Investigations on Convolutional Neural Network Based on Image Recognition

Haolin Dong *

Department of Intelligence Science and Technology, Beijing Information Science & Technology University, Beijing, China

* Corresponding Author Email: haolindong1021@gmail.com

Abstract. Many convolutional neural networks have emerged over the years, varying in accuracy, speed, and architecture. From LeNet-5 to GoogleNet, the CNN model developed rapidly, and the architecture of the model became more complex. These models are known for their accuracy on ImageNet. Hence, the theme of this study is to explore and study classical CNN models, because these models are similar in basic structure, but they modify the model in different ways. In addition, this paper selects four classical CNN models for learning and implementation, uses FashionMNIST datasets on four models, and records the performance differences of image classification of specific models, with the intention of studying the comparison of results among different models. The four models are LeNet-5, AlexNet, VGG-16, and GoogleNet. The training was carried out on the same equipment and under the same conditions. The study finds that GoogleNet achieves the best prediction accuracy. LeNet-5 spends the least amount of time on training and forecasting. The GoogleNet and AlexNet models can be considered for practical applications, while the VGG-16 and LeNet-6 are either inefficient due to the long training time of the models, or the accuracy cannot.

Keywords: LeNet-5, AlexNet, VGG-16, GoogleNet, Classification of Image.

1. Introduction

In the realm of neural networks, the task of image classification is inseparable from Convolutional Neural Networks (CNN). One classic model, LeNet-5 [1], was developed in 1998, primarily for handwritten digit recognition, and later adapted for Chinese character recognition. Over the following decades, numerous attempts were made to innovate and to enhance CNN for improved image classification accuracy, yet none succeeded in significantly upgrading the fundamental architecture.

In the year 2010, the ILSVRC competition was initiated by the ImageNet [2], or "ImageNet Large-Scale Visual Recognition Challenge," a competition aimed at advancing image classification models and fostering the development of novel CNN architectures. Finally, in 2012, Alex Krizhevsky and his team introduced and developed a regularization technique called "Dropout" to mitigate overfitting in fully connected layers, proving to be highly effective. This model came to be known as AlexNet [3].

In 2014, the Visual Geometry Group (VGG) at the University of Oxford, collaborating with Google DeepMind, unveiled a new model, aptly named VGG-16 [4]. They delved into another crucial aspect of CNN architecture design–depth. Compared to the preceding AlexNet, VGG-16 boasted deeper layers, more parameters, and superior performance and versatility. However, VGG-16 secured the second place in the 2014 ILSVRC, overshadowed by its formidable opponent, GoogleNet [5]. GoogleNet introduced the innovative Inception module and a more intricate design, securing the championship title. Although new models have been introduced, such as ResNet [6], this paper does not cover ResNet.

This paper primarily employs these four CNN models–LeNet-5, AlexNet, VGG, and GoogleNet–within the PyTorch deep learning framework. Using the publicly available FashionMNIST [7, 8] datasets of clothing categories within PyTorch [9]. The primary aim of this scholarly article is to undertake a comparative analysis. the strengths and weaknesses of these models in the context of clothing classification, ultimately drawing conclusions on their usability. The models are constructed and utilized in a similar manner, evaluated on the same baseline and training equipment.

Subsequent sections of this paper focus on the principles and delve into the distinguishing features of each model, followed by a comparative analysis of their results. Finally, the paper summarizes the key findings, explores their implications in depth, and offers potential future directions.

2. Method

2.1. Data Description

The datasets imported in this paper are the FashionMNIST datasets. FashionMNIST is a dataset consisting of 70,000 grayscale images of clothing items from the front, each with a resolution of 28 by 28 pixels and encompasses 10 different categories. This dataset is also a perfect alternative to the classic MNIST handwritten digit datasets, provided by the research department of Zalando, a German fashion company. The datasets comprise 70,000 photographs, of which 60,000 are used to construct the training set samples, and 10,000 are used to construct the specimens or instances comprising the test datasets. The datasets are selected from the public datasets in PyTorch. Fig. 1 is part of FashionMNIST datasets.

