Applied Deep Learning

Dr. Philippe Blaettchen
Bayes Business School (formerly Cass)

www.bayes.city.ac.uk

Learning objectives of today

Goals: Understand how convolution is used to enable different computer vision applications

- Typical network structures in convolutional networks
- Specific adjustments to layers and connections that allow to overcome training and use challenges

How will we do this?

- We first consider image classification and the network architectures that allow to perform the task effectively
- We then turn to transfer learning: how we can use existing network architectures to apply to our own computer vision problems
- Finally, we study some other computer vision problems and the relevant adjustments required in convolutional networks to tackle them

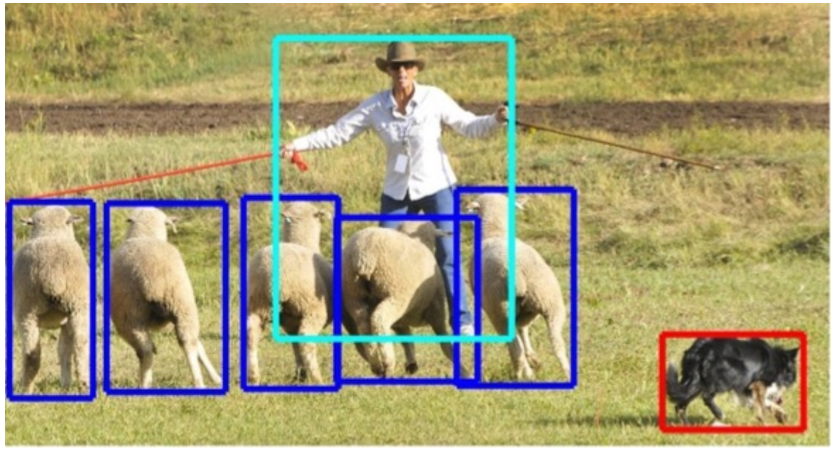
BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

