SPFormer: Enhancing Vision Transformer with Superpixel Representation

Jieru Mei[1]   Liang-Chieh Chen[2]   Alan Yuille[1]   Cihang Xie[3]

[1]Johns Hopkins University   [2]Bytedance   [3]UC Santa Cruz

Abstract

In this work, we introduce SPFormer, a novel Vision Transformer enhanced by superpixel representation. Addressing the limitations of traditional Vision Transformers' fixed-size, non-adaptive patch partitioning, SPFormer employs superpixels that adapt to the image's content. This approach divides the image into irregular, semantically coherent regions, effectively capturing intricate details and applicable at both initial and intermediate feature levels.

SPFormer, trainable end-to-end, exhibits superior performance across various benchmarks. Notably, it exhibits significant improvements on the challenging ImageNet benchmark, achieving a 1.4% increase over DeiT-T and 1.1% over DeiT-S respectively. A standout feature of SPFormer is its inherent explainability. The superpixel structure offers a window into the model's internal processes, providing valuable insights that enhance the model's interpretability. This level of clarity significantly improves SPFormer's robustness, particularly in challenging scenarios such as image rotations and occlusions, demonstrating its adaptability and resilience.

1. Introduction

Over the past decade, the vision community has witnessed a remarkable evolution in visual recognition systems, from the resurgence of Convolutional Neural Networks (CNNs) in 2012 [19] to the cutting-edge innovation of Vision Transformers (ViTs) in 2020 [9]. This progression has instigated a significant shift in the underlying methodology for feature representation learning, transitioning from pixel-based (for CNNs) to patch-based (for ViTs).

Conventionally, pixel-based representations organize an image as a regular grid, allowing CNNs [13, 30, 35, 41] to extract local detailed features through sliding window operations. Despite the inductive bias inherent in CNNs, like translation equivariance, aiding their success in effectively learning visual representations, these networks face a challenge in capturing global-range information, typically necessitating the stacking of multiple convolutional operations and/or additional operations [6, 21] to enlarge their receptive fields.

On the other hand, ViTs [9] regard an image as a sequence of patches. These patch-based representations, usually of a much lower resolution compared to their pixel-based counterparts, enable global-range self-attention operations in a computationally efficient manner. While the attention mechanism successfully captures global interactions, it does so at the expense of losing local details, like object boundaries. Moreover, the low resolution of patch-based representations poses challenges to adaptation for high-resolution dense prediction tasks such as segmentation and detection, which require both local detail preservation and global context information.

This leads us to ponder an interesting question: *can we derive benefits from both preserved local details and effective long-range relationship capture*? In response, we explore superpixel-based solutions, which have been employed extensively in computer vision prior to the deep learning era [4, 28, 29, 33, 36, 44, 65]. These solutions provide locally coherent structures and reduce computational overhead compared to pixel-wise processing. Specifically, adaptive to the input, superpixels partition an image into irregular regions, with each region grouping pixels with similar semantics. This approach allows for a small number of superpixels, making it amenable to modeling global interactions through self-attention.

Motivated by the inherent limitations of patch representations in ViTs, we introduce an innovative transition to superpixel representation through our Superpixel Cross Attention (SCA). The resulting architecture, Superpixel Transformer (SPFormer), adeptly marries local detail preservation with global-range self-attention, enabling end-to-end trainability. In comparison to standard ViT architectures, SPFormer demonstrates remarkable enhancements across various tasks. For instance, it achieves impressive gains on the challenging ImageNet benchmark, such as 1.4% for DeiT-T and 1.1% for DeiT-S. Notably, the superpixel representation in SPFormer aligns seamlessly with semantic boundaries, even in unseen data. Crucially, the interpretability afforded by our superpixel representation deepens our understanding of the model's decision-making process, elucidating its robustness against rotations and occlusions. These findings highlight the potential of superpixel-based approaches in advancing
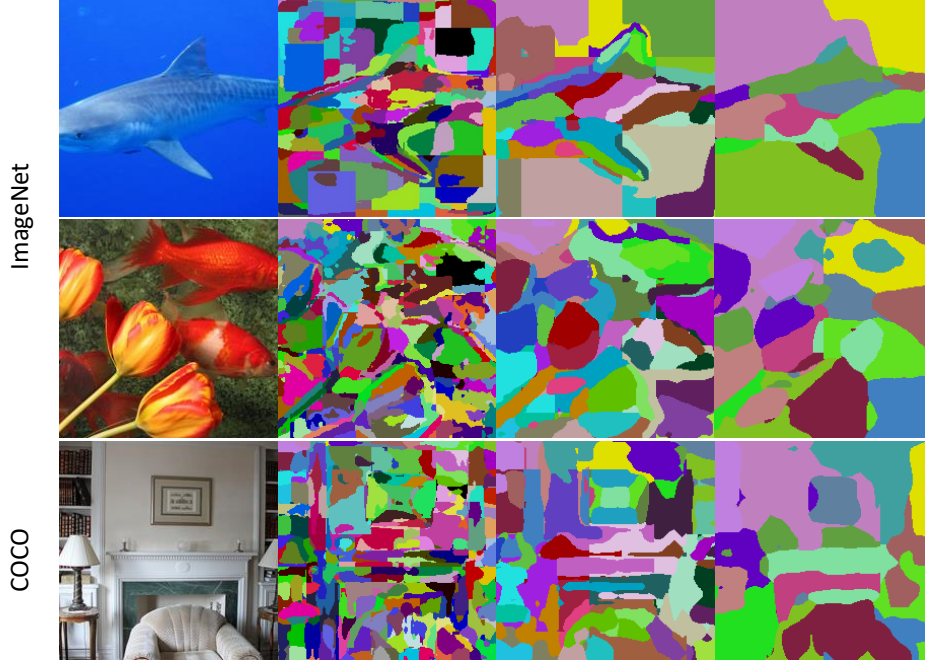
Figure 1. Visualization of learned superpixels with our SPFormer trained on ImageNet with category labels only. For each row, we show input image, visualization of 196, 49, and 16 superpixels. The learned superpixel aligns well with the object boundaries even with 16 superpixels. The last row shows results from a COCO image (not trained), demonstrating SPFormer's zero-shot ability.

the field, inspiring future research beyond traditional pixel and patch-based paradigms in visual representation.

