How Segmentation Model Learns?

Review: Per-Pixel Classification is Not All You Need for Semantic Segmentation

Bowen Cheng et al.

NeurIPS 2021

20 Dec. 2021

Sungguk Cha

TL; DR

- We introduce **how segmentation models can learn.**
- We review Per-Pixel Classification is Not All You Need for Semantic Segmentation

  which **compares learning methods**.

Contents

Contents

Introduction to Learning to Segment

- Image Classification to Segmentation
- Per-pixel Classification vs Mask Classification

Introduction to Learning to Segment

- Image Classification to Segmentation
- Per-pixel Classification vs Mask Classification

Image Classification to Segmentation

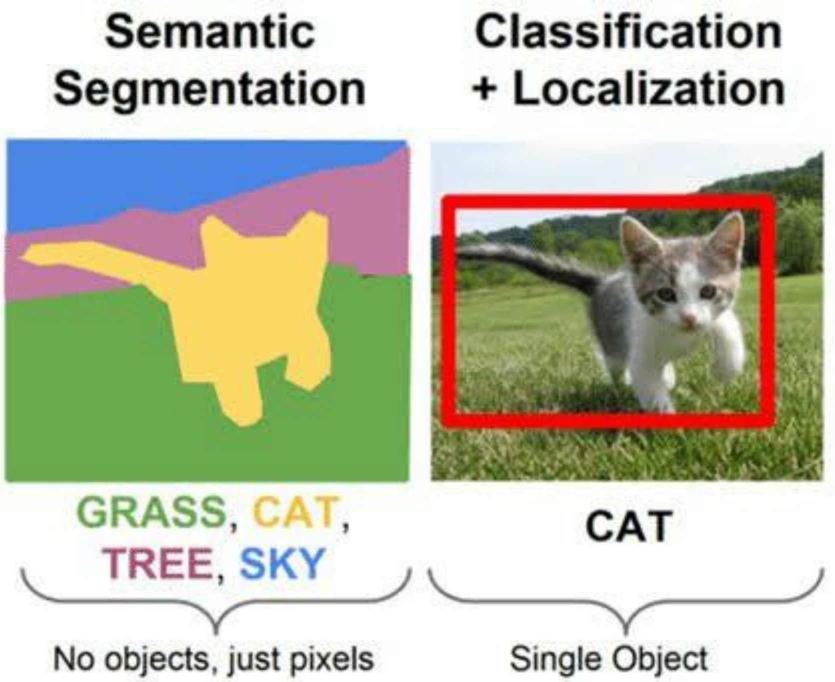


Image Source: https://www.quora.com/What-is-the-difference-between-segmentation-and-classification-in-image-processing

