Edge Detection Based Feature Extraction for Intelligent Car Parking System

K. Kalirajan[#1], R.Jamuna*[2], G. Shanthi[3], L.Raja[4], G. Harikrishnan[5]

[#]*Associate Professor, Department of ECE, KPR Institute of Engineering and Technology, Coimbatore.*

**Assistant Professor (S.G), Department of ECE, SNS College of Engineering, Coimbatore.*

[3,4]*Associate Professor, Department of ECE, Vel Tech, Chennai.*

[5]*Associate Professor, Holy Grace Academy of Engineering. Kerala.*

[1]kalirajan.k@kpriet.ac.in

[2]jamuname9@gmail.com

[3]sdrshanthi@veltechengg.com,[4]rajabvni@gmail.com,[5]harichandanam780@gmail.com

***Abstract*— This article presents a novel feature extraction approach for intelligent car parking system. The proposed approach makes use of Log-Gabor filter followed by contour analysis to find vacant and filled slots. The wrongly parked cars if any are also identified and message alert will be sent to that particular vehicle driver for immediate action. The camera sensor is used to continuously monitor the parking slots. The image captured by the camera module is pre-processed and boundaries of parking slots are identified using Gabor filter. Then, contours are analyzed to extract the boundary points of parking slot. The status of the occupied parking slot is updated in red color region whereas the unoccupied parking slot is updated in green color region in the monitor at the front end of parking area. On the other hand, if any car is wrongly parked, then it is updated in blue color and the alert will be sent to the driver of that car for immediate action. Experiment was conducted on OpenCV with Raspberry-pi and the performance was evaluated based on the frame based metrics. The results clearly indicate that the proposed system performs well in terms of precision and accuracy.**

***Keywords*— Intelligent car parking, Log-Gabor filter, contouring analysis, Edge features.**

I. INTRODUCTION

In India, the major cities are facing a problem nowadays for vehicle parking due to lack of sufficient parking space. With families getting smaller and the total number of motor vehicles exceeding the total number of heads per family, the parking scenario is woefully falling short of the current requirements in the country. The situation is such that on any given working day approximately 40% of the roads in urban India are taken up for just parking the cars. The problem has been further exacerbated by the fact that nowadays even people from low income group are able to own cars. The number of families with cars has become much more than what the country is able to manage. As it is, the cities in India are highly congested and on top of that the parked cars claim a lot of space that could otherwise be used in a better way. Most of the public places such as shopping malls and apartments have its parking area in basement due to the unavailability of free spaces. In such case, the vehicle parking especially, car parking is difficult and suffers serious problems because of wrongly parked cars. According to the management team of the malls, apartments in city, they are suffering a lot of parking problems due to wrongly parked vehicle in the parking area. Most of the car parking today are not functioning efficiently. This means that on busy days drivers may take a long time driving around a city in order to find a free parking space. So, there is a need of designing such a system to reliably find the exact status of the parking area. This problem can be solved by using sensor modules at the parking slot. However, these sensor modules are not reliable and managing the setup is also very difficult because of misalignments of the cars in parking slot. In existing approach, it is necessary to use 4 to 6 sensors per slot and it involves more complexity. For smart parking lots monitoring, various sensor solutions such as magnetic sensor, acoustic sensors, RFIDs, ultrasounds, inductive loops, smart cameras, etc. can be used [1].

L. Ruizhi et al. proposed the smart parking system in which an ultrasonic range finder is placed in each vehicle to detect free places near the parked vehicle. The collected information is then disseminated to the other vehicles' drivers to help them find these free slots [2]. Vera-Gómez et al. presented photoelectric sensors based parking system to detect the vehicles in a controlled zone. The sensors were installed at the entry/exit of the parking and connected with the server for information processing and dissemination. Such arrangement can provide real-time information about the number of free parking lots but does not provide exact location of the free space. In addition, the problem of finding exact free parking lot grows exponentially with the number of parking lots [3]. Amato et al. proposed convolutional neural network algorithm based parking system to handle various challenging scenarios such as low light, shadows, and occlusions [4].

However, it is not integrated with a fully distributed Intelligence Transport system architecture. Neto et al. presented the computational mechanisms and image processing techniques to monitor and authorize vehicles entering the parking area. However, the presented systems only detect and recognize the vehicles and verify if they are the registered users or not [5]. Maria et al. used drone cameras to capture the videos and then analyze it for the available parking lots and congestion on the roads. However, the presented work shows less precision and also not fully integrated into a distributed ITS system [6]. In view of above, an edge detection based intelligent car parking system is proposed in this article. This new method mainly contributes the easiness of the parking authorities in shopping malls, apartments, hospitals and other places where the car parking is a vital one. The implementation of this proposed system will help to resolve the terrific issue of traffic congestion, wasted time, wasting money, and to help better public service, to reduce car emissions and pollution, to improve city visitor experience, to increase parking utilization, and to prevent unnecessary capital investments. The proposed system can be installed at the entrance and exit of the parking with a computer system, display panels and lights to manage the whole process in order to help the driver to park the car easily. The proposed system indicates the status of the parking area at the entrance so that it will simplify the vehicle parking even at peak time in traffic cities. The remaining part of this article is organized as follows. Section. 2 describes the overall methodology of the proposed system. Section 3 deliberates the results and discussions. Section 4 concludes the article with possible future directions.

