SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers

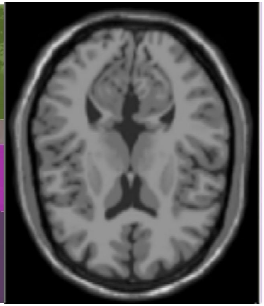
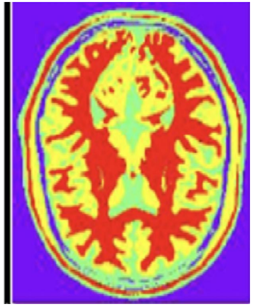
Authors: Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, Ping Luo

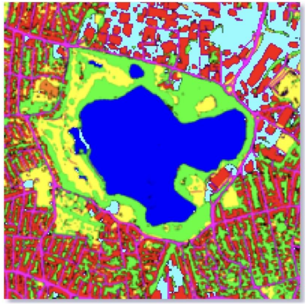
Submitted May 31, 2021; Last Revised October 28, 2021

Presented By: Alex Georgiev, David Zhang, Pan Lu

Motivation

- Provide a more lightweight solution to semantic segmentation
  - Group together things that have a similar meaning
  - Objects of the same class share the same segmentation mask
  - Applications: autonomous navigation, medical imaging, satellite imagery
- Investigate how altering the encoder *and* decoder of a transformer architecture can affect its performance

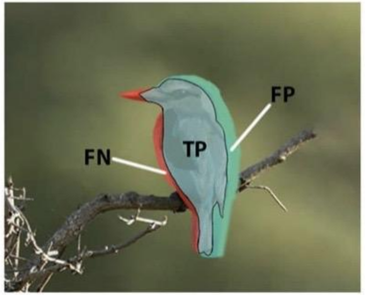


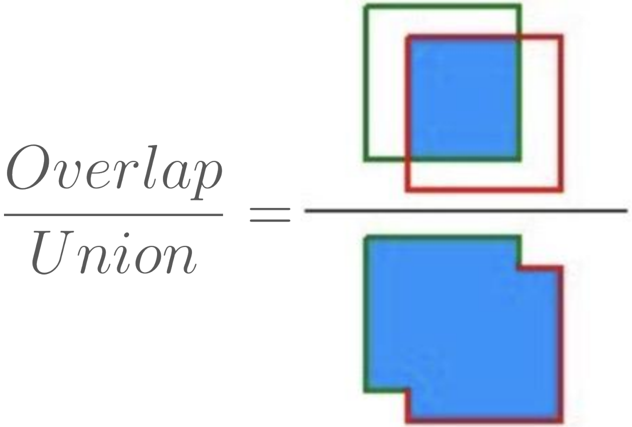
Metrics

- FLOPs - floating point operations
  - Since Mask2Former and SegFormer both use Nvidia V100s, lower FLOPs → less computational overhead
- mIoU - mean intersection over union
  - Measures overlap between predicted segmentation mask and the ground truth mask
    - 0% → no overlap; 100% → perfect overlap
  - Averages intersection over union across all classes

Ground Truth Mask

Predicted Mask





Weaknesses of Prior Work

- Computationally demanding and inefficient
  - Complex decoders with high parameter counts
  - Cannot be deployed for real-time applications
- Only alter the design of the transformer encoder and neglect the decoder as an avenue for improved performance
- Positional encoding can lead to decreased performance when the testing resolution differs from the training resolution

Key Contributions

- Novel architectural components:
  - Hierarchical encoder that doesn’t rely on positional encoding
  - Lightweight all-MLP (Multilayer Perceptron) decoder design
    - Much smaller parameter count
  - Efficient self-attention with reduced computational complexity
- Model is robust to noise, blurs, weather effects, and digital corruptions like JPEG compression
- Small variant designed for real-time applications

Demo from NVIDIA

Methods

Methods: The Model

- Transformer encoder (MiT)
  - Multi-level features
- All-MLP decoder
  - Segmentation mask

