International Core Journal of Engineering Volume 7 Issue 3, 2021
ISSN: 2414-1895 DOI: 10.6919/ICJE.202103_7(3).0043

MobileNet and EfficientNet Demonstration on Google Landmark Recognition Dataset

Kangrui Wang[1,*], Xiaobing Yu[2]

[1]Shanghai Jiao Tong University, Shanghai, 200240, China;

[2]Miami University, Oxford, 45056, USA.

*nightinx@sjtu.edu.cn

Abstract

In image processing, the convolutional neural net- work extracts better features than the previous manual features because of the special structure of the CNN. The combination of convolution and pooling layers enables the CNN to extract better features in the image.There are many network models of convolution neural networks, but a convolution neural network model generally consists of several convolution layers, pooling layers and full connection layers. In this paper, we will discuss some newest models, MobileNet and EfficientNet, using Google Landmark Recognition's dataset to visualize these two CNN models' performance on image classification by comparing their accuracy and public score.

Keywords

Efficientnet; MobileNet; Google Landmark Recognition; CNN.

1. Introduction

Image classification technology has made significant progress in the past few years. For example, in the Imagenet Classification Challenge, the error rate has dropped significantly every year. In order to continue to advance the development of computer vision, many researchers are now focusing more on fine and instance-level recognition problems, and many people are designing machine learning algorithms that can recognize the Eiffel Tower, Mount Fuji or Persian cats instead of recognizing buildings and general entities such as mountains and cats. However, a relatively large research obstacle in this field is the lack of large labeled datasets. This year, Google updated the largest artificial and natural land-mark recognition data set, and released Google-Landmarks- v2. Given a landmark image, querying similar images in the dataset is this year's major task, which is common image recognition [1]. The difficulty lies in that the dataset is very large and noisy. Google Landmarks Dataset v2 (GLD2) contains nearly 5 million images, 4132914 images in the training set, 20,094 categories, 761757 images in the index set, and 117577 images in the test set [2]. This dataset is the landmark image data obtained by network data mining, so it contains a lot of noisy annotations. In addition, the data set has a version CGLD2 that was cleaned up by the smlyaka team in the previous competition using an automatic cleanup tool.

In the training set, it contains 1,580,470 images and 81,313 categories. At the same time, it contains a large amount of non- landmark data, which is in line with the actual situation and is very challenging. Based on this data set, this year attracted more than 500 teams from around the world to participate in the landmark retrieval and recognition competition hosted by Google. Many Kagglers are familiar with image classification challenges like the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), which aims to recognize 1K general object categories. Landmark recognition is a little different from that: it contains a much larger number of classes (there are more than 81K classes in

ISSN: 2414-1895 DOI: 10.6919/ICJE.202103_7(3).0043

this challenge), and the number of training examples per class may not be very large. This feature makes this year's competition remarkably challenging. Landmark recognition has some significant differences from other issues. For example, even in a large labeled dataset, there may not be a lot of training data for some unknown landmarks. A specific example is shown in Fig. 1[2].

(a) Images corresponding to label 1 (b) Images corresponding to label 7

Fig. 1. Example of unknown landmarks[2]

In addition, since landmarks are usually immovable rigid objects, the changes within the category are very small (in other words, the appearance of the landmark does not change much in its different images). Therefore, the changes only come from the photographing conditions such as occlusion, different viewing angles, weather and lighting, which makes landmarks different from other image recognition datasets. Another feature of this dataset is that there are some noisy images in both train dataset and test dataset like Fig. 2[2]. We will set the random stuff, and process all images with the same methods because we want to test how efficient these two models can be when applying in this complicated dataset.

