International Journal of Combinatorial Optimization Problems and Informatics, 14(1), Jan-April 2023, 49-65. ISSN: 2007-1558.

www.editada.org

ICM image separation based available parking space detection

Víctor Romero Bautista[1], Aldrin Barreto Flores[1], Salvador E. Ayala Raggi[1], Verónica E. Bautista López[2]

Benemérita Universidad Autónoma de Puebla, Av. San Claudio, Ciudad Universitaria, 72570, Puebla City, Puebla, México.
[1]Facultad de Ciencias de la Electrónica.
[2]Facultad de Ciencias de la Computación.
victor.romerobau@alumno.buap.mx,aldrin.barreto@correo.buap.mx,salvador.raggi@correo.buap.mx,
veronica.bautistalo@correo.buap.mx

Abstract. Parking lot systems based on computer vision have been used frequently in recent years to improve vehicle traffic on urban areas. These systems provide information about the availability of a parking place, furthermore these systems contribute for a better organization and to reduce the time to looking for a free space. One of the main challenges for these systems is the occlusion effects that occur frequently due to the location of the cameras in the parking lots. In this paper, we present a method for available parking place detection, where we contribute with a mechanism to reduce occlusion effects that consists of feature extraction using ICM based on input image separation. The results shows that our contribution significantly reduces the noise generated by occlusion effects. The proposed method was evaluated by our dataset and an external dataset, where the experiments results achieving up 0.98 accuracy.

Keywords: Classification · Feature Extraction · ICM · Parking Space.

Article Info
Received Sep 30, 2022
Accepted Jan 20, 2023

1 Introduction

The detection of available parking slots is commonly used in parking assistance systems, which provide information to users about the availability of a place. Also contribute to improving the vehicle flow and organization within the parking lot on the system which is implemented.

Parking space detection systems based on computer vision have been increasingly used because it does not demand high maintenance costs or the physical implementation of sensor within a parking lot. However, these systems often present some inconveniences. Some of these, correspond to the changes of illumination that occur throughout the day, shadow effects, color variation in cars, and occlusion, where these inconveniences are commonly presented in outdoor parking lots. Occlusion and changes of illumination introduces information that does not represent the object of interest, due this, mistakes may be presented during the classification.

The detection of available parking places turns a difficult work when these inconveniences are presented. Methods proposed in the state of art try to decrease these effects by applying different paradigms, some of these corresponds to use of characteristics than present low susceptibility to changes of illumination, i.e., texture features, HOG, edge detection, background subtraction, however, these features present high susceptibility when occlusion occurs, in this case, frequently the region or object affected by occlusion is not considered because this causes errors in detection.

A method that try to reduce the effects of the changes of illumination is presented in [1], where is proposed a method to obtain the characteristics of cars using Histogram of Oriented Gradient (HOG), and for classification, a set of Support Vector Machines (SVM) with linear kernel are combined using AdaBoost algorithm, this method presents low susceptibility to scale, low illumination and color variation, however, this algorithm demands a high computational cost and can present susceptibility to occlusion effects.

Other method that reduces the changes of illumination is based on corners and edges detection, corresponding to the marks that delimit the parking spaces, as is the case of [2], in which the Harris corner detector is used, this method is robust to different parking space types, but it is necessary that the markings of the parking spaces kept correctly painted.

The proposed system for parking lot occupancy detection in [3], consists of the generation of region of interest (ROI) manually, then the subtraction of the background to each ROI, and for object recognition, the HOG and Scale Invariant Feature Transform (SIFT) are employed, to improve the classification results, YUV, HSV and YCrCb color spaces features are attached; the classification process is based on the comparation of the features with a default threshold, this method presents high performance on homogeneous lighting, although, accuracy decreases in the presence of shadows.

The study presented in [4] consist of the use of Convolutional Neural Network (CNN) for the extraction of characteristics, and for classification the SVM algorithm is used, this method presents low susceptibility under different weather conditions like rainy, sunny, and cloudy, although, objects with the presence of shadow or occlusion effects are wrong detected.

In [5], is proposed an approach based on Dilated Convolutional Neural Network designed for detecting parking space occupancy in a parking lot, called CarNet,this method presents a good performance when training on a certain dataset and testing on an different dataset, but when it is validate with the same dataset the performance decrease.

In [6] is presented a method for real-time car parking occupancy detection that uses a Convolutional Neural Network (CNN) implemented in a board of a smart camera with limited resources, the results present high accuracy, even in presence of light variation and shadows, however, when an empty parking space is partially occluded, it can be detected by an occupied parking space.

An unique parking vacancy slot detection method is proposed in [7], where a Faster R-CNN is used for vehicle detection; the detection process consists on marking all possible parking slots manually, and each mark is considered as occupied if and only if it is inside the rectangle of the vehicle detected by Faster R-CNN; this method shows an efficient performance, nevertheless, the system do not behave good for occluded vehicles.

Occlusion into outdoor parking lots is difficult to predict when it occurs, because it depends on different variables such as, pedestrians walking over parking spaces, structures that covers regions of the captured image, cars wrong parked, and many others. However, sometimes occlusion may not cover whole object, and is possible to obtain the characteristics that represents it to be detected effectively by not considering the region occluded, this can be possible when the object is not covered by at least 40%.

In this paper, we propose a method for available parking space detection that reduces the noise effect introduced by the occlusion presented in images of Regions of Interest (ROI) up to 40% of occlusion. The proposed method is based on the semi-automatic generation of ROIs, time signal generated by ICM image separation method used for feature extraction, Principal Components Analysis (PCA) algorithm for feature reduction, and for classification we employed K-Nearest Neighbor (KNN), weighted K-Nearest Neighbor (wKNN), and Radial

Basis Function (RBF), these classifiers were combined by voting ensembled method. Our method was tested by using our dataset and rainy condition images from PKLot dataset, from these datasets we designed three experiments to evaluate our proposed method, first experiment consists of use our dataset, on the second experiment we use a hybrid dataset that consist of combine images of our dataset and PKLot dataset, and the third experiment consist of to use rainy conditions images from PKLot dataset. The experiments were evaluated by k-fold cross validation, where the best reached result shows an accuracy over 0.98.

2 Methodology

The proposed method has the intention to be implemented in a parking assistance system to help improve the organization in a parking lot, showing to users in the access area the number of available parking spaces, so that they can decide if accept or decline go into the parking lot, thus avoiding spending time searching an available parking space when there is not, and helping to identify an available parking space when there is, also this can improve the traffic flow into the parking lot. The parking lot images were not captured with a calibrated camera (a special camera located strategically to capture pictures or video of the parking lot), instead these were captured by hand with a smartphone camera that lets to capture pictures with size of 3264x2448 pixels, these images were resized to VGA.

This method is planned to work with outdoor parking lot images, where different inconveniences can be presented, e.g., occlusion caused by structures outside the parking lot or people walking into the parking spaces; changes of illumination that can occur along the day, such as shadow effects that can cover the regions of interest and flashes on the windshield of cars and on the parking lot floor.

