ConvMLP: Hierarchical Convolutional MLPs for Vision

Jiachen Li[1], Ali Hassani[1], Steven Walton[1], Humphrey Shi[1,2]
[1]SHI Lab @ University of Oregon & UIUC, [2]Picsart AI Research (PAIR)

Abstract

MLP-based architectures, which consist of a sequence of consecutive multi-layer perceptron blocks, have recently been found to reach comparable results to convolutional and transformer-based methods on image classification. However, most methods adopt spatial MLPs which take fixed-dimension inputs, therefore making it difficult to apply them as backbones to downstream tasks such as object detection and semantic segmentation, which require inputs with arbitrary dimension. Moreover, single-stage designs further limit the performance in other computer vision tasks and fully-connected layers bear heavy computation. To tackle these problems, we propose ConvMLP: a Hierarchical Convolutional MLP for visual recognition, which is a lightweight, stage-wise, co-design of convolution layers, and MLPs. In particular, ConvMLP-S achieves 76.8% top-1 accuracy on ImageNet-1k with 9M parameters and 2.4 GMACs (15% and 19% of MLP-Mixer-B/16, respectively). Experiments on object detection and semantic segmentation further show that visual representation learned by ConvMLP can be seamlessly transferred to downstream tasks and achieve competitive results with fewer parameters. Our code and pre-trained models are open-sourced at https://github.com/SHI-Labs/Convolutional-MLPs.

1. Introduction

Image classification is a fundamental problem in computer vision, and most milestone solutions in the past five years have been dominated by deep convolutional neural networks. Since late 2020, with the rise of Vision Transformer [6], researchers have not only been applying Transformers [38] to image classification and other computer vision tasks, but explored more meta-models other than convolutional neural networks for visual recognition. MLP-Mixer [35] proposes token-mixing and channel-mixing MLPs to allow communication between spatial locations and channels. ResMLP [36] uses cross-patch and cross-channel sublayers as the building block, following the design of ViT. gMLP [25] connects channel MLPs by adding spatial gating units. In essence, MLP-based models show that simple feed-forward neural networks can compete with operators like convolution and self-attention on image classification.

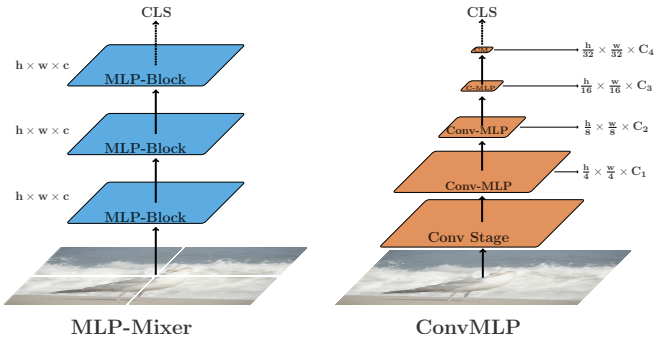

Figure 1. Comparing MLP-Mixer to ConvMLP. MLP-Mixer is designed for image classification with fixed representation of visual feature maps. ConvMLP adopts a simple hierarchical multi-stage co-design of convolutions and MLPs, which achieves both more flexible mutli-scale representations as well as better accuracy vs computation trade-offs for visual recognition tasks including classification, detection and segmentation.

However, using MLPs to encode spatial information requires fixing dimension of inputs, which makes it difficult to be deployed on downstream computer vision tasks – such as object detection and semantic segmentation – since they usually require arbitrary resolutions of input sizes. Furthermore, single-stage design, following ViT [6], may constrain performances on object detection and semantic segmentation since they make predictions based on feature pyramids. Representation learned by single resolution hurt the performance on small object recognition as shown in DETR [1]. Large consecutive MLPs also bring heavy computation burden and more parameters with high dimension of hidden layers. For instance, MLP-Mixer is only able to slightly surpass ViT-Base with its large variant, which is over twice as large and twice as expensive in terms of computation. Similarly, ResMLP suffers from over 30% more parameters and complexity, compared to a transformer-based model of similar performance.

Based on these observations, we propose ConvMLP: A Hierarchical Convolutional MLP backbone for visual

