MetaFormer is Actually What You Need for Vision

Yu, Weihao, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan

Sea AI Lab & National University of Singapore

April 3, 2023

Presented by Boyao Li

Overview

- Introduction
- Background
- MetaFormer
- Experiments and Future Work

Introduction

This paper abstracts transformers into a general architecture MetaFormer, and empirically demonstrates that the success of transformer/MLP-like models is largely attributed to the MetaFormer architecture rather than specific token mixers.

- PoolFormer: only employ a pooling operator as a weak token mixer for MetaFormer.
- Achieve competitive performance compared with the SOTA models using sophistic design of token mixers on multiple vision tasks.

Why matters?

- Inspire more future research dedicated to improving MetaFormer instead of the token mixer modules.
- PoolFormer could serve as a good start baseline for future MetaFormer architecture design.

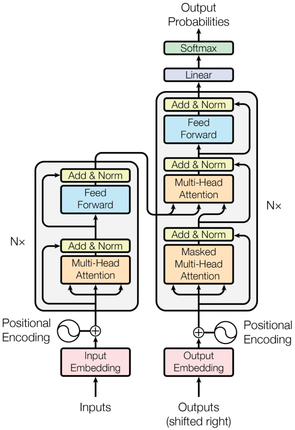
Background

- Transformer
- Vision Transformer (ViT)
- MLP-like Models

Transformer

Encoder and Decoder Stacks

- N(In this work $N = 6$) identical layers. Each layer has two sub-layers.
- Output for each sub-layer: $Y = \text{LayerNorm}(X + \text{SubLayer}(X))$



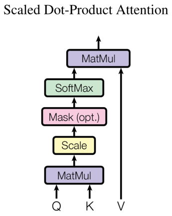
Transformer

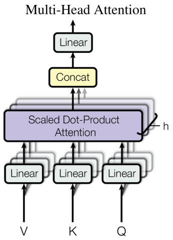
Scaled Dot-Product Attention

- Queries(d_k), keys(d_k) and values(d_v)
- $\text{Attention}(Q, K, V) = \text{softmax}(QK^T/\sqrt{d_k})V$

Multi-Head Attention

- $\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O$, where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$
- $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, $W_i^O \in \mathbb{R}^{hd_v \times d_{model}}$

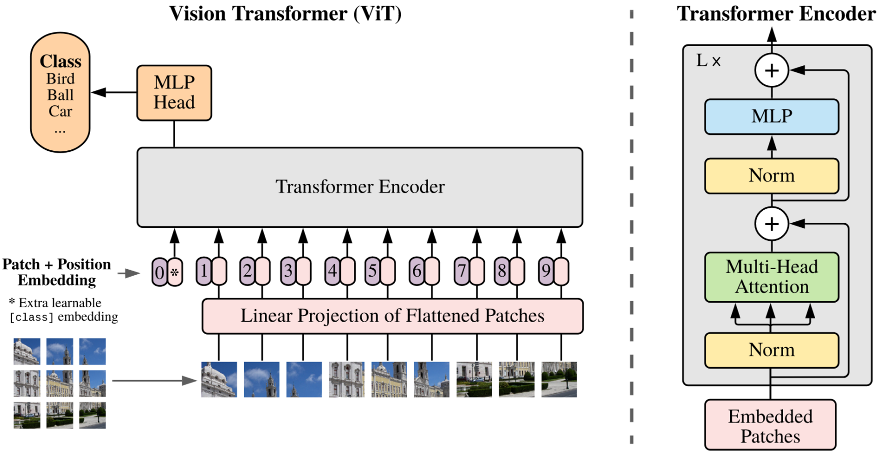




In this work, $d_{model} = 512, h = 8, d_{d\,v} = d_{model}/h = 64$.

