RESEARCH ARTICLE

INTERNATIONAL RESEARCH JOURNAL OF
MULTIDISCIPLINARY TECHNOVATION

Asian Research Association

DOI: 10.54392/irjmt2418

AOA based Masked Region-CNN model for Detection of Parking Space in IoT Environment

K. Sri Vijaya [a], V. Gokula Krishnan [b,*], D. Arul Kumar [c], B. Prathusha Laxmi [d], B. Yasaswi [e]

[a] Department of IT, PVP Siddhartha Institute of Technology, Vijayawada-520007, Andhra Pradesh, India

[b] Department of CSE, Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Thandalam, Chennai- 602105, Tamil Nadu, India

[c] Department of ECE, Panimalar Engineering College, Poonamallee, Chennai- 600123, Tamil Nadu, India

[d] Department of AIDS, R.M.K. College of Engineering and Technology, Gummidipoondi- 601206, Tamil Nadu, India

[e] Department of CSE, PVP Siddhartha Institute of Technology, Vijayawada-520007, Andhra Pradesh, India

*Corresponding Author Email: gokul_kris143@yahoo.com

DOI: https://doi.org/10.54392/irjmt2418

Received: 20-10-2023; Revised: 25-12-2023; Accepted: 05-01-2024; Published: 27-01-2024

Check for updates

Abstract: Uneven illumination has a significant impact on vision-based automatic parking systems, making it impossible to conduct a correct assessment of parking places in the presence of complicated picture data. In to address this issue, this work provides a deep learning-based system for visual recognition of parking spaces and picture processing. Artificial intelligence (AI) approaches can be used to identify a less expensive and easier-to-implement solution to the parking spot identification challenge, especially since the discipline of deep learning is reshaping the world. Using deep learning techniques, this study offers a dynamic, straightforward, and cost-effective algorithm for the detection of parking spots. In order to determine which parking spots are available and which are occupied, this method employs a Masked Region Based Convolutional Neural Network (MR-CNN) and the intersection over union approach. Cars in the training dataset were spaced more apart than those actually seen, which increased the accuracy of the identification between cars and parking spots. The AOA mechanism enhances the model's ability to focus on relevant regions within an image, improving accuracy in detecting parking spaces. This leads to precise identification of parking slots, reducing false positives and negatives. The sequence and quantity of parking spots, as well as the capacity to predict empty spots, were tested in a case study and found to be accurate. In the experimental results as the AOA based MR-CNN model stretched the accuracy as 98.50 and the recall value as 40.59 then the precision as 96.34 F1-measure as 57.95 correspondingly.

Keywords: Automatic Parking System, Masked Region based Convolutional Neural Network, Artificial Intelligence, Parking Spaces, Deep learning techniques.

1. Introduction

Almost every sector relies heavily on IoT-enabled apps. In the context of smart cities, finding a parking spot in a dense urban area is next to impossible. If someone wants to leave the building in their own automobile, the first thing they'll discuss is where they'll park [1]. It's really difficult to get a parking spot in a city without any restrictions. The vast majority of the time, motorists find that there are no open parking spots when they arrive, requiring them to find alternative parking. Parking garage in the hopes of finding a spot. It's also difficult to get street parking throughout holidays and other peak times [2]. Illegal parking and subsequent traffic congestion are the inevitable results. This has led to increased instances of inefficient space utilisation, blocked views for pedestrians, polluted air, and wasted time. This problem is getting steadily worse in Sri Lanka and other emerging nations compared to the developed world. The government now prioritises investments in digital infrastructure. There are many tasks that can be improved with the use of Google, and an astute vehicle improves management [3–4].

Since there is nowhere for the growing number of private automobiles to park, drivers are forced to abandon any semblance of order and leave their vehicles wherever they happen to be, leading to increased congestion and potential road care hazards [5]. There is also a lack of advice and an inability to satisfy the demands of car owners due to the absence of supervision of parking spots on the road and the parking lot being designated merely with or without parking spaces [6]. That's why it's crucial for urban planners to figure out how to make parking easy and accessible [7]. This project's solution allows drivers to easily and correctly interpret parking spot information on their mobile devices in real time, allowing them to pick the best option for themselves and their vehicles, therefore reducing pollution and the risk of accidents [8].

Recent years have been a focus on visual detection in AI studies, especially as they relate to autonomous vehicles. In conventional parking, when distance is detected via radar, a reference car must be present on either side of the parking place [9]. In addition, there are quite specific rules about where and how you can park. The aforementioned issues have prompted the suggestion of visible parking systems [10]. Over the past few decades, numerous researchers have amassed a wealth of knowledge on the topic of vision detecting parking systems. There have been a lot of studies and trials of intelligent parking systems. Using the current several intelligent systems have been built [11]. The most well-known of these are

- To get accurate parking spot advice, use computer vision and image processing technologies. Second, an optimum path planning guidance model is developed based on the classic Dijkstra algorithm [12]. Third, knowledge of graph theory is used to abstract modelling.
- ZigBee wireless communication technology underpins an intelligent parking guiding system. Parking other functions are all made possible by the system's use of magneto resistive sensors in conjunction with parking space detection procedures [13].

