MobileNetV2 Model for Image Classification

Ke Dong[1st]
School of Computer Science and Information
Hefei University of Technology
Hefei, China
coliapaston@163.com

Chengjie Zhou[1st]
College of Letters and Science
University of California, Los Angeles
Los Angeles, CA
*Charles2secret@g.ucla.edu

Yihan Ruan[1st]
The Grainger College of Engineering
University of Illinois at Urbana Champaign
Champaign, IL
yihan2@illinois.edu

Yuzhi Li[1st]
Faculty of Electronic and Information Engineering
Xi'an Jiaotong University
Xi'an, China
nidhogg@stu.xjtu.edu.cn

[1st]These authors contributed equally.

Abstract—**Machine learning has been increasingly prevailing all over the world, especially in the computer vision field. This paper mainly focused on the performance of MobileNetV2 model for image classification. To verify the advanced performance of MobileNetV2 model better, this paper adopted MobileNetV1 model as the control group and introduced an experiment of identifying images in a variety of datasets extracted from TensorFlow. With the T-SNE visualization tool, the conclusion can be generated by comparing the accuracy and effectiveness of these two models. The experimental results demonstrated that the proficiency of MobileNetV2 model achieved higher accuracy rates compared to MobileNetV1 model. In order to enhance the performance of MobileNetV2, extensive experiments are performed.**

Keywords- MobileNetV2; MobileNetV1Image; Classification; T-SNE

I. Introduction

With the application of machine learning and especially convolutional neural networks (CNNs), image classification has shown great potential in disease verification [1], face recognition [2], and vehicle detection [3]. The ability and advantages of pre-trained CNN models are mainly due to the parameters, which are trained on the dataset with larger samples. Compared with training a new model from scratch, the computational costs can be decreased by using pre-trained models. In the previous works, Howard et al. [4] verified that the MobileNets model is efficient for mobile, embedded vision, and other applications. However, Howard has not adequately demonstrated the performance of the MobileNets model in image classification. To remedy this issue, we tested and analyzed MobileNetV2 model on two datasets from TensorFlow (Colorectal_histology and Eurosat [5]). The results demonstrated that MobileNetV2 model achieved higher accuracy and shorter training duration than other models, such as MobileNetV1, Xception [6], Inception-ResNetV2, and ResNet152.

Taken together, the key contributions of this paper can be summarized as follows:

(1) This paper compared the performance of different models on two datasets and utilized saliency maps to visualize the outputs.

(2) The experimental result demonstrated that the highest accuracy of training datasets is achieved by MobileNetV2 model when performing image classification missions.

(3) After analyzing the experimental result, MobileNetV2 model provides assistance in future applications of the image classification field.

The rest of this paper is organized as follows: Section 2 reviewes related techniques used in the paper. Section 3 describes and analyzes two convolutional neural network models, MobileNetV1 and its improved version MobileNetV2. Section 4 reports the experimental steps we took in comparing and contrasting the performance of MobilNetV2 and other models when handling image classification. Finally, section 5 summarizes the experiment's results and proposes future directions for contemporary researchers.

II. Related Work

T-distributed stochastic neighbor embedding (t-SNE) is non-linear technique for dimensionality reduction, and this technique can be used to visualize high-dimensional data successfully. T-SNE can map the multi-dimensional data to a relatively low dimensional space. The algorithm for T-SNE can be mainly divided into two steps. First of all, researchers need to create a probability distribution demonstrating relationships between neighboring points. Then, they recreate a lower-dimensional space based on the probability distribution generated from higher dimensions. The recent studies demonstrated that T-SNE is able to handle crowding problems in a great performance, which basically comes from the curse of dimensionality.

Another conception related to this article is the Convolutional Neural Network (CNN). CNN is a typical class of deep neural networks, which is supervised learning and trained through backpropagation algorithms. CNNs are commonly used for computer vision by learning spatial hierarchies of features. CNNs are generally composed of following layers: convolution layers, pooling layers, and fully

978-0-7381-1141-4/20/$31.00 ©2020 IEEE
DOI 10.1109/ITCA52113.2020.00106

connected layers. Convolution layers use filters that can perform convolution operations to scan the input images. Pooling layers are commonly used to preserve detected features or downsample feature maps. In the end, fully connected layers collect results from previous layers and classify images with labels based on those results.