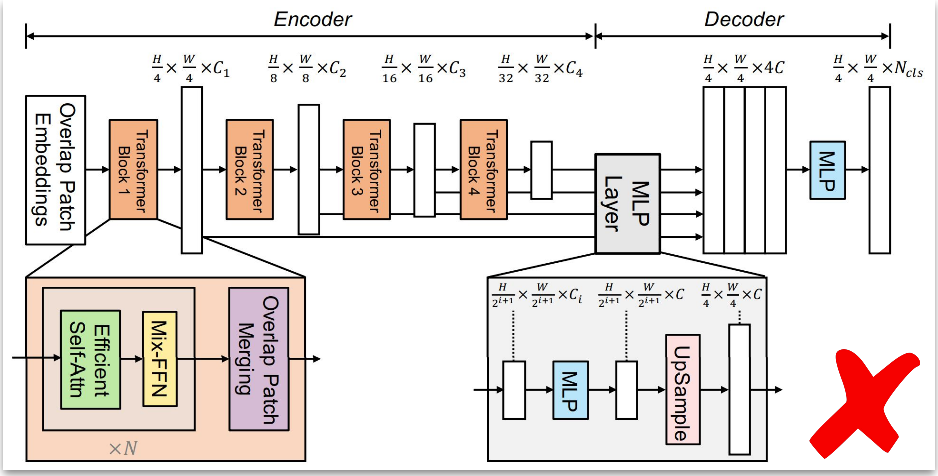
$$H \times W \times 3 \text{ Image}$$

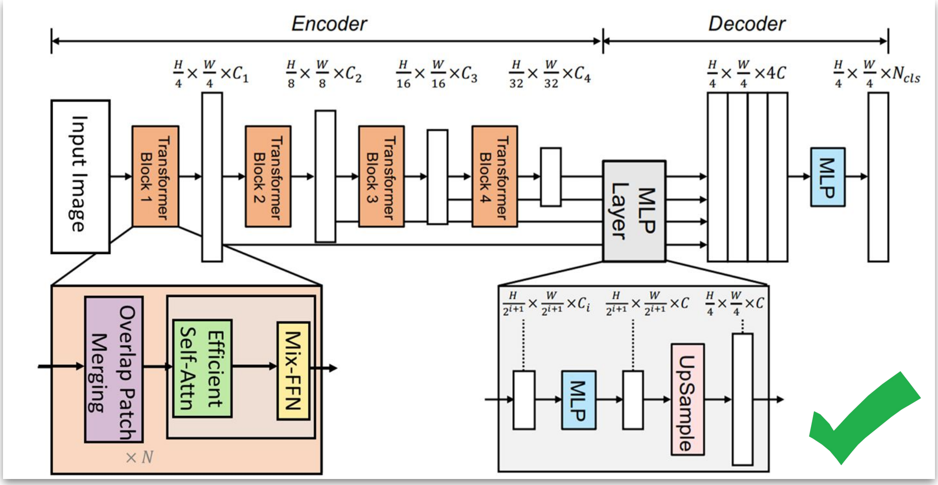
$$\downarrow$$

$$\text{SegFormer}$$

$$\downarrow$$

$$\frac{H}{4} \times \frac{W}{4} \times N_{cls} \text{ Output}$$

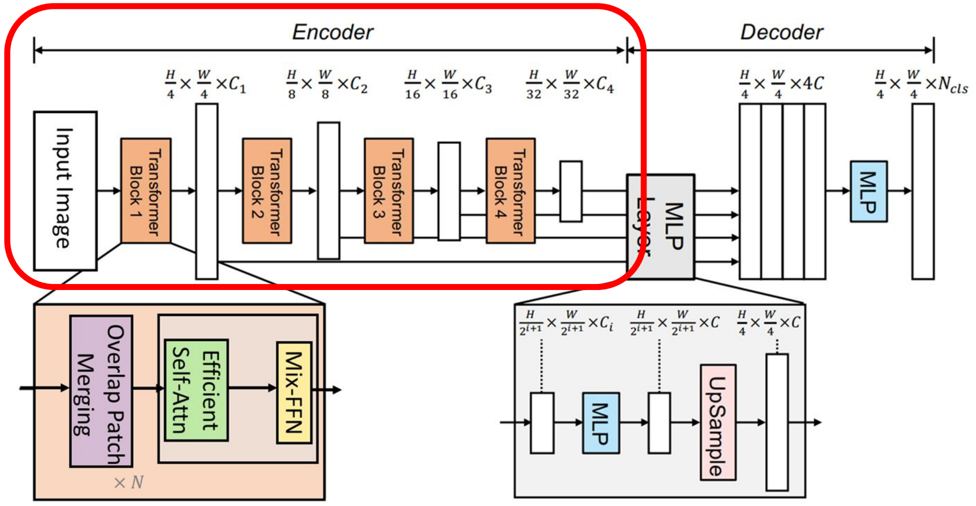


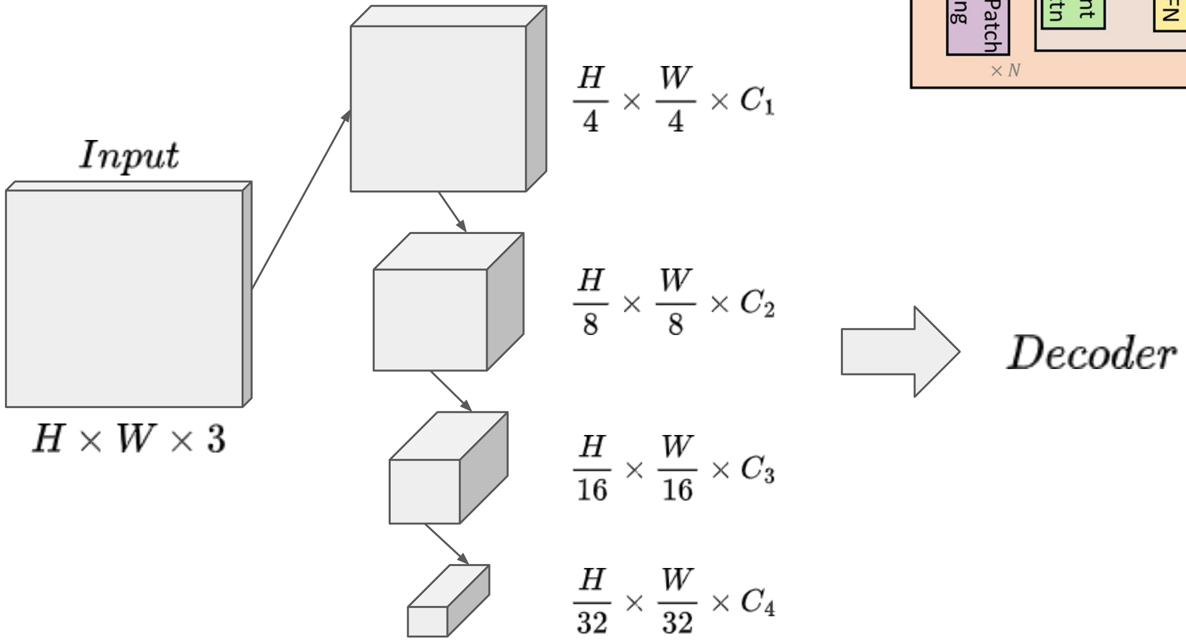


@Jayden9912

Methods: The Encoder

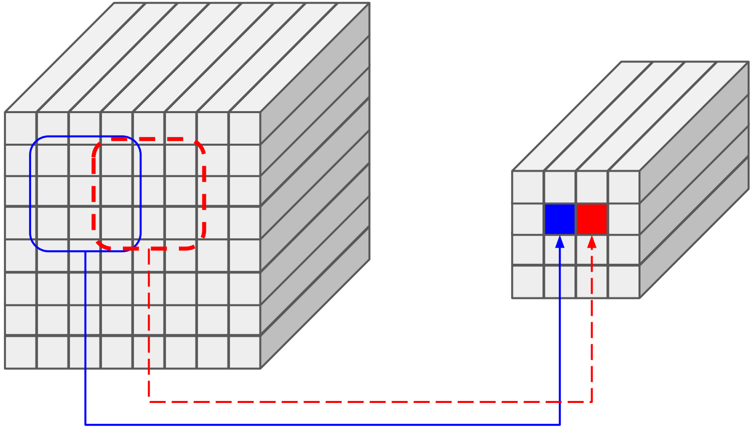
- Hierarchical feature representation

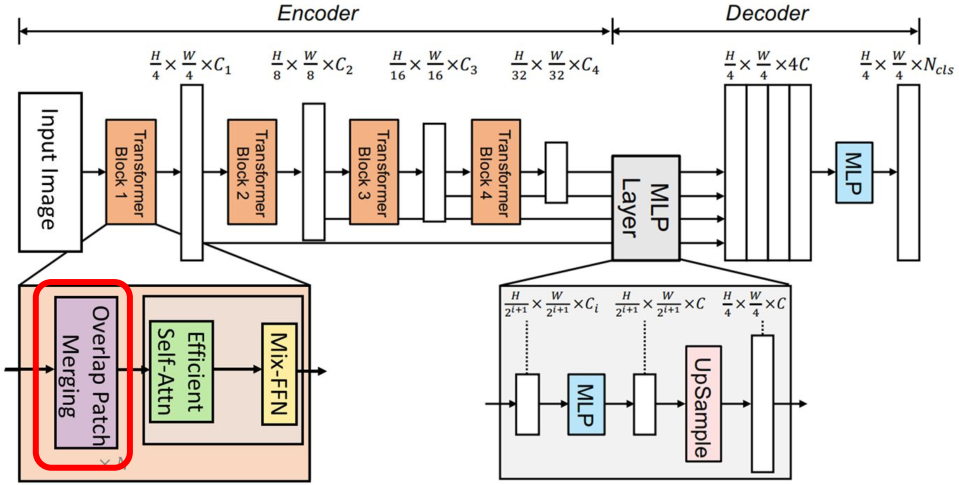


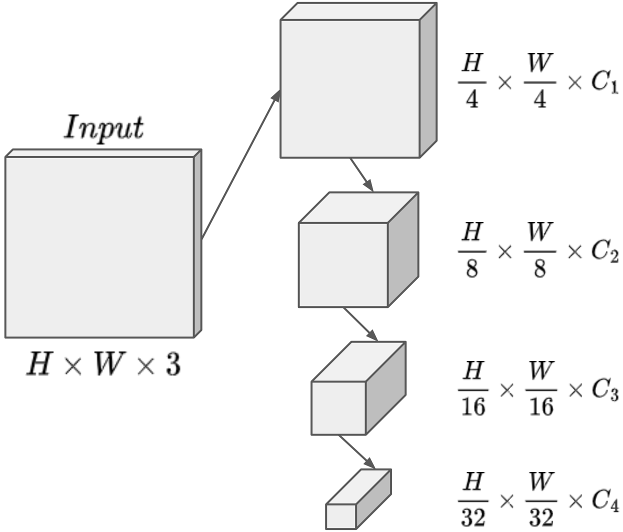