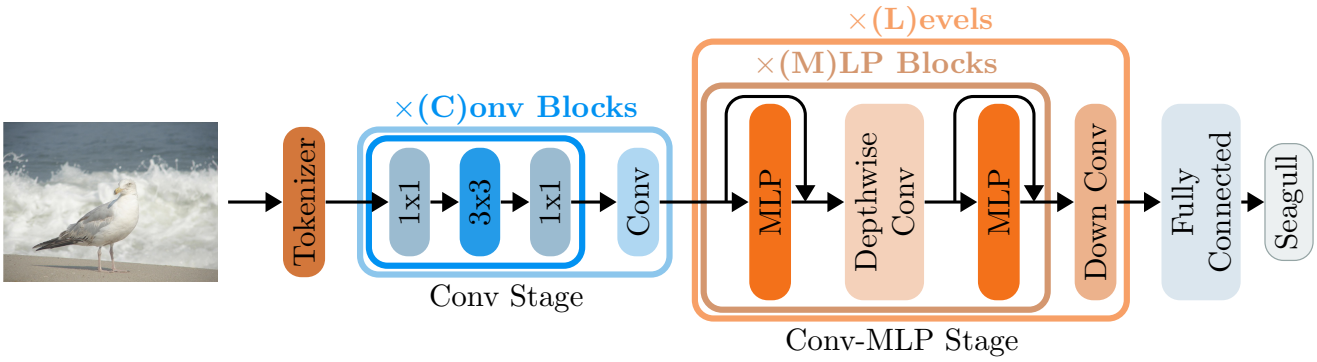

Figure 2. Overview of ConvMLP framework. The Conv Stage consists of C convolutional blocks with 1×1 and 3×3 kernel sizes. The Conv-MLP Stage consists of Channel MLPs with skip layers and a 3×3 depthwise convolution layer. This is repeated M times before a down-sampling convolution is utilized to express a level L. A level down samples an image $\mathcal{L} : h \times w \times c \mapsto \frac{h}{2^L} \times \frac{w}{2^L} \times 2^L c$

recognition, which is a combination of convolution layers and MLP layers for image classification and can be seamlessly used for downstream tasks like object detection and segmentation as shown in Figure 1. To remove constraints on input dimension in other MLP-like frameworks, we first replace all spatial MLPs with channel MLPs for augmenting cross-channel connections and build a pure-MLP baseline model. To make up spatial information interaction, we add a light-weight convolution stage on top of the rest MLP stages and use convolution layers for down-sampling. Furthermore, to make up spatial connections in MLP stages, we add a simple 3×3 depth-wise convolution between the two channel MLPs in each MLP block, hence calling it a Conv-MLP block. This co-design of convolution layers and MLP layers builds the prototype of ConvMLP model for image classification. To make ConvMLP scalable, we extend ConvMLP model by scaling the depth and width of both convolution and Conv-MLP stages. It achieves competitive performances on ImageNet-1k with fewer parameters compared to recent MLP-based models. We also fine-tune the model on CIFAR and Flowers-102 for transfer learning. On object detection and semantic segmentation, we conduct experiments on MS COCO and ADE20K benchmarks. It shows that using ConvMLP as a backbone achieves better trade-off between performance and model size compared to other MLP-based methods.

In conclusion, our contributions are as follows:

- We analyze the constraints of current MLP-based models for image classification, which only take inputs of fixed dimensions and are difficult to be used in downstream computer vision tasks as backbones. Single-stage design and large computation burden further limit their applications.

- We propose ConvMLP: a Hierachical Convolutional MLP backbone for visual recognition with co-design of convolution and MLP layers. It is scalable and can be seamlessly deployed on downstream tasks like object detection and semantic segmentation.

- We conduct extensive experiments on ImageNet-1k for image classification, CIFAR and Flowers-102 for transfer learning, MS COCO for object detection and ADE20K for semantic segmentation to evaluate the effectiveness of our ConvMLP model.

2. Related Work

2.1. Convolutional Methods

Image classification has been dominated by convolutional neural networks for almost a decade, since the rise of AlexNet [21], which introduced a convolutional neural network for image classification, and won the 2012 ILSRVC. Following that, VGGNet [33] proposed larger and deeper network for better performance. ResNet [12] introduced skip connections to allow training even deeper networks, and DenseNet [17] proposed densely connected convolution layers. In the meantime, researchers explored smaller and lightweight models that would be deployable to mobile devices. MobileNet [16, 32] consisted of depth-wise and point-wise convolutions, which reduced the number of parameters and computations required. ShuffleNet [29] found channel shuffling to be effective, and EfficientNet [34] further employs model scaling to width, depth, and resolution for better model scalability.

2.2. Transformer-based Methods

Transformer [38] was proposed for machine translation and has been widely adopted in most natural language processing. Recently, researchers in the computer vision area have adopted transformers to image classification.