2. Related Work

Pixel Representation Convolutional Neural Networks (CNNs) [13, 17, 19, 20, 24, 37, 40, 41] process an image as a grid of pixels in a sliding window manner. CNN has been the dominant network choice since the advent of AlexNet [19], benefiting from several design choices, such as translation equivariance and the hierarchical structure to extract multi-scale features. However, it requires stacking several convolution operations to capture long-range information [13, 37], and it could not easily capture global-range information, as the self-attention operation [46].

Patch Representation The self-attention mechanism [2] of Transformer architectures [46] effectively captures long-range information. However, its computation cost is quadratic to the number of input tokens. Vision Transformers (ViTs) [9] alleviate the issue by tokenizing (or patchifying) the input image with a sequence of patches (*e.g.*, patch size 16×16). The patch representation [9] unleashes the power of Transformer architectures [46] in computer vision, significantly impacting multiple visual recognition tasks [3, 5, 14, 32, 34, 42, 48, 57, 58, 66]. Due to the lack of the built-in inductive biases as in CNNs, learning with ViTs requires special training enhancements, *e.g.*, large-scale datasets [39], better training recipes [38, 42],

or architectural designs [23, 49]. To mitigate the issue, a few works exploit convolutions [20, 35] to tokenize the images, resulting in hybrid CNN-Transformer architectures [8, 12, 30, 45, 50, 51, 53, 59]. Unlike those works that simply gather knowledge from existing CNNs and ViTs, we explore a different superpixel representation in ViTs.

Superpixel Representation Before the deep learning era, superpixel is one of the most popular representations in computer vision [4, 28, 29, 33, 36, 44, 65]. Ren and Malik [33] preprocess images with superpixels that are locally coherent, preserving the structure necessary for the following recognition tasks. It also significantly reduces the computation overhead, compared to the pixel-wise processing. The superpixel clustering methods include graph-based approaches [11, 36], mean-shift [7, 47], or k-means clustering [1, 25]. Thanks to its effective representation, recently some works attempt to incorporate clustering methods into deep learning frameworks [16, 18, 26, 27, 52, 54–56, 60, 64]. For example, SSN [18] integrates the differentiable SLIC [1] to CNNs, allowing end-to-end training. Yu et al. [55, 56] regard object queries [5, 48] as cluster centers in Transformer decoders [46]. SViT [16] clusters the tokens to form the super tokens, where the clustering process has no gradient passed through[1]. Consequently, their network is not aware of the clustering process and could not recover from the clustering

---

[1]From official code: https://github.com/hhb072/STViT/blob/main/models/stvit.py#L206

error. CoCs [27] groups pixels into clusters, while aggregating features within each cluster by regarding the image as a set of points with coordinates concatenated. In contrast, our proposed method groups pixels into superpixels, and models their global relationship via self-attention. Furthermore, during clustering, CoCs uses a Swin-style window partition [23] that introduces visual artifacts, especially around the window boundaries.

3. Method

In this section, we formalize our superpixel representation and compare it with traditional methods in Sec. 3.1. We then detail the integration of this representation with our Superpixel Cross Attention mechanism in Sec. 3.2. Building on these concepts, our model SPFormer, which exemplifies an explainable and efficient approach to image processing, is presented in Sec. 3.3.

3.1. Superpixel Representation: Bridging Pixel and Patch Approaches

In the evolving landscape of feature representation, the transition from pixel to patch-based methods in Vision Transformers has opened new avenues for image processing. However, each method has limitations, inspiring our exploration of a more adaptive and efficient representation: superpixels.

Pixel Representation Conventional pixel representation treats an image $\mathbf{I}$ as a grid of high-resolution pixels, with $\mathbf{I} \in \mathcal{R}^{c \times h \times w}$. This representation, dominant in CNN-based methods, suffers from restricted contextual integration due to limited receptive fields. While self-attention mechanisms could theoretically enhance this integration, their application at this resolution is computationally burdensome due to the quadratic complexity with respect to the number of pixels.

Patch Representation Vision Transformers typically use a lower resolution patch representation, $\mathbf{P} \in \mathcal{R}^{c \times p_h \times p_w}$, reducing input length significantly. This reduction facilitates the application of self-attention mechanisms but at the cost of finer details and contextual nuances due to the coarse granularity of patches.

Superpixel Representation Our superpixel representation synthesizes the detail of pixel-based methods with the efficiency of patch-based approaches, consisting of superpixel features $\mathbf{S} \in \mathcal{R}^{c \times s_h \times s_w}$ and pixel-to-superpixel associations $\mathbf{A} \in \mathcal{R}^{n \times h \times w}$.

1. **Neighboring Superpixels ($\mathcal{N}_i$):** For each pixel i, neighboring superpixels are defined as $\mathcal{N}_i$, including the nearest superpixel and its Moore Neighborhood for $n = 9$. The association matrix $\mathbf{A}$ delineates relationships between pixels and their neighboring superpixels, fostering a competitive dynamic among them.

