Advanced Topics in Computer Vision and Image Processing

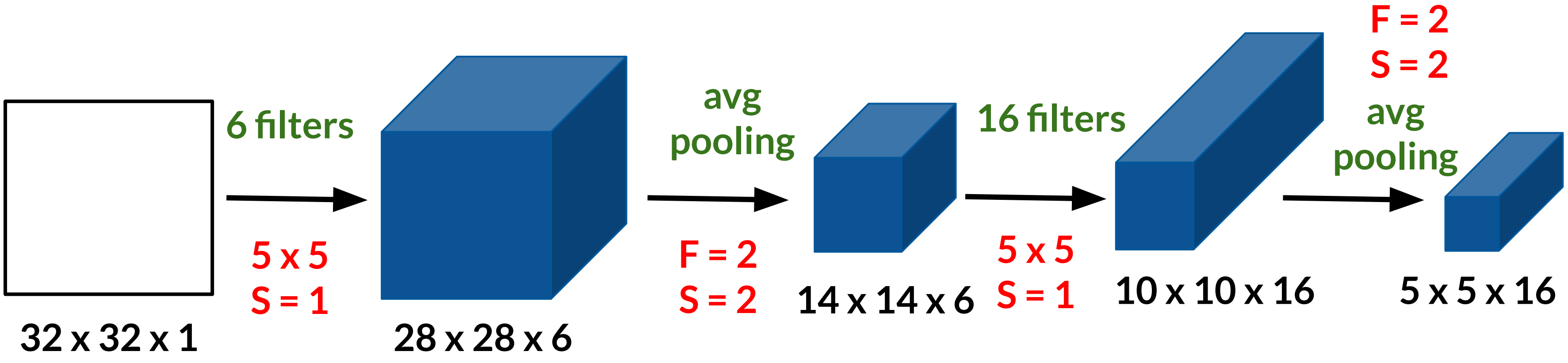
Lecture 12
Case Studies (Deep Convolutional Models)

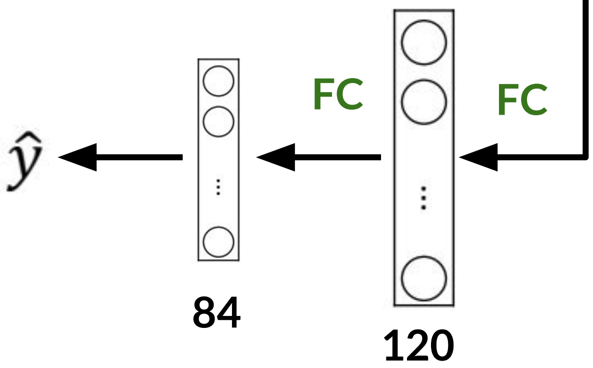
Asim D. Bakhshi, PhD
Military College of Signals

Object Classification

LeNet-5

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

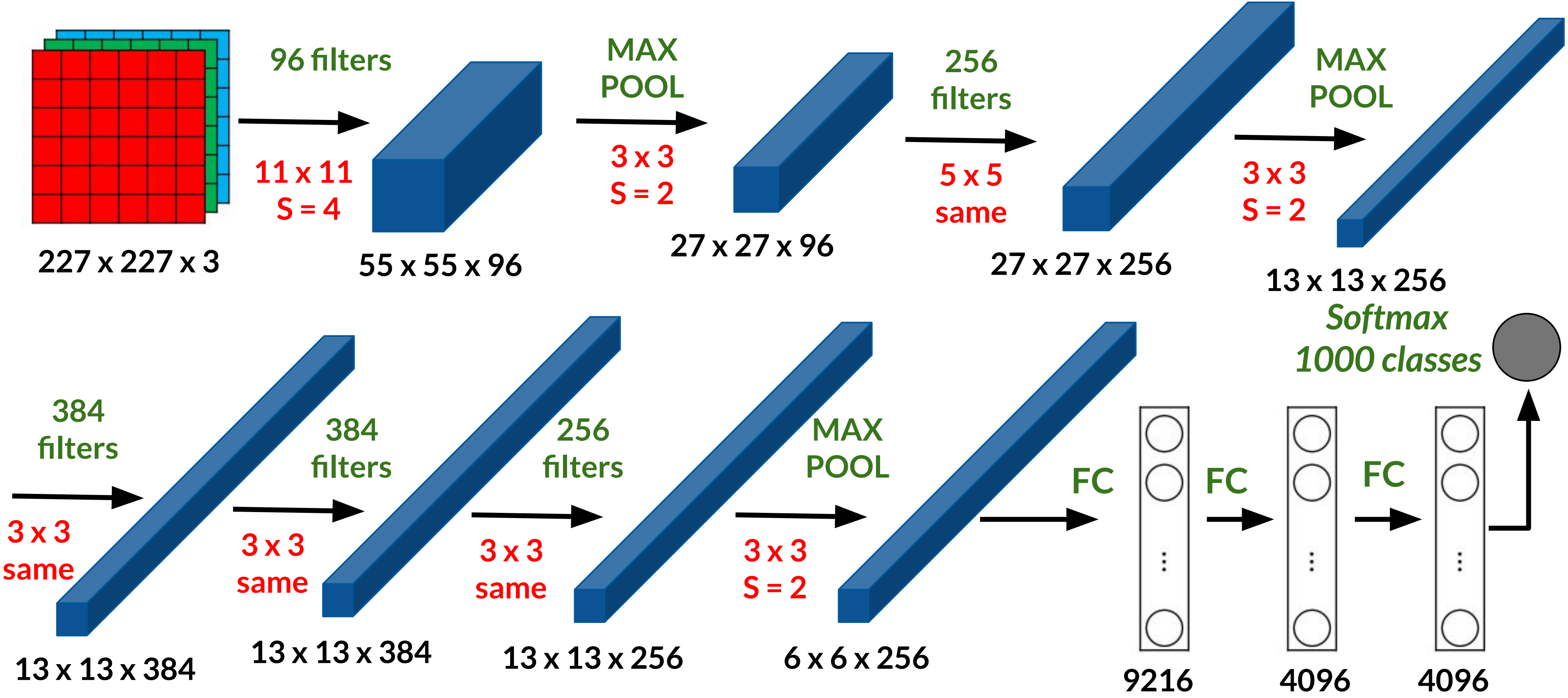




- *No padding in 1998*
- *About 60k parameters*
- *Size shrinks; channels increase*
- *conv-pool-conv-pool-fc-fc trend*
- *Sigmoid/Tanh; no ReLU*
- *Tough paper to read*

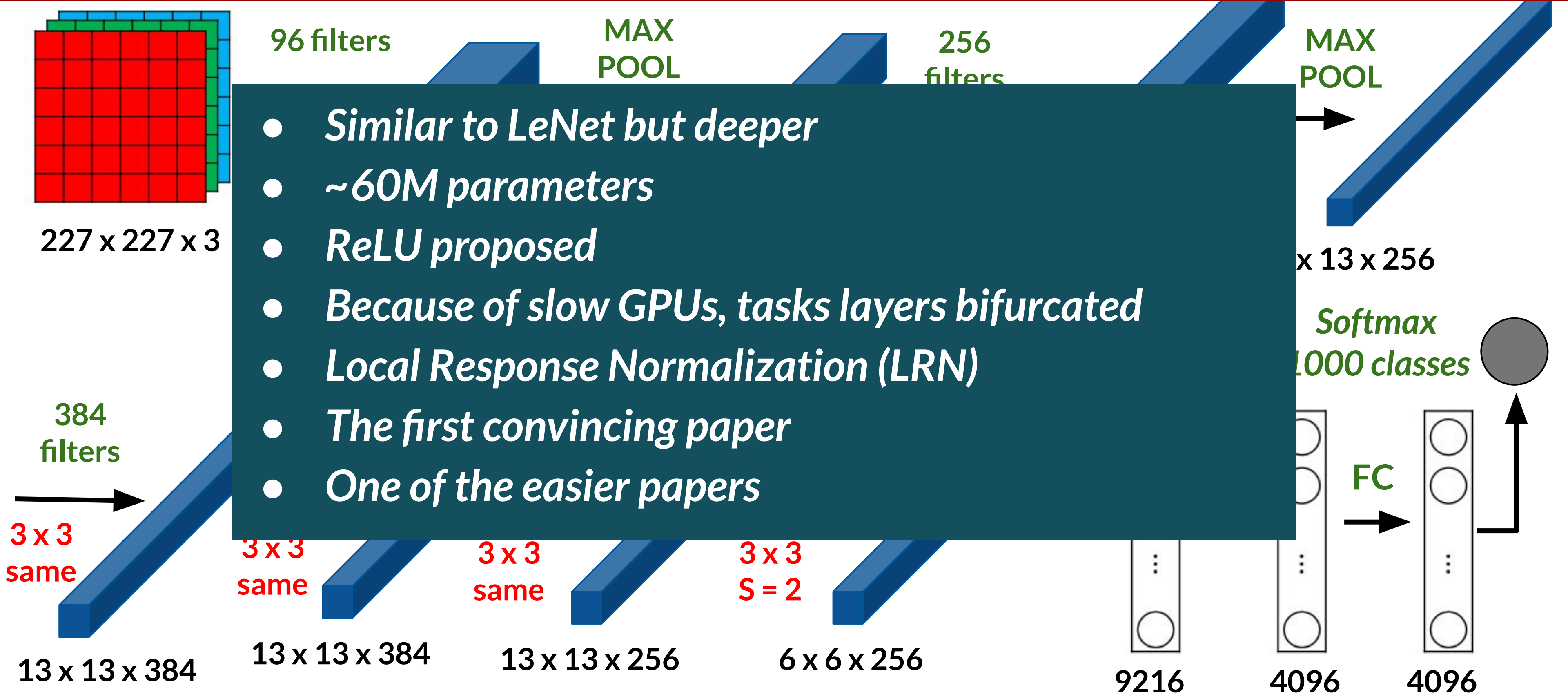
AlexNet

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



AlexNet

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

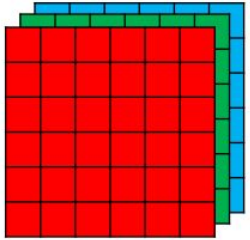


- *Similar to LeNet but deeper*
- *~60M parameters*
- *ReLU proposed*
- *Because of slow GPUs, tasks layers bifurcated*
- *Local Response Normalization (LRN)*
- *The first convincing paper*
- *One of the easier papers*

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 → CONV64 x2 → 224 x 224 x 64 → POOL → 112 x 112 x 64 → CONV128 x2 → 112 x 112 x 128

→ POOL → 56 x 56 x 128 → CONV256 x2 → 56 x 56 x 256 → POOL → 28 x 28 x 256

→ CONV512 x3 → 28 x 28 x 512 → POOL → 14 x 14 x 512 → CONV512 x3 → 14 x 14 x 512 → POOL → 7 x 7 x 512

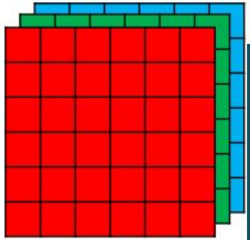
→ FC → 4096 → FC → 4096 →

Softmax
1000 classes

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

x 128

256

CONV512 x3

2

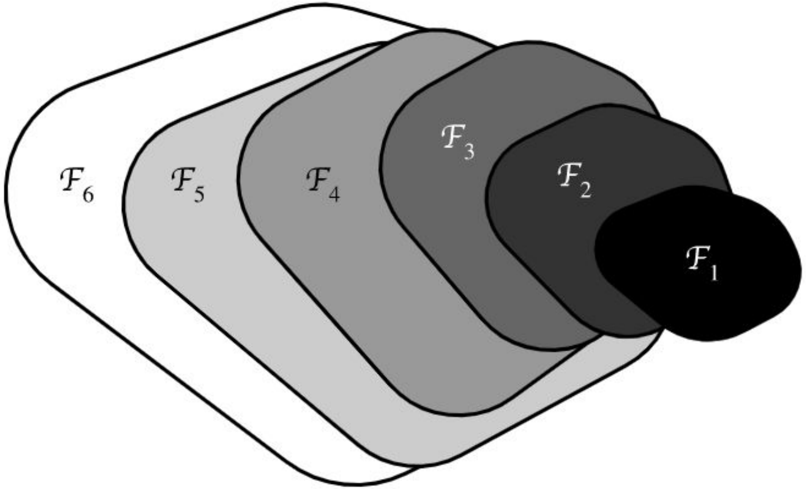
7 x 7 x 512

- *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*

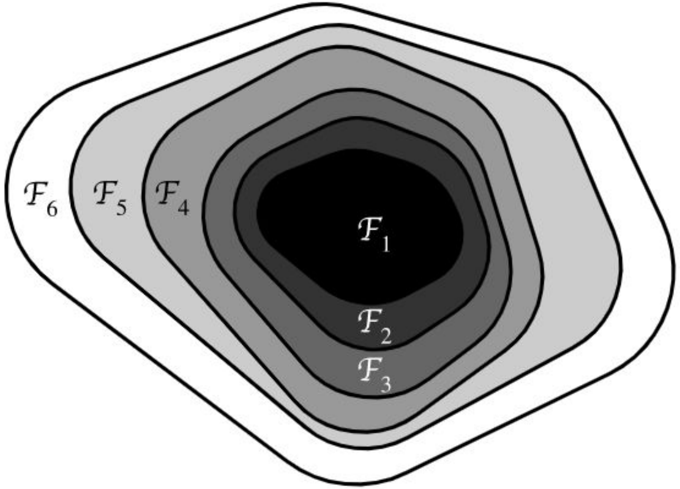
FC → 4096 FC → 4096 → *Softmax 1000 classes*

