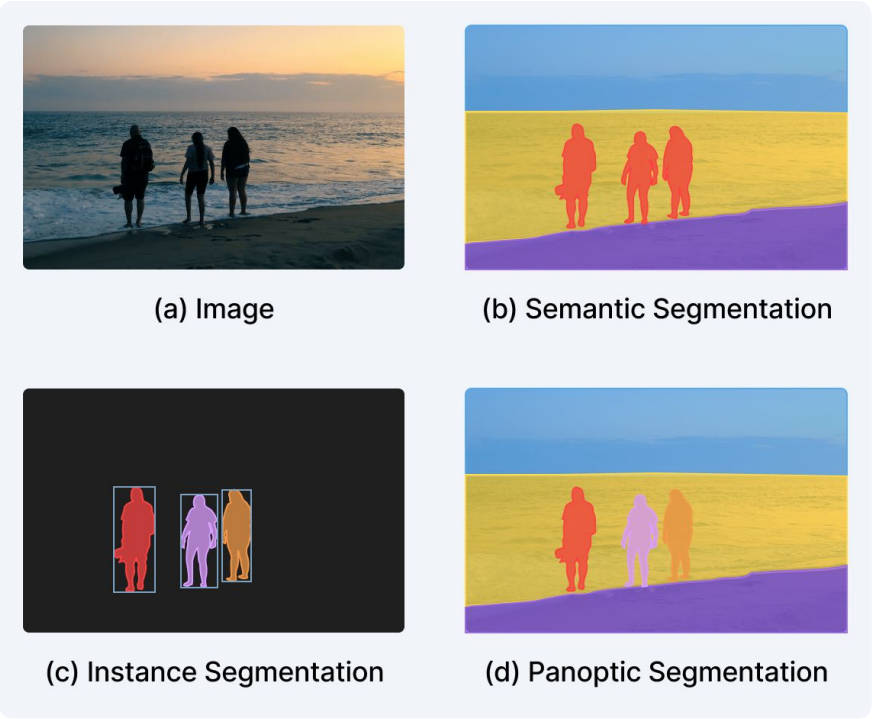
Image Segmentation

Matt Hyatt
Mujtaba Nazari

Image Segmentation

For each pixel,

- What class does that pixel belong to?
- Which instance of that class?

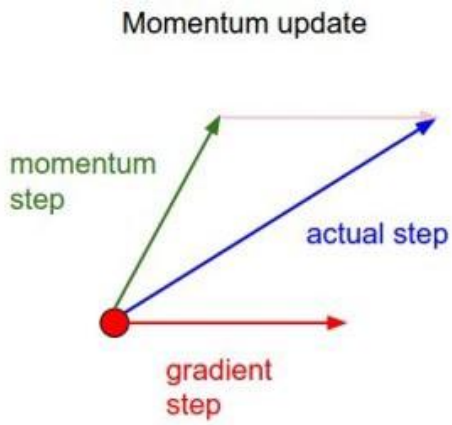


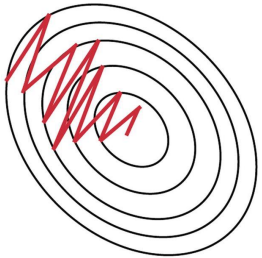
Important Prerequisites

Training details

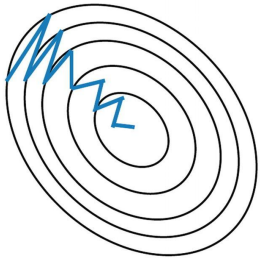
Weight decay

- Also called L2 regularization
  - $L_{new}(w) = L_{original}(w) + \lambda w\ Tw.$
- Penalizes large weights
- Encourages building models with less complexity





Stochastic Gradient Descent **without** Momentum

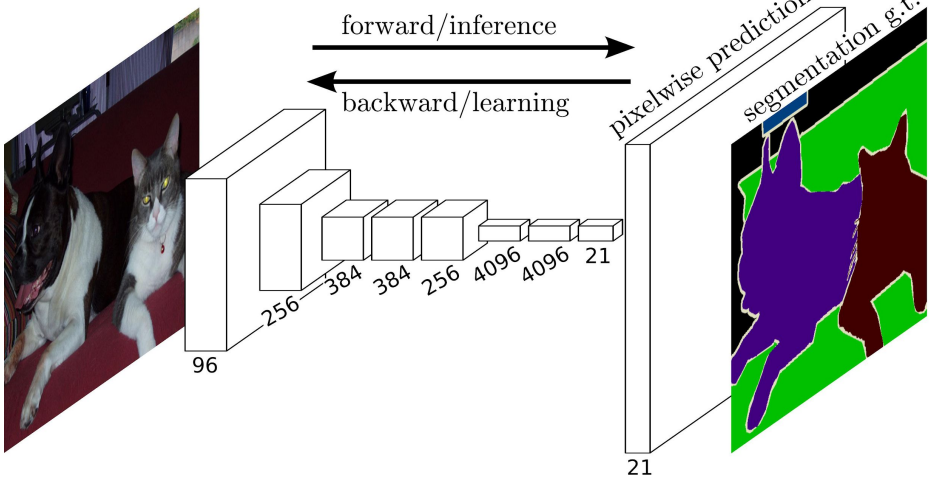


Stochastic Gradient Descent **with** Momentum

Momentum

- extension to gradient descent optimization
- build inertia in a direction in the search space
- overcome oscillations of noisy gradients and coast across flat spots

FCN

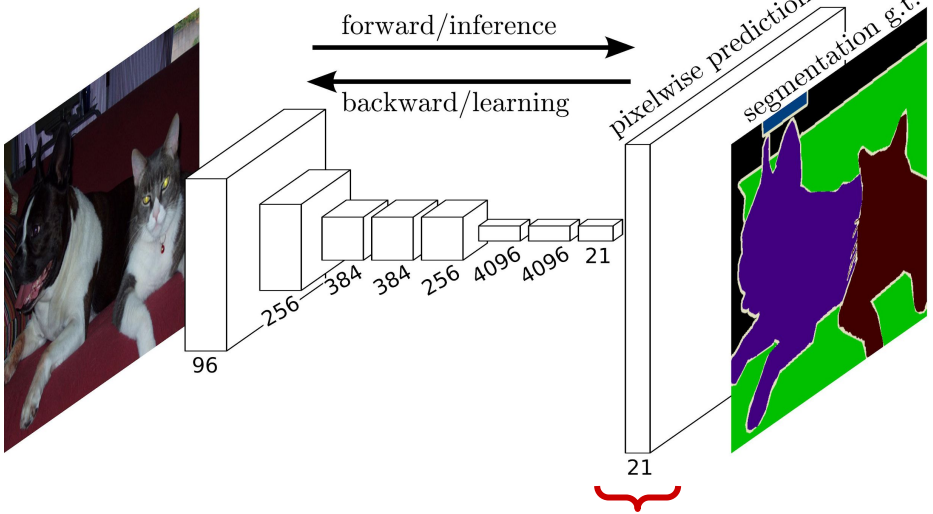


CVPR 2015

FCN

Contributions:

- End-to-end (no pre/postprocessing)
- Fully convolutional (no MLP)
  - channelwise class activation map …
- Deconvolution for upsampling