Methods: Overlapped Patch Merging

- 4 × 4 patches from input image
  - K=7, S=4, P=3
- Downsample by 2 between transformer blocks
  - K=3, S=2, P=1

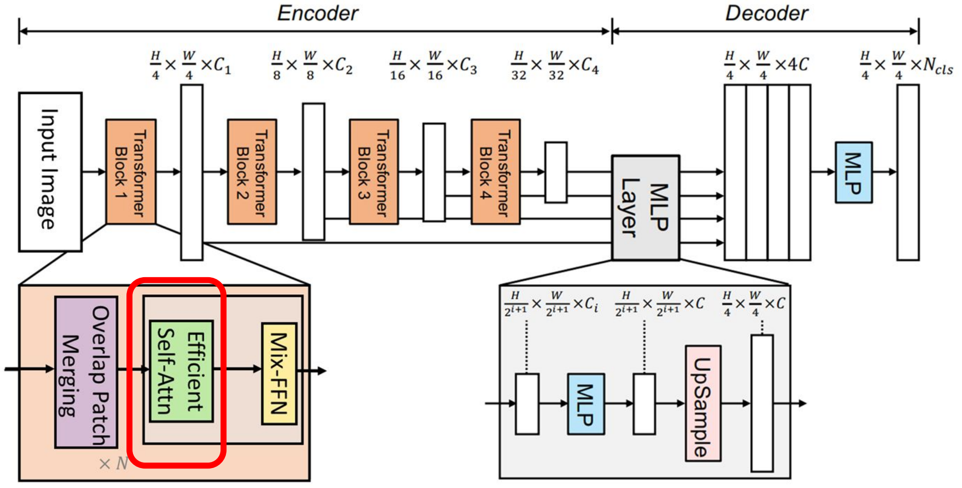


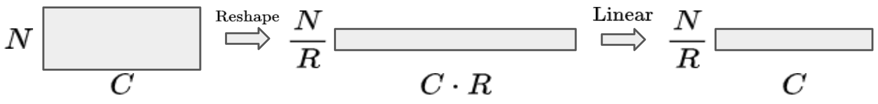




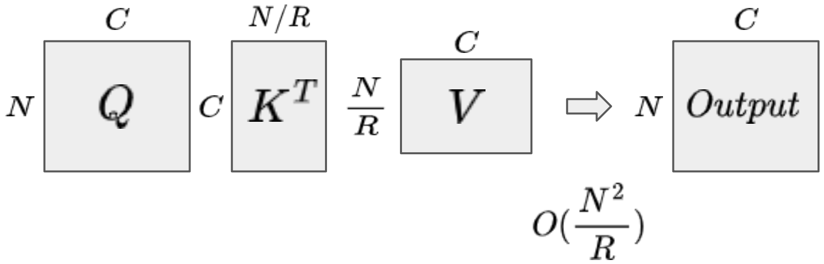
Methods: Efficient Self-Attention

- Use sequence reduction to reduce the length of Key (K) and Value (V) by a factor of R





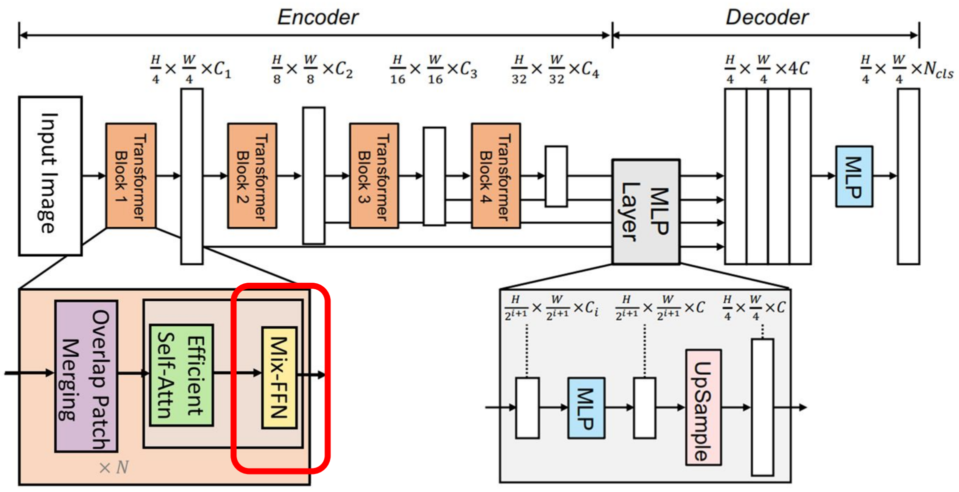
$$\text{Attention}(Q,K,V) = \text{Softmax}\left(\frac{QK^{\mathsf{T}}}{\sqrt{d_{head}}}\right)V.$$

