

---

# Computer Vision

## Lecture 12 Case Studies (Deep Convolutional Models)

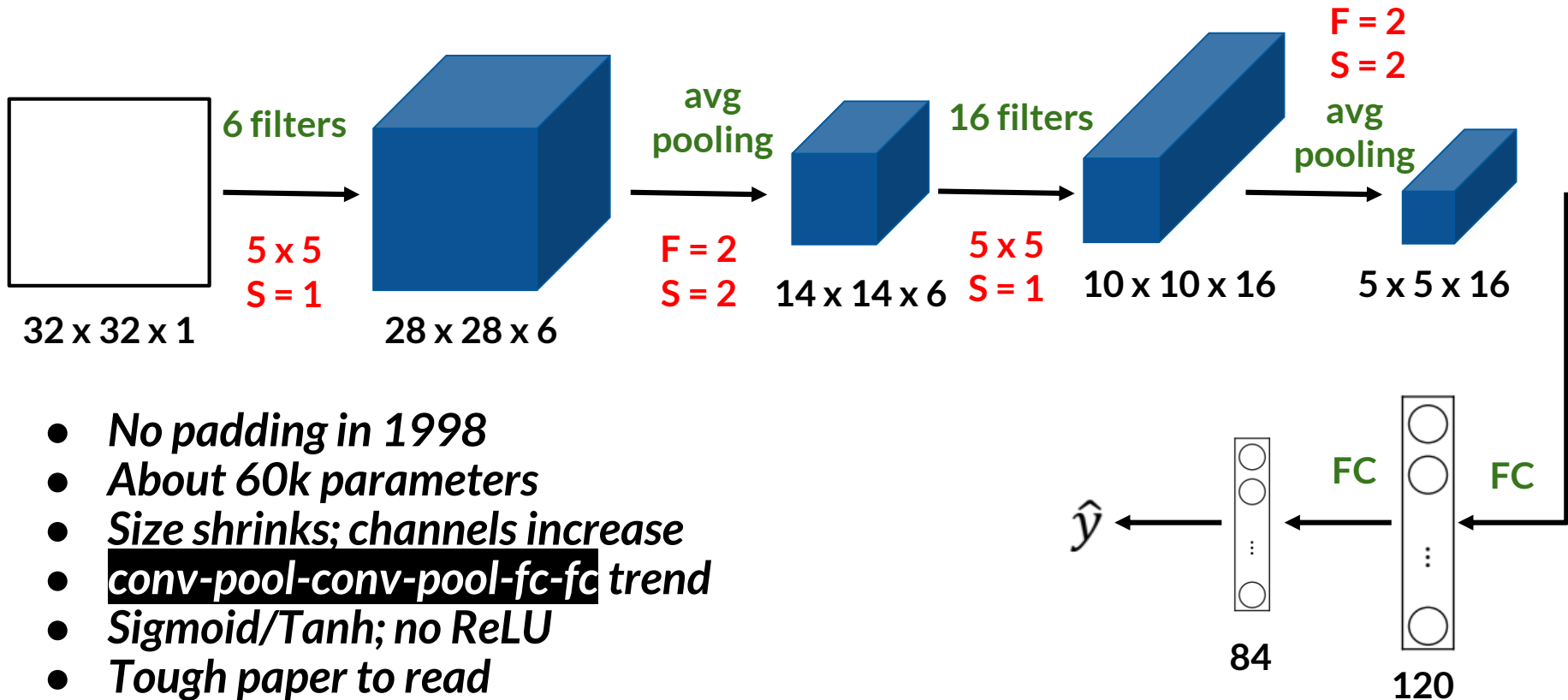
*Asim D. Bakhshi, PhD*  
*Military College of Signals*

---

# Object Classification

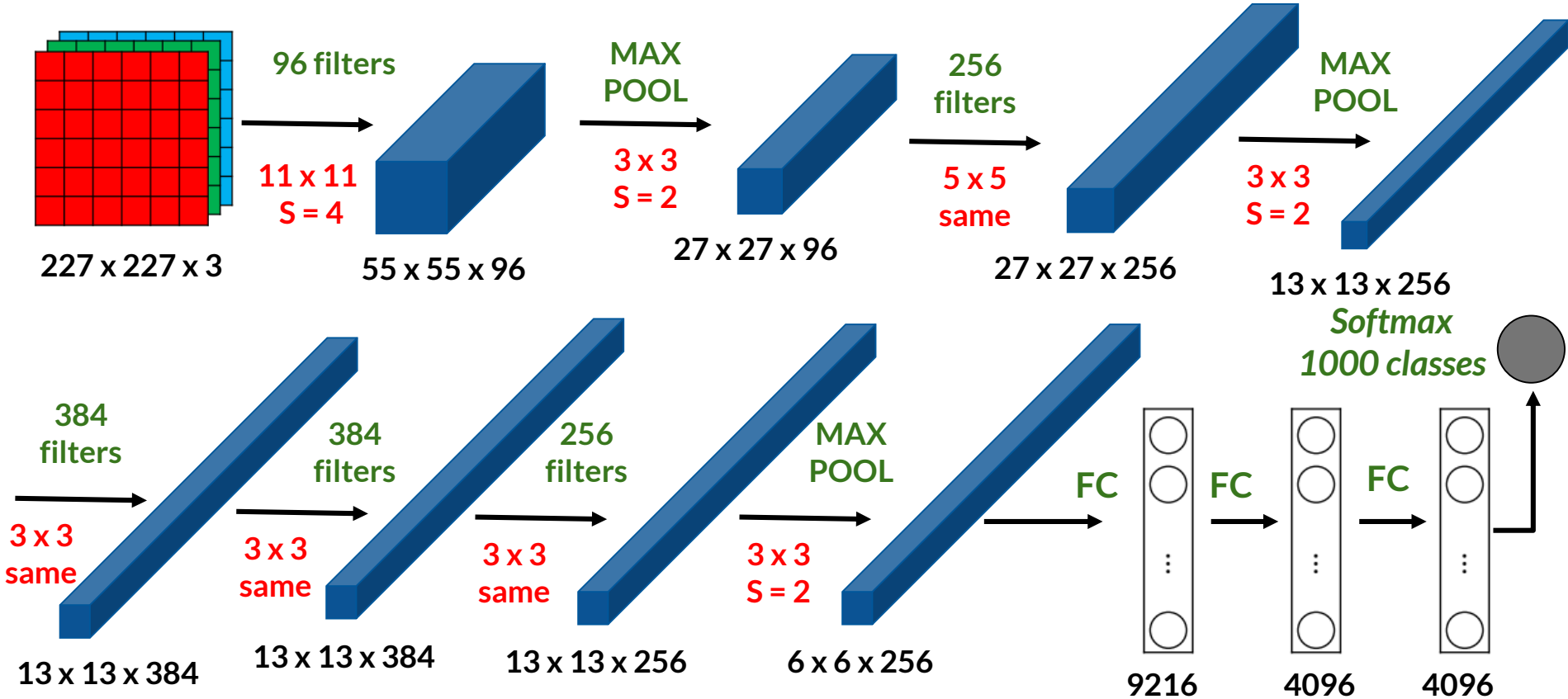
# LeNet-5

LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.



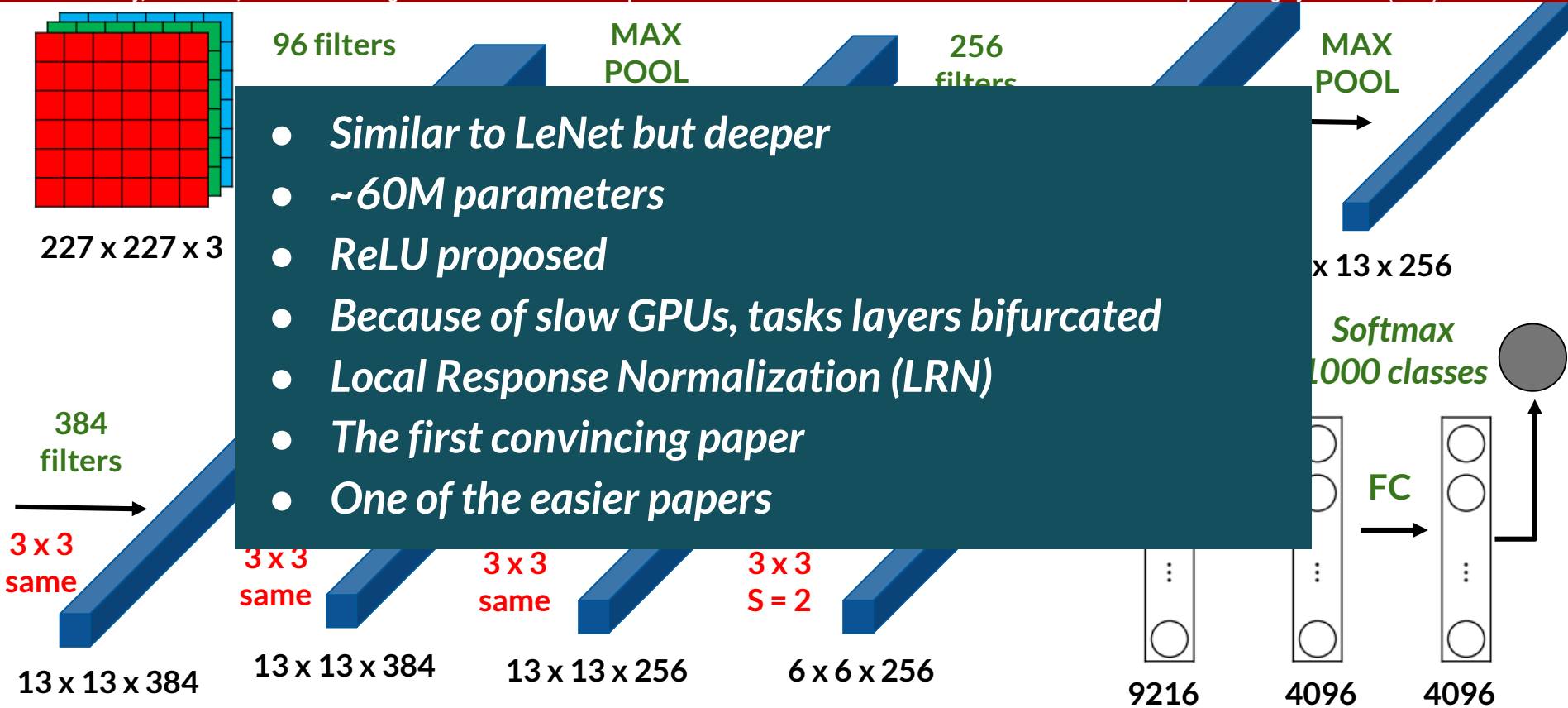
# AlexNet

Krizhevsky, Sutskever, and Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012): 1097-1105.



