Fully Convolutional Network With Edge Labels For Semantic Segmentation

Ruyi Zhang
Yifan Wang
Yi Zheng

USC Viterbi School of Engineering
University of Southern California

Content

- Background
- Methods
- Results
- Conclusion

1. Background

- Semantic Segmentation
- Fully Convolutional Network
- Atrous Convolution
- PASCAL VOC

1.1 Semantic Segmentation

- Classification vs. detection vs. segmentation:
  - Classification: classify an image into a label.
  - detection: classify objects in an image and bound them by bounding boxes .
  - segmentation: classify and label the pixels in an image.
- All we have done in class is Classification.

Classification

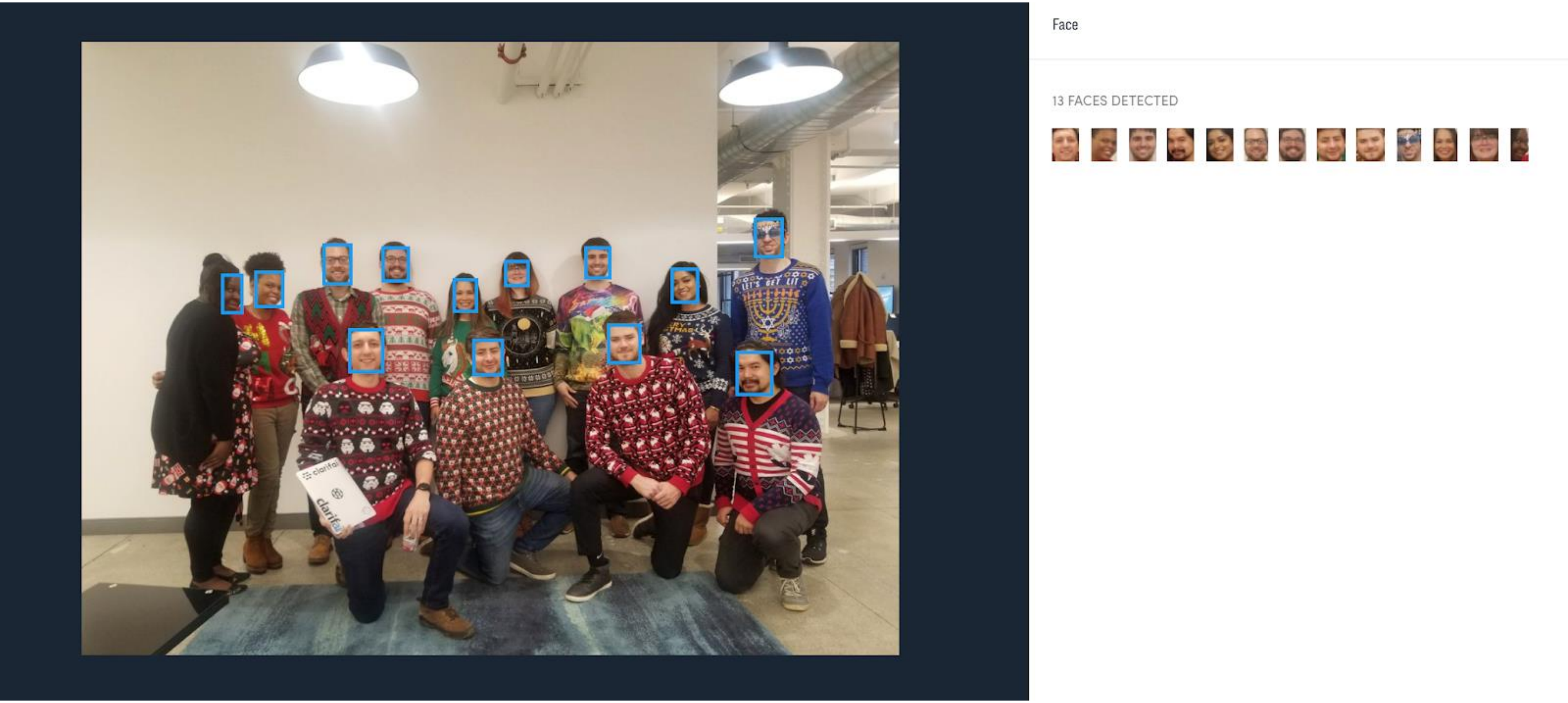
General

LANGUAGE

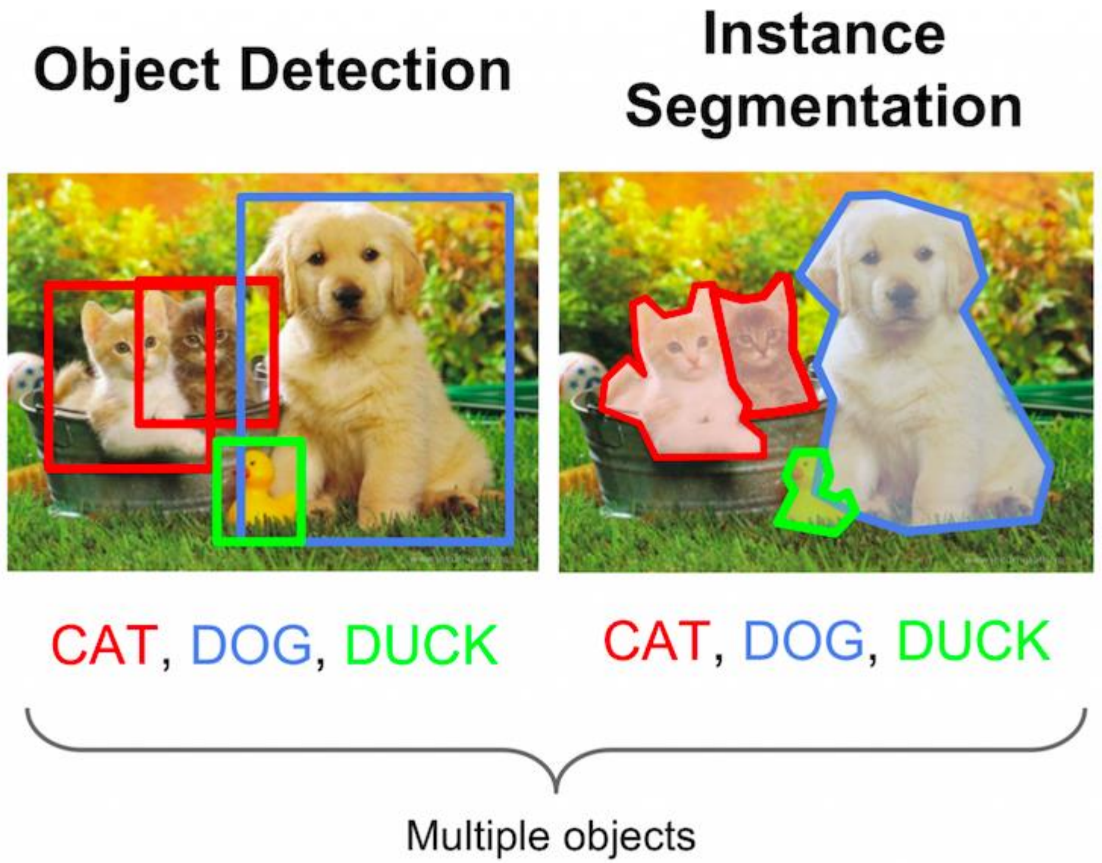
English (en)