II. System Overview

Fig. 1 shows the prototype model of the proposed method. The proposed system involves various steps such as pre-processing, edge feature detection, contour analysis and feature classification.

A. Pre-processing

Initially, the camera captures the image frame and the captured image is pre-processed before the foreground segmentation. Fig. 2 shows the example of captured image frames obtained from camera module. Pre-processing before the feature extraction is very useful to compensate the illumination changes and to remove the image noises. Mostly the captured image is affected by the camera jitter noise due to wrong setting apertures and environment conditions. Hence, input image frames are initially smoothed by average smoothing filter to improve the system performance with respect to illumination irregularities. One method to remove noise is by convolving the original image with a mask that represents a low-pass filter or smoothing operation. For example, the Gaussian mask comprises elements determined by a Gaussian function. This convolution brings the value of each pixel into closer harmony with the values of its neighbors. In general, a smoothing filter replaces each pixel by the average value of all pixels in the subimage called window. Smoothing filters tend to blur an image, because pixel intensity values that are significantly higher or lower than the surrounding neighborhood would smear across the area. Because of this blurring, linear filters are seldom used in practice for noise reduction. RGB color space is an additive color model in which red, green, and blue light are added together in various ways to reproduce a various secondary colors.

Fig. 1 Model of proposed intelligent car parking system

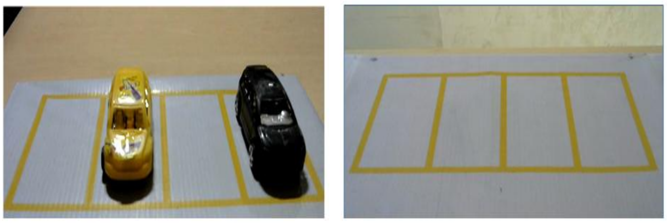

Fig. 2 Template frames captured by camera sensor module

Fig. 3 shows the overall flow diagram of proposed approach. RGB is mostly a device-dependent color space and its characteristics gets influenced to intensity variations. Different devices reproduce a given RGB value differently and it often requires pre-processing before the implementation.

Therefore, the captured RGB image frame is converted into gray scale image to avoid the influence of illumination changes. Fig. 4 shows the grayscale image obtained through color space conversion. Grayscale images are often the result of measuring the intensity of light at each pixel in a single band of the electromagnetic spectrum and it has many shades of gray in between.

