ConciseCarNet: convolutional neural network for parking space classification

Marwan Ramli[1], Sayuti Rahman[2], Rahmad Bayu Syah[2]
[1]Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Syiah Kuala, Banda Aceh, Indonesia
[2]Department of Informatics, Engineering Faculty, Universitas Medan Area, Medan, Indonesia

Article Info

Article history:

Received Jul 24, 2023
Revised Apr 2, 2024
Accepted Apr 17, 2024

Keywords:

Concise car network
Convolutional neural networks
Deep learning
Parking space
Smart parking system

ABSTRACT

The car is a mode of transportation that brings numerous benefits to the community. As a result, the growth of vehicles is increasing, which has a negative impact. Some of the negative impacts include noise, air pollution, traffic congestion, and the need for parking spaces. Drivers that drive around looking for parking places increase the negative impact as well as boredom and even worry for the driver. Therefore, the driver needs this information on the availability of parking spaces. A convolutional neural network (CNN) using a camera is one of the best methods that can be used to solve this problem. We built a more efficient CNN architecture for classifying parking spaces, which was named ConciseCarNet. ConciseCarNet uses 3×3 and 1×1 convolution filters, which cause fewer parameters than previous architectures. ConciseCarNet has two branches, each with a different branch structure. This branch is designed to generate additional feature variations, which will help improve the accuracy. Based on testing, the accuracy of ConciseCarNet2x outperforms the accuracy of mAlexnet, Carnet, EfficientParkingNet, and you look once (YOLO)+MobilNet architectures, which is 99.37%. ConciseCarNet has fewer parameters, file sizes, and floating point operations (FLOPs) compared to other architectures.

Corresponding Author:

Sayuti Rahman
Department of Informatics, Engineering Faculty, Universitas Medan Area
Jalan Kolam No. 1 Medan Estate, Medan, Indonesia
Email: sayutirahman@staff.uma.ac.id

1. INTRODUCTION

Nowadays, the use of cars for transportation, both within urban areas and beyond, has become essential. The popularity of cars is on the rise due to their greater capacity compared to motorcycles and their ability to shield occupants from external air pollution. Despite the advantages, the surge in the prevalence of cars brings about adverse effects, including air pollution, noise, limited parking availability, traffic congestion, and heightened fuel consumption [1]. Vehicle exhaust emissions contribute to air pollution, posing risks to public health, and are a significant source of anthropogenic carbon dioxide (CO_2) [2]. Furthermore, the escalating number of cars has intensified the demand for parking spaces, leading to challenges such as traffic congestion caused by drivers searching for parking space. Consequently, implementing an intelligent parking system emerges as a viable solution to address this issue.

Smart parking systems offer drivers various benefits, aiding them in avoiding traffic congestion, inefficient fuel usage, long vehicle queues, anxiety, and environmental pollution. Research in the realm of smart parking systems encompasses aspects such as parking space availability detection, parking meters, crowdsensing, integrated vehicle development, simulation, parking data analysis, and competition in parking space reservations [3]. The identification of vacant parking spaces stands out as a crucial element of smart parking systems. This involves gathering data to inform drivers about available parking spots, with sensors or computer vision being common methods for parking space detection. Computer vision cameras, for instance, can simultaneously classify numerous parking spaces, presenting a more cost-effective alternative compared to installing and maintaining sensors in each individual parking space.

Many approaches for determining the availability of parking places with good computer vision have been developed by researchers. Some of the methods produced include detecting parking spaces using image subtraction [4], giving markers to parking spaces to be recognized [5], [6], adaptive approach to space constraints with cubes [7], and learning base [8], [9]. Among these, learning-based methods, particularly deep learning, are frequently developed to address this problem.

Deep learning is a particularly successful method for handling computer vision difficulties. The deep learning method that is often used to classify objects in images is convolutional neural networks (CNN) [10]. Researchers have successfully used CNN to classify objects in images. Existing CNN with large enough parameters include Alexnet [10], visual geometry group network (VGGNet) [11], GoogleNet [12], and Resnet [13]. Currently, CNN architectures for mobile applications have been found, including MobileNet V2 [14], and ShuffleNet [15] which have a smaller number of parameters. This existing CNN is aimed at classifying 1,000 objects from the Imagenet dataset. Thus, the existing CNN has a large amount of computing and storage space. Classification of objects with fewer classes than Imagenet, the new architecture as needed will be better. however, some researchers prefer to exploit CNN, such as transfer learning [16], fine-tuning [17], or pruning [18] against existing CNN.

mAlexnet and mLenet are CNN architectures specifically developed for parking space classification. mAlexnet and mLenet adopt the Alexnet [10] and Lenet [19] architectures, but are more compact. mAlexnet has better accuracy than mLenet. In the parking lot (PKLot) and consiglio nazionale delle ricerche parking (CNRParkEXT) datasets, mAlexnet produces more than 90% accuracy. Alexnet and mAlexnet exhibit nearly identical accuracy levels, with a marginal difference of approximately 1% [8]. Subsequently, CarNet was developed [9], and it effectively enhanced the accuracy of parking space classification within the same dataset. However, due to its extensive architecture, CarNet possesses a high number of parameters, making it unsuitable for low-computing devices like the Raspberry Pi. Not satisfied with this, because the number of parameters for mAlexnet is still much smaller than that of CarNet, fine-tuning is done on mAlexnet. Fine-tuning the filter on the convolution layer and the activation function can increase the accuracy of mAlexnet [20], although not significantly. After that, it was found that EfficientParkingNet [21] was better in accuracy and speed for parking space classification. However, for real-time information speed, it is necessary to develop CNN architecture that can classify parking spaces more quickly and accurately.

