Enhancing Vision Transformer with Superpixel Representation
Pixel Representation

- Typically have high-resolution
- Need a local sliding window approach for efficient processing
- **Intractable for global self-attention**, due to the quadratic complexity
Patch Representation

- Image as a set of 16x16 patches, which enables us to learn global information
- Low-resolution, thus *sacrifices image details.*
Can superpixels help?

Superpixels **over-segment the image** into similar regions.

Usually used as the **preprocessing** step to reduce the complexity
Superpixel Representation

- Superpixel features
- Superpixel association

Pixel Unfolding
Superpixel Representation

- **Efficient**: lower resolution than pixel/patch
- **Explainable**: formed by grouping pixel features with similar semantics
- **Robust**: rotation & occlusion, driven by Explainability
Patch → Superpixel

It’s straightforward to directly replace the patches by the superpixels.
- However, there are additional challenges.
Challenges

Traditional superpixel methods only use low level features (RGB + position)

- Sensitive to low level data augmentation
- Not aware of semantic information
Challenges

The problem is amplified as

- **Not recoverable** from superpixel errors
- **Not differentiable** due to the hard assignment
  - Each pixel is assigned to only one superpixel
Challenges

Traditional superpixel method compute a unique over-segmentation

- As an over-segmentation method, there is built-in ambiguity
- We may require different granularity for understanding the image
Superpixel as Multi-head Sliding-Window Cross Attention
Preliminary

$Q_{pi}^t = e^{-D(I_p, S_i^{t-1})} = e^{-||I_p - S_i^{t-1}||^2}$

Reformulation

Superpixel Cross Attention

- Multi-head mechanism
  - Multiple superpixel assignment for Ambiguity and granularity
- Superpixel features are updated in a residual manner
  - Ensures the training stability

Superpixels will emerge
Superpixel Transformers for Efficient Semantic Segmentation
Dense prediction tasks are expensive
- It requires **expensive decoders** which are often stacked upsample-convs.

Motivation

Dense prediction tasks require extracting contextual information efficiently


Motivation

- Effectively leverage global context information
- Significantly reduce the computational cost
  - Due to the operation on high resolution pixel features
Superpixel Transformer

Replaces the decoder using our superpixel representation
Superpixel Transformer

Extract Superpixel features with reformulation, initialized by learnable queries
Superpixel Transformer

Meanwhile we compute the association between each superpixel and pixel
We use global self-attention to enrich the Superpixel features
- 16x32 superpixels vs 64x256 pixels
Superpixel Transformer

Direct classify each superpixel
As for each pixel, the final output is a **weighted combination** of the surrounding superpixels’ logits, using the **association** instead of bilinear upsampling
We decompose the pixel features into a low dimensional superpixel representation.

By reducing the number of the latent features, we are able to perform efficient global self-attention between the superpixel features.

Generating the final semantic segmentation predictions is done entirely by projecting the superpixels back into the image.
## Results on Cityscapes

<table>
<thead>
<tr>
<th>Method</th>
<th>Backbone</th>
<th>Params ↓</th>
<th>FLOPs ↓</th>
<th>FPS↑</th>
<th>mIoU ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>RegProxy* [53]</td>
<td>ViT-S [17]</td>
<td>23M</td>
<td>270G</td>
<td>-</td>
<td>79.8</td>
</tr>
<tr>
<td>(k\text{MaX-DeepLab}^{†} [50])</td>
<td>ResNet-50 [21]</td>
<td>56M</td>
<td>434G</td>
<td>9.0</td>
<td>79.7</td>
</tr>
<tr>
<td>SP-Transformer</td>
<td>ResNet-50 [21]</td>
<td>29M</td>
<td>253G</td>
<td>15.3</td>
<td>80.4</td>
</tr>
<tr>
<td>RegProxy* [53]</td>
<td>ViT-L/16 [17]</td>
<td>307M</td>
<td>-</td>
<td>-</td>
<td>81.4</td>
</tr>
<tr>
<td>SegFormer [47]</td>
<td>MiT-B5 [47]</td>
<td>85M</td>
<td>1,448G</td>
<td>2.5</td>
<td>82.4</td>
</tr>
<tr>
<td>(k\text{MaX-DeepLab}^{†} [50])</td>
<td>ConvNeXt-L [32]</td>
<td>232M</td>
<td>1,673G</td>
<td>3.1</td>
<td>83.5</td>
</tr>
<tr>
<td>SP-Transformer</td>
<td>ConvNeXt-L [32]</td>
<td>202M</td>
<td>1,557G</td>
<td>3.6</td>
<td>83.1</td>
</tr>
</tbody>
</table>
Visualization

