A Battle of Network Structures: An Empirical Study of CNN, Transformer, and MLP

University of Science and Technology of China, Microsoft Research Asia

2021.08.30

arXiv:2108.13002v1

Introduction

- Convolutional neural networks (CNNs) have been dominating the computer vision (CV) field since the renaissance of deep neural networks (DNNs).

- The transformer structure finally made its grand debut in CV last year.

- Interestingly, before the heat of Transformer dissipated, the structure of multi-layer perceptrons (MLPs)

Introduction

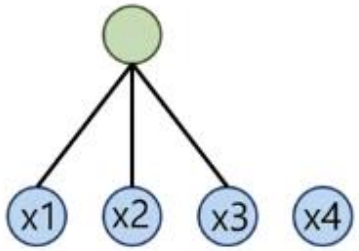
- However, as the reported accuracy on image classification benchmarks continues to increase by new network designs from various camps, no conclusion can be made as which structure among CNN, Transformer, and MLP performs the best or is most suitable for vision tasks.

- We first develop a unified framework called SPACH which adopts separate modules for spatial and channel processing.

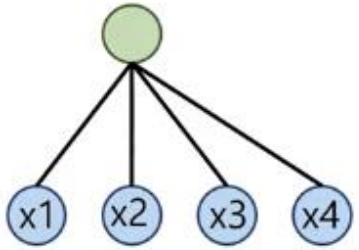
Comparison of Conv, SA, MLP

Properties	Convolution	Self-Attention	MLP
Sparse Connectivity	✓		
Dynamic Weight		✓	
Global Receptive Field		✓	✓

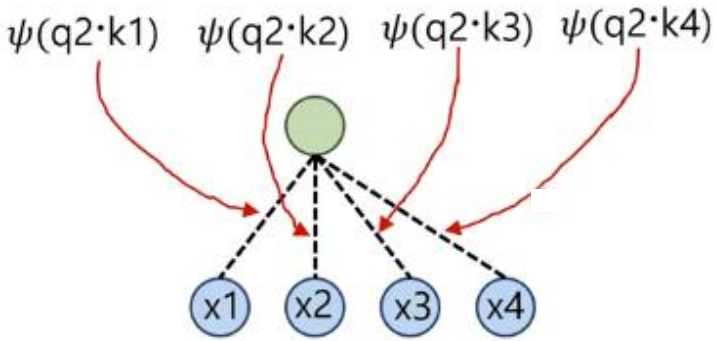
Table 2. Three desired properties in network design are seen in different network structures.



Convolution layer
- Locally connected
- Same weights for all inputs



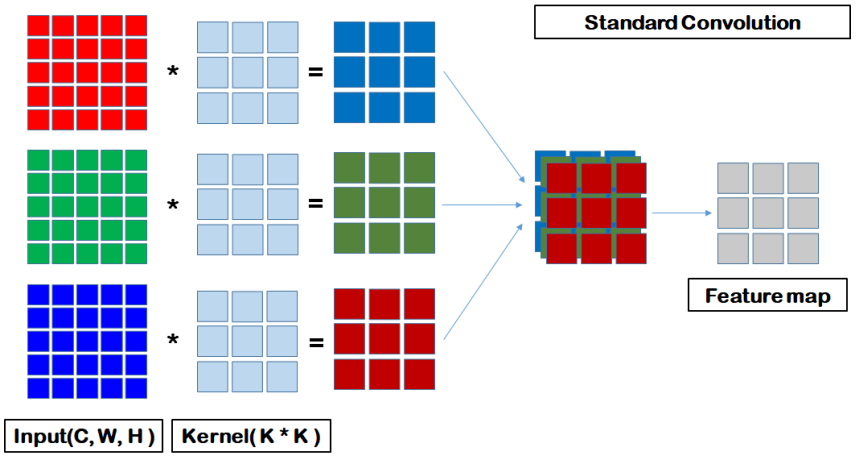
Dense layer
- Fully connected
- Same weights for all inputs

