Global J. Environ. Sci. Manage. 8(3): 315-326, Summer 2022, Serial #31

Global Journal of Environmental Science and Management
(GJESM)

Homepage: https://www.gjesm.net/

ORIGINAL RESEARCH ARTICAL

Enhancement of convolutional neural network for urban environment parking space classification

S. Rahman[1], M. Ramli[2,*], F. Arnia[3], R. Muharar[3], M. Ikhwan[4], S. Munzir[2]

[1] School of Engineering, Universitas Syiah Kuala, Banda Aceh 23111, Indonesia

[2] Department of Mathematics, Universitas Syiah Kuala, Banda Aceh 23111, Indonesia

[3] Department of Electrical Engineering, Universitas Syiah Kuala, Banda Aceh 23111, Indonesia

[4] Graduate School of Mathematics and Applied Sciences, Universitas Syiah Kuala, Banda Aceh 23111, Indonesia

ARTICLE INFO

Article History:
Received 29 May 2021
Revised 17 August 2021
Accepted 01 October 2021

Keywords:
Deep learning
Efficient Parking Network
mAlexnet
MobileNet
Parking space

ABSTRACT

BACKGROUND AND OBJECTIVES: The increase in the number of vehicles has several negative impacts, including traffic congestion, air pollution, noise levels, and the availability of parking spaces. Drivers looking for parking spaces can cause traffic jams and air pollution. The solution offered at this time is the development of a smart parking system to overcome these problems. The smart parking system offers a parking availability information feature in a parking area to break up congestion in the parking space. Deep learning is a successful method to solve parking space classification problems. It is known that this method requires a large computational process. Th aims of this study are to modified the architecture of Convolutional Neural Networks, part of deep learning to classify parking spaces. Modification of the Convolutional Neural Networks architecture is assumed to increase the work efficiency of the smart parking system in processing parking availability information.
METHODS: Research is focusing on developing parking space classification techniques using camera sensors due to the rapid advancement of technology and algorithms in computer vision. The input image has 3x3 dimensions. The first convolution layer accepts the input image and converts it into 56x56 dimensions. The second convolution layer is composed in the same way as the first layer with dimensions of 25x25. The third convolution layer employs a 3 x 3 filter matrix with padding of up to 15 and converts it into 10x10 dimensions. The fourth layer is composed in the same way as the third layer, but with the addition of maximum pooling. The software used in the test is Python with a Python framework.
FINDINGS: The proposed architecture is the Efficient Parking Network or EfficientParkingNet. It can be shown that this architecture is more efficient in classifying parking spaces compared to some other architectures, such as the mini–Alex Network (mAlexnet) and the Grassmannian Deep Stacking Network with Illumination Correction (GDSN-IC). EfficientParkingNet has not been able to pass the accuracy of Yolo Mobile Network (Yolo+MobileNet). Furthermore, Yolo+MobileNet has so many parameters that it cannot be used on low computing devices. Selection of EfficientParkingNet as a lightweight architecture tailored to the needs of use. EfficientParkingNet's lightweight computing architecture can increase the speed of information on parking availability to users.
CONCLUSION: EfficientParkingNet is more efficient in determining the availability of parking spaces compared to mAlexnet, but still cannot match Yolo+MobileNet. Based on the number of parameters, EfficientParkingNet uses half of the number of parameters of mAlexnet and is much smaller than Yolo+MobileNet. EfficientParkingNet has an accuracy rate of 98.44% for the National Research Council parking dataset and higher than other architectures. EfficientParkingNet is suitable for use in parking systems with low computing devices such as the Raspberry Pi because of the small number of parameters.

DOI: 10.22034/gjesm.2022.03.02

NUMBER OF REFERENCES	NUMBER OF FIGURES	NUMBER OF TABLES
38	3	10

*Corresponding Author:
Email: *marwan.math@unsyiah.ac.id*
Phone: +62813 9766 8376
ORCID: 0000-0003-1225-9063

Note: Discussion period for this manuscript open until October 1, 2022 on GJESM website at the "Show Article.

