Multi-Scale High-Resolution Vision Transformer for Semantic Segmentation

Jiaqi Gu[1]*, Hyoukjun Kwon[2], Dilin Wang[2], Wei Ye[2], Meng Li[2], Yu-Hsin Chen[2],
Liangzhen Lai[2], Vikas Chandra[2], David Z. Pan[1]
[1]University of Texas at Austin, [2]Meta Platforms Inc.
jqgu@utexas.edu, dpan@ece.utexas.edu
{hyoukjun, wdilin, weiye, meng.li, yhchen, liangzhen, vchandra}@fb.com

Abstract

Vision Transformers (ViTs) have emerged with superior performance on computer vision tasks compared to the convolutional neural network (CNN)-based models. However, ViTs mainly designed for image classification will generate single-scale low-resolution representations, which makes dense prediction tasks such as semantic segmentation challenging for ViTs. Therefore, we propose `HRViT`*, which enhances ViTs to learn semantically-rich and spatially-precise multi-scale representations by integrating high-resolution multi-branch architectures with ViTs. We balance the model performance and efficiency of* `HRViT` *by various branch-block co-optimization techniques. Specifically, we explore heterogeneous branch designs, reduce the redundancy in linear layers, and augment the attention block with enhanced expressiveness. Those approaches enabled* `HRViT` *to push the Pareto frontier of performance and efficiency on semantic segmentation to a new level, as our evaluation results on ADE20K and Cityscapes show.* `HRViT` *achieves 50.20% mIoU on ADE20K and 83.16% mIoU on Cityscapes, surpassing state-of-the-art MiT and CSWin backbones with an average of* ***+1.78*** *mIoU improvement,* ***28%*** *parameter saving, and* ***21%*** *FLOPs reduction, demonstrating the potential of* `HRViT` *as a strong vision backbone for semantic segmentation. Our code is publicly available* [1].

1. Introduction

Dense prediction tasks such as semantic segmentation are important computer vision workloads on emerging intelligent computing platforms, e.g., AR/VR devices. Convolutional neural networks (CNNs) have rapidly evolved with significant performance improvement in semantic segmentation [1, 4, 19, 21, 25, 29]. Beyond classical CNNs, vision Transformers (ViTs) have emerged with competitive performance in computer vision tasks [2, 3, 6, 12, 13, 18, 20, 28, 31, 32, 35, 36, 39, 43]. Benefiting from the self-attention operations, ViTs embrace strong expressivity with long-distance information interaction and dynamic feature aggregation. However, ViT [13] produces single-scale and low-resolution representations, which are not friendly to semantic segmentation that requires high position sensitivity and fine-grained image details.

To cope with the challenge, various ViT backbones that yield multi-scale representations were proposed for semantic segmentation [6, 12, 20, 30, 31, 35, 38]. However, they still follow a classification-like network topology with a *sequential* or *series* architecture. Based on complexity consideration, they gradually downsample the feature maps to extract higher-level *low-resolution (LR) representations* and directly feed each stage's output to the downstream segmentation head. Such sequential structures lack enough cross-scale interaction thus cannot produce high-quality *high-resolution (HR) representations*.

HRNet [29] was proposed to solve the problem outside of ViT context, which enhances the cross-resolution interaction with a multi-branch architecture maintaining all resolutions throughout the network. HRNet extracts multi-resolution features in parallel and fuses them repeatedly to generate high-quality HR representations with rich semantic information. Such a design concept has achieved great success in various dense prediction tasks. Nevertheless, its expressivity is limited by small receptive fields and strong inductive bias from cascaded convolution operations. To deal with the challenge, some HRNet variants such as Lite-HRNet [37] and HR-NAS [11] are proposed. However, those improved HRNet designs are still mainly based on the convolutional building blocks, and their demonstrated performance on semantic segmentation is still far behind the state-of-the-art (SoTA) scores of ViT counterparts.

Therefore, *synergistically* integrating HRNet with ViTs is an approach to be explored for further performance improvement. By combining those two approaches, ViTs can obtain rich multi-scale representability from the HR architecture, while HRNet can gain a larger receptive field

*Work done during an internship at Meta Platforms Inc.
[1] https://github.com/facebookresearch/HRViT

from the attention operations. However, migrating the success of HRNet to ViT backbones is non-trivial. Given the high complexity of multi-branch HR architectures and self-attention operations, simply replacing all convolutional residual blocks in HRNet with Transformer blocks will encounter severe scalability issues. The inherited high representation power from *multi-scale* can be overwhelmed by the prohibitive latency and energy cost on hardware without careful architecture-block co-optimization.

Therefore, we propose `HRViT`, an efficient *multi-scale high-resolution* vision Transformer backbone specifically optimized for semantic segmentation. `HRViT` enables multi-scale representation learning in ViTs and improves the efficiency based on the following approaches: (1) `HRViT`'s multi-branch HR architecture extracts multi-scale features in parallel with cross-resolution fusion to enhance the multi-scale representability of ViTs; (2) `HRViT`'s augmented local self-attention removes redundant keys and values for better efficiency and enhances the model expressivity with extra parallel convolution paths, additional nonlinearity units, and auxiliary shortcuts for feature diversity enhancement; (3) `HRViT` adopts mixed-scale convolutional feedforward networks to fortify the multi-scale feature extraction; (4) `HRViT`'s HR convolutional stem and efficient patch embedding layers maintain more low-level fine-grained features with reduced hardware cost. Also, distinguished from the HRNet-family, `HRViT` follows a unique heterogeneous branch design to balance efficiency and performance, which is not simply an improved HRNet or a direct ensemble of HRNet and self-attention but a new topology of pure ViTs mainly constructed by self-attention with careful branch-block co-optimization.

