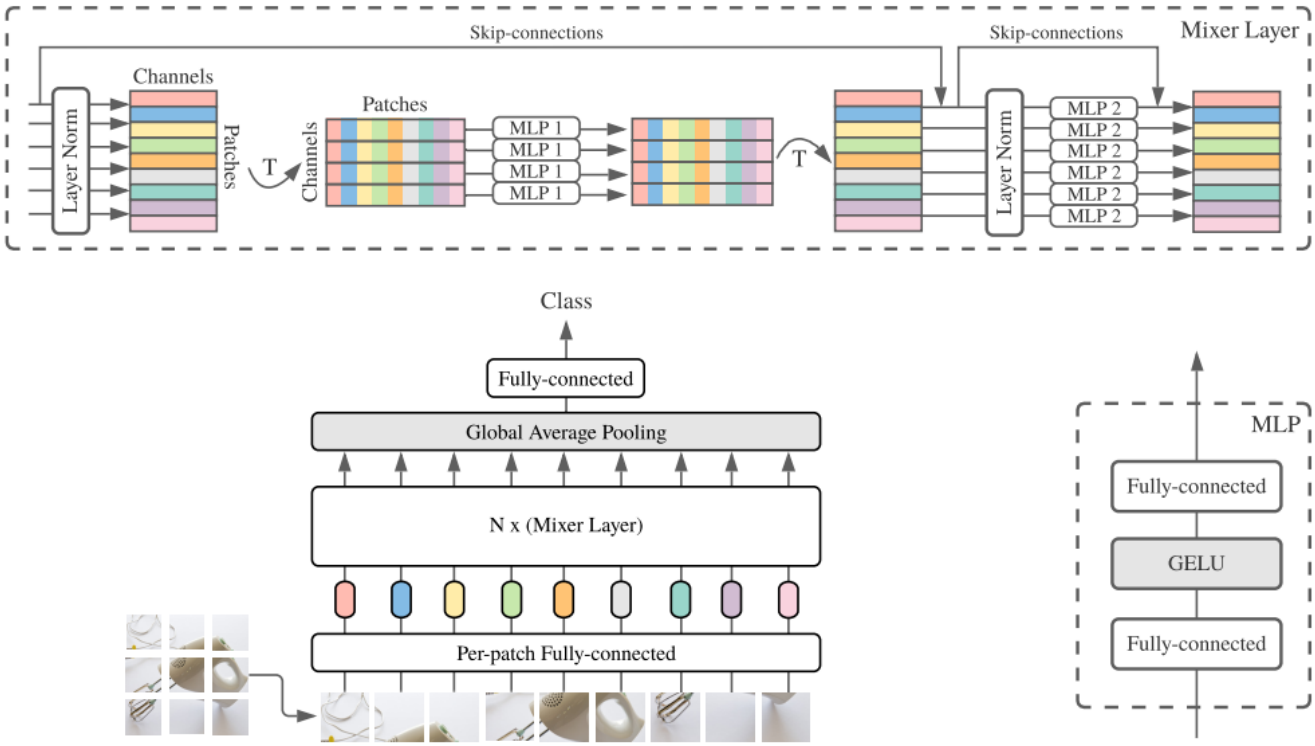
MLP Mixer: An all-MLP Architecture for Vision

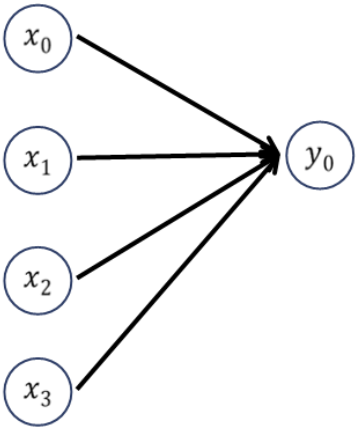
https://arxiv.org/pdf/2105.01601v1.pdf (https://arxiv.org/pdf/2105.01601v1.pdf)