A brief recap of convolutional and pooling layers

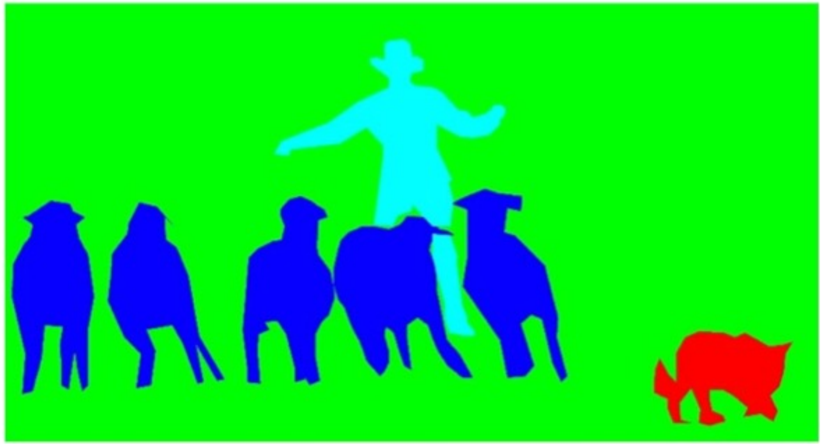
Typical computer vision problems

Image classification

Object detection



Semantic segmentation

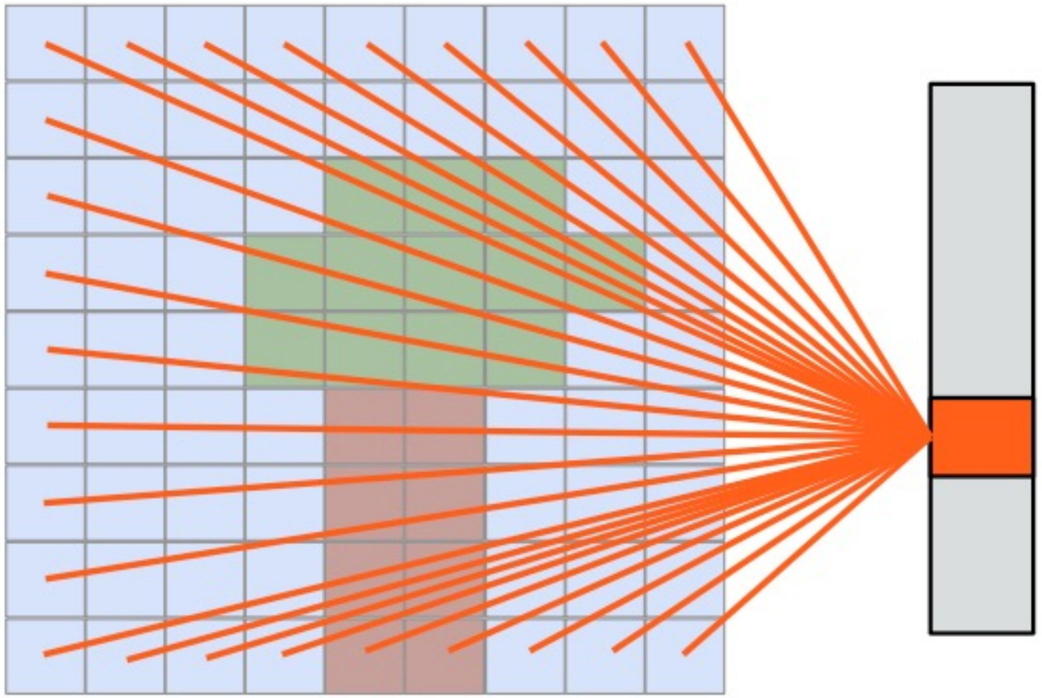


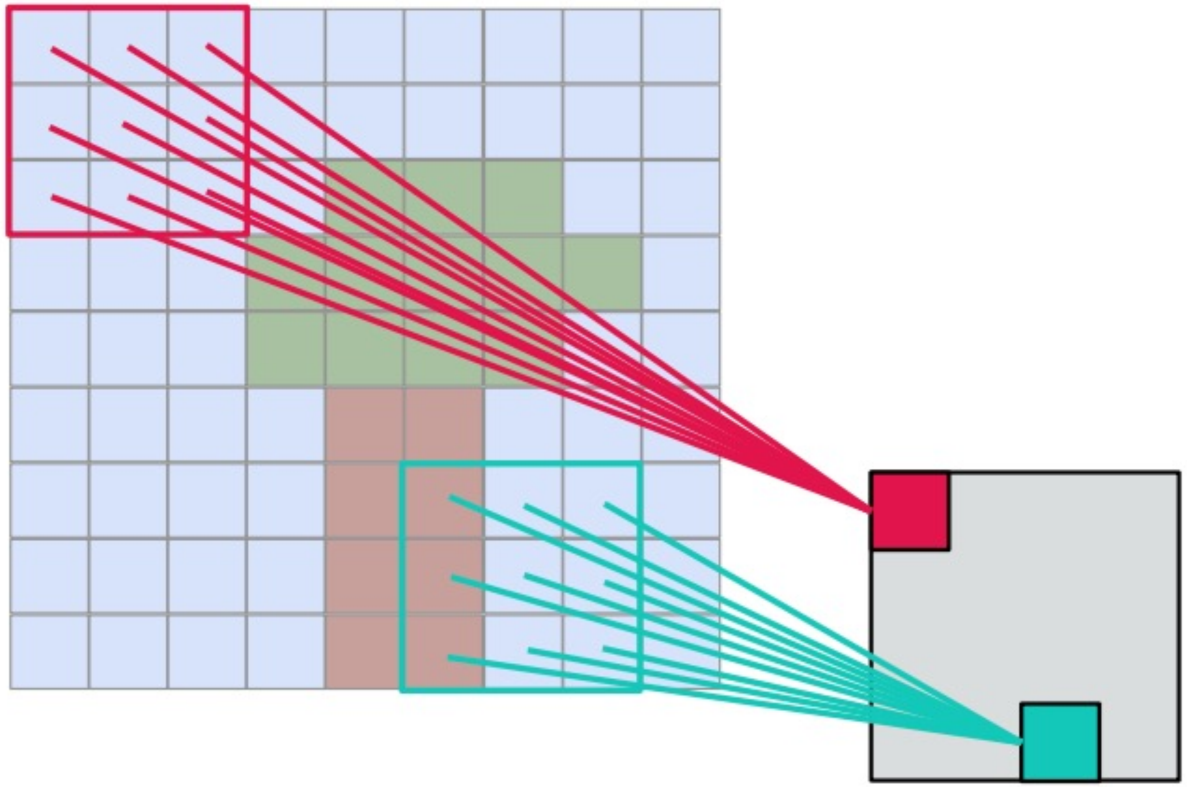
Neural style transfer

Source: Lin, reiinakano.com

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

From fully connected to locally connected





Source: Dieleman

The convolution operator

$$
\begin{bmatrix} 1 & 3 & 1 & 2 \\ 6 & 1 & 5 & 4 \\ 5 & 4 & 2 & 5 \\ 3 & 3 & 1 & 2 \end{bmatrix}
*
\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}
=
\begin{bmatrix} 20 & & \\ & & \\ & & \end{bmatrix}
$$

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The convolution operator

$$\begin{bmatrix}1&3&1&2\\6&1&5&4\\5&4&2&5\\3&3&1&2\end{bmatrix} * \begin{bmatrix}1&2\\2&1\end{bmatrix} = \begin{bmatrix}20&12&\\ & & \\ & & \end{bmatrix}$$

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The convolution operator

$$