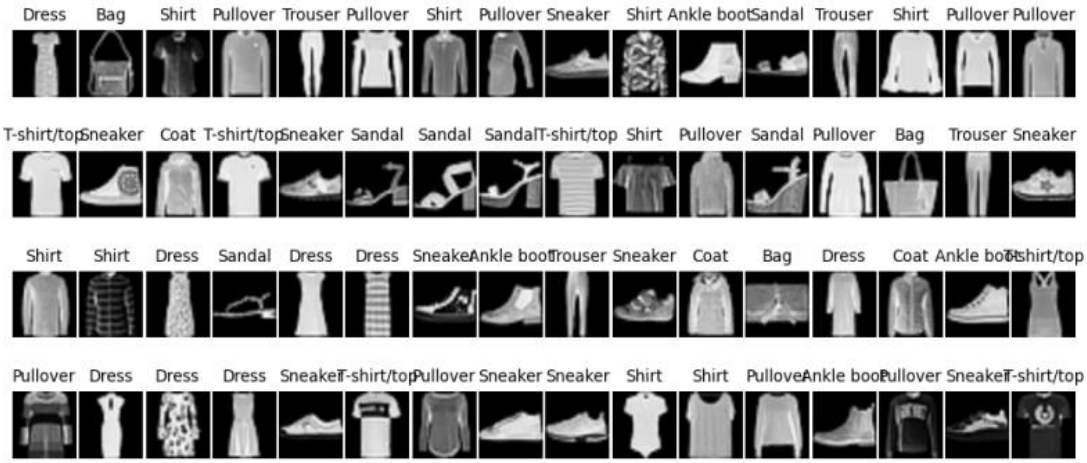

Fig. 1 Part of FashionMNIST datasets [6]

2.2. A brief introduction to CNN principles

CNN is a class of networks that use convolutional layers. The convolutional layer in this neural network, the nonlinear activation function Sigmoid, and the Fully Connected (FC) layer are all merged. In order to construct a high-performance CNN, convolutional layers need to be arranged, the spatial resolution of their representation is progressively diminishing the degree of spatial detail or precision of their representation and increasing the number of channels. In CNN architectures, the feature representations encoded within the convolutional blocks typically undergo further processing through one or more FC layers prior to the final output generation. This hierarchical processing paradigm facilitates the extraction of abstract, high-level features from the raw input data, ultimately contributing to the network's decision-making capabilities. CNN is used to implement all models in this paper, and the principles behind each model are described below.

2.2.1 A brief introduction to LeNet-5 principles

LeNet-5 is mainly composed of the two components or segments, namely the first part is the feature extraction part and the second part is the FC layer. The feature extraction of the first part consists of two convolution layers. The second part contains two layers of pooling, while the layers of the fully connected feature are made up of three fully linked layers. In the LeNet-5 model, each unit structure is composed of each convolution layer, one sigmoid activation function, and one pooling layer. The architecture of LeNet-5 model can be depicted in Fig. 2, presenting a graphical depiction of its intrinsic architecture.

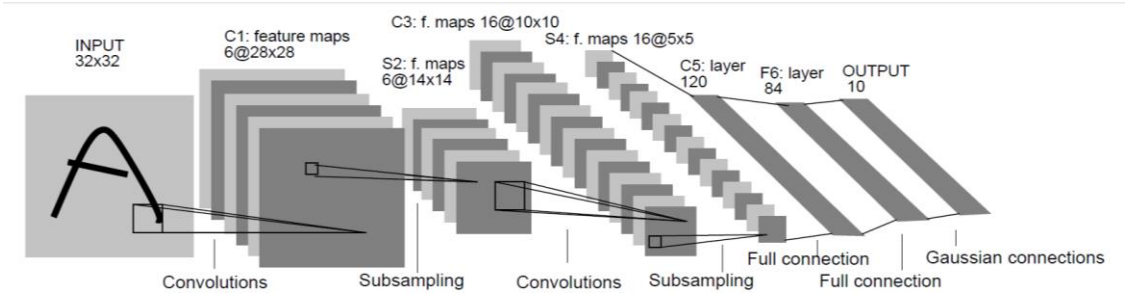

Fig. 2 Architecture of the LeNet-5 model [1]