The CNN computational process is very large, especially during training, so it can take quite a long time [22]. As a result, several researchers have focused on speeding up CNN training and testing times [23], [24]. Motivated by the need for information on the availability of parking spaces, the research proposed in this paper is to propose a new, more efficient architecture. The proposed architecture can be applied to low-cost computing devices, to reduce classification costs. The proposed architecture also has better accuracy than previous studies. In addition, this architecture has a number of parameters, floating point operations (FLOPs), and smaller file size so that it is faster in parking space classification. This architecture can provide information on the availability of parking spaces in real-time at a low cost.

The availability of parking spaces is an important part of a smart parking system. Drivers require quick and accurate parking space availability information. Drivers may quickly select their preferred parking area, reducing the time it takes to find a parking space. Sensors or vision-based technology are often utilized in the classification of parking spaces [25]. Based on budget usage, the application of sensor-based technology for parking space classification requires the more expensive installation and maintenance costs than using vision-based technology. A smart camera that uses computer vision can classify multiple parking spaces at once. In addition, computer vision can generate some additional information.

Computer vision is widely applied to help humans in various fields. Computer vision works like the human eye. Computers can see and provide information about what they see to humans. Therefore, researchers have applied computer vision to classify parking spaces. Researchers develop research to provide fast and accurate parking space availability information. Several computer vision methods for parking space classification include marking parking spaces, line-based detection, and learning base. Research related to parking space classification is presented in Table 1 [8], [9], [21], [26]–[34].

Table 1 shows several methods for computer vision-based parking space classification. mAlexnet is the first CNN used for parking space classification [8]. After that, CarNet appeared to improve the accuracy of mAlexnet, but the number of parameters and FLOPs is vast. The classification accuracy of Yolo+MobileNet on the CNRParkEXT dataset reaches up to 98.97% [31]. GDSN-IC classified the CNRParkEXT sub-dataset with an accuracy of 97.10% [32]. Mini Shufflenet [33] by pruning ShuffleNet V2 produces an accuracy of 98.37. EfficientParkingNet is an architecture that can be implemented on low-end computing devices [21]. In addition to its accuracy reaching 98.44%, EfficientParkingNet also has fewer parameters and FLOPs. However, accuracy still needs to be improved to produce more accurate and faster information.

Table 1. Research related to parking space classification

Methods	Dataset	Accuracy (%)
Feature point detection and color histogram classification [26]	Self Collecting	85.20
Support Vector Machine [27]	Self Collecting	85.57
Rotation-based method and mosaic 3D structures [28]	Self Collecting	90.00
Threshold value [29]	Self Collecting	98.00
Local Binary Patterns (LBP) and Local Phase Quantization (LPQ [30]	PKLot	< 89.83
mAlexnet [8]	PKLot and CNRParkEXT	91.34
CarNet [9]	CNRParkEXT	97.24
You Look Only Once (Yolo) and MobileNet [31]	CNRParkEXT	98.97
Grassmannian Deep Stacking Network with Illumination Correction (GDSN-IC) [32]	CNRParkEXT	97.10
EfficientParkingNet [21]	CNRParkEXT	98.44
Mini Shufflenet [33]	CNRParkEXT	98.37
YoloV5-Pretrained [34]	PKLot	99.50

Our motivation in this research is to find a new architecture that can improve the accuracy PKLot classification. This architecture also has a smaller number of parameters and FLOPs so it is faster in classification. This architecture is suitable for implementation on budget-friendly computing devices, such as the Raspberry Pi, so it can be implemented in low-cost parking systems and be a solution to problems that arise when driving around looking for a parking space.

2. METHOD

This study focuses on creating a more efficient CNN architecture for classifying parking spaces. In this work, we compared the architecture discovered with existing approaches. We make observations in order to identify architectures that are more efficient in terms of accuracy and speed. The proposed CNN dataset and architecture are further described in sections 2.1 and 2.2.

2.1. Datasets

CNN architecture testing using the PKLot dataset and CNRParkEXT. The PKLot and CNR Park-EXT data sets represent the real situation of parking spaces in terms of weather conditions, camera angles, and obstacles in the parking space. PKLot has 12,417 full images with a resolution of 1280×720 pixels. PKLot contains datasets taken from two parking areas, named parking1 and parking2. Parking1 is a parking space dataset from two cameras, namely parking1a and parking1b. Parking2 is a parking space dataset captured from a camera. The PKLot dataset consists of three weather conditions such as sunny, rainy, and cloudy. The dataset consists of two labels (free/busy) consisting of 695,900 parking space images. The image on parking1a, parking1b, and parking2 are as shown in Figures 1(a) to 1(c):

(a) (b) (c)

Figure 1. PKLot dataset showing two parking areas and three camera points, namely (a) Parking1a, (b) Parking1b, and (c) Parking2 [30]

The second dataset is CNRParkEXT consisting of CNRPark and CNR-EXT. The CNRParkEXT dataset originates from CNR Research in the City of Pisa, Italy. It comprises 242 complete images and 12,584 images specifically focusing on parking spaces in the CNRPark subset. Additionally, CNR+EXT includes

4,081 complete images and a substantial 144,965 parking space images captured from nine different cameras. CNRPark contains datasets taken during sunny weather, while CNR+EXT contains datasets captured in sunny, rainy, and cloudy weather. CNRParkEXT is a dataset with many obstacles such as balls, trees, and lamp posts. CNRParkEXT is presented in Figure 2.