The operation of the proposed method is based on a static parking lot image as an input and returns the same image with the identification of the available and occupied parking spaces. The methodology proposed to identify the available and occupied parking spaces is shown in figure 1, it is based into four main steps: (a) generation of regions of interest, (b) feature extraction based on ICM image separation method, (c) feature reduction by PCA and (d) classification by ensembled method.

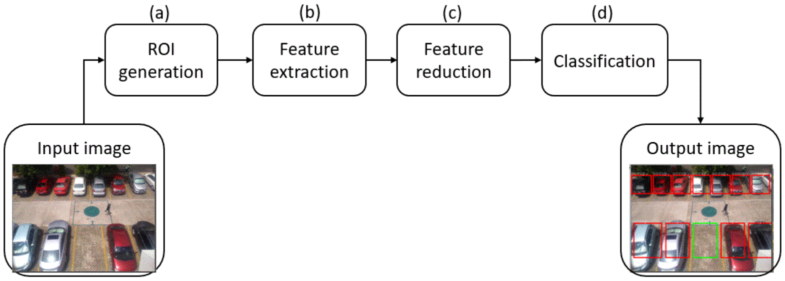
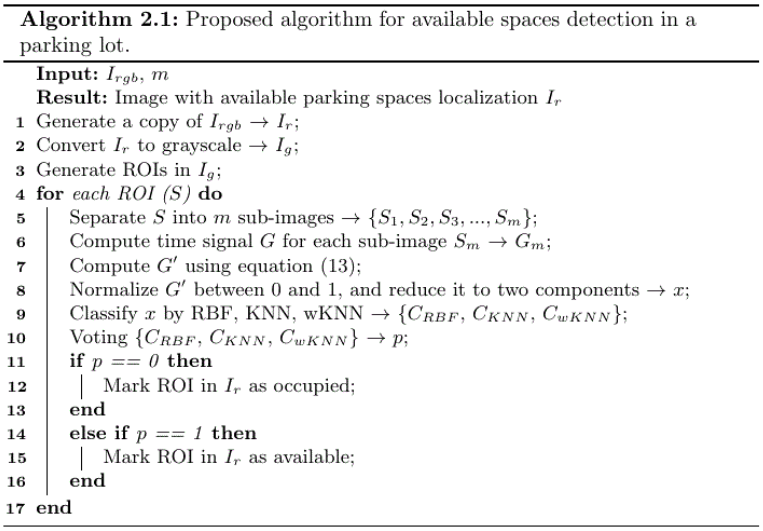


Fig. 1 Methodology proposed.

Figure 2 presents the proposed algorithm for available parking spaces detection in a parking lot image, it requires as input parameters, the parking lot image I_{rgb} with a VGA size, and the m sub-images to separate the input image, the result (I_r) is the same input image I_{rgb} with the identification of the parking spaces as the methodology shows, where bounding boxes marked with red color represent occupied spaces, while those are marked with green color represent available spaces.



```
Algorithm 2.1: Proposed algorithm for available spaces detection in a
parking lot.
   Input: $I_{rgb}$, $m$
   Result: Image with available parking spaces localization $I_r$
 1 Generate a copy of $I_{rgb} \rightarrow I_r$;
 2 Convert $I_r$ to grayscale $\rightarrow I_g$;
 3 Generate ROIs in $I_g$;
 4 for each ROI (S) do
 5     Separate $S$ into $m$ sub-images $\rightarrow \{S_1, S_2, S_3, ..., S_m\}$;
 6     Compute time signal $G$ for each sub-image $S_m \rightarrow G_m$;
 7     Compute $G'$ using equation (13);
 8     Normalize $G'$ between 0 and 1, and reduce it to two components $\rightarrow x$;
 9     Classify $x$ by RBF, KNN, wKNN $\rightarrow \{C_{RBF}, C_{KNN}, C_{wKNN}\}$;
10     Voting $\{C_{RBF}, C_{KNN}, C_{wKNN}\} \rightarrow p$;
11     if $p$ == 0 then
12         Mark ROI in $I_r$ as occupied;
13     end
14     else if $p$ == 1 then
15         Mark ROI in $I_r$ as available;
16     end
17 end
```

Fig. 2. Proposed algorithm for parking space detection.

Steps 1 and 2 in the algorithm consist of copy input image (I_{rgb}) and convert it to grayscale (I_g). Then ROIs in I_g are generated in step 3, these are appended into a list where each ROI represents a stimulus image (S). Step 5 consists of to apply the image separation to i-th ROI, here sub-images (S_m) are produced. In step 6, time signal characteristic vector (G) for each sub-image (S_m) is computed, obtaining the set G_m which contains the time signals for each S_m generated in the previous step. In step 7 is computed the general feature vector called G' that is produced according to the information in G_m. Next, in step 8 G' is normalized between 0 and 1, then feature reduction is applied to it, and are chosen the first two principal component to obtain x vector. In Step 9, machine learning algorithms RBF, KNN and wKNN are employed to classify x, classes ($C_{RBF}, C_{KNN}, C_{wKNN}$) obtained by each classifier are append into a list. Classifiers results are ensembled by voting method in step 10, where the class that has more than the 50% of votes into the elements of the list generated in previous step is chosen, it is represented by p. In the remaining steps (11 to 16), if p is equal to 0 then i-th ROI is identified as an occupied space, and if p is equal to 1 then i-th ROI is identified as an available space, both identifications are marked in the image I_r, also can be created a counter for count each available and occupied spaces identification. The main steps of the proposed method shown in figure 1 are described below.

2.1 Generation of regions of interest

The method for generating regions of interest (ROI) is based on the parking lot morphology presented on the images in our dataset. The morphology of the parking lot is shown on (a) in the figure 3, where there are three areas: upper parking area, lower parking area and transit area. According with this, the process for ROI generation consists of dividing into three rows the image to generate upper, transit and lower areas. The height of the upper and lower areas is set according to the size of the parking area (in pixels), and the transit area is not considered, therefore, only upper and lower areas remain, this process is shown on (b) in figure 3. Then, upper and lower parking areas are divided into the quantity of parking slots that can be contained within each area, this process is shown on (c) in figure 3.

To exemplify this process, consider the parking lot image presented on (a) in figure 4. According with the first step, this image is divided into three rows generating upper, transit and lower areas, then, upper and lower areas are divided into columns according with the number of parking slots considered that could be contained in each area, in such a way that each column can generate a box that contains a parking space. The image presented on (b) in figure 4 presents the result of this process, where the upper area is divided into seven and lower into four, because with this configuration is possible to contain a parking space in each box. The number of columns is the parameter that lets to generate the ROIs, and this is set according to the frame obtained.