Vision Transformer(ViT)

- A pure transformer applied on sequences of image patches can perform very well on image classification tasks, and the reliance on CNN may not be necessary.
- Yield modest accuracies of a few percentage points below ResNets of comparable size when pre-trained on mid-sized datasets such as ImageNet-1K.
- Reach/beat SOTA models when trained on larger datasets(14M-300M images) such as ImageNet-21K(14M images with 21K classes) and JFK-300M dataset.



Vision Transformer (ViT)
Class
Bird
Ball
Car
...
MLP Head
Transformer Encoder
Patch + Position Embedding
* Extra learnable [class] embedding
0 *
1
2
3
4
5
6
7
8
9
Linear Projection of Flattened Patches
Transformer Encoder
L x
MLP
Norm
Multi-Head Attention
Norm
Embedded Patches

Vision Transformer(ViT)

- Patch Embedding
  - Reshape a 2D (H, W) image $\boldsymbol{x} \in \mathbb{R}^{H \times W \times C}$ into a sequence of 2D (P, P) patches $\boldsymbol{x}_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$.
  - C is the number of the channels.
  - $N = HW/P^2$ is the number of the patches.
- Use standard learnable 1D position embeddings.
- Prepend a learnable embedding to the sequence of embedded patches $\boldsymbol{z}_0^0 = \boldsymbol{x}_{\text{class}}$ similar as BERT's CLS token. Only the class vector is used to predict the output.
- ViT has much less images-specific inductive bias than CNNs due to that self-attention layers are global.

Vision Transformer(ViT)

Related Work: Data-efficient Image Transformers (DeiT)

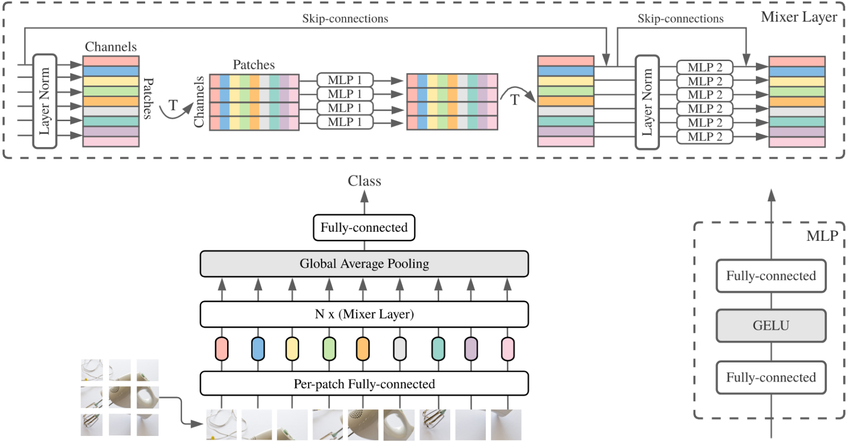
- Build upon ViT and produce competitive transformers by training on ImageNet only with less computing resource.
- New distillation strategy specific to transformers
  - Add a distillation token to the sequence of embedded patches to reproduce the label given by the "teacher" instead of true label on output of the network.
  - Both the class and distillation tokens are learned by backpropagation.

MLP-like Models

MLP-Mixer(and ResMLP)

- While convolutions and attentions are both sufficient for good performance, neither of them are necessary.
- Two types of MLP layers
  - token-mixing MLP: $\mathbb{R}^N \rightarrow \mathbb{R}^N$, applied across patches.
  - channel-mixing MLP: $\mathbb{R}^C \rightarrow \mathbb{R}^C$, applied independently to patches.
- Position embeddings are not included because token-mixing MLP are sensitive to the order of the input tokens.
- Attain competitive performance on image classification benchmarks when trained on large datasets or with modern regularization schemes.

