Universität Augsburg
Fakultät für Angewandte Informatik

Segformer++: Efficient Token-Merging Strategies for High-Resolution Semantic Segmentation

Daniel Kienzle, Marco Kantonis, Robin Schön, Rainer Lienhart

IEEE International Conference on Multimedia Information Processing and Retrieval 2024

August 7th 2024, San Jose, California, USA

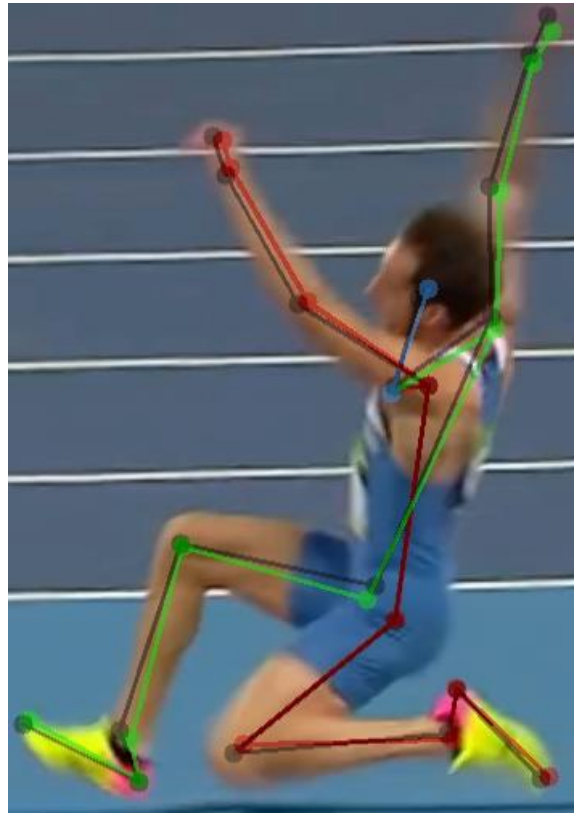
Motivation

Segformer++: Efficient **Token-Merging** Strategies for **High-Resolution** Semantic Segmentation

Core Ideas:

- **Reduction in number of tokens** using Token Merging
- Specialize in **dense pixel tasks** →SegFormer architecture
- Evaluation of **high-resolution performance**
- **Inference-only** possible → no need for finetuning

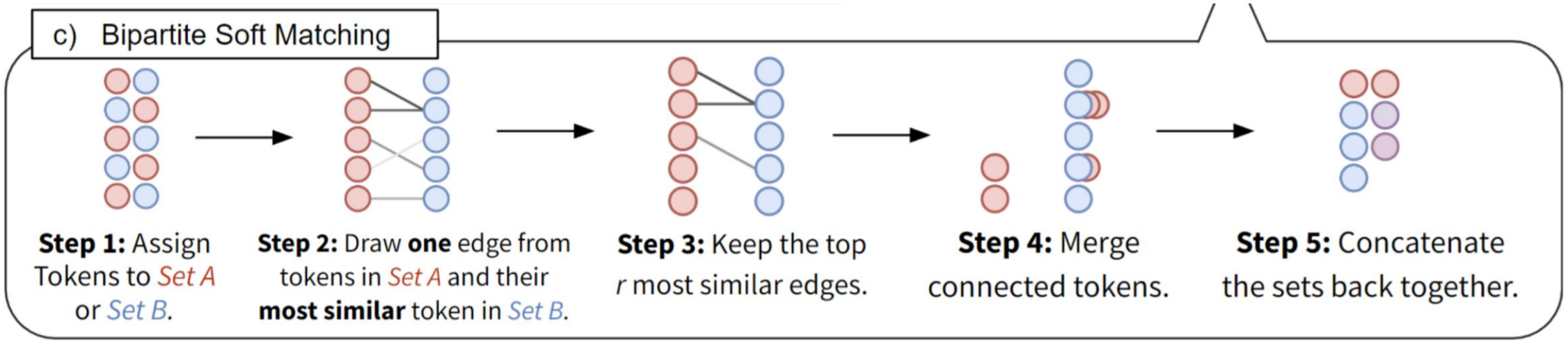
➔ Introduce Segformer++ architecture



Standard Token Merging

Reducing tokens in standard transformers

Bolya, D., et al, "Token Merging: Your ViT but Faster," in International Conference on Learning Representations (ICLR), 2023.



