

Proposing Real-time Parking System for Smart Cities using Two Cameras

Phat Nguyen Huu*

School of Electronics and Telecommunications, Hanoi University of Science and Technology, Hanoi, Vietnam
phat.nguyenhuu@hust.edu.vn

Loc Hoang Bao

School of Electronics and Telecommunications, Hanoi University of Science and Technology, Hanoi, Vietnam
Loc.HB202786M@sis.hust.edu.vn

Received: 03/Oct./2020

Revised: 18/March/2021

Accepted: 18/June/2021

Abstract

Today, cars are becoming a popular means of life. This rapid development has resulted in an increasing demand for private parking. Therefore, finding a parking space in urban areas is extremely difficult for drivers. Another serious problem is that parking on the roadway has serious consequences like traffic congestion. As a result, various solutions are proposed to solve basic functions such as detecting a space or determining the position of the parking to orient the driver. In this paper, we propose a system that not only detects the space but also identifies the vehicle's identity based on their respective license plate. Our proposal system includes two cameras with two independent functions, Skyeeye and LPR cameras, respectively. Skyeeye module has function to detect and track vehicles while automatic license plate recognition system (ALPR) module detects and identifies license plates. Therefore, the system not only helps drivers to find suitable parking space but also manages and controls vehicles effectively for street parking. Besides, it is possible to detect offending vehicles parking on the roadway based on its identity. We also collect a set of data that correctly distributes for the context in order to increase the system's performance. The accuracy of proposal system is 99.48% that shows the feasibility of applying into real environments.

Keywords: object detection; single shot detector; multi-view cameras; automatic license plate recognition; ALPR.

1- Introduction

With the development of the economy today, cars are becoming a popular means of life. This rapid development has resulted in an ever-increasing demand for private parking beyond the available supply. Therefore, the situation of not meeting the demand for parking is increasingly serious. Not only in terms of available parking space that does not meet but also the main common management method that depends on human communication with physical space entities. This leads to the waste of manpower as well as efficiency of the parking space. The main difficulty comes from not knowing about the status of parking spaces that are empty for a given period of time. Most of the parking always have space. However, the parking efficiency is low due to inefficient storage.

Finding a suitable parking space in an urban city without knowing the location of the parking lot is extremely difficult for the driver. That's why driving around in search of a parking space is a frustrating experience. The problems lead to unsafe driving, time consuming, fuel consumption, and increasing emissions that pollute the

environment and cause traffic congestion. Besides, illegal parking is more popular. To solve this problem, parking information and guidance systems (PGI) were developed. The PGI system requires real-time accuracy and responsive updates on the status of a parking space since it can provide users with vacancy and optimize management.

The outstanding advantages of PGI systems that use cameras comparing to other existing systems include: first, no additional infrastructure are required; secondly, camera-based systems provide exact location of empty parking spaces; thirdly, the method of using cameras are highly applicable to the parking spaces on the roads and residential areas.

Image-based parking occupancy detection is basically related to the problem of vehicle detection in the parking space. One of the most important issues with the detection is whether the training, development and test data share the actual distribution. Therefore, we have collected a new data set with the correct distribution for the context. We then use transition learning to increase system performance.

To manage the parking space and information about its status for drivers is an important issue. Therefore, a proposal system not only detects the total amount of free

* Corresponding Author

space but also identifies the identity of each respective vehicle in the paper. Besides, sharing information about the status of the parking space helps to reduce the time to search for a parking location. Vehicle identification is based on a license plate that can be paid online, a parking time notification for the driver, and a vehicle offender. Those are our main contributions.

2- Related Work

Smart parking is one of the most applied and fastest growing solutions worldwide. Due to the inadequacies leading to serious consequences such as time, traffic congestion, and environmental pollution, smart parking has become an attractive field for many researchers. By using low-cost sensors, real-time data and response allows the user to monitor parking spaces. The goal is to automate and reduce the search time for drivers. Besides, we also propose solutions that include a complete set of services such as online payment, parking time notification, and driver orientation.

Existing PGI systems are divided into 4 categories: 1) counter-based systems, 2) wired sensor-based systems, 3) wireless magnetic sensor-based systems, and 4) camera or imaging based system. The counter-based system is dependent on the sensors at the parking. This system can only provide information about the total number of seats instead of guiding the driver to the exact location of the parking space. Therefore, it is not applicable for on-road parking. Wired sensor-based systems and wireless magnetic sensor-based systems depend on ultrasonic sensors, infrared light or wireless magnetic sensors which are installed on each parking space location. Both systems have been adopted in real commercial and used for large shopping malls. However, these methods require the installation of wireless technology sensors, processors, and transceivers. Although sensor-based systems are highly reliable, their high installation and maintenance costs limit for a wide range of applications. To compare with systems using sensors, systems using cameras are cost effective since both monitoring and blank detection functions can be performed simultaneously.

Various methods and techniques have been proposed to deal with parking problems in urban areas. In many modern car parks, popular systems such as ground sensors are used to determine the status of parking locations. The methods use range from using RFID radio frequency identification [1] [2], or infrared sensor [3] to ultrasonic sensor. One of these is the RFID-based smart parking system developed by [4] that implements an automatic process of entry control and exit control using an RFID. Another system is proposed by [5] based on the effect of magnetic on sensors. It is used to determine the state of the parking space. Besides, authors propose smart parking

service systems using short message services (SMS) or a global system for mobile devices (GSM) or microcontroller to enhance security by [6]. ZigBee technique is used together with GSM module to manage and reserve parking by [7]. These systems achieve the function of which is primarily required by blank detection. However, it is affected by factors such as changing weather conditions or the presence of electromagnetic interference.

