Proceedings of CONF-SPML 2026 Symposium: The Artificial Intelligence Tools & Applications
DOI: 10.54254/2755-2721/2026.CH31978

Comparative Analysis and Optimization of Model Architectures for Fashion-MNIST Image Classification

Nan Jiang

School of Management, Shenzhen University, Shenzhen, China
1917341608@qq.com

Abstract. The performance of image classification models depends greatly on the architectural decisions made. Fashion-MNIST, as the mainstream adopted by researchers for model performance analysis, provides another avenue for the systematic comparison of different model architectures. In this paper, we have comparatively studied and analyzed the performances of Multi-Layer Perceptrons (MLP), Convolutional Neural Networks (CNN), Random Forests and Residual Networks (ResNet) on this dataset, and found that one of the reasons for the excellent performance of convolutional networks may lie in the ability of extracting spatial features inherently possessed by convolutional layers. Although a deeper ResNet-34 shows an excellent performance (91.15%), its large number of parameters makes it less efficient for general tasks. To improve the efficiency, we find that by increasing the number of channels in the first convolutional layer from 32 to 64, the achieved accuracy (92.44%) is superior to any single task, which verifies the effectiveness of width optimization. In summary, for fashion-mnist such applications, an optimized width convolutional network architecture achieves the best accuracy-to-efficiency balance. We empirically prove that for image classification tasks, model selection and light design are significantly influenced by adopting appropriate architectural optimizations.

Keywords: Fashion-MNIST, image classification, convolutional neural network, ResNet, model comparison, structural optimization

1. Introduction

Image classification is one of the fundamental tasks in computer vision, which focuses on the requirement of automatically labeling input images into certain categories. With the rapid development of technology, the image classification model has experienced tremendous progress in terms of performance. Most of these progress can be attributed to the continuous innovation and evolution of model architectures. From the earliest traditional machine learning methods that rely on manually extracted features, through the emergence of multi-layer perceptrons (MLPs) that can automatically learn the feature representation mechanism, all the way to the field-changing convolutional neural networks (CNNs) that revolutionized the field of deep learning, and even until the recent breakthroughs in deep network training caused by the introduction of residual networks (ResNets) [1], the reason why each mainstream architecture could significantly improve the generalization performance is that each of them has injected different inductive biases, which means

DOI: 10.54254/2755-2721/2026.CH31978

fundamental assumptions about data distribution or model form, that facilitate learning those particular patterns more efficiently.

Among them, the convolutional neural networks (CNNs) have demonstrated impressive success, and the key component underlying their success is the convolutional kernel (also named simply filter). To understand why CNNs can achieve such great success, it is crucial to have a grasp of the convolutional kernel. Firstly, from an operational perspective, the convolutional kernel is a small weight matrix (e.g., 3 × 3 or 5 × 5). It works like a sliding window that scans over the entire region of the input image. At each position of scanning, it will perform dot product and summation with the corresponding local pixel region of the image and obtain the output value at that position, which is called a feature map, providing information about the distribution of local features in the preceding layer that match the weight pattern of the convolutional kernel.

This study aims to address this gap by designing and implementing a rigorous, controlled comparative experiment. We will not only faithfully implement and reproduce the benchmarking tests on several popular and influential models such as Random Forests, MLPs, and CNNs but also compare them with a few novel architectures (resated networks, or ResNets), which are among the latest state-of-the-art evidence of what is possible with deep learning. Perhaps more importantly, instead of keeping the model's depth constant, we creatively explore the effect of model capacity by varying the width of the initial convolutional network architectures by introducing a series of tests on their widened counterparts. Our main assumption in this paper is that our readers should not be limited to a mere accuracy comparison, but instead should be allowed to follow the structural path that leads to the performance divergence among different model architectures and assess the feasibility of structural optimization. In other words, we aim to provide definitive and accurate guideline principles for model selection and lightweight design depending on the application tasks.