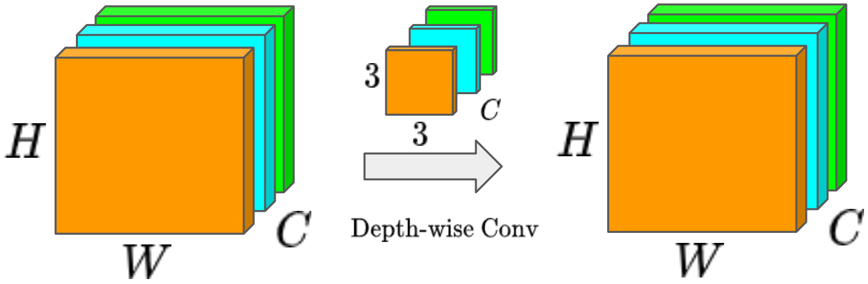


Methods: Mix-FFN

- Use convolution to provide positional information instead of positional encoding
- Avoid interpolating PE which is bad
- Conv with zero-padding leak location information*

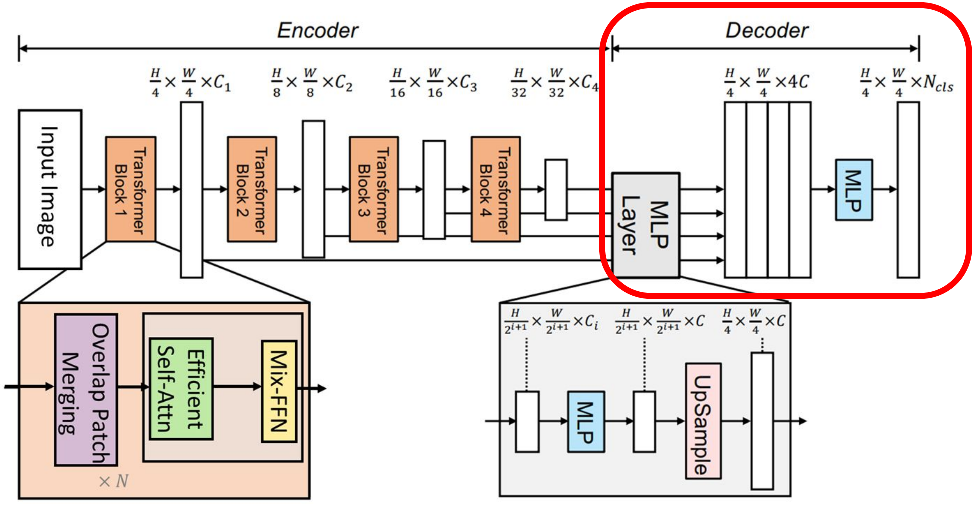
$$\mathbf{x}_{out} = \text{MLP}(\text{GELU}(\text{Conv}_{3\times3}(\text{MLP}(\mathbf{x}_{in})))) + \mathbf{x}_{in},$$





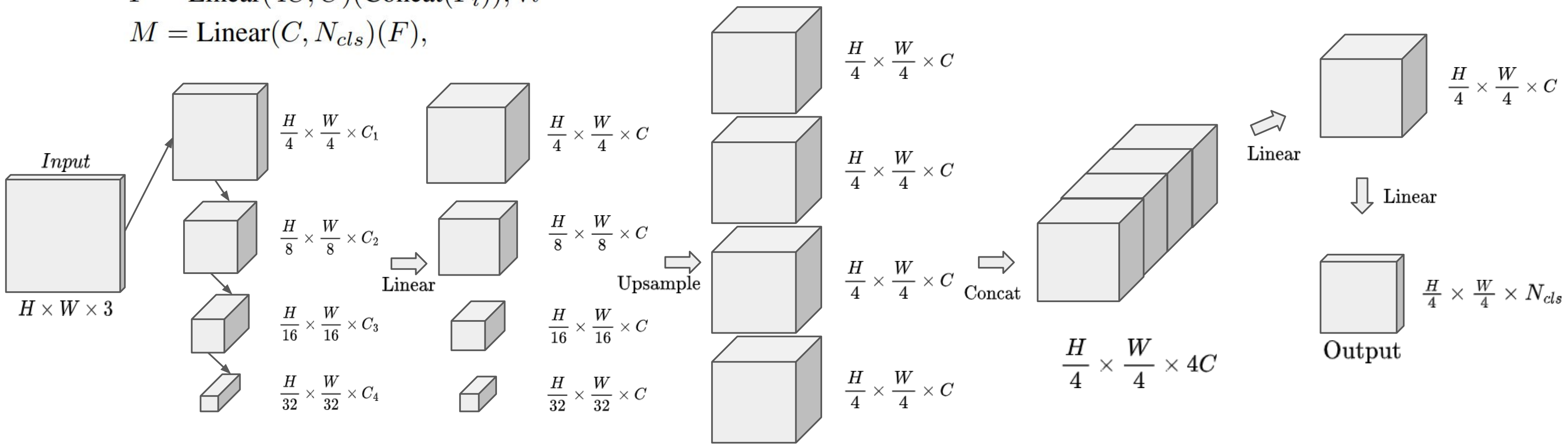
*Md Amirul Islam, Sen Jia, and Neil DB Bruce. How much position information do convolutional neural networks encode? arXiv, 2020

Methods: All-MLP Decoder



- Lightweight decoder that unifies the features from the encoder and produces the output

$$\hat{F}_i = \text{Linear}(C_i, C)(F_i), \forall i$$
$$\hat{F}_i = \text{Upsample}(\frac{W}{4} \times \frac{W}{4})(\hat{F}_i), \forall i$$
$$F = \text{Linear}(4C, C)(\text{Concat}(\hat{F}_i)), \forall i$$
$$M = \text{Linear}(C, N_{cls})(F),$$



