

Perspicacious Vacant Car Parking Detection System

Nita Saware¹, Sneha Waje², Mangesh M Kore³, Tejaswini Patil⁴, Preeti Nanaware⁵

Professor, Dept. of Computer Engineering, Jaywantrao Sawant College Of Engineering, Hadapsar, Pune-411028, India¹

U.G Student, Dept. of Computer Engineering, Jaywantrao Sawant College of Engineering, Hadapsar, Pune-411028, India²

U.G Student, Dept. of Computer Engineering, Jaywantrao Sawant College of Engineering, Hadapsar, Pune-411028, India³

U.G Student, Dept. of Computer Engineering, Jaywantrao Sawant College of Engineering, Hadapsar, Pune-411028, India⁴

U.G Student, Dept. of Computer Engineering, Jaywantrao Sawant College of Engineering, Hadapsar, Pune-411028, India⁵

Abstract - : Traditional car parking needs man power at large scale. Also there is no facility for pre-booking of the slot, and no accurate bills. Therefore, there is a need to design and implement an automated, advanced and smart oriented assistance parking system to monitor the above mentioned issues and help to improve the driver parking skills and reduce the risk for car damages. There is also a need of designing a subsystem which can issue a payment receipt analyzing and calculating the time for which the parking space was acquired since the car entered in the parking space. Here it demands for implementation of various methods or techniques under the domains: Artificial Intelligence, Image Processing and Android Applications. It includes issues such insufficient illumination, complicated lightning conditions in which we have to alleviate the interference of unpredictable lightning changes and shadows. We also have to filter out the distortions appeared in the observed image due to inter-object occlusion and perspective distortion.

Secondly, for automating, guiding, reserving and notification services to the authenticated user it requires implementation of the technologies and techniques of Android.

Key Words: iSCAPS, Bayesian Hierarchical Framework (BHF), PS, NFC, IoT, RFID, Pattern search(PS)

1. INTRODUCTION

Initially car parking were totally manual. External person had to guide the user for parking the car in the available free parking slot. As it was manual, it had many errors in the bill generation process. Later a system was developed in which an external person guided the user to the available free parking slot but electronic boom barriers were used to control the entry and exit of the cars.

Later a system, iSCAPS, was proposed in which the concept of e-valet was used in which user can reserve the parking slot using an android application through smart phone. Instead of using conventional monthly pass or ticket, this system uses NFC (Near Field Communication). In this system, at entrance and exit sensors are used to detect the arriving car. LCD displays are used to show the number of available free parking slots at the entrance. Using these sensors accurate bill generation can be done as these sensors automatically senses the entry and exit time.

