International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 09 Issue: 11 | Nov 2022 www.irjet.net p-ISSN: 2395-0072

Smart Parking Solution using Camera Networks and Real-time Computer Vision

Vidana Gamage C.N., Irugal Bandara S.S., Nugaliyadde S.S., Silva W.P.S.M.

[1] *Department of Computer Systems Engineering Sri Lanka Institute of Information Technology Malabe, Sri Lanka*
[2] *Department of Computer Systems Engineering Sri Lanka Institute of Information Technology Malabe, Sri Lanka*
[3] *Department of Computer Systems Engineering Sri Lanka Institute of Information Technology Malabe, Sri Lanka*
[4] *Department of Computer Systems Engineering Sri Lanka Institute of Information Technology Malabe, Sri Lanka*

Abstract - *At present, the world is seeing an unprecedented push to an electric vehicle future which is forced mainly due to the climate change concerns associated with the internal combustion engine-based cars. These traditional automobiles are significantly less efficient, as a result vehicle infrastructure around cars holds a major role in reducing the carbon footprint. Thus, this research is focused on resolving a major problem that automatically comes with cars and parking. A modern single car takes up a significant space when compared with cars in the early days. Now with most of the world's population are now living in cities, invaluable space in sprawled urban infrastructure is becoming increasingly concerning. Even the electric future won't be any help to this situation. Therefore, this is suggesting improving the existing infrastructure using AI to process the existing surveillance footage. Even though most modern cars equipped with parking sensors they are limited when making small maneuvers at low speeds and cease working as soon as the car's ignition is off. There is a lack of personalized individually serving system of surveillance which the drivers can check whether a crash had been detected associated with his or her car which is currently parked.*

***Key Words*: AI, Computer Vision, Smart Parking, Surveillance footage, parking infrastructure**

1. INTRODUCTION

Under current circumstances, drivers do not have the ability to foresee the park they are about to enter from anywhere they prefer. From the driver's point of view, if the park is busy drivers would find it stressful to manoeuvre inside the facility. This is a much more significant problem in a tight urban space and could lead to miles of traffic jams and ultimately wasting millions of man hours per day. Our solution would be much easier to implement because it doesn't require any additional hardware components unlike IoT systems which mostly use proximity sensors for each parking lot. This whole transformation process of a single garage would be much costlier as well. This system avoids the need to install such hardware and nullifies all the maintenance toll that is guaranteed with that kind of upgrade since this is mostly based on software platform.

Mobile applications would be provided via major OS platforms so users can easily install and get into the service. This mobile application would act as the gateway for accessing personalized real-time information about the park and the vehicle status. When it comes to data visualization aspect s in the application, they are intuitively displayed where most important data is in the upfront and easily accessible viewing modes are presented to the user with a few touches away. There are several view modes which users can check out the latest stats of the parking garage.

The system uses surveillance cameras to keep an eye on the movements and behaviour of vehicles in a certain region to assume if they are approaching or leaving a parking garage. These observations allow computer vision to identify license plate numbers and show drivers the location of the closest vacant parking lot in real-time via a cloud-based application. The system tracks the driver's departure when he or she comes back to retrieve the car, and it determines the cost based on how long the car has been left in the garage.

1.1 Available Parking Lot Detection

Major pillars of this system include parking spot recognition, license plate detection, collision detection and data visualization which is being developed by using Artificial Intelligence and Machine learning based frameworks. By eliminating the need for sensors this system has the adaptability and scalability which would be vital for the efficiency of all kinds of aspects. Functioning of this system needs a minimum of three surveillance camera feeds: at the entrance, view from the top, and at the exit.

The live footage obtained by the entrance would be extracted and processed to identify vehicle license plates.

Fig -1: Graphical Overlook view of the Parking

Top level feed would be processed to achieve two major functions; to detect empty parking lots while detecting the vacant ones while providing a graphical UI for the user implementing data visualization techniques. Additionally, to detect collisions that may occur during making small maneuvers inside the park as drivers trying to park their vehicles into a vacant lot or when they are trying to leave the park after keeping the car parked for some time. The footage delivered by the camera feed at the exit would be used to detect the parking time of a certain vehicle according to the time they have kept the vehicle parked.

