Practical Machine Learning and Deep Learning

Vladimir Ivanov

WEEK 2

Quiz

- a paper-based quiz, 5-7 min.
- Q1: What is i.i.d.? How the concept used in statistics?
- Q2: Why this concept is important in Machine Learning? (explain and provide an example).

Objectives (for today)

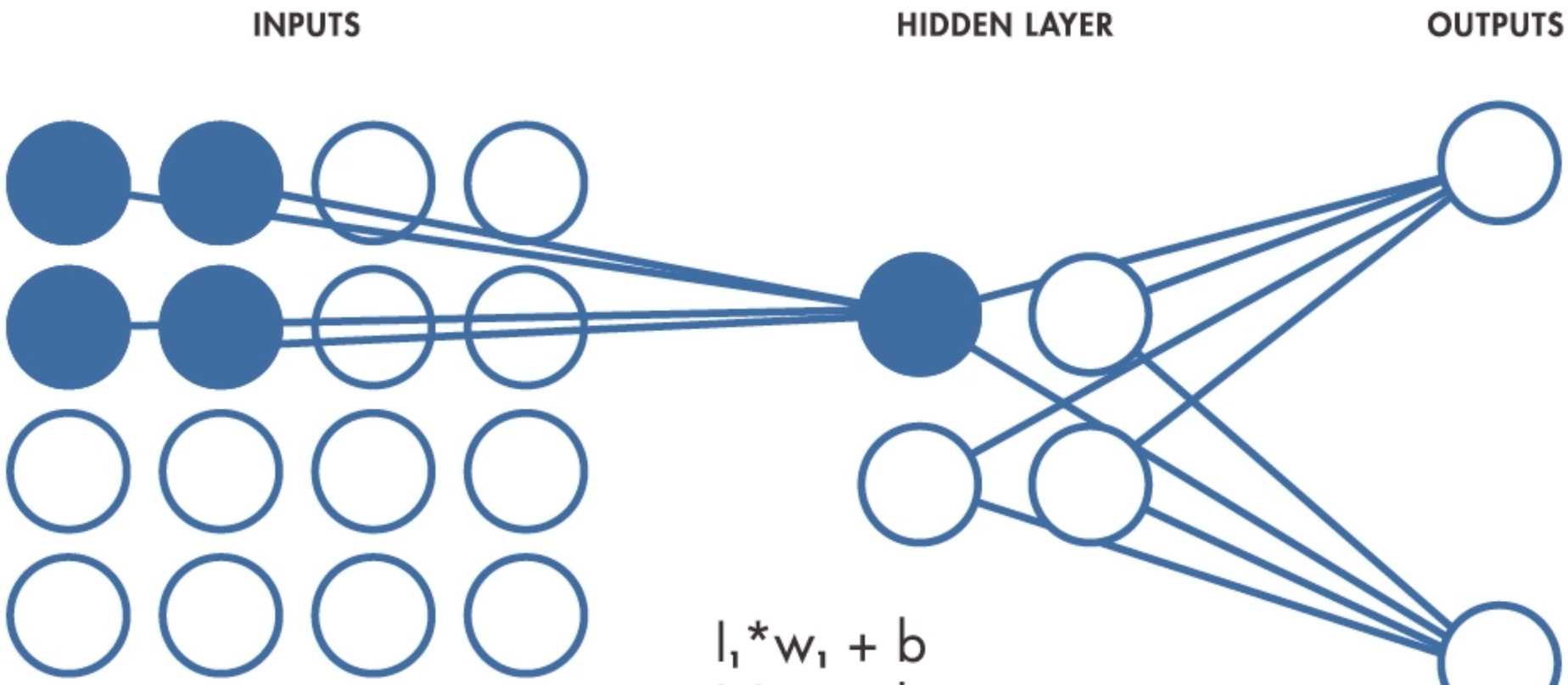
- to recall CNNs in Computer Vision
- to review object detection
- to understand style transfer
- CNNs in other domains

Computer Vision...

just go to:

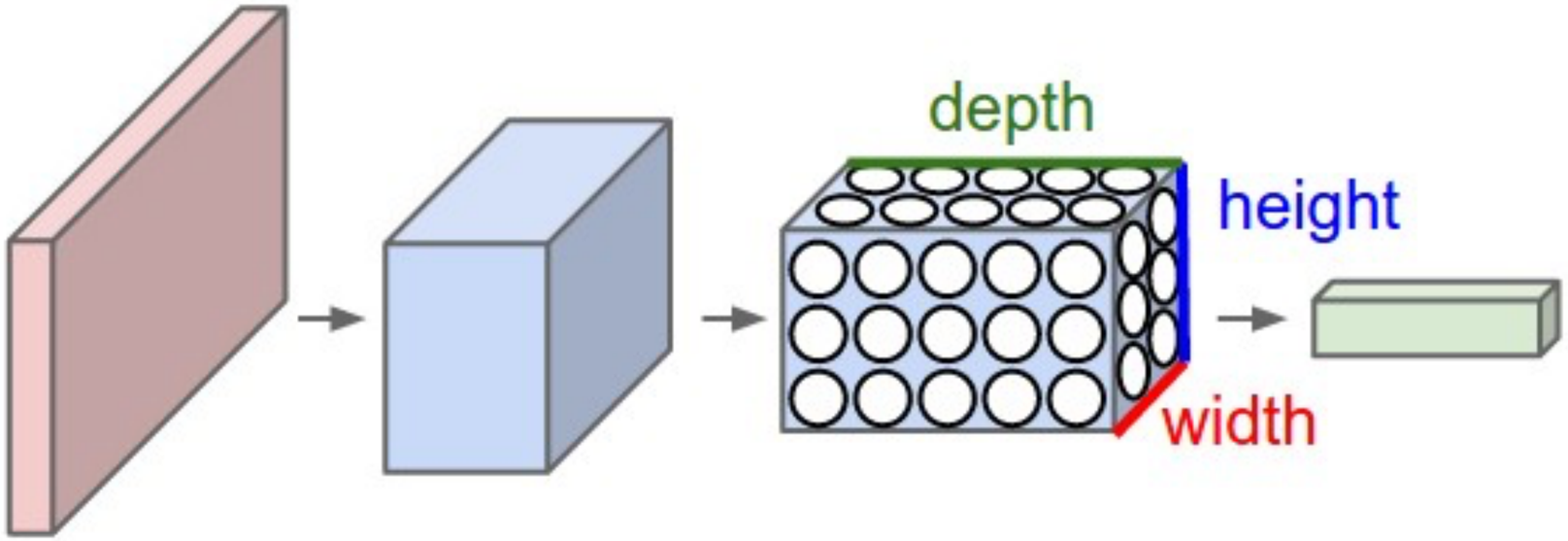
* https://paperswithcode.com/area/computer-vision

Review of ConvNets

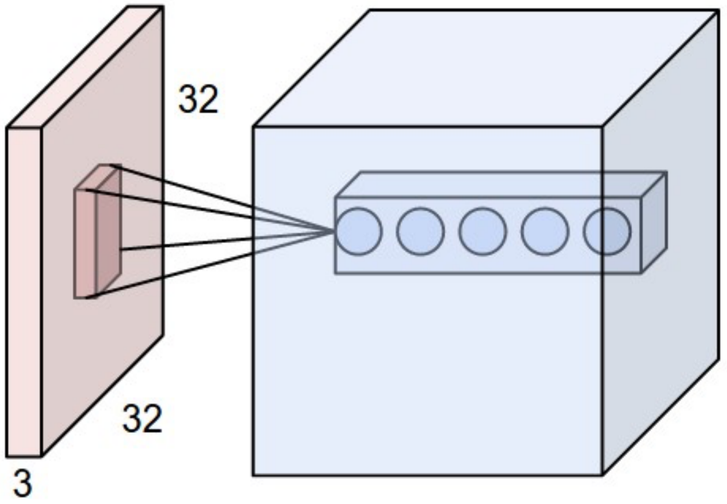


$$I_1 * W_1 + b$$
$$I_2 * W_2 + b$$
$$I_3 * W_3 + b$$
$$I_4 * W_4 + b$$

Review of ConvNets

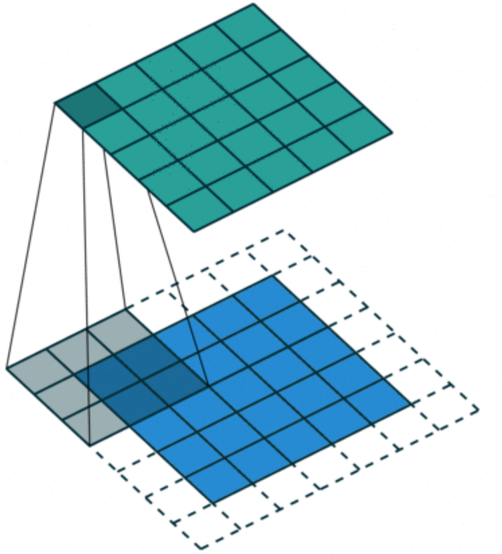


Review of ConvNets



1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

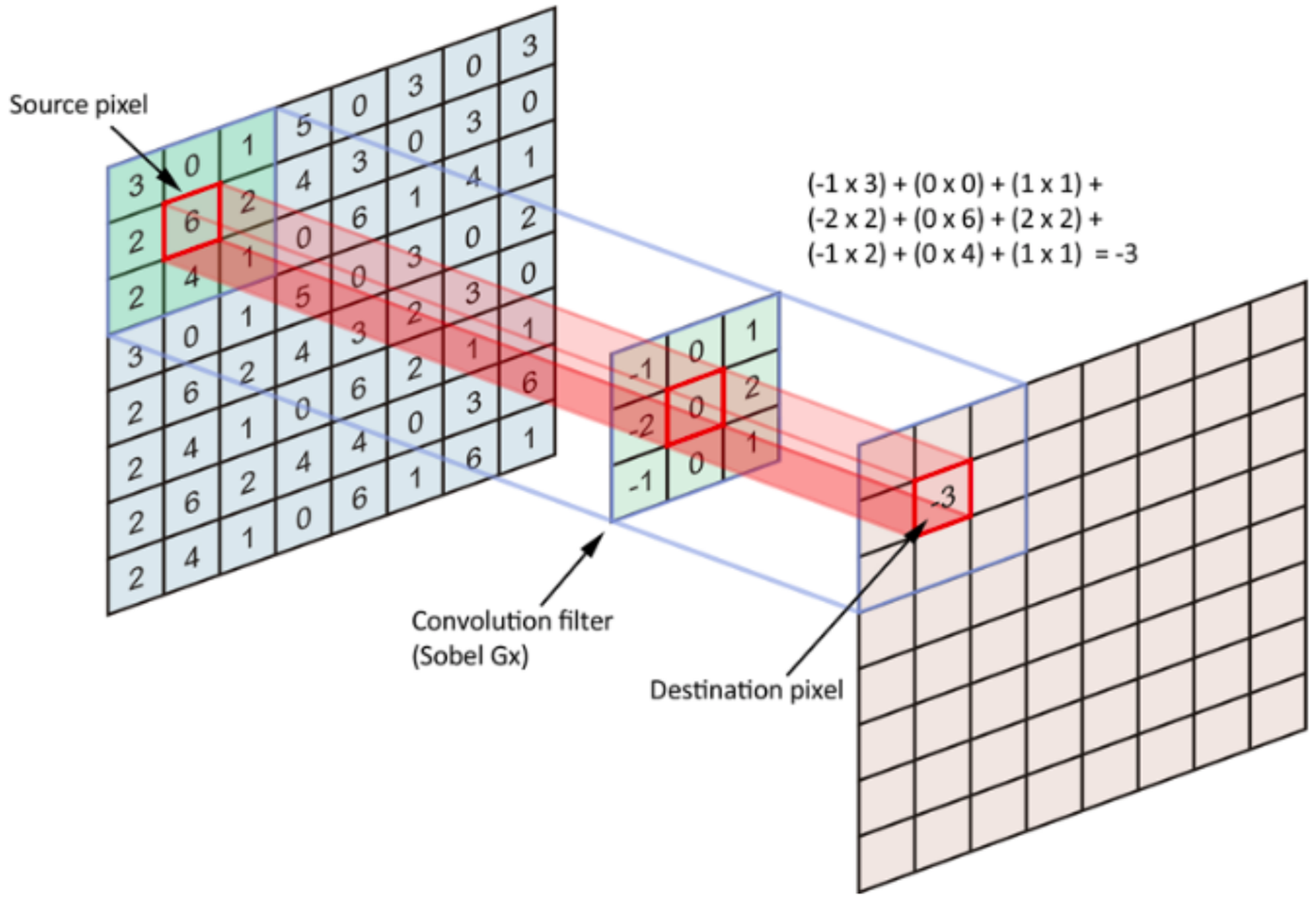
4		



1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

4		

Review of ConvNets



Review of ConvNets

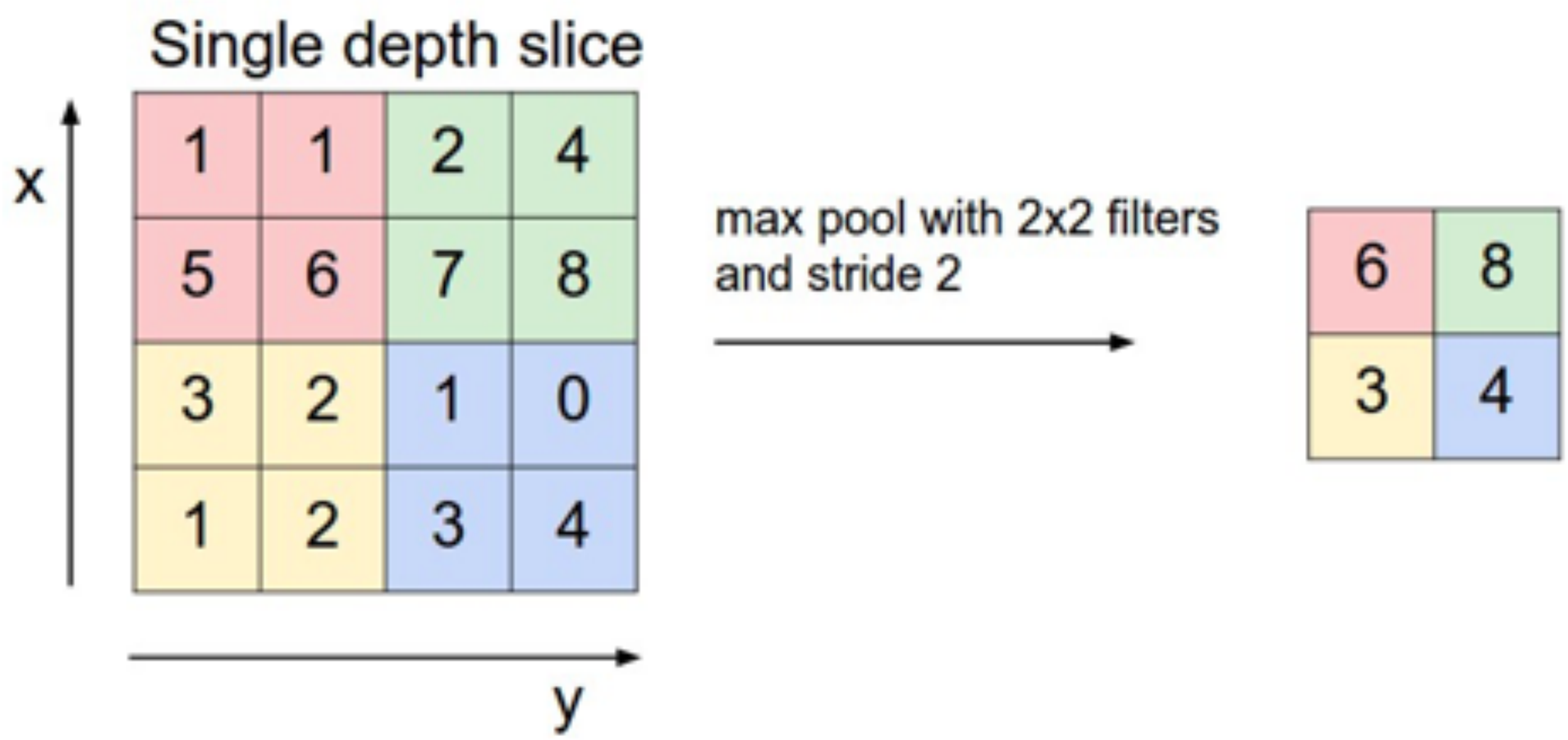
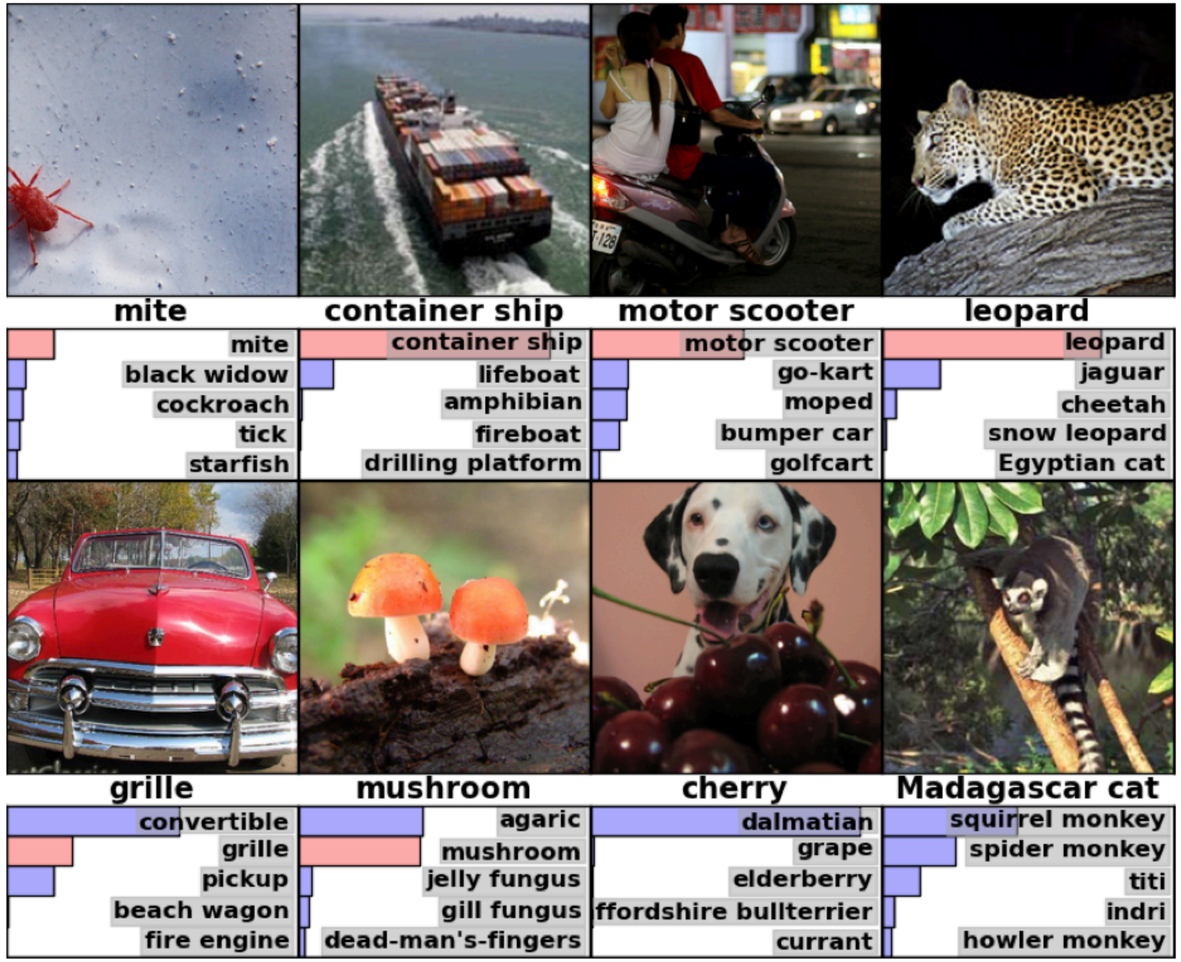
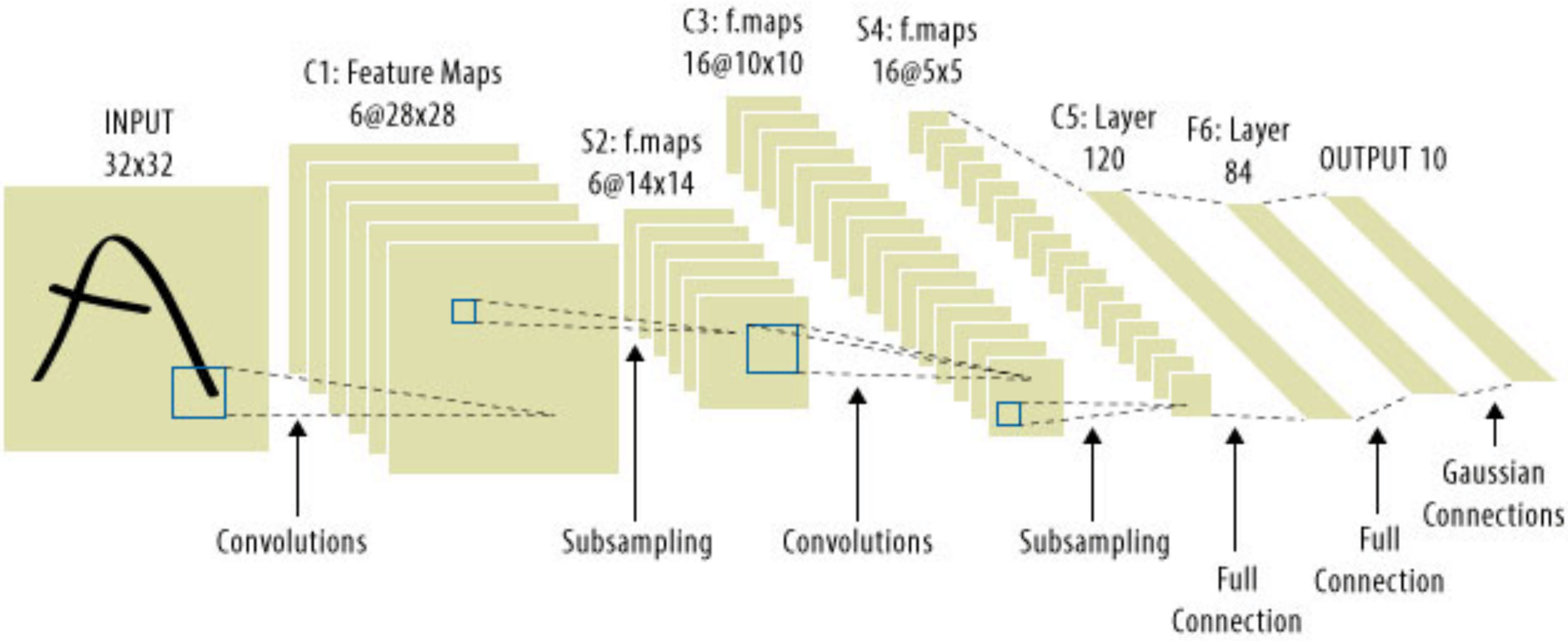


