A guide to image segmentation

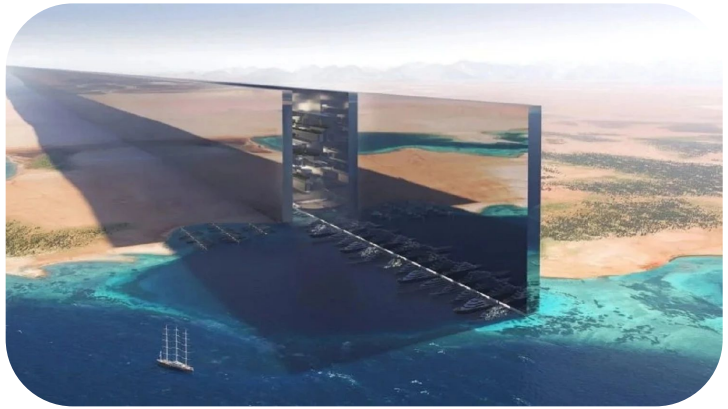
Jakub Fajkowski

Background

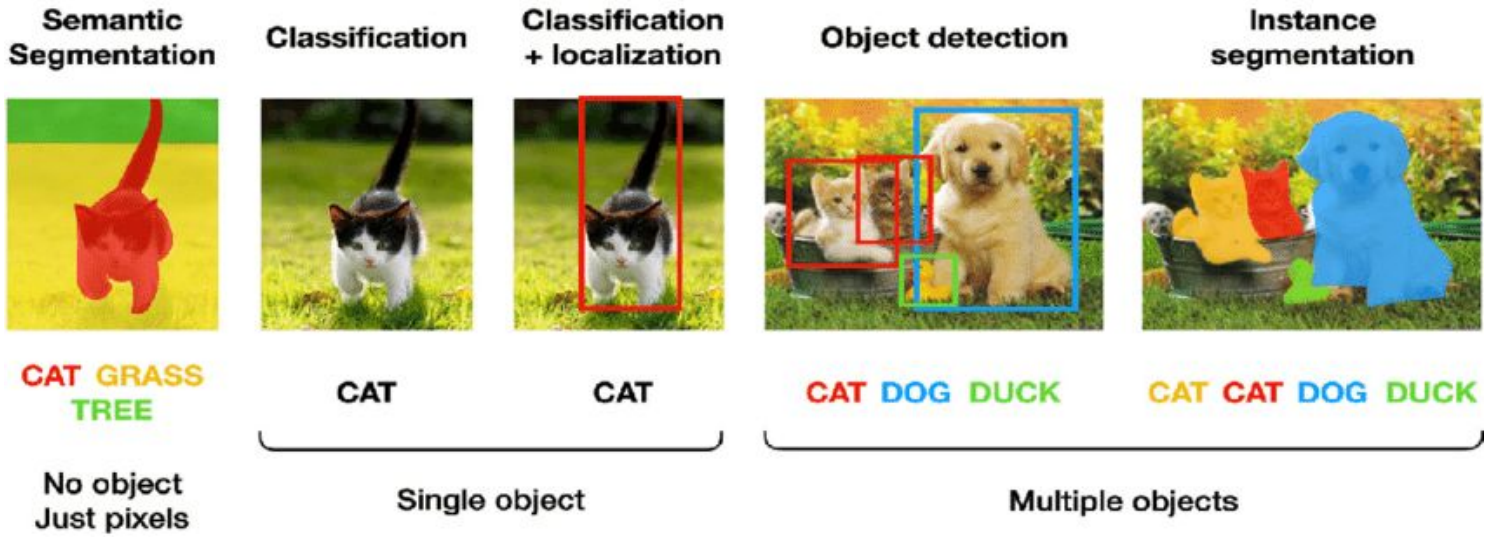
- Master of Science in CS in 2020
- Professionally active in AI since 2017
- Experience in:
  - Natural language processing
  - Time series forecasting
  - Computer vision
  - Selling carpets
  - Making balloon animals

Currently @ AI Clearing

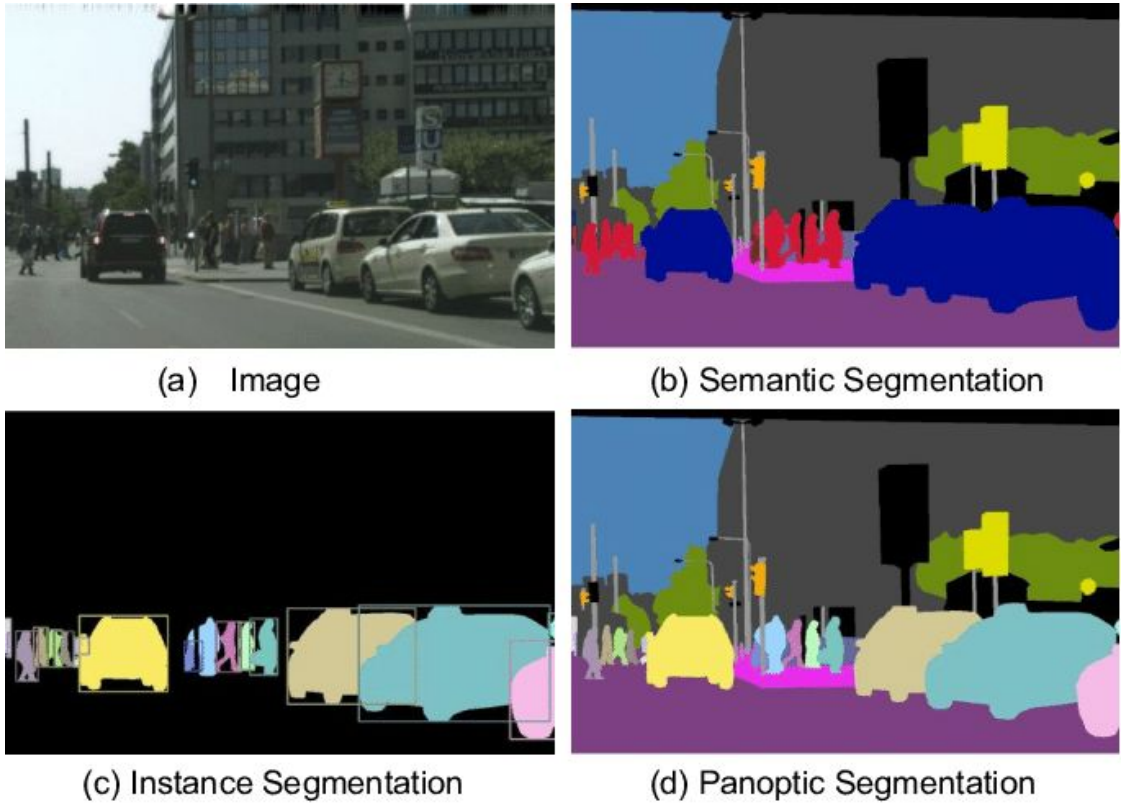
- AI Tech Lead at AI Clearing
- Production grade models for construction site analytics
- Successful model deployment for different construction domains:
  - Solar
  - Earthworks
  - Pipelines
- Dozens of construction sites monitored on 6 continents
- Participation in the largest construction site in the world!



