Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 23, No. 3, July 2024, pp. 557∼568
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 200/M/KPT/2020
DOI: 10.30812/matrik.v23i3.3871

Normalization Layer Enhancement in Convolutional Neural Network for Parking Space Classification

Sayuti Rahman[1], Marwan Ramli[2], Arnes Sembiring[1], Muhammad Zen[3], Rahmad B.Y Syah[1]
[1]Universitas Medan Area, Medan. Indonesia
[2]Universitas Syiah Kuala. Banda Aceh. Indonesia
[3]Universitas Pembangunan Panca Budi, Medan, Indonesia

Article Info

Article history:

Received February 02, 2024
Revised March 12, 2024
Accepted May 30, 2024

Keywords:

Convolutional Neural Network
Classification
Enhancement
Normalization Layer
Parking Space

ABSTRACT

The **research problem** of this study was the urgent need for real-time parking availability information to assist drivers in quickly and accurately locating available parking spaces, aiming to improve upon the accuracy not achieved by previous studies. This research aimed to enhance the classification accuracy of parking spaces using a Convolutional Neural Network (CNN) model, specifically by integrating an effective normalizing function into the CNN architecture. **The research method** employed involved the application of four distinct normalizing functions to the EfficientParkingNet, a tailored CNN architecture designed for the precise classification of parking spaces. **The results of this study** indicated that the EfficientParkingNet model, when equipped with the Group Normalization function, outperforms other models using Batch Normalization, Inter-Channel Local Response Normalization, and Intra-Channel Local Response Normalization in terms of classification accuracy. Furthermore, it surpasses other similar CNN models such as mAlexnet, you only look once (Yolo)+mobilenet, and CarNet in the same classification task. This demonstrates that EfficientParkingNet with Group Normalization significantly enhances parking space classification, thus providing drivers with more reliable and accurate parking availability information.

Corresponding Author:

Sayuti Rahman, +6281262847435,
Department of Informatics, Engineering Faculty,
Universitas Medan Area, Medan, Indonesia,
Email: sayutirahman@staff.uma.ac.id.

How to Cite:
sayuti rahman, M. Ramli, A. Sembiring, M. Zen, and R. B. Syah, ”Normalization Layer Enhancement in Convolutional Neural Network for Parking Space Classification”, *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol. 23, No. 3, pp. 557-568, July, 2024.

1. INTRODUCTION

Drivers going around looking for parking spaces have negative impacts such as traffic jams, waste of fuel, air pollution by exhaust gases, panic, and others [1]. Information on the availability of parking spaces is an important part of a smart parking system. This information can guide drivers to empty spaces more quickly and efficiently. Information on the availability of parking spaces is obtained from the results of the classification of parking spaces using sensors or based on computer vision. Currently, the use of computer vision is increasing because the implementation costs are cheaper than using sensors. A camera can monitor many parking spaces at once. Neural Network (CNN) is a widely used computer vision method. CNN architecture is growing, so it can classify objects with high accuracy. CNN is used in many fields such as industry [2, 3], agriculture [2], health [3], transportation, and parking. Many existing CNNs for parking space classification have been developed, including mAlexnet [4] with an accuracy of 98.13% and several parameters of 32,000, Yolo+MobileNet [5] with an accuracy of 97.24% and several parameters of 4,000,000, Mini ShuffelNet [6] with an accuracy of 98.37% and several parameters of 24,000, GDSN-IC [7] with an accuracy of 97.10%, and EfficientParkingNet [8] with an accuracy of 98.44% and several parameters of 22,000. EfficientParkingNet is an architecture with better accuracy and speed in classifying parking spaces, which is 98.44% on the CNRPark+EXT dataset. **There is a gap** that has not been resolved by previous research, namely low classification accuracy and needs to be improved.