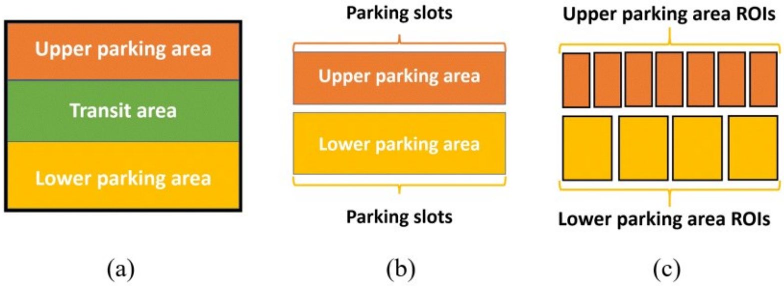


Fig. 3 Localization of regions of interests: (a) morphology of the parking lot; (b) upper and lower parking areas; (c) ROI generated

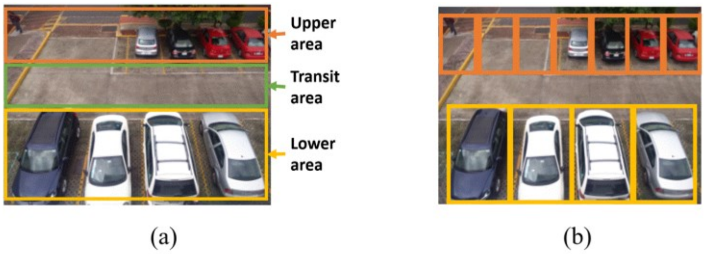


Fig. 4 ROIs localization; (a) rows generation; (b) bounding boxing generation.

According to the parking lot structure, a region that must not be considered can be present as the image on (a) in figure 5 shows, where the red box shows the parking lot entrance, therefore, the ROIs generated into this region must not be considered. For this, a parameter is stablished manually to indicate that some ROIs must be ignored into each area generated (upper and lower), the value of this parameter was stablished according with previous tests. For the case of the image presented on (a) in figure 5, is stablished that the first three ROIs (from left to right) of the upper area must be ignored, the result of this process is shown on (b) in figure 5, that presents the ROIs generated. ROIs generated are append into a list to be employed in the next step.

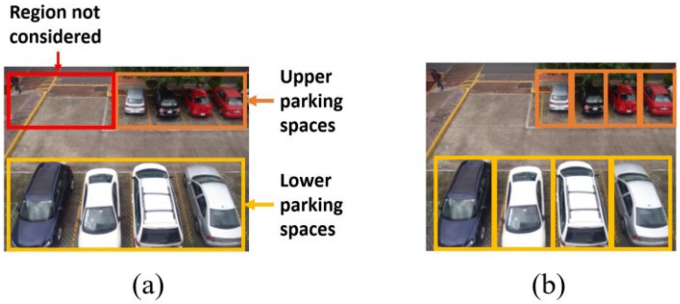


Fig. 5 Results to apply the proposed method for ROI localization.

2.2 Feature extraction

In this step, ROIs generated previously are received and feature extraction process is applied for each one to obtain its characteristics vector. The proposed method for feature extraction is based on the Intersection Cortical Model (ICM) proposed by Ekblad in [8], it is a Pulse-Coupled Neural Network (PCNN) special case, where β parameter value of the Linking field is set to 0, as a result, the ICM does not receive Linking field information and reduces the quantity of the parameters to 3 [9], [10]. PCNN is considered as 3rd generation neural networks due not require a training phase instead these need a correctly adjust to their parameters to obtain a desired output. An ICM neuron representation is shown in figure 6, where the parameters to adjust (f, g, h) are highlighted in red.

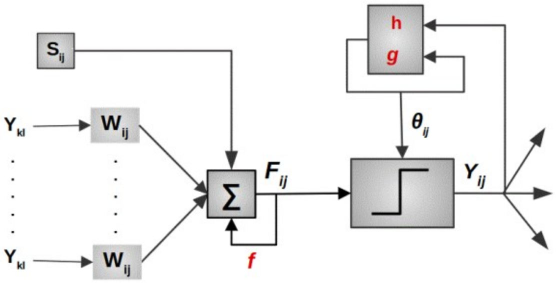


Fig. 6 ICM neuron representation.

If the stimulus (S) is an image, then the neural network will have as many neurons as pixels the image contains. ICM model is given by the following equations:

$$F_{ij}[n] = fF_{ij}[n-1] + \sum W_{ijkl}\, Y_{kl}[n-1] + S_{ij} \tag{1}$$

$$\theta_{ij}[n] = g\theta_{ij}[n-1] + hY_{ij}[n] \tag{2}$$

$$Y_{ij}[n] = \begin{cases} 1, if\, F_{ij}[n] > \theta_{ij} \\ 0, otherwise \end{cases} \tag{3}$$

$$W = \begin{pmatrix} 0.25 & 0.5 & 0.25 \\ 0.5 & 0.5 & 0.5 \\ 0.25 & 0.5 & 0.25 \end{pmatrix} \tag{4}$$

Here, F_{ij} is the receptive field called feeding, it receives S_{ij} pixel from the stimulus image and it is added to the neighbor neurons information and the output of feeding at previous time weighted by f parameter; θ_{ij} is the dynamic threshold, where g is the decrement parameter and h is the increment value for neurons that have been activated in a time; Y_{ij} is the output of the neuron; and W is the synaptic weights matrix to obtain the information of the neighbor neurons. PCNN is widely used for image processing, some of its applications are edge detection [11], bright images enhanced [12], noise reduction [13], segmentation [14], feature extraction [15],[16], and others. The time signal (G) is a feature that can be obtained by PCNN model, it was proposed by Johnson in [17], and it is expressed on the following equation:

$$G[n] = \sum Y_{ij}\,[n] \tag{5}$$

The time signal G is a vector that contains the number of activated neurons in a time n, this feature presents low susceptibility to rotation and scaling. We proposed the following modification to ICM model in order to have a better control in obtaining information from neighbor neurons, this modification is showed in the following equations:

$$Y \otimes W = \begin{cases} conv, \sum W_{ijkl}\, Y_{kl}[n-1] > 0 \\ 0, otherwise \end{cases} \tag{6}$$

$$F_{ij}[n] = fF[n-1] + Y \otimes W + S_{ij} \tag{7}$$

$$\theta_{ij}[n] = g\theta_{ij}[n-1] + hY_{ij}[n] \tag{8}$$

$$Y_{ij}[n] = \begin{cases} 1, F_{ij}[n] > \theta_{ij}[n] \\ 0, otherwise \end{cases} \tag{9}$$

$$W = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{pmatrix} \tag{10}$$

Here, we introduce the $conv$ value that will be added to the feeding field when at least one neighbor neuron is activated using the kernel W shown in equation (10). This lets to apply a specific increment to the internal potential of the neuron (feeding field), when is detected that at least one neighbor neuron has a similar pixel intensity. Therefore, correlated neurons with a difference less than $1 - g$ on its pixel intensity can be activated at the same time by applying $conv$ increase. This is only applicable when f parameter is nearly equal to 0.