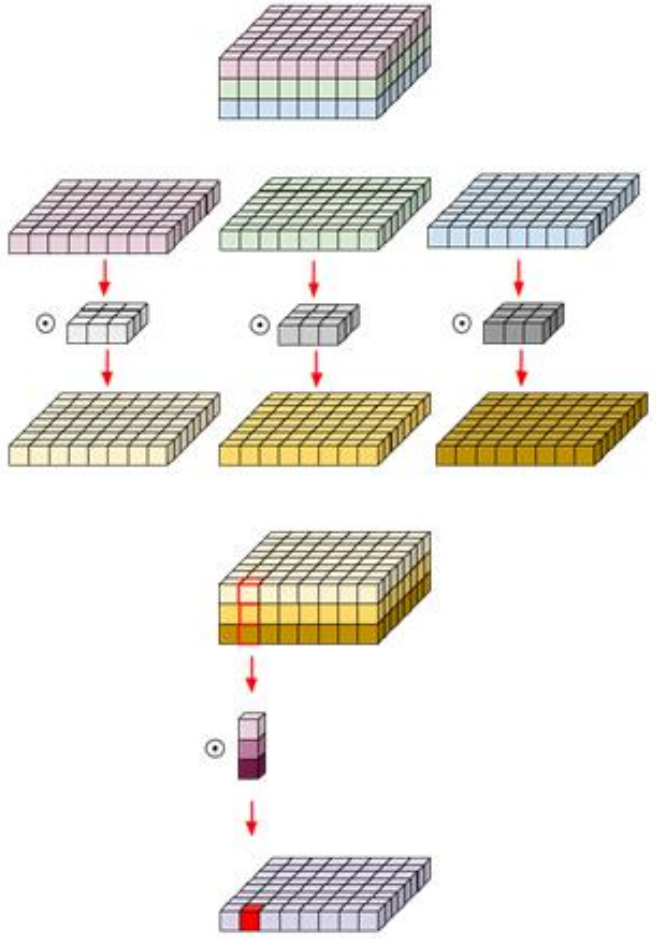


Attention layer
- Fully connected
- Different weights for all inputs

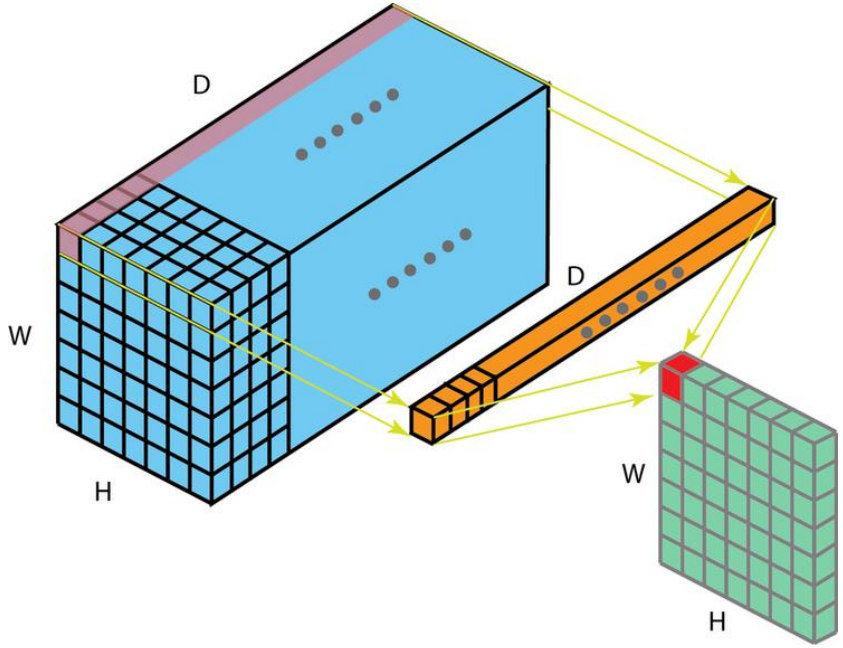
- https://www.youtube.com/watch?v=KQmZlxdnnuY

Background -CNN





Depthwise, Pointwise Convolution



1x1 Convolution

Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017
Network In Network, 2013