While the aforementioned two conventional approaches are capable of performing the necessary tasks, they also have glaring drawbacks: they are more difficult to install, and if the application is accepted, it requires extensive investment in costly and time-consuming infrastructure [14]. Research on the topic of target detection is extensive. Face recognition and pedestrian detection are two of its primary applications. However, spotting more targets is a difficult task. In this study, an MR-CNN-based model is used to locate available parking spots; the suggested model's weights are chosen using an AOA optimisation algorithm.

The remaining sections of the paper are structured as shadows: In Section 2, we offer the relevant literature; in Section 3, we provide a concise description of the suggested model; in Section 4, we present the experimental analysis; and in Section 5, we draw the necessary conclusions.

2. Related Works

In this review based on the AOA-based Masked Region-CNN model presents a cutting-edge solution for the detection of parking spaces in IoT environments. By leveraging the power of deep learning, spatial attention mechanisms, and seamless integration with IoT technologies, this model holds great promise in addressing the challenges posed by dynamic urban parking scenarios. As smart city initiatives continue to evolve, the adoption of such innovative models is crucial for enhancing urban mobility and optimizing parking space utilization.

To help those in need of a parking spot find one for free, Nithya *et al.* [15] suggest a Smart Parking System. The technology will be able to process photos of the parking lot and the available parking spots in real time, alerting the user to any open spots. Users have the option of selecting a parking spot that best suits their requirements. The technology will record events so that parking patterns may be compared and contrasted on various days. Faster Recurrent Convolutional Neural Networks (Faster R-CNN) are used in conjunction with the YOLOv3 method to accomplish the parking lot identification. This method uses a collection of images of cars to train a model that can identify vehicles in a parking lot. The suggested system exclusively detects parked automobiles, thus it is a highly constructive approach because it won't be confused with other temporary items in the parking lot. This system is stronger, uses less energy, and has the potential to be upgraded further.

In order to improve the accuracy with which parking spots will become available in smart parking, Tekouabou *et al.* [16] present a new approach that cartels the Internet of Things with a predictive perfect built using ensemble techniques. Our experiments on the Birmingham parking data set showed that the Bagging Regression (BR) technique could achieve an average Mean Absolute Error (MAE) of 0.06%. As a consequence, the best previously achieved performance has been enhanced by over 6.6%, while system complexity has been drastically reduced.

To this end, Wang *et al.* [17] select TechIO technology to enable wireless communiqué lot, examine the benefits of TechIO technology and the requirements of a parking management system, and construct a simulation based on the system design to conduct relevant tests on the system's components. From the results of the tests, it is clear that the system's use of TechIO technology results in its strong adaptability, removes the need for expensive and time-consuming wiring in the parking lot but also greatly simplifies the deployment of parking lot nodes.

The research by Abdellatif *et al.* [18] uses pre-processing, segmentation, and character recognition as the three image processing phases necessary to achieve high accuracy in licence plate identification. Finding the contours of the automobile and the licence plate requires a deft use of edge detection, contour detection, and masking techniques. This report presents the results of an experiment in which 200 photographs of Egyptian licence plates were used to identify the plates. With a 93% success rate, the algorithm was able to correctly identify Arabic licence plates. To evaluate the effectiveness of the system, a prototype is built with ESP32 cameras and a Raspberry-Pi. In addition, the RPi serves as host to a database and website that lets users locate their cars in the parking lot by entering their licence plate number, which is recorded in the database upon detection.

A cheap passive tag optimised for UHF RFID-band operation is proposed by Zahid *et al.*, which may be placed into a plastic display for parking spots [19]. Using the Python package Pymongo, we gather data from the RFID readers and send it to the MongoDB database using the message queuing telemetry transport (MQTT) protocol and the Scotland 5G-network so that we can make decisions and show open parking spots to passing drivers. Using the embedded JavaScript template (EJS) for each slot, a Python-based web app retrieves information from the database and displays it on a web-app display. A significant decline in the received-signal-strength-index (RSSI) value at the receiver end has been seen in experiments performed at parking slots when cars were present; this suggests that RSSI signal drop can be used for the identification of vehicle presence in the parking slot. The proposed method is meant to ease the process of finding affordable and efficient parking spots. Recorded and given are test results proving the suggested system's superior performance and accuracy in the circumstances used.

In Rajyalakshmi *et al.* [20], offer a model based on the Internet of Things that helps drivers make better use of available parking spots. This work uses Machine Learning (ML) and deep learning techniques to construct an algorithm called Hybrid Deep DenseNet Optimisation (HDDNO) to estimate parking space availability. Secondary data from Pisa, Italy's National Research Council Park (CNRPark) is used in the HDDNO-based ML model. Predictions of parking space availability at a specified time are made using a number of different regression techniques. While the HDDNO model provided more precise results, the DenseNet method has shown promise. Minimising the model's loss has been made possible with the use of five optimizers: Adaptive Delta (AdaDelta), and Stochastic Gradient Descent (SGD). Adam's contribution allowed the HDDNO model to accomplish a high level of accuracy (99.19%) and low loss (0.0306%) in its predictions. The proposed solution will greatly enhance environmental security and serve as a step towards the creation of "smart cities."

3. Proposed system

The ideal interpretation of the video stream, parking and collation of parking spaces, and judgement of parking spaces are the four components that make up the suggested technique.