Based on the approaches in `HRViT`, we make the following contributions:

- We deeply investigate the multi-scale representation learning in vision Transformers and propose `HRViT` that integrates multi-branch high-resolution architectures with vision Transformers.
- To enhance the efficiency of `HRViT` for scalable HRViT integration, we propose a set of approaches as follows: exploiting the redundancy in Transformer blocks, developing performance-efficiency co-optimized building blocks, and adopting heterogeneous branch designs.
- We evaluate `HRViT` on ADE20K and Cityscapes and present results that push the Pareto frontier of performance and efficiency forward as follows: `HRViT` achieves 50.20% mIoU on ADE20K `val` and 83.16% mIoU on Cityscapes `val` for semantic segmentation tasks, outperforming SoTA MiT and CSWin backbones with 1.78 higher mIoU, 28% fewer parameters, and 21% lower FLOPs, on average.

2. Proposed HRViT Architecture

Recent advances in vision Transformer backbone designs mainly focus on attention operator innovations. A new topology design can create another dimension to unleash the potential of ViTs with even stronger vision expressivity. Extending the sequential topology of ViTs to the multi-branch structure, inspired by HRNet, is a promising approach for performance improvement. An important question that remains to be answered is whether the *success of HRNet can be efficiently migrated to ViT backbones* to consolidate their leading position in dense prediction tasks such as semantic segmentation.

In this section, we delve into the multi-scale representation learning in ViTs and introduce an efficient integration of the HR architecture and Transformer.

2.1. Architecture overview

As illustrated in Figure 1, the first part of `HRViT` consists of a convolutional stem to reduce spatial dimensions while extracting low-level features. After the convolutional stem, `HRViT` deploys four progressive Transformer stages where the n-th stage contains n *parallel multi-scale* Transformer branches. Each stage can have one or more modules. Each module starts with a lightweight dense fusion layer to achieve cross-resolution interaction and an efficient patch embedding block for local feature extraction, followed by repeated augmented local self-attention blocks (`HRViTAttn`) and mixed-scale convolutional feedforward networks (`MixCFN`). Unlike sequential ViT backbones that progressively reduce the spatial dimension to generate pyramid features, we *maintain the HR features throughout the network* to strengthen the quality of HR representations via cross-resolution fusion.

2.2. Efficient HRViT component design

A straightforward choice to fuse HRNet and ViTs is to replace convolutions in HRNet with self-attentions. However, given the high complexity of multi-branch HRNet and self-attentions, this brute-force combining can cause an explosion in memory footprint, parameter size, and computational cost. In this section, we will discuss how to design `HRViT` blocks with balanced efficiency and performance.

Augmented cross-shaped local self-attention. To achieve high performance with improved efficiency, a hardware-efficient self-attention operator is necessary. We adopt one of the SoTA efficient attention designs, cross-shaped self-attention [12], as our baseline attention operator. Based on that, we design our *augmented cross-shaped local self-attention* `HRViTAttn` illustrated in Figure 2, which provides the following benefits: (1) *Fine-grained attention*: Compared to globally-downsampled attentions [30,35], this one has fine-grained feature aggregation that preserves detailed information. (2) *Approximate global view*: By using

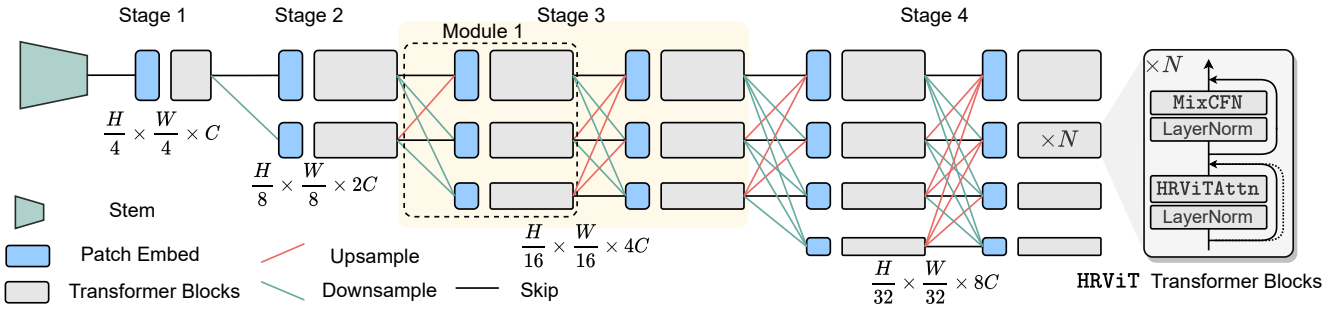


Figure 1. The overall architecture of our proposed HRViT. It progressively expands to 4 branches. Each stage has multiple modules. Each module contains multiple Transformer blocks.

two parallel orthogonal local attentions, this attention can collect global information. (3) *Scalable complexity*: one dimension of the window is fixed, which avoids quadratic complexity to image sizes.

To balance the performance and efficiency, we introduce our augmented version, denoted as HRViTAttn, with several key optimizations. In Figure 2a, we follow the cross-shaped window partitioning approach in CSWin that separates the input $x \in \mathbb{R}^{H\times W\times C}$ into two parts $\{x_H, x_V \in \mathbb{R}^{H\times W\times C/2}\}$. x_H is partitioned into disjoint horizontal windows, and the other half x_V is chunked into vertical windows. The window is set to $s \times W$ or $H \times s$. Within each window, the patch is chunked into K d_k-dimensional heads, then a local self-attention is applied,