Image Classification

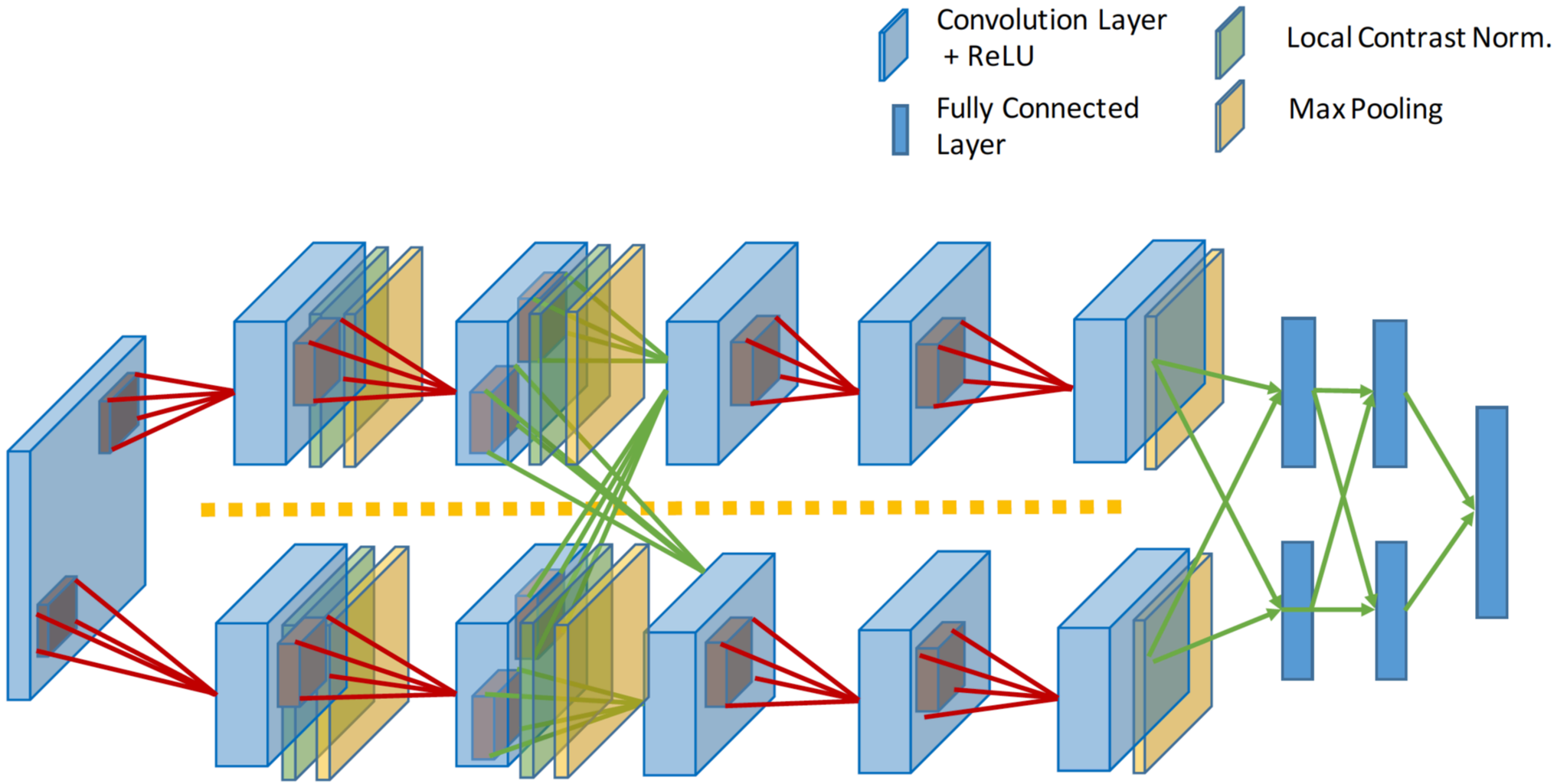
- LeNet-5
- AlexNet
- GoogLeNet
- VGG
- ResNet
- Inception



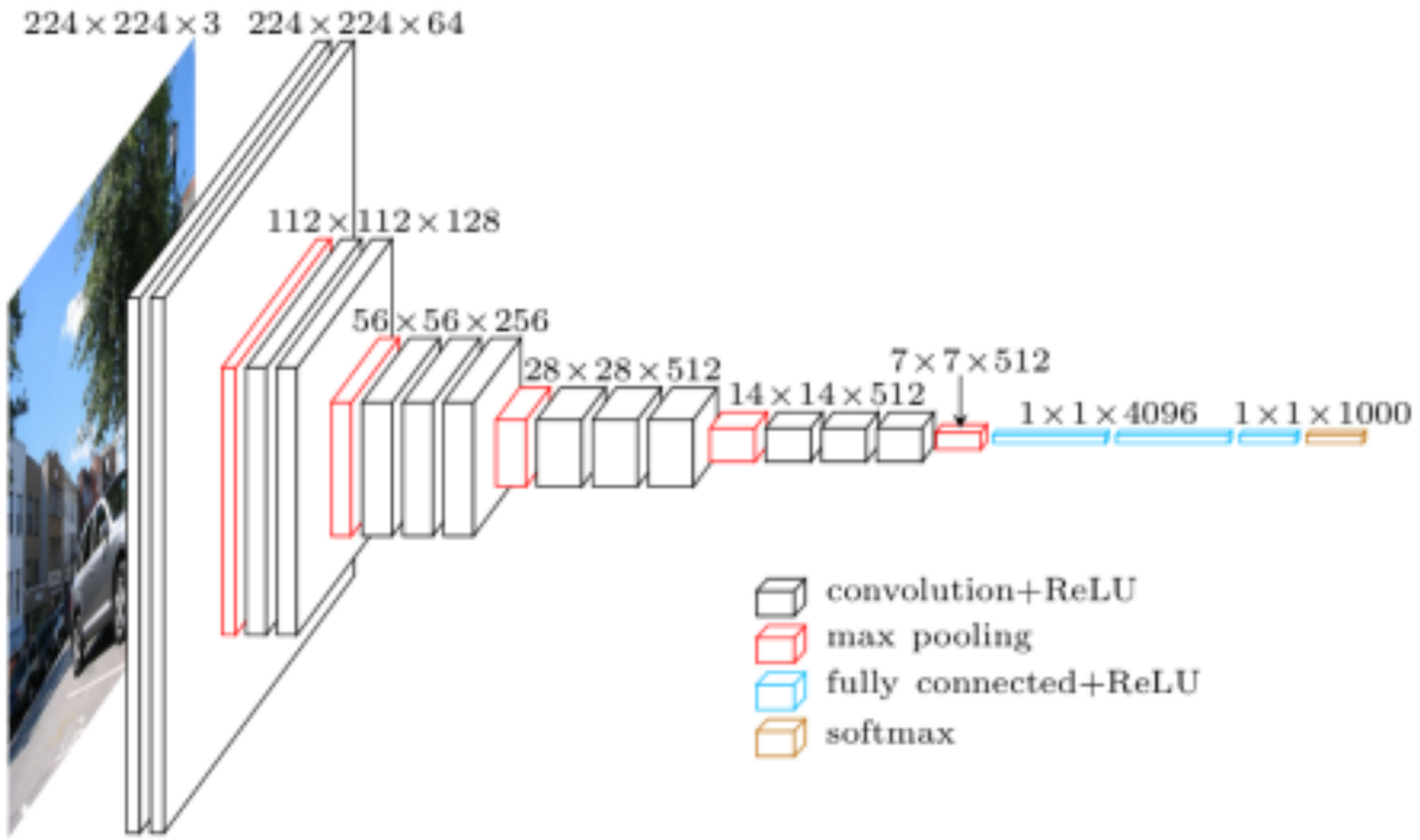
Review of ConvNets



Review of ConvNets

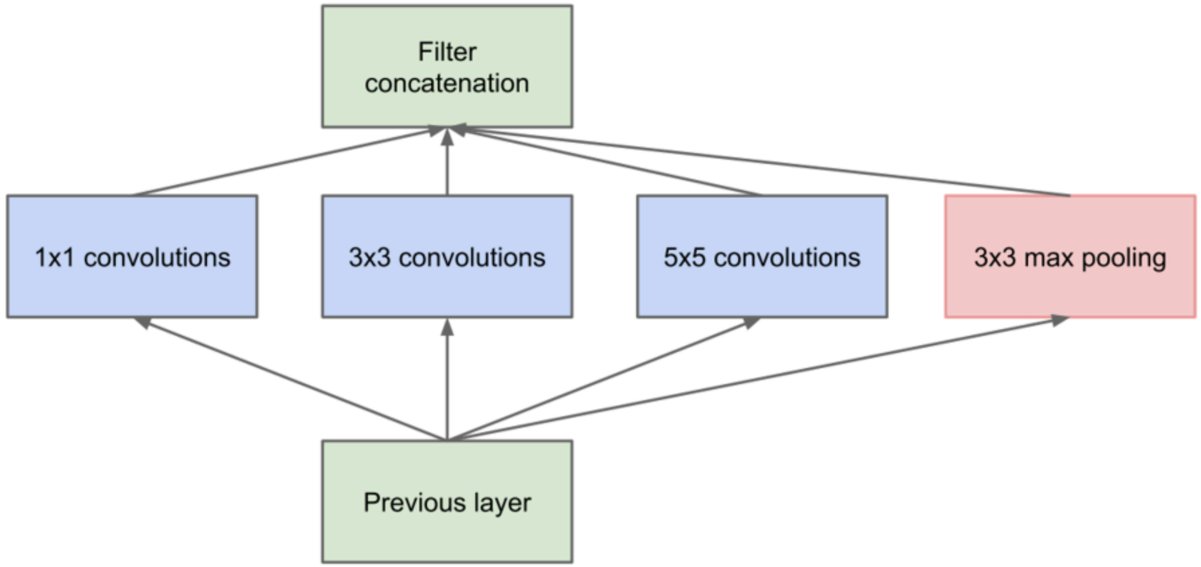


Review of ConvNets

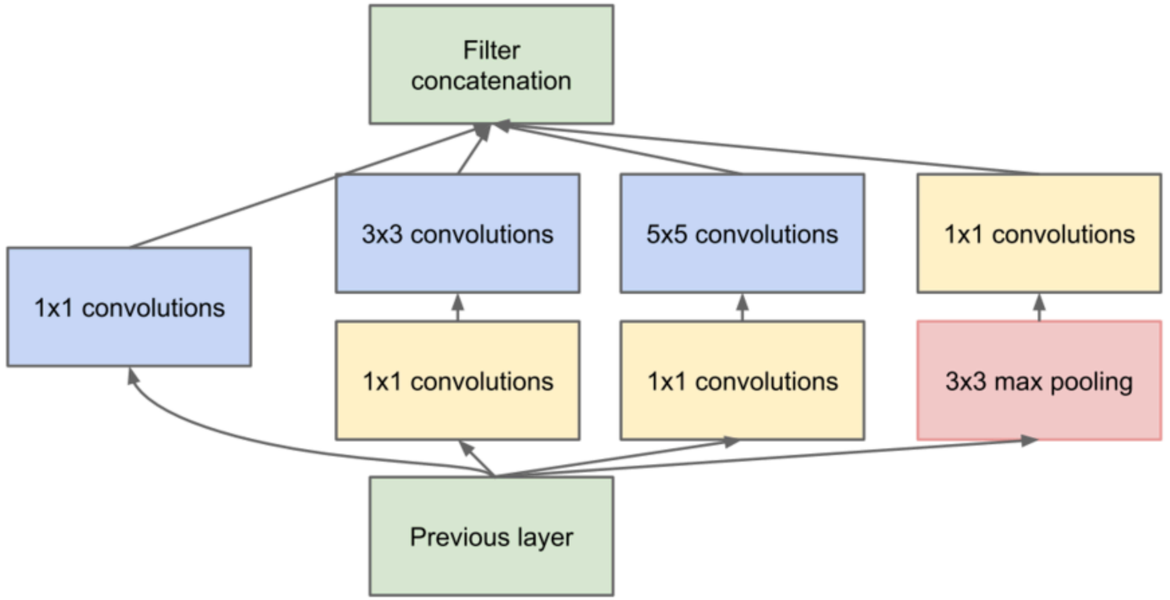


* http://ethereon.github.io/netscope/#/preset/vgg-16

Review of ConvNets



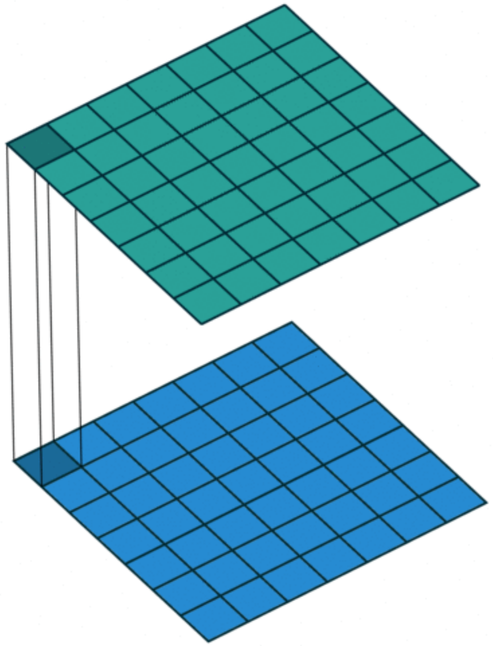
(a) Inception module, naïve version



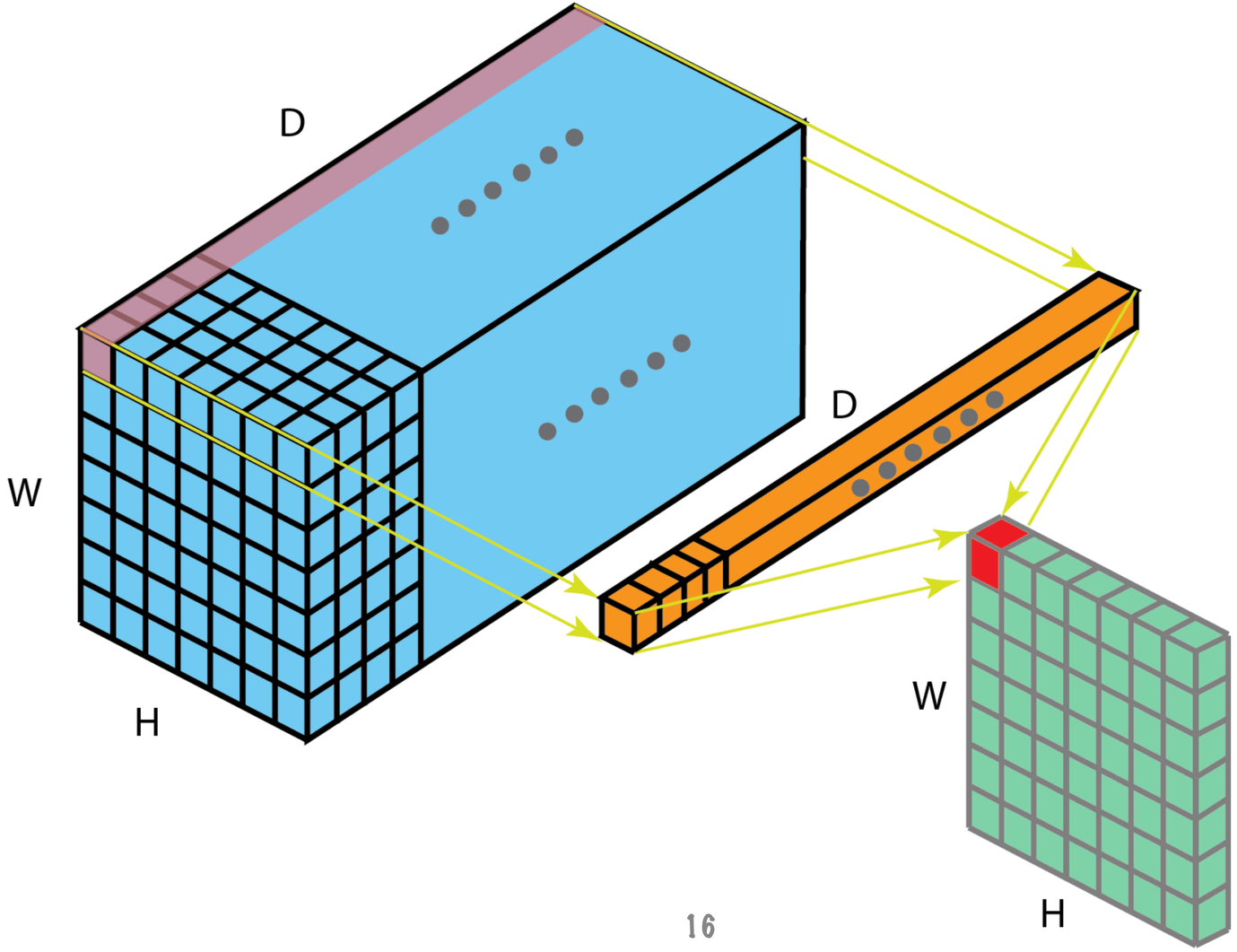
(b) Inception module with dimension reductions