2.2.2 A brief introduction to AlexNet principles

In the realm of computer vision, while LeNet-5 demonstrates commendable performance on modest-sized datasets, the viability and performance of training CNN on larger, more comprehensive datasets remain an area of ongoing investigation. The advent of AlexNet in 2012, which emerged victorious in a seminal competition, marked a pivotal shift towards the widespread adoption of deep learning methodologies within computer vision. Notably, the foundational principles of AlexNet and LeNet-5 exhibit a degree of congruence, albeit accompanied by pronounced distinctions. Firstly, AlexNet surpasses LeNet-5 in depth, possessing a more intricate architecture. Secondly, AlexNet comprises eight layers: an intricate arrangement of five convolutional layers, two fully connected hidden layers, and a solitary fully connected output layer. Thirdly, AlexNet innovatively incorporates ReLUs [3] as its activation function, forsaking the traditional Sigmoid function. Consequently, it is evident that AlexNet boasts a total of eight layers, exceeding the layer count of LeNet-5.

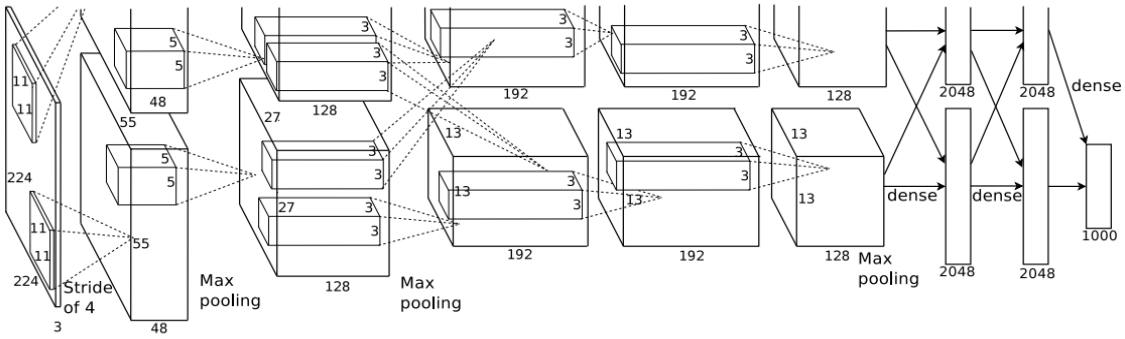

Fig. 3 Architecture of the AlexNet model [3]

Fig. 3 is the architecture of the AlexNet model. To improve model accuracy and reduce overfitting, AlexNet uses Dropout [3], an approach that has since been widely adopted. In the context of the Dropout regularization technique, the process unfolds as follows: Initially, a stochastic and provisional subset comprising approximately half of the hidden neurons within the neural network is temporarily deactivated, leaving the input and output neurons unaltered. Subsequently, the input data traverses this altered network configuration in a forward pass, and the resulting loss is then subjected to a backward pass through the same modified network, enabling the computation of gradients. Following the completion of this process for a mini-batch of samples, adhering to the stochastic gradient descent algorithm, the parameters associated with the active (non-dropped) neurons undergo an update, refining the model's learning capabilities while concurrently incorporating a regularization mechanism that alleviates the risk of overfitting. This iterative process is then continued.