The level of accuracy and speed of classification measures CNN's success rate. EfficientParkingNet has fewer learning parameters, so it is faster in the classification process, but the accuracy level still needs improvement. One technique to increase the accuracy of a CNN architecture is fine-tuning. Resetting the hyperparameters or primary components of the CNN allows for fine-tuning. The number and size of filters [9], stride [9], padding, learning rate [10], epoch [11], and optimizer function [12] are all frequently specified hyperparameters. The convolution layer [13], fully-connected layer, activation layer [6], dropout layer, and normalization layer [13] are some of the primary components of CNN that researchers frequently fine-tune.

The difference between this research and previous research is that EfficientParkingNet's accuracy has improved by modifying the normalization layer. It is hoped that the better the accuracy, the more precise the information provided to the driver and the ability to overcome the negative impacts arising from the driver looking for a parking space. In addition, research was conducted to analyze the influence of normalization techniques on CNN architecture. Currently, there is no research on analysis normalization techniques such as local response normalization, batch normalization, and group normalization. **The objectives of this research** are to develop a CNN model that achieves higher accuracy while maintaining fast classification speed. By using a smaller number of parameters, the classification process becomes quicker and can be executed on low-power computing devices. Consequently, **this research contributes** to improving the accuracy of parking space classification and providing real-time information to drivers. With precise information, drivers can save fuel, reduce traffic congestion, and avoid the stress of driving around in search of parking spaces.

2. RESEARCH METHOD

The normalization function can increase CNN network accuracy for the following reasons: 1. The normalization function aids in resolving major scale differences between features or inputs in a dataset. The sizes of features in a dataset might vary greatly, which can affect network learning; 2. When the gradient sent over the network layers becomes extremely tiny or very large, the missing gradient problem emerges. This can impede training and result in poor convergence. The gradient is kept within a tolerable range by applying normalization, allowing for more steady and efficient learning; 3. In neural networks, normalization serves as an effective regularization strategy. Normalization reduces overfitting by minimizing correlations and dependencies between features. Overfitting is the tendency for networks to learn too well from training data but not generalize well to new data.

The process of normalization involves transforming input values into a generic form. This change is based on a scale only and does not change the input's brightness, contrast, or value. Normalization is used on CNN to improve accuracy and speed up the training process [14–16]. The normalizations known in CNN include local response normalization (LRN), Batch Normalization (BN), and Group Normalization. The strategy used in this study is to fine-tune EfficientParkingNet in the normalization layer. The research method is illustrated in Figure 1. As depicted in Figure 2, the dataset of parking space images is classified as vacant or occupied. The method employed for parking space classification utilizes an existing Convolutional Neural Network (CNN) known as EfficientParkingNet. To enhance the CNN's accuracy, we perform fine-tuning specifically on the normalization layers, aiming for a more substantial improvement in accuracy.

