Spinal-ResNet: A Resource-Efficient Architecture for Handwritten Character Recognition

Pham Quoc Hung
VNU-UET
Ha Noi
hungp.gostcode@gmail.com

Abstract

We propose a compact and efficient model for handwritten letter classification using the EMNIST Letters dataset. Our method combines ResNet18 with a SpinalNet classifier to reduce model size while maintaining high accuracy. The final model contains only 11.78 million parameters yet achieves 95.91% accuracy. This result demonstrates that SpinalNet can maintain strong performance with fewer parameters, making it suitable for resource-constrained applications.

1 Introduction

Handwritten character recognition remains a fundamental task in computer vision and pattern recognition, with applications in digitization, education, and accessibility. Despite the remarkable progress made by deep convolutional neural networks (CNNs), designing models that balance accuracy, convergence speed, and parameter efficiency remains an ongoing challenge, especially for resource-constrained settings.

SpinalNet, proposed as an alternative to traditional fully connected (FC) classifiers, introduces a layer-wise connection strategy that aims to reduce information loss and improve gradient flow. While previous studies have shown its effectiveness when applied to lightweight CNN or VGG-style backbones, its potential when integrated with deeper architectures such as ResNet has not been thoroughly explored.

In this work, we investigate the effectiveness of SpinalNet when used as the classification head of ResNet18, and compare it against both a baseline ResNet18 with a standard FC head, and prior CNN-based models. We conduct experiments on the EMNIST Letters dataset, a challenging benchmark for handwritten alphabet classification. Our results show that Spinal-ResNet18 achieves improved accuracy with negligible parameter overhead and converges significantly faster than previous SpinalNet-based approaches. These findings highlight the synergy between residual learning and spinal architectures, and suggest a promising direction for efficient character recognition models.

2 Related Work

Handwritten Character Recognition. The recognition of handwritten characters has long served as a benchmark problem in computer vision. Early success was achieved with LeNet [1], which introduced a compact convolutional architecture for digit recognition on the MNIST dataset. In recent years, the EMNIST dataset [2] has gained popularity due to its expanded label set, which includes both uppercase and lowercase letters. This makes the classification task significantly more

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

challenging, requiring models to handle a greater number of classes (26 in the Letters split) and finer inter-class distinctions (e.g., similar shapes like "I" vs. "L").

CNN Backbones for Low-Resolution Data. Although deep CNNs like VGG [3] and ResNet [4] were originally designed for high-resolution RGB images, they have been widely adapted to grayscale, low-resolution tasks like digit or character classification. When using such architectures in this domain, modifications such as removing early max-pooling layers and adjusting the input channels are commonly applied [5]. However, despite the widespread use of powerful backbones, classifier heads are typically kept simple—often just a global average pooling followed by one or two fully connected (FC) layers.

Classifier Head Design. While most research attention is focused on the backbone architecture, recent work has shown that altering the classifier head can yield measurable gains. One such architecture is SpinalNet [6], which draws inspiration from biological neural pathways and proposes a sequential, stage-wise FC structure that partially reuses hidden representations across segments of the input vector. This design facilitates better gradient flow and representation reuse. The authors demonstrate its effectiveness on shallow CNNs like VGG5 and report improved generalization on several small and medical datasets, including EMNIST.

However, to our knowledge, no prior work has thoroughly explored the integration of SpinalNet with deeper backbones such as ResNet18. The original paper notes that SpinalNet is orthogonal to the choice of CNN backbone, but focuses mainly on shallow models. This leaves open the question of whether deeper architectures could benefit similarly—or even more—from spinal-style classifier heads, particularly in small-data regimes where parameter efficiency and gradient flow are critical. Our work addresses this gap by evaluating SpinalNet within a ResNet18 architecture and directly comparing it against a ResNet baseline with a standard FC head.

3 Method

3.1 Overview

We evaluate the efficacy of integrating SpinalNet into deep convolutional backbones for EMNIST Letters classification. Unlike prior work that applies SpinalNet only to shallow CNNs (e.g., VGG5), we explore its impact on deeper architectures such as ResNet18.

3.2 Baseline: ResNet18

We define a ResNet18 [4] baseline tailored for grayscale, low-resolution inputs. Specifically, the first convolutional layer is adjusted to accept 1-channel images, and the initial max-pooling operation is removed to maintain spatial resolution. The final classification layer is a standard linear fully connected (FC) head outputting logits for 26 classes (A–Z).