To sum up Alexnet, there are similarities and differences between AlexNet and LeNet-5 in design. In order to adapt to ImageNet large-size images, AlexNet employs a hierarchical approach to convolution kernel sizes, ranging from sizable 11x11 to more refined 5x5 and 3x3 kernels. This design, coupled with the inclusion of both convolutional and fully connected layers, introduces an extensive parameter space exceeding 60 million, significantly advancing the capabilities of CNNs in the realm of image classification beyond traditional machine learning methods. Moreover, AlexNet innovatively adopts Rectified Linear Units (ReLUs) as the activation function, a departure from the

traditional sigmoid function, resulting in expedited convergence rates and a substantial reduction in training time. To enhance model accuracy and mitigate overfitting, AlexNet integrates dropout layers, a strategy that has since garnered widespread adoption. Furthermore, AlexNet introduces overlapping maximum pooling within its CNN architecture, deviating from the prevalent use of average pooling, which tends to induce blurring effects. By exclusively utilizing maximum pooling, AlexNet mitigates these issues, contributing to its overall effectiveness. In addition, AlexNet proposes that the concession length is smaller than the size of the pooled core, so overlaps and overlays occur among the export of the pooling layer, leading to enhanced feature richness. AlexNet employed two data augmentation strategies to substantially augment the data set of training, enhance the resilience, and mitigate overfitting. LRN [3] regularized, multi-GPU parallel training mode was used (it was not widely used later years).

2.2.3 A brief introduction to GoogleNet

An inception block is the term used to refer to a volume block of a base in GoogleNet [5]. The Inception block architecture encompasses four parallel processing streams, each designed to capture information at distinct spatial scales. The initial three streams enable the extraction of features from varying receptive field sizes. To mitigate model complexity, the two intermediate streams precede their respective convolutional layers with a 1×1 convolutional filter, which acts as a dimensionality reduction technique by decreasing the parameter of import channels. The fourth stream integrates a 3×3 max-pooling layer, followed by a 1×1 convolutional layer, to modify the channel depth while preserving the spatial dimensions. To ensure consistency in height and width between inputs and outputs across all streams, appropriate padding is applied. Subsequently, along the channel dimension, the outputs of these parallel streams are concatenated, amalgamating their respective feature representations to form the output of the Inception block. Within the Inception block design, a pivotal hyperparameter that is often tuned is the parameter of export channels per layer, enabling the fine-tuning of the model's representational capacity.

Through the innovative inception module and a more complex network design, the author team was able to beat VGG-16 to win the title. The input features are obtained by applying parallel filters and then combining them in series along the channel dimension. These features are then used as the input for the next layer. The Inception blocks are then piled on top of each other, one layer at a time. In this way, the sparse type is introduced and the optimal topology of CNN is simulated. It also gives the network the ability to automatically select, rather than artificially setting convolution or pooling or determining the kernel size of the convolution layer. To reduce and decrease these overall parameter numbers, 1×1 convolution is added to the Inception block, which effectively reduces the feature dimension and avoids parameter explosion.

2.2.4 A brief introduction to VGG-16

The structure of the VGG-16 is similar to the classical CNN, but there are a few changes. The layers in VGG-16 are connected by blocks, and the structure of convolution layers in blocks is the same. Outside blocks, blocks are connected to each other through maxpool. The convolutional layers within the blocks exhibit a homogeneous structural design, facilitating their stacking and reuse with a consistent kernel dimension, stride, and padding parameter of 3×3. Additionally, the incorporation of a max-pooling layer serves to halve the feature representation of the preceding layer (the output of the block layer), yielding a notably regular and systematic reduction in the dimension. The depth of the VGG-16 network is deep. There are sufficient numbers of parameters, and there are sufficient number of deep network layers, which makes the model classification effect of training excellent, but the larger parameters put forward greater resource requirements for training and model preservation. The VGG-16 network has a small filter size, a small kernel size, and the global size is set as 3×3. This model exhibits a notably reduced size in comparison to preceding models, making it comparatively more compact. Fig. 4 is the architecture of the VGG-16 model.