III. MobileNet Models

A. MobileNet V1

As a small and low latency model with efficient network architecture and hyper-parameters, Mobilenet v1 was proposed by A. G. Howard et al. in 2017 [7]. Instead of only focusing on size, Mobilenet v1 is primarily based on depthwise separable convolutions and a set of two hyper-parameters. As a class of lightweight networks, MobileNetV1 allows people to specifically choose a small network for individual applications matching the resource restrictions (latency and size). Mobilenet v1 is often used in monitoring systems [8], image classification [9], and augmented reality [10].

Depthwise separable convolutions were first proposed in image classification problems [11]. The depthwise separable convolutions were subsequently utilized in Inception models [12] to reduce computation cost. Besides, there are many other networks that employ similar structures to optimize the network structure and reduce the amount of computation. Shrinking, decomposing, or compressing pre-trained networks is another way to obtain small networks, i.e., compression based on product quantization [13] and hash I [14], pruning, vector quantization and Huffman coding [15], distillation [16], etc. This approach utilizes larger networks to train smaller networks. The calculation of depthwise separable convolutions can be summarized as follows:

1) Depthwise Convolution. Depthwise

convolution refers to convolution that does not cross channels, which means each channel of the feature map has an independent convolution kernel only acting on. The calculation cost of the depthwise convolution also can be $1/N$ of the traditional convolution:

$$D_k \cdot D_k \cdot W \cdot D_W \cdot D_H \tag{1}$$

2) Pointwise convolution.

Although the operation of depthwise convolution is very efficient, the depthwise convolution is only equivalent to applying a filter to a channel of the current FeatureMap, instead of merging several features to generate new features. As the FeatureMap is the output of Depthwise convolution, the number of channels is equal to the number of channels input to FeatureMap. The ability of Depthwise convolution cannot increase or reduce the dimensionality.

In order to solve these problems, Pointwise convolution was introduced in MobileNetV1 for feature merging and dimension alternating. Naturally, using convolution to complete this function can be considered. The number of parameters of Pointwise is , and the amount of calculation can be denoted as:

$$M \cdot N \cdot D_W \cdot D_H \tag{2}$$

3) Depthwise Separable Convolution.

Depthwise Separable convolution is a set of operations obtained by combining a 3×3 Depthwise convolution and a Pointwise convolution. Compared with a 3×3 convolution, the parameter amount and calculation cost of MobileNetV1 are about the $1/8$ of ordinary convolution.

4) Hyper-parameters.

Width multiplier and resolution multiplier are two hyper-parameters utilized in MobileNetV1. The role of the width multiplier α is to thin a network uniformly at each layer. Resolution multiplier ρ is applied to the input image, while the internal representation of each layer is subsequently reduced by the same multiplier. The computational cost for the core layer of MobileNetV1 can be expressed as the following form:

$$D_k \cdot D_k \cdot \alpha M \cdot \rho D_F \cdot \rho D_F + \alpha M \cdot \alpha N \cdot \rho D_F \cdot \rho D_F \tag{3}$$

The structure of Mobilenet v1 is established on depthwise separable convolutions except for the first layer, which is a full convolution. For a MobileNetV1 with 28 layers, all layers are followed by a batchnorm and ReLU non-linearity with the exception of the final fully connected layer, which is followed by no non-linearity and feeds into a softmax layer for classification [7].

Although MobileNetV1 could achieve certain promising performance, it still has some problems, such as the vanishing gradient problem. To solve this problem, MobileNetV2 was discussed in detail in the subsequent section.