Argmax of the soft association -> hard assignment

- Can capture thin objects like the poles.
- Sharper edges than the GT
- No direct supervision on the superpixel associations, implicitly learned by the network
Can we learn superpixels without pixel annotation?
Superpixel Transformer for Classification
Superpixel Embedding

ViT as our baseline

- Pixel features from patch embedding with stride 4
- Superpixel features initialized by 4x4 average pooling
Revisit the Challenges

- Non-differentiability
- Ambiguity & Granularity
- Superpixels are generated only in low level features
- Not recoverable from superpixel errors
Superpixel Regeneration

Regenerate the superpixels within the middle of the network using the enriched features.
Superpixel Regeneration

We partition the ViT into multiple stages
Superpixel Regeneration

Then, we use the **global self-attention** to enrich superpixel features
Superpixel Regeneration

We enhance the pixel features through the updated superpixel features, utilize the association for upsampling.
Architecture

Our network can be viewed as a dual-branch architecture.
- Pixel branch: high resolution
- Superpixel branch: low resolution
Visualization

Our method can generate reasonable superpixels even with just 16 tokens
Visualization - Rotation

Our approach produces meaningful superpixels when the image is rotated
Visualization - Occlusion

Our method effectively finds the boundaries and separates them into distinct superpixels.

- Patch representation inherently mixes foreground objects with occluders in certain patches.
Visualization - Transferability

Our method can transfer to MSCOCO images \textit{without fine-tuning}
Visualization - Transferability

Our method can produce reasonable results using less superpixels

COCO
Superpixel Quality

Alignment with ground truth boundaries in *Zero-shot* setting

<table>
<thead>
<tr>
<th>Method</th>
<th>Pascal Voc2012</th>
<th>Pascal-Parts-58</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mIoU</td>
<td>mAcc</td>
</tr>
<tr>
<td>Patch</td>
<td>87.8</td>
<td>92.8</td>
</tr>
<tr>
<td>SPFormer-T†</td>
<td>91.5</td>
<td>95.7</td>
</tr>
<tr>
<td>SPFormer-S†</td>
<td>92.0</td>
<td>96.6</td>
</tr>
<tr>
<td>SPFormer-B†</td>
<td>91.2</td>
<td>96.3</td>
</tr>
<tr>
<td>SLIC [1]</td>
<td>92.5</td>
<td>95.4</td>
</tr>
</tbody>
</table>
Empirical Results

Our models have much larger capacity, empowered by the superpixel representation.

- Better regularization methods are desired
  - Suitable data augmentation for superpixels?

<table>
<thead>
<tr>
<th>Method</th>
<th>params</th>
<th>FLOPs</th>
<th>ImageNet Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeiT-T</td>
<td>5M</td>
<td>1.3G</td>
<td>72.2</td>
</tr>
<tr>
<td>DeiT-S</td>
<td>22M</td>
<td>4.6G</td>
<td>79.9</td>
</tr>
<tr>
<td>SPFormer-T</td>
<td>5M</td>
<td>1.3G</td>
<td>73.6 (+1.4)</td>
</tr>
<tr>
<td>SPFormer-S</td>
<td>22M</td>
<td>4.9G</td>
<td>81.0 (+1.1)</td>
</tr>
<tr>
<td>DeiT-Base</td>
<td>87M</td>
<td>17.6G</td>
<td>81.8</td>
</tr>
<tr>
<td>SPFormer-B*</td>
<td>88M</td>
<td>18.5G</td>
<td>82.4 (+0.6)</td>
</tr>
</tbody>
</table>

*Drop path: 0.1 -> 0.6
## Empirical Results

Our method adheres to a **distinct scaling rule** compared to the vanilla ViT.