2. Review section

2.1. Theoretical foundations

LeCun et al. [2] defined the cornerstone theory of deep learning in their Nature review, who not only explained the significance of setting up multilayer perceptrons (MLPs) and convolutional neural networks (CNNs) but also emphasized the advantage of convolutional layers in weight sharing and local connectivity when dealing with images. Schmidhuber [3] summarized the history of neural networks and explained how backpropagation and long short-term memory (LSTM) networks led to deep learning progress. Li et al. [4] introduced the design idea of each module in CNNs, such as convolutional kernels, pooling, and normalization, and explained the inductive bias of CNNs in visual feature extraction. However, Tolstikhin et al. [5] questioned the requirement of convolution and attention and proposed MLP-Mixer, which achieved image classification through pure MLPs and opened up new theoretical exploration spaces. Raghu et al. [6] pointed out that between CNNs and Vision Transformers (ViTs), there were differences in the mechanism of intermediate feature formation; that is, CNNs had more hierarchical structure, whereas ViTs were prone to early aggregation of global information. Further comparing ViTs and CNNs, Khan et al. [7] highlighted the differences in inductive bias, structural assumptions, and application prospects.

2.2. Research methods

In terms of methodology, LeCun [2] and Schmidhuber [3] mainly adopted literature reviews and theory description to summarize key ideas and typical applications of deep learning. Li et al. [4]

DOI: 10.54254/2755-2721/2026.CH31978

performed systematic literature review and comprehensively summarized the structure evolution and application scenarios of CNNs. Tolstikhin et al. [5] proposed and verified their MLP-Mixer model through experimental method, performing training experiments and ablation experiments on large-scale image dataset ImageNet, and comparing its performance with CNNs and Transformers; Raghu et al. [6] used multiple analytical methods (including CKA-based centralized kernel alignment (CKA), attention distance metrics, effective receptive field analysis, skip-connection intervention experiments, and linear probing) to thoroughly compare the internal representation structure information retention capability, and transfer learning performance, and so on) to systematically compare the internal representation structure, the ability to retain spatial information, and transfer learning performance, and so on) between ViT and CNNs; Khan et al. [7] mainly relied on the review work, which has been systematically compared by synthesizing the experimental results from previous works.

2.3. Research findings

Despite their popularity, CNNs have maintained their effectiveness in visual tasks. LeCun [2] and Schmidhuber [3] demonstrated that the locality and sharing of parameters of CNNs still offer advantages even when compared to small amounts of data and restricted computational resources. Li et al. [4] also agreed with this viewpoint in their survey, as they showed that CNNs are applied today in object detection, image segmentation, and facial recognition. On the other hand, Tolstikhin et al. [5] experimentally verified that MLP-Mixer can match CNNs and Transformers in their performance when trained on large-scale datasets, showing that convolution is not the only way to succeed in visual tasks. Raghu et al. [6] showed that ViTs have a more uniform distribution of features in the intermediate representations and receive global information more frequently than local information. Conversely, CNN features are local and hierarchical. Khan et al. [7] summarized that ViTs' competitiveness increases with large-scale pretraining and plentiful computing resources; however, they tend to be outperformed by CNNs on small-scale data tasks.

2.4. Research discrepancies

Disagreements have arisen. Some reports (e.g. Tolstikhin [5]) show that non-convolutional architectures can match the performance of convolutional ones when dealing with large-scale data and limited computation, but it is review studies (LeCun [2], Schmidhuber [3], Li [4]) and systematic comparison (Khan [7]) that all confirm that CNNs maintain their advantages in many types of applications. Why there is such a disagreement? In fact, as analysed by Raghu et al. [6], this might be due to the underlying mechanisms; for ViTs focusing on global feature modeling, whereas for convolutional architectures the inductive biases come into play when dealing with small datasets and specific tasks. Therefore, the divergence of opinions may arise from the difference in the scale of training data, the model structural bias, and experimental settings.