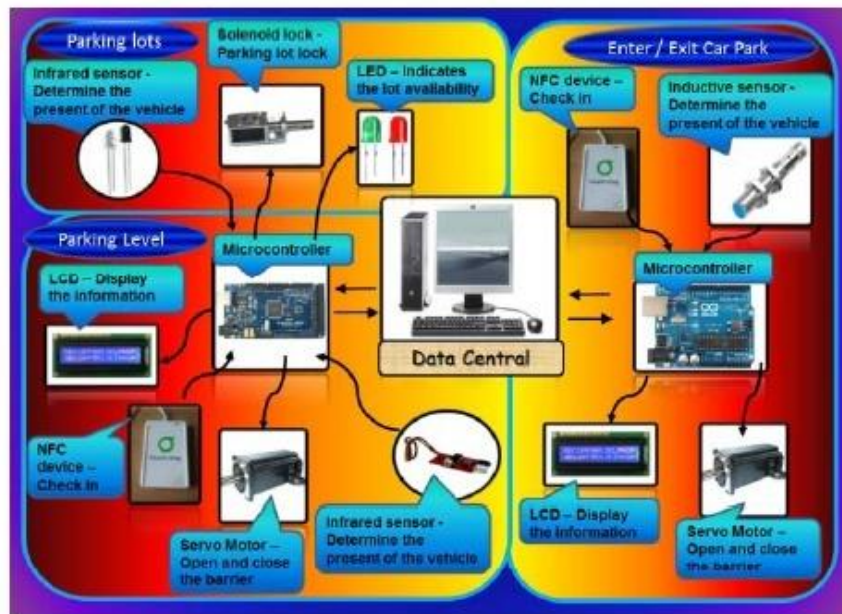


Fig. the overview of the process of iSCAPS with NFC technology

2. HISTORY & BACKGROUND

Vacant car parking detection have the first issue about how to obtain well exposed images for inference: For this a plane based Bayesian Hierarchical Framework (BHF) is proposed. Second issue is about how to improve the performance of vacant parking space detection and how to speed up the system for practical applications [1]. The method of combining background subtraction and vertex-edge features are followed. It uses color features and corner features, through training SVM classifier to get the result. [2] ICampus parking allocation using patterns suggests algorithms such as Pattern search (PS) and Pattern search warn(PSwarn) for solving the paring detection problem. [3].Cloud-based parking uses three components as Cloud Tier, OSGi Web Servers Tier, Mobile Applications Tier but don not solve ground level issues [4].The temporal clustering and anomaly detection deals with the various algorithms and their use for clustering parking spaces [5].Smart car parking systems with guidance and damage notification gives route to allocated parking slot and inform in case of any damage to car

[6].An innovative system for car parking work separately for entrance, parking allocation and exit phenomenon [7]The Advanced License Plate Recognition System, consists of two section: Radio frequency Identification (RFID) based detection and license plate recognition by using Open Computer Vision(CV). RFID is an automatic secure and convenient identifier which identifies the objects automatically. Automatic license plate recognition(ALPR) is used to obtain the license number of the vehicles written on the license plate and it possesses various applications but also many challenges [8].Parking availability based on sensor-enabled network uses the data and select some key parameters to form three feature sets to use as the input for the parking availability prediction models that we derive. The algorithms used for modeling the occupancy rate and for prediction are: 1)Regression Tree 2)Support Vector 3)Neural Networks [9]. The entering and leaving from the parking lot is commanded by an Android based application. On allocation of free slot, the car will further trace its path to free parking spot [10].

3. PERFORMANCE REVIEW TABLE

Sr.No.	Title	Techniques	Parameters Achieved
1.	Vacant Parking Space Detection Based on Plane-based Bayesian Hierarchical Framework (Proposed Method)	Bayesian Hierarchical Framework (BHF)	Performance, consistency
2.	A Method of Parking space detection based on image segmentation and LBP	Novel parking space detection, LBP	Better feature extraction, good detection results
3.	Campus Parking Space Allocation Distribution Using Pattern Search and Particle Swarm Pattern Search Algorithms	Particle Swarm Pattern Search	Good and feasible solution for college campus place.
4.	A Cloud-Based Intelligent Car Parking Services for Smart Cities	Cloud computing, IoT	Best availability and guidance to nearest parking slot
5.	Smart Car Parking: Temporal Clustering and Anomaly Detection in Urban Car Parking	Farthest First(FF) Clustering, Expectation Maximization(EM) Clustering, One-Class SVM, IoT.	Reliability, low cost, maximum productivity
6.	Intelligent Parking System for Car Parking Guidance and Damage Notification	Ultrasonic sensors, shock sensors	Damage notification, guidance to free space
7.	iSCAPS - Innovative Smart Car Park System integrated with NFC technology and e-Valet function	NFC, Radio Frequency Identification(RFID)	Automatic bill generation, reservation in advance
8.	Advanced License Plate Recognition System For Car Parking	RFID,CV	High speed, accuracy, uniqueness
9.	Parking Availability Prediction for Sensor-Enabled Car Parks in Smart Cities	Regression Tree, Support Vector, Neural Networks.	Feasibility, security, robustness
10.	Automated Car Parking System Commanded By Android Application	Autonomous searching method, Development of an Android application	Automation, Security

