet [2] was used as the fire detection model.

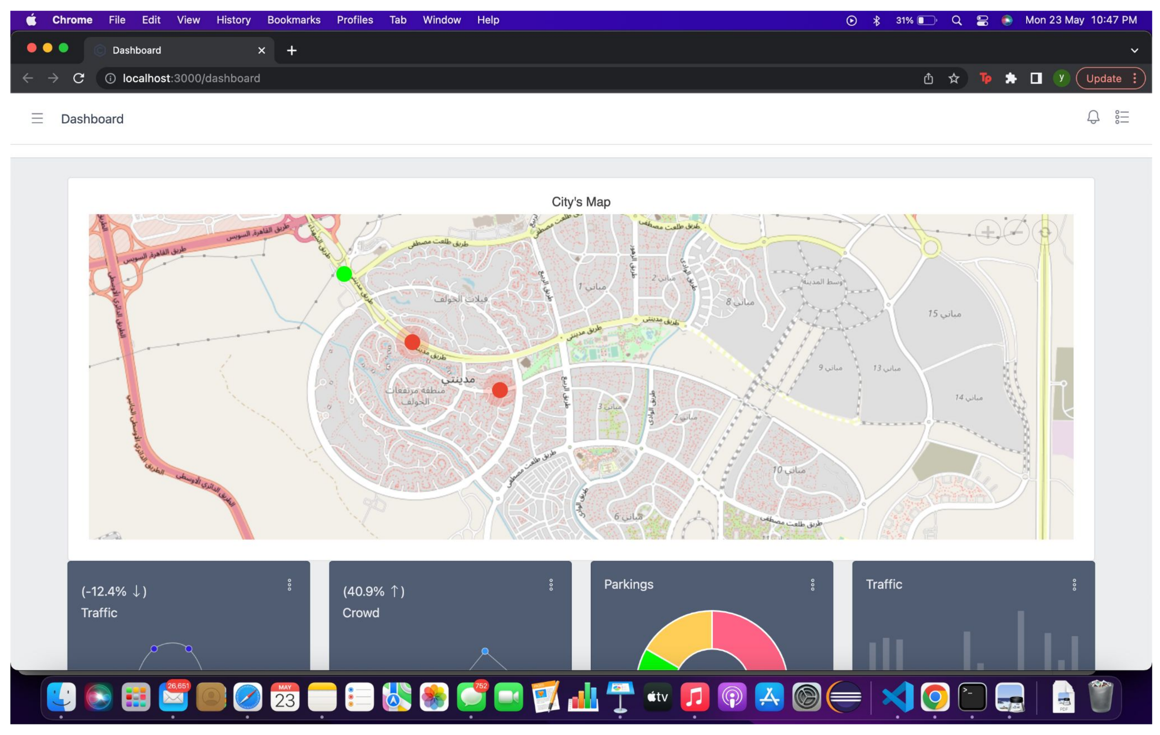


For data representation and pre processing

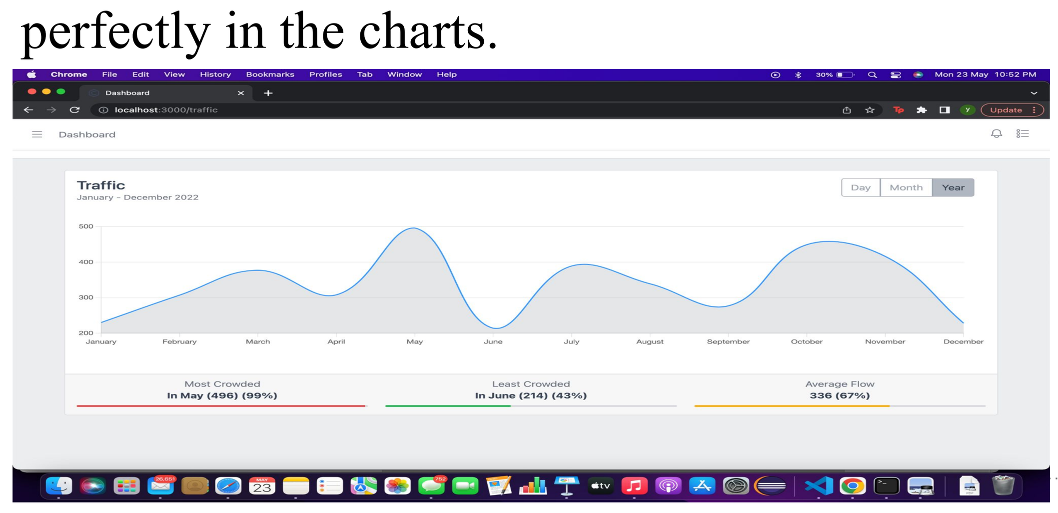


After the pre-processing step being done as shown in figure 3, the data that was fed from the video cameras all over the city to the pre-trained models in the backend are classified and detected then stored on a cloud database to be represented by graphs and charts on the dashboard to be easily represented and read by the users who will use the interface so they can easily understand and read the information being represented. TensorFlow and OpenCV libraries were also used in the pre- processing step as these libraries have a huge importance in computer vision.

Results



The pre-trained model SSD-MobileNet [3] is having an mAP (mean average pre- cision) of 67% and for YOLOv3 FireNet [2] pre-trained model has an accuracy of 93.91%. On the other side, the front end is already getting the information from the API and representing them perfectly in the charts.



References

[2] M. Shorfuzzaman, M. S. Hossain, and M. F. Alhamid, "Towards the sustainable development of smart cities through mass video surveillance: A response to the covid-19 pandemic," Sustainable Cities and Society , vol. 64, p. 102582, 2021.

[3] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in Proceedings of the IEEE conference on computer vision and pattern recognition , pp. 3339–3348, 2018.

[4] W. Fang, L. Wang, and P. Ren, "Tinier-yolo: A real-time object detection method for constrained environments," IEEE Access , vol. 8, pp. 1935–1944, 2019.

[5] S. Zhai, D. Shang, S. Wang, and S. Dong, "Df-ssd: An improved ssd object detection algorithm based on densenet and feature fusion," IEEE access , vol. 8, pp. 24344–24357, 2020.

Prepared for Thesis Poster Display Conference

11^{th} -12^{th} June 2022