Efficient Semi-Supervised Learning for Handwritten Digit Classification using FixMatch

Wenbin Huang
hartmann_psi@sjtu.edu.cn

Abstract

We explore the use of FixMatch[1], a state-of-the-art semi-supervised learning framework, for handwritten digit classification under limited labeled data. Our setup involves a dataset containing only 200 labeled MNIST[2] images (20 images for each category) and 10,000 unlabeled images with images out of distribution (80% MNIST, 20% EMNIST-Letters[3]). We evaluate several lightweight image classification models, including CNN, ResNet18[4], MobileNetV2[5], ShuffleNetV2[6], and Vision Transformer (ViT)[7], under a unified FixMatch training protocol. Empirical results show that with proper data augmentation and training strategies, ResNet18 achieves over 99% test accuracy, demonstrating FixMatch's strong potential under extreme low-resource settings. We further analyze the impact of model architecture, data augmentation, and optimizer choices on training efficiency and generalization.

Introduction

Background. Supervised deep learning models typically require large labeled datasets to achieve high accuracy. However, in many practical applications such as medical imaging, OCR, or low-resource language processing, labeled data is extremely limited. Semi-supervised learning (SSL) aims to leverage unlabeled data to improve generalization by combining supervised and unsupervised losses. Among SSL methods, FixMatch[1] has emerged as a simple yet powerful framework that combines consistency regularization with confidence-based pseudo-labeling.

Task Overview. In this work, we explore to apply FixMatch to the task of handwritten digit recognition on MNIST[2]. We simulate a low-resource environment with only very few labeled data for each category but a large set of unlabeled ones. To simulate the noisy working environment in real-time tasks, we added EMNIST-letters[3] images into the unlabeled dataset. Thus the model must learn meaningful features by utilizing a large set of unlabeled images. Our goal is to explore how model architecture and training strategies affect final performance under such constraints.

Methodology

Dataset Setup. Labeled Data: 200 MNIST images, 20 per class.

Unlabeled Data: 10,000 images, composed of 8000 MNIST + 2000 EMNIST-Letters.

Test Data: 10,000 MNIST test images.

We use standard preprocessing (resizing to 28x28 and normalization). For ViT, inputs are interpolated to 224x224.

Models Evaluated. We benchmark the following architectures:

SimpleCNN: Baseline convolutional model with two Conv-Pool blocks.

ResNet18[4]: Standard residual network with modified input for grayscale.

MobileNetV2[5]: Lightweight model with depthwise separable convolutions.

ShuffleNetV2[6]: Highly efficient model optimized for mobile.

ViT-Tiny[7]: Transformer-based image classifier adapted for grayscale.

FixMatch Framework. On a mixed dataset with batch size B for labeled data and μB for unlabeled data, FixMatch minimizes a combined loss which is defined as follows.[1]

Supervised Loss: Cross-entropy on labeled data.

$$\ell_s := \frac{1}{B}\sum_{b=1}^{B} H(p_b, p_m(y|x_b)).$$

Unsupervised Loss: Confidence-filtered cross-entropy on strongly-augmented unlabeled data with pseudo-labels generated from weakly-augmented views. The hyperparameter τ is the confidence threshold for pseudo-labeling.

$$\ell_u := \frac{1}{\mu B} \sum_{b=1}^{\mu B} \mathbb{I}(\max(q_b) > \tau) H(\hat{q}_b, q_b).$$

The total loss is a weighted sum of the supervised and unsupervised losses, which is defined as:

$$\ell := \ell_s + \lambda \cdot \ell_u$$

where λ is a hyperparameter controlling the balance between the two losses.

According to the suggestions[1] in Fixmatch paper, we choose $\lambda = 1.0$ and conventional SGD optimizer with momentum instead of Adam[8]. We also use a cosine learning rate decay schedule[9].

We enhance the standard FixMatch protocol by:

- Applying RandAugment[10] for strong augmentation.
- Introducing dynamic confidence thresholding.
- Optionally disabling horizontal flip, which is detrimental on MNIST[2].

We use PyTorch[11] for implementation.

