What Makes for Hierarchical Vision Transformer?
Yuxin Fang, Xinggang Wang, Member, IEEE, Rui Wu, and Wenyu Liu, Senior Member, IEEE

Abstract—Recent studies indicate that hierarchical Vision Transformer (ViT) with a macro architecture of interleaved non-overlapped window-based self-attention & shifted-window operation can achieve state-of-the-art performance in various visual recognition tasks, and challenges the ubiquitous convolutional neural networks (CNNs) using densely slid kernels. In most recently proposed hierarchical ViTs, self-attention is the de-facto standard for spatial information aggregation. In this paper, we question whether self-attention is the only choice for hierarchical ViT to attain strong performance, and study the effects of different kinds of cross-window communication methods. To this end, we replace self-attention layers with embarrassingly simple linear mapping layers, and the resulting proof-of-concept architecture termed TRANSLINEAR can achieve very strong performance in ImageNet-1k image recognition. Moreover, we find that TRANSLINEAR is able to leverage the ImageNet pre-trained weights and demonstrates competitive transfer learning properties on downstream dense prediction tasks such as object detection and instance segmentation. We also experiment with other alternatives to self-attention for content aggregation inside each non-overlapped window under different cross-window communication approaches. Our results reveal that the macro architecture, other than specific aggregation layers or cross-window communication mechanisms, is more responsible for hierarchical ViT’s strong performance and is the real challenger to the ubiquitous CNN’s dense sliding window paradigm.

Index Terms—Vision transformer, self attention, translinear.

I. INTRODUCTION

Recently, the impregnable position of convolutional neural networks (CNNs) in computer vision seems to be weakened by the emerging hierarchical Vision Transformer families [1], [2], [3], [4]. As one representative, Swin Transformer [3] and its variants (e.g., [5], [6]) with an interleaved non-overlapped window-based self-attention & cross-window token mixing paradigm are able to achieve state-of-the-art performance in image recognition, and demonstrate excellent transferability on various downstream computer vision tasks such as object detection, instance segmentation, and scene parsing. Therefore it is meaningful to conduct an in-depth study on this new architecture family, and analyze what makes it so strong.

Methodologically, Swin Transformer (Fig. 1) first partitions feature maps to a series of non-overlapped local windows, and uses multi-head self-attention (MHSA) layers to aggregate information in each window individually. Then, instead of using a dense sliding window paradigm for cross-window token mixing like CNNs [9], [10], [11], Swin Transformer proposes to shift windows between consecutive layers. By alternating these two operations, each token is able to interact with all other tokens, and the overall architecture can obtain very strong capacities.

Manuscript received 2 January 2022; revised 23 December 2022; accepted 16 May 2023. Date of publication 9 June 2023; date of current version 5 September 2023. Recommended for acceptance by R. Feris. (Corresponding author: Xinggang Wang.)

Yuxin Fang, Xinggang Wang, and Wenyu Liu are with the School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China (e-mail: yxf@ hust.edu.cn; xg wang@ hust.edu.cn; liu wy@ hust.edu.cn).

Rui Wu is with Horizon Robotics, Beijing 100094, China (e-mail: rui.wu@horizon.ai).

Digital Object Identifier 10.1109/TPAMI.2023.3282019

Most successors of Swin Transformer mainly focus on replacing the shifted-window operation with other kinds of cross-window communication, such as the spatial token shuffle operation from [5], and the messenger tokens exchange proposed in [6]. While the use of window-based MHSA is usually taken for granted and treated as the de-facto standard for token fusion within each local window.

In this paper, we ponder the question: “What Makes for Hierarchical Vision Transformer?”. Specifically, we investigate whether MHSA is the only choice to aggregate information for the hierarchical ViT family, and the effects of different cross-window communication schemes such as spatial token shuffle and auxiliary messenger tokens exchange. Previous practice from computer vision suggests that MHSA is good at capture dense and long-range contextual information [12], [13] in dense scene parsing tasks [14], [15]. Intuitively, it is somewhat too aggressive to use MHSA to capture dense contextual relation inside a local window with only $7 \times 7 = 49$ spatial tokens.