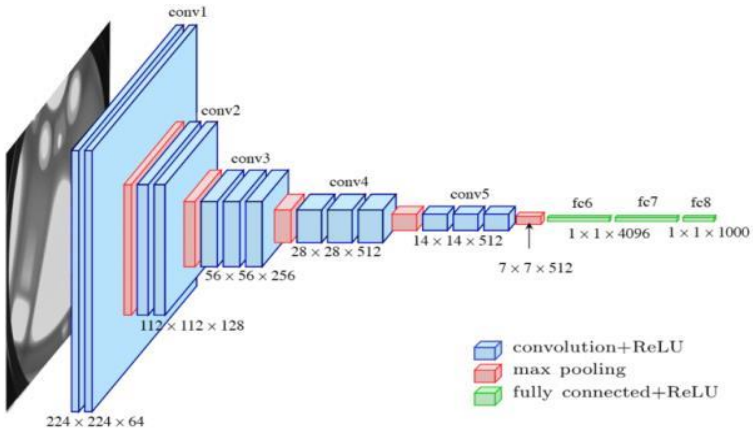

Fig. 4 Architecture of the VGG-16 model [10]

Through the extension of the traditional CNN model (AlexNet), it can be understood that in addition to the design of relatively complex model structures (such as GoogLeNet), Furthermore, with the aim of improving predictive accuracy detected in this model, depth is crucial. Compared with the previous AlexNet, VGG has deeper depth and more parameters. The effect and portability are better, and the model design is simple and regular, so it is widely used. There are also some characteristics summarized as follows: First, the small size filter not only makes the parameters less, but the effect is not weaker than the large size filter, such as 5×5. Second, the use of blocks results in a very concise network definition. Complex networks can be effectively designed using blocks. Third, local response normalization in AlexNet does not play a significant role.

3. Experiments and Results

3.1. Testing and prediction accuracy

In the training set, all models contain 60,000 clothing category images. Following each round of training, the code can show the accuracy rate and training time. Finally, upon completion of the training, the total training time and final training accuracy rate can be output, and the weight can be saved in the path. Each model was trained 50 times, respectively, and the effect and training time between the models were compared respectively in the two training cycles. Table 1 compares the accuracy and time of different models under different training times.

Table 1. Accuracy and training time, 50 epochs.

Model	Accuracy(%)	Training Time(minutes)
LeNet-5	88.41	11 minutes and 38 seconds
AlexNet	90.69	29 minutes and 33 seconds
VGG-16	91.56	260 minutes and 57 seconds
GoogleNet	92.84	57 minutes and 2 seconds

According to table 1, it can be seen that the efficiency of the VGG-16 is not high because it has the longest training time. While LeNet-5 has the lowest accuracy, it is also not suitable for practical application. However, the advantage of Lenet-5 is that it requires the shortest training time. GoogleNet is the model with the highest accuracy in this study, while the AlexNet model ranks third in accuracy but second in training time, which is a model with no advantages but no disadvantages.

3.2. Visualization and result analysis

The following figures show that all models have training accuracy, training loss, validation accuracy, and validation loss recorded during the training process.