PREDICTED CONCEPT	PROBABILITY
group	0.988
people	0.982
woman	0.973
festival	0.963
portrait	0.953
adult	0.943
child	0.942
singer	0.935
election	0.935
music	0.934

Detection



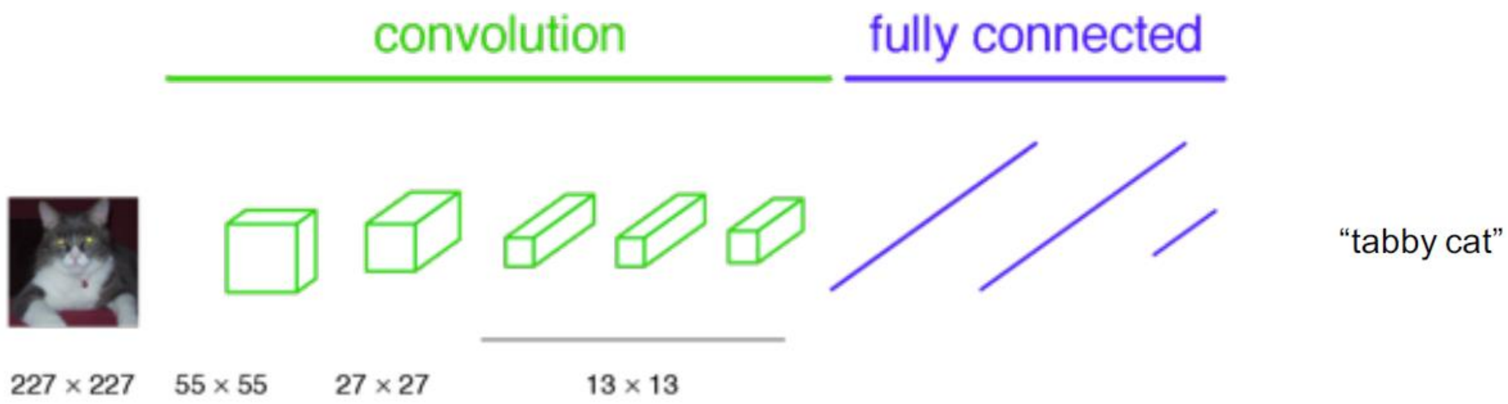
Segmentation



1.2 Fully Convolutional Network

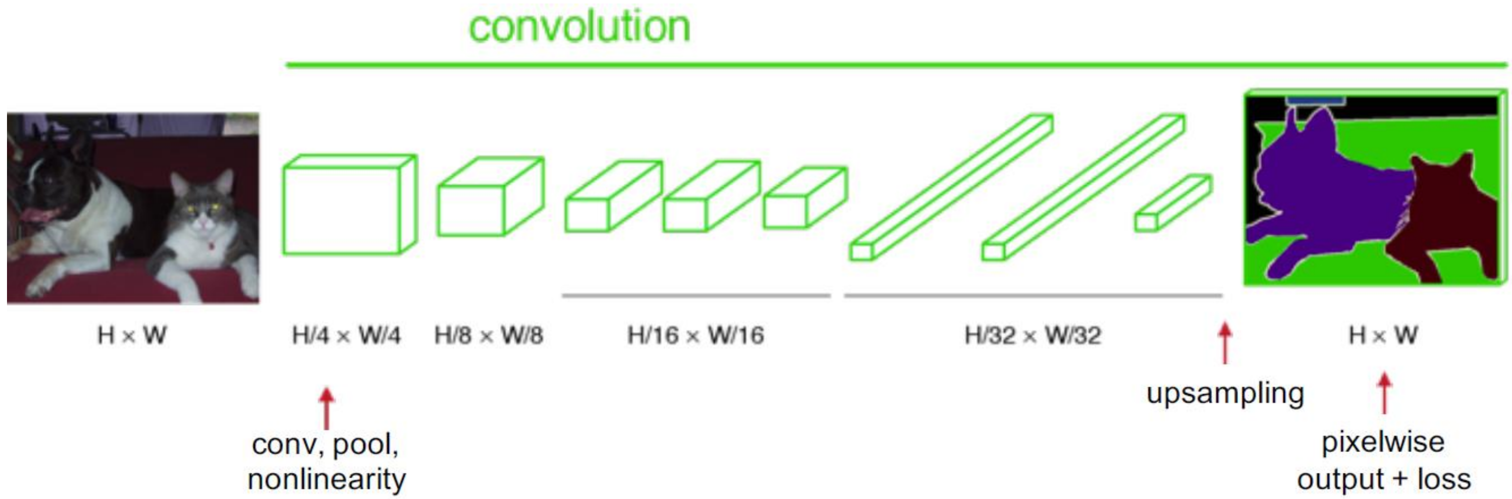
CNN(what)	FCN(what and where)
Down sampling convolution + fully connected + output	Down sampling conv + 1*1 conv with 21 channels(classes) + up sampling conv
down sampling: capture semantic information	up sampling: recover spatial information
Input: fixed dimensions	Input: any size
Output: one predicted label	Output: pixelwise prediction
throw away spatial coordinates	make spatial output maps