This motivates us to replace MHSA layers with linear mapping layers, one of the most common & simplest components in neural architecture design, in three representative hierarchical Vision Transformer instantiations, i.e., Swin Transformer [3], Shuffle Transformer [5], and MSG Transformer [6]. The resulting proof-of-concept model is termed as TRANSLINEAR. We find that TRANSLINEAR with embarrassingly simple linear mapping layers is sufficient for local...
content aggregation, and is able to achieve very competitive performance in ImageNet-1k image recognition benchmark [7]. Moreover, TransLinear can better leverage the pre-trained representations from image recognition and demonstrate excellent transferability in downstream dense prediction tasks such as object detection & instance segmentation.

Furthermore, we experiment with other variants for information aggregation inside each local window, e.g., depth-wise linear mapping with separable weights (denoted as “DW Linear” in Table I), and multi-layer perceptrons with intermediate activation functions (denoted as “MLP”) in Table I. We also conduct a study on different cross-window communication approaches such as spatial token shuffle (denoted as “Shuffle” in Table I) and auxiliary message tokens exchange (denoted as “MSG” in Table I). As shown in Table I, we find different content aggregation layers all give similar competitive results under the same cross-window communication approaches, and vice versa: different cross-window communication methods also achieve similar strong performance under the same content aggregation layer.

Based on the available evidence, we hypothesize that the macro architecture of Swin model families, i.e., interleaved non-overlapped window-based token mixing & cross-window communications, other than specific aggregation layers such as MHSA or specific means of cross-window communication such as shifted-window or spatial shuffle, may be more responsible for their strong performance, and is the real challenger to the ubiquitous CNN’s dense sliding window paradigm.

Please note that this paper is not an attempt to show that simple linear or MLP layers are superior to MHSA. On the contrary, we find MHSA is better than linear mapping & MLP in terms of accuracy with even fewer budgets (see Table VIII). Our goal is to abstract away from specific aggregation layers as well as cross-window communication approaches, and highlight the importance and contribution of the macro architecture of the Swin Transformer family. We hope our work can encourage the community to rethink the role of attention in neural architecture design, and shed a little light on future studies of general visual representation learning.

|   | MHSA | Linear | DW Linear | MLP |
|---|------|--------|-----------|-----|
| Shift | 80.5 | 79.7   | 79.8      | 79.9|
| Shuffle | 80.6 | 79.6   | 79.6      | 79.8|
| MSG | 80.4 | 79.5   | 79.7      | 79.7|

II. BACKGROUND AND RELATED WORK

Highly mature and robust training recipes [16] enable standard Transformer architecture [17], [18] directly inherited from natural language processing to attain excellent performance in the image recognition task even with limited data & model sizes [19], [20]. Standard Transformer [17] models sequence-to-sequence relationship in a pairwise manner with minimal structural prior via global scaled dot-product multi-head self-attention (MHSA), which scales quadratically with the sequence length. Therefore Transformer suffers from the scaling problem in spatial dimensions and fails to process high-resolution inputs with varying sizes in vision tasks.1

To more efficiently apply Vision Transformers to other downstream tasks in computer vision such as object detection, instance segmentation, and scene parsing, three key issues need to be solved:

- Involving hierarchical architectures to establish multi-scale feature representations for better handling of large variations in scales.
- Reducing memory & computation costs from global MHSA to efficiently process high-resolution inputs with varying token lengths.
- Introducing appropriate inductive biases & prior knowledge of the target task for better performance.

To mitigate the aforementioned issues, [2], [4], [22] process features with multi-resolution stages using spatial pooling operations instead of in a columnar manner. [1], [3] further propose to compute MHSA in weakly-overlapped or non-overlapped local windows. After that, many follow-up hierarchical local window-based Vision Transformers emerge and challenge the hegemonic position of CNN in computer vision [5], [6], [23].