Basic vision tasks



Side note - panoptic segmentation



(a) Image

(b) Semantic Segmentation

(c) Instance Segmentation

(d) Panoptic Segmentation

Perception by humans and machines

Things:

- Constrained shape
- Individual instances
- Easier to point finger at
- Person, cat, car, etc.

Stuff:

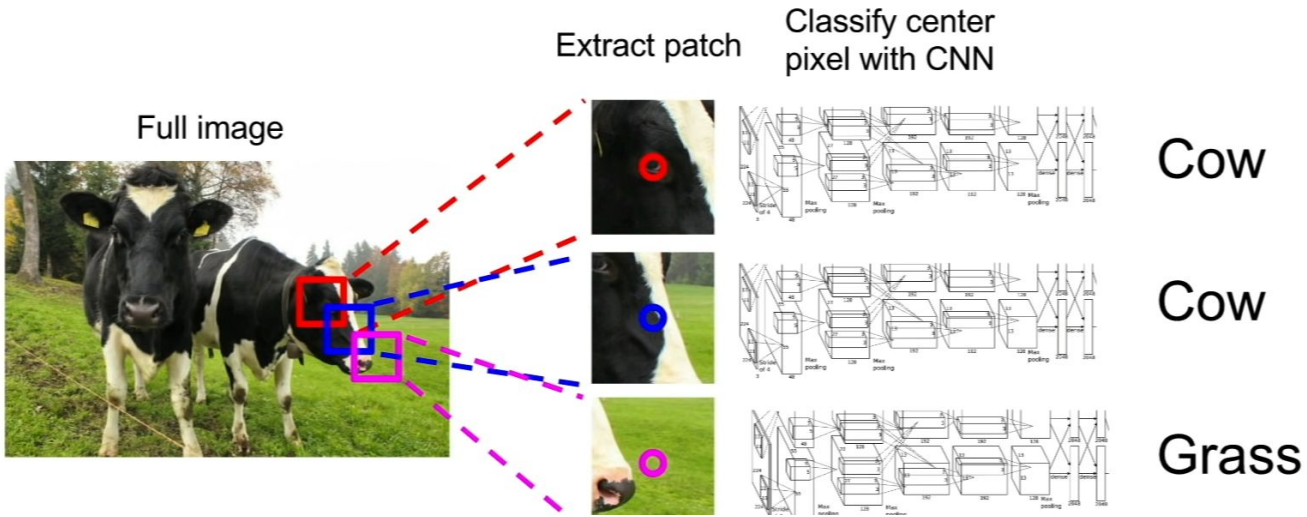
- Amorphous (shapeless)
- No notion of instances
- Harder to point finger at
- Sky, water, sand, etc.

Semantic segmentation

Year is 2014

- Deep Neural Networks trained in a supervised manner are breaking another records on ImageNet classification
- Typical flow for classification is to resize the image to 224x224 an pass it through series of convolutional layers
- Popular architectures at the time:
  - AlexNet
  - VGG
  - GoogLeNet

Initial idea - Sliding Window Classification



Idea behind Fully Convolutional Networks

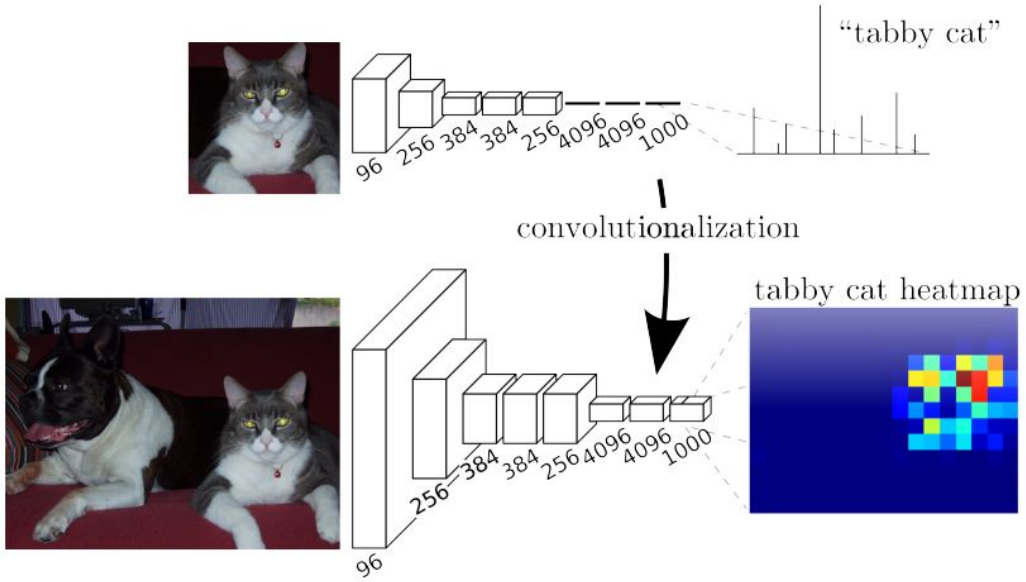


Fig. 2. Transforming fully connected layers into convolution layers enables a classification net to output a spatial map. Adding differentiable interpolation layers and a spatial loss (as in Figure 1) produces an efficient machine for end-to-end pixelwise learning.

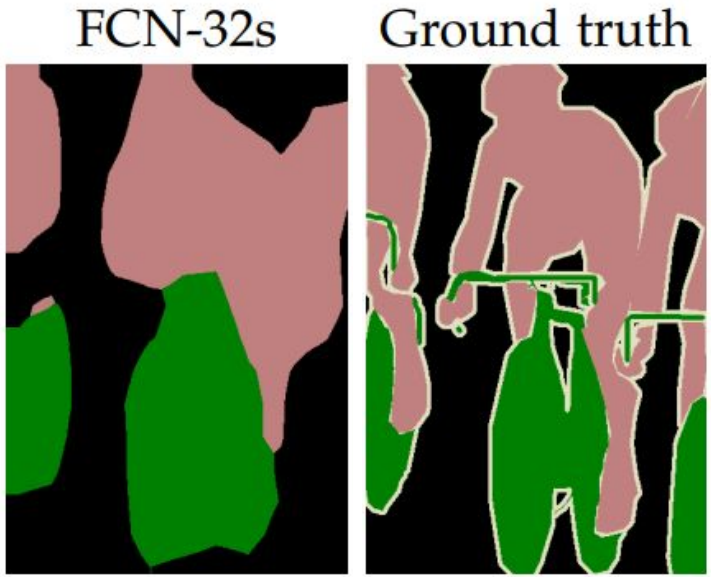
Results on PASCAL VOC 2011

TABLE 1

We adapt and extend three classification convnets. We compare performance by mean intersection over union on the validation set of PASCAL VOC 2011 and by inference time (averaged over 20 trials for a 500×500 input on an NVIDIA Titan X). We detail the architecture of the adapted nets with regard to dense prediction: number of parameter layers, receptive field size of output units, and the coarsest stride within the net. (These numbers give the best performance obtained at a fixed learning rate, not best performance possible.)