B. MobileNet V2

This work employed the MobileNetV2 model for image classification and focused on the model's portability. The main structure of MobileNetV2 is based on its previous version -- MobileNetV1. MobileNetV2 applies the technique of Depthwise Separable Convolutions (DSC) for portability and not only improved the problem of information destroying in non-linear layers in convolution blocks by using Linear Bottlenecks, but also introduced a new structure named Inverted residuals to preserve the information.

1) Depthwise Separable Convolutions.

The Depthwise Separable Convolutions used in MobileNetV1 have also been applied in MobileNetV2. Combining with the Depthwise convolution and the Pointwise convolution, the total number of parameters and computational cost can be reduced to about 18 of the ordinary convolution.

2) Linear Bottlenecks.

There are two properties that inspire the idea of Linear Bottlenecks. On the one hand, it is clear that if a result of a layer has the form of ReLU (Bx) and the result remains non-zero, the corresponding part of input space (x) is limited to a linear transformation (Bx). In other words, the ability of the deep networks is limited to a linear classifier on the non-zero volume part of the output domain. On the other hand, when ReLU collapses a channel, it inevitably loses information in that channel. However, if there are lots of channels, and there is a structure in the set of activation functions in which information can still be preserved in other channels. Here this kind of structure turns out to be a layer that can embed the input into a lower-dimensional subspace of the activation space. The discussions above seem to show that if the input space can be embedded into low-dimensional when using ReLU as an activation function, a linear transformation is enough to hold all the useful information. To this end, linear bottleneck layers can be inserted into the convolution blocks.

TABLE I. BOTTLENECK RESIDUAL BLOCK TRANSFORMING FROM K TO K' CHANNELS, WITH STRIDES, AND EXPANSION FACTOR T

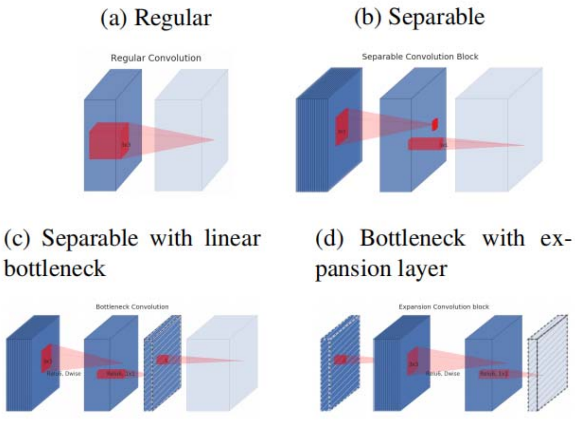
Input	Operator	Output
h × w × k	1x1 conv2d , ReLU6	h × w × (tk)
h × w × tk	3x3 dwise s=s, ReLU6	hs × ws × (tk)
hs × ws × (tk)	linear 1x1 conv2d	hs × ws × k'

3) *Inverted Residuals.*

The bottleneck blocks are similar to residual blocks, where each block contains an input followed by several bottlenecks and then followed by expansion [17]. The shortcuts can be added between different bottlenecks to improve the gradient propagate ability in multiplier layers is mainly due to the two factors/aspects: (1) The bottlenecks consist of almost all the information. (2) The expansion layer can be regarded as an implementation detail with a non-linear transformation of the tensor. And using the inverted design (Inverted Residual) is more memory efficient compared to the traditional structure.

4) *Model Architecture.*

The basic building block is a bottleneck depthwise separable convolution with residuals. The detailed structure of a sample DSC block is shown in Figure 1. By taking advantage of transfer learning, MobileNetV2 can be deemed the head, and some layers follow behind for specific classification tasks. Here the project employs the version of MobileNetV2 whose input is 160 × 160 RGB picture. The model first expands the low-dimensional compressed representation of the input to high dimension and filters it with a lightweight depthwise convolution. Features are subsequently projected back to a low-dimensional representation with a linear convolution. This kind of structure in the model can preserve the information, solve the inflexible number of filters in MobileNetV1, and keep the block lightweight simultaneously.