2. **Superpixel's Local Window ($\mathcal{W}_p$):** Derived from the neighboring superpixels ($\mathcal{N}_i$), each superpixel p is associ-
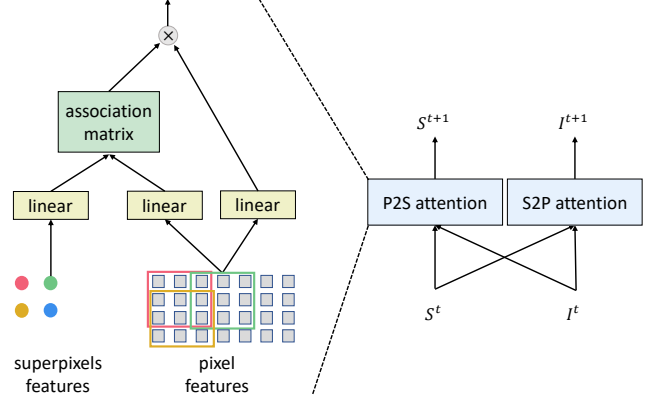

Figure 2. Illustration of our SCA module for iterative refinement of both superpixel and pixel features using a sliding window-based cross-attention mechanism. Each superpixel cross-attends to a localized region of pixels, as highlighted in the colored rectangle. On the left, we detail the Pixel-to-Superpixel (P2S) cross-attention process, while the Superpixel-to-Pixel (S2P) cross-attention is depicted similarly, albeit with reversed roles for superpixel and pixel.

ated with a local window $\mathcal{W}_p$. This window encompasses the neighboring pixels that contribute to the superpixel's feature representation. The overlapping nature of these local windows is pivotal for the implementation of SCA with a sliding window approach in Sec. 3.2, enabling more nuanced attention mechanisms.

The transformation from superpixel to pixel representation is captured by:

$$\mathbf{I}_i = \sum_{p \in \mathcal{N}_i} \mathbf{A}_{ip} \cdot \mathbf{S}_p \tag{1}$$

where $\mathbf{I}_i$ signifies the feature of the i-th pixel. This approach ensures boundary information preservation and finer granularity compared to direct patch upsampling, translating from a dense pixel grid to a coarser superpixel grid.

The superpixel representation uniquely conserves boundary information, enabling the maintenance of high-resolution features crucial for detailed tasks. Its robustness to image distortions, such as rotation and occlusion, makes it robust compared to traditional pixel or patch methods. In summary, superpixel representation is efficient due to its reduced resolution, explainable through semantic pixel clustering, and robust against challenging image transformations.

3.2. Superpixel Cross Attention

Given an initial pixel representation $\mathbf{I}^0$ and superpixel features $\mathbf{S}^0$, our method iteratively updates these features. At each iteration t, both superpixel features $\mathbf{S}^t$ and the association $\mathbf{A}^t$ are refined using a cross-attention mechanism within a sliding window, as depicted in Fig. 2. This mechanism is designed to maintain the locality of superpixels while ensuring high computational efficiency.

The SCA module, pivotal to our approach, encompasses two types of cross-attention: Pixel-to-Superpixel (P2S) and Superpixel-to-Pixel (S2P). For the P2S cross-attention, superpixel features cross-attend to pixel features within a localized region, enhancing the superpixel representations. Conversely, in the S2P cross-attention, pixel features are updated based on neighboring superpixels, refining the pixel representation.

The P2S cross-attention updates the superpixel features $\mathbf{S}^t$ by aggregating relevant pixel features, computed as:

$$\mathbf{S}_p^t = \mathbf{S}_p^{t-1} + \sum_{i \in \mathcal{W}_p} \text{softmax}_i \left(\mathbf{q}_{\mathbf{S}_p^{t-1}} \cdot \mathbf{k}_{\mathbf{I}_i^{t-1}} \right) \mathbf{v}_{\mathbf{I}_i^{t-1}} \quad (2)$$

Here, $\mathcal{W}_p$ indicates the set of pixels within the local window of superpixel p. The vectors $\mathbf{q}$ (query), $\mathbf{k}$ (key), and $\mathbf{v}$ (value) are derived from linear transformations of the prior iteration's superpixel features $\mathbf{S}_p^{t-1}$ and pixel features $\mathbf{I}_i^{t-1}$.

In the S2P cross-attention, pixel features $\mathbf{I}^t$ are updated using the updated associations $\mathbf{A}_{ip}^t$, calculated as follows:

$$\mathbf{A}_{ip}^t = \text{softmax}_{p \in \mathcal{N}_i} \left(\mathbf{q}_{\mathbf{I}_i^{t-1}} \cdot \mathbf{k}_{\mathbf{S}_p^{t-1}} \right) \quad (3)$$

where $\mathcal{N}_i$ denotes the neighboring superpixels of pixel i. The updated pixel representation $\mathbf{I}_i^t$ is then derived by:

$$\mathbf{I}_i^t = \mathbf{I}_i^{t-1} + \sum_{p \in \mathcal{N}_i} \mathbf{A}_{ip}^t \cdot \mathbf{v}_{\mathbf{S}_p^{t-1}} \quad (4)$$

Here, the value vector $\mathbf{v}$ is obtained through a linear transformation of the preceding superpixel features $\mathbf{S}_p^{t-1}$.