	FCN-AlexNet	FCN-VGG16	FCN-GoogLeNet[3]
mean IU	39.8	**56.0**	42.5
forward time	16 ms	100 ms	20 ms
conv. layers	8	16	22
parameters	57M	134M	6M
rf size	355	404	907
max stride	32	32	32

Problem - too coarse results



Solution - fuse finer feature maps

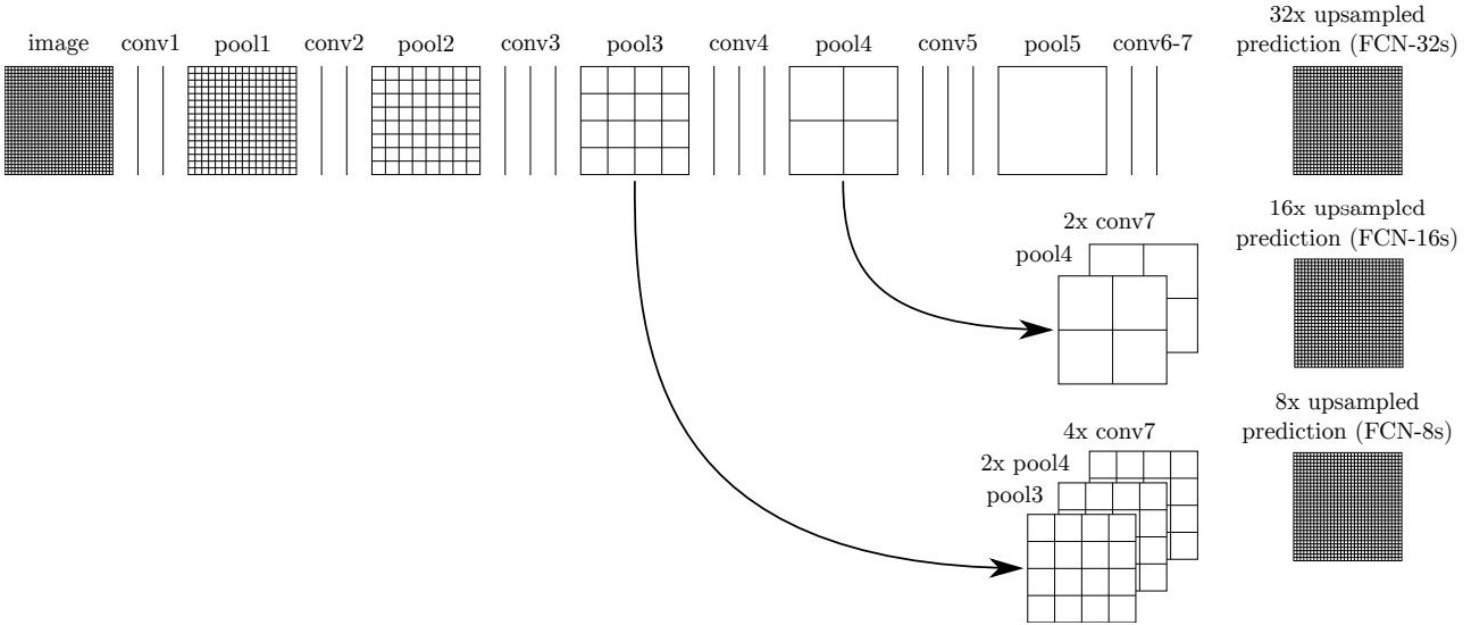


Fig. 3. Our DAG nets learn to combine coarse, high layer information with fine, low layer information. Pooling and prediction layers are shown as grids that reveal relative spatial coarseness, while intermediate layers are shown as vertical lines. First row (FCN-32s): Our single-stream net, described in Section 4.1, upsamples stride 32 predictions back to pixels in a single step. Second row (FCN-16s): Combining predictions from both the final layer and the `pool4` layer, at stride 16, lets our net predict finer details, while retaining high-level semantic information. Third row (FCN-8s): Additional predictions from `pool3`, at stride 8, provide further precision.

Improved results

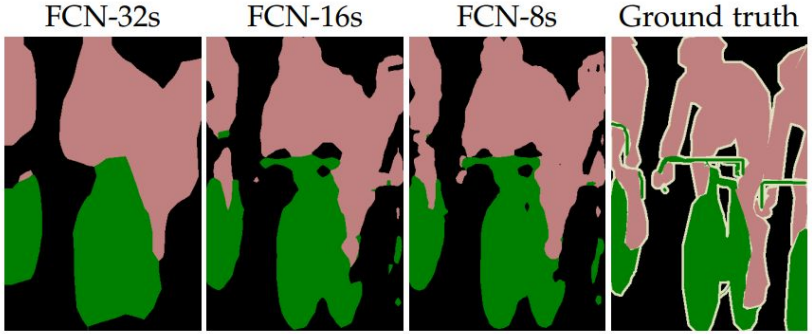


Fig. 4. Refining fully convolutional networks by fusing information from layers with different strides improves spatial detail. The first three images show the output from our 32, 16, and 8 pixel stride nets (see Figure 3).

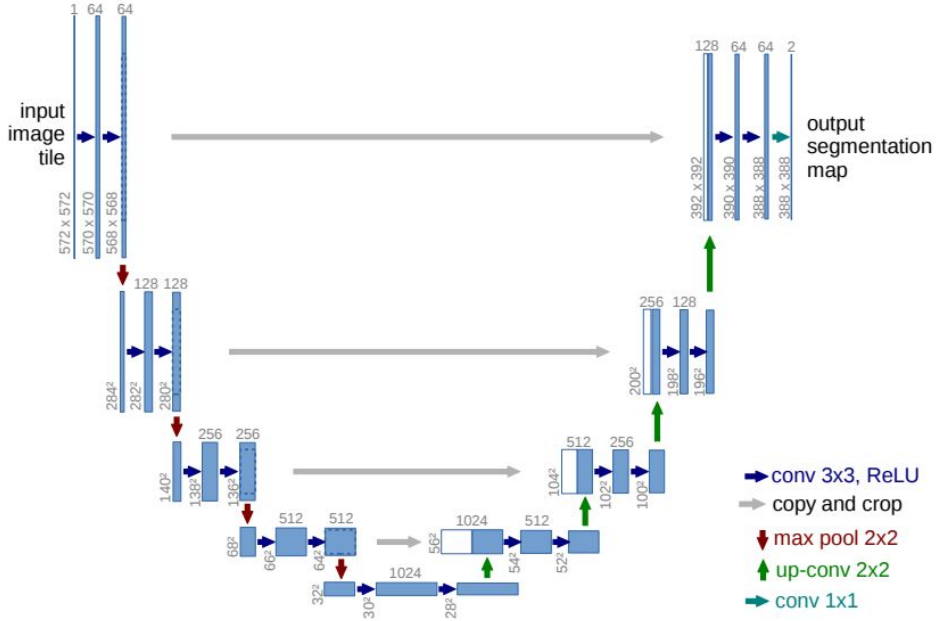
TABLE 3
Comparison of FCNs on a subset[5] of PASCAL VOC 2011 segval. Learning is end-to-end with batch size one and high momentum, with the exception of the fixed variant that fixes all features. Note that FCN-32s is FCN-VGG16, renamed to highlight stride, and the FCN-poolX are truncated nets with the same strides as FCN-32/16/8s.