Experiments and Results

Training Configuration. Optimizer: SGD with momentum=0.9, weight_decay=5×10^{-4}, nesterov=True;

Learning Rate: 0.03 with cosine decay;

Batch Sizes: 64 for labeled, 128 for unlabeled unless noted otherwise;

Epochs: 1000 unless noted otherwise;

Dynamic Threshold Strategy: Caculate the threshold τ based on current epoch n and hyperparameter k, $\tau_{\max}$ and $\tau_{\min}$:

$$\tau := \max\left(\tau_{\max},\ \tau_{\min} + (1 - \tau_{\min}) \cdot \left(1 - e^{-\frac{n}{k}}\right)\right)$$

in which $\tau_{\max} = 0.95$, $\tau_{\min} = 0.5$, $k = 50$.

Other hyperparameter settings can be found in the code repository.

Performance Comparison. The following table summarizes the performance of different models on the MNIST test set after 1000 epochs of training. The accuracy is reported as the percentage of correctly classified images.

Model	Parameters	Accuracy (%)
SimpleCNN	~225K	~97.3
ResNet18	~11M	~**99.0**
MobileNetV2	~2.2M	~98.2
ShuffleNetV2	~351K	~97.0
ViT-Tiny	~5.4M	-

In our experiment, ResNet18 outperforms all other models in accuracy and convergence speed. ShuffleNet underperforms in this small-scale setting, and ViT without pretraining fails to converge within 1000 steps.

To explore the model capacity of our method in extreme low-resource settings, we also evaluate the performance of CNN and ResNet18 under one-shot condition (only **1** labeled data per class) within **100** training epochs. Despite the challenging setting, CNN achieved an accuracy of **94.5%** and ResNet18 **96.3%** respectively. The results indicate that our method is capable of achieving high accuracy even with extremely limited labeled data.

Impact of Augmentation and Flip Removal. Using RandAugment and 4x augmentation per labeled sample improves early convergence: ResNet18 reaches 95% accuracy in only 100 epochs.

Removing horizontal flip improves accuracy across all models (e.g., 5↔2 confusion reduced), and further improves convergence speed, after which ResNet18 reaches 97% accuracy in only 50 epochs.

Loss Curves. We plot the training loss curves in first 100 epochs for selected models as follows:

