ARTA: Adaptive Mixed-Resolution Token Allocation for Efficient Dense Feature Extraction

David Hagerman†, Roman Naeem†, Erik Brorsson, Fredrik Kahl & Lennart Svensson
{david.hagerman, nroman, erik.brorsson, lennart.svensson, fredrik.kahl}@chalmers.se

Chalmers University of TechnologyChalmers University of technology, 412 96 Gothenburg, Sweden kontakt@chalmers.se
http://www.chalmers.se/

Abstract. We present **ARTA**, a mixed-resolution coarse-to-fine vision transformer for efficient dense feature extraction. Unlike models that begin with dense high-resolution (fine) tokens, ARTA starts with low-resolution (coarse) tokens and uses a lightweight allocator to predict which regions require more fine tokens. The allocator iteratively predicts a semantic (class) boundary score and allocates additional tokens to patches above a low threshold, concentrating token density near boundaries while maintaining high sensitivity to weak boundary evidence. This targeted allocation encourages tokens to represent a single semantic class rather than a mixture of classes. Mixed-resolution attention enables interaction between coarse and fine tokens, focusing computation on semantically complex areas while avoiding redundant processing in homogeneous regions. Experiments demonstrate that ARTA achieves state-of-the-art results on ADE20K and COCO-Stuff with substantially fewer FLOPs, and delivers competitive performance on Cityscapes at markedly lower compute. For example, ARTA-Base attains **54.6 mIoU** on ADE20K in the ∼100M-parameter class while using fewer FLOPs and less memory than comparable backbones.

Keywords: Segmentation · Adaptive token allocation · Mixed-resolution

[1]

1 Introduction

In semantic segmentation tasks, not all pixels are equally important. In a typical scene, large regions are often homogeneous (e.g., sky, road, or walls) and exhibit little spatial or semantic variation, making dense per-pixel computation unnecessary. In contrast, regions with high object density or class boundaries require higher spatial resolution for accurate predictions. This uneven distribution of

[1] † These authors contributed equally to this work.

semantic content motivates architectures that adaptively allocate higher spatial resolution to more informative regions of the image.

Most computer vision architectures process images using a uniform grid of square patches, assigning equal computational cost to each patch regardless of its semantic content. For example, in Vision Transformers [8], it is common to divide the input image into fixed-size patches (e.g., 16×16), each of which is embedded into a token, applying the same representation capacity to both simple and complex patches. Convolutional neural networks such as ConvNext [27] similarly apply filters over regularly spaced patches, implicitly treating all areas of the image with equal importance. While some recent models [19, 25, 35] introduce mechanisms for content-aware processing or adaptive computation, the majority of widely used architectures still rely on uniform spatial partitioning, which does not align with the uneven distribution of information in real-world images.

While uniform spatial partitioning is inefficient, most non-uniform approaches still follow a high-to-low resolution processing pipeline. In these architectures, the input image is initially represented at high resolution, and information is gradually compressed through pooling, striding, or token merging. As a result, all image patches, regardless of their semantic importance, are initially processed at the finest spatial granularity. This leads to unnecessary computation in homogeneous or uninformative patches and limits the potential efficiency gains of token pruning or merging in later stages. An ideal approach would avoid assigning high-resolution capacity to patches that do not require it in the first place, allocating computational resources only where semantic complexity demands it.

Beyond where computation is spent, there is also the question of how representational capacity is used within each token. When a token aggregates pixels from multiple semantic classes, its feature vector must encode the class mixture (which classes are present and in what proportions), the spatial layout and extent of each class within the patch, and the patch's overall position. By contrast, if a token predominantly corresponds to a single class, its representation can devote more dimensions to modeling intra-class variation and higher-order semantics rather than resolving class boundaries. Constraining tokens to be largely class-specific improves the efficiency of the representation, allowing models with modest width to rival the semantic expressiveness of wider baselines.

To address these computational and representational inefficiencies, we propose **ARTA** (**A**daptive Mixed-**R**esolution **T**oken **A**llocation), a two-stage encoder. In Stage 1, a lightweight allocator adaptively assigns token density across the image through three hierarchical allocation rounds. Starting from coarse tokens, it predicts a semantic class-boundary score for each token and allocates additional finer-grained tokens to patches whose scores exceed a low threshold, increasing resolution only where boundary evidence is present. In subsequent rounds, scoring is re-applied only to the newly allocated finest-resolution tokens, reducing allocator compute by avoiding repeated dense evaluation over the full token set. Repeating this process yields a mixed-resolution token set that concentrates spatial detail near class boundaries and avoids unnecessary allocation in uniform regions, ensuring that no patch is processed at higher resolution than

needed. In Stage 2, we refine the resulting mixed-resolution tokens with a deeper hierarchical encoder to capture higher-level semantics. To summarize, our main contributions are:

- A lightweight, class-boundary scoring allocator for adaptive mixed resolution token allocation. Starting from coarse tokens, it iteratively re-scores and allocates finer-grained tokens to patches containing class-boundaries, increasing token density in semantically rich regions while leaving uniform areas at coarse granularity.
- **ARTA:** A two-stage encoder that couples the proposed mixed-resolution token allocation with a deeper hierarchical refinement network. Mixed resolution attention enables interaction between coarse and fine tokens, concentrating computation on semantically complex patches.
- We validate ARTA on ADE20K, COCO-Stuff, and Cityscapes, achieving state-of-the-art results on ADE20K and COCO-Stuff with substantially fewer FLOPs, and competitive performance on Cityscapes at lower compute.

2 Related Work

2.1 Token dropping and merging

Many methods improve transformer efficiency by removing tokens deemed uninformative or by aggregating nearby/similar tokens [1, 3, 12, 17, 20, 22, 23, 26]. These fine-to-coarse strategies typically start from a dense high-resolution grid of tokens and progressively reduce the number of tokens by pruning or merging similar tokens to gain efficiency. This yields a content-adaptive token density, but only through token reduction. The spatial resolution never exceeds that of the initial grid, and training still processes an initially dense token set. In contrast, ARTA follows a coarse-to-fine strategy that starts from coarse tokens and allocates additional tokens only where finer detail is needed, increasing resolution on demand rather than uniformly.