\begin{bmatrix} 1 & 3 & 1 & 2 \\ 6 & 1 & 5 & 4 \\ 5 & 4 & 2 & 5 \\ 3 & 3 & 1 & 2 \end{bmatrix}
*
\begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}
=
\begin{bmatrix} 20 & 12 & 19 \\ & & \\ & & \end{bmatrix}
$$

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The convolution operator

$$
\begin{array}{|c|c|c|c|}
\hline
1 & 3 & 1 & 2 \\
\hline
6 & 1 & 5 & 4 \\
\hline
5 & 4 & 2 & 5 \\
\hline
3 & 3 & 1 & 2 \\
\hline
\end{array}
\quad * \quad
\begin{array}{|c|c|}
\hline
1 & 2 \\
\hline
2 & 1 \\
\hline
\end{array}
\quad = \quad
\begin{array}{|c|c|c|}
\hline
20 & 12 & 19 \\
\hline
22 & & \\
\hline
 & & \\
\hline
\end{array}
$$

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The convolution operator

$$
\begin{array}{|c|c|c|c|}
\hline
1 & 3 & 1 & 2 \\ \hline
6 & 1 & 5 & 4 \\ \hline
5 & 4 & 2 & 5 \\ \hline
3 & 3 & 1 & 2 \\ \hline
\end{array}
\;*\;
\begin{array}{|c|c|}
\hline
1 & 2 \\ \hline
2 & 1 \\ \hline
\end{array}
\;=\;
\begin{array}{|c|c|c|}
\hline
20 & 12 & 19 \\ \hline
22 & 21 & \\ \hline
 & & \\ \hline
\end{array}
$$