1.1.1 Frameworks

This function is developed by utilizing artificial intelligence models like YOLOv5, PyTorch and OpenCV based on the Python language. This YOLOv5 AI model can detect objects in real-time by dividing each frame of video footage or pictures extracted from the surveillance cameras into a grid of cells. It has the ability to shed it attention to specific part of the footage as well. This model is named after the phrase 'You only look once' because it has accuracy speed and efficiency adequate enough to serve in a lightweight mobile based platform. When considering PyTorch, it is a great open-source framework which is often used in this domain.

For the purpose of addressing the feature loss brought on by the compression of high-resolution photographs during the normalization stage, an adaptive clipping strategy based on the YOLO object identification algorithm is recommended for the data pre-processing and detection stage. A high-resolution training dataset is first improved using the adaptive clipping method. Then, by developing a fresh training set, the detailed features of the object detection network are preserved. During the network detection phase, the adaptive clipping approach is utilized to identify the image in segments, and position mapping is employed to merge the coordinates of the detection findings. The output of the chunked detection results is combined with the output of the global detection results.

With the rapid growth of the social economy and urbanization, traffic problems are becoming more and more of a problem. Effective traffic monitoring can help to solve serious traffic problems. As AI becomes a priority on the national agenda, the appeal of intelligent transportation systems will rise. In the transportation sector, an unmanned aircraft with high-definition cameras has a variety of applications and an advantage in parking lot management, intelligent traffic control, and disaster relief. The improved YOLO algorithm can effectively capitalize on the advantages of auxiliary decision-making in a variety of challenging traffic circumstances thanks to the characteristics of quick identification speed, high accuracy, and good detection effect.

Top level views and ground images for detecting vehicles are slightly different because the ground view is mostly acquired by a stationary camera. The top view is photographed from the top view using a camera and a bird's eye lens. Some vehicle side information is thus lost. The image conveys a tremendous amount of information, and top-level cameras' image quality is much superior to that of ground cameras. Therefore, it is important to use images wisely and appropriately.

Identification of ground targets using deep learning is an established method. When it comes to camera-based vehicle recognition, there are still certain problems with the current technology, such as the small number of targets in parking lots that are made up of car parts. By way of illustration, the YOLO object detection network creates a 13*13 prediction grid with a down sampling factor of 32. When there are fewer than 32 pixels separating two target objects, target differentiation errors happen in the network.

The top-tier camera takes high-resolution pictures or footage, which are then edited to provide an image library. The YOLOv5 object detection method is then trained using these images or videos, which are then arranged into an initial training dataset using human labelling and divided into a final training dataset following processing by the suggested adaptive clipping approach. We get the appropriate model weights.

1.1.2 PyTorch

This function is developed by using key artificial intelligence technologies like EasyOCR, YOLOv5 and Pytorch which would be running in Python. The gradient values for the built-in neural networks can be found using the PyTorch framework. PyTorch employs dynamic computational graphs. During the forward calculation, the graph is inferentially defined using operator overloading.

Users have more flexibility when using dynamic graphs as opposed to static ones because they may simultaneously design and assess the graph. They are easy to debug since they can run code line by line. Finding problems in code is made much easier with PyTorch Dynamic graphs, which is a crucial feature that makes PyTorch such a well-liked choice in the industry.

Computational graphs are built from scratch in PyTorch during each cycle. This makes it possible to utilize any Python control flow expression, changing the graph's size and structure in the process. The advantage is that training can start without encoding every possible path. You put what you identify into action.