Question

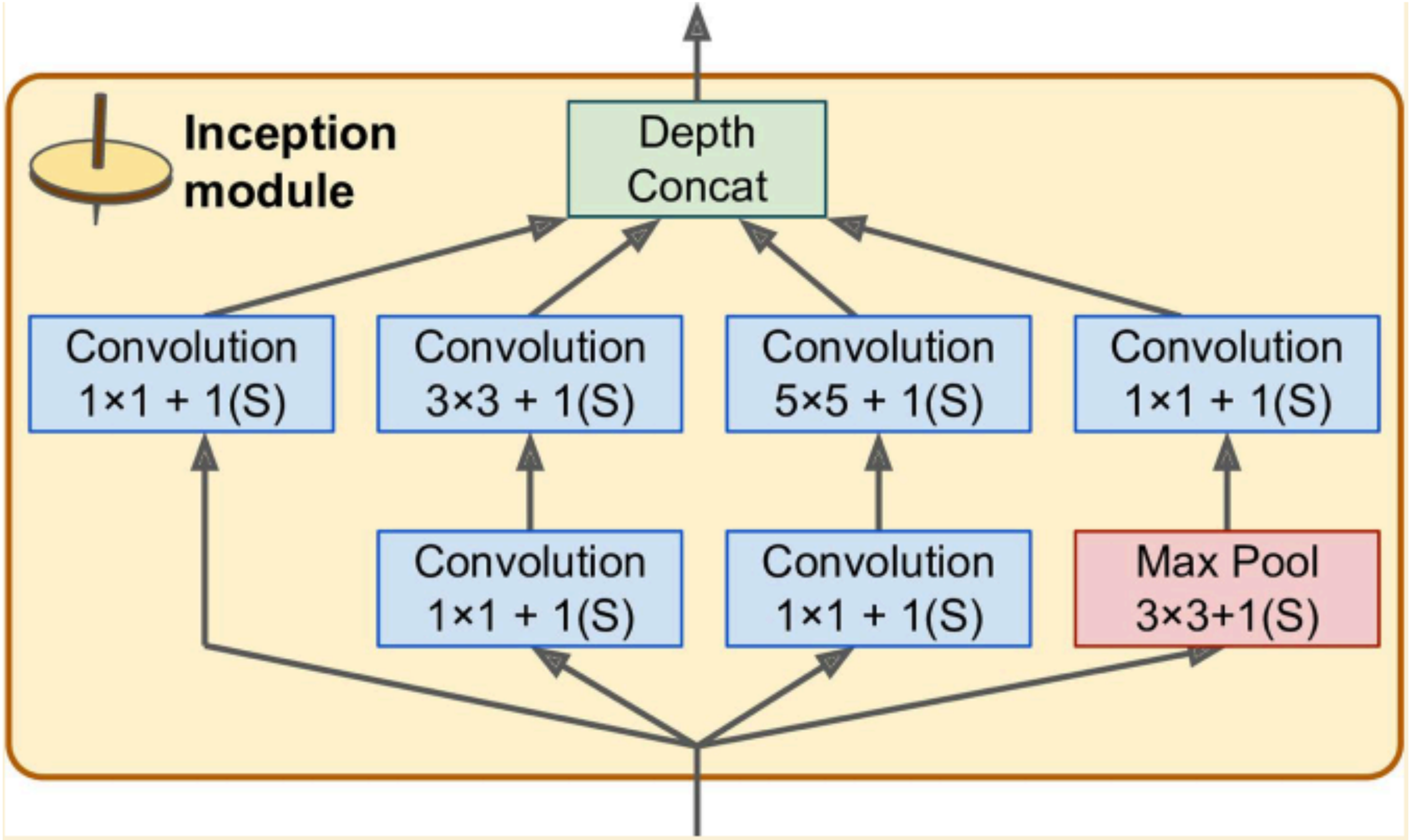
* Why 1x1 convolution?



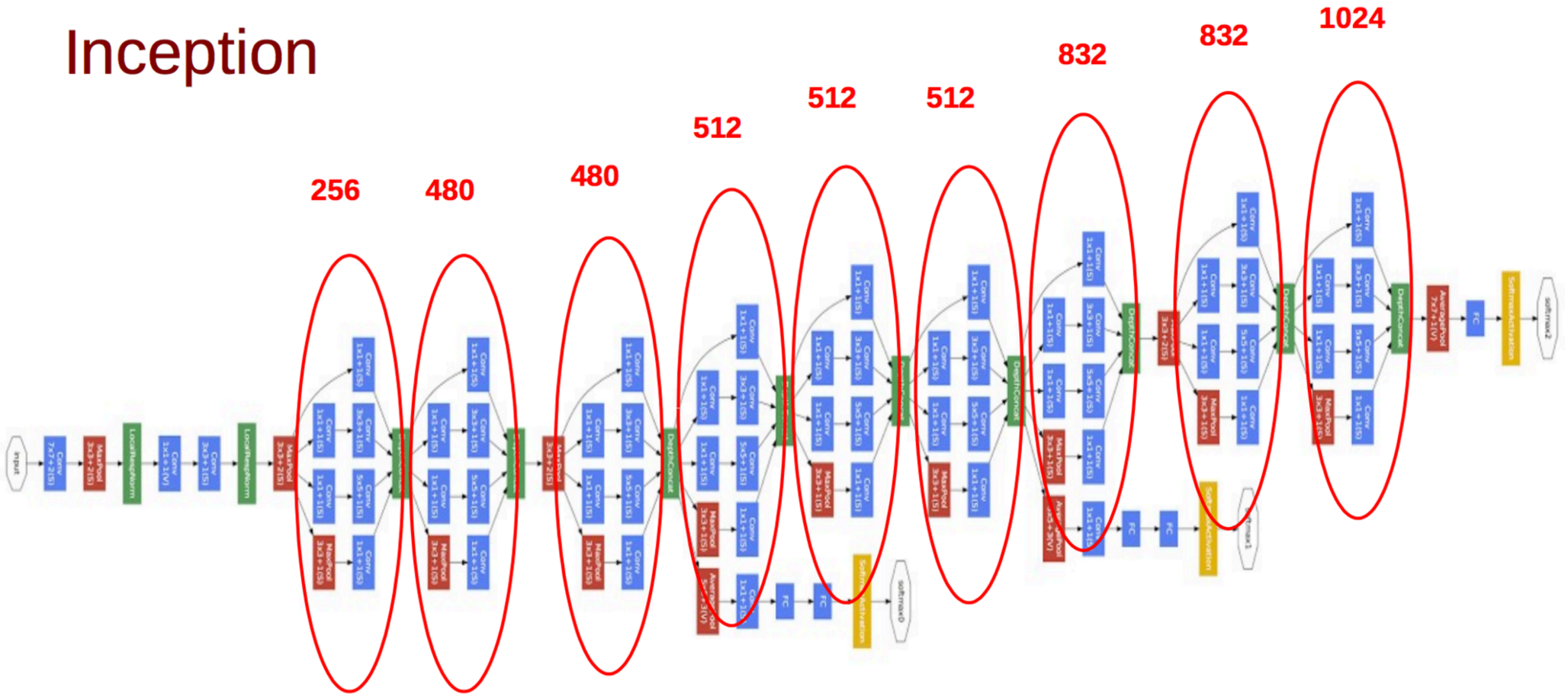
1x1 convolution



Review of ConvNets



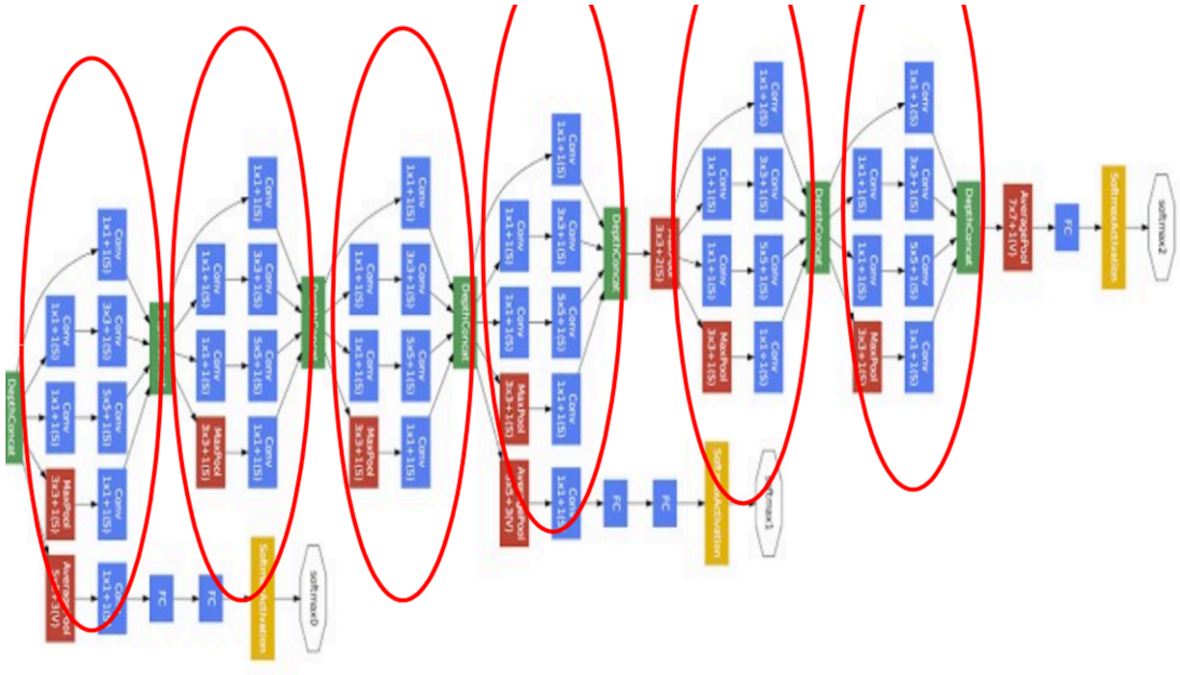
Review of ConvNets



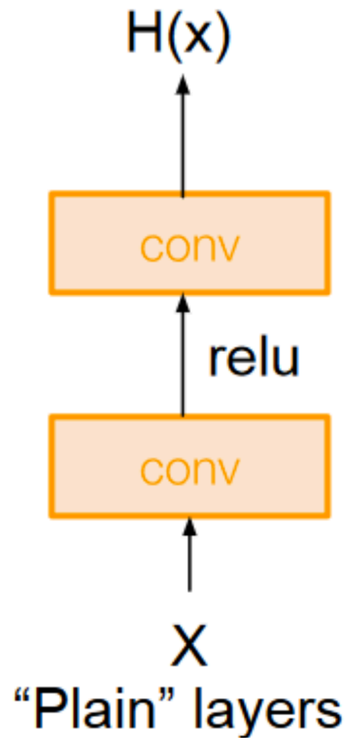
Width of **inception modules** ranges from 256 filters (in early modules) to 1024 in top inception modules.

Question

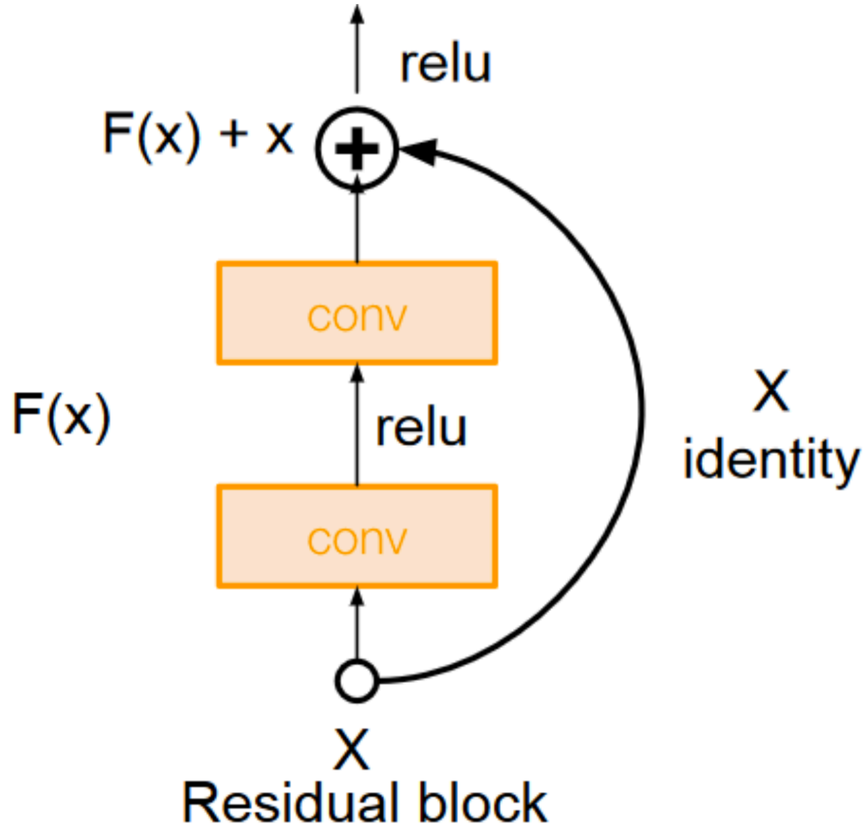
- Why do we need the side branches?



Residual learning



"Plain" layers

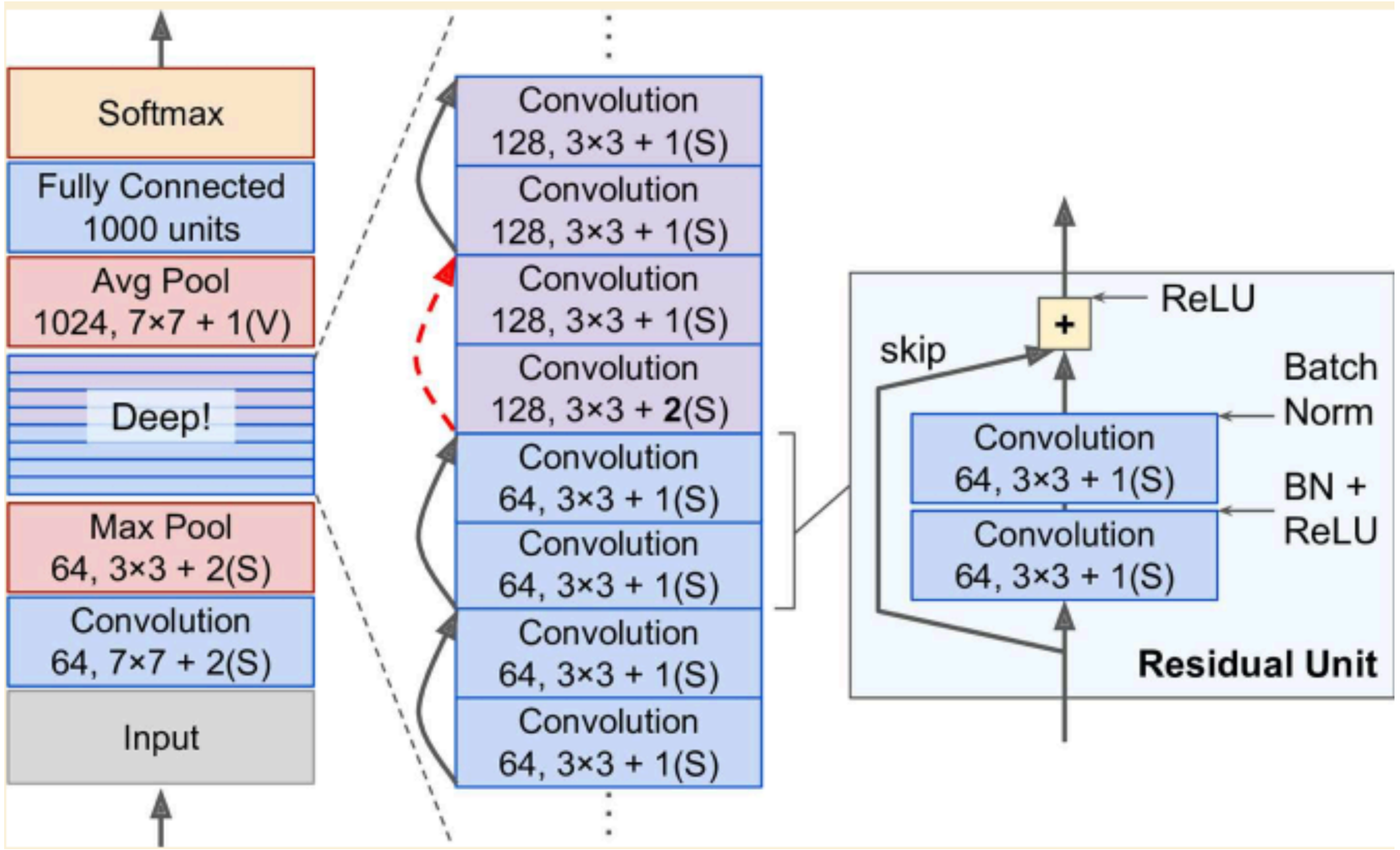


Residual block

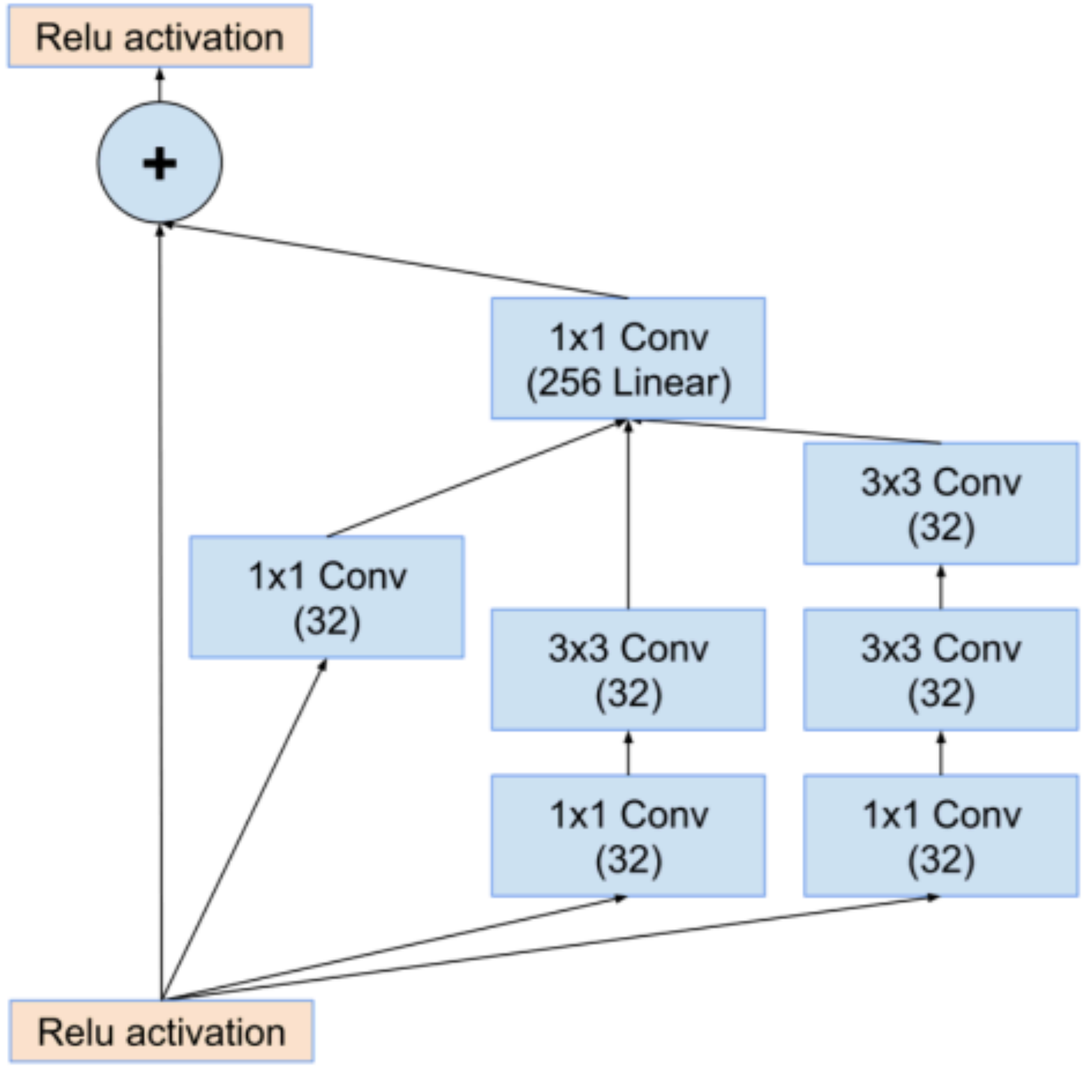
Question

* What types of Residual Blocks do you know?