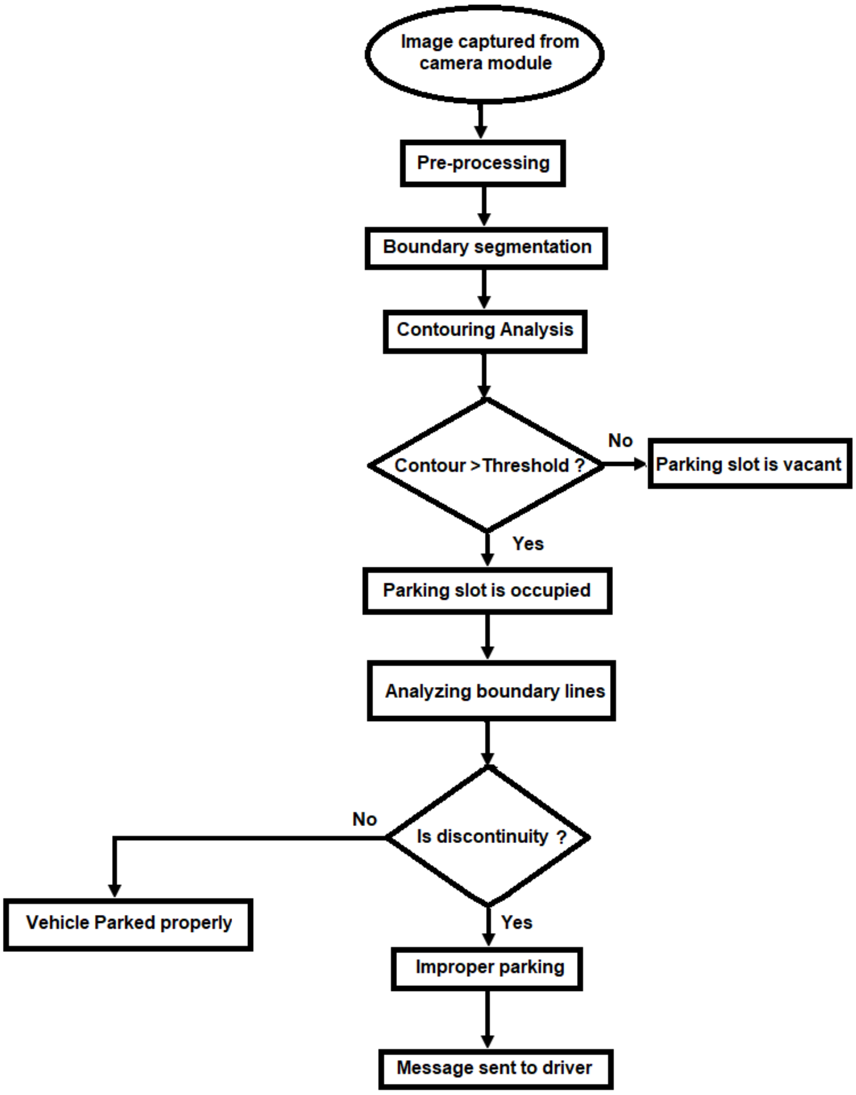

Fig. 3 Overall flow diagram of proposed approach.

Fig. 4 Grayscale image obtained from RGB image

Commonly, photometry or colorimetric is used to match the luminance of the grayscale image to the luminance of the original color image [7-8]. This also ensures that both images will have the same absolute luminance in any given area of the image. In addition, matching luminance provides matching perceptual lightness measures. To convert RGB color space into a grayscale representation of its luminance, RGB values of each pixel must be considered and form as a single output value which gives the brightness of that pixel. However, the perceived brightness is often dominated by the green component in the processed image. So, a weighted sums must be calculated in a linear RGB space for RGB color space conversion. In practice, the humans are able to see small differences when luminance is low, but at high luminance levels, humans are much less sensitive to them. In order to avoid wasting effort representing imperceptible differences at high luminance, the color scale is warped, so that it concentrates more values in the lower end of the range, and spreads them out more widely in the higher end. This is called gamma compression.

The gamma correction brings the luminance very close to that of the original image so that the grayscale representation will have better visibility. On the other hand, the gamma expansion is applied to do gamma compression before calculating grayscale luminance as per (1):

$$C_{linear} = \begin{cases} \frac{C_{srgb}}{12.92}, & ; \ C_{srgb} \le 0.04045 \\ \left(\frac{C_{srgb}+0.055}{1.055}\right)^{2.4} & ; \ C_{srgb} > 0.04045 \end{cases} \quad (1)$$

where C_{srgb}, is the gamma compressed RGB values and C_{linear} is the corresponding linear intensity value of RGB values. This gamma correction needs more computational cost when compared to weight averaging. To reduce the computational cost, luminance is calculated as a weighted sum of the three linear-intensity value as follows.

$$Y = 0.2126R' + 0.7152G' + 0.0722B' \quad (2)$$

The coefficients in (1) represent the measured intensity perception of typical trichromatic humans. The linear luminance are converted back to non-linear representation as in (3) and finally, the brightness and contrast adjustment are done in the processed grayscale image by point processing as per (4).

$$Y_{srgb} = \begin{cases} 12.92\ Y & ; \ \mathrm{Y} \le 0.0031308 \\ 1.055\ Y^{\frac{1}{2.4}} - 0.055 & ; \ \mathrm{Y} > 0.0031308 \end{cases} \quad (3)$$

$$g(x) = \alpha f(x) + \beta \quad (4)$$

where α and β are the gain and bias parameters used for contrast and brightness control respectively. The term $f(x)$ is the original image and $g(x)$ is the processed image. Fig. 5 shows the processed image in which the brightness and contrast are adjusted for further processing.

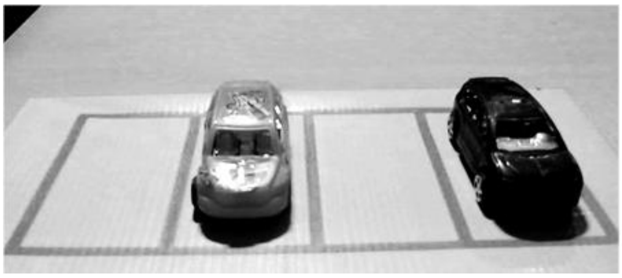

Fig. 5 Processed image after brightness and contrast adjustments

B. Edge Feature Detection

After pre-processing the captured image, the foreground features (i.e., boundaries of parking slots) are extracted from the pre-processed gray scale image using edge detection method. Edge detection is a set of mathematical calculations which identifies disjoints where the brightness changes sharply or tapering. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction. The edges extracted from a two-dimensional image of a three-dimensional scene can be classified as either viewpoint dependent or viewpoint independent. A viewpoint independent edge typically reflects inherent properties of the three-dimensional objects, such as surface markings and surface shape. A viewpoint dependent edge may change as the viewpoint changes, and typically reflects the geometry of the scene, such as objects occluding one another. For a line, there may usually be one edge on each side of the line. A number of researchers have used a Gaussian smoothed step edge as the simplest extension of the ideal step edge model for modeling the effects of edge blur in practical applications. However, it is not useful in many cases. Gabor filter is a good choice for edge detection and its impulse response is equal to the multiplication of a harmonic function with a Gaussian function. Based on the convolution theorem, the convolution of Fourier Transformation (FT) of harmonic function and FT of Gaussian function gives the FT of a Gabor filter's impulse response. Normally, Gabor filters with different frequencies and orientations in different directions are used to localize and extract edges from images.