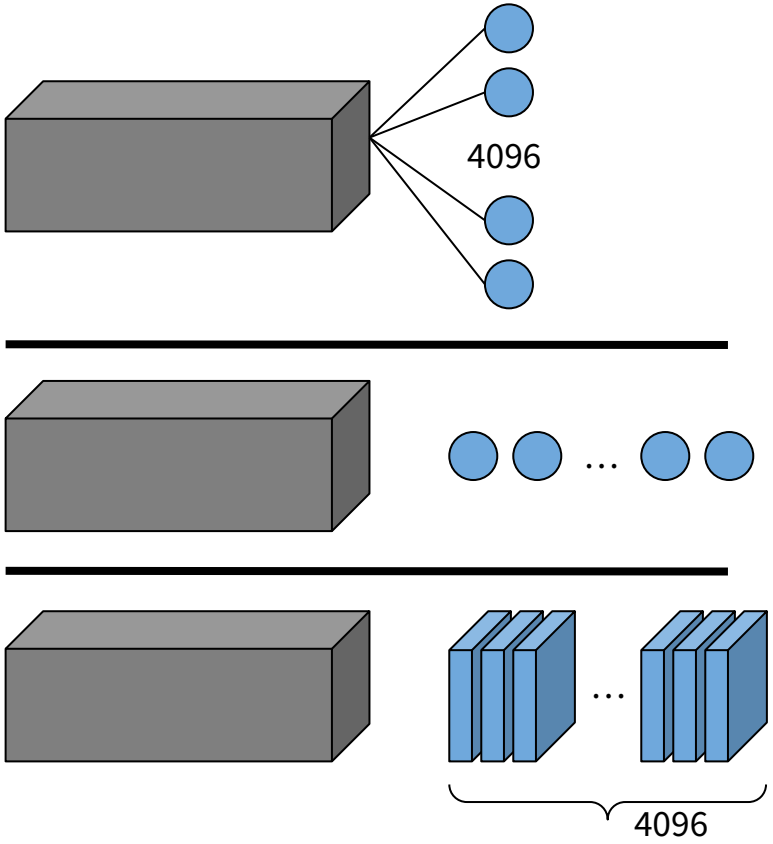


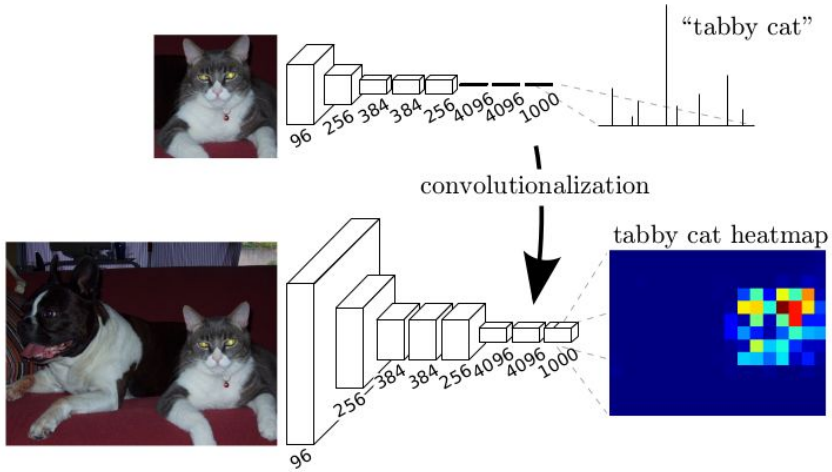
for each pixel, which (of 21) classes is most likely?

Notes:

- inference less than ⅕ second
- “We find that __ is unnecessary”
  - Many experimental changes make no difference