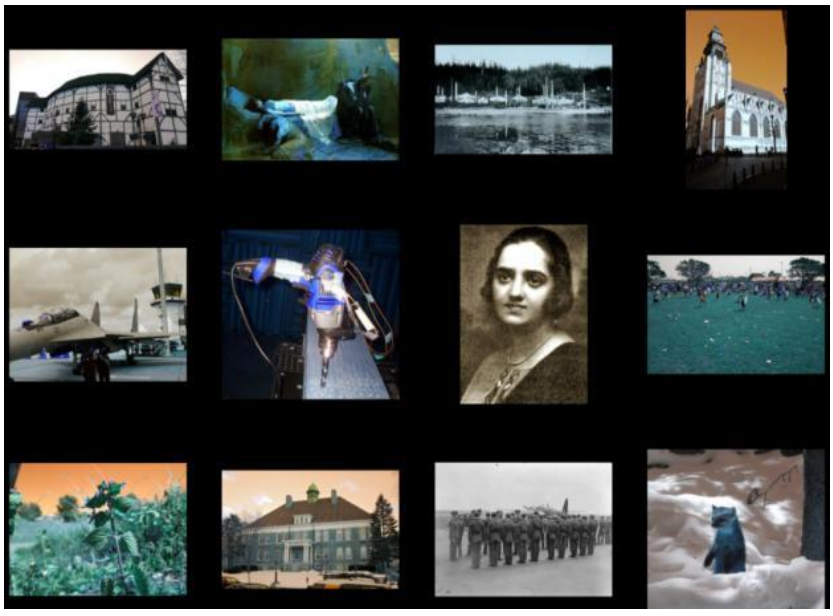


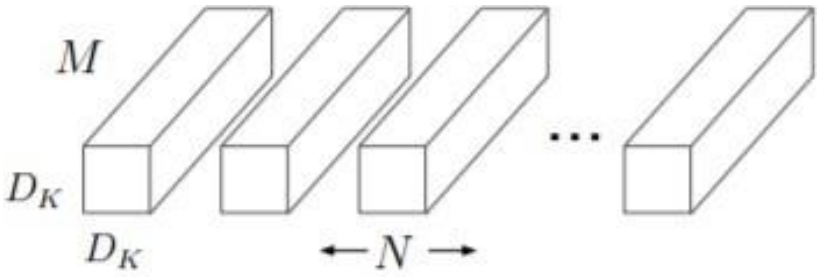
Fig. 2. Disturbing images[2]

ISSN: 2414-1895 DOI: 10.6919/ICJE.202103_7(3).0043

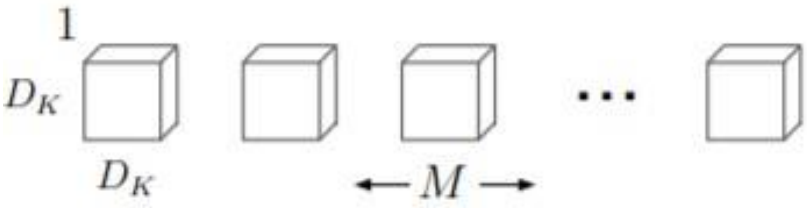
Since this competition has its own grading policy, which is called public score, we will directly use it to compare their performance by using the same method of data cleaning. By implementing different models with different batch sizes and epoches, we will get different public scores by submitting our final result. We will also compare the precision and recall value of these two models in an orthdox way.

2. Mobilenet

The basic unit of MobileNet is depthwise separable convolu- tion. In fact, this structure has been used in the Inception model before. Depth-level separable convolution is actually a factor- ized convolution operation, which can be decomposed into two smaller operations: depthwise convolution and pointwise convolution, as shown in Fig. 3 [3]. Depthwise convolution is different from standard convolution. For standard convolution, the convolution kernel is used on all input channels, while depthwise convolution uses different convolution kernels for each input channel, which is a convolution kernel corre- sponding to an input channel, so depthwise convolution is a depth-level operation. The pointwise convolution is actually an ordinary convolution, but it uses a 1x1 convolution kernel. Fig. 4 shows the two operations more clearly. For depthwise separable convolution, first this method uses depthwise con- volution to convolve different input channels separately, and then uses pointwise convolution to combine the above outputs, so that the overall effect is similar to a standard convolution, but it will greatly reduce the calculation quantity and model parameter quantity. We believe this model will be efficient since Google Landmark dataset has over 81k classes.