Stage	ConvMLP-S	ConvMLP-M	ConvMLP-L
Conv	$\begin{bmatrix} 1\times1 \text{ Conv} \\ 3\times3 \text{ Conv} \\ 1\times1 \text{ Conv} \end{bmatrix} \times 2$	$\begin{bmatrix} 1\times1 \text{ Conv} \\ 3\times3 \text{ Conv} \\ 1\times1 \text{ Conv} \end{bmatrix} \times 3$	$\begin{bmatrix} 1\times1 \text{ Conv} \\ 3\times3 \text{ Conv} \\ 1\times1 \text{ Conv} \end{bmatrix} \times 3$
Scale	$C_1 = 64$	$C_1 = 64$	$C_1 = 96$
Conv-MLP	$\begin{bmatrix} \text{Channel MLP} \\ 3\times3 \text{ DW Conv} \\ \text{Channel MLP} \end{bmatrix} \times 2$	$\begin{bmatrix} \text{Channel MLP} \\ 3\times3 \text{ DW Conv} \\ \text{Channel MLP} \end{bmatrix} \times 3$	$\begin{bmatrix} \text{Channel MLP} \\ 3\times3 \text{ DW Conv} \\ \text{Channel MLP} \end{bmatrix} \times 4$
Scale	$C_2 = 128$, $R = 2$	$C_2 = 128$, $R = 3$	$C_2 = 192$, $R = 3$
Conv-MLP	$\begin{bmatrix} \text{Channel MLP} \\ 3\times3 \text{ DW Conv} \\ \text{Channel MLP} \end{bmatrix} \times 4$	$\begin{bmatrix} \text{Channel MLP} \\ 3\times3 \text{ DW Conv} \\ \text{Channel MLP} \end{bmatrix} \times 6$	$\begin{bmatrix} \text{Channel MLP} \\ 3\times3 \text{ DW Conv} \\ \text{Channel MLP} \end{bmatrix} \times 8$
Scale	$C_3 = 256$, $R = 2$	$C_3 = 256$, $R = 3$	$C_3 = 384$, $R = 3$
Conv-MLP	$\begin{bmatrix} \text{Channel MLP} \\ 3\times3 \text{ DW Conv} \\ \text{Channel MLP} \end{bmatrix} \times 2$	$\begin{bmatrix} \text{Channel MLP} \\ 3\times3 \text{ DW Conv} \\ \text{Channel MLP} \end{bmatrix} \times 3$	$\begin{bmatrix} \text{Channel MLP} \\ 3\times3 \text{ DW Conv} \\ \text{Channel MLP} \end{bmatrix} \times 3$
Scale	$C_4 = 512$, $R = 2$	$C_4 = 512$, $R = 3$	$C_4 = 768$, $R = 3$

Table 1. Detailed model architecture of ConvMLP in different scales. R denotes scaling ratio of hidden layers in MLP.

They propose ViT [6] that reshapes image to patches for feature extraction by transformer encoder, which achieves comparable results to CNN-based models. DeiT [37] further employs more data augmentation and makes ViT comparable to CNN-based models without ImageNet-22k or JFT-300M pretraining. DeiT also proposes an attention-based distillation method, which is used for student-teacher training, leading to even better performance. CCT [10] proposes a convolutional tokenizer and compact vision transformers, leading to better performance on smaller datasets training from scratch, with fewer parameters compared with ViT. TransCNN [26] also proposes a co-design of convolutions and multi-headed attention to learn hierarchical representations. To make models friendly to downstream tasks, PVT [39] proposes feature pyramids for vision transformers. Swin Transformer [27] uses patch-level multi-headed attention and stage-wise design, which also increase transferability to downstream tasks. Shuffle Swin Transformer [18] proposes shuffle multi-headed attention to augment spatial connection between windows. NAT [9] and DiNAT [8] adopt dense and sparse sliding window attention patterns to achieve a linear cost attention.

2.3. MLP-based Methods

MLP-Mixer [35] was recently proposed as a large scale image classifiers that was neither convolutional nor transformer-based. At its core, it consisted of basic matrix multiplications, data layout changes and scalar non-linearities. ResMLP [36] followed a ResNet-like structure with MLP-based blocks instead of convolutional ones. Following that, gMLP [25] proposed a Spatial Gating Unit to process spatial features. S^2-MLP [41] adopts shifted spatial feature maps to augment information communication. ViP [14] employs linear projection on the height, width and channel dimension separately. All these methods have MLPs on fixed spatial dimensions which make it hard to be used in downstream tasks since the dimensions of spatial MLPs are fixed. Cycle MLP [3] and AS-MLP [22] are concurrent works. The former replaces the spatial MLPs with cycle MLP layers and the latter with axial shifted MLPs, which make the model more flexible for varying inputs

sizes. They reach competitive results on both image classification and other downstream tasks. Hire-MLP [7] is another concurrent work that uses Hire-MLP blocks to learn hierarchical representations and achieves comparable result to transformer-based model on ImageNet.

3. ConvMLP

In this section, we first introduce the overall design and framework of ConvMLP. Then, we follow that design pattern including convolutional tokenizer, convolution stage and Conv-MLP Stage. We also explain how model scaling is applied to ConvMLP on convolution and Conv-MLP stages.

3.1. Overall Design

The overall framework of ConvMLP is illustrated in Figure 2. Unlike other MLP-based models, we use a convolutional tokenizer to extract the initial feature map $F_1 \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times C_1}$. To reduce computation and improve spatial connections, we follow tokenization with a pure convolutional stage, producing feature map $F_2 \in \mathbb{R}^{\frac{H}{8} \times \frac{W}{8} \times C_2}$. Then we place 3 Conv-MLP stages, generating 2 feature maps $F_3 \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C_3}$ and $F_4 \in \mathbb{R}^{\frac{H}{32} \times \frac{W}{32} \times C_4}$. Each Conv-MLP stage includes multiple Conv-MLP blocks and each Conv-MLP block has one channel MLP followed by a depth-wise convolutional layer, succeeded by another channel MLP. Similar to previous works, we include residual connections and Layer Normalization applied to inputs in the block. Each channel MLP consists of two fully connected layers with a GeLU activation [13] and dropout. We then apply global average pooling across to the output feature map, F_4, and send it through the classification head. When applying ConvMLP to downstream tasks, the feature maps F_1, F_2, F_3, and F_4 can be used to generate feature pyramids with no constraints on input size.