ResNet

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



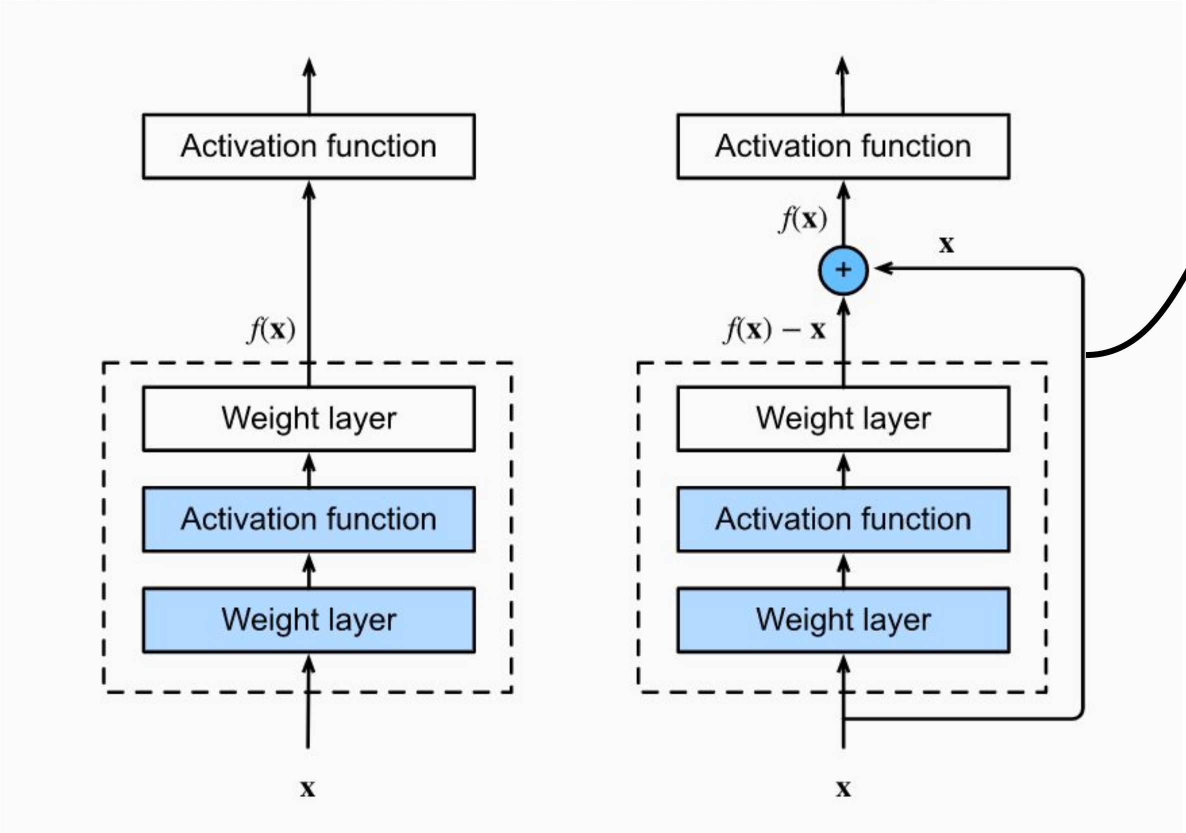
Non-nested function classes



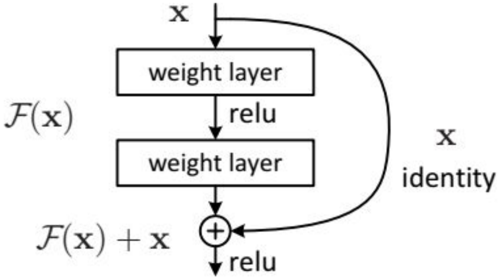
Nested function classes

ResNet

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

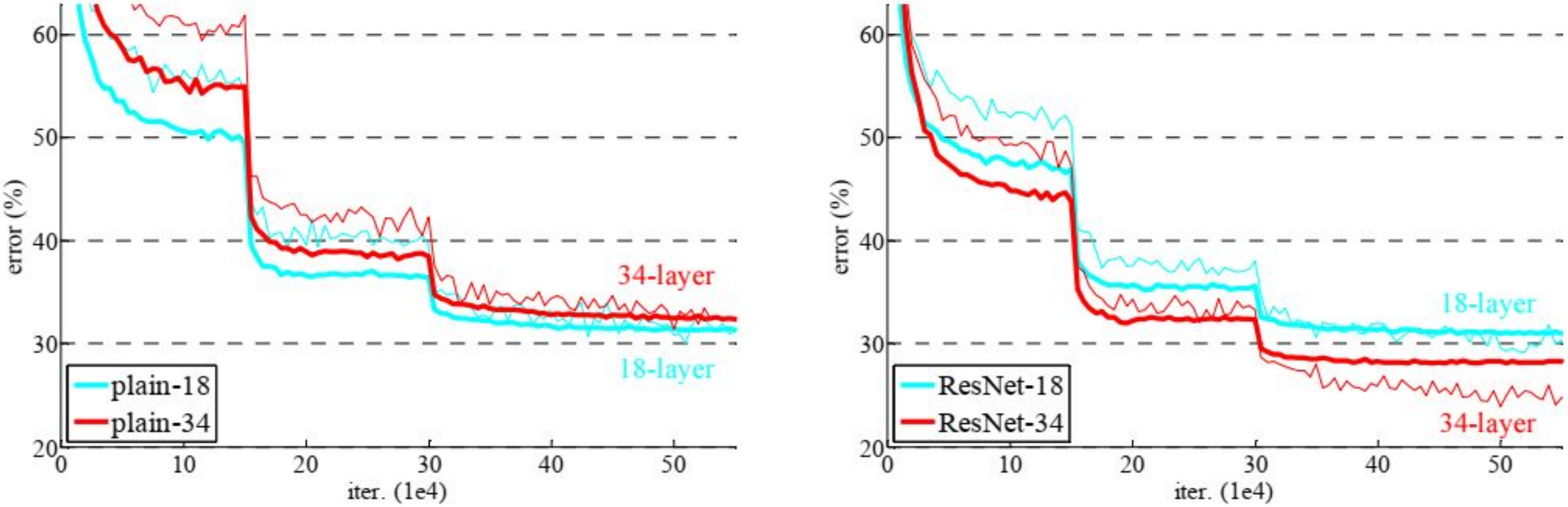


skip connection



ResNet

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



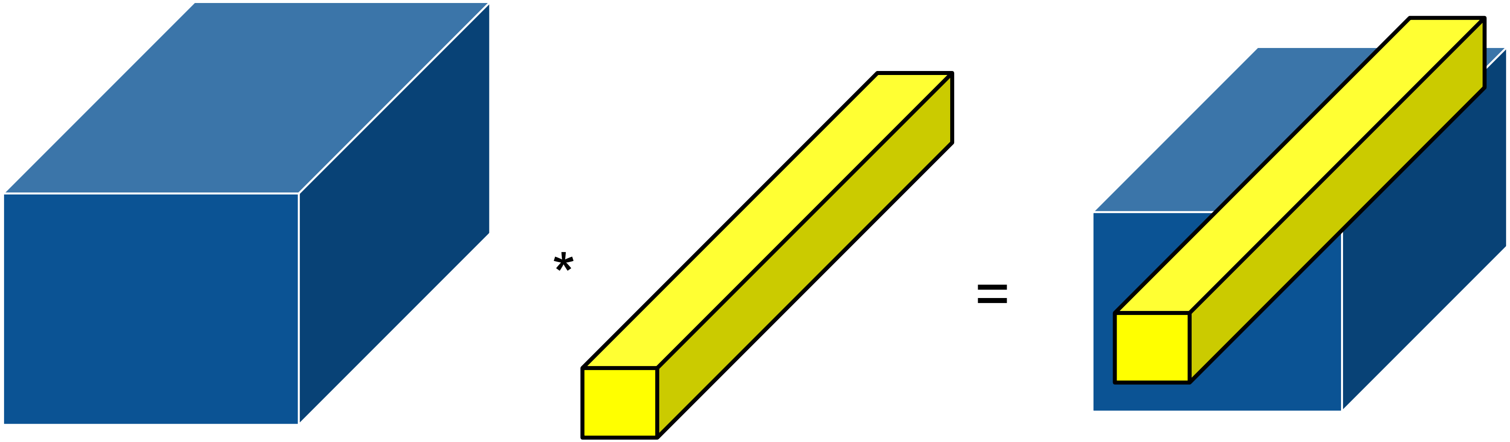
Network in Network

Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." *arXiv preprint arXiv:1312.4400* (2013).

$$
\begin{array}{|c|c|c|c|c|c|}
\hline
1 & 2 & 3 & 4 & 5 & 6 \\ \hline
3 & 4 & 6 & 7 & 1 & 0 \\ \hline
3 & 9 & 8 & 5 & 2 & 0 \\ \hline
2 & 1 & 0 & 9 & 8 & 4 \\ \hline
2 & 3 & 4 & 9 & 0 & 0 \\ \hline
1 & 1 & 6 & 5 & 8 & 3 \\ \hline
\end{array}
\quad * \quad \boxed{4} \quad = \quad
\begin{array}{|c|c|c|c|c|c|}
\hline
 & & & & & \\ \hline
 & & & & & \\ \hline
 & & & & & \\ \hline
 & & & & & \\ \hline
 & & & & & \\ \hline
 & & & & & \\ \hline
\end{array}
$$

Network in Network

Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." *arXiv preprint arXiv:1312.4400* (2013).



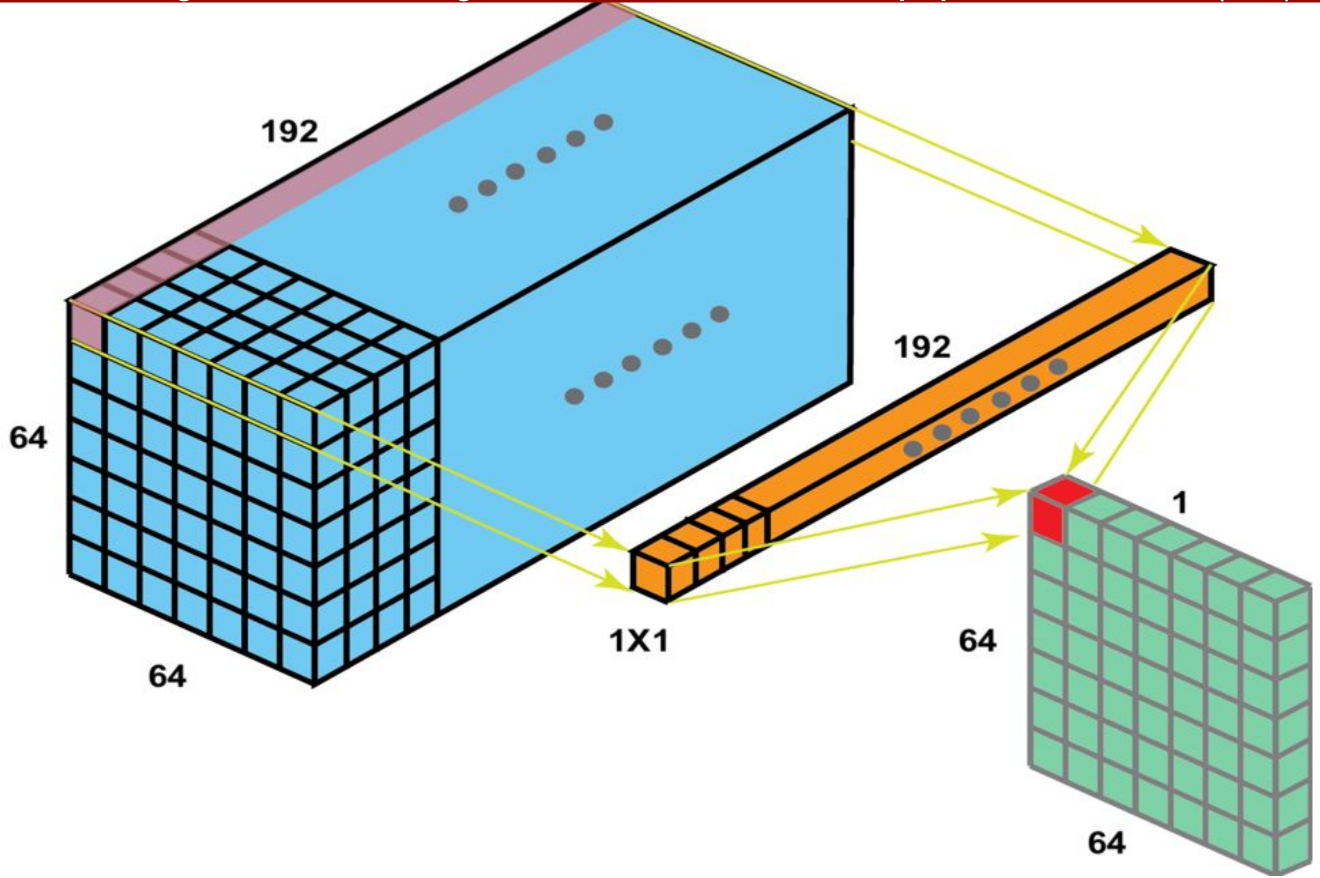
6 x 6 x 32 * 1 x 1 x 32 = 6 x 6 x #filters

Apply ReLU

above illustration shows one slice and depth of the output would depend on number of CONV1 filters

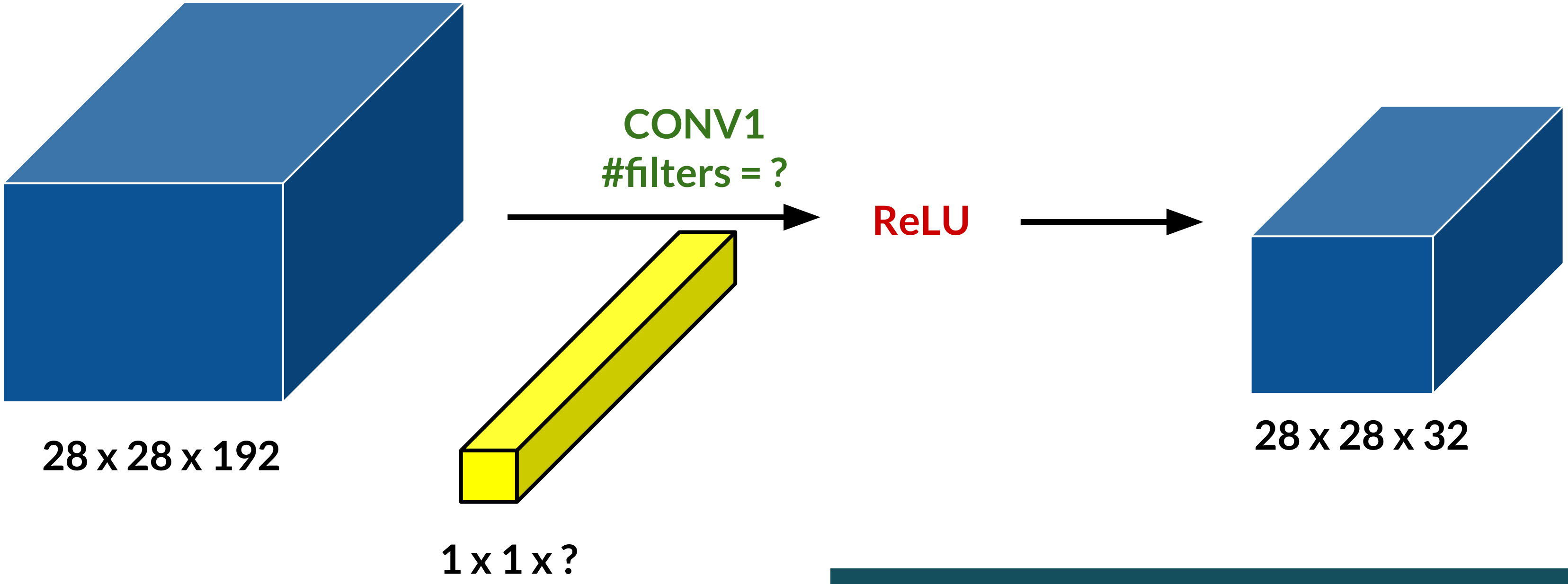
Network in Network

Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." *arXiv preprint arXiv:1312.4400* (2013).