Node-to-node implementations require a lot of time. For these systems, it is required to set up, install, and maintain sensors in each parking space. Therefore, the cost is very expensive especially in parking spaces with a large number of locations.

Through camera-based methods, empty parking positions can be identified. The data being processed and the generated result will precisely determine the specific number and location of the parking space. Zhang Bin et al. proposed a vision-based parking detection method that is easy to install, low cost. Besides, the detector can be easily adjusted according to requirements. Furthermore, image data is increasingly plentiful and diverse. However, the disadvantage of these methods is that the accuracy is highly dependent on the camera position. H. Ichihashi et al. has proposed a parking space detection system based on a vision that is affected mainly by weather and lighting conditions such as raindrops falling on the lens of camera. For these reasons, cameras are predominantly used for detecting vehicles in indoor parking spaces [8].

In [18] the authors used the wide-angle camera as a sensor. It detects only empty parking spaces and records them. Post-processing information is used to designate a parking space for the newly arrived driver. The Intelligent Traffic System (ITS) and Electronic Toll Collection (ETC) use character recognition (OCR) to create a record for all incoming vehicles. This produces less entry tags for all vehicles in the parking but it does not assign a position for the new driver. A generalized character recognition algorithm is not possible that makes it difficult to create the records shown in [17]. Another proposing system by image processing is proposed in [9]. In this article, an image processing technique is demonstrated to capture the brown circle drawn on the parking area and processed it to detect empty space. In [10], an image of vehicle is saved as a reference and the others are matched with the reference using edge detection and information about the positions and displays. Besides, several methods have been proposed to extract features from images such as [11-15]. With the great progress and achievements gained from deep learning, the authors have come up with other methods by using Artificial Intelligence (AI) to process input images from cameras. Advances of multicore architecture allow to use of convolutional deep neural network architectures (CNN) to detect and classify objects [16,17]. Mauri et al. [22] mentioned the object detection

methods that are divided into two categories: (i) single-stage object detector and (ii) two-stage object detection detector. Several popular single-stage object detection algorithms are Single Shot Detector (SSD) [32], versions of You Only Look Once (YOLO) [18,19]; Two-phase object detection algorithms such as RCNN (Region proposal CNN) [20,21]. By using the proposal algorithms, [23] shows that the model for the media detection problem has the ability to extract good features and high results. As analysis above, we found that current methods focus on detecting space and identifying license plates. In this article, we therefore propose an automatic vehicle management system for street car parks by applying deep learning. Besides, our system can also identify each vehicle since it

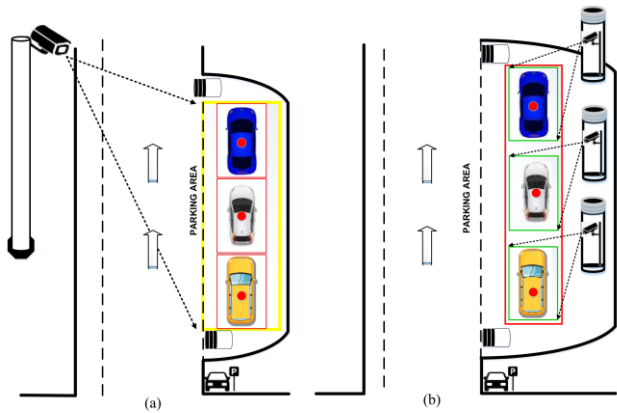


Fig. 1 System using (a) one camera, (b) multiple cameras

can be automatically charged electronically. Experimental results have high accuracy when comparing with existing methods. Therefore, it shows that the feasibility of the application system in practice is very high. The proposed parking model is discussed in detail in section 3. In part 3, the proposed algorithm of a parking system is discussed in detail. Part 4 presents experimental results achieved and evaluated. Finally, conclusions will be drawn in section 5.

3- Proposed System

Realistic requirements for the system include:

- Accurately detecting empty space;
- Identifying the vehicle based on the license plate.

During the analysis and design, the following cases occur: 1) a system using one camera and 2) a system using multiple cameras as shown in Fig. 1. For a system using one camera, the camera performs both functions. The system is low cost and easy to use. However, the license plate recognition function has low accuracy since it is not possible to recognize the number plate for cars with a long distance.

For a system using multiple cameras, each parking position will be installed with a camera with a low angle in order to be able to observe the license plate when the vehicle enters the parking. The advantage of system is its high accuracy. Its disadvantage is high energy consumption and cost.

Therefore, we propose to use two cameras to perform two separate functions in the paper. Specifically, a camera has a high resolution and is mounted in a low position for the purpose of detecting and identifying the license plate most accurately. The other camera will have a wide angle to maximize the parking area to detect and track objects entering the parking.