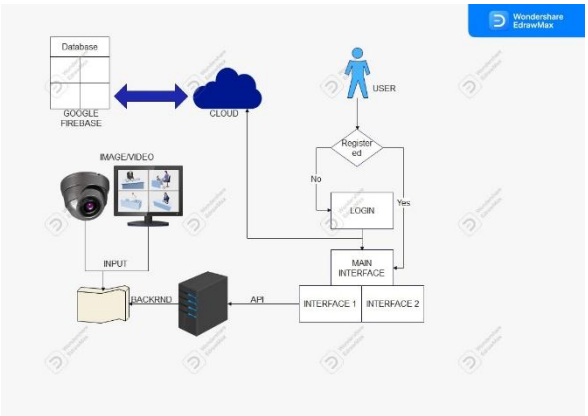

Fig -2: System Diagram

2. LICENSE PLATE DETECTION

This feature is developed using the above discussed technologies and EasyOCR character recognition algorithm. Installing the Easy OCR Python package creates the primary key for optical character recognition. EasyOCR takes the lead in number plate identification, making it possible to construct the model more simply. This process is much more sophisticated than it seems as it has the capability to crop down the frame which is delivering the footage consisting of the license plate so that it works more efficiently.

To start the procedure, first an EOCR object would be built. There are two ways the process can go forward from there. After creating the EOCR object, the first path is to load the font file. The recognize method should then be called, and the final step is to turn the recognition settings. The detection of number plate characters does not typically use this method. The second approach, in which a learning mode is derived from the EOCR object, should be employed for that. The image of the text that must be read must then be loaded. In this instance, it must be read with a loaded video. Both the NewFront and LearnPattern methods should be called after the video has loaded.

Fig -3: License Plate detection featuring a ticketing system

2.1 Technologies used

The system uses a substantial open-source library for computer vision, machine learning, and image processing, called OpenCV. Real-time operation, which is essential in contemporary systems, is one area where it currently plays a vital role. It may be used to search for people, objects, and even human handwriting in images and movies. When Python is used in conjunction with other libraries, like NumPy, it can handle the OpenCV array structure for analysis. To recognize visual patterns and their various qualities, we use vector space and mathematical operations on these properties. Computer vision transforms data from a still or video camera into a decision or a new representation. All these alterations are meant to achieve a certain goal.

3. COLLISION DETECTION

We offer a method for detecting car crashes in parking lots. The suggested process consists of two parts. First, foreground extraction and motion mapping are used to identify a car. In the final stage, it is decided whether an accident has happened based on the speed and direction of the car. Experimental results show that the suggested method accurately detects car accidents that occur while parked. This system uses cameras to capture video of the parking lot and the automobiles there. It also uses cameras to capture photos of the vehicles and image processing to determine the distances between them. Python is the language utilizing here, along with a few libraries. Following image processing, the system uses surveillance cameras to track the movements of vehicles and the space between them. Calculate the distance after taking the picture. If the value exceeds the predetermined value and there are cars present without the allocated standard distance system, the region is considered a risk zone.

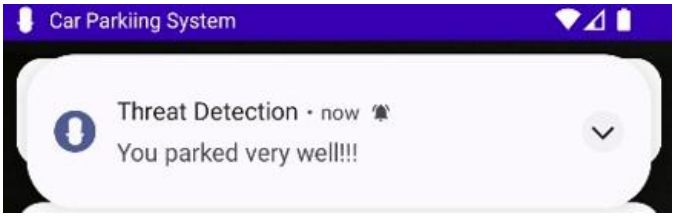

Fig -4: Collision detection notifications

4. DATA VISUALIZATION

The processed data accumulated by the above methodology should be visualized in a way that is straightforward and easily digestible manner. Unless this development will not help the driver already being busy steering their vehicles through tight corners and narrow spaces in an overcrowded urban parking space. Remedy comes as a development of a mobile application which can handle multiple functionalities at once including several views and a dashboard regarding the parking space.

4.1 Methodology

Application intended for the drivers is developed based on Java, XML and retrofit okhttp3 is used to help in API calls. Square has created Retrofit, a type-safe REST client for Android, Java, and Kotlin. The library offers a robust framework for communicating with and authenticating APIs as well as sending network requests via OkHttp. See this manual to learn how OkHttp functions. When retrofit is returned, it typically creates the object that was retrieved using the Gson Factory constructor that we supply. Most items are defined; therefore, the amount of personalization is constrained.

