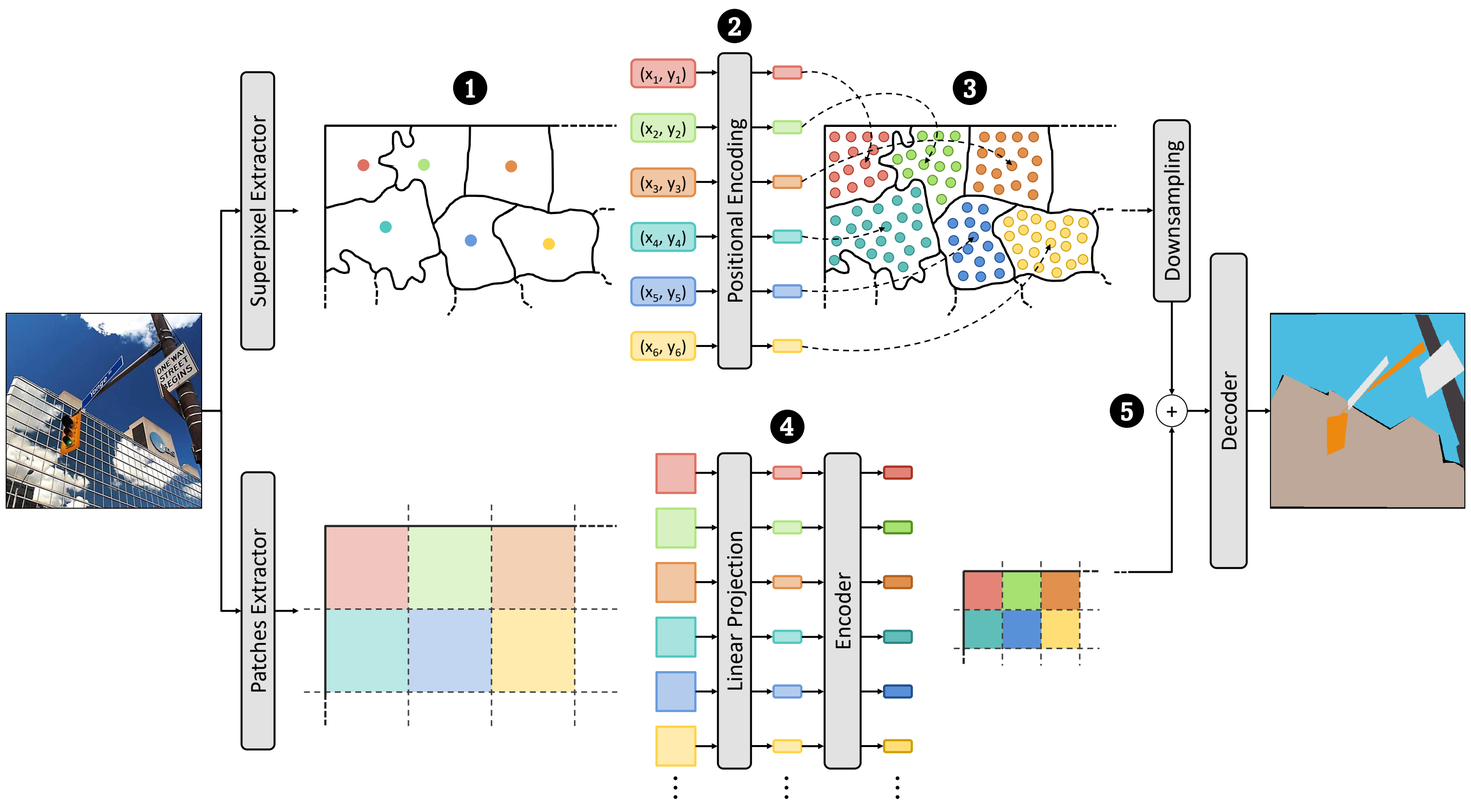
SUPERPIXEL POSITIONAL ENCODING TO IMPROVE VIT-BASED SEMANTIC SEGMENTATION MODELS

Roberto Amoroso[1] Matteo Tomei[2] Lorenzo Baraldi[1] Rita Cucchiara[1]
[1]University of Modena and Reggio Emilia, Italy [2]Prometeia, Italy
[1]{name.surname}@unimore.it [2]matteo.tomei@prometeia.com