Network in Network

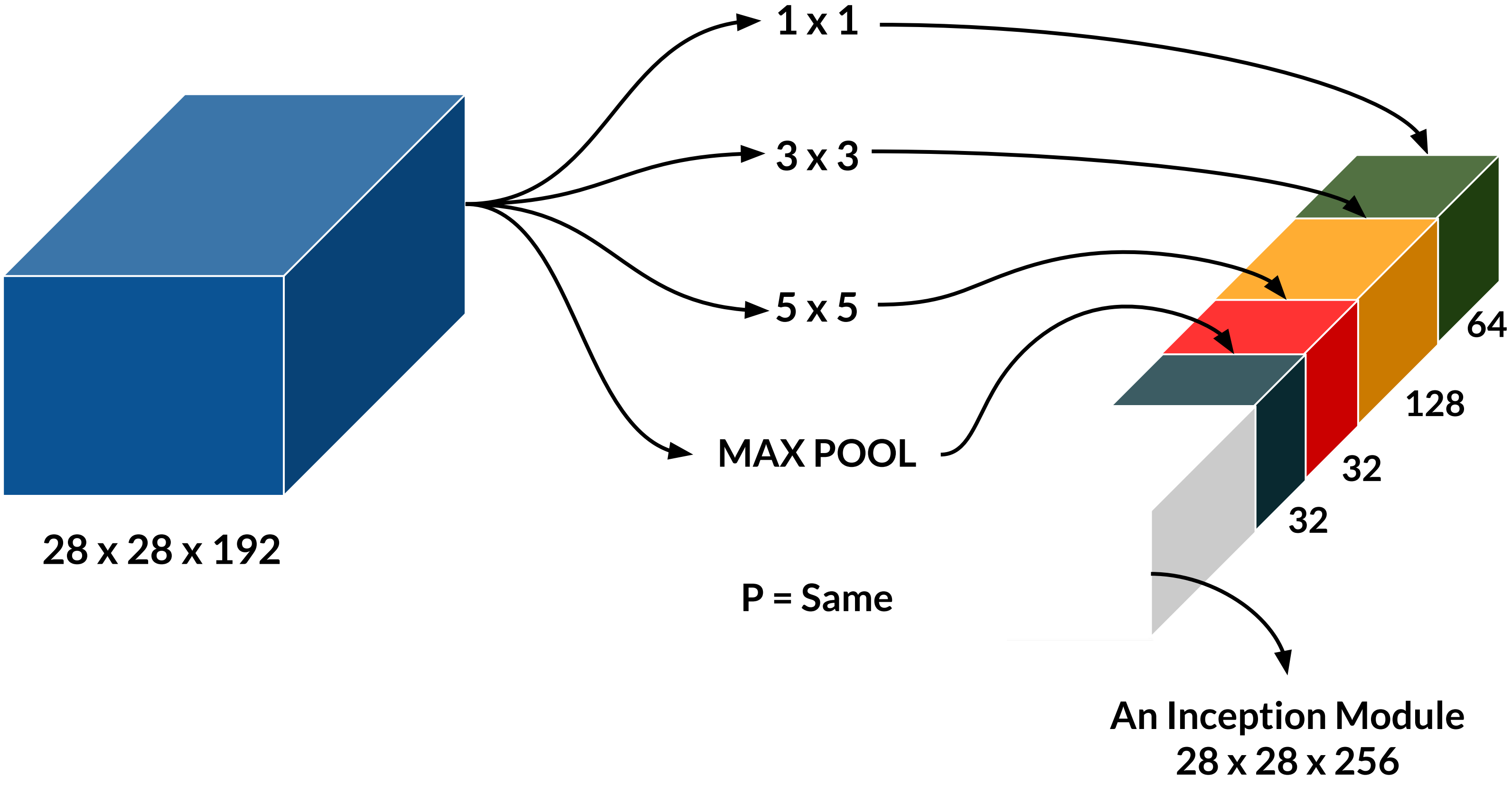
Lin, Min, Qiang Chen, and Shuicheng Yan. "Network in network." *arXiv preprint arXiv:1312.4400* (2013).



what if we don't want to shrink channels?

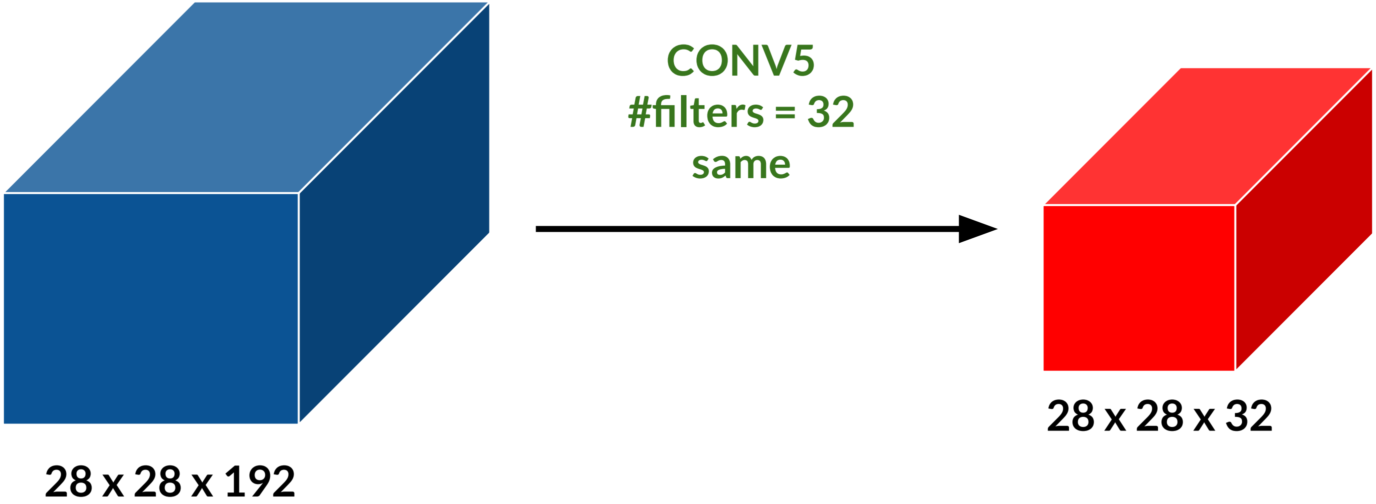
Inception Network

Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.



Inception Network

Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

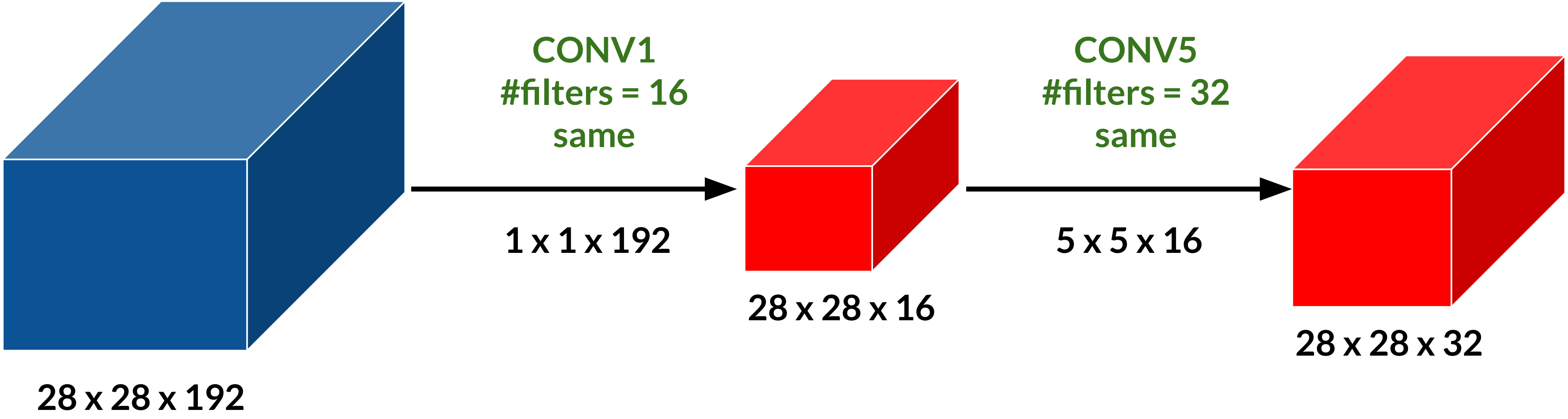


What is the computational cost?

- filter size = 5 x 5 x 192
- #filters = 32
- nos to compute = 28 x 28 x 32
- for each no. = 28 x 28 x 32 x 5 x 5 x 192 = 120 M

Inception Network

Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

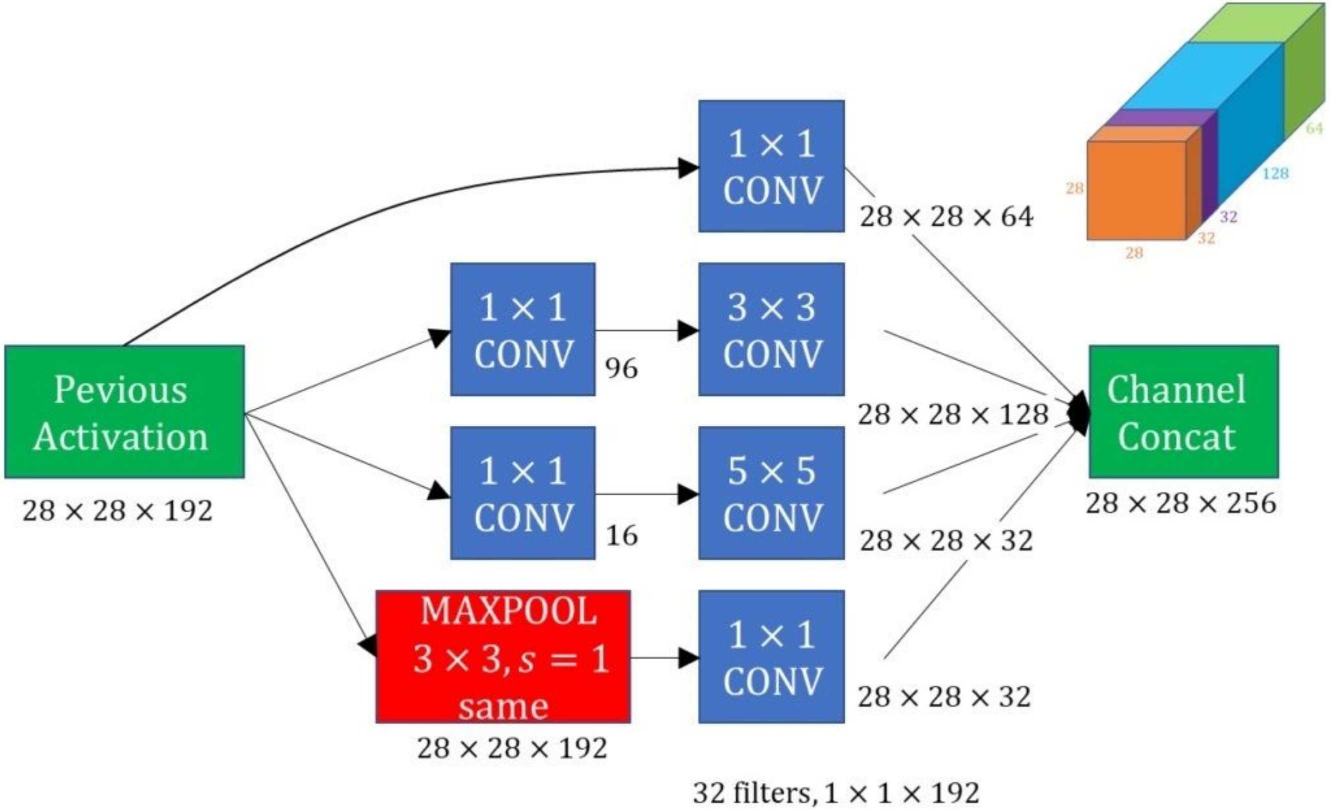


$28 \times 28 \times 192 \times 16 = 2.4$ M

$28 \times 28 \times 32 \times 5 \times 5 \times 16 = 10$ M

Inception Network

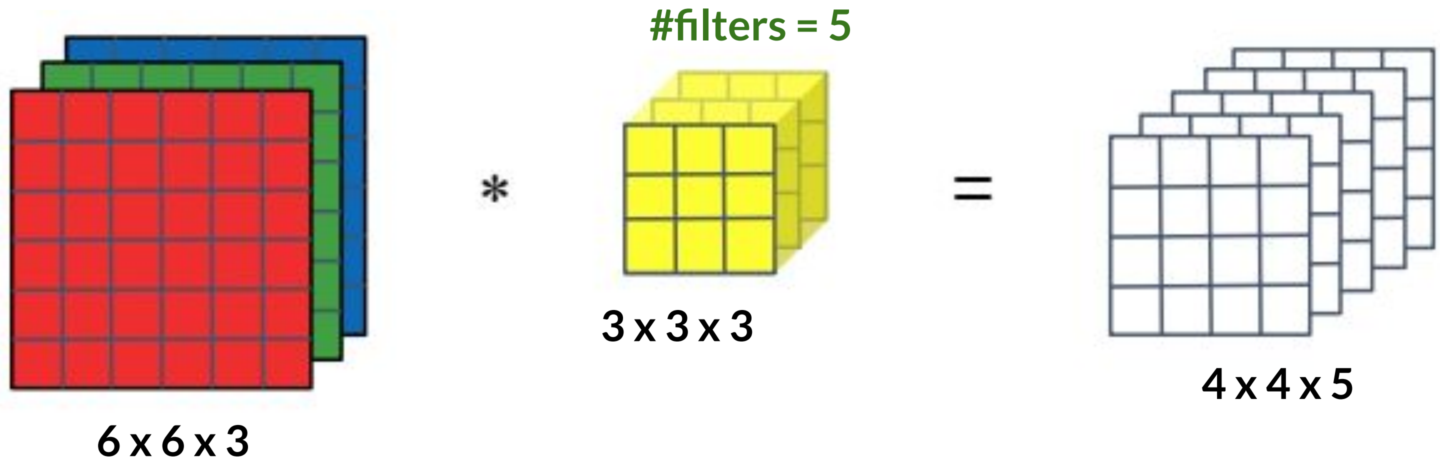
Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.



MobileNet

Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).

NORMAL CONVOLUTION

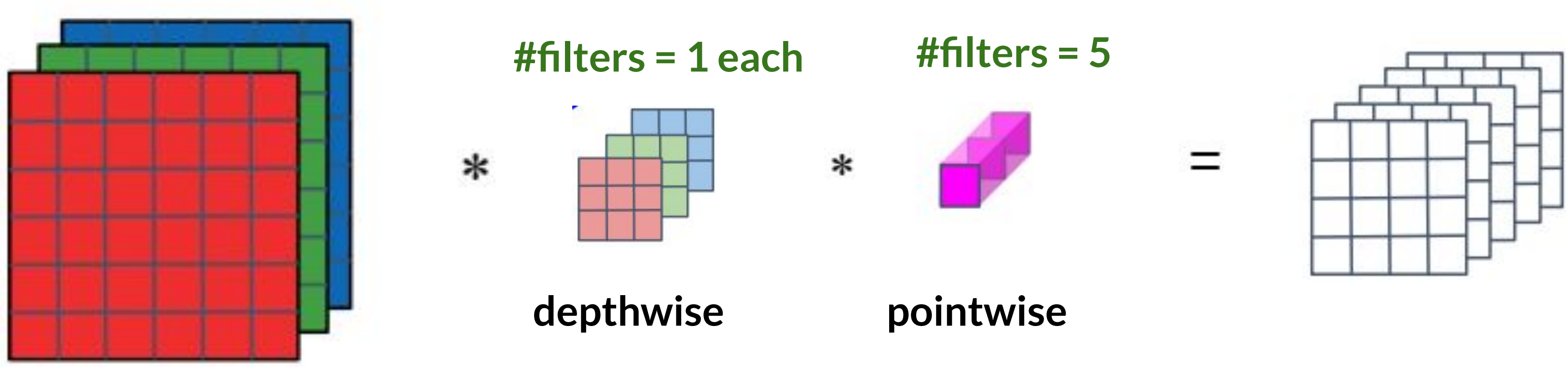


Computational Cost = #filter parameters x filter positions x #filters = 2160 muls

MobileNet

Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).