Because, edges are relatively present in high frequency components, whereas other regions of an image are relatively smooth in nature. Gabor filter achieves maximum frequency resolution for the given set of time resolution. Nevertheless, Gabor filter response has non-zero DC term and it is symmetrically distributed around the centre frequency which has redundant information. This DC component exhibits features that depend on the mean value of the input and do not fit with statistics of the images. To overcome this problem, Log-Gabor filter is used for boundary edge detection in proposed method. Log-Gabor filter does not have DC component in its response and it requires fewer filters to extract the boundaries of natural images. In polar coordinates, the Log-Gabor filter is divided into two parts such as radial filter and angular filter. The radial filter and angular filter frequency responses are given by (5) and (6) respectively.

$$G_{rad}(r) = \exp\left(-\frac{\left| \log\left(\frac{r}{f_o} \right) \right|^2}{2\log(\sigma)^2} \right) \quad (5)$$

$$G_{ang}(\theta) = \exp\left(-\frac{(\theta-\theta_0)^2}{2(\sigma_0)^2} \right) \quad (6)$$

where (r,θ_0) is the polar co-ordinates, f_0 is the centre frequency of the filter, θ_0 is the orientation angle, σ is scale bandwidth and σ_0 is the angular bandwidth. The overall Log-Gabor filter transfer is expressed as follows.

$$G(r,\theta) = G_{rad}(r).G_{ang}(\theta) \quad (7)$$

The boundary edge features are extracted by applying a log-Gabor filter given in (7) at different scales and orientations. The energy functions are computed by convolving Log-Gabor filters with the test input images and based on these energy feature functions the test image is matched with template. Fig. 6 and Fig.7 illustrate the boundary edge features of template and test image frames, respectively.

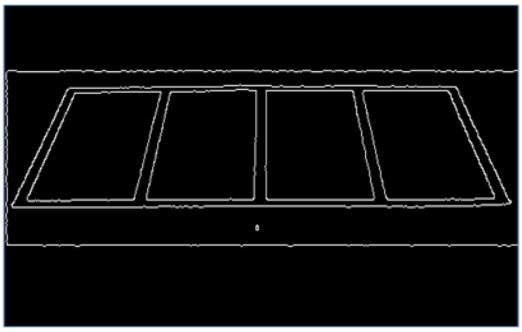

Fig. 6 Edge features of template image frame.

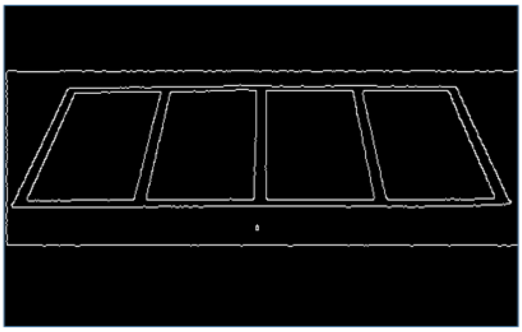

Fig. 7 Edge features of test image frame.

C. Contour Analysis

In this phase, contours of the boundary features are extracted to localize the foreground. Connected component labelling can be used for contour features extraction. Connected component is a group of pixels in which each pixel is interconnected to each other. In connected component labelling process, all the connected components are identified and labels are assigned. In addition to that, it assigns logic 1(white) for foreground and logic 0 (black) for background as shown in Fig. 8. The contour may be 4-connected or 8-connected. Nevertheless, it is not enough to merely identify the boundary pixels of a pattern in order to extract its contour.

Hence, it is required to have an ordered sequence of the boundary pixels from which the general shape of the pattern can be obtained as shown in Fig. 9. Contour tracing is one of many pre-processing techniques performed on digital images in order to extract information about their general shape. Once the contour of a given pattern is extracted, its different characteristics will be examined and used as features which will be used in pattern classification. Therefore, correct extraction of the contour will produce more accurate features which will increase the chances of correctly classifying a given pattern.

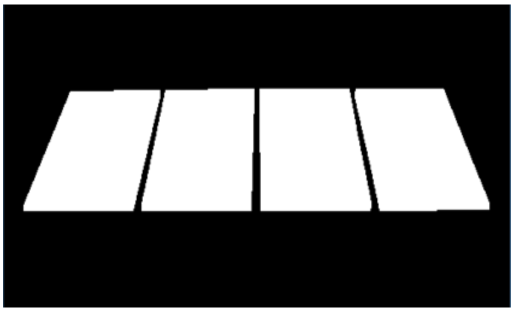

Fig. 8 Foreground and background representation.

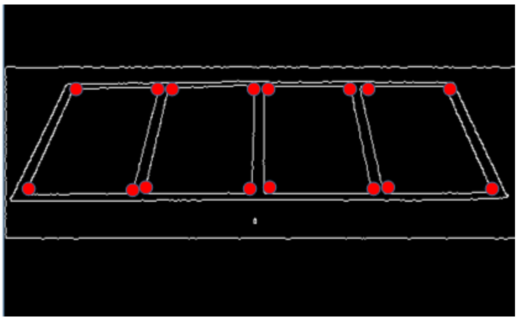

Fig. 9 Contour points obtained in template image frame.