To incorporate positional information within SCA, we utilize Convolution Position Embedding (CPE) [16], which captures the spatial relationships within the image. Prior to applying P2S and S2P cross-attentions, both superpixel and pixel features are augmented with CPE, implemented as a 3×3 depthwise convolution with a skip connection. This enhancement strengthens the association between pixels and superpixels based on their spatial proximity, fostering more accurate pixel-superpixel alignments.

These iterative update equations are fundamental to refining both pixel and superpixel feature representations, thereby enhancing the overall quality and accuracy of the feature representation. The proposed SCA module, capable of multiple iterations t, is a cornerstone of our method and will be further elaborated in the subsequent architecture SPFormer in the next subsection.

3.3. SPFormer Architecture

Our architecture, designed to leverage the advantages of the proposed superpixel representation, introduces minimal alterations from the standard ViT [9]. In alignment with the ViT methodology, we employ a non-overlapping patchify layer. However, we utilize a smaller window size of 4×4

for extracting initial pixel features, as opposed to the conventional 16×16. This reduction in window size is made feasible by the effective superpixel representation, which significantly decreases the input length.

Initially, the superpixel features $\mathbf{S}^0$ are derived using a 1×1 convolution and 4×4 average pooling, based on the pixel features $\mathbf{I}^0$. We utilize the SCA module to iteratively update these superpixel features $\mathbf{S}^0$ and the association $\mathbf{A}^0$ (across t iterations), as outlined in Sec. 3.2. The SCA module capitalizes on the local spatial context within a superpixel to enhance its representation. Subsequently, the updated superpixel features undergo Multi-Head Self-Attention (MHSA), enabling the network to discern long-range dependencies and contextual information across various superpixels, thus facilitating a comprehensive understanding of the image.

We observed that, even with multiple iterations (e.g., $t > 2$) within an SCA module, generating lower-level superpixel representations may not perfectly align with the overall context, mainly due to insufficient semantic information. To address this, we propose a gradual refinement strategy for the superpixel representations through multiple SCA modules, each comprising a few iterations (e.g., $t = 2$). Each subsequent SCA module utilizes the updated pixel features to generate semantically richer superpixels.

Specifically, prior to advancing to the next SCA module, the superpixel features are projected through a 1×1 convolution. The pixel features are then updated according to Eq. (1), incorporating a skip connection [13]. This method ensures that pixel representations are refined in light of the globally context-enhanced superpixel features obtained from preceding SCA and MHSA modules. Rather than reinitializing superpixel features from the beginning, we utilize the contextually enriched superpixel features from the preceding stage as the starting point. This methodology, depicted in Fig. 3, systematically enhances the superpixel representations, allowing for the capture of progressively more complex semantic information.

In the concluding stage, global average pooling is applied to the superpixel features, and the resultant representation is fed into a linear classifier for image classification tasks.

Conceptually, our network architecture can be envisaged as a dual-branch structure. One branch maintains a dense pixel representation with high resolution, while the other branch is dedicated to our proposed low-resolution superpixel representation. Minimal direct operations are performed on the dense pixel representation, allowing us to concentrate most computational efforts on the more efficient superpixel representation. This dual-branch approach achieves computational efficiency without compromising the preservation of local details in the image representation.
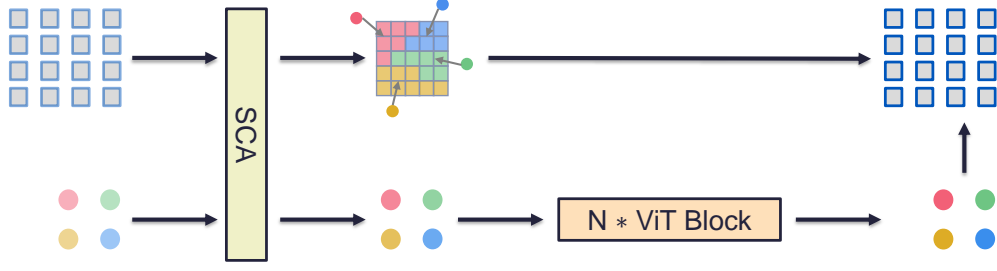
Figure 3. Illustration of a single stage of the SPFormer architecture. It starts with initial superpixel features and pixel features as inputs. The SCA module iteratively refines superpixel features, enhancing their semantic richness. These features are then processed by the Multi-Head Self-Attention (MHSA) for global contextual understanding. The stage concludes by updating the pixel features based on the enriched superpixel information, readying them for the next stage or for final pooling and classification. This design showcases the efficient integration of local detail and global context in SPFormer.

4. Experiments

We first detail our method's implementation in Sec. 4.1. We then demonstrate its explainability in Sec. 4.2 and assess efficiency on image classification and segmentation in Sec. 4.3.

4.1. Implementation Details

SPFormer establishes a specific ratio between the dimensions of superpixel and pixel features. By design, the spatial dimensions of superpixel features are reduced to $\frac{1}{4} \times \frac{1}{4}$ of their corresponding pixel features. This downscaling strategy is instrumental in encoding contextually rich information at a more abstract level, while simultaneously preserving essential details at the pixel level.

In the SCA module, we employ multi-head attention to manage attention and interaction between superpixels effectively. This setup not only leverages global contextual information optimally but also produces multiple superpixel representations, as shown in Fig. 4. These varied representations capture different granularities, addressing the ambiguity commonly associated with superpixel over-segmentation. Specifically, we allocate two heads for our smaller variants (SPFormer-T and SPFormer-S) and three heads for the base model (SPFormer-B).

Figure 4. The multi-head SCA design generates multiple superpixel representations, each capturing different semantic relationships and addressing the ambiguity in superpixel over-segmentation.

