Fully Convolutional Networks for Semantic Segmentation

NNFL Project

Akarsh Chaturvedi : 2016B5A80582P

Aman Garg : 2016B4A70584P

Harshavardhana Shrirup: 2016B3A70501P

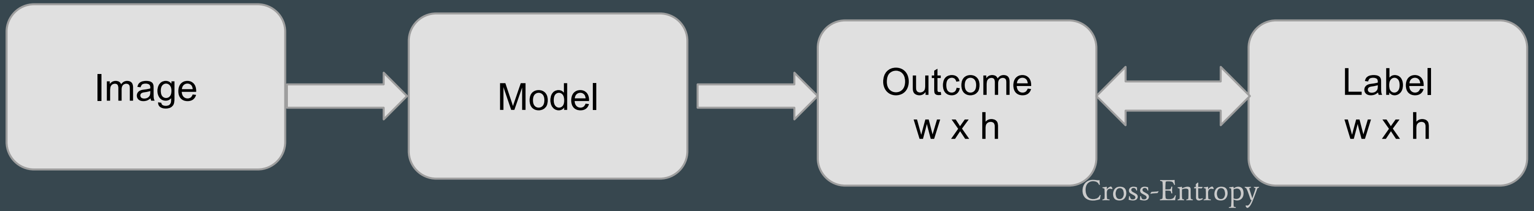
Semantic Segmentation

An example of semantic segmentation, where the goal is to predict class labels for each pixel in the image.

Input Segmentation

Image Source: http://pages.cs.wisc.edu/~jiaxu/projects/weak-label-seg/

The Typical Way



Loss is calculated for each pixel independently.

Issue

How to create a dense prediction?

Related works:

- Patchwise training
- Small model -> small receptive field
- Post-processing (eg. superpixel projection, random field regularization, filtering.....)
- Saturating tanh nonlinearities
- Restricted receptive field
- Input shifting and output interlacing
- Multi-scale pyramid processing

Idea

Semantics and Location

Global information resolves **what** while **local** information resolves **where**.

- Global information → What (Semantics)
- Local information → Where (location)

- User train entire image, instead of patch.
- Let Receptive field overlap significantly to improve efficiency.
- Transfer learning from classification net to fully convolutional network.
- For pixel-wise prediction, connect coarse outputs to pixels.