	pixel acc.	mean acc.	mean IU	f.w. IU
FCN-32s	90.5	76.5	63.6	83.5
FCN-16s	91.0	78.1	65.0	84.3
FCN-8s at-once	91.1	**78.5**	65.4	84.4
FCN-8s staged	**91.2**	77.6	**65.5**	**84.5**
FCN-32s fixed	82.9	64.6	46.6	72.3
FCN-pool5	87.4	60.5	50.0	78.5
FCN-pool4	78.7	31.7	22.4	67.0
FCN-pool3	70.9	13.7	9.2	57.6

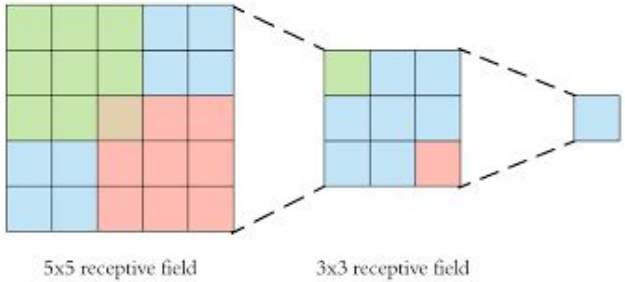
TABLE 4
Our FCN gives a 30% relative improvement on the previous best PASCAL VOC 11/12 test results with faster inference and learning.

	mean IU VOC2011 test	mean IU VOC2012 test	inference time
R-CNN [5]	47.9	-	-
SDS [14]	52.6	51.6	~ 50 s
FCN-8s	**67.5**	**67.2**	~ **100 ms**

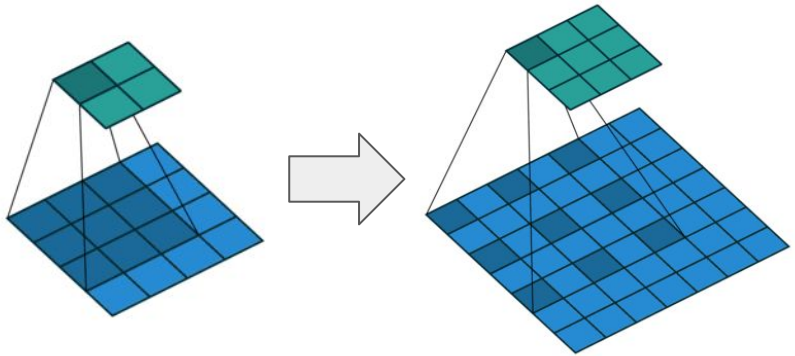
Follow up for FCN feature map fusion - UNet



Problem with such approach?



Small field of view



Atrous (dilated) convolution

DeepLabv3

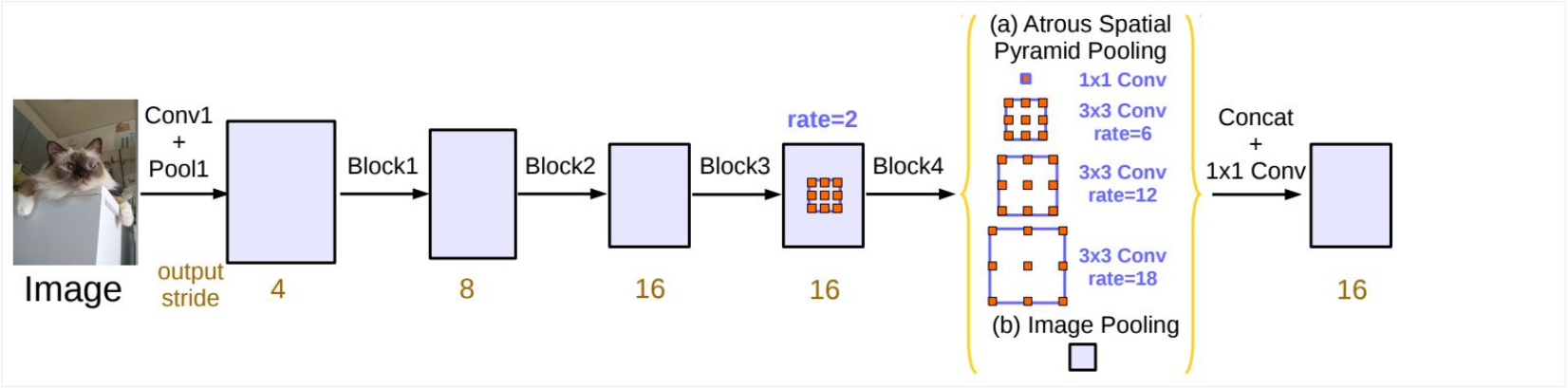


Figure 5. Parallel modules with atrous convolution (ASPP), augmented with image-level features.

DeepLabv3	85.7
DeepLabv3-JFT	86.9

Table 7. Performance on PASCAL VOC 2012 *test* set.

Shift of paradigm

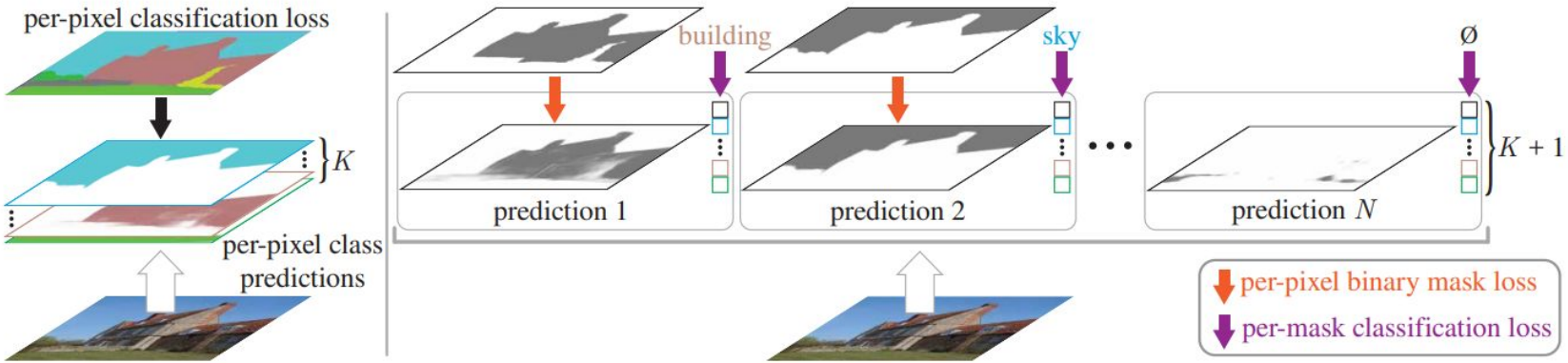


Figure 1: **Per-pixel classification *vs*. mask classification. (left)** Semantic segmentation with per-pixel classification applies the same classification loss to each location. **(right)** Mask classification predicts a set of binary masks and assigns a single class to each mask. Each prediction is supervised with a per-pixel binary mask loss and a classification loss. Matching between the set of predictions and ground truth segments can be done either via *bipartite matching* similarly to DETR [4] or by *fixed matching* via direct indexing if the number of predictions and classes match, *i.e.*, if $N = K$.