The SCA blocks are integrated into the standard ViT architecture, strategically positioned just before the first and third self-attention blocks. We adopt the LayerScale technique, as described in Touvron et al. [43], to regulate gradient flow, thereby enhancing training stability and convergence. It is noteworthy that, combined with the residual connection as formulated in Eq. (2) and Eq. (4), our method initially resembles vanilla patches, evolving to leverage superpixels as training progresses.

Following the training protocols detailed in DeiT [42], we utilize strong data augmentations, the AdamW optimizer, and a cosine decay learning rate schedule. All models undergo training on the ImageNet dataset [34] for a duration of 300 epochs. During the training phase of SPFormer-B/16, we faced significant overfitting challenges. To counteract this, we increased the Stochastic Depth [15] rate from 0.1 to 0.6, effectively mitigating the overfitting. This adjustment underscores the potential need for more sophisticated regularization techniques, especially those tailored to the superpixel representation, which we aim to investigate in future research endeavors.

For SPFormer variants, the default configuration employs a 4×4 patchify layer. Variants augmented with two convolution layers of kernel size 3 with stride 2 are denoted by $\dagger$. Other configurations, adapting different superpixel sizes, are indicated by their ViT-equivalent patch sizes, such as SPFormer/32 for a 32×32 patch size. This notation ensures clear distinction between each variant in our experiments.

4.2. Unveiling SPFormer's Explainability

Integrating superpixel representation into the Vision Transformer architecture adds a significant layer of explainability compared to conventional fixed patch partition methods. This section first discusses the inherent explainability of our superpixel representation, followed by an evaluation of its semantic alignment and generalizability to unseen data.

4.2.1 Superpixel Representation as an Explainability Tool

Our method's superpixel representation can be visualized through the association matrix $\mathbf{A}$, providing insights into

the model's internal processing. In Fig. 1, we visualize the learned soft associations by selecting the argmax over the superpixels:

$$\hat{\mathbf{A}} = \mathrm{argmax}(\mathbf{A}) \tag{5}$$

These visualizations reveal that, even with a soft association, the superpixels generally align with image boundaries. This alignment is noteworthy as it emerges even though the network is only trained on image category labels. Thus, the model segments images into irregular, semantics-aware regions while reducing the number of tokens needed for representation.

Furthermore, we assess the generalizability of our superpixel representation using the COCO dataset [22], which consists of high-resolution images with complex scenes. For this evaluation, we resize and center-crop COCO images to align with the ImageNet evaluation pipeline. Fig. 5 showcases the visual representation of superpixels on these images. Remarkably, the superpixels generated by SPFormer, trained exclusively on ImageNet, adapt well to this unseen data, capturing intricate structures such as thin objects. This adaptability highlights the model's capability to preserve detail and generalize its superpixel representation to new contexts.
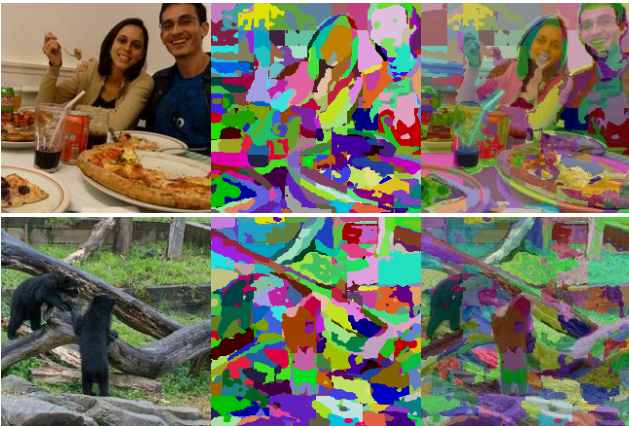

Figure 5. Zero-shot transferability on the COCO dataset. Trained solely on ImageNet, SPFormer demonstrates effective segmentation of unseen COCO images into detailed superpixels. 196 superpixels are used in this visualization.

4.2.2 Semantic Alignment of Superpixels

Our evaluation of superpixel representation focuses on its ability to align with ground truth boundaries in images, despite the model not being trained on the datasets used for this assessment. This test involved a quantitative analysis on both object and part levels using the Pascal VOC 2012 dataset [10] and Pascal-Part-58 [61]. Remarkably, these

assessments were performed without any training on these specific datasets, underscoring the model's generalization capabilities.

In our approach, each superpixel or patch's prediction is derived by aggregating the ground truth labels of the pixels it encompasses. We assign the most frequently occurring label within a superpixel as its prediction, assuming optimal classification. This method leverages the soft associations produced by our SCA module, where predictions are formed by combining pixel labels with their corresponding weights and upscaled as per Eq. (1).

Diverging from the single-superpixel outcome of traditional patch representation, our model employs a multi-head design in the SCA module. This allows for the creation of multiple, distinct superpixels for each head, enhancing the richness and diversity of the extracted features (see Fig. 4). For our evaluations, we computed an average of predictions across all heads. It's noteworthy that effective feature extraction in our model is deemed successful if even a single head accurately identifies a superpixel.

The proficiency of our superpixel approach is demonstrated in its performance compared to vanilla ViTs that utilize patch representations with a stride of 16. ViTs often suffer from a granularity trade-off, losing finer details in favor of broader patch representations. In contrast, the superpixels from our SCA module, as shown in Tab. 1, manifest substantial improvements — achieving a 4.2% increase in object-level and 4.6% in part-level mean Intersection over Union (mIoU) with SPFormer-S†. Furthermore, these superpixels display a quality comparable to those from traditional superpixel methods like SLIC [1], highlighting our method's effectiveness in capturing detailed semantic information without direct training on the evaluation datasets.