Fully Convolutional Network

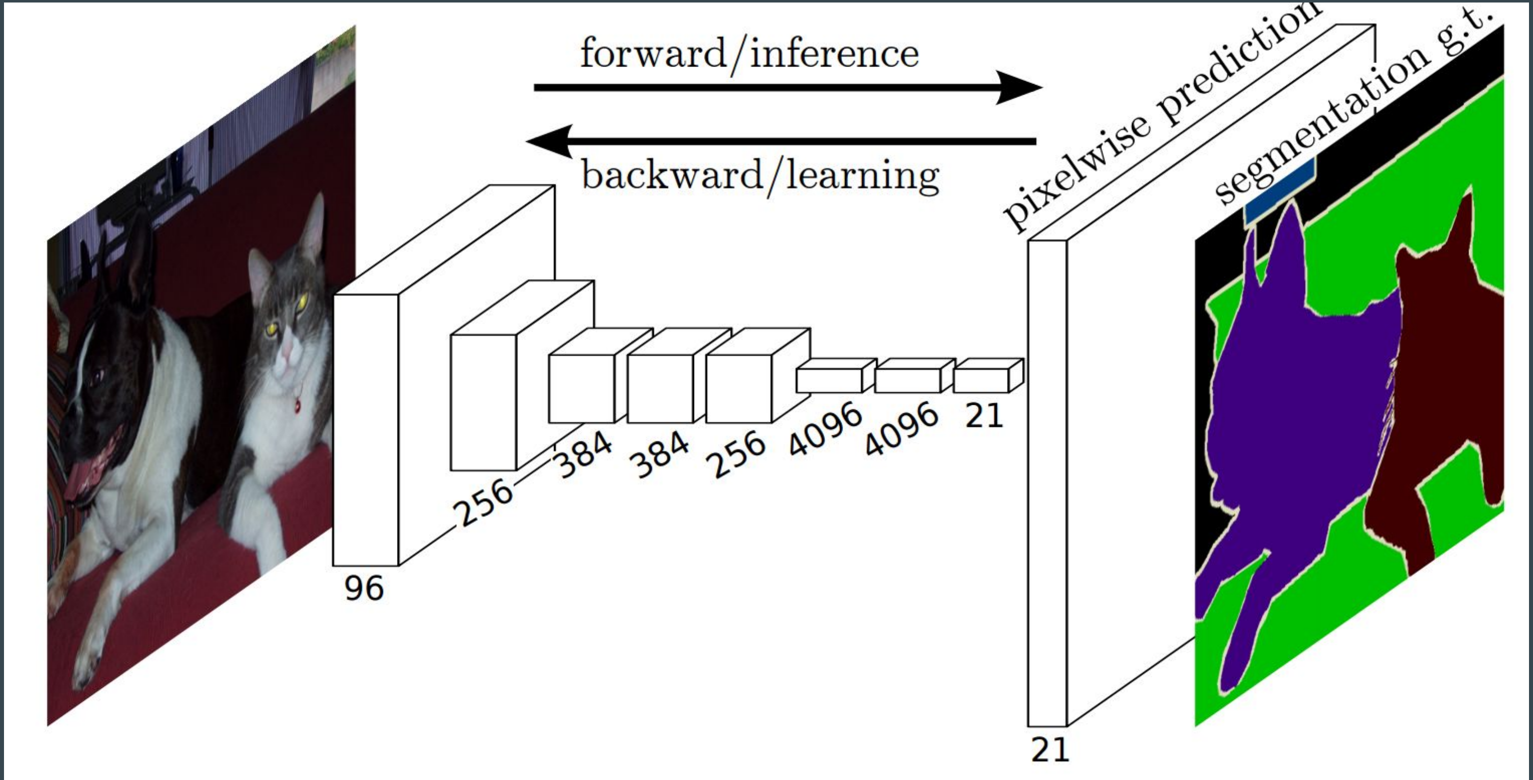
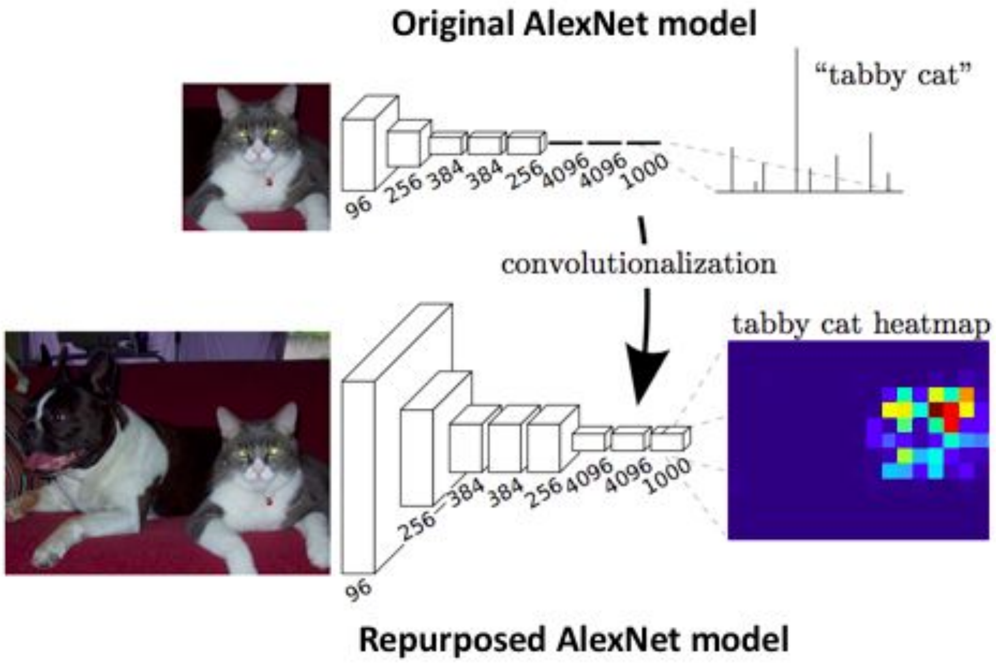


Image Source: http://deeplearning.net/tutorial/_images/cat_segmentation.png

Convert classification net to Fully Convolutional Network



The encoder produces a ***coarse*** feature map which is then refined by the decoder module.