© 2020-2021 Asim D. Bakhshi

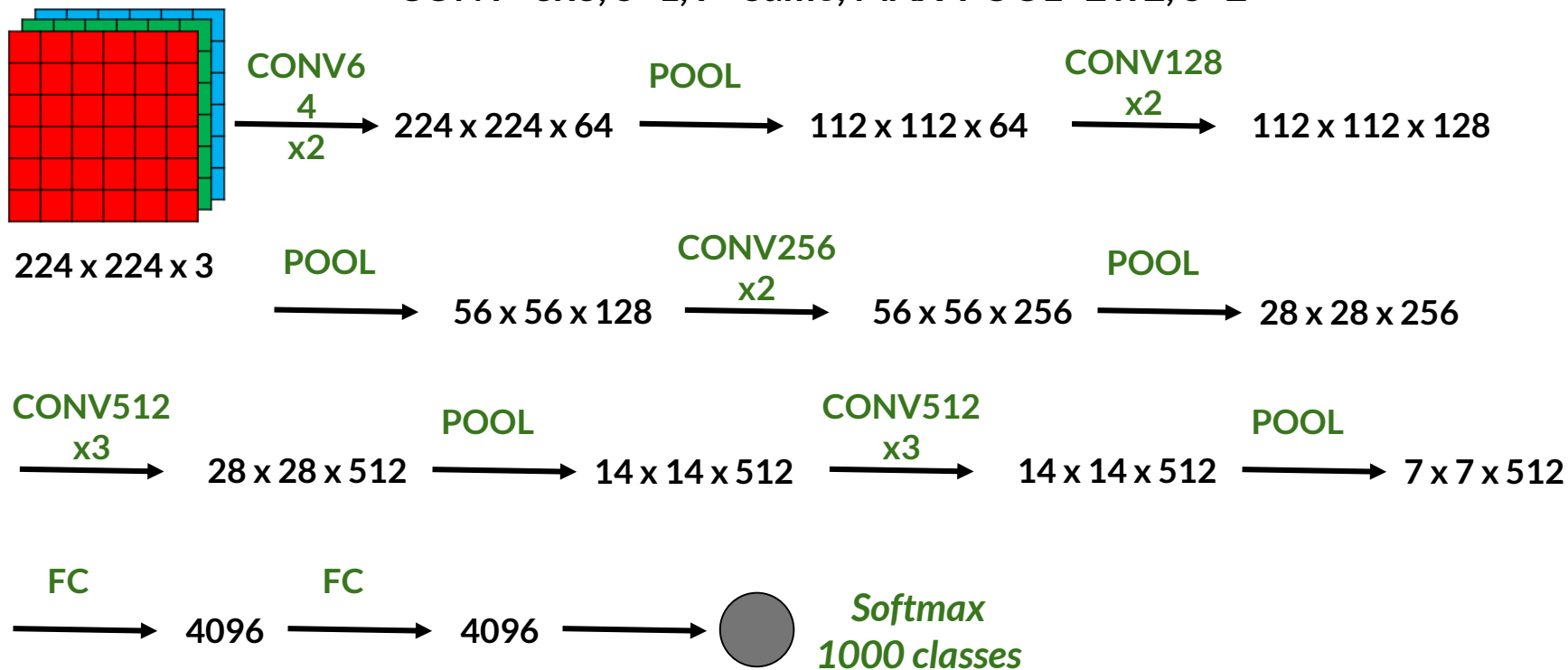
5



# VGG-16

Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

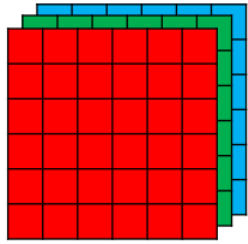
CONV=3x3, S=1, P=Same; MAX-POOL=2 x 2, S=2



# VGG-16

Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." *arXiv preprint arXiv:1409.1556* (2014).

CONV=3x3, S=1, P=Same; MAX-POOL=2 x 2, S=2



224 x 224 x 3

- *Simplicity of architecture and uniformity*
- *Very deep: ~138M parameters*
- *More versions: VGG-19*
- *Good exercise*
  - *Read the papers*
  - *Build these three networks from scratch*
  - *Reproduce results*

x 128

256

7 x 7 x 512

CONV512  
x3

2

FC

4096

FC

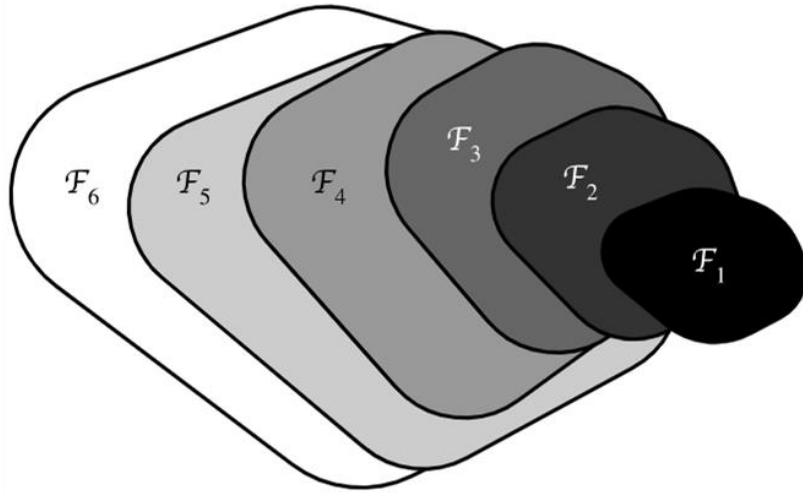
4096



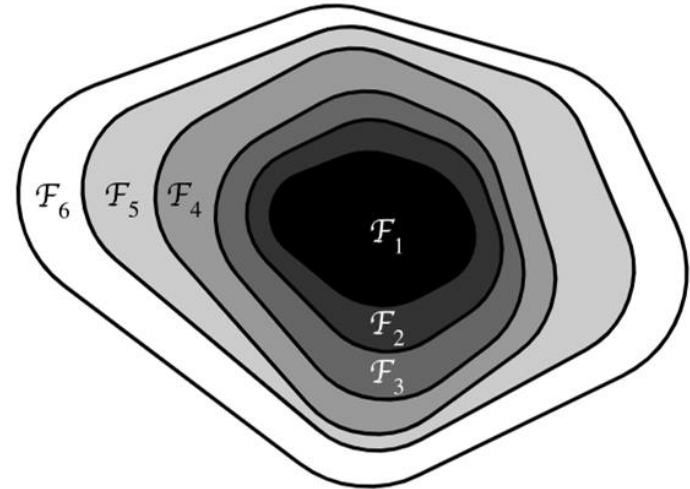
Softmax  
1000 classes

# ResNet

He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.



Non-nested function classes

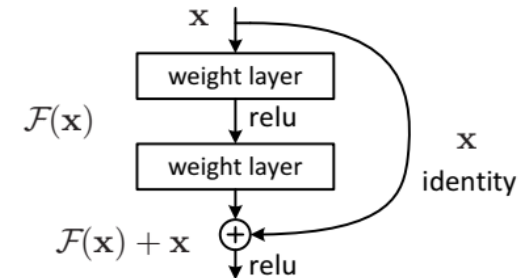
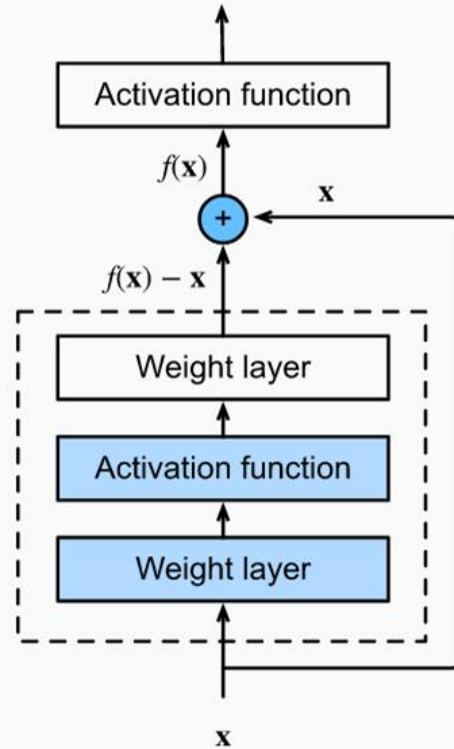
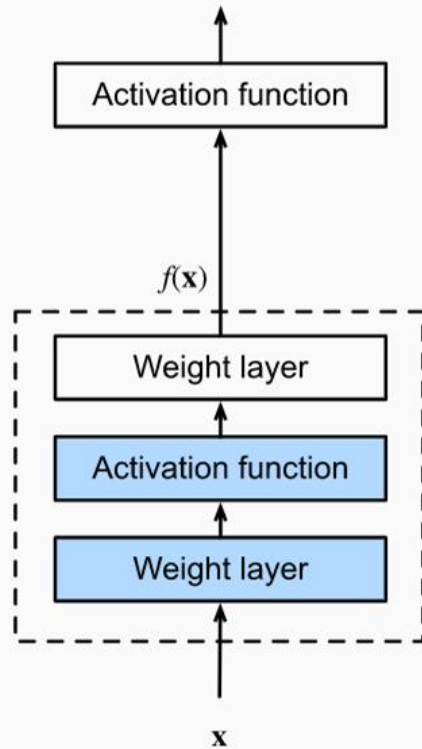