Converter is used for data serialization and is configured in Retrofit. Typically, a free Java library called Gson is used to serialize and deserialize items to and from JSON. To parse XML or other protocols, you can also add your own converters to Retrofit if necessary. To send HTTP requests, Retrofit takes use of the OkHttp framework. It manages all low-level network activities, caching, and manipulation of requests and answers. OkHttp is a pure HTTP/SPDY client. Retrofit, in contrast, is a high-level REST abstraction built on OkHttp. Retrofit is closely linked to OkHttp and largely depends on it.

OkHttp provides the following functionalities, among others: All queries to the same host may share a socket thanks to HTTP/2 capability, Request latency is decreased via connection pooling (if HTTP/2 is not supported), GZIP transparency reduces download size.

Retrofit offers the following crucial characteristics: Support for query parameters and replacement of URL parameters, Converting an object to the request body e.g., JSON, protocol buffers, Upload of files and multipart requests. Users are greeted with the login screen where they can enter the vehicle license plate numbers and create the login. This ID is significant because it is used to bring a personalized experience to the driver.

Fig -5: Dashboard View of the application

3. CONCLUSIONS

The world is currently experiencing an unparalleled drive toward an electric vehicle future, which is being compelled mostly by concerns about climate change related to internal combustion engine-based vehicles. Because these conventional cars are so much less efficient, the infrastructure built around them plays a key part in lowering the carbon impact. Thus, the goal of this research is to find a solution to parking, a significant issue that arises frequently with cars. In comparison to early cars, a single modern car occupies a substantial amount of area. Given that cities currently house most of the world's population, valuable space in sprawling metropolitan infrastructure is a growing source of concern. This problem won't be solved by the electric future at all.

This suggests employing AI to process the current surveillance footage in order to improve the infrastructure that is already in place. Even though the majority of modern cars come with parking sensors, these devices are only effective for tiny maneuvers at moderate speeds and stop functioning as soon as the ignition is turned off. There isn't a tailored, individually served monitoring system that allows drivers to verify if a crash has been discovered linked to their currently parked automobile.

ACKNOWLEDGEMENT

It is a great opportunity to pursue research in the domain of automotive infrastructure development because the whole system seems to be crippled at this rate of new car production and car buying rates. We are excited and grateful to all the supervisors, friends and family for the guidance and courage to pursue this journey.

REFERENCES

[1] FadiAl Turjmana, Arman Malekloob (2019, August). Smart parking in IoT-enabled cities: A survey ScienceDirect Volume 49, 101608 Available at: https://www.sciencedirect.com/science/article/abs/pii/S2210670718327173

[2] Vijay Paidi, Hasan Fleyeh, Johan Håkansson, Roger G. Nyberg (2018 May) Smart parking sensors, technologies, and applications for open parking lots: a review. IET research Available at: https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-its.2017.0406

[3] Liehuang Zhu, Meng Li, Zijian Zhang, Zhan Qin. (2018 June) ASAP: An Anonymous Smart-Parking and Payment Scheme in Vehicular Networks. IEEE Xplore Volume 17 Issue 4. Available at: https://ieeexplore.ieee.org/abstract/document/8396301

[4] Fotios Zantalis, Grigorios Koulouras, Sotiris Karabetsos, Dionisis Kandris (April 2019) A Review of Machine Learning and IoT in Smart Transportation. Mdpi volume 11, Issue 4. Available at: https://www.mdpi.com/1999-5903/11/4/94

[5] Ali Hassan Sodhro, Sandeep Pirbhulal, Zongwei Luod, Victor Hugo C.de Albuquerque (May 2019) Towards an optimal resource management for IoT based Green and sustainable smart cities. ScienceDirect. Volume 220. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0959652619302082

[6] Moneeb Gohar, Muhammad Muzammal, Arif Ur Rahman (August 2018) SMART TSS: Defining transportation system behavior using big data analytics in smart cities. ScienceDirect. Volume 41. Available at: https://www.sciencedirect.com/science/article/abs/pii/S2210670717309757