3-1- Proposal System

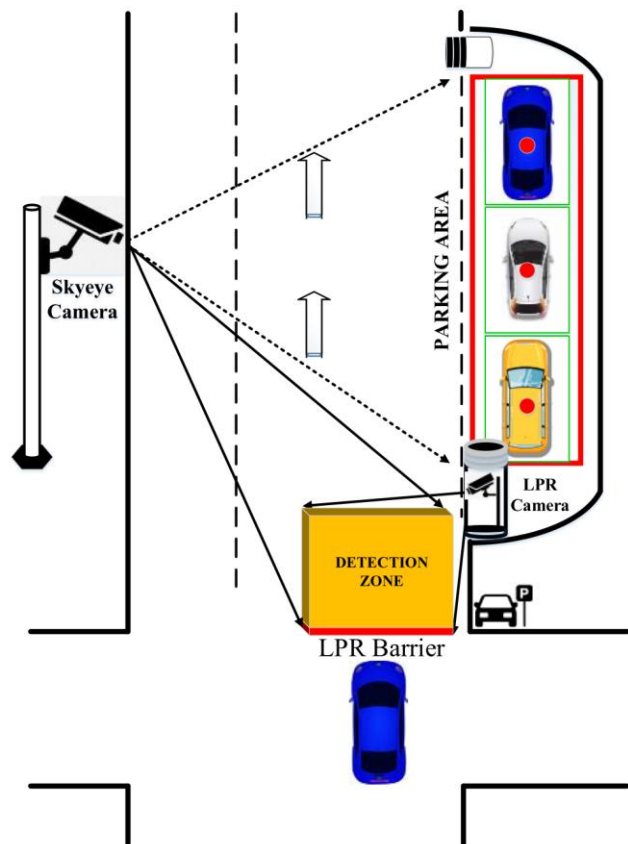


Fig. 2 Diagram of proposal system

Due to the close relationship of object detection to video and image analysis, the proposed and implemented object detection methods have achieved certain achievements recently. Traditional object detection methods are built on image characteristics drawn from experience. The performance of the methods is easily affected by the construction of complex clusters that combine low-level image characteristics and high-level features. With the rapid development of deep learning, as

well as computer science bring real-time response systems with very high accuracy to solve problems that exist in traditional methods.

The initial requirement of problem is to accurately detect the space position of car parking and control the vehicle entering by determining its identity. Our proposed system uses two cameras with two independent functions, Skyeeye and LPR cameras, respectively. We compare with real-world feasible systems as follows:

For an example, a smart car park for a condominium or planned parking lot that includes:

Objects are apartment car or planned parks. The system recognizes the license plate number correctly since the vehicle must stop in front of the barrier for procedures. From there, easily identify the vehicle's identity and collect the toll. However, this system is only suitable for planned parking lots and available facilities.

Other case is an image of a smart parking space with wireless sensor networks. Subjects are street parking lots. The system only detects whether the parking lot has space or not. However, the system does not recognize the license plate since it will not be able to identify the vehicle. Deploying the system will be expensive since it depends on the number of slots corresponding to the number of sensors needed. Besides, the system is affected by environmental conditions.

Therefore, we design the two respective main modules as follows:

- Skyeeye module has function to detect and track vehicles.
- ALPR module detects and identifies license plates.

Our proposal system is shown in Fig. 2. It includes:

- Skyeeye camera is a wide angle camera that covers the car parking.
- LPR camera is a high resolution camera to detect the license plate of vehicle.
- Detection zone is the area that detects when new vehicles enter.
- LPR barrier is a virtual barrier to detect the license plate of vehicle when passing the barrier.

With the management of information and locating the assistive vehicle for the operator, we use images to capture from a wide-angle camera (Skyeye camera) as input to the Skyeeye module that allows to determine the positions. Therefore, the system also detects the number of vehicles in the parking area. Furthermore, the Skyeeye camera also detects and tracks new objects entering the car parking.

With the function of identifying number plates to determine the identity of each vehicle when entering the car parking, LPR camera has a high resolution and long zoom capability.

There are two main difficulties of proposal system:

- One issue related to the accuracy of the LPR module is camera placement. When the vehicle

has entered the parking spaces, the license plate is likely to be obscured by the front and rear objects. Therefore, we have installed a license plate detection camera as shown in Fig. 2 to detect it before the vehicle enters the car parking to increase the performance of ALPR module.

- The second is a matter of synchronizing two modules in order to know how to be the same object. Our solution is to process each module sequentially.

The system operation is shown in Fig. 3 as follows:

- Skyeeye module is always active with the input of images collecting from the Skyeeye camera. After processing, information about parking status will be shared to users. In the case of a new vehicle entries, the Skyeeye module detects the vehicle in the detection zone and sends trigger signal to the LPR module.
- LPR module is activated to detect and recognize number plates. The results are the letters of the number plate.
- Assigning #ID to the newly detected vehicle with license plate characters and tracking the vehicle using centroid tracking algorithm based on [24] and checking if the vehicle enters the parking.
- Updating and displaying information.

Above is an overview of the proposal system. In next section, we will present more details about the modules.

3-2- Skyeeye Module

The Skyeeye module consists of two main blocks: 1) Vehicle detection and 2) vehicle tracking. Vehicle detection is one of the important applications of object detection problem in ITS. It is intended to extract vehicle-specific information from images or videos containing the vehicle. To solve current vehicle detection problems, such as vehicle classification, low detection accuracy and real-time non-response, we perform evaluation and comparison of state-of-the-art detection algorithms. Using the network architectures is to extract features to detect the object with one-time detectors such as SSD, YOLO on data sets such as ImageNet, COCO [30].

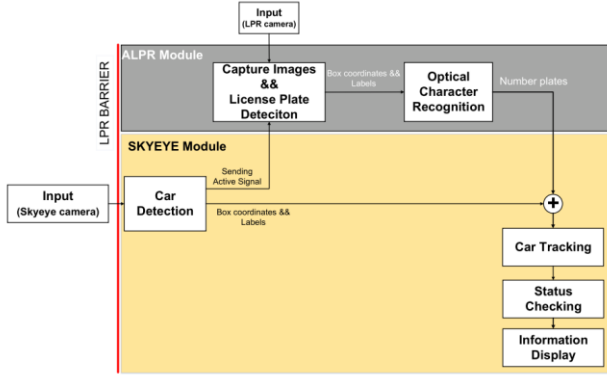


Fig. 3 The proposed system's operational modules

Vehicles are one of the basic objects present in many basic identification and detection datasets, such as PASCAL-VOC [25], ImageNet [26], and COCO [30]. Therefore, we decided to select a trained model for vehicle detection after comparing the performance of different network architectures as shown in Tab. 1.