To demystify the relation between CNN and hierarchical Vision Transformer, [24] study the inhomogeneous depth-wise convolution under modern training & optimization recipe from [3], and demonstrates that CNN can achieve similar competitive performance compared with the Swin Transformer family in various vision tasks. Furthermore, [25] and [26] study the convolution-attention hybrid architecture. Combining the strengths from both camps, convolution-attention hybrid architecture can achieve state-of-the-art performance under different resource constraints across various datasets.

Previous studies show that simple multi-layer perceptrons (MLPs) architectures are competitive with CNNs in digit recognition [27], [28], keyword spotting [29] and handwriting recognition [30]. Recently, a series of works [31], [32], [33] revisit the architecture based exclusively on columnar structured MLPs in image recognition tasks under modern training and transfer learning recipes.

As cursorily summarized in Table II, there are still two “missing pieces”, i.e., CNNs with columnar architectures, and MLPs with hierarchical architectures. [24] touches the former topic with the columnar architecture proposed in [34]. This paper conducts a primitive study to the latter one: a straightforward, simple, yet must-know model in computer vision. We argue it is inevitable to investigate the potential

1 Concurrent to our work, ConvMixer [21] studies this kind of architecture design.
of hierarchical linear mapping & MLP structures now, and we hope
the proposed proof-of-concept TRANSLINEAR model can encourage the
community to rethink the role between macro model design method-
ologies and specific network building blocks.

III. WHAT MAKES FOR HIERARCHICAL VISION
TRANSFORMER?

We first briefly review the Swin Transformer family in Section III-A,
and then introduce the proposed TRANSLINEAR in Section III-B.

A. The Swin Architecture Family

Methodologically, Swin Transformer [3] processes high-resolution
input hierarchically using multi-head self-attention (MHSA) within
non-overlapped local windows. Specifically, MHSA is used as the
feature aggregation layer to fuse content information of spatial to-
kens inside each window. Since the non-overlapped partition scheme
lacks connection across windows, Swin Transformer proposes to use
shifted-window operations between every two successive window-
based MHSA layers to encourage cross-window communications. Hier-
archical architecture design is also adopted to produce multi-resolution
representations for better handling of large scale & size variations in
visual entities.

Most successors of Swin Transformer mainly focus on replacing
shifted-window operations with other kinds of cross-window communi-
cations, such as spatial token shuffle [5] or information exchange based
on auxiliary message tokens [6], while keeping other components
unchanged, especially the MHSA block.

Overall, Swin Transformer and its variants all adopt the MHSA as
the aggregation layer for spatial token fusion, and use non-overlapped
window-based token mixing & cross-window communications in an
alternating fashion with hierarchical representations as the macro ar-
chitecture. We refer readers to [3], [5], [6] for more details of the Swin
Transformer family architectures investigated in this paper.

B. A Proof-of-Concept Model: TRANSLINEAR

Despite being greatly successful in various tasks, we question
whether MHSA is the only choice to aggregate information for the Swin
Transformer family. To this end, we attempt to use linear mapping,
one of the simplest components in neural architecture design, as a
touchstone & probe to reveal that the macro architecture (i.e., inter-
leaved non-overlapped window-based token mixing & cross-window
communication) seems to be more responsible for Swin model families’
strong performance other than specific aggregation layers such as
MHSA.

We choose three representative and publicly available instantia-
tions from the Swin model family, i.e., Swin Transformer [3], Shuffle
Transformer [5], and MSG Transformer [6]. We directly replace their
window-based MHSA layers with TRANSLINEAR layers described in
Algorithm 1, and align other components and configurations with Swin
Transformer [3].

In our default instantiation, the weights and biases of lin_map_1() and
lin_map_2() in Algorithm 1 are shared across different groups. Linear
mappings with separate parameters for different groups are
denoted as depth-wise linear mapping (“DW Linear” in Tables I and
IX) in this paper. Our controlled experiments demonstrate that using
separate weight cannot bring further improvements in linear mapping
given similar model sizes, which echoes the observation in previous
multi-head versus single-head self-attention studies [35], [36].