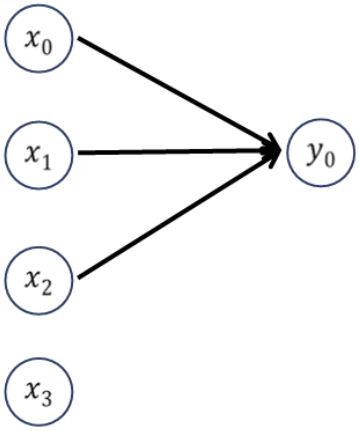
1. Abstract

- **Convolutional Neural Networks (CNNs)** are the go-to model for computer vision.
- Recently, attention-based networks, such as the **Vision Transformer**, have also become popular.
- In this paper we show that while convolutions and attention are both sufficient for good performance, **neither of them are necessary**.
- We present **MLP-Mixer**, an architecture based exclusively on **multi-layer perceptrons (MLPs)**.
- MLP-Mixer contains two types of layers: one with MLPs applied independently to image patches (i.e. “mixing” the per-location features), and one with MLPs applied across patches (i.e. “mixing” spatial information).
- When trained on large datasets, or with modern regularization schemes, MLP-Mixer attains **competitive scores** on image classification benchmarks, with pre-training and inference cost comparable to state-of-the-art models.
- We hope that these results spark further research beyond the realms of well established CNNs and Transformers.

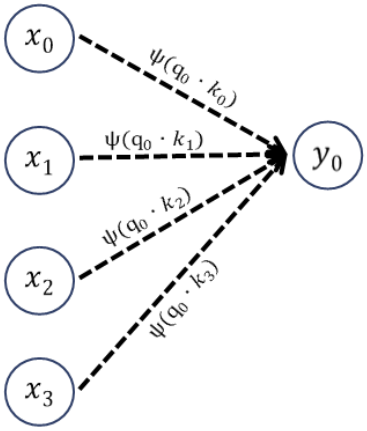
+)FullyConnected vs Convolution vs Attention



Fully-Connected layer
- Fully connected
- Same weights for all inputs



Convlution layer
- Fully connected
- Weight sharing
- Same weights for all inputs



Attention layer
- Fully connected
- Different weights for all inputs

2. Mixer Architecture

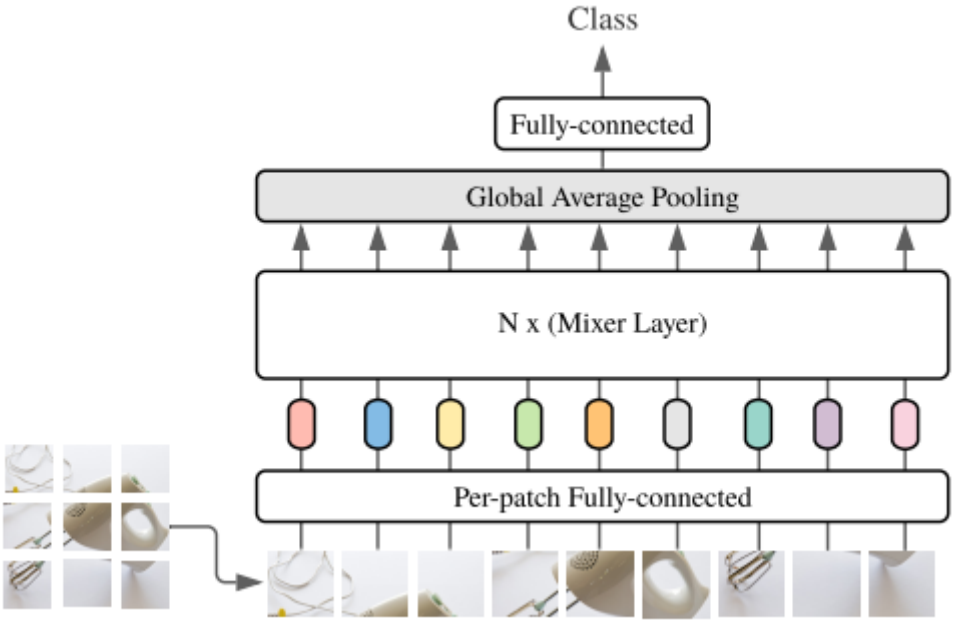
- Modern deep vision architectures consist of layers that **mix features**
  - **1) at a spatial location**
  - **2) between different spatial locations**