2.2 Adaptive downsampling

A complementary direction makes tokenization or routing content-aware, for example via saliency-driven multi-resolution grids, attention-guided sampling, or per-input policy decisions [9, 21, 24]. For dense prediction, AutoFocusFormer (AFF) [35] proposes point-based local attention with balanced clustering and a learnable neighborhood-merging module to support segmentation heads. Other works merge/share or prune tokens based on semantics or difficulty [19,25]. While these approaches introduce adaptivity, most still start from a dense tokenization and then select, share, or prune tokens thereafter. As a result, they generally do not increase spatial token density on demand, and training compute remains dominated by the initial high-resolution token set.

2.3 Learned upsampling operators

Another line of work studies learned upsampling operators that incorporate local context. Dynamic upsampling operators such as DySample [14] and frequency-aware fusion modules such as FreqFusion [4] operate on dense feature maps. DySample learns sampling locations for each output position, while FreqFusion applies adaptive low-/high-pass filtering and offsets to boundary sharpness during upsampling. These approaches focus on how to upsample by improving the upsampling operator itself, and they typically upsample densely across spatial locations. In contrast, ARTA addresses what to upsample by starting from a coarse tokenization and allocating additional tokens only when required, producing a mixed-resolution representation that keeps homogeneous patches compact. Therefore, learned upsampling operators are orthogonal to ARTA.

2.4 Coarse-to-fine architectures

Several coarse-to-fine architectures follow an overview-to-detail schedule, redistributing capacity toward coarse scales [18, 34]. OverLoCK [18] begins with a coarse global context derived from low-resolution features, and later refines the representation using fine-grained attention. These architectures alter how capacity is distributed across scales but generally keep resolution schedules fixed and input-agnostic rather than content-driven within an image.

Beyond encoder design, boundary-refinement post-processing methods such as SegFix [32] improve boundary quality by redirecting boundary pixels toward more reliable interior predictions using learned offsets. In contrast, ARTA incorporates boundary awareness directly into the encoder through class-boundary scoring and hierarchical token allocation. SegFix operates as a post-processing step on model outputs, whereas ARTA controls where spatial resolution is allocated during feature extraction.

Overall, most prior work either reduces token count after a dense beginning, improves upsampling or fusion on dense feature maps, or redistributes capacity with fixed multi-scale plans. A gap remains for architectures that start from coarse tokens and adaptively allocate higher token density only where semantic complexity warrants it, while maintaining mixed-resolution representations that interact across scales during dense feature extraction.

3 Method

We present **ARTA**, a two-stage hierarchical encoder that constructs mixed-resolution feature representations by adaptively allocating token density to semantically complex image patches. ARTA is designed to encourage tokens to correspond to a single semantic class, rather than mixing multiple classes within the same token. This is achieved by identifying tokens likely to overlap semantic (class) boundaries and selectively allocating additional tokens to only those patches, while keeping homogeneous patches compact.

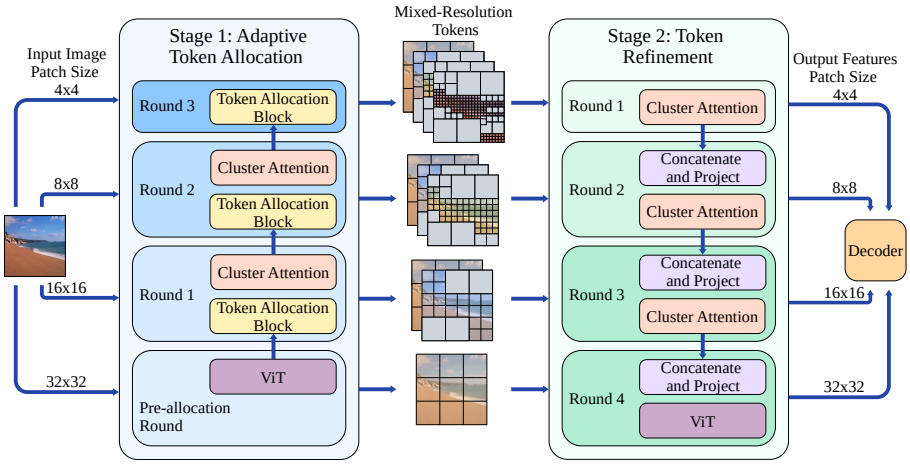

Fig. 1: ARTA overview. ARTA has two stages: Adaptive Token Allocation (Stage 1) and Token Refinement (Stage 2). Stage 1 proceeds bottom-up from coarse 32×32 tokens. A pre-allocation ViT produces boundary-aware features, after which each allocation round applies a token allocation block (rounds 1–2 also use cluster attention). The allocation block scores the finest tokens and allocates finer-grained tokens to the corresponding patches containing class-boundaries (Figure 2), progressively building mixed-resolution sets up to $[32\times32, 16\times16, 8\times8, 4\times4]$. Stage 2 proceeds top-down from the final set and refines tokens using cluster attention, with a ViT in the final round. At each refinement round, the finest tokens are output, and the remaining tokens continue. Lateral Stage 1 outputs are fused into Stage 2 by concatenation and projection before attention. The decoder uses these multi-scale features for dense prediction.

ARTA differs from token-pruning or token-merging approaches that begin with a dense high-resolution grid and later reduce token count. Instead, ARTA starts from coarse tokens and increases spatial resolution only where needed. This avoids ever allocating high-resolution tokens to semantically simple patches and enables a principled and efficient use of computation.

3.1 Overview

ARTA follows a two-stage design. In Stage 1, an adaptive mixed-resolution token allocator iteratively predicts a class-boundary score for tokens and hierarchically allocates additional high-resolution tokens to the image patches containing class-boundaries over multiple allocation rounds, producing a mixed-resolution token set. In Stage 2, a deeper hierarchical encoder refines token features over multiple refinement rounds to capture higher-level semantics using mixed-resolution attention, allowing coarse and fine tokens to interact while concentrating compute on semantically rich patches. The overall architecture is illustrated in Figure 1 and configurations details can be found in Table 1.