4. DESIGN ISSUES AND PROPOSED SYSTEM

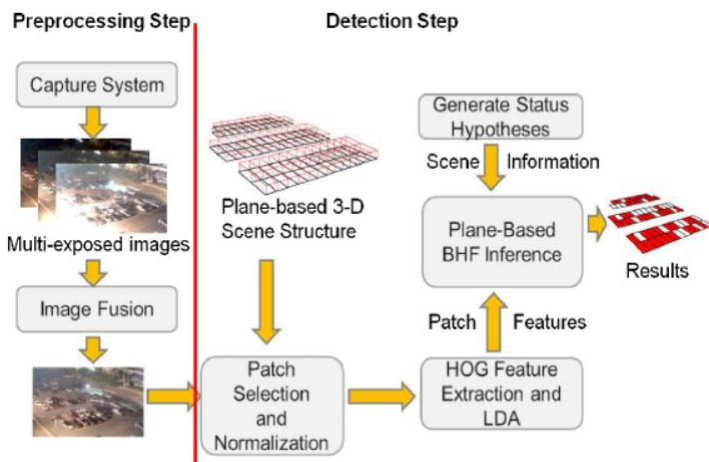


Fig.1 System flow of the proposed method

A. Plane-Based Structure and Feature Extraction

The figure shows the overall flow of the system to detect the vacant space or slot in the parking lot.

As shown the images are captured in the form of image bags. To get multi exposure images, we use the AXIS M1114 IP camera which can adjust the exposure value (EV) during image capturing. By using the software development kit (SDK) provided by AXIS, we capture images from a short exposure period to a long exposure period in a cyclic manner with the period of N image frames, as illustrated in Fig. 1 In our system, the longest and shortest exposure period is 3 s and 0.33 s, respectively. Then by using the two-step exposure fusion method to combine every N images, we get images of improved contrast.

In this approach, the parking is treated as spaces as a set of cuboids. Each cuboid is composed of six patches, based on the 3-D cuboid model, we represent the structure of parking by a set of 3-D planar surfaces. By projecting those 3-D surfaces onto the image, we get image patches of parallelogram shape. These patches are to be used for the status inference of parking spaces.

To overcome perspective distortion, we normalize each image patch into a rectangle, with Rl pixels in length and Rw pixels in width. After that, we extract features from the normalized patches. For feature

extraction, we adopt the HOG feature extraction method which is less affected by shadows and the changes of illumination. To extract HOG features, a normalized image patch is regularly segmented into non overlapping cells, with each cell containing $Cl \times Cw$ pixels. In total, would be $(Rl/Cl) \cdot (Rw/Cw)$ cells in each normalized patch. For each cell, a histogram of oriented gradients, is built. Each histogram has Hb histogram bins. By combining the histograms of all cells in the normalized patch, we obtain the HOG feature. In our system, the parameters $(Rl \times Rw, Cl \times Cw, Hb)$ are empirically chosen to be $(64 \times 32, 16 \times 16, 10)$. That is, each normalized patch contains eight cells and the dimensions of its HOG feature is 80.

In Fig. 2 illustrate the processes of patch normalization and HOG feature extraction. As will be explained later, these high-dimensional HOG features will be converted into low dimensional features via LDA so that the following inference process can be implemented in a more efficient way.

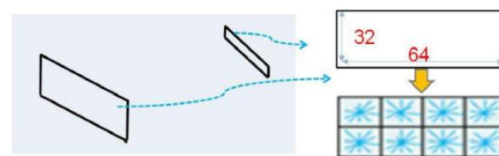


Fig. 2 : Patch normalization and HOG feature extraction.

B. Patch Classification:

For patch classification in the proposed plane-based BHF framework. As mentioned before, in our plane-based model, each parking space is approximated as a cuboid with six 3-D planar surfaces. We classify these surfaces into four different types: ground surface (G), side surface (S), front (or rear) surface (F), and top surface (T). Via perspective projection, these four types of planar surfaces are projected onto four types of image patches:

G-patch, S-patch, F-patch, and T-patch.

Due to the inter-object occlusion in the 3-D space, we further classify each type of image patch into a few sub-classes.