$$\begin{aligned}
\texttt{HRViTAttn}(x) &= \texttt{BN}\big(\sigma(W^O[y_1, \cdots, y_k, \cdots, y_K])\big) \\
y_k &= z_k + \texttt{DWConv}\big(\sigma(W_k^V x)\big) \\
[z_k^1, \cdots, z_k^M] = z_k &= \begin{cases} \texttt{H-Attn}_k(x), & 1 \le k < K/2 \\ \texttt{V-Attn}_k(x), & K/2 \le k \le K \end{cases} \\
z_k^m &= \texttt{MHSA}(W_k^Q x^m, W_k^K x^m, W_k^V x^m) \\
[x^1, \cdots, x^m, \cdots, x^M] &= x, \quad x^m \in \mathbb{R}^{(H/s)\times W\times C},
\end{aligned} \tag{1}$$

where $W_k^Q, W_k^K, W_k^V \in \mathbb{R}^{d_k\times C}$ are projection matrices to generate query Q_k, key K_k, and value V_k tensors for the k-th head, $W^O \in \mathbb{R}^{C\times C}$ is the output projection matrix, and σ is Hardswish activation. If the image sizes are not a multiple of window size, e.g., $s\lceil H/s\rceil > H$, we apply zero-padding to inputs x_H or x_V to allow a complete K-th window, shown in Figure 2b. Then the padded attentions are masked to 0 to avoid incoherent semantic correlation.

The original QKV linear layers are quite costly in computation and parameters. We share the linear projections for key and value tensors in HRViTAttn to save computation and parameters as follows,

$$\texttt{MHSA}(W_k^Q x^m, W_k^V x^m, W_k^V x^m) = \texttt{softmax}\Big(\frac{Q_k^m\,(V_k^m)^T}{\sqrt{d_k}}\Big)V_k^m, \tag{2}$$

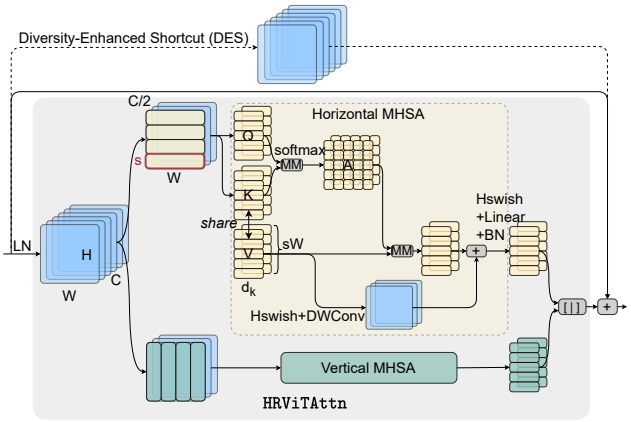


Figure 2. (a) HRViTAttn: augmented cross-shaped self-attention with a parallel CONV path and an efficient diversity-enhanced shortcut. (b) Window zero-padding with attention map masking.

In addition, we introduce an auxiliary path with parallel depth-wise convolution to inject inductive bias to facilitate training. Unlike the local positional encoding in CSWin, our parallel path is *nonlinear and applied on the entire 4-D feature map $W^V x$ without window-partitioning*. This path can be treated as an inverted residual module sharing point-wise convolutions with the linear projection layers in self-attention. This shared path can effectively inject inductive bias and reinforce local feature aggregation with marginal hardware overhead.

As a performance compensation for the above key-value sharing, we introduce an extra Hardswish function to improve the nonlinearity. We also append a BatchNorm (BN)

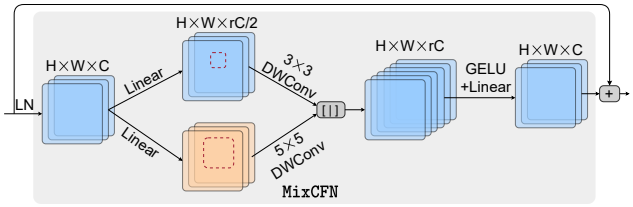


Figure 3. MixCFN with multiple depth-wise convolution paths to extract multi-scale local information.

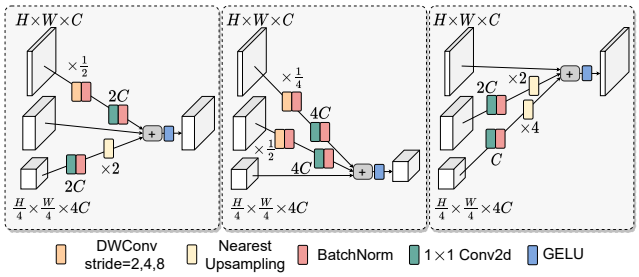


Figure 4. Cross-resolution fusion layers with channel matching, up-scaling, and down-sampling.

layer initialized to an identity projection to stabilize the distribution for better trainability. Motivated by recent studies on the importance of shortcuts in ViTs [24], we add a channel-wise projector as a diversity-enhanced shortcut (DES). Unlike the augmented shortcut [27], our shortcut has higher nonlinearity and does not depend on hardware-unfriendly Fourier transforms. The projection matrix in our DES $\mathcal{P}^{C\times C}$ is approximated by Kronecker decomposition $\mathcal{P} = A^{\sqrt{C}\times\sqrt{C}} \otimes B^{\sqrt{C}\times\sqrt{C}}$ to minimize parameter cost. Then we fold x as $\tilde{x} \in \mathbb{R}^{HW\times\sqrt{C}\times\sqrt{C}}$ and convert $(A \otimes B)x$ into $(A\tilde{x}B^T)$ to save computations. We further insert Hardswish after the B projection to increase the nonlinearity,

$$\texttt{DES}(x) = A \cdot \texttt{Hardswish}(\tilde{x}B^T). \tag{3}$$

Mixed-scale convolutional feedforward network. Inspired by the *MixFFN* in MiT [35] and multi-branch inverted residual blocks in HR-NAS [11], we design a mixed-scale convolutional FFN (MixCFN) by inserting two multi-scale depth-wise convolution paths between two linear layers, shown in Figure 3. After LayerNorm, we expand the channel by a ratio of r, then split it into two branches. The 3×3 and 5×5 depth-wise convolutions (DWConvs) are used to increase the multi-scale local information extraction of HRViT. For efficiency consideration, we exploit the channel redundancy by reducing the MixCFN expansion ratio r from 4 [20, 35] to 2 or 3 with marginal performance loss on medium to large models.