3.2. Convolutional Tokenizer

As stated, we replace the original patch tokenizer with a convolutional tokenizer. It includes three convolutional blocks, each consisting of a 3×3 convolution, batch normalization and ReLU activation. The tokenizer is also appended with a max pooling layer. Our experiments show that a convolutional tokenizer brings faster convergence and better performance in the end.

3.3. Convolution Stage

In order to augment spatial connections, we adopt a fully-convolutional first stage. It consists of multiple blocks, where each block is comprised of two 1×1 convolution layers with a 3×3 convolution in between. It brings more stable training and improvements on accuracy with few extra parameters.

3.4. Conv-MLP Stage

To reduce constraints on input dimension, we replace all spatial MLPs with channel MLPs. Since channel MLP only share weights across channels which lacks spatial interactions, we make up it by adding convolution layers in early stage, down-sampling and MLP blocks.

Convolutional Downsampling In the baseline model, we follow Swin Transformer [27] that uses a patch merging method based on linear layers to down-sample feature maps. To augment adjacent spatial intersection, we replace patch merging with a 3×3 convolution layer under stride 2. It improves the classification accuracy while only brings a few more parameters.

Convolution in MLP block We further add a depth-wise convolution layer between two channel MLPs in one MLP block and name it Conv-MLP block. It is a 3×3 convolution layer with the same channel to the two channel MLPs, which is also used in recent Shuffle Swin Transformer [18] to augment neighbor window connections. It makes up the deficiency of removing spatial MLPs, which improves the performance by a large margin while only brings few parameters.

3.5. Model Scaling

To make ConvMLP scalable, we scale up ConvMLP on both width and depth of convolution stages and Conv-MLP stages. We present 3 ConvMLP variants. Our smallest ConvMLP-S starts with only a two convolutional blocks, and has 2, 4 and 2 Conv-MLP blocks in the three Conv-MLP stages respectively. ConvMLP-M and ConvMLP-L start with three convolutional blocks. ConvMLP-M has 3, 6 and 3, and ConvMLP-L has 4, 8 and 3 Conv-MLP blocks in the three Conv-MLP stages. Details are also presented in Table 1. Experiments show that the performance of image classification and downstream tasks improves consistently with model scaling.

4. Experiments

In this section, we mainly introduce our experiments on ImageNet-1k, CIFAR-10/100, Flowers-102, MS COCO and ADE20K. We first show ablation studies on different modules in the ConvMLP framework to evaluate their effectiveness. Then, we compare ConvMLP to other state-of-the-art models on ImageNet-1k with three variants: ConvMLP-S, ConvMLP-M and ConvMLP-L. We also show transferring ability to CIFAR-10/100 and Flowers-102. On MS COCO and ADE20K benchmark, we use ConvMLP as backbones of RetinaNet, Mask R-CNN, Semantic FPN and it shows consistent improvements on these different downstream models.

Conv Stage	Conv Downsampling	Depth-Wise Conv	Epochs	# Params (M)	GMACs	Top-1 Acc (%)
-	-	-	100	7.88	1.47	63.29
✓	-	-	100	7.89	1.59	66.69
✓	✓	-	100	8.71	1.65	69.56
✓	-	✓	100	7.91	1.59	73.84
✓	✓	✓	100	8.73	1.65	74.04
✓	✓	✓	300	8.73	1.65	76.33
✓†	✓	✓	300	9.02	2.40	76.81

Table 2. Ablation study on ImageNet-1k validation set. All experiments are based on ConvMLP-S. † denotes replacing 1×1 with 3×3 convolution layers in Conv Stage with improved accuracy in the long run, which is used in our final ConvMLP-S model.

4.1. ImageNet-1k

ImageNet-1k [21] contains 1.2M training images and 50k images on 1000 categories for evaluating performances of classifiers. We follow standard practice provided by `timm` [40] toolbox. We use RandAugment [5] Mixup [44], and CutMix [43] for data augmentation. AdamW [28] is adopted as optimizer with momentum of 0.9 and weight decay of 0.05. The initial learning rate is 5e-4 with batch size of 128 on each GPU card. We use 8 NVIDIA RTX A6000 GPUs to train all models for 300 epochs and the total batch size is 1024. All other training settings and hyper-parameters are adopted from DeiT [37] for fair comparisons. For those results in ablation study, we train these models for 100 epochs with batch size 256 on each GPU and use 4 GPUs with learning rate at 1e-3.

