Development of an Automatic Data-centered Parking Space Occupancy Detection with Vehicle License Plate Web-Based Monitoring

Abdul Ghaffar U. Tagolanao, Hashim Monir M. Ambia, and Cristina P. Dadula

Abstract

This paper is a two-step camera-based system running on a web-based monitoring center that utilizes Convolutional Neural Networks (CNN) to achieve the detection of license plates and the identification of the vacancy of parking spaces for a display to entering drivers. The license plate detection is achieved using a retrained tiny You Only Look Once (YOLO) model in conjunction with Tesseract OCR for character recognition. The detected plate images are stored in separate directories according to date, and filenames according to time. The parking occupancy network employs a simplified AlexNet to classify parking spaces as either 'vacant' or 'occupied'. Testing was done on an emulated parking lot, where the live camera feed on the entrance and exit are simulated by prerecorded videos. The license plate detection works on an average of 0.023 seconds with 100% accuracy on still images. Parking Space occupancy identifier can classify the vacancy of 37 parking slots in an average of 0.06 seconds with 10.14% of false occupancy identification, 0.92% of false vacancy identification, and 88.94% correct identification. Though tested in a simulated environment, the test results show that the system can be implemented and applied to actual parking lots.

***Keywords*—Character Recognition, License Plate Detection, Parking Space Occupancy Detection, Smart Trasnportation**

EEE Department, Mindanao State University- General Santos City

General Santos City, Philippines

Email: skieyfar@gmail.com, monirambia@gmail.com, cristina.dadula@msugensan.edu.ph

I. Introduction

The rapid industrial growth is reflected by the increase in the number of vehicles on the streets throughout the world which causes parking-related problems. Currently, most of the existing car parks do not have a systematic approach, and most are manually managed. For motorists, the search for a parking space is a time-consuming process that affects the efficiency of economic activities, social interactions, and resource usage. This problem usually occurs in urban areas, where the number of vehicles exceeds the availability of parking spaces. These inefficient conditions arise due to a lack of exploitation of currently available technologies in the market [1] [2]. Various systems have been developed to ensure the smoothness of traffic congestion in some car parking areas, from manual implementations used in older systems to modern implementations used in current computerized systems [1] [2] [3].

Some of the recent advancements in parking space management include the implementation of the Internet of Things to assist drivers in parking space selection [4], the identification of available parking spaces using optical character recognition [5], and the estimation of parking lot capacity by using image processing on aerial images [6]. Intelligent Transportation System is a concept of multi-technology that integrates Cyber Technology, electronic technology, information technology, systems engineering [7], which deals with several vehicular problems. ITS is therefore practiced by many developed countries to deal with traffic [3] as well as parking problems.

This paper intends to create a web-based monitoring system that will display information about parking space availability by using image processing. The goals of the system are as follows: provide real-time information about the available parking space in a parking area [1] [8], display the parking space availability in a monitor to motorists, capture the license plates of the vehicles passing through the entrance and exit of the parking space using cameras, identify the characters of the captured license plates, and create a log containing the in-and-out timestamps, the captured image of the license plates, and the identified characters of the license plates.

The usage of cameras in this study will greatly help in monitoring the activities inside the parking area. The web-based motoring center means that the system could be accessed by any device connected to the server and could, therefore, reduce the costs for interfacing. The database will ensure that there will be a record of all vehicles entering the parking area. Lastly, the parking space occupancy detection will also be of great help in assisting motorists in locating vacant parking slots inside the parking area.

II. Methodology

A. Overview of the Setup

The system, as shown in Fig. 1 incorporates two key hardware components: the computer and the cameras. All processing, including plate detection, character recognition, parking occupancy detection, and data logging, is done on the computer. The cameras provide the input images for the system. On the software side, the system adopts python language to make the hardware and the software components communicate with each other. The monitor serves as the display for the status of the parking area by getting real-time data from the computer server via a wireless network. This is summarized as shown in Fig. 2. Explained further in this chapter are the implementation and development of the core components of the system: hardware and software.