Figure 2. Two out of nine camera points CNRParkEXT [8]

Figures 1 and 2 are full images. The full image is cropped according to the shape of the parking space and resized to a size of 224×224 pixels for CNN input. The cropped images results are shown in Figure 3. The evaluation was performed on a computer equipped with the following specifications: 4 gigabytes of RAM, an Intel Core i3 processor clocked at 3.30 GHz, running on a 64-bit operating system. The testing process employed Python, utilizing the PyTorch framework.

Figure 3. Results of image cropping for parking spaces

The dataset is divided into subsets that will be utilized for data training and testing. This study employed identical data sharing and testing approaches to prior experiments on the mAlexnet and CarNet models. Table 2 contains detailed information about the dataset.

Table 2. Details of the parking space dataset used as training and test data

Subset	Free space	Busy space	Total
CNRPark A	2,549	3,622	6,171
CNRPark B	1,632	4,781	6,413
CNRPark	4,181	8,403	12,584
CNRParkEXT Train	46,877	47,616	94,493
CNRParkEXT Train C1 C8	21,769	16,784	38,553
CNRParkEXT Test	13,589	18,276	31,825
CNRParkEXT	65,658	79,307	144,965
PKLot UFPR04	59,718	46,125	105,843
PKLot UFPR05	68,359	97,426	165,785
PKLot PUC	130,040	194,229	324,269

2.2. Proposed convolutional neural network architecture

The CNN architecture was built to detect cars in parking spaces and was named ConciseCarNet. This architecture is called ConciseCarNet because it is the same as its function as a car parking space detector. ConciseCarNet is also more efficient than CarNet [31] in terms of accuracy, many parameters, and FLOPs.

The ConciseCarNet architecture comprises several vital components aimed at enhancing both accuracy and classification speed. Some of the main CNN components include the following:
- The architecture employs 3×3 and 1×1 filter in each convolution to effectively decrease the number of parameters.
- Architecture has two branches with different structures. This branch serves to obtain feature variations to improve CNN accuracy.
- Average Pooling 7×7 is used before fully-connected to reduce the number of parameters, without losing features.

ConciseCarNet uses 3×3 and 1×1 convolution filters to extract spatial features. Researchers always use 3×3 convolution filter in developing CNN architecture. The filter size of 3×3 is better than 5×5 and 7×7 [35]–[37] in most cases. While the 1×1 filter is used to retrieve local features without involving its neighbors. The 1×1 filter is linear weighted or as input projection. This filter is also useful for increasing or decreasing the number of output map features without changing the height and width of the feature. ConciseCarNet uses a 1× 1 convolution filter to balance the output of each branch. This 1×1 filter has been used on network in network (NIN) [38], GoogleNet [12], neural architecture search network (NasNet) [39], and until now such as multiplexing network (MUXNet) [40].

Branches on the ConciseCarNet architecture can be used to capture spatial information at various spatial resolution ranges in order to generate feature variants. The popular existing CNN with branches has succeeded in increasing the accuracy for classifying objects in images such as GoogleNet [12], Resnext [41], squeeze-and-excitation network (SENet) [38], and pathway associated sparse network (PasNet) [42]. The accuracy of CNN architectures with branches is always better than branchless architectures such as VGGNet and Alexnet. The branch outputs are concatenated before being passed to the next layer. In addition, ConciseCarNet also uses the rectified linear unit (ReLU) activation function and batch normalization (BN).

Overfitting causes a decrease in the accuracy of CNN for classification. To avoid overfitting, the researchers used functions after activation functions such as local response normalization (LRN), dropout and BN. LRN was first introduced to Alexnet [10]. LRN has succeeded in improving CNN's performance. After that, "dropout" was introduced in 2014 to avoid overfitting and improve CNN performance [43]. Like dropout, BN was introduced in 2015 to improve CNN accuracy [44]. BN impacted CNN training faster with high accuracy. The ConciseCarNet architecture is shown in Figure 4.

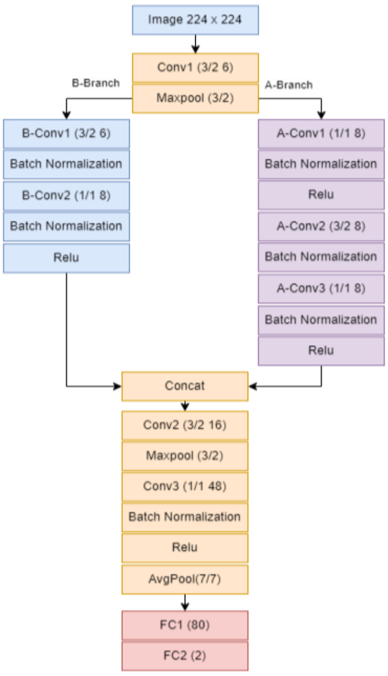

Figure 4. ConciseCarNet architecture

As shown in Figure 4, Conv1 (3/2 6) means that the first convolution layer uses a 3×3 filter with stride 2 and the number of outputs 6. Maxpool (3/2) means MaxPooling with a filter size of 3×3 and the number of strides 2. The ConciseCarNet architecture has three convolution layers, namely conv1, conv2, and conv3. Between conv1 and conv2 there are two branches, namely A-Branch and B-Branch. A-Branch has three convolution layers, namely A-conv1, A-conv2, and A-conv3. Each convolution layer in the A-Branch uses filters 1×1, 3×3, and 1×1. In B-Branch, there are two convolution layers, namely B-conv1 and B-conv2. B-conv1 uses a 1×1 filter and B-conv1 uses a 3×3 filter. The outputs of both branches are concatenated as input conv2. Conv3 uses BN and the ReLU activation function. Conv3 output is pooled using AvgPooling with a filter size of 7×7 and stride 7. Pooling outputs are classified as the first fully connected (FC1) with class 80 outputs. FC1 followed by FC2 with two class outputs according to the desired parking space information.