For the extraction of G, we proposed to separate the input image (S) into m sub-images with the same size, to reduce the occlusion effects presented in the ROI generated, this process is represented by the following equation:

$$S = \{S_1, S_2, S_3, \ldots, S_m\} \tag{11}$$

Then, for each sub-image generated, compute time signal (G) setting the value of the parameters to: f = 0, g = 0.8, h = 20, $conv$ = standard deviation of the m sub-image (σ_m) and a quantity of 50 pulses. This process generates G, which contains $G_m$time signals as equation (12) shows.

$$G = \{G_1, G_2, G_3, \ldots, G_\mathrm{m}\} \tag{12}$$

Subsequently, compute G' according to equation (13), where, if the components of the $G_m$vectors are not equal in a time n, the value assigned will be the mean of G, in otherwise assign the mode of G.

$$G'[n] = \begin{cases} \dfrac{\sum_{i=1}^{m} G_i[n]}{m}, & if \quad G_1[n] \neq G_2[n] \neq G_3[n] \neq, \ldots \neq G_m[n] \\ \quad mode\{G[n]\}, & \quad otherwise \end{cases} \tag{13}$$

The resultant time signal vector (G') is the feature that we use for recognition process. This method is showed in figure 7.

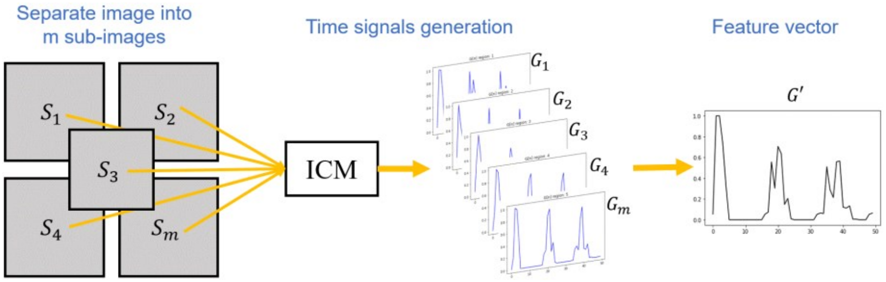


Fig. 7. Feature extraction by ICM splitting image method.

In this work we implemented three configurations to separate the input image (S), which are: four, five and nine sub-images, these configurations are presented in figure 8.

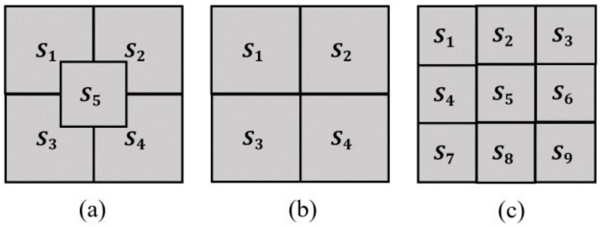


Fig. 8. Configuration to split input image: (a) m = 5; (b) m = 4; (c) m = 9.

In order to provide the classification result, we also use the average value of the standard deviation (σ_m) obtained according with the sub-images as the following equation shows.

$$\sigma_m = \frac{\sum_{i=1}^{m} \sigma(S_i)}{m} \qquad (14)$$

Where $S_i$is the sub-image generated by image separation method; σ_m is attached to the vector G'.

2.3 Feature reduction

In this step a feature reduction is applied to the vector G' generated in the previous step. To reduce the number of components in the features vector we use the Principal Component Analysis (PCA) algorithm [18]. PCA reduce the dimensionality of a dataset that contains a large number of related variables while retaining as much as possible of the variation presented in this, by transforming to a new set of variables called principal components (PCs) which are ordered so that the first components retain most of the variation present in all of the original variables [24]. In this step firstly, vector G' is normalized between zero and one, then, the dimensionality reduction is applied to obtain a vector x, with two components (C_1, C_2) that correspond to the two principal components that contain the 60 % of the total variation, this vector is showed in the following equation:

$$x = [C_1, C_2] \qquad (15)$$

According to the analysis of the information obtaining by PCA and tests made, we decided to employ only the firths two principal components in order to reduce the memory consumption.

2.4 Classification

This step consists of recognize the vector x to identify if it corresponds to an available parking space or to an occupied space, this process is called classification. For this stage, we use machine learning algorithms: K-Nearest Neighbors (KNN) [19], Radial Basis Function (RBF) [20] and weighted K-Nearest Neighbors (wKNN) [21]. KNN and wKNN are not parametric classifiers, these are based on memory, KNN and wKNN store all data train, so that when a query is given, they find the k closest classes in the data stored and is selected the class with more votes, in the wKNN particular case, distance between the query and k closest classes of the data training is taken into account to weight each k closest classes, giving more importance to classes that are closer to the query. RBF is composed by three layers, input, hidden, and output; input layer receives the feature vector, hidden layer contains neurons with radial basis function as presented in the following equation.

$$g_j = e^{-\frac{\sum_{i=1}^{n}(x_i-w_i)^2}{2\sigma_j^2}} \tag{16}$$

Where, $g_j(u)$ is the output of the jth neuron in the hidden layer; x_i is the nth component in the input vector; w_i is the centroid computed during the training phase by applying K-Means algorithm, and σ_j^2 is the variance of the jth neuron in the hidden layer, also computed during the training phase. In the output layer was employed a logistic activation function presented in the following equation.

$$C_{RBF} = \frac{1}{1 + e^{-s}} \tag{17}$$

Where,C_{RBF} is the output of the RBF; s is the internal potential of the neuron. These classifiers were employed according to tests, where the data distribution obtained presented a better performance with KNN, RBF, and wKNN than with multi-layer perceptron (MLP). MLP demands a considerable time for training phase (more than 5000 epochs are needed to obtain similar performance than KNN), on the other hand, using KNN, wKNN and RBF separately a similar performance is obtained, and less epochs for training phase for RBF classifier are needed (100 epochs), due this, to get a better accuracy employing these classifiers we used an ensembled method.

We set k= 3 for KNN, k = 5 for wKNN and 5 neurons in the hidden layer for RBF with 100 epochs for the training phase. The value of k for KNN and wKNN was established according to previous tests, where was obtained a better performance with these k values. The number of neurons and epochs for RBF was established according to previous test, where was observed that employing more than 100 epochs an overfitting affect occurs, whereas that, employing less or more neurons in the hidden layer the accuracy decrease. These algorithms were combined by voting ensemble method as schema in figure 9 shows, which can provide better result [22]. The value of prediction (p) corresponds to the class label that has majority of votes. This method is expressed by equation (18).

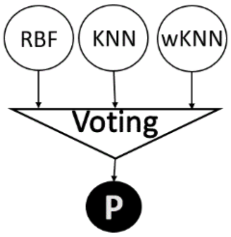


Fig. 9. Schema for voting ensembled method.

$$p = mode\{C_{RBF}, C_{KNN}, C_{wKNN}\} \tag{18}$$

Here, C is the class predicted by each classifier and p is the class that has more than the 50% of votes.