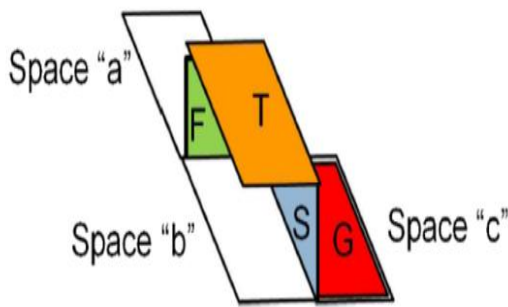


Fig 3. Illustration for the subclass definitions of image patch.

Here, without loss of generality, we use a camera configuration with a 45-degree view to explain the proposed patch classification. For the G-patch of the parking space “c” in Fig. 3, its image content is affected not only by the status of the parking space “c” but also by the status of the adjacent parking space “b.” Depending on whether these two parking spaces are occupied or vacant, there are four types of image patterns according to four different parking statuses: 1) “c” is occupied while “b” is vacant; 2) “c” is vacant while “b” is occupied; 3) both “c” and “b” are occupied; and 4) both “c” and “b” are vacant. Similarly, for the S-patch shared by “b” and “c” or the F patch shared by “a” and “b,” there would be four major kinds of image patterns. For the T-patch of the parking space “b,” on the other hand, we may either classify its image patterns into four sub-classes that relate to the four status combinations of the spaces “b” and “c,” or into eight sub-classes that relate to the eight status combinations of “a,” “b,” and “c.” In our experiments, for the sake of simplification, we choose the four subclass classification for T-patches.

Given a parking lot, we first set up an IP camera on the roof of a building near the parking lot. The camera is geometrically calibrated to obtain the 3-D to 2-D projection model and to construct the 3-D plane-based scene model. After that, we capture a few image sequences of the monitored parking lot and extract plentiful image patches for each type of planar surface. For each image patch, we manually collect its patch label and extract its HOG feature from the normalized patch. Based on the labeled patch type and the HOG feature, we learn the conditional probability function.

$$p(\mathbf{o}|l) \equiv \frac{1}{(2\pi)^{\frac{3}{2}}|\Sigma_l|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{x} - \mu_l)^T \Sigma_l^{-1}(\mathbf{x} - \mu_l)\right\}$$

where \mathbf{o} denotes the observed feature of an image patch, for each of the four subclasses, say l , we model the conditional probability function of observing the 3-D feature \mathbf{x} given l as a Gaussian function and 3-D features for the four subclass of the surface type $T, l \in L$ denotes the label of the image patch.

C. Inference of Parking Status

Once we have learned the conditional probability function $p(\mathbf{o}|l)$, we can start to infer the status of parking spaces in new coming images. For example, given an image patch, we first determine its surface type based on the pre-established plane-based 3-D scene model and the pre-calibrated 3-D to 2-D transformation. After that, based on the surface type, the image patch \mathbf{o} , we obtain the likelihood function $p(\mathbf{o}|l)$. By finding the label that maximizes the likelihood function, we can classify the image patch into one of the four subclasses and then deduce the status of the corresponding parking space. This maximum likelihood approach is pretty simple and efficient.

However, the deduced classification label, which depends on a small image patch, provides only local decision of the parking status. The inevitable errors in subclass classification may lead to a wrong decision of the parking status. To unify the 3-D scene information and patch classification results for vacant parking space detection, we propose a three layer hierarchical framework, named plane-based BHF, as shown in Fig 4

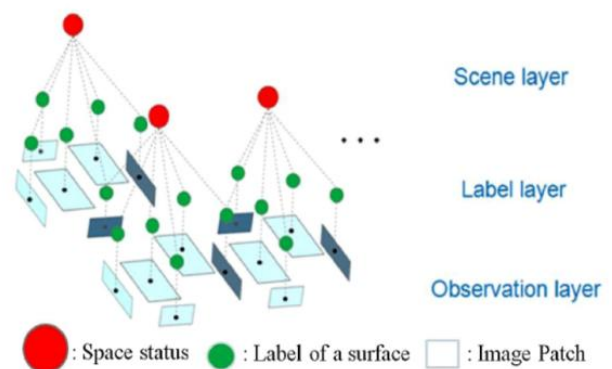


Fig 5 : Proposed plane-based BHF

Based on this framework, we build the connections between the 3-D scene model and 2-D image patches in a plane-based manner so that we can infer the statuses of parking spaces in a unified way. The proposed hierarchical framework consists of three layers:

the scene layer (*SL*), the label layer (*LL*), and the observation layer (*OL*). Each layer contains either nodes or parallelograms. In the scene layer, each node indicates the status of a parking space. For example, the node *sj* in *SL* denotes the status of the *j*th parking space, which can be either occupied or vacant.