CNN



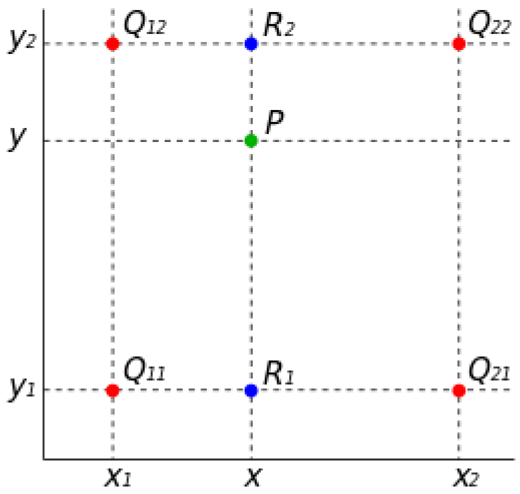
convolution
fully connected
"tabby cat"
227 × 227
55 × 55
27 × 27
13 × 13

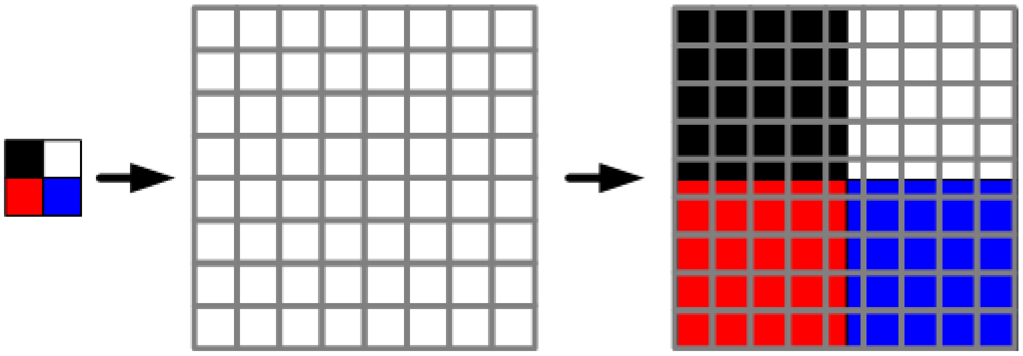
FCN


convolution
H × W
H/4 × W/4
H/8 × W/8
H/16 × W/16
H/32 × W/32
H × W
upsampling
conv, pool,
nonlinearity
pixelwise
output + loss

Up Sampling

Bilinear interpolation:

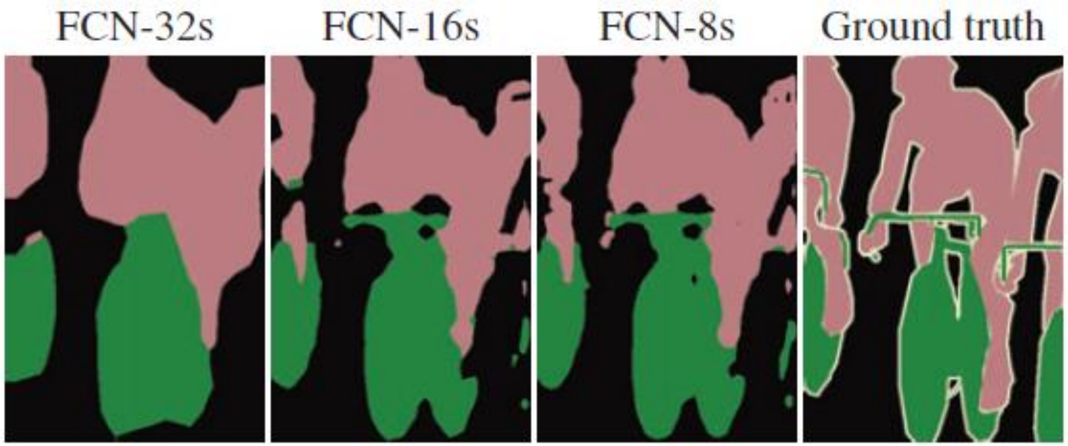




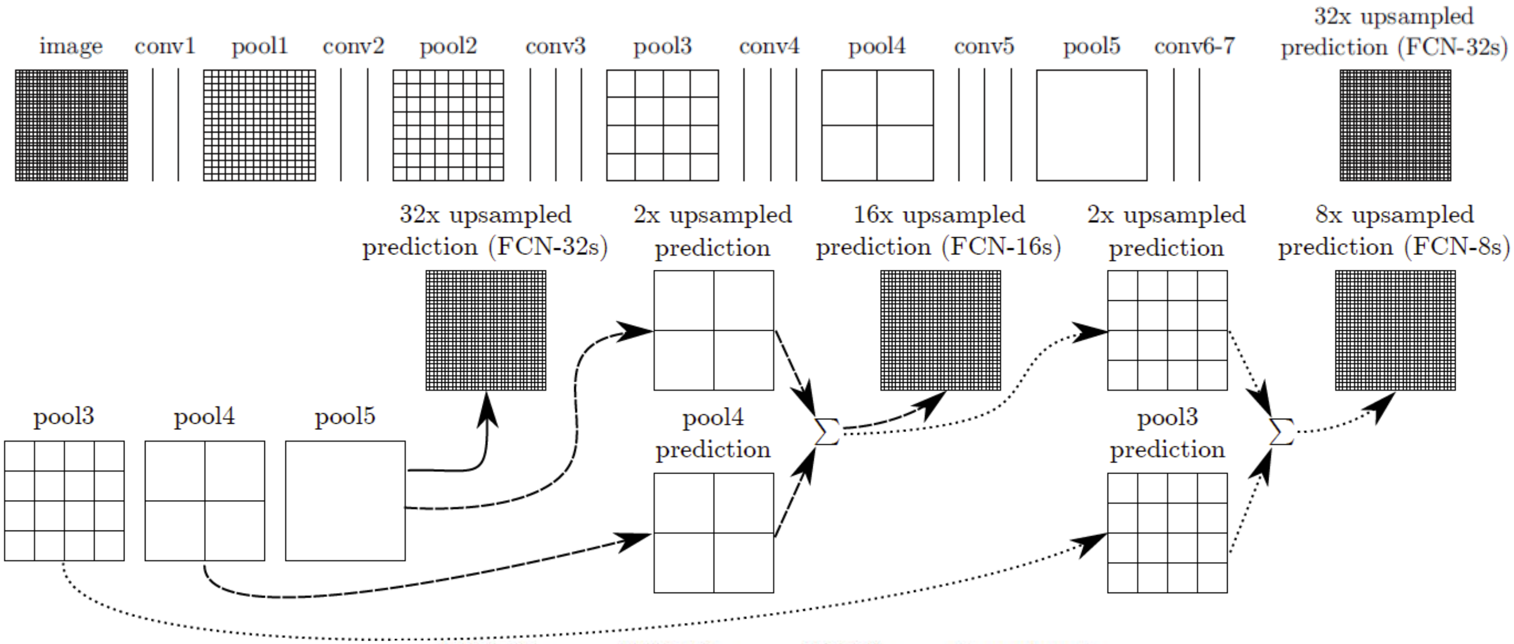
Deconv layer need not to be fixed, but can be learned.

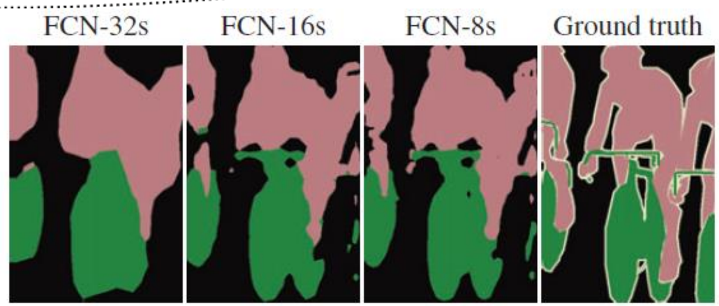
1.2 Fully Convolutional Network

- Deep features can be obtained when going deeper
- Spatial location information is also lost when going deeper
- Output is dissatisfyingly coarse, because stride limits the detail.
- Add skips to fuse layer outputs(by element-wise addition).



1.2 Fully Convolutional Network





1.3 Atrous Convolution

- In previous FCN, to have a good feature map, the output feature map is very small.
- Downsampling is a loss compression.
- 32× upsampling is aggressive.
- Use Atrous Convolution.

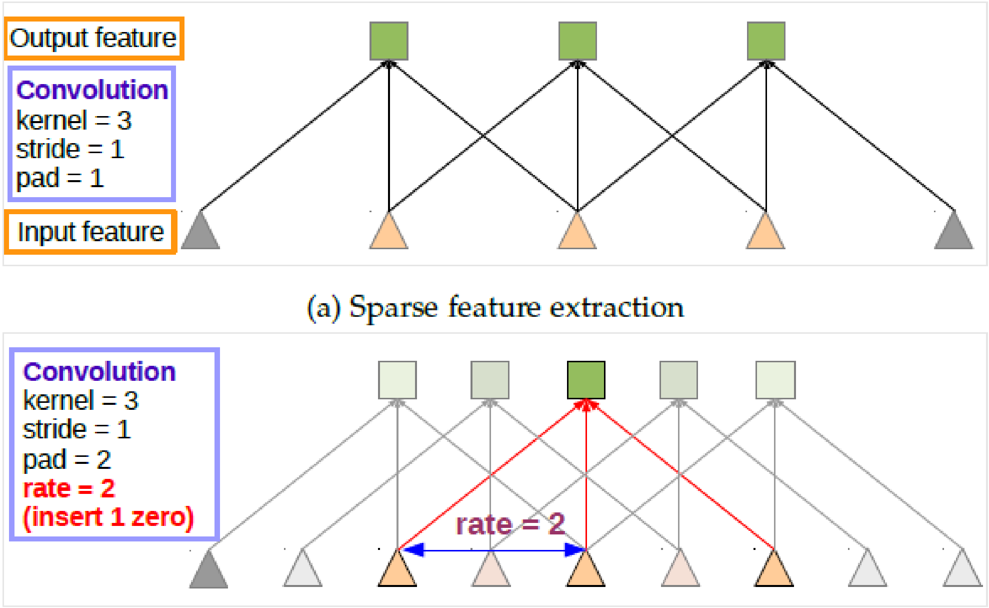
1.3 Atrous Convolution

$$y[i] = \sum_{k=1}^{K} x[i + r \cdot k] w[k]$$

- When r=1, standard convolution.
- When r>1, atrous convolution which is the stride to sample the input sample during convolution.
- By adjusting r, we can adaptively modify filter's field-of-view.