The contour pixels are generally a small subset of the total number of pixels representing a pattern. Therefore, the amount of computation is greatly reduced when feature extraction algorithm is used on the contour instead of on the whole pattern. Since the contour shares a lot of features with the original pattern, the feature extraction process becomes much more efficient when performed on the contour rather on the original pattern. Contour representations include chain code, crack code and run code. In chain code, a contour is followed in a clockwise manner and keep the track of the directions as it goes from one contour pixel to the next. Fig. 10 shows the typical contour lines extracted from the connected component mask. These contours are used to map the slot borders and the edge features are extracted from the contour image as shown in Fig. 11.

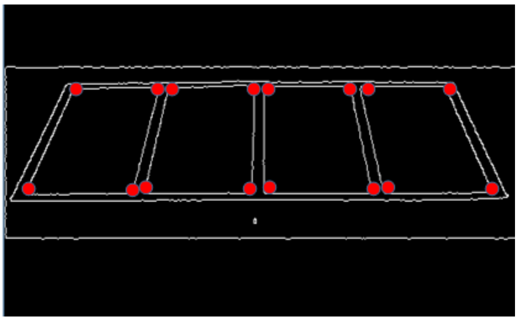

Fig. 10 Contour lines extracted from connected component.

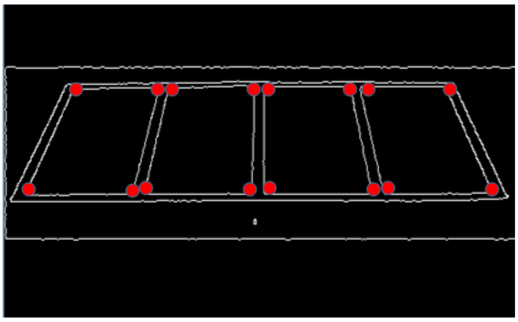

Fig. 11 Contour points obtained in template image frame.

D. Feature Classification

In feature classification, the edge features of the parking slot and the bounding box of vehicle are used to classify the status of the parking slot. First, the area of the vehicle (Car) occupied bounding box is calculated from the test frame. Then, the area of boundaries of parking slot is obtained. Let a_1 be the area of bounding box of the vehicle and a_2 be the area of boundaries of parking slot. Now, the status of the parking slot is classified by using area of overlapping. In this method of classification, the amount of overlapping is compared with threshold as per (8). A typical value of threshold for good classification is set to 0.5. From (9), if area of overlapping is greater than the threshold, then the status is classified as occupied and the bounding box is filled with red color image pixels. Otherwise, it is classified as unoccupied slot and the bounding box is filled with green color pixels. Fig. 12-13 represent the test image frame and the status of the respective parking slot. Here, all four parking slots are occupied and hence, it is updated in red color region.

$$Area\ \text{of}\ overlapping = \frac{a_1 \cap a_2}{a_1 \cup a_2} \quad (8)$$

$$Area\ \text{of}\ overlapping > Threshold \quad (9)$$

Fig. 12 Test image frame.

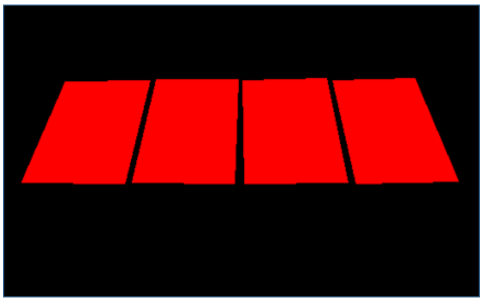

Fig. 13 Status of the parking slot.

III. RESULTS AND DISCUSSIONS

The proposed method is implemented in prototype model and experiment is conducted using Emgu CV platform. Emgu CV is a cross platform which allows OpenCV from .NET language. The biggest advantage of Emgu CV is that it can able to run on any platform. The interactive front end designs created on Emgu CV platform are portrayed in Figs.14 – 17. Here, the first screen is used to display the captured image frame. The second, fourth and fifth screens are used to display the boundaries, binary mask and contour lines of the template parking slot respectively. On the other hand, third and sixth screens are used to display the edge feature patterns and the status of the respective parking slots with slot number. From the Fig. 14, it can be seen that all four parking slots are unfilled by the vehicle and thus the status of the parking slot is displayed in green color region with slot number G0G1G2G3. That means all four slots are with vacancy. From the Fig. 15, it is inferred that all four slots are filled by the vehicles and thus the status of parking slots are displayed with red color.