Downsampling stem. In semantic segmentation tasks, images are high resolution, e.g., 1024×1024. Self-attention operators are expensive as their complexity is quadratic to image sizes. To address the scalability issue when processing large images, we down-sample the inputs by 4× before feeding into the main body of HRViT. We do not use attention operations in the stem since early convolutions are more effective to extract low-level features than self-attentions [15, 34]. As early convolutions, we follow the design in HRNet and use two stride-2 CONV-BN-ReLU blocks as a stronger downsampling stem to extract C-channel features with more information maintained, unlike prior ViTs [6, 20, 35] that used a stride-4 convolution.

Efficient patch embedding. Before Transformer blocks in each module, we add a patch embedding block (CONV-LayerNorm) on each branch, which is used to match channels and extract patch information with enhanced inter-patch communication. However, the patch embedding layers have a non-trivial hardware cost in the HR architecture since each module at stage-n will have n embedding blocks. Therefore, we simplify the patch embedding to be a point-wise CONV followed by a depth-wise CONV [16],

$$\texttt{EffPatchEmbed}(x) = \texttt{LN}\big(\texttt{DWConv}(\texttt{PWConv}(x))\big). \tag{4}$$

Cross-resolution fusion layer. The cross-resolution fusion layer is critical for HRViT to learn high-quality HR representations, shown in Figure 4. To enhance cross-resolution interaction, we insert repeated cross-resolution fusion layers at the beginning of each module following the approach in HRNet [29, 37].

To *help LR features maintain more image details and precise position information*, we merge them with down-sampled HR features. Instead of using a progressive convolution-based downsampling path to match tensor shapes [29, 37], we employ a direct down-sampling path to minimize hardware overhead. In the down-sampling path between the i-th input and j-th output ($j > i$), we use a depth-wise separable convolution with a stride of 2^{j-i} to shrink the spatial dimension and match the output channels. The kernel size used in the DWConv is (2^{j-i}+1) to create patch overlaps. Those HR paths inject more image information into the LR path to *mitigate information loss* and *fortify gradient flows* during backpropagation to facilitate the training of deep LR branches.

On the other hand, the receptive field is usually limited in the HR blocks as we minimize the window size and branch depth on HR paths. Hence, we merge LR representations into HR paths to help them *obtain higher-level features with a larger receptive field*. Specifically, in the up-scaling path ($j < i$), we first increase the number of channels with a point-wise convolution and up-scale the spatial dimension via a nearest neighbor interpolation with a rate of 2^{i-j}. When i=j, we directly pass the features to the output as a skip connection. Note that in HR-NAS [11], the dense

Feature/Arch.	HR ($\frac{1}{4}\times, \frac{1}{8}\times$)	MR ($\frac{1}{16}\times$)	LR ($\frac{1}{32}\times$)
Memory cost	High	Medium	Low
Computation	Heavy	Moderate	Light
#Params	Small	Medium	Large
Eff. on class.	Not quite useful	Important	Important
Feat. granularity	Fine	Medium	Coarse
Receptive field	Local	Region	Global
Window size	Narrow (s=1,2)	Wide (s=7)	Wide (s=7)
Depth	Shallow ($\sim$5-6)	Deep (20-30)	Shallow ($\sim$4)

Table 1. Qualitative cost and functionality analysis. Window sizes and depth are given for each branch. *Eff. on class.* and *Feat. granularity* are short for effectiveness on image-level classification and feature granularity.

fusion is simplified by a sparse fusion module where only neighboring resolutions are merged. This technique is not considered in HRViT since it saves marginal hardware cost but leads to a noticeable accuracy drop, which will be discussed in subsection 3.2.

2.3. Heterogeneous HRViT branch design

Given the efficient HRViT components we introduced above, the second challenge is how to integrate them in a scalable and efficient way. In this section, we provide a solution by introducing a heterogeneous multi-branch architecture to further push the efficiency boundary.

Heterogeneous branch configuration. For the branch architecture in HRViT, we need to determine the number of Transformer blocks assigned for each branch. Simply assigning the same number of blocks with the same local self-attention window size on each module will result in intractably high computational costs. Therefore, we analyze the functionality and cost of each branch in Table 1, and we propose a simple design heuristic based on the analysis.

We analyze (1) the number of parameters and (2) the number of floating-point operations (FLOPs) in HRViTAttn and MixCFN blocks on the i-th branch ($i = 1, 2, 3, 4$) as follows:

$$\begin{aligned}
\texttt{Params}_{\texttt{HRViTAttn},i} &= \mathcal{O}(4^{i-1}C^2 + 2^{i-1}C),\\
\texttt{Params}_{\texttt{MixCFN},i} &= \mathcal{O}(4^{i-1}C^2 r_i + 2^{i-1}C r_i),\\
\texttt{FLOPs}_{\texttt{HRViTAttn},i} &= \mathcal{O}\Big(HWC^2 + \frac{CHW(H+W)s_i}{4^{i-1}}\Big),\\
\texttt{FLOPs}_{\texttt{MixCFN},i} &= \mathcal{O}\Big(r_i HWC^2 + \frac{r_i HWC}{2^{i-1}}\Big).
\end{aligned} \tag{5}$$

We use Equation 5 to compare the memory cost, the computation, the number of parameters, and computation in Table 1.

Based on the complexity analysis, we observe that the first and second HR branches ($i = 1, 2$) involve a high memory and computational cost. Hence, those HR branches typically can not afford a large enough receptive field for image-level classification. On the other hand, they are parameter-efficient and able to provide fine-grained detail calibration in segmentation tasks. Thus, we *use a narrow attention window size and use a minimum number of blocks on two HR paths*.