1.3 Atrous Convolution

1D Atrous Convolution

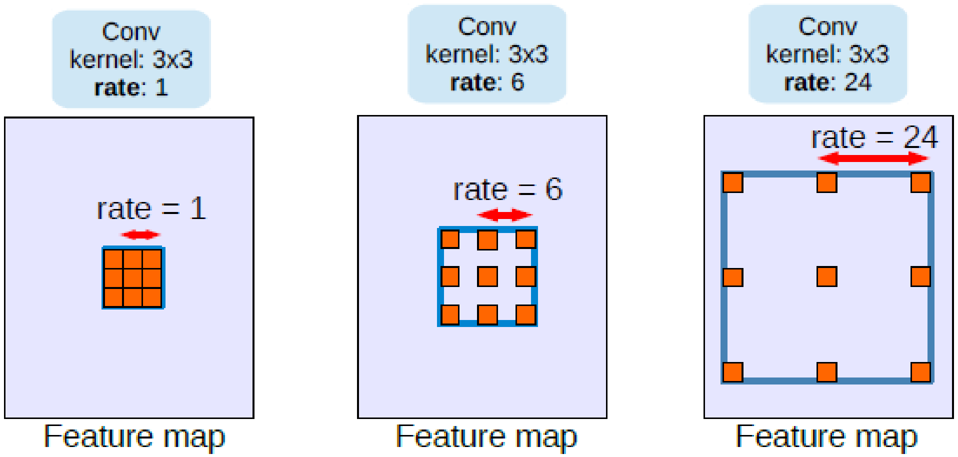


(a) Sparse feature extraction

(b) Dense feature extraction

1.3 Atrous Convolution

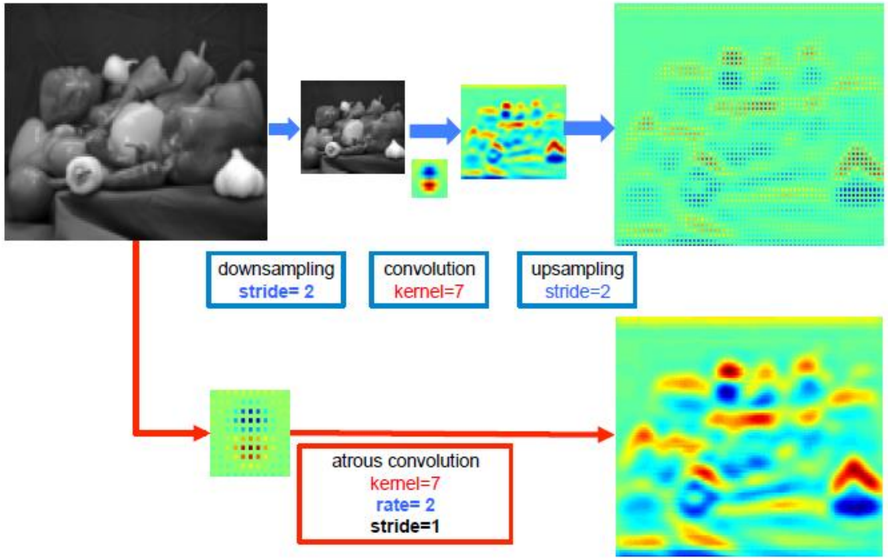
2D Atrous Convolution



- Increase the effective filter size.
- only consider non-zero filter values.
- number of filter parameters and operations per position stay constant.

- Makes the output feature map larger.
- Enlarge the field-of-view of filters to incorporate larger context.

1.3 Atrous Convolution

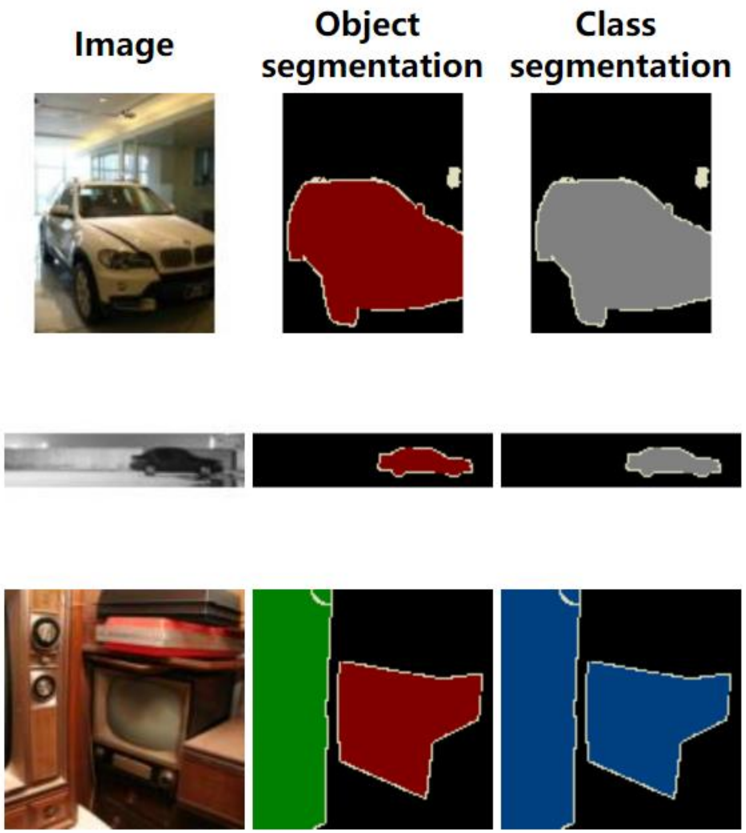


- Atrous Convolution can control the field-of-view
- Trade-off between accurate localization (small field-of-view) and context assimilation (large field-of-view).
- Easily and explicitly control the spatial resolution of responses.

1.3 PASCAL VOC

- recognize objects from a number of visual object classes in realistic scenes.
- a supervised learning learning problem.
- Segmentation training examples:
  - the training image.
  - the object segmentation: pixel indices correspond to the first, second object etc.
  - the class segmentation: pixel indices correspond to classes.
- For both segmentation image, index 0 corresponds to background and index 255 corresponds to 'void' or unlabelled.