Gradually reduce the number of tokens by…

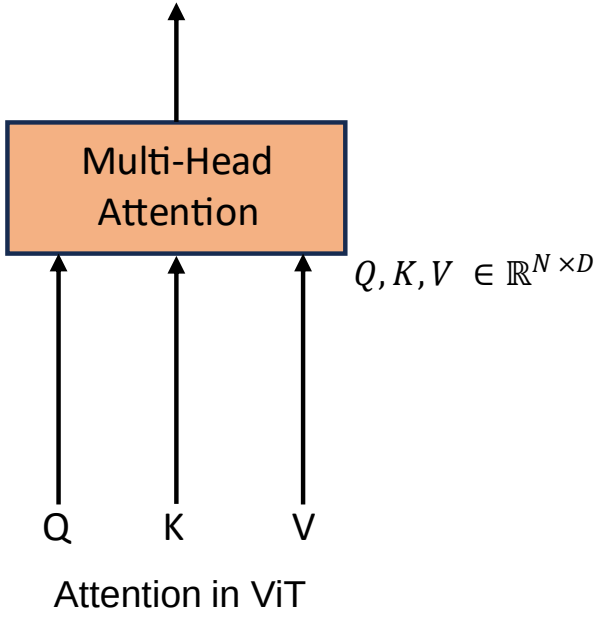
…merging a fixed quantity of tokens per stage

…merging the most similar tokens (Cosine Similarity)

SegFormer

Specialized transformer architecture for dense pixel tasks

Xie, E., et al, "SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers," in *Neural Information Processing Systems (NeurIPS)*, 2021.



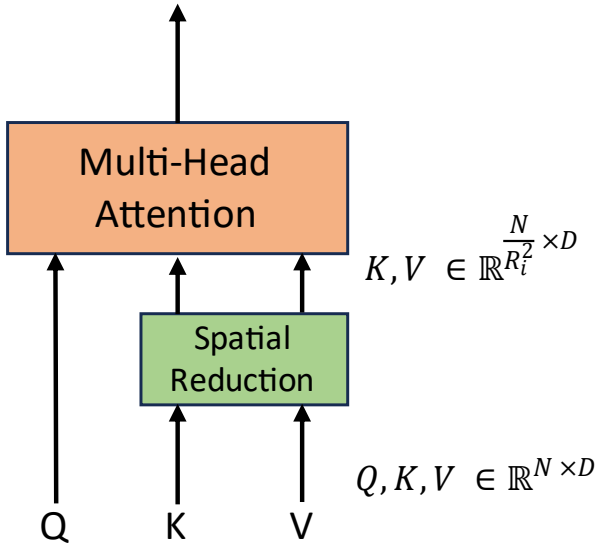
Attention in ViT

Major changes to ViT:

- **Overlapping 7x7 patches** instead of 16x16 patches
- **Pyramid Structure:** Reduce #token after each stage with strided convolution
- **Mix-FFN:** Convolution in FFN block
- **Spatial reduction attention:** Reduce #token before attention with strided convolution