[7] Mehdi Nourinejad, Sina Bahrami, Matthew J.Roorda. (March 2018) Designing parking facilities for autonomous vehicles. ScienceDirect. Volume 109 Available at: https://www.sciencedirect.com/science/article/abs/pii/S0191261517305866

[8] M. Mazhar Rathor, Anand Paul, Won-Hwa Hong, Hyun Cheol Se Imtiaz Awan Sharjil Saeed. (July 2018) Exploiting IoT and big data analytics: Defining Smart Digital City using real-time urban data. ScienceDirect. Volume 40 Available at: https://www.sciencedirect.com/science/article/abs/pii/S2210670717309782

[9] Prince Waqas Khan, Yung-Cheol Byun, Namje Park. (March 2020) A Data Verification System for CCTV Surveillance Cameras Using Blockchain Technology in Smart Cities. mdpi.com Volume 9 Issue 3. Available at: https://www.mdpi.com/2079-9292/9/3/484

[10] Harshitha Bura; Nathan Lin; Naveen Kumar; Sangram Malekar; Sushma Nagaraj; Kaikai Liu (July 2018) An Edge Based Smart Parking Solution Using Camera Networks and Deep Learning. IEEE Xplore Volume Available at: https://ieeexplore.ieee.org/abstract/document/8457691

[11] Chyn Ira C. Crisostomo; Royce Val C. Malalis; Romel S. Saysay; Renann G. Baldovino (November 2019) A Multi-storey Garage Smart Parking System based on Image Processing. IEEE Xplore Available at: https://ieeexplore.ieee.org/abstract/document/8932899

[12] Chandra Kiran B. Krishnamurthy, Nicole S.Ngoc. (January 2020) The effects of smart parking on transit and traffic: Evidence from SFpark. ScienceDirect. Volume 99, 102273. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0095069619301664

[13] Faris Alshehri; A. H. M. Almawgani; Ayed Alqahtani; Abdurahman Alqahtani. (May 2019) Smart Parking System for Monitoring Cars and Wrong Parking. IEEE Xplore Available at: https://ieeexplore.ieee.org/abstract/document/8769463

[14] Kuchi N S S S S Utpala, Suresh Kumar, K.Praneetha, D.Hema Sruthi, K.Sai Avinash Varma. (2019) Authenticated IoT Based Online Smart Parking System with Cloud. Pramana Research Journal. ISSN NO: 2249-2976, Available at: https://www.pramanaresearch.org/gallery/prj-p622.pdf

[15] Noah Sieck; Cameron Calpin; Mohammad Almalag. (March 2020) Machine Vision Smart Parking Using Internet of Things (IoTs) In A Smart University. IEEE Xplore. Available at: https://ieeexplore.ieee.org/abstract/document/9156121

[16] Tayo Fabusuyi, Victoria Hill (2020) Designing an integrated smart parking application. ScienceDirect, Volume 48. Available at https://www.sciencedirect.com/science/article/pii/S2352146520305500

[17] Jan Šilar; Jirí Růžička; Zuzana Bělinovà; Martin Langr; Kristýna Hlubučková; (October 2018) Smart parking in the smart city application. IEEE Xplore Available at: https://ieeexplore.ieee.org/abstract/document/8402667

[18] Amal O. Hamada, Fatma Zahran; Noha Ezz ElDin; Mohamed Azab; Mohamed Eltoweissy; Denis Gračanin.(October 2019) smartpark: A Location-Independent Smart Park and Transfer System. IEEE Xplore. Available at: https://ieeexplore.ieee.org/abstract/document/8936180

[19] Garisa, Shankara Sree Vatsava Konanki Rangaiahgari, Dinesh Chakravarthi. (2022) Smart Parking Assisting System. Divaportal. Available at: https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1690643&dswid=3298

[20] Mingyan Bai; Shenghua Zhong; Pengyu Yan; Zhibin Chen; Zhixian Zhang. (December 2021) A Data-Driven Near-Optimization Approach for Smart Parking Management Platforms. IEEE Xplore. Available at: https://ieeexplore.ieee.org/abstract/document/9702219