3. RESULTS AND DISCUSSION

mAlexnet is a mini architecture pruned by Alexnet [8]. mAlexnet has an average accuracy of 91.34% for the PKLot and CNRPark datasets based on different test sets. Training on the PKLot dataset and testing on the CNRPark dataset or vice versa results in low accuracy. Overfitting and underfitting cause low accuracy. Overfitting is when CNN recognizes training data too much so that it can't recognize test data, while underfitting is training data that does not represent all test data [45]. Testing by dividing the dataset as training and test data, mAlexnet produces higher accuracy than training and testing on different datasets. Therefore, in this test, we did not use a different dataset.

We tested the architectural capabilities step by step. In the first stage, we tested ConciseCarNet with the CNRParkEXT dataset. At this stage, we compare the accuracy of ConciseCarNet with mAlexnet. In the second stage, we try to increase the number of outputs in several layers of ConciseCarNet convolution. ConciseCarNet fine-tuning serves to improve CNN accuracy. In the third stage, we use the best ConciseCarNet and compare it against several other methods with all datasets. In the fourth step, we compare the efficiency of each architecture by calculating the number of parameters and FLOPs.

3.1. Comparison of ConciseCarNet and mAlexnet

The CNRParkEXT dataset is the data used to test ConciseCarNet. The resulting accuracy of ConciseCarNet is compared with mAlexnet to see the increase in accuracy obtained. The composition of the training data used is CNRPark, CNRParkEXT TRAIN C1-C8, and CNRParkEXT TRAIN. The test data used is CNRParkEXT TEST. The comparison of the accuracy of ConciseCarNet and mAlexnet is shown in Table 3.

Table 3. Accuracy Comparison between ConciseCarNet and mAlexNet

Train dataset	Test dataset	Methods	
		mAlexnet (%)	ConciseCarNet (%)
CNRPark	CNRParkEXT TEST	93.52	92.93
CNRParkEXT TRAIN C1-C8	CNRParkEXT TEST	95.88	96.93
CNRParkEXT TRAIN	CNRParkEXT TEST	97.70	98.30
Average accuracy		95.70	96.05

Based on Table 3, the accuracy of ConciseCarNet is better than mAlexnet in two tests, namely the CNRParkEXT TRAIN C1-C8 and CNRParkEXT TRAIN training data. Testing on CNRPark training data, mAlexnet's accuracy is better than ConciseCarNet. ConciseCarNet is 0.35% better than mAlexnet on average accuracy. ConciseCarNet's accuracy is better, but it still needs to be more efficient. Therefore, fine-tuning ConciseCarNet is necessary.

3.2. Fine tuning ConciseCarNet

Fine tuning ConciseCarNet is a way to improve the accuracy of parking space classification. We increase the number of filters in A-Branch and B-Branch to 15 filters and the number of filters in conv2 to 30 filters. This fine-tuning certainly increases the number of parameters by almost double the original, so we named this architecture ConciseCarNet 2x. ConciseCarNet 2x architecture is shown in Figure 5. As seen in Figure 5, the addition of the number of filters is expected to produce a large feature variance, to increase accuracy. ConciseCarNet2x does not change the structure of ConciseCarNet. The difference between ConciseCarNet2x and ConciseCarNet only lies in the number of filters used in branch convolution and conv2.

The tests conducted utilized identical data to that of the previous researchers. The training data comprised CNRPark, CNRParkEXT TRAIN C1-C8, and CNRParkEXT TRAIN, while the test data consisted of CNRParkEXT test. ConciseCarNet2x demonstrated superior accuracy compared to all test schemes. ConciseCarNet2x has an average accuracy of 96.82%, 1.12% better than mAlexnet and 0.4% than

EfficientParkingNet. A comparison of the accuracy of ConciseCarNet2x, ConciseCarNet, mAlexnet, and EfficientParkingNet is presented in Table 4.