The convolution operator

1	3	1	2
6	1	5	4
5	4	2	5
3	3	1	2

$*$

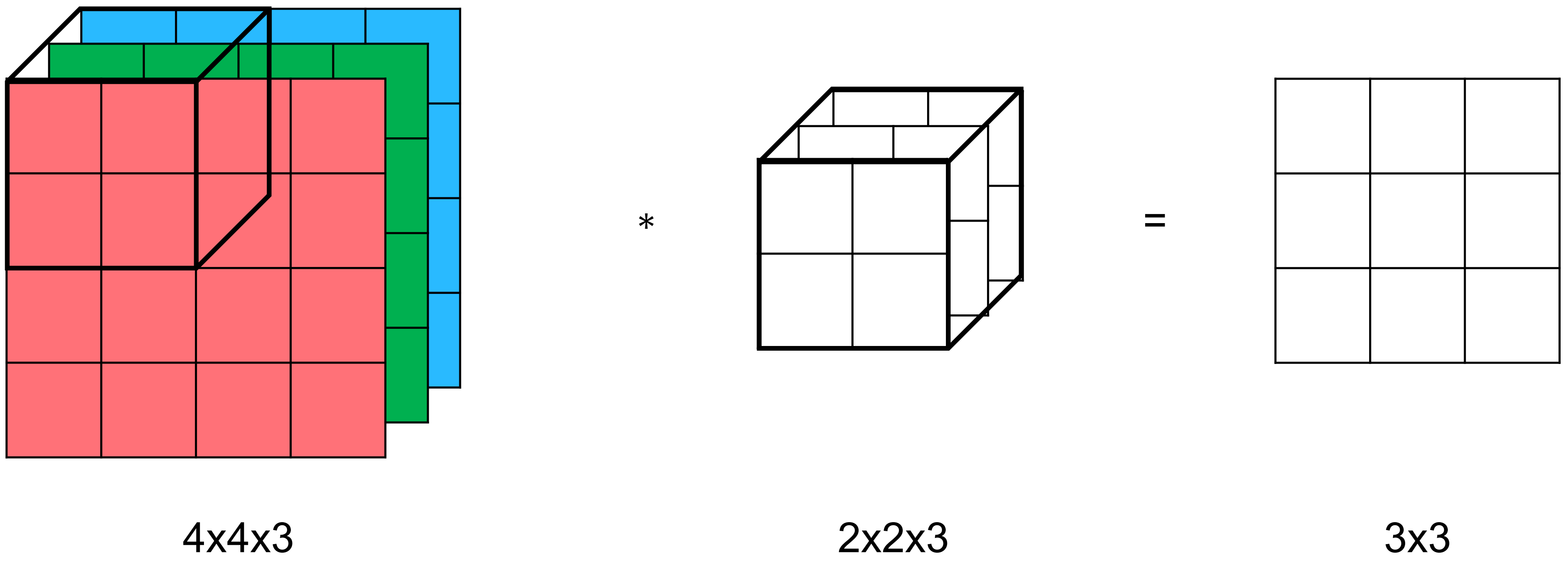
1	2
2	1

$=$

20	12	19
22	21	22
22	15	16

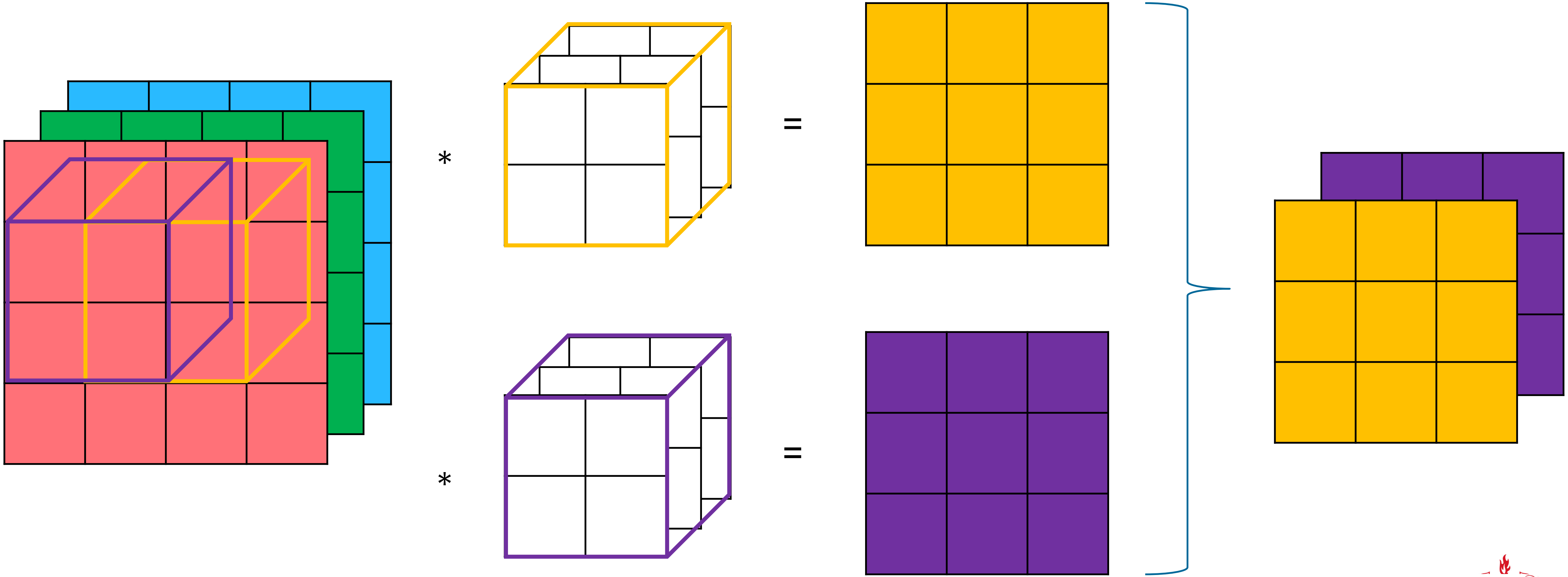
BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Convolution on a 3D array



BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Multiple 3D convolutions



4x4x3 $(n_H^{[0]}, n_W^{[0]}, \mathrm{n}_C^{[0]})$

2x2x3 $(f_H, f_W, \mathrm{n}_C^{[0]})$

3x3 $(n_H^{[1]}, n_W^{[1]}, \mathrm{n}_C^{[1]})$

$n_C^{[1]}$= number of filters

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Max pooling in practice

$$\begin{bmatrix} 1 & 3 & 1 & 2 \\ 6 & 1 & 5 & 4 \\ 5 & 4 & 2 & 5 \\ 3 & 3 & 1 & 2 \end{bmatrix} * \begin{bmatrix} \; & \; \\ \; & \; \end{bmatrix} = \begin{bmatrix} 6 & \; & \; \\ \; & \; & \; \\ \; & \; & \; \end{bmatrix}$$

Source: Géron

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Max pooling in practice

1	3	1	2
6	1	5	4
5	4	2	5
3	3	1	2

*

=

6	5	

Source: Géron

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Max pooling in practice

$$
\begin{bmatrix} 1 & 3 & 1 & 2 \\ 6 & 1 & 5 & 4 \\ 5 & 4 & 2 & 5 \\ 3 & 3 & 1 & 2 \end{bmatrix}
*
\begin{bmatrix} & \\ & \end{bmatrix}
=
\begin{bmatrix} 6 & 5 & 5 \\ & & \\ & & \end{bmatrix}
$$