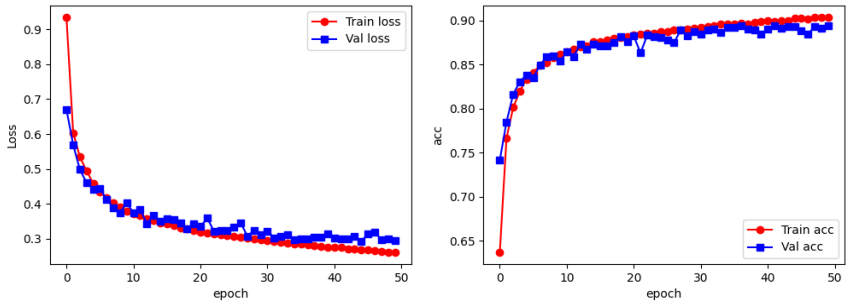

Fig. 5 Accuracy and loss for training and validation of the model of the LeNet-5

In the LeNet-5 model in Fig. 5. It can be seen that LeNet-5 can achieve high accuracy in very few epochs, starting at about 10 training sessions, after which the increase in accuracy is not very significant. In addition, according to the above table, it can be found that although LeNet-5 has the shortest training time of 50 times, its accuracy is also the lowest. Meanwhile, by observing the table, it can be found that among the four models, the accuracy of LeNet-5 is the only model that lower than is 90%, indicating that LeNet-5 can be applied to image recognition and classification. However, according to the original design intention of the LeNet-5 model, it seems that it is not necessary to use this model and do more training times, and it can achieve better accuracy in more than ten training times. However, even if the LeNet-5 model is further trained with more times (such as 50 times of training), the improvement is limited. However, the advantage of LeNet-5 model is that its loss is less and less.

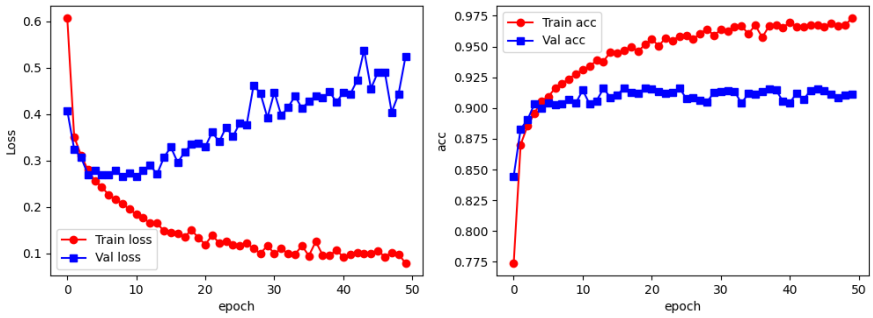

Fig. 6 Accuracy and loss for training and validation of the AlexNet

By observing Fig. 6. It can be found that overfitting occurs in the AlexNet to accompany the increase in accuracy and decrease of the loss, but the time of training is not as similar to the validation number during training as presented in LeNet-5 training output figure. However, with the increase in the number of trainings, the loss increased slightly, but this phenomenon, unfortunately, does not have a significant impact on the improvement of its accuracy, and it has been proved that compared with LeNet-5, the loss has indeed increased.

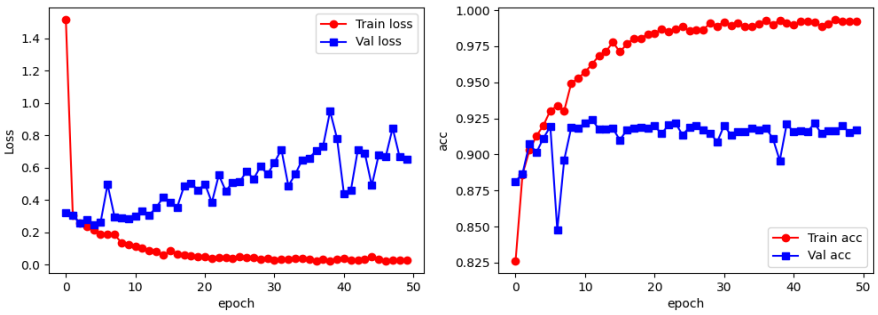

Fig. 7 Accuracy and loss for training and validation of the VGG-16

In Fig. 7. About VGG-16 model, it can be found that the validation loss in the left figure is somewhat non-convergent with longer training sessions, and the validation accuracy in the right figure is slightly increased compared with that of the AlexNet model, and the total accuracy of the

dot plot is obviously more stable than that of the AlexNet. Nevertheless, the disadvantage of the VGG-16 is that the training is very timeconsuming, which is not conducive to practical application because the efficiency is not high, and the improvement in accuracy is steep on the dot plot at about 10 training times.