- In **CNNs**
  - 1x1 convolution perform 1).
  - pooling operation perform 2).
  - NxN convolution perform 1) and 2).

- In **Transformer**
  - self-attention layers perform 1) and 2).
  - MLP blocks perform 1).

- The idea of the Mixer architecture.
  - clearly separte the per location operation (**channel-mixing MLP block**) and cross-location operations (**token-mixing MLP block**).

1) Per-patch linear embddings.

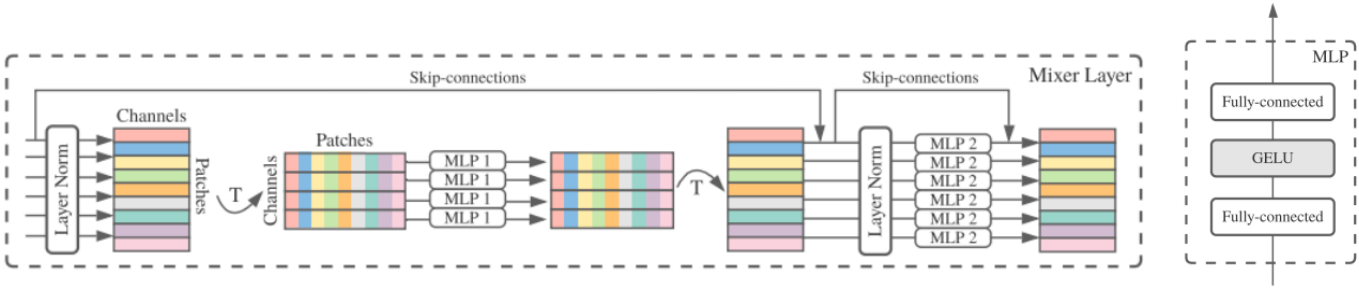


- Mixer takes as input a sequence of non-overlapping image S patches, each one projected to a desired hidden dimension C.
- input : $X \in \mathbb{R}^{S \times C}$
- If the original input image has resolution (H, W), and each patch has resolution (P, P), then the number of patches is $S = HW/P^2$
- All patches are linearly projected with the *same* projection matrix.

2) Two MLP blocks

Mixer consists of multiple layers of identical size, and each layer consists of two MLP blocks.

Each MLP block contains two fully-connected layers and a non-linearity applied independently to each row of its input data tensor.



1. Token-mixing MLP block
   - It acts on columns of X (i.e. it is applied to a transposed input table X^T), maps $\mathbb{R}^S \to \mathbb{R}^S$, and is shared across all columns.

2. Channel-mixing MLP block
   - it acts on rows of X, maps $\mathbb{R}^C \to \mathbb{R}^C$, and is shared across all rows.

3)

- The computational complexity of the network is linear in the number of input patches, unlike ViT whose complexity is quadratic.

- Mixer does not use positional embeddings because the token-mixing MLPs are sensitive to the order of the input tokens.
- Tying the parameters of the channel-mixing MLPs (within each layer) is a natural choice.
- However, tying parameters across channels is much less common.

Experiments

1. Specifications of the Mixer architectures

Table 1: Specifications of the Mixer architectures used in this paper. The "B", "L", and "H" (base, large, and huge) model scales follow Dosovitskiy et al. [14]. We use a brief notation: "B/16" means the model of base scale with patches of resolution 16×16. "S" refers to a small scale with 8 Mixer layers. The number of parameters is reported for an input resolution of 224 and does not include the weights of the classifier head.