3. Methodology

This chapter will explain the experimental design, model architecture and optimization strategy adopted in this study. Firstly, we explain the Fashion-MNIST dataset and the workflow for preprocessing tasks. Secondly, we explain the four model architectures (random forest, multi-layer perceptron, convolutional neural network and residual network) involved in our comparison and their implementation details in detail. Thirdly, we present a width optimization method for the

DOI: 10.54254/2755-2721/2026.CH31978

baseline convolutional neural network (CNN). Finally, we explain the metrics used to evaluate the model performance. We aim to fairly compare their performances on the same task through controlled experiments on the same dataset.

3.1. Dataset and experimental setup

This paper strictly uses the Fashion-MNIST dataset [1], in which there are 10 categories of clothes, and the training set has 60,000 images and the test set has 10,000 images, all sized 28x28 pixels. And this dataset has been standardized and pixel values have been normalized between 0 and 1, and also divided by the mean and standard deviation value of the dataset itself (mean=0.1307, standard deviation=0.3081). All experiments were implemented on CPU, and PyTorch [8] was chosen for the deep learning framework. In order to make the results comparable and reproducible, each model's training process (optimizer, learning rate, number of epochs) was strictly repeated according to the original open-source project's settings. Only the model architecture was changed.

3.2. Comparing model architectures

To ensure a thorough and representative comparison, we selected four types of models from different perspectives:

1. Random Forest: As the representative of the traditional non-parametric machine learning methods, this paper implemented it using the Scikit-learn library with the default parameters and observed the basic performance (before fusing multiple probability distributions into a single one) of the random forest algorithm.

2. Multi-Layer Perceptron (MLP): As the simplest fully-connected type deep learning model, this experiment implemented a model with 5 layers of hidden layers (the neuron count for each layer was 2000:1000:500:100, with the activation function of each layer being ReLU except for the last layer which outputs the probability distribution vector with a total of 10 classes), which can be regarded as a preliminary attempt to implement the image classification task.

3. Convolutional Neural Network (CNN): As a model that has achieved excellent results in image data classification tasks, this paper implemented the basic structure of a convolutional network, which consists of four convolutional blocks (each block consisted of a Conv2D-BatchNorm2d-ReLU layer and a MaxPool2d layer except for the last two layers), and added a full connection layer at the end for classification. The number of convolutional kernels in the first layer (that is, the number of output channels) was set to 32.

4. Residual Network [8]: As another representative of modern deep architectures, this paper implemented the widely tested two-depth versions of residual networks, namely, ResNet-18 and ResNet-34 [2], to observe the effect of increasing depth.

3.3. Architectural modification: CNN width optimization

Apart from the comparison between different benchmark versions, to explore the effect of model capacity, we implemented a targeted structural modification:

Original CNN: The network's first layer has 32 convolutional kernels.

Modified CNN: The number of output channels of the network's first convolutional layer is expanded to 64, which makes the first layer broader and allows the layer above (i.e., the second convolutional layer) to learn more diverse basic foundations (e.g., different orientations of edges,

DOI: 10.54254/2755-2721/2026.CH31978

different textures) for its own layers, which may expand the whole network's representational capacity.

3.4. Evaluation metrics

Model performance is primarily assessed based on the overall classification accuracy calculated on an independent test set—the proportion of correctly classified samples relative to the total sample count. This serves as the most intuitive and fundamental performance metric for classification tasks. Additionally, we analyze the confusion matrix to examine the model's classification behavior across specific categories, identify prevalent confusion patterns, and understand common weaknesses.

4. Results and discussion

All the following analysis and experiments will strictly base on the experimental output data (even with some figure files provided), as shown in Table 1. It clearly shows the final performance of these models on the test set, which is used as the basis of the following discussion.