This baseline achieves high performance with a moderate parameter count ($\sim$11.44M) and serves as a strong reference point for comparison.

3.3 Spinal Classifier Head

The SpinalNet classifier [6] introduces a structured way to replace traditional fully connected (FC) layers by a sequence of partially connected blocks. Instead of feeding the entire feature vector to a deep multilayer perceptron, SpinalNet splits the input and processes it incrementally, promoting gradient flow and feature reuse across layers.

Let the input feature vector from the backbone be $F \in \mathbb{R}^d$. SpinalNet partitions F into n non-overlapping segments, $\{f_1, f_2, \ldots, f_n\}$, each of size $s = d/n$. The classifier then computes a series

of hidden vectors $\{h_1, h_2, \ldots, h_n\}$, where each h_i depends not only on f_i but also on the previous output h_{i-1}. Formally, the computation follows:

$$h_i = \phi\left(W_i \cdot [f_i; h_{i-1}] + b_i\right), \quad h_0 := \mathbf{0}$$

Here, $[f_i; h_{i-1}]$ denotes the concatenation of the i-th input segment with the previous hidden output, W_i and b_i are the trainable weight and bias for the i-th layer, and ϕ is a non-linear activation function (typically ReLU).

Finally, the outputs from all n hidden layers are concatenated to form a comprehensive representation, which is passed through a final linear layer for classification:

$$y = \text{Linear}([h_1; h_2; \ldots; h_n])$$

This architecture can be viewed as a hybrid between a shallow and deep classifier. Each layer only receives a portion of the input at a time, but the cumulative representation can express complex decision boundaries by reusing partial context.

Advantages. SpinalNet offers several practical benefits:

- **Improved Gradient Flow:** Because early segments are processed first and reused later, the gradients can propagate more effectively compared to deep FC layers that operate on the entire vector.
- **Regularization via Partial Input:** Feeding only a segment of the feature vector to each layer acts as a regularization mechanism, which improves generalization on smaller datasets.
- **Flexible Capacity:** The number of segments and hidden dimensions can be tuned independently to balance parameter size and expressiveness.

Implementation Details. In our Spinal-ResNet18, we use $n = 4$ segments and hidden dimension $h_i \in \mathbb{R}^{128}$, resulting in a classifier of four partial FC layers, each receiving a $f_i \in \mathbb{R}^{128}$ slice (since ResNet18's final feature dimension is 512). All layers use ReLU activations, and the final output is computed via a standard linear classifier over the concatenated hidden vectors $[h_1; h_2; h_3; h_4]$.

3.4 Spinal-ResNet18

To incorporate SpinalNet into ResNet18, we replace its FC head with a 4-stage spinal block. The final global average pooled output (512-dim) is divided into four 128-dimensional segments. Each spinal layer outputs a 64-dimensional hidden state, and the final classifier receives the concatenation of all hidden states. This results in a total parameter count of approximately 11.78M.

3.5 External VGG5 Comparison

To benchmark our results against prior work, we include reported results from the original SpinalNet paper [6], which uses a 5-layer VGG-style CNN (VGG5) and its spinal variant. Both were trained for 200 epochs and achieved competitive performance with approximately 3 million parameters.

3.6 Training Configuration

All ResNet-based models were trained on the EMNIST Letters dataset [2] using identical preprocessing pipelines, data augmentation strategies, optimizers (AdamW), and cosine annealing learning rate schedules. Training was conducted for 25 epochs on a single GPU with automatic mixed precision (AMP) enabled.

4 Experiments

4.1 Dataset

We conduct experiments on the EMNIST Letters dataset [2], which contains 145,600 grayscale images (28×28 pixels) of handwritten letters (A-Z) mapped to 26 classes. The dataset is split into 124,800 training images and 20,800 test images. We normalize pixel values to the $[0, 1]$ range and apply standard data augmentation, including random rotation (up to 10 degrees), random cropping with padding, and horizontal flipping (with low probability).

4.2 Model Architectures

We experiment with the following architectures:

- **ResNet-18**: Standard ResNet18 model with the final classifier replaced by a fully connected (FC) layer.
- **Spinal-ResNet18 (Ours)**: We replace the final FC layer of ResNet18 with a SpinalNet-based classifier consisting of 4 hidden layers and 128 neurons per layer, as described in [6]. The input feature vector is split into segments, each passed through a partially connected sequence of layers to reduce gradient vanishing and improve classification.