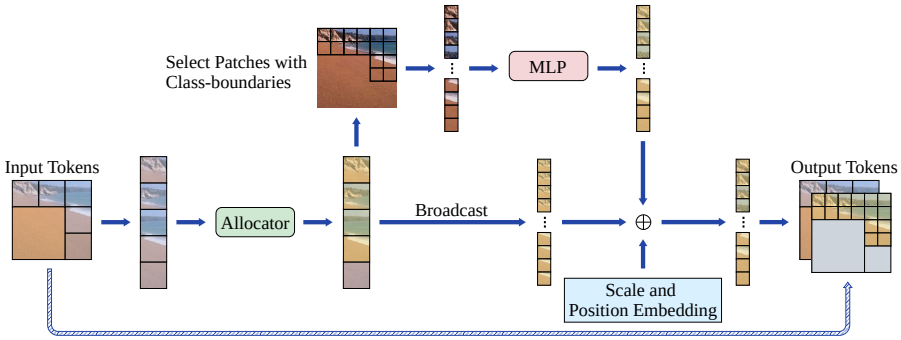

Fig. 2: Token Allocation Block. The allocator scores the current finest-resolution tokens and selects the corresponding patches containing class-boundaries. Selected patches are split into 2×2 sub-patches to allocate finer tokens. Each new token is initialized from sub-patch image features (MLP) and combined with a broadcast residual from the parent token. Finally, scale and position embeddings are added to get output tokens.

3.2 Adaptive Mixed-Resolution Token Allocation

Coarse initialization. Given an input image, ARTA begins with a coarse patch embedding using a 32×32 kernel to form an initial low-resolution token grid. During a pre-allocation round, these tokens are processed by a lightweight Vision Transformer to produce boundary-aware coarse features, enabling reliable class-boundary scoring during the allocation rounds.

Hierarchical allocation rounds. Stage 1 allocates tokens through R allocation rounds (we use $R = 3$). At allocation round r, the allocator predicts a class-boundary score $c_i^{(r)}$ for the candidate patch represented by token i at the current finest resolution under consideration. Patches with scores above a low threshold are selected for finer allocation, and additional tokens are introduced by partitioning each selected region into finer sub-patches (in our implementation, a 2×2 partition yielding four tokens per selected patch).

The resulting mixed-resolution token set is then processed by a cluster attention block from AFF [35], which computes attention over localized clusters that are recomputed each forward pass under the current allocation. While AFF adaptively downsamples by merging less important tokens into important ones, yielding non-overlapping tokens, ARTA can contain overlapping tokens at multiple resolutions covering the same region. We use the same clustering mechanism to enable efficient cross-resolution interaction, so that fine tokens can attend to nearby coarse tokens for high-level context, while coarse tokens absorb local detail carried by fine tokens that inject newly extracted sub-patch features.

At the next round, the allocator is re-applied only to the finest-resolution tokens produced by the previous allocation round, rather than re-scoring the entire token set, reducing allocation overhead by avoiding repeated dense evaluation.

Mixed-resolution token set. After R rounds, Stage 1 produces a mixed-resolution representation consisting of coarse tokens for homogeneous regions and finer tokens concentrated around predicted semantic boundaries. This ensures that no region is represented at a higher resolution than necessary.

3.3 Token Allocation Block

The token allocation block assigns a scalar class-boundary score to each token at the finest resolution in allocation round r. For an input token i, the allocator outputs a score $c_i^{(r)}$ estimating the amount of semantic (class) boundary content within the token patch.

Class-boundary score. The allocator is trained to predict a class-boundary score for each token, indicating how important it is to allocate additional tokens to the corresponding image patch. Ground-truth scores are derived from the segmentation labels by first computing a binary map of class-boundary pixels, where a pixel is marked as a boundary (1) if any of its neighboring pixels has a different class label, and as non-boundary (0) otherwise. For each token patch, we sum the boundary pixels within the patch and normalize the result to obtain the target score. The allocator MLP is trained to regress this target using mean squared error (MSE).

This scoring choice encourages each token to represent a single semantic class, simplifying downstream reasoning. A token that spans multiple classes must encode both class identities and their spatial arrangement, increasing representational burden. In contrast, single-class tokens allow the encoder to focus on semantic meaning rather than intra-token class composition.

Threshold-based allocation. We allocate additional tokens only for regions whose predicted scores exceed a round-specific threshold τ_r. This induces a variable token budget per sample. Let N_r denote the number of candidate tokens for a given sample at round r, and let

$$K_r = \sum_{i=1}^{N_r} \mathbf{1}\Big(c_i^{(r)} > \tau_r\Big) \tag{1}$$

denote the number of selected tokens. The threshold τ_r is set slightly above zero to tolerate small prediction noise and to ensure that only regions with sufficient predicted boundary complexity are selected for further allocation.

Because K_r varies across samples, token counts differ within a mini-batch. During training, we pad each sample's token set (per allocation round) to the maximum token count in the batch and mask the padded tokens so they do not participate in attention or loss computation. This enables mini-batch training while ensuring each sample uses only the number of tokens required by its content.

Table 1: Encoder architecture hyperparameters for ARTA-Tiny, ARTA-Small, and ARTA-Base. For each round, we report the token embedding dimensions and the number of attention blocks. Entries are listed in execution order, following ARTA's coarse-to-fine-to-coarse layout.