Image Source: https://www.jeremyjordan.me/semantic-segmentation/

Dense Prediction

Strategy for upsampling:

- Shift-and-Stitch
- Deconvolution
- Bilinear

Deconvolution

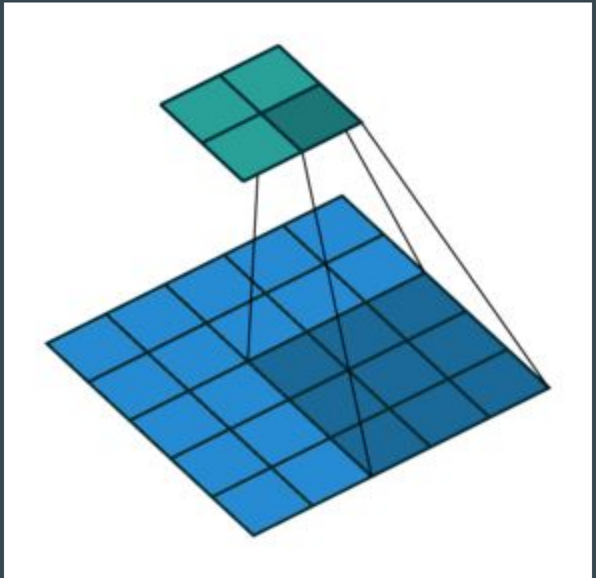
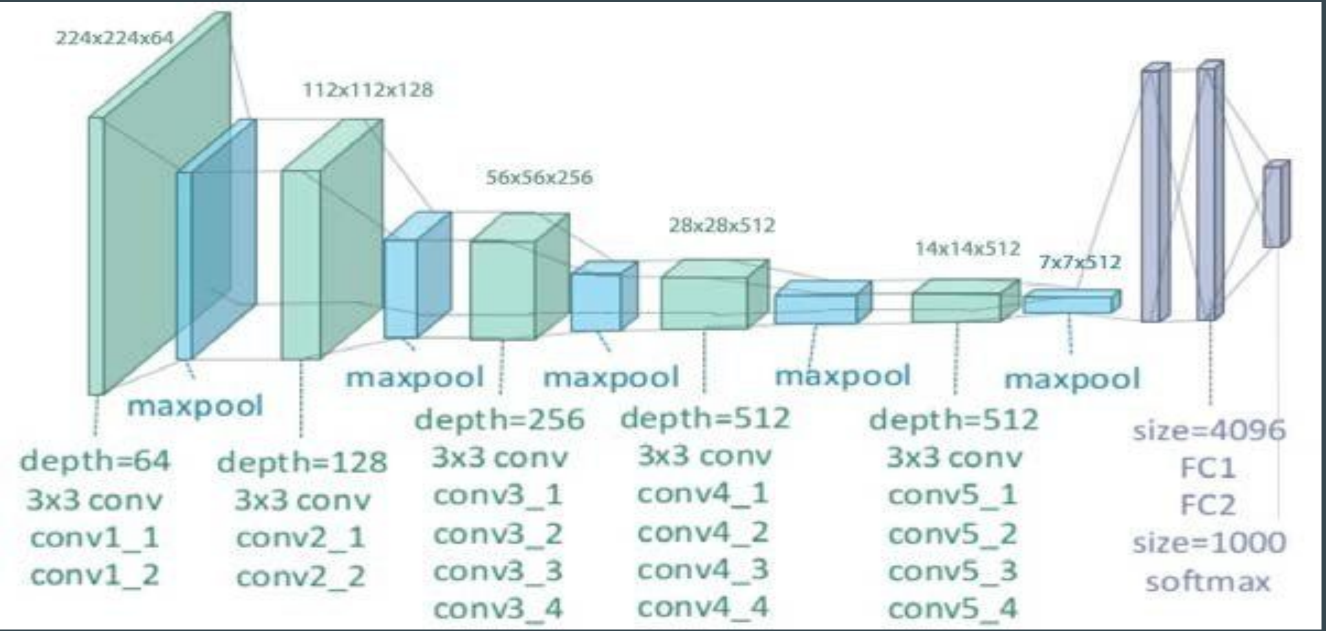


Image Source: https://cdn-images-1.medium.com/max/600/1*BMngs93_rm2_BpJFH2mS0Q.gif

VGG-16 Architecture



https://www.researchgate.net/figure/llustration-of-the-network-architecture-of-VGG-19-model-conv-means-convolution-FC-means_fig2_325137356