(a) Regular

(b) Separable

(c) Separable with linear bottleneck

(d) Bottleneck with expansion layer

Figure 1. Evolution of separable convolution blocks. The diagonally hatched texture indicates layers that do not contain non-linearities. The last (lightly colored) layer indicates the beginning of the next block. Note: 2d and 2c are equivalent blocks when stacked.

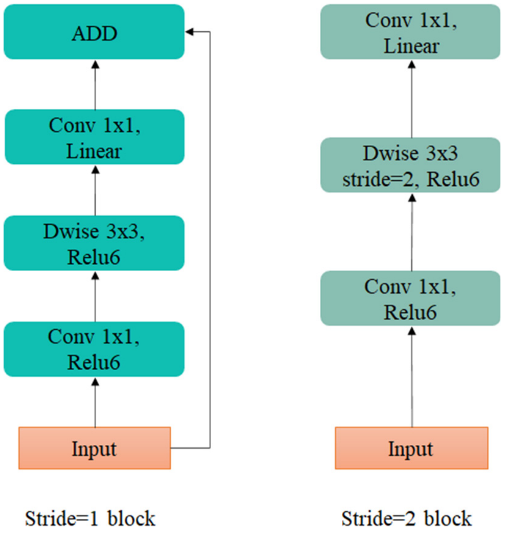


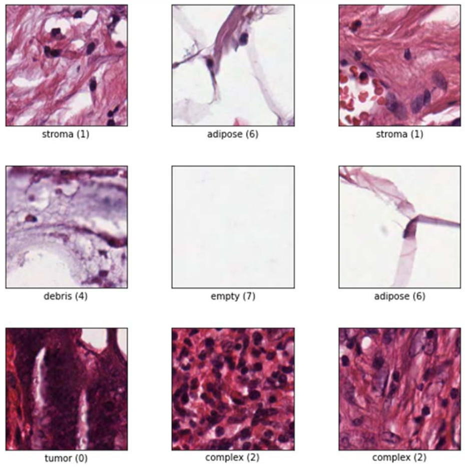
Figure 2. The convolutional blocks in MobileNetV2.

IV. EXPERIMENT RESULT

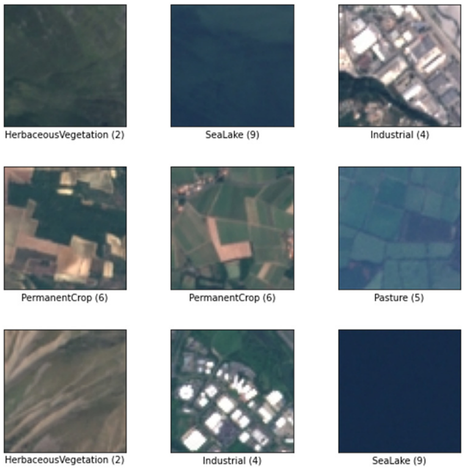
The datasets utilized in this work come from the TensorFlow datasets. Colorectal_histology is a classification of textures in colorectal cancer histology. Each example is composed of a 150 x 150 x 3 RGB image of one of 8 classes (see Figure 3.a). And eurosat is based on Sentinel-2 satellite images covering 13 spectral bands and consisting of 10 classes with 27000 labeled and geo-referenced samples (see Figure 3.b). The detailed information is in Table 2.

TABLE II. THE DATASETS USED IN THE EXPERIMENT

Name	Samples	Classes	Size
colorectal_histology	5000	8	150×150×3
eurosat/rgb	27000	10	64×64×3



(a)



(b)

Figure 3. (a) Examples of colorectal_histology dataset. (b) Examples of eurosat dataset.

In the experiment, the accuracy of the validation set is used for the evaluation of the performance of a model. To specify the advantage of MobileNets, a CNN with a typical and traditional structure built manually is applied as a comparison, and it is trained for more epochs to narrow the gap between untrained and pretrained parameters. The information about the networks in the experiment can be found in Table 3. Also, to make the result more direct and clear, the training process and the change of the loss and accuracy during training are plotted in Figure 4 and Figure 5.