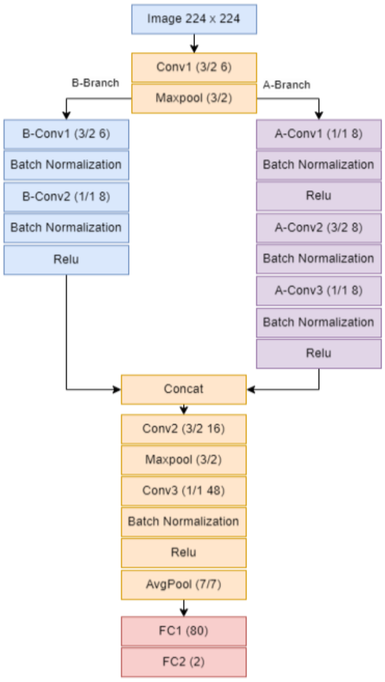

Figure 5. ConciseCarNet 2x architecture

Table 4. Accuracy comparison of ConciseCarNet2x, ConciseCarNet, mAlexNet, and EfficientParkingNet

Train dataset	Methods			
	mAlexnet (%)	EfficientParkingNet (%)	ConciseCarNet (%)	ConciseCarNet2x (%)
CNRPark	93.52	93.97	92.93	94.30
CNRParkEXT TRAIN C1-C8	95.88	96.85	96.93	97.55
CNRParkEXT TRAIN	97.70	98.54	98.30	98.60
Average	95.70	96.42	96.05	96.82

The CNRParkEXT dataset is a dataset taken from nine different cameras, so it has slightly different features. Training with one subset and testing with another subset is not the right solution for CNN implementation, this causes overfitting or underfitting. To overcome this, we use all the data to be able to represent all the cameras. All CNRParkEXT datasets were combined and divided into 80% training data and 20% as validation data. This test is closer to the real implementation of parking.

The test results show that ConciseCarNet2x has better accuracy than some other methods. The validation accuracy rate on ConciseCarNet reaches 99.37%, better than mAlexnet and much better than CarNet. ConciseCarNet matches the accuracy of Yolo+MobileNet which is a well-known existing CNN, even slightly better. A comparison of the accuracy of several methods is shown in Table 5.

Table 5. Validation accuracy results on the CNRParkEXT dataset

Methods	Accuracy (%)
Re-trained mAlexnet	98.13
Yolo+MobileNet [31]	98.97
CarNet [9]	97.24
EfficientParkingNet [21]	98.44
Mini Shufflenet [33]	98.37
ConciseCarNet2x	99.37

3.3. ConciseCarNet2x on PKLot dataset

Testing on the CNRParkEXT dataset shows that ConciseCarNet is better than mAlexnet and EfficientParkingNet. The next test uses the PKLot dataset [30] which has three subsets. Each subset is captured by a camera. The PKLot subsets named UFPBR04, UFPBR05, and PUC are derived from three different cameras from two parking areas. PKLot represents a scenario where one parking space is utilized as the training dataset, while another parking space serves as the testing dataset. The PKLot dataset is easier to identify than the CNRParkEXT dataset. ConciseCarNet2x accuracy is greater than 99.9% for datasets with the same camera source. The accuracy of ConciseCarNet2x is better than the accuracy of mAlexnet and CarNet. The results of the comparison of accuracy are presented in Table 6.

Table 6. Accuracy comparison on the PKLot dataset

Train Dataset	Test dataset	Methods		
		MAlexnet (%)	ConciseCarNet 2x (%)	CarNet (%)
UFPR04	UFPR04	99.54	99.93	95.60
	UFPR05	93.23	97.68	97.60
	PUC	98.27	99.02	98.30
UFPR05	UFPR04	93.69	93.90	95.20
	UFPR05	99.49	99.93	97.50
	PUC	92.72	96.86	98.40
PUC	UFPR04	98.03	98.92	94.40
	UFPR05	96.00	96.07	97.60
	PUC	99.90	99.92	98.80
Average accuracy		96.74	97.27	97.04

As seen in Table 6, testing with 9 schemes, ConciseCarNet2x 6 times and CarNet 3 times being better than all methods. ConciseCarNet2x has better average accuracy than mAlexnet and CarNet. ConciseCarNet2x has an average accuracy of 97.27%, while CarNet is 97.04% and mAlexnet is only 96.74%. ConciseCarNet achieves a maximum accuracy of 99.93%, surpassing larger architectures such as YOLOv5-pretrained [34], which attains 99.50% accuracy. Therefore, ConciseCarNet is better at classifying parking spaces on the PKLot dataset.

3.4. Comparison of number of parameters and floating point operations

The mAlexnet architecture has a total of 32,332 parameters, so the classification process is faster. In contrast to CarNet, which has a very wide architecture, so the parameter size is too large. Carnet has three convolution layers, the first convolution layer uses 96 filters with a size of 11×11 and is followed by max-pooling with 2×2 filters. The second and third convolution layers are the same as the first layer, but the number of filters is 192 in the second convolution layer and 384 in the convolution layer. Third CarNet has three fully connected layers, the first and second layers use 4096 total outputs and the third fully connected layer has two outputs [9].

The CarNet architecture has 64,139,354 parameters. CarNet has very large parameters, causing relatively large file sizes. In addition, the computational process is very large and requires computing devices with adequate specifications to carry out the training and testing process. CarNet has a larger number of parameters than all architectures for parking space classification, even against Yolo+MobileNet. MobileNet parameters only range from 4,000,000 parameters [31]. Compared to other architectures, ConciseCarNet2x has the smallest parameters. ConciseCarNet2x has 14,642 parameters and is smaller than EfficientParkingNet with 21,834 parameters [21]. This is caused by the size of the filter used 3×3 and 1×1. The comparison of the number of parameters is presented in Table 7.

Table 7. Parameter comparison

Methods	Parameters (million)
mAlexnet	0.032
CarNet	64.139
Yolo+MobileNet	4.000
EfficientParkingNet	0.022
ConciseCarNet 2 x	0.015
ConciseCarNet	0.008

Based on Table 7, ConciseCarNet has the fewest parameters. ConciseCarNet2x is only the number of parameters for mAlexnet, 1/273 the number of parameters for Yolo+mobilenet, and 2/3 the number of parameters for EfficientParkingNet. CarNet has the largest number of parameters, it is difficult to implement on low-computing devices such as Raspberry Pi. Based on the number of parameters, ConciseCarNet is better than other architectures to be implemented on low-end computing devices. The following comparison of file sizes and FLOPs for each architecture is presented in Table 8.