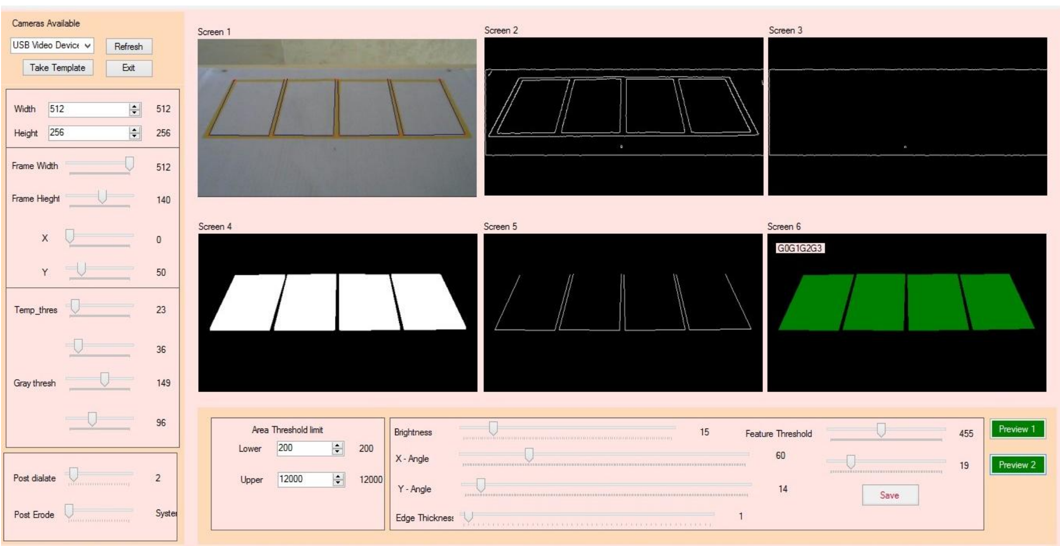

Fig. 14 Simulation results of empty parking slots with its status.

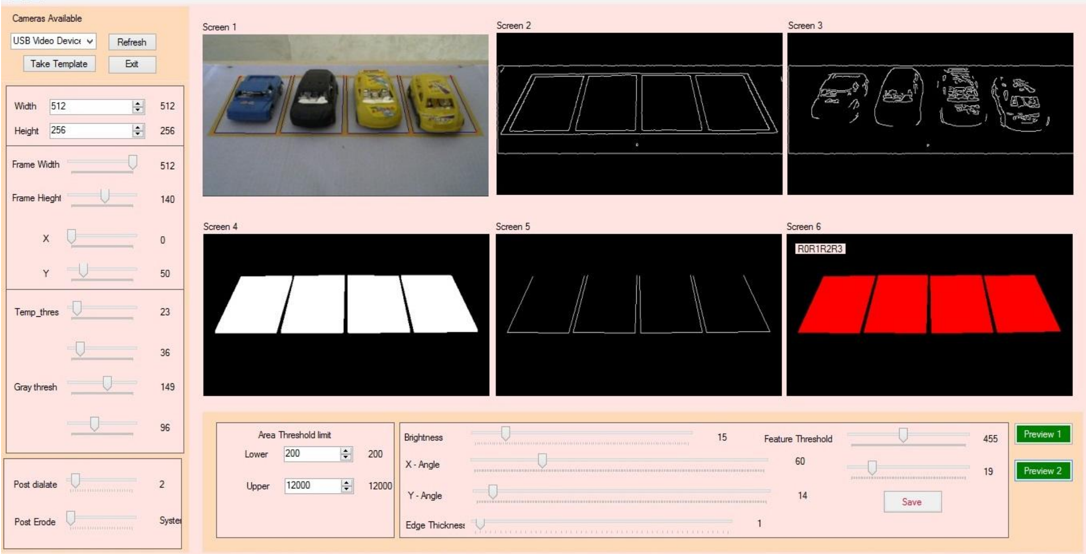

Fig. 15 Simulation results of occupied parking slots with its status.

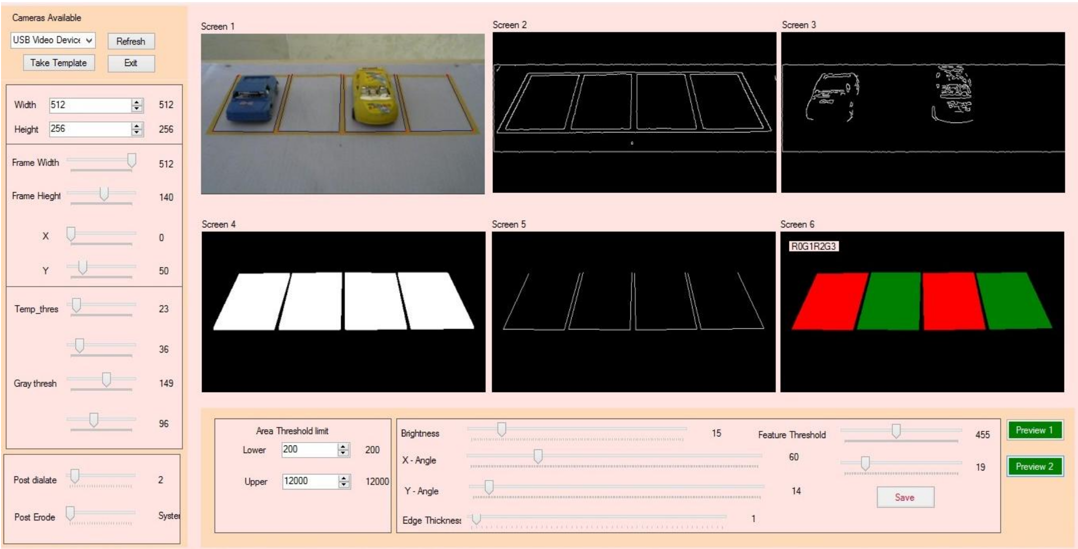

Fig. 16 Simulation results of occupied and unoccupied parking slots with its status.