Table 1: Object detection results on data set PASCAL VOC 2007 and MS-COCO

CNN	VOC07		COCO	
	FLOPs	mAP	FLOPs	mAP
SSD-512 [32]	90.2 B	74.9	99.5 B	26.8
SSD-300 [32]	31.3 B	72.4	35.2 B	23.2
YOLOv2 [33]	6.8 B	69.0	17.5 B	21.6
MobileNetv1-320 [34]	-	-	1.3 B	22.2
MobileNetv2-320 [35]	-	-	0.8 B	22,1

We use a model that has been trained on COCO data set to save time and initialize initial parameters to learn features faster. The using featured extractor is MobileNet_v2 paired with an SSD single-phase detector [32]. Overall, Single Shot Multibox Detector (SSD) [32] is used to reduce model size and complexity. It works by using multiple feature maps along a network. Therefore, the network can use this information to predict large objects through deeper layers, as well as predict small objects using shallow layers. We do not make any changes or refinements to the SSD- MobileNetv2 and only use it as a black box to merge outputs related only to one vehicle being a car and ignore the remaining layers. The process of implementing consists of two main phases: one is the training phase and the other is the test phase to evaluate the performance of the model as shown in Fig. 4. After detecting the object, it will track using the centroid tracking algorithm to check if the vehicle enters the car parking.

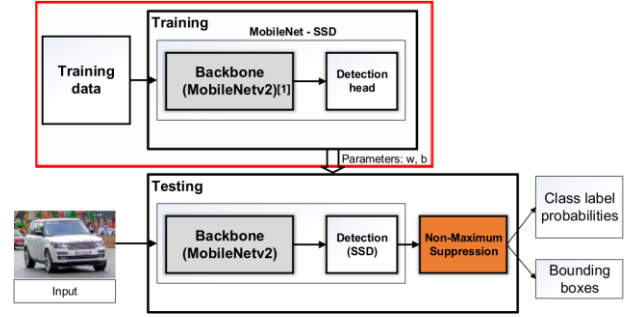


Fig. 4 Diagram of vehicle detection block implementation

Training Objectives: The training target of SSD is based on MultiBox [36,37]. It is extended to handle a wide variety of objects. Setting $x_{ij}^p = \{1, 0\}$ is specified for the first default box to the j^{th} truth of type p . The total loss function is the weighting of the localization loss (loc) and the confidence loss (conf) defining as follows:

$$L(x, c, l, g) = \frac{1}{N} \left(L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g) \right), \quad (1)$$

where N is the appropriate number of default boxes. If $N = 0$, the loss function is 0. The localization loss function is the L_1 loss smooth function [38] between the predicted box (l) and the truth box (g) as

$$L_{\text{loc}}(x, l, g) = \sum_{i \in \text{Pos}} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k \text{smooth}_{L1} \left(l_i^m - g_j^m \right), \quad (2)$$

$$g_j^{cx} = (g_j^{cx} - d_i^{cx}) / d_i^w; g_j^{cy} = (g_j^{cy} - d_i^{cy}) / d_i^h, \quad (3)$$

$$g_j^w = \log \left(\frac{g_j^w}{d_j^w} \right); g_j^h = \log \left(\frac{g_j^h}{d_j^h} \right);$$

The reliability loss function is the softmax loss over many layers (c) as

$$L_{\text{conf}}(x, c) = - \sum_{i \in \text{Pos}} x_{ij}^k \log \left(c_i^{\wedge p} \right) - \sum_{i \in \text{Neg}} \log \left(c_i^{\wedge 0} \right), \quad (4)$$

$$\text{Where } c_i^{\wedge p} = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}. \quad (5)$$

Training Details: For the training of SSD_Mobinetv2, we created a data set with 18000 images collecting from locations where the camera angle was installed according to the recommended system. The aim is to create a properly distributed data set to increase system performance. For each image, we manually labeled the objects of the car. From the selection of transformations to enhance the input, we then train the model using Tensorflow [39]. We trained the network with 300,000 mini-batch loops of 16 size. We used a standard optimization algorithm RMSPropOptimizer with momentum and decay values 0.9 and 0.9, respectively. Besides, using batch normalization technique after each

layer and the standard decay weight is set to 0.0001. The results will be presented in the following section.

3-3- ALPR Module

In order to be able to identify each vehicle based on its license plate, we use an automatic license plate recognition system (ALPR). In the module, we use an ALPR system that operates on a variety of scenarios [27]. One of the main advantages of system is its ability to detect number plates in a variety of contexts that allows a process of aligning the number plate before character recognition. Therefore, the system has the flexibility to detect and identify highly accurate number plates in independent test datasets using the same system parameters. ALPR has two main tasks: finding and identifying number plates in input images. Typically, it is divided into four missions, vehicle detection, number plate detection, character segmentation and character recognition. In [27], they combined the last two missions, OCR. The method implemented consists of three main blocks: vehicle detection, license plate display method, and OCR. With high-resolution LPR camera input image data, the module will first detect the media in the image. For each region, the curved plane object detection network (WPOD-NET) searches for the license plate containers and regress an affine transformation. This allows the license plate area to be edited into a rectangle like the one from the front view. These edited findings are sent to the OCR network for character recognition. The following is the block diagram of the ALPR module implementation as shown in Fig. 5.