Table 1. Evaluation of superpixel quality in a zero-shot setting on Pascal VOC 2012 and Pascal-Parts-58 datasets, using 196 patches/superpixels. Our SPFormer variants demonstrate notable improvements over traditional patch representations and are competitive with the SLIC method.

Method	Pascal Voc2012		Pascal-Parts-58	
	mIoU	mAcc	mIoU	mAcc
Patch	87.8	92.8	68.7	78.2
SPFormer-T†	91.5	95.7	71.5	79.9
SPFormer-S†	92.0	96.6	73.3	82.4
SPFormer-B†	91.2	96.3	72.5	81.4
SLIC [1]	92.5	95.4	74.0	81.7

4.2.3 Explainability-Driven Robustness

The robustness of SPFormer is deeply intertwined with its explainability, particularly through the superpixel representation. This section explores how the model's transparent and interpretable features contribute to its resilience against image modifications like rotation and occlusion.

Robustness to Rotation Our model's capacity to generate coherent superpixels even under rotational transformations showcases the robustness afforded by its explainable structure. By visualizing how superpixels adapt to rotated images, we gain insights into the model's stability in varied orientations. While SPFormer demonstrates a heightened robustness to rotation, it still exhibits some limitations, likely due to the learnable absolute position embeddings not being inherently rotation-invariant. These observations suggest potential avenues for enhancing rotational robustness, possibly through integrating rotation-invariant mechanisms within the superpixel representation or the network architecture. Figure 6 illustrates the model's performance under rotation, and Table 2 quantifies this robustness, under patch size 32.
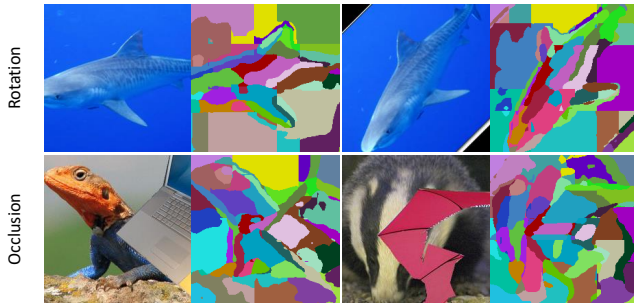

Figure 6. Visualization of SPFormer's superpixel representation under rotation and occlusion, highlighting the model's adaptability and robustness.

Robustness to Occlusion The occlusion robustness of SP-Former is another facet where explainability plays a key role. By examining superpixel behavior in occluded images, we observe the model's ability to distinguish between occluders and the object of interest. Unlike traditional patch-based representations, which tend to blend occluders with the object, our superpixel representation more effectively isolates and identifies obscured parts of the image. This nuanced differentiation is a direct result of the model's explainable superpixel structure, which provides a more detailed and context-aware interpretation of the image content, as demonstrated in Figure 6.

4.3. Efficiency in Image Classification and Segmentation

4.3.1 Main Results on ImageNet

Our assessment of SPFormer on ImageNet underlines its enhanced efficiency and performance compared to the DeiT

Table 2. Quantitative evaluation of SPFormer's robustness to rotation, comparing performance at different angles. Variants augmented with two convolution layers of kernel size 3 with stride 2 are denoted by †.

Model	Clean	15	30	45
DeiT-S/32 [42]	73.3	71.1	67.7	59.5
SPFormer-S/32†	77.9	75.2	73.4	66.9

Table 3. Comparative analysis of SPFormer's performance on ImageNet classification against DeiT baselines. Variants augmented with two convolution layers of kernel size 3 with stride 2 are denoted by †.

Model	#Params	#FLOPs	Top-1
SPFormer-S/56	22M	0.5G	72.3
DeiT-T [42]	5M	1.3G	72.2
SPFormer-T	5M	1.3G	73.6
SPFormer-T†	5M	1.3G	75.0
DeiT-S/32 [42]	22M	1.1G	73.3
SPFormer-S/32	22M	1.2G	76.4
SPFormer-S/32†	22M	1.3G	77.9
DeiT-S [42]	22M	4.6G	79.9
SPFormer-S	22M	5.2G	81.0
SPFormer-S†	22M	5.3G	81.7
DeiT-B [42]	87M	17.5G	81.8
SPFormer-B	87M	19.2G	82.4
SPFormer-B†	87M	19.2G	82.7

baseline across diverse configurations, as shown in Tab. 3. Notably, SPFormer-S, utilizing the standard ViT configuration with 196 tokens, surpasses DeiT-S by a margin of 1.1% (achieving 81.0% *vs*. DeiT's 79.9%). In the case of SPFormer-T, it exceeds DeiT-T by 1.4% (73.6% *vs*. 72.2%). This advantage becomes more pronounced when larger patch sizes, such as 32, are used. While DeiT-S/32 exhibits a decline in performance due to its coarse granularity, SPFormer-S/32 maintains robust performance at 76.4%, even outperforming DeiT-T by a significant 4.2% with less FLOPs.

A notable aspect of SPFormer is the shift in computational load, with MLPs taking precedence over self-attention mechanisms. This redistribution suggests an alternative scaling strategy, namely increasing image resolution to encapsulate finer details. By adapting to a higher resolution of 448, through doubling the window size from 4 to 8, SPFormer retains computational efficiency and achieves a 0.3% improvement in performance compared to its standard configuration. In contrast, when DeiT-S employs a similar strategy, it gains a marginal improvement of only 0.1%, limited by the granularity of its patch representation.