Source: Géron

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Max pooling in practice

1	3	1	2
6	1	5	4
5	4	2	5
3	3	1	2

*

=

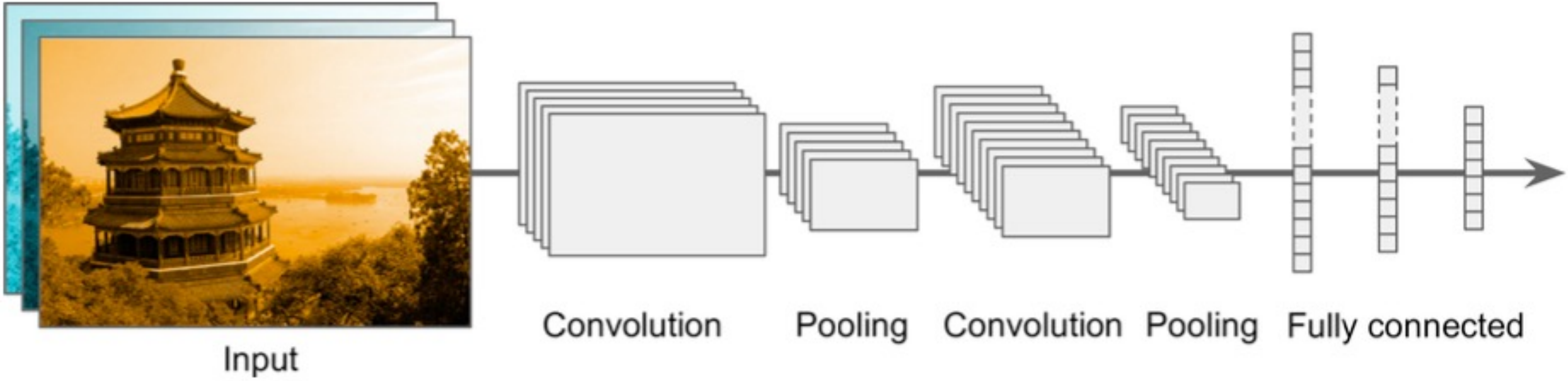
6	5	5
6	5	5
5	4	5

Source: Géron

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Typical architectures

Typical architecture



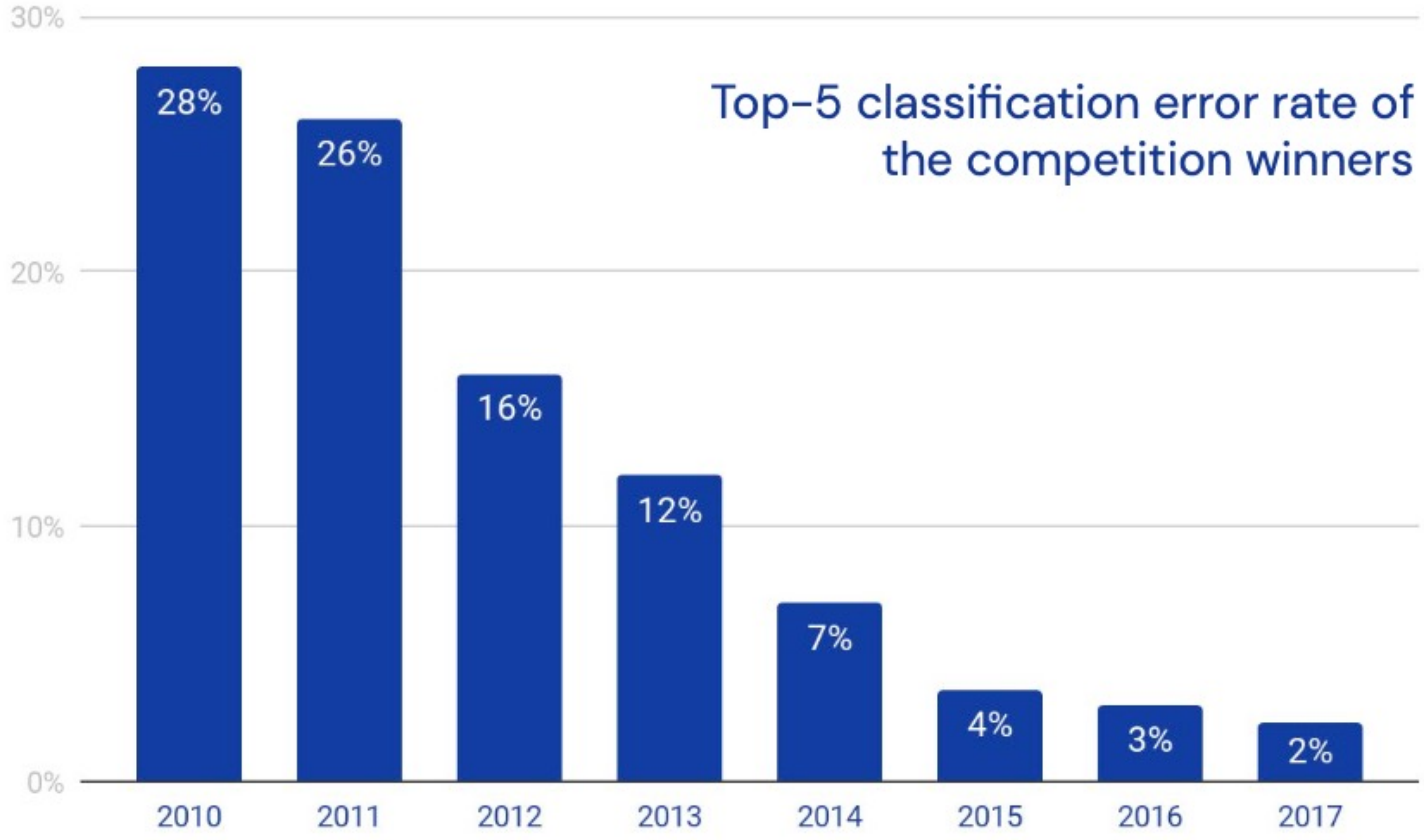
Source: Géron

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

The ImageNet challenge