DEPTH WISE SEPARABLE CONVOLUTIONS

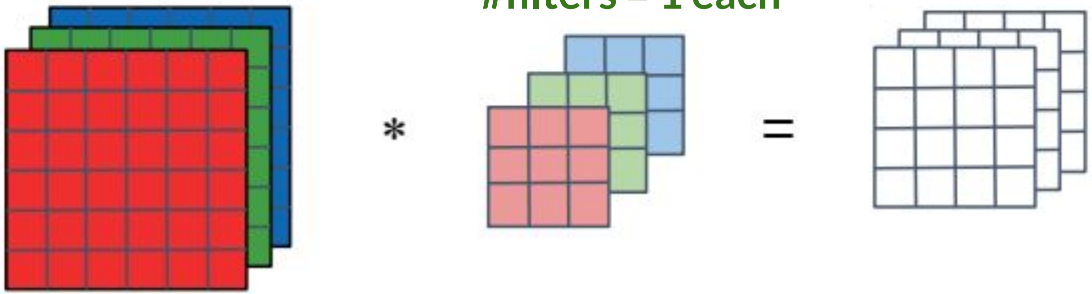


MobileNet

Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).

DEPTH WISE CONVOLUTIONS

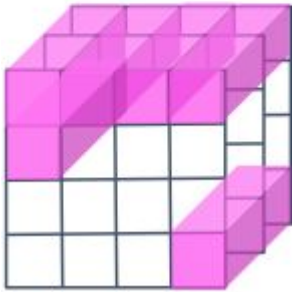
#filters = 1 each



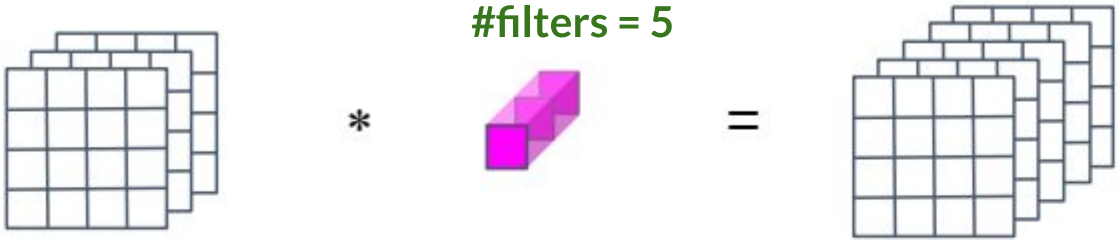
Depthwise Cost = #filter parameters x filter positions x #filters = 432 muls

MobileNet

Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).



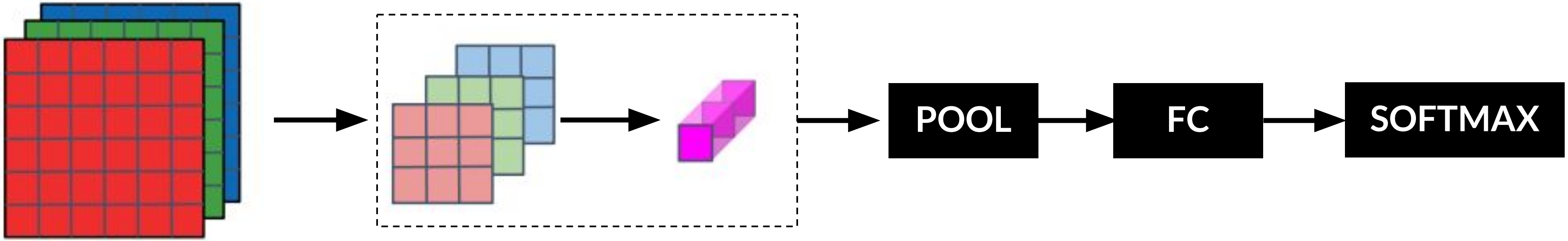
POINTWISE CONVOLUTIONS



Pointwise Cost = #filter parameters x filter positions x #filters = 240 muls

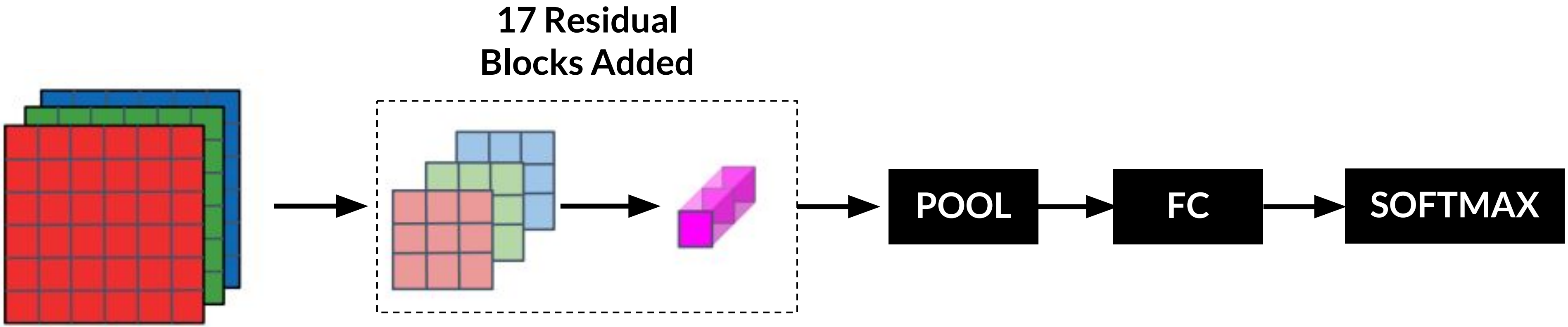
MobileNetv1

Howard, Andrew G., et al. "Mobilenets: Efficient convolutional neural networks for mobile vision applications." *arXiv preprint arXiv:1704.04861* (2017).



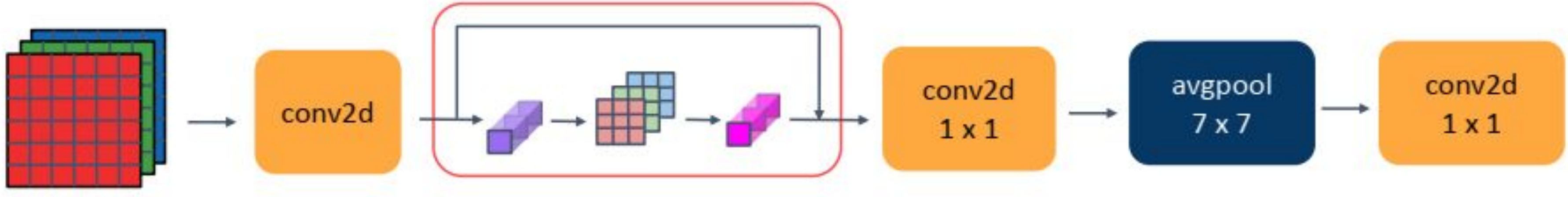
MobileNetv2

Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.



MobileNetv2

Sandler, Mark, et al. "Mobilenetv2: Inverted residuals and linear bottlenecks." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.



EfficientNet

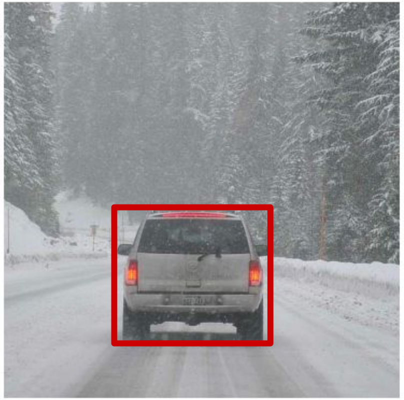
Tan, Mingxing, and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks." *International Conference on Machine Learning*. PMLR, 2019.

Object Detection

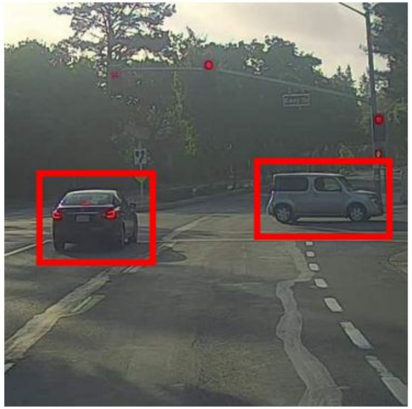
Localization and Detection

CLASSIFICATION

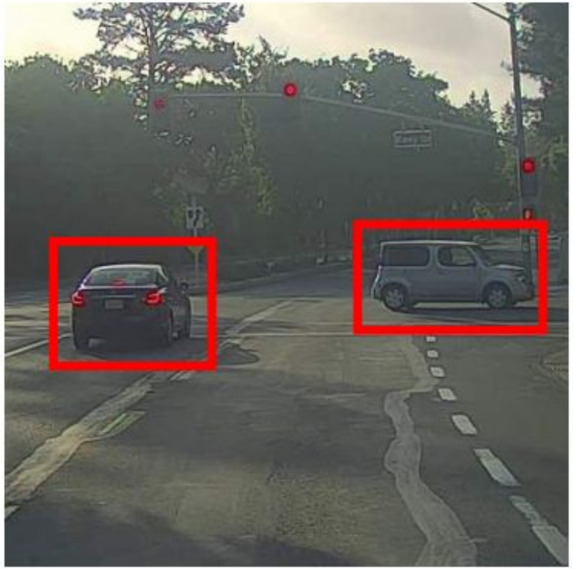
LOCALIZATION



DETECTION



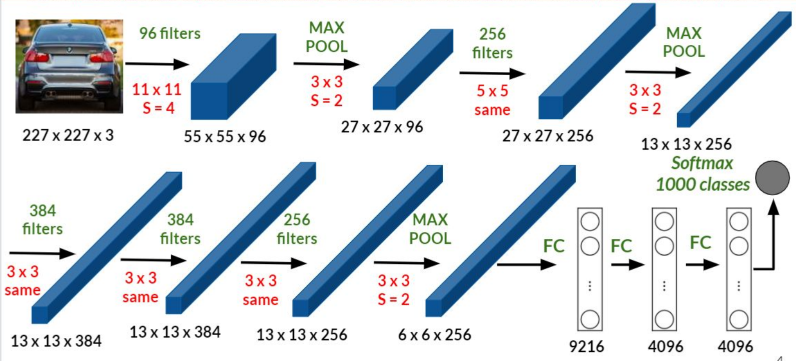
Localization and Detection



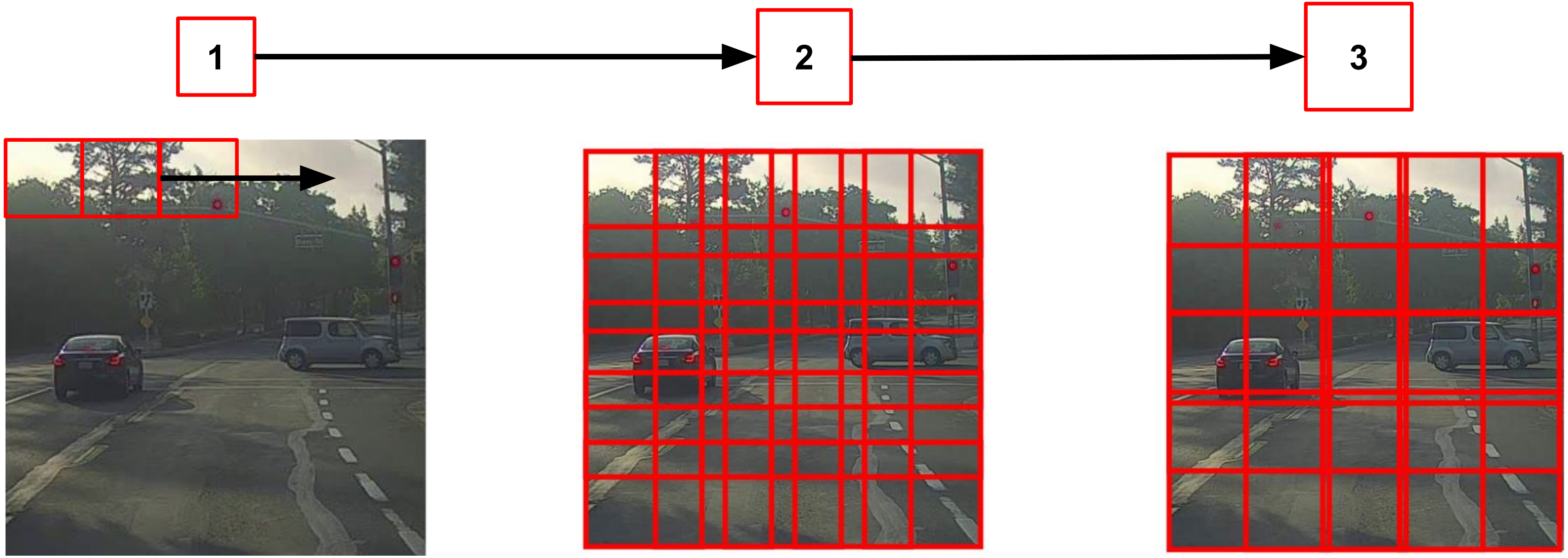
Training Set

x	y
	1
	1
	1
	0
	0

Train the ConvNet

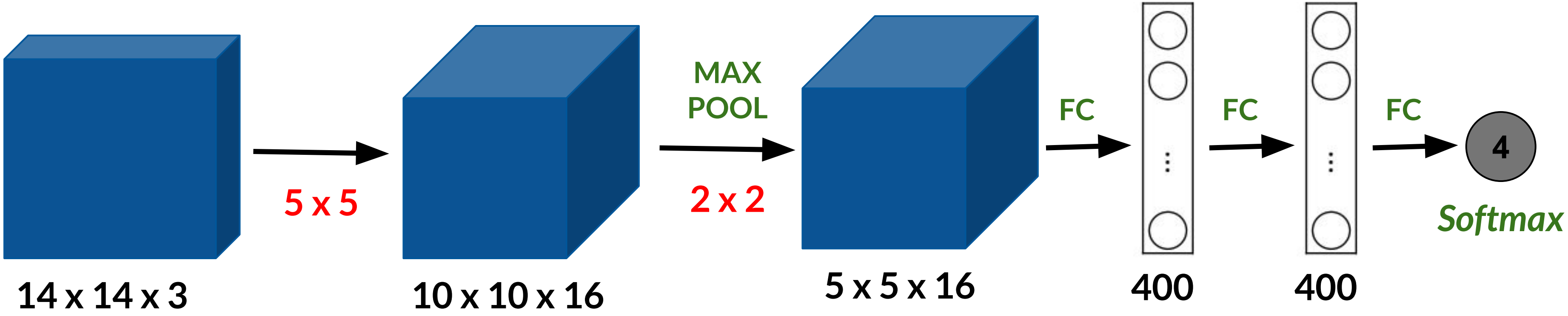


Localization and Detection (Sliding Windows)