INTRODUCTION

Environmental problems often occur in big cities due to urbanization, such as land insecurity, worsening water quality, excessive air pollution, noise, and the problems of waste disposal (Uttara *et al.*, 2012). One of the factors causing this problem is vehicles. The increasing number of vehicles can cause air pollution, noise, lack of parking areas, traffic jams, and increased fuel use. Vehicle exhaust is the main source of anthropogenic carbon dioxide (CO_2) in the metropolis that is dangerous to public health (Uttara *et al.*, 2012). Emissions and the use of vehicle fuels have also increased (Sukarno *et al.*, 2016). Urban planning used to solve various problems in big cities by integrating information technology and communication (ICT) and the internet of things (IoT) technology is called smart cities. Smart parking systems are part of a smart city. Smart parking systems can help drivers avoid traffic jams, use less efficient fuel, avoid vehicle queues, panic, and minimize pollution. Bura *et al.* (2018) stated that 30% of traffic jams are caused by drivers searching for a parking area. Smart parking systems can give direction to vehicles to parking spaces based on the type of vehicle, the appropriate rate, and the available space by selecting the closest route in a short time (Ikhwan *et al.*, 2017). Based on the technology, smart parking systems have been developed based on agent-based, fuzzy systems, wireless sensors, global positioning systems (GPS) and computer-based (Mahmud *et al.*, 2013). Much research has been developed to improve the reliability of smart parking systems. Some studies related to smart parking systems includes detection of parking space availability, parking meters, integrated vehicle development, simulation and analysis of parking data, and competition in booking available parking spaces. Most implementations of smart parking are focused on sensing technology and mobile application development (Lin *et al.*, 2017). One smart camera can monitor multiple parking spaces simultaneously at a lower cost compared to the cost of installing and maintaining sensors in each parking space (Amato *et al.*, 2017). Detection of parking spaces' availability is an important part of a smart parking system. It can provide information to drivers in parking spaces about availability for vehicle parking. Detection of parking spaces by using computer vision has been widely developed, including the detection of parking spaces by using image subtraction (Chiang and Knoblock, 2013), marking in the parking area (Rahman *et al.*, 2020a; Zhang *et al.*, 2018), adaptive approach to space constraint with cube (Masmoudi and Wali, 2019), using feature recognition algorithms such as deep learning (Amato *et al.*, 2017) and many others. Deep learning is very effective in solving computer vision problems. Convolutional Neural Network (CNN) is a part of deep learning. It is often used to classify objects in an image (Krizhevsky *et al.*, 2012). Some researchers have built CNN models to classify objects in images. CNN model is also known as the pre-trained CNN. Some well-known and widely developed pre-trained CNN such as Alexnet (Krizhevsky *et al.*, 2012), VGGNet (Simonyan and Zisserman, 2014), GoogleNet (Szegedy *et al.*, 2015), Resnet (He *et al.*, 2016), MobileNetV2 (Sandler *et al.*, 2018), and ShuffleNet (Ma *et al.*, 2018), managed to classify objects well. Pre-trained CNN is designed to classify 1000 objects in an image dataset with high computational and large storage space. For fewer class cases, the researchers exploit CNN, such as transfer learning (Hussain *et al.*, 2018), fine-tuning or pruning (Li *et al.*, 2016) towards pre-trained CNNs or creating a new CNN architecture. The first CNN models developed specifically to identify parking space availability were mAlexnet and mLenet (Amato *et al.*, 2016; 2017). mAlexnet and mLenet are named based on their names, namely Alexnet (Krizhevsky *et al.*, 2012) and Lenet (LeCun *et al.*, 1998), which are the prunings of the two architectures. mAlexnet has a greater level of accuracy than 90% in classifying PKLot and CNRPark+EXT datasets, and this is better than mLenet. It is known that mAlexnet and Alexnet have the same level of accuracy in several experiments, with a difference of accuracy of 1% (Amato *et al.*, 2017). However, if it is measured from the average level of accuracy, mAlexnet is still lower than Alexnet. Increasing the size of the kernel at the convolutional layer of mAlexnet (Rahman *et al.*, 2020b) and the activation function of ReLu to LeakyReLu can improve the accuracy of mAlexnet (Rahman *et al.*, 2021) on CNRPark A and CNRPark B data. By increasing the kernel size, the number of parameters used will increase. An increasing number of parameters can cause a time increase in training and testing. In this case, CNRPark A and CNRPark B are sub-datasets of CNRPark+EXT. Therefore, an efficient CNN model needs to be developed to increase the accuracy and speed of parking space classification. These quantities are important parts of the smart parking system. In terms of speed, mAlexnet is better than Alexnet in the

Table 1: The Summary of vision-based methods for parking space classification

Methods	Dataset	Accuracy	References
mAlexnet	PKLot and CNRPark	>90%	Amato *et al*., 2017
Background subtraction and SVM	PKLot and CNRPark	Avg Acuracy 96.53%	Varghese and Sreelekha, 2019
Yolo and MobileNet	CNRPark+EXT	99%	Chen *et al*., 2020
GDSN-IC	CNRPark+EXT	97.10%	Connie *et al*., 2021

training and testing process. CNN architecture training is a large computational process and can take a long time (de Gusmao *et al*., 2016). As a result, some researchers focused on speeding up CNN's training time (van Grinsven *et al*., 2016; Yamanaka *et al*., 2017; Zhang *et al*., 2019). Meanwhile, the speed of testing is needed to support real-time information systems. mAlexnet has a speed of 15 seconds in detecting parking spaces on a real-time camera using a Raspberry Pi 3b. The number of parameters of mAlexnet is approximately 1/1340 of the total Alexnet parameters, so it can be applied to low computing devices such as the Raspberry Pi (Amato *et al*., 2017). The CNN architecture proposed in this study is expected to be excellent in accuracy and speed compared to other architectures that have been developed in previous research. This architecture is built by creating a new, better architecture in terms of accuracy and speed. The dataset used for CNN testing is CNRPark-EXT taken from 10 different cameras. The study aims to modify the architecture of Convolutional Neural Networks, part of deep learning, to classify parking spaces. Modification of the Convolutional Neural Networks architecture is assumed to increase the work efficiency of the smart parking system in processing parking availability information. This study is simulated at the Modeling and Simulation Laboratory, Department of Mathematics, Syiah Kuala University, Indonesia in 2021.

MATERIALS AND METHODS

Many researchers are interested in developing a parking system that can be managed by using a smart system. Detection of parking space generally uses two technologies, namely sensor-based and vision-based. The development of technology and algorithms in computer vision has made researchers focus on developing parking space classification techniques using camera sensors. Researchers consider the use of computer vision to be more efficient than using sensors. A camera can monitor multiple parking spaces simultaneously. Computer vision-based parking space detection has been developed using the use of color histograms and Harris corner detection (True, 2007), 3D parking lot models (Huang *et al*., 2013), and the use of background subtraction with a combination of several classification techniques (del Postigo *et al*., 2015; Màrmol and Sevillano, 2016), deep learning, and more. Deep learning becomes one of the solutions used to solve classification problems in computer vision. mAlexNet was first used to classify parking spaces by using CNN (Amato *et al*., 2017). mAlexnet outperforms Support Vector Machine (SVM) accuracy-wise (de Almeida *et al*., 2015) on PKLot dataset. mAlexnet is designed for low computing devices by simplifying the Alexnet architecture. mAlexnet has 45,000 parameters, which is about 1/1340 parameters of Alexnet, so it can be run on a Raspberry Pi. However, the accuracy level of mAlexnet's for classifying parking spaces needs to be improved. Recently, several methods have been developed by researchers to improve the classification accuracy of mAlexnet. These methods include R-CNN (Sairam *et al*., 2020), Bag of Features (Varghese and Sreelekha, 2019), Yolo+MobileNet (Chen *et al*., 2020), and GDSN-IC (Connie *et al*., 2021), as presented in Table 1.