Encoder	Stage 1 Patch Size: [32^2, 16^2, 8^2, 4^2]		Stage 2 Patch Size: [4^2, 8^2, 16^2, 32^2]	
	Dims	Blocks	Dims	Blocks
ARTA-Tiny	[512, 256, 128, 64]	[1, 1, 1, 0]	[64, 128, 256, 512]	[4, 4, 16, 4]
ARTA-Small	[512, 256, 128, 64]	[2, 2, 2, 0]	[64, 128, 256, 512]	[4, 6, 24, 3]
ARTA-Base	[768, 384, 192, 96]	[2, 2, 2, 0]	[96, 192, 384, 768]	[8, 6, 18, 4]

Token allocation. For each selected token, the corresponding image patch is partitioned into 2×2 sub-patches, and allocate four finer-grained tokens, one per sub-patch. To initialize a new token, we extract the raw pixel values of its sub-patch, flatten them into a vector, and project this vector to the token embedding dimension using a learned linear layer. The projected vector is then passed through an MLP to produce an image feature. To reuse the selected token's boundary-aware coarse representation, we add it as a residual to each new token. Each token further receives (i) a learned scale embedding and (ii) a learned sub-patch position embedding to disambiguate within-patch spatial identity.

3.4 Mixed-Resolution Token Refinement

Stage 2 takes the mixed-resolution token set produced by the final allocation round in Stage 1 and refines token features in a top-down manner to capture higher-level semantics. Stage 2 is organized into multiple refinement rounds. In each round, we apply cluster attention [35] over localized token clusters, which is computationally efficient for sparse mixed-resolution token sets and enables cross-resolution interactions. Fine tokens gain broader semantic context by attending to nearby coarse tokens, while coarse tokens are enriched with local structure information carried by fine tokens.

Across refinement rounds, Stage 2 progressively outputs the finest-resolution tokens to form multi-scale features for dense prediction. Starting from mixed-resolution tokens at all four scales, each refinement round outputs the current finest scale (e.g., 4×4, then 8×8, then 16×16, then 32×32), while the remaining coarser tokens are passed to the next round. The final refinement round uses a ViT block to strengthen high-level semantic representations at the coarsest scale.

Cross-stage residual connections. To preserve low-level and boundary-aware information from the allocation stage, Stage 2 incorporates lateral connections

from Stage 1. Before the attention block in each refinement round, we fuse the current Stage 2 tokens with a Stage 1 lateral output at the same resolution via concatenation followed by linear projection. Concretely, refinement rounds 2–4 use lateral outputs from allocation round 2, allocation round 1, and the pre-allocation round, respectively. This pre-attention fusion injects Stage 1 features into Stage 2 refinement while maintaining the mixed-resolution structure. The output finest-scale tokens from each refinement round are forwarded to the decoder as multi-scale features.

3.5 Decoder strategy

We use the point-based Mask2Former decoder [6] as in AFF [35]. A pixel decoder first processes and aligns features across scales, followed by masked cross-attention that iteratively refines a set of class queries using multi-scale token features.

Before masked cross-attention, we densify only the highest-resolution feature map. At each spatial location, we select the finest token available, replicate its feature over the corresponding region, and add learned positional embeddings for spatial differentiation. Lower-resolution feature maps remain sparse. This preserves sparse computation in early decoding while enabling dense, fine-grained updates in the final masked attention stage.

4 Experiments

4.1 Datasets

We evaluate our method on three publicly available semantic segmentation benchmarks: ADE20K [33], Cityscapes [7], and COCO-Stuff [2,13]. ADE20K is a scene parsing dataset containing 20,210 images annotated with 150 fine-grained semantic classes, covering a diverse range of indoor and outdoor environments. Cityscapes is a street-level driving dataset consisting of 5,000 high-resolution images with fine annotations for 19 semantic categories. COCO-Stuff extends the COCO dataset with dense pixel-level annotations, providing 172 semantic labels across 164,000 images.

4.2 Experimental Setup

We build on Detectron2 with components from AFF and Mask2Former, and train on NVIDIA A100 GPUs. For complete details of the experimental protocol and additional settings, see the supplementary material.

For pre-training, we use ImageNet classification. During this stage the token allocation blocks are disabled and tokens are allocated at random according to a fixed ratio schedule (no content-based selection). ImageNet classification results can be found in the supplement.

For fine-tuning, we follow AutoFocusFormer and Mask2Former for data augmentation and general hyperparameters. Models were optimized using AdamW

Table 2: Comparisons with state-of-the-art methods on ADE20K val.

Model	Params	FLOPs ↓	mIoU (SS) ↑	mIoU (MS) ↑
ViT-CoMer-T [29]	38.7M	-	43.0	44.3
SegFormer-B3 [30]	47.3M	79G	49.4	50.0
SegNeXt-L [11]	48.9M	70G	51.0	52.1
SegMAN-B [10]	51.8M	58G	**52.6**	-
Mask2Former-Swin-T [6]	46.5M	74G	47.7	49.6
AFF-Tiny-1/5 [35]	46.5M	51G	50.0	-
ARTA-Tiny	48.5M	**44±7G**	51.5	**52.6**
ViT-CoMer-S [29]	61.4M	-	46.5	47.7
VMamba-T [15]	62.0M	-	47.9	48.8
ViT-Adapter-S [5]	57.6M	-	46.2	47.1
OverLoCK-Tiny [18]	63.0M	-	50.3	-
HRFormer-B [31]	56.2M	-	48.7	50.0
SegFormer-B4 [30]	64.1M	96G	50.3	51.1
LRFormer-B [28]	69M	75G	51.0	-
Mask2Former-Swin-S [6]	66.5M	98G	51.3	52.4
AFF-Small-1/5 [35]	62.1M	67G	51.9	-
ARTA-Small	61.7M	**60.7±9.3G**	**52.1**	**53.9**
ViT-CoMer-B [29]	144.7M	-	48.8	49.4
ViT-Adapter-B [5]	133.9M	-	48.8	49.7
VMamba-B [15]	122.0M	-	51.0	51.6
ConvNeXt V2-B [27]	122.0M	-	52.1	-
OverLoCK-Base [18]	124M	-	51.7	-
SegFormer-B5 [30]	84.7M	183G	51.0	51.8
LRFormer-L [28]	113M	183G	52.6	-
SegMAN-L [10]	92.4M	97G	53.2	-
Mask2Former-Swin-B [6]	106.5M	222.7G	52.4	53.7
ARTA-Base	111.5M	**82.4±14.0G**	**53.5**	**54.6**

with a learning rate of 4×10^{-5}. Crop sizes are fixed for all reported methods: 512×512 for ADE20K and COCO-Stuff, and 1024×1024 for Cityscapes. Training was conducted for 80k iterations with a batch-size of 32 on ADE20K and COCO-Stuff, and for 90k iterations on Cityscapes with a batch-size of 16. The threshold τ_r is set to [0.005, 0.01, 0.02] for the three allocation rounds.