4.2. Ablation Study

Our baseline model Pure-MLP Baseline is composed of one patch converter adopted from Swin [27] and a sequence of channel MLPs in following stages. In Table 2, the baseline model reaches 63.29% top-1 accuracy on ImageNet-1k and we replace the first stage of MLPs with a convolution stage that has two 1×1 convolution layers with a 3×3 convolution layer in between. Then, we replace the downsampler from patch merging used in Swin into a single 3×3 convolution layer with stride 2, which further improves top-1 accuracy to 69.56%. To further make up spatial information communication, we add a 3×3 depth-wise convolution between the two channel MLPs and extend training epochs to 300. Finally, we modify the convolution stage with successive 1×1, 3×3, 1×1 convolution blocks and builds ConvMLP-S model.

4.3. Comparisons with SOTA

In Table 3, we compare ConvMLP to other state-of-the-art image classification models on ImageNet-1k. We use three variants ConvMLP-S, ConvMLP-M, ConvMLP-L and the detailed architecture are shown in Table 1. We include Convolution-based, Transformer-based and MLP-based methods under different scales: Small models (5M-15M), Medium-sized models (16-30M) and Large models ($>$ 30M). We also present number of parameters, GMACs Acc/GMACs, ACC/MParams of these models to show the efficiency on model size and computation. It turns out that ConvMLP-S reaches better accuracy vs computation trade-offs compared with other MLP-based methods.

4.4. Transfer learning

Dataset We use CIFAR-10/CIFAR-100 [20] and Flowers-102 [30] to evaluate transferring ability of ImageNet-pretrained ConvMLP variants. Each model was fine-tuned for 50 epochs with a learning rate of 3e-4 (with cosine scheduler), weight decay of 5e-2, 10 warmup and cooldown epochs. We used the same training script and therefore augmentations as the ImageNet-1k experiments. We also resized all images to 224×224.

Results The results are presented in Table 4. We report results from ResMLP, ViT and DeiT as well. ConvMLP reaches the top performance with less computations.

4.5. Object Detection

Dataset MS COCO [24] is a widely-used benchmark for evaluating object detection model. It has 118k images for training and 5k images for evaluating performances of object detectors. We follow standard practice of RetinaNet [23] and Mask R-CNN [11] with ResNet as backbones in `mmdetection` [2]. We replace ResNet backbones with ConvMLP and adjust the dimension of convolution layers in feature pyramids accordingly. We also replace SGD optimizer with AdamW and adjust learning rate to 1e-4 with weight decay at 1e-4, which follows the configs in PVT [39]. We train both RetinaNet and Mask R-CNN for 12 epochs on 8 GPUs with total batch size of 16.

Results We transfer ResNet, Pure-MLP and ConvMLP variants to object detection on MS COCO and the results

Model	Backbone	# Params (M)	GMACs	Top-1 (%)	Acc/GMACs	Acc/MParams
Small models (5M-15M)						
ResNet18 [12]	Convolution	11.7	1.8	69.8	38.8	6.0
Mobilenetv3 [15]	Convolution	5.4	0.2	75.2	376.0	13.9
EfficientNet-B0 [34]	Convolution	5.3	0.4	77.1	192.8	14.5
ResMLP-S12 [36]	MLP	15.3	3.0	76.6	25.5	5.0
CycleMLP-B1 [3]	MLP	15.2	2.1	78.9	37.6	5.2
ConvMLP-S (**ours**)	ConvMLP	9.0	2.4	76.8	32.0	8.5
Medium-sized models (16M-30M)						
ResNet50 [12]	Convolution	25.6	4.1	76.1	18.6	3.0
EfficientNet-B4 [34] ↑380	Convolution	19.0	4.2	82.9	19.7	4.4
ViT-S [6] †	Transformer	22.1	4.6	79.9	17.4	3.6
DeiT-S [37]	Transformer	22.1	4.6	81.2	17.7	3.7
PVT-S [39]	Transformer	24.5	3.8	79.8	21.0	3.3
CCT-14t [10]	Transformer	22.4	5.1	80.7	15.8	3.6
MLP-Mixer-S/16 [35]	MLP	18.5	3.8	73.8	19.4	4.0
ResMLP-S24 [36]	MLP	30.0	6.0	79.4	13.2	2.6
gMLP-S [25]	MLP	19.4	4.5	79.6	17.7	4.1
AS-MLP-Ti [22]	MLP	28.0	4.4	81.3	18.7	2.9
ViP-Small/7 [14]	MLP	25.1	6.9	81.5	11.8	3.2
ConvMLP-M (**ours**)	ConvMLP	17.4	3.9	79.0	20.3	4.5
Large models (>30M)						
ResNet101 [12]	Convolution	44.6	7.8	78.0	10.0	1.7
RegNetY-8GF [31]	Convolution	39.2	8.0	79.0	9.9	2.0
RegNetY-16GF [31]	Convolution	83.6	15.9	80.4	5.1	1.0
ViT-B [6] †	Transformer	86.6	17.5	81.8	4.7	0.9
DeiT-B [37]	Transformer	86.6	17.5	83.4	4.8	1.0
PVT-L [39]	Transformer	61.4	9.8	81.7	8.3	1.3
Swin Transformer-B [27]	Transformer	87.8	15.4	83.5	5.4	1.0
Shuffle Swin-B [18]	Transformer	87.8	15.6	84.0	5.4	1.0
MLP-Mixer-B/16 [35]	MLP	59.9	12.6	76.4	6.1	1.3
S^2-MLP-wide [41]	MLP	71.0	14.0	80.0	5.7	1.1
ResMLP-B24 [36]	MLP	115.7	23.0	81.0	3.5	0.7
gMLP-B [25]	MLP	73.1	15.8	81.6	5.2	1.1
ViP-Large/7 [14]	MLP	87.8	24.4	83.2	3.4	0.9
CycleMLP-B5 [3]	MLP	75.7	12.3	83.2	6.7	0.9
AS-MLP-B [22]	MLP	88.0	15.2	83.3	5.4	1.0
ConvMLP-L (**ours**)	ConvMLP	42.7	9.9	80.2	8.1	1.9