MLP-Mixer

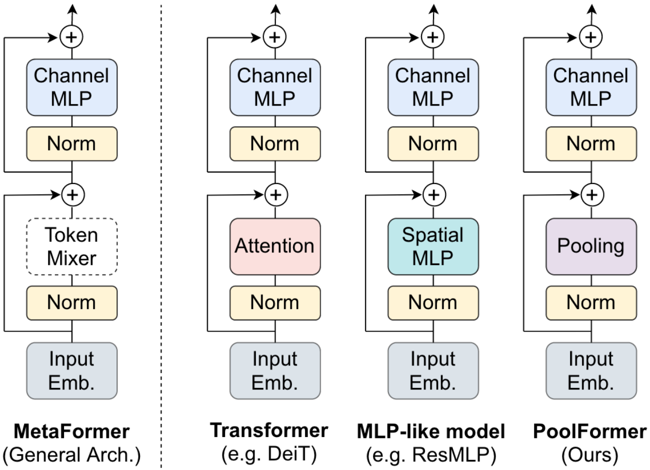

Skip-connections
Skip-connections
Mixer Layer
Channels
Layer Norm
Patches
T
Channels
Patches
MLP 1
MLP 1
MLP 1
MLP 1
T
Layer Norm
MLP 2
MLP 2
MLP 2
MLP 2
MLP 2
MLP 2
Class
Fully-connected
Global Average Pooling
N x (Mixer Layer)
Per-patch Fully-connected
MLP
Fully-connected
GELU
Fully-connected

MLP-like Models

Related Work: Spatial-shift MLP

- MLP-Mixer cannot achieve as outstanding performance as its CNN/ViT counterparts.
- Replace token-mixing MLP to some spatial-shift operations for the communications between patches.

MetaFormer


Channel MLP
Norm
Token Mixer
Norm
Input Emb.
Channel MLP
Norm
Attention
Norm
Input Emb.
Channel MLP
Norm
Spatial MLP
Norm
Input Emb.
Channel MLP
Norm
Pooling
Norm
Input Emb.
MetaFormer
(General Arch.)
Transformer
(e.g. DeiT)
MLP-like model
(e.g. ResMLP)
PoolFormer
(Ours)

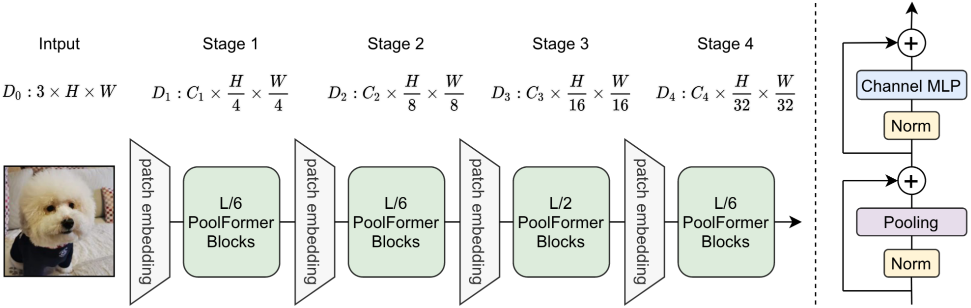
MetaFormer

- For an input image I, $X = \text{InputEmb}(I), X \in \mathbb{R}^{N \times C}$
- $Y = \text{TokenMixer}(\text{Norm}(X)) + X$. $\text{Norm}(\cdot)$ denotes normalization such as Layer Normalization or Batch Normalization.
- $Z = \sigma(\text{Norm}(Y)W_1)W_2 + Y$ where $W_1 \in \mathbb{R}^{C \times rC}$, $W_2 \in \mathbb{R}^{rC \times C}$. r is the MLP (expansion) ratio, and $\sigma(\cdot)$ is a non-linear activation function such as GeLU or ReLU.

PoolFormer

- Use (average) pooling as the token mixer.
- Assuming input $T \in \mathbb{R}^{C \times H \times W}$ is channel-first, the pooling operator is $T'_{:,i,j} = \left(\frac{1}{K \times K} \sum_{p,q=1}^{K} T'_{:,i+p-\frac{K+1}{2},i+q-\frac{K+1}{2}} \right) - T_{:,i,j}$, where K is the pooling size.
- The subtraction of the input itself is added due to that MetaFormer block already has a residual connection.