Boosting Semantic Segmentation with Superpixel Shape and Position Priors

Recent ViT-based segmentation approaches adopt an encoder-decoder architecture, where the self-attentive encoder features are used to predict segmentation maps. In this work, we propose a **Superpixel Positional Encoding** (**PE**) that injects superpixel shape and position priors into the ViT encoder features, creating more boundary-aware semantic latent representations. Our approach is ***model-agnostic***, ***parameters-free***, ***plug-and-play*** and operates by computing superpixels over the input image. Our strategy improves segmentation performance while *mitigating overfitting*, ensuring a good balance between *generalization* and *specificity*.

***Superpixel-PE**: Injecting Superpixels Priors in ViT-based Architectures*



Superpixel-based positional encoding (*PE*):

❶ For a given input image, we extract a map of superpixels $\mathcal{L}$ along with their centroids $\mathcal{C}$ [5, 6];

❷ For each superpixel $\mathcal{L}_i$, we compute its position encoding PE_i with shape d_{model}. PE_i can be:
- ***absolute***: sinusoidal encoding of the coordinates (x, y) of the corresponding centroid $\mathcal{C}_i \rightarrow$ ***SinPE***
- ***relative***: progressive index i associated with the superpixel $\mathcal{L}_i$, normalized in $[0, 1] \rightarrow$ ***LinearPE***

❸ We replicate the positional encoding PE_i over every pixel of the superpixel $\mathcal{L}_i \rightarrow$ we obtain a superpixel-based positional encoding map $PE_{\mathcal{L}}$ with shape $H \times W \times d_{model}$;

❹ We extract N_p squared patches that we feed to the j^{th} layer of the ViT [1] encoder, obtaining a feature vector f^j;

❺ We downsample $PE_{\mathcal{L}}$ and sum to f^j, before feeding the result to the downstream segmentation decoder.

Superpixels Improve Semantic Segmentation Performance

	#Superpixel	Compact.	ADE20K (mIoU)	Cityscapes (mIoU)
DPT-B [4]	-	-	44.9	71.0
DPT-B+SinPE	16,000	20	45.4	71.7
DPT-B+LinearPE	28,000	10	**45.8**	**72.0**
SegFormer-B0 [2]	-	-	37.5	71.4
SegFormer-B0+SinPE	16,000	20	38.2	71.8
SegFormer-B0+LinearPE	28,000	10	**38.4**	**72.2**
SegFormer-B4 [2]	-	-	49.0	78.4
SegFormer-B4+SinPE	16,000	20	**49.3**	**78.6**
SegFormer-B4+LinearPE	28,000	20	**49.3**	**78.6**
SETR-T [3]	-	-	35.2	69.3
SETR-T+SinPE	8,192	10	**36.3**	**70.1**
SETR-T+LinearPE	16,384	10	36.1	70.0
SETR-S [3]	-	-	42.7	74.6
SETR-S+SinPE	8,192	10	43.0	74.9
SETR-S+LinearPE	8,192	10	**43.4**	**75.0**

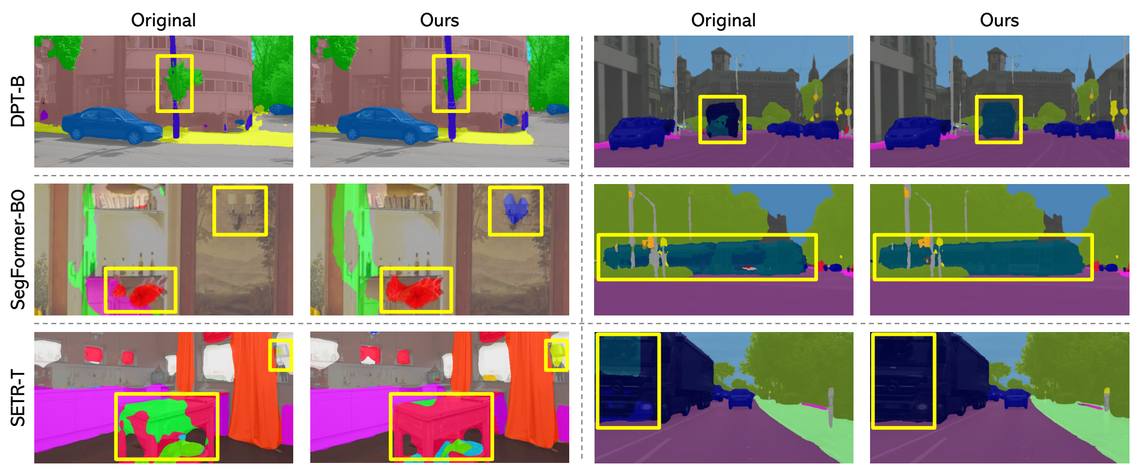
mIoU on ADE20K and Cityscapes using DPT-Base, SegFormer-B0, SegFormer-B4, SETR-Tiny, and SETR-Small with/without superpixel positional encoding.