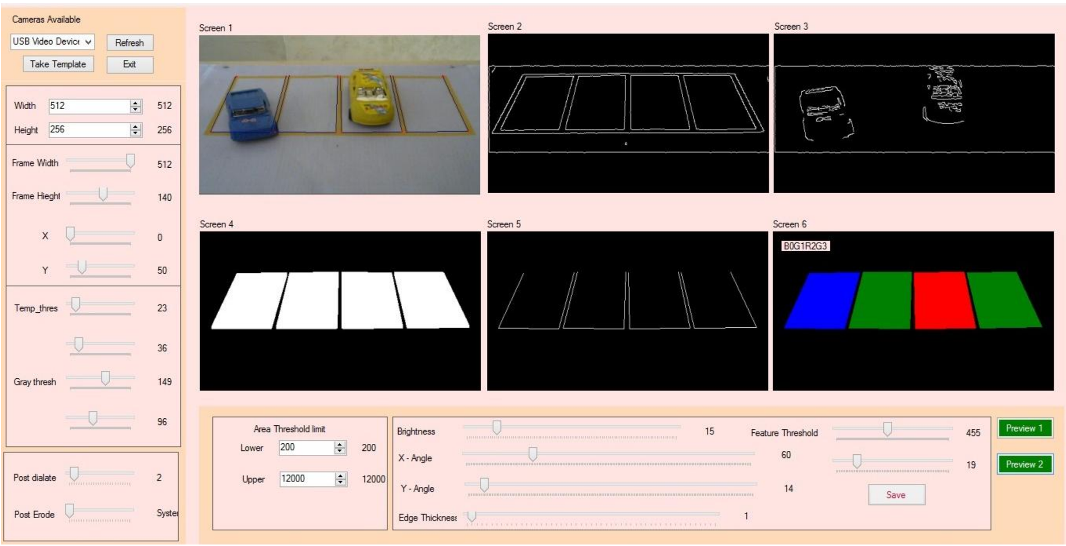

Fig. 17 Simulation results of wrongly parked slots, occupied slots and unoccupied slots with its status.

From the Fig. 16, it is understood that the first and fourth parking slots are filled and the other two slots are empty. Therefore, the slot number is displayed as R0G1R3G4 in sixth screen. In Fig.17, the first slot is wrongly parked and third slot is correctly parked whereas the other two slots are empty. Hence, the status of the wrongly parked slot is displayed in blue color and properly parked slot is displayed with red color. Additionally, the status of other two empty slots are displayed with green color. The experiment results clearly indicates that the proposed approach is best suited for intelligent car parking applications. In addition to qualitative analysis, the performance of the proposed approach is quantitatively evaluated based on the metrics such as precision and accuracy. These metrics are calculated using frame based constraints such as true positives, false positives, true negatives and false negatives. Fig.18 depicts the frame based metrics in confusion matrix.

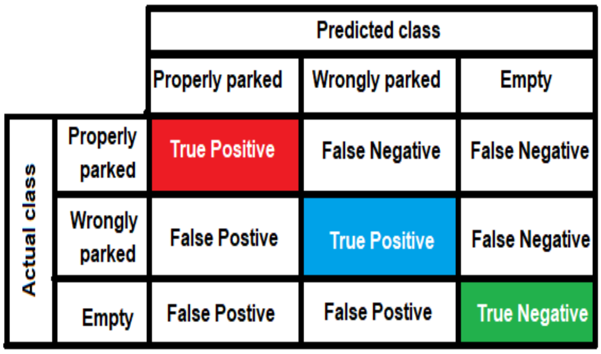

Fig. 18 Frame based metrics.

From the Fig.18, the correct predictions of properly and wrongly parked vehicle are considered as true positives $\left(T^P\right)$ whereas, the correct prediction of empty slot is considered as true negative $\left(T^N\right)$. If the vehicle is parked but it is predicted as empty, then it is referred as false negative $\left(\mathrm{F}^N\right)$. On the other hand, if the empty slot is predicted as properly/wrongly parked slot, then it is considered as false positive $\left(\mathrm{F}^P\right)$. Based on these constraints [9], the metrics such as precision and accuracy are calculated as follows.

$$Precision = \frac{T^P}{T^P + F^P} \tag{9}$$

$$Accuracy = \frac{T^P + T^N}{T^P + F^P + T^N + F^N} \tag{10}$$

Experiment is conducted at four different time slots such as 9 AM, 1 PM, 5 PM and 8 PM to assess the performance of proposed approach. In addition to that, the metrics are calculated quantitatively and listed in Table 1. From Table 1, it is inferred that the proposed approach achieves 90.77% of precision and 88.6 % of accuracy on average time slots. Fig. 19 shows the quick review of quantitative results in terms of precision and accuracy. From the results, it is concluded that the proposed approach is best suitable for intelligent car parking system.