3 Results

The proposed method was implemented in a Raspberry Pi 4 which uses a Quad Core Cortex-A72 processor with 1.5 GHz speed, 4 GB of RAM, and 16 GB on disk. The algorithm presented in figure 2 was implemented using Python 3 programming language, with the libraries: Numpy, Pillow, OpenCV, Pandas; and for graphic user interface (GUI) was employed Tkinter library. To evaluate the algorithm three experiments were designed, and for validate them k-fold cross validation with 10 folds was employed, these validation tests were running in Google Collaboratory virtual machine with 12 GB of RAM and 100 GB on disk, CPU speed information is not shared by Google. The training phase for implementation in Raspberry Pi was ran in an Intel Core i5 8400T

CPU with 1.7GHz speed with 16 GB of RAM. The proposed method only was proved with static images pre-captured; the system has not been implemented into a parking lot yet, this is planned for a future work. Results presented in subsection 3.4, where obtained by k-fold cross validation using k = 10, in consequence ROIs for training correspond to 90% of total samples, whereas 10% corresponds to validation. For operation phase, was used 30% of samples for training, and 70% for validation.

In this section, results of the four main steps of proposed method are presented. Firstly, datasets employed are shown, consequently ROIs generation results are presented, next the proposed method for feature extraction based on image separation is exposed, and the accuracy obtained by the designed experiments for testing the proposed algorithm using our dataset, hybrid dataset and PKLot dataset are presented below.

3.1 Datasets

For evaluate the proposed method we employed two datasets, first correspond to our dataset generated with images of the parking lot of the faculty of electronics of science (FCE) at Benemérita Universidad Autónoma de Puebla (BUAP), the second is the PKLot dataset, that contains images of parking lot at Federal University of Parana and Pontifical Catholic University of Parana. These datasets are described below.

Our dataset

For testing our method, we accessed to our dataset generated in the parking lot of the faculty of electronics science (FCE) at Benemérita Universidad Autónoma de Puebla (BUAP). This dataset contains 407 images with size of 3264x2448 pixels captured by a cellphone camera resized to VGA, which has four different frames captured by a timelapse during the day, these frames are showed on the figure 10, where the image (a) is the parking lot center part, image (b) correspond to right part, image (c) is the left part and image (d) correspond to parking lot lateral part. These images included some weather conditions like sunny and overcast; images with sunny conditions present variation of lighting due to shadow effects that can cover the regions of interest, and flashes on the windshield of the cars, on the other hand, images with cloudy conditions can present a homogeneous lighting.

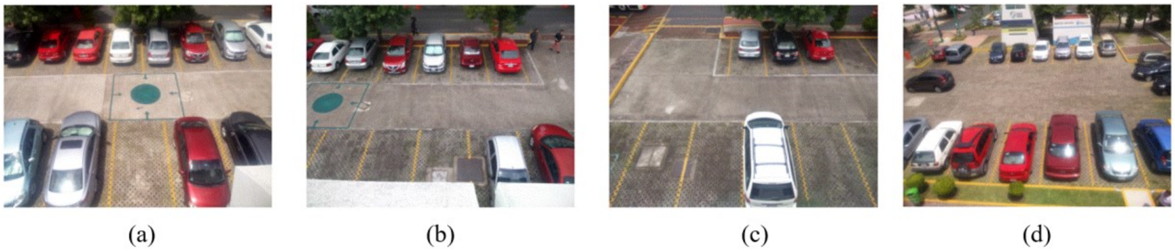


Fig. 10. Frames from out dataset.

PKLot dataset

Furthermore, we use dataset PKLot to test the performance of our method for another kind of parking lot; this dataset contains 12,417 images with size of 1280x720 pixels captured at the parking lots of the Federal University of Parana (UFPR) and the Pontifical Catholic University of Parana (PUCPR), both located in Curitiba, Brazil. This dataset contains 695,899 images of parking spaces segmented from the images of the parking lots UFPR and PUCPR, which were manually checked and labeled [23]. Also, the images are categorized according to the weather condition. For our tests, we use rainy condition segmented images from the PUCPR parking lot. The figure 11 shows images of PKLot dataset corresponding to PUCPR parking lot, where 100 segmented images can be obtained by each image. Figure 12 shows some segmented images of PUCPR parking lot, where (a), (b) correspond to empty parking spaces, whereas (c) and (d) correspond to occupied parking space.

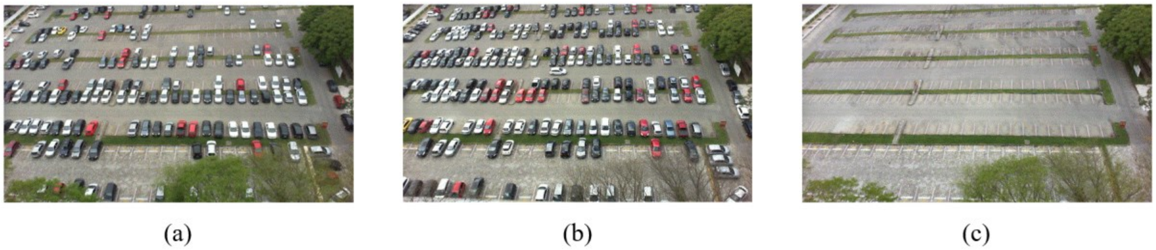


Fig. 11. Parking lot images from PKLot dataset [23].

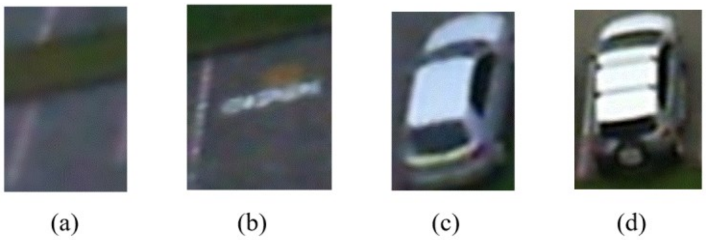


Fig. 12. Segmented parking spaces provides by dataset PKLot [23].

3.2 ROI generation

The method to generate ROIs semi-automatically exposed above was applied only to our dataset, for PKLot dataset we use the segmented images that this dataset provides. Some segmented parking places by the method to generate ROI semi-automatically for our dataset are shown in figure 13, where (a) and (b) correspond to empty parking space, on the other hand, (c) and (d) correspond to occupied parking space. The average of ROIs that can be generate in parking lot images of our dataset correspond to 13 ROIs for center part images, 11 ROIs for right part images, 8 ROIs for left part images, and 15 for lateral part. In total we obtained 3504 unbalanced samples, table 1 shows the quantity parking spaces segmented from our dataset.