Image Classification to Segmentation

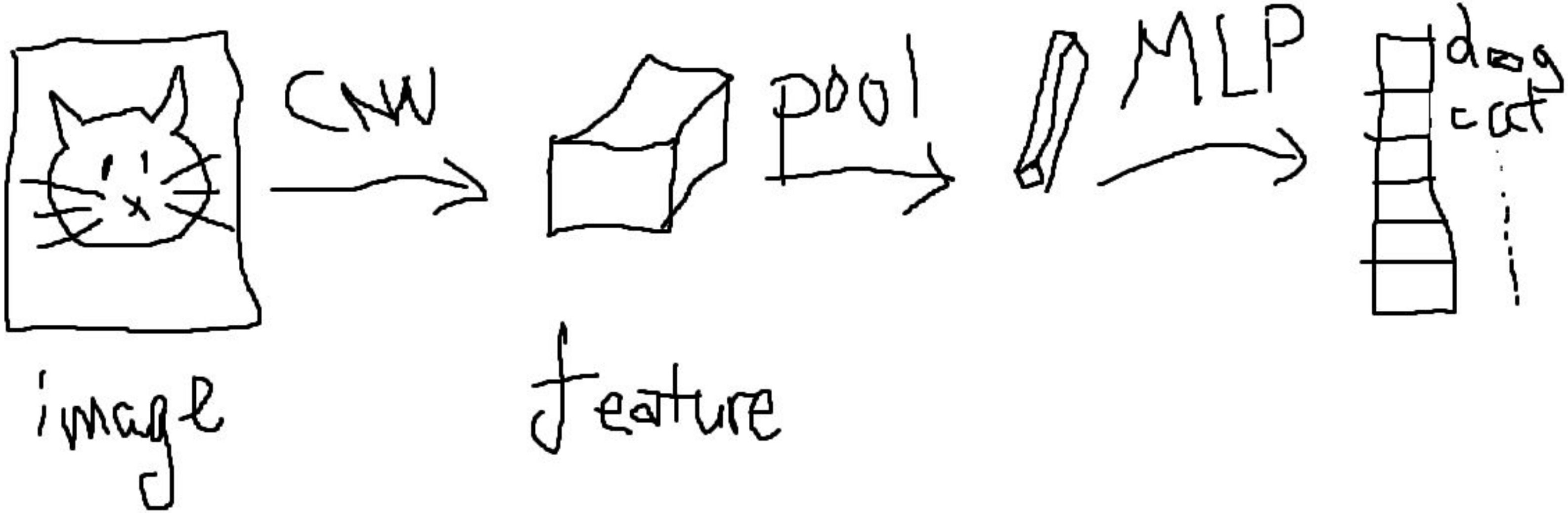


Fig. Image classification overview

Image Classification to Segmentation

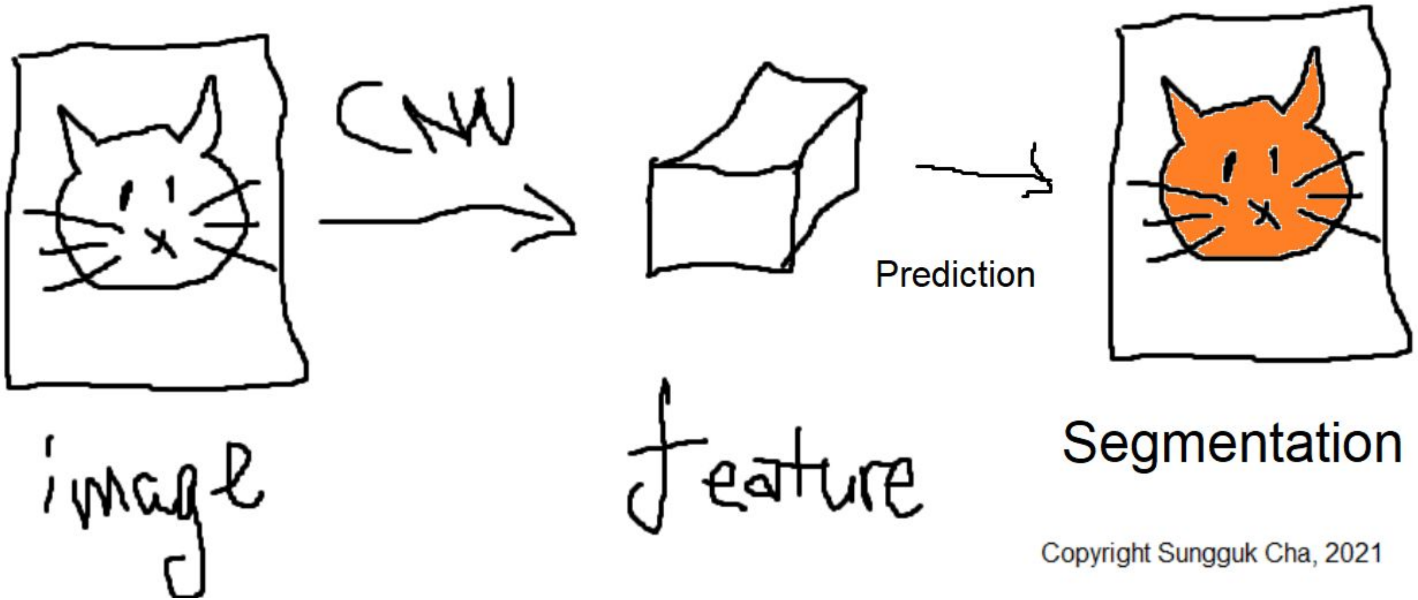


Fig. Image segmentation overview

Introduction to Learning to Segment

- Image Classification to Segmentation
- Per-pixel Classification vs Mask Classification

MINDs Lab BRAIN

Per-pixel Classification vs Mask Classification

Per-pixel Classification vs Mask Classification

- It can be seen as comparing

pixel-level supervision

vs

object-level supervision

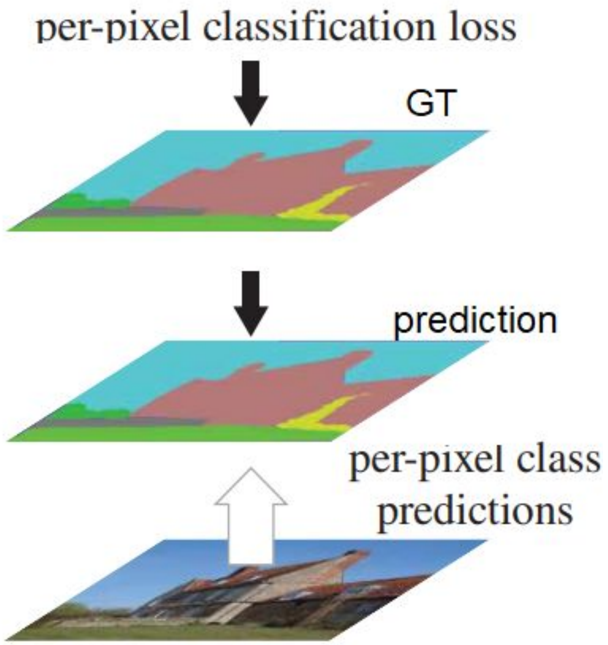
Pixel-level Supervision

Pixel-level Supervision

Fully convolutional networks (FCN), U-Net, Deeplabs and SegFormer represent it.

Pixel-level Supervision

Fully convolutional networks (FCN), U-Net, Deeplabs and SegFormer represent it.



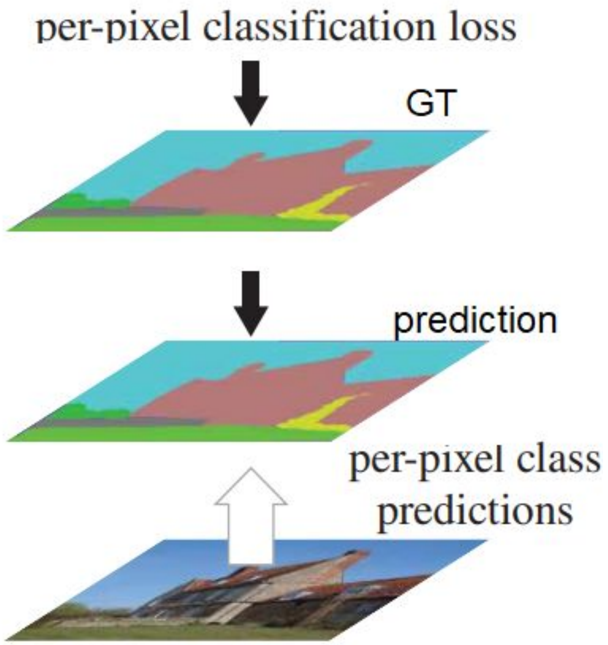
Original source: MaskFormer, Sungguk Cha edited

Pixel-level Supervision

Fully convolutional networks (FCN), U-Net, Deeplabs and SegFormer represent it.

It predicts the whole at once,

and supervises it at once.



Original source: MaskFormer, Sungguk Cha edited

Object-level Supervision

Object-level Supervision

Mask-RCNN and SETR represent it.

Object-level Supervision

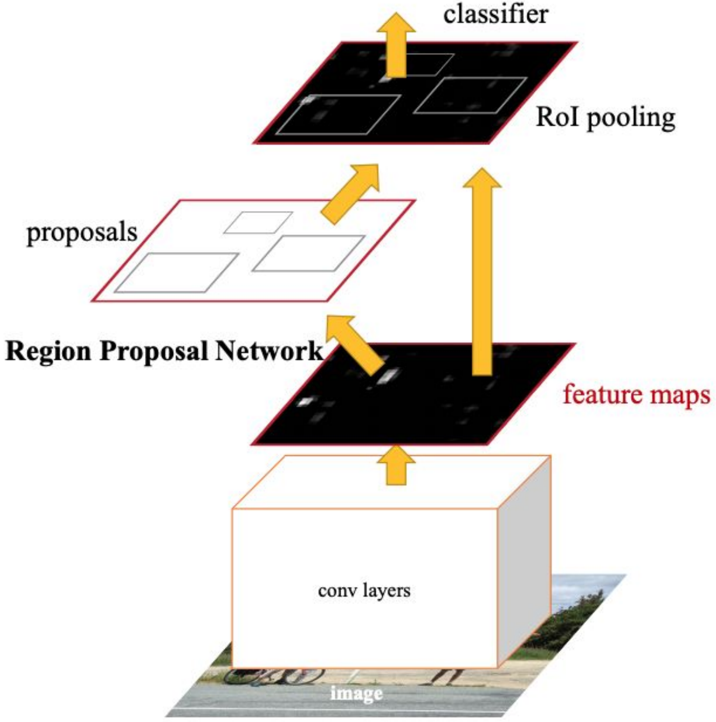
Mask-RCNN and SETR represent it.