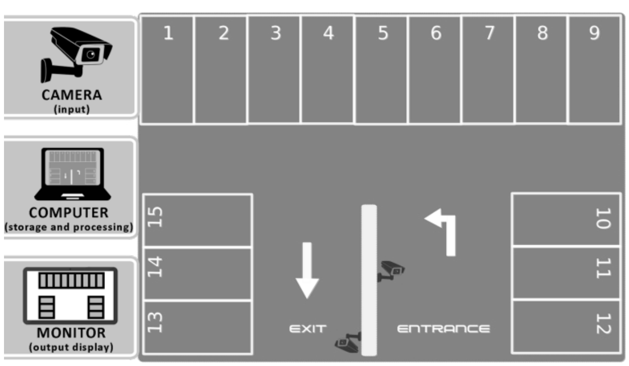

Fig. 1. The working operation of the system, consisting of 3 cameras, a parking lot, a computer, and a monitor display

B. Hardware Setup

1. *Camera:* The system incorporate three (3) cameras into its workflow. One of the cameras was used to cover the field of the parking area. The remaining two (2) were used for covering the entrance point and the exit point of the parking area.
2. *Computer:* A computer, particularly a laptop with the specifications shown in Table I, was used for the processing of the program of the system.

Table I.
Technical Specifications of the Local Machine Used in the Development of the Computer Vision System

Model	HP Omen 15-DC0100TX Shadow Black
Processor	Intel Core i7-8750H Processor (9M Cache, up to 4.10 GHz)
Memory	8 GB DDR4-2666 SDRAM (1 x 8 GB)
Storage	1 TB 7200 rpm SATA
Graphics	NVIDIA GeForce GTX 1050 (4 GB GDDR5 dedicated)
Connectivity	Intel Wireless-AC 9560 802.11b/g/n/ac (2x2) Wi-Fi and Bluetooth 5 Combo

3. *Wireless Network:* The wireless network will serve as the connection between the laptop server and the client display for the entrance of the parking area.

Fig. 2. The connection of the laptop server and the display at the entrance

C. Development of the System

The system components were developed and tested individually before merging into a single web-based monitoring system.

1) Parking Space Occupancy Detection: The parking space occupancy detection for the system is an image classifier that will classify a cropped image of a given parking space into 'vacant' or 'occupied'. The network used for the classification is based on *AlexNet*, a high-performance convolutional neural network developed by Alex Krizhevsky [9].

The AlexNet was modified to classify between two classes instead of 1000 to reduce the complexity of the network, thus reducing the processing time of classifying between two classes. Shown in Fig. 3 is the architecture of the modified AlexNet.

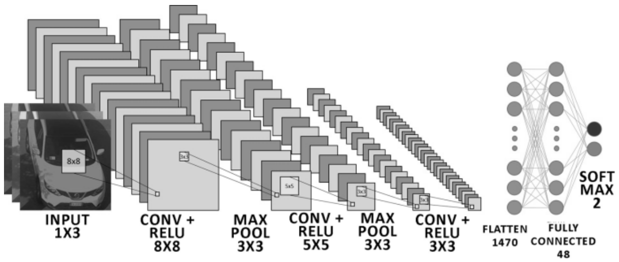

Fig. 3. The modified model of AlexNet used in the system

The parking space occupancy detection can be summarized into the following steps: A representation of the parking lot having rectangles with unique ids representing a parking space is drawn ahead of time as shown on the right side of Fig. 16. Bounding squares are labeled on the live image of the parking lot ahead of time, with each square representing the same parking space as shown on the left side of Fig. 16. Each bounding square is assigned with a unique id corresponding to each rectangle in the representation. The coordinates and ids of each bounding square are stored as a text file in JSON format. The current frame from the camera feed is cropped at regions specified by each of the predetermined bounding squares. All cropped images are resized to 150x150 to match the input size of the modified AlexNet. All cropped images are input to the modified AlexNet for classification into 'vacant' or 'occupied'. All squares classified as 'vacant' are drawn on the current frame with a green border. All squares classified as 'occupied' are drawn on the current frame with a red border. The representation of the parking lot is rendered with colors.

2) License Plate Detection: Automatic license plate detection and recognition are already widely used by many countries for surveillance and monitoring purposes.