Table 1 shows that Yolo+mobilnet succeeded in increasing the accuracy level in parking space classification to 99%. However, the number of parameters used is around 4 million parameters. This is higher than mAlexnet, which only uses approximately 45,000 parameters. The more parameters used, the greater the computational process gets. As a result, it takes time to classify parking spaces. Parking space classification requires high speed to provide real-time information to parking users. Fine-tuning is a way to increase the accuracy of mAlexnet. Fine-tuning mAlexnet by changing the filter size in the first layer has succeeded in increasing the level of accuracy (Rahman *et al*., 2020b). Furthermore, changing the ReLu activation function to LeakyReLu at each convolution layer can also increase the accuracy level (Rahman *et al*., 2021). Increasing the accuracy level by fine-tuning mAlexnet has not been optimal. Therefore, an architecture with a high level of classification accuracy

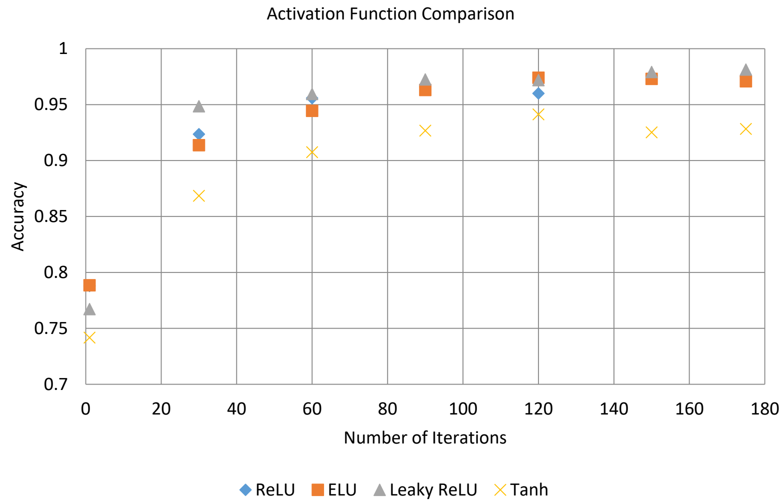

Fig. 1: Comparison of the activation function

and a small number of parameters to produce an efficient architecture is necessary. This study discusses the architecture in question by combining CNN parameters such as convolution layer, kernel, padding, stride, activation function, and fully-connected. Setting CNN parameters in the right order gives good results (Skourt *et al*., 2021).

CNN architecture for parking space classification

The characteristics of an efficient CNN architecture are balanced convolution on each channel, takes the cost of convolution groups into account, reduces the degree of fragmentation, and reduces element-based operations (Ma *et al*., 2018). The CNN architecture for parking space classification developed here is called EfficientParkingNet. This architecture is the development of mAlexnet, with added convolution and LeakyReLu used as an activation function. Similar to mAlexnet, EfficientParkingNet does not have branches, it only consists of one channel. In addition, EfficientParkingNet also has a small degree of fragmentation so that the processing cost of each convolution layer is quite low. The use of LeakyReLu as an activation function on mAlexnet can increase the accuracy rate by 0.57% in the CNRPark B subdataset (Rahman *et al*., 2021). CNRPark B is a small part of the CNRPark+EXT dataset (Amato *et al*., 2017). The results of the comparison of the activation function on mAlexnet are presented in Fig. 1. LeakyRelu is a better activation function compared to other activation functions as shown in Fig. 1. Based on the number of iterations, Leaky Relu managed to get high accuracy. Leaky Relu succeeded in increasing the accuracy of mAlexnet starting from the 30th iteration, and the value continued to increase to 97.70%.

EfficientParkingNet uses LeakyRelu on each convolution layer. Filter size on each convolution layer uses a filter matrix of 3 x 3 to reduce the number of parameters used. EfficientParkingNet has four convolution layers. This layer is deeper than mAlexnet, which consists of only three layers. The EfficientParkingNet architecture is presented in Fig. 2.

The EfficientParkingNet architecture has four convolution layers as shown in Fig. 2. The input image with a size of 224 x 224 is entered in the first convolution layer. In this case, the first convolution layer uses a filter matrix of 3 x 3, padding 4, and the number of filters 30, followed by the activation functions LeakyRelu, Local Response Normalization (LRN), and Maximum Pooling (MaxPooling) with a filter matrix with padding 2. The second convolution layer has the same composition as the first layer. The third convolution layer uses a filter matrix 3 x 3 with padding of 2 as much as 15, followed by the LeakyRelu activation function, and LRN. The fourth layer has the same composition as the third layer with added MaxPooling. The results of the fourth layer are sent to fully-connected by the platten process. The fully connected (FC1) output has 48 classes with a LeakyReLu activation function and the second fully connected (FC2) has a class binary

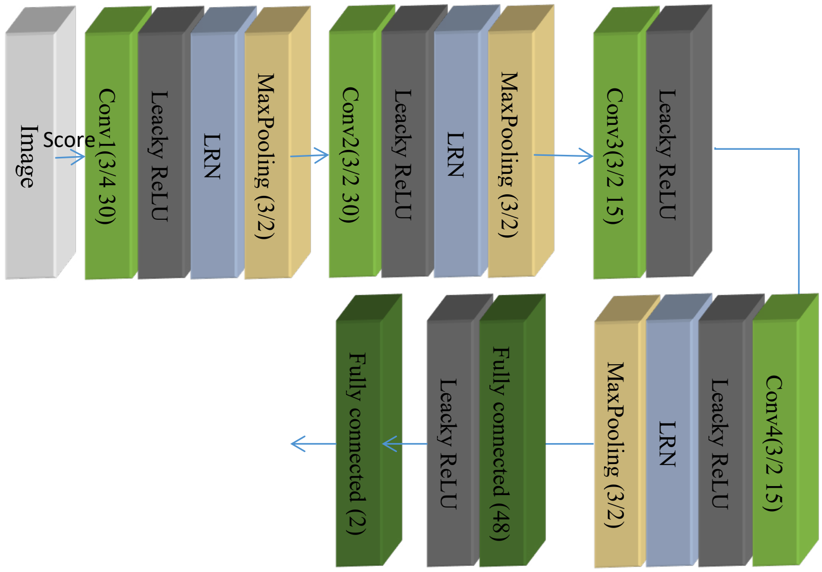

Fig. 2: EfficientParkingNet architecture

output and a softmax activation function. The number of parameters in the convolution layer is determined using Eq. 1.

