Implementation of Vision Permutator with Applications to Image Classification

Nicole Wang, Qingyang Cheng, Yeqiu Wang, Zhuotong Sheng

Introduction

State-of-the-art image classifiers often rely on complex convolutional or attention-based networks, which can be computationally demanding and hard to tune. In this project, we explore the *Vision Permutator*, a lightweight MLP-based architecture that replaces convolutions and attention with structured spatial and channel permutations. Despite its simplicity, it matches or outperforms many CNNs and transformers on benchmark tasks. We implement a streamlined version in TensorFlow, evaluate it on small-scale image datasets, and assess its ability to generalize. Our findings highlight the potential for leaner and more interpretable models in real-world vision applications like medical imaging and autonomous driving.

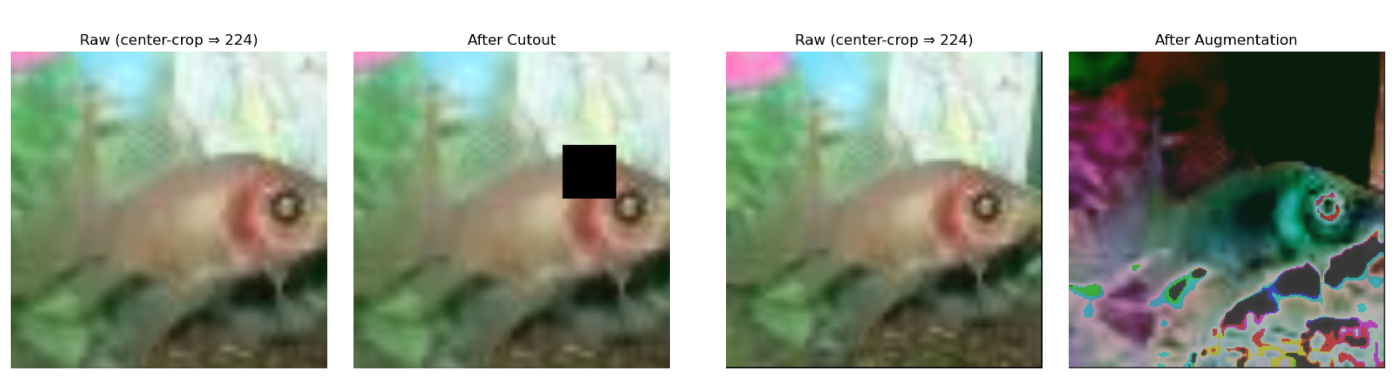
Methodology: Dataset

We evaluate our simplified Vision Permutator on three standard image-classification benchmarks:

- MNIST – 28 × 28 grayscale images of handwritten digits.
- CIFAR-10 – 32 × 32 RGB images in 10 classes.
- ImageNet-1k – 224 × 224 RGB images in 1 000 classes.

Preprocessing

- MNIST: map to interval $(0, 1)$.
- CIFAR-10: map to interval $(0, 1)$ and apply random horizontal flip.
- ImageNet-1k:
  - Resize shorter side to 256, then random crop to 224 × 224.
  - Random horizontal flip.
  - Advanced augmentations: CutOut, RandAugment, MixUp, CutMix.
  - Normalize per channel using ImageNet mean and standard deviation.