3.1. Pre-processing

3.1.1. Image Enhancement

When capturing images, the camera will be disrupted by a variety of variables, leading to an abundance of noise in the final product. It will make locating parking spots more difficult. To increase recognition accuracy and suppress unwanted background noise, just the area of interest in the image is processed. The goal of filtering is to keep as much of the image's original features as possible while reducing the amount of background noise. There will be repercussions for later processing and identification due to the filtering effect. For the purposes of this work, parking detection, spatial domain improvement was chosen. When it comes to preserving fine details and reducing picture noise, median filtering is superior to Gaussian filtering. Median filtering was chosen after extensive evaluation. As an illustration, a 33 filter sample is used. The 8 surrounding pixels plus the pixels you want processed make up the filter template. Median filtering makes use of the fuzzy radius notion. The 3x3 grid has a fuzzy radius of 1. From the grey values of the template pixels, the grey value of the pixel to be processed is derived. The 9 pixels to the left, for instance, are assigned a value of 5 for calculations. It's used as the new pixel center's coordinate in the filter. After processing the entire image, you may do the median filtering procedure.

3.1.2. Binarization

The choice of threshold values or methods may require careful tuning based on the characteristics of the input images.

Threshold segmentation is a simple yet powerful tool in image processing, and it is often employed as a preprocessing step in more complex computer vision tasks. It provides a basis for isolating regions of interest and extracting meaningful information from images. Before parking spot detection, the picture needs undergo threshold segmentation. Binarization is the process of distinguishing between the foreground and background. Global binarization is the simplest kind of threshold segmentation. By calculating the grey levels throughout the whole image, it is able to derive the global threshold. Above that point, we use a value of 255 for the pixels. The values of the pixels below thresh are all zeroed out. The equation is (6) where x is the grey value of the input pixel, f(x) is the grey value of the computed output pixel, and T is the binarization threshold. To determine the best cutoff point, T, we apply the greatest inter-class variance technique OSTU.

$$f(x) = \begin{cases} 0 & x < T \\ 255 & x > T \end{cases} \qquad (1)$$

3.1.3. Data Augmentation

Data augmentation is a technique widely used in machine learning and computer vision to artificially increase the size of a dataset by applying various transformations to the existing data. The primary goal of data augmentation is to improve the generalization and robustness of machine learning models by exposing them to a more diverse range of examples during training. In the context of the AOA-based Masked Region-CNN model for parking space detection in an IoT environment, data augmentation plays a crucial role in enhancing the model's performance and adaptability. Here's an exploration of data augmentation in this specific application:By inverting the image of the data set, zooming in on the subset of data, and modelling a wide range of complicated situations, we are able to increase the precision and accuracy of the data set.

3.2. Detection of free space using MR-CNN

For this purpose, we will employ Mask R-CNN, a variant of R-CNNs that makes use of a Region Proposal Network (RPN) and the instance segmentation property for detection. The threshold value is calculated locally for each pixel based on the characteristics of its neighbourhood. This helps in handling variations in lighting across different regions of the image. Use Case: Effective when there is uneven illumination or varying contrast within the image.

3.2.1 Mask R-CNN

Thresholds are determined based on characteristics of the image histogram, such as peaks and valleys. Using the foundational Faster R-CNN architecture, Mask R-CNN is a cutting-edge deep learning technique for instance segmentation. Segmentation and object detection are combined in the instance segmentation algorithm. To find out where in a picture a specific item is located in terms of pixels, a segmentation method initially uses object detection. Figure 1 shows the result. The first step of Mask R-CNN analyses the photos and finds potential candidates for the item. The second phase involves making a pixel-level mask for the object, refining the bounding box, and predicting its class.

While the region suggestion construction stage is the similar in both Mask R-CNN and Faster R-CNN, the second stage in Mask R-CNN predicts class and creates bounding boxes for each RoI in parallel in addition to producing a binary mask. The core is a top-bottom lateral connection deep neural network composed of RestNet101 + FPN. This multi-layer feature pyramid network generates ROI at many scales, which improves the efficiency of the ResNet architecture, and it does so by extracting the feature map from the photos. The feature map's size will be cut in half or perhaps doubled based on the feature pyramid network. Feature maps are generated using the top-down methodology. The suggested model's internal structure is seen in Figure 2.

To go from the smallest to the largest, it starts with the highest-level feature map and works its way down. Prior to upsampling, the number of streams is limited to 256 using the 1x1 convolution. Next, the result of the previous upsampling cycle is gradually added to the preceding. The last 4 retrieved features are generated by subjecting all outputs to a 3 X 3 convolution layer. A max pooling procedure on the fifth feature map yields the sixth feature map. The RPN network's previous convolution layer's complete convolution feature map is then subjected to a 3x3 convolution layer. The results from these sub-processes—which determine the object's score and regress its bounding box coordinates—are then combined and sent on.