Tabel 8. Comparison of file size and FLOPs

Methods	File Size	FLOPs
mAlexnet	133Kb	21,329,778
CarNet	257Mb	1,325,155,234
EfficientParkingNet	90Kb	8,469,776
ConciseCarNet2x	73Kb	6,084,818
ConciseCarNet	45Kb	3,963,916

Based on Table 8, ConciseCarNet has a small file size of 45 KB and ConciseCarNet2x only 73 KB. mAlexnet has a file size of 133 KB, almost 2 times the size of ConciseCarNet2x. Based on the number of FLOPs, ConciseCarNet2x is also smaller than mAlexnet and EfficientParkingNet, which is less than 1/3 the number of FLOPs of mAlexnet and FLOPs of EfficientParkingNet. The fewer number of parameters and FLOPs will speed up the classification time, which is needed in real-time systems. Speed testing by comparing time is rarely done by researchers because the central processing unit (CPU) task when performing computations is unpredictable. Therefore, the tests were alternated and repeated ten times testing for each method. The results of the comparison of the classification speed of a parking space are presented in Table 9.

Table 9. Parking space classification speed comparison

Methods	Classification Time (Second)
mAlexnet	0.00948079
EffecientParkingNet	0.00859164
ConciseCarNet 2 X	0.00771057
ConciseCarNet	0.00661905

A parking area contains many parking spaces that can be classified simultaneously. Camera A in the CNRPark dataset is capable of monitoring 51 parking spaces at once. The faster the time for classifying a parking area, the faster the information will be received by the driver. EffecientCarNet is the fastest architecture for classifying parking spaces in parking areas. The classification times for camera A in the CNRPark dataset are presented in Table 10.

Table 10. Parking area classification time

Methods	Clasification Time (Second)	Frames/Second
EffecientParkingNet	0.438174	2.282200
ConciseCarNet 2 X	0.393239	2.542982
ConciseCarNet	0.337572	2.962334

Based on Table 10, it is clear that the classification time for a single parking space using mAlexnet is 0.48 seconds, while ConciseCarNet obtains a faster time of 0.337 seconds. In terms of processing speed, mAlexnet can categorize PKLot at a rate of 2 picture frames per second, but ConciseCarNet outperforms at 2.9 frames per second. Our study not only demonstrates the improved accuracy of the new architecture but also its better speed compared to other existing architectures.

4. CONCLUSION

This study found a new CNN architecture dubbed ConciseCarNet for identifying parking spaces, which addresses the difficulty of parking space availability in urban areas. ConciseCarNet uses small filters, namely 3×3 and 1×1 with a small number of filters, this causes the number of parameters to be less. ConciseCarNet has branches with different structures, after the first convolution layer. This branch is intended to generate more feature variance, thereby contributing to increased accuracy. Based on the test, the accuracy of ConciseCarNet2x outperforms the accuracy of mAlexnet, EfficientParkingNet, and Carnet architectures. The accuracy of ConciseCarNet2x is 99.37% on the CNRParkEXT dataset and an average of 97.27% for the PKLot dataset, even reaching 99.93% in the same subsets on the PKLot dataset. In addition, ConciseCarNet also has fewer parameters, file size, and FLOPs compared to other architectures. Therefore, ConciseCarNet is faster and can be implemented for parking space classification with low-computing devices, such as the Raspberry Pi. This will save costs in purchasing computing tools. The weakness of this research is the manual determination of the coordinates for each parking space. This technology is expected to be able to detect parking space coordinates independently in the future, thereby eliminating the need for manual labeling.

REFERENCES

[1] I. Sukarno, H. Matsumoto, and L. Susanti, "Transportation energy consumption and emissions - a view from city of Indonesia," *Future Cities and Environment*, vol. 2, no. 1, Jun. 2016, doi: 10.1186/s40984-016-0019-x.

[2] S. Uttara, N. Bhuvandas, and V. Aggarwal, "Impacts of urbanization on environment," *International Journal of Research in Engineering and Applied Sciences*, vol. 2, no. 2, pp. 1637–1645, 2012.

[3] T. Lin, H. Rivano, and F. Le Mouel, "A survey of smart parking solutions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 12, pp. 3229–3253, Dec. 2017, doi: 10.1109/TITS.2017.2685143.

[4] Y.-Y. Chiang and C. A. Knoblock, "A general approach for extracting road vector data from raster maps," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 16, no. 1, pp. 55–81, Mar. 2013, doi: 10.1007/s10032-011-0177-1.

[5] R. Yusnita, F. Norbaya, and N. Basharuddin, "Intelligent parking space detection system based on image processing," *International Journal of Innovation, Management and Technology*, vol. 3, no. 3, pp. 232–235, 2012.

[6] H. Al-Kharusi and I. Al-Bahadly, "Intelligent parking management system based on image processing," *World Journal of Engineering and Technology*, vol. 2, no. 2, pp. 55–67, 2014, doi: 10.4236/wjet.2014.22006.