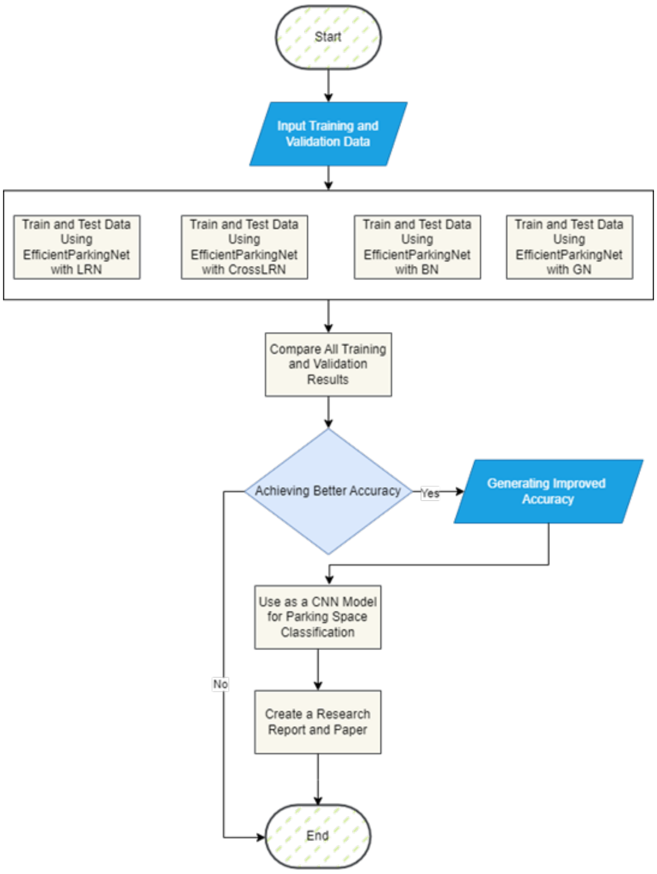

Figure 1. Research Method

2.1. Dataset

The dataset used is CNRPark+EXT, which consists of 150,000 images from 164 parking spaces collected by the CNR Research Area in Pisa, Italy. This dataset was first used to test mAlexnet [4]. The CNRPark+EXT dataset has a more difficult classification difficulty than PKLot in [16]. CNRPark+EXT has obstructions such as trees, lamp posts, and more. The experiment was done by dividing the dataset into 80% training data and 20% test data. The following is an example of the CNRPark+EXT dataset shown in Figure 2.

Figure 2. Parking Spaces on the CNRPark+EXT Dataset

2.2. EfficientParkingNet

EfficientParkingNet is a dedicated Convolutional Neural Network (CNN) architecture designed specifically for parking space categorization, with four convolution layers and two fully connected layers. The design includes a sequence in which each convolutional layer is followed by activation, normalization, and pooling functions, which improves the network's overall efficiency. Figure 3 depicts EfficientParkingNet's complete architecture, including structural components flow over the network.

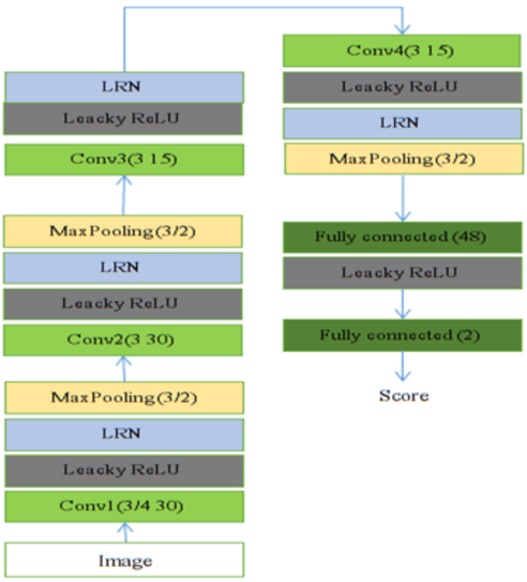

Figure 3. EfficientParkingNet Architecture [8]

Based on Figure 3, it can be seen that the EfficientParkingNet Architecture uses normalization for each convolution layer, namely Local Response Normalization (LRN). The LNR used is Inter-Channel LRN. In this study, we compared the effect of the normalization function. Four normalization functions were compared: LRN, CrossMapLRN (intra-channel LRN), Batch normalization, and group normalization. The best normalization function from the test results will be applied to CNN for parking space classification with high accuracy. When using the Group Normalization function, in the first and second convolution layers, the group used is 10, and the number of groups in the third and fourth layers is five. The test results are analyzed to determine the best function that can be implemented on EfficientParkingNet.

2.3. Local Response Normalization

Local Response Normalization (LRN) is used to normalize input values; LRN was first introduced in the AlexNet architecture. On Alexnet, LRN is paired with the ReLu activation function. LRN serves to promote lateral inhibition. Neurobiology's concept refers to a neuron's capacity to reduce its neighbors' activity. This lateral inhibition aims to perform a local contrast enhancement so that the local maximum pixel value is used as an excitation for the next layer. The LRN is a non-trainable layer that normalizes the square of the pixel values in the feature map in the local environment. Two types of LRN are based on the environment, namely Inter-Channel LRN and Intra-Channel LRN. Inter-Channel LRN is an LRN introduced in Alexnet. The normalization value is determined by the channel or different channels at each position (x, y). Inter-channel LRN normalization is generated by the formula in Equation 1.