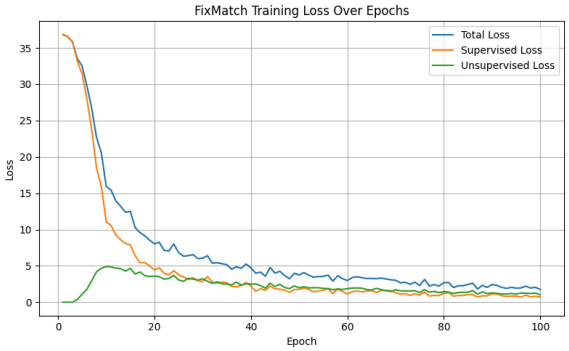

Figure 1: SimpleCNN

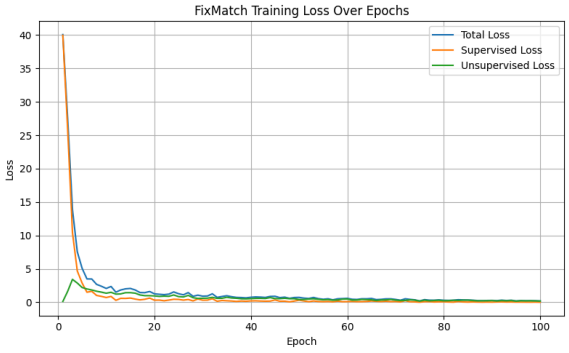

Figure 2: ResNet18

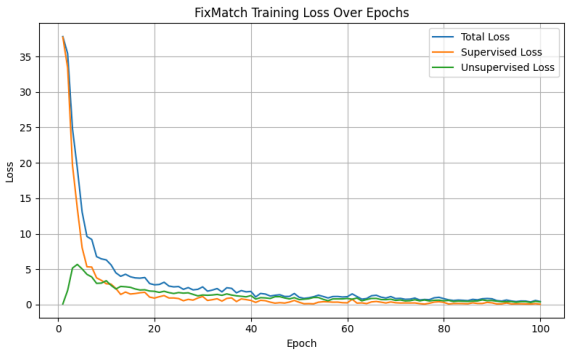

Figure 3: MobileNetV2

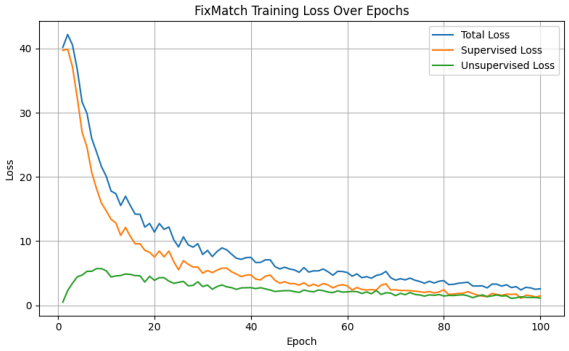

Figure 4: ShuffleNetV2

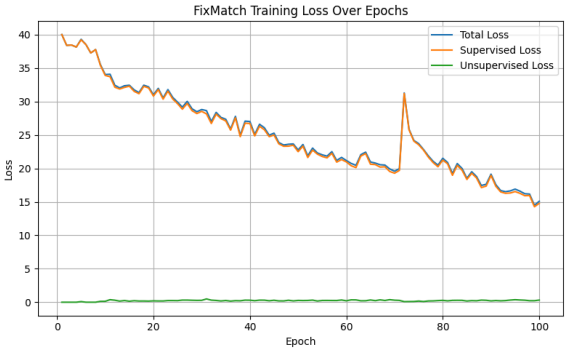

Figure 5: ViT-Tiny

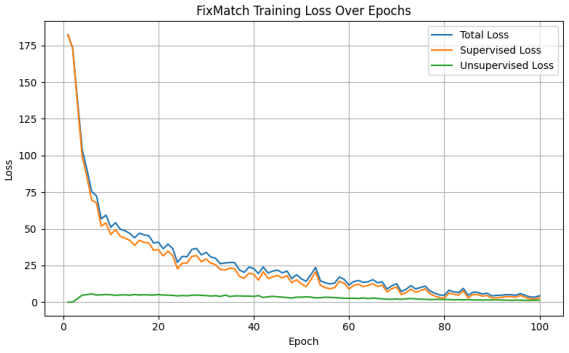

Figure 6: SimpleCNN Under Extreme Low-Resource Condition

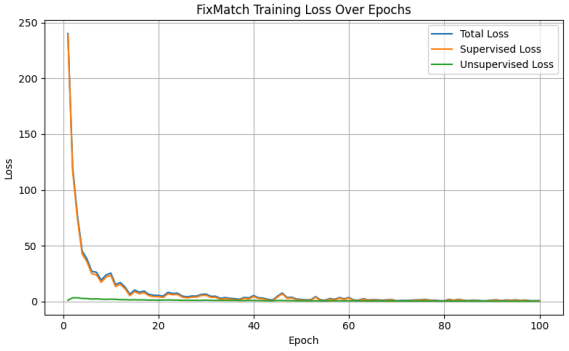

Figure 7: ResNet18 Under Extreme Low-Resource Condition

Discussion and Conclusion

This work shows that FixMatch is highly effective for handwritten digit classification with extremely limited labels. Among all tested architectures, ResNet18 shows the best trade-off between capacity, speed, and final accuracy. SimpleCNN also performs competitively with much fewer parameters. Removing inappropriate augmentations like horizontal flip significantly boosts performance. Future work includes pretraining ViT or applying contrastive pretraining for stronger feature extractors.

Bibliography

[1] K. Sohn *et al.*, "FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence," *CoRR*, 2020, [Online]. Available: https://arxiv.org/abs/2001.07685

[2] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998, doi: 10.1109/5.726791.

[3] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik, "EMNIST: an extension of MNIST to handwritten letters," *CoRR*, 2017, [Online]. Available: http://arxiv.org/abs/1702.05373

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *CoRR*, 2015, [Online]. Available: http://arxiv.org/abs/1512.03385

[5] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation," *CoRR*, 2018, [Online]. Available: http://arxiv.org/abs/1801.04381

[6] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design," *CoRR*, 2018, [Online]. Available: http://arxiv.org/abs/1807.11164

[7] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *CoRR*, 2020, [Online]. Available: https://arxiv.org/abs/2010.11929

[8] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *CoRR*, 2014, [Online]. Available: https://api.semanticscholar.org/CorpusID:6628106

[9] I. Loshchilov and F. Hutter, "SGDR: Stochastic Gradient Descent with Restarts," *CoRR*, 2016, [Online]. Available: http://arxiv.org/abs/1608.03983

[10] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "RandAugment: Practical data augmentation with no separate search," *CoRR*, 2019, [Online]. Available: http://arxiv.org/abs/1909.13719

[11] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," *CoRR*, 2019, [Online]. Available: http://arxiv.org/abs/1912.01703