Background - Transformer

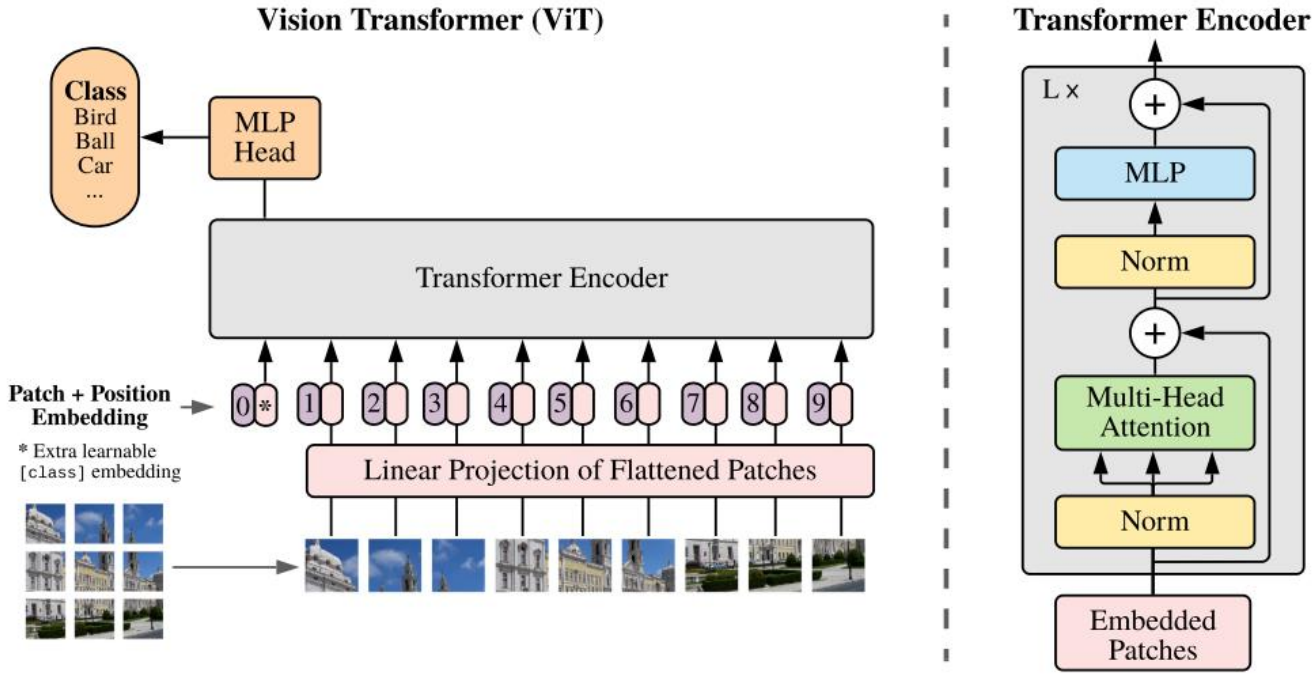


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

- An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, 2020

Yann LeCun
@ylecun

Well, not *actually* conv free.
1st layer: "Per-patch fully-connected" == "conv layer with 16x16 kernels and 16x16 stride"
other layers: "MLP-Mixer" == "conv layer with 1x1 kernels"

> **Neil Houlsby** @neilhoulsby · May 5
> New paper from Brain Zurich and Berlin!
>
> We try a conv and attention free vision architecture: MLP-Mixer (arxiv.org/abs/2105.01601)
>
> Simple is good, so we went as minimalist as possible (just MLPs!) to see whether modern training methods & data is sufficient...
>
> Show this thread

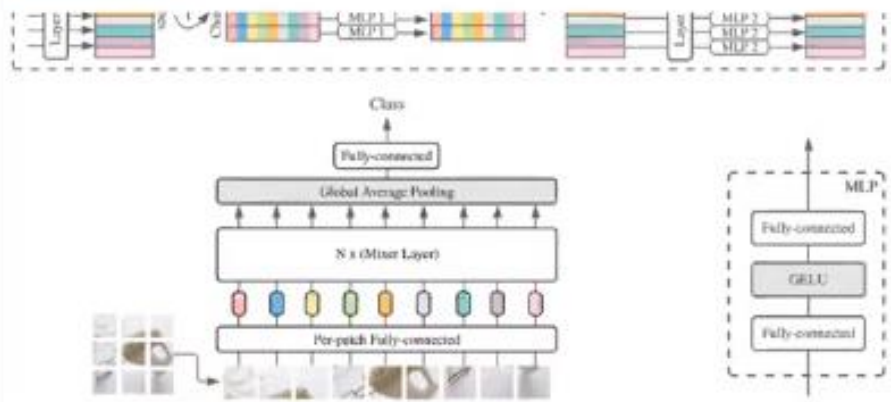


Figure 1: MLP-Mixer consists of per-patch linear embeddings, Mixer layers, and a classifier head. Mixer layers contain one token-mixing MLP and one channel-mixing MLP, each consisting of two fully-connected layers and a GELU nonlinearity. Other components include: skip-connections, dropout, layer norm on the channels, and linear classifier head.