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters

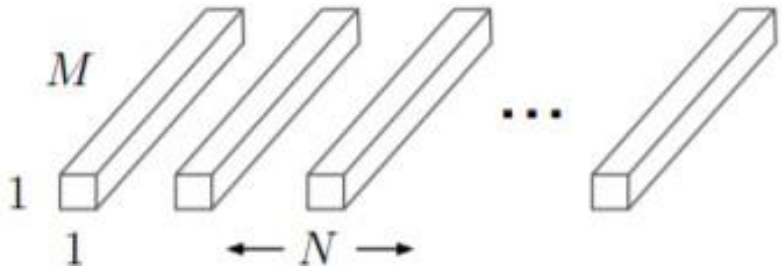


Fig. 3. MobileNet Structure(1)[3]

ISSN: 2414-1895 DOI: 10.6919/ICJE.202103_7(3).0043

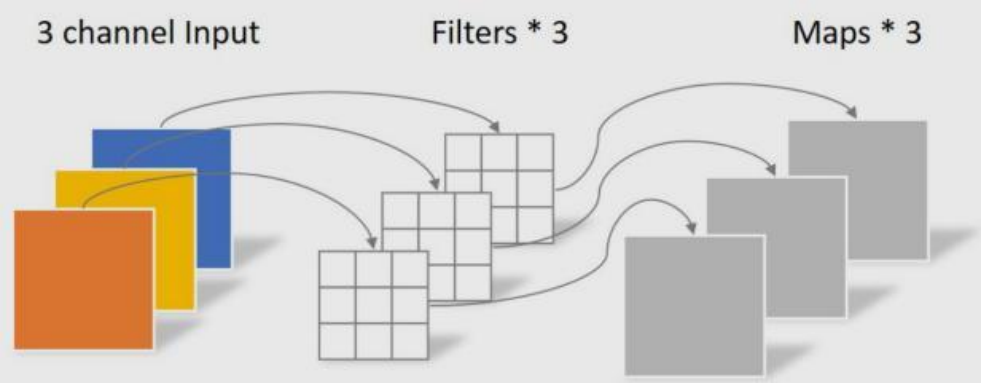


Fig. 4. MobileNet Structure (2)

The characteristic of MobileNet is that inference is very fast [4]. Take SSD as an example, if MobileNet is used as a backbone, the speed is about 3 times that of VGG as a backbone, so it is a network that can be landed on the mobile side. But everything has two sides. Under the same data set, such as the open image data set released this year, after 100- 200 epochs in many categories, the map of mobilenet is usually about 20 points lower than VGG.

(1) Tasks done by MobileNet are equivalent to splitting the convolution kernel. For example, if a layer requires 33 convolution kernels with 256 channels, we want to split them into 33 convolution kernels with 1 channel and 11 convolution kernels with 256 channels. After such splitting, the parameter amount can be reduced to one-ninth of the original. This is why the amount of parameters, and the size of the model becomes smaller.

(2) The point-by-point convolution is the convolution of 11 mentioned in (1), since in the neural network, the number of channels can usually reach 64 or 256, so basically all the parameters are concentrated in the point-by-point convolution, of course the amount of calculation are all concentrated in this part. This amount of calculation accounts for about 94% of the total, and the parameters account for about 75% of the total [5]. The purpose of this MobileNet is to apply knowledge learned from one environment to a new environment. The knowledge learned by the neural network is mainly reflected in the weight parameters obtained by training on a specific task, so the essence of migration learning is the migration of weights. Traditional machine learning requires training data and test data to have the same distribution, and it is impossible to obtain enough valid samples. A large amount of sample data needs to be relabeled to meet the demand. Using the migration learning method can solve the above two problems, allowing the learned and existing knowledge to be applied to new fields with only a small amount of sample data. This means that for an existing model that has been trained, and retraining its last layer or layers of network, you can get a better network model without too long training time.