They **predict object regions** and learn by **object-level supervision**.

Object-level Supervision

Mask-RCNN and SETR represent it.

They **predict object regions** and learn by **object-level supervision**.



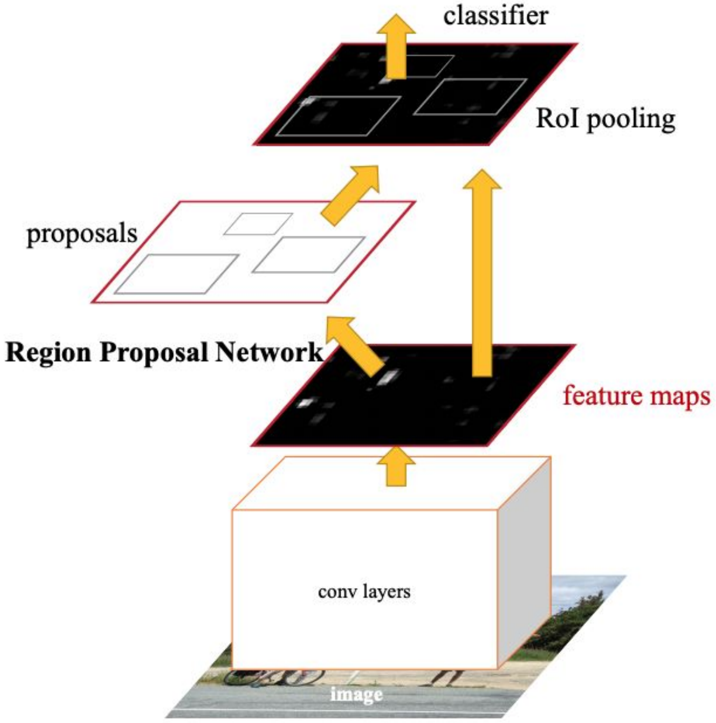
Original source: Faster R-CNN, He K. et al.

Object-level Supervision

Mask-RCNN and SETR represent it.

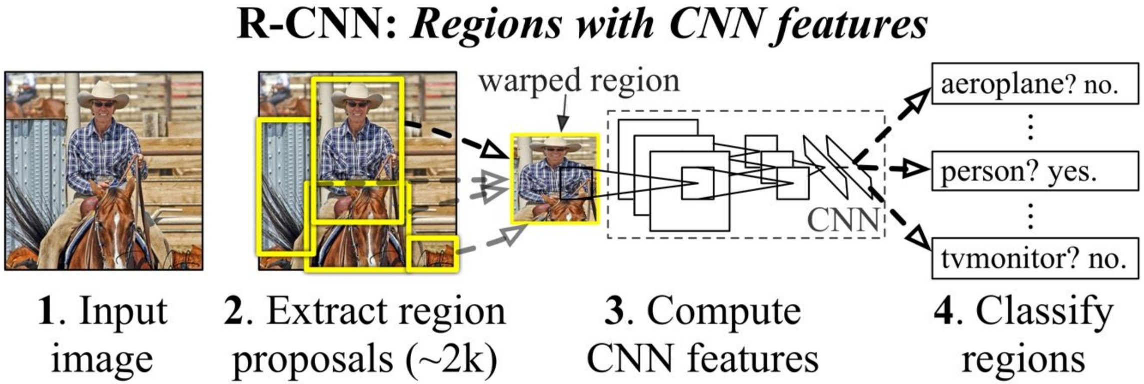
They **predict object regions** and learn by **object-level supervision**.

R-CNN is region proposal network that proposes target objects.



Original source: Faster R-CNN, He K. et al.

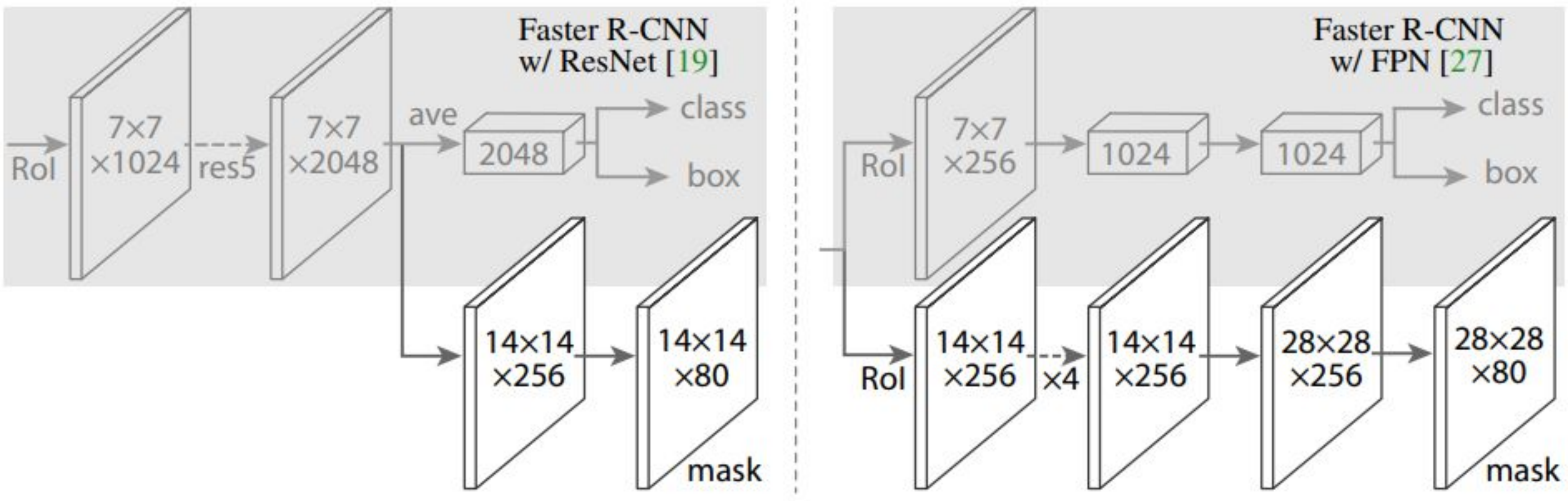
Object-level Supervision



Original source: R-CNN, Ross Girshick et al.