Table 3. ImageNet-1k validation top-1 accuracy comparison between ConvMLP and state-of-the-art models. Compared to other MLP-based methods, ConvMLP achieved the best Acc/GMACs and Acc/MParams in different model size ranges. †: *reported from DeiT for fairer comparison; ViT-S was not proposed in the original paper.* ↑ *specifies image resolution, if different from 224×224.*

are presented in Figure 3. It can be observed that ConvMLP achieves better performance on object detection and instance segmentation consistently as backbones of RetinaNet and Mask R-CNN compared with Pure-MLP and

Model	# Params (M)	ImageNet-1k (%)	CIFAR-10 (%)	CIFAR-100 (%)	Flowers-102 (%)
ConvMLP-S	9.0	76.8	98.0	87.4	99.5
ResMLP-S12 [36]	15.4	76.6	98.1	87.0	97.4
ConvMLP-M	17.4	79.0	98.6	89.1	99.5
ResMLP-S24 [36]	30.0	79.4	98.7	89.5	97.4
ConvMLP-L	42.7	80.2	98.5	89.2	99.6
ViT-B [6]	86.6	81.8	99.1	90.8	98.4
DeiT-B [37]	86.6	83.4	99.1	91.3	98.9

Table 4. Fine-tuning top-1 accuracy on CIFAR-10/100 and Flowers-102 with pre-training on ImageNet-1k. ConvMLP is the top performing model on Flowers-102 compared with ResMLP, ViT and DeiT.