Nested function classes



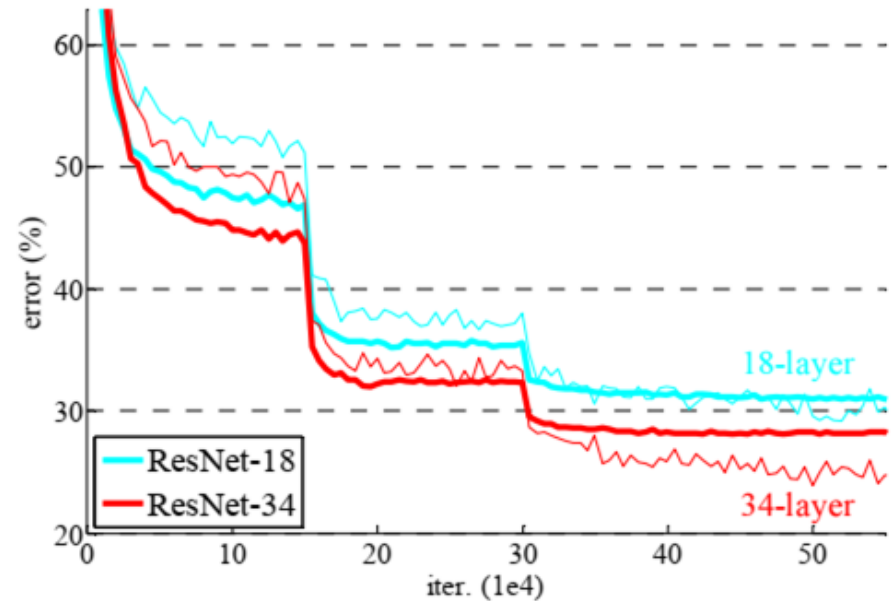
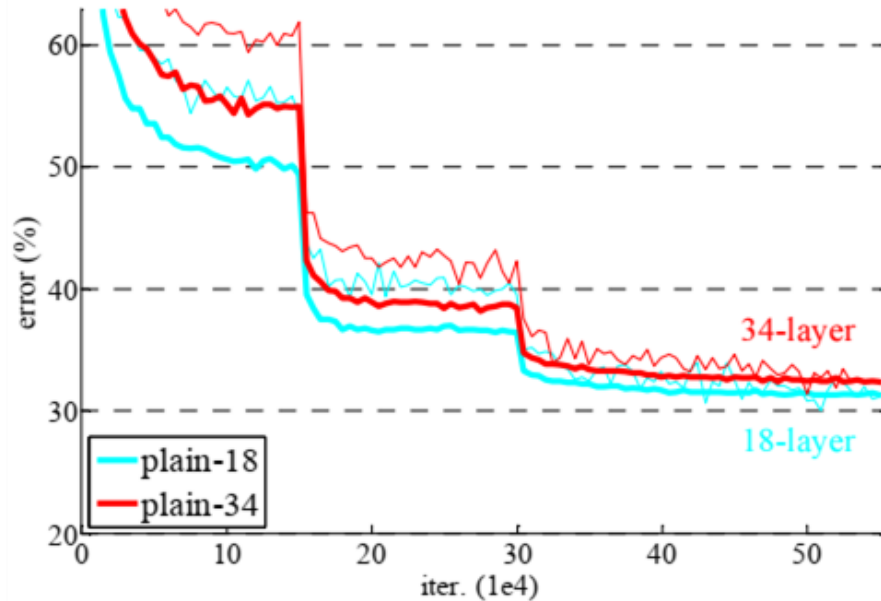
# ResNet

He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.



# ResNet

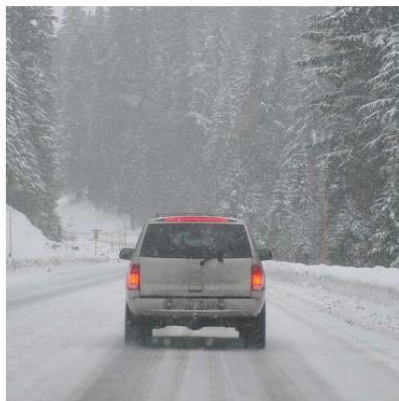
He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.