FCN convolutionization

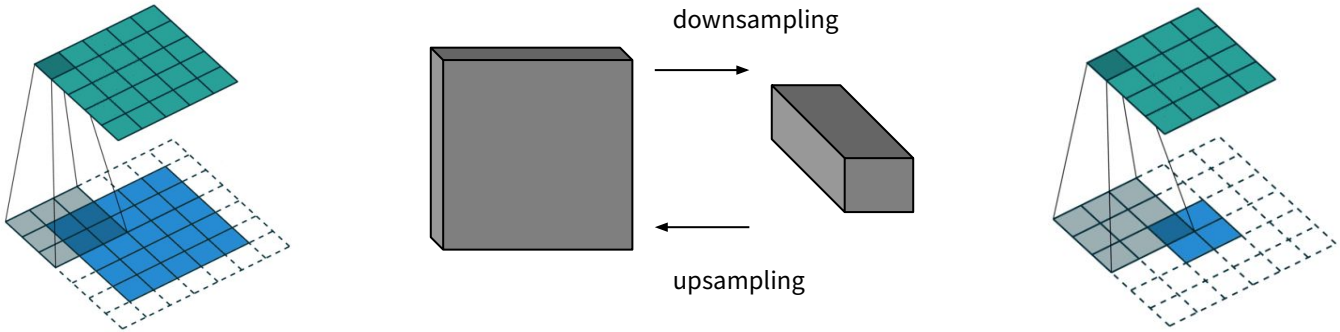




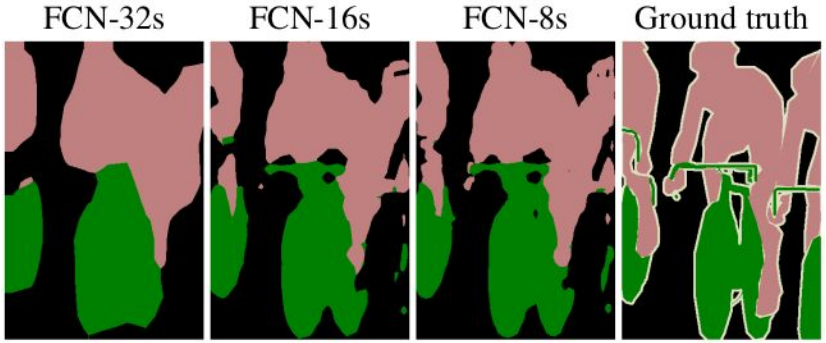
FCN deconv

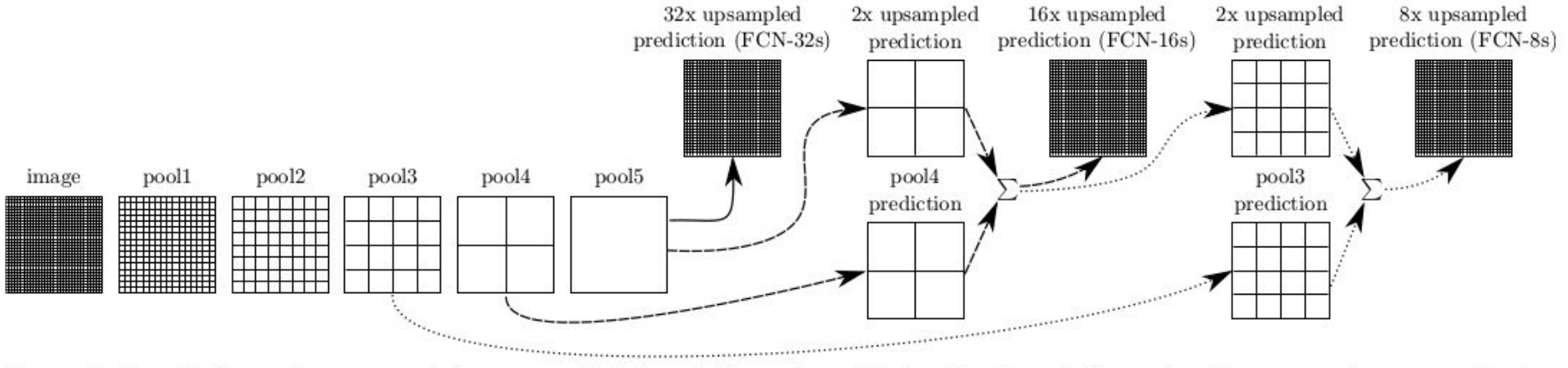
Called transposed convolution now

- *tf.keras.layers.Conv2DTranspose()*



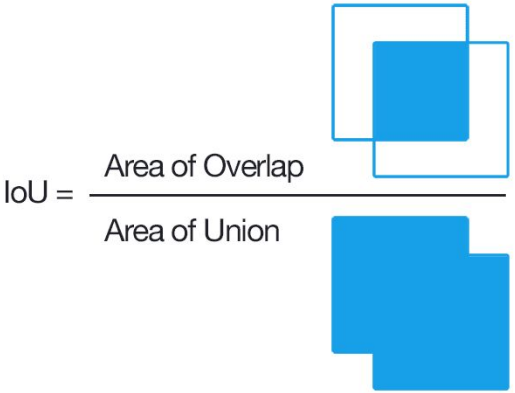
FCN Architecture

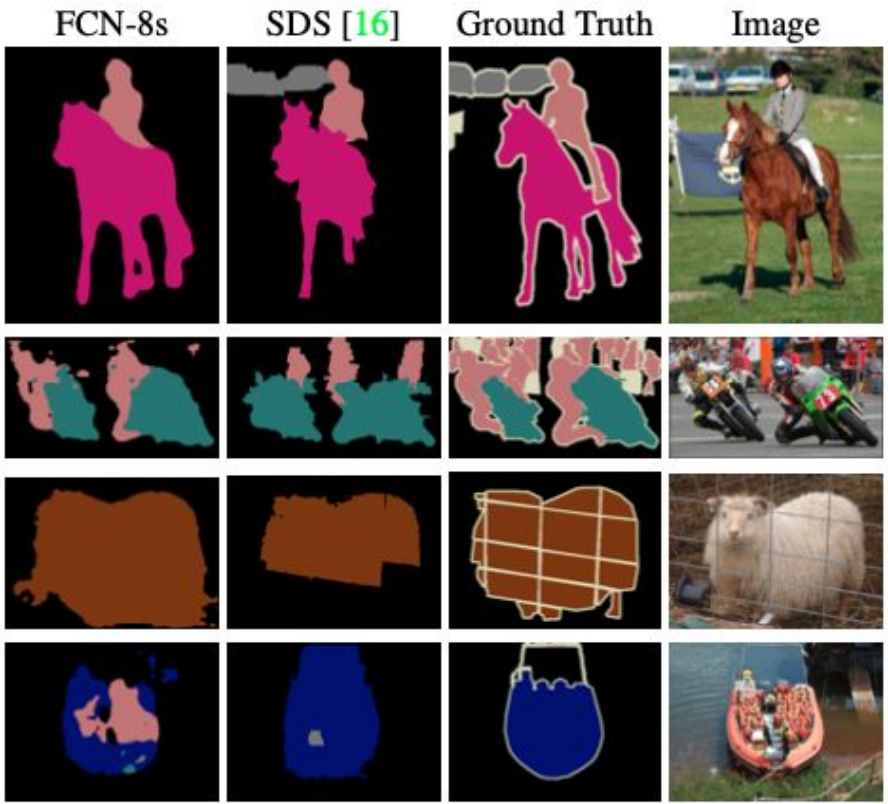


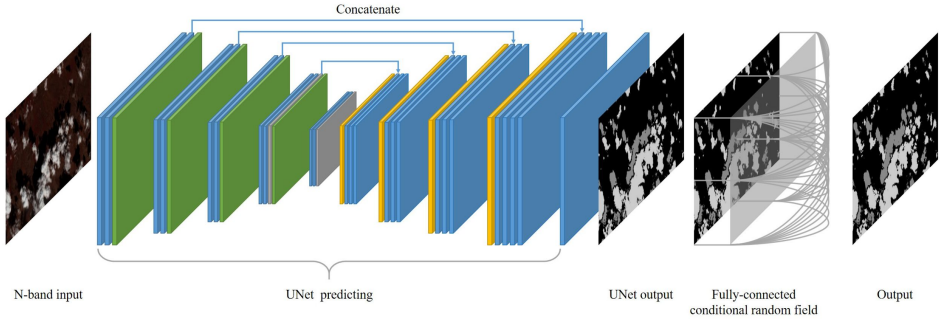


FCN results

- SOTA on performance metrics
  - (20% relative improvement)
- Pascal VOC
  - 62.2 mIoU



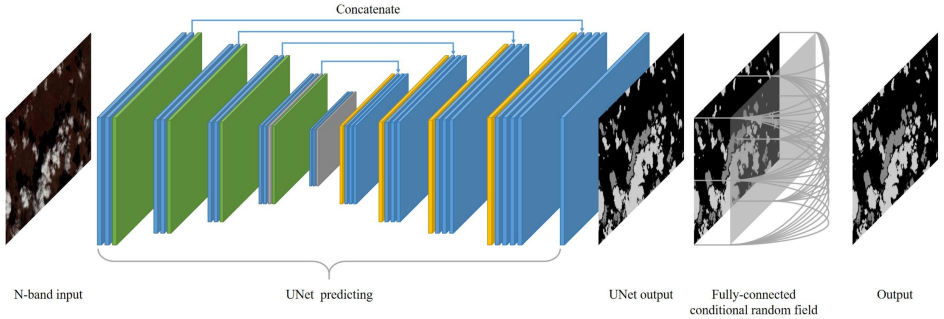




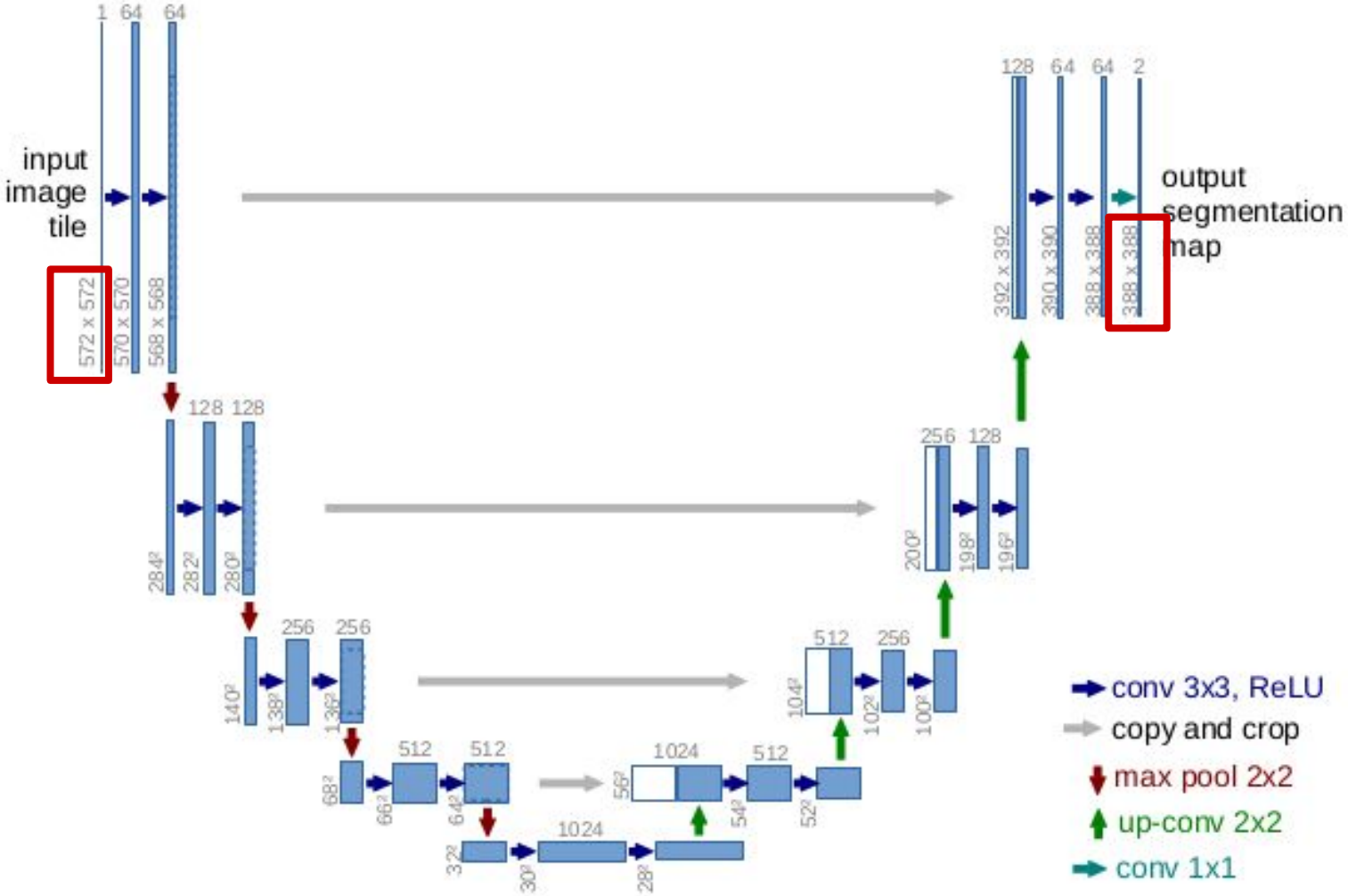
UNet

MICCAI 2015

UNet

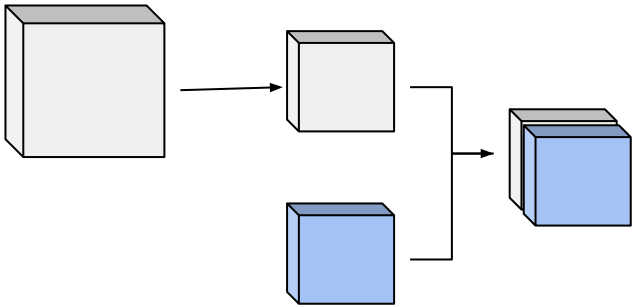


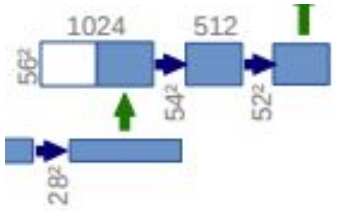
- Present a network and training strategy that relies on the strong use of data augmentation to use the available annotated samples more efficiently.
- We show that such a network can be trained end-to-end from very few images and outperforms the prior best method
- The network is fast. Segmentation of a 512x512 image takes less than a second on a recent GPU.
- their success was limited due to the size of the available training sets and the size of the considered networks.