[7] I. Masmoudi and A. Wali, "Vision based approach for adaptive parking lots occupancy estimation," *Pattern Recognition and Image Analysis*, vol. 29, no. 3, pp. 515–522, Jul. 2019, doi: 10.1134/S1054661819030143.

[8] G. Amato, F. Carrara, F. Falchi, C. Gennaro, C. Meghini, and C. Vairo, "Deep learning for decentralized parking lot occupancy detection," *Expert Systems with Applications*, vol. 72, pp. 327–334, Apr. 2017, doi: 10.1016/j.eswa.2016.10.055.

[9] S. Nurullayev and S.-W. Lee, "Generalized parking occupancy analysis based on dilated convolutional neural network," *Sensors*, vol. 19, no. 2, Jan. 2019, doi: 10.3390/s19020277.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd International Conference on Learning Representations (ICLR 2015), Computational and Biological Learning Society*, 2015, pp. 1–14.

[12] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2015, pp. 1–9, doi: 10.1109/CVPR.2015.7298594.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.

[14] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[15] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: practical guidelines for efficient CNN architecture design," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 122–138.

[16] M. Klaiber, D. Sauter, H. Baumgartl, and R. Buettner, "A systematic literature review on transfer learning for 3D-CNNs," in *2021 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2021, pp. 1–10, doi: 10.1109/IJCNN52387.2021.9533302.

[17] Z. Zhou, J. Shin, L. Zhang, S. Gurudu, M. Gotway, and J. Liang, "Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, pp. 4761–4772, doi: 10.1109/CVPR.2017.506.

[18] A. Alqahtani, X. Xie, and M. W. Jones, "Literature review of deep network compression," *Informatics*, vol. 8, no. 4, Nov. 2021, doi: 10.3390/informatics8040077.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[20] S. Rahman, M. Ramli, F. Arnia, R. Muharar, and A. Sembiring, "Performance analysis of mAlexnet by training option and activation function tuning on parking images," *IOP Conference Series: Materials Science and Engineering*, vol. 1087, no. 1, Feb. 2021, doi: 10.1088/1757-899X/1087/1/012084.

[21] S. Rahman, M. Ramli, F. Arnia, R. Muharar, M. Ikhwan, and S. Munzir, "Enhancement of convolutional neural network for urban environment parking space classification," *Global Journal of Environmental Science and Management*, vol. 8, no. 3, pp. 315–326, 2022, doi: https://doi.org/10.22034/gjesm.2022.03.02.

[22] P. P. B. D. Gusmao, G. Francini, S. Lepsøy, and E. Magli, "Fast training of convolutional neural networks via kernel rescaling," *arXiv-Computer Science*, pp. 1-13, 2016.

[23] M. J. J. P. V. Grinsven, B. V. Ginneken, C. B. Hoyng, T. Theelen, and C. I. Sanchez, "Fast convolutional neural network training using selective data sampling: application to hemorrhage detection in color fundus images," *IEEE Transactions on Medical Imaging*, vol. 35, no. 5, pp. 1273–1284, May 2016, doi: 10.1109/TMI.2016.2526689.

[24] J. Yamanaka, S. Kuwashima, and T. Kurita, "Fast and accurate image super resolution by deep CNN with skip connection and network in network," in *International Conference on Neural Information Processing*, Springer, 2017, pp. 217–225, doi: 10.1007/978-3-319-70096-0_23.

[25] S. D. Khan and H. Ullah, "A survey of advances in vision-based vehicle re-identification," *Computer Vision and Image Understanding*, vol. 182, pp. 50–63, May 2019, doi: 10.1016/j.cviu.2019.03.001.

[26] N. True, "Vacant parking space detection in static images," *University of California, San Diego*, vol. 17, pp. 659–662, 2007.

[27] Q. Wu, C. Huang, S. Wang, W. Chiu, and T. Chen, "Robust parking space detection considering inter-space correlation," in *Multimedia and Expo, 2007 IEEE International Conference on*, Jul. 2007, pp. 659–662, doi: 10.1109/ICME.2007.4284736.

[28] J. K. Suhr, H. G. Jung, K. Bae, and J. Kim, "Automatic free parking space detection by using motion stereo-based 3D reconstruction," *Machine Vision and Applications*, vol. 21, no. 2, pp. 163–176, Feb. 2010, doi: 10.1007/s00138-008-0156-9.

[29] N. Bibi, M. N. Majid, H. Dawood, and P. Guo, "Automatic parking space detection system," in *2017 2nd International Conference on Multimedia and Image Processing (ICMIP)*, Mar. 2017, pp. 11–15, doi: 10.1109/ICMIP.2017.4.

[30] P. R. L. D. Almeida, L. S. Oliveira, A. S. Britto, E. J. Silva, and A. L. Koerich, "PKLot – A robust dataset for parking lot classification," *Expert Systems with Applications*, vol. 42, no. 11, pp. 4937–4949, Jul. 2015, doi: 10.1016/j.eswa.2015.02.009.

[31] L.-C. Chen, R.-K. Sheu, W.-Y. Peng, J.-H. Wu, and C.-H. Tseng, "Video-based parking occupancy detection for smart control system," *Applied Sciences*, vol. 10, no. 3, Feb. 2020, doi: 10.3390/app10031079.

[32] T. Connie, M. K. O. Goh, V. C. Koo, K. T. Murata, and S. P. Amnuaisuk, "Improved parking space recognition via grassmannian deep stacking network with illumination correction," in *International Conference on Computational Intelligence in Information System*, Springer, 2021, pp. 150–159, doi: 10.1007/978-3-030-68133-3_15.