You Only Look Once (YOLO), specifically the tiny YOLO implementation, is used in this system. Tiny YOLO was selected due to its small size, high framerate, and lightweight. The Tiny Yolo model, shown in Fig. 4, has been retrained to predict the bounding boxes of vehicle license plates in an image. The model, on the other hand, provides several bounding boxes for an image. These superfluous bounding boxes are filtered out, leaving only those with a confidence level greater than 0.6. The retained bounding boxes are given an ID and tracked until it reaches an imaginary line, at which it will be cropped out, resized to 200x50, and stored in the database.

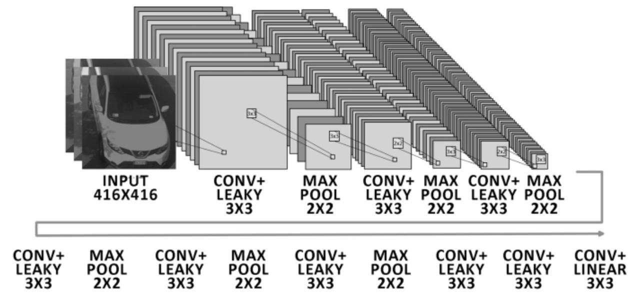

Fig. 4. The YOLO architecture model used in the system

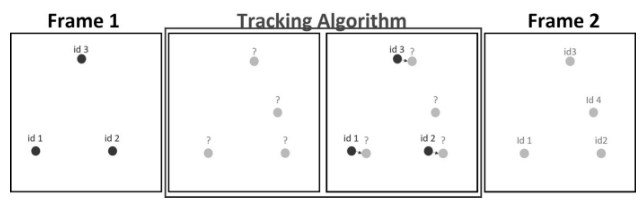

Fig. 5. The flow of the tracking algorithm used to tract license plates.

The cropped image of the detected license plate is saved to a folder with a file naming format of /yyyy-mm-dd/mm-ss/jpg.

The system's license plate detection is implemented for both the entrance and exit of the parking area and therefore requires two cameras. The license plate detection for both the entrance and the exit can be summarized into the following steps:

1. Grab the image from the camera.
2. Predict the location of the bounding boxes in the grabbed image using the retrained tiny YOLO model.
3. Filter the bounding boxes to only retain those with a confidence of more than 0.6.
4. The centroid of each bounding box in a frame is calculated and stored for comparison with the next frame, shown in Fig. 5 Frame 1.
5. The centroids of the current frame (c curr) are compared to the centroids of the previous frame (c prev), shown in Fig. 5 bounded in a red box.
6. The Euclidean distances between all centroids in c curr and c prev are calculated using the formula in Eq. 1.

$$Dist = \sqrt{(x2 - x1)2 + (y2 - y1)2} \tag{1}$$

- A centroid from c_curr with the nearest distance to a corresponding centroid in c prev will be identified as a single plate as shown in Fig. 5 Frame 2.

Centroids in c_curr (i.e., id4 in Fig. 5 Frame 2) with no corresponding centroid in c_prev are labeled as new id.

- When the centroid of a plate passes through the imaginary line, the plate is cropped, aligned horizontally, resized to 250x50, and saved to file with the naming format of */yyyy-mm-dd/hh-mm-ss.jpg*.

3) Data Logging: By definition, data logging is the process of using a computer to collect data over time and save the output for analysis or archiving. The data logging of the system saves three types of data. The cropped and aligned image of the detected license plate. The recognized characters of the detected license plate. The in-and-out timestamp of the vehicle when the license plate was captured. The in-and-out timestamp of the vehicle license plate, as well as the recognized characters of the license plate, is saved to the database as a text file in JSON format. A link to the license plate image is also included among the data logged on the database.

4) Character Recognition: The character recognition ability of the neural network will depend on its training set. The process of gathering the dataset is time-consuming. Moreover, the great variation of the license plates used in the Philippines is an obstacle in gathering a reliable training set. These reasons are the basis for using the pre-trained network called Tesseract OCR. Initial testing of the Tesseract OCR using the cropped license plates did not give acceptable results, thus image preprocessing was done to improve the recognition of the characters.