Review of ConvNets



Review of ConvNets



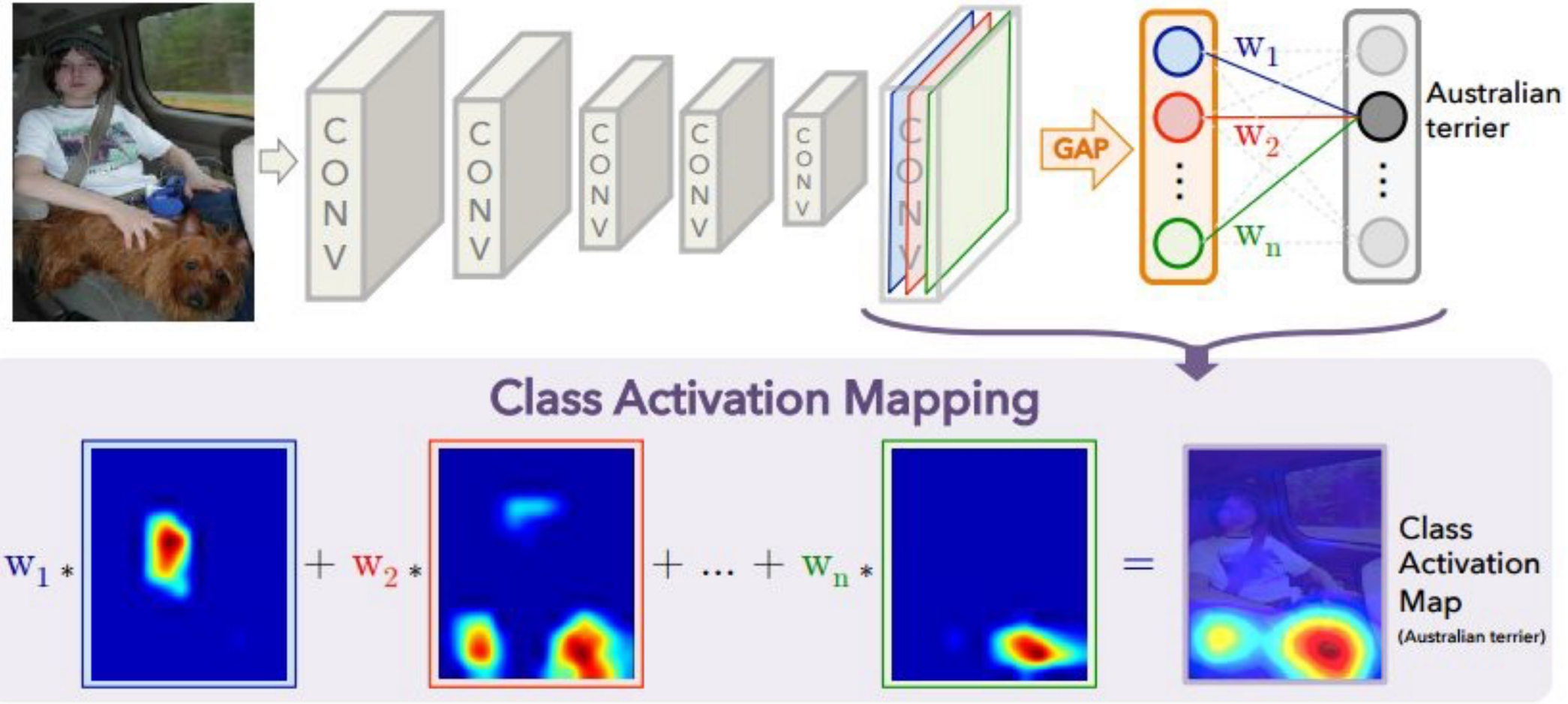
- Inception-v4
- Inception-ResNet

https://arxiv.org/abs/1602.07261

Demo

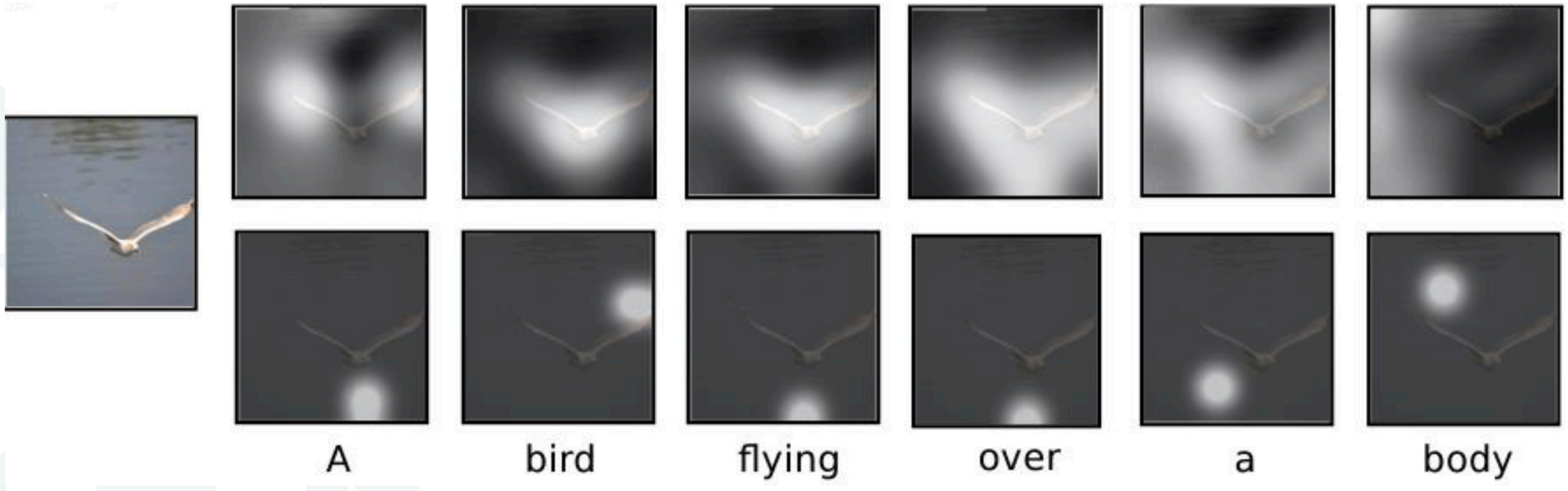
- https://cs.stanford.edu/people/karpathy/convnetjs/demo/cifar10.html

Aside: Class activation maps



https://towardsdatascience.com/demystifying-convolutional-neural-networks-using-class-activation-maps-fe94eda4cef1

Aside: Attention in CNNs



* Attention is a masking mechanism to blank out irrelevant areas of the image
* The attention mask is multiplied with the CNN feature output (aka gating)

Xu et. al. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention

Break, 5 min.

Exercise

- Calculate memory size needed for training the following layer:
  - The input is a 150 × 100 RGB image (three channels)
  - A convolutional layer with 5 × 5 filters,
  - Layer outputs 200 feature maps of size 150 × 100, with stride 1 and SAME padding,
  - Batch size = 100

From Classification to Localization and Detection

Localization and Detection

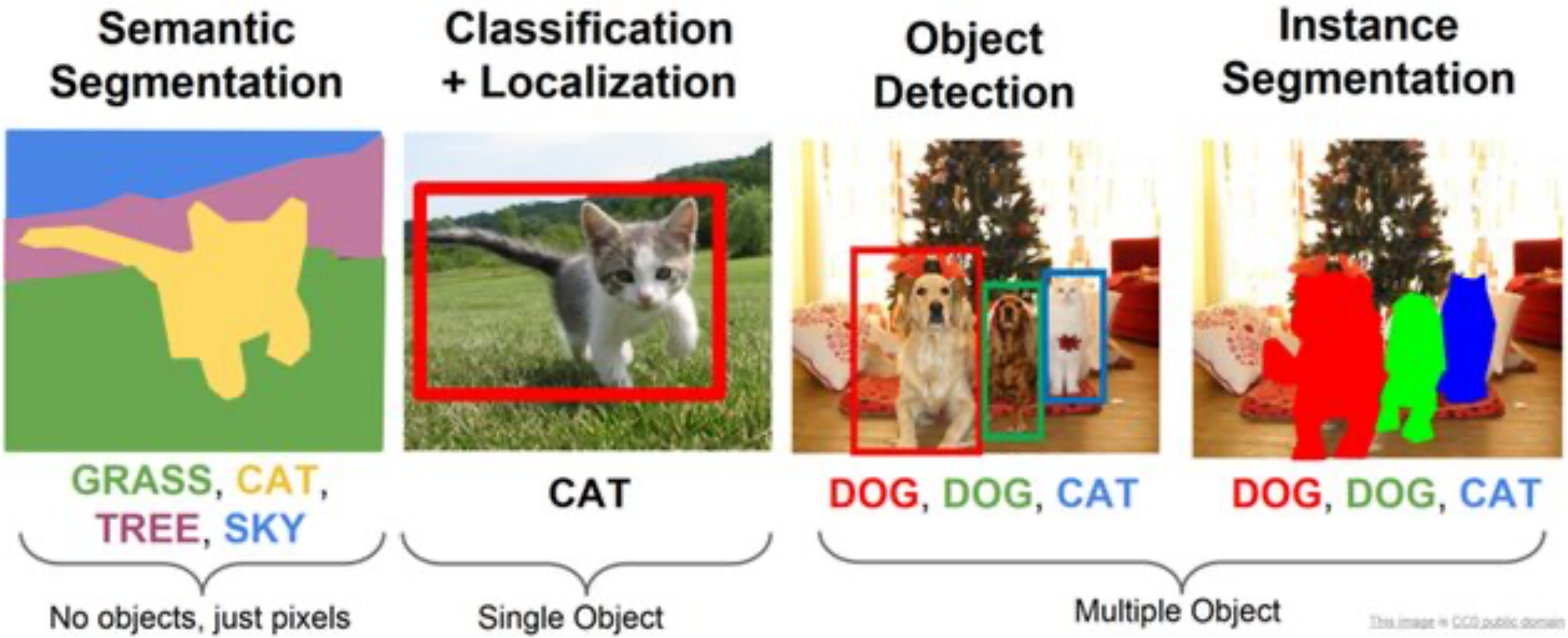
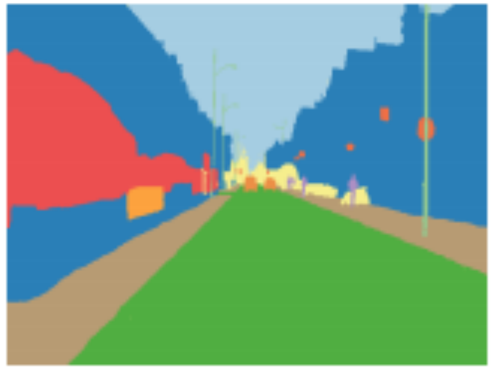
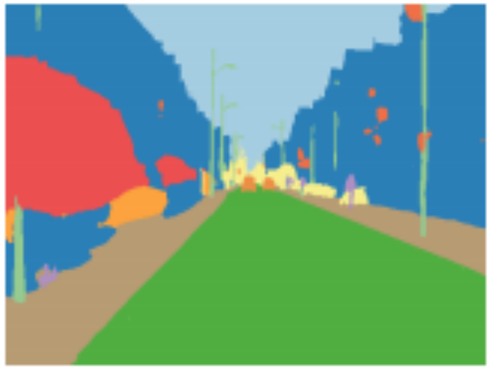


Image segmentation





* linking each pixel in an image to a class label

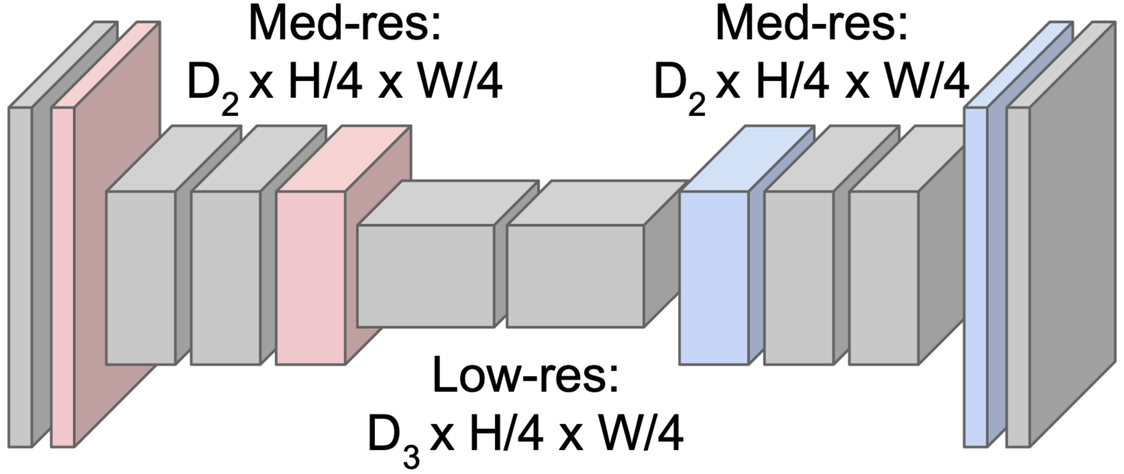
General Idea

Downsampling:
Pooling, strided convolution

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Upsampling:
Unpooling or strided transpose convolution

Input:
3 x H x W



High-res:
D_1 x H/2 x W/2

High-res:
D_1 x H/2 x W/2

Predictions:
H x W

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf

Question

* What is transpose convolution?
* What is unpooling?

Unpooling

Max Pooling
Remember which element was max!

1	2	6	3
3	5	2	1
1	2	2	1
7	3	4	8

Input: 4 x 4

5	6
7	8

Output: 2 x 2

...

Rest of the network

Max Unpooling
Use positions from pooling layer

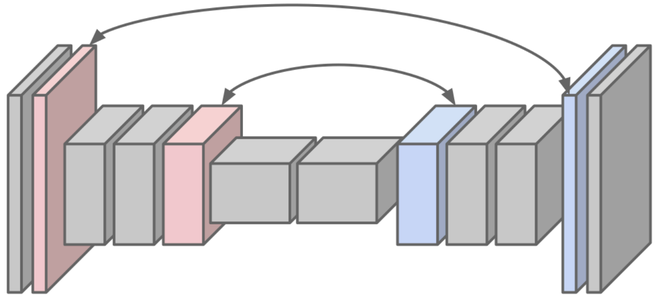
1	2
3	4

Input: 2 x 2

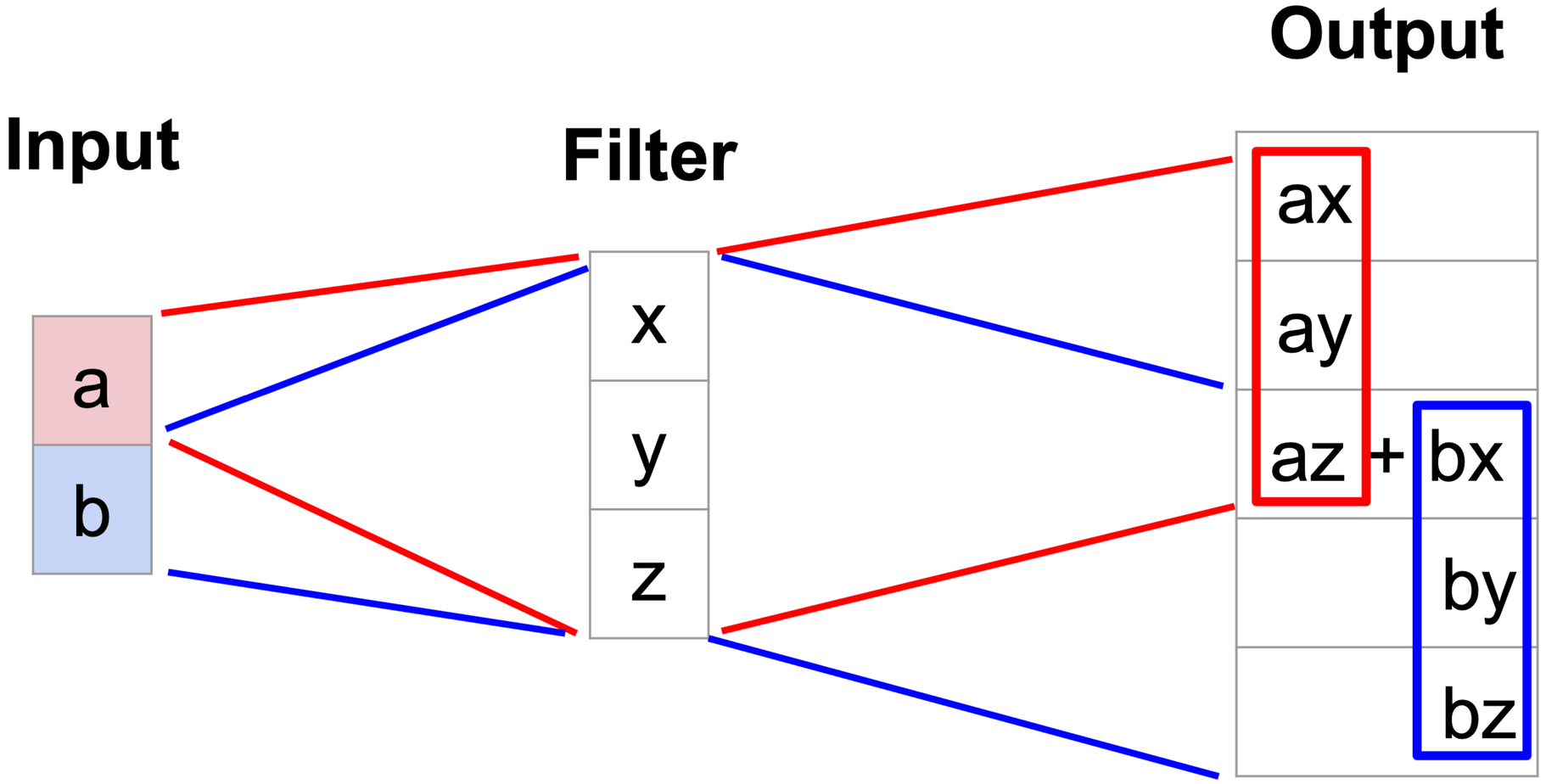
0	0	2	0
0	1	0	0
0	0	0	0
3	0	0	4

Output: 4 x 4

Corresponding pairs of downsampling and upsampling layers



Transpose Convolution: 1D Example



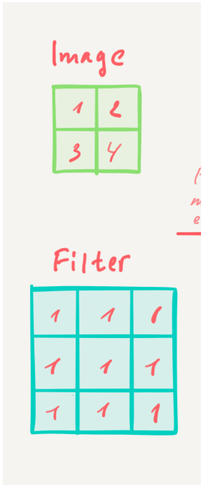
Output contains copies of the filter weighted by the input, summing at where at overlaps in the output