$$P\_C = \left((F_{Size} - N) + 1\right) \times N\_F, \quad (1)$$

P_C, $F_{\_Size}$, N and N_F states the number of parameters in convolution layer, the filter size used, the number of channels in the previous layer and the number of filters. To determine the number of parameters in the Fully Connected layer, using Eq. 2.

$$P_{Fc} = Cl \times Pl + 1 \times Cl, \quad (2)$$

Where, $P_{\_Fc}$ is the number of parameters in Fully Connected layer, *Cl* is the number of output Current Layer, and *Pl* is the number of output Previous Layer. Next, Activation Shape calculated using Eq. 3.

$$O_{width} = \frac{W-F+2*P}{S} + 1, \quad (3)$$

Where, $O_{\_width}$, W, F, p and s represents output width, tensor size, filtersize, number of padding and stride value.

Dataset and tools

The EfficientParkingNet architecture was developed to increase the accuracy of parking space classification in the CNRPark+EXT dataset. CNRPark+EXT consists of CNRPark taken from the parking area of CNR Research in the City of Pisa, Italy. CNRPark consists of 242 full images and 12,584 image spaces captured with two cameras, A and B. CNR-EXT consists of 4,081 full images and 144,965 image spaces captured with nine cameras. CNRPark was taken in sunny weather conditions, while CNR-EXT was taken in three weather conditions, namely sunny, rainy and cloudy. In this case, the entire CNRPark+EXT dataset is cut off and grouped into two groups, namely free and busy. Busy contains images of a parking space containing vehicles, and free contains images of an empty parking space. The various image sizes were converted to a size of 224 x 224. These images are the input for the Convolutional neural network to be classified. Testing was carried out on mAlexnet using the CNRPark+EXT and PKLot datasets. mAlexnet succeeded in classifying the PKLot dataset with an accuracy rate of 99% with an average accuracy rate of 96.77%, for the CNRPark+EXT dataset the accuracy rate reached 97.71%, and the average accuracy rate was 95.70% (Amato *et al.*, 2017). With the CNRPark+EXT dataset has bigger resistance than PKLot in parking space classification. The CNRPark+EXT dataset, mAlexnet has succeeded in detecting parking spaces with a low level of accuracy. Classification of CNRPark+EXT dataset becomes the focus of this study. Therefore, EfficientParkingNet is focused on increasing

Table 2: Dataset details used as training and test data

Subset	Empty slots	Filled slots	Total
CNRPark A	2549	3622	6171
CNRPark B	1632	4781	6413
CNRPark	4181	8403	12584
CNRPark-EXT Train	46877	47616	94493
CNRPark-EXT Train C1 C8	21769	16784	38553
CNRPark-EXT Test	13589	18276	31825
CNRPark+EXT	65658	79307	144965

the accuracy level for tCNRPark+EXT dataset with fewer parameters than mAlexnet. The dataset sharing scheme for training and testing is the same as mAlexnet. In this case the test is added by dividing the CNRPark+EXT data into 2 (two) parts, namely 80% for training and 20% for randomly selected tests. Details of the data used are presented in Table 2. Testing is done by using a computer with the following specifications: 4 GigaByte RAM, Processor Intel(R) core i3-2120 @3.30GHz 3.30GHz, 64-bit operating system. The software used in the test is python with a Python framework.

RESULTS AND DISCUSSION

The CNRPark+EXT dataset as test data is used to compare the performance of EfficientParkingNet to mAlexnet. The tests were carried out based on mAlexnet test scheme. In this case, 3 (three) test schemes were carried out, namely CNRPark, CNRPark-EXT Train C1-C8, and CNRPark+EXT Train as training data. Each test schema uses CNRPark_EXT TEST as test data. The test results are presented in Table 3.

Based on Table 3, it can be seen that EfficientParkingNet's accuracy level is better than each test scheme. EfficientParkingNet is better in the CNRPark dataset by 0.35%,CNRPark+EXT TRAIN C1-C8 excels by 0.97%, and CNRPark+EXT TRAIN is the best by 0.84%. The average accuracy rate of mAlexnet is 95.70% and EfficientParkingNet is 96.42%. EfficientParkingNet's accuracy level is higher with an average of 0.72% compared to mAlexnet's. The low accuracy level is caused by differences between training and the test data, called overfitting or the training data, it does not represent all the underfitting test data (Gavrilov *et al*., 2018). The first test with CNRPark training data and CNRPark-EXT TEST test data shows a low accuracy rate of 93.87%. It is because the training data does not represent the features in the test data. CNRPark was taken from two cameras, namely A and B in sunny weather while CNRPark-EXT TEST was taken from 10 different cameras which were collected randomly under various weather conditions. The difference in weather, location and less training data compared to test data cause overfitting and underfitting. Overfitting and underfitting cause a lower level of validation accuracy in training. This happened in the first test scheme. The comparison results of the training accuracy level and validation accuracy level are presented in Fig. 3.