FCN Architecture

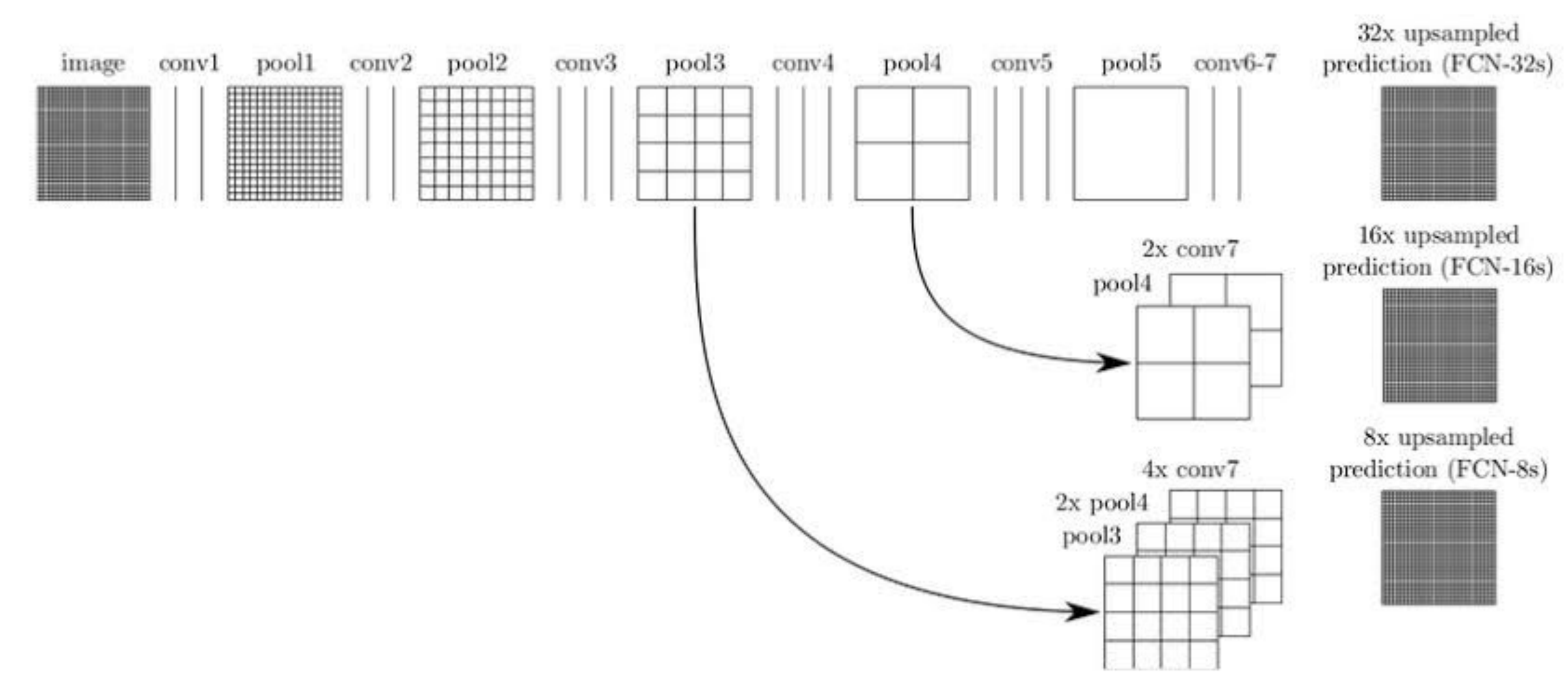
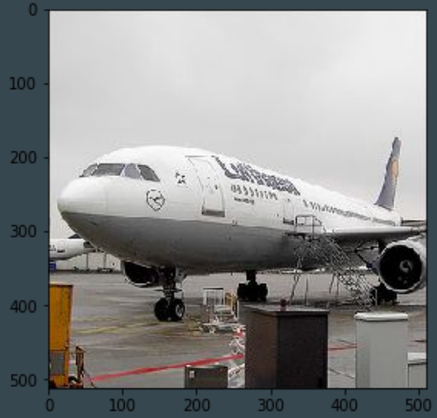
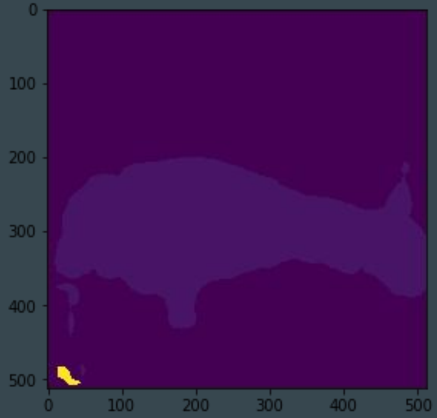
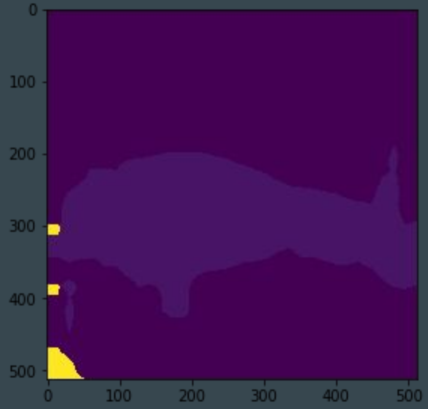
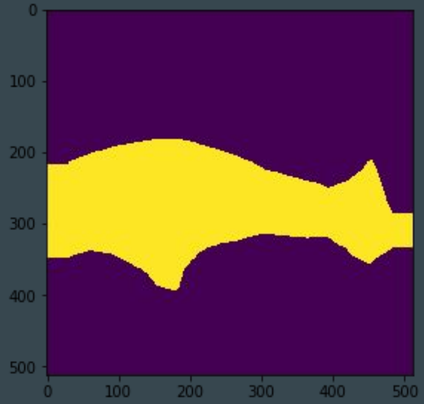