[33] S. Rahman *et al.*, "Mini shufflenet for efficient parking space classification," in *AIP Conference Proceedings*, 2023, vol. 2480, no. 1, doi: 10.1063/5.0103430.

[34] S. Rafique, S. Gul, K. Jan, and G. M. Khan, "Optimized real-time parking management framework using deep learning," *Expert*

Systems with Applications, vol. 220, Jun. 2023, doi: 10.1016/j.eswa.2023.119686.

[35] X. Lian, Z. Liu, Z. Song, J. Dai, W. Zhou, and X. Ji, "High-Performance FPGA-Based CNN accelerator with block-floating-point arithmetic," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 8, pp. 1874–1885, Aug. 2019, doi: 10.1109/TVLSI.2019.2913958.

[36] Z. Zhou, W. Zhou, H. Li, and R. Hong, "Online filter clustering and pruning for efficient convnets," in *2018 25th IEEE International Conference on Image Processing (ICIP)*, Oct. 2018, pp. 11–15, doi: 10.1109/ICIP.2018.8451123.

[37] P. Singh, V. K. Verma, P. Rai, and V. P. Namboodiri, "Leveraging filter correlations for deep model compression," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Mar. 2020, pp. 824–833, doi: 10.1109/WACV45572.2020.9093331.

[38] M. Gerla, E. Leonardi, F. Neri, and P. Palnati, "Routing in the bidirectional shufflenet," *IEEE/ACM Transactions on Networking*, vol. 9, no. 1, pp. 91–103, 2001, doi: 10.1109/90.909027.

[39] A. M. Alayba, V. Palade, M. England, and R. Iqbal, "A combined CNN and LSTM model for Arabic sentiment analysis," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*, Springer, 2018, pp. 179–191, doi: 10.1007/978-3-319-99740-7_12.

[40] S. Mostafa and F.-X. Wu, "Diagnosis of autism spectrum disorder with convolutional autoencoder and structural MRI images," in *Neural Engineering Techniques for Autism Spectrum Disorder*, Elsevier, 2021, pp. 23–38, doi: 10.1016/B978-0-12-822822-7.00003-X.

[41] P. Aswathy, Siddhartha, and D. Mishra, "Deep GoogLeNet features for visual object tracking," in *2018 IEEE 13th International Conference on Industrial and Information Systems (ICIIS)*, Dec. 2018, pp. 60–66, doi: 10.1109/ICIINFS.2018.8721317.

[42] A. Varghese and G. Sreelekha, "An efficient algorithm for detection of vacant spaces in delimited and non-delimited parking lots," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 10, pp. 4052–4062, Oct. 2020, doi: 10.1109/TITS.2019.2934574.

[43] B. Zoph, V. Vasudevan, J. Shlens, and Q. V Le, "Learning transferable architectures for scalable image recognition," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 8697–8710, doi: 10.1109/CVPR.2018.00907.

[44] Z. Lu, K. Deb, and V. Naresh Boddeti, "MUXConv: information multiplexing in convolutional neural networks," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 12041–12050, doi: 10.1109/CVPR42600.2020.01206.

[45] A. D. Gavrilov, A. Jordache, M. Vasdani, and J. Deng, "Preventing model overfitting and underfitting in convolutional neural networks," *International Journal of Software Science and Computational Intelligence*, vol. 10, no. 4, pp. 19–28, Oct. 2018, doi: 10.4018/IJSSCI.2018100102.

BIOGRAPHIES OF AUTHORS

Prof. Marwan Ramli is a Professor of Applied Mathematics and an active professor and researcher at Department of Mathematics and Mathematics Graduated Study Program, Universitas Syiah Kuala. He was born in Tanjung Pinang, Kepulauan Riau, on November 25, 1971. He received his Ph.D. in Applied Mathematics from the Bandung Institute of Technology, specializing on extreme water waves generation. In 1999, he received his Master's degree in Mathematics from the Bandung Institute of Technology in Indonesia. In 1996, he received his B.Sc. in Mathematics from Syiah Kuala University in Indonesia. He can be contacted at email: marwan.math@unsyiah.ac.id.

Dr. Sayuti Rahman is lecturer at the University of Medan Area, has recently earned his Ph.D. degree from Syiah Kuala University in Banda Aceh. Throughout his teaching career. He has demonstrated a strong interest in exploring the domains of digital imagery and machine learning. As a result, his current focus revolves around research related to extracting information from digital images, particularly in the realm of computer vision. Python serves as the primary programming language extensively utilized in his research pursuits. Currently, he is actively involved in a research project dedicated to the development of an intelligent parking system, with a specific emphasis on optimizing convolutional neural networks for effective parking space classification. He was born on June 18, 1987, in Bukit Gantung, situated in the Wampu District of Langkat Regency. He can be contacted at email: sayutirahman@staff.uma.ac.id.

Dr. Rahmad Bayu Syah is an Associate Professor in Universitas Medan Area and received head of Excellent Centre Innovations and New Science (PUIN) Universitas Medan Area. His research interests are modeling and computing, artificial intelligence, data science, business intelligent, metaheuristics hybrid algorithm and computational intelligent. He is a member of the IEEE, institute for system and technologies of information, control, and communication, International Association of Engineers (IAENG). He can be contacted at email: rahmadsyah@uma.ac.id.