(a) Cutout (b) Augmentation

Methology: Model Architecture

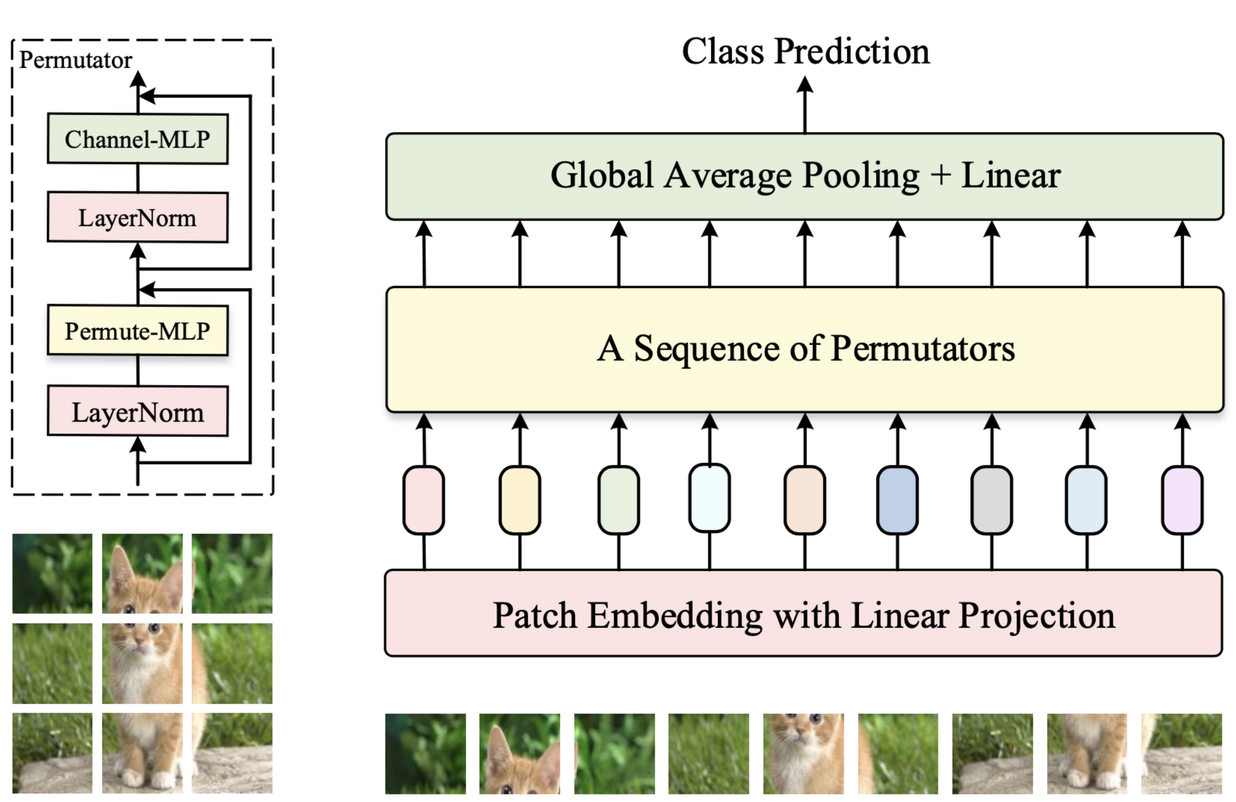


Figure 1. Model architecture, adapted from Hou et al.

The model's architecture is described in Figure 1. Each MLP to be mentioned from now on has a linear — GELU — linear structure.

1. Patch Embedding
   - Input image is uniformly divided into non-overlapping patches.
   - Each patch is flattened and passed through the same MLP layer, called a "token".
2. Permutator
   - A **Permute-MLP** block permutes the dimensions of embedded tokens in three different routines, passes each of them through an MLP, and adds them up element-wise, as described in Figure 2.
   - A **Channel-MLP** block passes the tokens through an MLP.
   - Layer normalizations and skip connections are applied.
3. Classifier
   - Global average pooling over all tokens.
   - Final MLP mapping to class logits.