Overall Framework of PoolFormer



(a) Overal framework with L PoolFormer blocks

(b) PoolFormer block

Experiments: Model Configurations

Stage	#Tokens	Layer Specification		PoolFormer S12	S24	S36	M36	M48
1	$\frac{H}{4} \times \frac{W}{4}$	Patch Embedding	Patch Size	7×7, stride 4				
			Embed. Dim.	64			96	
		PoolFormer Block	Pooling Size	3×3, stride 1				
			MLP Ratio	4				
			# Block	2	4	6	6	8
2	$\frac{H}{8} \times \frac{W}{8}$	Patch Embedding	Patch Size	3×3, stride 2				
			Embed. Dim.	128			192	
		PoolFormer Block	Pooling Size	3×3, stride 1				
			MLP Ratio	4				
			# Block	2	4	6	6	8
3	$\frac{H}{16} \times \frac{W}{16}$	Patch Embedding	Patch Size	3×3, stride 2				
			Embed. Dim.	320			384	
		PoolFormer Block	Pooling Size	3×3, stride 1				
			MLP Ratio	4				
			# Block	6	12	18	18	24
4	$\frac{H}{32} \times \frac{W}{32}$	Patch Embedding	Patch Size	3×3, stride 2				
			Embed. Dim.	512			768	
		PoolFormer Block	Pooling Size	3×3, stride 1				
			MLP Ratio	4				
			# Block	2	4	6	6	8
Parameters (M)				11.9	21.4	30.8	56.1	73.4
MACs (G)				1.9	3.5	5.1	9.0	11.8

Experiments: Image Classification

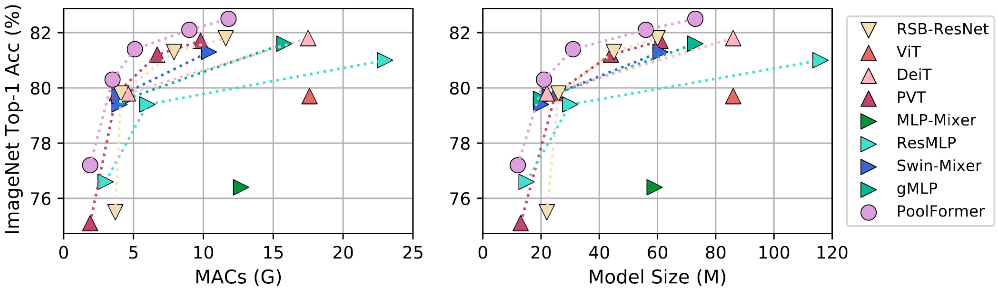
Setup

- Dataset: ImageNet-1K
- Data augmentation: MixUp, CutMix, CutOut and RandAugment
- Use Modified Layer Normalization (MLN) to compute the mean and variance along token and channel dimensions compared to only channel dimension in vanilla Layer Normalization.

Experiments: Image Classification

General Arch.	Token Mixer	Outcome Model	Image Size	Params (M)	MACs (G)	Top-1 (%)
Convolutional Neural Netowrks	—	RSB-ResNet-18 [57]	224	12	1.8	70.6
		RSB-ResNet-34 [57]	224	22	3.7	75.5
		RSB-ResNet-50 [57]	224	26	4.1	79.8
		RSB-ResNet-101 [57]	224	45	7.9	81.3
		RSB-ResNet-152 [57]	224	60	11.6	81.8
MetaFormer	Attention	ViT-B/16* [17]	224	86	17.6	79.7
		ViT-L/16* [17]	224	307	63.6	76.1
		DeiT-S [51]	224	22	4.6	79.8
		DeiT-B [51]	224	86	17.5	81.8
		PVT-Tiny [55]	224	13	1.9	75.1
		PVT-Small [55]	224	25	3.8	79.8
		PVT-Medium [55]	224	44	6.7	81.2
		PVT-Large [55]	224	61	9.8	81.7
	Spatial MLP	MLP-Mixer-B/16 [49]	224	59	12.7	76.4
		ResMLP-S12 [50]	224	15	3.0	76.6
		ResMLP-S24 [50]	224	30	6.0	79.4
		ResMLP-B24 [50]	224	116	23.0	81.0
		Swin-Mixer-T/D24 [35]	256	20	4.0	79.4
		Swin-Mixer-T/D6 [35]	256	23	4.0	79.7
		Swin-Mixer-B/D24 [35]	224	61	10.4	81.3
		gMLP-S [34]	224	20	4.5	79.6
		gMLP-B [34]	224	73	15.8	81.6
	Pooling	PoolFormer-S12	224	12	1.9	77.2
		PoolFormer-S24	224	21	3.5	80.3
		PoolFormer-S36	224	31	5.1	81.4
		PoolFormer-M36	224	56	9.0	82.1
		PoolFormer-M48	224	73	11.8	82.5