TABLE I

TABLE I. PERFORMANCE METRICS

Time Slots	Metrics	
	Precision	**Accuracy**
9 AM	**0.9003**	**0.8988**
1 PM	**0.9331**	**0.9288**
5 PM	**0.8987**	**0.8745**
8 PM	**0.8987**	**0.8419**
Average	**0.9077**	**0.8860**

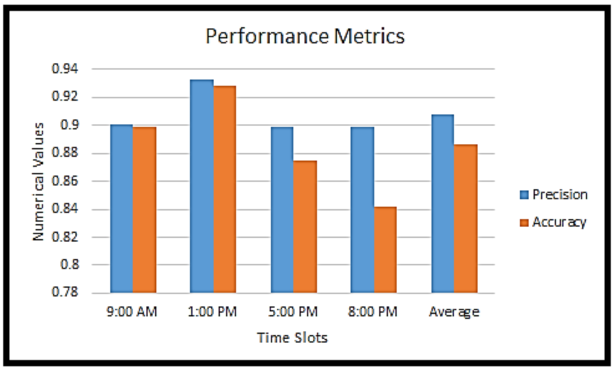

Fig. 19 Quantitative results based on performance metrics.

IV. CONCLUSIONS

An intelligent car parking system based on edge features and contour tracing is proposed. This approach involves four steps such as pre-processing, edge feature detection, contour analysis and image classification. In pre-processing, the RGB image is converted into grayscale color space and smoothened using averaging filter to remove the effect of camera jitter and image noises. Then, brightness and contrast are adjusted to improve the visual features of grayscale image. In edge feature detection, the edge features are extracted from the pre-processed image using log-Gabor filter and contour points are identified. In contour analysis, these edge features used to trace the contour lines and boundary of parking slots are identified. Finally, the overlapping of bounding box of vehicle and parking slot is calculated using Area of Overlapping method and the status of the parking slot is classified as occupied, empty and wrongly parked in three different colors. Red color is used for occupied slots, green color is used for empty slots and blue color is used for wrongly parked slots. Prototype model is designed and experiment is conducted at different time slots. Both visual analysis and quantitative results in terms of precision and accuracy indicate that the proposed method works well in all aspects. In future, it is planned to upgrade the proposed system to include the concept of Internet of Things (IoT) and sophisticated deeplearning techniques to upload the status of the parking slots in cloud server so that the driver can able to locate nearby available parking area using GPS tracking. Furthermore, the driver will get update every time with available number of empty parking slots before reaching the spot to avoid the unnecessary long driving during peak hours.

REFERENCES

[1] T. Lin, H. Rivano, and F. Le Mouel, "A survey of smart parking solutions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3229–3253, 2017.

[2] L. Ruizhi, R. Cristian, B. Peter, O. Shumao, and C. Liping, "Crowdsourcing on-street parking space detection," Available: https://arxiv.org/abs/1603.00441, 2016.

[3] J. Vera-Gómez, A. Quesada-Arencibia, C. García, R. Suárez Moreno, and F. Guerra Hernández, "An Intelligent Parking Management System for Urban Areas," *Sensors*, vol. 16, no. 6, pp. 931, 2016.

[4] G. Amato, F. Carrara, F. Falchi, C. Gennaro, and C. Vairo, "Car parking occupancy detection using smart camera networks and Deep Learning," *in proceedings of IEEE Symposium on Computers and Communication*, ISCC 2016, pp. 1212–1217, July 2016.

[5] E. C. Neto, E. S. Reboucas, J. L. De Moraes, S. L. Gomes, and P. P. R. Filho, "Development control parking access using techniques digital image processing and applied computational intelligence," *IEEE Latin America Transactions*, vol. 13, no. 1, pp. 272–276, 2015.

[6] G. Maria, E. Baccaglini, D. Brevi, M. Gavelli, and R. Scopigno, "A drone-based image processing system for car detection in a smart transport infrastructure," *in Proceedings of the 18th Mediterranean Electro technical Conference*, MELECON 2016, pp. 1–5, April 2016.

[7] Chengjun Liu and Harry Wechsler, "Gabor Feature Based Classification Using the Enhanced Fisher Linear Discriminant Model for Face Recognition" *IEEE Trans Image Processing*, vol. 11, no. 4, pp. 467-476, 2002.

[8] Z. Bin, J. Dalin, W. Fang, and W. Tingting, "A design of parking space detector based on video image", *The Ninth International Conference On Electronic Measurement & Instruments* ICEMI, 2009.

[9] K.Kalirajan, D. Venugopal, V.Seethalakshmi, K. Balaji, " Feature extraction for face recognition using edge detections and thresholding", *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 9, no. 7, pp. 59-63, 2020.