Specification	S/32	S/16	B/32	B/16	L/32	L/16	H/14
Number of layers	8	8	12	12	24	24	32
Patch resolution $P\times P$	32×32	16×16	32×32	16×16	32×32	16×16	14×14
Hidden size C	512	512	768	768	1024	1024	1280
Sequence length S	49	196	49	196	49	196	256
MLP dimension D_C	2048	2048	3072	3072	4096	4096	5120
MLP dimension D_S	256	256	384	384	512	512	640
Parameters (M)	10	10	46	46	188	189	409

2. Transfer performance, inference throughput, and training cost.

Table 2: Transfer performance, inference throughput, and training cost. The rows are sorted by inference throughput (fifth column). Mixer has comparable transfer accuracy to state-of-the-art models with similar cost. The Mixer models are fine-tuned at resolution 448. Mixer performance numbers are averaged over three fine-tuning runs and standard deviations are smaller than 0.1.

	ImNet top-1	ReaL top-1	Avg 5 top-1	VTAB-1k 19 tasks	Throughput img/sec/core	TPUv3 core-days
Pre-trained on ImageNet-21k (public)						
● HaloNet [49]	85.8	—	—	—	120	0.10k
● Mixer-L/16	84.15	87.86	93.91	74.95	105	0.41k
● ViT-L/16 [14]	85.30	88.62	94.39	72.72	32	0.18k
● BiT-R152x4 [22]	85.39	—	94.04	70.64	26	0.94k
Pre-trained on JFT-300M (proprietary)						
● NFNet-F4+ [7]	89.2	—	—	—	46	1.86k
● Mixer-H/14	87.94	90.18	95.71	75.33	40	1.01k
● BiT-R152x4 [22]	87.54	90.54	95.33	76.29	26	9.90k
● ViT-H/14 [14]	88.55	90.72	95.97	77.63	15	2.30k
Pre-trained on unlabelled or weakly labelled data (proprietary)						
● MPL [33]	90.0	91.12	—	—	—	20.48k
● ALIGN [21]	88.64	—	—	79.99	15	14.82k