3. Efficientnet

Being a new technology of model scaling for Convolu- tional Neural Networtks (ConvNets), efficeintnet has attracted research attention since 2019 after Mingxing Tan and Quoc V.Le from Google Brain introduced the new concept [5]. Convolutional Neural Networks are usually developed under a fixed resource budget. If more resources are available, we can scale up the ConvNets to obtain a better accuracy. For example, the depth, width and resolution of the network can be increased. However, it is very difficult to manually decide the scale-up-coefficient of the depth, width, and resolution. Based on the above background, Efficientnet is proposed as a new model scaling method, which uses a simple and efficient composite coefficient to scale up the network from the three dimensions. And based on the neural structure search technol- ogy, an optimal set of parameters (composite coefficients) can be obtained.

There are three dimensions for Convolutional Neural Net- works to implement scale up strategy:

ISSN: 2414-1895 DOI: 10.6919/ICJE.202103_7(3).0043

Width: As shown in Fig. 5. (b), by increasing the channel dimension of the baseline, the network scale can be scaled up. A wider network can capture more fine-grained features and make the training process easier.

Depth: As shown in Fig. 5. (c), the number of convolutional layers is increased. A deeper network can capture richer and more complex features, but due to gradient dispersion, problems such as gradient explosion make the training process more difficult.

Resolution: As shown in Fig. 5. (d) above, by increasing the resolution of the baseline input image, the resolution network can capture more fine-grained features.

If the resolution of the image is larger, a deeper network is needed to improve the receptive field, and more channels are needed to capture more fine-grained features. Therefore, a fixed ratio to width, Depth, and resolution are scaled up at the same time to achieve a balance of the three in a convolutional neural network, as shown Fig. 5. (e), which is exactly the scale up strategy of EfficeintNet.

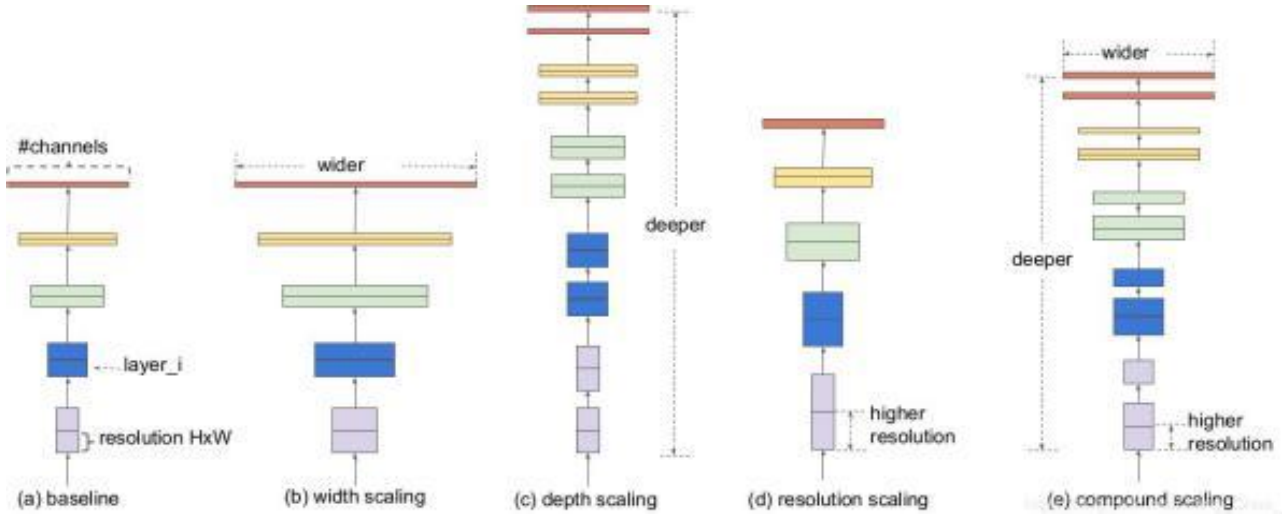


Fig. 5. Scale up straregy[5]

4. Experiment

We conduct experiments between MobileNet and Efficient- Net on Google Landmark Recognition 2020 data set in this section.