Further enhancing SPFormer's initial feature extraction phase in the superpixel cross-attention stage, we introduce a lightweight convolution stem comprising two or three 3×3 convolutions with a stride of 2. This enhancement has consistently improved performance, exemplified by SPFormer-S/32†, which witnesses an additional increase of 1.5% in ImageNet accuracy, reaching 77.9%.

Table 4. Ablation study on the design choices in SPFormer.

Model	#Params	#FLOPs	Top-1
SPFormer-S/32	22M	1.2G	76.4
Single Iteration in SCA	22M	1.2G	75.4
SCA at Initial Layer Only	22M	1.2G	74.8
Single-Head SCA	22M	1.2G	75.6
Learnable Position Embeddings	22M	1.2G	76.1

4.3.2 Ablation Study: Design Choices in SPFormer

We evaluates key design elements of SPFormer-S/32 on the ImageNet validation set. We investigate the impacts of iteration count in the SCA module, the placement of SCA within the architecture, the use of multi-head attention, and the choice of position embeddings.

The findings highlight the importance of multiple iterations in SCA for performance enhancement, with a single iteration leading to a 1.0% drop in accuracy. The strategic placement of SCA across different layers is crucial, as restricting it to the initial layer causes a 1.6% accuracy reduction, indicating that higher-level features play a vital role in augmenting semantic depth and in rectifying early-stage superpixel inaccuracies. Furthermore, employing multi-head attention in SCA is significant for capturing diverse superpixel relationships, with its absence leading to a 0.8% decrease in accuracy. Lastly, using learnable position embeddings over CPE results in a slight drop in performance.

This ablation study validates the effectiveness of the considered design choices in SPFormer, affirming their contributions to the overall performance of the model.

4.3.3 Semantic Segmentation: Utilizing SPFormer's High-Resolution Feature Preservation

SPFormer's superpixel representation intrinsically maintains higher resolution features, making it particularly suitable for semantic segmentation tasks. This characteristic allows for detailed and context-rich segmentation outputs, a key advantage over traditional patch-based methods.

Incorporating SPFormer into the SETR [62] framework, we enhance segmentation performance by directly classifying individual superpixels. This direct approach leverages SPFormer's high-resolution feature preservation, allowing for more nuanced segmentation. The final segmentation maps are generated by upscaling the superpixel-based logits using Eq. (1).

We evaluate SPFormer on the ADE20K [63] and Pascal Context [31] datasets. Utilizing ImageNet-pretrained models, SPFormer demonstrates significant improvements in mIoU, highlighting its effectiveness in detailed segmentation tasks. Detailed training parameters and methodologies for these evaluations are provided in the supplementary material.

As shown in Tab. 5 and Tab. 6, the performance gains in mIoU are noteworthy when using ImageNet-pretrained models: 4.2% improvement on ADE20K and 2.8% on Pascal Context. These results not only highlight the detailed nature of SPFormer's superpixel representation but also its adaptability to diverse and complex datasets. To further validate the intrinsic segmentation capabilities of SPFormer, we conduct additional training from scratch. This approach reiterates the model's strength in maintaining high-resolution features independently of pretraining influences, leading to mIoU improvements of 3.0% on ADE20K and 3.1% on Pascal Context compared to baseline methods.

SPFormer's preservation of high-resolution features within its superpixel representation thus proves to be a powerful asset for semantic segmentation, enabling detailed analyses of diverse image datasets.

Table 5. Semantic segmentation on ADE20K val split.

Method	#Params	#FLOPs	Pretrained	mIoU
DeiT-S	22M	32G	✗	20.1
SPFormer-S	23M	35G	✗	23.1
DeiT-S	22M	32G	✓	42.3
SPFormer-S	23M	35G	✓	46.5

Table 6. Semantic segmentation on Pascal Conext val split.

Method	#Params	#FLOPs	Pretrained	mIoU
DeiT-S	22M	27G	✗	18.0
SPFormer-S	23M	30G	✗	21.1
DeiT-S	22M	27G	✓	48.3
SPFormer-S	23M	30G	✓	51.2

5. Conclusion

In this work, we introduced SPFormer, a novel approach for feature representation in vision transformers, emphasizing superpixel representation. This method marks a shift from traditional pixel and patch-based approaches, offering three distinct advantages: explainability through semantic grouping of pixels, efficiency due to a reduced number of superpixels facilitating global self-attention, and robustness against image distortion. SPFormer not only demonstrates improved performance on ImageNet classification tasks but

also excels in semantic segmentation, showcasing its versatility. Our results underscore the potential of superpixel-based methods in diverse vision tasks, paving the way for future exploration in this promising direction.

References

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 34 (11):2274–2282, 2012. 2, 6

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR*, 2015. 2

[3] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *ICLR*, 2022. 2

[4] Eran Borenstein and Shimon Ullman. Class-specific, top-down segmentation. In *ECCV*, 2002. 1, 2

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

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *TPAMI*, 40(4):834–848, 2018. 1

[7] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *TPAMI*, 24(5):603–619, 2002. 2

[8] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in Neural Information Processing Systems*, 34:3965–3977, 2021. 2

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 2, 4

[10] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *IJCV*, 111(1):98–136, 2015. 6

[11] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 59:167–181, 2004. 2

[12] Jianyuan Guo, Kai Han, Han Wu, Yehui Tang, Xinghao Chen, Yunhe Wang, and Chang Xu. Cmt: Convolutional neural networks meet vision transformers. In *CVPR*, 2022. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 4

[14] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, 2022. 2

[15] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q. Weinberger. Deep networks with stochastic depth. In *ECCV*, pages 646–661. Springer, 2016. 5