To some extent, TRANSLINEAR is one of the simplest possible instan-
tiations of architectures with interleaved non-overlapped window-based
token mixing & cross-window communications scheme. Despite being
simple, the TRANSLINEAR layer is more lightweight than the MHSA
layer. Therefore the lack in capacity compared with MHSA can be
compensated by deeper or wider architecture design given similar
FLOPs & parameters budgets, and the resulting architecture is still
able to achieve competitive performance (Section IV-B).

It is noteworthy that the proposed TRANSLINEAR layer enables a
fully linear mapping & MLP architecture to directly process high-
resolution input images with arbitrary shapes. This property allows
MLP-like architectures to be easily transferred to different computer
vision downstream tasks, which is infeasible for previous columnar
MLP counterparts such as [31], [32], [37]. In Table VII we demonstrate
that the transfer learning performance of TRANSLINEAR can be even on
a par with Swin Transformer in MS-COCO [38] object detection and
instance segmentation even with relatively worse supervised pre-trained
representations on ImageNet-1k [7].

Please note that TRANSLINEAR is not an attempt to show that simple
linear or MLP layers are superior to MHSA. On the contrary, we find
MHSA is better than linear mapping & MLP in terms of accuracy with
even fewer budgets (see Table VIII). Overall, the merit of TRANSLINEAR
is to abstract away from specific aggregation layers as well as cross-
window communication schemes, and highlights the importance of the
macro architecture, which seems overlooked in previous research on
hierarchical Vision Transformer.

IV. EXPERIMENTS

We first give the general experimental setup in Section IV-A, and
then report the pre-training, scaling, and transfer learning performance
of TRANSLINEAR in Section IV-B. The model analysis and ablation
study are finally conducted in Section IV-C.

A. Setup

Pre-Train Settings. The experiments are conducted on the public
available codebase of [3], [5], [6] and the timm library [39].

During pre-training, all models are trained and evaluated on
ImageNet-1k [7] benchmark following the setup in [3], [16]. We train
models with 300 epochs on ImageNet-1k for main results. For model

Algorithm 1: TRANSLINEAR Pseudocode.

```plaintext
Algorithm 1: TRANSLINEAR Pseudocode.

| B: num_windows, C: channel |
|-----------------------------|
| ws: window size, gs: group size |
| t(dim1, dim2): # transpose dim1 & dim2 |
| lin_map_1 = Linear(ws*gs, ws*gs) |
| lin_map_2 = Linear(ws*gs, ws*gs) |
| proj = PointWiseConv(C, C) |

| def LinMapper(x): |
| f1 = x.view(B, C//gs, gs*ws, ws) |
| f1 = lin_map_1(f1.t(-1, -2)).t(-1, -2) |
| f2 = x.view(B, C, C//gs) |
| f2 = lin_map_2(f2) |
| return proj(f1 + f2) |
```
analysis and ablation study, we study different content aggregation layers with model width and model depth\(^2\) same as Swin Transformer-Tiny (i.e., model width = 96, model depth = \(\{2, 2, 6, 2\}\)) using 200 epochs training schedule on ImageNet-1k unless specified.

The input resolution is \(224 \times 224\) and the window size is \(7 \times 7\) for all TRANS\(L\)INEAR models in all experiments. For a clearer study of different aggregation layers in non-overlapped windows, we remove all densely split conv-layers in the network stem and each block of [5] in this paper.

Model throughput data during inference are measured using a single Titan Xp GPU with batch size = 64 with input resolution = \(224 \times 224\). Model FLOPs during inference are measured with batch size = 1 with input resolution = \(224 \times 224\).

Transfer Learning Settings. The experiments are conducted on the public available codebase of [3] and the mmdetection library [43]. We study the transfer learning performance of ImageNet-1k 300-epoch supervised pre-trained TRANS\(L\)INEAR in the challenging MS-COCO [38] object detection and instance segmentation benchmarks using the canonical Mask R-CNN [42] framework. We fine-tune the pre-trained TRANS\(L\)INEAR with standard 1× schedule [42] on MS-COCO train split and report the transfer learning results on MS-COCO \(m\) \GW op\M s following the training and testing configurations from [3].