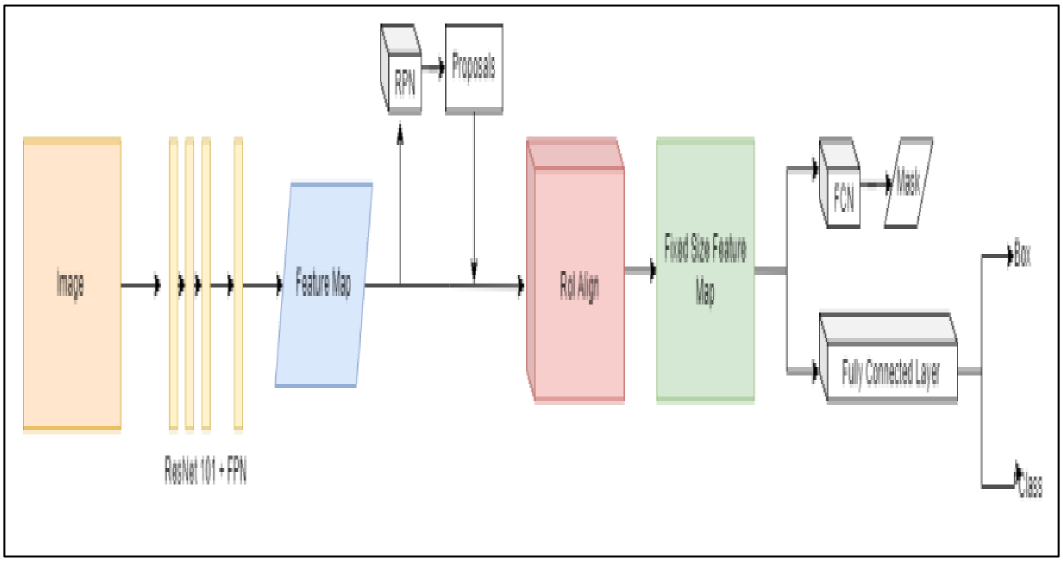

Figure 1. The construction diagram of Mask R-CNN

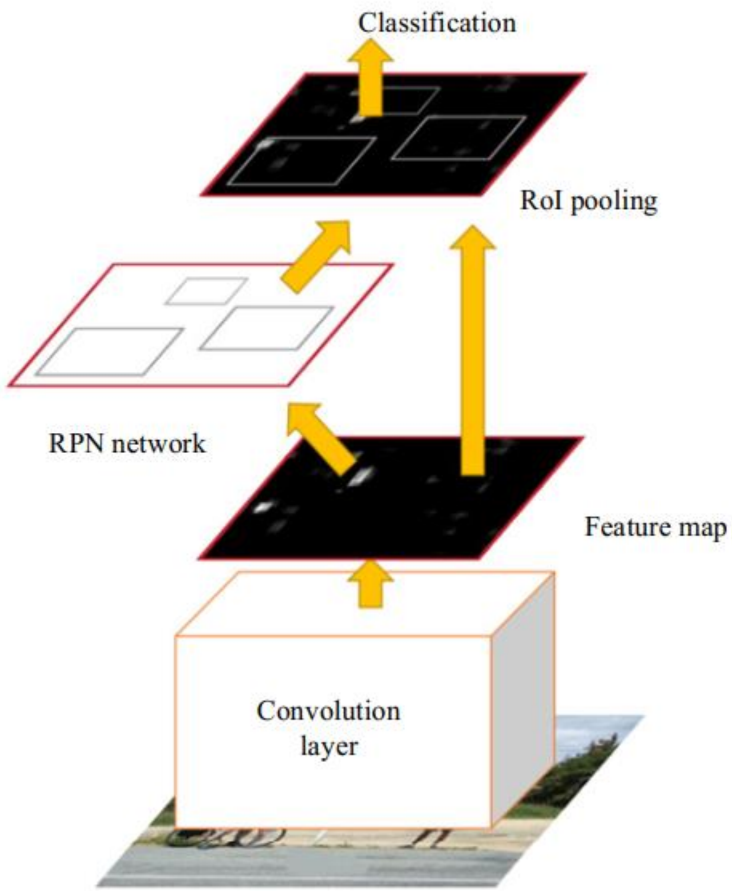

Figure 2. MR-CNN model structure

3.2.2 ROI Align

RoI Align has replaced RoI pooling because RoI pooling quantizes at the first and second stages, degrading input significantly if given to the final layer. As a consequence, RoI Align will reduce redundant data, as it does not perform quantization at any step. Starting with the HxW feature map from the preceding Convolution layer, the ROI Align grids it into MxN squares of the same size. By including a segmented branch into the architecture, we were able to quadruple the frames per second of the Mask R-CNN's interpretation performance.

3.2.3 The Mask Representation

In this case, the Mask R-CNN will predict the mask using a fully connected network. The pixel correlation with the aforementioned layers is necessary because combining the results of the classification and regression layers improves performance. The mask of form MxM is the result of a fully connected convolution network processing features input from the region of interest. A convolutional layer (1x1) in the fully connected layer improves the mask, while the layer above it reduces the channel size to 256.

Image segmentation is a key component of Mask R-CNN. Segmenting an electronic photograph into its constituent parts is known as picture segmentation. Segmenting digital images into fixed categories (semantic segmentation) and variable categories (instance segmentation) are two of Mask R-CNN's primary methods of segmentation. The implementation of a dynamic algorithm that favours intersection over union will benefit from this. As the model needs to be retrained numerous times in response to changes in the environment, the alternative method of training a model from scratch using the empty and filled parking spot categories will be a simpler methodology, but the entire process will not be dynamic. A model could never be trained with a negligible investment of time, energy, or information.