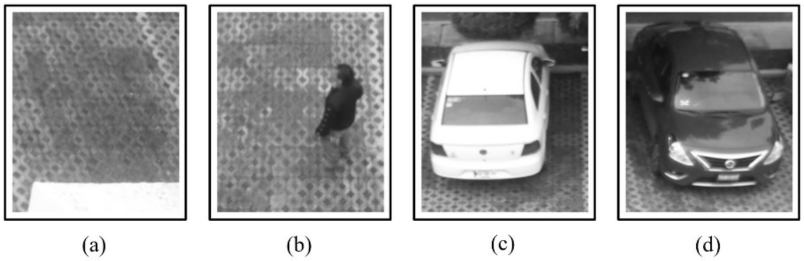


Fig. 13. Segmented parking places by semi-automatic ROI generating method.

Table 1. Parking spaces segmented from our dataset.

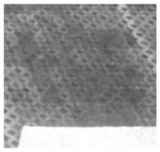
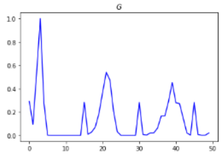
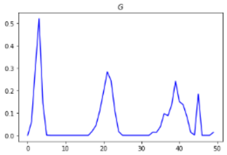
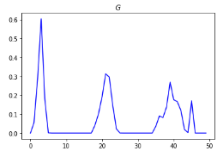
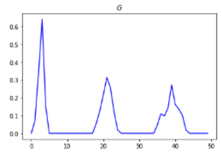
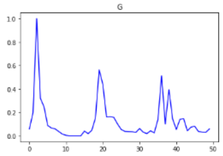
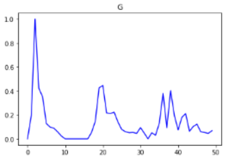
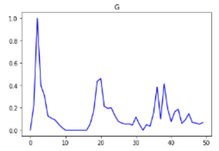
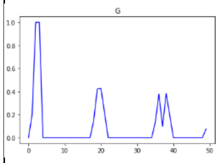
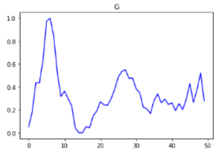
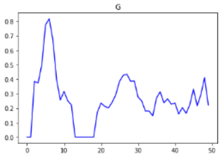
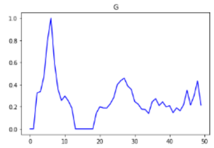
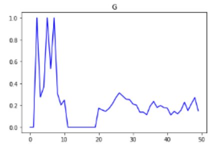
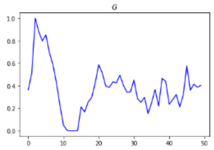
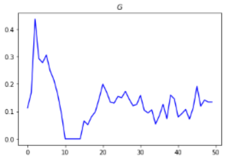
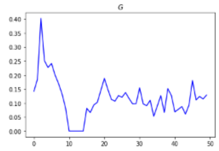
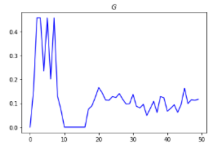
Parking space	Quantity
Occupied	2200
Empty	1298
Total	3498

3.3 Feature extraction

As exposed above, the proposed method for feature extraction using ICM is based on separating input image into m sub-images to obtain G_m time signals, then computing G'. We proposed this method to eliminate the noise effects in G caused by occlusion. In our experiments, we use three configurations to split the input image in order to observe which configuration obtains the best classification result.

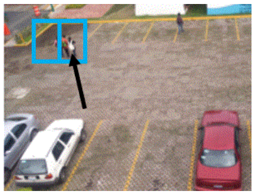
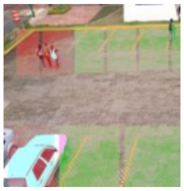
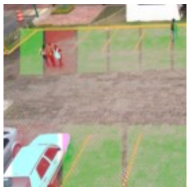
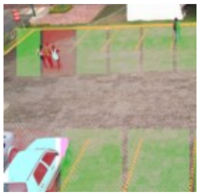
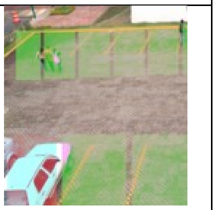
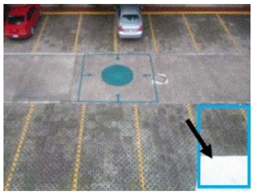
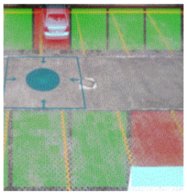
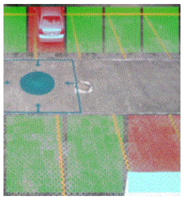
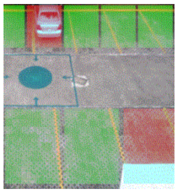
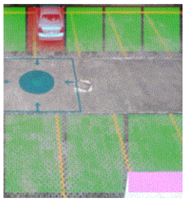
The result observed in the proposed method is shown in the table 2, where we can see the input image, and the time signal G' generated by m sub-images; when $m = 0$, there is not image separation, therefore, the time signal G' is obtained by global image. In the result presented in (a), the G' obtained by the global image contains peaks that correspond to the occlusion effect generated by the lower white region, whereas, using separation image method these peaks are reduced by $m = 4$ and $m = 5$, and the peaks are completely deleted by $m = 9$. In the result presented in (b), the input image contains a larger occlusion region, due this, the G' obtained by $m = 4$ and $m = 5$ are similar to G' obtained by $m = 0$, whereas, using $m = 9$ the occlusion effect is reduced. In the result presented in (c), the occlusion effect is induced by pedestrians in an occupied space, but due to the dispersion of pixel intensities is high, we obtain a similar G' for $m = 0, 4, 5, 9$ generated largely by the mean of G, a similar case is showed in the result presented in (d).

Table 2. Results of feature extraction by splitting input image method.

Test	Input image	$m = 0$	$m = 4$	$m = 5$	$m = 9$
(a)					
(b)					
(c)					
(d)					

During the operation phase can be observed the results presented on table 3, where is pointed the object that cause the occlusion effect in the input image, and ROIs affected by occlusion are marked with a blue box. The first row of the table presents the occlusion effect caused by people transit that covers a certain percent of two ROIs, whereas the second row of the table presents the occlusion caused by a part of roof near the parking lot, that covers approximately a 30% of a ROI. Detection results are presented in table 3 by each m configuration, where green marked ROIs indicates an available space detected whereas red marked ROIs indicates an occupied space detected. According to the input image presented on the first row in table 3, using $m = 0$ the ROIs that are covered by persons cannot be detected as an available parking space because as test (b) in table 2 presents, this occlusion introduces noise to the time signal (G), as consequence these ROIs are classified as occupied parking spaces, on the other hand, by using $m = 4, 5, 9$ the ROI where the occlusion effect is less (first ROI from left to right), the noise is reduced and then it is detected as an available parking space, but the ROI where the occlusion is present in at least in 30%, and is need to use $m = 9$ to detect this ROI as an available space. According to the input image in the second row on table 3, here the occlusion is caused by the roof near the parking lot, as test (a) in table 2 presents, this kind of occlusion add noise to G of the ROI, as a consequence this cannot be detected as an available parking space by using $m = 0, 4, 5$, and when $m = 9$ the noise effect is reduced and this ROI can be detected as an available parking space.