$$b_{x,y}^{i} = \frac{a_{x,y}^{i}}{\left(k + \alpha \Sigma_{j=max(0,i-\frac{n}{2})}^{min(N-1,i+\frac{n}{2})} (a_{x,y}^{j})^2\right)^{\beta}} \quad (1)$$

Where $b^i_{x,y}$ is the pixel value at position (x, y) after normalization, $a^i_{x,y}$ is the input image pixel value at position (x, y), i is the output of the filter to i, k is the value used to avoid division by zero, is the constant of contrast, n represents the number of neighbors involved, and N is the number of channels. Intra-channel LRN is different from Inter-Channel LRN. Inter-channel LRN is normalized by involving different input channels, while intra-channel LRN is normalized to involve neighboring pixels on the same channel. Intra-channel LRN is calculated by Equation 2.

$$b^k_{x,y} = a^k_{x,y} / \left(kx + \alpha \sum_{i=max(0,x-n/2)}^{min(W,x+\frac{n}{2})} \sum_{j=max(0,y-n/2)}^{min(H,y+\frac{n}{2})} (\alpha^k_{i,j})^2 \right)^\beta \tag{2}$$

Where, $b^k_{x,y}$ is the normalized value at position (x, y) for channel k, $a^k_{x,y}$ is the original input value at position (x, y) for channel k before normalization, k is a constant added to the denominator to prevent division by zero, providing numerical stability. α is a scaling parameter that multiplies with the summation term, β is the exponent applied to the summation in the denominator, controlling the strength of the normalization, n is represents the number of neighboring pixels considered in each direction (left/right for x and up/down for y) around a central pixel for normalization. W and H are the width and height of the map feature. In the following discussion, Inter-Channel is called LRN, and Intra-Channel is called CrossMapLRN.

2.4. Batch Normalization

A Batch is a group of data, and the input data is grouped into small parts or, called a mini-batch. This batch is useful to ease the computation of the CNN training process. Therefore, batch normalization (BN) is the process of normalizing the input data in each group. Batch normalization has different ways during training and testing. During training, the mean and variance of the features are calculated for each mini-batch. Each data in the batch is normalized to the mean and variance values with Equation 3.

$$\hat{x}_i = \frac{x_i - \mu}{\sqrt{\sigma^2 + \epsilon}} \tag{3}$$

Where $\hat{x}$ is the feature value in the normalized mini-batch, i is the index, μ is the mean value in the mini-batch, σ^2 is the variance value in the mini-batch, and ϵ is a constant value to stabilize the numerical value. The results of normalization are scaled and shifted with Equation 4, which functions to return the normalized feature distribution to before normalization; if without normalization, it is more optimal. The output after this scale and shift is the output of batch normalization. Calculate the running average mean and variance with Equations 5 and 6 to estimate the mean and variance of the entire dataset.

$$y_i = y\hat{x}_i + \beta \tag{4}$$

Where y_i is the result of the scale and shift of the normalized value, y is the variable for the normalized value scale and β is the variable for the normalized value shift.

$$\mu_{run} = (1 - \alpha)\mu_{run} + \alpha\mu \tag{5}$$

$$\sigma^2_{run} = (1 - \alpha)\sigma^2_{run} + \alpha\sigma^2 \tag{6}$$

Where μ_{run} is the running average of the mean value during training and α^2_{run} is the running average of the variance value during training. This running average value is used for testing using Equation 7. Without batch normalization, the feature distribution depends on the previous layer, while with batch normalization, the feature distribution depends on γ and β. This makes the CNN training process with batch normalization easier.

$$\hat{x}_i = \frac{x_i - \mu run}{\sqrt{\sigma^2_{run}} + \epsilon} \tag{7}$$

Batch normalization is widely used in architecture because it can improve CNN accuracy. Some applications that use batch normalization include lip motion recognition, signature recognition [17], and other object recognition.