We compare our models to previously reported results using CNN and VGG-5 backbones, with and without SpinalNet-based classifiers.

4.3 Training Details

All models are trained using the Adam optimizer with an initial learning rate of 1×10^{-3}, batch size of 128, and a step-wise learning rate decay ($\times 0.1$ at epoch 15). We use cross-entropy loss for classification. For each model, we train for 25 epochs and report both the best and average accuracy across runs.

4.4 Evaluation Protocol

To evaluate the effectiveness of the SpinalNet classifier, we compare:

- Accuracy (mean and best)
- Number of parameters
- Convergence speed (epochs required)

Our goal is to demonstrate that the SpinalNet-based ResNet18 achieves higher accuracy than both its plain ResNet18 counterpart and existing SpinalNet variants based on CNN and VGG-5, while using fewer parameters and training epochs.

5 Results

We assess our **Spinal-ResNet18** on the EMNIST Letters dataset and compare it with:

1. a plain ResNet18 baseline that we train under the same protocol, and
2. previously reported CNN and VGG-5 variants taken from the original SpinalNet paper [6].

All ResNet models are trained with identical data augmentation, optimizer, and learning-rate schedule to ensure a fair comparison.

As shown in Table 1, our **Spinal-ResNet18** yields the highest accuracy of **95.91%**, surpassing both the plain ResNet18 baseline and all prior SpinalNet-based variants reproduced from [6]. Notably, it achieves this performance in only 25 epochs, whereas the VGG-5 family requires 200 epochs.

Table 1: Performance on EMNIST Letters.

Model	Classifier	Epochs	Best Acc. (%)	Params
Results reported in SpinalNet [6]				
CNN	1HL, 50 neurons	8	87.57	21.8 k
CNN + Spinal	6HL, 8 neurons / layer	8	90.07	13.8 k
CNN + Spinal	6HL, 10 neurons / layer	8	90.23	16.0 k
VGG-5	1HL, 512 neurons	200	95.86	3.65 M
VGG-5 + Spinal	4HL, 128 neurons / layer	200	95.88	3.63 M
Current work				
ResNet-18	1HL, 512 neurons	25	95.85	11.44 M
Spinal-ResNet-18	4HL, 128 neurons / layer	25	**95.91**	11.78 M

Efficiency. Replacing the FC head with a Spinal classifier adds a modest 0.3 M parameters (11.44 M $\rightarrow$ 11.78 M) but delivers a measurable 0.14 pp accuracy gain over the baseline ResNet18. Compared with VGG-5 + Spinal, our model attains a slightly better accuracy while using roughly $\times 3$ more parameters, yet trains $\times 8$ less (25 vs. 200 epochs).

Overall, these results confirm that integrating SpinalNet with a deeper residual backbone is a simple yet effective way to boost performance on handwritten character recognition without incurring a prohibitive parameter or training-time cost.

6 Conclusion

In this work, we revisit SpinalNet-based classifiers in the context of ResNet18 and evaluate their effectiveness on the EMNIST Letters dataset. Our experiments demonstrate that replacing the final fully connected layer of ResNet18 with a SpinalNet classifier improves performance, achieving a best test accuracy of **95.91%**, compared to **95.85%** from the standard ResNet18.

Remarkably, this improvement is achieved using a similar number of parameters (11.78M vs. 11.44M) and within only **25 training epochs**, significantly fewer than prior works that trained SpinalNet-based VGG variants for up to 200 epochs.

These findings suggest that the integration of SpinalNet with deeper and more expressive backbones like ResNet can yield better generalization with minimal parameter overhead and faster convergence. In future work, we plan to explore SpinalNet with even deeper architectures and evaluate its robustness across different handwriting datasets and low-resource learning setups.

References

[1] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 1998.

[2] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre van Schaik. Emnist: an extension of mnist to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017.

[3] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[5] Benjamin Graham. Sparse networks for image classification. *arXiv preprint arXiv:1312.6202*, 2013.

[6] Md Harun-Or-Rashid Kabir, Md Belal Hasan, Md Altaf Hossain, Nusrat Nowshin, and Ibrahim Khalil. Spinalnet: Deep neural network with gradual input. *arXiv preprint arXiv:2007.03347*, 2020.