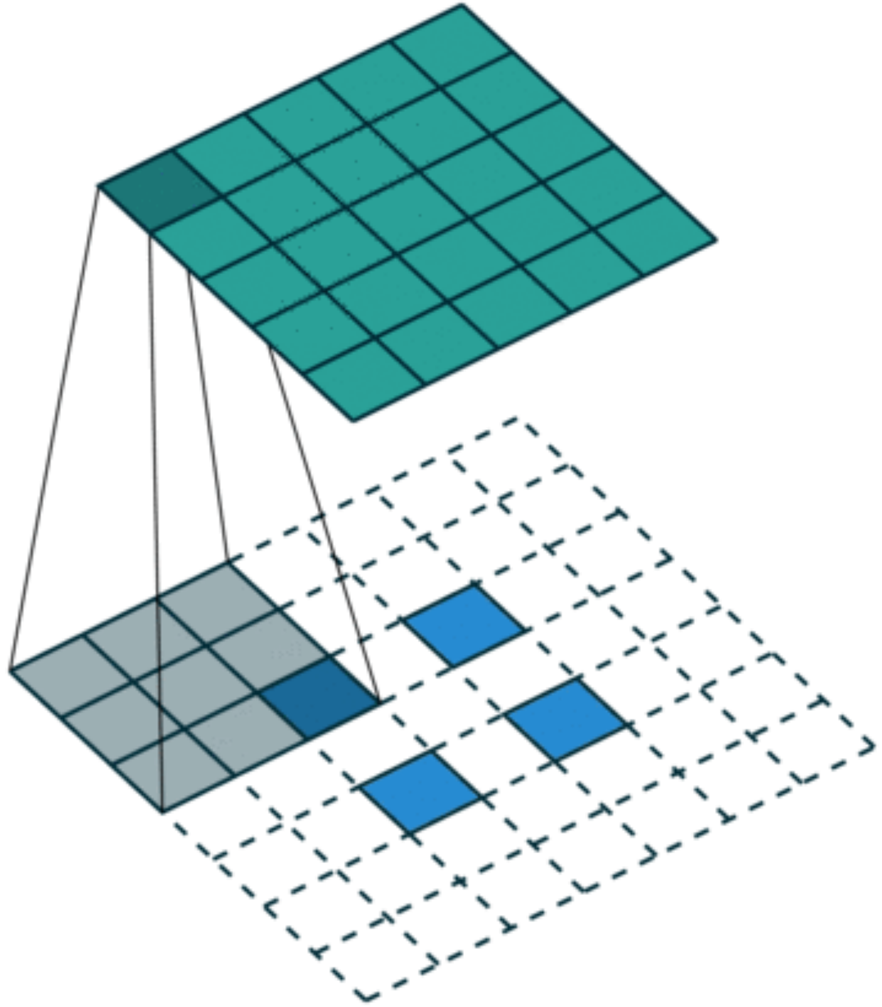
Need to crop one pixel from output to make output exactly 2x input

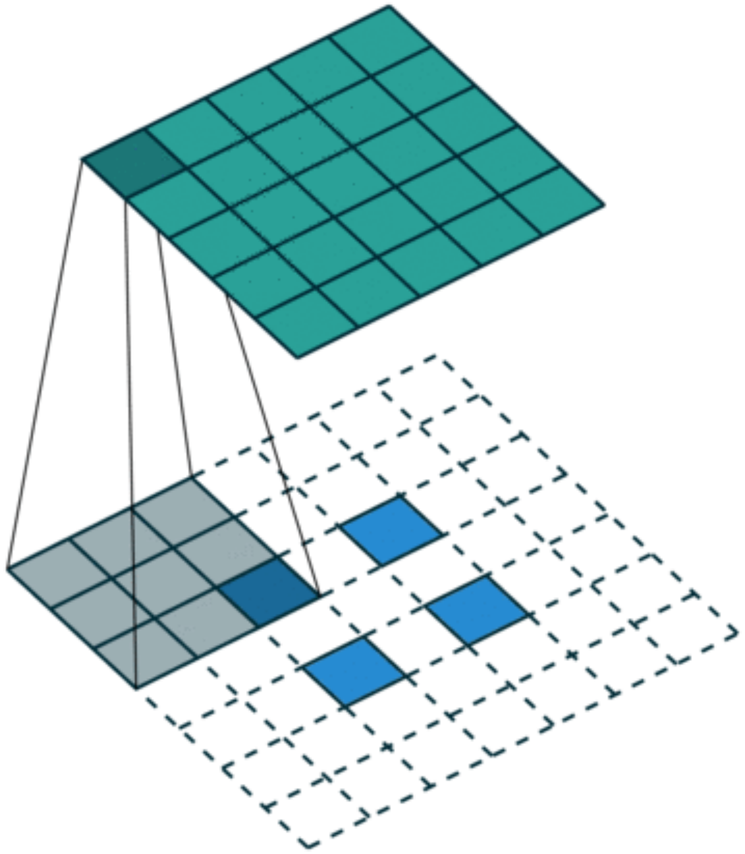
https://github.com/vdumoulin/conv_arithmetic

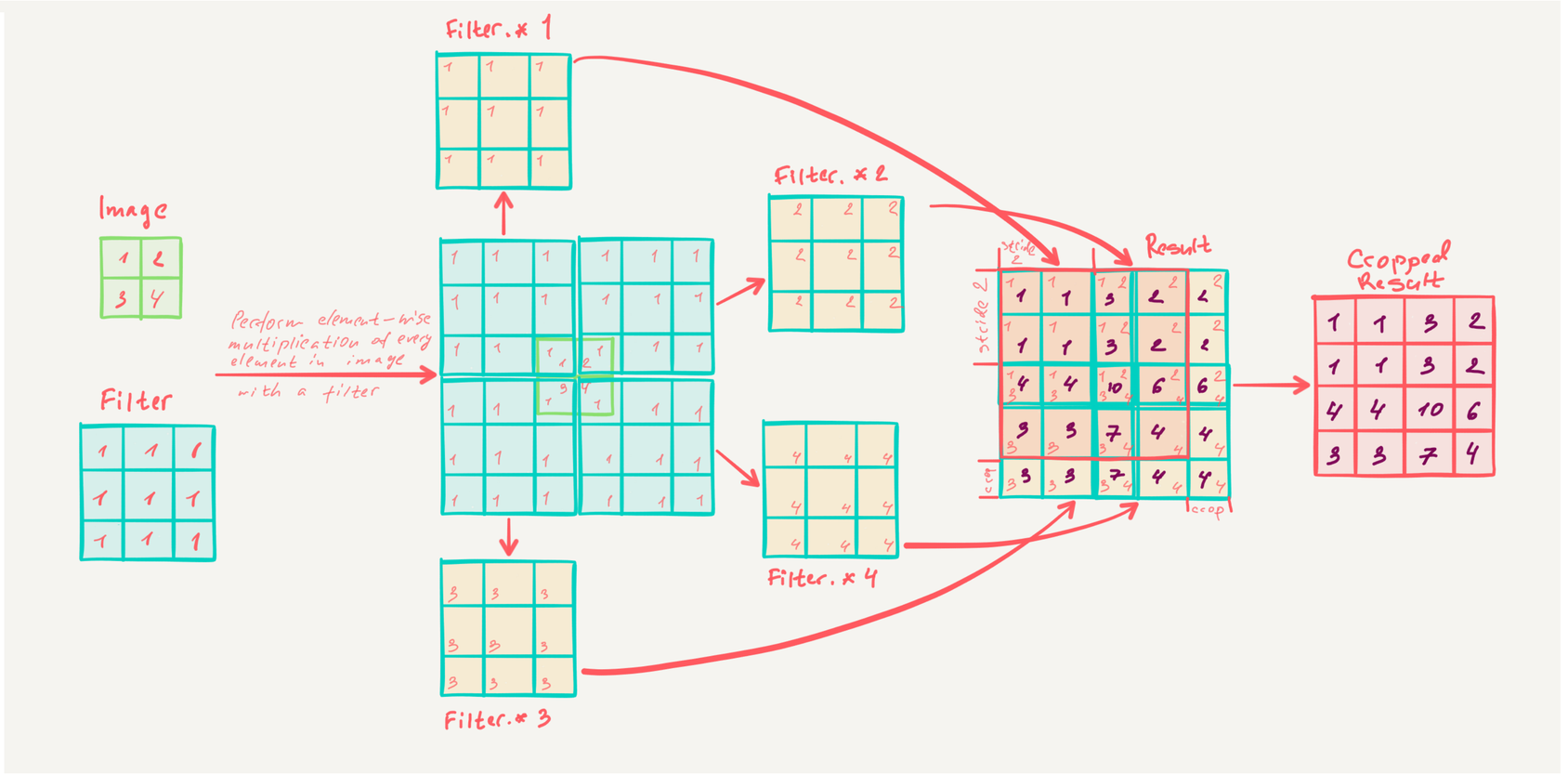
Exercise

- Run a transpose convolution upsampling with 3x3 filter and stride of 2








Filter.* 1
Image
Filter
Perform element-wise multiplication of every element in image with a filter
Filter.* 2
Filter.* 3
Filter.* 4
Result
Cropped Result
stride 2
crop

Fully Convolutional Networks, 2016

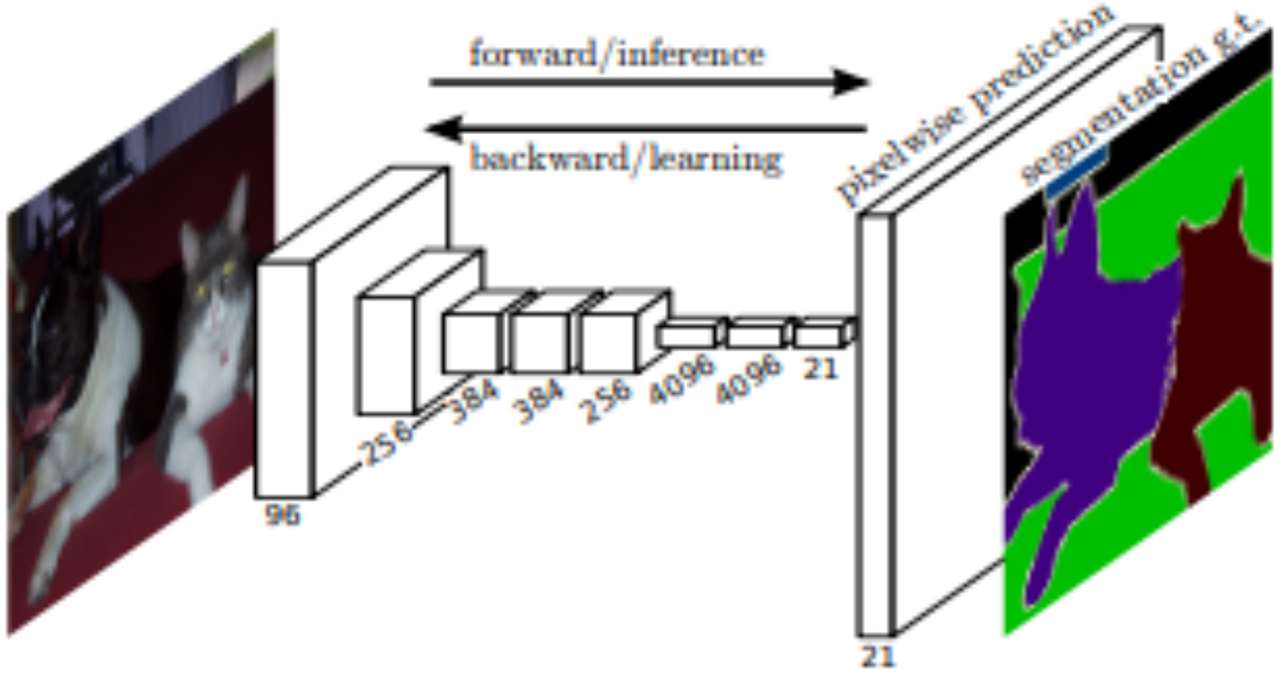
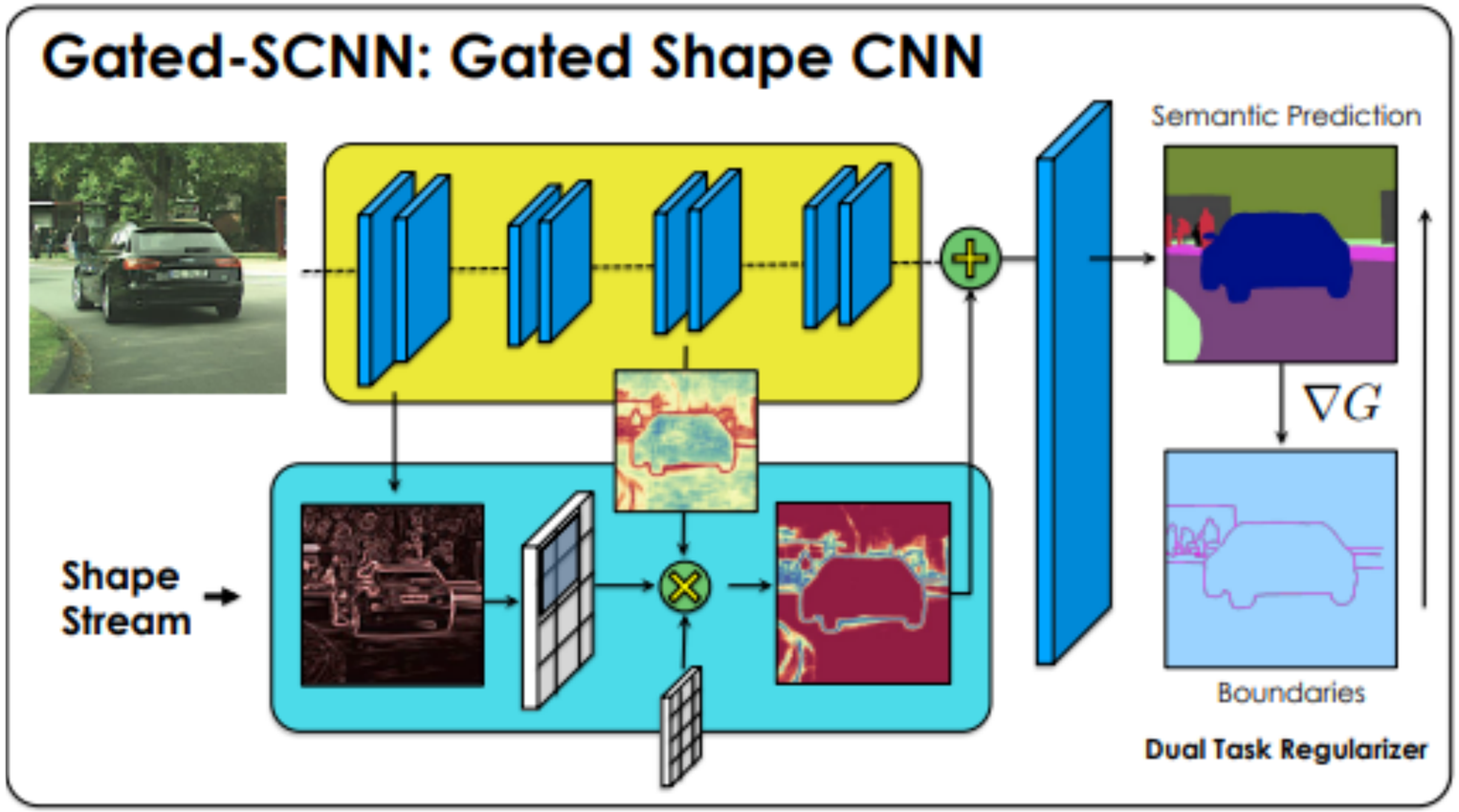


Fig. 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

- turned classification networks into FCN
- fine-tuning to the segmentation task

https://arxiv.org/abs/1605.06211

SOTA, 2019



* explicitly wires shape information as a separate processing branch

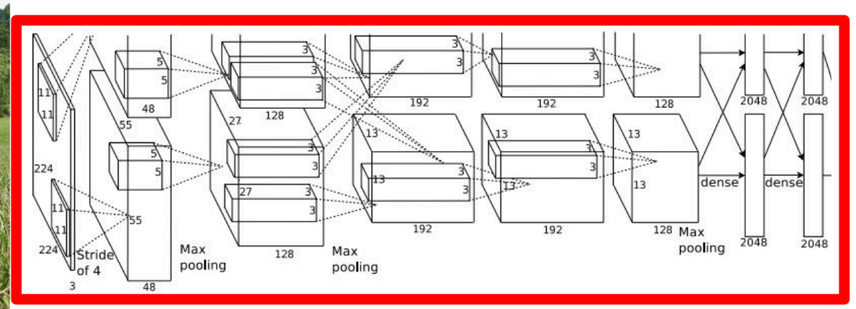
https://arxiv.org/abs/1907.05740

Localization

* Classification + Localizaion (1 object)
* Softmax loss + regression loss for bounding box (x,y,h,w) using MSE

Classification + Localization

This image is CC0 public domain



Often pretrained on ImageNet
(Transfer learning)

Vector: 4096

Fully Connected: 4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Softmax Loss

Correct label: Cat

Fully Connected: 4096 to 4

Box Coordinates (x, y, w, h)

L2 Loss

Correct box: (x', y', w', h')

+ → **Loss**

Treat localization as a regression problem!

