Journal of Sensor Science and Technology
Vol. 30, No. 5 (2021) pp. 286-289
http://dx.doi.org/10.46670/JSST.2021.30.5.286
pISSN 1225-5475/eISSN 2093-7563

Lightweight image classifier for CIFAR-10

Akshay Kumar Sharma[1], Amrita Rana[1], and Kyung Ki Kim[1,+]

Abstract

Image classification is one of the fundamental applications of computer vision. It enables a system to identify an object in an image. Recently, image classification applications have broadened their scope from computer applications to edge devices. The convolutional neural network (CNN) is the main class of deep learning neural networks that are widely used in computer tasks, and it delivers high accuracy. However, CNN algorithms use a large number of parameters and incur high computational costs, which hinder their implementation in edge hardware devices. To address this issue, this paper proposes a lightweight image classifier that provides good accuracy while using fewer parameters. The proposed image classifier diverts the input into three paths and utilizes different scales of receptive fields to extract more feature maps while using fewer parameters at the time of training. This results in the development of a model of small size. This model is tested on the CIFAR-10 dataset and achieves an accuracy of 90% using .26M parameters. This is better than the state-of-the-art models, and it can be implemented on edge devices.

Keywords : Computer vision, Convolutional neural networks, Image classification, Lightweight CNN.

1. INTRODUCTION

Computer vision is the field of computer science that empowers computers to identify, detect, and understand various objects in an image or a video. Computer vision replicates the method by which humans observe and understand the world. It has a large variety of applications, including image classification. In addition, it can be considered as a significant problem because it is the basis for other computer-vision-related problems. Traditional machine learning approaches have been used for image classification purposes. However, the outcomes have not effective [1]. Deep learning can produce better results by utilizing large datasets and backpropagation algorithms [2]. Of the various deep neural networks, the convolutional neural network is most prominent because it provides exceptional results in computer-vision-related problems.

Hubel and Wiesel determined that animal visual cortex cells detect light in the small receptive field [3]. Inspired by this, in 1980, Kunihiko Fukushima introduced neocognitron, which is a multi-layered neural network capable of identifying visual patterns hierarchically through learning [4], and is considered the theoretical inspiration for convolutional neural networks (CNNs). In the 1990's, a practical CNN model (called LeNet-5) was introduced by LeCun [5]. The CNN was then used for various visual tasks that facilitate training. However, training data and computing capability were inadequate. Those issues have been resolved because at present, a large number of datasets with good quality images (e.g., CIFAR-10 [6], CIFAR-100 [7], and ImageNet [8]) and substantial computing power are available.

Recently, the use of computer-vision-related work has advanced beyond computer applications to edge devices. This makes the implementation of larger CNN models on these hardware devices more difficult. Most of the CNN algorithms require a large amount of data and high computing power. Hence, cloud servers constitute the most reliable option. It is infeasible to accomplish substantial computations on mobile devices. Considering this, many researchers are focusing on lightweight CNN architectures that can be implemented on hardware devices. Lightweight ConvNets are in high demand, and many researchers are working on this topic. Studies on lightweight networks, such as MobileNet [9] and ShuffleNet [10], reveal the scope for improvement in the construction of more memory-efficient architecture for edge devices.

The convolutional layer in conjunction with batch normalization and the activation function yields better results than other machine learning algorithms. We can obtain highly effective results using

[1] Department of Electronic Engineering, Daegu Universtiy
Daegudaero 201, Gyeongsan, Gyeongbuk 38543, Korea
[+]Corresponding author: kkkim@daegu.ac.kr
(Received: Sep. 12, 2021, Revised: Sep. 20, 2021, Accepted: Sep. 27, 2021)

CNNs. However, the need for less power-consuming devices continues to increase. As a model becomes deeper, the accuracy generally increases. However, the size of the model increases simultaneously, which causes delays in inference time [9]. The focus of this work is to develop a model that can be used for software as well as hardware devices.

The two main functions of CNNs are filtering and combining. A normal convolution network carries out both of these simultaneously. In this study, a lightweight CNN is proposed by using pointwise and depthwise convolution. It provides good accuracy and uses fewer parameters than state-of-the-art models. The pooling layer is also used for the downsampling.

The proposed lightweight model relies on the method proposed by the MobileNet framework [9], which divides the filtering and combining tasks into two layers. This results in the use of a smaller number of parameters and the achievement of an accuracy comparable to that of normal convolution. Fig. 1 shows how depthwise and pointwise convolution performs filtering and combining tasks. Depthwise convolution divides all three input channels and then applies the filter on these individually, as shown in Fig. 1. Pointwise convolution is used for the combining process. An image classification model is proposed by utilizing depthwise and pointwise convolution.