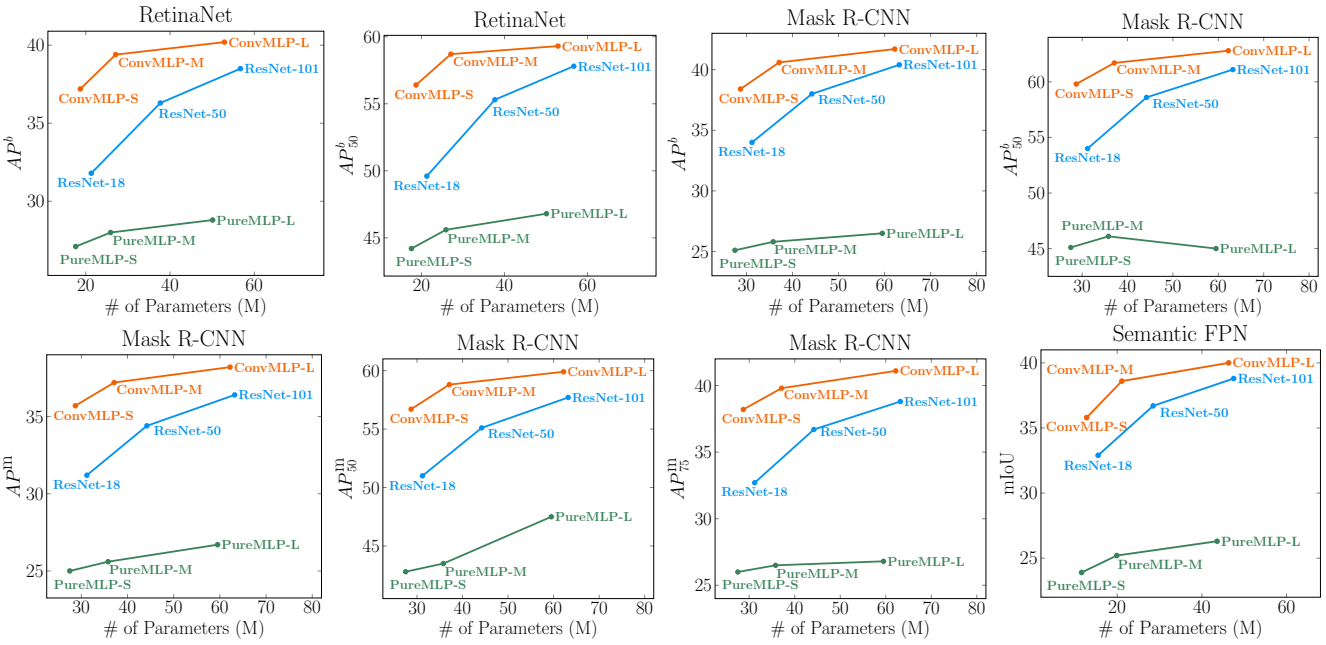

Figure 3. Comparisons between ConvMLP, Pure-MLP and ResNet as backbones of RetinaNet, Mask R-CNN on MS COCO and Semantic FPN on ADE20K. ConvMLP-based models show consistent improvements under different evaluation metrics and tasks.

ResNet. More details of the results are presented in Appendix.

4.6. Semantic Segmentation

Dataset ADE20K [45] is a widely-used dataset for semantic segmentation, which has 20k images for training and 2k images for evaluating the performance of semantic segmentation models. We employ standard practice of Semantic FPN [19] implemented based on `mmsegmentation` [4]. Following PVT in semantic segmentation, we train ConvMLP-based Semantic FPN on 8 GPUs with total batch size of 16 for 40k iterations. We also replace optimizer from SGD to AdamW with learning rate at 2e-4 and weight decay at 1e-4. The learning rate decays with polynomial rate at 0.9 and input images are randomly resized and cropped to 512×512.

Results All experimental results on ADE20K are presented in Figure 3. Similar to the object detection results presented in 4.5, it can be observed that visual representations learned by ConvMLP can also be successfully transferred to pixel-level prediction tasks, such as semantic segmentation. We present further details of these experiments in Appendix.

4.7. Visualization

We visualize feature maps of ResNet50, MLP-Mixer-B/16, Pure-MLP Baseline and ConvMLP-M under $(1024, 1024)$ input size (MLP-Mixer-B/16 under $(224, 224)$

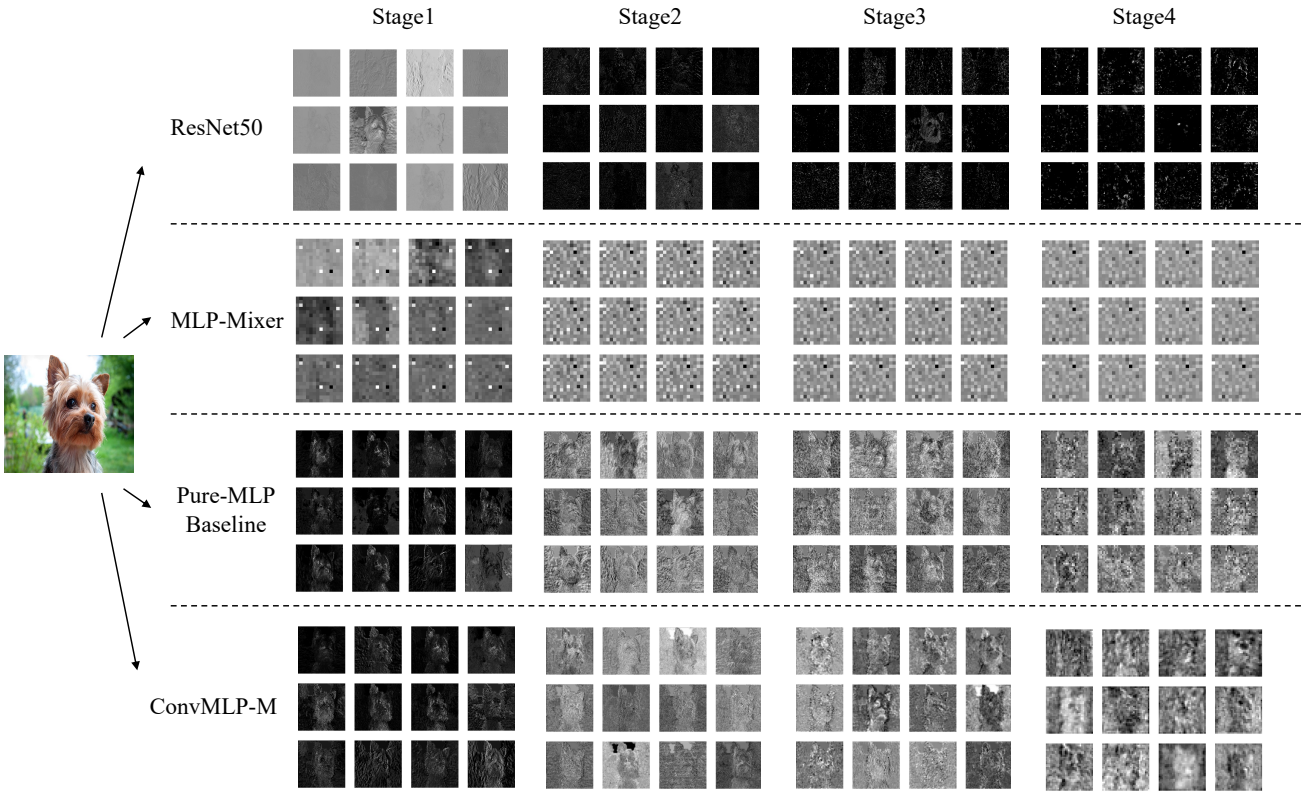

Figure 4. Visualization of feature maps in different stages of ResNet50, MLP-Mixer, Pure-MLP Baseline and ConvMLP-M. Visual representations learned by ConvMLP-M show both semantic and low-level information.