3. Graphs

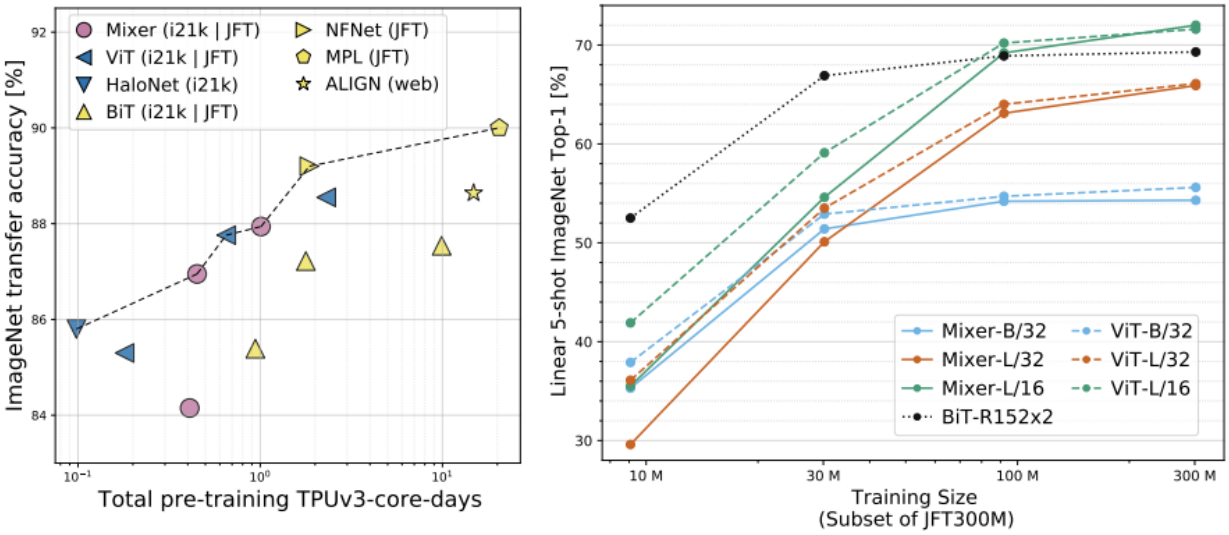


Figure 2: **Left:** ImageNet accuracy/training cost Pareto frontier (dashed line) for the SOTA models presented in Table 2. These model are pre-trained on ImageNet-21k, or JFT (labelled, or pseudo-labelled for MPL), or noisy web image text pairs. In addition, we include ViT-L/16, Mixer-L/16, and BiT-R200x3 (Adam) for context. Mixer is as good as these extremely performant ResNets, ViTs, and hybrid models, and sits on frontier with HaloNet, ViT, NFNet, and MPL. **Right:** Mixer (solid) catches or exceeds BiT (dotted) and ViT (dashed) as the data size grows. Every point on a curve uses the same pre-training compute; they correspond to pre-training on 3%, 10%, 30%, and 100% of JFT-300M for 233, 70, 23, and 7 epochs, respectively. Mixer improves more rapidly with data than ResNets, or even ViT, and the gap between large scale Mixer and ViT models shrinks until the performance is matched on the entire dataset.

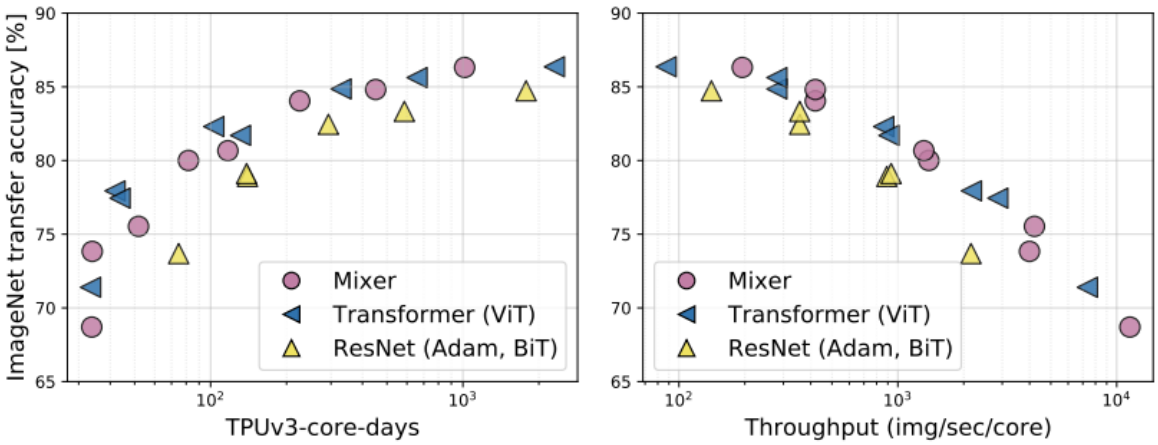


Figure 3: The role of the model scale. ImageNet validation top-1 accuracy vs. total pre-training compute (**left**) and throughput (**right**) of ViT, BiT, and Mixer models at various scales. All models are pre-trained on JFT-300M and fine-tuned at resolution 224, which is lower than in Figure 2 (left).