2. Proposed Method

The proposed image classifier comprises two principal blocks: the first and second blocks are named as Main_block and Transition_block, respectively. Main_block divides the input into three paths to extract more features, and Transition_block is majorly responsible for downsampling.

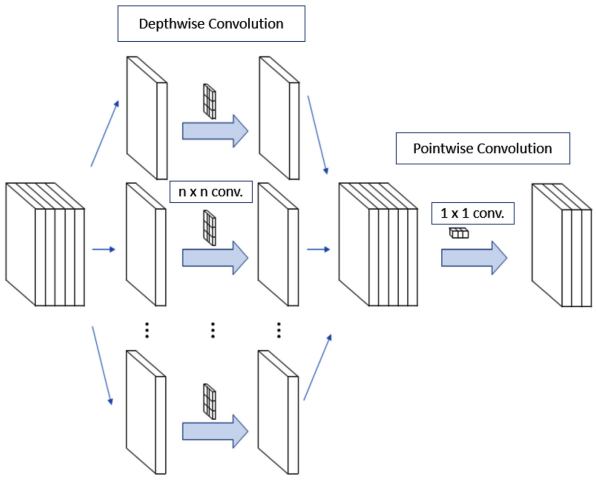

Fig. 1. Depthwise & pointwise convolution [10].

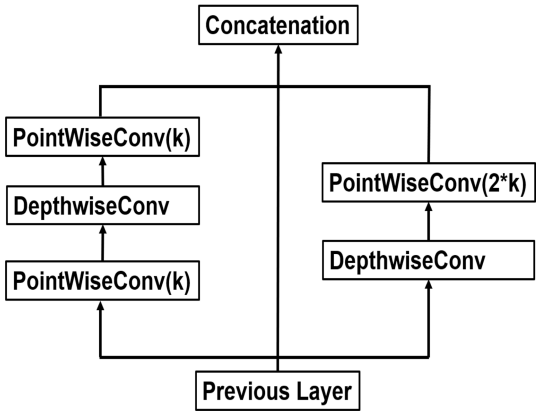

Fig. 2. Main Block of the architecture.

2.1 Main_block

In Fig. 2, Main_block receives the input from the previous layer and divides it into three paths, as follows:

(1) The first path includes a sequence of a pointwise convolutional layer followed by a depthwise and a pointwise convolutional layer. The pointwise layer is used as the first layer in this path to create the projection of feature maps for the next layers.

(2) The second path comprises a depthwise convolutional layer followed by a pointwise convolutional layer.

(3) The third path directly goes to the concatenation layer.

The term "k" used in Main_block determines the number of filters to be used in the specific layer. "k" is set as 64 in the model for CIFAR-10. After passing through the three paths and performing convolution, concatenation is conducted. After concatenation, the output is sent to the transition layer for further processing.

2.2 Transition_block

Transition_block receives input from the previous Main_block and sends it first to a pointwise convolutional layer, then to the average pooling layer (2×2) that is responsible for the downsampling in the architecture (see Fig. 3). The transition layer plays a key role in the architecture and is based on the DenseNet architecture [12].

The output from the transition block goes to Main_block again and repeats the process. The repetition is set to four for this work.

2.3 Pseudo-code

The pseudo-code of the proposed method is shown in Fig. 4.

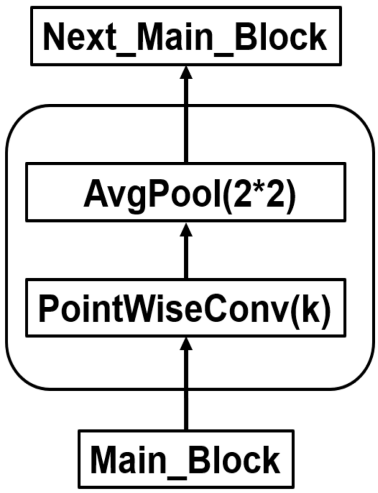

Fig. 3. Transition Block.

```
Input ← Training Labeled Data
repetitions ← number of repetitions
e ← number of epochs
1.  for e = 1:
2.      pointwise_conv(Input)
3.      for reps in repetitions:
4.          m = main_block(reps)
5.          #m runs for given number of reps
6.          t = transition_block(m, reps - 1)
7.          #t runs for given number of (reps-1)
8.  end for
9.  pointwise_conv (t)
10. pooling(pointwise_conv)
11. dropout(pooling)
12. dense wd softmax(dropout)
13. end for
```

Fig. 4. Pseudo-code of the proposed image classifier.

The entire network consists of the initial layer and four repetitions of Main_block and Transition_block. The first layer is a pointwise convolutional layer that is followed by the batch normalization and ReLU activation function layer.

