

A cluster of overlapping, semi-transparent geometric shapes in shades of blue, green, and red, located in the top-left corner of the slide.

SegFormer

Simple and Efficient Design for Semantic Segmentation with Transformers


(<https://arxiv.org/abs/2105.15203>)

한성대학교 1971336 김태민

A cluster of overlapping, semi-transparent geometric shapes in shades of light gray, located in the bottom-right corner of the slide.

A cluster of colorful, overlapping geometric shapes (triangles and polygons) in shades of blue, green, and red, located in the top-left corner of the slide.

목차

- Semantic Segmentation
 - Abstract
 - Method
 - Hierarchical Transformer Encoder
 - Hierarchical Feature Representation
 - Overlapped Patch Merging
 - Efficient Self-Attention
 - Mix-FFN
 - Lightweight All-MLP Decoder
 - Experiments
 - Competition
- 
- A cluster of light gray, overlapping geometric shapes (triangles and polygons) in the bottom-left corner of the slide.

Semantic Segmentation



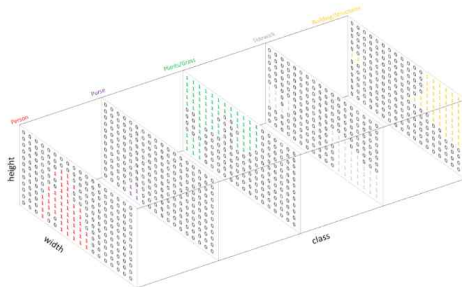
Semantic Segmentation



segmented

- 1: Person
- 2: Purse
- 3: Plants/Grass
- 4: Sidewalk
- 5: Building/Background


3	3	3	3	3	3	3	3	3	3	3	5	5	5	5	5
3	3	3	3	3	3	3	3	3	3	3	5	5	5	5	5
3	3	3	3	3	3	1	1	3	3	3	5	5	5	5	5
3	3	3	3	3	1	1	1	1	3	3	5	5	5	5	5
3	3	3	3	3	3	1	1	3	3	3	5	5	5	5	5
5	5	3	3	3	3	1	1	3	3	5	5	5	5	5	5
4	4	3	4	1	1	1	1	1	1	4	4	4	4	5	5
4	4	3	4	1	1	1	1	1	1	4	4	4	4	5	5
4	4	4	1	1	1	1	1	1	1	4	4	4	4	4	4
3	3	3	1	1	1	1	1	1	1	4	4	4	4	4	4
3	3	3	1	2	2	1	1	1	1	4	4	4	4	4	4
3	3	3	1	2	2	1	1	1	1	4	4	4	4	4	4





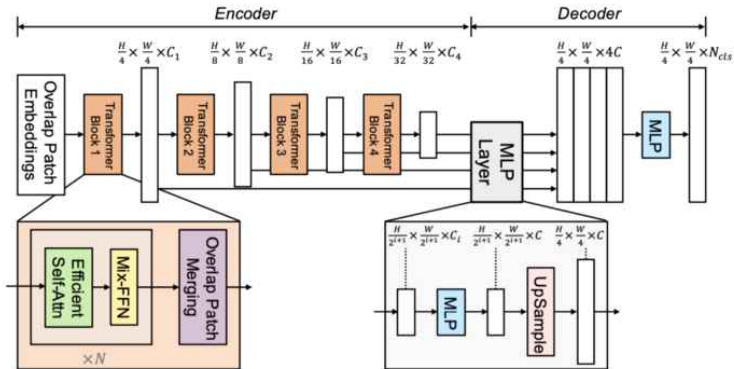
SegFormer

Abstract

- **features: 0)** 트랜스 포머의 경량화
 - **features: 1)** multiscale feature를 output으로 뽑는 계층적 구조의 Transformer encoder로 구성되며 Transformer에 사용되는 positional encoding을 제거하여 학습과 다른 이미지 사이즈를 테스트할 경우 성능이 하락된 부분을 개선
 - **features: 2)** MLP로만 이루어진 MLP decoder를 사용하며 encoder에서 얻은 multiscale feature를 결합하여 각 feature map에서의 local attention과 합쳐진 feature map에서 global attention을 통하여 높은 성능을 달성
- 

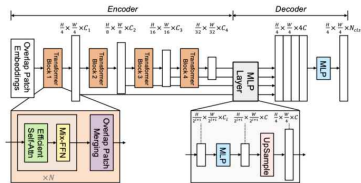
SegFormer

Method



SegFormer

Method



- ViT와 다르게 4x4 patch를 사용하는것이 dense prediction task에 더 유리하다고 한다.
- 각 패치들은 입력으로 들어가 Transformer encoder에 들어간다.
- 마치 Residual blocks 처럼 각 feature map을 뽑아 주는데 각 사이즈는 $\{1/4, 1/8, 1/16, 1/32\}$ 로 나오게 된다.
- MLP decoder에서 multi-level feature map을 여러 레이어를 거침으로써 최종적으로 $H/4 \times H/4 \times N_{cls}$ 를 갖는 Segmentation mask를 예측