In the observation layer, each parallelogram represents an image patch obtained by projecting the corresponding 3-D surface onto the image. These image patches in the observation layer provide image features for patch classification.

Between the scene layer *SL* and observation layer *OL*, we design a label layer *LL*. Each node in the label layer indicates the inferred label of the corresponding image patch. The label can be one of the 16 labels in the aforementioned label set *L*. The label layer *LL* links to both the scene layer *SL* and the observation layer *OL*.

These links play important roles in connecting the 3-D scene model and the 2-D image observation for the inference of vacant parking spaces. The links between *SL* and *LL* pass the information from the 3-D scene model to enforce consistent labels in the labeling layer. On the other hand, the links between *OL* and *LL* convey the likelihood messages from image patches.

To utilize both image observations and 3-D scene information for parking status inference, in our framework, we denote a status hypothesis *St* as a combination of the status of the parking spaces. For example, as shown in Fig. 11, if we consider only three parking spaces in the parking lot, the status hypothesis *St* can be one of the eight combinations: $\{(v, v, v), (v, v, o), (v, o, v), (v, o, o), (o, v, v), (o, v, o), (o, o, v), (o, o, o)\}$, where “v” denotes vacant and “o” denotes occupied. On the other hand, we denote our image observation as a set of image patches:

$$O = \{o_i\}_{i=1 \sim N},$$

where *i* is the patch index and *N* denotes the total number of image patches in the parking lot. For the three parking spaces in Fig. 6,

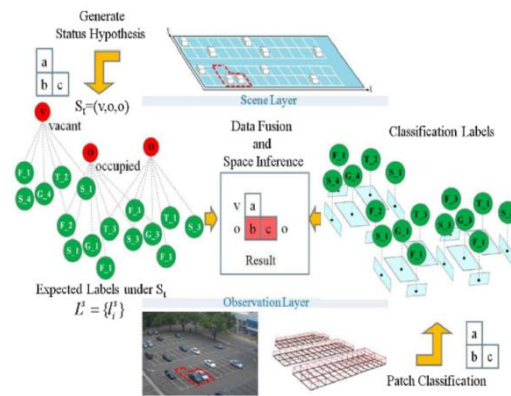


Fig. 6 Example of parking status inference in the proposed framework.

there are $N = 16$ image patches in total. Based on the above definitions, the inference of the optimal parking status

is equivalent to the maximization of $p(St | O)$. That is

$$S_t^* = \arg \max_{St} p(St | O). (3)$$

To solve (3), we treat the set of classification labels *L* as a hidden layer that bridges the gap between *St* and *O*. Based on

a status hypothesis, like the $St = \{v, o, o\}$ hypothesis in Fig. 11, we deduce the expected label set $LSt = \{l_{St}^i\}_{i=1 \sim N}$ for all *N* patches.

Here, the superscript in *LSt* indicates that this label set is constrained by the status hypothesis *St*. Since there is

a one-to-one mapping between *St* and *LSt*, we have

$$p(L|St) = \delta(L - LSt) \text{ and } p(O|St) = \int p(O|L) p(L|St) dL = \int p(O|L) \delta(L - LSt) dL = p(O|LSt). (4)$$

Moreover, based on the Bayes' rule, we can rewrite (3) as

$$S_t^* = \arg \max_{St} [p(O|St) p(St)] = \arg \max_{St} [p(O|LSt) p(St)]. (5)$$

In our system, $p(O|LSt)$ is formulated as

$$p(O|LSt) = \prod_i^N p(o_i | l_i^{St}) p(St) (6)$$

where we assume that the observation nodes $\{o_i\}_{i=1 \sim N}$ are conditionally independent once if the labels of the labeling layer are given.