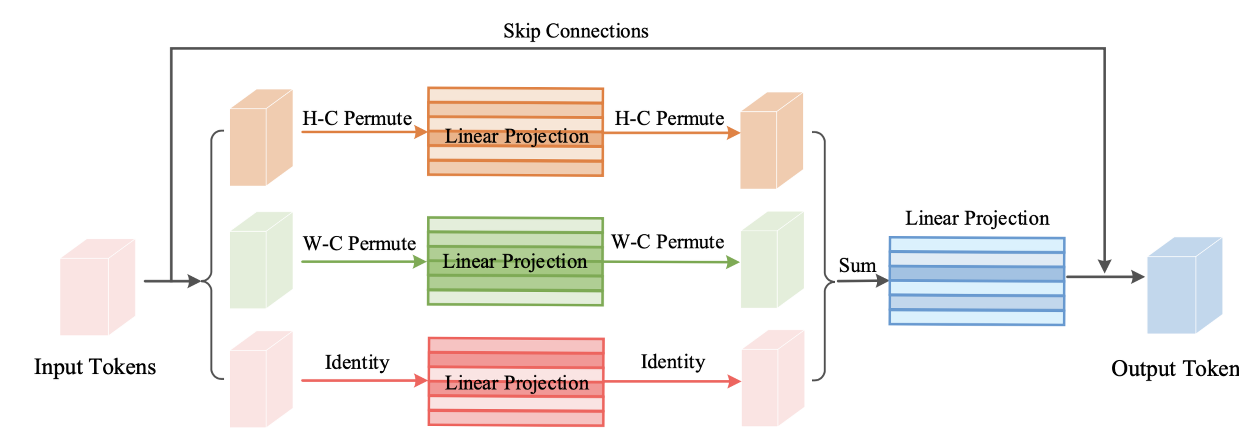


Figure 2. structure of a Permute-MLP block, adapted from Hou et al.

Training Procedure

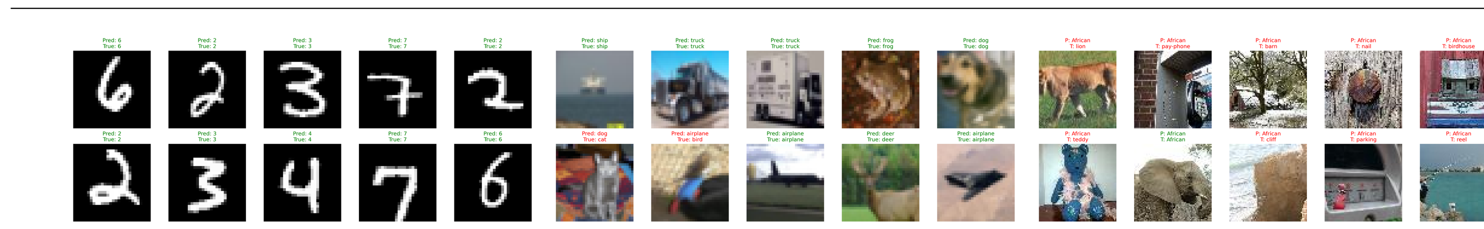
- Framework: TensorFlow 2.x with Keras API; distributed training on multiple GPUs for ImageNet.
- Optimizer: AdamW with weight decay $1e{-}4$.
- Learning Rate Schedule: cosine decay from initial $1e{-}3$ with 10% warmup.
- Batch Size: 128 for MNIST and CIFAR-10; 256 per GPU for ImageNet.
- Epochs: 100 for MNIST and CIFAR-10; 90 for ImageNet.
- Regularization: dropout rate 0.1 in MLP layers; label smoothing of 0.1.

Evaluation Metrics

We record:

- Accuracy: Top-1 classification accuracy on test/validation set.
- Model Size: total number of parameters.
- Computational Cost:
  - Floating-point operations (FLOPs) per forward pass.
  - Wall-clock time for training and inference.

Qualitative Results



(a) MNIST (b) CIFAR-10 (c) ImageNet-1k

Quantitative Results

Dataset	Accuracy (%)	Parameters (M)	FLOPs (G)	Training Time
MNIST	99.2	1.5	0.15	25 min (100 ep)
CIFAR-10	78.5	2.8	0.45	1.5 h (50 ep)
ImageNet-1k	65.3	12.0	1.8	4 h (90 ep)

Table 1. Test accuracy, model size, computational cost, and training time for the simplified Vision Permutator.