3.2.4. Hyper-parameter tuning using Archimedes optimization algorithm (AOA)

Population-based is the key concept of AOA. The suggested method treats members of the population as submerged entities. An initial volumes, densities, and accelerations is used in AOA, as it is in other population-based metaheuristic algorithms. Each object's initial fluid location is likewise initialised at this time. AOA iteratively improves a population until a termination condition is met by assessing the fitness of the original population. AOA revises the mass and volume of all objects in each cycle. When an item collides with another nearby object, the collision condition is used to update the object's acceleration. The new location of an item is calculated using its current density, volume, and acceleration. Here is the whole mathematical equation for each stage in AOA.

A) Algorithmic steps

Here, we present the AOA formulation. Since AOA theoretically incorporates both exploration and exploitation, it may be viewed as a global optimisation method. The suggested AOA is broken down into the following mathematical steps:

Step 1: Initialization Prime the places of altogether objects using (2):

$$O_i = lb_i + rand \times (ub_i - lb_i); i = 1,2,\ldots,N \quad (2)$$

where O_i is the ith object in a population of N objects. lb_i and ub_i are the bounds of the search-space, correspondingly.

Initialize volume for each ith object using (3):

$$den_i = rand$$

$$vol_i = rand \quad (3)$$

Where rand is a D-dimensional vector generating a number between zero and one at random. Finally, using (4), set the starting acceleration (acc) of the ith object.:

$$acc_i = lb_i + rand \times (ub_i - lb_i) \quad (4)$$

In this step, evaluate initial populace and choice the object with the finest value. Allocate x_{best}, den_{best}, vol_{best}, and acc_{best}.

Step 2: Inform densities, volumes, the object i for the iteration t + 1 is updated using (5):

$$den_i^{t+1} = den_i^t + rand \times (den_{best} - den_i^t)$$

$$vol_i^{t+1} = vol_i^t + rand \times (vol_{best} - vol_i^t) \quad (5)$$

where vol_{best} and den_{best} are the allied is consistently distributed random sum.

Step 3: Operators of transfer and density factors Initially, items collide, and then, after some time passes, they attempt to settle into a stable equilibrium. Transfer operator T F, defined by (6), is used in AOA to accomplish this transition from exploratory to exploitative searching.

$$TF = exp\left(\frac{t - t_{max}}{t_{max}}\right) \quad (6)$$

where the time constant of the transfer, T F, rises steadily until it reaches 1. Here, t is the total sum of iterations, and t_max is the extreme sum of iterations. The density-reducing factor d also helps AOA with its global-to-local search. The formula for its decay over time is (7):

$$d^{t+1} = exp\left(\frac{t_{max} - t}{t_{max}}\right) - \left(\frac{t}{t_{max}}\right) \quad (7)$$

where d^{t+1} gradually diminishes with time, allowing convergence on a location already recognised as promising. Keep in mind that this variable, if managed correctly, will keep AOA's exploration and exploitation in check.

Phase of exploration (in which items collide): Step 4.1 Select a random material (mr) and adjust the object's iteration t + 1 using (8) if the collision threshold is less than half.

$$acc_i^{t+1} = \frac{den_{mr} + vol_{mr} \times acc_{mr}}{den_i^{t+1} \times vol_i^{t+1}} \quad (8)$$

where den_{i}, vol_{i}, and acc_{i} are object i. Whereas acc_{mr}, den_{mr} and vol_{mr} are the acceleration, density, and volume of accidental material. It's worth noting that a TF of 0.5 guarantees exploration occurs every third iteration. The exploration-exploitation trade-off is affected by the value applied.

Phase of exploitation (no object collisions) 4.2. If the collision threshold is greater than 0.5, then the acceleration of each object is updated for the next iteration at time t+1 using (9).

$$acc_i^{t+1} = \frac{den_{best} + vol_{best} \times acc_{best}}{den_i^{t+1} \times vol_i^{t+1}} \quad (9)$$

where acc_{best} is the hastening of the best object.

Normalize hastening to compute the percentage of change using (10):

$$acc_{i-norm}^{t+1} = u \times \frac{acc_i^{t+1} - \min(acc)}{\max(acc) - \min(acc)} + l \quad (10)$$

Where u is the normalisation range and l is 0.9, and so on. Each agent's proportion of step change is calculated using the acc_(i-norm)(t+1). A high acceleration value indicates that object i is in the exploration phase, whereas a low value indicates that object i is in the exploitation phase. This exemplifies the progression from the exploratory to the exploitative stages. The acceleration factor often starts at a high number and gradually diminishes over time. This directs search agents' attention away from suboptimal local options and towards the world's best answer. It is worth noting, however, that there may be a small number of search agents that require a longer investigation period than is typical. Thus, AOA strikes a happy medium between discovery and exploitation.

Step 5: Change of status Position of the ith item at the end of the next cycle t+1 using (11)

$$x_i^{t+1} = x_i^t + C_1 \times rand \times acc_{i-norm}^{t+1} \times d \times (x_{rand} - x_i^t) \quad (11)$$

where C_1 is constant equals to 2. Else, if $TF > 0.5$ the objects update their sites using (12).

$$x_i^{t+1} = x_{best}^t + F \times C_2 \times rand \times acc_{i-norm}^{t+1} \times d \times (T \times x_{best} - x_i^t) \quad (12)$$

C_2 = 6, where C is a constant. T = C_3 TF defines T as an increasing function of time that is directly time in initially steals a fixed proportion of the top spot. The initial percentage is low because a wide gap between the optimal and current positions means the random walk will proceed with huge steps. This percentage continuously rises as the search progresses in order to narrow the gap among the best possible position and the present one. This ultimately leads to striking a healthy middle ground between discovery and exploitation.