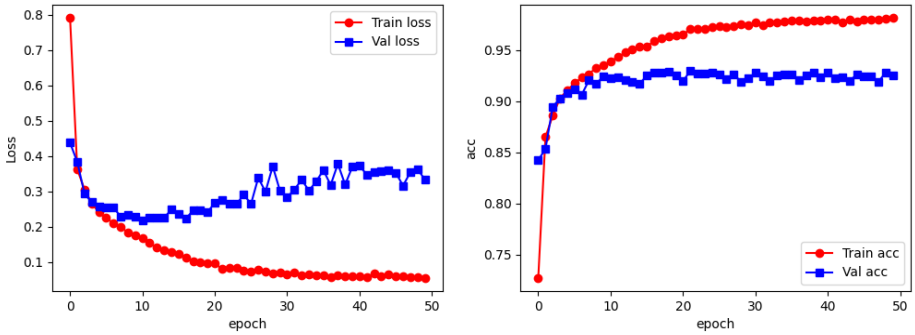

Fig. 8 Accuracy and loss for training and validation of the GoogleNet

Figure 8 illustrates the accuracy metrics detected by code and all metrics loss for both training and validation phases of the GoogleNet. Notably, GoogleNet emerges as the optimal model in this study, achieving the highest accuracy while maintaining a feasible training duration of 50 epochs. GoogleNet performs best on this FashionMNIST dataset because it has the highest accuracy, and the total training time is second only to VGG-16 among the three models with more than 90% accuracy.

3.3. Result analysis

In this study, the GoogleNet model has the best accuracy. It does not take as much time as the VGG model, nor does it have obvious dithering in the accuracy of training as other models. This GoogleNet model also had the smoothest training accuracy of the four models, with a result of nearly 93%. The innovative inception module is really useful for image classification and works very well. Among the models, the Global Average Pooling (GAP) [5] layer does have advantages, which can suppress overfitting and directly flatten the fully connected layer, which still retains a large amount of spatial information.

4. Conclusions and Future Work

All three models applied to the FashionMNIST datasets achieved an accuracy of more than 90%; GoogleNet obtained the highest prediction accuracy of 92.84%, followed by VGG-16 with 91.56% and AlexNet with 90.69%, while LeNet-5 only achieved an accuracy of 88.41%. The average time required for each model to train each epoch varies: LeNet-5 training takes the shortest average time (13.96 seconds per epoch), and each AlexNet epoch takes 35.46 seconds, while VGG-16 takes an average of 5.22 minutes to train each epoch and GoogleNet takes an average of 1.14 minutes to train each epoch. It can be concluded that GoogleNet performs best on the FashionMNIST datasets because it has the highest accuracy and the total training time is second only to VGG-16 among the three models with more than 90% accuracy. AlexNet model can be considered for simple clothing category classification applications, such as clothing image search by users in mobile phone software. Because among the three models above 90% accuracy, it has the shortest total training time.

References

[1] Yann Lecun, Leon Bottou, Yoshua Bengio and Patrick Haffner. Gradient-based learning applied to document recognition. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009: 248-255.

[3] Alex Krizhevsky, Ilya Sutskever and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. Proceedings of the Advances in Neural Information Processing Systems, 2012(25).

[4] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-scale Image Recognition. arXiv preprint arXiv:1409.1556, 2014.

[5] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke and Andrew Rabinovich. Going deeper with convolutions. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015: 1-9.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770-778.

[7] Han Xiao. Fashion-MNIST. Github, https://github.com/zalandoresearch/fashion-mnist, 2017.

[8] Han Xiao, Kashif Rasul and Roland Vollgraf. Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. arXiv preprint arXiv:1708.07747, 2017.

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. Advances in neural information processing systems, 2019, 32.

[10] Max Ferguson, Ronay Ak, Yung-Tsun Tina Lee and Kincho H. Law. Automatic localization of casting defects with convolutional neural networks. Proceedings of the IEEE International Conference on Big Data (Big Data), 2017: 1726-1735.