MaskFormer

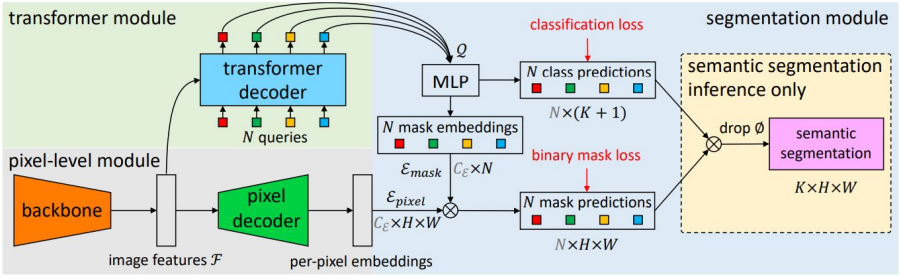


Table 2 Baseline performance on the validation set of SceneParse150.

Networks	Pixel Acc.	Mean Acc.	Mean IoU	Weighted IoU
FCN-8s	71.32%	40.32%	0.2939	0.5733
SegNet	71.00%	31.14%	0.2164	0.5384
DilatedVGG	73.55%	44.59%	0.3231	0.6014
DilatedResNet-34	76.47%	45.84%	0.3277	0.6068
DilatedResNet-50	76.40%	45.93%	0.3385	0.6100
Cascade-SegNet	71.83%	37.90%	0.2751	0.5805
Cascade-DilatedVGG	74.52%	45.38%	0.3490	0.6108

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 $\pm std$. 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

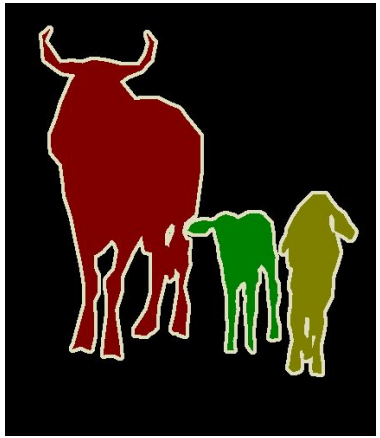
Data

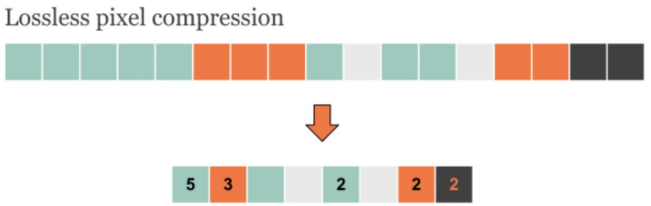
Datasets:

- Pascal VOC (kinda legacy)
- Cityscapes
- ADE20K
- COCO-Stuff

Formats:

- masks
- polygons
- RLE

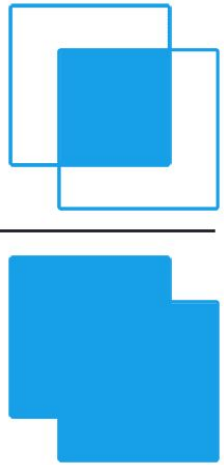




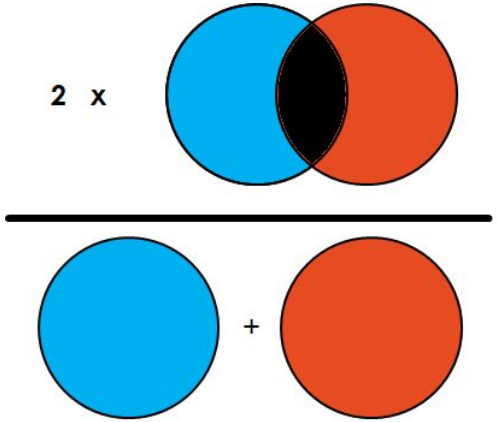
Metrics

IoU

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

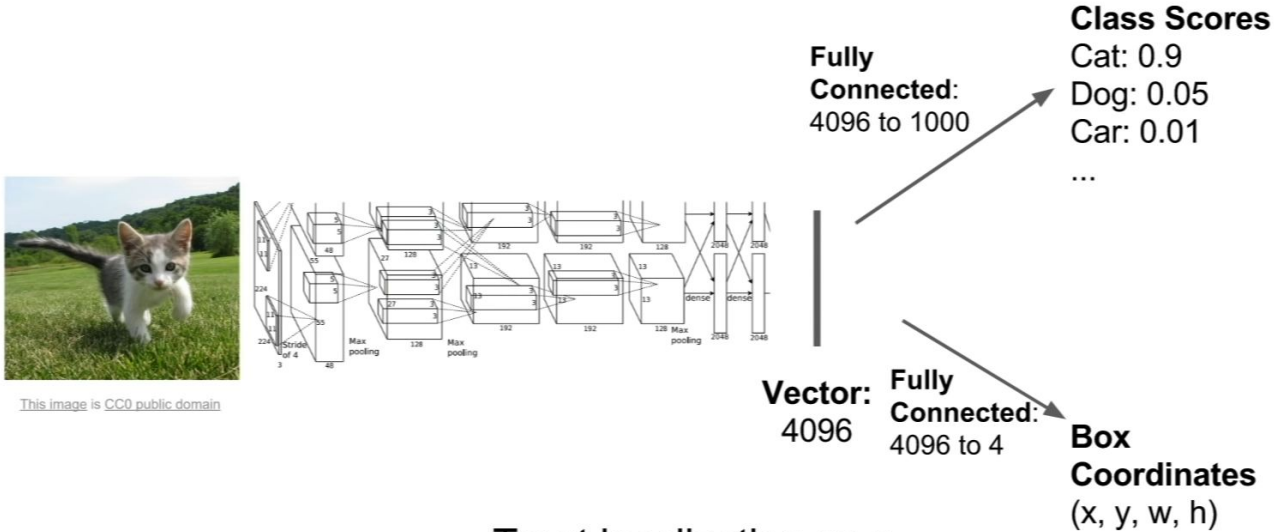


Dice Coefficient



Instance segmentation

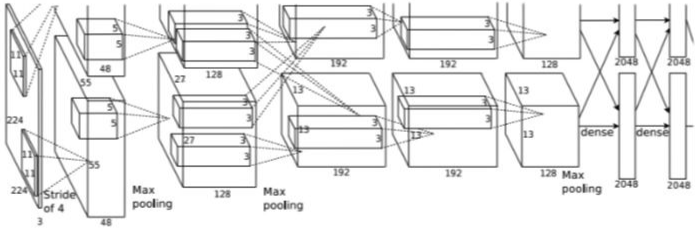
Classification + Localization



Treat localization as a regression problem!