Experiments: Image Classification



Experiments: Object Detection and Instance Segmentation

Backbone	RetinaNet 1×							Mask R-CNN 1×						
	Params (M)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Params (M)	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet-18 [23]	21.3	31.8	49.6	33.6	16.3	34.3	43.2	31.2	34.0	54.0	36.7	31.2	51.0	32.7
PoolFormer-S12	21.7	36.2	56.2	38.2	20.8	39.1	48.0	31.6	37.3	59.0	40.1	34.6	55.8	36.9
ResNet-50 [23]	37.7	36.3	55.3	38.6	19.3	40.0	48.8	44.2	38.0	58.6	41.4	34.4	55.1	36.7
PoolFormer-S24	31.1	38.9	59.7	41.3	23.3	42.1	51.8	41.0	40.1	62.2	43.4	37.0	59.1	39.6
ResNet-101 [23]	56.7	38.5	57.8	41.2	21.4	42.6	51.1	63.2	40.4	61.1	44.2	36.4	57.7	38.8
PoolFormer-S36	40.6	39.5	60.5	41.8	22.5	42.9	52.4	50.5	41.0	63.1	44.8	37.7	60.1	40.0

Table 3. **Performance of object detection using RetinaNet, and object detection and instance segmentation using Mask R-CNN on COCO `val2017` [33].** 1× training schedule (*i.e.*12 epochs) is used for training detection models. AP^b and AP^m represent bounding box AP and mask AP, respectively.

Experiments: Object Detection and Instance Segmentation

Backbone	RetinaNet 1×							Mask R-CNN 1×						
	Params (M)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	Params (M)	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ResNet-18 [23]	21.3	31.8	49.6	33.6	16.3	34.3	43.2	31.2	34.0	54.0	36.7	31.2	51.0	32.7
PoolFormer-S12	21.7	36.2	56.2	38.2	20.8	39.1	48.0	31.6	37.3	59.0	40.1	34.6	55.8	36.9
ResNet-50 [23]	37.7	36.3	55.3	38.6	19.3	40.0	48.8	44.2	38.0	58.6	41.4	34.4	55.1	36.7
PoolFormer-S24	31.1	38.9	59.7	41.3	23.3	42.1	51.8	41.0	40.1	62.2	43.4	37.0	59.1	39.6
ResNet-101 [23]	56.7	38.5	57.8	41.2	21.4	42.6	51.1	63.2	40.4	61.1	44.2	36.4	57.7	38.8
PoolFormer-S36	40.6	39.5	60.5	41.8	22.5	42.9	52.4	50.5	41.0	63.1	44.8	37.7	60.1	40.0

Table 3. **Performance of object detection using RetinaNet, and object detection and instance segmentation using Mask R-CNN on COCO `val2017` [33].** 1× training schedule (*i.e.*12 epochs) is used for training detection models. AP^b and AP^m represent bounding box AP and mask AP, respectively.