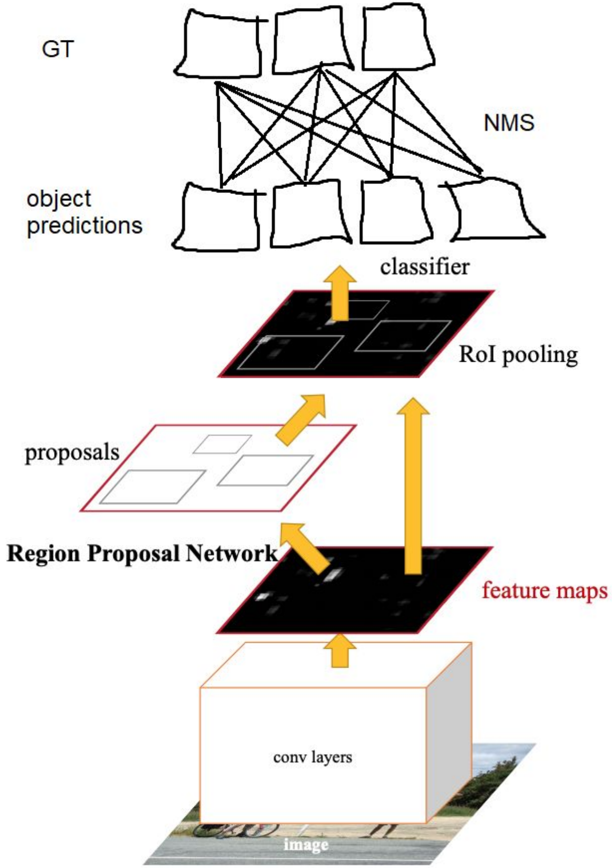
MINDs Lab BRAIN

Object-level Supervision



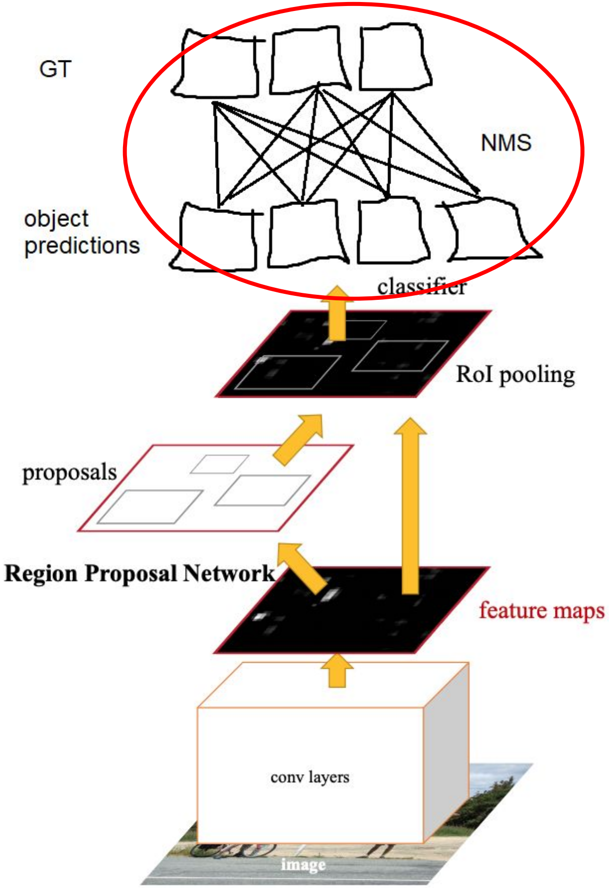
Original source: Mask R-CNN, He K. et al.

Object-level Supervision



Original source: Faster R-CNN, He K. et al.
Sungguk Cha edited

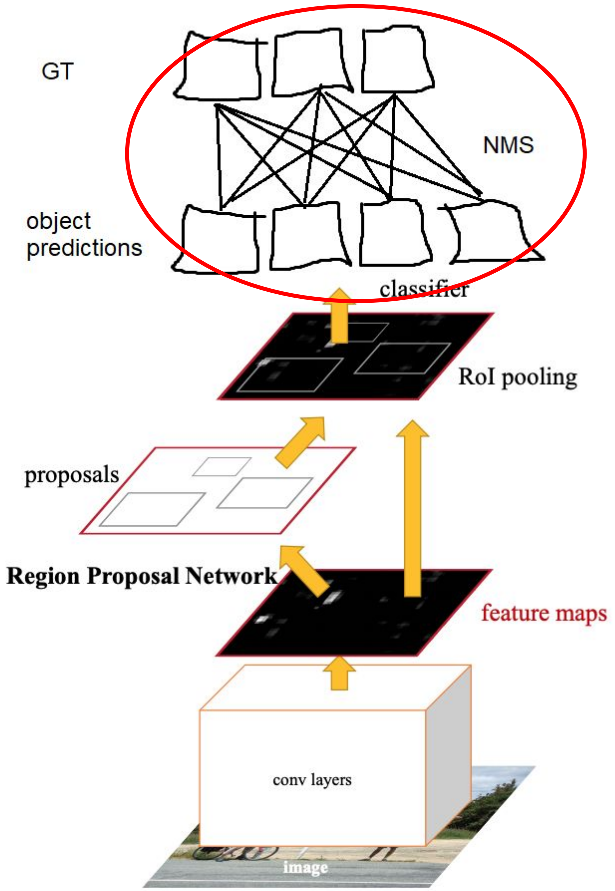
Object-level Supervision



Original source: Faster R-CNN, He K. et al.
Sungguk Cha edited

Object-level Supervision

These model learns by object-level supervision



Original source: Faster R-CNN, He K. et al.
Sungguk Cha edited

Per-pixel Classification vs Mask Classification Summary

	Pixel-level supervision	Object-level supervision
Model	U-Net, DeepLab	Mask R-CNN
Supervision	Pixel-level loss	Object-level loss

MINDs Lab BRAIN

Question now, is which one **superior**?