http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf

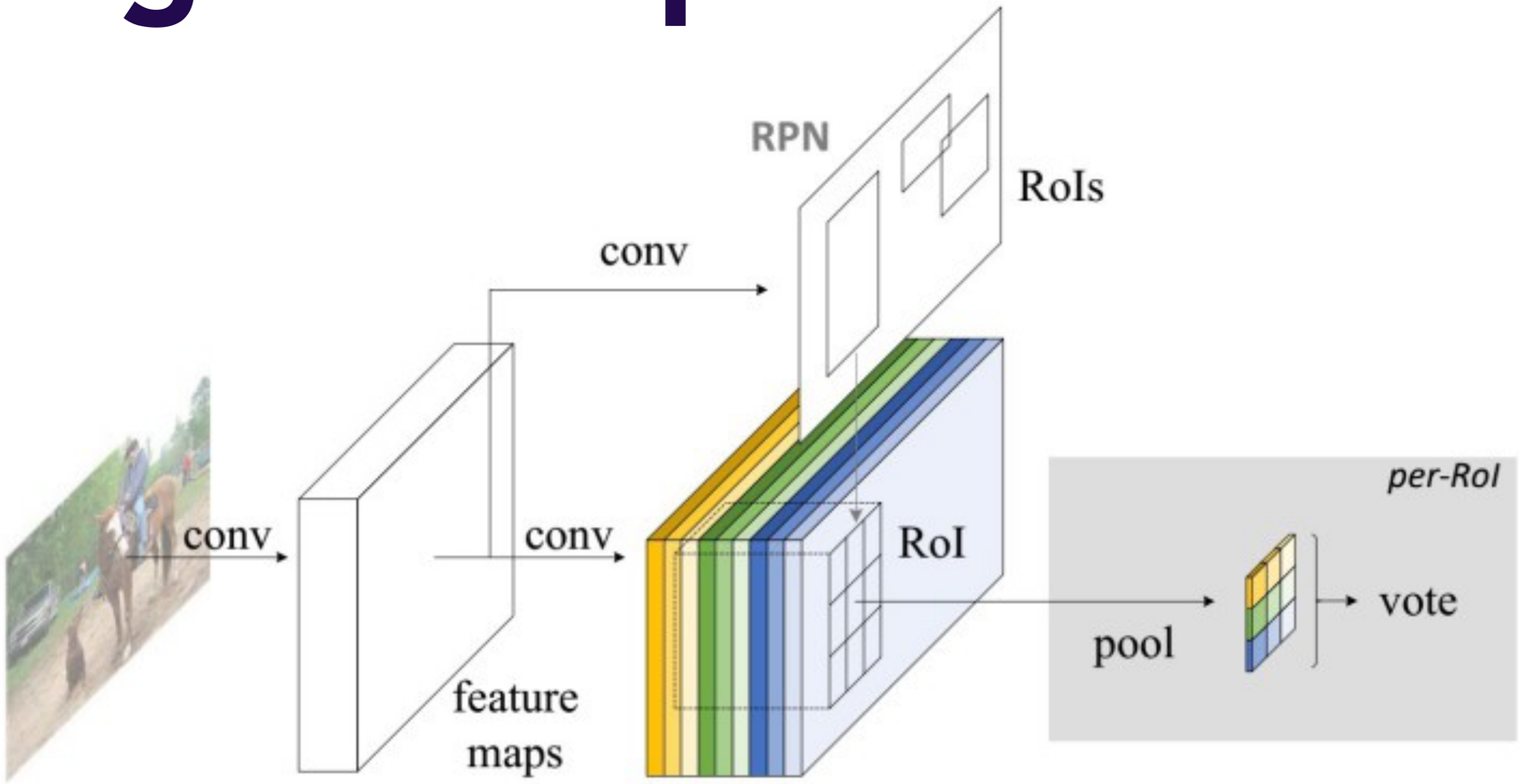
Object Detection

* multiple objects,
* bounding box for each

Object Detection

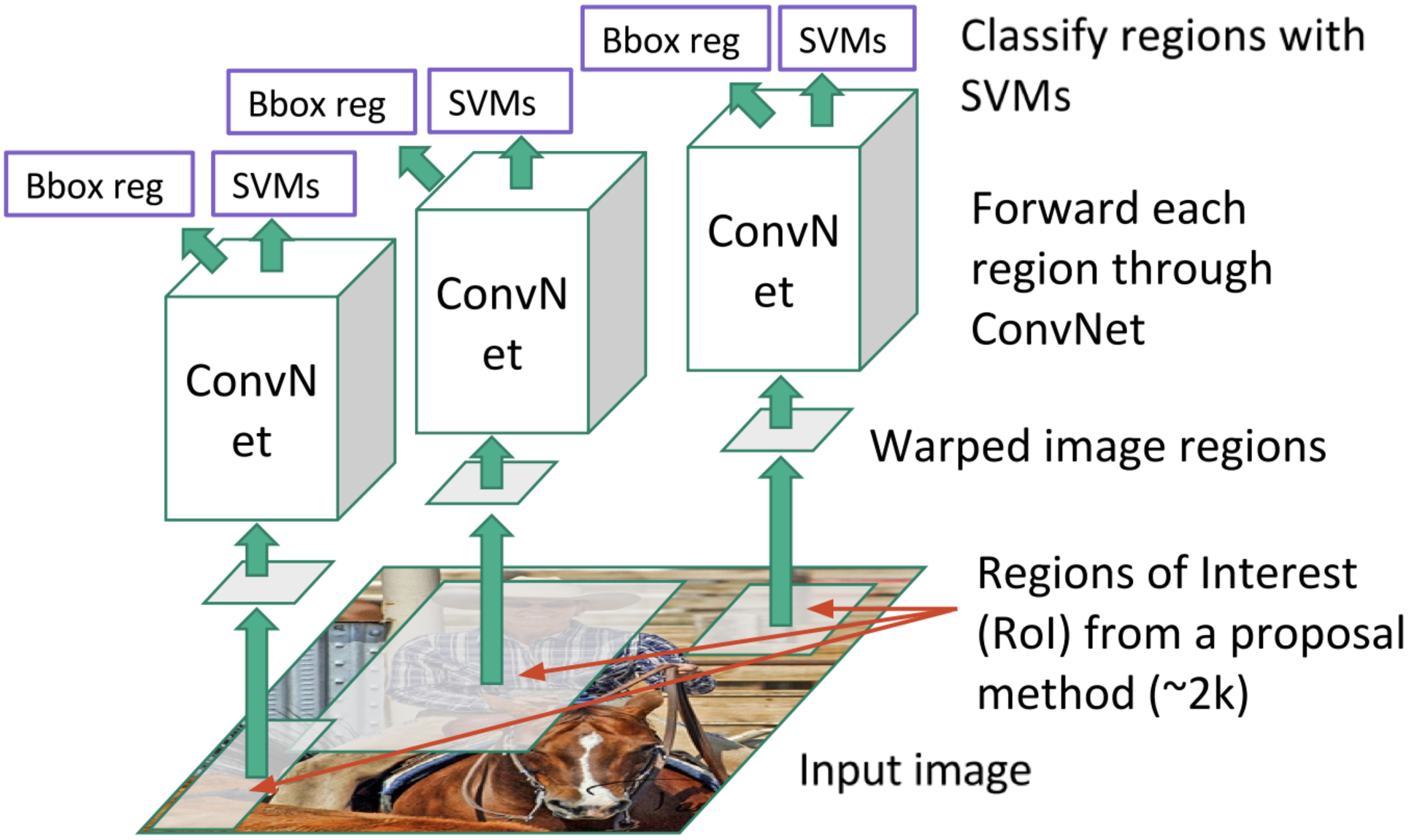
* R-CNN , Fast/er R-CNN
* SSD, YOLO

Region Proposal Networks

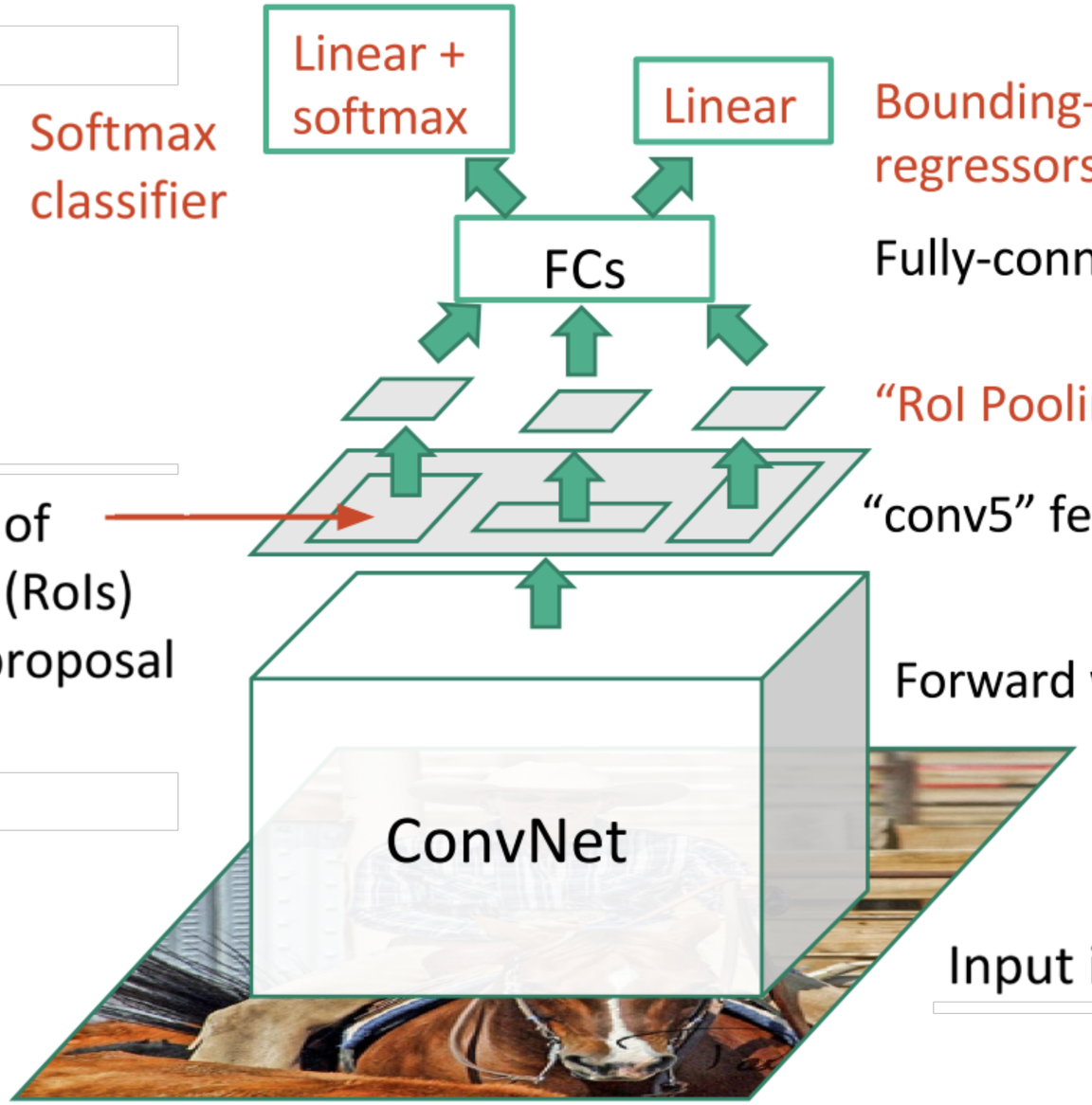


https://arxiv.org/pdf/1506.01497.pdf

R-CNN


Linear Regression for bounding box offsets
Bbox reg
SVMs
Classify regions with SVMs
Bbox reg
SVMs
Bbox reg
SVMs
ConvNet
ConvNet
ConvNet
Forward each region through ConvNet
Warped image regions
Regions of Interest (RoI) from a proposal method (~2k)
Input image

Fast R-CNN



Softmax classifier

Bounding-box regressors

Fully-connected layers

"RoI Pooling" layer

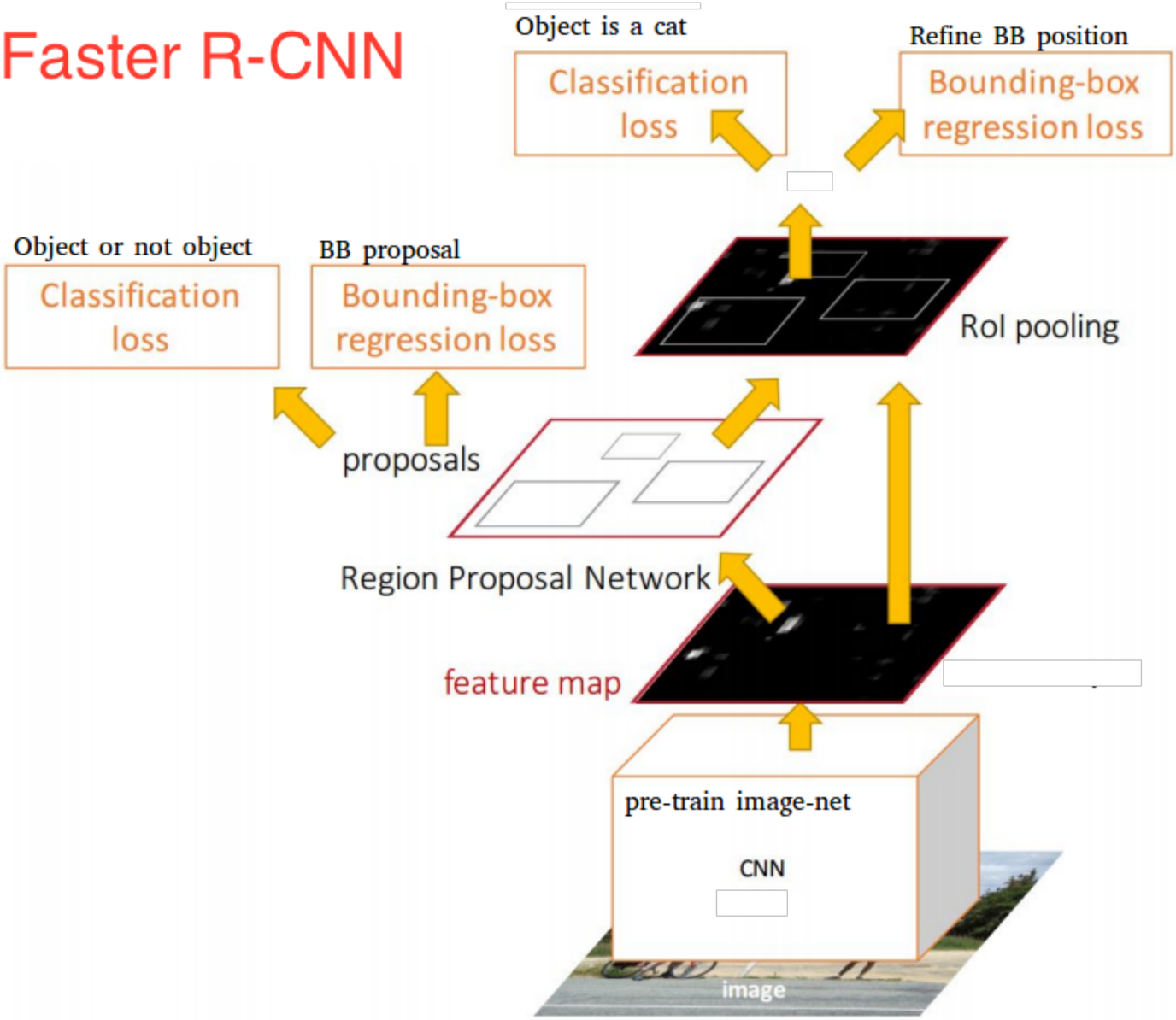
"conv5" feature map of image

Regions of Interest (RoIs) from a proposal method

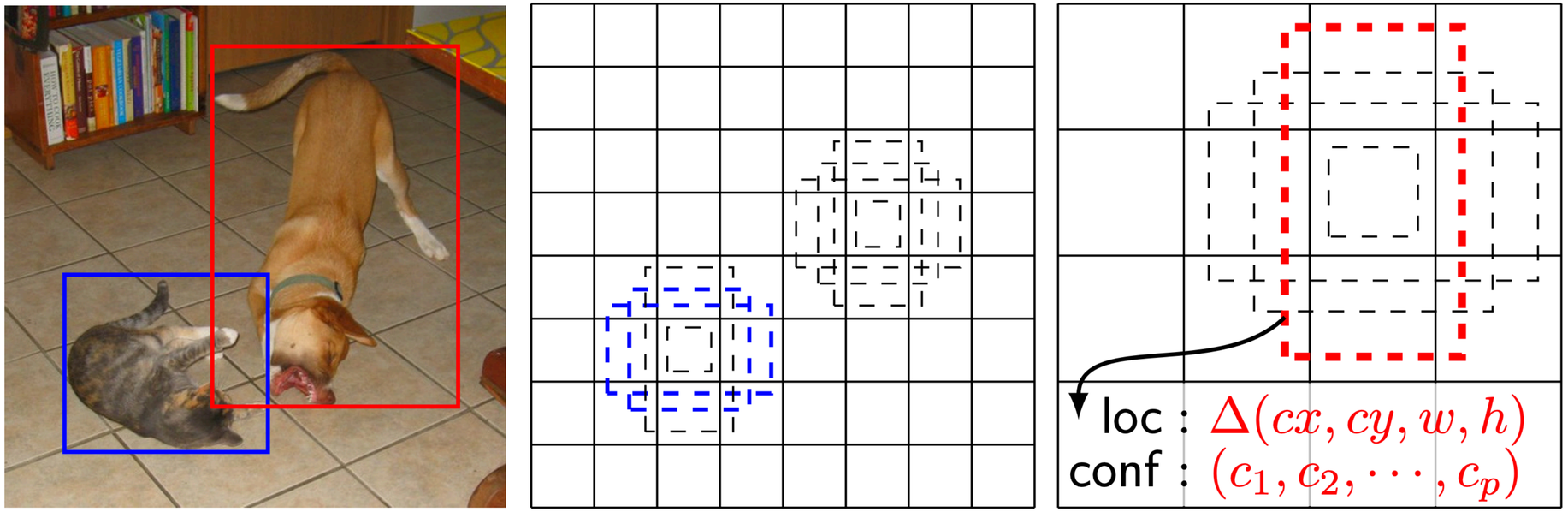
Forward whole image through ConvNet

Input image

Faster R-CNN


Object is a cat
Classification loss
Refine BB position
Bounding-box regression loss
Object or not object
Classification loss
BB proposal
Bounding-box regression loss
RoI pooling
proposals
Region Proposal Network
feature map
pre-train image-net
CNN
image

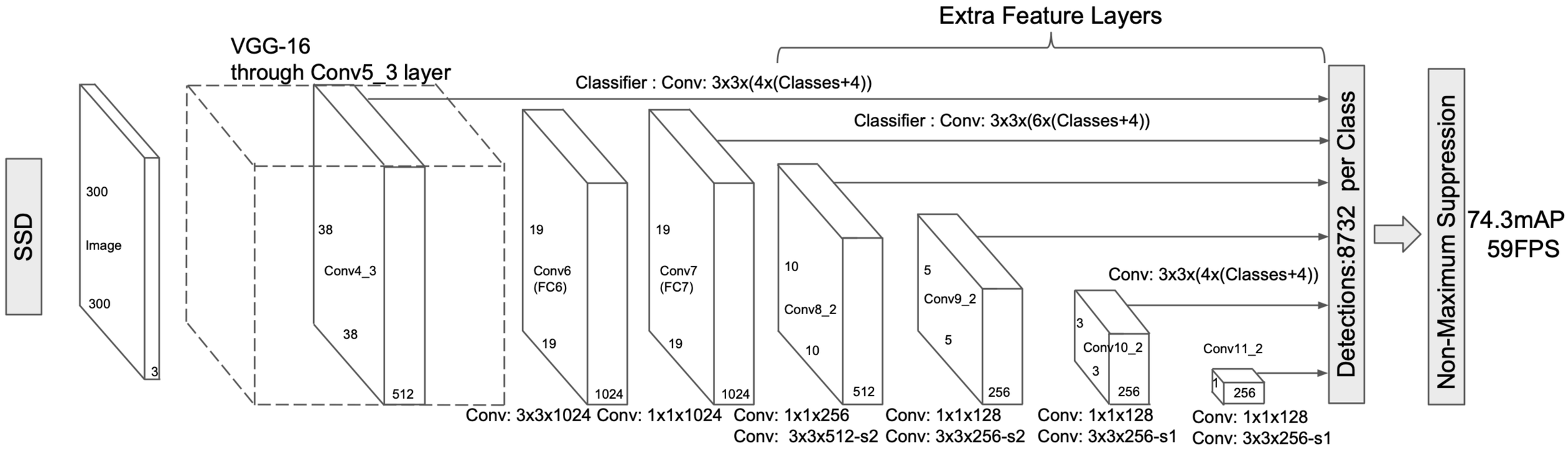
Single-Shot MultiBox Detector (SSD)