SegFormer

Hierarchical Transformer Encoder

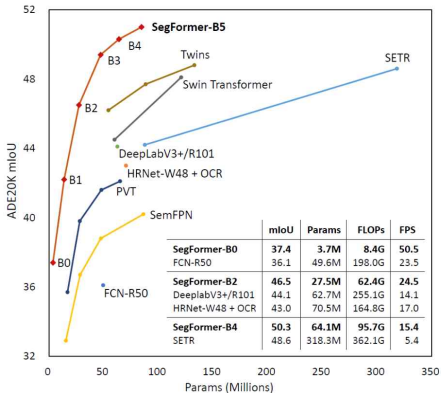


Figure 1: Performance of SegFormer-B0 to SegFormer-B5.

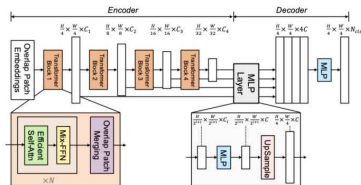
- 저자 들은 Segformer의 Encoder 를 Mix Transformer Encoder(MiT) 라 부르며 모델은 사이즈 별로 B0~B5까지 존재한다.
- 숫자가 높을 수록 높은 성능

SegFormer

Hierarchical Feature Representation

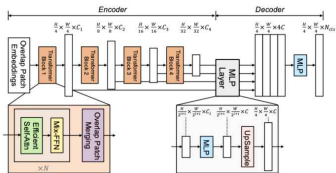
- ViT와 다르게 4x4 patch를 사용하는것이 dense prediction task에 더 유리하다고 한다.
- 각 패치들은 입력으로 들어가 Transformer encoder에 들어간다.
- 마치 Residual blocks 처럼 각 feature map을 뽑아 주는데 각 사이즈는 $\{1/4, 1/8, 1/16, 1/32\}$ 로 나오게 된다.
- MLP decoder에서 multi-level feature map을 여러 레이어를 거침으로써 최종적으로 $H/4 \times H/4 \times N(\text{cls})$ 를 갖는 Segmentation mask를 예측
- ViT와 다르게 여러 feature map
- (multi-level feature map)을 생성하여 성능을 향상

$\frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}} \times C_i$, where $i \in \{1, 2, 3, 4\}$, and C_{i+1} is larger than C_i .



SegFormer

Efficient Self-Attention



- 기존 multi-head self-attention 연산량의 문제 ViT는 16x16이었지만 Segformer는 4x4이므로 더 많은 연산을 요구함으로 **Efficient Self-Attention**을 도입 시간복잡도($O(N^2)$)

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

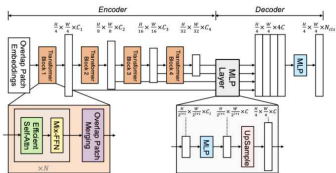
$$\hat{K} = \text{Reshape}\left(\frac{N}{R}, C \cdot R\right)(K)$$

$$K = \text{Linear}(C \cdot R, C)(\hat{K}),$$

- 저자들은 R(reduction ratio)를 사전에 정의하여 K,V의 N(HxW)채널을 줄이는sequence reduction process를 적용
- 위 수식과 같이 N을 R로 나누고 C에 R을 곱하면 Reshape이 가능해지고 이때 $C \times R$ 을 Linear 연산을 통해 다시 C로 줄임으로써 $(N/R) \times C$ 차원으로 Key와 Value로 만들어 줄 수 가 있다. 저자들은 실험을 통해 Stage-1부터 Stage-4 까지의 R을 [64, 16, 4, 1]로 설정하였다

SegFormer

Mix-FFN



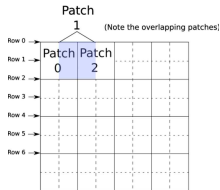
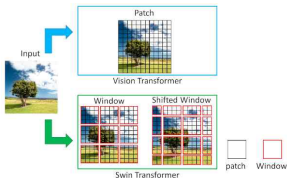
- 기존 ViT에서는 positional encoding을 적용하는데 이 방식은 input resolution이 고정되어야함 이는 문제가 있어 input resolution이 달라지면 성능이 하락함으로 이를 대신하여
- 3x3 Conv(stride:1/padding:1)을 FFN에 적용 (3 x 3 Conv의 zero padding을 통해 leak location의 정보를 고려할 수 있다고 주장)