The plate character recognition steps are implemented as follows:

– Preprocessing the cropped license plate image for image enhancement. It employs converting the image into grayscale, improving the license plate contrast using Contrast Limited Adaptive Histogram Equalization, applying gaussian blur to reduce noise for the next step, and applying adaptive thresholding to produce a binary image.
– Drawing the contours of alphanumeric characters from the preprocessed image. It employs getting the contours from the preprocessed image, filtering the contours from the previous step to remove unwanted "noise contours", and drawing and filling the contours that meet the following criteria below for possible character contours. Take note that contour bounding area is greater than 100 square pixels, and the contour bounding width greater than 4 pixels, contour bounding height greater than 16 pixels contour bounding aspect ratio must be between 0.15 and 1.33.
– Recognizing the alphanumeric characters from the drawn contours in Tesseract-OCR.

D. Data Gathering and Training

The datasets used for training the license plate detection, and the parking occupancy detection networks were collected from open source databases produced for research purposes.

1. *Car Dataset for License Plate Detection* A Croatian database of cars [10] was used as the training data for the license plate detection. The images are available for download from http://www.zemris.fer.hr/ projects/ LicensePlates/english/ results.shtml. The database contains 602 images of cars, mostly in the rearview. Some contents of the Croatian database are shown in Fig. 6.

Fig. 6. Sample images from the Croatian car database with different vehicle models and car plate texts.

2. *Carpark Dataset for Parking Space Occupancy Detection* CNRPark+EXT [11] [12] provided a database for the training of parking occupancy detection network. Dataset provided contains cropped images of parking slots called by the author as "Patches" as shown in Fig. 7. These patches are categorized into two; occupied parking slots and vacant parking slots.

The datasets were processed further using the following methods:

1. *Splitting the Patches for the Parking Space Occupancy Detection* The carpark dataset, "patches", was split into two categories; training set and evaluation set. The training set was used to train the network for vehicle presence on the slot, and the other was used to validate and know the

accuracy of the network. The ratio of data was 9:1 for training and evaluation, respectively.

2. *License Plate Region Annotation* To compensate for the small number of datasets, data augmentation was applied to the images in the car dataset gathered to artificially expand the size of the training data from the original number 602 images to 4188 images. The data augmentation applied the following modifications:

- Shifting of the images to the right.
- Shifting of the images to the left.
- Zooming of the images from the center.
- Randomly increasing or decreasing the brightness of the images.

Fig. 7. The dataset for the parking occupancy detection with multiple obstructions (trees, light post), shadowing and natural light variations.

The augmented dataset was annotated for training using LabelImg, a graphical image annotation tool written in Python used to label object bounding boxes in images (Fig. 8). Annotation in machine learning is the process of labeling the data on images containing objects. In this paper, the objects to be labeled are the license plate region of vehicles. The image annotation was done manually for each of the 4188 images using LabelImg. Annotations are saved as XML files in PASCAL VOC format [13]. When opening a previously annotated image in LabelImg, it will show the location of the saved bounding box as shown in Fig. 8.

Fig. 8. LabelImg, the annotation tool used to annotate the images in the dataset

E. Designing the Web-Based Monitoring Center

The web-based monitoring center was developed using an HTML-CSS-Javascript combo running over a Flask server. Flask is the web framework used to develop the web monitoring center. Flask was chosen because it is lightweight and novice-friendly. The web implementation was made to utilize the widespread availability of wireless networks and enable remote viewing of the monitoring center when connected within the same local network. The web pages for the flask application are shown using an HTML-CSS-Javascript combo. HTML gives the structure of the web page while CSS provides the visual layout of the web page. Javascript, on the other hand, is a client-side scripting language embedded within the HTML codes to give commands and dictate the web page behavior. The monitoring center has two main pages: the homepage that houses all the details about the page and the project, and the program shows all the outputs of the system. This includes the license plate detection, the parking space occupancy detection, and the cropped license plate image. The program button is a special button on the homepage to redirect the user to the program page when clicked, see Fig. 9.

a) The Homepage: The homepage navigation bar is used to navigate through the pages available on the homepage, which are: the authors' page that contains details about the authors of the project, and the project page contains the description of the project.

b) Program Page: The program page shows: the video feed of the entrance and the exit point of the parking area, the navigation bar for the program page, the image of the license plates captured from the entrance and exit point, and the navbar in the program page is used to navigate through the page showing the actual feed of the parking lot with the bounding boxes of the parking slots and the page showing the status of the parking slots that are either occupied or vacant.