Image FCN-8 FCN-16 FCN-32

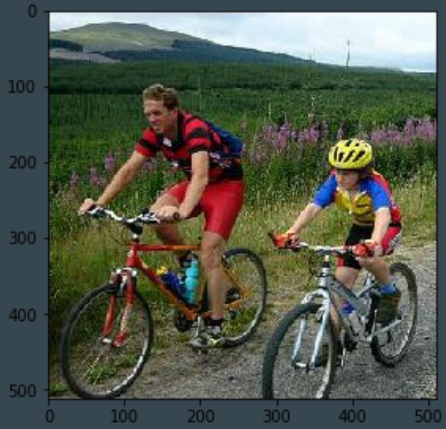




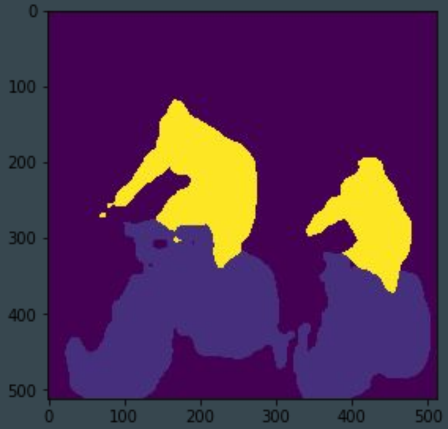




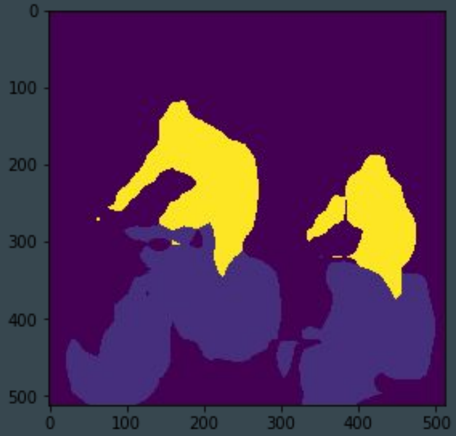
Image



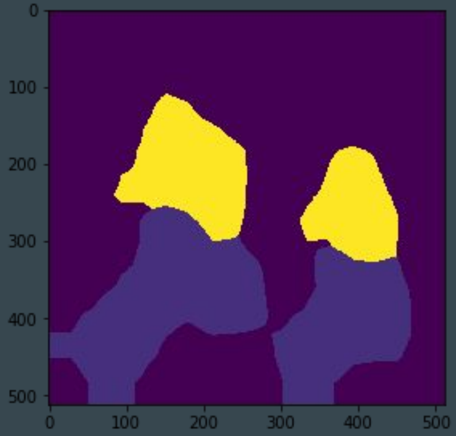
FCN-8



FCN-16



FCN-32



Results

	FCN-32s		FCN-16s		FCN-8s	
	Paper	Ours	Paper	Ours	Paper	Ours
Pixel acc.	89.1	79.2	90.0	77.1	90.3	77.3
Mean acc.	73.3	48.6	75.7	46.0	75.9	45.5
Mean IU	59.4	40.6	62.4	38.2	62.7	37.3
F.w. IU	81.4	70.8	83.0	67.6	83.2	67.1

We have made use of:

- We have used pre-trained weights of **VGG-16 to train FCNs using Adam-optimizer.**
- Pascal VOC 2012 dataset has been used.
- Upsampling via Deconvolution.