Question now, is which one **superior**?

This paper compares and contrasts the supervisions.

Question now, is which one **superior**?

This paper compares and contrasts the supervisions

and presents ***MaskFormer*** that takes advantages of mask-supervision.

Question now, is which one **superior**?

This paper compares and contrasts the supervisions

and presents ***MaskFormer*** that takes advantages of mask-supervision

using the exact same model, loss, and training procedure.

Contents

- Introduction to Learning to Segment
  - Image Classification to Segmentation
  - Per-pixel Classification vs Mask Classification
- MaskFormer
- Conclusion

MINDs Lab BRAIN

MaskFormer

MaskFormer

They contrasted the two supervisions in a simple following manner.

MaskFormer

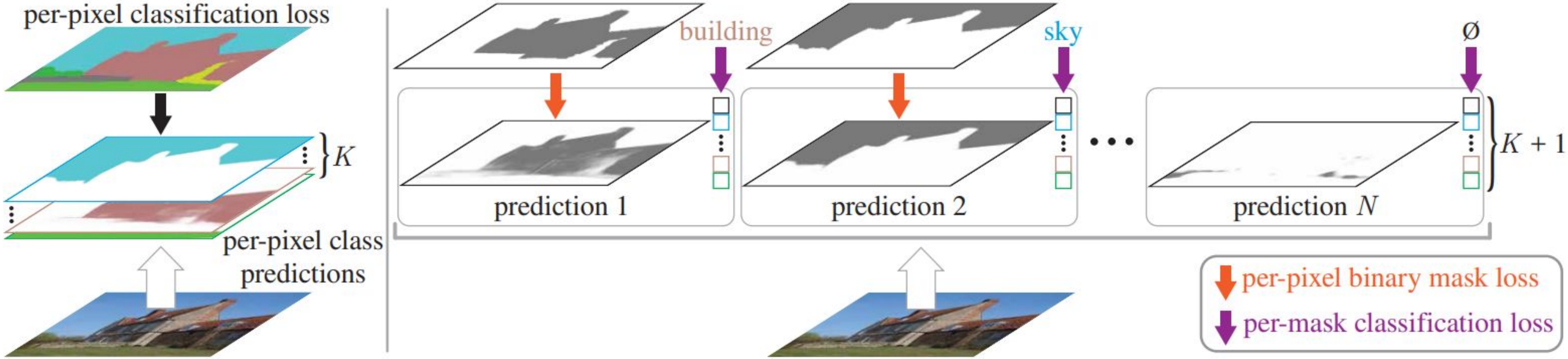


Figure: ***Per-pixel classification vs. mask classification***

MaskFormer

Instead of learning at once,
mask supervision teaches ***N*** categories independently.

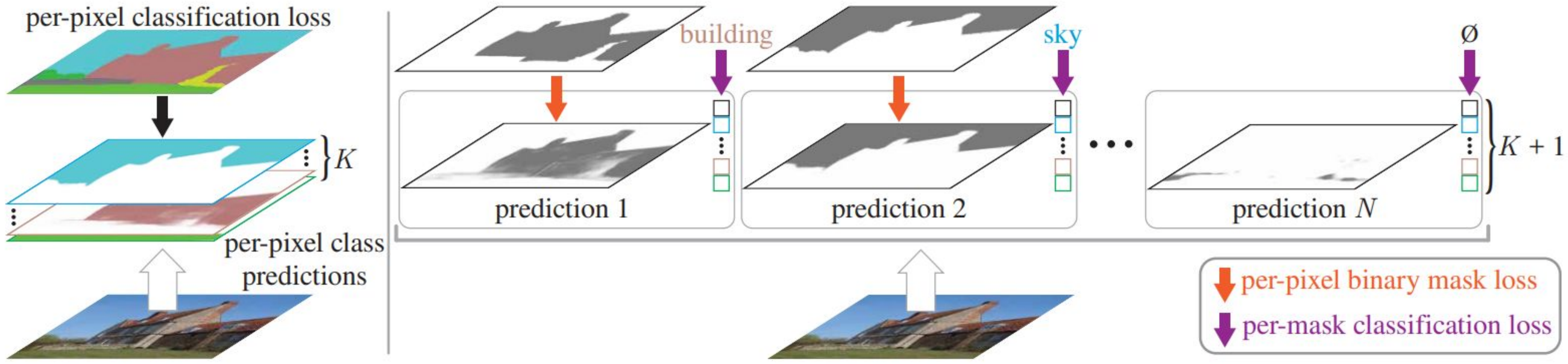


Figure: ***Per-pixel classification vs. mask classification***

MaskFormer

Instead of learning at once,
mask supervision teaches ***N*** categories independently.
I.e., mask classification has N category-predictors.