Fig. 5 Block diagram for detecting and identifying license plate [27]

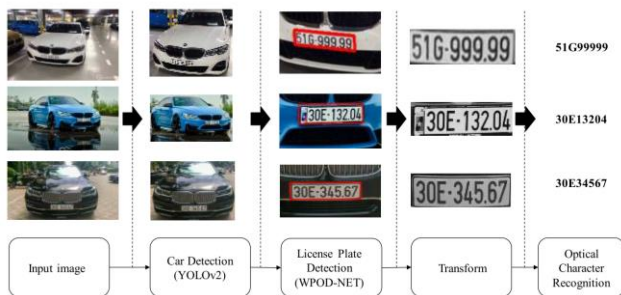


Fig. 6 Demonstration of the module implementation steps

Car Detection: Since vehicles are one of the basic objects that present in many basic detection and identification datasets (PASCAL-VOC [25], ImageNet [26], and COCO [30]), we do not decide to train a detector from scratch. Besides, we choose a trained model to

perform vehicle detection by considering several evaluation criteria such as AP, recall, and precision. Based on [27], we use the YOLOv2 network because of its fast execution speed, high precision and recall (76.8% mAP on the PASCAL-VOC dataset) [25]. The detected zones are then resized before inclusion in WPOD-NET to detect possible areas of number plates. As a rule of thumb, large sized input images allow detecting smaller objects. However, the downside is the increase in computation costs. Experiments have shown that if the license plate is taken from a front or front angle, the ratio between the size of the license plate and the vehicle's limit box will be high. However, this ratio tends to be smaller for the case of number plates taken from an oblique angle. Therefore, oblique frames should be changed to a larger size since the number plate can still be detected. More specifically, the coefficient of resizing is calculated as follows [27]:

$$f_{sc} = \frac{1}{\min\{W_v, H_v\}} \min \left\{ D_{\min} \frac{\max(W_v, H_v)}{\min(W_v, H_v)}, D_{\max} \right\}, \quad (6)$$

where W_v, H_v are the width and length of the limited box of the vehicle respectively with $D_{\max}=608; D_{\min}=208$.

License plate detection: License plates are rectangular and flat objects that are attached to each vehicle for the purpose of identifying it. In the module, we use the WPOD-NET network [27] to learn how to detect number plates in a variety of contexts and the regression coefficients of affine transformations to convert the curved plate into a rectangle. The detection process using WPOD-NET is illustrated in Fig. 7. Besides, the network input is resized by the vehicle detection unit. The result after being forwarded to the WPOD-NET network is a feature map with 8 channels - encoding the object/no-object probability and affine conversion parameters. To extract the warped license plate, the author first considers a fictional square of fixed size around the center position of a cell (m, n). If the probability of this object exceeds a certain detection threshold, a part of regression parameters will be used to construct an affine matrix that transforms the fictional square into number plate. The WPOD-NET network model architecture consists of a total of 21 convolution layers with 14 residual blocks [31]. All convolution sizes are 3x3. The ReLU trigger function is used on the entire network except for block detection. There are 4 layers of Max pooling of size 2x2 and stride 2 that will reduce the input size by a factor of 16. Finally, the detect block consists of two parallel convolution layers: (i) for calculating probabilities and triggering by the function. softmax, and (ii) for affine matrix regression with no trigger function.

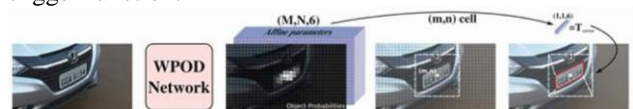


Fig. 7 Detection of license plates using WPOD-NET [27]

The loss function of the lattice is composed of two components [27]: the first part looks at the error between a deforming version of normal square and normalizing points of license plate; the second part handles the probability that there is or not object at position (m, n) . Combining the two components above, the loss function is defined as follows:

$$loss = \sum_{m=1}^M \sum_{n=1}^N [I_{obj} f_{affine}(m, n) + f_{affine}(m, n)], \quad (7)$$

where

$$f_{affine}(m, n) = \sum_{i=1}^4 \|T_{mn}(\mathbf{q}_i) - A_{mn}(\mathbf{p}_i)\|_1 \quad (8)$$

$$f_{probs}(m, n) = \log loss(I_{obj}, v_1) + \log loss(1 - I_{obj}, v_2) \quad (9)$$

In the context of the system, we propose the solution to implement ALPR module as shown in Figs. 8 and 9.



Fig. 8 Block diagram of the proposed ALPR module implementation



Fig 9: Illustration of ALPR module result

In the system, the image is collected from LPR camera that only contains the area with the license plate of the vehicle. Therefore, we skipped the vehicle detection step before detecting the license plate and adding a preprocessing block to increase the accuracy of character recognition. One of the factors that affect the character recognition block accuracy is the quality of input image. Therefore, we have used homomorphic filtering to minimize the problem. This technique uses a light reflection model. This model considers an image based on two components: 1) illumination on the field being viewed $L(x, y)$ (low frequency); 2) the reflective composition of the objects on the scene $R(x, y)$ (high frequency) is defined as follows:

$$I(x, y) = L(x, y) \cdot R(x, y) \quad (10)$$

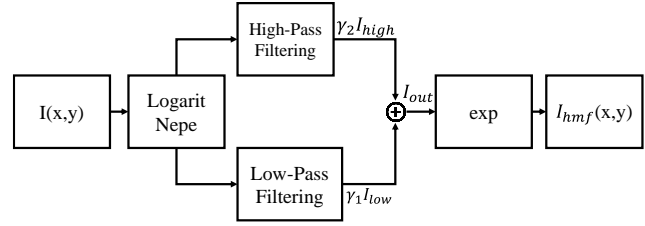


Fig. 10 Block diagram of preprocessing

The input image of length and width are H, W respectively. By creating a Gaussian filter of size (M, N) (where $M = 2.H + 1, N = 2.W + 1$) to reduce the jagged effect, low pass filters (H_{low}, H_{high}) are defined as follows:

$$H_{low}(x, y) = e^{-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma^2}}, \quad (11)$$

$$H_{high}(x, y) = 1 - H_{low} \quad (12)$$

After passing two filters, we get I_{low} and I_{high} . To add these two results, we have

$$I_{out} = \gamma_1 I_{low} + \gamma_2 I_{high} \quad (13)$$

where $\gamma_1 = 0.3, \gamma_2 = 1.5$.

Finally, the output image is defined as follows:

$$I_{hmf} = e^{I_{out}} \quad (14)$$



Fig. 11 Image before and after pretreatment

4- Experiment and Results

After the analysis and design process, we build a test scenario for the system and evaluate the achieved results to its feasibility. Evaluation method consists of two main stages: one is to evaluate the results achieved on each module, other is to evaluate the system.

4-1- Experiment Scenario

4.1.1 Realistic Background

Parking is located on a one-way lane as shown in Fig. 8.



Fig. 12 Actual parking lot

4.1.2 Circumstances Occurred

We divided into two cases:

- A car enters
- Many vehicles go in sequence.

4-2- Build the Data Set

With the proposal system, Skyeeye module retrieves image data from camera mounted on the overhead. We built a realistic reflection dataset to increase the performance of the module. Table 2 shows the results of the collecting data.

Table 2: Statistical table of collected data sets

Place	Features	Number of images	Number of objects
Tan Trieu K Hospital	Taken from camera phone, high angle view.	5400	6292
Tunnel of Thang Long	Taken from camera phone, high angle view	7800	9475
Street	Taken from camera phone, high angle view	3600	4425
Hanoi University of Science and Technology	Taken from camera phone, high angle view	1200	4000
Total		18000	24192



Fig. 13 Illustration of collecting data samples

4-3- Skyeeye Module Results

Table 3: The results obtained after performing the Skyeeye module

Criteria Method	AP@05 (%)	Precision (%)	Recall (%)	Processing time (second)	
				416 × 416	960 × 540
YOLOv2 [3], [5]	99.95	99.20	99.32	0.533	1.274
Skyeye module	99.48	99.28	99.18	0.273	0.829



Fig. 14 Illustration of a test example of the Skyeeye module

After labeling the self-collected data set, we retrained the vehicle detection model. The results are presented in Tab. 3.

The parameters are defined as follows:

- True Positive (TP): A correct detection. Detection with $IOU \geq$ threshold
- False Positive (FP): A wrong detection. Detection with $IOU <$ threshold
- False Negative (FN): A ground truth not detected
- True Negative (TN): We do not apply. It would represent a corrected misdetection. In the object detection task, there are many possible bounding boxes that should not be detected within an image. TN would be all possible bounding boxes that were correctly not detected (many possible boxes within an image). Therefore, it is not used by the metrics

The number of objects is 12226 and threshold is 0.5 in our paper. The parameters are shown in Tab. 4.

Table 4: Parameter setup

	Predicted as Positive	Predicted as Negative
Actual: Positive	TP = 12126	FN = 100
Actual: Negative	FP = 88	

From the results achieved, it shows that the Skyeeye module has a very high accuracy. Although the model's accuracy is lower than that of YOLOv2 [23], the processing time is only half.

In this article, we focus on research and propose a smart parking management system solution using multiple cameras. Therefore, when a solution is proposed, we accelerate the process of implementing the solution by installing cameras and collecting datasets that share the

actual context. On the other hand, the performance of deep learning application detection and identification models in particular comes from the fact that the quality of data and the amount of data that can cover all the contexts will be decisive for model training. Currently, we collect data in environmental conditions including: sunny, cloudy, rainy, etc. Our system achieved in hitting performance evaluation over the availability of data collected including the aforementioned contexts.

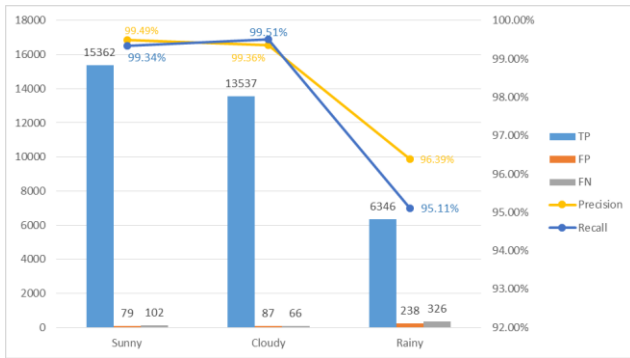


Fig. 15 Result of sunny, cloudy, and rainy environment

Our system still achieves high accuracy through the Precision and Recall evaluation criteria shown in Fig. 15. However, our system has not yet handled all the actual cases. In the future, we are implementing to improve system performance and work well in the following contexts, namely sunny, rainy, afternoon and evening.

4-4- ALPR Module Results

To evaluate the proposed method, we used two test datasets with different characteristics: 1) a car_long tuple and 2) a self-collection dataset called VN_LP, which is presented in Tab. 3.