2.5. Group Normalization

Group Normalization (GN) is a normalization function developed based on Batch Normalization to improve the error rate in small batch sizes. GN is almost the same as BN, Layer Normalization (LN) [35], and Instance Normalization (IN) [36] by calculating the mean and variance for normalization. The difference between these four functions is that the map features involved in calculating the mean and variance are different. The differences in BN, IN, LN, and GN [37] are shown in Figure 4.

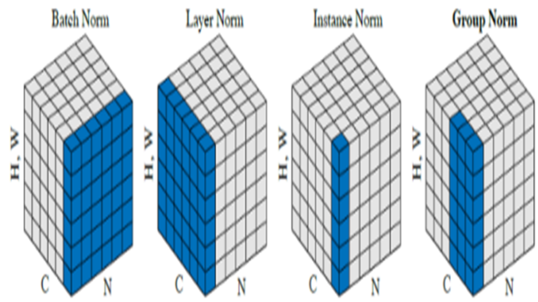

Figure 4. Normalization Functions. C is the channel axis, N is the batch, and (H,W) is the height and width of the spatial feature [37]

As can be seen in Figure 4, BN works on each channel and the entire batch. BN is defined by the following Equation 8 [36].

$$S_i = k \mid k_C = i_C \tag{8}$$

Where S_i is the data set that produces the mean and variance for normalization. k_C and i_C represent the index of the C axis. This shows that normalization is carried out on the same channel. BN is calculated along the axis (N, H, W). The map features involved in calculating the mean and variance with LN are presented in the following equation 9.

$$S_i = k \mid k_N = i_N \tag{9}$$

Equation 9 shows that the mean and variance are calculated on all channels in the same batch. The map features involved in calculating the mean and variance with IN are presented with the following equation 10.

$$S_i = k \mid k_C = i_N, k_C = i_C \tag{10}$$

Equation 10 shows that each channel and batch's mean and variance are calculated on (H, W). Meanwhile, GN is presented in Equation 11.

$$S_i = k \mid k_C = i_N, [\frac{k_C}{C/G}] = [\frac{i_C}{C/G}] \tag{11}$$

Where G is the number of groups and C/G is the number of channels in one group to be normalized.

3. RESULT AND ANALYSIS

This study's findings and analysis involve the utilization of datasets during the EfficientParkingNet architecture's training and testing stages. Then, the effect of layer normalization on parking space classification was investigated. It is envisaged that a normalizing layer capable of obtaining the highest accuracy can be used to improve EfficientParkingNet's performance, and the resulting data can be used to construct intelligent parking systems.

Smart parking systems emphasize the importance of accuracy and computational efficiency. The effectiveness of a CNN model in these systems is gauged by its ability to operate on devices with limited computational power. Key factors assessed in improving models include Peak Accuracy, which Measures the highest accuracy reached during tests, showing the effectiveness of different normalization methods like Group Normalization, which excels in dataset management; 2. Comparative Performance: Involves direct comparisons of different models using the same dataset; 3. Computational Efficiency: Focuses on the number of parameters each model uses, which is crucial in environments with limited computational resources. This helps determine the practicality of deploying specific models in real-world applications.

3.1. Training Accuracy Comparison

The programming language used in this study was Python, and the framework used was PyTorch. The number of epochs used was 20. All CNRPark+EXT data are divided into 80% as training data and 20% as validation data. So, all experiments used the same data. The training result showed that LRN accuracy was 94.3%, CrossMapLRN was 94.4%, Batch Normalization was 96.3%, and Group Normalization was 96.6%. The following training results are shown in Figure 5.