2:44 PM · May 7, 2021 · Twitter Web App

74 Retweets **12** Quote Tweets **619** Likes

Background - MLP

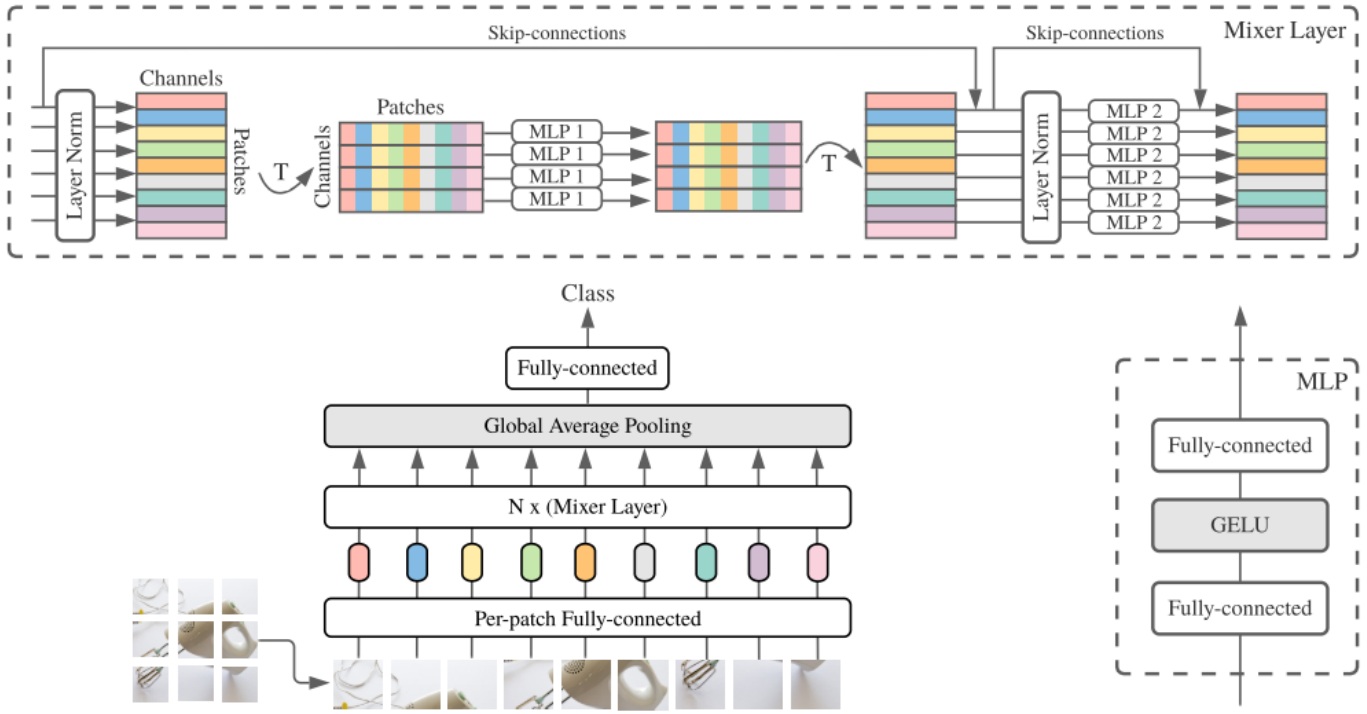


Figure 1: MLP-Mixer consists of per-patch linear embeddings, Mixer layers, and a classifier head. Mixer layers contain one token-mixing MLP and one channel-mixing MLP, each consisting of two fully-connected layers and a GELU nonlinearity. Other components include: skip-connections, dropout, and layer norm on the channels.

- MLP-Mixer: An all-MLP Architecture for Vision, 2021
- Do You Even Need Attention? A Stack of Feed-Forward Layers Does Surprisingly Well on ImageNet, 2021