In (6), $p(o_i|I_i^{S_i})$ represents how likely the image observation o_i on the i th image patch belongs to the patch label $I_i^{S_i}$. Note that $I_i^{S_i}$ is the expected label of the i th patch under the status hypothesis S_t . Moreover, the probability function $p(S_t)$ in (5) represents the prior knowledge about the parking statuses in the parking lot. In our system, we simply assume $p(S_t)$ to be uniformly distributed. Under this uniform distribution assumption, the $p(S_t)$ term in (5) can be ignored and we get the final formulation as follows:

$$S_t^* = \arg \max_{S_t} [(\prod_i^N p(o_i|I_i^{S_i})) p(S_t)] = \arg \max_{S_t} [(\prod_i^N p(o_i|I_i^{S_i})] \quad (7)$$

In above equation, the calculation of $p(o_i|I_i^{S_i})$ is based on the pre-learned conditional probability model and the expected label $I_i^{S_i}$ for the i th image patch. To find the optimal status hypothesis S_t^* , an exhaustive search over all possible status hypotheses is used. However, for a parking lot containing tens of parking spaces, the number of status hypotheses would be extremely high. Hence, instead of generating the status hypotheses for all parking spaces in the parking lot at one time, we adopt a sequential block-wise approach. In Fig. 7(a), we illustrate how we perform the generation of status hypotheses. In this example, we sequentially infer the parking status from left to right.

Note that there is no difference if we infer the status from right to left. Due to inter-object occlusion, we need to take into account the relevant parking spaces as we infer the status of a parking space. In our system, we consider six spaces at one time and determine the parking statuses of the two central parking spaces. In the example in Fig. 7(a),

these parking spaces whose statuses have already been inferred are labeled as either vacant (v) or occupied (o). The region within the red boundary indicates the six parking spaces to be considered at this moment. Within the region, the parking spaces to be inferred are marked by yellow squares while the relevant parking spaces are marked by green triangles. For these six parking spaces, a plane-based BHF framework is built. Since the statuses of the two parking spaces in the left have already been inferred, we only need to generate 16 status hypotheses relating to the 16 status combinations of the parking spaces A, B, C, and D.

By calculating the value of $\prod_i^N p(o_i|I_i^{S_i})$ for each of the 16 hypotheses, the optimal combination can be picked and the status of the two central parking spaces is inferred.

Based on this sequential way to generate status hypotheses, the system complexity grows only linearly as the number of parking spaces increases. To detect the parking status on the border, the inference process is slightly modified. As shown in Fig. 7(b), the red boundary indicates the six parking spaces to be considered and there are four parking spaces marked by the yellow squares. These four parking spaces need to be inferred at the same time. Similar to the aforementioned inference process, a plane based BHF framework for six parking spaces is built. However, without knowing the two parking spaces on the left, we need to test 64 status combinations of the six parking spaces. By finding the optimal status combination, the status of the four yellow-marked parking spaces is inferred. In the proposed plane-based BHF framework, the inclusion of the 3-D scene information provides a few benefits. First, the image patches under analysis can be systematically selected via geometric projection and the surface type of each image patch can be easily determined.

This allows us to apply LDA to deal with four surface types separately. Second, the status hypotheses in the 3-D space can be systematically converted into expected classification labels that consistently relate relevant image patches. This automatically avoids the inconsistency problem mentioned in the aforementioned maximum likelihood approach.