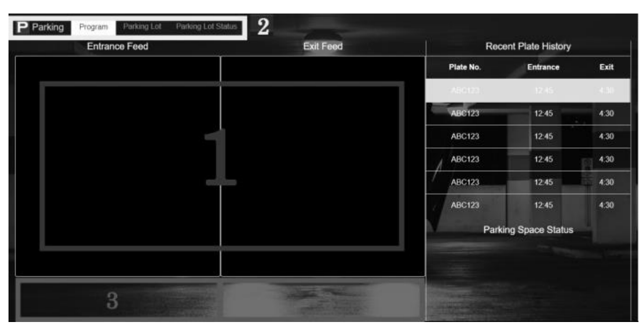

Fig. 9. The design of the program page of the monitoring center.

c) Parking Occupancy Status: The status of the parking area is shown in Fig. 10. The layout of the parking area is manually drawn according to the layout of the parking area

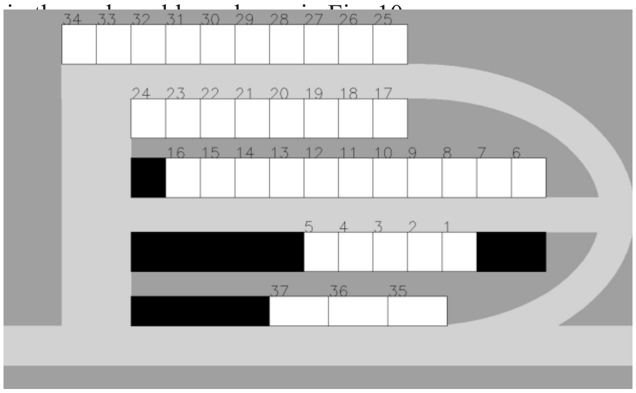

Fig. 10. The parking space occupancy output.

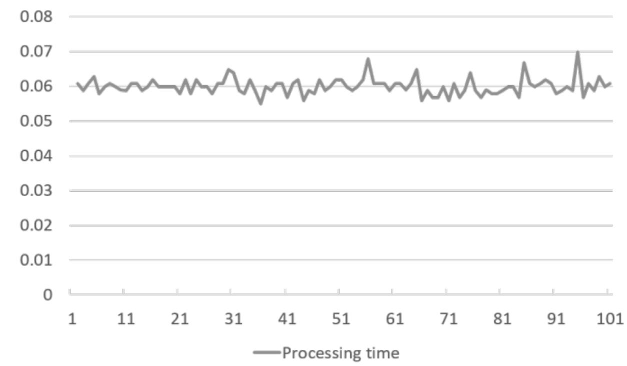

Fig. 11. Parking Space Occupancy Processing Times

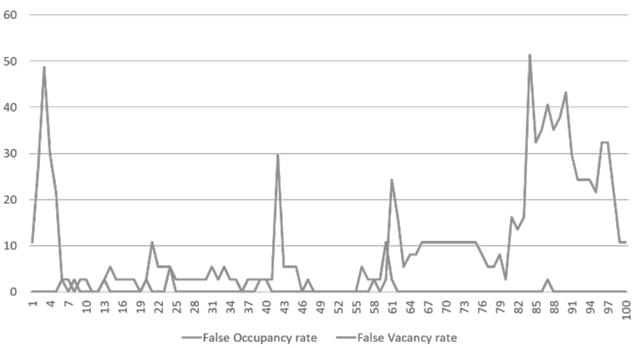

Fig. 12. Parking Space Occupancy false detection rates

F. Testing of the System

The testing of the individual components of the system was done separately to ensure the reliability of each component its integration on the web-based monitoring center. License plate detection was tested for both still images and live videos to test the accuracy and processing time of the detection of the license plates. Parking Space Detection was tested on still images of a car park from the dataset gathered from CNRPark+EXT [11] [12]. For this component of the system, the parking area camera was simulated by showing still images of a fixed camera angle of a parking lot from the CNRPark+EXT dataset as a sequence. The testing was done to measure the accuracy and processing time of the identification of 'vacant' and 'occupied' parking slots.