SPACH Framework

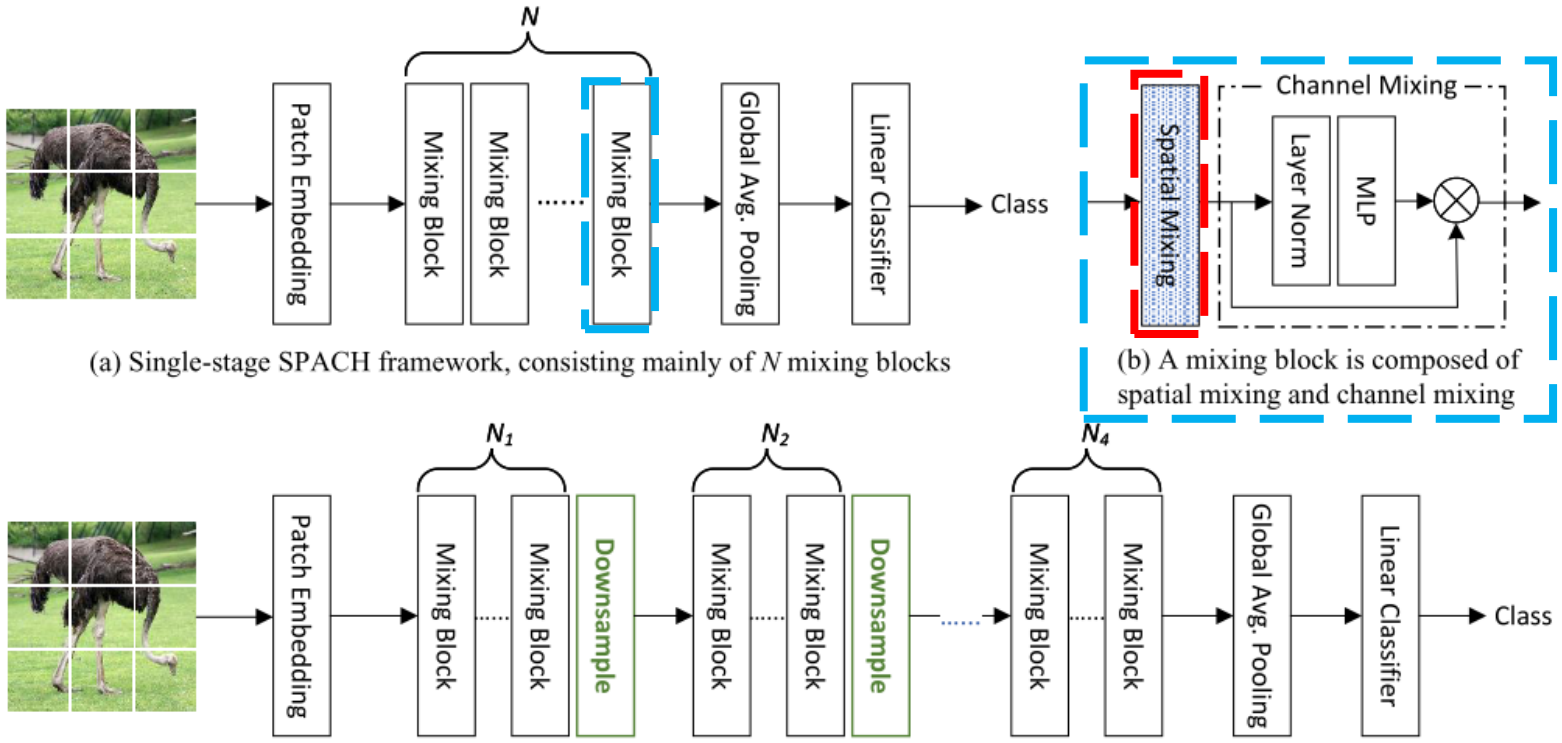


Figure 1. Illustration of the proposed experimental framework named SPACH.

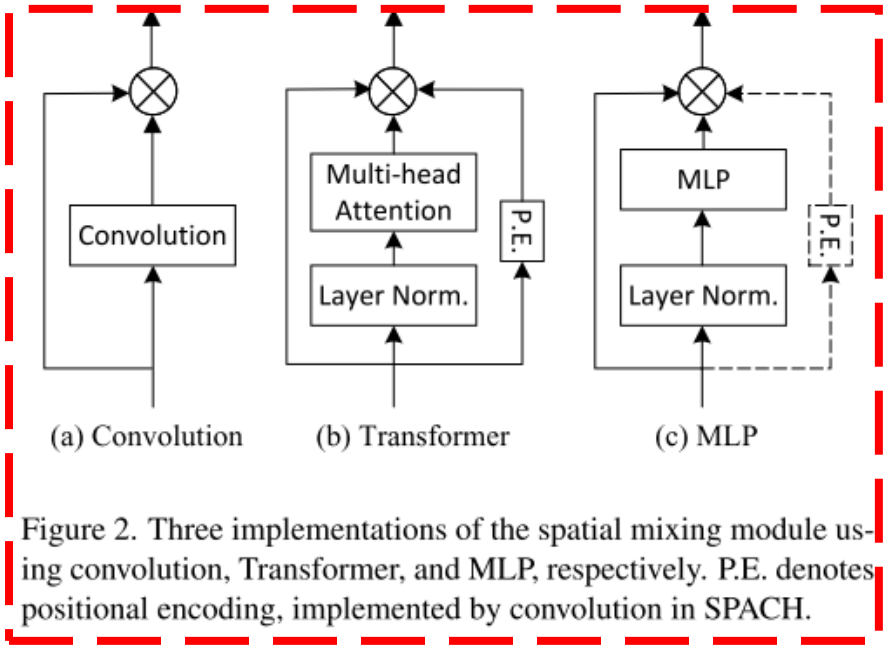


Figure 2. Three implementations of the spatial mixing module using convolution, Transformer, and MLP, respectively. P.E. denotes positional encoding, implemented by convolution in SPACH.