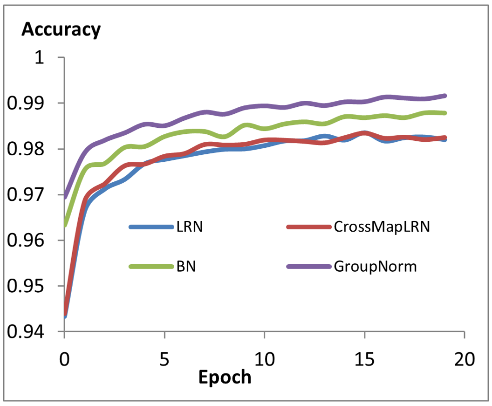

Figure 5. EfficientParkingNet Training Graph with Various Normalization Functions

Based on Figure 5, it can be seen that Group Normalization has the highest accuracy from the beginning of training till 20 epoch. Batch Normalization is in the second position, better than LRN and CrossMapLRN. The group normalization function achieves better accuracy by dividing the input channel into several groups and then normalizing each group. This study has shown that this method improves CNN classification accuracy in the classification task. The other three normalization functions normalize the input for each channel simultaneously (in the Layer Normalization function) or for each batch (in the Batch Normalization function). It is natural for the training time to be longer. Still, the difference will be insignificant because the difference is only in the number of stages, while the normalized data remains unchanged.

3.2. Validation Accuracy Comparison

Validation accuracy is critical during the testing phase of Convolutional Neural Network (CNN) models before deployment for parking space classification. The significance lies in the fact that a higher validation accuracy suggests a well-designed architecture capable of correctly distinguishing and categorizing information. Figure 6 depicts the results of the validation accuracy comparison across different models, providing a clear depiction of the performance differences and aiding in assessing the usefulness of each design in the context of parking space classification.

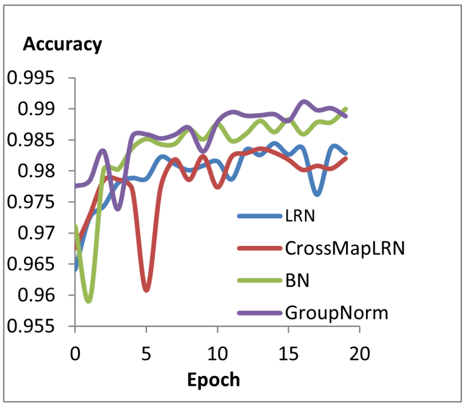

Figure 6. Comparison Graph of Validation Accuracy

Based on Figure 6, LRN and CrossMapLRN produce low accuracy, and batch normalization shows the most stable accuracy results and relatively increases in each epoch. Group Normalization has the highest accuracy compared to other methods. The highest validation accuracy in 20 epochs is shown in Table 1.

Table 1. Validation accuracy of some normalization functions on EfficientParkingNet

Normalization Methods	Validation Accuracy
LRN	98.44%
CrossMapLRN	98.35%
BN	98.99%
GroupNorm	99.11%

Table 1 presents the validation accuracies of different normalization methods applied within the EfficientParkingNet across 20 epochs, evaluating their performance in parking space classification. Local Response Normalization (LRN) and CrossMapLRN recorded accuracies of 98.44% and 98.35%, respectively, indicating moderate effectiveness. Batch Normalization (BN) showed greater stability and achieved an accuracy of 98.99%. Notably, Group Normalization delivered the best performance with an accuracy of 99.11%, demonstrating its better capability in managing dataset variations effectively. These findings underscore Group Normalization as the most effective approach for improving classification accuracy in smart parking systems. Further comparisons were made utilizing multiple architectures that also performed classification tasks on the CNRpark+EXT dataset. As comparisons, Yolo+MobileNet [5], Mini ShuffleNet [6], EfficientParkingNet, which just utilizes the LN normalization function [8], and mAlexnet [4] were chosen.