III. RESULTS AND DISCUSSION

A. Detection of the vacancy of parking area

The occupancy detection network was able to correctly classify the parking slots 88.95 % based on the 100 images of the parking area that were tested. However, this indicates that the network is far from perfect, as there were false occupancy and false vacancy detections. On average, false occupancies occur at 10.14 %, while false vacancies occur at 0.92 %. The processing times of parking occupancy detection for the 100 images are depicted in Fig. 11. Each of the 37 parking spaces in camera 4 takes an average of 0.059976697 seconds to fully identify. Fig. 12 depicts the percentage of parking-lot slots that are falsely detected.

The high occurrence of false occupancy detection happens mostly on the images taken on afternoons to night times, as shown in Fig. 13.

B. License Plate Detection

The license plate detection was tested using still images and video feed to see how the detection behaves with the two different file types. Shown in Fig. 14 are the prediction and processing time of the network for the detection of the license plates for the 100 images. The total processing includes detection and drawing of the bounding boxes. The tested images have dimensions of 640 x 480 pixels. The average time for plate detection is 0.023448862 seconds. On the other hand, the whole process of detection and bounding box drawing takes about 0.02377052 seconds on average.

C. Character Recognition

The mixed results produced by the recognition of characters using tesseract OCR were done using 50 random images of license plates extracted from cars used in license plate detection testing. The tesseract OCR was only able to properly recognize 18% of license plate images under test without errors, 60% with more than four (4) correctly recognized characters, and 22% had very bad reading.

Fig. 13. Parking Space Occupancy high false occupancy rates

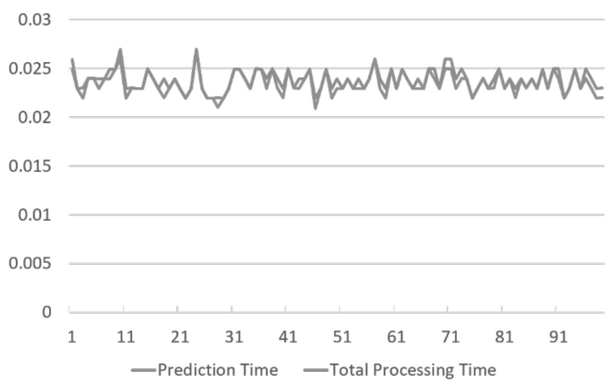

Fig. 14. The prediction and processing times of license plate detection for 100 images.

These results show that the Tesseract OCR should be retrained and not be used as-is. The variance in fonts used in license plates may contribute to bad character recognition. The image processing helped in improving the recognition accuracy. However, it is not enough to compensate for the inherent bad recognition of Tesseract OCR for license plate characters.

D. The Monitoring Center

The license plate detection and parking occupancy detection were successfully integrated into the web-based monitoring center. The entrance and exit camera feeds are shown in the monitoring center, as shown in Fig. 15. The camera feed shows the bounding boxes of the license plate of vehicles. The cropped license plates are displayed below the camera feed. The cropped license plates are displayed after their processing and logging into the database. The resolution of video input to the system is lower and subpar compared to the quality of images that were used in the testing of the standalone network causing more problems to the already bad character recognition.

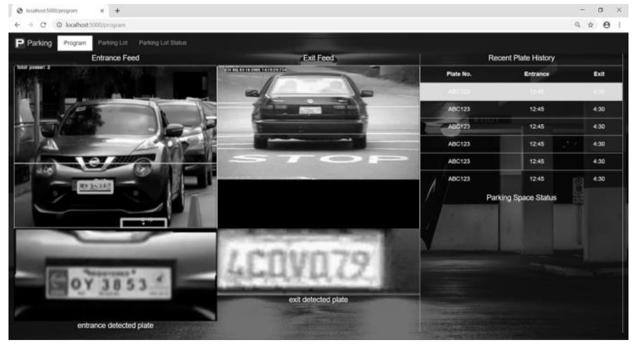

Fig. 15. The program tab with the working license plate detection.

The parking space occupancy was also integrated into the system successfully. The performance of parking space occupancy detection was the same as it was tested alone. Fig. 16 shows the successful integration of the parking space occupancy detection into the monitoring center. The dataset used for training is the main factor for the ability of the parking space occupancy detection network to correctly differentiate between the vacant slots from the occupied slots.