We observe that the most important branch is the third one with a medium resolution (MR). Given its medium hardware cost, we can afford *a deep branch with a large window size on the MR path* to provide large receptive fields and well-extracted high-level features.

The lowest resolution (LR) branch contains most parameters and is very useful to provide high-level features with a global receptive field to generate coarse segmentation maps. However, its small spatial sizes result in too much loss of image details. Therefore, we only deploy *a few blocks with a large window size on the LR branch* to improve high-level feature quality under parameter budgets.

Nearly-even block assignment. A unique problem in HRViT is to determine *how to assign blocks to each module* In HRViT, we need to assign 20 blocks to 4 modules on the 3rd path. To maximize the average depth of the network ensemble and help input/gradient flow through the deep Transformer branch, we employ a *nearly-even partitioning*, e.g., 6-6-6-2, and exclude an extremely unbalanced assignment, e.g., 17-1-1-1. Different block assignment strategies are compared in Appendix. B.1.

2.4. Architectural variants

As shown in Table 2 with three design variants of HRViT, variants of HRViT scale in both network depth and width. We follow the aforementioned design guidance and evenly assign 5-6 Transformer blocks to HR branches, 20-24 blocks to the MR branch, and 4-6 blocks to the LR branch. Window sizes are set to (1,2,7,7) for 4 branches. We use relatively large MixCFN expansion ratios in small variants for higher performance and reduce the ratio to 2 on larger variants for better efficiency. We gradually follow the scaling rule from CSWin [12] to increase the basic channel C for the highest resolution branch from 32 to 64. #Blocks and #channels can be flexibly tuned for the *3rd/4th branch* to match a specific hardware cost.

3. Experiments

We pretrain all models on ImageNet-1K [10] and conduct experiments on ADE20K [45] and Cityscapes [8] for semantic segmentation. We compare the performance and efficiency of our HRViT with SoTA ViT backbones, i.e., Swin [20], Twins [6], MiT [35], and CSWin [12].

3.1. Semantic segmentation on Cityscapes and ADE20K

On semantic segmentation, HRViT achieves the best performance-efficiency Pareto front, surpassing the SoTA

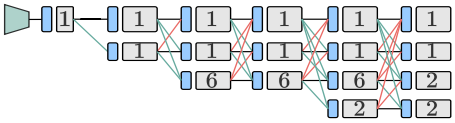
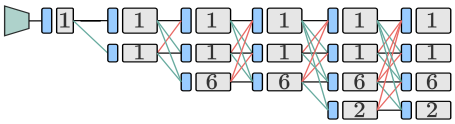
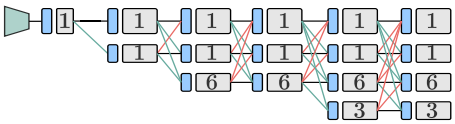
Variant	Architecture design	Window s	MixCFN ratio r	Channel C	Head dim d_k
HRViT-b1		1	4	32	16
		2	4	64	32
		7	4	128	32
		7	4	256	32
HRViT-b2		1	2	48	24
		2	3	96	24
		7	3	240	24
		7	3	384	24
HRViT-b3		1	2	64	32
		2	2	128	32
		7	2	256	32
		7	2	512	32

Table 2. Architecture variants of HRViT. The number of Transformer blocks is marked in each module, followed by per branch settings.

Variant	Image Size	#Params (M)	GFLOPs	IMNet-1K top-1 acc.
HRViT-b1	224	19.7	2.7	80.5
HRViT-b2	224	32.5	5.1	82.3
HRViT-b3	224	37.9	5.7	82.8

Table 3. ImageNet-1K pre-training results of HRViT. FLOPs are measured on an image size of 224×224. #Params includes the classification head as used in HRNetV2 [29].

MiT and CSWin backbones. HRViT (b1-b3) outperform the previous SoTA SegFormer-MiT (B1-B3) [35] with **+3.68**, **+2.26**, and **+0.80** higher mIoU on ADE20K val, and **+3.13**, **+1.81**, **+1.46** higher mIoU on Cityscapes val.

ImageNet-1K pre-training. All HRViT variants are pre-trained on ImageNet-1K, shown in Table 3. We follow the same pre-training settings as DeiT [28] and other ViTs [12, 20, 35]. We adopt stochastic depth [17] for all HRViT variants with the max drop rate of 0.1. The drop rate is gradually increased on the deepest 3rd branch, and other shallow branches follow the rate of the 3rd branch within the same module. We use the HRNetV2 [29] classification head in HRViT on ImageNet-1K pre-training. More details can be found in Appendix A.1.

Settings. We evaluate HRViT for semantic segmentation on the Cityscapes and ADE20K datasets. We employ a lightweight SegFormer [35] head based on the mmsegmentation framework [7]. We follow the training settings of prior work [12, 35]. The training image size for ADE20K and Cityscapes are 512×512 and 1024×1024, respectively. The test image size for ADE20K and Cityscapes is set to 512×2048 and 1024×2048, respectively. We do inference on Cityscapes with sliding window test by cropping 1024×1024 patches. More details are in Appendix A.2.

Results on ADE20K. We evaluate different ViT backbones in single-scale mean intersection-over-union (mIoU), #Params, and GFLOPs. Figure 5 plots the Pareto curves in the #Params and FLOPs space. On ADE20K val, HRViT outperforms other ViTs with better performance and efficiency trade-off. For example, with the SegFormer head, HRViT-b1 outperforms MiT-B1 with 3.68% higher mIoU, 40% fewer parameters, and 8% less computation. Our HRViT-b3 achieves a higher mIoU than the best CSWin-S but saves 23% parameters and 13% FLOPs. Compared with HRNetV2+OCR, our HRViT shows considerable performance and efficiency advantages. We also evaluate HRViT with UperNet [33] head in Appendix B.2.