Table 2. CNN Validation Accuracy for Parking Space Classification

CNN Methods	Validation Accuracy
Re-trained mAlexnet	98,13%
Yolo+MobileNet [5]	99.00%
Mini ShuffleNet [6]	98.37%
EfficientParkingNet [8]	98.44%
EfficientParkingNet + Group Norm	99.11%

Table 2 summarizes the accuracy of numerous architectures from previous studies such as [7, 8], and [18], as well as the findings of this work. In this study, mAlexnet was retrained because the dataset composition of 80% training data and 20% validation data was not employed in the source publications. Table 3 compares the number of parameters involved in the various architectures employed in this study. The number of parameters is compared to see the relationship between the proposed architecture and its computing requirements. Of course, the fewer the parameters, the lower the processing requirements of the CNN. As demonstrated in Table 3, the variation in the normalization function in EfficientParkingNet has no major impact on the number of parameters involved. This indicates that changes in the normalization function have no effect on the addition of computational requirements. These findings suggest that the group normalization function greatly increases accuracy without increasing computational requirements, as shown in Table 3.

Table 3. The number of parameters from various architectures being compared

Methods	Parameters (million)
mAlexnet	0.0326
Mini ShuffleNet	0.024
Yolo+MobileNet	4
EfficientParkingNet + LRN	0.0218
EfficientParkingNet + CrossMapLRN	0.0217
EfficientParkingNet + BN	0.0219
EfficientParkingNet +Group Normalization	0.0219

According to Table 3, Group normalization is the optimal normalization function to use with EfficientParkingNet. The accuracy of EfficientParkingNet is enhanced to 0.67% by group normalization, which is better than prior research in [7, 8] and [18]. According to Table 2, EfficientParkingNet with Group Normalization offers higher accuracy than other approaches without requiring additional processing resources. The likelihood is almost certain that increasing efficiency without increasing computational requirements will keep it quicker than earlier systems.

The findings of this research indicate an improvement in the accuracy of EfficientParkingNet when utilizing group normalization layers. Furthermore, EfficientParkingNet with Group Normalization performs better than previous studies, as evidenced by higher accuracy levels. **This research aligns** with previous research that choosing the right normalization layer can increase CNN classification accuracy. Therefore, it can be concluded that this CNN is a more favorable choice for implementation in smart parking systems.

4. CONCLUSION

The findings of this study reveal that EfficientParkingNet with Group Normalization excels over other normalization functions in terms of accuracy. Importantly, this accuracy improvement is achieved without a significant increase in the number of parameters, ensuring that it has no adverse effects on processing requirements or classification time. Additionally, EfficientParkingNet, with Group Normalization, surpasses the accuracy of mAlexnet, Yolo+Mobilenet, and Carnet in terms of architectural precision. This research significantly contributes to enhancing the accuracy of the existing EfficientParkingNet.Implementing EfficientParkingNet with Group Normalization is expected to yield superior information when incorporated into a smart parking system. The provision of more accurate information to drivers is anticipated to reduce the time required to locate a suitable parking space, thus improving overall efficiency.

Furthermore, this research's experiments underscored the normalization approach's impact on the accuracy level of CNNs. Hence, careful consideration of the normalization approach is essential when designing a CNN architecture. This study's comparison of normalization functions conclusively identifies Group Normalization as the optimal choice for classifying parking lots within the EfficientParkingNet framework. It is noteworthy that both this and previous studies involve the manual labeling of parking space coordinates, making accuracy susceptible to changes in camera angles. Future research endeavors should prioritize the development of automated methods for reading coordinates, mitigating the impact of variations in camera angles, and further advancing the accuracy of parking space classification systems.

5. ACKNOWLEDGEMENTS

The Acknowledgments section is optional. Research sources can be included in this section.

6. DECLARATIONS

AUTHOR CONTIBUTION

Sayuti Rahman was responsible for the conceptualization, methodology, supervision, writing of the original draft, review and editing. Marwan Ramli contributed to the conceptualization and data curation. Arnes Sembiring carried out formal analysis, investigation, and writing review and editing. Muhammad Zen handled the software, validation, resources, visualization, and writing review and editing. Rahmad B.Y Syah was in charge of project administration and writing review and editing.

