ECE371 Neural Networks and Deep Learning Assignment 1

聂芷琦
22365080
Business School
Sun Yat-sen University, Shenzhen Campus
niezhq@mail2.sysu.edu.cn

Abstract: This report systematically evaluates the evolution of image classification techniques through two key exercises. First, I compare the performance of **ResNet-18**, **ResNet-50**, and **ShuffleNet** architectures with different optimizers (SGD vs. Adam) on a flower classification task. Our experiments demonstrate that Adam optimizer significantly outperforms SGD, achieving **96.33%** top-1 accuracy with ResNet-50 (+21.33% improvement over SGD), while ShuffleNet reaches **92.13%** accuracy with lower computational cost. Second, I conduct an ablation study on classifier head designs for transfer learning, testing four variants: (1) baseline linear layer, (2) wider linear layer, (3) 3-layer MLP, and (4) SEBlock-enhanced linear layer. Counter to initial hypotheses, the simplest **single linear layer** achieves the highest accuracy (**93.33%**), outperforming more complex variants by up to 7.54%. We identify **overparameterization** and **feature distortion** as primary causes of performance degradation in nonlinear classifiers.

Keywords: optimizer,Resnet,Image classification

1 Introduction

Deep learning models, particularly Convolutional Neural Networks (CNNs), have revolutionized image classification tasks by leveraging transfer learning from large pre-trained models. However, when adapting these models to small datasets, the design of the classifier head—the final layers responsible for class prediction—plays a critical role in balancing performance and generalization. In this study, I explore the impact of different classifier architectures on fine-tuning performance using a small-scale flower classification dataset (2,800 images, 5 classes). Ifig begin with Exercise 1, where I fine-tune a pre-trained ResNet-18 model with a single linear layer as the classifier head. This baseline approach achieves strong accuracy (90Building on this, Exercise 2 conducts an ablation study to examine how modifications to the classifier architecture influence model performance. We compare: A wider linear layer (increased capacity), A multilayer perceptron (MLP) with BatchNorm and Dropout (nonlinear mapping), and A Squeeze-and-Excitation (SE) block + linear layer (channel attention mechanism). Our findings reveal that simpler classifiers (single linear layer) outperform more complex variants, suggesting that excessive complexity can lead to overfitting, training instability, or distortion of pre-trained features in small-data regimes. These results emphasize the principle that ”less is more” in fine-tuning scenarios where the backbone already provides strong discriminative features. This study provides practical insights into classifier design for transfer learning, particularly in resource-constrained applications where parameter efficiency and stability are crucial. By systematically evaluating different architectures, we offer guidelines for optimizing model adaptation while avoiding unnecessary complexity.

, .

2 Related Work: Evolution of Image Classification Techniques

The field of image classification has undergone significant advancements through the development of deep convolutional neural networks (CNNs). Early breakthroughs were achieved by **AlexNet** [?], which demonstrated the power of deep architectures by winning the ImageNet challenge in 2012. This work established the effectiveness of **ReLU activations** and **GPU-accelerated training** for large-scale visual recognition tasks.

A major leap forward came with the introduction of **Residual Networks (ResNets)** [1], which addressed the degradation problem in very deep networks through identity shortcut connections. The authors showed that residual learning enables training of networks with over 100 layers while maintaining accuracy, with **ResNet-152** achieving a top-5 error of just 4.49% on ImageNet [?]. This architecture became foundational for many subsequent vision tasks, including object detection and segmentation [?].

The current state-of-the-art continues to build upon these fundamental advances, with residual learning remaining a cornerstone of modern architectures. As demonstrated in Table **??**, the error rate reduction from 2012 to 2018 highlights the rapid progress enabled by these innovations.

3 Method

3.1 Exercise 1: Model Architectures and Optimization

3.1.1 Model Selection and Architecture

We evaluated three CNN architectures chosen for their balance between computational efficiency and representational power:

- **ResNet-18/50** [1]: Utilize residual blocks with identity shortcuts to mitigate gradient vanishing. ResNet-50 employs bottleneck layers (1×1 convolutions) for reduced FLOPs while maintaining accuracy.
- **ShuffleNet** [2]: Lightweight architecture using grouped convolutions and channel shuffling, ideal for resource-constrained scenarios.

3.1.2 Optimization Strategies

The learning rate of SGD optimizer is 0.1, and the momentum is 0.9, the learning rate of Adam optimizer is 0.01.

3.1.3 Performance Analysis

Key observations from Table **??**:

- Adam improved ResNet-50 accuracy by 21.33% (75.0% → 96.33%)
- ShuffleNet's lower capacity limited peak accuracy (92.13% vs. ResNet-50's 96.33%)
- SGD underperformed due to suboptimal learning rate scheduling

3.2 Exercise 2: Classifier Head Ablation Study

3.2.1 Baseline Model (Linear Layer)

Advantages:Minimal parameters (512×5=2,560),Preserves pretrained feature semantics,Low overfitting risk for small datasets (∼2,800 images)

Table 1: Classifier Architectures

Variant	Parameters	Accuracy (%)
Baseline (Linear)	2.5K	93.33
Wider Linear	525K	93.16
MLP	394K	85.79
SEBlock + Linear	35K	92.28

3.2.2 Variants and Implementations

3.2.3 Key Findings

- **Simpler is better**: Baseline linear layer achieved highest accuracy (93.33%)
- **Overparameterization harms**:
  - MLP accuracy dropped by 7.54% despite 157× more parameters
  - SEBlock added complexity without benefits (92.28% vs. 93.33%)
- **Feature preservation**: Pretrained ResNet-18 features were already linearly separable

4 Experiment and Analysis

4.1 Comparative Study of Model Architectures and Optimizers (Exercise 1)

I evaluated three CNN architectures (**ResNet-18, ResNet-50, ShuffleNet_v2**) on a **flower classification dataset** (5 classes, ~2,800 images) using two optimizers (**SGD, Adam**). The models were fine-tuned with a **single linear classifier head**, and performance was measured by **test accuracy**.