Table 3. Results of detection phase for each m configuration.

Input image	$m = 0$	$m = 4$	$m = 5$	$m = 9$
				
				

3.5 Feature Reduction

For feature reduction we decide to employ only the first two components that correspond to the 60% approximately of the total variation, to reduce the memory consumption. Table 3 presents de percent of variation of the first two components employed by each configuration of m.

Table 4. Variation of the first two principal components employed.

$m = 0$	$m = 4$	$m = 5$	$m = 9$
[0.39,0.15]	[0.46, 0.17]	[0.43,0.16]	[0.42,0.16]

3.4 Experimental results

To test our method, we designed three experiments. The first experiment consists of to use our dataset, the second test consists of to use a hybrid dataset that contains segmented images of our dataset and of PKLot dataset, and the third experiment consists of to use segmented images of PKLot dataset. For these experiments we only use segmented images with rainy weather conditions from PKLot dataset. To evaluate the experiments, we use k-fold cross validation with 10 folds to obtain the accuracy on each experiment. Accuracy is computed by equation (19).

$$accuracy = \frac{tp + fn}{tp + tn + fp + fn} \tag{19}$$

Where, tp (true positive) is the number of segmented images that represents an occupied space classified as occupied, tn (true negative) is the number of segmented images that represents an available space classified as available, fp (false positive) is the number of segmented images that represents an occupied space classified as available, and fn (false negative) is the number of segmented images that represents an available space classified as occupied. Experiments and results are described below:

1.- In the first experiment, we use our dataset, where we obtained segmented parking places by the semiautomatic ROI generator method exposed above. To balance the number of samples between occupied and empty spaces, we added 902 samples of empty spaces from the PKLot dataset. Therefore, the number of samples for this test corresponds to 4400, with 2200 samples for empty and occupied spaces. This experiment is composed by the following configurations of the feature vector:

a) Use the G' computed according with the m sub-images employed. Table 4 presents the results obtained by using 4, 5 and 9 sub-images, where the values of tp, tn, fp, fn are presented in average obtained during k-fold cross validation.

Table 5. Results of cross validation using G' computed by m sub-images.

m sub-images	**Average tp**	**Average tn**	**Average fp**	**Average fn**	**Average accuracy**
4	211.6	8.4	16.6	203.4	0.943
5	211.2	8.8	16	204	0.943
9	202.6	17.4	18.2	201.8	0.919

b) Attach to G' the average of standard deviation (σ_m), obtained according to the information in each m sub-image. Table 5 shows the results obtained, where the values of tp, tn, fp, fn shows the average obtained during cross validation.

Table 6. Results of cross validation using G' and σ_m computed by m sub-images.

m sub-images	**Average tp**	**Average tn**	**Average fp**	**Average fn**	**Average accuracy**
4	211.9	8.1	11.4	208.6	0.957
5	214.1	5.9	7.3	212.7	0.97
9	211.7	8.3	11.3	208.7	0.955

2.- In the second experiment, we use a hybrid dataset, this contained 4000 samples and was generated by combining our dataset with PKLot dataset, using 2000 samples for each dataset, where 1000 samples correspond to empty spaces and 1000 to occupied spaces. Here, we introduced another feature vector, composed by G' and standard deviation of the global image (σ_{global}). The result of this test is shown in the graph in figure 14, where the blue bar indicates accuracy achieved by the use of G' as a vector of characteristics, by m = 0 4, 5, 9; the orange bar indicates the accuracy achieved by the use of $G'$and the standard deviation of the global image (σ_{global}); the green bar indicates the accuracy achieved by the use of G' and the average of standard deviation (σ_m) obtained by each sub-image. When $m = 0$, there is not splitting image, therefore, the time signal G' is obtained by global image.

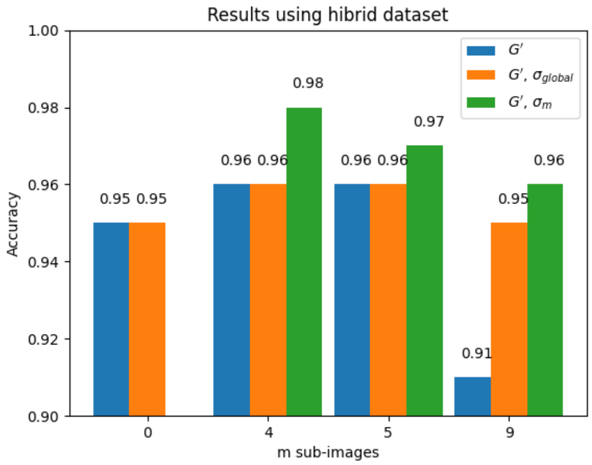


Fig. 14. Results of the second experiment.

3.- In the third experiment, we used 24,650 rainy condition segmented parking places from the PKLot dataset. The results of this test are shown in the figure 15, where blue bar indicates the accuracy reached by using G', and orange bar indicates the accuracy reached by σ_m attached to G'. When $m = 0$, there is not splitting image, therefore, the time signal G' is obtained by global image.

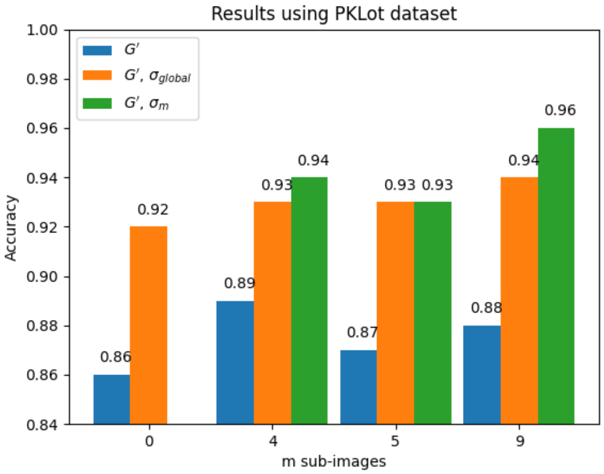


Fig. 15. Results of the third experiment.

4 Comparison

Table 4 presents a comparison of the proposed method with some methods in the state of art, empty spaces in the table are presented when authors do not report the respective information. Methods presented in [4], [5] and [7] employ deep learning methods (Convolutional Neural Networks), while [3] uses HOG, SIFT and color space features and the classification is based on threshold comparison.

The proposed method shows the largest number of samples used in the experiments that corresponds to PKLot dataset samples exposed in the third experiment presented above, and the accuracy value belongs to the highest obtained in third experiment; run time presented in the proposed method is based in the average obtained by using m configurations. According with tests, using $m = 0$ takes 3.7 secs, $m = 4$ takes 3.9 secs, $m = 5$ takes 4.3 secs and $m = 9$ takes 4.5 secs to process an entire image with twelve ROIs, while to process a ROI for $m = 0$ takes 0.25 secs, $m = 4$ takes 0.29 secs, $m = 5$ takes 0.31 secs, $m = 9$ takes 0.32 secs.

