Enhancing MP-MLP with Patch Mixing

Taehyeon Kim
Department of Computer Engineering, Kyonggi University
Suwon, Republic of Korea
dannykim05@kyonggi.ac.kr

***Abstract*—Lightweight vision architectures are important for image classification in resource-constrained environments. Among CNN-free approaches, multi-layer perceptron (MLP)-based models provide a simple and computationally efficient alternative. MP-MLP is a lightweight vision model that divides an image into non-overlapping micro-patches and applies a shared MLP to each patch independently. While this design is simple and efficient, it does not explicitly model interactions across patches before classification. To address this limitation, we introduce a simple patch mixing module inspired by the token-mixing idea of MLP-Mixer. The proposed module is applied after local patch encoding and performs mixing along the patch dimension through a lightweight MLP block. Experimental results on MNIST and SVHN show that the proposed method consistently improves performance over the baseline MP-MLP, with especially large gains on the more complex SVHN dataset. These results suggest that patch mixing is an effective way to enhance lightweight MLP-based vision models.**

I. Introduction

Deep learning has significantly advanced computer vision, with convolutional neural networks (CNNs) becoming the dominant approach for image recognition tasks [1], [2]. CNNs are highly effective because they exploit local spatial structure through convolution and weight sharing. However, they also rely on a strong inductive bias based on convolutional operations.

More recently, transformer-based models such as Vision Transformer (ViT) showed that image classification can also be performed by treating images as sequences of patches and modeling long-range dependencies with self-attention [3]. These models achieve strong performance, but often require relatively high computational cost and large-scale training data.

To explore simpler alternatives, recent research has investigated MLP-based vision architectures. MLP-Mixer demonstrated that competitive image classification performance can be achieved using only MLP layers without convolution or self-attention [4]. A key idea in MLP-Mixer is that, besides per-token feature processing, information should also be mixed across tokens.

MP-MLP is a lightweight model developed from this perspective [7]. It divides an input image into non-overlapping micro-patches and applies the same shared MLP to each patch independently. This design yields a simple and efficient local processing mechanism with low architectural complexity. However, in the baseline MP-MLP, the encoded patch features are directly flattened and fed to the classifier, without explicit interaction across patches.

This limitation becomes more important when the visual pattern is complex. For simple datasets, local patch features may already be sufficient. In contrast, for datasets with greater appearance variation and more complicated spatial structure, inter-patch interaction can become more important.

In this paper, we propose a simple extension of MP-MLP that introduces a patch mixing module after local patch encoding. Inspired by MLP-Mixer, the proposed module mixes information along the patch dimension while preserving the simplicity of the original architecture. The main contributions of this work are as follows:

- We identify the lack of explicit inter-patch interaction as a limitation of the baseline MP-MLP.
- We introduce a simple patch mixing module inspired by MLP-Mixer and apply it to MP-MLP.
- We empirically show that the proposed modification improves performance on both MNIST and SVHN, with especially large gains on the more challenging dataset.

II. Related Work

A. CNN-Based Vision Models

CNNs have long been the standard backbone for visual recognition because they efficiently learn hierarchical local features. AlexNet demonstrated the effectiveness of deep CNNs on large-scale image classification [1]. Later, ResNet introduced residual connections that enabled much deeper networks and further improved recognition accuracy [2].

B. Transformer-Based Vision Models

Vision Transformers replaced convolutional feature extraction with patch tokenization and self-attention [3]. This allows global dependencies to be modeled explicitly and has shown strong performance on image classification benchmarks. However, self-attention can be computationally expensive when the number of tokens becomes large, which motivates the search for simpler alternatives.

C. MLP-Based Vision Models

MLP-based architectures aim to simplify visual modeling by replacing convolution and attention with fully connected layers. MLP-Mixer is a representative example, using one MLP for per-token feature transformation and another for cross-token mixing [4]. Follow-up works such as ResMLP [5] and gMLP [6] explored variants with residual connections and gating mechanisms, showing that pure MLP architectures can be expressive and practical. These results suggest that explicit

cross-token interaction is an important component of effective MLP-based vision models.

MP-MLP [7] is closely related to this direction, but focuses on a particularly lightweight design based on shared local patch encoding without cross-patch interaction. Our work extends this baseline by adding a patch mixing stage after local encoding.

III. Method

A. Overview

The proposed model extends MP-MLP [7] with a patch mixing block. The overall pipeline consists of four stages: (1) patch extraction, (2) local patch encoding using a shared MLP, (3) patch mixing across encoded patches, and (4) final classification.