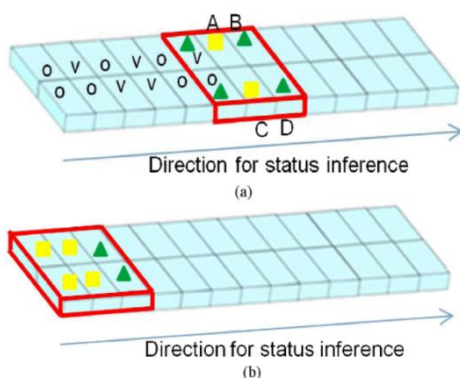


Fig 7 (a) Illustration of status hypothesis generation and the order of parking status inference. (b) Status hypothesis generation on the border

5.RESULTS AND ANALYSIS

These results includes the results and graphs related to the survey of proposed method

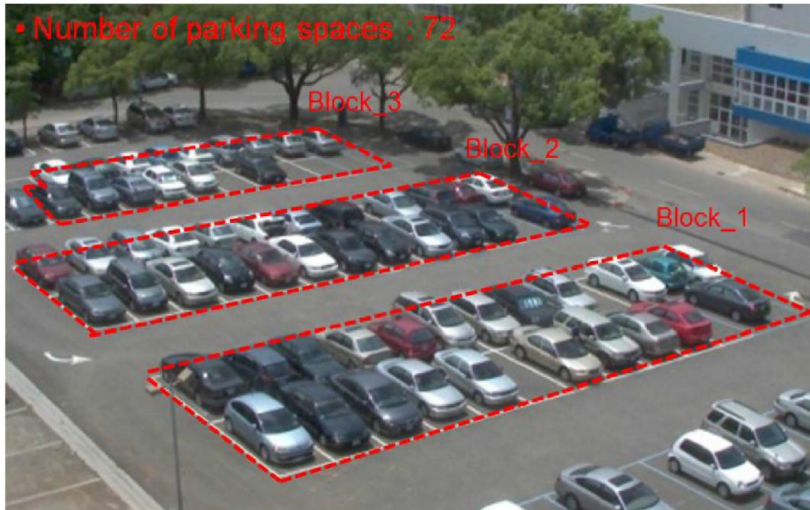


Fig. Experiment environment for evaluating the influence of perspective distortion.

To quantitatively evaluate the performance of our system, we calculate false positive rate (FPR), false negative rate (FNR), and accuracy (ACC). Their definitions are expressed as follows:

FPR = number of parked spaces being detected as vacant / total number of parked spaces

FNR = number of vacant spaces being detected as parked / total number of vacant spaces

ACC = number of correct detection in both parked and vacant spaces / total number of tested spaces

1) Daytime Performance Evaluation:

Table1 : Space Detection Results in the Daytime

Data Set ID	Number of tested Spaces			Proposed Method		
	vacant	Parked	Total	FPR	FNR	ACC
Typical(DS_1)	4123	7397	11520	0.280%	1.190%	99.39%
Sunny(DS_2)	8121	3399	11520	0.088%	0.860%	99.37%
Cloudy(DS_7)	5353	6167	11520	0.047%	0.260%	99.85%
Rainy(DS_4)	3544	6896	10440	0.640%	0.370%	99.45%

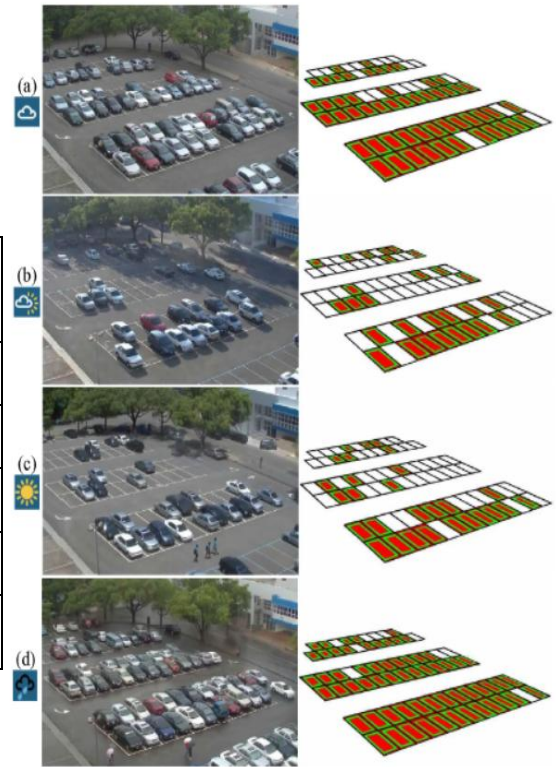


Fig: Results of vacant space detection in the daytime.