Model FLOPs during inference are measured with batch size = 1 and input resolution = \(1280 \times 800\).

### B. Main Results

**Results of TRANS\(L\)INEAR on ImageNet-1k.** As shown in Table III, given limited FLOPs and parameters budgets, tiny-sized TRANS\(L\)INEAR models are able to achieve competitive performance on ImageNet-1k [7] image recognition benchmark with three different cross-window communication paradigm, i.e., shifted-window [3], spatial token shuffle [5], and auxiliary message tokens exchange [6].

Along with the results in Table I, it is noteworthy that (1) shifted-window, spatial token shuffle, and auxiliary message tokens exchange all give similar strong results under the same aggregation layer, and moreover, (2) different aggregation layers are all quite competitive under the same cross-window token mixing approach.

All these results support our proposal: the macro architecture of the Swin model family, other than specific aggregation layers or specific means of cross-window communication, may be more responsible for its strong performance.

\(^2\)In this paper, we define model width is the number of channels in the first stage of the network, and model depth is the number of content aggregation layers in each stage of the network.

**Scaling TRANS\(L\)INEAR-Tiny.** There is little literature available on the scaling properties of hierarchical MLP-like models. Here we demonstrate that the tiny-sized TRANS\(L\)INEAR model is scalable.

We choose Swin TRANS\(L\)INEAR-Tiny as the model scaling start point. Width scaling [44], [45] is adopted while the model depth and input resolution are kept unchanged. The training and testing configurations are aligned with [3]. Results in Table IV show that both TRANS\(L\)INEAR-Small & TRANS\(L\)INEAR-Base can be successfully optimized, converged, and consistently benefit from more computations and larger model size.

Finding appropriate model scaling laws tailored for TRANS\(L\)INEAR as well as other MLP variants is non-trivial, since the model complexity of MLP-like architectures is a linear combination of two different parts: (1) the content aggregation layer part for depth-wise or group-wise spatial token mixing only, and (2) the feed-forward network part for point-wise or token-wise feature transformation only, which is different from the complexity of CNNs. Consequently, previous successful practice in CNNs scaling cannot apply to TRANS\(L\)INEAR in a principled way and can only be verified one-by-one experimentally. We leave the study of sophisticated model scaling laws on TRANS\(L\)INEAR for future work.

**Comparisons With Other Columnar MLP-Like Variants.** We summarize some recently proposed columnar MLP-like architectures in Table V. The proposed TRANS\(L\)INEAR demonstrates superior performance with fewer FLOPs and parameters. The window partitioning operation and hierarchical representations introduce 2D locality bias and invariance to TRANS\(L\)INEAR. Therefore it is not a surprise that TRANS\(L\)INEAR is more efficient and competitive than global & columnar MLP architectures such as MLP-Mixer and ResMLP by leveraging these design priors.

**Comparisons With Representative Hierarchical MLP-Like Variants and ConVs.** In Table VI, we compare our proposed TRANS\(L\)INEAR with representative hierarchical MLP-like architectures as well as ConvNets. Swin-Mixer [3] is a recently proposed MLP-like architecture with hierarchical representations based on the Swin Transformer family, which is concurrent to our TRANS\(L\)INEAR. We demonstrate that both the tiny-sized and small-sized TRANS\(L\)INEAR can outperform the corresponding Swin-Mixer counterparts even with fewer computation budgets.

**Transfer Learning Performance of TRANS\(L\)INEAR.** There is little literature available on the transferability of MLP-like architecture to downstream dense prediction tasks such as object detection and instance segmentation. Most available MLP variants using global kernels for