Results

Experimental Setup

Datasets: Cityscapes, ADE20K, and COCOStuff

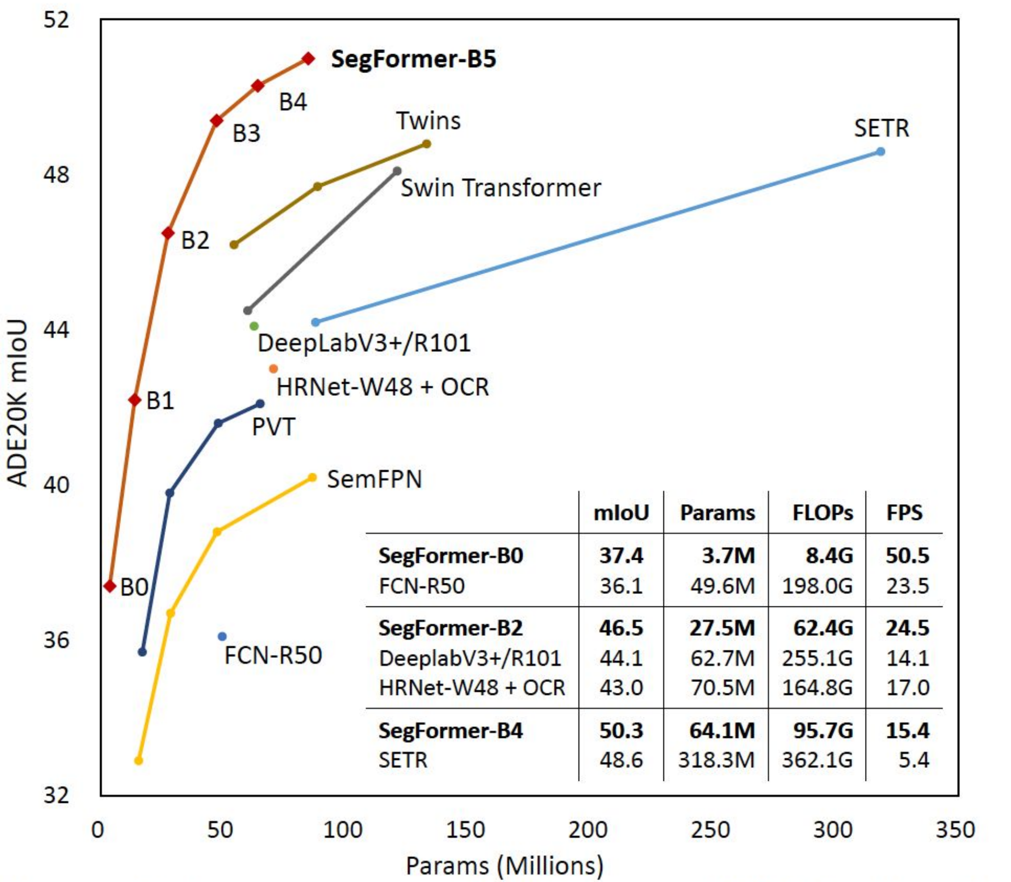
Implementation details:

- 8 Tesla V100
- Encoder is pretrained on the Imagenet-1K dataset
- Decoder is initially randomized
- Models are trained using AdamW optimizer
- The learning rate was set to an initial value of 0.00006
- “Poly” LR schedule with factor 1.0

Ablation Studies

Influence of the size of model

- Increasing the size of the encoder yields consistent mIoU improvements
- Lower parameter count and higher FPS compared to prior work



	mIoU	Params	FLOPs	FPS
SegFormer-B0	**37.4**	**3.7M**	**8.4G**	**50.5**
FCN-R50	36.1	49.6M	198.0G	23.5
SegFormer-B2	**46.5**	**27.5M**	**62.4G**	**24.5**
DeeplabV3+/R101	44.1	62.7M	255.1G	14.1
HRNet-W48 + OCR	43.0	70.5M	164.8G	17.0
SegFormer-B4	**50.3**	**64.1M**	**95.7G**	**15.4**
SETR	48.6	318.3M	362.1G	5.4

Figure 1: **Performance *vs.* model efficiency on ADE20K.** All results are reported with single model and single-scale inference. SegFormer achieves a new state-of-the-art 51.0% mIoU while being significantly more efficient than previous methods.

Influence of C, the MLP decoder channel dimension

- C = 256 provides a very competitive performance and computational cost.
- Bigger C leads to larger and less efficient models.

(b) Accuracy as a function of the MLP dimension C in the decoder on ADE20K.

C	Flops $\downarrow$	Params $\downarrow$	mIoU $\uparrow$
256	25.7	24.7	44.9
512	39.8	25.8	45.0
768	62.4	27.5	45.4
1024	93.6	29.6	45.2
2048	304.4	43.4	45.6

Mix-FFN vs. Positional Encoder (PE)

- Mix-FFN outperforms

positional encoding

- Mix-FFN is less sensitive

to differences in the test resolution

(c) Mix-FFN vs. positional encoding (PE) for different test resolution on Cityscapes.

Inf Res	Enc Type	mIoU ↑
768×768	PE	77.3
1024×2048	PE	74.0
768×768	Mix-FFN	80.5
1024×2048	Mix-FFN	79.8

Effective receptive field evaluation

- Coupling proposed Transformer encoder with the MLP decoder leads to the best performance

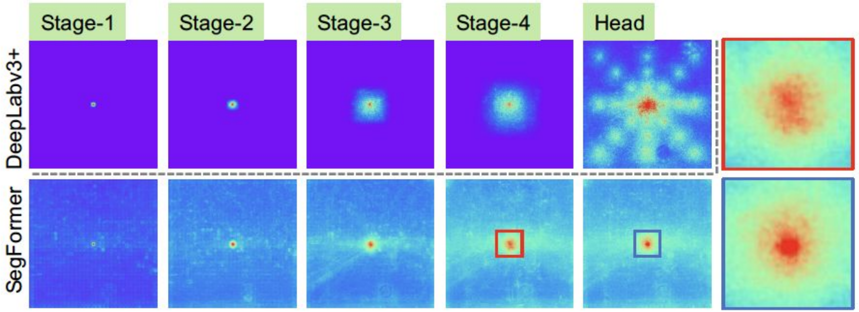


Figure 3: **Effective Receptive Field (ERF) on Cityscapes** (average over 100 images). Top row: Deeplabv3+. Bottom row: SegFormer. ERFs of the four stages and the decoder heads of both architectures are visualized. Best viewed with zoom in.

(d) Accuracy on ADE20K of CNN and Transformer encoder with MLP decoder. "S4" means stage-4 feature.

Encoder	Flops ↓	Params ↓	mIoU ↑
ResNet50 (S1-4)	69.2	29.0	34.7
ResNet101 (S1-4)	88.7	47.9	38.7
ResNeXt101 (S1-4)	127.5	86.8	39.8
MiT-B2 (S4)	**22.3**	**24.7**	43.1
MiT-B2 (S1-4)	62.4	27.7	45.4
MiT-B3 (S1-4)	79.0	47.3	**48.6**

Comparison to state of the art methods

Table 2: **Comparison to state of the art methods on ADE20K and Cityscapes.** SegFormer has significant advantages on #Params, #Flops, #Speed and #Accuracy. Note that for SegFormer-B0 we scale the short side of image to {1024, 768, 640, 512} to get speed-accuracy tradeoffs.