due to dimension constraint) in Figure 4 to analyze the differences in visual representations learned by these models, and similar feature maps of transformer-based model are presented in T2T-ViT [42]. We observe that representations learned by ConvMLP involve more low-level features like edges or textures compared with ResNet and more semantics compared with Pure-MLP Baseline.

5. Conclusion

In this paper, we analyze the constraints of current MLP-based models for visual representation learning: 1. Spatial MLPs only take inputs with fixed resolutions, making the transfer to downstream tasks, such as object detection and segmentation, difficult. 2. The single-stage design and fully connected layers further constrain usage due to the added complexity. To tackle these problems, we propose ConvMLP: a Hierarchical Convolutional MLP for visual representation learning through combining convolutional layers and MLPs. The architecture can be seamlessly prepended to downstream networks like RetinaNet, Mask R-CNN and Semantic FPN. Experiments further show that it can achieve competitive results on different benchmarks with fewer parameters compared to other methods. The main limitation of ConvMLP is that ImageNet performance scales slower with model size. We leave this to be explored in future works.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 1

[2] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 5

[3] Shoufa Chen, Enze Xie, Chongjian Ge, Ding Liang, and Ping Luo. Cyclemlp: A mlp-like architecture for dense prediction. *arXiv preprint arXiv:2107.10224*, 2021. 3, 6

[4] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and

benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 7

[5] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 5

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 3, 6, 7

[7] Jianyuan Guo, Yehui Tang, Kai Han, Xinghao Chen, Han Wu, Chao Xu, Chang Xu, and Yunhe Wang. Hire-mlp: Vision mlp via hierarchical rearrangement. *arXiv preprint arXiv:2108.13341*, 2021. 4

[8] Ali Hassani and Humphrey Shi. Dilated neighborhood attention transformer. *arXiv preprint arXiv:2209.15001*, 2022. 3

[9] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. *arXiv preprint arXiv:2204.07143*, 2022. 3

[10] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers. *arXiv preprint arXiv:2104.05704*, 2021. 3, 6

[11] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 5

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2, 6

[13] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 4

[14] Qibin Hou, Zihang Jiang, Li Yuan, Ming-Ming Cheng, Shuicheng Yan, and Jiashi Feng. Vision permutator: A permutable mlp-like architecture for visual recognition. *arXiv preprint arXiv:2106.12368*, 2021. 3, 6

[15] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1314–1324, 2019. 6

[16] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2

[17] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 2

[18] Zilong Huang, Youcheng Ben, Guozhong Luo, Pei Cheng, Gang Yu, and Bin Fu. Shuffle transformer: Rethinking spatial shuffle for vision transformer. *arXiv preprint arXiv:2106.03650*, 2021. 3, 4, 6

[19] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6399–6408, 2019. 7

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images, 2009. 5

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 2, 5

[22] Dongze Lian, Zehao Yu, Xing Sun, and Shenghua Gao. As-mlp: An axial shifted mlp architecture for vision. *arXiv preprint arXiv:2107.08391*, 2021. 3, 6

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 5

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 5

[25] Hanxiao Liu, Zihang Dai, David R So, and Quoc V Le. Pay attention to mlps. *arXiv preprint arXiv:2105.08050*, 2021. 1, 3, 6

[26] Yun Liu, Guolei Sun, Yu Qiu, Le Zhang, Ajad Chhatkuli, and Luc Van Gool. Transformer in convolutional neural networks. *arXiv preprint arXiv:2106.03180*, 2021. 3

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 3, 4, 5, 6

[28] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[29] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018. 2

[30] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing*, pages 722–729, 2008. 5

[31] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10428–10436, 2020. 6

[32] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 2

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 2

[34] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International*

Conference on Machine Learning, pages 6105–6114. PMLR, 2019. 2, 6

[35] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021. 1, 3, 6

[36] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021. 1, 3, 6, 7

[37] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 3, 5, 6, 7

[38] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1, 2

[39] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021. 3, 5, 6

[40] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 5

[41] Tan Yu, Xu Li, Yunfeng Cai, Mingming Sun, and Ping Li. S^2-mlp: Spatial-shift mlp architecture for vision. *arXiv preprint arXiv:2106.07477*, 2021. 3, 6

[42] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021. 8

[43] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019. 5

[44] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 5

[45] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 7