Initial idea - sliding window[again]

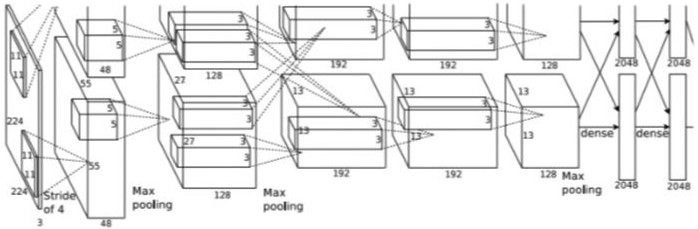
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? NO
Background? YES

Initial idea - sliding window[again]

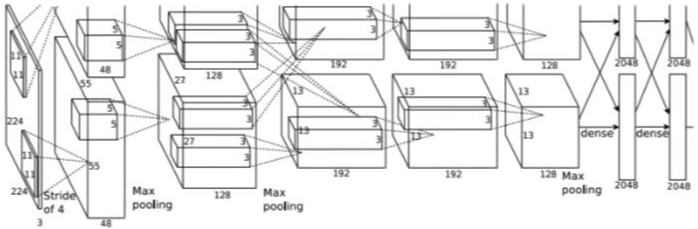
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

Initial idea - sliding window[again]

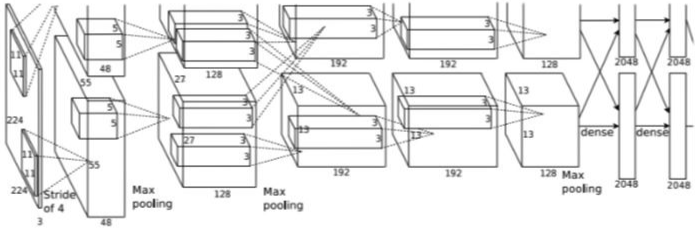
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

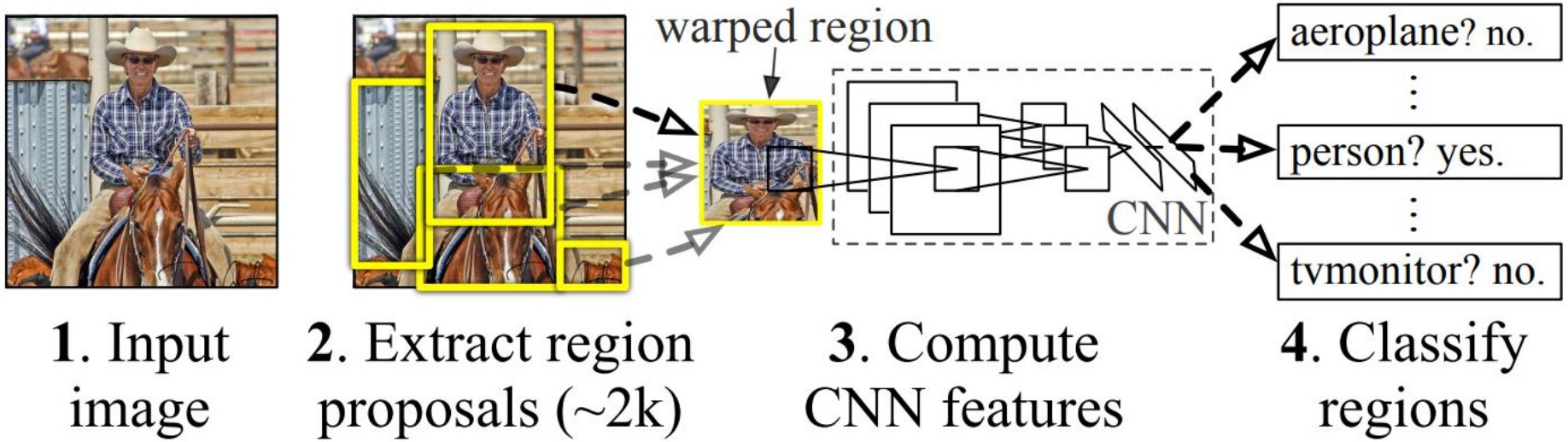
Initial idea - sliding window[again]