TABLE III. . INFORMATION AND FEATURES OF THE NEURAL NETWORKS IN THE EXPERIMENTS

Name	Total Parameters	Trainable Parameters	Epoch
Simple_CNN	670,056	670,056	50
MobileNetV1	3,369,738	140,490	10
MobileNetV2	2,431,496	173,128	10

TABLE IV. THE COMPARISON OF PERFORMANCE ON COLORECTAL CANCER

Name	*Performance (Test Set Accuracy)*	*Training time*
Simple_CNN	82.88 %	400s
MobileNetV1	88.60 %	72s
MobileNetV2	89.00 %	75s

V. CONCLUSION

This article discusses the experiment implementing the MobielNetV1 model and MobileNetv2 model with multiple datasets for image classification and corresponding results. By comparing and analyzing the experimental data, it can be found that the MobileNetV2 model achieves a higher accuracy rate than the MobileNetV1 model. Therefore, MobileNetV2 is able to be considered as an appropriate technique to address image classification problems.

In the future, image classification can be improved based on the current work by expanding the dataset. The larger dataset can eliminate errors to the largest extent because the majority of possibilities will be tested based on the large dataset, which could help researchers develop a more competitive model. In addition, implementing some derivative functions might enhance the performance of the MobileNetV2 model for image classification, for example, segmentation algorithm and object detection.

REFERENCES

[1] S. Srdjan, et al. "Deep neural networks based recognition of plant diseases by leaf image classification." Computational intelligence and neuroscience 2016 (2016).

[2] G. Shenghua, I. Wai-Hung Tsang, and L. Chia. "Kernel sparse representation for image classification and face recognition." European conference on computer vision. Springer, Berlin, Heidelberg, 2010.

[3] C. Xueyun, et al. "Vehicle detection in satellite images by hybrid deep convolutional neural networks." IEEE Geoscience and remote sensing letters 11.10 (2014): 1797-1801.

[4] A. G. Howard, G. Andrew, et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications", arXiv preprint arXiv:1704.04861, 2017.

[5] P. Helbe, B. Bischke, et al. "EuroSAT: A Novel Dataset and Deep Learning Benchmark for Land Use and Land Cover Classification", arXiv preprint arXiv:1709.00029, 2019.

[6] C. François. "Xception: Deep learning with depthwise separable convolutions." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[7] A. G. Howard, G. Andrew, et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications", arXiv preprint arXiv:1704.04861, 2017.

[8] W. Kim, W. S. Jung, H. K. Hyun, "Lightweight driver monitoring system based on multi-task mobilenets", Sensors , vol.19, no.14, pp. 3200, 2019.

[9] N. R. Gavai, "MobileNets for flower classification using TensorFlow", 2017 International Conference on Big Data, IoT and Data Science (BID), IEEE, 2017.

[10] G. K. Upadhyay, et al. "Augmented Reality and Machine Learning based Product Identification in Retail using Vuforia and MobileNets", 2020 International Conference on Inventive Computation Technologies (ICICT). IEEE, 2020.

[11] L. Sifre. S. Mallat, "Rigid-motion scattering for image classification", PhD thesis, vol.1, no.3, 2014.

[12] S. Ioffe, C. Szegedy. "Batch normalization: Accelerating deep network training by reducing internal covariate shift", arXiv preprint arXiv:1502.03167, vol.1, no.3, 2015.

[13] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices", In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4820-4828, 2016.

[14] W. Chen, J. T. Wilson, S. Tyree, K. Q. Weinberger, and Y. Chen. "Compressing neural networks with the hashing trick", International conference on machine learning, pp.2285-2294, 2015.

[15] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. CoRR, abs/1510.00149, 2, 2015. 2

[16] G. Hinton, O. Vinyals, J. Dean, "Distilling the knowledge in a neural network", arXiv preprint arXiv:1503.02531, vol.2, no.7, 2015.

[17] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (references)

[18] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[19] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350

[20] K. Elissa, “Title of paper if known,” unpublished.

[21] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[22] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[23] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.