[16] Huaibo Huang, Xiaoqiang Zhou, Jie Cao, Ran He, and Tieniu Tan. Vision transformer with super token sampling. In *CVPR*, 2023. 2, 4

[17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 2

[18] Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Superpixel sampling networks. In *ECCV*, 2018. 2

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 2012. 1, 2

[20] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[21] Yingwei Li, Xiaojie Jin, Jieru Mei, Xiaochen Lian, Linjie Yang, Cihang Xie, Qihang Yu, Yuyin Zhou, Song Bai, and Alan L. Yuille. Neural architecture search for lightweight non-local networks. In *CVPR*, 2020. 1

[22] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 2014. 6

[23] 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 *ICCV*, 2021. 2, 3

[24] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 2

[25] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982. 2

[26] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *Advances in Neural Information Processing Systems*, 33:11525–11538, 2020. 2

[27] Xu Ma, Yuqian Zhou, Huan Wang, Can Qin, Bin Sun, Chang Liu, and Yun Fu. Image as set of points. In *ICLR*, 2023. 2, 3

[28] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *IJCV*, 43:7–27, 2001. 1, 2

[29] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, 2001. 1, 2

[30] Sachin Mehta and Mohammad Rastegari. Mobilevit: lightweight, general-purpose, and mobile-friendly vision transformer. In *ICLR*, 2022. 1, 2

[31] Roozbeh Mottaghi, Xianjie Chen, Xiaobai Liu, Nam-Gyu Cho, Seong-Whan Lee, Sanja Fidler, Raquel Urtasun, and Alan L. Yuille. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, pages 891–898. IEEE Computer Society, 2014. 8

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2

[33] Xiaofeng Ren and Jitendra Malik. Learning a classification model for segmentation. In *ICCV*, 2003. 1, 2

[34] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115:211–252, 2015. 2, 5

[35] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 1, 2

[36] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *TPAMI*, 22(8):888–905, 2000. 1, 2

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 2

[38] Andreas Steiner, Alexander Kolesnikov, Xiaohua Zhai, Ross Wightman, Jakob Uszkoreit, and Lucas Beyer. How to train your vit? data, augmentation, and regularization in vision transformers. *arXiv:2106.10270*, 2021. 2

[39] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017. 2

[40] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 2

[41] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 2019. 1, 2

[42] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 2, 5, 7

[43] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *ICCV*, pages 32–42. IEEE, 2021. 5

[44] Zhuowen Tu and Song-Chun Zhu. Image segmentation by data-driven markov chain monte carlo. *TPAMI*, 24(5):657–673, 2002. 1, 2

[45] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *ECCV*, 2022. 2

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[47] Andrea Vedaldi and Stefano Soatto. Quick shift and kernel methods for mode seeking. In *ECCV*, 2008. 2

[48] Huiyu Wang, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *CVPR*, 2021. 2

[49] 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 *ICCV*, 2021. 2

[50] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *ICCV*, 2021. 2

[51] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. *Advances in Neural Information Processing Systems*, 34:30392–30400, 2021. 2

[52] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *CVPR*, 2022. 2

[53] Chenglin Yang, Siyuan Qiao, Qihang Yu, Xiaoding Yuan, Yukun Zhu, Alan Yuille, Hartwig Adam, and Liang-Chieh Chen. Moat: Alternating mobile convolution and attention brings strong vision models. In *ICLR*, 2023. 2

[54] Fengting Yang, Qian Sun, Hailin Jin, and Zihan Zhou. Superpixel segmentation with fully convolutional networks. In *CVPR*, 2020. 2

[55] Qihang Yu, Huiyu Wang, Dahun Kim, Siyuan Qiao, Maxwell Collins, Yukun Zhu, Hartwig Adam, Alan Yuille, and Liang-Chieh Chen. Cmt-deeplab: Clustering mask transformers for panoptic segmentation. In *CVPR*, 2022. 2

[56] Qihang Yu, Huiyu Wang, Siyuan Qiao, Maxwell D. Collins, Yukun Zhu, Hartwig Adam, Alan L. Yuille, and Liang-Chieh Chen. k-means Mask Transformer. In *ECCV*, 2022. 2

[57] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *CVPR*, 2022. 2

[58] Yaodong Yu, Tianzhe Chu, Shengbang Tong, Ziyang Wu, Druv Pai, Sam Buchanan, and Yi Ma. Emergence of segmentation with minimalistic white-box transformers. *CoRR*, abs/2308.16271, 2023. 2

[59] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. In *ICCV*, 2021. 2

[60] Yifan Zhang, Bo Pang, and Cewu Lu. Semantic segmentation by early region proxy. In *CVPR*, 2022. 2

[61] Yifan Zhao, Jia Li, Yu Zhang, and Yonghong Tian. Multi-class part parsing with joint boundary-semantic awareness. In *ICCV*, pages 9176–9185. IEEE, 2019. 6

[62] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H. S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, pages 6881–6890. Computer Vision Foundation / IEEE, 2021. 8

[63] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *CVPR*, pages 5122–5130. IEEE Computer Society, 2017. 8

[64] Alex Zihao Zhu, Jieru Mei, Siyuan Qiao, Hang Yan, Yukun Zhu, Liang-Chieh Chen, and Henrik Kretzschmar. Superpixel transformers for efficient semantic segmentation. *CoRR*, abs/2309.16889, 2023. 2

[65] Song Chun Zhu and Alan Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multiband image segmentation. *TPAMI*, 18(9):884–900, 1996. 1, 2

[66] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. In *ICLR*, 2021. 2