Comparison with Alternative PE Strategies

	#Superpixel	Compact.	ADE20K (mIoU)	Cityscapes (mIoU)
SegFormer-B0 [2]	-	-	37.5	71.4
+PixelPE	-	-	37.8	71.5
+PatchPE-1	-	-	37.6	71.7
+PatchPE-4	-	-	37.5	71.7
+PatchPE-16	-	-	37.7	71.7
+WeightedPE	16,000	20	37.7	**72.4**
+LearnablePE	4,096	10	38.3	**72.2**
+SinPE	16,000	20	38.2	71.8
+LinearPE	28,000	10	**38.4**	**72.2**

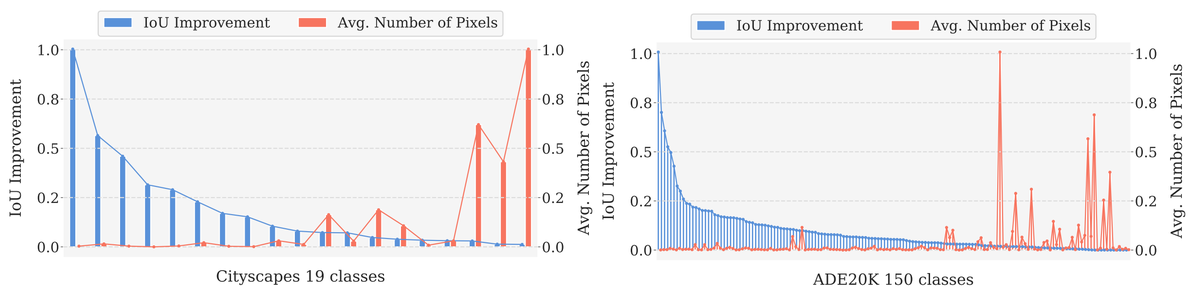
- **Pixel-PE**: sinusoidal positional encoding of individual pixel coordinates;
- **Patch-PE**: normalized numeric index that progressively identifies each squared patch;
- **Weighted-PE**: weighted sum of ViT features and superpixel-PE, with learnable weights;
- **Learnable-PE**: each superpixel is assigned a learnable vector, allowing the model to learn the optimal encoding during training (**additional training parameters**).

Qualitatively Improved Segmentation Masks



Qualitative comparison w./w.o. our superpixel positional encoding.

Superpixels Mitigate Overfitting



Our Superpixel-PE improves mIoU scores particularly on classes with low occurrence, such as `traffic light`, `flower`, and `sconce` in ADE20K and `bus`, and `train` in Cityscapes. This suggests that our approach is effective for infrequent classes and helps **avoid overfitting** for highly represented classes.

References

[1] Dosovitskiy A, et al. *An image is worth 16x16 words: Transformers for image recognition at scale*, In ICLR 2021
[2] Xie E, et al. *SegFormer: Simple and efficient design for semantic segmentation with transformers*, In NeurIPS 2021
[3] Zheng S, et al. *Rethinking semantic segmentation from a sequence-to-sequence perspective*, In CVPR 2021
[4] Ranftl R, et al. *Vision Transformers for Dense Prediction*, In ICCV 2021
[5] Achanta R, et al. *SLIC superpixels compared to state-of-the-art*, In TPAMI 2012
[6] Stutz D, et al. *Superpixels: An evaluation of the state-of-the-art*, In CVIU 2018