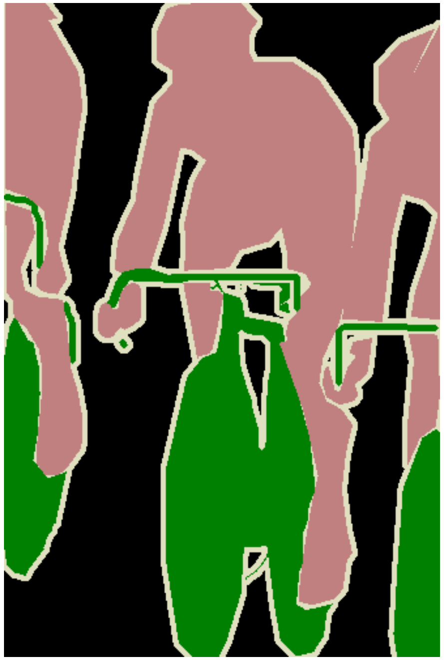
1.3 PASCAL VOC



2. Methods:

Motivation (1):

- original FCN do not use edges when computing loss;
- prediction on edges would be bad;



Ground truth

Prediction

Methods (1):

- Adding a new edge loss.
- A hard decision, loss for each pixel is either 0 or 1;

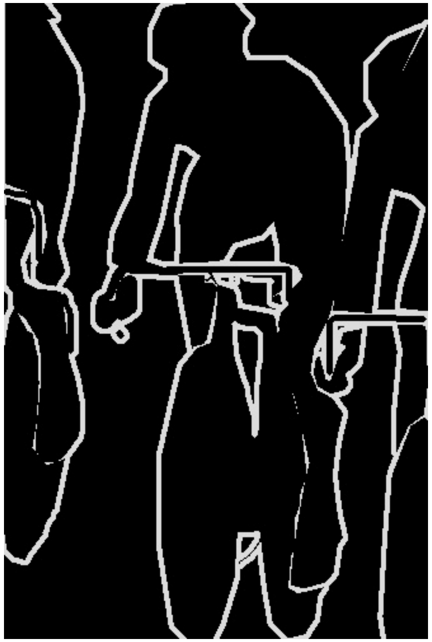
$$L_{edge} = \frac{(1 - E_{I_{true}}) \odot E_{I_{pred}}}{\sum(1 - E_{I_{true}}) + \epsilon}$$

- An opposition process operated on true edge labels
- Achieve the negative effect of loss;

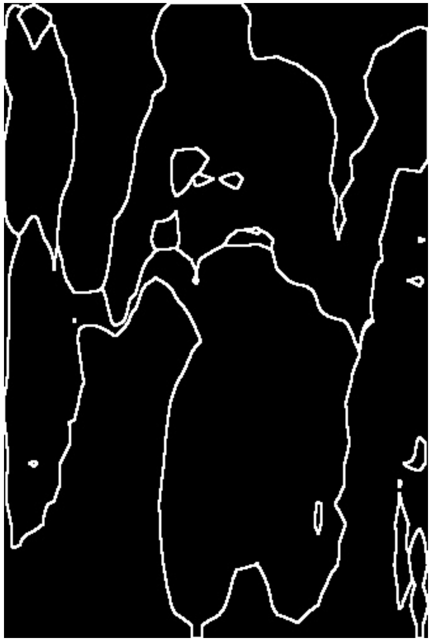
Methods (2):

loss		ground truth	
		is edge	none edge
prediction	is edge	0	1
	none edge	1	0

Methods (3):



True label



Predicted edge

Pixel with none zero loss

3. Teat Results:

	Without edge loss	With edge loss
meanIOU	0.5613	0.5760
pixel acc	0.8911	0.8970

Table 1: Accuracy when train both network for 30 epochs

Segmentation results on Pascal VOC dataset :

(1) Original

(2) Ground truth

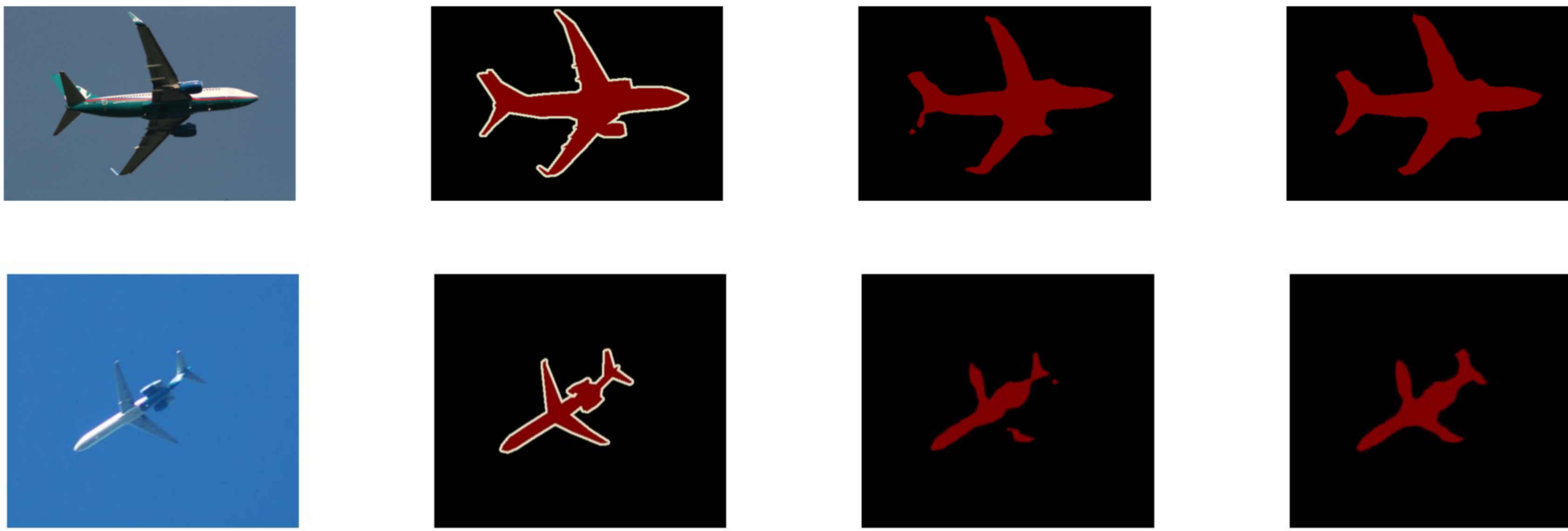
(3) Without edge loss

(4) With edge loss

Common factors of successful results:

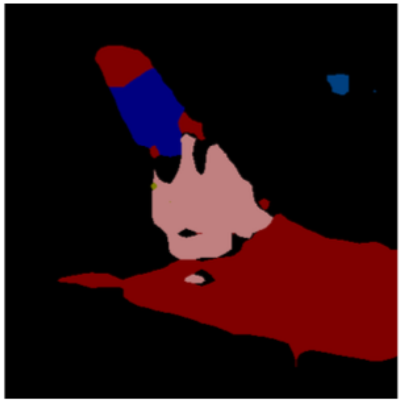
- relative simple boundaries
- no complex texture on objects

Original | Ground truth | Without edge loss | With edge loss



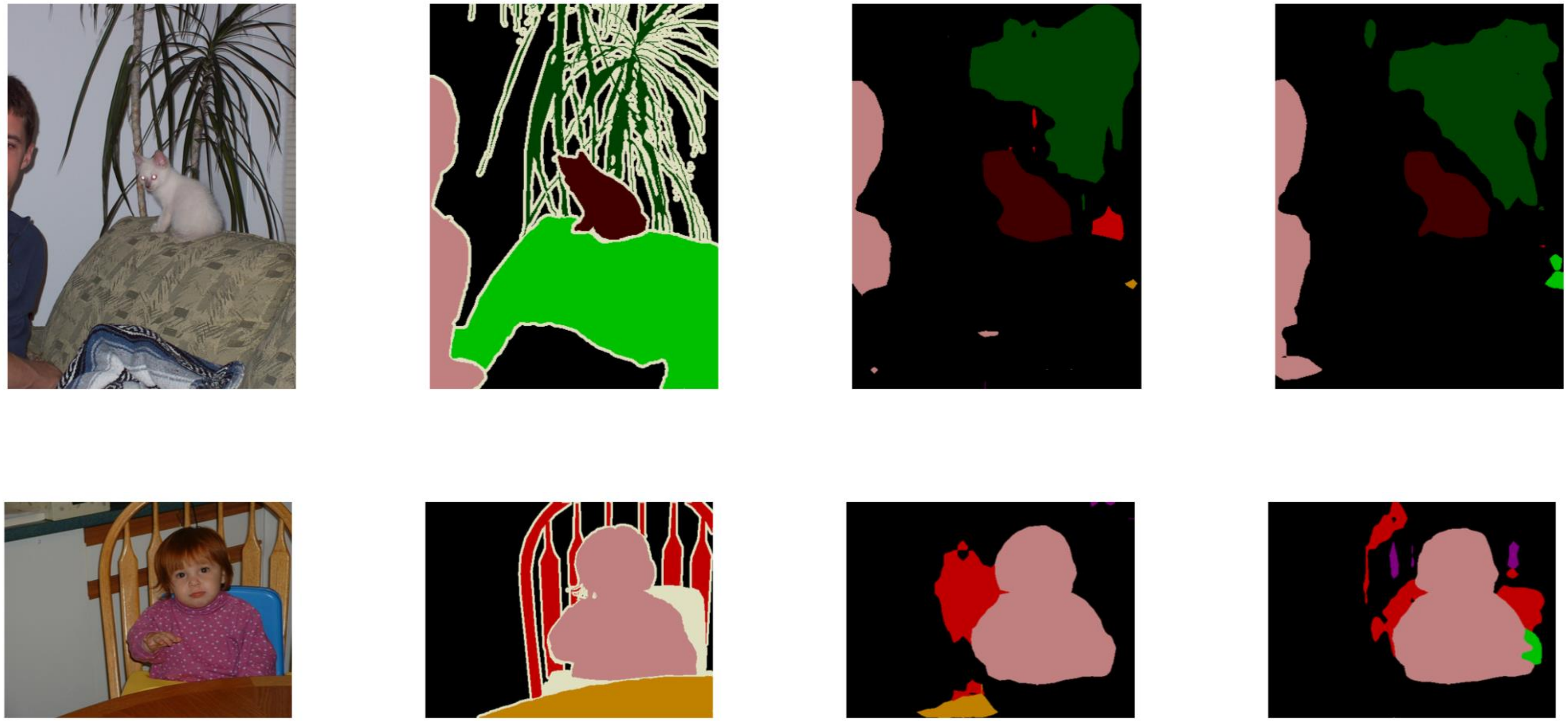
Case 1: Object with complex texture on image