➔ **Multiple convolutional layers are introduced**

➔ **Number of tokens is increased**



Attention in SegFormer

From Segformer to Segformer++

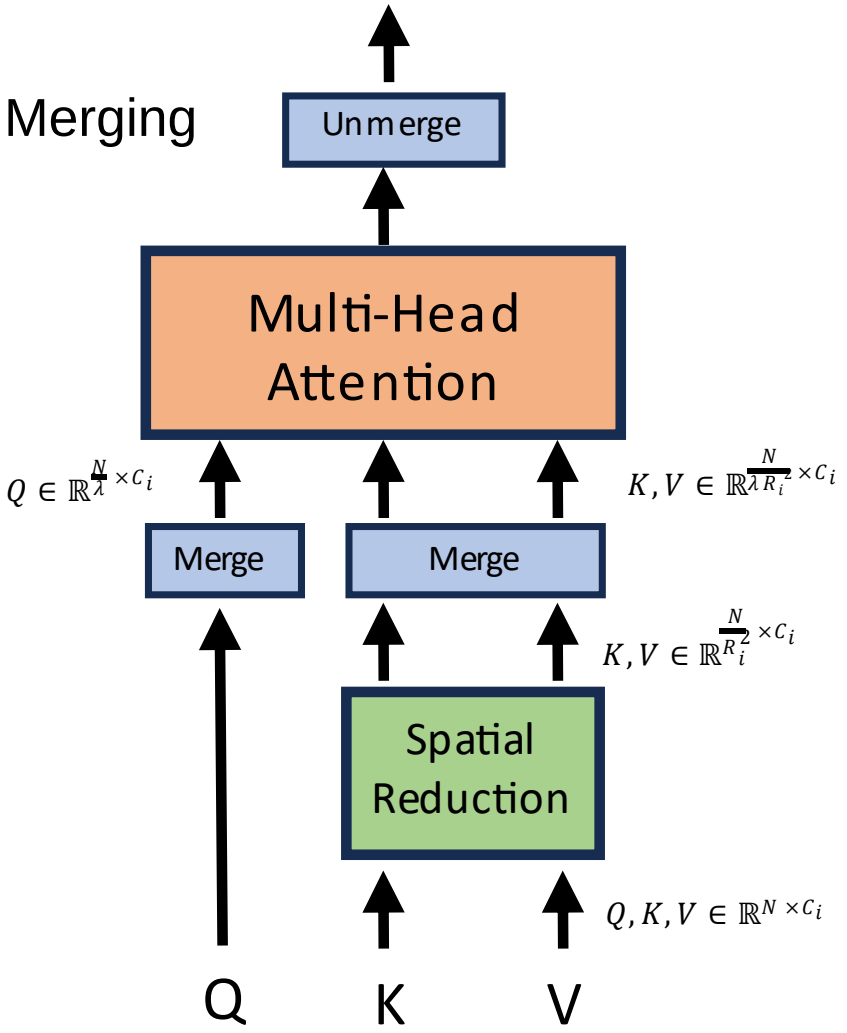
How to use Token Merging with convolutional layers

Idea: Further reduction in the number of token with Token Merging
→ Enable High Resolution Inputs

Convolutions depend on the 2D structure of the tokens.
Problem: Token Merging breaks the 2D structure.

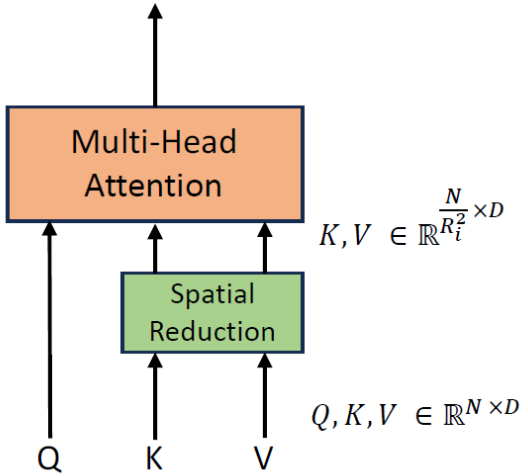
Solution:
- Merge directly before attention layer
- Unmerge (copy) directly after attention layer