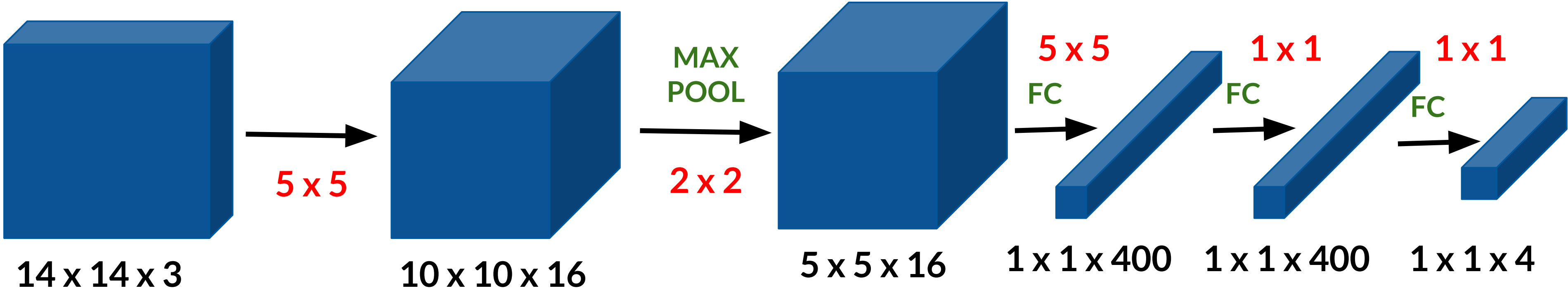


Problem: Computational Cost

Localization and Detection (Convolutional)

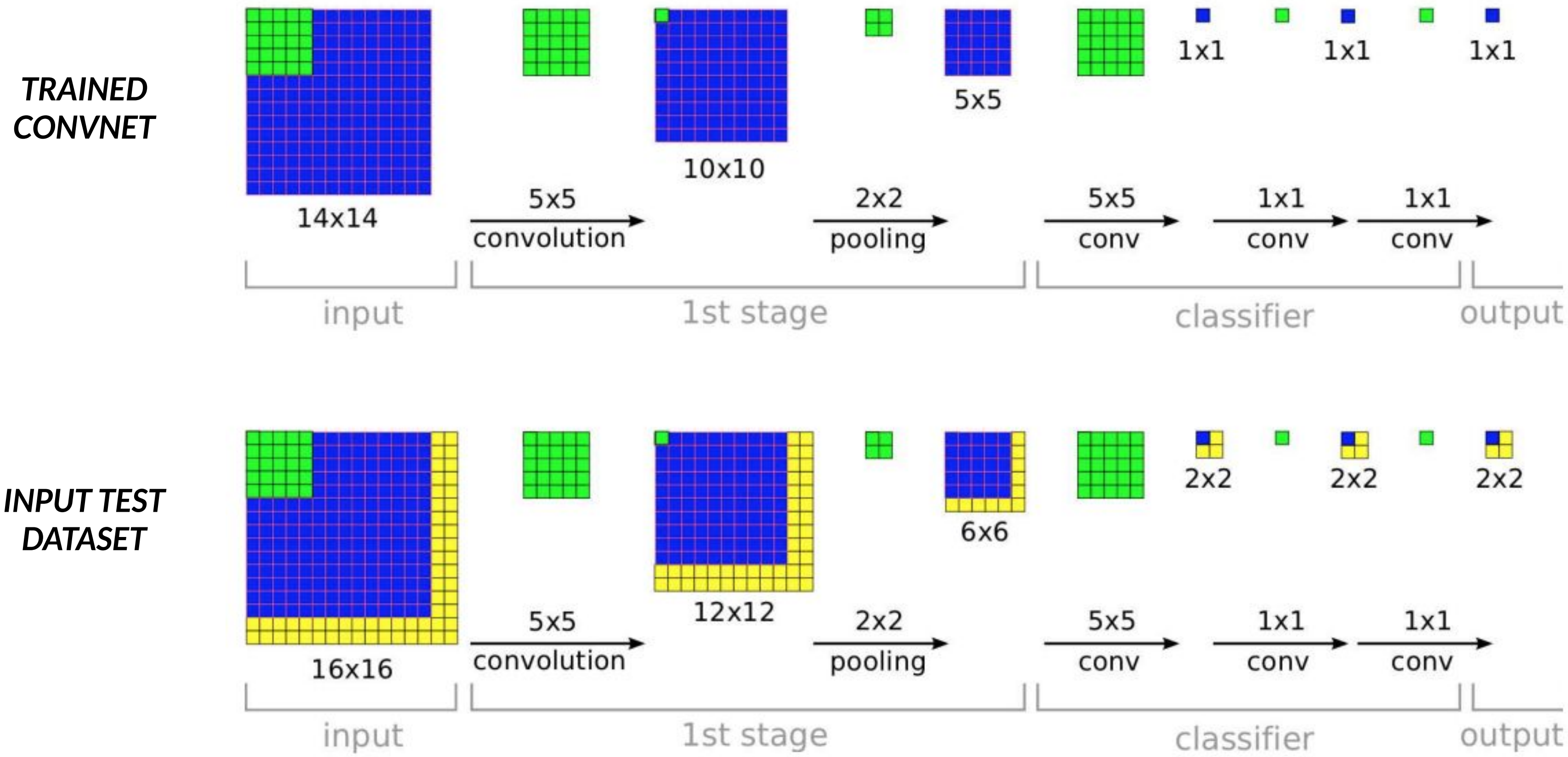


Turning FC Layers into Convolutional Layers



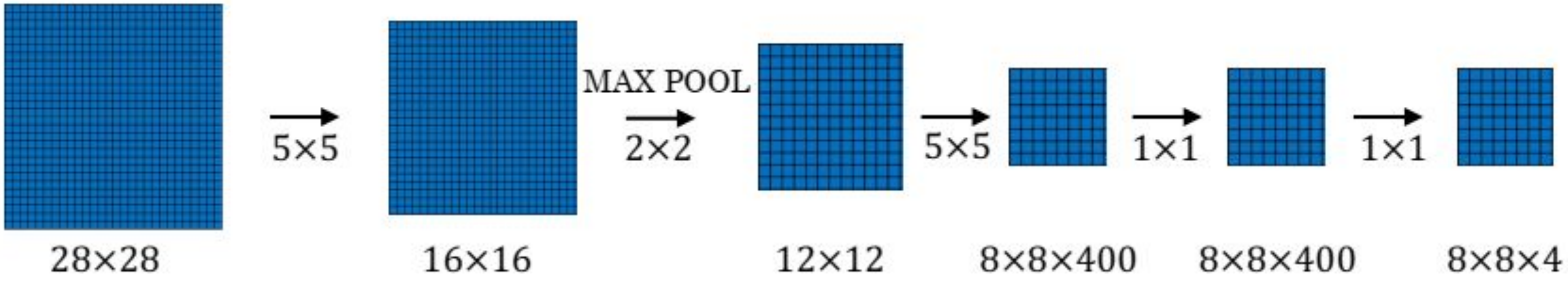
Overfeat

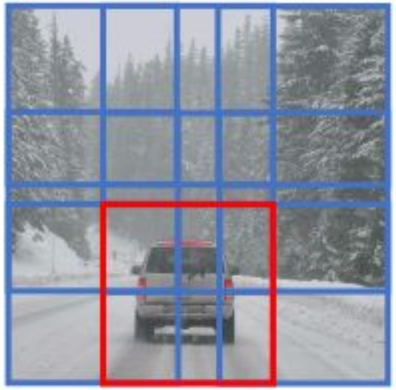
Sermanet, Pierre, et al. "Overfeat: Integrated recognition, localization and detection using convolutional networks." *arXiv preprint arXiv:1312.6229* (2013).



Overfeat

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



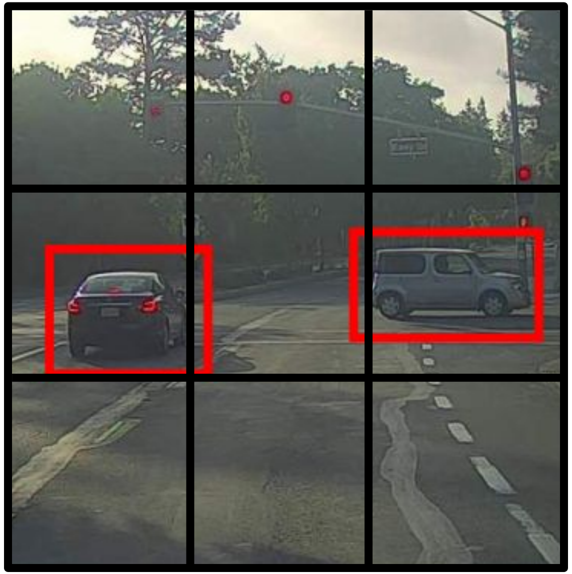


Problem: Accuracy of Bounding Boxes

Improving Bounding Box Predictions - YOLO

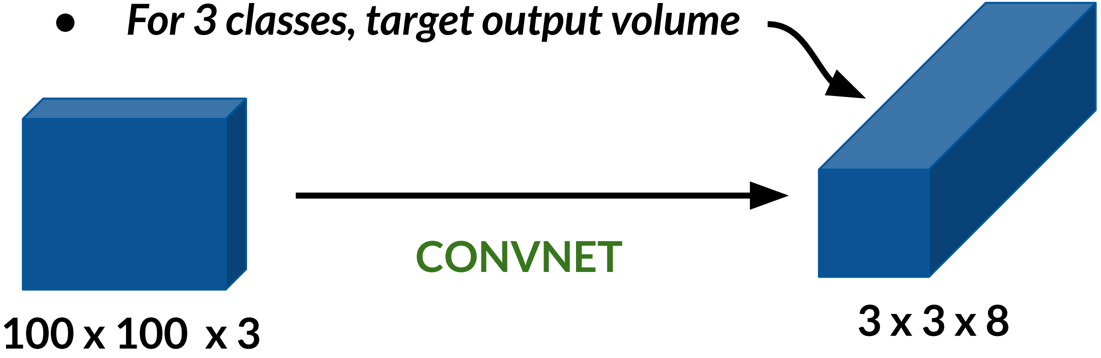
Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

100 x 100

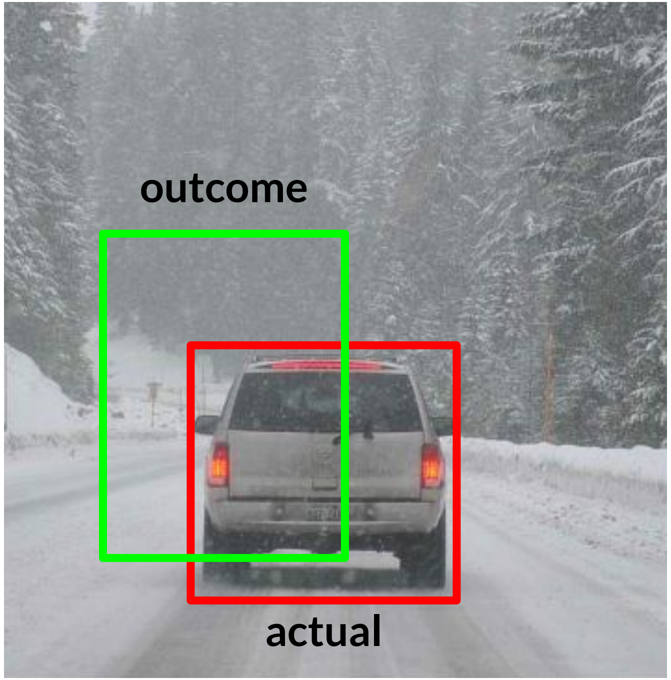


Labels for Training Set

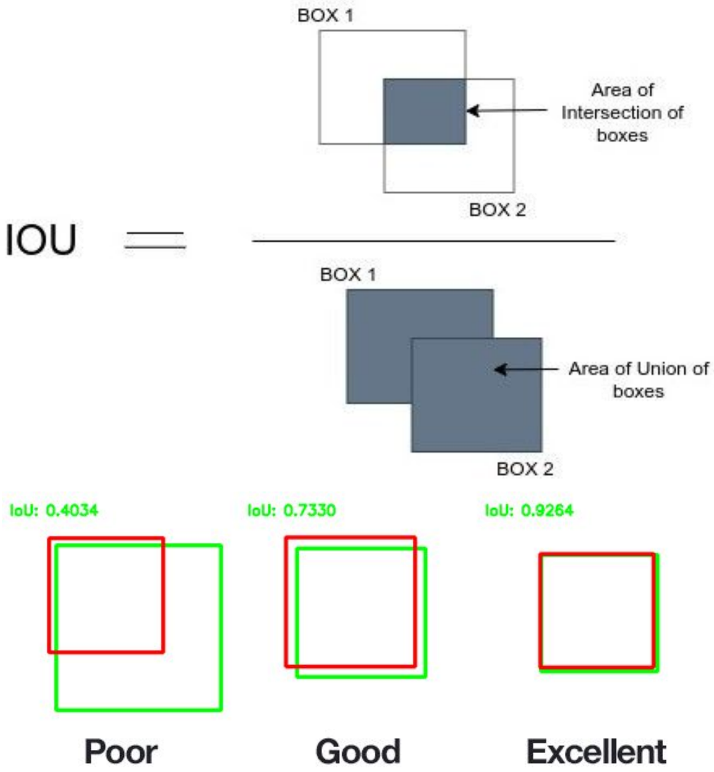
- *A total of 9 grids*
- *Training labels for each grid*
- *Elements in each label vector = 1 + 4 + no. of classes*
- *Objects assigned to single grid cell (centers)*
- *For 3 classes, target output volume*



Evaluating Object Localization - Intersection Over Union

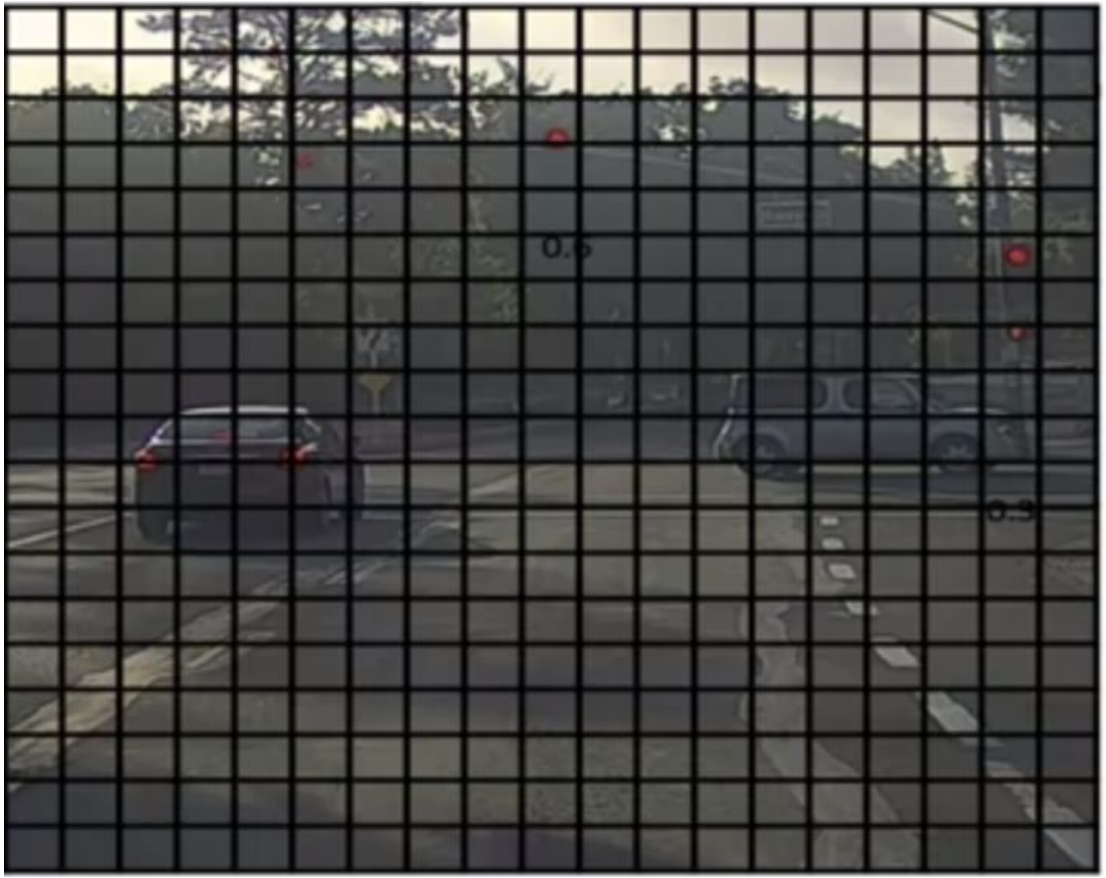


Question: Is this a good or bad outcome?