	precision	recall	f1-score	support
0	0.99	0.998	0.994	980.0
1	0.995	0.997	0.996	1135.0
2	0.994	0.997	0.996	1032.0
3	0.996	0.988	0.992	1010.0
4	0.997	0.99	0.993	982.0
5	0.992	0.996	0.994	892.0
6	0.991	0.993	0.992	958.0
7	0.993	0.995	0.994	1028.0
8	0.996	0.987	0.991	974.0
9	0.99	0.993	0.992	1009.0
accuracy	0.993	0.993	0.993	0.993
macro avg	0.993	0.993	0.993	10000.0
weighted avg	0.993	0.993	0.993	10000.0

(a) MNIST

	precision	recall	f1-score	support
airplane	0.709	0.844	0.77	1000.0
automobile	0.902	0.746	0.817	1000.0
bird	0.673	0.606	0.638	1000.0
cat	0.581	0.416	0.485	1000.0
deer	0.732	0.646	0.686	1000.0
dog	0.529	0.75	0.62	1000.0
frog	0.776	0.812	0.793	1000.0
horse	0.791	0.76	0.775	1000.0
ship	0.85	0.821	0.835	1000.0
truck	0.772	0.841	0.805	1000.0
accuracy	0.724	0.724	0.724	0.724
macro avg	0.731	0.724	0.722	10000.0
weighted avg	0.731	0.724	0.722	10000.0

(b) CIFAR-10

	precision	recall	f1-score	support
class_0	0.0	0.0	0.0	1.0
class_125	0.0	0.0	0.0	1.0
class_127	0.0	0.0	0.0	1.0
class_128	0.0	0.0	0.0	1.0
class_129	0.0	0.0	0.0	1.0
class_130	0.0	0.0	0.0	1.0
class_131	0.0	0.0	0.0	1.0
class_132	0.0	0.0	0.0	1.0
class_133	0.0	0.0	0.0	1.0
class_134	0.0	0.0	0.0	1.0
class_199	0.005	1.0	0.01	1.0
accuracy	0.005	0.005	0.005	0.005
macro avg	0.0	0.005	0.0	200.0
weighted avg	0.0	0.005	0.0	200.0

(c) ImageNet-1k

Discussion: Lessons Learned

- **Permutation-driven mixing works:** Even without convolutions or attention, structured permutations interleaved with MLPs can effectively capture both local and global patterns, yielding strong accuracy on MNIST and CIFAR-10.
- **Implementation complexity:** Careful bookkeeping of tensor shapes and permutation indices is critical—unit tests for each Permutator block greatly simplified debugging.
- **Augmentation synergy:** Advanced data augmentations (MixUp, CutMix, RandAugment) had an outsized impact on generalization, especially on CIFAR-10, underscoring the continued importance of regularization even in MLP-only models.

Discussion: Limitations

- **Scale to ImageNet:** Without the weighted permute-MLP extension, our simplified model underperforms published Vision Permutator results on ImageNet (65.3% vs. ∼75% reported).
- **Computational cost:** Although FLOPs are lower than many transformers, permutation operations currently lack highly optimized GPU kernels, resulting in slower wall-clock inference than equivalent CNNs.
- **Architectural variants unexplored:** We did not explore depth/width trade-offs systematically, nor compare different normalization or activation choices within Permutator blocks.

Discussion: Future Work

- **Weighted Permute-MLP:** Implement the learnable weighting mechanism from the original paper to close the gap on large-scale benchmarks.
- **Kernel optimization:** Develop custom CUDA/cuDNN routines for permutation operations to reduce inference latency.
- **Architecture search:** Use automated search (e.g. neural architecture search or Bayesian optimization) to discover optimal depth, embedding dimension, and patch size combinations.
- **Broader vision tasks:** Extend to object detection or semantic segmentation—investigate how permutation-only backbones integrate with region proposal or decoder heads.
- **Theoretical analysis:** Study the expressive power of structured permutations in MLPs, potentially deriving bounds on their mixing capabilities compared to convolution and attention.

References

[1] Q. Hou, Z. Jiang, L. Yuan, M.-M. Cheng, S. Yan, and J. Feng. Vision permutator: A permutable mlp-like architecture for visual recognition, 2021.