<table>
<thead>
<tr>
<th>Method</th>
<th>params</th>
<th>FLOPs</th>
<th>ImageNet Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeiT-S</td>
<td>22M</td>
<td>4.6G</td>
<td>79.9</td>
</tr>
<tr>
<td>SPFormer-S</td>
<td>22M</td>
<td>4.9G</td>
<td>81.0 (+1.1)</td>
</tr>
<tr>
<td>DeiT-S /32</td>
<td>22M</td>
<td>1.1G</td>
<td>73.3</td>
</tr>
<tr>
<td>SPFormer-S /32</td>
<td>22M</td>
<td>1.2G</td>
<td>76.1 (+2.8)</td>
</tr>
<tr>
<td>DeiT-Tiny</td>
<td>5M</td>
<td>1.3G</td>
<td>72.2</td>
</tr>
<tr>
<td>SPFormer-S /56</td>
<td>22M</td>
<td>0.5G</td>
<td>72.3</td>
</tr>
</tbody>
</table>
Harness Finer Details

Superpixel representation could benefit from detailed information

<table>
<thead>
<tr>
<th>Method</th>
<th>params</th>
<th>FLOPs</th>
<th>ImageNet Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeiT-S</td>
<td>22M</td>
<td>4.6G</td>
<td>79.9</td>
</tr>
<tr>
<td>DeiT-S 448</td>
<td>22M</td>
<td>4.6G</td>
<td>80.0 (+0.1)</td>
</tr>
<tr>
<td>SPFormer-S</td>
<td>22M</td>
<td>4.9G</td>
<td>81.0 (+1.1)</td>
</tr>
<tr>
<td>SPFormer-S 448</td>
<td>22M</td>
<td>4.9G</td>
<td>81.3 (+1.4)</td>
</tr>
</tbody>
</table>
Ablation

Our reformulation mitigates the aforementioned challenges

<table>
<thead>
<tr>
<th>Method</th>
<th>params</th>
<th>FLOPs</th>
<th>ImageNet Acc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPFormer-S /32</td>
<td>22M</td>
<td>1.2G</td>
<td>76.1</td>
</tr>
<tr>
<td>- multi iterations</td>
<td>22M</td>
<td>1.2G</td>
<td>75.4 (-0.7)</td>
</tr>
<tr>
<td>- multi stages</td>
<td>22M</td>
<td>1.2G</td>
<td>74.8 (-1.3)</td>
</tr>
<tr>
<td>- multi head</td>
<td>22M</td>
<td>1.2G</td>
<td>75.6 (-0.5)</td>
</tr>
</tbody>
</table>
Superpixel Transformer V2

We successfully learn meaningful superpixels using only category annotations.

Leveraging the superpixel representation, our method surpasses the performance of the standard vision transformer, offering improved efficiency, enhanced explainability, and increased robustness.
From Pixels to Objects: A Hierarchical Approach
Hierarchy Scene Understanding

**Huge redundancy** in areas like the sky
- Arises from the nature of superpixels as an **over-segmentation**
- **Merge similar superpixels** for further increasing the efficiency
  - Through another abstract level: **groups**
Multi-Level Representation

- Pixel -> Superpixel -> Group
What’s Next?
Can we get rid of Annotations?

Combine with MAE

**MAE couldn’t further scale up**
- Problem is not hard enough
- One can almost directly copy-paste the pixels to reconstruct the image.

Masking at the superpixel level compels the network **learn to do reasoning**
Supervoxel Transformer

Much more redundancy in 3D

- FLOPs saving: $(s^3)^2$ in 3D
- Combined with the hierarchical scene understanding, we may get **video segmentation & tracking, with or without annotations.**