In the baseline MP-MLP, each patch is independently encoded and the resulting features are directly flattened for classification. In contrast, the proposed model inserts an additional patch mixing block after local encoding so that information can be exchanged across patches before classification.

B. Patch Extraction

Let the input image be represented as

$$X \in R^{H\times W\times C}, \tag{1}$$

where H, W, and C denote the image height, width, and number of channels, respectively. The image is divided into non-overlapping patches of size $p \times p$. Each patch is flattened into a vector

$$x_i \in R^{p^2C}, \quad i = 1, 2, \ldots, N, \tag{2}$$

where the number of patches is

$$N = \left\lfloor \frac{H}{p} \right\rfloor \left\lfloor \frac{W}{p} \right\rfloor. \tag{3}$$

For MNIST, we use grayscale images of size 28×28 with $C = 1$. With patch size $p = 4$, this gives $N = 49$. For SVHN, we use RGB images of size 32×32 with $C = 3$. With the same patch size, this gives $N = 64$.

C. Shared Local Patch Encoding

Each flattened patch is processed independently using the same shared two-layer MLP:

$$h_i = f_\theta(x_i), \tag{4}$$

where

$$h_i = W_2\, \phi(W_1 x_i + b_1) + b_2. \tag{5}$$

Here, $\phi(\cdot)$ denotes the ReLU activation function.

The encoded patch feature dimension is denoted by d. Thus, each patch vector is mapped from R^{p^2C} to R^d. All encoded patch features are stacked as

$$H = [h_1, h_2, \ldots, h_N] \in R^{N\times d}. \tag{6}$$

D. Patch Mixing

After local patch encoding, the encoded patch features are arranged as

$$H \in R^{N\times d}, \tag{7}$$

where N is the number of patches and d is the feature dimension. To enable inter-patch interaction, we transpose the feature map to

$$H^\top \in R^{d\times N}, \tag{8}$$

and apply an MLP along the patch dimension, following the token-mixing idea of MLP-Mixer [4]. The patch mixing operation is defined as

$$H' = \left(H^\top + \mathrm{MLP}(H^\top)\right)^\top. \tag{9}$$

In this way, each patch feature can incorporate information from other patch positions before the final prediction layer, while the overall architecture remains simple.

E. Final Classification

In the baseline MP-MLP, the encoded patch matrix is flattened and directly passed to the classifier. In the proposed model, the mixed patch matrix is flattened instead. The classifier then produces the final class logits.

Thus, the only architectural difference between the baseline and the proposed model is the insertion of the patch mixing block between local patch encoding and final classification.

F. Model Summary

The full architecture of the proposed model can be summarized as

$$X \to \{x_i\}_{i=1}^N \to \{h_i\}_{i=1}^N \to H \to H' \to \hat{y}. \tag{10}$$

More explicitly, the processing flow is:

Image $\to$ Micro-patches $\to$ Shared Local MLP $\to$ Patch Mixing $\to$ Classifier.

IV. Experiments

A. Datasets

We evaluate the proposed method on MNIST and SVHN. MNIST is a grayscale handwritten digit dataset with simple visual patterns and clean backgrounds. SVHN is a more challenging real-world digit dataset containing richer variation and more complex spatial structure.

B. Training Setup

All models are trained using the Adam optimizer with learning rate 10^{-3} for 5 epochs. Cross-entropy loss is used for classification.

For MNIST, we use patch size $p = 4$, hidden dimension $d = 16$, and batch size 64. For SVHN, we use patch size $p = 4$, hidden dimension $d = 32$, and batch size 128.

For both datasets, the baseline model corresponds to the original MP-MLP without patch mixing, while the proposed model includes the additional patch mixing module after local patch encoding. All other settings are kept the same for fair comparison.

TABLE I
PERFORMANCE COMPARISON BETWEEN THE BASELINE MP-MLP AND THE PROPOSED PATCH-MIXING MODEL.

Dataset	Baseline	Proposed	Gain (pp)
MNIST	95.91%	97.83%	+1.92
SVHN	76.07%	82.13%	+6.06

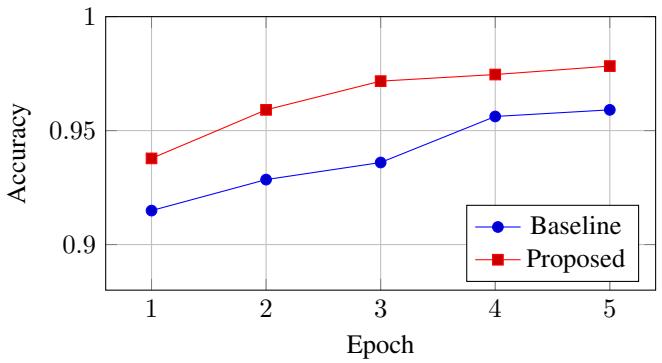

Fig. 1. Test accuracy over training epochs on MNIST.