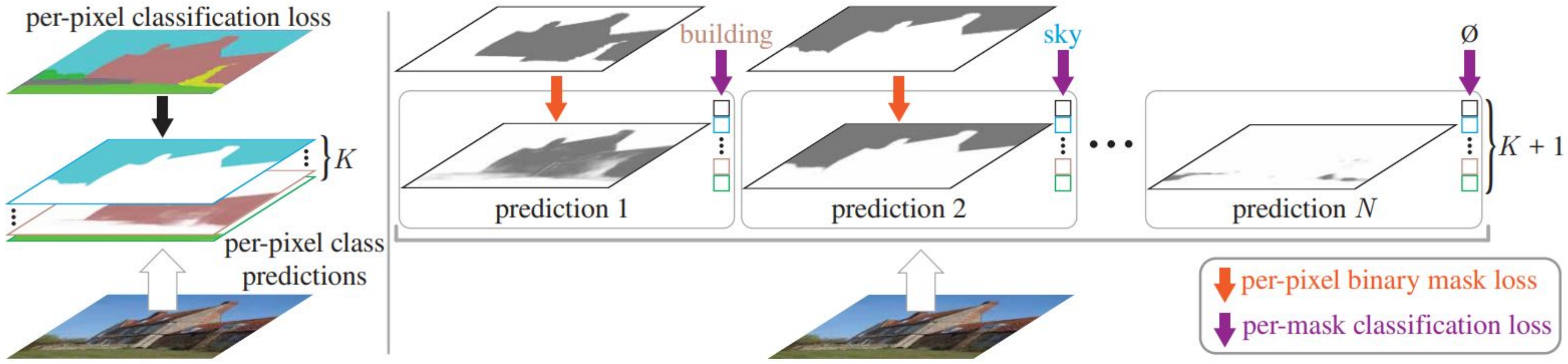


Figure: ***Per-pixel classification vs. mask classification***

MaskFormer

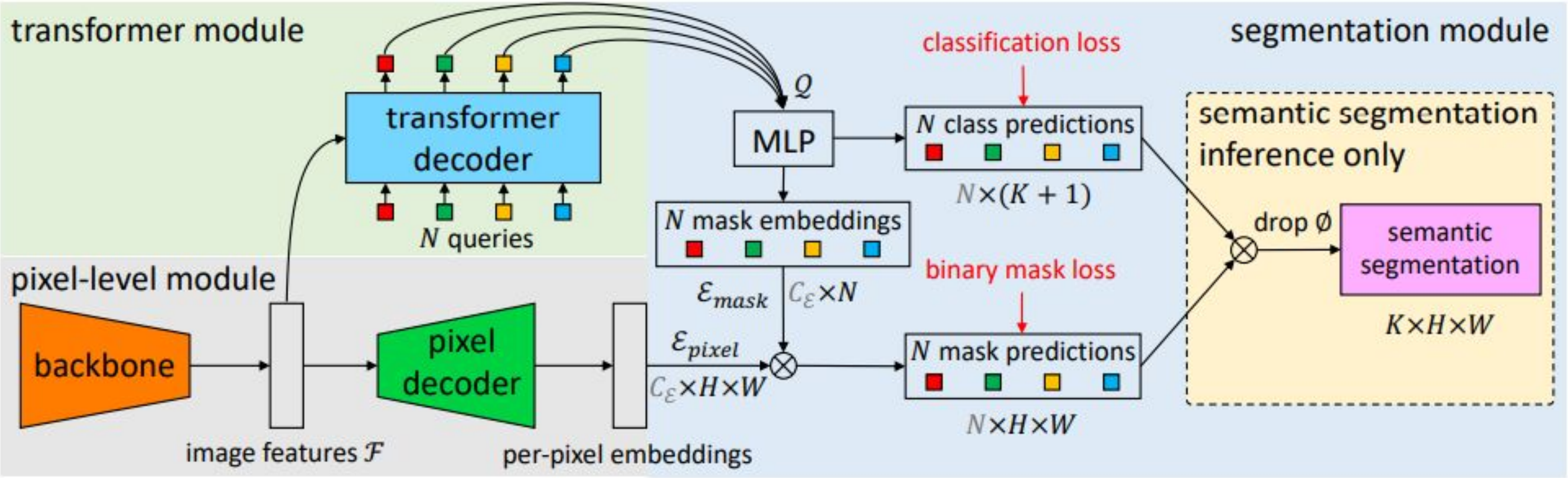
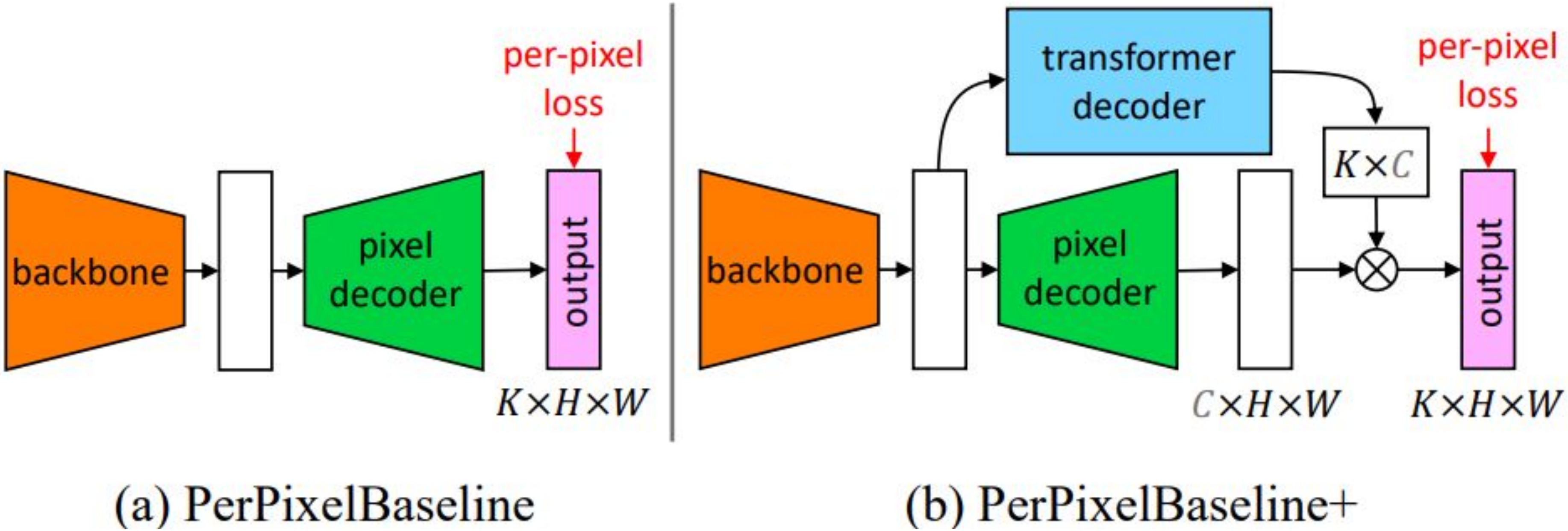