(a) Cloud day. b) Typical day. (c) Sunny day. (d) Rainy day.

Green rectangles indicate the ground truth and red rectangles indicate the detection results.

2) Night time Performance Evaluation:

	#of tested spaces			Proposed Method		
	vacant	Parked	Total	FPR	FNR	ACC
Low Illumination(DS_5)	4165	3035	7200	1.610%	2.420%	97.92%
Medium Illumination(DS_6)	4569	2631	7200	0.800%	1.730%	98.61%
High Illumination(DS_7)	1428	2172	3600	0.46%	3.29%	98.42%



Table2: Space Detection Results in the Nighttime
Fig: Result for vacant space detection in nighttime

6. CONCLUSION

Intelligent vacant car parking detection system is based on artificial intelligence along with Android application development. It is useful for users as it provides secure and reliable parking and pre-reservation of parking slot. Smart vacant car parking detection system will become rapid and an efficient way of parking in smart cities and projects related to it. It also simplify conflicts occurring due to absence of pre-reservation and requires minimum number of manpower. Results are quickly transferred to centralized database and ultimately to the Android application.

7. REFERENCES

- [1] Wang Lixia, Jiang Dalin, "A Method of Parking space detection based on image segmentation and LBP", Fourth International Conference on Multimedia Information Networking and Security, 2012 IEEE DOI 10.1109/MINES.2012.27.
- [2] Aderemi O. Adewumi, Luke O. Joel and Babatunde A. Sawyerr, "Campus Parking Space Allocation Distribution Using Pattern Search and Particle Swarm Pattern Search Algorithms", SCIS&ISIS 2014, Kitakyushu, Japan, December 3-6, 2014.
- [3] Zhanlin Ji, Ivan Ganchev, Máirtín O'Droma and Xueji Zhang "A Cloud-Based Intelligent Car Parking Services for Smart Cities", IEEE, 2014.
- [4] Yanxu Zheng, Sutharshan Rajasegarar, Christopher Leckie, Marimuthu Palaniswami, The University of Melbourne, Australia, "Smart Car Parking: Temporal Clustering and Anomaly Detection in Urban Car Parking", 2014 IEEE Ninth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP) Symposium on Information Processing Singapore, 21-24 April 2014.
- [5] Sanaa Alfatihi, Soukaina Chihab and Yassine Salih Alj, School of Science and Engineering, Al Akhawayn University, Ifrane, Morocco, "Intelligent Parking System for Car Parking Guidance and Damage Notification", 2013 4th International Conference on Intelligent Systems, Modelling and Simulation.
- [6] Jin Teong Ang, Siew Wen Chin, Jia Hui Chin, Zyn Xian Choo, Yu Ming Chang, "iSCAPS - Innovative Smart Car Park System integrated with NFC technology and e-Valet function", IEEE June-2013.
- [7] Judith Sen E, Deepa Merlin Dixon K, Ansy, Jinesh K J, "Advanced License Plate Recognition System For Car Parking", International Conference on Embedded Systems - (ICES 2014).
- [8] Yanxu Zheng, Sutharshan Rajasegarar, Christopher Leckie National ICT Australia Victoria, Dept. of Computing and Information Systems, The University of Melbourne, Australia, "Parking Availability Prediction for Sensor-Enabled Car Parks in Smart Cities", 2015 IEEE Tenth International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP) Singapore, 7-9 April 2015.
- [9] Mrs. D.J. Bonde, Rohit Sunil Shende, Akshay Sambhaji Kedari, Ketan Suresh Gaikwad, Amol Uday Bhokre, "Automated Car Parking System Commanded By Android Application", 2014 International Conference on Computer Communication and Informatics (ICCCI - 2014), Jan. 03 - 05, 2014, Coimbatore, INDIA.
- [10] C. C. Huang, S. J. Wang, Y. J. Chang, and T. Chen, "A Bayesian hierarchical detection framework for parking space detection," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, Apr. 2008, pp. 2097- 2100.