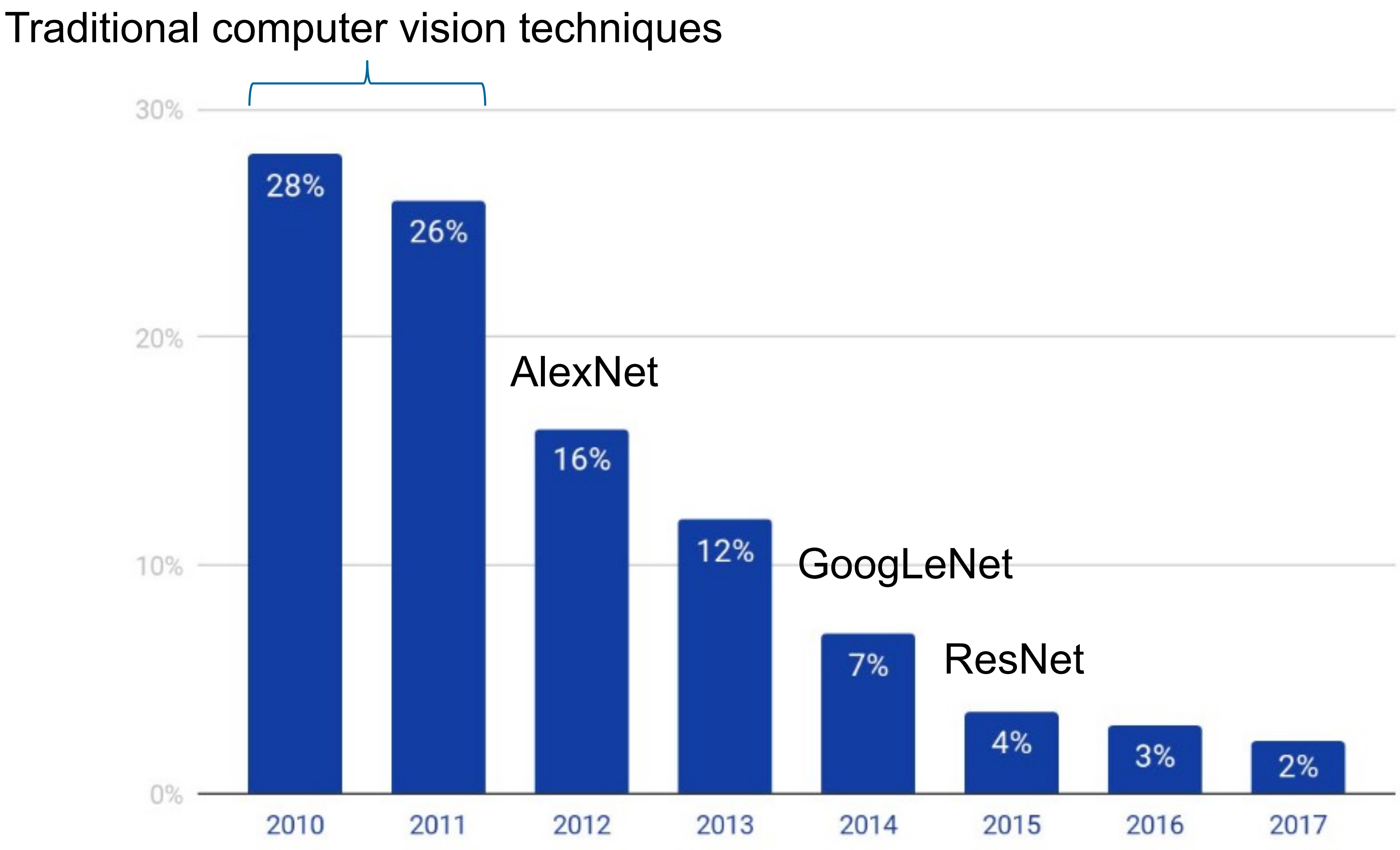
- Major computer vision benchmark for image classification (and later, more advanced stuff)
- From 2010-2017 (now transferred to Kaggle)
- 1.4 mio images in 1,000 classes
- Models need to predict the top 5 most likely labels
    - Winner: lowest “top-5 error rate” – percentage of test images for which true label is not among top 5 most likely labels
- More information: https://www.image-net.org/challenges/LSVRC/index.php