For evaluation on ADE20K and COCO-Stuff, we resize the short side to 512 with preserved aspect ratio; for Cityscapes, we use overlapping 1024×1024 sliding-window inference. Performance is reported using mean Intersection over Union (mIoU)

FLOPs are measured for the full network at fixed input sizes: 512×512 (ADE20K/COCO-Stuff) and 1024×2048 (Cityscapes), reported as mean ± std per image over the validation set. We compute FLOPs for **ARTA** using this protocol, while FLOPs for other methods are taken from their respective papers.

For fairness, we report FLOPs only for methods evaluated at the same input resolution.

Table 3: Comparisons with state-of-the-art methods on the validation sets of COCO-Stuff and Cityscapes. "†" indicates that ImageNet22k was used for pre-training.

Model	Params	COCO-Stuff		Cityscapes	
		FLOPs ↓	mIoU ↑ (SS/MS)	FLOPs ↓	mIoU ↑ (SS/MS)
SegFormer-B3 [30]	47.3M	79G	45.5 / -	963G	81.7 / 83.3
SegNext-L [11]	48.9M	70G	46.5 / 47.2	578G	83.2 / **83.9**
SegMAN-B [10]	51.8M	58G	**48.4**/ -	479G	**83.8** / -
Mask2Former-Swin-T [6]	46.5M	-	- / -	537G	82.1 / 83.0
ARTA-Tiny	49.5M	**45±9G**	47.2 / 47.7	**286±20G**	81.8 / 82.9
HRFormer-B [31]	56.2M	280G	42.4 / 43.3	2224G	81.9 / 82.6
SegFormer-B4 [30]	64.1M	96G	46.5 / -	1241G	82.3 / **83.9**
LRFormer-B [28]	67M	75G	47.2 / -	555G	**83.0** / -
Mask2Former-Swin-S [6]	66.5M	-	- / -	732G	82.6 / 83.6
ARTA-Small	61.7M	**57±12G**	**47.6** / **48.3**	**355±25G**	81.9 / 83.2
SegFormer-B5 [30]	84.7M	112G	46.7 / -	1460G	82.4 / 84.0
LRFormer-L [28]	111.0M	122G	47.9 / -	908G	83.2 / -
SegMAN-L [10]	92.4M	97G	48.8 / -	796G	**84.2** / -
Mask2Former-Swin-B† [6]	106.5M	-	- / -	1050G	83.3 / **84.5**
ARTA-Base	111.5M	**87±19G**	**49.0** / **49.4**	**590±40G**	82.6 / 83.3

ADE20K. We compare **ARTA** at three model scales against state-of-the-art baselines on ADE20K val (Table 2). At the smallest scale, SegMAN-B attains higher single-scale mIoU than ARTA-Tiny, but ARTA-Tiny is more compute-efficient. As model capacity increases, ARTA scales more favorably and at the large scale ARTA-Base surpasses SegMAN-L while using fewer FLOPs. When comparing to other encoders using a Mask2Former decoder such as Mask2Former [6] with Swin backbones and AFF [35], we can observe that ARTA has higher mIoU and lower FLOPs over all model sizes. Across all ARTA variants, multi-scale testing further boosts performance, with ARTA-Base reaching **54.6** mIoU.

COCO-Stuff. Table 3 mirrors the trend observed on ADE20K: SegMAN is strongest at the smallest scale, whereas ARTA improves more with capacity. In the ∼50M-parameter regime, SegMAN-B achieves higher single-scale mIoU than ARTA-Tiny, but ARTA-Tiny is notably more compute-efficient. As model size increases, ARTA gains substantially more mIoU from Tiny to Base than SegMAN does from B to L. At the large scale, **ARTA-Base** surpasses SegMAN-

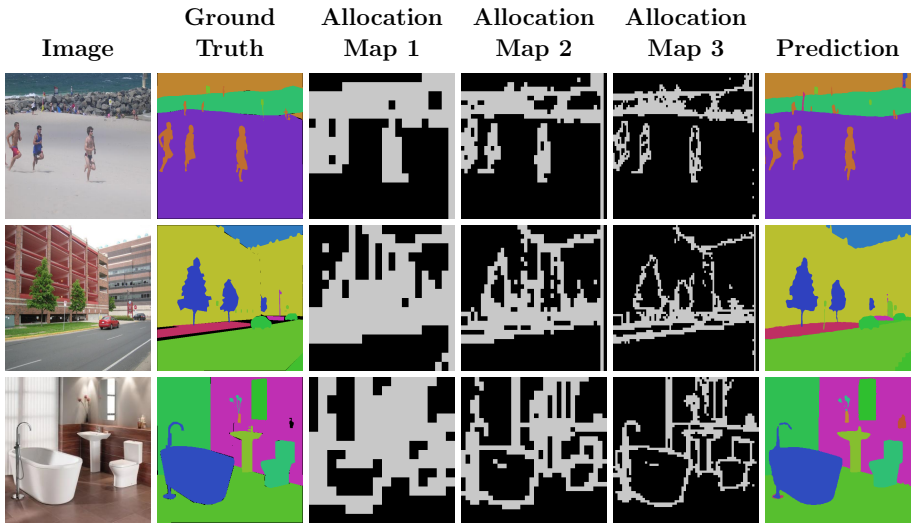

Fig. 3: From left to right: original image, ground truth, 32×32 patches selected for allocation, 16×16 patches selected for allocation, 8×8 patches selected for allocation, and prediction. Black in the ground truth means that the pixel was not labeled.