Experiments: Semantic segmentation

Backbone	Semantic FPN	
	Params (M)	mIoU (%)
ResNet-18 [23]	15.5	32.9
PVT-Tiny [55]	17.0	35.7
PoolFormer-S12	15.7	37.2
ResNet-50 [23]	28.5	36.7
PVT-Small [55]	28.2	39.8
PoolFormer-S24	23.2	40.3
ResNet-101 [23]	47.5	38.8
ResNeXt-101-32x4d [60]	47.1	39.7
PVT-Medium [55]	48.0	41.6
PoolFormer-S36	34.6	42.0
PVT-Large [55]	65.1	42.1
PoolFormer-M36	59.8	42.4
ResNeXt-101-64x4d [60]	86.4	40.2
PoolFormer-M48	77.1	42.7

Table 4. **Performance of Semantic segmentation on ADE20K [65] validation set.** All models are equipped with Semantic FPN [29].

Experiments: Ablation Studies

Ablation	Variant	Params (M)	MACs (G)	Top-1 (%)
Baseline	None (PoolFormer-S12)	11.9	1.9	77.2
Token mixers	Pooling $\rightarrow$ Identity mapping	11.9	1.9	74.3
	Pooling $\rightarrow$ Global random matrix* (extra 21M frozen parameters)	11.9	3.3	75.8
	Pooling $\rightarrow$ Depthwise Convolution [9, 37]	11.9	1.9	78.1
	Pooling size 3 $\rightarrow$ 5	11.9	1.9	77.2
	Pooling size 3 $\rightarrow$ 7	11.9	1.9	77.1
	Pooling size 3 $\rightarrow$ 9	11.9	1.9	76.8
Normalization	Modified Layer Normalization† $\rightarrow$ Layer Normalization [1]	11.9	1.9	76.5
	Modified Layer Normalization† $\rightarrow$ Batch Normalization [27]	11.9	1.9	76.4
	Modified Layer Normalization† $\rightarrow$ None	11.9	1.9	46.1
Activation	GELU [24] $\rightarrow$ ReLU [40]	11.9	1.9	76.4
	GELU $\rightarrow$ SiLU [18]	11.9	1.9	77.2
Other components	Residual connection [24] $\rightarrow$ None	11.9	1.9	0.1
	Channel MLP $\rightarrow$ None	2.5	0.3	5.7
Hybrid Stages	[Pool, Pool, Pool, Pool] $\rightarrow$ [Pool, Pool, Pool, Attention]	14.0	2.0	78.3
	[Pool, Pool, Pool, Pool] $\rightarrow$ [Pool, Pool, Attention, Attention]	16.5	2.6	81.0
	[Pool, Pool, Pool, Pool] $\rightarrow$ [Pool, Pool, Pool, SpatialFC]	11.9	1.9	77.5
	[Pool, Pool, Pool, Pool] $\rightarrow$ [Pool, Pool, SpatialFC, SpatialFC]	12.2	1.9	77.9

Future Work

- MetaFormer Baselines for Vision
  - IdentityFormer, RandFormer
  - ConvFormer, CAFormer (Convs+Attentions)
- Work on NLP tasks?

References

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H. (2021, July). Training data-efficient image transformers distillation through attention. In International conference on machine learning (pp. 10347-10357). PMLR.

Tolstikhin, I. O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., ... Dosovitskiy, A. (2021). Mlp-mixer: An all-mlp architecture for vision. Advances in neural information processing systems, 34, 24261-24272.

Touvron, H., Bojanowski, P., Caron, M., Cord, M., El-Nouby, A., Grave, E., ... Jégou, H. (2022). Resmlp: Feedforward networks for image classification with data-efficient training. IEEE Transactions on Pattern Analysis and Machine Intelligence.

Yu, T., Li, X., Cai, Y., Sun, M., Li, P. (2022). S2-mlp: Spatial-shift mlp architecture for vision. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 297-306).

Yu, W., Si, C., Zhou, P., Luo, M., Zhou, Y., Feng, J., ... Wang, X. (2022). Metaformer baselines for vision. arXiv preprint arXiv:2210.13452.