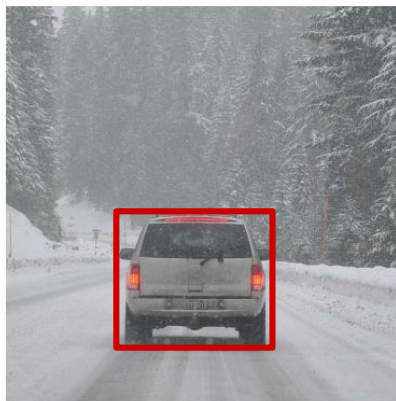
# Object Detection

# Localization and Detection

CLASSIFICATION



LOCALIZATION








DETECTION

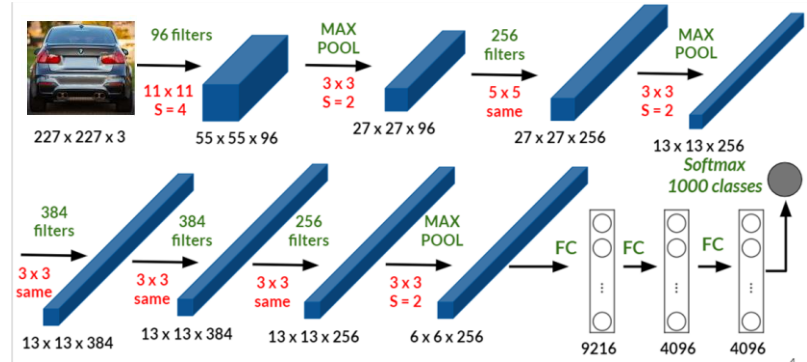


# Localization and Detection

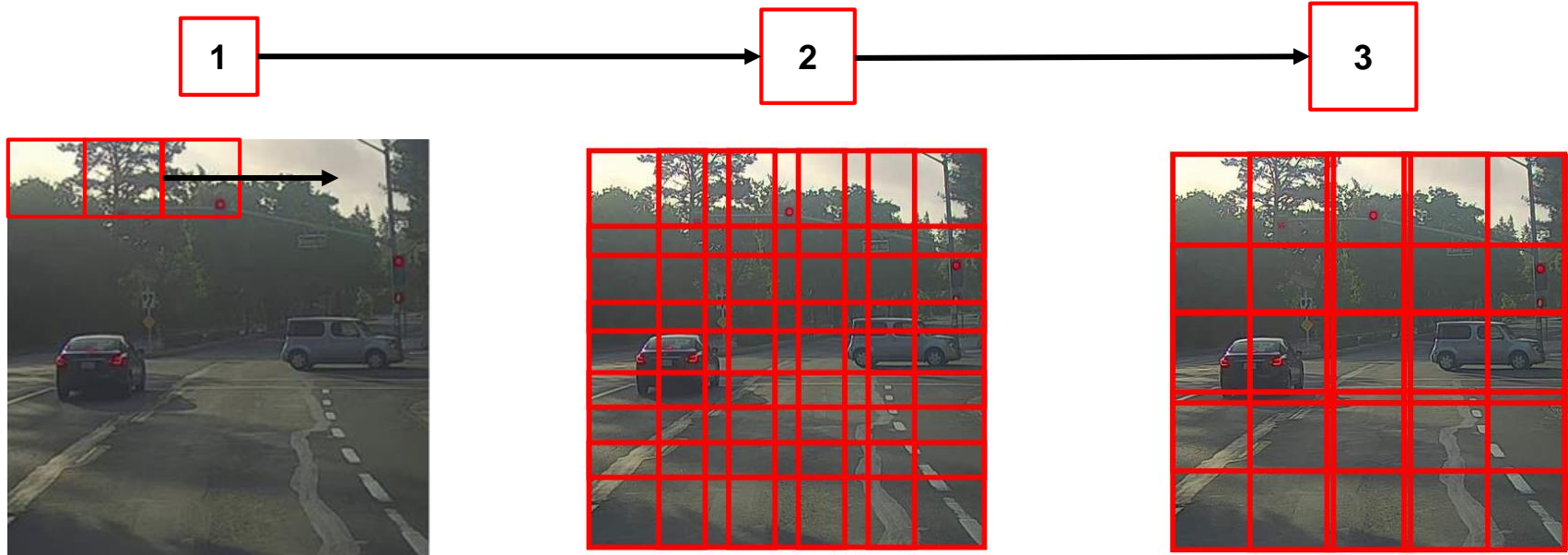


Training Set	
x	y
	1
	1
	1
	0
	0

## Train the ConvNet

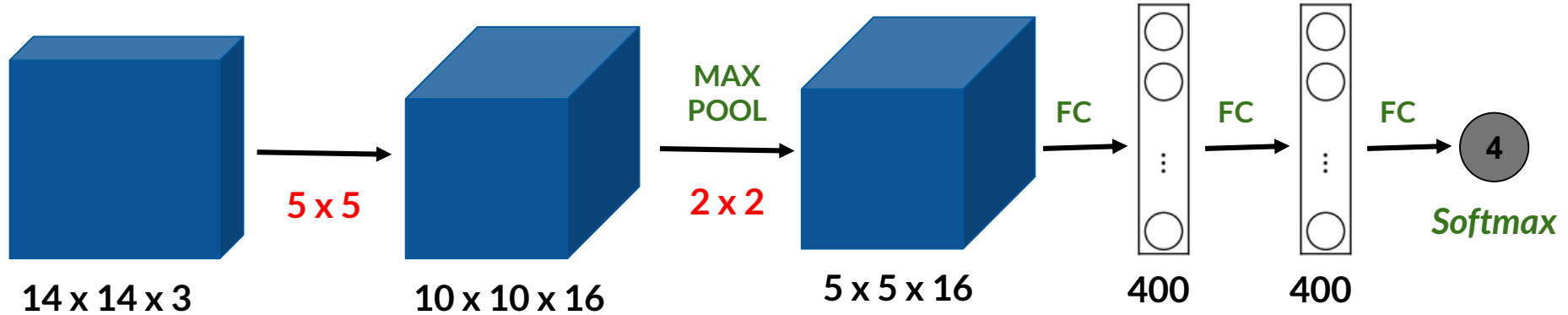


# Localization and Detection (Sliding Windows)

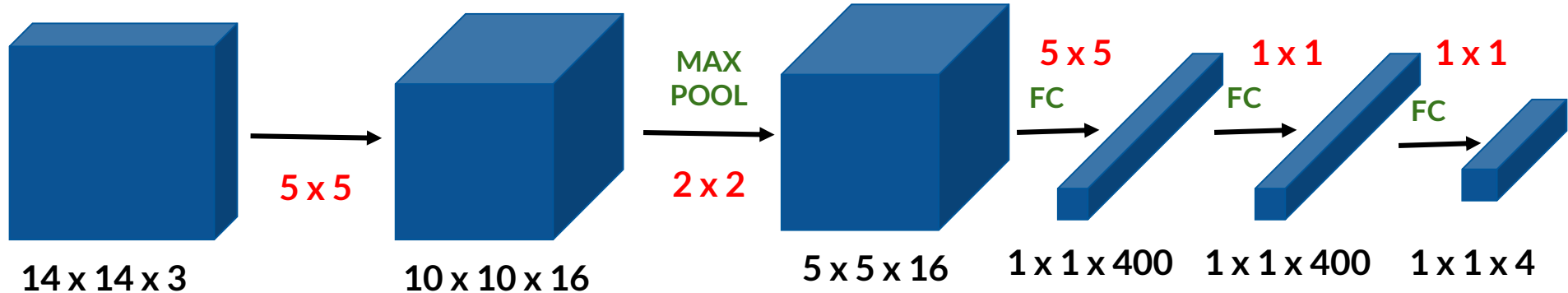


*Problem: Computational Cost*

# Localization and Detection (Convolutional)



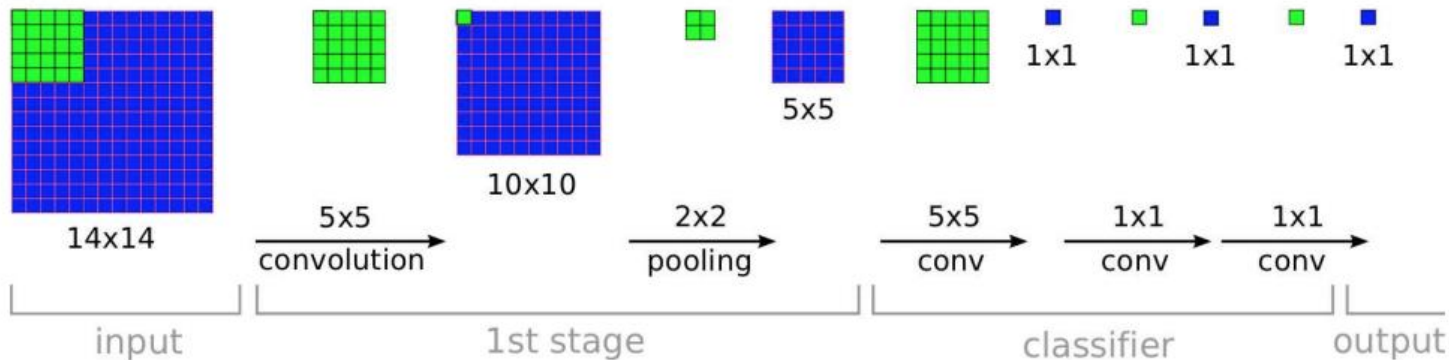
## Turning FC Layers into Convolutional Layers



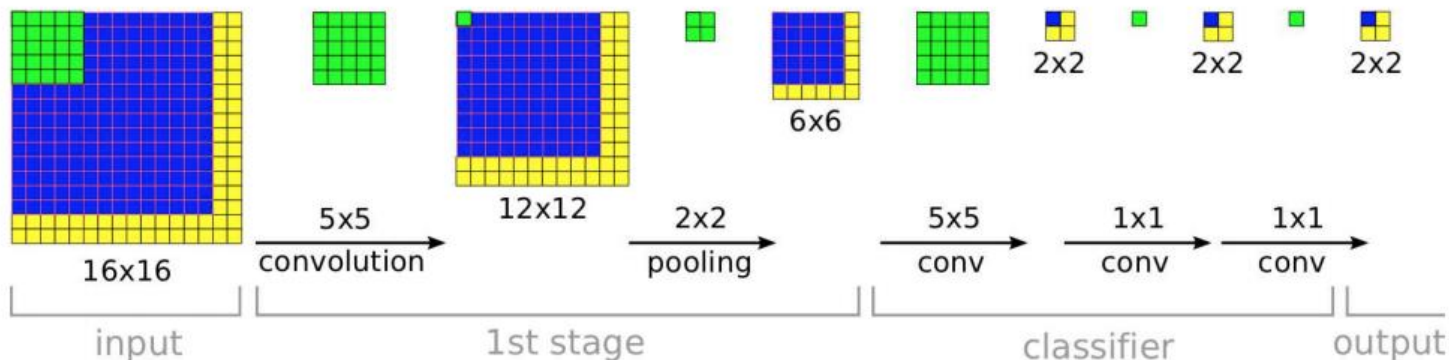
# Overfeat

Sermanet, Pierre, et al. "Overfeat: Integrated recognition, localization and detection using convolutional networks." *arXiv preprint arXiv:1312.6229* (2013).

TRAINED  
CONVNET



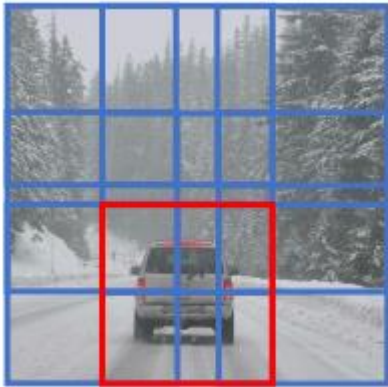
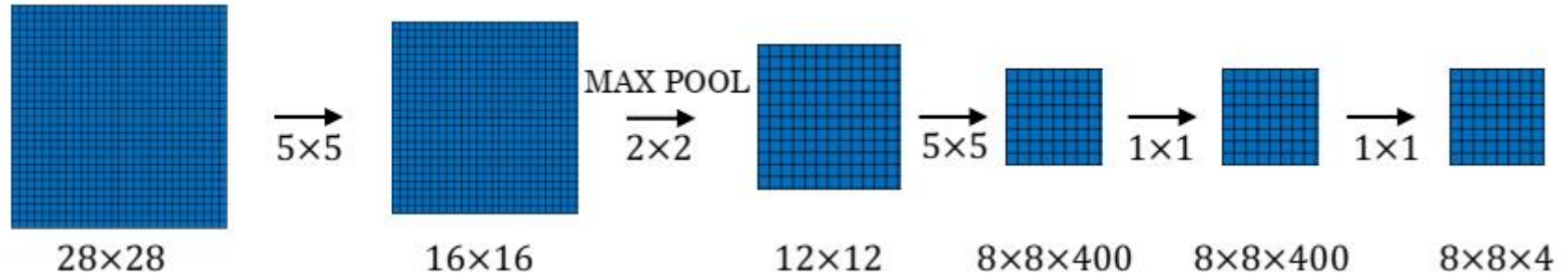
INPUT TEST  
DATASET





# Overfeat

Sermanet, Pierre, et al. "Overfeat: Integrated recognition, localization and detection using convolutional networks." *arXiv preprint arXiv:1312.6229* (2013).

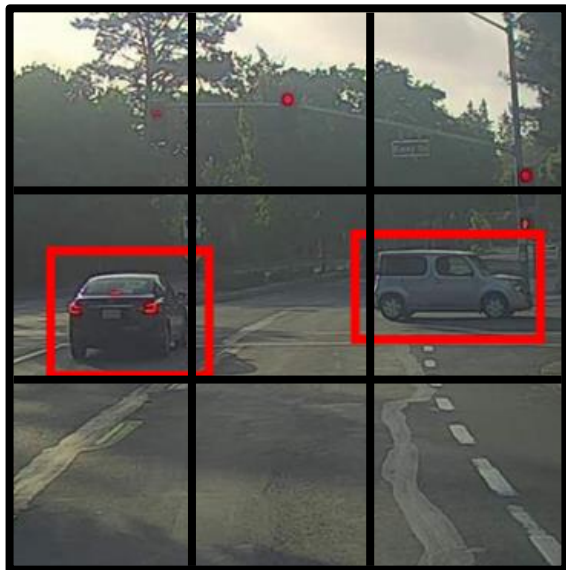


***Problem: Accuracy of Bounding Boxes***

# Improving Bounding Box Predictions - YOLO

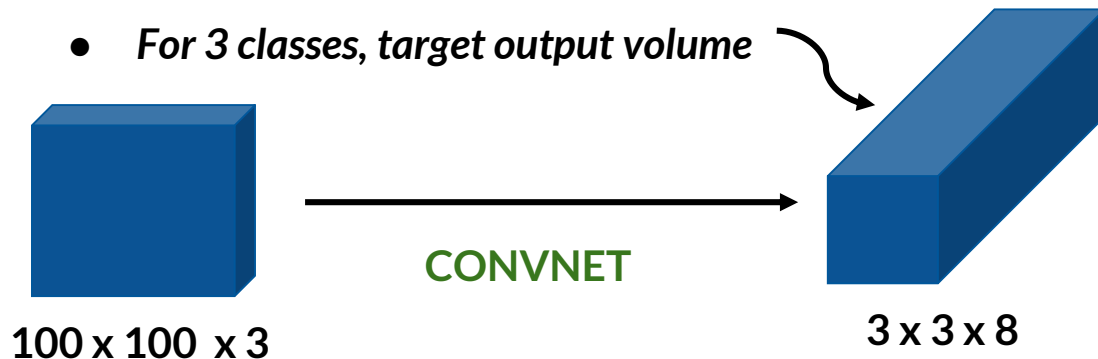
Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