L in single-scale mIoU while using fewer FLOPs on average, and further improves with multi-scale testing (Table 3).

Cityscapes. On Cityscapes, ARTA delivers competitive accuracy while using a fraction of the compute of comparable baselines (Table 3). Across model scales, ARTA attains mIoU close to state of the art, but with substantially lower FLOPs (e.g., **ARTA-Tiny** at 286 GFLOPs vs. 479–963 GFLOPs for similar-size baselines), providing a favorable accuracy–efficiency trade-off.

Speed Analysis. We benchmark inference speed and memory usage on Cityscapes with a single NVIDIA A100 GPU. Following SegMAN [10], we report FPS averaged over 128 inference steps with batch size 2, using full-resolution inputs (1024×2048) in FP32. We report peak allocated GPU memory over the same run using PyTorch's CUDA memory statistics. As shown in Table 4, ARTA's reduced FLOPs translate to higher throughput and lower memory usage across model scales.

Qualitative Examples. Figure 3 shows qualitative results and the tokens selected for refinement at each scale. The model keeps homogeneous regions (e.g., sky, road, walls) coarse while allocating finer tokens near boundaries and small/fine structures, aligning with the goal of spending resolution where semantic complexity is high.

Table 4: FPS and peak allocated memory during inference on Cityscapes with a single NVIDIA A100.

Model	Params	FLOPs	FPS	Peak mem (GB)
SegFormer-B3 [30]	47.3M	963G	6.6	21.9
SegNeXt-L [11]	48.8M	554G	**7.3**	16.9
SegMAN-B [10]	51.8M	479G	5.6	23.8
AFF-Tiny [35]	46.5M	463G	3.8	34.4
ARTA-Tiny	49.5M	**286±20G**	6.6	**14.6**
SegFormer-B4 [30]	64.1M	1241G	4.7	29.5
AFF-Small [35]	62.1M	639G	3.6	38.7
ARTA-Small	61.7M	**355±25G**	**6.0**	**19.8**
SegFormer-B5 [30]	84.7M	1460G	4.0	34.9
SegMAN-L [10]	92.4M	796G	3.9	31.6
ARTA-Base	111.5M	**590±40G**	**4.9**	**25.4**

4.3 Ablation Studies

Without adaptive token allocation We ablate adaptive token allocation by forcing dense tokenization: during ImageNet pre-training, every selected coarse token is always allocated finer tokens, producing dense token grids at all scales. We then fine-tune this ARTA-Tiny variant on ADE20K and compare it to the adaptive baseline under the same training setup. Removing adaptive allocation reduces accuracy and increases compute, achieving 50.4 mIoU at 74G FLOPs versus 51.5 mIoU at 44±7G for the baseline.

Choice of encoder blocks. We ablate the block type used at different locations in ARTA-Tiny, trained from scratch on ADE20K. The pre-allocation and final refinement rounds operate on coarse and dense token grids, where standard dense backbones are applicable. In contrast, all other rounds process sparse mixed-resolution token sets with substantially higher token counts, requiring layers that can handle multi-scale sparse inputs while keeping compute low. A vanilla ViT supports sparse inputs in principle, but becomes infeasible in the intermediate rounds due to its quadratic attention cost. We therefore consider two compatible alternatives that support sparse mixed-resolution processing with compute constraints: cluster attention and (point-based) multi-scale deformable attention (MSDeformAttn) [35]. Table 5 shows that ViT is best for the coarse dense initial/final rounds, while cluster attention performs best for intermediate mixed-resolution rounds.

Ablation: Oracle Token Allocation Scores. We test whether injecting ground-truth class-boundary scores during training improves segmentation. Models are fine-tuned on ADE20K; the allocation blocks are always trained and, at inference, allocation uses predicted scores. An oracle rate of $x\%$ means we randomly use oracle scores for $x\%$ of training batches (and predicted scores otherwise). Results on validation set are shown in Table 6.

Table 5: Block-type ablation for ARTA-Tiny on ADE20K. "OOM" indicates out of memory.

Initial/Final block type	Intermediate block type	mIoU
Initial/Final Block Ablation		
Cluster Attention [35]	Cluster Attention [35]	42.1
ConvNeXt V2 [27]		41.8
Swin [16]		42.2
ViT [8]		**42.4**
Intermediate Block Ablation		
ViT [8]	ViT [8]	OOM
	MSDeformAttn [35]	42.0
	Cluster Attention [35]	**42.4**

Table 6: Oracle token allocation scores on ADE20K with ARTA-Tiny.

Oracle rate	mIoU ↑
100%	35.3
50%	48.9
10%	50.8
0%	**51.5**

Table 7: Ablation of Stage 2 and Stage 1 token initialization for ARTA-Tiny on ADE20K.

Method	mIoU ↑
Baseline (Stage 1 + Stage 2)	**42.4**
Stage 1 only	41.6
w/o aux image init	41.0
w/o feature residual	41.3

Oracle scores do not improve validation mIoU and higher oracle rates degrade performance. While training losses (Dice/mask) decrease when using oracle scores, we hypothesize this reflects a train–test mismatch: oracle scores may act as a privileged boundary cue during training, encouraging the model to rely on boundary information that is not available at test time. When allocation reverts to predicted scores at inference, this reliance may not transfer and performance drops.

Ablation: Stage design and token initialization. We ablate Stage 2 mixed-resolution token refinement and two token initialization choices in Stage 1 allocation, training all variants from scratch on ADE20K for 80k iterations. The baseline uses Stage 1 allocation followed by Stage 2 refinement. *Stage 1 only* removes Stage 2, reallocates depth to Stage 1 to match parameters, and adds a final cluster-attention layer (as in Stage 2) to refine newly allocated high-resolution tokens. *No aux image data* disables adding sub-patch image features when initializing newly allocated tokens. *No feature residual* removes the residual copy of the selected token feature, initializing new tokens solely from sub-patch image features (plus embeddings/MLP). As shown in Table 7, removing Stage 2 or either initialization component reduces mIoU.