Table 1. Performance comparison of different models on the fashion-mnist test set (data obtained from experimental output)

Model	Test Accuracy	Parameter Estimation	Training Epochs
Random Forest	88.0%	(N/A)	(N/A)
Multi-Layer Perceptron (MLP)	89.0% (8866/10000)	~2.3M	10
Original CNN (32 channels)	92.0% (9237/10000)	~0.5M	10
Modified CNN (64 channels)	92.44% (9244/10000)	~0.7M	10
ResNet-34	91.15% (9115/10000)	~21M	5
ResNet-18	90.91% (9091/10000)	~11M	5

4.1. Inductive bias in model architecture: the fundamental cause of performance stratification

Up to 4.44 percentage points difference in accuracy between models may not seem like an extreme variation across different architectures, but this discrepancy is not an accident. As we saw in the introduction, the superior performance of convolutional neural networks (CNNs) (92.0%) is due solely to their architectural design. Recall that CNNs are designed to respect the spatial structure of images via their convolutional kernels, local connections, and weight sharing. Convolutional layers allow images to be flattened into higher dimensional vectors (as per the multi-layer perceptron or MLP shown below), while still efficiently extracting hierarchical features (e.g., edges and textures, to local patterns like the ones highlighted below), all while translating invariance. In contrast, the flattening of images into one-dimensional vectors (one for each pixel) removes all information regarding the relationship between pixels. Fully connected layers (performed by the multi-layer perceptron or MLP) have weights updated frequently due to the high sensitivity to even small translations of images. With approximately 4.6 times as many parameters as the convolutional network (as shown in Appendix A.2), there is also more room for overfitting. The performance floor of random forest (88.0%) suggests that although an old and venerable traditional machine learning method, its reliance on raw pixel values as features lacks the ability to automatically learn

DOI: 10.54254/2755-2721/2026.CH31978

discriminative hierarchical features. Therefore, this experiment confirmed that when selecting a model, it is the fundamental architecture of the model to respect (or not) the underlying characteristics of the data to determine the model's performance ceiling.

4.2. Model width optimization: enhancing accuracy through enhanced feature extraction

With CNN's essential advantage secured, we systematically test target-oriented optimization experiments on CNN, to verify the effectiveness of structural fine-tuning can really release the model potential. When increasing the width (i.e., number of channels) of convolutional layer's first kernel map from 32 to 64, the accuracy stabilizes at 0.44% (from 92.0% to 92.44%), which has meaningful significance compared with the state-of-the-art on this task. The possible reason might be that a larger number of convolutional kernels could allow the model to learn more diverse and finer-grained foundational features in parallel at the low layers, thereby providing more rich and discriminative information source for the following feature combination and abstraction. Surprisingly, this improvement is achieved at the cost of increasing the number of parameters from around 0.5M to about 0.7M (about 40%), which demonstrates an extremely high optimization efficiency. It implies that for mature tasks, instead of resorting to more complicated and heavyweight models by replacing existing well-performed architectures (e.g., CNN), target-oriented "fine-tuning" (e.g., width adjustment) might be a better solution.

4.3. Training efficiency differences due to mismatched model depth and task complexity

Both ResNet-34 (91.15%) and ResNet-18 (90.91%) outperformed the optimized CNN, calling into question the widely held intuition that "deeper models necessarily lead to better performance." However, this result re-emphasizes the fundamental idea that model capacity should match task difficulty. Fashion-MNIST, given its low resolution, small number of categories, and relatively simple background, may have information complexity much lower than that of a ResNet required to utilize the full capacity of convolutional layers on this shallow task. Fashion-MNIST is an easy task compared to the more difficult ImageNet classification task, where it is necessary to classify the thousand-valued labels of extremely high-resolution images. The 21 million weights (30 times the number of weights in a CNN) of ResNet-34 were likely a significant cause of parameter inefficiency for this task. The training behavior also shows that ResNets have significantly higher initial loss (e.g., ~2.5 for ResNet-34 compared to ~2.3 for a CNN) and more extreme loss variation (during training) than CNNs, which implies that their optimization trajectories were more unstable than those of the CNN. Therefore, when resources are limited and the task itself is not very complex, a well-optimized "shallow and wide" model often achieves a better accuracy-efficiency tradeoff compared to a "deep and narrow" architecture.