4.1 EfficientNet

On the pre-training process, we implement the pretrained model on ImageNet. Firstly, we train our model on CGLD2, which is a cleaned version of the landmark data set. Then, we use the data in GLD2, a data set with many disturbing pictures, to train our model. We filter out labels having samples less than a minimum threshhold. We use Global Average Precision as our competetion's metric [6]. The EfficientNet-B0 is the main version we implemented on this dataset, the structrue of which is given in Fig. 6. For the test data, We select only one sample for the classes with samples more than three in order to increase the diversty of the test set. We run our experiment on kaggle with TPU v3-8.

4.2 MobileNet

We first import Mobile Net from tensor flow. keras, and set the weight equals to ImageNet on the pretrained model. We decided to apply Mobile Net V2 in this dataset because this version is widely used and extremely suitable for this dataset. Since we already have cleaned GLD2 which is mentioned in the introduction part, we just apply the same method in the Efficient Net process which only deals with the labelled sample with minimum threshold to make sure that we process two models in the same way, and we also apply GAP as the standard rubric. The process of Mobile Net V2 is shown in Fig. 7[3].

ISSN: 2414-1895 DOI: 10.6919/ICJE.202103_7(3).0043

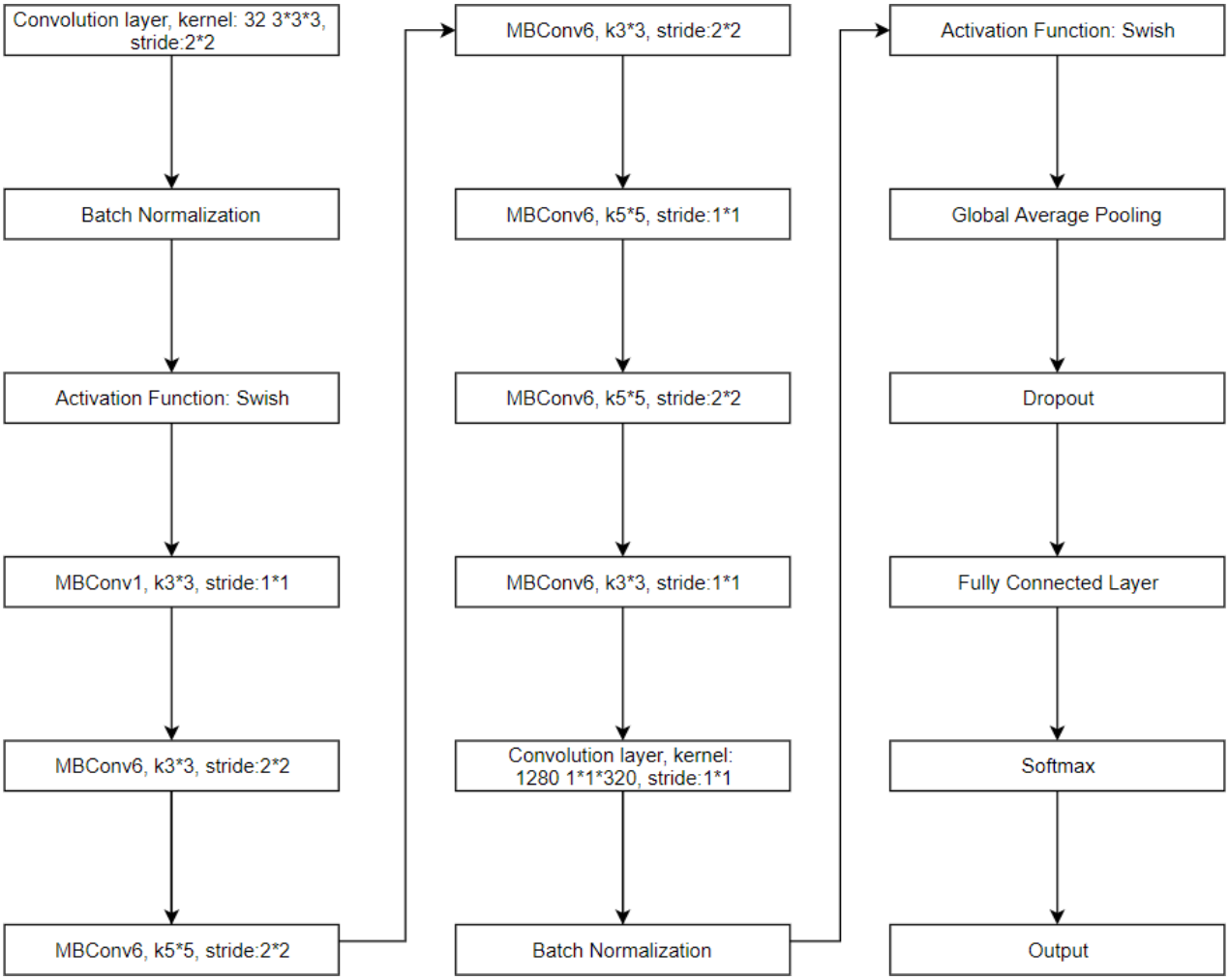


Fig. 6. Structrue of Efficientnet implementation

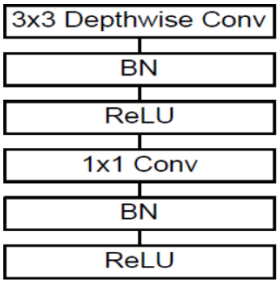


Fig. 7. Structrue of MobileNet implementation[3]

4.3 Experimental Results and Analyses

The result of our experiment is given in Table 1. We compare the two models with ResNet, which is used in the best solution to the Google Landmark Recognition 2019. We measure the accuracy of each model by selecting about 100 iamges as test inputs. And we get the predicted probabilities for each class. We weight average the possibilities which the test images really belong to and caculate