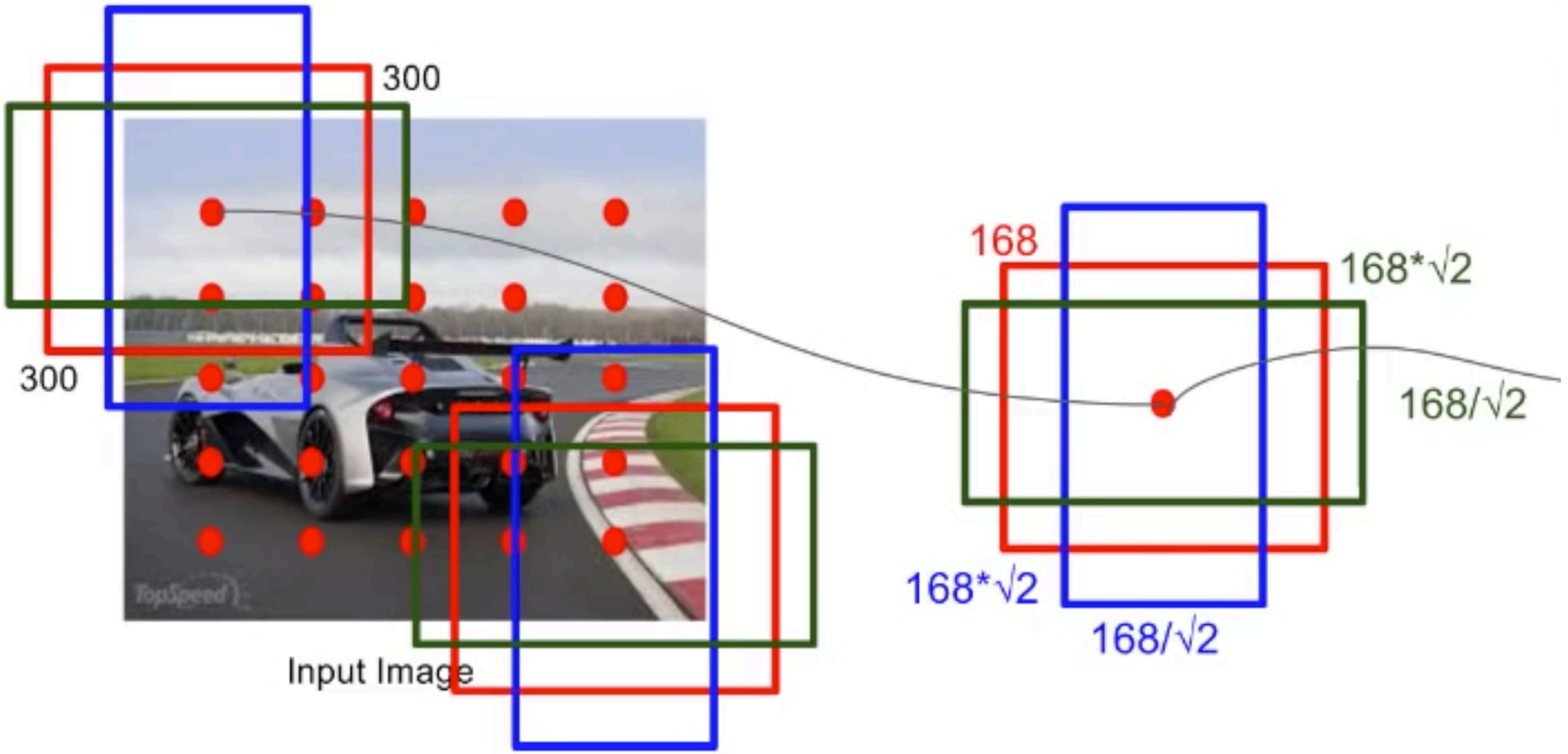
(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

https://arxiv.org/pdf/1512.02325.pdf

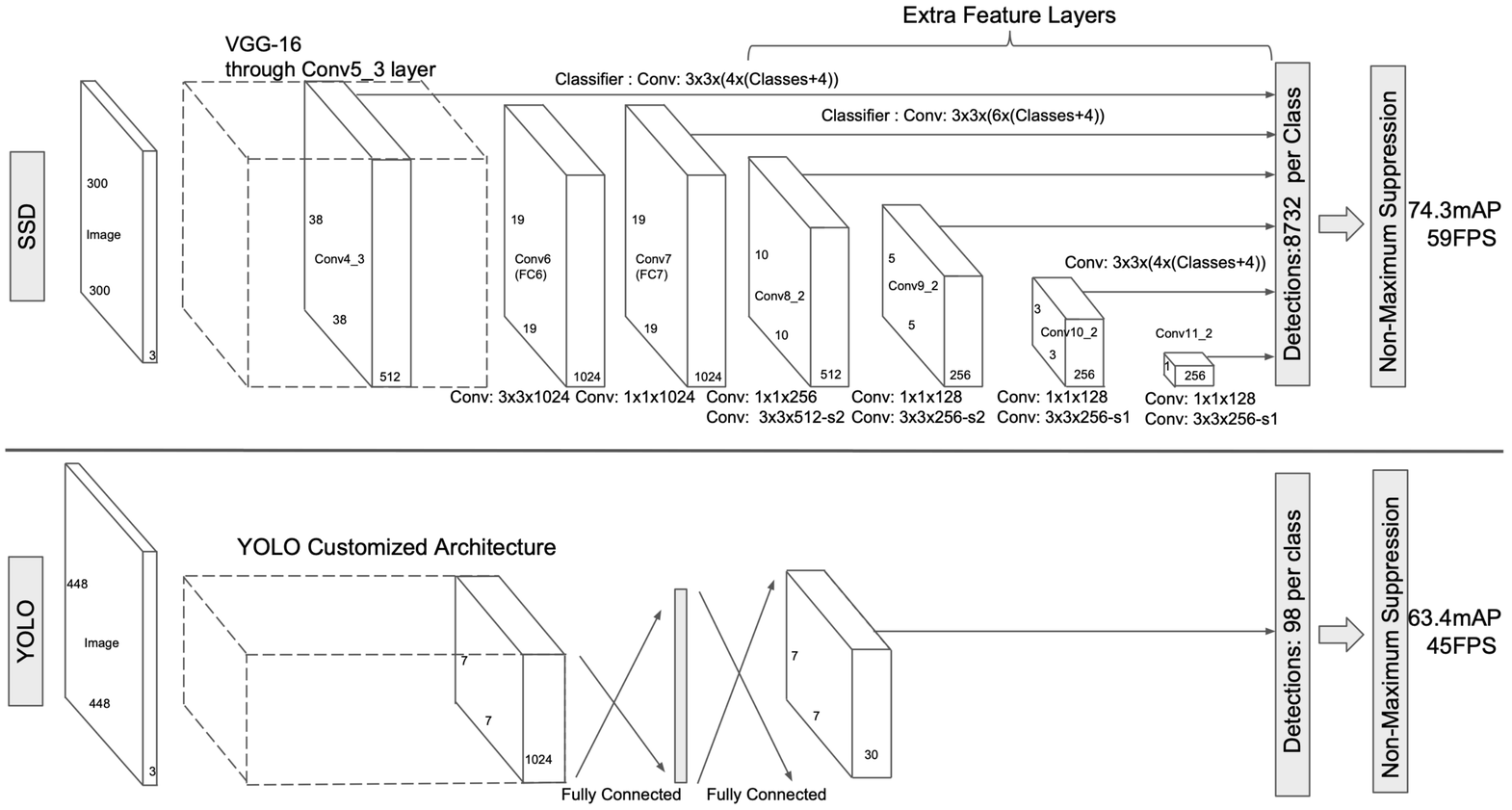
Architecture



Shape problem



SSD and YOLO



https://github.com/NVIDIA/DeepLearningExamples/tree/master/PyTorch/Detection/SSD

Instance Segmentation

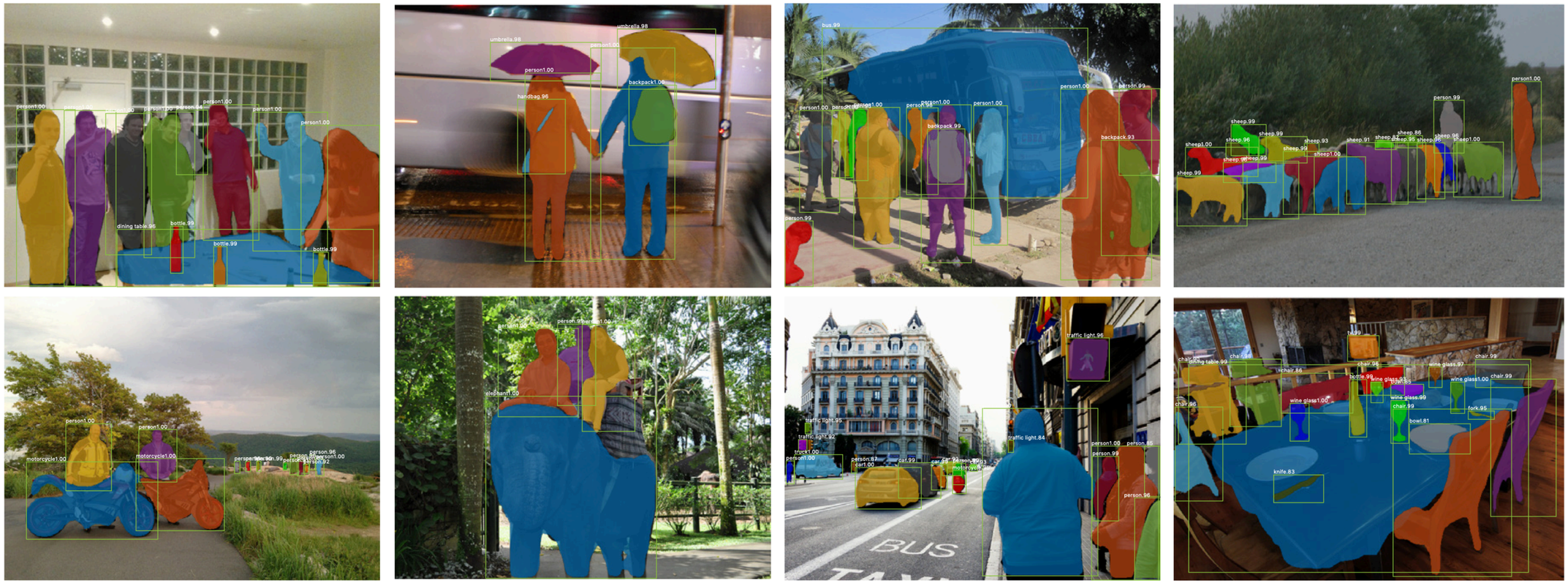
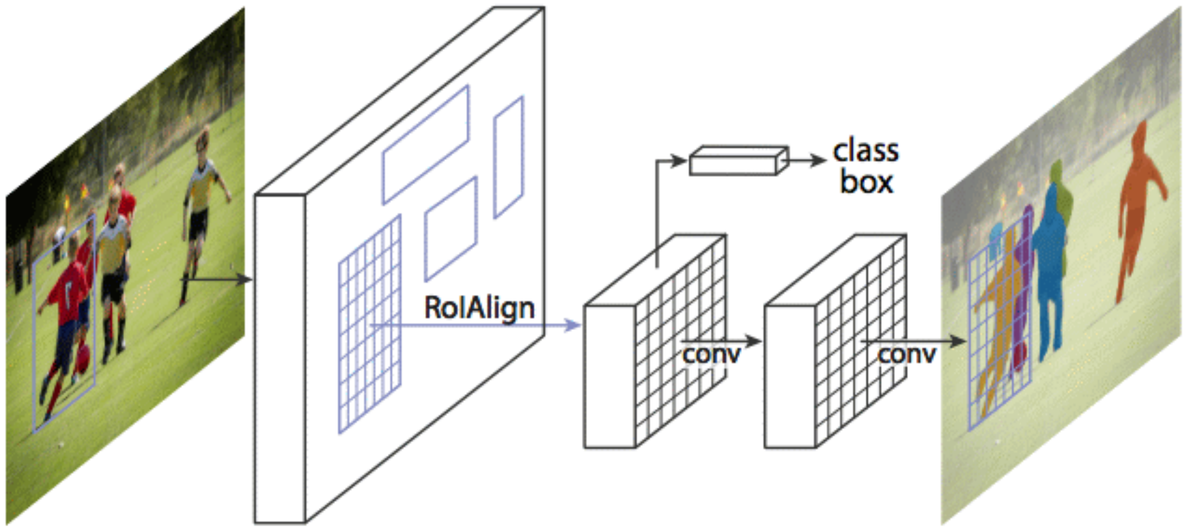


Figure 2. **Mask R-CNN** results on the COCO test set. These results are based on ResNet-101 [19], achieving a *mask* AP of 35.7 and running at 5 fps. Masks are shown in color, and bounding box, category, and confidences are also shown.

https://arxiv.org/pdf/1703.06870.pdf

Instance Segmentation



* Mask R-CNN = Faster R-CNN + FCN
* RoI Align

https://github.com/NVIDIA/DeepLearningExamples/tree/master/PyTorch/Segmentation/MaskRCNN

Neural Style Transfer

content image

Ancient city of Persepolis

+

style image

The Starry Night (Van Gogh)

=

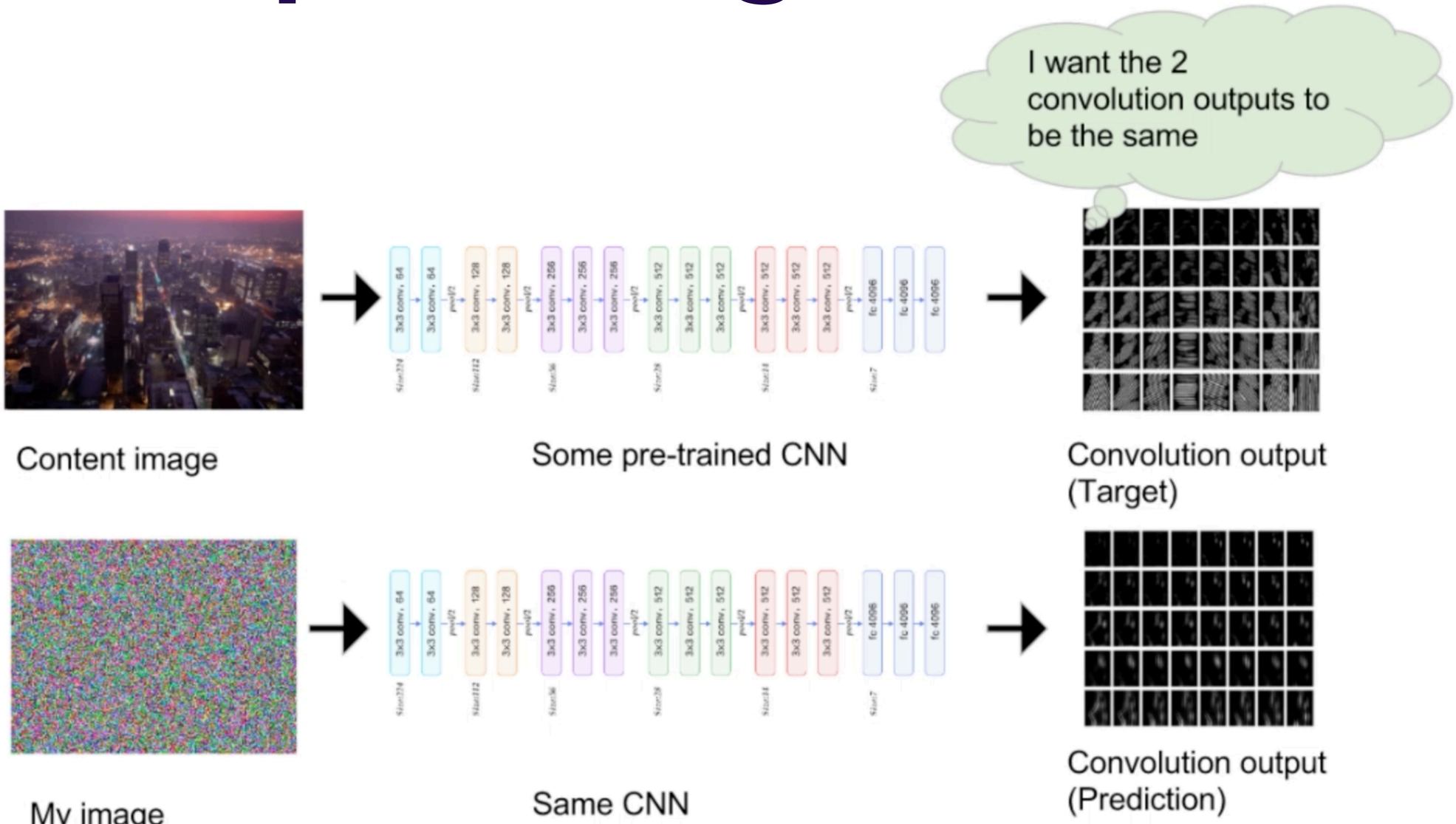
generated image

Persepolis
in Van Gogh style

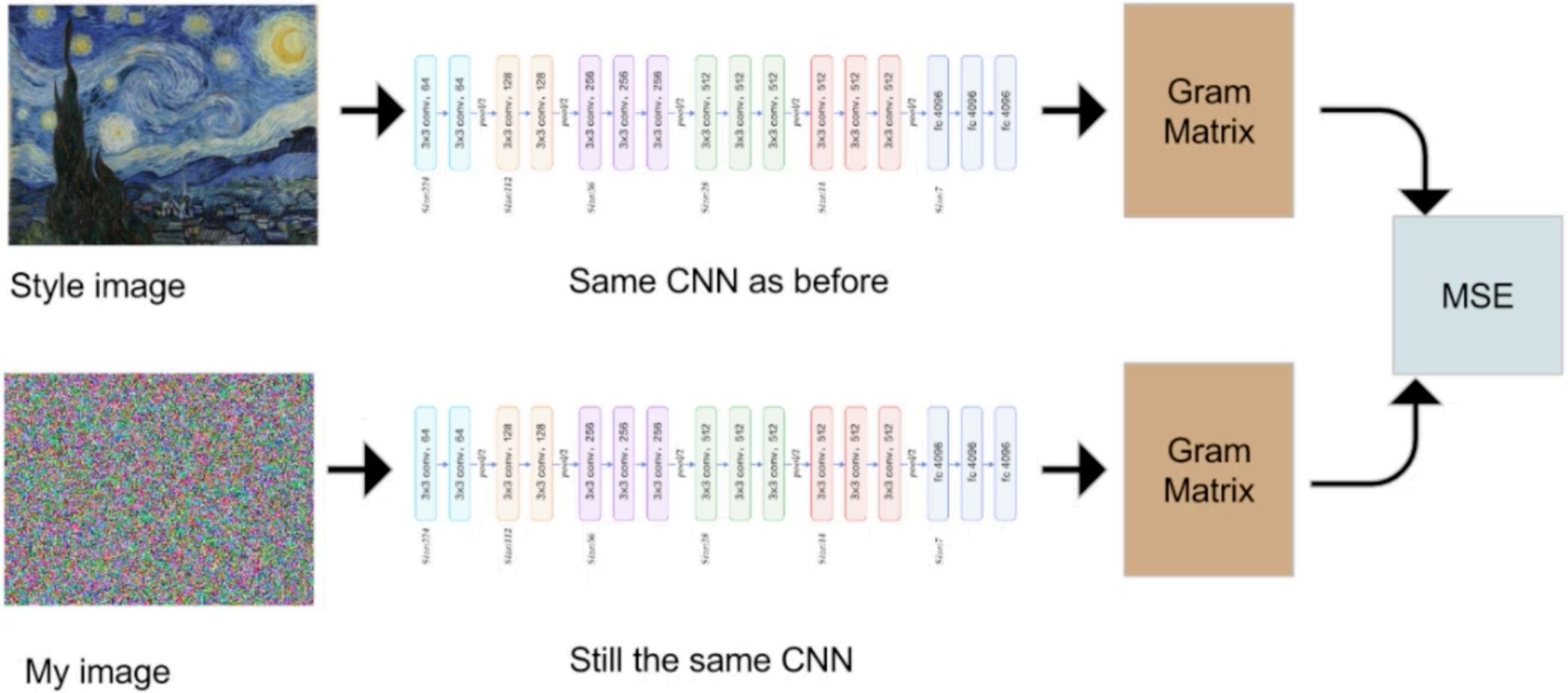
Idea

- We optimise input image instead of weights (recall the adversarial attacks)
- $$X^* = \arg\min_X L$$
- Combination of two losses:
- $$L = \alpha L_c + \beta L_s$$