Based on Fig. 3, it can be seen that there is a difference in accuracy level between training and validation. EffecientParkingNet can study the training dataset, but it cannot recognize the dataset in the test. It is because the difference in features between the training and the test data or the training data does not represent the test data. Hence, designing the dataset is needed to represent all the locations and cameras contained in the CNRPark+EXT dataset. In the implementation of a parking area, an application must be able to classify parking spaces simultaneously. Splitting datasets and training sub-datasets to recognize other sub-datasets is not a solution at the same time. In this study, all the CNRPark+EXT datasets are used and divided randomly. They are 80% for training data and 20% for validation test data. The use of all data is expected to represent the actual state of the parking system in real-time. The test results for the dataset are presented in Table 4.

Based on Table 3, it can be seen that mAlexnet has a better ability to recognize the CNRPark+EXT train dataset compared to other datasets. Furthermore, based on Table 4, the accuracy rate of mAlexnet reaches 98.13% in classifying the CNRPark+EXT dataset. EfficientParkingNet has an accuracy rate of 98.44% or 0.31%. It is better than mAlexnet. Some tests show that EffecientParkingNet is better than mAlexnet. Besides mAlexnet, the results of EfficientParkingNet are also compared with previous studies based on the data used. The results of the comparison of several

Tabel 3: Comparison of mAlexnet accuracy and EfficientParkingNet

Data train	Accuracy rate	
	mAlexnet	EfficientParkingNet
CNRPark	93.52%	93.87%
CNRPark +EXT TRAIN C1-C8	95.88%	96.85%
CNRPark +EXT TRAIN	97.70%	98.54%
Average	95.70%	96.42%

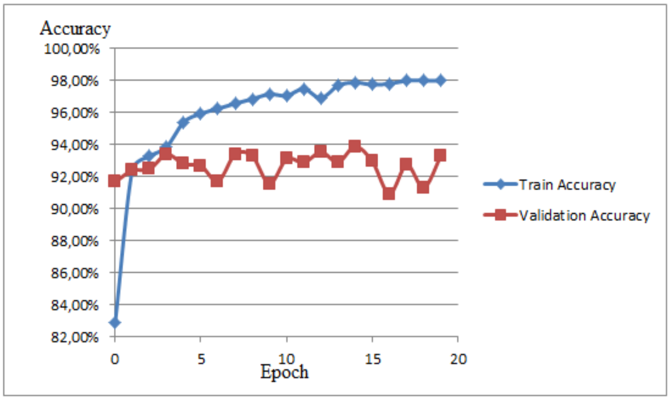

Fig. 3: Training accuracy rate with validation

methods for classifying CNRPark+EXT TEST with CNRPark_EXT TRAIN C1-C8 training data are presented in Table 5.

Based on Table 5, it can be seen that the accuracy level of EffecientParkingNet is better than the two previous studies which classified the sub dataset of CNRPark+EXT Test using CNRPark_EXT TRAIN C1-C8 training data. Furthermore, several researchers claim to have succeeded in increasing the accuracy level with different testing schemes on the CNRPark+EXT dataset by using the Yolo+MobileNet method (Chen *et al.*, 2020) and GDSN-IC (Connie *et al.*, 2021). The results of the comparison accuracy level of EffecientParking Net with the results of previous studies are in Table 6.

Based on Table 6, it can be seen that EfficientParkingNet is better than mAlexnet and GDSN-IC. Also, it can also be seen that the results of EfficientParkingNet are close to the results of Yolo+MobileNet. Yolo+MobileNet has a large number of parameters, more than 4 million parameters. It causes a large computational process and takes time. Therefore, Yolo+MobileNet is not suitable to be used on low-computing devices such as the Raspberry Pi. EfficientParkingNet has a number of parameters of 22.0000 or (half of) the number of parameters of mAlexnet. EfficientParkingNet is more suitable for use on low computing devices. The comparison of the number of parameters is presented in Table 7.

Based on Table 7, it can be seen that EfficientParkingNet has fewer parameters than the other two methods. The number of mAlexnet parameters is 1/1340 of Alexnet parameters (Amato *et al.*, 2017). The number of parameters of Alexnet is 61 million, so the number of parameters of mAlexnet is 45,522. By using equation 1 and equation 2, the number of parameters of mAlexnet is determined to be 32,648 parameters. Tables 8 and 9 show the number of parameters and activation shapes calculated using Eqs. 1 to 3 for mAlexnet, and EfficientParkingNet, either manually or using Pytorch Code.

Based on Tables 8 and 9, it can be seen that the number of EfficientParkingNet parameters is less than

Table 4: Comparison of the accuracy level all data

Methods	Accuracy rate
mAlexnet	98.13%
EfficientParkingNet	98.44%

Table 5: Comparison of CNRPark+EXT test classifications

Arsitecture	Accuracy Level	References
mAlexnet	95.88%	Amato *et al.*, 2017
background subtraction+SVM	96.59%	(Varghese and Sreelekha, 2019)
EfficientParkingNet	96.85%	Proposed method

Table 6: Comparison of accuracy of several architectures

Methods	Level of accuracy	References
mAlexnet	98.13%	Amato *et al.*, 2017
Yolo+MobileNet	99%	Chen *et al.*, 2020
GDSN-IC	97.10%	Connie *et al.*, 2021
EfficientParkingNet	98.44%	Proposed method

Table 7: Comparison of the number of parameters

Architecture	Parameters
mAlexnet	45K
EfficientParkingNet	22K
Yolo+MobileNet	4M

Table 8: The Number of parameters of mAlexnet

Layers	Activation Shape	Activation Size	Parameters
Input Layer	(224,224,3)	150,528	0
Conv1 (11/4 16)	(54,54,16)	46,656	5,824
Pooling (3/2)	(26,26,16)	10,816	0
Conv2 (5/2 20)	(22,22,20)	9,680	8,020
Pooling (3/2)	(10,10,20)	2,000	0
Conv3(3/2 30)	(8,8,30)	1,920	5,430
Pooling (3/2)	(3,3,30)	270	0
FC1 (48)	(48,1)	48	13,230
FC2(2)	(2,1)	2	144
Total			32,648

mAlexnet parameters. The number of parameters for mAlexnet is 33,000, and for EfficientParkingNet is about 22,000. These parameters determine the file size for each architecture. The smaller the file size, the more efficient the storage space, especially on embedded system devices. Table 10 presents the experimental results of the relationship between the number of parameters and file size.