Using (13) as an example, pressing F will reverse the current direction of travel:

$$F = \begin{cases} +1 \; if \; P \le 0.5 \\ -1 \; if \; P > 0.5 \end{cases} \tag{13}$$

Where $P = 2 \times rand - C_4$.

Step 6: Evaluation Assess each finest solution found so far. Allocatex_{best}, den_{best}, vol_{best}, andacc_{best}.

3.3. Judgment of Parking Space Vacancy

The percentage of parking spots not occupied was used to make that determination. Some parking places are obviously filled in the picture frame shown in Figure 3, while others are plainly empty. In order to determine whether or not a parking spot was available, random points were generated inside of each designated spot. Within a one-square-meter area, the distribution of random points was represented as:

$$a \le n \le b \tag{14}$$

Where a and b stand for the minimum and maximum values, respectively. Temporarily, the size of each designated parking spot was determined by using the formula::

$$S = (x_{max} - x_{min})(y_{max} - y_{min}) \tag{15}$$

Then, the possible distribution of random locations inside each designated parking spot was recorded as:

$$Sa \le n \le Sb \tag{16}$$

Here, we focused exclusively on that cap. In the end, there were Sb random spots inside each spotted parking spot.

The ratio of randomly chosen locations within a parking place was used as the primary metric for evaluating a vehicle's level of coverage. Using a density clustering technique, we can identify the random locations traversed by a vehicle. The model used to evaluate the parking spot was built using the indirect Monte Carlo approach. The percentage of total points that were in the vehicle was determined and compared to the cutoff. If the ratio was lower than the threshold, then the spot was taken. The parking spot was considered available if the ratio was less than the threshold.

The time cost of identifying parking spaces for varying statistics of available parking spots was used to arrive at a threshold value. There were obviously longer delays for the same number of empty parking spots. However, PSD requires at least 6 s to whole, thus the impact of the available parking spots on the process period is significant and should be taken into account. Concurrently, the time expenditures associated with the various cutoffs were quite similar. As a result, we shouldn't think about how various criteria can affect the duration of operations. The cutoff score ended up being 0.8.

4. Results and Discussion

There are already a plethora of options for creating deep learning applications. For the purposes of this article, TensorFlow was selected as the deep learning experimental framework. This system was used for everything related to the network: building, training, and tagging. Google's TensorFlow is a deep learning framework. The structure is very adaptable and easy to move around. It's widely recognised as a top deep learning development platform.

This is because there is a mountain of data produced when the deep learning model for parking spot recognition is being trained. It is simple to overload the memory, rendering the model untrainable. This resulted in doubling the amount of the RAM to 12 GB. The model employs a huge number of convolution operations, which makes deep learning CPU-intensive. Training a model might take weeks or months. In this test, a GeForce GTX 1060 graphics card was utilised to rapidly train a custom-made deep learning model. The original 20-day training period was cut down to fewer than 10. Table 1 displays the specific hardware device configuration data.

4.1. Data Set Production

Figure 4 shows the test the suggested approach for general use by applying it to photos of various automobiles. Training sets consisted of 6387 photos of automobiles, whereas validation sets consisted of 1146 images (see [21] for details). The 7533 photos were labelled by hand using tool, and then the resulting footnote files were transformed to TensorFlow's standard TFRecord data format using a Python script. The suggested model's sample results are shown in Figure 5.

4.2. Results of Proposed IoT Model

When a matching licence plate is detected, the Raspberry Pi activates the servo motor it controls. If the car is found, the system (controlled by a Raspberry Pi) prepares to take a picture of the plate. If it does, the system logs the car and triggers the DC motor.

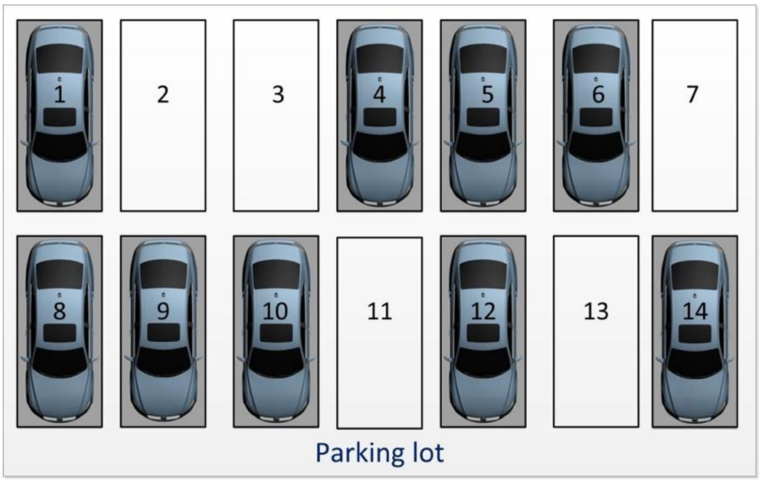

Figure 3. Parking space number

Table 1. Deep learning hardware stage info

Hardware Equipment	Amount	Model Stipulations
memory stick 1	1	Kingston DDR3 (8G memory)
memory stick 2	1	Kingston DDR3 (4G memory)
graphics card	1	NVIDIA GeForce GTX 1060 (6G VRAM)
CPU mainframe	1	Intel Core i5-3470

Figure 4. Partial vehicle imageries in the data set

Furthermore, we use a speed hump in the centre of the gap (X) to give the PI cameras time to spot the automobile and begin the number plate identification procedure. We've used the same strategy for tracking vehicles leaving the building. The Raspberry Pi's output is connected to the LCD so that it may show available slots and other data. No one checkpoint exists to record which vehicles arrive and leave a certain time period. As a result, it is necessary to keep an eye on each vehicle in its own designated parking spot.