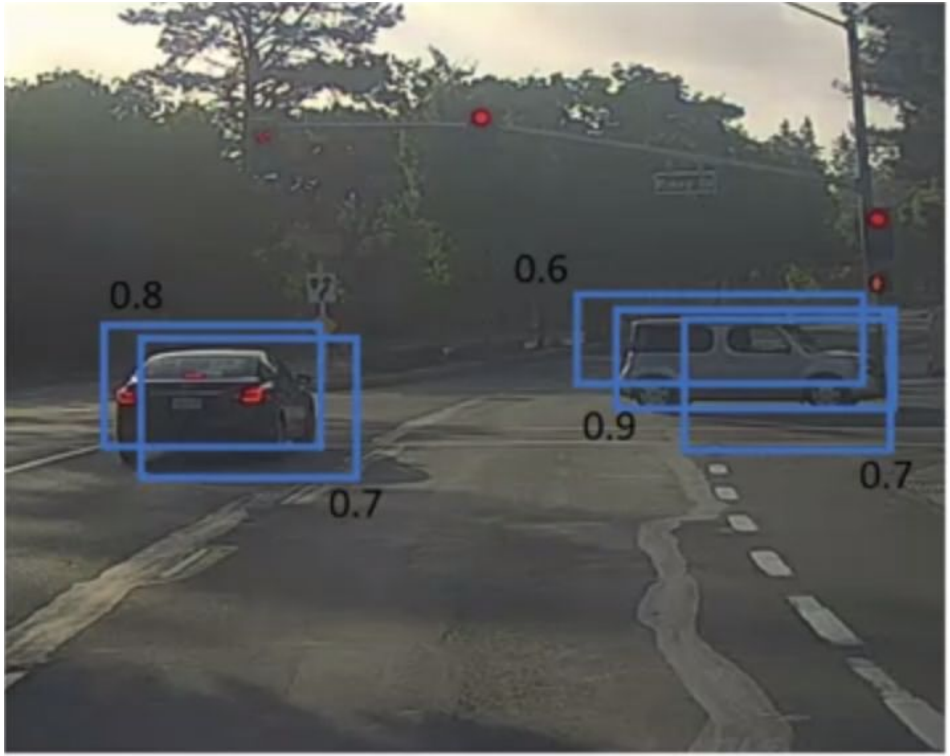


Non-max Suppression

Non-max Suppression



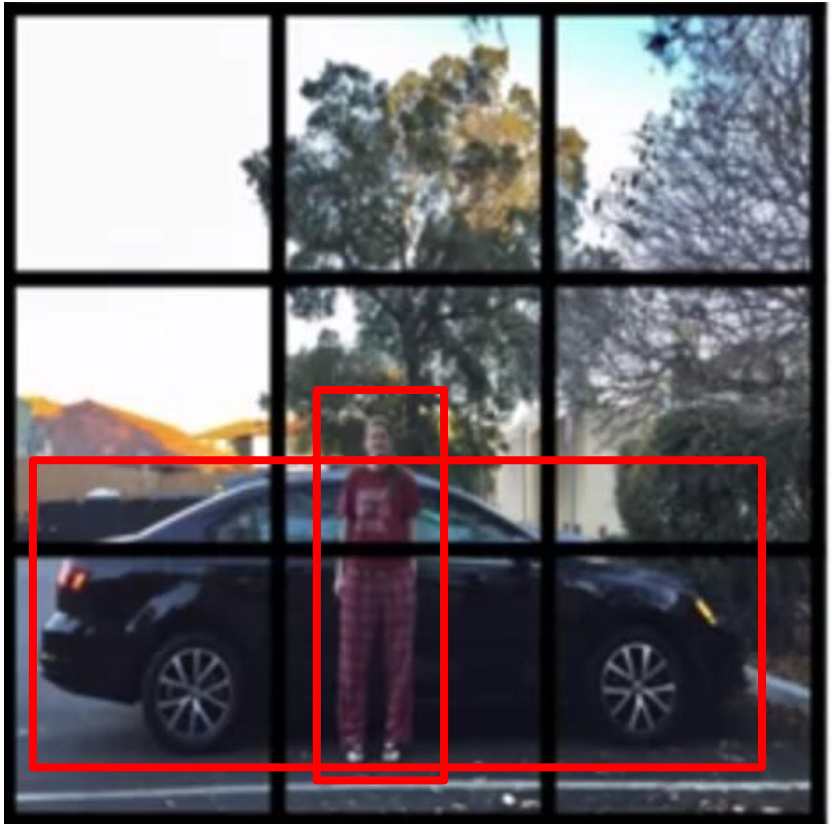
Non-max Suppression



Non-max Algorithm

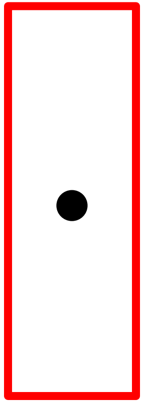
- Go to each detection box
- Look at the probabilities
- Discard thoses with *Pc* < *threshold*
- For remaining boxes:
  - Pick one with largest *Pc*
  - *Output that as prediction*
  - Discard remaining with higher IoUs

Multiple Objects: Anchor Boxes

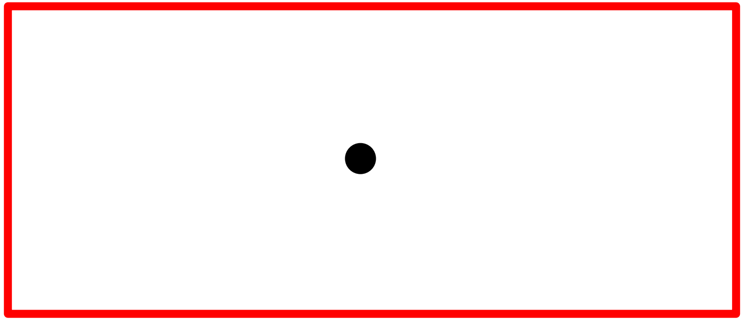


$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

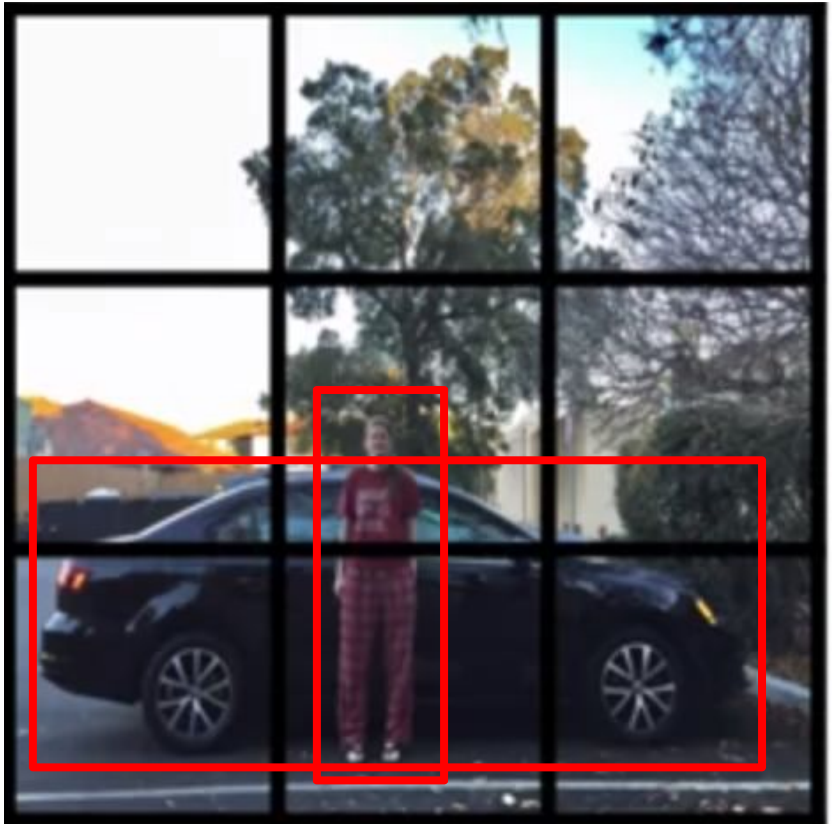
Anchor Box 1



Anchor Box 2



Multiple Objects: Anchor Boxes



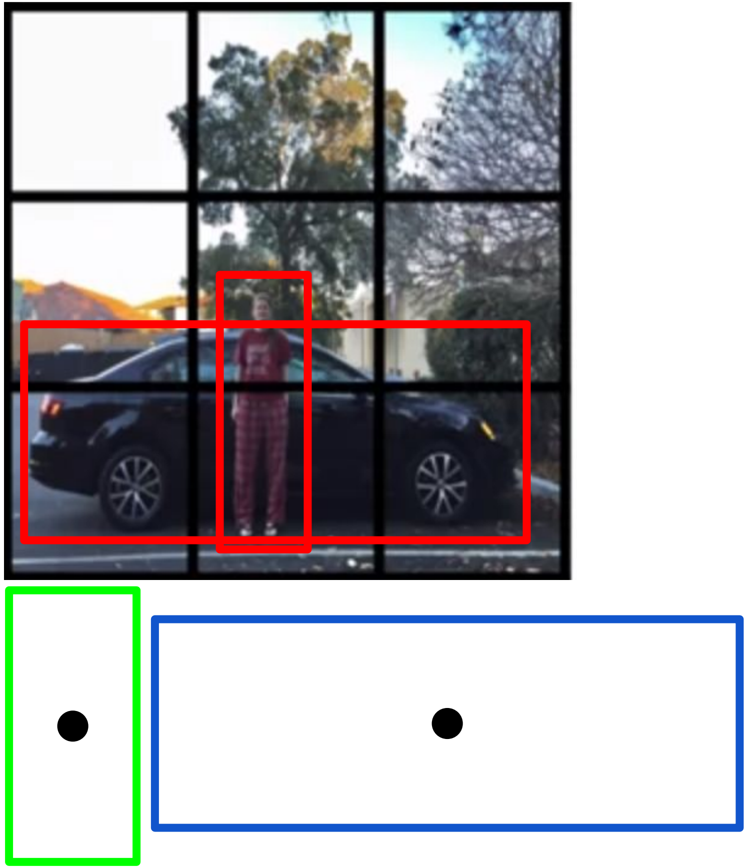
Previously (3 x 3 x 8)

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



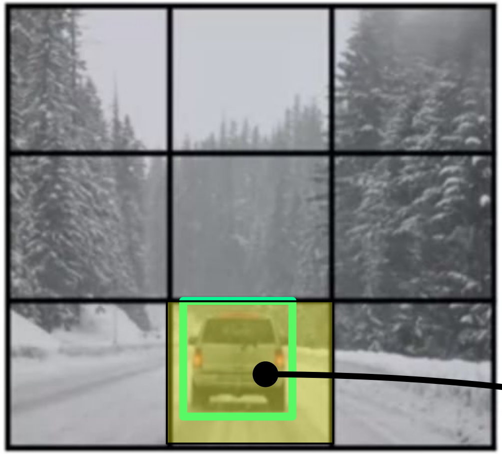
$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \\ p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

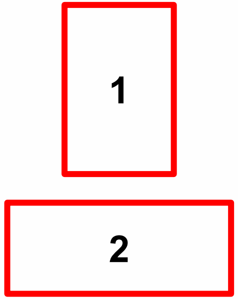
YOLO Example

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





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

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

<u>For 3 Classes; Two Anchor Boxes</u>

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*

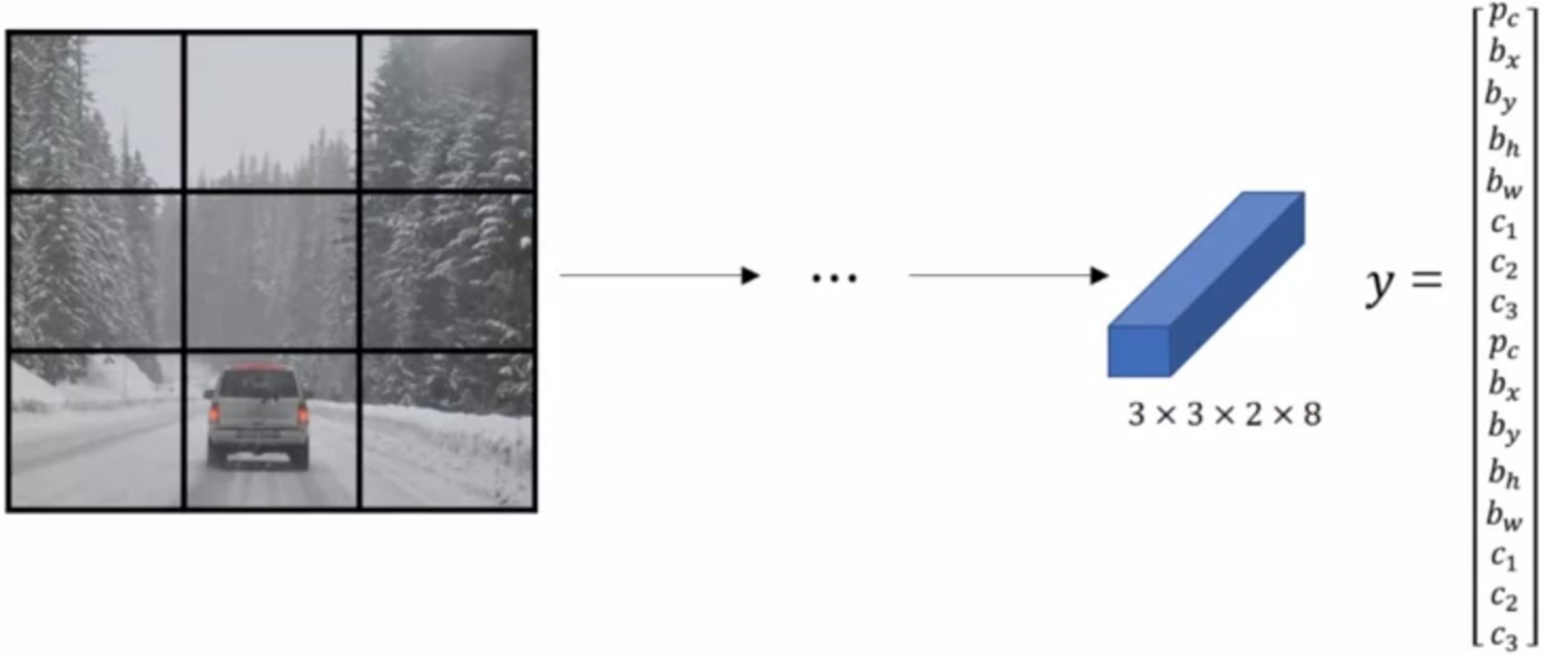
$$\begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