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? YES
Background? NO

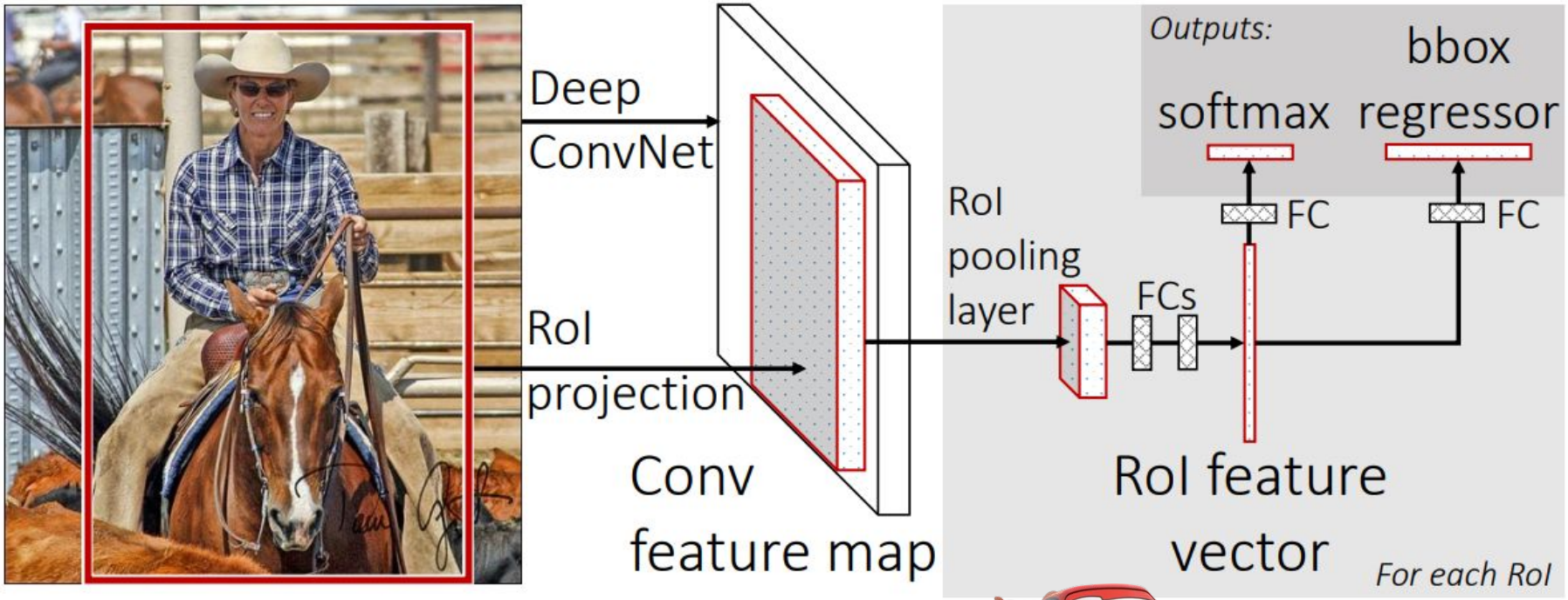
R-CNN - 2012



5. (x, y, w, h) BBox Regression

40 seconds of inference

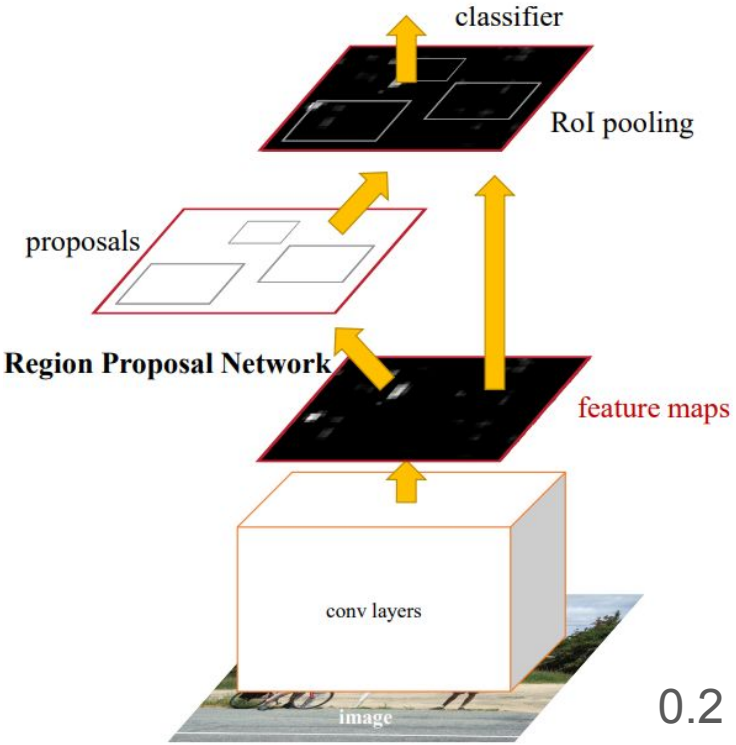
Fast-RCNN

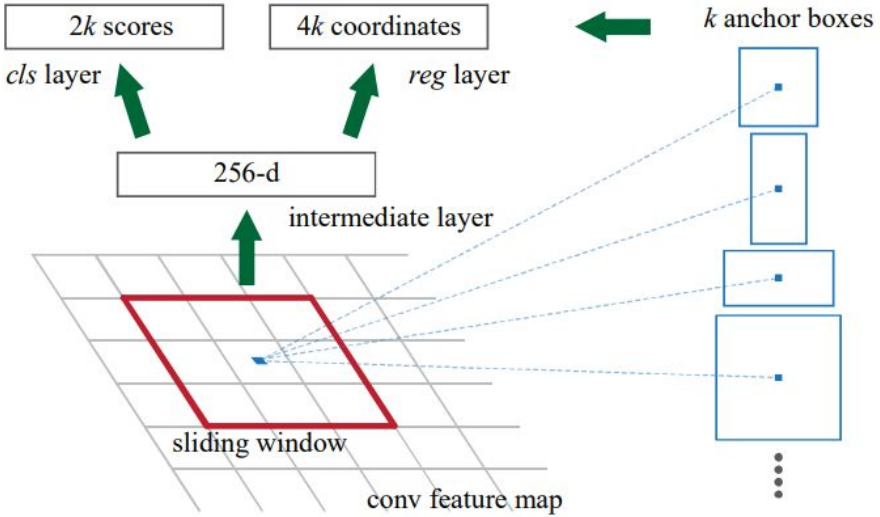


2 seconds of inference


MOAR!!!
MOAR!!!!!!
imgflip.com

Faster-RCNN





0.2 seconds of inference

Mask-RCNN

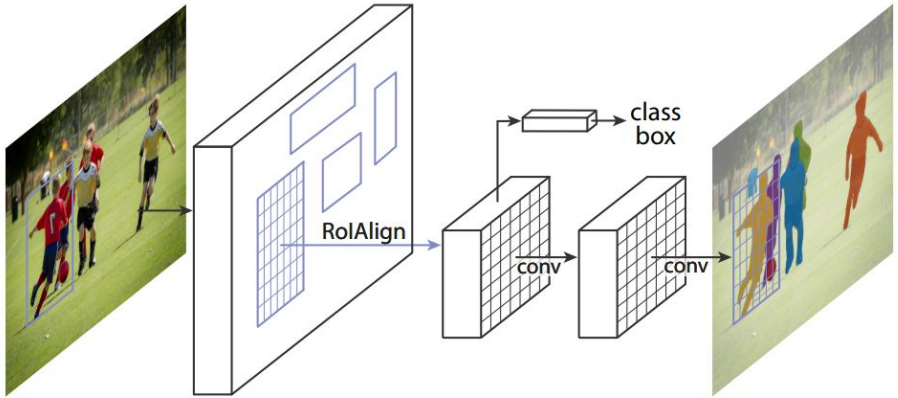
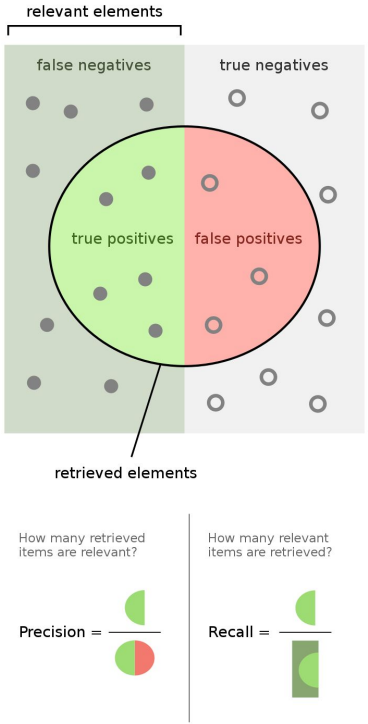


Figure 1. The **Mask R-CNN** framework for instance segmentation.

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MNC [10]	ResNet-101-C4	24.6	44.3	24.8	4.7	25.9	43.6
FCIS [26] +OHEM	ResNet-101-C5-dilated	29.2	49.5	-	7.1	31.3	50.0
FCIS+++ [26] +OHEM	ResNet-101-C5-dilated	33.6	54.5	-	-	-	-
Mask R-CNN	ResNet-101-C4	33.1	54.9	34.8	12.1	35.6	51.1
Mask R-CNN	ResNet-101-FPN	35.7	58.0	37.8	15.5	38.1	52.4
Mask R-CNN	ResNeXt-101-FPN	**37.1**	**60.0**	**39.4**	**16.9**	**39.9**	**53.5**

Metrics - mAP



Average Precision (AP):

AP	% AP at IoU=.50:.05:.95 **(primary challenge metric)**
$AP^{IoU=.50}$	% AP at IoU=.50 (PASCAL VOC metric)
$AP^{IoU=.75}$	% AP at IoU=.75 (strict metric)

AP Across Scales:

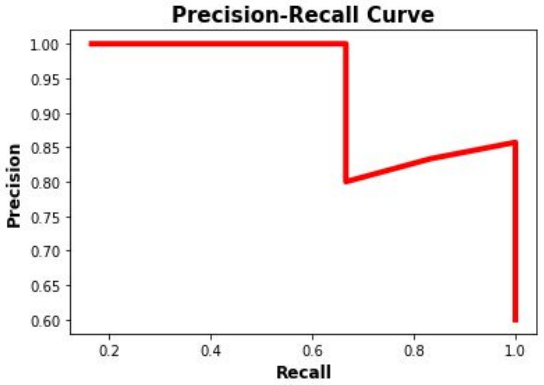
AP^{small}	% AP for small objects: area < 32^2
AP^{medium}	% AP for medium objects: 32^2 < area < 96^2
AP^{large}	% AP for large objects: area > 96^2

Average Recall (AR):