---

**TABLE III**

**RESULTS OF DIFFERENT TRANS\(L\)INEAR-TINY VARIANTS ON IMAGE NET-1 K: IMAGE RECOGNITION BENCHMARK**

| Method          | Model Width | Model Depth | #Params. (M) | FLOPs (G) | Throughput (Img/s) | Top-1 Acc. |
|-----------------|-------------|-------------|--------------|-----------|-------------------|------------|
| Swin TRANS\(L\)INEAR-Tiny | 64         | \(\{2, 2, 22, 4\}\) | 24.6        | 1.0       | 320               | 80.5       |
| Shuffle TRANS\(L\)INEAR-Tiny | 64         | \(\{2, 2, 22, 4\}\) | 24.6        | 1.0       | 328               | 80.6       |
| MSG TRANS\(L\)INEAR-Tiny     | 64         | \(\{2, 2, 22, 4\}\) | 30.6        | 1.5       | 380               | 80.4       |

---

**TABLE IV**

**SCALING THE TINY-SIZED TRANS\(L\)INEAR MODEL. WE USE THE SHIFTED-WINDOW OPERATION [3] AS THE DEFAULT CROSS-WINDOW COMMUNICATION SCHEME**

| Method          | Model Width | Model Depth | #Params. (M) | FLOPs (G) | Throughput (Img/s) | Top-1 Acc. |
|-----------------|-------------|-------------|--------------|-----------|-------------------|------------|
| Swin TRANS\(L\)INEAR-Tiny | 64         | \(\{2, 2, 22, 4\}\) | 24.6        | 1.0       | 320               | 80.5       |
| Swin TRANS\(L\)INEAR-Small | 96         | \(\{2, 2, 22, 4\}\) | 54.9        | 1.8       | 201               | 82.0       |
| Swin TRANS\(L\)INEAR-Base     | 128        | \(\{2, 2, 22, 4\}\) | 97.3        | 3.0       | 143               | 82.5       |

---

\(\) is the number of channels in the first stage of the network, and model depth is the number of content aggregation layers in each stage of the network.
spatial token mixing, therefore their spatial kernel sizes are fixed and highly correlated to the input resolution during pre-training, which largely limits the applications on high-resolution inputs with varying shapes and sizes during transfer learning.

The proposed TRANSLINEAR can naturally overcome this issue via window-based spatial token mixing inherited from the Swin model family. TRANSLINEAR is able to directly process arbitrary resolutions zero-padded to be divisible by the window size. Results in Table VII demonstrate that the performance of TRANSLINEAR is on a par with Swin Transformer in MS-COCO object detection and instance segmentation benchmarks. Meanwhile, as the macro architecture becomes larger, our TRANSLINEAR is less scalable compared with the MHSA counterpart in terms of the downstream transfer learning results.

C. TRANSLINEAR Analysis and Ablation Study

In this section, we study the impact of model width & model depth configurations, weight sharing properties of linear mapping layers, as well as the number of groups. Overall, we conclude that TRANSLINEAR is quite robust to different model choices and configurations thanks to the strong macro architecture.

Going Wider or Deeper? We study the model width and model depth configurations for the tiny-sized TRANSLINEAR model. A high-performance tiny-sized model can also be served as a promising start point for model scaling.

Since a single TRANSLINEAR layer is much lighter than a single window-based MHSA layer from [3] in terms of both parameters and computations, we need to adjust the model width and depth of TRANSLINEAR to align with the budgets for Swin Transformer-Tiny. The resulting models are termed as Swin TRANSLINEAR-Tiny (Wide) and Swin TRANSLINEAR-Tiny (Deep).

As shown in Table VIII, the wider model seems to be more speed friendly while the deeper model is a lot more parameters & FLOPs efficient. We choose Swin TRANSLINEAR-Tiny (Deep) as our default tiny-sized TRANSLINEAR model instantiation.

Linear or Depth-Wise Linear? In this paper, depth-wise linear (DW Linear) layers refer to linear mapping with separate or non-shared weights & biases for each group. Therefore the model parameters increase while the theoretical FLOPs are kept unchanged when using DW linear layers instead of shared linear layers. As shown in Table IX, DW linear layers bring no significant improvement, which echoes the observation in previous multi-head versus single-head self-attention studies [35], [36] to some extent. Therefore we choose to share linear
weights & biases across different groups as our default instantiation for better parameters efficiency.