4. weights

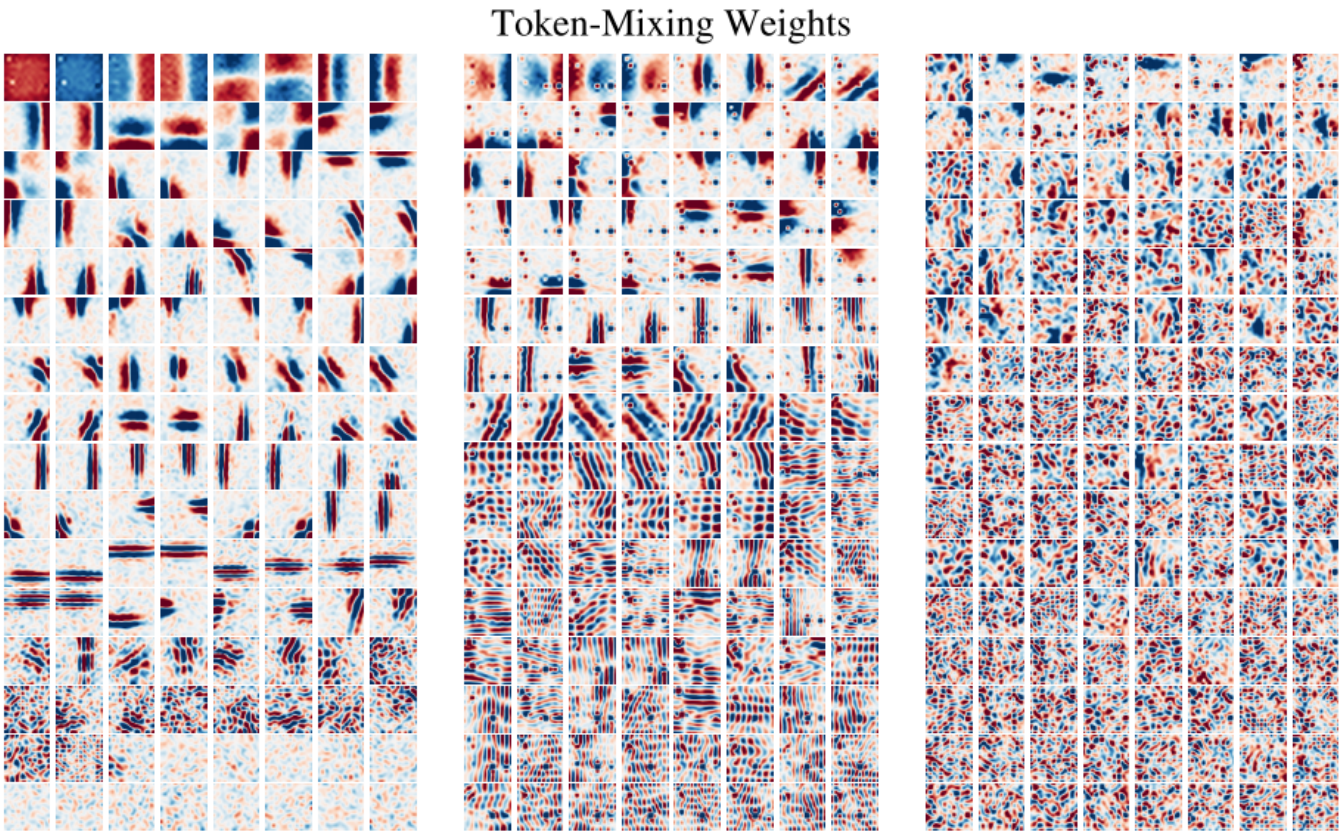


Figure 4: A selection of input weights to the hidden units in the first (**left**), second (**center**), and third (**right**) token-mixing MLPs of a Mixer-B/16 model trained on JFT-300M. Each unit has $14 \times 14 = 196$ weights, one for each of the 14×14 incoming patches. We pair units whose inverse is closest, to easily visualize the emergence of kernels of opposing phase. Pairs are sorted approximately by filter frequency. We highlight that in contrast to the kernels of convolutional filters, where each weight corresponds to one pixel in the input image, one weight in any plot from the left column corresponds to a particular 16×16 patch of the input image. Complete plots in Supplementary D.