SPACH Framework

Model	SPACH-XXS	SPACH-XS	SPACH-S
Conv	$C = 384, R = 2.0, N = 12$	$C = 384, R = 2.0, N = 24$	$C = 512, R = 3.0, N = 24$
Transformer	$C = 192, R = 2.0, N = 12$	$C = 384, R = 2.0, N = 12$	$C = 512, R = 3.0, N = 12$
MLP	$C = 384, R = 2.0, N = 12$	$C = 384, R = 2.0, N = 24$	$C = 512, R = 3.0, N = 24$
Conv-MS	$C = 64, R = 2.0$ $N_s = \{2, 2, 6, 2\}$	$C = 96, R = 2.0$ $N_s = \{3, 4, 12, 3\}$	$C = 128, R = 3.0$ $N_s = \{3, 4, 12, 3\}$
Transformer-MS	$C = 32, R = 2.0$ $N_s = \{2, 2, 6, 2\}$	$C = 64, R = 2.0$ $N_s = \{3, 4, 12, 3\}$	$C = 96, R = 3.0$ $N_s = \{3, 4, 12, 3\}$
MLP-MS	$C = 64, R = 2.0$ $N_s = \{2, 2, 6, 2\}$	$C = 96, R = 2.0$ $N_s = \{3, 4, 12, 3\}$	$C = 128, R = 3.0$ $N_s = \{3, 4, 12, 3\}$

Table 1. SPACH and SPACH-MS model variants. C: feature dimension, R: expansion ratio of MLP in $\mathcal{F}_c$, N: number of mixing blocks of SPACH, N_s: number of mixing blocks in the i_{th} stage of SPACH-MS.

Due to the extremely high computational cost of Transformer and MLP on high-resolution feature maps, we implement the mixing blocks in the first stage with convolutions only.

Experiments

Network Scale	Model	Single-Stage				Multi-Stage			
		#param.	FLOPs	Throughput (image/s)	IN-1K Top-1 acc.	#param.	FLOPs	Throughput (image/s)	IN-1K Top-1 acc.
XXS	Conv	8M	1.4G	1513	72.1	5M	0.7G (-0.7G)	1576	73.3 (+1.2)
	Trans	4M	0.9G	1202	68.0	2M	0.5G (-0.4G)	1558	65.4 (-2.6)
	MLP	9M	1.8G	980	74.1	6M	0.9G (-0.9G)	1202	74.9 (+0.8)
XS	Conv	15M	2.8G	770	77.3	17M	2.8G (-0.0G)	602	80.1 (+2.8)
	Trans	15M	3.1G	548	78.4	14M	3.1G (-0.0G)	441	80.1 (+1.7)
	MLP	17M	3.5G	503	78.5	19M	3.4G (-0.1G)	438	80.7 (+2.2)
S	Conv	39M	7.4G	374	80.1	44M	7.2G (-0.2G)	328	81.6 (+1.5)
	Trans	33M	6.7G	328	81.7	40M	7.6G (+0.9G)	246	82.9 (+1.2)
	MLP	41M	8.7G	272	78.6	46M	8.2G (-0.5G)	254	82.1 (+3.5)

Table 3. Model performance of SPACH and SPACH-MS at three network scales.

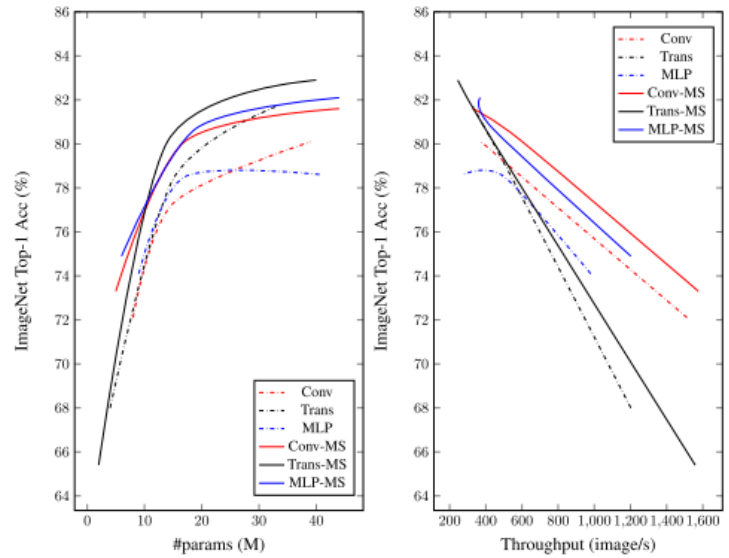


Figure 3. The multi-stage models (named with -MS suffix) always achieve a better performance than their single-stage counterparts.