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

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

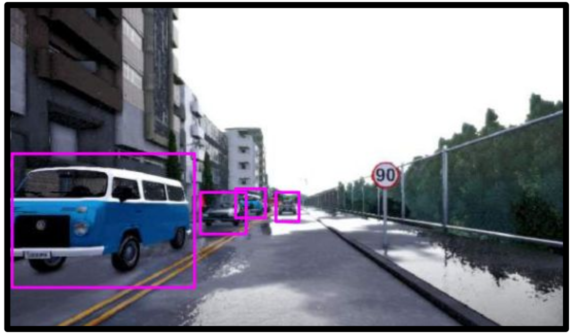


Semantic Segmentation

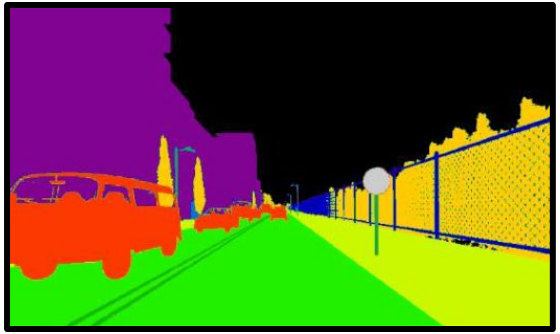
General Problem

Input Image

Object Detection

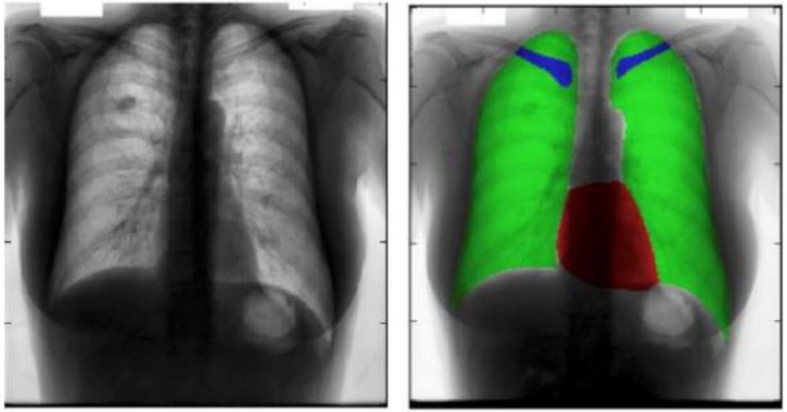


Semantic Segmentation

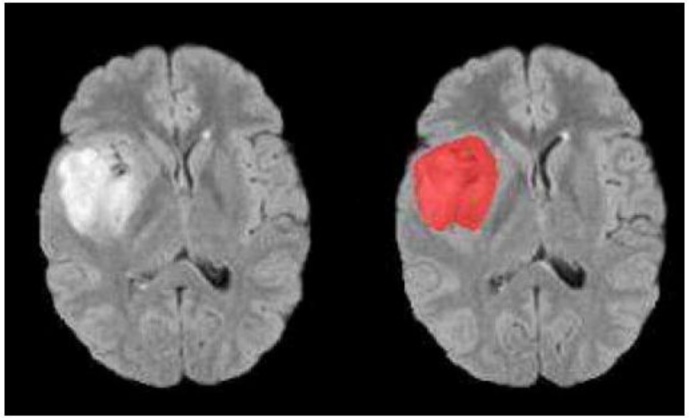


U-Net - Original Motivation

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.

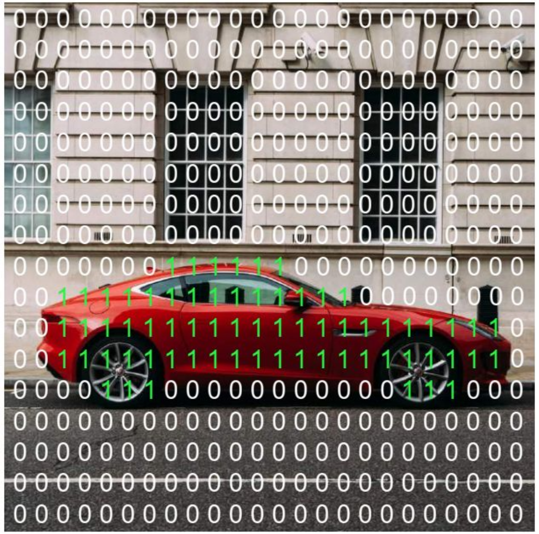


Chest X-Ray



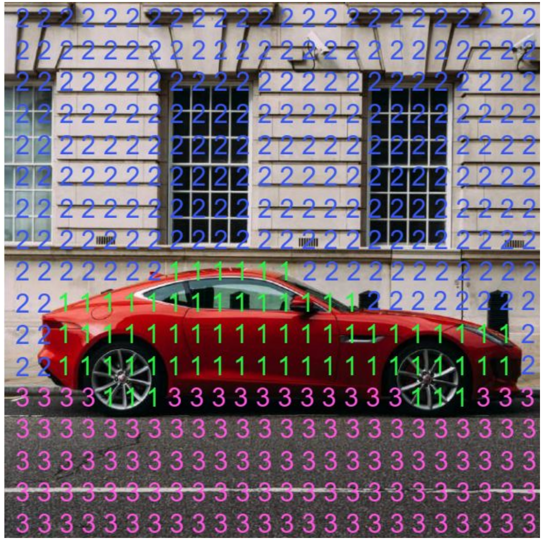
Brain MRI

Per Pixel Class Labels



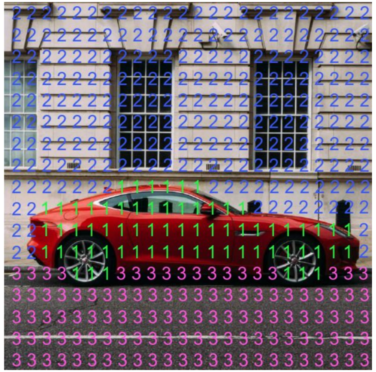
0. *Car*
1. *Not Car*

Per Pixel Class Labels



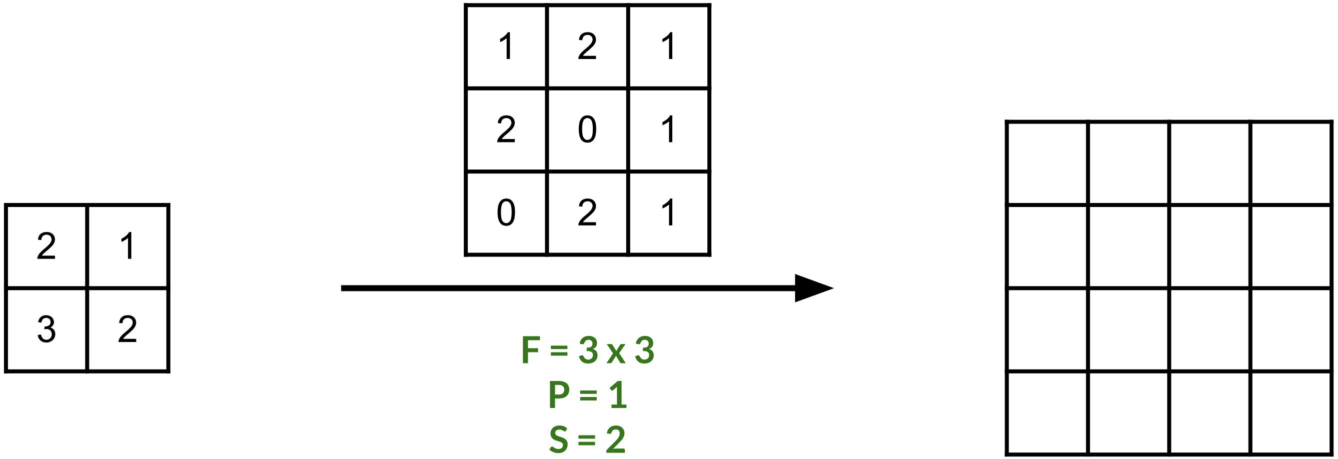
1. *Car*
2. *Building*
3. *Road*

Segmentation Map

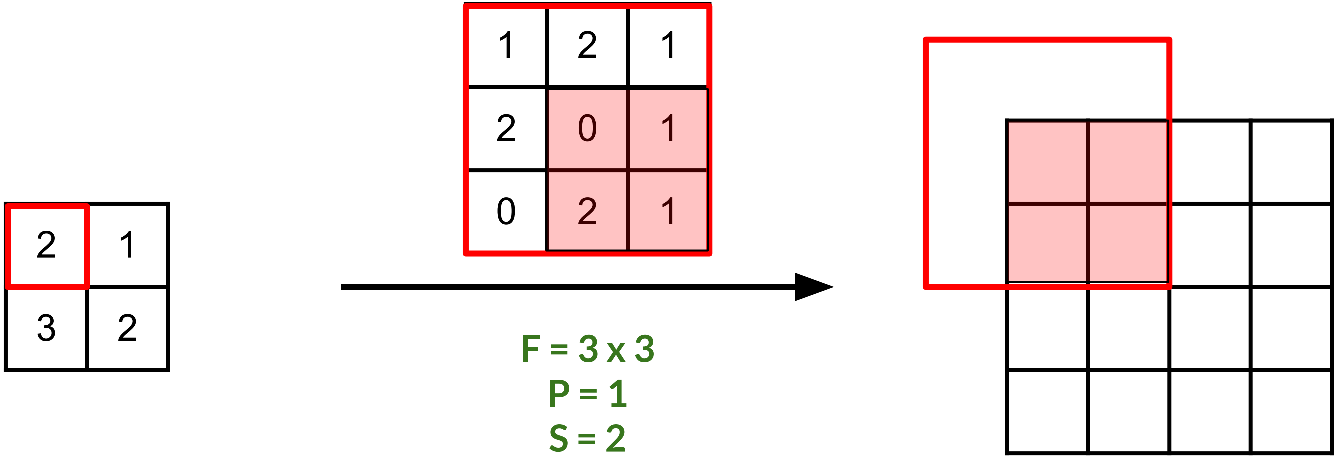


2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
3	3	3	3	1	1	1	3	3	3	3	3	3	3	3	3	3	3	1	1	1	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3