**Linear Mapping or MLP?** In Table X, we investigate the impact of using simple linear mapping layers and more sophisticated MLP layers for content aggravation inside each non-overlapped window. The “MLP” in Table X of our paper means we simply replace the Linear([ws*gs, ws*gs]) with Linear([ws*gs, ws*gs/2])→GELU()→Linear([ws*gs/2, ws*gs]) in Algorithm I of our paper. The results suggest that using MLPs with intermediate GELU() activation functions cannot bring further improvements.

This observation in our hierarchical TRANSLINEAR is somewhat in line with the findings from the columnar counterpart [32], where ResMLP models also adopt simple patch-to-patch (or token-to-token) linear transformations instead of MLP layers in MLP-Mixer [31] for cross-patch communications.

**Number of Groups (#Group) for Linear Layers.** In Table XI, we study the impact of different numbers of groups (#Groups) in linear layers. The weights & biases of linear layers are shared across groups. We find setting #Groups too large or too small is harmful to performance, while other choices yield similar results. In this paper, #Groups = 32 (i.e., gs = 3 x 2(#stage−1)) in Algorithm I is chosen as the default configuration for all-sized TRANSLINEAR models.

### References

[1] A. Vaswani, P. Ramachandran, A. Srinivas, N. Parmar, B. Hechtman, and J. Shlens, “Scaling local self-attention for parameter efficient visual backbones,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 12889–12899.

[2] W. Wang et al., “Pyramid vision transformer: A versatile backbone for dense prediction without convolutions,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 548–558.

[3] Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 9992–10002.

[4] H. Fan et al., “Multiscale vision transformers,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 6804–6815.

[5] Z. Huang, Y. Ben, G. Liao, P. Cheng, G. Yu, and B. Fu, “Shuffle transformer: Rethinking spatial shuffle for vision transformer,” 2021, arXiv:2106.03650.

[6] J. Fang, L. Xie, X. Wang, X. Zhang, W. Liu, and Q. Tian, “MSG-transformer: Exchanging local spatial information by manipulating mesenger tokens,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 12053–12062.

[7] O. Russakovsky et al., “ImageNet large scale visual recognition challenge,” Int. J. Comput. Vis., vol. 115, pp. 211–252, 2015.

[8] J. L. Ba, J. R. Kiros, and G. E. Hinton, “Layer normalization,” 2016, arXiv:1607.06450.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proc. 25th Int. Conf. Neural Inf. Process. Syst., 2012, pp. 1097–1105.

[10] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. Int. Conf. Learn. Representations, 2015.

[11] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[12] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2018, pp. 7794–7803.

[13] Z. Huang, X. Wang, L. Huang, C. Huang, Y. Wei, and W. Liu, “CCNet: Cross-cross attention for semantic segmentation,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 603–612.
[14] Z. Tu, X. Chen, A. L. Yuille, and S.-C. Zhu, “Image parsing: Unifying segmentation, detection, and recognition,” Int. J. Comput. Vis., vol. 63, pp. 113–140, 2005.

[15] J. Tighe, M. Niethammer, and S. Lazebnik, “Scene parsing with object instances and occlusion ordering,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2014, pp. 3748–3755.

[16] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, “Training data-efficient image transformers & distillation through attention,” in Proc. 38th Int. Conf. Mach. Learn., 2021, pp. 10347–10357.

[17] A. Vaswani et al., “Attention is all you need,” in Proc. 31st Int. Conf. Neural Inf. Process. Syst., 2017, pp. 6000–6010.

[18] A. Dosovitskiy et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” in Proc. Int. Conf. Learn. Representations, 2021.

[19] L. Yuan et al., “Tokens-to-token ViT: Training vision transformers from scratch on ImageNet,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 538–547.

[20] K. Han, A. Xiao, E. Wu, J. Guo, C. Xu, and Y. Wang, “Transformer in transformer,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2021.

[21] A. Trockman and J. Z. Kolter, “Patches are all you need?” 2022, arXiv:2201.09792.

[22] B. Heo, S. Yun, D. Han, S. Chun, J. Choe, and S. J. Oh, “Rethinking spatial dimensions of vision transformers,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 11196–11205.