Figure: MaskFormer architecture

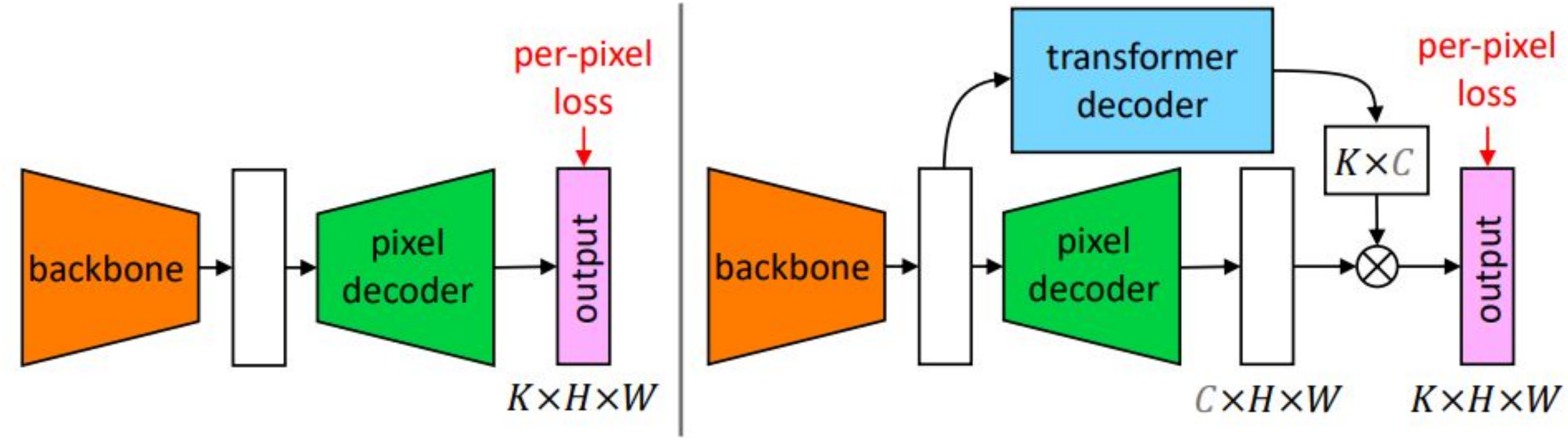
Baselines



MINDs Lab BRAIN

Baselines

For fair comparison w.r.t. #params,
they experimented PerPixelBaseline+ as well.



(a) PerPixelBaseline (b) PerPixelBaseline+

MINDs Lab
BRAIN

Experiments

Experiments

It outperforms SOTAs

in semantic segmentation (ADE20K)

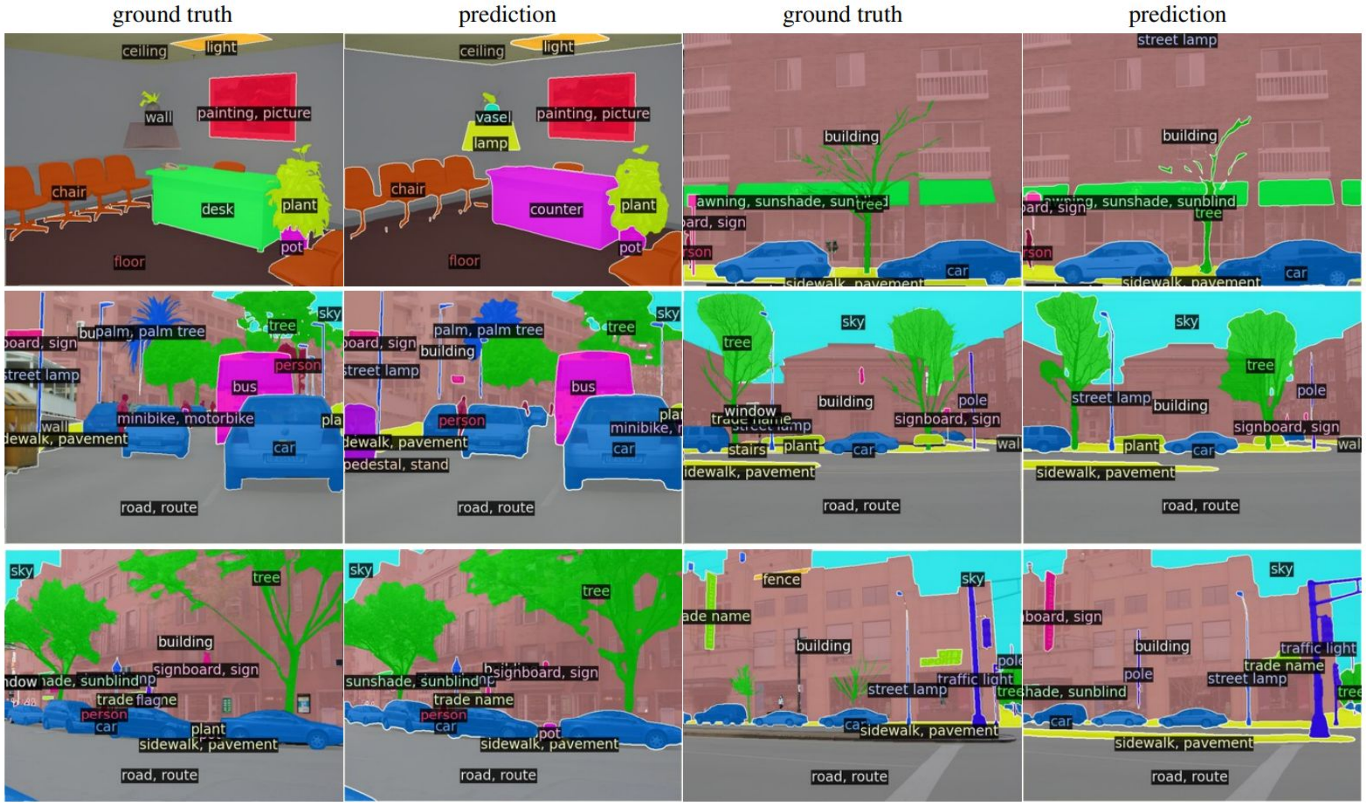
and in panoptic segmentation (COCO)

MINDs Lab BRAIN

Experiments on ADE20k

MINDs Lab
BRAIN

Experiments on ADE20k



Experiments on ADE20k

Table 1: **Semantic segmentation on ADE20K `val` with 150 categories.** Mask classification-based MaskFormer outperforms the best per-pixel classification approaches while using fewer parameters and less computation. We report both single-scale (s.s.) and multi-scale (m.s.) inference results with ±*std*. FLOPs are computed for the given crop size. Frames-per-second (fps) is measured on a V100 GPU with a batch size of 1.[3] Backbones pre-trained on ImageNet-22K are marked with †.