	Method	Encoder	Params ↓	ADE20K			Cityscapes		
				Flops ↓	FPS ↑	mIoU ↑	Flops ↓	FPS ↑	mIoU ↑
Real-Time	FCN [1]	MobileNetV2	9.8	39.6	64.4	19.7	317.1	14.2	61.5
	ICNet [11]	-	-	-	-	-	-	30.3	67.7
	PSPNet [17]	MobileNetV2	13.7	52.9	57.7	29.6	423.4	11.2	70.2
	DeepLabV3+ [20]	MobileNetV2	15.4	69.4	43.1	34.0	555.4	8.4	75.2
	SegFormer (Ours)	MiT-B0	**3.8**	**8.4**	**50.5**	**37.4**	125.5	15.2	**76.2**
				-	-	-	51.7	26.3	75.3
				-	-	-	31.5	37.1	73.7
				-	-	-	**17.7**	**47.6**	71.9
Non Real-Time	FCN [1]	ResNet-101	68.6	275.7	14.8	41.4	2203.3	1.2	76.6
	EncNet [24]	ResNet-101	**55.1**	218.8	14.9	44.7	1748.0	1.3	76.9
	PSPNet [17]	ResNet-101	68.1	256.4	15.3	44.4	2048.9	1.2	78.5
	CCNet [41]	ResNet-101	68.9	278.4	14.1	45.2	2224.8	1.0	80.2
	DeeplabV3+ [20]	ResNet-101	62.7	255.1	14.1	44.1	2032.3	1.2	80.9
	OCRNet [23]	HRNet-W48	70.5	164.8	**17.0**	45.6	1296.8	**4.2**	81.1
	GSCNN [35]	WideResNet38	-	-	-	-	-	-	80.8
	Axial-DeepLab [74]	AxialResNet-XL	-	-	-	-	2446.8	-	81.1
	Dynamic Routing [75]	Dynamic-L33-PSP	-	-	-	-	**270.0**	-	80.7
	Auto-Deeplab [50]	NAS-F48-ASPP	-	-	-	44.0	695.0	-	80.3
	SETR [7]	ViT-Large	318.3	-	5.4	50.2	-	0.5	82.2
	SegFormer (Ours)	MiT-B4	64.1	**95.7**	15.4	51.1	1240.6	3.0	83.8
	SegFormer (Ours)	MiT-B5	84.7	183.3	9.8	**51.8**	1447.6	2.5	**84.0**

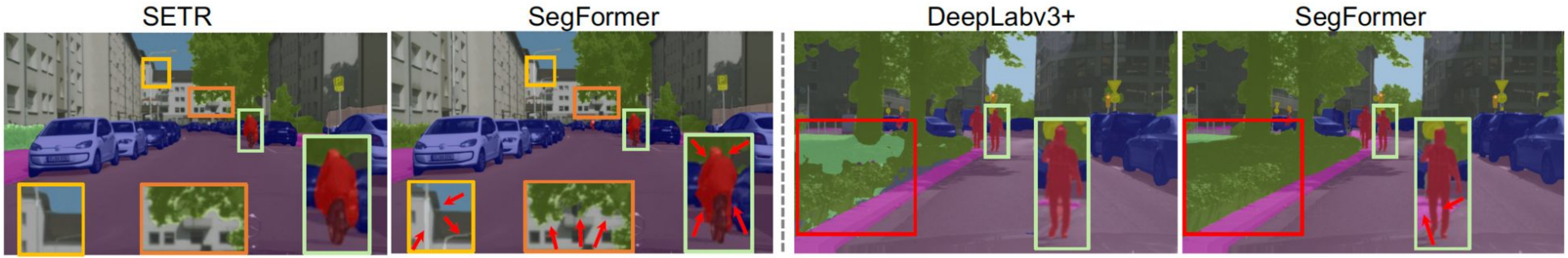


Figure 4: **Qualitative results on Cityscapes.** Compared to SETR, our SegFormer predicts masks with substantially finer details near object boundaries. Compared to DeeplabV3+, SegFormer reduces long-range errors as highlighted in red. Best viewed in screen.

Table 3: **Comparison to state of the art methods on Cityscapes test set.** IM-1K, IM-22K, Coarse and MV refer to the ImageNet-1K, ImageNet-22K, Cityscapes coarse set and Mapillary Vistas. SegFormer outperforms the compared methods with equal or less extra data.

Method	Encoder	Extra Data	mIoU
PSPNet [17]	ResNet-101	IM-1K	78.4
PSANet [43]	ResNet-101	IM-1K	80.1
CCNet [41]	ResNet-101	IM-1K	81.9
OCNet [21]	ResNet-101	IM-1K	80.1
Axial-DeepLab [74]	AxiaiResNet-XL	IM-1K	79.9
SETR [7]	ViT	IM-22K	81.0
SETR [7]	ViT	IM-22K, Coarse	81.6
SegFormer	MiT-B5	IM-1K	82.2
SegFormer	MiT-B5	IM-1K, MV	**83.1**

- Pre-training on Mapillary Vistas and Imagenet-1k produces new state-of-the-art result of 83.1% mIoU

Table 4: **Results on COCO-Stuff full dataset** containing all 164K images from COCO 2017 and covers 172 classes.

Method	Encoder	Params	mIoU
DeeplabV3+ [20]	ResNet50	**43.7**	38.4
OCRNet [23]	HRNet-W48	70.5	42.3
SETR [7]	ViT	305.7	45.8
SegFormer	MiT-B5	84.7	**46.7**

- SegFormer-B5 reaches 46.7% mIoU with only 84.7M parameters, which is 0.9% better and 4 smaller than SETR.

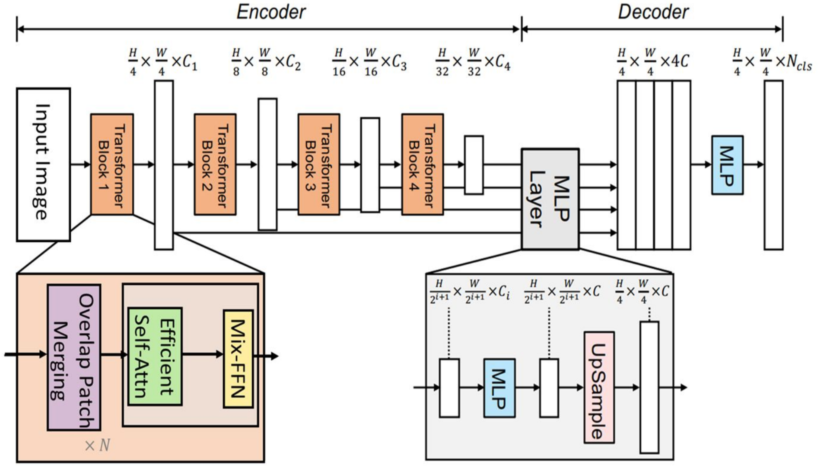
Robustness to natural corruptions

Table 5: **Main results on Cityscapes-C.** "DLv3+", "MBv2", "R" and "X" refer to DeepLabv3+, MobileNetv2, ResNet and Xception. The mIoUs of compared methods are reported from [77].