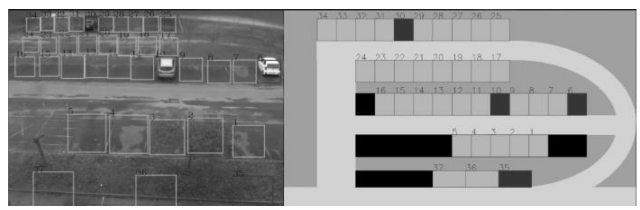

Fig. 16. The parking lot feed and the status update of the parking space occupancy place side by side.

IV. Conclusion

Even though the testing for this system is done on an emulated parking lot, the system was able to successfully recognize the car plate region with 100% accuracy on a controlled environment. However, the accuracy of the network for the video feed of the actual parking lot entrance was lower due to the low resolution of the input video and the high variance of cars passing through the camera. The parking space occupancy identifier proves satisfactory in classifying the vacancy of each parking space with 10.14% of false occupancy identification, 0.92% false vacancy identification, 88.94% correct identification. False identifications are more likely to occur in situations with bad lighting conditions and obstruction of the camera view.

A. Recommendations

- Gather a better dataset for parking space occupancy detection, which includes situations with bad lighting, bad weather, and an obstructed view of the area under test. Gather a better dataset for license plate detection, which should include images taken in bad lighting conditions, more camera angles, and greater variance of cars.

- Increase the resources available by creating a single neural network for plate localization and character recognition.
- Create a web server capable of serving multiple clients at the same time.
- Create a better character recognition algorithm or retrain the existing tesseract OCR suited for license plates character recognition.

References

[1] Faheem, Mahmud, Khan, Rahman, and Zafar, "A survey of intelligent car parking system," *Journal of Applied Research and Technology*, pp. 714–726, 2013.

[2] R. Yusnita, F. Norbaya, and N. Basharuddin, "Intelligent parking space detection system based onimage processing," *International Journal of Innovation, Management and Technology*, vol. 3, no. 3, p. 232, 2012.

[3] Y. Han, J. Shan, M. Wang, and G. Yang, "Optimization design and evaluation of parking route based on automatic assignment mechanism of parking lot," *Advances in Mechanical Engineering*, vol. 9, no. 7, p. 1687814017712416, 2017.

[4] M. Ramasamy, S. G. Solanki, E. Natarajan, and T. M. Keat, "Iot based smart parking system for large parking lot," in *2018 IEEE 4th International Symposium in Robotics and Manufacturing Automation (ROMA)*, 2018, pp. 1–4.

[5] A. Athira, S. Lekshmi, P. Vijayan, and B. Kurian, "Smart parking system based on optical character recognition," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, 2019, pp. 1184–1188.

[6] M. I. Ashqer and M. Bikdash, "Parking lot space detection based on image processing," in *2019 SoutheastCon*, 2019, pp. 1–6.

[7] Y. Ji, W. Deng, and W. Wang, "A planning method for determining optimal parking space display of parking guidance sign boards," in *ICCTP 2010: Integrated Transportation Systems: Green, Intelligent, Reliable*, 2010, pp. 2389–2396.

[8] N. Bibi, M. N. Majid, H. Dawood, and P. Guo, "Automatic parking space detection system," in *2017 2nd International Conference on Multimedia and Image Processing (ICMIP)*. IEEE, 2017, pp. 11–15.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12. USA: Curran Associates Inc., 2012, pp. 1097–1105. [Online]. Available: http://dl.acm.org/citation.cfm?id=2999134.2999257

[10] V. Srebric and prof.dr.sc.Slobodan Ribaric, "Diplomski zadatak br. 1397:postupci za poboljsanje kontrasta sivih slika,"˘ *University of Zagreb,*

[11] *Faculty of Electrical Engineering and Computer Science, School of Computing Department of Electronics, Microelectronics, Computer and Intelligent Systems: Pattern Recognition*, 2003.

[12] G. Amato, F. Carrara, F. Falchi, C. Gennaro, and C. Vairo, "Car parking occupancy detection using smart camera networks and deep learning," in *Computers and Communication (ISCC), 2016 IEEE Symposium on*. IEEE, 2016, pp. 1212–1217.

[13] G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini, and C. Vairo, "Deep learning for decentralized parking lot occupancy detection," *Expert Systems with Applications*, vol. 72, pp. 327–334, 2017.

[14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010.