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

- 실제로는 파라미터수를 줄이기 위해 3x3 convolution을 depth-wise convolution으로 사용

SegFormer

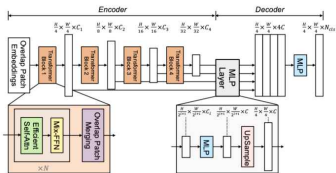
Overlapped Patch Merging



- ViT에서는 $N \times N \times 3$ patch를 $1 \times 1 \times C$ 의 벡터로 표현하는데 이럴 경우 패치들은 non-overlap 상태이므로 patch들 간의 local continuity (지역 연속성)이 보존되기가 어렵다.
- 보완하기 위해 추후 Swin Transformer에서는 Shifted Window를 통해 이를 보존하려고 했고 Segformer에서는 overlapping patch merging으로 접근하였다.
- K (patch size or kernel size), S (stride), P (padding)를 사전에 정의하여 B (batch) \times C (channel \times stride²) \times N (num of patch)의 차원으로 patch를 분할하고 B (batch) \times C (embedd dim) \times W (width) \times H (height)의 차원으로 Merging을 수행한다.

SegFormer

Lightweight All-MLP Decoder



- 저자들은 MLP layer로만 구성된 decoder를 설계하여 다른 모델의 decoder와 다르게 큰 연산량을 요구 하지 않는다고 한다.
- 작은 연산량으로 잘 작동되는 이유는 hierachical transformer encoder에서 기존 CNN 인코더보다 더 larger effective field를 가진다고 한다.

$$\hat{F}_i = \text{Linear}(C_i, C)(F_i), \forall i$$

$$\hat{F}_i = \text{Upsample}\left(\frac{W}{4} \times \frac{W}{4}\right)(\hat{F}_i), \forall i$$

$$F = \text{Linear}(4C, C)(\text{Concat}(\hat{F}_i)), \forall i$$

$$M = \text{Linear}(C, N_{cls})(F),$$

- 1. multi-level feature들의 channel을 모두 동일하게 통합시킨다.
- 2. feature size를 original image의 1/4 크기로 통합한다.
- 3. feature들을 concatenate시키고 이 과정에서 4배로 증가한 channel을 원래대로 돌린다.
- 4. 최종 segmentation mask를 예측한다. (shape: B(batch) x N(num of class) x H/4 x W/4)

SegFormer

Experiments

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		ADE20K		Cityscapes		COCO-Stuff	
	Encoder	Decoder	Flops ↓	mIoU(SS/MS) ↑	Flops ↓	mIoU(SS/MS) ↑	Flops ↓	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

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

C	Flops ↓	Params ↓	mIoU ↑
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

(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

(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	46.6

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
Non Real-Time	SETR [7]	ResNet-101	68.6	275.7	14.8	41.4	2203.3	1.2	76.6
		ResNet-101	55.1	218.8	14.9	44.7	1748.0	1.3	76.9
		PSPNet [17]	68.1	256.4	15.3	44.4	2048.9	1.2	78.5
		CCNet [41]	68.9	278.4	14.1	45.2	2224.8	1.0	80.2
		DeepLabV3+ [20]	62.7	255.1	14.1	44.1	2032.3	1.2	80.9
		OCRNet [23]	70.5	164.8	17.0	45.6	1296.8	4.2	81.1
		GSCNN [35]	-	-	-	-	-	-	80.8
		WideResNet38	-	-	-	-	-	-	80.8
		Axial-DeepLab [74]	-	-	-	-	2446.8	-	81.1
		Dynamic Routing [75]	-	-	-	-	-	-	80.7
		NAS-F48-ASPP	-	-	-	44.0	695.0	-	80.3
		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

SegFormer

Competition

- Github : <https://github.com/AlConnect-Army/qualify-test>

Public				
● 결과물을 뒤, 열람 시간(이른 후에 리미트도에 표시됩니다)				
순위	팀명	팀원	eval Public Score	작업시간
1	아이콘사주세요 작업 수 2회 최종작을 가결한	공공공공	0.79184083586257	2022-11-14 12:43:06
2	홍도환전 영는 아음 작업 수 35회 최종작을 가결한	공공공	0.781076860370475	2022-11-14 07:34:48
3	MUC 작업 수 12회 최종작을 가결한	공공공	0.75893547247329	2022-11-13 23:20:41
4	BlockAI 작업 수 4회 최종작을 가결한	공공공	0.74833873255994	2022-11-14 10:57:47
5	수학용어 작업 수 3회 최종작을 가결한	공공공	0.741558473162076	2022-11-14 16:19:34
6	한화아름스 데인 관심 부탁드립니다. 작업 수 24회 최종작을 가결한	공공공	0.73730773362485	2022-11-11 22:15:06
7	Deep Sleeping 작업 수 1회 최종작을 가결한	공공공	0.731980359269352	2022-11-14 16:48:18
8	KD1 작업 수 48회 최종작을 가결한	공공공	0.72185045474832	2022-11-14 10:55:54
9	hardyang 작업 수 16회 최종작을 가결한	공공공	0.71648333748759	2022-11-11 12:57:52
10	MMC_Lab 작업 수 29회 최종작을 가결한	공공공	0.71674429633570	2022-11-14 17:56:33