	method	backbone	crop size	mIoU (s.s.)	mIoU (m.s.)	#params.	FLOPs	fps
CNN backbones	OCRNet [50]	R101c	520 × 520	-	45.3	-	-	-
	DeepLabV3+ [9]	R50c	512 × 512	44.0	44.9	44M	177G	21.0
		R101c	512 × 512	45.5	46.4	63M	255G	14.2
	MaskFormer (ours)	R50	512 × 512	44.5 ±0.5	46.7 ±0.6	41M	53G	24.5
		R101	512 × 512	45.5 ±0.5	47.2 ±0.2	60M	73G	19.5
		R101c	512 × 512	**46.0** ±0.1	**48.1** ±0.2	60M	80G	19.0
Transformer backbones	SETR [53]	ViT-L†	512 × 512	-	50.3	308M	-	-
	Swin-UperNet [29, 49]	Swin-T	512 × 512	-	46.1	60M	236G	18.5
		Swin-S	512 × 512	-	49.3	81M	259G	15.2
		Swin-B†	640 × 640	-	51.6	121M	471G	8.7
		Swin-L†	640 × 640	-	53.5	234M	647G	6.2
	MaskFormer (ours)	Swin-T	512 × 512	46.7 ±0.7	48.8 ±0.6	42M	55G	22.1
		Swin-S	512 × 512	49.8 ±0.4	51.0 ±0.4	63M	79G	19.6
		Swin-B	640 × 640	51.1 ±0.2	52.3 ±0.4	102M	195G	12.6
		Swin-B†	640 × 640	52.7 ±0.4	53.9 ±0.2	102M	195G	12.6
		Swin-L†	640 × 640	**54.1** ±0.2	**55.6** ±0.1	212M	375G	7.9

Experiments on COCO panoptic

Table 3: **Panoptic segmentation on COCO panoptic `val` with 133 categories.** MaskFormer seamlessly unifies semantic- and instance-level segmentation without modifying the model architecture or loss. Our model, which achieves better results, can be regarded as a box-free simplification of DETR [4]. The major improvement comes from "stuff" classes (PQ^{St}) which are ambiguous to represent with bounding boxes. For MaskFormer (DETR) we use the exact same post-processing as DETR. Note, that in this setting MaskFormer performance is still better than DETR (+2.2 PQ). Our model also outperforms recently proposed Max-DeepLab [42] without the need of sophisticated auxiliary losses, while being more efficient. FLOPs are computed as the average FLOPs over 100 validation images (COCO images have varying sizes). Frames-per-second (fps) is measured on a V100 GPU with a batch size of 1 by taking the average runtime on the entire `val` set *including post-processing time*. Backbones pre-trained on ImageNet-22K are marked with †.

	method	backbone	PQ	PQ^{Th}	PQ^{St}	SQ	RQ	#params.	FLOPs	fps
CNN backbones	DETR [4]	R50 + 6 Enc	43.4	48.2	36.3	79.3	53.8	-	-	-
	MaskFormer (DETR)	R50 + 6 Enc	45.6	50.0 (+1.8)	39.0 (+2.7)	80.2	55.8	-	-	-
	MaskFormer (ours)	R50 + 6 Enc	**46.5**	**51.0** (+2.8)	**39.8** (+3.5)	**80.4**	**56.8**	45M	181G	17.6
	DETR [4]	R101 + 6 Enc	45.1	50.5	37.0	79.9	55.5	-	-	-
	MaskFormer (ours)	R101 + 6 Enc	**47.6**	**52.5** (+2.0)	**40.3** (+3.3)	**80.7**	**58.0**	64M	248G	14.0
Transformer backbones	Max-DeepLab [42]	Max-S	48.4	53.0	41.5	-	-	62M	324G	7.6
		Max-L	51.1	57.0	42.2	-	-	451M	3692G	-
	MaskFormer (ours)	Swin-T	47.7	51.7	41.7	80.4	58.3	42M	179G	17.0
		Swin-S	49.7	54.4	42.6	80.9	60.4	63M	259G	12.4
		Swin-B	51.1	56.3	43.2	81.4	61.8	102M	411G	8.4
		Swin-B†	51.8	56.9	**44.1**	81.4	62.6	102M	411G	8.4
		Swin-L†	**52.7**	**58.5**	**44.0**	**81.8**	**63.5**	212M	792G	5.2

Ablation: MaskFormer vs per-pixel

Table 2: **MaskFormer *vs*. per-pixel classification baselines on 4 semantic segmentation datasets.** MaskFormer improvement is larger when the number of classes is larger. We use a ResNet-50 backbone and report single scale mIoU and PQ^{St} for ADE20K, COCO-Stuff and ADE20K-Full, whereas for higher-resolution Cityscapes we use a deeper ResNet-101 backbone following [8, 9].

	Cityscapes (19 classes)		ADE20K (150 classes)		COCO-Stuff (171 classes)		ADE20K-Full (847 classes)	
	mIoU	PQ^{St}	mIoU	PQ^{St}	mIoU	PQ^{St}	mIoU	PQ^{St}
PerPixelBaseline	77.4	58.9	39.2	21.6	32.4	15.5	12.4	5.8
PerPixelBaseline+	**78.5**	60.2	41.9	28.3	34.2	24.6	13.9	9.0
MaskFormer (ours)	**78.5** (+0.0)	**63.1** (+2.9)	**44.5** (+2.6)	**33.4** (+5.1)	**37.1** (+2.9)	**28.9** (+4.3)	**17.4** (+3.5)	**11.9** (+2.9)

Ablation: MaskFormer vs per-pixel

Table 2: **MaskFormer *vs*. per-pixel classification baselines on 4 semantic segmentation datasets.** MaskFormer improvement is larger when the number of classes is larger. We use a ResNet-50 backbone and report single scale mIoU and PQ^{St} for ADE20K, COCO-Stuff and ADE20K-Full, whereas for higher-resolution Cityscapes we use a deeper ResNet-101 backbone following [8, 9].

	Cityscapes (19 classes)		ADE20K (150 classes)		COCO-Stuff (171 classes)		ADE20K-Full (847 classes)	
	mIoU	PQ^{St}	mIoU	PQ^{St}	mIoU	PQ^{St}	mIoU	PQ^{St}
PerPixelBaseline	77.4	58.9	39.2	21.6	32.4	15.5	12.4	5.8
PerPixelBaseline+	**78.5**	60.2	41.9	28.3	34.2	24.6	13.9	9.0
MaskFormer (ours)	**78.5** (+0.0)	**63.1** (+2.9)	**44.5** (+2.6)	**33.4** (+5.1)	**37.1** (+2.9)	**28.9** (+4.3)	**17.4** (+3.5)	**11.9** (+2.9)

Contents

MINDs Lab BRAIN

Conclusion

Conclusion

It can be seen as dividing a harder problem into easier problems.

Conclusion

It can be seen as dividing a harder problem into easier problems.

Also it can be seen as solving a problem with multi-task learning point of view.

Conclusion

It can be seen as dividing a harder problem into easier problems.

Also it can be seen as solving a problem with multi-task learning point of view.

It works better.