LED signs inform cars as to whether a parking spot is free or occupied by relaying information from infrared (IR) sensors installed in each space. Meanwhile, sensors communicate with an ESP32 through the internet to provide consumers with real-time information about open parking spots. Figures 6-8 provide examples of the suggested model's output.

Figure 5. Schematic design of vehicle documentation consequences

Figure 6. LCD Screen – Welcome

Figure 7. Vehicle Slots Availability Display

Figure 8. Output Message – Display

A liquid crystal display (LCD) is utilised as a parking display, with park slot detectors confirming the presence or absence of motorcars and displaying a total number of vehicles in the parking lot.

When a car pulls into a parking place, the infrared detector there alerts the system with a high reading in that slot's vicinity. In addition, the system provides an output for the LCD to show the current slot status. If every parking spot is taken, the LCD displays "Sorry not Space Available," and the parking gate shuts entirely.

4.3. Performance Analysis of Proposed Classifier

The presentation of projected model is tested with existing models in terms of different metrics that is shown in Table 2.

Table 2. Classifier Result analysis

Method	Accuracy	Recall	Precision	F1-measure
DBN	97.90	21.06	00.21	11.14
CNN	97.79	65.47	25.22	36.42
MR-CNN	85.00	15.11	00.45	10.64
AOA based MR-CNN	98.50	40.59	96.34	57.95

In Table 2 characterise that the Classifier Result analysis. In the analysis of DBN model gotten the accuracy as 97.90 and the recall value as 21.06 then the precision as 00.21 F1-measure as 11.14 correspondingly. Then the CNN model touched the accuracy as 97.79 and the recall value as 65.47 then the precision as 25.22 36.42 correspondingly. Then the MR-CNN model reached the accuracy as 85.00 and the recall value as 15.11 0 and the recall value as 0.45 F1-measure as 10.64 correspondingly. Then the AOA based MR-CNN model stretched the accuracy as 98.50 and the recall value as 40.59 then the precision as 96.34 F1-measure as 57.95 correspondingly.

5. Conclusion

The deployment of this technology will have a significant positive impact on parking. The economic, social, and safety aspects of society, as well as environmental conservation, can all benefit from the implementation of this system. The city's parking infrastructure has been upgraded in an effort to raise citizens' standard of living. It might take a while to find a parking place in densely populated areas, so it's crucial that drivers have access to helpful, cutting-edge technologies. In this research, we suggested using deep learning for object recognition to identify parking spots in a parking lot. The proposed strategy involved locating parking spots and pinpointing their specific distribution. Indirect Monte Carlo analysis was found to be useful for estimating parking demand. To test the efficacy of the suggested strategy, we performed detection in a big parking lot using actual settings. The detection results demonstrated that the suggested technique effectively identified parking spots when cars entered and exited them. However, the single graphics card we own isn't powerful enough to run both the training and testing software. The training application was transferred to the graphics processing unit for instruction. The central processing unit was used to run the test programme. Due to the test program's heavy demands on memory and processing power, the GPU's training speed has been significantly reduced. The total time spent training was 190 hours. Two graphics cards, with one used for training and the other for testing, might be used to speed up deep learning projects in the future. Investigate the applicability of transfer learning and domain adaptation techniques to fine-tune the model for different environments. This can facilitate the deployment of the model in diverse urban landscapes with varying characteristics.

References

[1] G. Manjula, G. Govinda Rajulu, R. Anand, J.T. Thirukrishna, (2022) Implementation of smart parking application using IoT and machine learning algorithms. In Computer Networks and Inventive Communication Technologies: Proceedings of Fourth ICCNCT 2021 Springer Singapore. https://doi.org/10.1007/978-981-16-3728-5_18

[2] W.A. Jabbar, C.W. Wei, N.A.A.M. Azmi, N.A. Haironnazli, An IoT Raspberry Pi-based parking management system for smart campus. Internet of Things, 14 (2021) 100387. https://doi.org/10.1016/j.iot.2021.100387

[3] S. GokulKrishna, J. Harsheetha, S. Akshaya, & D. Jeyabharathi, An IoT based smart outdoor parking system. In 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), IEEE, 1 (2021) 1502-1506. https://doi.org/10.1109/ICACCS51430.2021.9441766

[4] H. Mohapatra, A.K. Rath, An IoT based efficient multi-objective real-time smart parking system. International journal of sensor networks, 37 (2021) 219-232. https://doi.org/10.1504/IJSNET.2021.119483

[5] K.S. Kaleeem, A.S. Raju, N. Giweli, A. Dawoud, P.W.C. Prasad, M.A. Kashef, (2021) IoT Regression Techniques in Smart Parking Systems: Survey. In 2021 6th International Conference on Innovative Technology in Intelligent System and Industrial Applications (CITISIA), IEEE, Australia. https://doi.org/10.1109/CITISIA53721.2021.9719884