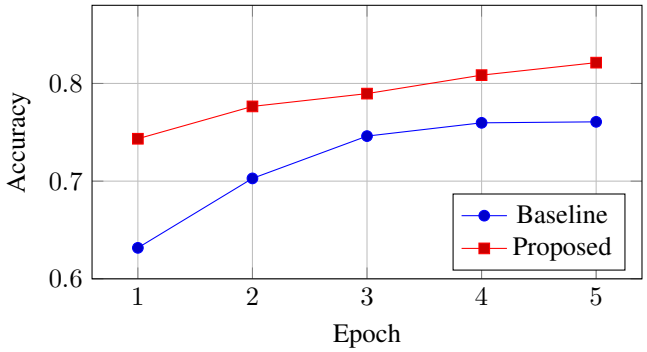

Fig. 2. Test accuracy over training epochs on SVHN.

C. Main Results

As shown in Table I, the proposed patch mixing module improves classification accuracy on both datasets. On MNIST, patch mixing yields a gain of 1.92 percentage points. On SVHN, the gain is much larger at 6.06 percentage points.

These results indicate that explicit interaction across patches is useful even for relatively simple handwritten digit classification, and becomes substantially more important for visually complex data such as SVHN.

D. Training Dynamics

Figure 1 and Figure 2 show the test accuracy over training epochs on MNIST and SVHN, respectively. On MNIST, the proposed model consistently outperforms the baseline across all epochs, though the gap is relatively small given that MNIST is a simple dataset. On SVHN, the improvement is more pronounced, and the accuracy gap appears from the first epoch and remains throughout training. These dynamics suggest that patch mixing improves both learning speed and final representation quality, with greater benefit on more complex data.

E. Analysis

The results on MNIST and SVHN provide a meaningful contrast. On MNIST, the baseline model already performs well, but patch mixing still provides a noticeable gain. On SVHN, the improvement is much larger, indicating that inter-patch interaction becomes more valuable when the dataset contains more complex visual patterns.

Furthermore, on MNIST, the baseline shows a relatively abrupt accuracy increase between epochs 3 and 4, whereas the proposed model converges more smoothly across epochs, suggesting that patch mixing may also contribute to more stable training dynamics.

Another important observation is that this improvement is achieved without using convolution or attention. The proposed method only introduces a lightweight MLP-based mixing block across patch positions. This makes the approach attractive for settings where simplicity and efficiency are important.

V. LIMITATIONS AND FUTURE WORK

Although the proposed method improves performance, it also has limitations. First, the current experiments are limited to MNIST and SVHN. Second, the architecture remains shallow and lightweight by design, which may limit its representational power on more challenging benchmarks. Third, we do not yet provide a full ablation study over patch size, hidden dimension, or mixing depth. Finally, the current paper focuses on classification accuracy and does not include explicit comparisons of parameter count or computational cost.

In future work, we plan to evaluate the model on additional datasets such as Fashion-MNIST, CIFAR-10, and CIFAR-100. We also plan to investigate deeper or more efficient patch mixing variants while preserving the lightweight design philosophy.

VI. CONCLUSION

We presented an enhanced MP-MLP architecture with a patch mixing module. The original MP-MLP is efficient and simple, but it lacks explicit interaction across encoded patches before classification. By introducing a simple MLP-based mixing block along the patch dimension, the proposed method enables inter-patch communication while maintaining architectural simplicity.

Experimental results on MNIST and SVHN show consistent accuracy gains, with especially large improvement on the more complex SVHN dataset. These findings suggest that patch mixing is an effective way to enhance compact MLP-based vision models.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2012, pp. 1097–1105.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 770–778.

[3] A. Dosovitskiy *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations (ICLR)*, 2021.

[4] I. Tolstikhin *et al.*, "MLP-Mixer: An all-MLP architecture for vision," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021, pp. 24261–24272.

[5] H. Touvron *et al.*, "ResMLP: Feedforward networks for image classification with data-efficient training," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 5314–5321, 2023.

[6] H. Liu, Z. Dai, D. R. So, and Q. V. Le, "Pay attention to MLPs," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2021, pp. 9204–9215.

[7] T. Kim, "CNN-Free Lightweight Vision Model Using Weight-Shared MLP on Micro-Patches," *engrXiv* (preprint), Mar. 2026. DOI: 10.31224/6542.