Method	Clean	Blur				Noise				Digital				Weather			
		Motion	Defoc	Glass	Gauss	Gauss	Impul	Shot	Speck	Bright	Contr	Satur	JPEG	Snow	Spatt	Fog	Frost
DLv3+ (MBv2)	72.0	53.5	49.0	45.3	49.1	6.4	7.0	6.6	16.6	51.7	46.7	32.4	27.2	13.7	38.9	47.4	17.3
DLv3+ (R50)	76.6	58.5	56.6	47.2	57.7	6.5	7.2	10.0	31.1	58.2	54.7	41.3	27.4	12.0	42.0	55.9	22.8
DLv3+ (R101)	77.1	59.1	56.3	47.7	57.3	13.2	13.9	16.3	36.9	59.2	54.5	41.5	37.4	11.9	47.8	55.1	22.7
DLv3+ (X41)	77.8	61.6	54.9	51.0	54.7	17.0	17.3	21.6	43.7	63.6	56.9	51.7	38.5	18.2	46.6	57.6	20.6
DLv3+ (X65)	78.4	63.9	59.1	52.8	59.2	15.0	10.6	19.8	42.4	65.9	59.1	46.1	31.4	19.3	50.7	63.6	23.8
DLv3+ (X71)	78.6	64.1	60.9	52.0	60.4	14.9	10.8	19.4	41.2	68.0	58.7	47.1	40.2	18.8	50.4	64.1	20.2
ICNet	65.9	45.8	44.6	47.4	44.7	8.4	8.4	10.6	27.9	41.0	33.1	27.5	34.0	6.3	30.5	27.3	11.0
FCN8s	66.7	42.7	31.1	37.0	34.1	6.7	5.7	7.8	24.9	53.3	39.0	36.0	21.2	11.3	31.6	37.6	19.7
DilatedNet	68.6	44.4	36.3	32.5	38.4	15.6	14.0	18.4	32.7	52.7	32.6	38.1	29.1	12.5	32.3	34.7	19.2
ResNet-38	77.5	54.6	45.1	43.3	47.2	13.7	16.0	18.2	38.3	60.0	50.6	46.9	14.7	13.5	45.9	52.9	22.2
PSPNet	78.8	59.8	53.2	44.4	53.9	11.0	15.4	15.4	34.2	60.4	51.8	30.6	21.4	8.4	42.7	34.4	16.2
GSCNN	80.9	58.9	58.4	41.9	60.1	5.5	2.6	6.8	24.7	75.9	61.9	70.7	12.0	12.4	47.3	67.9	32.6
SegFormer-B5	**82.4**	**69.1**	**68.6**	**64.1**	**69.8**	**57.8**	**63.4**	**52.3**	**72.8**	**81.0**	**77.7**	**80.1**	**58.8**	**40.7**	**68.4**	**78.5**	**49.9**

Summary

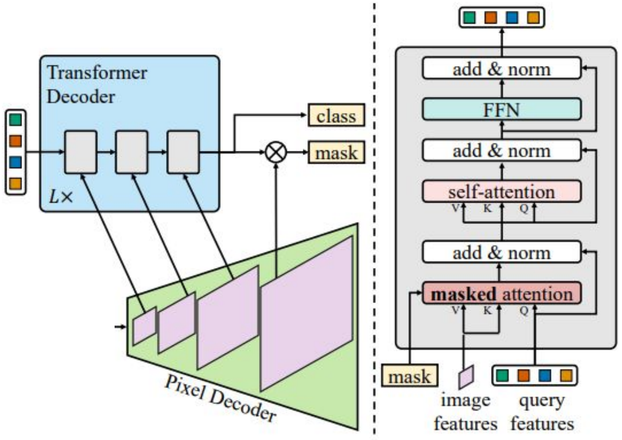
- Simple, efficient and effective design
- Positional-encoding-free hierarchical encoder captures high-resolution fine features and low-resolution coarse features
- Lightweight All-MLP decoder



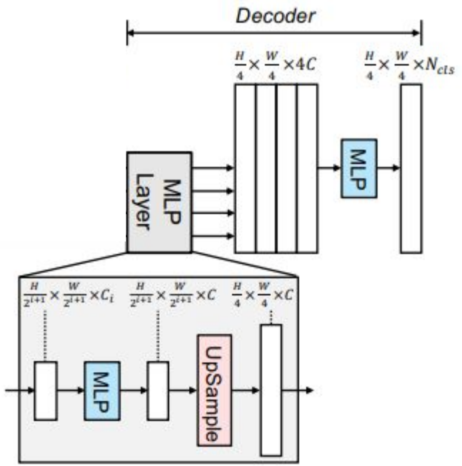
Arguments

Simple and lightweight design

- No widely-used tricks, such as auxiliary losses
- No positional encoding, so no interpolation when dealing with higher resolution images
- Lightweight decoder only has at most 3.3M parameters whereas theirs has ~20M
  - Our decoder only consist of MLP layers while theirs uses a transformer



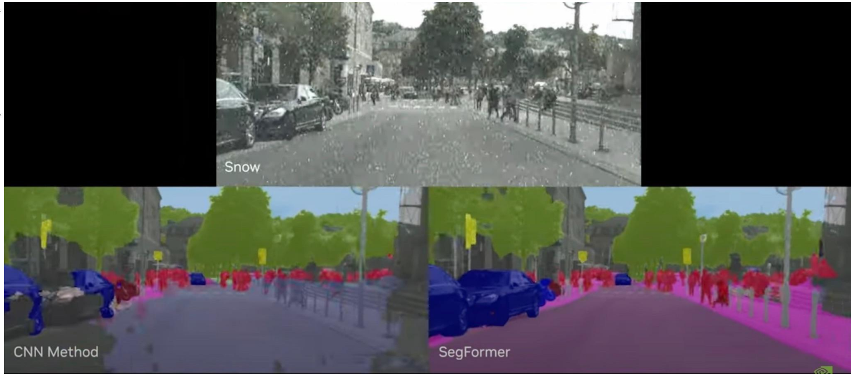
vs



Can be used for latency-critical real-time applications

- Our B0 model achieves a high mIoU and high FPS with a much lower number of FLOPS and only 3.8M parameters
- Robust to common corruptions such as weather conditions