Results on Cityscapes. We summarize the results on Cityscapes in Table 4. Our small model HRViT-b1 outperforms MiT-B1 and CSWin-Ti by +3.13 and +2.47 higher mIoU. The key insight is that the HR architecture can increase the effective width of models with narrow channels, leading to higher modeling capacity. Hence, the parallel multi-branch topology is especially beneficial for small networks. When training HRViT-b3 on Cityscapes, we set the window sizes to 1-2-9-9. HRViT-b3 outperforms the MiT-b4 with +0.86 higher mIoU, 55.4% fewer parameters, and 30.7% fewer FLOPs. Compared with two SoTA ViT backbones, i.e., MiT and CSWin, HRViT achieves an average of +2.16 higher mIoU with 30.7% fewer parameters and 23.1% less computation.

3.2. Ablation studies

In Table 5, we first compare with a baseline where all block optimization techniques are removed. Our proposed key techniques can *synergistically* improve the ImageNet accuracy by 0.73% and the Cityscapes mIoU by +1.18 with 20% fewer parameters and 13% less computation. Then we *independently remove each technique* from HRViT to

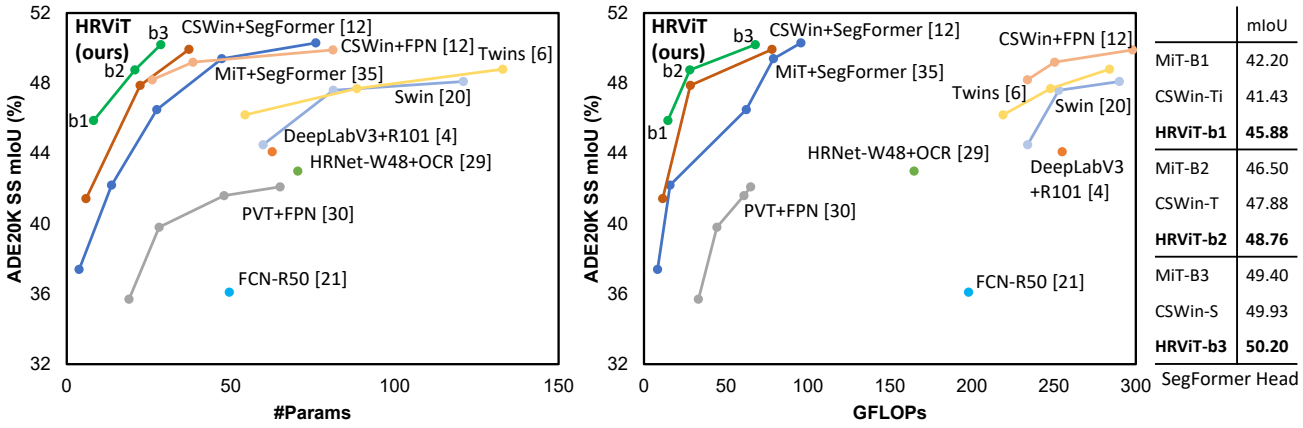


Figure 5. HRViT achieves the best performance-efficiency trade-off among all models on ADE20K val. The table on the right shows ADE20K val mIoUs of MiT, CSWin, and HRViT with the SegFormer [35] head.

Backbone	SegFormer Head [35] #Param. (M)↓	GFLOPs↓	mIoU (%)↑
MiT-B0 [35]	3.8	8.4	76.20
MiT-B1 [35]	13.7	15.9	78.50
CSWin-Ti [12]	5.9	11.4	79.16
HRViT-b1	**8.1**	**14.1**	**81.63**
MiT-B2 [35]	27.5	62.4	81.00
CSWin-T [12]	22.4	28.3	81.56
HRViT-b2	**20.8**	**27.4**	**82.81**
MiT-B3 [35]	47.3	79.0	81.70
MiT-B4 [35]	64.1	95.7	82.30
CSWin-S [12]	37.3	78.1	82.58
HRViT-b3	**28.6**	**66.8**	**83.16**
Avg improv.	-30.7%	-23.1%	+2.16

Table 4. Comparison on the Cityscapes val segmentation dataset. We reduce the channels (64→32) of CSWin-T and name it CSWin-Ti. FLOPs are based on the image size of 512×512.

Variants	#Params (M)	FLOPs (G)	IMNet top-1 acc.	City mIoU
HRViT-b1	**8.1**	**14.1**	**80.52**	**81.63**
– Key-value sharing	8.8	14.7	80.52	81.00
– Eff. patch embed	9.9	16.5	80.19	81.18
– MixCFN	7.9	13.6	79.86	80.52
– Parallel CONV path	8.1	14.0	80.06	80.82
– Nonlinearity/BN	8.1	14.1	80.37	81.12
– Dense fusion	8.0	14.0	79.95	81.26
– DES	8.1	14.0	80.36	81.38
– All block opt.	10.1	16.3	79.79	80.45

Table 5. Ablation on proposed techniques. Each entry removes one technique independently. The last one removes all techniques.

validate their individual contribution.

Sharing key-value. When removing key-value sharing, i.e., using independent keys and values, HRViT-b1 shows the same ImageNet-1K accuracy but at the cost of lower Cityscapes segmentation mIoU, 9% more parameters, and 4% more computations.

Patch embedding. Changing our EffPathEmbed to the CONV-based counterpart [35] leads to 22% more parameters and 17% more FLOPs without accuracy/mIoU benefits.

MixCFN. Replacing the MixCFN block with the original FFNs [13] directly leads to ∼0.66% ImageNet accuracy drop and 0.11 Cityscapes mIoU loss with marginal efficiency improvement. By adding multi-scale local feature extraction in feedforward networks, MixCFN can indeed boost the performance of HRViT.