Table 10 shows that the file size of EfficientParkingNet is smaller than mAlexnet. Based on the calculation results, EfficientParkingNet has a better accuracy level, a smaller number of parameters, and a smaller file size compared to mAlexnet. Therefore, EfficientParkingNet is suitable to be used on low computing devices and embedded systems.

Table 9: The Number of parameters of EfficientParkingNet

Layers	Activation Shape	Activation Size	Parameters
Input Layer	(224,224,3)	150,528	0
Conv1 (3/4 30)	(56,56,30)	94,080	840
Pooling (3/2)	(27,27,30)	21,870	0
Conv2 (3/2 30)	(25,25,30)	18,750	8,130
Pooling (3/2)	(12,12,30)	4,320	0
Conv3 (3/2 15)	(10,10,15)	1,500	4,065
Conv4 (3/2 15)	(8,8,15)	960	2,040
Pooling (3/2)	(3,3,15)	135	0
FC1 (48)	(48,1)	48	6,615
FC2(2)	(2,1)	2	144
Total			21,834

Table 10: Comparison of parameters and file size

Architecture	Parameters	File Size
mAlexnet	33K	133K
EfficientParkingNet	22K	90K

CONCLUSION

The increased number of vehicles resulted in traffic congestion and pollution. As much as 30% of traffic jams are caused by drivers searching for a parking area. Smart parking systems can direct vehicles to parking spaces based on the type of vehicle, the appropriate rate, and the available space by taking the shortest route. A CNN architecture used to detect parking space named EfficientParkingNet has been developed. It is a “small architecture” which is a development of the mAlexnet architecture, with four layers of convolution and LeakyRelu as an activation function. The low level of accuracy is caused by overfitting. Accuracy is also reduced if the training data does not represent all of the underfitting test data, which is caused by the difference between the training data and the test data. Modifications were also made to the distribution of training data and test data. The first test, which used CNRPark training data and CNRPark-EXT TEST test data, yielded a low accuracy rate of 93.87%. This is due to the fact that the training data does not accurately represent the features in the test data. EffecientParkingNet can analyze the training dataset but not recognize it in the dataset test. In this study, all CNRPark+EXT datasets are used and divided at random, with 80% used for training and 20% used for testing. The use of all data is expected to represent the real-time state of the parking system. An increase in the accuracy level of EfficientParkingNet has been seen, bringing it to 98.44%, higher than the accuracy level of mAlexnet. According to recent research, different Yolo+MobileNet testing schemes have been successful in increasing accuracy. Background subtraction plus SVM and GDSN-IC equals Yolo+MobileNet accuracy level. It can also be seen that the results of EfficientParkingNet are close to the results of Yolo+MobileNet. Yolo+MobileNet has a large number of parameters, more than 4 million parameters. It causes a large computational process. Therefore, Yolo+MobileNet is not suitable to be used on low-computing devices such as the Raspberry Pi. The number of parameters in EfficientParkingNet architecture is 22,000 parameters, or 2/3 of the number of mAlexnet parameters and 1/182 of Yolo+MobileNet parameters. Therefore, EfficientParkingNet is suitable to be used on low computing devices such as the Raspberry Pi. It has an impact on the reduction of implementation costs.

AUTHOR CONTRIBUTIONS

S. Rahman performed the literature review, running the model, analyzed and interpreted the data, prepared the manuscript text, and manuscript edition. M. Ramli performed the literature review, running the model, analyzed and interpreted the data, prepared the manuscript text, and manuscript edition. F. Arnia and R. Muharar performed the literature review, prepared numerical code, prepared the manuscript text, and manuscript edition. M. Ikhwan and S.

Munzir performed the literature review, analyzed and interpreted the data, prepared the manuscript text, and manuscript edition. All authors agreed on the final version of the manuscript.

ACKNOWLEDGEMENT

This study is funded by Professor Research Grant, Universitas Syiah Kuala 2021, with contract number [2/UN11.2.1/PT.01.03/PNBP/2021].

CONFLICT OF INTEREST

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

OPEN ACCESS

PUBLISHER'S NOTE

GJESM Publisher remains neutral with regard to jurisdictional claims in published maps and institutional afflictions.

ABBREVIATIONS

%	Percent
Alexnet	Alex Network, network architecture created by Alex Krizhevsky.
Cl	The number of output Current Layer
CNN	Convolutional Neural Networks
CNR	Council of National Research
CNRPark	CNR Parking dataset
CNRPark-EXT	New updated of CNRPark
CO_2	Carbon dioxide
Conv1, Conv2, Conv3, Conv4	Fisrt, second, third, and fourth convolution layer, respectively
ELU	Exponential linear unit
Eq. / Eqs.	Equation / equations
F	Filtersize
$F_{\_Size}$	The filter size used
FC1	The first fully connected
FC2	The second fully connected
Fig.	Figure
GDSN-IC	Grassmannian Deep Stacking Network with Illumination Correction
GoogleNet	Network architecture developed by a team at Google
GPS	Global positioning systems
ICT	Information and Communication Technology
IoT	Internet of things
Lenet	LeCun Network, convolutional neural network structure proposed by LeCun *et al.* (1998)
LeakyReLu	Leaky rectified linear unit
LRN	Local Response Normalization
MaxPooling	Maximum pooling
mAlexnet	mini Alexnet
mLenet	mini Lenet
MobileNet	Mobile Neural Network
MobileNetV2	Version 2 of MobileNet
N	The number of channels in the previous layer
N_F	The number of filters
$O_{\_width}$	Output width
p	The number of padding
P_C	The number of parameters in convolution layer
$P_{\_Fc}$	The number of parameters in Fully Connected layer
PKLot	Parking lot databased
Pl	The number of output Previous Layer
R-CNN	Region CNN

Relu	Rectified linear unit
Resnet	Residual neural network
s	The stride value
ShuffleNet	Shuffle Neural Network
SVM	Support vector machine
Tanh	Hyperbolic tangent
VGGNet	Network architecture that invented by VGG (Visual Geometry Group) from University of Oxford
W	Tensor size
Yolo	You only look once network architecture

REFERENCES

Amato, G.; Carrara, F.; Falchi, F.; Gennaro, C.; Meghini, C.; Vairo, C., (2017). Deep learning for decentralized parking lot occupancy detection. Expert Syst. Appl., 72: 327–334 (**8 pages**).