Reproducing content



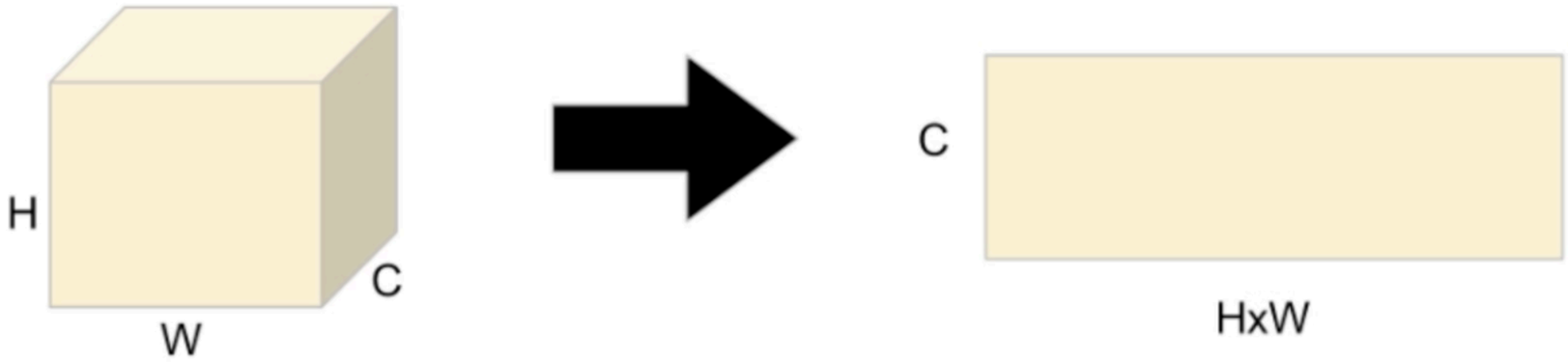
Recreating Style



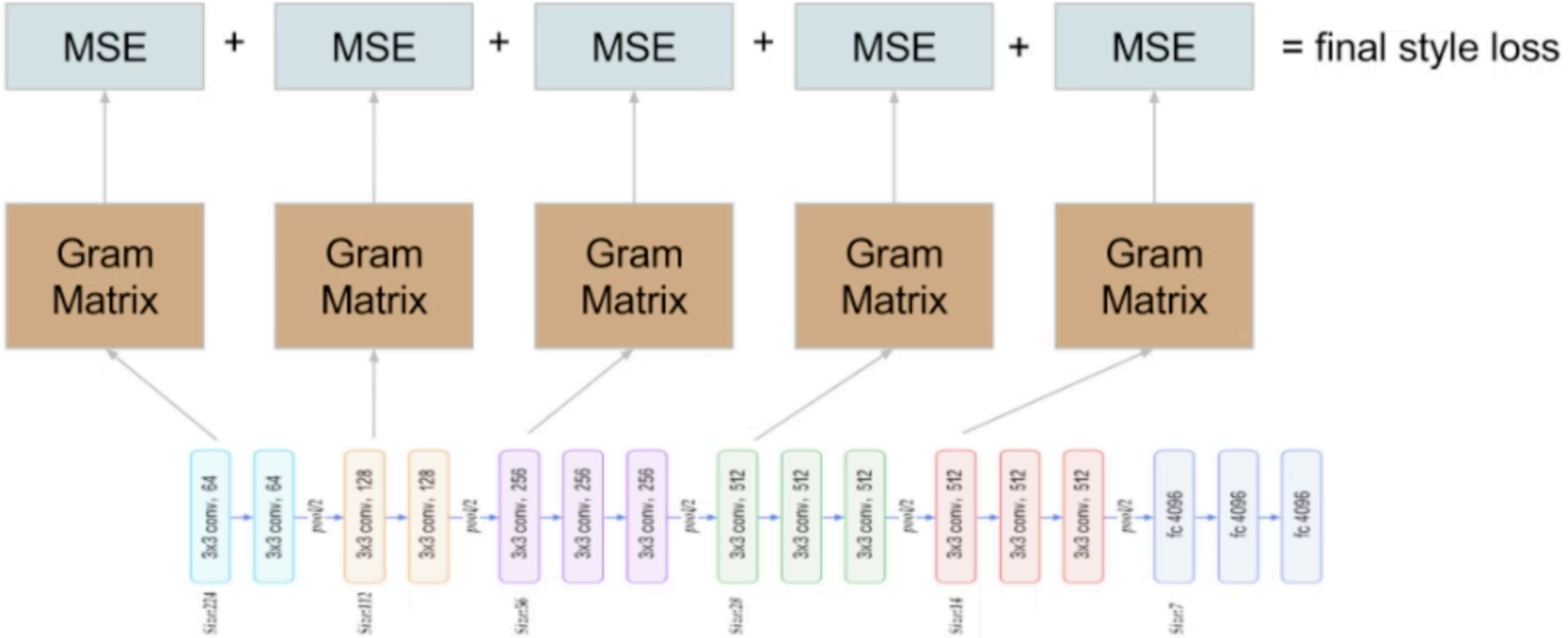
Gram Matrix

* $G = \frac{1}{N} X X^T$

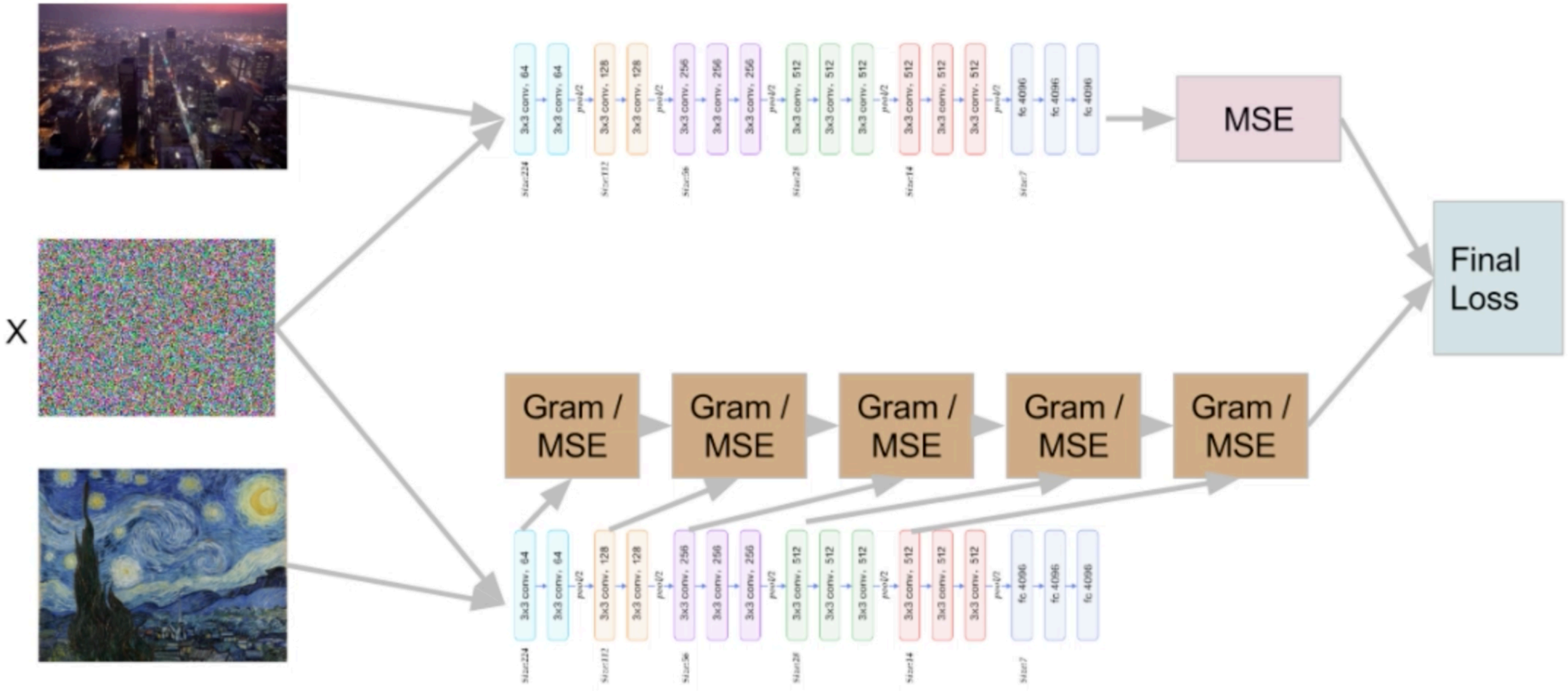
* Autocorrelation of features



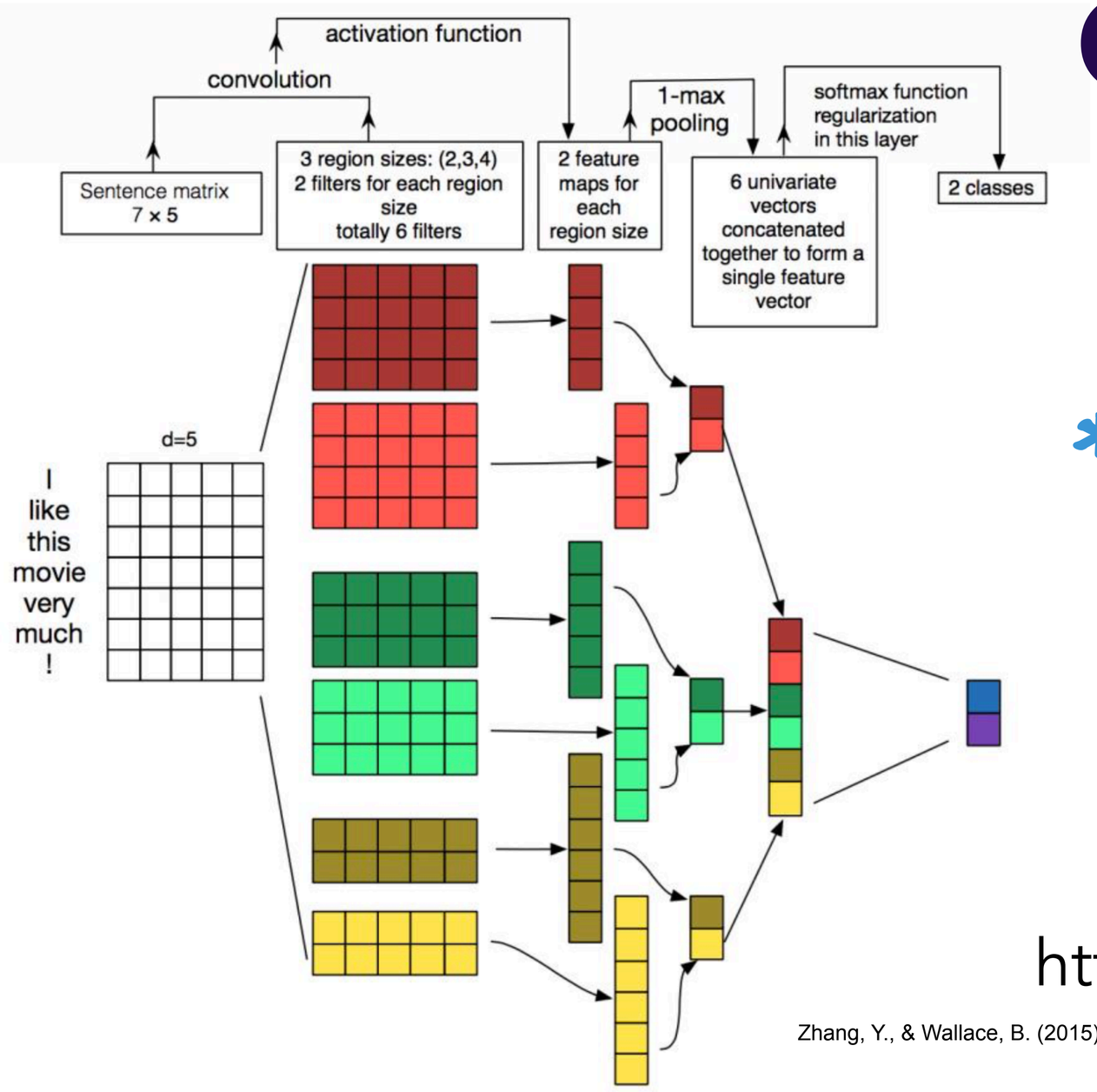
Style loss


MSE
+
MSE
+
MSE
+
MSE
+
MSE
= final style loss
Gram Matrix
Gram Matrix
Gram Matrix
Gram Matrix
Gram Matrix
3x3 conv, 64
3x3 conv, 64
pool/2
3x3 conv, 128
3x3 conv, 128
pool/2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
pool/2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool/2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool/2
fc 4096
fc 4096
fc 4096

Finally…


X
3x3 conv, 64
3x3 conv, 128
3x3 conv, 256
3x3 conv, 512
fc 4096
MSE
Gram / MSE
Gram / MSE
Gram / MSE
Gram / MSE
Gram / MSE
Final Loss

CNNs for Text

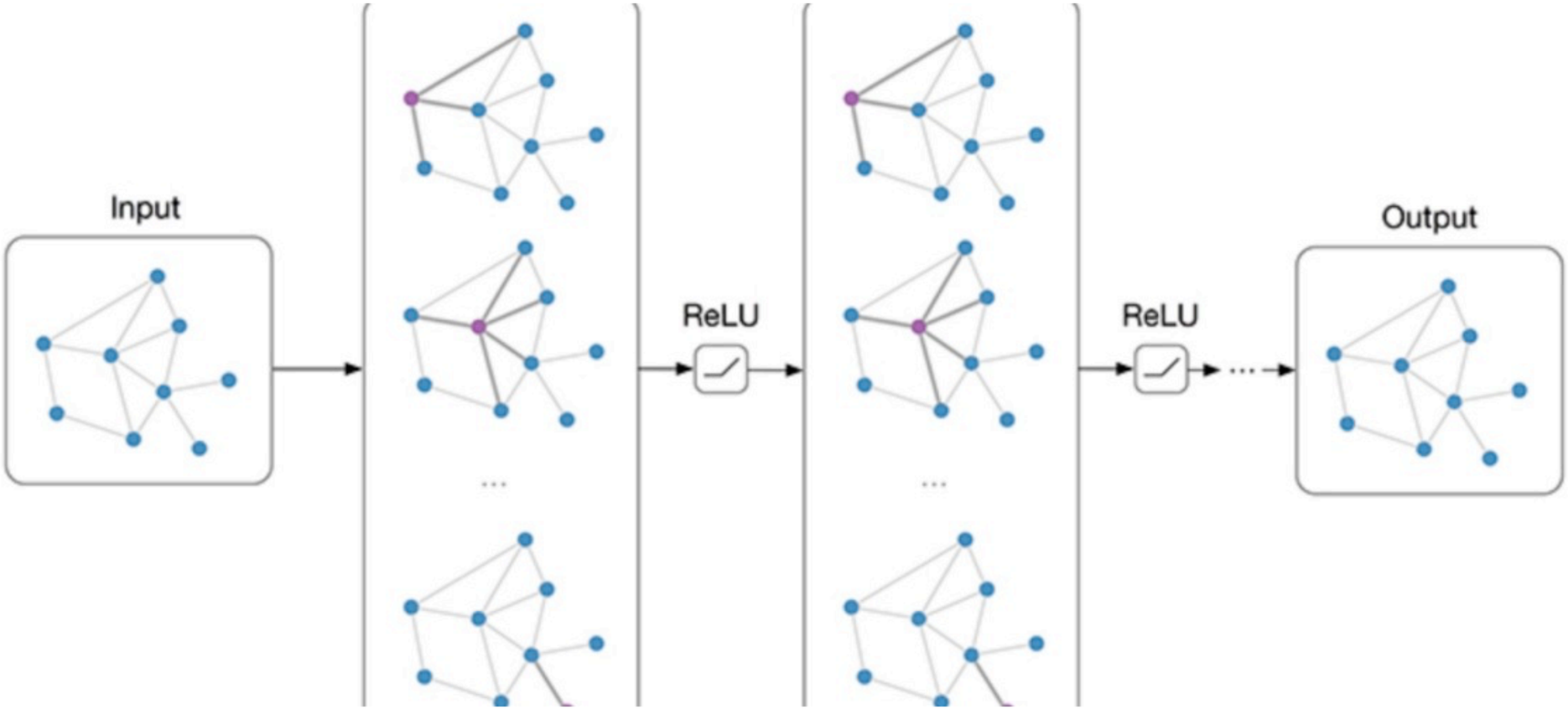


* an extensive experimental analysis of CNNs for sentence classification

https://arxiv.org/pdf/1510.03820.pdf

Zhang, Y., & Wallace, B. (2015). A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification.

Later in this course: Graph Convolution Networks



Summary

* Here comes your summary from today's class

Homework (in Moodle)

* “Ambitions” Homework
* (optional) Apply SSD to process video

Tools for Annotations

* LabelMe: One of the most known tools. The UI was a bit too slow, though, especially when zooming in on large images.
* RectLabel: Simple and easy to work with. Mac only.
* LabelBox: Pretty good for larger labeling projects and has options for different types of labeling tasks.
* VGG Image Annotator (VIA): Fast, light, and really well designed. This is the one I ended up using.
* COCO UI: The tool used to annotate the COCO dataset.

References / Sources

1.https://arxiv.org/pdf/1604.03640.pdf

2.https://habr.com/ru/company/mailru/blog/311706/

3.https://habr.com/ru/post/318970/

4.inception v3: https://josephpcohen.com/w/wp-content/uploads/inception-v3.pdf

5.https://www.cs.toronto.edu/~frossard/post/vgg16/

6.https://medium.com/machine-learning-bites/deeplearning-series-convolutional-neural-networks-a9c2f2ee1524

Videos

* https://youtu.be/Cgxsv1riJhI?t=181
* https://www.youtube.com/watch?v=4eIBisqx9_g
* https://www.youtube.com/watch?v=nDPWywWRIRo&list=WL&index=2&t=0s
* https://deeplizard.com/learn/video/cNBBNAxC8l4