100 x 100



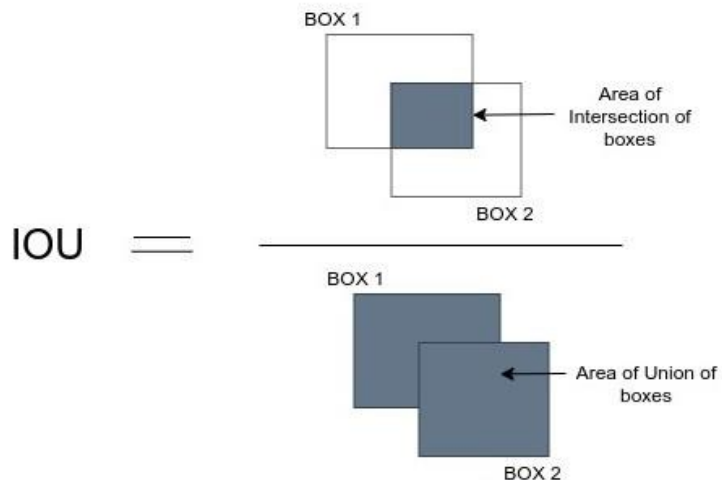
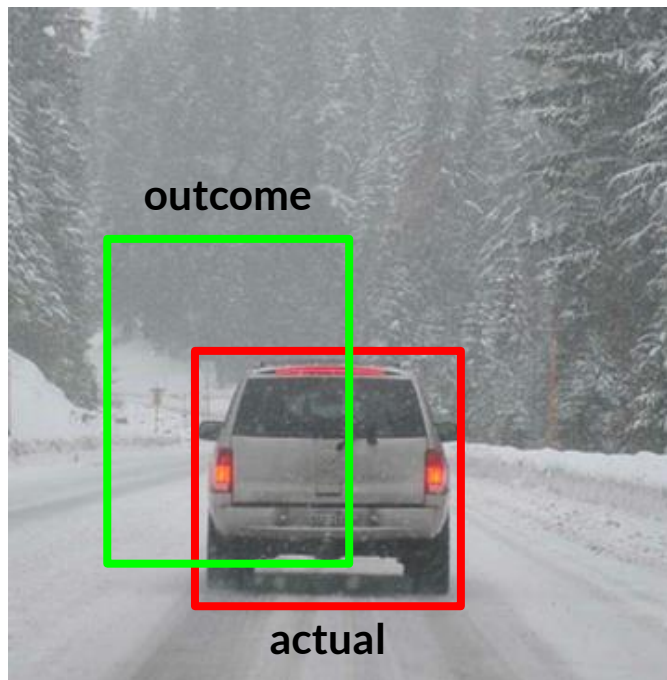
## Labels for Training Set

- A total of 9 grids
- Training labels for each grid
- Elements in each label vector = 1 + 4 + no. of classes
- Objects assigned to single grid cell (centers)
- For 3 classes, target output volume



# Evaluating Object Localization - Intersection Over Union

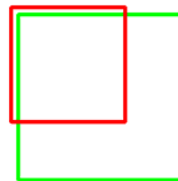
*Question: Is this a good or bad outcome?*



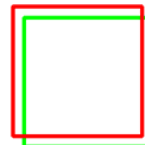
IoU: 0.4034

IoU: 0.7330

IoU: 0.9264



Poor

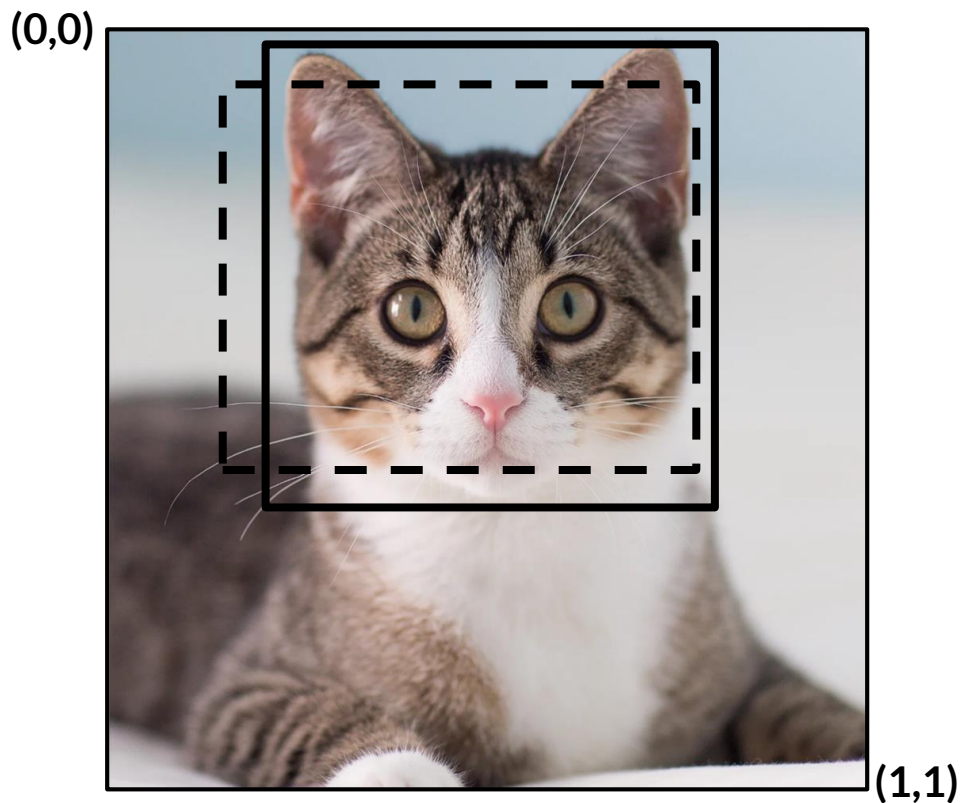


Good



Excellent

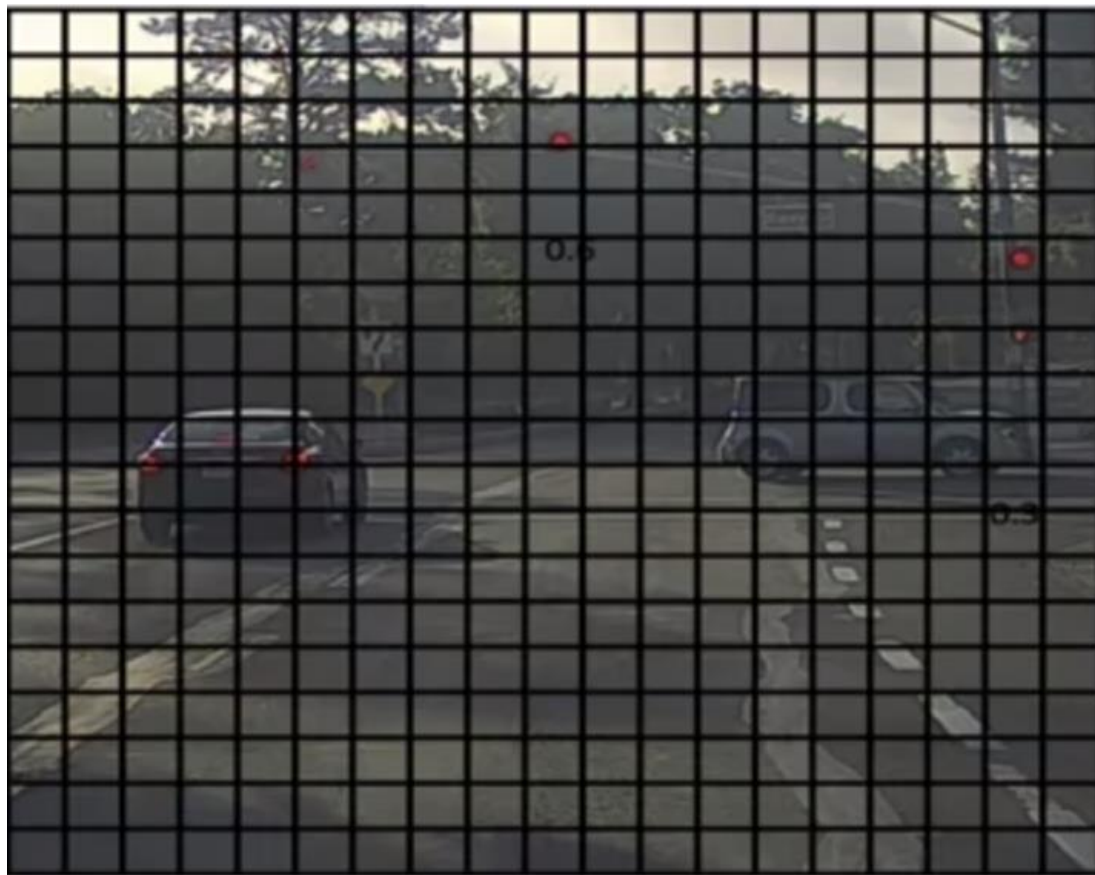
# Evaluating Object Localization - Intersection Over Union