Amato, G.; Carrara, F.; Falchi, F.; Gennaro, C.; Vairo, C., (2016). Car parking occupancy detection using smart camera networks and deep learning. In 2016 IEEE Symposium on Computers and Communication (ISCC), 1212–1217 (**6 pages**).

Bura, H.; Lin, N.; Kumar, N.; Malekar, S.; Nagaraj, S.; Liu, K., (2018). An edge based smart parking solution using camera networks and deep learning. In 2018 IEEE International Conference on Cognitive Computing (ICCC), 17–24 (**8 pages**).

Chen, L.; Sheu, R; Peng, W.; Wu, J; Tseng, C., (2020). Video-based parking occupancy detection for smart control system. Appl. Sci., 10(3): 1079 (**22 pages**).

Chiang, Y.; Knoblock, C.A., (2013). A general approach for extracting road vector data from raster maps. Int. J. Doc. Anal. Recognit., 16: 55–81 (**27 pages**).

Connie, T.; Goh, M.K.O.; Koo, V.C.; Murata, K.T.; Phon-Amnuaisuk, S., (2021). Improved parking space recognition via grassmannian deep stacking network with illumination correction. In International Conference on Computational Intelligence in Information System, 150–159 (**10 pages**).

de Almeida, P.R.L.; Oliveira, L.S.; Britto Jr, A.S.; Silva Jr, E.J.; Koerich, A.L., (2015). PKLot–A robust dataset for parking lot classification. Expert Syst. Appl., 42: 4937–4949 (**13 pages**).

de Gusmao, P.P.B.; Francini, G.; Lepsøy, S.; Magli, E., (2016). Fast training of convolutional neural networks via kernel rescaling. arXiv Prepr. arXiv1610.03623 (**13 pages**).

del Postigo, C.G.; Torres, J.; Menéndez, J.M., (2015). Vacant parking area estimation through background subtraction and transience map analysis. IET Intell. Transp. Syst., 9: 835–841 (**7 pages**).

Gavrilov, A.D.; Jordache, A.; Vasdani, M.; Deng, J., (2018). Preventing model overfitting and underfitting in convolutional neural networks. Int. J. Softw. Sci. Comput. Intell., 10: 19–28 (**10 pages**).

He, K.; Zhang, X.; Ren, S.; Sun, J., 2016. Deep residual learning for image recognition. In the IEEE Conference on Computer Vision and Pattern Recognition, 770–778 (**9 pages**).

Huang, C.; Tai, Y.; Wang, S., (2013). Vacant parking space detection based on plane-based Bayesian hierarchical framework. IEEE Trans. Circuits Syst. Video Technol., 23: 1598–1610 (**13 pages**).

Hussain, M.; Bird, J.J.; Faria, D.R., (2018). A study on cnn transfer learning for image classification. In UK Workshop on Computational Intelligence, 191–202 (**12 pages**).

Ikhwan, M; Mardlijah, M.; Arif, D.K., (2017). Model predictive parking control on four wheel vehicle with optimum parking space. In International Conference on Electrical Engineering and Informatics, 35-39 (**5 pages**).

Krizhevsky, A.; Sutskever, I.; Hinton, G.E., (2012). Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems, 1097–1105 (**9 pages**).

LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P., (1998). Gradient-based learning applied to document recognition. Proc. IEEE, 86: 2278–2324 (**10 pages**).

Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; Graf, H.P., (2016). Pruning filters for efficient convnets. In Proceedings of NIPS Workshop on Efficient Methods for Deep Neural Networks, 1-5 (**5 pages**).

Lin, T.; Rivano, H.; Le Mouël, F., (2017). A survey of smart parking solutions. IEEE Trans. Intell. Transp. Syst. 18, 3229–3253 (**25 pages**).

Ma, N.; Zhang, X.; Zheng, H.; Sun, J., (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), 116–131 (**16 pages**).

Mahmud, S.A.; Khan, G.M.; Rahman, M.; Zafar, H., (2013). A survey of intelligent car parking system. J. Appl. Res. Technol, 11: 714–726 (**13 pages**).

Màrmol, E.; Sevillano, X., (2016). QuickSpot: a video analytics solution for on-street vacant parking spot detection. Multimed. Tools Appl., 75: 17711–17743 (**33 pages**).

Masmoudi, I.; Wali, A., (2019). Vision based approach for adaptive parking lots occupancy estimation. Pattern Recognit. Image Anal., 29: 515–522 (**8 pages**).

Rahman, S.; Ramli, M.; Arnia, F.; Muharar, R.; Luthfi, M.; Sundari, S., (2020a). Analysis and comparison of hough transform algorithms and feature detection to find available parking spaces. J. Phys.: Conf. Ser., 1566: 12092 (**7 pages**).

Rahman, S.; Ramli, M.; Arnia, F.; Muharar, R.; Sembiring, A., (2020b). Convolutional neural network customization for parking occupancy detection. In 2020 International Conference on Electrical Engineering and Informatics, 1-6 (**6 pages**).