the accuracy. From the table we can easily see that EfficientNet performs the best among the three. It takes the least time to gain the highest GAP and accuracy. However, the performance of the MobileNet is disappointing, the accuracy of which is relatively. The result has proved the significance of EfficientNet. It has surpassed the two other models in terms of accuracy and speed. Considering the amount of parameters in EfficientNet, we believe that it may become a new foundation of computer vision in the future.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202103_7(3).0043

Table 1. Experiment result

Items / Model	Runtime	Accuracy	GAP
EfficientNet	1.5h	0.77	0.47
MobileNet	2.6h	0.36	0.24
ResNet	2.4h	0.67	0.47

5. Conclusion

In this paper, we test and compare the performance of two newest CNN models, EfficientNet and MobileNet, using Google Landmark Recognition Dataset. We introduce the structure and good features of these two models. We propose some innovative data processing methods to improve the efficiency and accuracy in the training process. We compare the perforamance of EfficeintNet, ResNet and MobileNet in the experiment section. The experiment result shows that EfficientNet is a powerful weapon in the field of computer vision.

Acknowledgements

Kangrui Wang and Xiaochun Yu contributed equally to this work and should be considered co-first authors.

References

[1] https://github.com/google/youtube-8m/blob/master/average precision calculator.py

[2] https://www.kaggle.com/c/landmark-recognition-2020..

[3] Howard, Andrew Zhu, Menglong Chen, Bo Kalenichenko, Dmitry Wang, Weijun Weyand, Tobias Andreetto, Marco Adam, Hartwig. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.

[4] Sinha, Debjyoti El-Sharkawy, Mohamed. (2019). Thin MobileNet: An Enhanced MobileNet Architecture.

[5] Mingxing Tan, Quoc V. Le. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.

[6] https://evaluations.readthedocs.io/en/latest/kaggle 2020/global average precision.html.