[23] Z. Zhang, H. Zhang, I. Zhao, T. Chen, and T. Pfister, “Aggregating nested transformers,” 2021, arXiv:2105.12723.

[24] Q. Han et al., “On the connection between local attention and dynamic depth-wise convolution,” in Proc. Int. Conf. Learn. Representations, 2022.

[25] W. Xu, Y. Xu, T. Chang, and Z. Tu, “Co-scale conv-attentional image transformers,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2021, pp. 9961–9970.

[26] Z. Dai, H. Liu, Q. V. Le, and M. Tan, “CoAtNet: Marrying convolution and attention for all data sizes,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2021, pp. 3965–3977.

[27] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, “Deep big multilayer perceptrons for digit recognition,” in Neural Networks: Tricks of the Trade, Berlin, Germany: Springer, 2012.

[28] P. Y. Simard, D. Steinkraus, and J. C. Platt, “Best practices for convolutional neural networks applied to visual document analysis,” in Proc. 7th Int. Conf. Document Anal. Recognit., 2003, pp. 958–963.

[29] C. Chatelain, “Extraction de séquences numériques dans des documents manuscrits quelconques,” Ph.D. dissertation, Dept. Inf. Sci., Université de Rouen, Mont-Saint-Aignan, France, 2006.

[30] T. Blué, “Deep neural networks for large vocabulary handwritten text recognition,” Ph.D. dissertation, Dept. Inf. Sci., Université Paris sud 11, France, 2015.

[31] I. O. Tolstikhin et al., “MLP-mixer: An all-MLP architecture for vision,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2021.

[32] H. Touvron et al., “ResMLP: Feedforward networks for image classification with data-efficient training,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 4, pp. 5314–5321, Apr. 2023.

[33] M.-H. Guo, Z.-N. Liu, T.-J. Mu, and S.-M. Hu, “Beyond self-attention: External attention using two linear layers for visual tasks,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 5, pp. 5436–5447, May 2023.

[34] L. Yuan, Q. Hou, Z. Jiang, J. Feng, and S. Yan, “VOLO: Vision outlooker for visual recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 5, pp. 6575–6586, May 2023.

[35] L. Liu, J. Liu, and J. Han, “Multi-head or single-head? An empirical comparison for transformer training,” 2021, arXiv:2106.09650.

[36] P. Michel, O. Levy, and G. Neubig, “Are sixteen heads really better than one?,” in Proc. 38th Int. Conf. Neural Inf. Process. Syst., H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, Eds., 2019, Art. no. 1257.

[37] H. Liu, Z. Dai, D. So, and Q. V. Le, “Pay attention to MLPs,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2021, pp. 9204–9215.

[38] T. Lin et al., “Microsoft COCO: Common objects in context,” in Proc. Eur. Conf. Comput. Vis., 2014, pp. 740–755.

[39] R. Wightman, “PyTorch image models,” 2019. [Online]. Available: https://github.com/rwightman/pytorch-image-models

[40] Y. Rao, W. Zhao, Z. Zhu, J. Lu, and J. Zhou, “Global filter networks for image classification,” in Proc. Int. Conf. Neural Inf. Process. Syst., 2021.

[41] T. Yu, X. Li, Y. Cai, M. Sun, and P. Li, “S2-MLP: Spatial-shift MLP architecture for vision,” in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis., 2022, pp. 3615–3624.

[42] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, “Mask R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 2980–2988.

[43] K. Chen et al., “MMDetection: Open MMLab detection toolbox and benchmark,” 2019, arXiv:1906.07155.

[44] S. Zagoruyko and N. Komodakis, “Wide residual networks,” in Proc. Brit. Mach. Vis. Conf., 2016, pp. 87.1–87.12.

[45] A. G. Howard et al., “MobileNets: Efficient convolutional neural networks for mobile vision applications,” 2017, arXiv:1704.04861.

[46] D. Hendrycks and K. Gimpel, “Gaussian error linear units (GELUs),” 2016, arXiv:1606.08415.