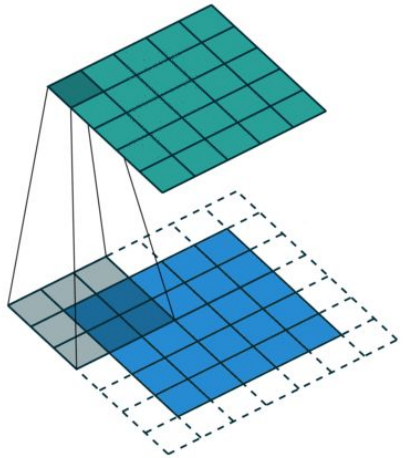

input image tile
572 x 572
570 x 570
568 x 568
1 64 64
128 128
284²
282²
280²
256 256
140²
138²
136²
512 512
68²
66²
64²
1024
32²
30²
28²
1024
56²
512
54²
512
52²
256
104²
102²
100²
256 128
200²
198²
196²
128 64 64 2
392 x 392
390 x 390
388 x 388
388 x 388
output segmentation map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

Unet Architecture

- The authors have used the unpadded convolution, So, the output is smaller than the input.
- Unet can be divided two encoder and decoder section.
- During upsampling the corresponding feature map from the encoder is concatenated and cropped to fit previous upsampled features

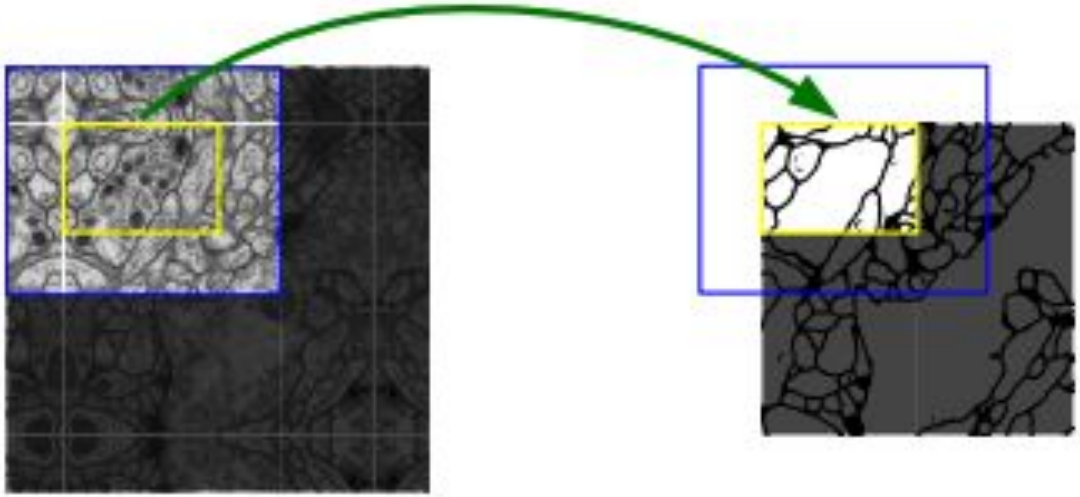




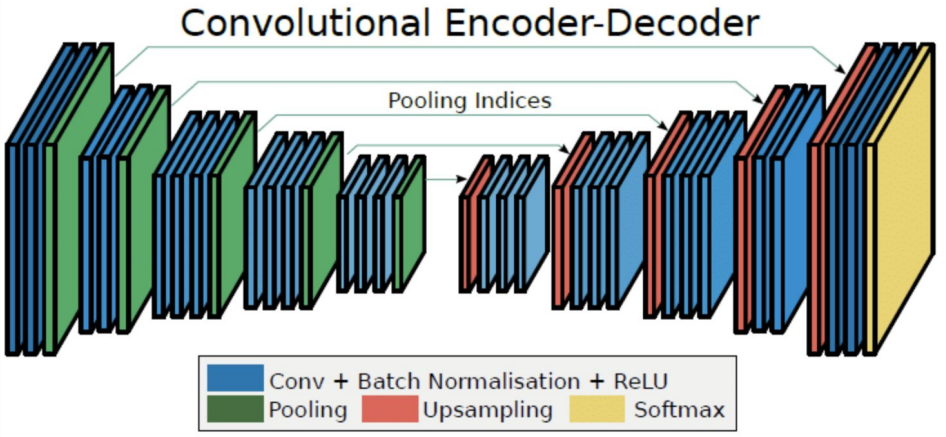


UNet Padding

- mirror the feature map when upsampling edges



SegNet

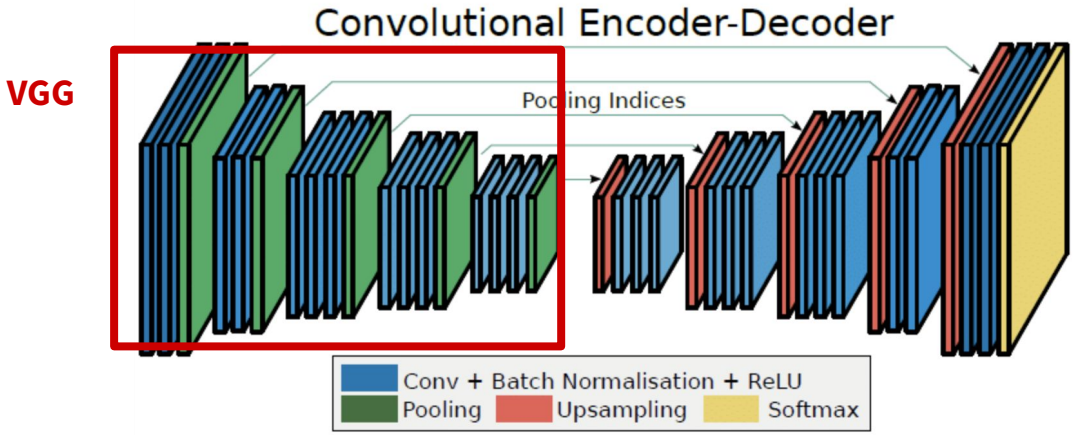


TPAMI 2017

SegNet

Contributions:

- New upsampling technique
  - decreases the need for learning
  - improves boundary delineation
- Study FCN decoder design



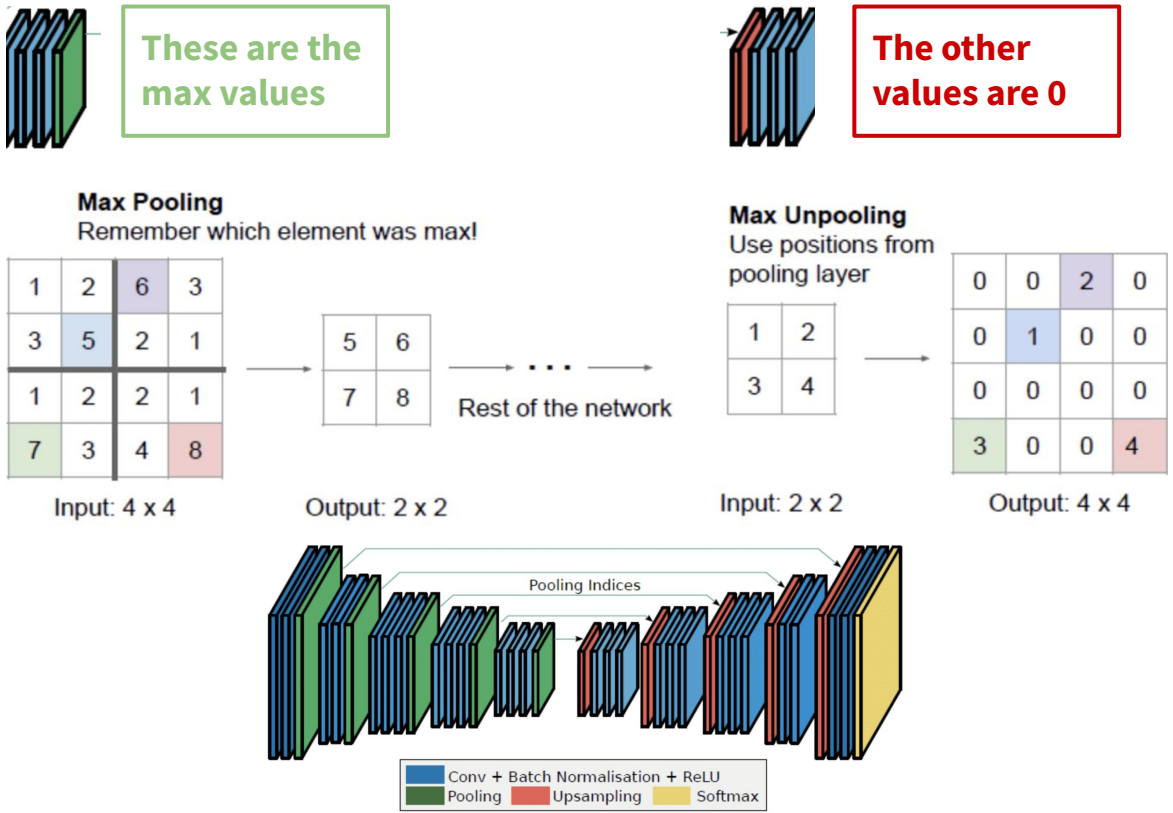
Design Principles:

- Smooth segmentation of large objects
- Attend to small objects
- End to end architecture

SegNet Unpooling

“bed of nails” unpooling

- preserves spacial info
- does not have to be learned
- Copy the indices instead of the unit.
  - Better GPU usage
- Uses convolution instead of deconvolution (layer, weight, biases, learning weight)
-



SegNet Results

- improvements on global and class average accuracy
- more impressively, segnet is able to draw clean boundaries
- https://www.youtube.com/watch?v=CxanE_W46ts

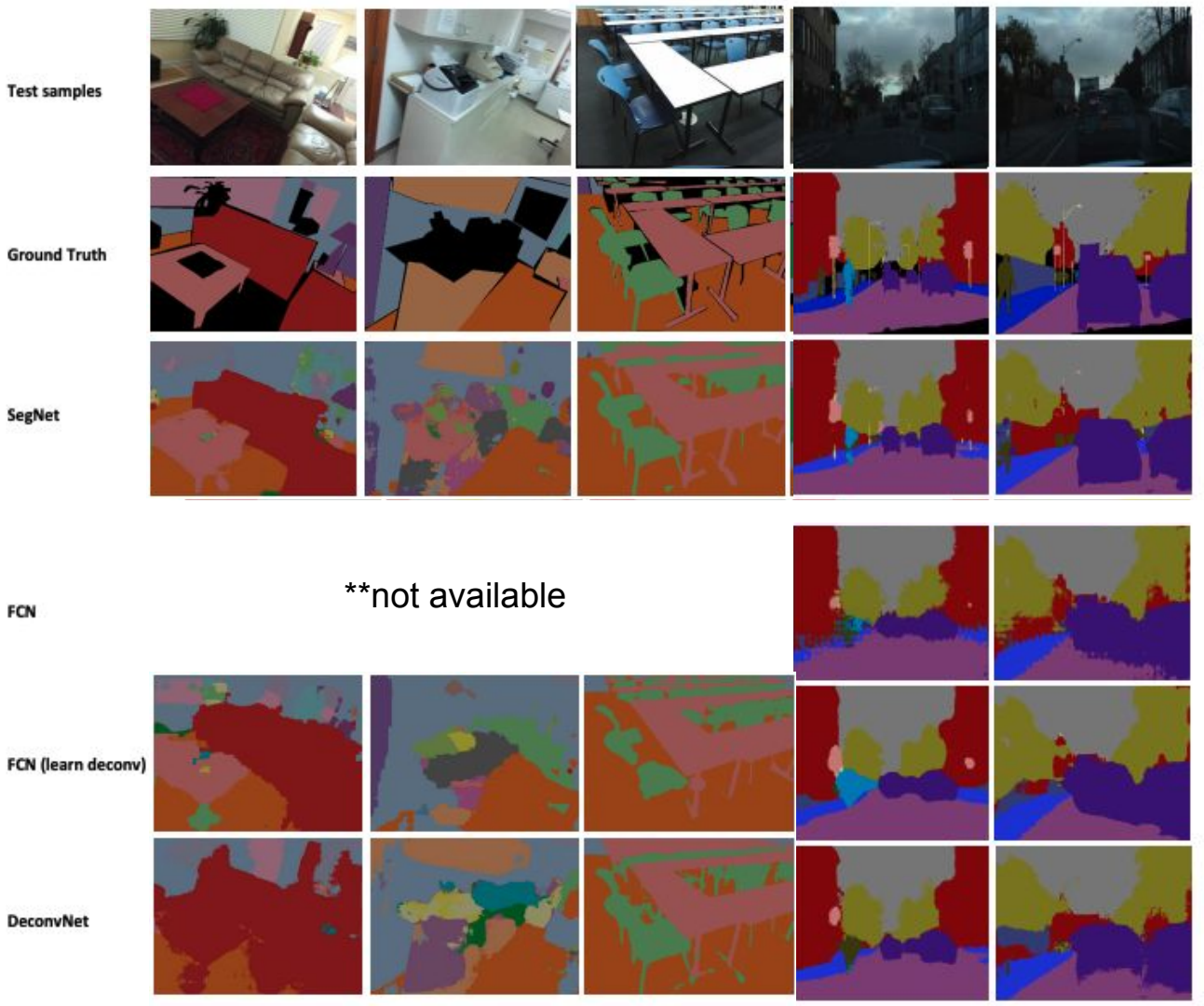
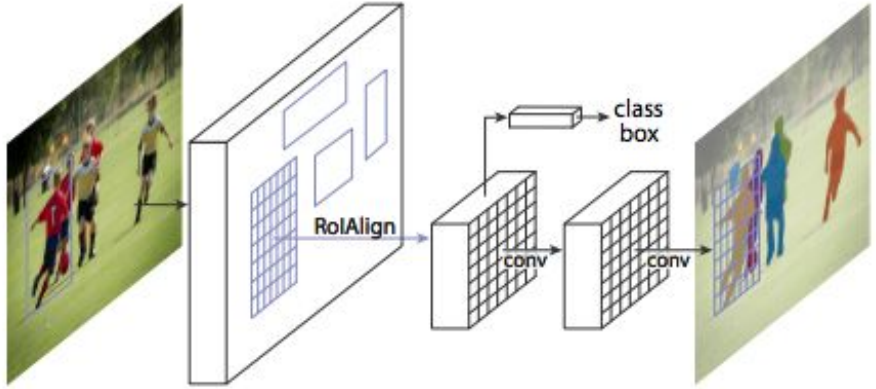


TABLE 2
Quantitative Comparisons of SegNet with Traditional Methods on the CamVid 11 Road Class Segmentation Problem [22]

Method	Building	Tree	Sky	Car	Sign-Symbol	Road	Pedestrian	Fence	Column-Pole	Side-walk	Bicydist	Class avg.	Global avg.	mIoU	BF
SfM+Appearance [28]	46.2	61.9	89.7	68.6	42.9	89.5	53.6	46.6	0.7	60.5	22.5	53.0	69.1	n/a*	
Boosting [29]	61.9	67.3	91.1	71.1	58.5	92.9	49.5	37.6	25.8	77.8	24.7	59.8	76.4	n/a*	
Dense Depth Maps [32]	85.3	57.3	95.4	69.2	46.5	**98.5**	23.8	44.3	22.0	38.1	28.7	55.4	82.1	n/a*	
Structured Random Forests [31]	n/a											51.4	72.5	n/a*	
Neural Decision Forests [64]	n/a											56.1	82.1	n/a*	
Local Label Descriptors [65]	80.7	61.5	88.8	16.4	n/a	98.0	1.09	0.05	4.13	12.4	0.07	36.3	73.6	n/a*	
Super Parsing [33]	87.0	67.1	96.9	62.7	30.1	95.9	14.7	17.9	1.7	70.0	19.4	51.2	83.3	n/a*	
SegNet (3.5K dataset training - 140K)	**89.6**	**83.4**	96.1	**87.7**	52.7	96.4	**62.2**	**53.45**	**32.1**	**93.3**	**36.5**	**71.20**	**90.40**	60.10	46.84
CRF based approaches															
Boosting + pairwise CRF [29]	70.7	70.8	94.7	74.4	55.9	94.1	45.7	37.2	13.0	79.3	23.1	59.9	79.8	n/a*	
Boosting+Higher order [29]	84.5	72.6	**97.5**	72.7	34.1	95.3	34.2	45.7	8.1	77.6	28.5	59.2	83.8	n/a*	
Boosting+Detectors+CRF [30]	81.5	76.6	96.2	78.7	40.2	93.9	43.0	47.6	14.3	81.5	33.9	62.5	83.8	n/a*	

*SegNet outperforms all the other methods, including those using depth, video and/or CRF's on the majority of classes. In comparison with the CRF based methods SegNet predictions are more accurate in 8 out of the 11 classes. It also shows a good ≈ 10 percent improvement in class average accuracy when trained on a large dataset of 3.5 K images. Particularly noteworthy are the significant improvements in accuracy for the smaller/thinner classes. * Note that we could not access predictions for older methods for computing the mIoU, BF metrics.*