4.4. Classification difficulties reveal inherent challenges in the dataset

When we dig deeper into the confusion matrix, it shows that no matter what the big picture performance is like, all the models always have the highest difficulty in dealing with category 6 (shirts), which are all wrongly classified as T-shirt, jacket, and sweatshirt. We find that such a globally highly consistent error pattern might be caused by a subtle issue underlying the dataset, that is, clothes such as T-shirt, jacket, and sweatshirt seem to have visually similar but distinct silhouettes and shapes. On the other hand, categories that are difficult to recognize with distinct silhouettes or shapes, namely pants, bags, and boots, are all easily recognized by the three models. This

DOI: 10.54254/2755-2721/2026.CH31978

encourages us not to focus on the "which model performs better" viewpoint, but explore what kind of problems these models have struggled with. Based on the above findings, we suggest a more targeted direction for future improvement in model upgrade, that is, to make models able to differentiate difficult categories better by improving their architectures or training strategies.

5. Conclusions

1. Architectural inductive bias is a decisive factor in performance. Taking convolutional layers as an example, their introduction of local connections, weight sharing, and translation invariance are all in perfect agreement with the spatial characteristics of image data, allowing CNNs to significantly outperform MLPs and random forests in terms of classification accuracy and establish their position in image classification.

2. Targeted structural optimization can effectively enhance performance. Significantly increasing the width of the first convolutional layer (from 32 to 64 channels) also led to a clear 0.44% improvement in model accuracy, showing that the fine-tuned structured design within a fixed architecture could achieve a higher return with fewer adjustments.

3. Model complexity should match task difficulty. Deeper ResNet models did not exhibit obvious advantages in this task because their large parameter counts made them inefficient models. This shows that for relatively simple tasks, pursuing deeper models is not always beneficial; an "shallow and wide" CNN reaches an optimal accuracy-efficiency balance point.

In conclusion, through rigorous experimental analysis, this paper shows that the model architecture and match with task characteristics are the key factors to some extent for the Fashion-MNIST image classification task. The convolutional layer with width optimized network architecture is the ideal model architecture that balances accuracy and efficiency. In addition, future work can be carried out from three aspects: 1) Focusing on the optimization of convolutional layer hyperparameters (e.g., depth and width), in-depth research on how to combine convolutional layers with other modules (e. g., attention mechanism) to better solve the problem of difficult category discrimination will also be meaningful; 3) As this comparative framework can meet the requirements of most people's visual perception, it can also be extended to other image datasets to test the universality of the above research conclusions and investigate the boundary conditions under which deep architecture has advantages.

References

[1] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 770-778).
[2] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436-444. https: //doi.org/10.1038/nature14539
[3] Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85-117. https: //doi.org/10.1016/j.neunet.2014.09.003
[4] Li, Z., Yang, W., Peng, S., & Liu, F. (2020). A survey of convolutional neural networks: Analysis, applications, and prospects. arXiv preprint arXiv: 2004.02806. https: //arxiv.org/abs/2004.02806
[5] Tolstikhin, I., Houlsby, N., Kolesnikov, A., et al. (2021). MLP-Mixer: An all-MLP architecture for vision. In Advances in Neural Information Processing Systems (NeurIPS).
[6] Raghu, M., Unterthiner, T., Kornblith, S., et al. (2021). Do vision transformers see like convolutional neural networks? arXiv preprint arXiv: 2108.08810. https: //arxiv.org/abs/2108.08810
[7] Khan, A., Rauf, Z., Sohail, A., et al. (2023). A survey of the Vision Transformers and their CNN-Transformer based Variants. Artificial Intelligence Review. https: //doi.org/10.1007/s10462-023-10595-0
[8] Xiao, H., Rasul, K., & Vollgraf, R. (2017). Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv: 1708.07747.