Character Recognition: To segment and recognize characters on license plate after pre-processing, we use pre-trained YOLO network [27] to evaluate the effectiveness. The results are presented in Tabs. 5 and 6 and Fig. 16.

Table 5: Test data set

Data set	Characteristic	Quantity
Car long [28]	Number plate images from the front with fixing angle	998 (images)
VN_LP	Images of number plates from many different angles	1000 (images)

The following results are shown in Tab. 6.

Table 6: Test results of ALPR modules

Criteria	Car long [29]		VN_LP	
	Processing Time (s)	Accuracy	Processing Time (s)	Accuracy
YOLOv2 [29]	0.218	86.34%	0.481	70%
Module ALPR	0.142	94%	0.366	60%

The results show that the result of proposal solution has lower processing time and higher accuracy for our dataset. However, the accuracy is significantly lower for other datasets.



Fig. 16 Illustration of ALPR module test result

4-5- Proposed System Results

To evaluate the system's performance based on the results obtained from the Skyeeye and ALPR modules, we give a simple formula that is the average of two-module results. The following results are achieved by proposal system as shown in Tab. 7 and Figs. 17 and 18. The results show that the system has high accuracy for our dataset. Therefore, the system is highly feasible when applying in practice.

Table 7: Results achieved by the proposal system

Method	Accuracy of Skyeeye module (%)	Accuracy of ALPR module (%)	System	Processing Time (second)	
				416 x416	960 x540
YOLO v2 [23][27]	99.95	86.34	89.83	0.533	1.274
Proposal system	94.48	94	96.74	0.273	0.829

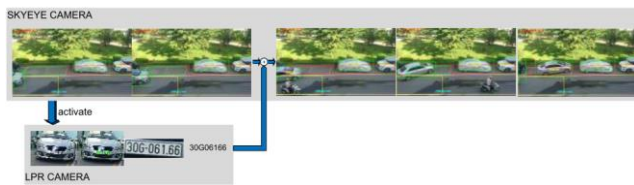


Fig. 17 Image of a vehicle entering a parking space

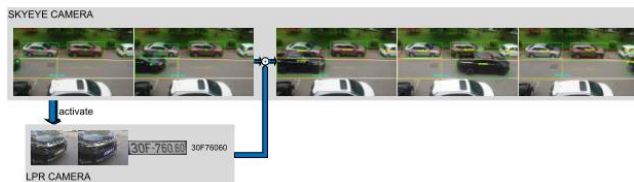


Fig. 18 Image of a vehicle not entering a parking space

5- Conclusion

The image-based proposal system in the article not only detects the location of parking space but also determines the vehicle's identity based on its respective license plate to share status information for user. Therefore, we can control and manage the parking space effectively. In addition to the application of the system to street parking spaces, the system can also be used for other purposes such as detecting illegal parking on the roadway as well as expanding functions. It helps to automatically collect electronic fees based on the license plate of vehicle that is linked to the owner's e-wallet. The system achieves high accuracy of 96.74% in actual setting. It points out the feasibility of the system into practice. However, there are many challenges that affect the performance of the system such as input image quality, detection algorithms, and identification. Therefore, we will try to improve the performance of the system by solving real-world problems in the future.

References

- [1] Z. Pala and N. Inanc, "Smart Parking Applications Using RFID Technology," *2007 1st Annual RFID Eurasia*, 2007, pp. 1-3.
- [2] J. P. Benson et al., "Car-Park Management using Wireless Sensor Networks," *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*, 2006, pp. 588-595.
- [3] A. Kianpisheh, N. Mustafa, P. Limtrairut, and P. Keikhosrokiani, "Smart Parking System (SPS) architecture using ultrasonic detector," *Int. J. Softw. Eng. its Appl.*, vol. 6, no. 3, pp. 51-58, 2012.
- [4] G. Ostojic, S. Stankovski, M. Lazarevic and V. Jovanovic, "Implementation of RFID Technology in Parking Lot Access Control System," *2007 1st Annual RFID Eurasia*, 2007, pp. 1-5.
- [5] J. Wolff, T. Heuer, Haibin Gao, M. Weinmann, S. Voit and U. Hartmann, "Parking monitor system based on magnetic field senso," *2006 IEEE Intelligent Transportation Systems Conference*, 2006, pp. 1275-1279.
- [6] N. H. H. Mohamad Hanif, Mohd Hafiz Badiozaman and H. Daud, "Smart parking reservation system using short message services (SMS)," *2010 International Conference on Intelligent and Advanced Systems*, 2010, pp. 1-5.
- [7] A. Sayeeraman and P. S. Ramesh, "Zigbee and gsm based secure vehicle parking management and reservation system," *J. Theor. Appl. Inf. Technol.*, vol. 37, no. 2, pp. 199-203, 2012.
- [8] H. Ichihashi, T. Katada, M. Fujiyoshi, A. Notsu and K. Honda, "Improvement in the performance of camera based vehicle detector for parking lot," *International Conference on Fuzzy Systems*, 2010, pp. 1-7.
- [9] R. Yusnita, F. Norbaya, and N. Basharuddin. "Intelligent parking space detection system based on image processing," *International Journal of Innovation, Management and Technology*, vol. 3, no. 3, pp. 232-235, 2012.
- [10] S. Banerjee, P. Choudekar and M. K. Muju, "Real time car parking system using image processing," *2011 3rd International Conference on Electronics Computer Technology*, 2011, pp. 99-103.
- [11] S. Khaled, and H. Tounsi, "Parking space detection system using video images," *Transportation Research Record: J. of the Transportation Research Board* 2537, no. 1, pp. 137-147, 2015.
- [12] S. H. Pratap, O. P. Uniyal, and K. Joshi, "An approach to implement cost efficient space detection technology with lower complexity for smart parking system," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 15, no. 3, pp. 415-419, 2015.
- [13] S. Saleh Al-Amri, N. V. Kalyankar, and Khamitkar S, "Image segmentation by using thershold techniques," *Journal of Computing*. vol. 2, no. 5, 2010.
- [14] S. Li, H. Dawood and P. Guo, "Comparison of linear dimensionality reduction methods in image annotation," *2015 Seventh International Conference on Advanced Computational Intelligence (ICACI)*, 2015, pp. 355-360.
- [15] R. Mehmood, R. Bie, H. Dawood and H. Ahmad, "Fuzzy clustering by fast search and find of density peaks," *2015 International Conference on Identification, Information, and Knowledge in the Internet of Things (IIKI)*, 2015, pp. 258-261.
- [16] A. R. Pathak, M. Pandey, S. Rautaray, "Application of deep learning for object detection," *Proceeding Comput. Sci*, vol. 132, pp. 1706-1717, 2018.