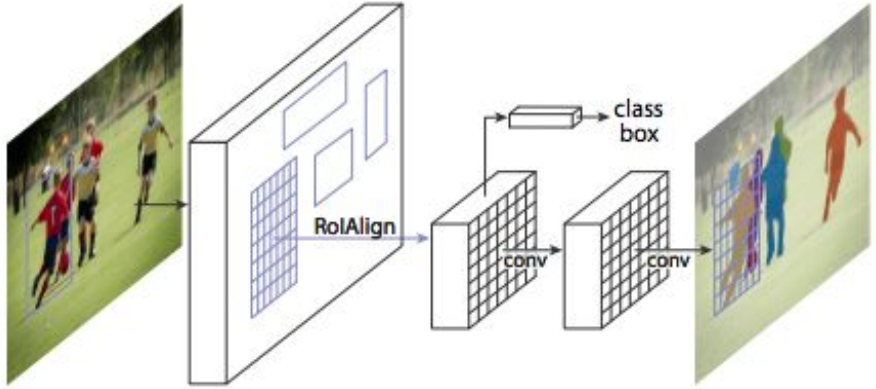
Mask R-CNN



ICCV 2017

MRCNN

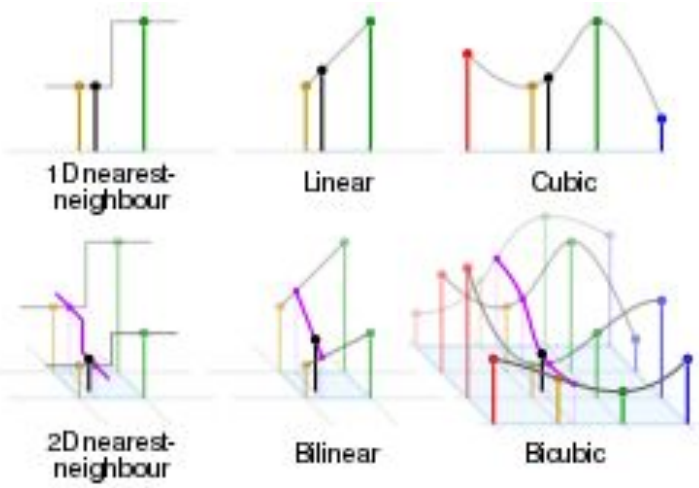
- Uses a region proposal network (RPN)
- ROI Align instead of ROI Pool
  - see Faster RCNN
- Mask R-CNN is simple to train and adds only a small overhead to Faster R-CNN, running at 5 fps.
  - Adding a branch for predicting segmentation masks on each Region of Interest (RoI).
- Parallel head (cls, bbox, mask)

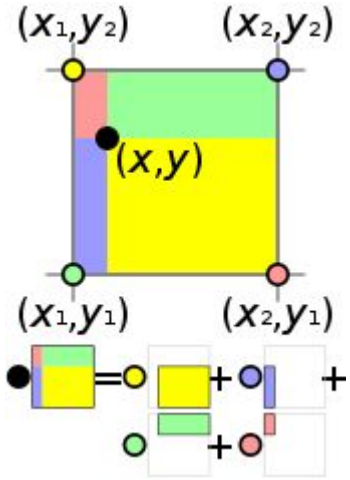


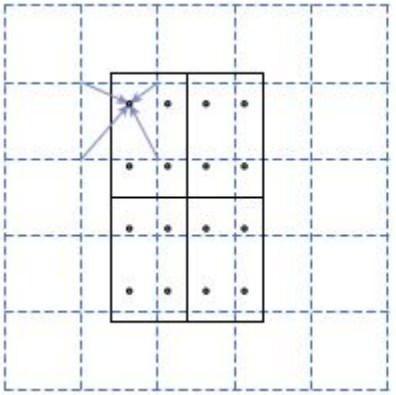
- Provide two output for each object
  - Class label
  - Bounding-box offset
- Decouple mask and class prediction

MRCNN ROIAlign

- For each region of interest (RoI)
  - How can we sample m**2 pixels?
  - Mask head will produce K m**2 feature maps
- quantization-free layer
  - preserves spatial location
  - allows us to build a per-pixel mask from the feature mask
  - pixels are sampled via bilinear interpolation

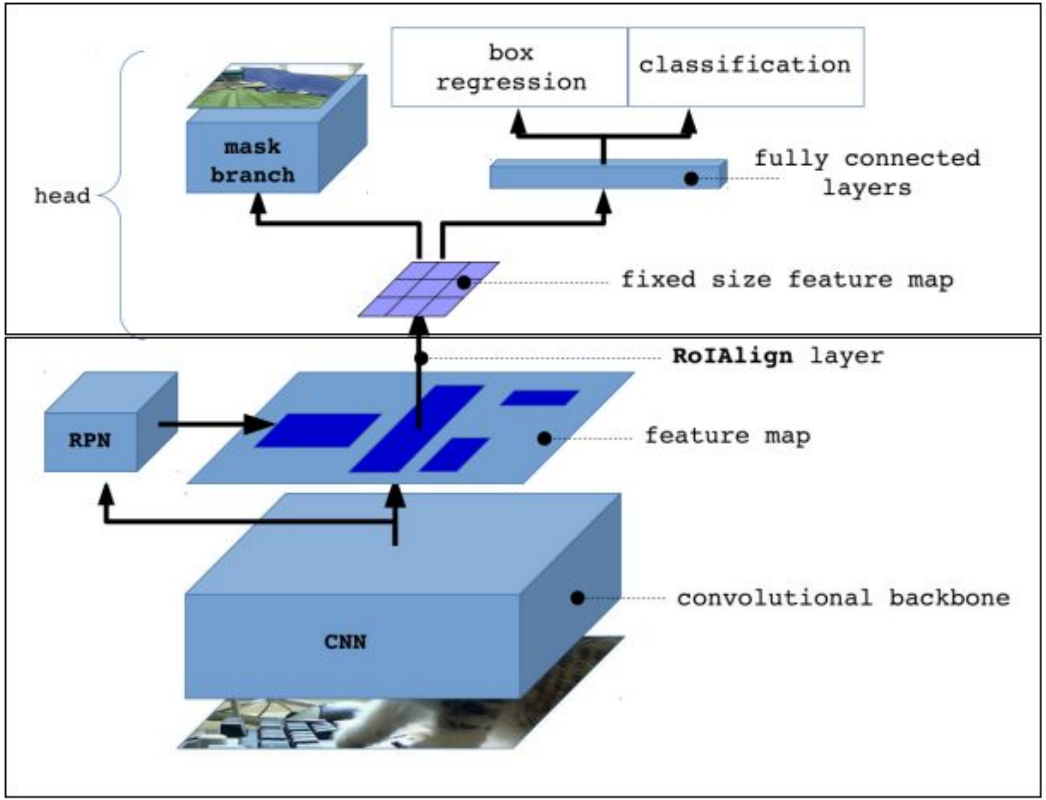






MRCNN Architecture

- Fast R-CNN
  - Region Proposal Network
- R-CNN
  - Object Detection

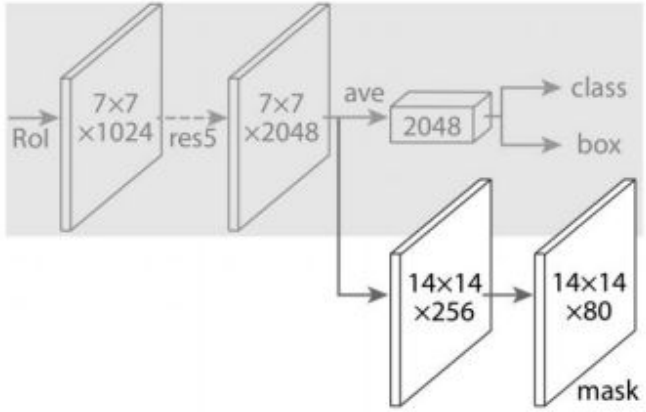


Head

Backbone
- Resnet50
- Resnet100
- FPN

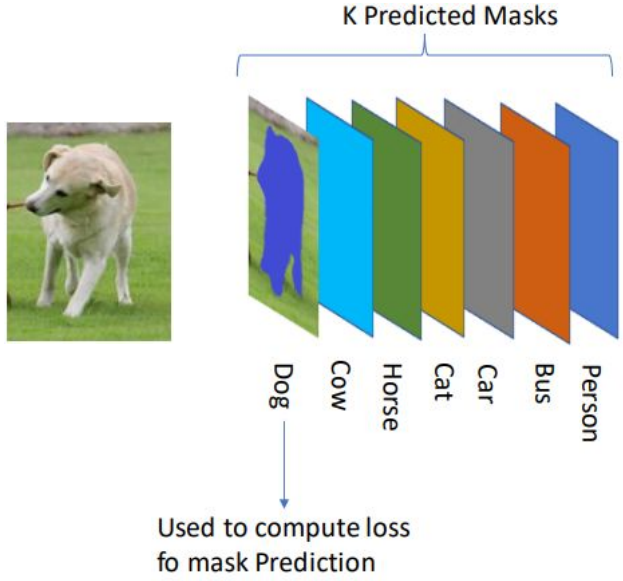
MRCNN Head

- K · (m × m) sigmoid outputs
  - pixel-wise binary classification
  - one mask for each class
- L_{mask}: mean binary loss-entropy
- RoI is positive if **IoU> 0.5** with ground-truth box else negative