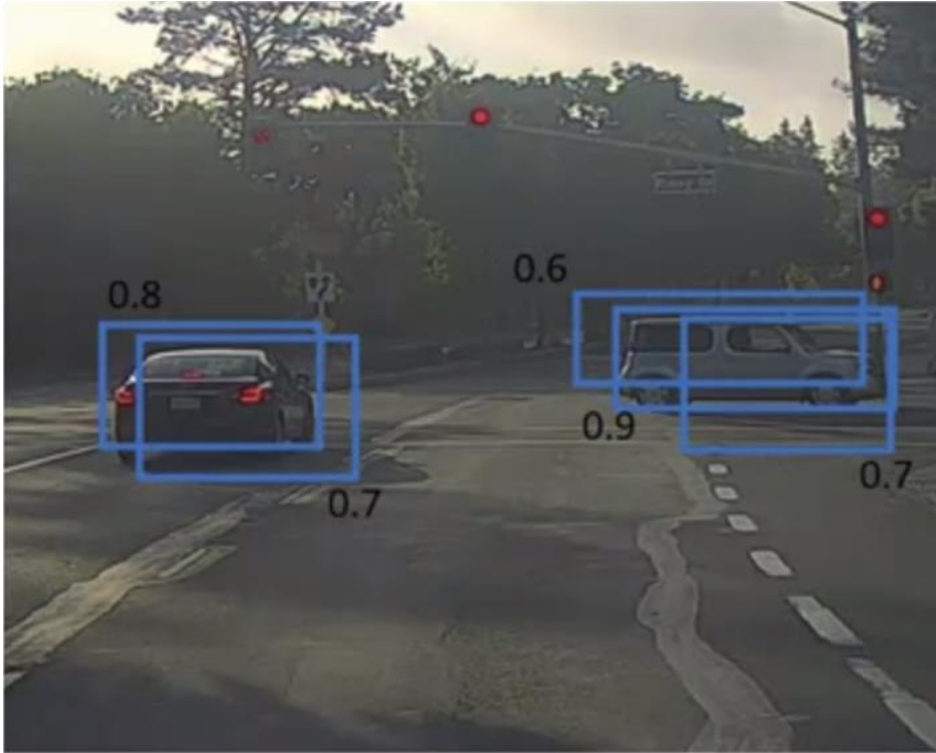
# Non-max Suppression



# Non-max Suppression



# Non-max Suppression

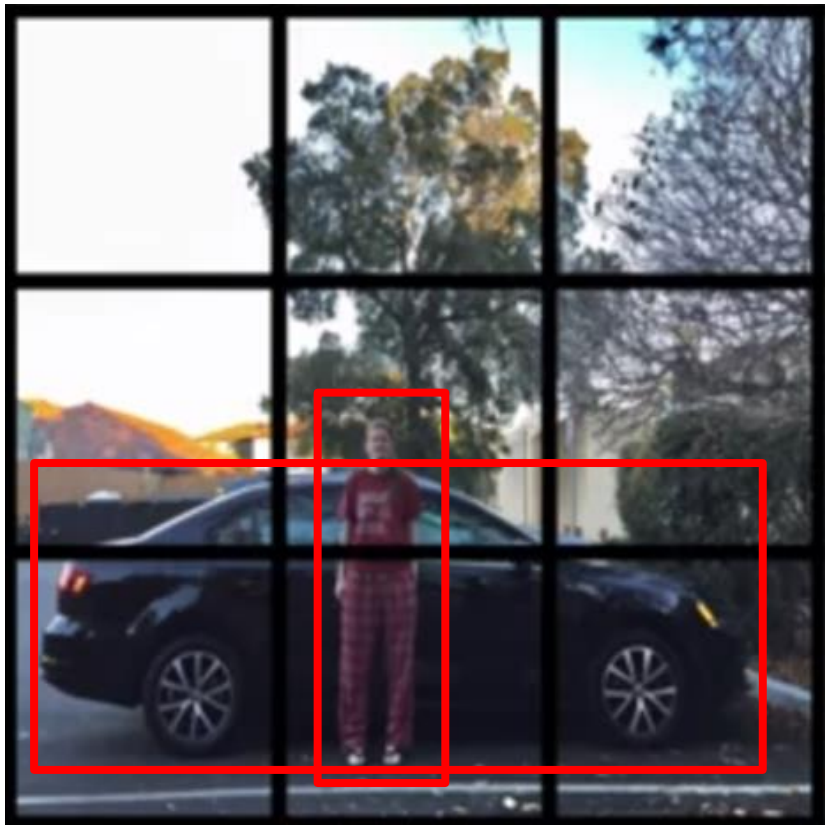


## Non-max Algorithm

- Go to each detection box
- Look at the probabilities
- Discard those with  $P_c < \text{threshold}$
- For remaining boxes:
  - Pick one with largest  $P_c$
  - Output that as prediction
  - Discard remaining with higher IoUs

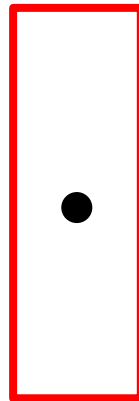


# Multiple Objects: Anchor Boxes

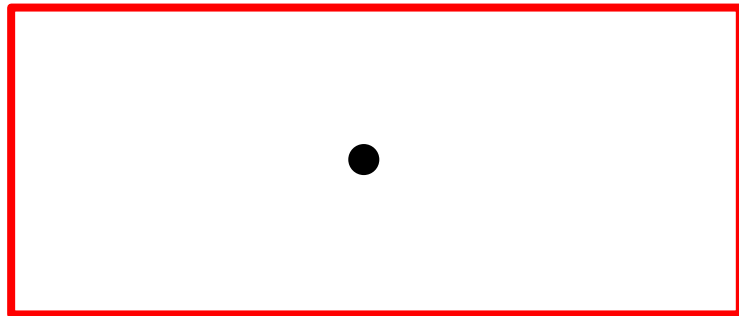


$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

*Anchor Box 1*

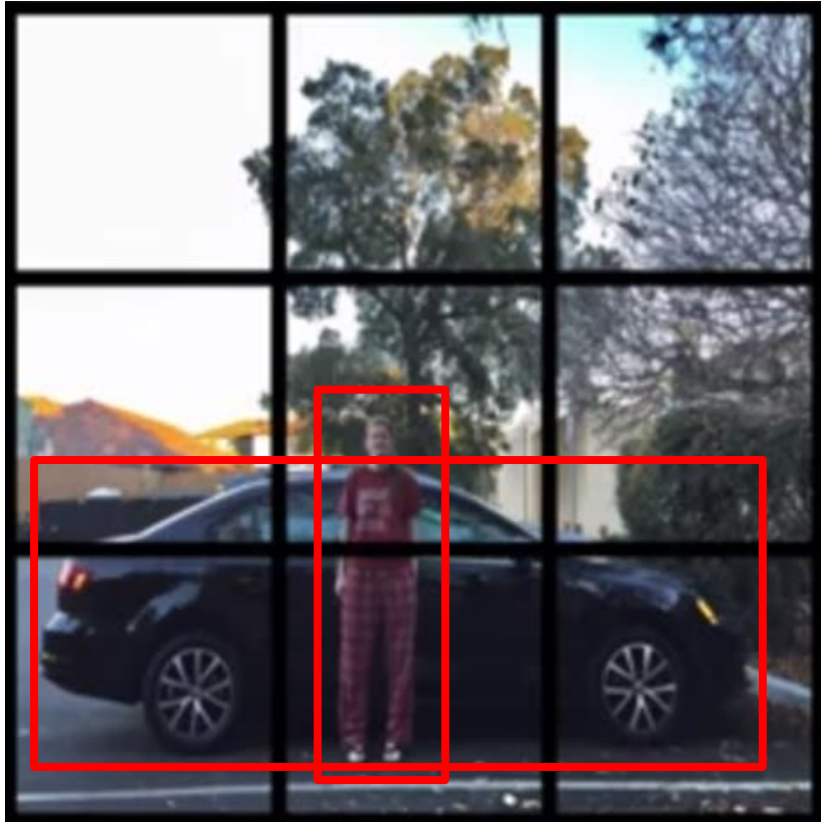


*Anchor Box 2*





# Multiple Objects: Anchor Boxes



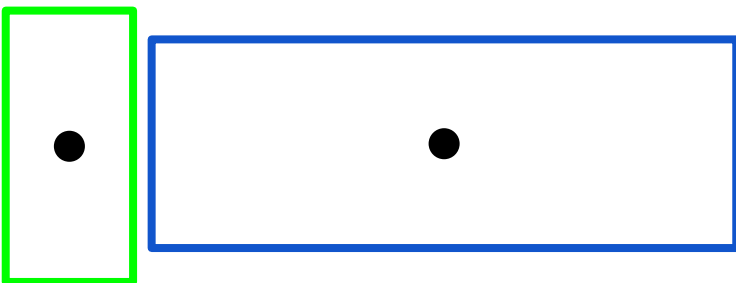
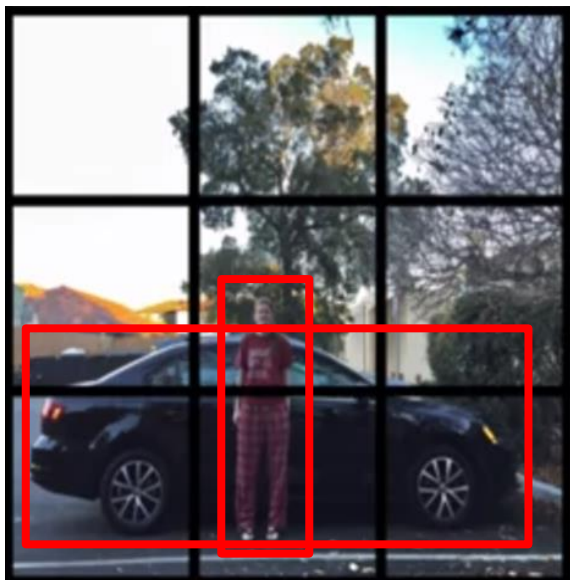
*Previously (3 x 3 x 8)*

Each object in the training image is assigned to grid cell that contains that objects midpoint

*With Two Anchor Boxes (3 x 3 x 2 x 8)*

Each object in the training image is assigned to grid cell that contains that objects midpoint **and anchor box for the grid cell with higher IoU**

# Multiple Objects in a Grid: Anchor Boxes



$y =$

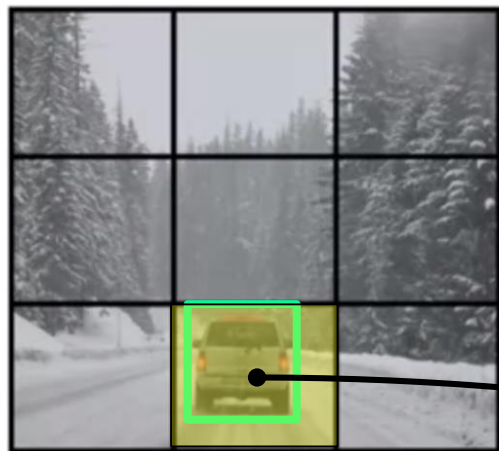
$$\begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