Rahman, S.; Ramli, M.; Arnia, F.; Muharar, R.; Sembiring, A., (2021). Performance analysis of mAlexnet by training option and activation function tuning on parking images. IOP Conf. Ser.: Mater. Sci. Eng., 1087: 12084 (**7 pages**).

Sairam, B.; Agrawal, A.; Krishna, G.; Sahu, S.P., (2020). Automated vehicle parking slot detection system using deep learning. In 2020 Fourth International Conference on Computing Methodologies and Communication, 750–755 (**6 pages**).

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L., (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In IEEE Conference on Computer Vision and Pattern Recognition, 4510–4520 (**11 pages**).

Simonyan, K.; Zisserman, A., (2014). Very deep convolutional networks for large-scale image recognition. In International Conference on Learning Representations, 1-14 (**14 pages**).

Skourt, B.A.; El Hassani, A.; Majda, A., (2021). Mixed-pooling-dropout for convolutional neural network regularization. J. King Saud Univ. Inf. Sci., 33:1-7 (**7 pages**).

Sukarno, I.; Matsumoto, H.; Susanti, L., (2016). Transportation energy consumption and emissions-a view from city of Indonesia. Futur. Cities Environ., 2: 6 (**11 pages**).

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A., (2015). Going deeper with convolutions. In IEEE Conference on Computer Vision and Pattern Recognition, 1–9 (**9 pages**).

True, N., (2007). Vacant parking space detection in static images. Univ. California, San Diego.

Uttara, S.; Bhuvandas, N.; Aggarwal, V., (2012). Impacts of urbanization on environment. Int. J. Res. Eng. Appl. Sci., 2: 1637–1645 (**9 pages**).

van Grinsven, M.J.J.P.; van Ginneken, B.; Hoyng, C.B.; Theelen, T.; Sánchez, C.I., (2016). Fast convolutional neural network training using selective data sampling: Application to hemorrhage detection in color fundus images. IEEE Trans. Med. Imaging, 35: 1273–1284 (**12 pages**).

Varghese, A.; Sreelekha, G., (2019). An efficient algorithm for detection of vacant spaces in delimited and non-delimited parking lots. IEEE Trans. Intell. Transp. Syst., 21: 4052–4062 (**11 pages**).

Yamanaka, J.; Kuwashima, S.; Kurita, T., (2017). Fast and accurate image super resolution by deep CNN with skip connection and network in network. In International Conference on Neural Information Processing, 217–225 (**9 pages**).

Zhang, L.; Li, X.; Huang, J.; Shen, Y.; Wang, D., (2018). Vision-based parking-slot detection: a benchmark and a learning-based approach. Symmetry, 10: 64 (**18 pages**).

Zhang, Yu; Zhang, Yan; Shi, Z.; Zhang, J.; Wei, M., (2019). Design and training of deep CNN-based fast detector in infrared SUAV surveillance system. IEEE Access, 7: 137365–137377 (**13 pages**).

AUTHOR (S) BIOSKETCHES

Rahman, S., Ph.D. Candidate, School of Engineering, Universitas Syiah Kuala, Banda Aceh 23111, Indonesia.
- Email: *sayuti_r@mhs.unsyiah.ac.id*
- ORCID: 0000-0002-4410-4026
- Web of Science ResearcherID: AAE-4070-2021
- Scopus Author ID: 57203457552
- Homepage: *http://dit.unsyiah.ac.id/*

Ramli, M., Ph.D., Professor, Department of Mathematics, Universitas Syiah Kuala, Banda Aceh 23111, Indonesia.
- Email: *marwan.math@unsyiah.ac.id*
- ORCID: 0000-0003-1225-9063
- Web of Science ResearcherID: A-2686-2017
- Scopus Author ID: 57217110324
- Homepage: http://math.unsyiah.ac.id/ind/marwan/

Arnia, F., Ph.D., Professor, Department of Electrical Engineering, Universitas Syiah Kuala, Banda Aceh 23111, Indonesia.
- Email: *f.arnia@unsyiah.ac.id*
- ORCID: 0000-0001-6020-1275
- Web of Science ResearcherID: R-5711-2017
- Scopus Author ID: 14027791000
- Homepage: http://www.fsd.unsyiah.ac.id/fitri.arnia/

Muharar, R., Ph.D., Associate Professor, Department of Electrical Engineering, Universitas Syiah Kuala, Banda Aceh 23111, Indonesia.
- Email: *r.muharar@unsyiah.ac.id*
- ORCID: 0000-0002-5079-4174
- Web of Science ResearcherID: T-9108-2018
- Scopus Author ID: 37085406000
- Homepage: http://fsd.unsyiah.ac.id/rusdha/

Ikhwan, M., Ph.D. Candidate, Graduate School of Mathematics and Applied Sciences, Universitas Syiah Kuala, Banda Aceh 23111, Indonesia.
- Email: *m.ikhwan@mhs.unsyiah.ac.id*
- ORCID: 0000-0002-8162-1479
- Web of Science ResearcherID: D-1570-2018
- Scopus Author ID: 57201743214
- Homepage: http://dmas.unsyiah.ac.id/

Munzir, S., Ph.D., Associate Professor, Department of Mathematics, Universitas Syiah Kuala, Banda Aceh 23111, Indonesia.
- Email: *smunzir@unsyiah.ac.id*
- ORCID: 0000-0003-4717-5521
- Web of Science ResearcherID: ABD-9651-2021
- Scopus Author ID: 57195062107
- Academic or organizational Homepage link: http://fsd.unsyiah.ac.id/smunzir/

HOW TO CITE THIS ARTICLE

Rahman, S.; Ramli, M.; Arnia, F.; Muharar, R.; Ikhwan, M.I.; Munzir, S., (2022). Enhancement of convolutional neural network for urban environment parking space classification. Global J. Environ. Sci. Manage., 8(3): 315-326.

DOI: 10.22034/gjesm.2022.03.02

url: https://www.gjesm.net/article_246490.html