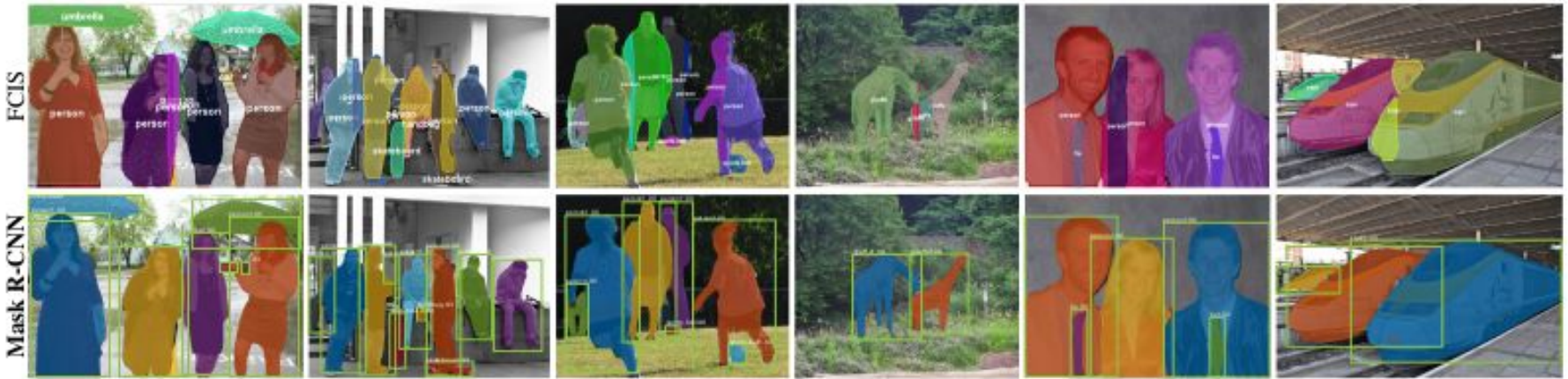
$$L = L_{cls} + L_{box} + L_{mask}$$

RoI = classification loss + bounding-box loss + mask loss

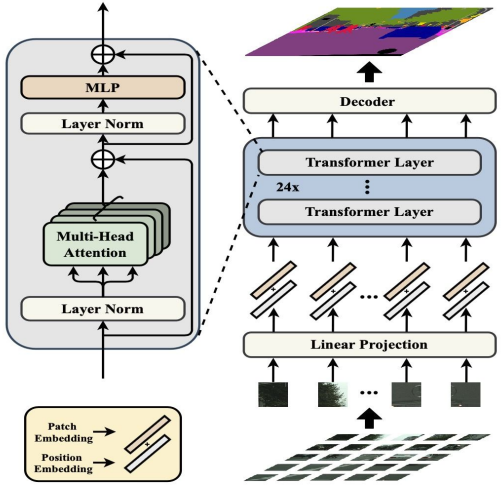


MRCNN Results

[Live Result]



SETR

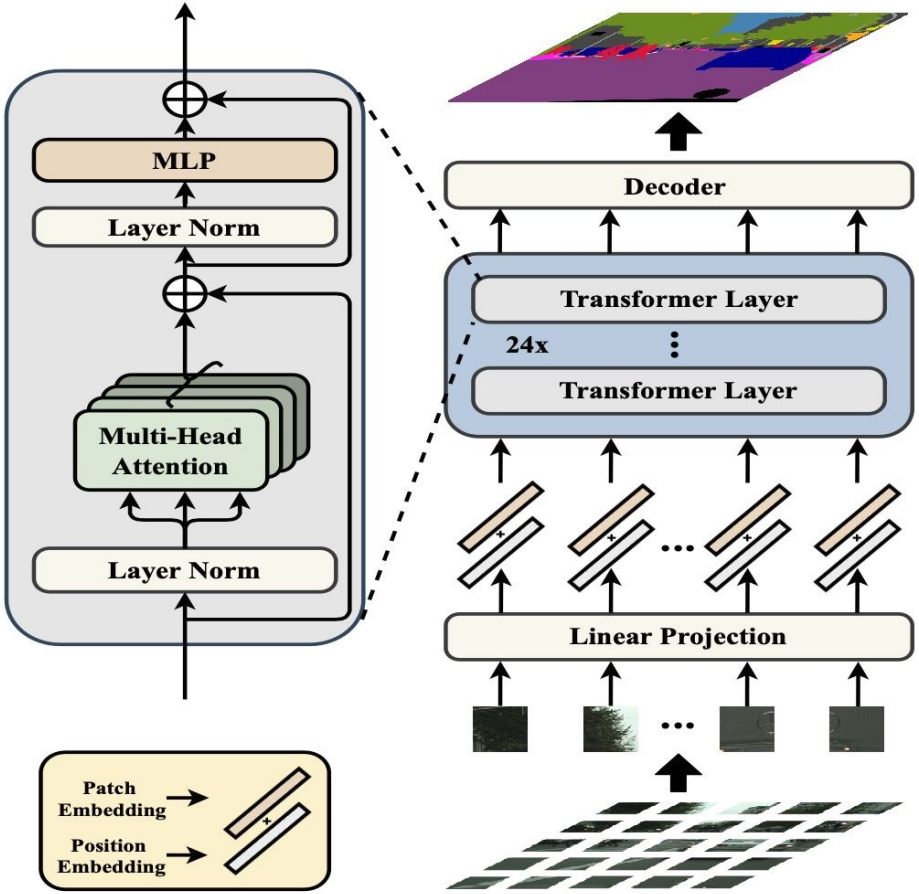


CVPR 2021

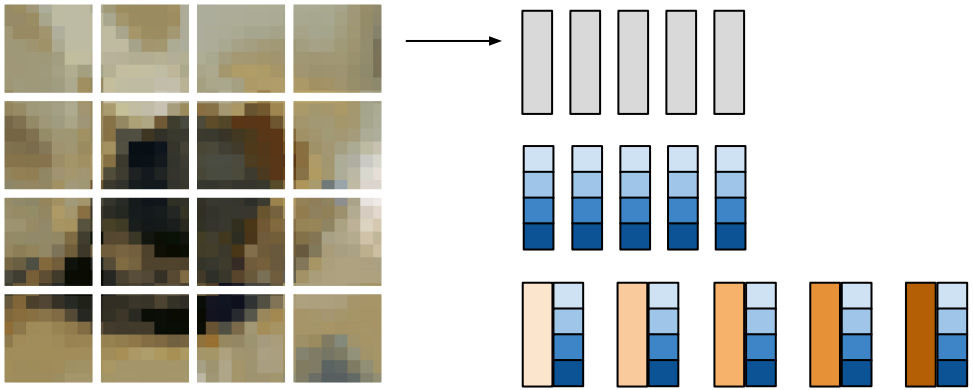
SETR

Contributions:

- “treat semantic segmentation as *sequence-to-sequence* prediction task”
- Purely transformer architecture
- No resolution downsampling