Table 7. Proposed method comparison with methods in state of art.

Author	Acharya [4]	Nurullayev [5]	Khan [7]	Tatulea [3]	Proposed
Samples employed	504,139	840,864	695,899	106,192	24,650
Accuracy	0.982	0.97	0.90	0.94	0.96
Run time (secs.)	2	-	-	-	4.1
CPU speed (GHz)	2.5	-	-	-	1.5
GPU	-	-	Nvidia 1080 Ti	-	-

5 Conclusions

The proposed method can be able to reduces the noise caused by lighting changes like shadows and flashes that commonly occur during the day, and the inconvenience caused by occlusion effects are reduced applying ICM image separation method exposed above.

According with the reached results, the splitting input image method can improve the classification results, as it is able to reduce the occlusion effects in the image. When the occlusion occupied 25% of the image, the use of 4 or 5 sub-images may be sufficient to reduce the noise effect. On the other hand, when the occlusion occupied 40% of the image, it is advisable to use 9 sub-images. The first experiment reached an accuracy of 0.943, which was obtained by using 4 and 5 sub-images, whereas, attaching the average of standard deviation (σ_m), computed according to the information in each m sub-image, a better accuracy was reached, of 0.97 with 5 sub-images, while with 4 and 9 sub-images, the accuracy reached was similar, of 0.95. In the second experiment was demonstrated that, using splitting input image method, better accuracy can be obtained than using global image, as in the case of using 4 and 5 sub-images, where an accuracy result of 0.98 was obtained using 4 sub-images. In the third experiment the better accuracy of 0.96 was obtained by using 9 sub-images. In comparison with other methods in literature, the proposed method presents a better accuracy than [3] and [7], and slightly less than [4] and [5], which suggest that the proposed method presents an acceptable performance.

The number of sub-images used implies a certain time for processing, it means that a greater number of sub-images (m) takes more processing time. For future work should optimize the feature extraction by ICM separation image by introducing a mechanism that lets to automatically select the quantity of m sub-images to generate, also it is advisable to incorporate more parking scenarios and a comparison study with other methods in the literature.

References

1. Wahyono, Hoang, V.D., Jo, K, H.: Multiscale car detection using oriented gradient feature and boosting machine. In: Nguyen, N.T., Trawinski, B., Fujita, H., Hong, T. P. (eds.) Intelligent Information and Database Systems, pp. 731-740. Springer Berlin Heidelberg (2016).
2. Suhr, J.K., Jung, H.G.: Fully automatic recognition of various parking slot markings in around view monitor (avm) image sequences. In: 15th International IEEE Conference on Intelligent Transportation Systems, pp. 1294–1299. IEEE (2012).
3. Tatulea, P., Calin, F., Brad, R., Brancovean, L., Greavu, M.: An Image Feature-Based Method For Parking Lot Occupancy. Future Internet (2019) doi:10.3390/fi11080169.
4. Acharya, D., Yan, W., Khoshelham, K.: Real-Time Image-Based Parking Occupancy Detection Using Deep Learning. 5th Annual Research@Locate Conference. Vol: 2087, (2018).
5. Nurullayev, S., Lee, S.W.: Generalized Parking Occupancy Analysis Based on Dilated Convolutional Neural Network. Sensors. Vol: 19, (2019) doi:10.3390/s19020277.
6. Amato, G., Carrara, F., Falchi, F., Gennaro, C., Vairo, C.: Car Parking Occupancy Detection Using Smart Camera Networks and Deep Learning. 2016 IEEE Symposium on Computers and Communication, pp. 1212-1217. IEEE (2016).

7. Khan, G., Farooq, M. A., Tariq, Z., Khan, M. U. G.: Deep-Learning Based Vehicle Count and Free Parking Slot Detection System. 22nd International Multitopic Conference, pp. 1-7, (2019).
8. Ekblad, U., Kinser, J., Atmer, J., Zetterlund, N.: The Intersecting Cortical Model in Image Processing. Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, pp. 392-396. (2004).
9. Ma, Y., Zhan, K., Wang, Z.: Applications of Pulse-Coupled Neural Networks. Springer Berlin Heidelberg (2011).
10. Lindblad, T., Kinser, J., Taylor, J.: Image Processing Using Pulse-Coupled Neural Networks. Springer Berlin Heidelberg (2013).
11. Chacon, M. I., Prieto, C. R., Sandoval, R., Rodriguez, A.: A Soft Image Edge Detection Approach Based on the Time Matrix of a PCNN. Proceedings of the International Joint Conference on Neural Networks, pp. 463-469. (2008).
12. Domínguez, K., Lavalle, M. M., Salazar, A. M., Salgado, G. R.: Pulsed Neural Net plus Time Matrix for Bright Images Enhancement. 2019 International Conference on Mechatronics, Electronics and Automotive Engineering, pp. 79-83. (2019).
13. Rangel, E., Lavalle, M., Sossa, H.: Filtrado de Ruido Gaussiano Mediante Redes Neuronales Pulso-Acopladas. Computación y Sistemas, pp. 381–395. Vol: 21 (2017).
14. Li, H., Guo, L., Yu, P., Chen, J., Tang, Y.: Image Segmentation Based on Iterative Self-Organizing Data Clustering Threshold of PCNN. 2016 2nd International Conference on Cloud Computing and Internet of Things, pp. 73–77 (2016).
15. Gollamudi, A., Calvin, P., Yuen, G., Bodruzzaman, M., Malkani, M.: Pulse Coupled Neural Network-Based Image Classification. Proceedings of Thirtieth Southeastern Symposium on System Theory, pp. 402-406, (1998).
16. Gu, X. Feature Extraction using Unit-linking Pulse Coupled Neural Network and its Applications. Neural Processing Letters, pp. 25–41. Vol:27 (2008).
17. Johnson, J. L.: Time Signatures of Images. IEEE International Conference on Neural Network, pp. 1279-1284. Vol: 2 (1994).
18. Turk, M., Pentland, A.: Face recognition using eigenfaces. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 586–591 (1991).
19. Mitchell, T.: Machine Learning, McGraw-Hill Science Engineering Math (1997).
20. Marsland, S.: Machine Learning an Algorithmic Perspective. CRC Press (2015).
21. Zuo, W., Zhang, D., Wang, K.: On kernel difference-weighted k-nearest neighbor classification. Pattern Analysis and Applications, pp. 247-257. Vol: 11 (2007).
22. Raihan-Al-Masud, M., Mustafa, H.: Network Intrusion Detection System Using Voting Ensemble Machine Learning. 2019 IEEE International Conference on Telecommunications and Photonics, pp. 1-4 (2019).
23. Almeida P. R. L., Oliveira, L., Britto, A., da Silva, J., Koerich, A.: PKLot: A robust dataset for parking lot classification. Expert Systems with Applications, pp. 4937-4949. Vol: 42 (2015).
24. Jollife I.T.: Principal Components Analysis. Springer Berlin Heidelberg (2002).