Experiments

Model	#param.	FLOPs	throughput (image/s)	IN-1K Top-1 acc.
Trans-MS-S	40M	7.6G	246	82.9
Trans-MS-S$^-$	40M	7.6G	259	82.3
MLP-MS-S	46M	8.2G	254	82.1
MLP-MS-S$^-$	46M	8.2G	274	80.1

Table 4. Both Transformer structure and MLP structure benefit from local modeling at a very small computational cost. The superscription - indicates without local modeling.

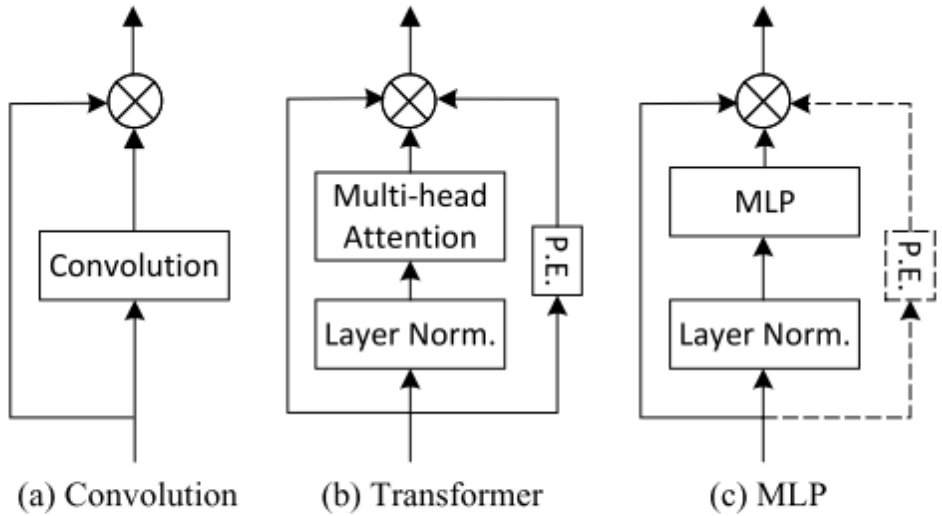


Figure 2. Three implementations of the spatial mixing module using convolution, Transformer, and MLP, respectively. P.E. denotes positional encoding, implemented by convolution in SPACH.

Experiments

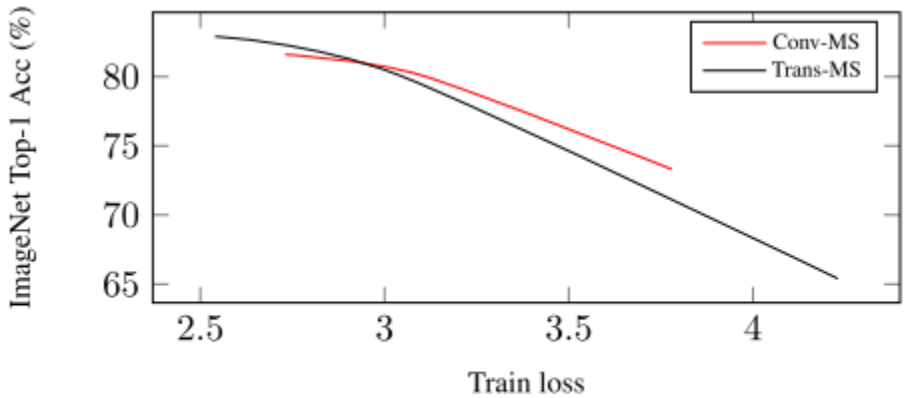


Figure 5. Conv-MS has a better generalization capability than Trans-MS as it achieves a higher test accuracy at the same training loss before the model saturates.