- [17] X. Wang, "Deep learning in object recognition, detection, and segmentation," *Trends Signal Process*, vol. 8, pp. 217–382, 2016.
- [18] M. Shafiee, B. Chywl, F. Li, A. Wong, "Fast YOLO: A fast you only look once system for real-time embedded object detection in video," arXiv:1709.05943, 2017.
- [19] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: unified, real-time object detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788.
- [20] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580-587.
- [21] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2017.
- [22] A. Mauri, R. Khemmar, B. Decoux, N. Ragot, R. Rossi, R. Trabelsi, R. Bouteau, R. Ertaud, J. Savatier, "Deep learning for real-time 3D multi-object detection, localisation, and tracking: application to smart mobility," *Sensors*, vol. 20, no. 2, pp. 1-15, 2020.
- [23] J. Sang, Z. Wu, P. Guo P, et al, "An improved YOLOv2 for vehicle detection," *Sensors (Basel)*, vol 18, no. 12, pp. 1-15, 2018.
- [24] J. C. Nascimento, A. J. Abrantes and J. S. Marques, "An algorithm for centroid-based tracking of moving objects," 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings. ICASSP99 (Cat. No.99CH36258), 1999, pp. 3305-3308, vol.6.
- [25] M. Everingham, L. V. Gool, C.K.I. Williams, J. Winn, A. Zisserman, "The pascal visual object classes (voc) challenge," *International J. of Computer Vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [26] G. R. Gonalves, S.P.G. Silva, D. Menotti, W.R. Schwartz, "Benchmark for license plate character segmentation," *J. of Electronic Imaging*, vol. 25, no. 5, pp. 1–5, 2016.
- [27] S. M. Silva and C.R. Jung, "License plate detection and recognition in unconstrained scenarios," *Lecture Notes in Computer Science*, vol 11216, Springer, Cham, pp. 593–609, 2018.
- [28] P.N. Huu, L.H. Bao, and H.L. The, "Proposing an image enhancement algorithm using CNN for applications of face recognition system," *J. Adv. Math. Comput. Sci.*, vol. 34, no. 6, pp. 1–14, 2020.
- [29] Computer Vision, "Aggregate data", Online: <https://thigiacmaytinh.com/tai-nguyen-xu-ly-anh/tong-hop-data-xu-ly-anh/> (accessed Jun. 30, 2020).
- [30] T.Y. Lin, et al, "Microsoft COCO: Common objects in context," in *ECCV 2014: Computer Vision – ECCV 2014*, vol. 8693, pp. 740-755, 2014.
- [31] K. He, X. Zhang, S. Ren and J. Sun, "Deep residual learning for image recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.
- [32] W. Liu, et al., "SSD: Single shot multiBox detector," In *ECCV 2016: Computer Vision – ECCV 2016*, vol. 9905, pp. 21-37, 2016.
- [33] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6517-6525.
- [34] A. G. Howard, et. al, "Mobilenets: efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, pp. 1-9, 2017.
- [35] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510-4520.
- [36] D. Erhan, C. Szegedy, A. Toshev and D. Anguelov, "Scalable object detection using deep neural networks," 2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2155-2162.
- [37] C. Szegedy, S. Reed, D. Erhan, D. Anguelov, "Scalable, high-quality object detection," arXiv:1412.1441 v3, pp. 1-10, 2015.
- [38] R. Girshick, "Fast R-CNN," 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440-1448.
- [39] M. Abadi, et. al, "TensorFlow: a system for large-scale machine learning," In *Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation (OSDI'16)*. USENIX Association, USA, pp. 265–283, 2016.

Phat Nguyen Huu received his B.E. (2003), M.S. (2005) degrees in Electronics and Telecommunications at Hanoi University of Science and Technology (HUST), Vietnam, and Ph.D. degree (2012) in Computer Science at Shibaura Institute of Technology, Japan. Currently, he lecturer at School of Electronics and Telecommunications, HUST Vietnam. His research interests include digital image and video processing, wireless networks, ad hoc and sensor network, and intelligent traffic system (ITS) and internet of thing (IoT). He received the best conference paper award in SoftCOM (2011), best student grant award in APNOMS (2011), hisayoshi yanai honorary award by Shibaura Institute of Technology, Japan in 2012.

Loc Hoang Bao received his B.E. (2020) degrees in Electronics and Telecommunications at Hanoi University of Science and Technology (HUST), Vietnam. Currently, he is the master student at School of Electronics and Telecommunications, HUST, Vietnam. His research interests include digital image, video processing, and embedded system.