Parallel convolution path. The embedded inverted residual path in the HRViTAttn block is very lightweight and contributes 0.46% higher ImageNet accuracy as well as 0.81 higher mIoU on Cityscapes.

Additional nonlinearity/BN. The extra Hardswish and BN introduce negligible overhead but boost expressivity and trainability, bringing 0.15% higher ImageNet-1K accuracy 0.51 higher mIoU on Cityscapes val.

Dense vs. sparse fusion layers. The sparse fusion layer proposed in HR-NAS [11] is not very effective in HRViT as it saves tiny hardware cost (<1%) but leads to 0.57% accuracy drop and 0.37 mIoU loss.

Diversity-enhanced shortcut. As an auxiliary path, the proposed shortcut (DES) helps enhance the feature diversity and effectively boosts the performance to a higher level both on classification and segmentation tasks. The hardware overhead is negligible due to the high efficiency of the Kronecker decomposition-based projector.

Vanilla HRNet-ViT baselines vs. HRViT. In Table 6, we directly replace residual blocks in HRNetV2 with MiT/CSWin Transformer blocks, which we refer to as a vanilla baseline. When comparing HRNet-MiT with the sequential MiT, we notice the HR variants have comparable mIoUs while significantly saving hardware cost. This shows that

Backbone	#Params (M)	FLOPs (G)	IMNet top-1 acc.	City mIoU
HRNet18-MiT	8.4	29.3	79.3	80.30
HRNet18-CSWin	8.1	22.3	79.5	80.95
HRViT-b1	**8.1**	**14.1**	**80.5**	**81.63**
HRNet32-MiT	24.4	52.4	81.5	82.05
HRNet32-CSWin	23.9	42.2	81.1	82.11
HRViT-b2	**20.8**	**27.4**	**82.3**	**82.81**
HRNet40-MiT	40.1	108.0	82.3	82.10
HRNet40-CSWin	39.5	96.3	82.4	82.38
HRViT-b3	**28.6**	**66.8**	**82.8**	**83.16**
Avg Improv.	-14.4%	-38.2%	+0.92	+0.89

Table 6. Compare vanilla HRNet-ViT baselines with HRViT on ImageNet-1K and Cityscapes val. With heterogeneous branch designs and optimized blocks, HRViT is more efficient than the vanilla HRNet-MiT and HRNet-CSWin.

window size s	7	9	11	13	15
GFLOPs	66.28	66.78	67.09	68.07	69.22
Cityscapes mIoU (%)	82.82	**83.16**	83.15	82.88	82.90

Table 7. Evaluate HRViT-b3 on Cityscapes val with different window sizes on the MR and LR paths.

the *multi-branch architecture is indeed helpful to boost the multi-scale representability*. However, the vanilla HRNet-ViT baseline overlooks the expensive cost of Transformers and is not efficient as the hardware cost quickly outweighs its performance gain. In contrast, HRViT benefits from *heterogeneous branches and optimized components with less computation, fewer parameters, and enhanced model representability* than the vanilla HRNet-ViT baselines.

Different window sizes. In Table 7, we evaluate HRViT-b3 on Cityscapes with different window sizes on the 3rd (MR) and 4th (LR) paths. In general, different window sizes give similar mIoUs, while window sizes of 7 and 9 show the best performance-efficiency trade-off. Increasing the window size from 7 to 9 helps HRViT-b3 achieve +0.34 mIoU improvement with only 0.8% more FLOPs. However, overly-large window sizes bring no performance benefits with unnecessary computation overhead. For example, further enlarging the window size from 9 to 15 causes a 0.26 mIoU drop and 3.7% more FLOPs.

4. Related Work

Multi-scale representation learning for semantic segmentation. Previous segmentation frameworks progressively down-sample the feature map to compute the LR representations [4, 13, 21], and recover the HR features via up-sampling, e.g., SegNet [1], UNet [25], Hourglass [23]. HRNet [29] maintains the HR representations throughout the network with cross-resolution fusion. Lite-HRNet [37] proposes conditional channel weighting blocks to exchange information across resolutions. HR-NAS [11] searches the channel/head settings for inverted residual blocks and the auxiliary Transformer branches. HRFormer [40] improves HRNetV2 by replacing residual blocks with Swin Transformer blocks. Different from the convolutional HRNet-family, HRViT is a pure ViT backbone with a novel *multi-branch topology* that benefits both from HR architectures and self-attentions. Distinguished from the direct CONV-to-Attention substitution in HRFormer, we explore a novel *heterogeneous branch* design and various *block optimization* techniques with higher performance and efficiency.

Multi-scale ViT backbones. Several multi-scale ViTs adopt hierarchical architectures to generate progressively down-sampled pyramid features [3, 6, 12, 14, 30, 35]. For example, PVT [30] integrates a pyramid structure into ViTs for multi-scale feature extraction. Twins [6] interleaves local and global attentions to learn multi-scale representations. SegFormer [35] proposes an efficient hierarchical encoder to extract coarse and fine features. CSWin [12] further improves the performance with multi-scale cross-shaped local attentions. However, they still follow the design concept of classification networks with a sequential topology. There is no information flow from LR to HR path inside those sequential ViTs, and the HR features are still very shallow ones of relatively low quality. In contrast, our HRViT adopts a multi-branch topology with enhanced multi-scale representability and improved efficiency. Our HRViT can serve as a ready-to-use backbone for advanced segmentation frameworks [5, 26].