The repetitions of Main_block and Transition_block is set to four. In the fourth repetition, only Main_block is repeated. After the final repetition, a pointwise convolutional layer followed by global average pooling and dropout (0.5) is set. The pseudo-code shown in Fig. 4 explains the entire architecture of the proposed model. All the depthwise and pointwise convolutional layers are followed by the batch normalization and the ReLU activation function.

Table 1. Accuracy & parameter comparison.

Models	No. of Parameters	Accuracy
Lightweight [13]	4.1M	84.25%
MobileNet [Acc. to 13]	4.2M	83.91%
Proposed classifier	.26M	90%

3. RESULTS AND DISCUSSIONS

This section discusses the dataset used for the work and compares the proposed model with the state-of-art models.

3.1 Dataset Used

The dataset used for this work is CIFAR-10 [6]. It consists of 10 classes and contains 50,000 training images and 10,000 testing images. The size of the images is 32 × 32. Because the images are colored, the channel size is three (32 × 32 × 3).

3.2 Performance Analysis

The proposed model is trained on Titan X GPU for 120 epochs. The batch size is set to 64. Adam is used as the optimizer. The number of parameters used for the training is 266,534, which is less than the number in Refs. [9] and [13]. The model is small-sized (3.5 MB). This makes it suitable for implementation on edge devices. The proposed model is compared with two previously constructed models, as shown in Table 1. The three models are tested on the CIFAR-10 dataset. The results show that the proposed model uses very few parameters and also provides good accuracy compared with the other works.

4. CONCLUSION

In this study, a lightweight image classifier is proposed using depthwise and pointwise convolution. To assess the efficacy of the model, it is tested on the CIFAR-10 dataset. The result shows that the proposed model yields better results in terms of the number of parameters and accuracy. There is a substantial difference in the number of parameters used for training the model on CIFAR-10: the proposed method uses. 26M parameters. The size of the model

is also very small (3.5 MB). This renders it preferable for implementation in edge devices. The proposed classifier can be used as the base network for the object detection model. In future work, the proposed classifier would be implemented on larger datasets and on hardware devices.

ACKNOWLEDGMENT

This research was supported by the Ministry of Science and ICT (MSIT), Korea, under the Information Technology Research Center (ITRC) support program (IITP-2021-0-02052) supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP).

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (2020-0-01080, Variable-precision deep learning processor technology for high-speed multiple object tracking).

REFERENCES

[1] F. Sultana, A. Sufian, and P. Dutta, "Advancements in image classification using convolutional neural network", *Proc. of IEEE 2018 Fourth Int. Conf. on Res. Comput. Intell. Commun. Netw.,* pp. 122-129, 2018.

[2] R. Hecht-Nielsen, "Theory of the backpropagation neural network", *Proc. of IEEE IJCNN*, pp. 593-605, San Diego, CA, 1989.

[3] D. H. Hubel and T. N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex", *J. Physiol.*, Vol. 195, No. 1, pp. 215-243, 1968.

[4] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position", *Biol. Cybern.*, Vol. 36, No. 4, pp. 193–202, 1980.

[5] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition", *Proc. of IEEE*, Vol. 86, No. 11, pp. 2278–2324, 1998.

[6] https://www.cs.toronto.edu/~kriz/cifar.html (retrieved on Aug. 25, 2021).

[7] J. Deng, W. Dong, R. Socher, L. J. Li, K. Li, and F. F. Li, "Imagenet: A large-scale hierarchical image database", *Proc. of IEEE Conf. on Comput. Vis. Pattern Recognit.*, pp. 248-255, Miami, Florida, 2009.

[8] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications", *Proc. Conf. on Comput. Vis. Pattern Recognit.*, pp. 1704.04861(1)-1704.04861(9), Honolulu, Hawaii 2017.

[9] X. Zhang, X. Zhou, M. Lin, and J. Sun. "ShuffleNet: An extremely efficient convolutional neural network for mobile devices", *Proc. of IEEE Conf. on Comput. Vis. Pattern Recognit.*, pp. 6848-6856, Salt Lake City, Utah, 2018.

[10] I. N. Junejo and N. Ahmed, "Depthwise separable convolutional neural networks for pedestrian attribute recognition", *SN Comput. Sci.*, Vol. 2, No. 2, pp. 1-11, 2021.

[11] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks", *Proc. of IEEE Conf. on Comput. Vis. Pattern Recognit.*, pp. 4700-4708, Honolulu, Hawaii, 2017.

[12] W. Sun, X. Zhang, and X. He, "Lightweight image classifier using dilated and depthwise separable convolutions", *J. Cloud Comp.,* Vol. 9, No. 1, pp. 1-12, 2020.