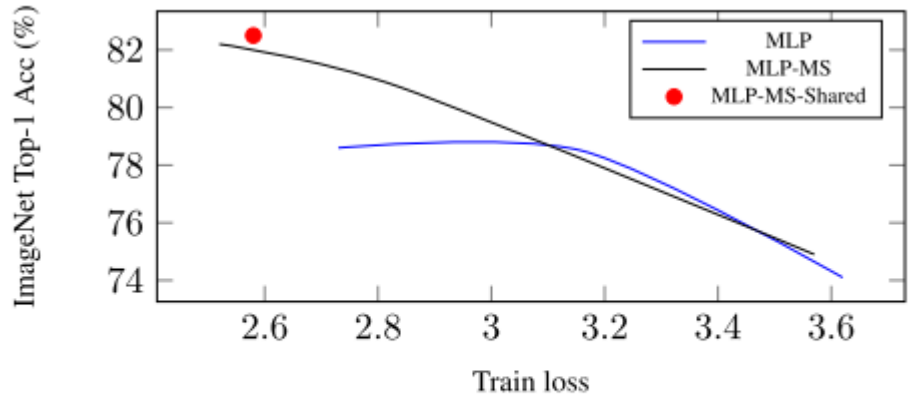


Figure 4. Illustration of the over-fitting problem in MLP-based models. Both multi-stage framework and weight sharing alleviate the problem.

Model	#param.	FLOPs	throughput (image/s)	IN-1K Top-1 acc.
MLP-S	41M	8.7G	272	78.6
+Shared	39M	8.7G	274	80.2
MLP-MS-S	46M	8.2G	254	82.1
+Shared	45M	8.2G	244	82.5

Table 5. The performance of MLP models are greatly boosted when weight sharing is adopted to alleviate over-fitting.

Proposed Method - Hybrid model

- In order to unleash the full potential of hybrid models, we further adopt the deep patch embedding layer (PEL) implementation

- Hybrid-MS-XS:
  It is based on Conv-MS-XS. The last ten layers in Stage 3 and the last two layers in Stage 4 are replaced by Transformer layers. Stage 1 and 2 remain unchanged.

- Hybrid-MS-S:
  It is based on Conv-MS-S. The last two layers in Stage 2, the last ten layers in Stage 3, and the last two layers in Stage 4 are replaced by Transformer layers. Stage 1 remains unchanged.

Model	#param.	FLOPs	IN-1K Top-1 acc.
CNN			
RegNetY-4G [25]	21M	4.1G	80.0
RegNetY-8G [25]	39M	8.0G	81.7
RegNetY-16G [25]	84M	16.0G	82.9
Transformer			
ViT-B/16* [9]	86M	-	77.9
DeiT-S [35]	22M	4.6G	79.8
DeiT-B [35]	86M	17.5G	81.8
Swin-T [21]	29M	4.5G	81.3
Swin-S [21]	50M	8.7G	83.0
Swin-B [21]	88M	15.4G	83.5
CaiT-XS24 [36]	26.6M	5.4G	81.8
CaiT-S36 [36]	68.2M	13.9G	83.3
CvT-13 [39]	20M	4.5G	81.6
CvT-21 [39]	32M	7.1G	82.5
MLP			
FF-Base [23]	62M	-	74.9
Mixer-B/16 [33]	79M	-	76.4
ResMLP-S24 [34]	30M	6.0G	79.4
ResMLP-B24 [34]	45M	23.0G	81.0
gMLP-S [20]	20M	4.5G	79.4
gMLP-B [20]	73M	15.8G	81.6
Ours			
Conv-MS-XS	17M	2.8G	80.1
Conv-MS-S	44M	7.2G	81.6
Trans-MS-XS	14M	3.1G	80.1
Trans-MS-S	40M	7.6G	82.9
Hybrid-MS-XS	28M	4.5G	82.4
Hybrid-MS-XS+	28M	5.6G	82.8
Hybrid-MS-S	63M	11.2G	83.7
Hybrid-MS-S+	63M	12.3G	83.9

Table 6. Comparison of different models on ImageNet-1K classification. Compared models are grouped according to network structures, and our models are listed in the last, Most models are pre-trained with 224x224 images, except ViT-B/16*, which uses 384x384 images.

Results

- Multi-stage design is standard in CNN models, but its effectiveness is largely overlooked in Transformer-based or MLP-based models.

- Local modeling is efficient and crucial. With only light-weight depth-wise convolutions, the convolution model can achieve similar performance as a Transformer model in our SPACH framework.

Results

- MLP can achieve strong performance under small model sizes, but it suffers severely from overfitting when the model size scales up.

- Convolution and Transformer are complementary in the sense that convolution structure has the best generalization capability while Transformer structure has the largest model capacity among the three structures.