FUNDING STATEMENT

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

COMPETING INTEREST

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

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, pp. 120–133, jun 2020, https://doi.org/10.1186/s40984-016-0019-x.

[2] A. Kamilaris and F. X. Prenafeta-Boldú, "A review of the use of convolutional neural networks in agriculture," *The Journal of Agricultural Science*, vol. 15, no. 3, pp. 312–322, 2021.

[3] S. P. Singh, L. Wang, S. Gupta, H. Goli, P. Padmanabhan, and B. Gulyás, "3D deep learning on medical images: a review," *Sensors*, vol. 20, no. 18, pp. 5097–5107, 2020.

[4] 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, no. 2, pp. 327–334, apr 2020, https://doi.org/10.1016/j.eswa.2016.10.055.

[5] 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, pp. 1079–1089, feb 2020, https://doi.org/10.3390/app10031079.

[6] S. Nurullayev and S.-W. Lee, "Generalized Parking Occupancy Analysis Based on Dilated Convolutional Neural Network," *Sensors*, vol. 19, no. 2, pp. 277–287, jan 2020, https://doi.org/10.3390/s19020277.

[7] T. Connie, M. K. O. Goh, V. C. Koo, K. T. Murata, and S. Phon-Amnuaisuk, "Improved Parking Space Recognition via Grassmannian Deep Stacking Network with Illumination Correction," in *International Conference on Computational Intelligence in Information System*. Springer, 2021, vol. 17, no. 1, pp. 150–159, https://doi.org/10.1007/978-3-030-68133-3_15.

[8] 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. 1–12, 2022, https://doi.org/10.22034/gjesm.2022.03.02.

[9] H. Naseri and V. Mehrdad, "Novel CNN with investigation on accuracy by modifying stride, padding, kernel size and filter numbers," *Multimedia Tools and Applications*, pp. 1–19, 2023.

[10] T. Takase, S. Oyama, and M. Kurihara, "Effective neural network training with adaptive learning rate based on training loss," *Neural Networks*, vol. 101, no. 1, pp. 68–78, 2020.

[11] O. G. Ajayi and J. Ashi, "Effect of varying training epochs of a Faster Region-Based Convolutional Neural Network on the Accuracy of an Automatic Weed Classification Scheme," *Smart Agricultural Technology*, vol. 3, pp. 100–128, 2023.

[12] X. Yin, W. Chen, X. Wu, and H. Yue, "Fine-tuning and visualization of convolutional neural networks," in *2017 12th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, vol. 1, no. 2. IEEE, 2020, pp. 1310–1315.

[13] A. Ismail, S. A. Ahmad, A. C. Soh, K. Hassan, and H. H. Harith, "Improving convolutional neural network (CNN) architecture (miniVGGNet) with batch normalization and learning rate decay factor for image classification," *International Journal of Integrated Engineering*, vol. 11, no. 4, 2021.

[14] A. Kumar, "Effects of different normalization techniques on the convolutional neural network," in *2021 8th International Conference on Computing for Sustainable Global Development (INDIACom)*, vol. 17, no. 1. IEEE, 2021, pp. 201–204.

[15] E. Chai, M. Pilanci, and B. Murmann, "Separating the effects of batch normalization on cnn training speed and stability using classical adaptive filter theory," in *2020 54th Asilomar Conference on Signals, Systems, and Computers*, vol. 25, no. 20. IEEE, 2020, pp. 1214–1221.

[16] P. R. de 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 2022, https://doi.org/10.1016/j.eswa.2015.02.009.

[17] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” *arXiv preprint arXiv:1607.08022*, vol. 5, no. 1, 2020, https://doi.org/10.48550/arXiv.1607.08022Focustolearnmore.

[18] A. Bianchi, M. R. Vendra, P. Protopapas, and M. Brambilla, “Improving image classification robustness through selective CNN-filters fine-tuning,” *arXiv preprint arXiv:1904.03949*, vol. 2, no. 1, 2020, https://doi.org/10.48550/arXiv.1904.03949.

[This page intentionally left blank.]