Without edge loss



With edge loss

Case 2: Objects with lots of boundaries



Without edge loss With edge loss

Case 3: Texture is almost the same as the background

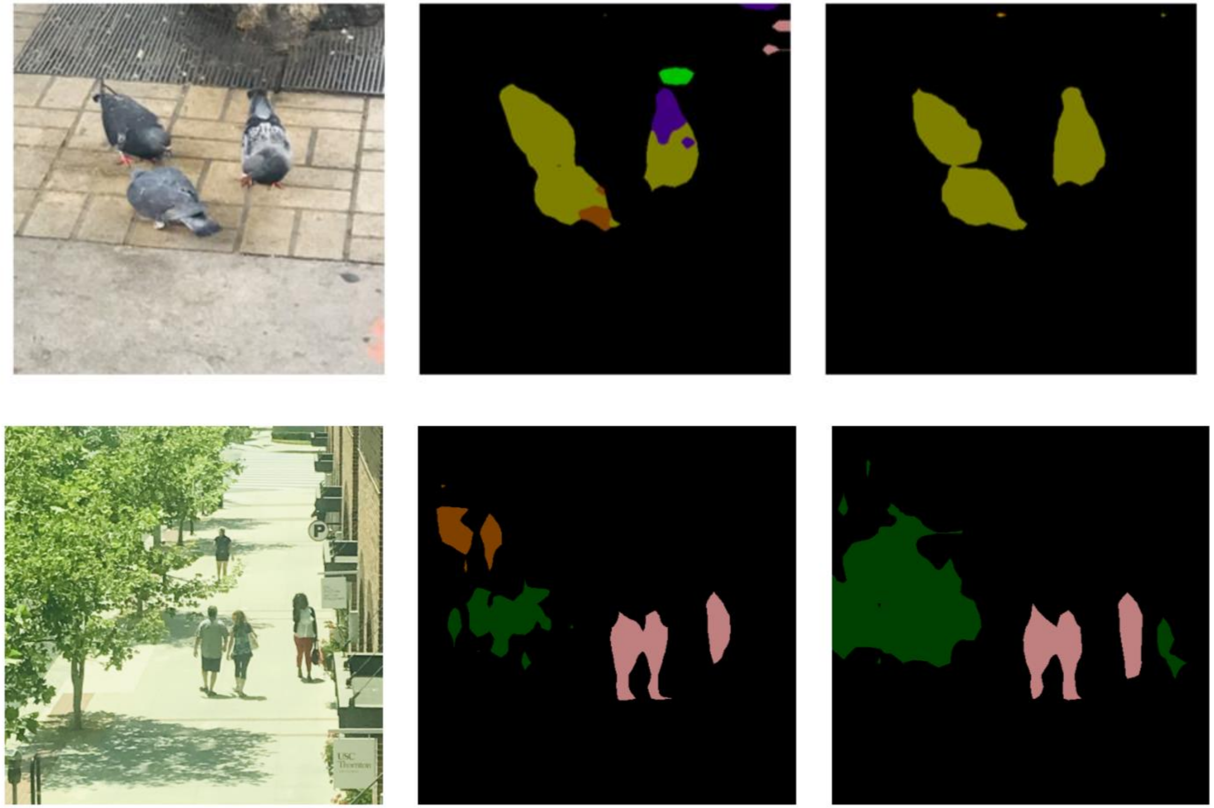
Without edge loss

With edge loss

Images we taken in our daily life :

Good

Bad

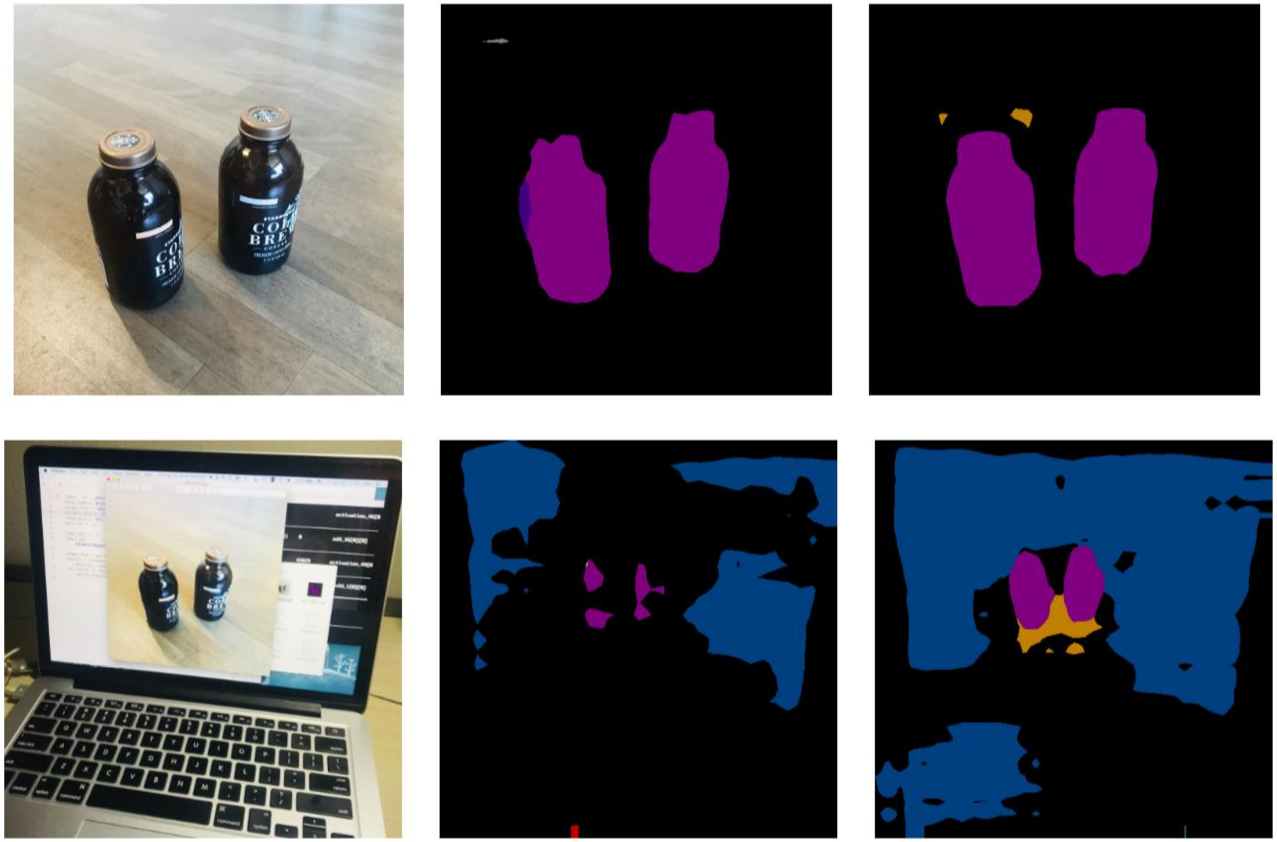


Without edge loss

With edge loss

Fancy cases :

- **An object has texture of another object**



Without edge loss　　With edge loss

4. Conclusion:

1. **Our Project:**

- Based on FCN, adding edge loss and atrous convolution method would improve segmentation results.

2. **Future Work:**

- More training epochs
- Soft decision
- Develop new methods to deal with more complex edges
- Develop more advanced network for fancy cases.