	Method	Encoder	Params ↓	ADE20K			Cityscapes		
				Flops ↓	FPS ↑	mIoU ↑	Flops ↓	FPS ↑	mIoU ↑
Real-Time	FCN [1]	MobileNetV2	9.8	39.6	64.4	19.7	317.1	14.2	61.5
	ICNet [11]	-	-	-	-	-	-	30.3	67.7
	PSPNet [17]	MobileNetV2	13.7	52.9	57.7	29.6	423.4	11.2	70.2
	DeepLabV3+ [20]	MobileNetV2	15.4	69.4	43.1	34.0	555.4	8.4	75.2
	SegFormer (Ours)	MiT-B0	**3.8**	**8.4**	**50.5**	**37.4**	125.5	15.2	**76.2**
				-	-	-	51.7	26.3	75.3
				-	-	-	31.5	37.1	73.7
				-	-	-	**17.7**	**47.6**	71.9



Comparable performance despite earlier publication

Table 1: Ablation studies related to model size, encoder and decoder design.

(a) Accuracy, parameters and flops as a function of the model size on the three datasets. "SS" and "MS" means single/multi-scale test.

Encoder Model Size	Params Encoder	Params Decoder	ADE20K Flops ↓	ADE20K mIoU(SS/MS) ↑	Cityscapes Flops ↓	Cityscapes mIoU(SS/MS) ↑	COCO-Stuff Flops ↓	COCO-Stuff mIoU(SS) ↑
MiT-B0	3.4	0.4	8.4	37.4 / 38.0	125.5	76.2 / 78.1	8.4	35.6
MiT-B1	13.1	0.6	15.9	42.2 / 43.1	243.7	78.5 / 80.0	15.9	40.2
MiT-B2	24.2	3.3	62.4	46.5 / 47.5	717.1	81.0 / 82.2	62.4	44.6
MiT-B3	44.0	3.3	79.0	49.4 / 50.0	962.9	81.7 / 83.3	79.0	45.5
MiT-B4	60.8	3.3	95.7	50.3 / 51.1	1240.6	82.3 / 83.9	95.7	46.5
MiT-B5	81.4	3.3	183.3	51.0 / 51.8	1460.4	82.4 / 84.0	111.6	46.7

method	backbone	panoptic model PQ (s.s.)	PQ (m.s.)	AP^{Th}_{pan}	$mIoU_{pan}$	instance model AP	AP50	semantic model mIoU (s.s.)	mIoU (m.s.)
Panoptic-DeepLab [11]	R50	60.3	-	32.1	78.7	-	-	-	-
	X71 [15]	63.0	64.1	35.3	80.5	-	-	-	-
	SWideRNet [9]	66.4	67.5	40.1	82.2	-	-	-	-
Panoptic FCN [31]	Swin-L†	65.9	-	-	-	-	-	-	-
Segmenter [45]	ViT-L†	-	-	-	-	-	-	-	81.3
SETR [64]	ViT-L†	-	-	-	-	-	-	-	82.2
SegFormer [59]	MiT-B5	-	-	-	-	-	-	-	84.0
Mask2Former (ours)	R50	62.1	-	37.3	77.5	37.4	61.9	79.4	82.2
	R101	62.4	-	37.7	78.6	38.5	63.9	80.1	81.9
	Swin-T	63.9	-	39.1	80.5	39.7	66.9	82.1	83.0
	Swin-S	64.8	-	40.7	81.8	41.8	70.4	82.6	83.6
	Swin-B†	66.1	-	42.8	82.7	42.0	68.8	**83.3**	**84.5**
	Swin-L†	**66.6**	-	**43.6**	**82.9**	**43.7**	**71.4**	**83.3**	84.3

MiT-B5: 84.7M
M2F-Swin-B: 107M

MiT-B4: 64.1M
M2F-Swin-S: 69M

Encoder Model Size	Params Encoder	Params Decoder	ADE20K Flops ↓	ADE20K mIoU(SS/MS) ↑	Cityscapes Flops ↓	Cityscapes mIoU(SS/MS) ↑	COCO-Stuff Flops ↓	COCO-Stuff mIoU(SS) ↑
MiT-B0	3.4	0.4	8.4	37.4 / 38.0	125.5	76.2 / 78.1	8.4	35.6
MiT-B1	13.1	0.6	15.9	42.2 / 43.1	243.7	78.5 / 80.0	15.9	40.2
MiT-B2	24.2	3.3	62.4	46.5 / 47.5	717.1	81.0 / 82.2	62.4	44.6
MiT-B3	44.0	3.3	79.0	49.4 / 50.0	962.9	81.7 / 83.3	79.0	45.5
MiT-B4	60.8	3.3	95.7	50.3 / 51.1	1240.6	82.3 / 83.9	95.7	46.5
MiT-B5	81.4	3.3	183.3	51.0 / 51.8	1460.4	82.4 / 84.0	111.6	46.7

	method	backbone	crop size	mIoU (s.s.)	mIoU (m.s.)	#params.	FLOPs
CNN	MaskFormer [14]	R50	512 × 512	44.5	46.7	41M	53G
		R101	512 × 512	45.5	47.2	60M	73G
	Mask2Former (ours)	R50	512 × 512	47.2	49.2	44M	71G
		R101	512 × 512	**47.8**	**50.1**	63M	90G
Transformer backbones	Swin-UperNet [36, 58]	Swin-L†	640 × 640	-	53.5	234M	647G
	FaPN-MaskFormer [14, 39]	Swin-L†	640 × 640	55.2	56.7	-	-
	BEiT-UperNet [2, 58]	BEiT-L†	640 × 640	-	57.0	502M	-
	MaskFormer [14]	Swin-T	512 × 512	46.7	48.8	42M	55G
		Swin-S	512 × 512	49.8	51.0	63M	79G
		Swin-B	640 × 640	51.1	52.3	102M	195G
		Swin-B†	640 × 640	52.7	53.9	102M	195G
		Swin-L†	640 × 640	54.1	55.6	212M	375G
	Mask2Former (ours)	Swin-T	512 × 512	47.7	49.6	47M	74G
		Swin-S	512 × 512	51.3	52.4	69M	98G
		Swin-B	640 × 640	52.4	53.7	107M	223G
		Swin-B†	640 × 640	53.9	55.1	107M	223G
		Swin-L†	640 × 640	56.1	57.3	215M	403G
		Swin-L-FaPN†	640 × 640	**56.4**	**57.7**	217M	-

MiT-B3 performs better than Mask2-Swin-T with the same number of parameters