5. Conclusion

In this paper, we delve into the multi-scale representation learning in ViTs and present an efficient multi-scale high-resolution ViT backbone design, named HRViT, for semantic segmentation. We enhance ViTs with a multi-branch architecture to learn high-quality HR representations via cross-scale interaction. To scale up HRViT with high efficiency, we introduce heterogeneous branch designs and jointly optimize key building blocks with efficient embedding layers, augmented cross-shaped attentions, and mixed-scale convolutional FFNs. In our evaluation, we observe that the multi-branch architecture can effectively boost the semantic segmentation performance of ViTs. Besides, we find that branch-block co-optimization is the key to improving the efficiency of HR-ViT integration. Experiments show that HRViT outperforms SoTA ViT backbones on semantic segmentation with significant performance improvement and efficiency boost. As a future direction, we look forward to evaluating HRViT on more dense prediction vision tasks, e.g., object detection, to thoroughly demonstrate the potential of HRViT as a strong vision backbone.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 2481–2495, 2017. 1, 8

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-End object detection with transformers. In *Proc. ECCV*, 2020. 1

[3] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. CrossViT: Cross-attention multi-scale vision transformer for image classification. In *Proc. ICCV*, 2021. 1, 8

[4] Liang-Chieh Chen, Yukun Zhu, George Papandreou Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proc. ECCV*, 2018. 1, 8

[5] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-Pixel Classification is Not All You Need for Semantic Segmentation. In *Proc. NeurIPS*, 2021. 8

[6] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the Design of Spatial Attention in Vision Transformers. In *Proc. NeurIPS*, 2021. 1, 4, 5, 8, 12

[7] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 6

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. CVPR*, 2016. 5, 11

[9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshop*, 2020. 11

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, , and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, 2009. 5

[11] Mingyu Ding, Xiaochen Lian, Linjie Yang, Peng Wang, Xiaojie Jin, Zhiwu Lu, and Ping Luo. HR-NAS: Searching Efficient High-Resolution Neural Architectures with Lightweight Transformers. In *Proc. CVPR*, 2021. 1, 4, 7, 8

[12] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. CSWin Transformer: A General Vision Transformer Backbone with Cross-Shaped Windows. *arXiv preprint arXiv:2107.00652*, 2021. 1, 2, 5, 6, 7, 8, 12

[13] A. Dosovitskiy, L. Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, M. Dehghani, Matthias Minderer, G. Heigold, S. Gelly, Jakob Uszkoreit, and N. Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proc. ICLR*, 2021. 1, 7, 8

[14] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale Vision Transformers. In *Proc. ICCV*, 2021. 8

[15] Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. LeViT: a Vision Transformer in ConvNet's Clothing for Faster Inference. In *Proc. ICCV*, 2021. 4

[16] Daniel Haase and Manuel Amthor. Rethinking Depthwise Separable Convolutions: How Intra-Kernel Correlations Lead to Improved MobileNets. In *Proc. CVPR*, pages 14588–14597, 2020. 4

[17] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *Proc. ECCV*, 2016. 6

[18] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. LocalViT: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021. 1

[19] Iasonas Kokkinos Liang-Chieh Chen, George Papandreou, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected CRFs. In *Proc. ICLR*, 2015. 1

[20] 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. In *Proc. ICCV*, 2021. 1, 4, 5, 6, 12

[21] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*, 2015. 1, 8

[22] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. ICLR*, 2019. 11

[23] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *Proc. ECCV*, page 483–499, 2016. 8

[24] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do Vision Transformers See Like Convolutional Neural Networks? *arXiv preprint arXiv:2108.08810*, 2021. 4

[25] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *Proc. MICCAI*, May 2015. 1, 8

[26] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for Semantic Segmentation. In *Proc. ICCV*, 2021. 8

[27] Yehui Tang, Kai Han, Chang Xu, An Xiao, Yiping Deng, Chao Xu, and Yunhe Wang. Augmented Shortcuts for Vision Transformers. In *Proc. NeurIPS*, 2021. 4

[28] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers: distillation through attention. In *Proc. ICML*, pages 10347–10357, 2021. 1, 6, 11

[29] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep High-Resolution Representation Learning for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3349–3364, 2021. 1, 4, 6, 8

[30] 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. In *Proc. ICCV*, 2021. 1, 2, 8

[31] Wenxiao Wang, Lu Yao, Long Chen, Deng Cai, Xiaofei He, and Wei Liu. CrossFormer: A Versatile Vision Transformer Based on Cross-scale Attention. *arXiv preprint arXiv:2108.00154*, 2021. 1

[32] Yuqing Wang, Zhaoliang Xu, Xinlong Wang, Chunhua Shen, Baoshan Cheng, Hao Shen, , and Huaxia Xia. End-to-end video instance segmentation with transformers. In *Proc. CVPR*, 2021. 1

[33] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proc. ECCV*, page 418–434, 2018. 6, 11, 12

[34] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early Convolutions Help Transformers See Better. In *Proc. NeurIPS*, 2021. 4

[35] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In *Proc. NeurIPS*, 2021. 1, 2, 4, 5, 6, 7, 8, 12

[36] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *Proc. ICCV*, 2021. 1

[37] Changqian Yu, Bin Xiao, Changxin Gao, Lu Yuan, Lei Zhang, Nong Sang, and Jingdong Wang. Lite-HRNet: A Lightweight High-Resolution Network. In *Proc. CVPR*, 2021. 1, 4, 8

[38] Qihang Yu, Yingda Xia, Yutong Bai, Yongyi Lu, Alan Yuille, and Wei Shen. Glance-and-Gaze Vision Transformer. *arXiv preprint arXiv:2106.02277*, 2021. 1

[39] 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. 1

[40] Yuhui Yuan, Rao Fu, Lang Huang, Weihong Lin, Chao Zhang, Xilin Chen, and Jingdong Wang. HRFormer: High-Resolution Transformer for Dense Prediction. In *Proc. NeurIPS*, 2021. 8

[41] 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 *Proc. ICCV*, 2019. 11

[42] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, , and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 11

[43] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *Proc. ICCV*, 2021. 1

[44] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proc. AAAI*, 2020. 11

[45] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proc. CVPR*, 2017. 5, 11