SETR embedding



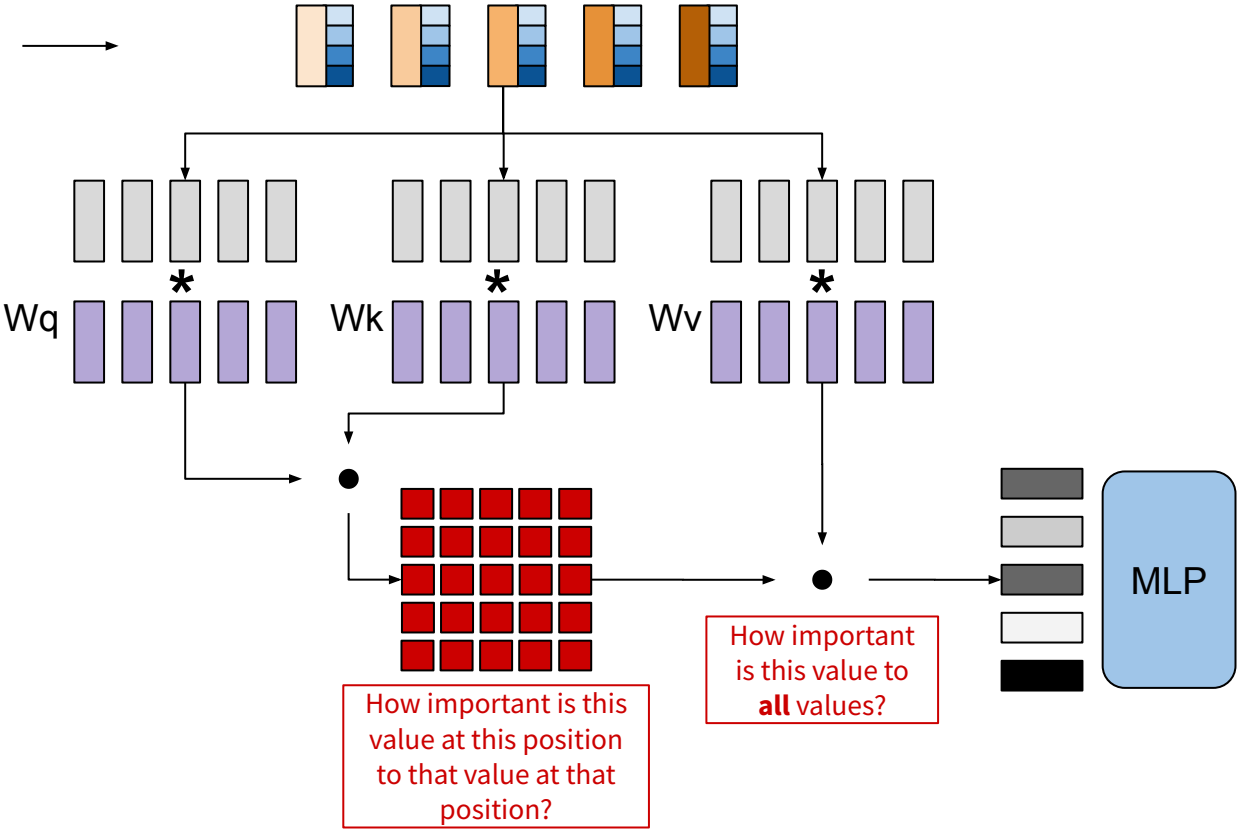
img patches

Patch embedding

Positional embedding

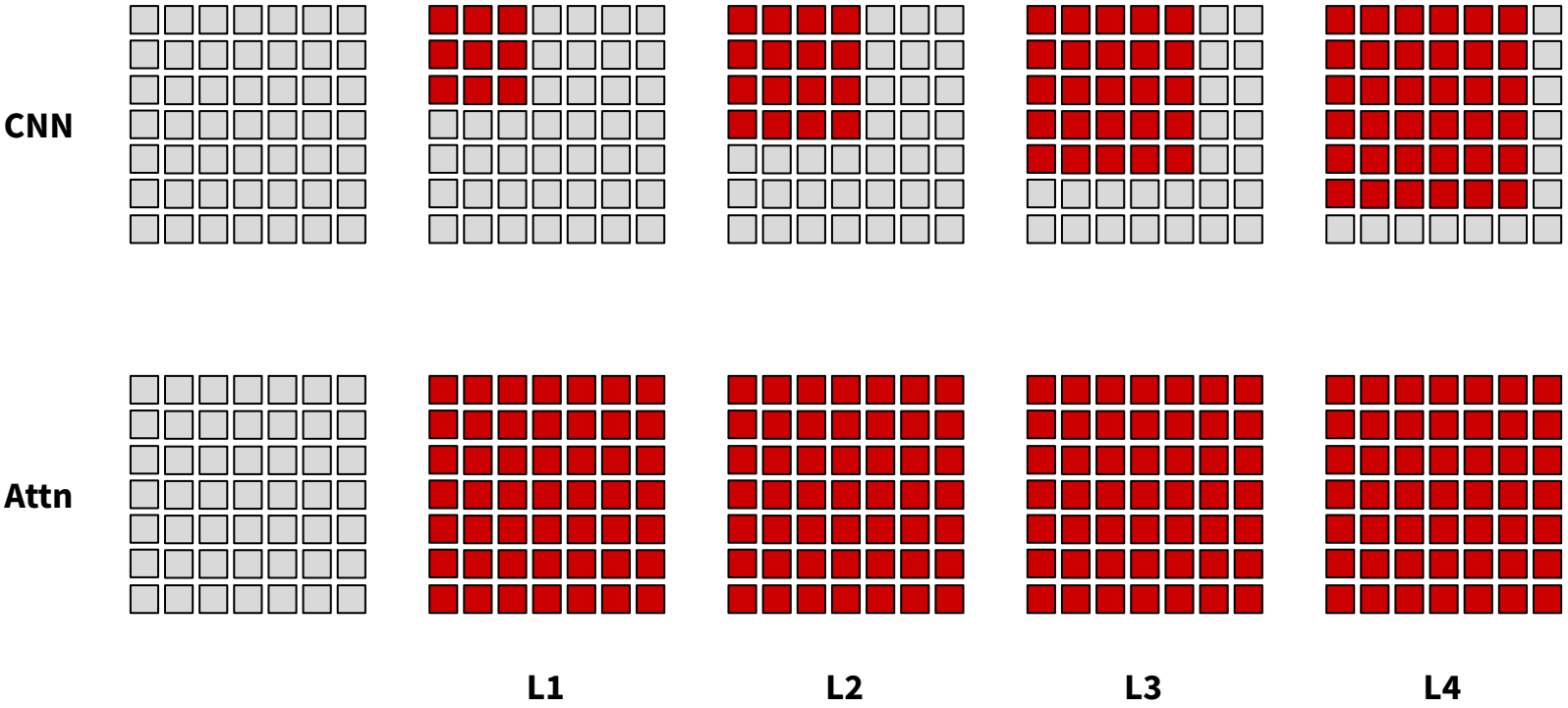
- f: $p \in R^{H*W*3} \rightarrow e \in R^{C}$
- E = {e1 + p1, e2 + p2, … , eL + pL}

SETR multi-headed self attention MHA

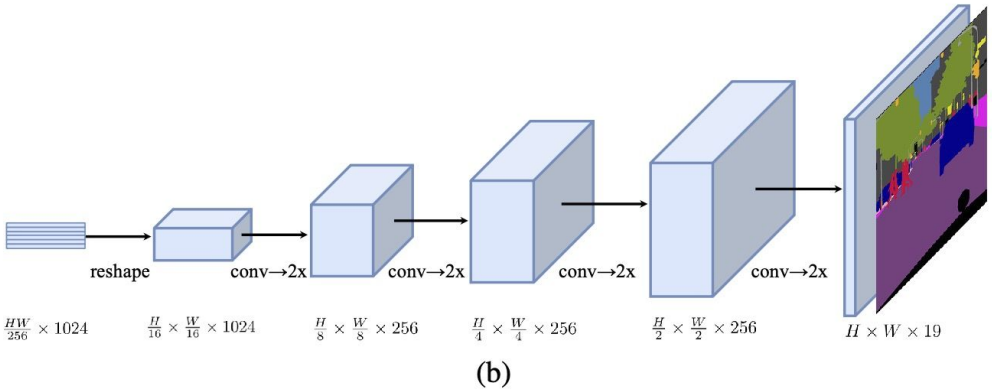


**normalization not shown

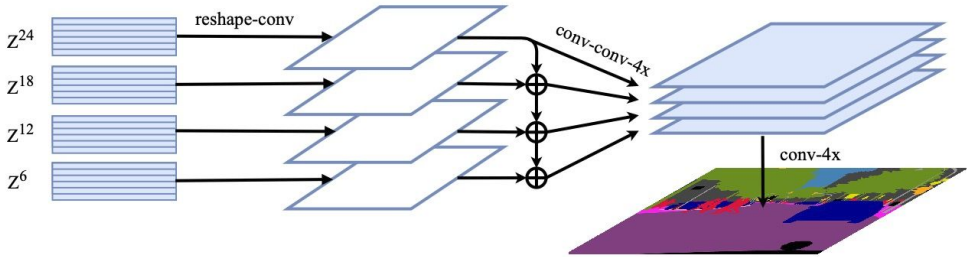
SETR receptive field



SETR decoder



(b)



Not quite “purely transformer”

SETR Results

- SOTA mIoU on ADE20K (50.28%)

(qualitative results on cityscapes)

SETR Result

Live Result Example

DETR for segmentation [huggingface]

Q&A from Summary

What makes a model fully convolutional?

- It consists of only **conv** layers … no **FC** layers

How can we deal with irregularly shaped images?

- Use a torch.Transform() to reshape or pad the image size before training

Can UNet be used outside of medical applications?

What if we increase the number of layers and implement the U-net or FCN on aspects of DNA-Mutation; and graph analysis?