$AR^{max=1}$	% AR given 1 detection per image
$AR^{max=10}$	% AR given 10 detections per image
$AR^{max=100}$	% AR given 100 detections per image

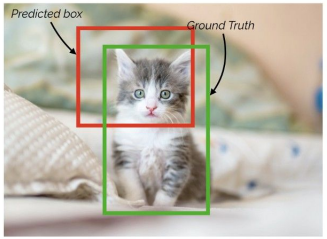
AR Across Scales:

AR^{small}	% AR for small objects: area < 32^2
AR^{medium}	% AR for medium objects: 32^2 < area < 96^2
AR^{large}	% AR for large objects: area > 96^2



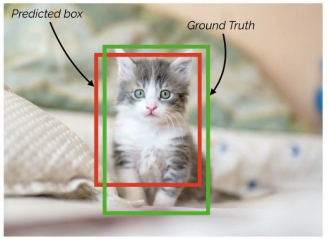
If IoU threshold = 0.5

False Positive (FP)



IoU = ~0.3

True Positive (TP)



IoU = ~0.7

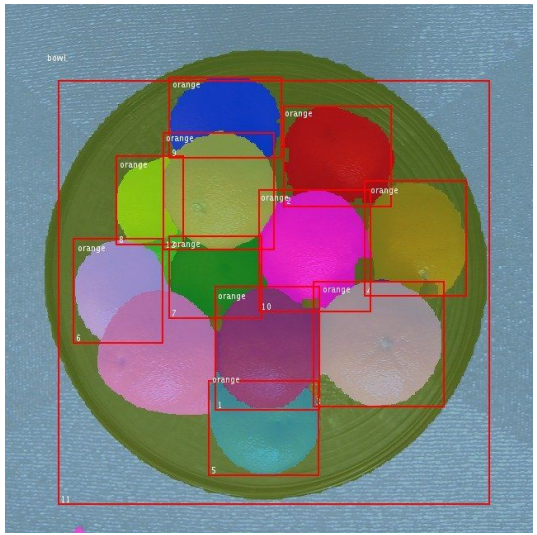
Data

Datasets

- COCO
- CityScapes
- ADE20K

Formats:

- RGB masks
- COCO-format (json)



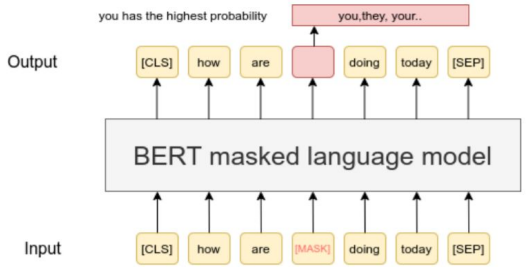
mmDetection demo

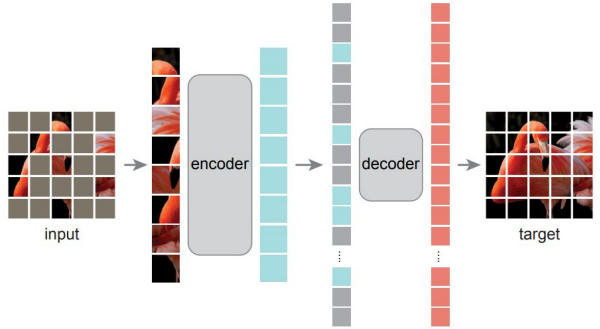
State of the art

Segment Anything

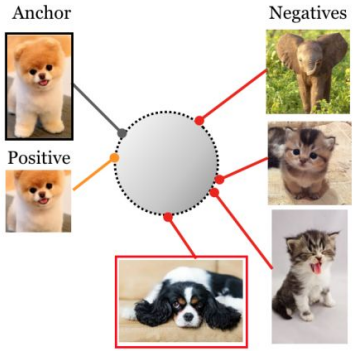
AI Clearing

Self-supervised models





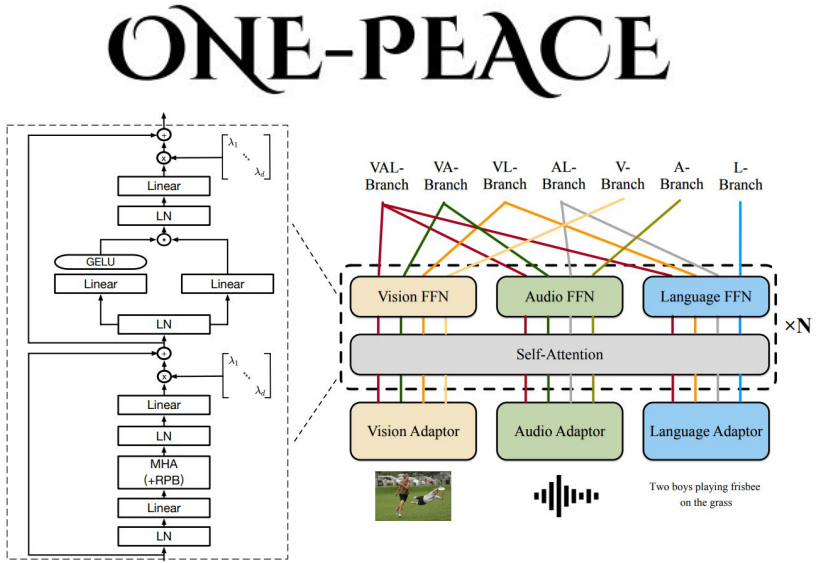
Masked Image Modelling



Contrastive Learning

Multimodal models

Rank	Model	Validation mIoU	Test Score	Params (M)	GFLOPs (512 x 512)	GFLOPs	Extra Training Data	Paper	Code	Result	Year	Tags
1	**ONE-PEACE**	63.0		1500			×	ONE-PEACE: Exploring One General Representation Model Toward Unlimited Modalities			2023	
2	**InternImage-H**	62.9		1310		4635	×	InternImage: Exploring Large-Scale Vision Foundation Models with Deformable Convolutions			2022	
3	**M3I Pre-training** (InternImage-H)	62.9		1310			×	Towards All-in-one Pre-training via Maximizing Multi-modal Mutual Information			2022	



We are looking for talented people

We are looking for you!

- Paid internships
- ESOP - after internship participate in company’s success
- Learn from the best experts in ML and Software Development
- Train state of the art models for construction sites analytics
- We have:
  - Terabytes of data
  - Compute clusters
  - World’s most advanced projects

AI Clearing

Send CV to:
jf@aiclearing.com

Sources

- https://persci.mit.edu/pub_pdfs/adelson_spie_01.pdf
- https://arxiv.org/abs/1605.06211
- https://arxiv.org/abs/1505.04597
- https://arxiv.org/abs/1706.05587
- https://arxiv.org/abs/2107.06278
- https://www.youtube.com/watch?v=nDPWywWRIRo
- https://arxiv.org/abs/1311.2524
- https://arxiv.org/abs/1504.08083
- https://arxiv.org/abs/1506.01497
- https://arxiv.org/abs/1703.06870
- https://arxiv.org/abs/2304.02643
- https://arxiv.org/abs/2002.05709
- https://arxiv.org/abs/2106.08254
- https://arxiv.org/abs/2305.11172
- https://github.com/open-mmlab/mmdetection

Questions?

Thank You!