5 Conclusion

We presented ARTA, a coarse-to-fine segmentation backbone that predicts semantic boundary density early and allocates additional tokens to regions needing higher spatial detail. Starting from coarse tokens and allocating finer tokens adaptively, ARTA preserves compute by avoiding fine processing in homogeneous areas while maintaining interacting mixed-resolution features across scales. ARTA achieves state-of-the-art accuracy on ADE20K and COCO-Stuff with substantially fewer FLOPs, and remains competitive on Cityscapes at a fraction of the compute of comparable baselines. These results show that content-adaptive resolution is a strong approach for accurate, efficient dense prediction. Scalability to much larger backbones and longer pre-training remains untested. Future work includes extending ARTA to 3D segmentation.

References

1. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token Merging: Your ViT But Faster. In: The Eleventh International Conference on Learning Representations (Sep 2022), https://openreview.net/forum?id=JroZRaRw7Eu 3
2. Caesar, H., Uijlings, J., Ferrari, V.: COCO-Stuff: Thing and Stuff Classes in Context. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1209–1218 (2018), https://openaccess.thecvf.com/content_cvpr_2018/html/Caesar_COCO-Stuff_Thing_and_CVPR_2018_paper.html 9
3. Chang, S., Wang, P., Lin, M., Wang, F., Zhang, D.J., Jin, R., Shou, M.Z.: Making Vision Transformers Efficient from A Token Sparsification View. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6195–6205 (Jun 2023). https://doi.org/10.1109/CVPR52729.2023.00600, https://ieeexplore.ieee.org/document/10204437 3
4. Chen, L., Fu, Y., Gu, L., Yan, C., Harada, T., Huang, G.: Frequency-Aware Feature Fusion for Dense Image Prediction. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(12), 10763–10780 (Dec 2024). https://doi.org/10.1109/TPAMI.2024.3449959, https://ieeexplore.ieee.org/document/10648934 4
5. Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., Qiao, Y.: Vision Transformer Adapter for Dense Predictions. In: The Eleventh International Conference on Learning Representations (Sep 2022), https://openreview.net/forum?id=plKu2GByCNW 10
6. Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R.: Masked-Attention Mask Transformer for Universal Image Segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1290–1299 (2022), https://openaccess.thecvf.com/content/CVPR2022/html/Cheng_Masked-Attention_Mask_Transformer_for_Universal_Image_Segmentation_CVPR_2022_paper.html 9, 10, 11
7. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The Cityscapes Dataset for Semantic Urban Scene Understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3213–3223 (2016), https://openaccess.thecvf.com/content_cvpr_2016/html/Cordts_The_Cityscapes_Dataset_CVPR_2016_paper.html 9

8. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In: International Conference on Learning Representations (Oct 2020), https://openreview.net/forum?id=YicbFdNTTy 2, 14
9. Fayyaz, M., Koohpayegani, S.A., Jafari, F.R., Sengupta, S., Joze, H.R.V., Sommerlade, E., Pirsiavash, H., Gall, J.: Adaptive Token Sampling for Efficient Vision Transformers. In: Avidan, S., Brostow, G., Cissé, M., Farinella, G.M., Hassner, T. (eds.) Computer Vision – ECCV 2022. pp. 396–414. Springer Nature Switzerland, Cham (2022). https://doi.org/10.1007/978-3-031-20083-0_24 3
10. Fu, Y., Lou, M., Yu, Y.: SegMAN: Omni-scale Context Modeling with State Space Models and Local Attention for Semantic Segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19077–19087 (2025), https://openaccess.thecvf.com/content/CVPR2025/html/Fu_SegMAN_Omni-scale_Context_Modeling_with_State_Space_Models_and_Local_CVPR_2025_paper.html 10, 11, 12, 13
11. Guo, M.H., Lu, C.Z., Hou, Q., Liu, Z.N., Cheng, M.M., Hu, S.m.: SegNeXt: Rethinking Convolutional Attention Design for Semantic Segmentation. In: Advances in Neural Information Processing Systems (Oct 2022), https://openreview.net/forum?id=VgOw1pUPh97 10, 11, 13
12. Kim, M., Gao, S., Hsu, Y.C., Shen, Y., Jin, H.: Token Fusion: Bridging the Gap Between Token Pruning and Token Merging. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 1383–1392 (2024), https://openaccess.thecvf.com/content/WACV2024/html/Kim_Token_Fusion_Bridging_the_Gap_Between_Token_Pruning_and_Token_WACV_2024_paper.html 3
13. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common Objects in Context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 740–755. Springer International Publishing, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48 9
14. Liu, W., Lu, H., Fu, H., Cao, Z.: Learning to Upsample by Learning to Sample. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 6027–6037 (2023), https://openaccess.thecvf.com/content/ICCV2023/html/Liu_Learning_to_Upsample_by_Learning_to_Sample_ICCV_2023_paper.html 4
15. Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., Ye, Q., Jiao, J., Liu, Y.: VMamba: Visual State Space Model. Advances in Neural Information Processing Systems **37**, 103031–103063 (Dec 2024), https://proceedings.neurips.cc/paper_files/paper/2024/hash/baa2da9ae4bfed26520bb61d259a3653-Abstract-Conference.html 10
16. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 9992–10002. IEEE, Montreal, QC, Canada (Oct 2021). https://doi.org/10.1109/ICCV48922.2021.00986, https://ieeexplore.ieee.org/document/9710580/ 14
17. Long, S., Zhao, Z., Pi, J., Wang, S., Wang, J.: Beyond Attentive Tokens: Incorporating Token Importance and Diversity for Efficient Vision Transformers. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10334–10343 (Jun 2023). https://doi.org/10.1109/CVPR52729.2023.00996, https://ieeexplore.ieee.org/document/10204564 3