# YOLO Example

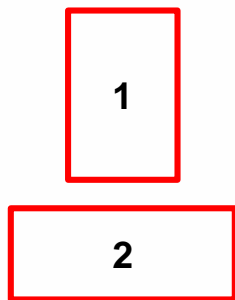
Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

## Training

- 1 - pedestrian
- 2 - car
- 3 - motorcycle



$y$  is  $3 \times 3 \times 2 \times 8$



$y =$

$\begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$

## For 3 Classes; Two Anchor Boxes

1. Which grid has center point?
2. Which class?
3. Which anchor box with more IoU with actual bounding box?
4. For each grid, generate a label vector
5. Train a ConvNet

$\begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}$

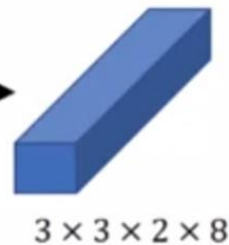
# YOLO Example

Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

## Making predictions



→ ... →



$y =$

$$\begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

1. Get predictions for each grid
2. Get rid of low  $P_c$  values
3. For each class, run non-max algo

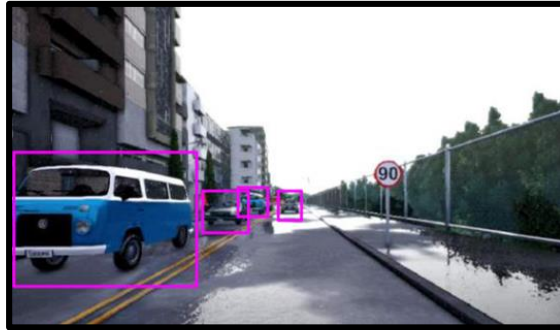
# Semantic Segmentation

# General Problem

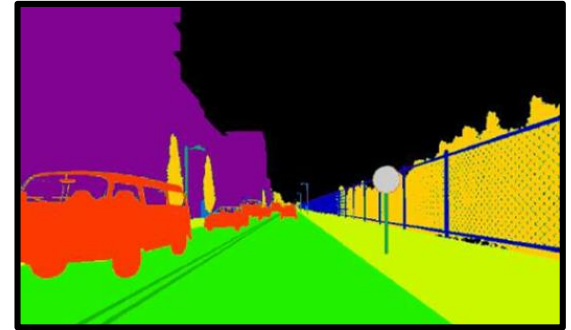
*Input Image*



*Object Detection*



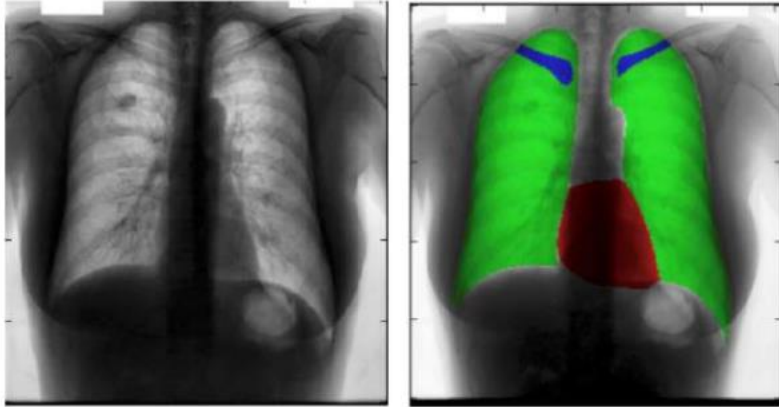
*Semantic Segmentation*



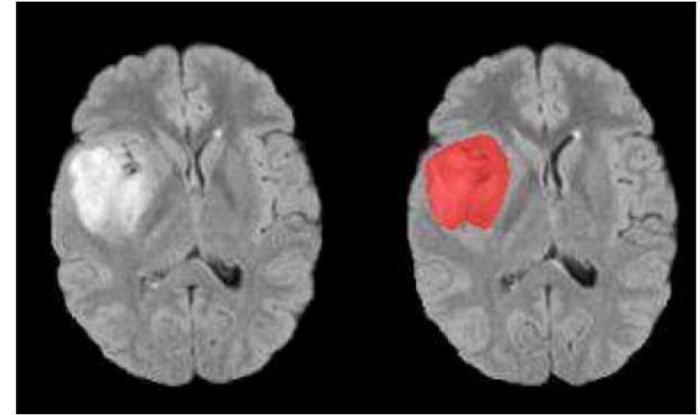
# U-Net - Original Motivation

Novikov, Alexey A., et al. "Fully convolutional architectures for multiclass segmentation in chest radiographs." *IEEE transactions on medical imaging* 37.8 (2018): 1865-1876.

Dong, Hao, et al. "Automatic brain tumor detection and segmentation using U-Net based fully convolutional networks." *annual conference on medical image understanding and analysis*. Springer, Cham, 2017.

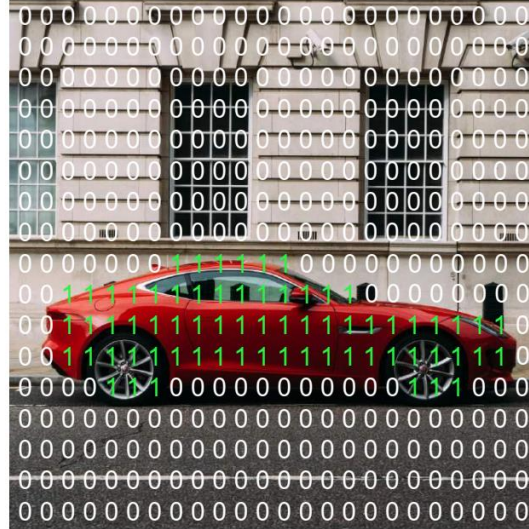


Chest X-Ray



Brain MRI

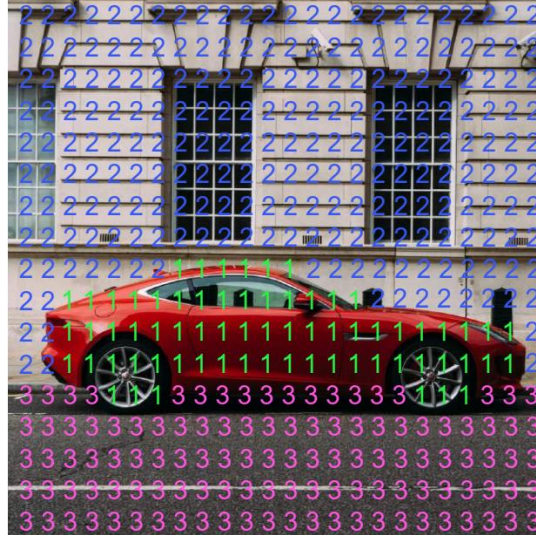
# Per Pixel Class Labels



0. *Not Car*  
1. *Car*

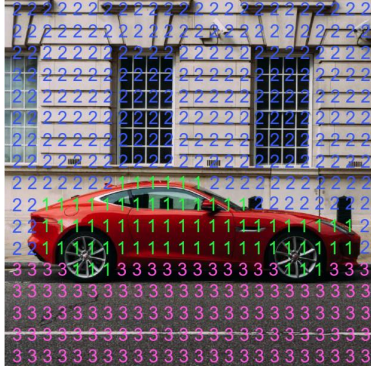


# Per Pixel Class Labels



1. Car
2. Building
3. Road

# Segmentation Map

[illegible]