[6] D. Vuk, D. Androcec, (2022) Application of machine learning methods on IoT parking

sensors' data. In Proceedings of Sixth International Congress on Information and Communication Technology: ICICT 2021, Springer Singapore. London. https://doi.org/10.1007/978-981-16-2380-6_14

[7] O. Makke, O. Gusikhin, (2021) Robust IoT based parking information system. In Smart Cities, Green Technologies, and Intelligent Transport Systems: 9th International Conference, SMARTGREENS 2020, Springer International Publishing. https://doi.org/10.1007/978-3-030-89170-1_11

[8] S. Shukla, R. Gupta, S. Garg, S. Harit, R. Khan, (2022) Real-Time Parking Space Detection and Management with Artificial Intelligence and Deep Learning System. In Transforming Management with AI, Big-Data, and IoT Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-86749-2_7

[9] A.T. Kabir, P.K. Saha, M.S. Hasan, M. Pramanik, A.J. Ta-Sin, F.T. Johura, A.M. Hossain, (2021) An IoT based intelligent parking system for the unutilized parking area with real-time monitoring using mobile and web application. In 2021 International Conference on Intelligent Technologies (CONIT), IEEE, India. https://doi.org/10.1109/CONIT51480.2021.9498286

[10] H. Huang, C. Song, J. Ting, T. Tian, C. Hong, Z. Di, D. Gao, Design of An IoT based Smart Parking Lock. In Journal of Physics: Conference Series, IOP Publishing, 1952 (2021) 042025. https://doi.org/10.1088/1742-6596/1952/4/042025

[11] Y. Agarwal, P. Ratnani, U. Shah, P. Jain, (2021) IoT based smart parking system. In 2021 5th international conference on intelligent computing and control systems (ICICCS), IEEE. 464-470. https://doi.org/10.1109/ICICCS51141.2021.9432196

[12] S. Suthir, P. Harshavardhanan, K. Subramani, P. Senthil, T. Veena, V. Nivethitha, Conceptual approach on smart car parking system for industry 4.0 internet of things assisted networks. Measurement: Sensors, 24 (2022) 100474. https://doi.org/10.1016/j.measen.2022.100474

[13] S. Park, D-park: User-centric smart parking system over ble-beacon based internet of things. Electronics, 10 (2021) 541. https://doi.org/10.3390/electronics10050541

[14] G. Pau, F. Arena, Smart city: the different uses of IoT sensors. Journal of Sensor and Actuator Networks, 11 (2022) 58. https://doi.org/10.3390/jsan11040058

[15] R. Nithya, V. Priya, C. Sathiya Kumar, J. Dheeba, K. Chandraprabha, A smart parking system: an IoT based computer vision approach for free parking spot detection using faster R-CNN with YOLOv3 method. Wireless Personal Communications, 125 (2022) 3205-3225. https://doi.org/10.1007/s11277-022-09705-y

[16] S.C.K. Tekouabou, W. Cherif, H. Silkan, Improving parking availability prediction in smart cities with IoT and ensemble-based model. Journal of King Saud University-Computer and Information Sciences, 34 (2022) 687-697. https://doi.org/10.1016/j.jksuci.2020.01.008

[17] A. Wang, Z. Qin, Y.H. Dong, Development of an IoT-Based Parking Space Management System Design. International Journal for Applied Information Management, 3 (2023) 91-100. https://doi.org/10.47738/ijaim.v3i2.54

[18] M.M. Abdellatif, N.H. Elshabasy, A.E. Elashmawy, M.A. Abdel Raheem, low cost IoT-based Arabic license plate recognition model for smart parking systems. Ain Shams Engineering Journal, 14 (2023) 102178. https://doi.org/10.1016/j.asej.2023.102178

[19] A. Zahid, N. Mufti, S. Ullah, M.W. Nawaz, A. Sharif, M.A. Imran, Q.H. Abbasi, IoT-Enabled Vacant Parking Slot Detection System Using Inkjet-Printed RFID Tags. IEEE Sensors Journal, 23 (2023) 7828-7835. https://doi.org/10.1109/JSEN.2023.3246382

[20] V. Rajyalakshmi, K. Lakshmanna, (2023) Detection of car parking space by using Hybrid Deep DenseNet Optimization algorithm. International Journal of Network Management, e2228. https://doi.org/10.1002/nem.2228

[21] Q. An, H. Wang, X. Chen, EPSDNet: Efficient Campus Parking Space Detection via Convolutional Neural Networks and Vehicle Image Recognition for Intelligent Human–Computer Interactions. Sensors, 22 (2022) 9835. https://doi.org/10.3390/s22249835

Authors Contribution Statement

The author K. Sri Vijaya performed the initial screening and defined the problem statement. Author V. Gokula Krishnan involved in manuscript preparation/writing and reviewing the literature. Author D. Arul Kumar performed the data collection and analysis part of this research work. Author D. Prathusha Laxmi performed the simulation work and generated the results and author B. Yasaswi done the paper formatting and proof-reading works. All authors read and approved the final version of this manuscript.

Funding

The authors declare that no funds, grants, or other supports were received during the preparation of this manuscript.

Competing Interests

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Data Availability

Data will be provided upon request.

Has this article screened for similarity?

Yes

About the License