Table 2: Test Accuracy (%) Across Models and Optimizers

Model	SGD Best Accuracy (%)	Adam Best Accuracy (%)
ResNet-18	73.08	92.13
ResNet-50	75.00	95.28
ShuffleNet	77.45	96.33

Optimization Strategy Comparison Key Observations:

- **Adam consistently outperformed SGD** across all models, achieving >**90% accuracy** compared to SGD’ s <**80%**.
- **SGD struggled with convergence**, likely due to:
  - Fixed learning rate requiring careful tuning.
  - Small batch size (default: 32) amplifying gradient noise.
- **Adam’ s adaptive learning rates** stabilized training, making it more robust for this task.

Model Architecture Comparison

- Under **SGD**, **ShuffleNet performed best (77.45%)**, likely due to its lightweight design being less prone to overfitting.
- Under **Adam**, **ResNet-50 achieved the highest accuracy (95.28%)**, benefiting from deeper feature extraction.
- **ShuffleNet + Adam reached 96.33%**, suggesting that efficient architectures can match deeper models when paired with adaptive optimization.

Discussion: Optimizer choice had a larger impact than model architecture—Adam improved accuracy by **~20%** over SGD, while model differences contributed **~5%**. **SGD may still be preferable** in scenarios where:Training data is extremely small (Adam' s adaptive updates may overfit noise).Batch sizes are very small (Adam's gradient estimation becomes unstable).

4.2 Ablation Study on Classifier Head Design (Exercise 2)

I modified the **classifier head** of ResNet-18 to study its impact on performance, testing:

1. **Baseline**: Single linear layer (`nn.Linear`).
2. **Wider Linear Layer**: Expanded hidden dimension (512 → 256 → 5).
3. **MLP with Dropout**: Added nonlinearity (`ReLU + Dropout`).
4. **SE-Block Integration**: Added Squeeze-and-Excitation before classification.

All experiments used **SGD** (to isolate architecture effects).

Table 3: Impact of Classifier Design on Accuracy (%)

Classifier Design	**Best Accuracy (%)**
Baseline (Linear)	93.33
Wider Linear Layer	93.16
MLP (ReLU + Dropout)	85.79
SE-Block + Linear	92.28

Key Findings:

- **Simpler classifiers performed best**—the baseline linear layer (93.33%) outperformed all modifications.
- **Adding complexity hurt performance**:
  - Wider linear layer (93.16%) marginally reduced accuracy, possibly due to information bottleneck (compressing 512 → 256).
  - MLP (85.79%) degraded significantly, likely due to overfitting (Dropout did not compensate for excessive parameters).
  - SE-Block (92.28%) slightly underperformed, suggesting channel attention was unnecessary for this simple task.

Discussion: "Less is more" in small-data fine-tuning: The pre-trained backbone (ResNet-18) already provided strong features, making complex classifiers redundant.
Overparameterization risks:MLPs introduced unnecessary nonlinearity, disrupting useful pre-trained features.SE-Blocks added computational cost without gains, as the dataset lacked fine-grained texture dependencies.
Practical Implication:For small datasets, a single linear layer is often sufficient; modifications should be validated rigorously.

4.3 Critical Analysis

Optimizer Sensitivity:Adam' s superiority was clear, but its hyperparameters (e.g., β_1, β_2) were not tuned—potential gains may be unexplored.SGD' s poor performance suggests the learning rate or momentum needed adjustment.**Classifier Over-Engineering**:Complex modifications (MLP, SE) degraded performance, highlighting misalignment with task simplicity.

4.4 Conclusion

This experiment demonstrated that Adam is superior to SGD for fine-tuning CNNs on small datasets. Model architecture matters less than optimization strategy in transfer learning.Simple linear classifiers often outperform complex heads when pre-trained features are strong.

5 Reference

K. He, X. Zhang, S. Ren, and J. Sun, ”Deep Residual Learning for Image Recognition,” *CVPR*, pp. 770–778, 2016.

X. Zhang, X. Zhou, M. Lin, and J. Sun, ”ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices,” *CVPR*, pp. 6848–6856, 2018.

P. Goyal et al., ”Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour,” *arXiv:1706.02677*, 2017.

D. P. Kingma and J. Ba, ”Adam: A Method for Stochastic Optimization,” *ICLR*, 2015.

J. Hu, L. Shen, and G. Sun, ”Squeeze-and-Excitation Networks,” *CVPR*, pp. 7132–7141, 2018.

J. Yosinski et al., ”How Transferable Are Features in Deep Neural Networks?,” *NeurIPS*, pp. 3320–3328, 2014.