# Transpose Convolution (Example)

1	2
3	2

2	2	1
2	1	2
3	2	1

$F = 3 \times 3$

$P = 1$

$S = 2$


# Transpose Convolution (Example)

2	1
3	2

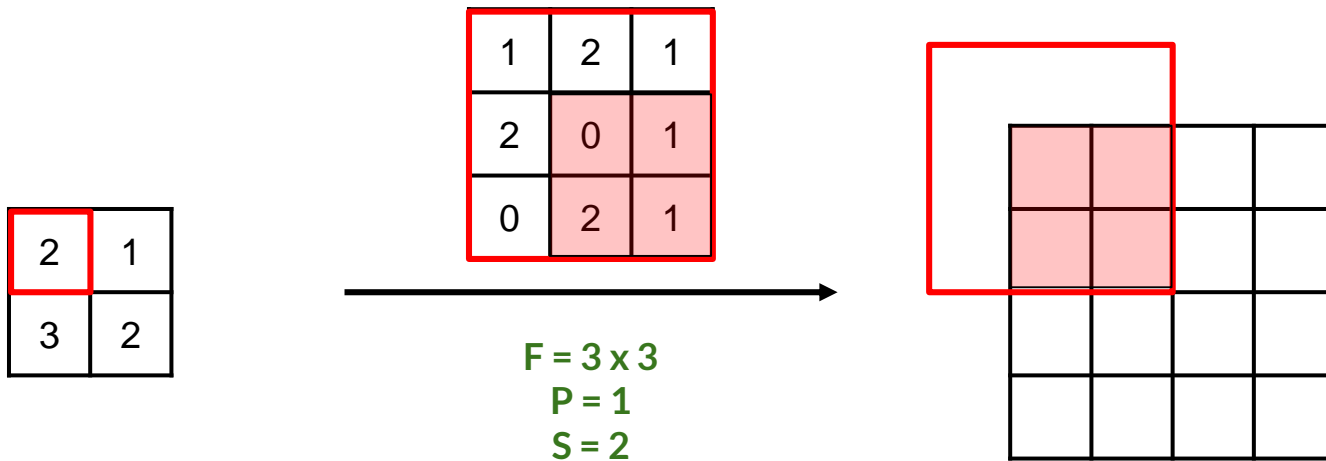
1	2	1
2	0	1
0	2	1

$F = 3 \times 3$

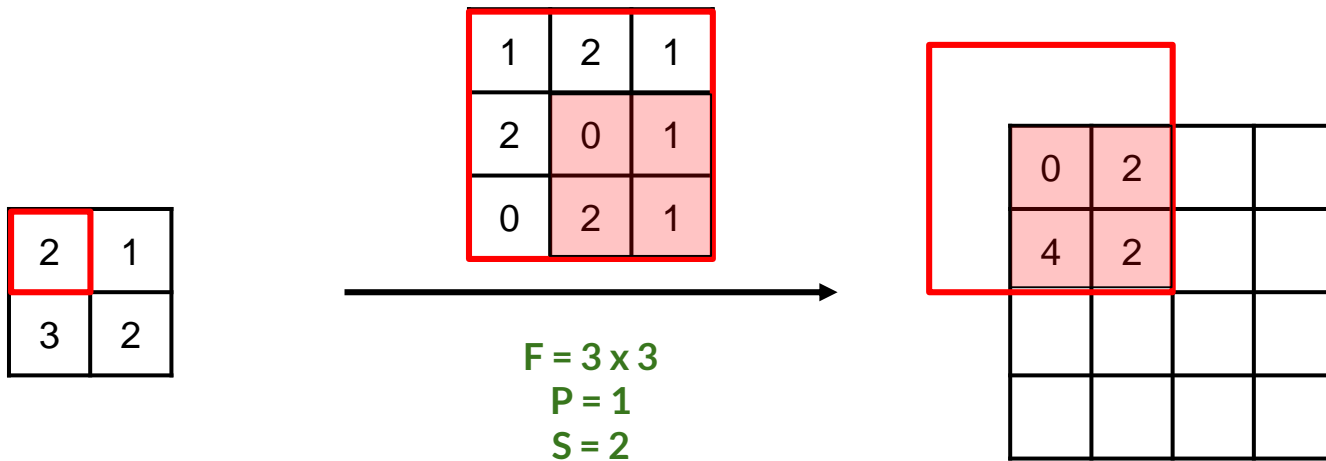
$P = 1$

$S = 2$

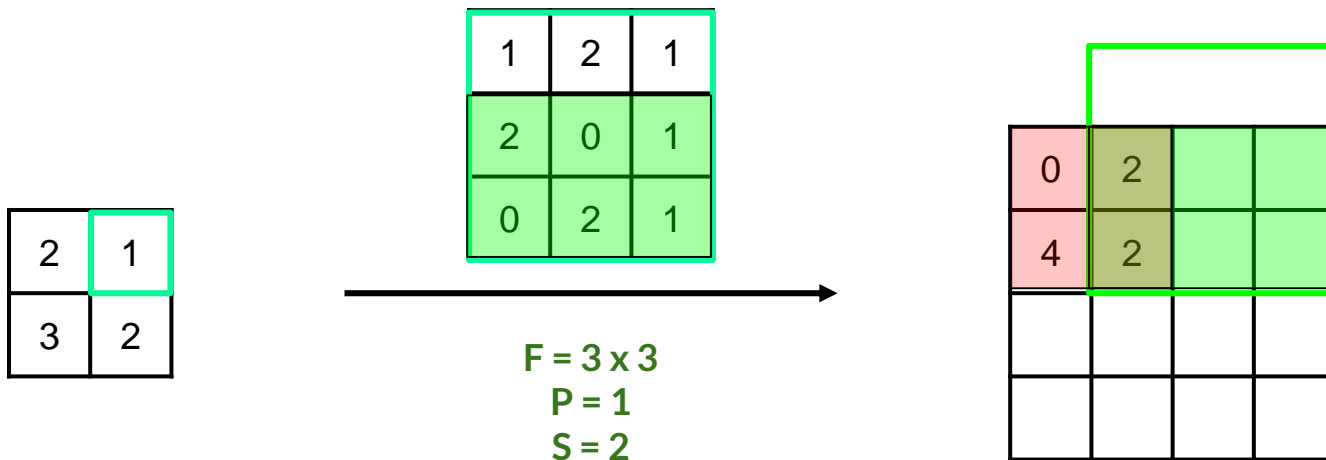

# Transpose Convolution (Example)



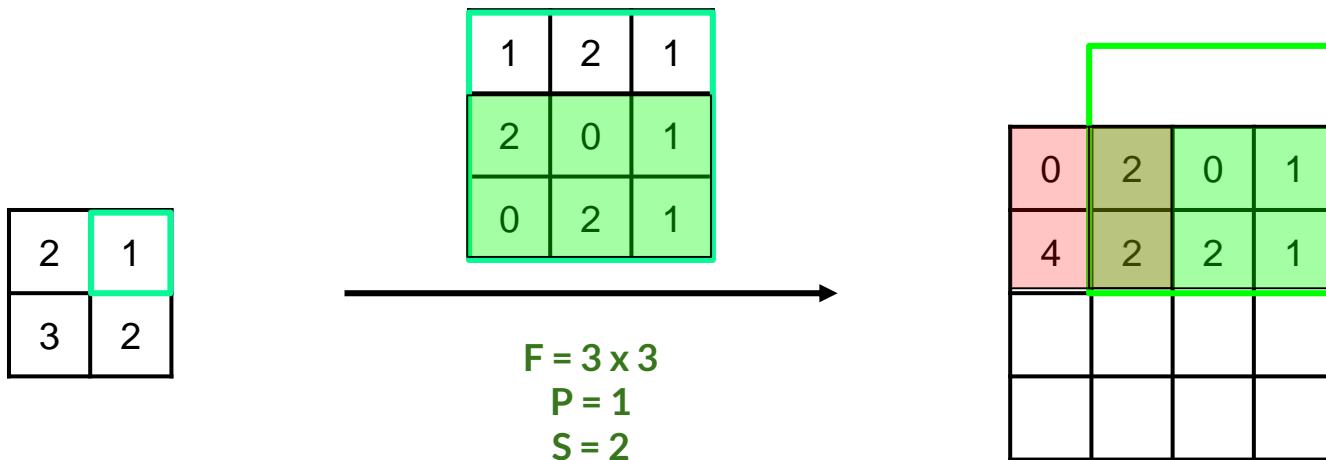
# Transpose Convolution (Example)



# Transpose Convolution (Example)

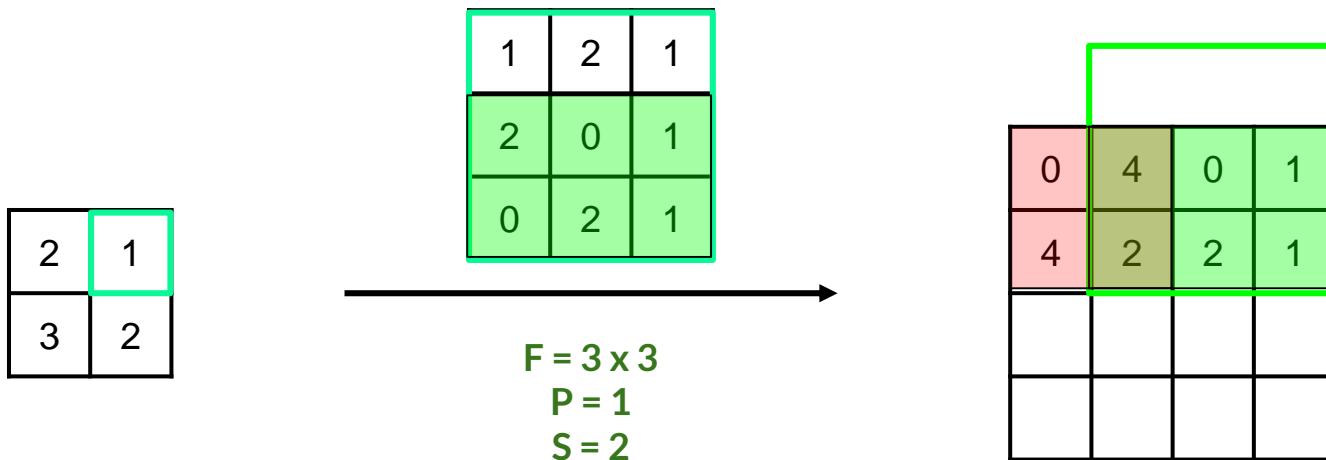


# Transpose Convolution (Example)





# Transpose Convolution (Example)



# Transpose Convolution (Example)

2	1
3	2

1	2	1
2	0	1
0	2	1

$F = 3 \times 3$   
 $P = 1$   
 $S = 2$

	0	4	0	1
	4	2	2	1

# Transpose Convolution (Example)

2	1
3	2

1	2	1
2	0	1
0	2	1

$F = 3 \times 3$   
 $P = 1$   
 $S = 2$

	0	4	0	1
	4	2	2	1
	0	3		
	6	3		

# Transpose Convolution (Example)

2	1
3	2

1	2	1
2	0	1
0	2	1

$F = 3 \times 3$   
 $P = 1$   
 $S = 2$

	0	4	0	1
	10	5	2	1
	0	3		
	6	3		

# Transpose Convolution (Example)

2	1
3	2

1	2	1
2	0	1
0	2	1

$F = 3 \times 3$   
 $P = 1$   
 $S = 2$

0	4	0	1
10	5	2	1
0	3		
6	3		

# Transpose Convolution (Example)

2	1
3	2

1	2	1
2	0	1
0	2	1

$F = 3 \times 3$   
 $P = 1$   
 $S = 2$

0	4	0	1
10	5	2	1
0	3	0	2
6	3	4	2

# Transpose Convolution (Example)

2	1
3	2

1	2	1
2	0	1
0	2	1

$F = 3 \times 3$   
 $P = 1$   
 $S = 2$

0	4	0	1
10	7	6	3
0	7	0	2
6	3	4	2

# Transpose Convolution (Example)

2	1
3	2

1	2	1
2	0	1
0	2	1

$F = 3 \times 3$

$P = 1$

$S = 2$

0	4	0	1
10	7	6	3
0	7	0	2
6	3	4	2



# U-Net - Original Motivation

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." *International Conference on Medical image computing and computer-assisted intervention*. Springer, Cham, 2015.