18. Lou, M., Yu, Y.: OverLoCK: An Overview-first-Look-Closely-next ConvNet with Context-Mixing Dynamic Kernels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 128–138 (2025), https://openaccess.thecvf.com/content/CVPR2025/html/Lou_OverLoCK_An_Overview-first-Look-Closely-next_ConvNet_with_Context-Mixing_Dynamic_Kernels_CVPR_2025_paper.html 4, 10
19. Lu, C., de Geus, D., Dubbelman, G.: Content-Aware Token Sharing for Efficient Semantic Segmentation With Vision Transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23631–23640 (2023), https://openaccess.thecvf.com/content/CVPR2023/html/Lu_Content-Aware_Token_Sharing_for_Efficient_Semantic_Segmentation_With_Vision_Transformers_CVPR_2023_paper.html 2, 3
20. Marin, D., Chang, J.H.R., Ranjan, A., Prabhu, A., Rastegari, M., Tuzel, O.: Token Pooling in Vision Transformers for Image Classification. In: 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). pp. 12–21 (Jan 2023). https://doi.org/10.1109/WACV56688.2023.00010, https://ieeexplore.ieee.org/document/10030157 3
21. Meng, L., Li, H., Chen, B.C., Lan, S., Wu, Z., Jiang, Y.G., Lim, S.N.: AdaViT: Adaptive Vision Transformers for Efficient Image Recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12309–12318 (2022), https://openaccess.thecvf.com/content/CVPR2022/html/Meng_AdaViT_Adaptive_Vision_Transformers_for_Efficient_Image_Recognition_CVPR_2022_paper.html 3
22. Pan, Z., Zhuang, B., Liu, J., He, H., Cai, J.: Scalable Vision Transformers with Hierarchical Pooling. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 367–376 (Oct 2021). https://doi.org/10.1109/ICCV48922.2021.00043, https://ieeexplore.ieee.org/document/9710380 3
23. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. In: Advances in Neural Information Processing Systems. vol. 34, pp. 13937–13949. Curran Associates, Inc. (2021), https://papers.neurips.cc/paper_files/paper/2021/hash/747d3443e319a22747fbb873e8b2f9f2-Abstract.html 3
24. Ronen, T., Levy, O., Golbert, A.: Vision Transformers With Mixed-Resolution Tokenization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4613–4622 (2023), https://openaccess.thecvf.com/content/CVPR2023W/ECV/html/Ronen_Vision_Transformers_With_Mixed-Resolution_Tokenization_CVPRW_2023_paper.html 3
25. Tang, Q., Zhang, B., Liu, J., Liu, F., Liu, Y.: Dynamic Token Pruning in Plain Vision Transformers for Semantic Segmentation. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 777–786 (Oct 2023). https://doi.org/10.1109/ICCV51070.2023.00078, https://ieeexplore.ieee.org/document/10378640 2, 3
26. Wei, S., Ye, T., Zhang, S., Tang, Y., Liang, J.: Joint Token Pruning and Squeezing Towards More Aggressive Compression of Vision Transformers. In: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2092–2101 (Jun 2023). https://doi.org/10.1109/CVPR52729.2023.00208, https://ieeexplore.ieee.org/document/10203122 3
27. Woo, S., Debnath, S., Hu, R., Chen, X., Liu, Z., Kweon, I.S., Xie, S.: ConvNeXt V2: Co-Designing and Scaling ConvNets With Masked Autoencoders. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

pp. 16133–16142 (2023), https://openaccess.thecvf.com/content/CVPR2023/html/Woo_ConvNeXt_V2_Co-Designing_and_Scaling_ConvNets_With_Masked_Autoencoders_CVPR_2023_paper.html 2, 10, 14
28. Wu, Y.H., Zhang, S.C., Liu, Y., Zhang, L., Zhan, X., Zhou, D., Feng, J., Cheng, M.M., Zhen, L.: Low-Resolution Self-Attention for Semantic Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **47**(9), 8180–8192 (Sep 2025). https://doi.org/10.1109/TPAMI.2025.3577035, https://ieeexplore.ieee.org/document/11029508/ 10, 11
29. Xia, C., Wang, X., Lv, F., Hao, X., Shi, Y.: ViT-CoMer: Vision Transformer with Convolutional Multi-scale Feature Interaction for Dense Predictions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5493–5502 (2024), https://openaccess.thecvf.com/content/CVPR2024/html/Xia_ViT-CoMer_Vision_Transformer_with_Convolutional_Multi-scale_Feature_Interaction_for_Dense_CVPR_2024_paper.html 10
30. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P.: SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. In: Advances in Neural Information Processing Systems (Nov 2021), https://openreview.net/forum?id=OG18MI5TRL 10, 11, 13
31. Yuan, Y., Fu, R., Huang, L., Lin, W., Zhang, C., Chen, X., Wang, J.: HRFormer: High-Resolution Vision Transformer for Dense Predict. In: Advances in Neural Information Processing Systems (Nov 2021), https://openreview.net/forum?id=DF8LCjR03tX 10, 11
32. Yuan, Y., Xie, J., Chen, X., Wang, J.: SegFix: Model-Agnostic Boundary Refinement for Segmentation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 489–506. Springer International Publishing, Cham (2020). https://doi.org/10.1007/978-3-030-58610-2_29 4
33. Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A.: Semantic Understanding of Scenes Through the ADE20K Dataset. International Journal of Computer Vision **127**(3), 302–321 (Mar 2019). https://doi.org/10.1007/s11263-018-1140-0, https://doi.org/10.1007/s11263-018-1140-0 9
34. Zhu, X., Yang, X., Wang, Z., Li, H., Dou, W., Ge, J., Lu, L., Qiao, Y., Dai, J.: Parameter-Inverted Image Pyramid Networks. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (Nov 2024) 4
35. Ziwen, C., Patnaik, K., Zhai, S., Wan, A., Ren, Z., Schwing, A.G., Colburn, A., Fuxin, L.: AutoFocusFormer: Image Segmentation off the Grid. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 18227–18236 (2023), https://openaccess.thecvf.com/content/CVPR2023/html/Ziwen_AutoFocusFormer_Image_Segmentation_off_the_Grid_CVPR_2023_paper.html 2, 3, 6, 8, 9, 10, 11, 13, 14