→ Largely reducing the number of tokens by factor λ

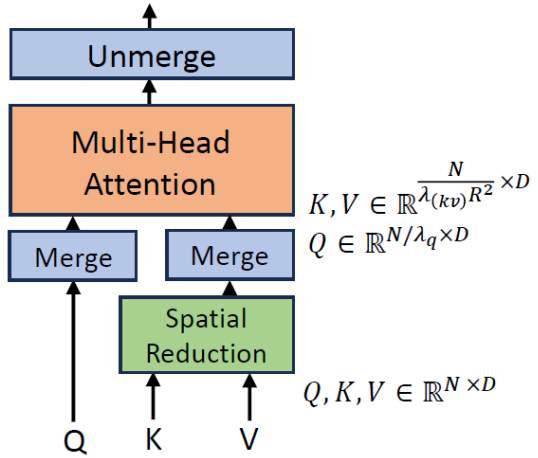


Overview

We compare multiple strategies



SegFormer



Segformer++

Introduce 2 versions:

- **Segformer++$_{HQ}$:** moderate efficiency gains + great performance
- **Segformer++$_{fast}$:** high efficiency gains + slightly worse performance

Inference-only possible:

→ Large pretrained models on cheap hardware

Results

Inference only & with training

Method	mIoU ↑	$mIou_{small}$ ↑	Speedup ↑
SegFormer	82.39	72.97	1.00
$Segformer++_{HQ}$	82.31	72.93	1.61
$Segformer++_{fast}$	82.04	72.24	1.94

Inference-only results on Cityscapes

Method	mIoU ↑	$mIoU_{small}$ ↑	Steps/s ↑	Memory (GB) ↓
SegFormer	82.39	72.97	0.80	48.30
$Segformer++_{HQ}$	82.19	72.77	1.12	33.95
$Segformer++_{fast}$	81.77	72.39	1.24	30.50

Results with training on Cityscapes

Inference-only:
- Significant speedups
- Great performance on small classes

With training:
- Good speedups + significantly lower memory requirements

→ **Nearly no performance losses** with our strategies

Results

High-resolution speedups

Method	512x512	640x640	1024x1024	2048x2048	3840x2160
Segformer++$_{HQ}$	1.18	1.32	1.61	2.04	2.66
Segformer++$_{fast}$	1.27	1.45	1.94	2.75	4.31

Inference speedups calculated on random tensors

➔ Impressive speedups on high-resolution images

Visualizations

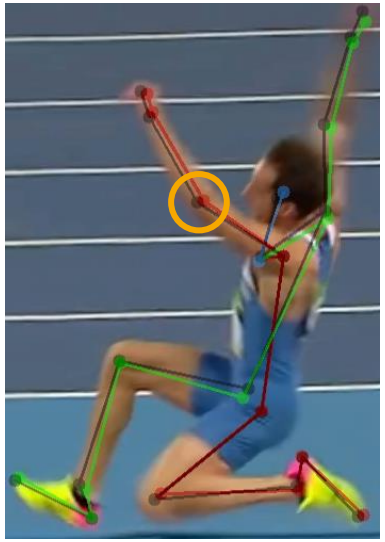
Semantic Segmentation + Human Pose Estimation

Segformer

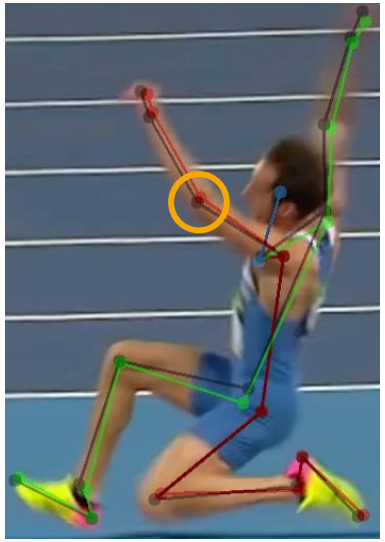
Segformer++$_{HQ}$

Segformer++$_{fast}$

Segformer++$_{HQ}$



Segformer++$_{fast}$



Conclusion

Segformer++: Efficient **Token-Merging** Strategies for **High-Resolution** Semantic Segmentation

➔ Largely **increasing efficiency**, especially for high resolution images
➔ Strategy can be applied at **training** or **inference only**
➔ Segformer++ preserves **small details**

Moreover:
- Applicable to **multiple dense pixel tasks**
- Applicable to **multiple architectures**