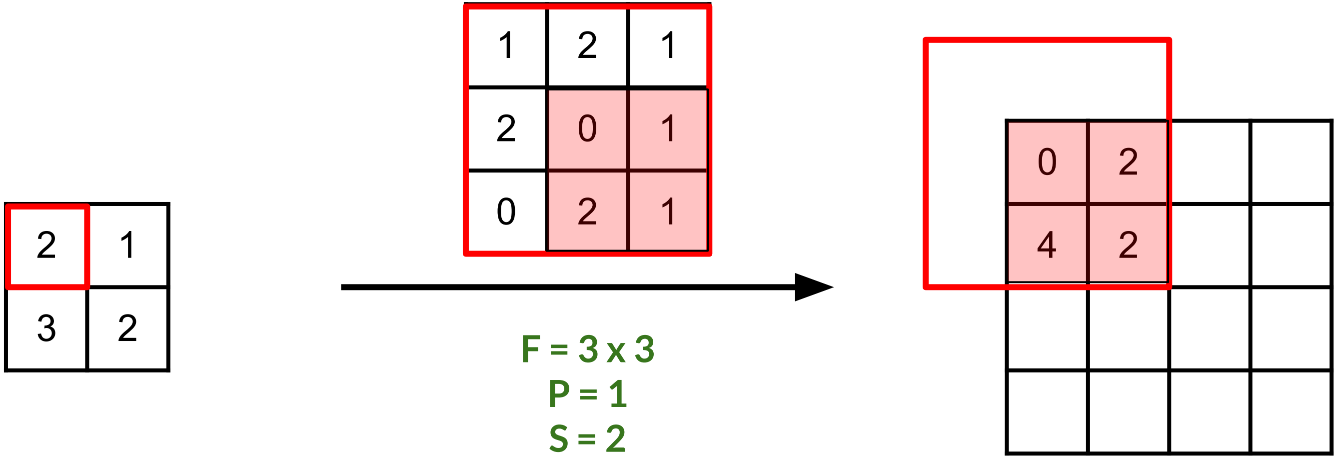
Segmentation Map



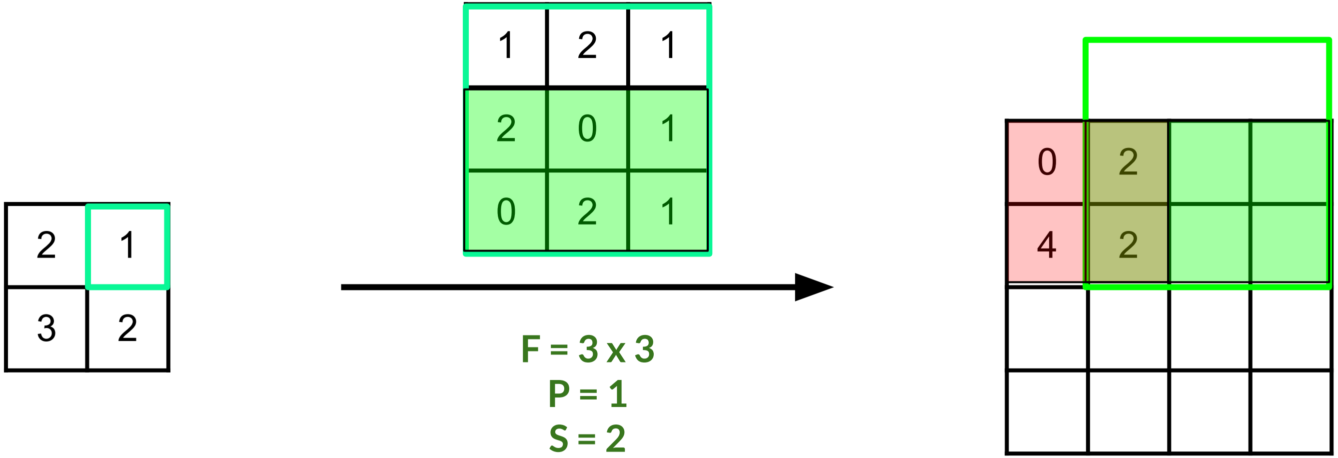
Segmentation Map



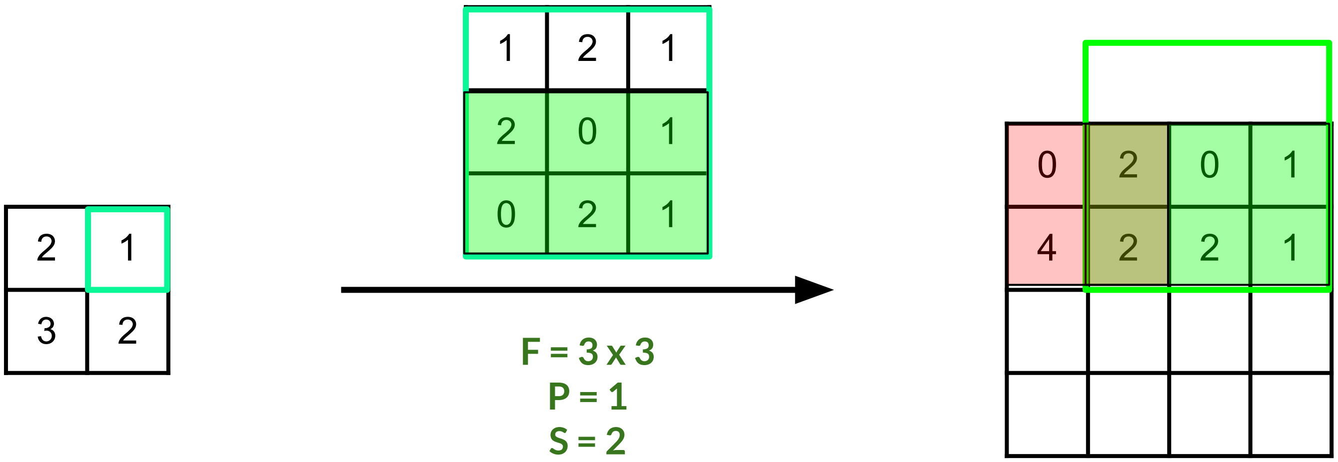
Segmentation Map



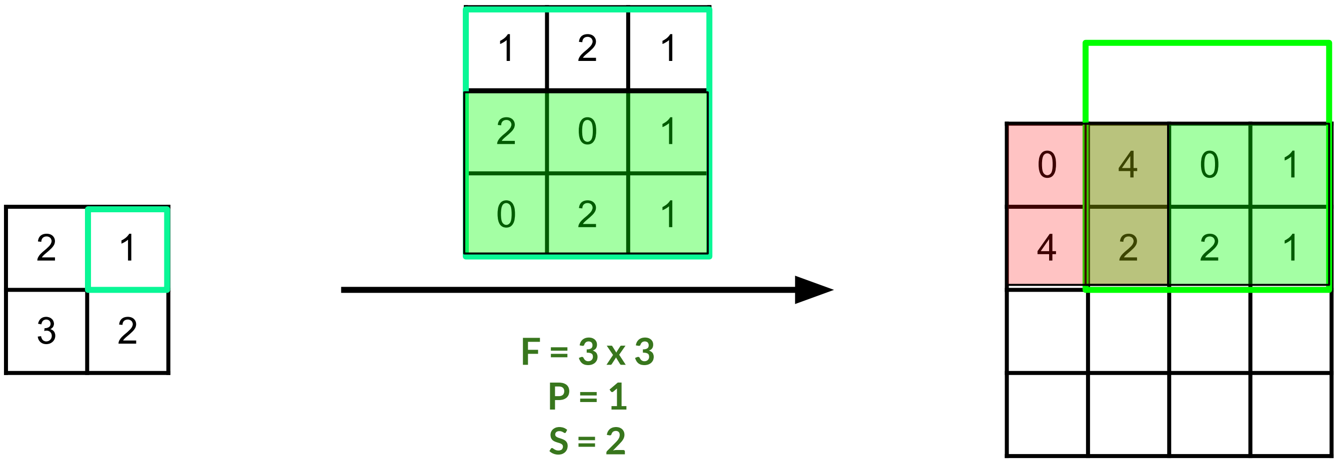
Segmentation Map



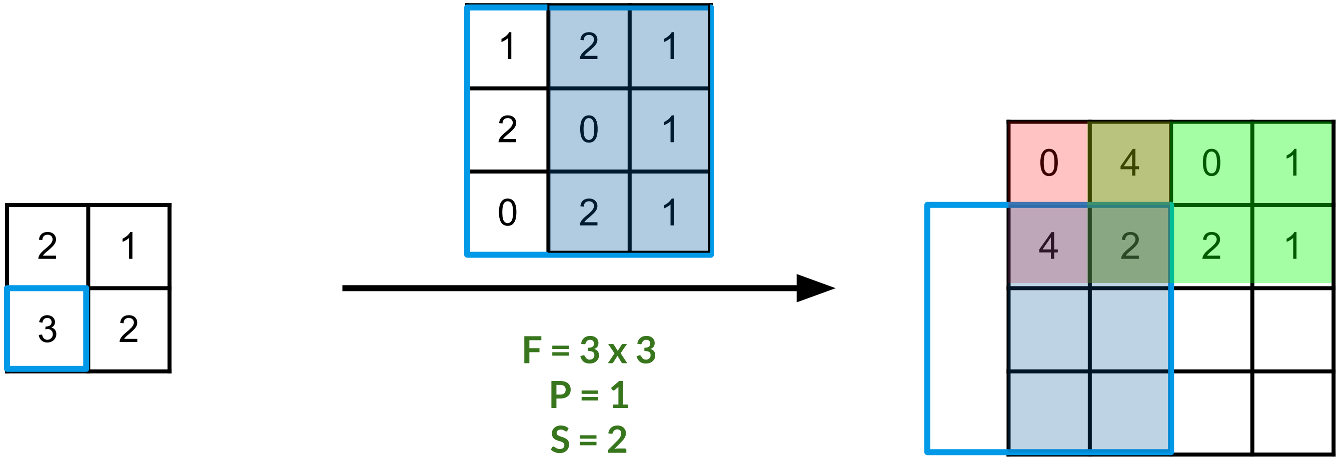
Segmentation Map



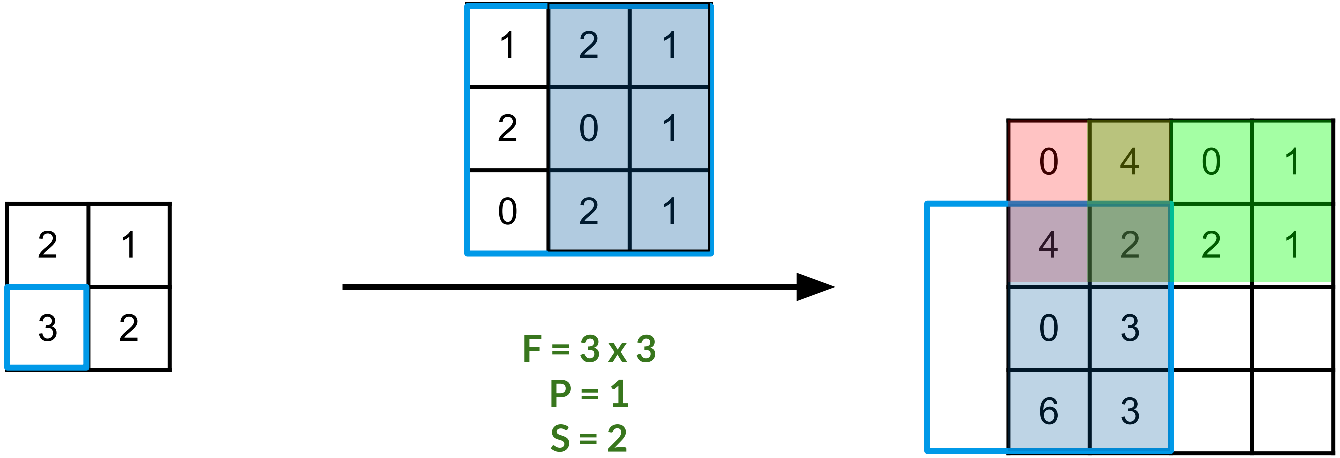
Segmentation Map



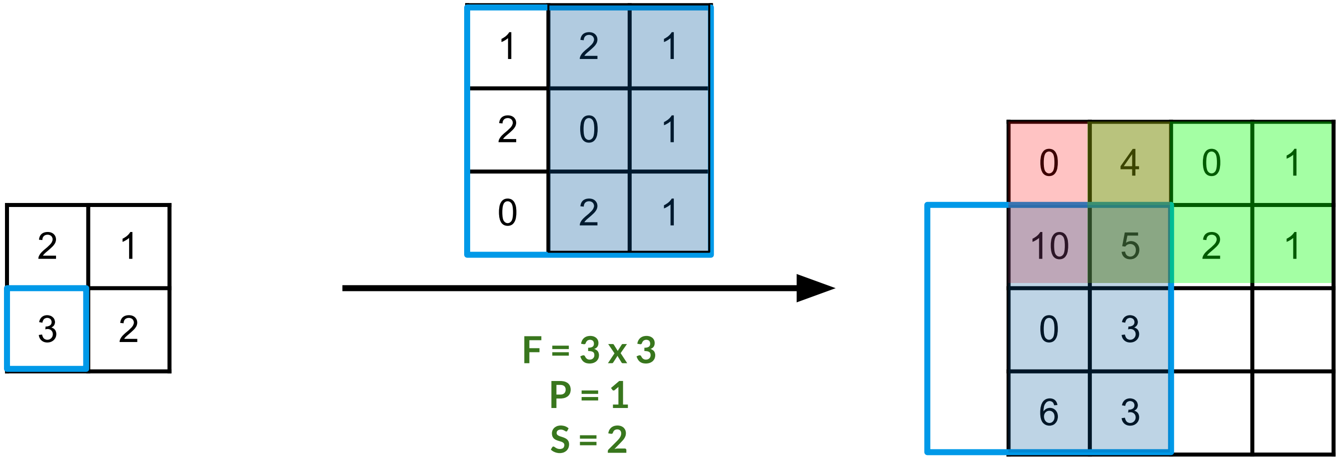
Segmentation Map



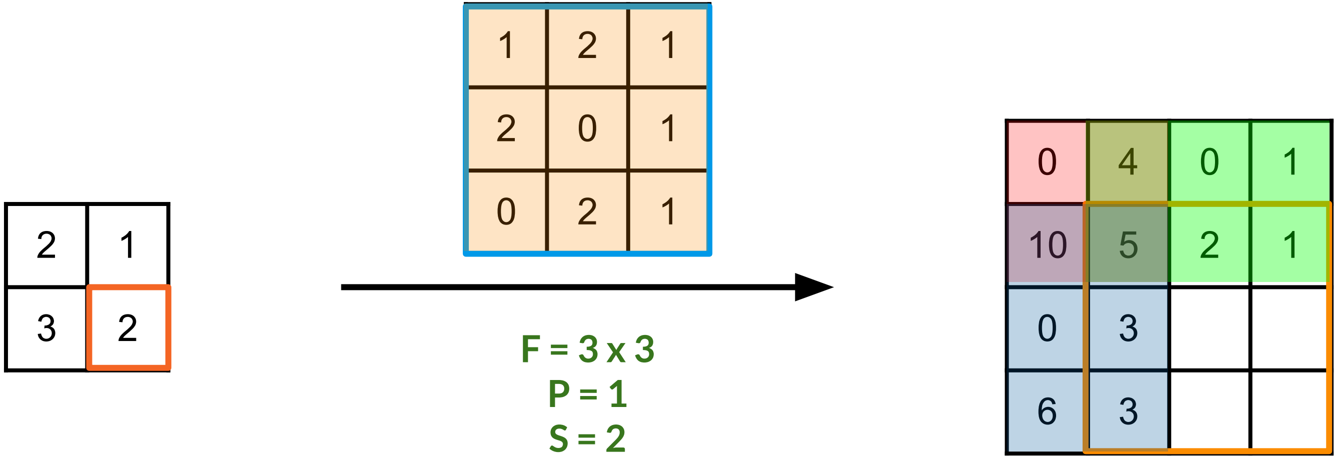
Segmentation Map



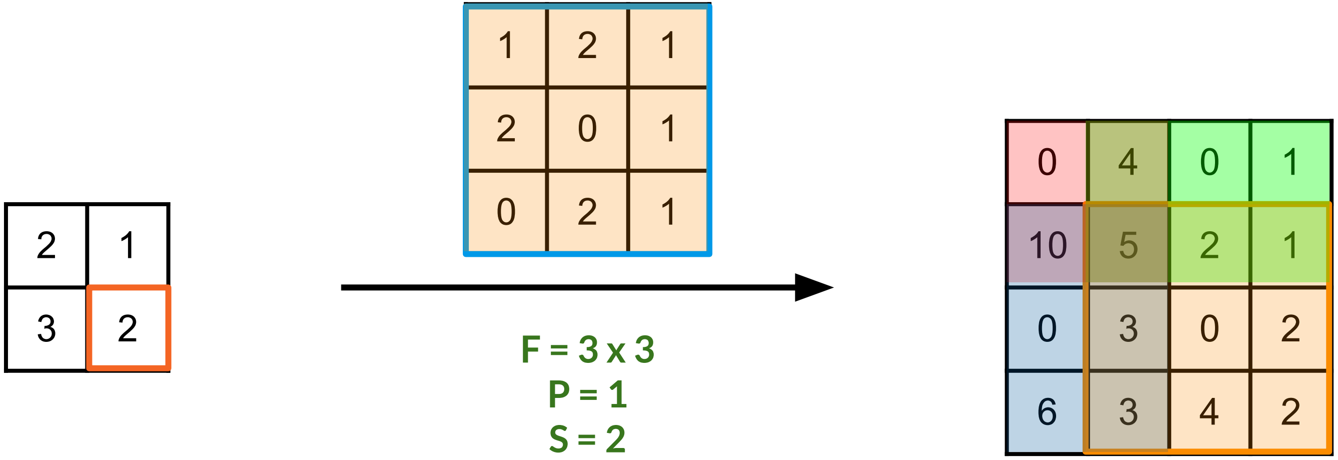
Segmentation Map



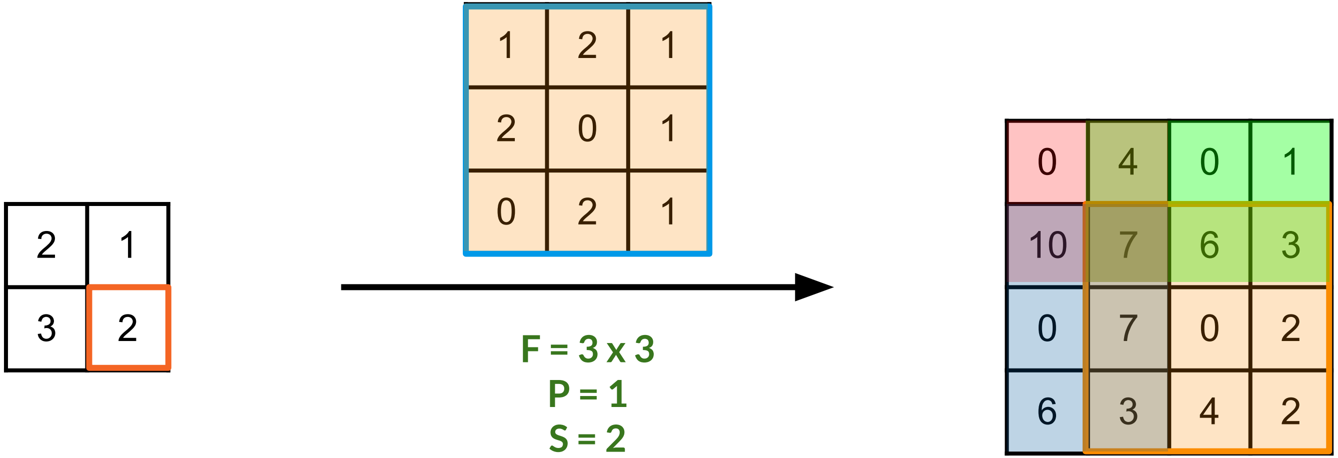
Segmentation Map



Segmentation Map



Segmentation Map



Segmentation Map

$$\begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix} \xrightarrow[\substack{F = 3 \times 3 \\ P = 1 \\ S = 2}]{\begin{bmatrix} 1 & 2 & 1 \\ 2 & 0 & 1 \\ 0 & 2 & 1 \end{bmatrix}} \begin{bmatrix} 0 & 4 & 0 & 1 \\ 10 & 7 & 6 & 3 \\ 0 & 7 & 0 & 2 \\ 6 & 3 & 4 & 2 \end{bmatrix}$$

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.