Architectures over time



Source: Dieleman

Architectures over time

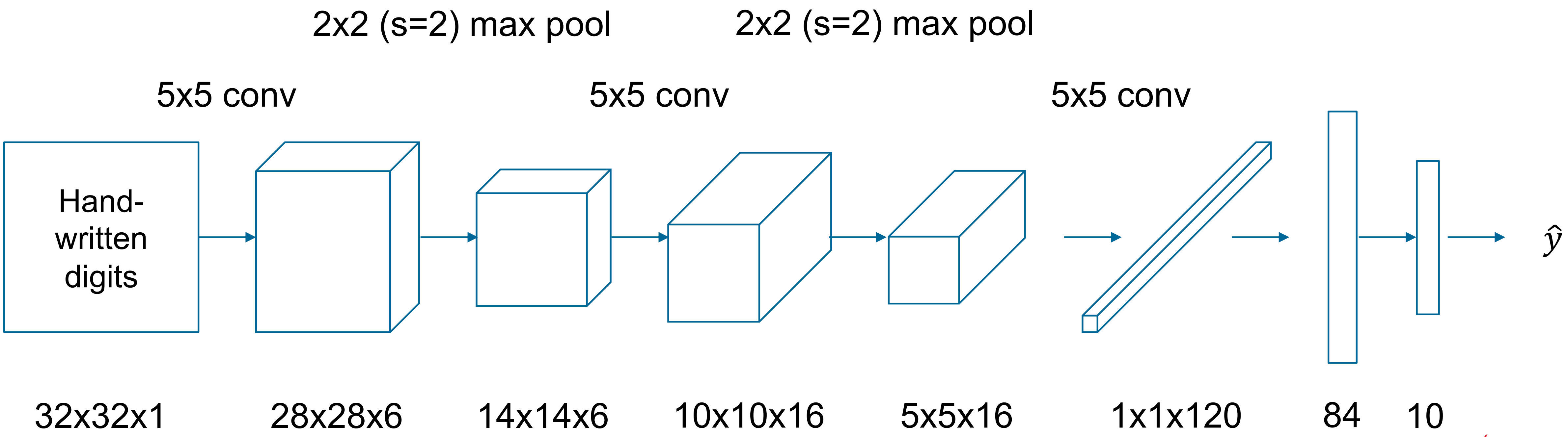


Source: Dieleman

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

LeNet-5

- Created by Yann LeCun, 1998
- Widely used for handwritten digit recognition (e.g., bank cheques)



- Uses mostly tanh
- See http://yann.lecun.com/exdb/lenet/index.html for demos

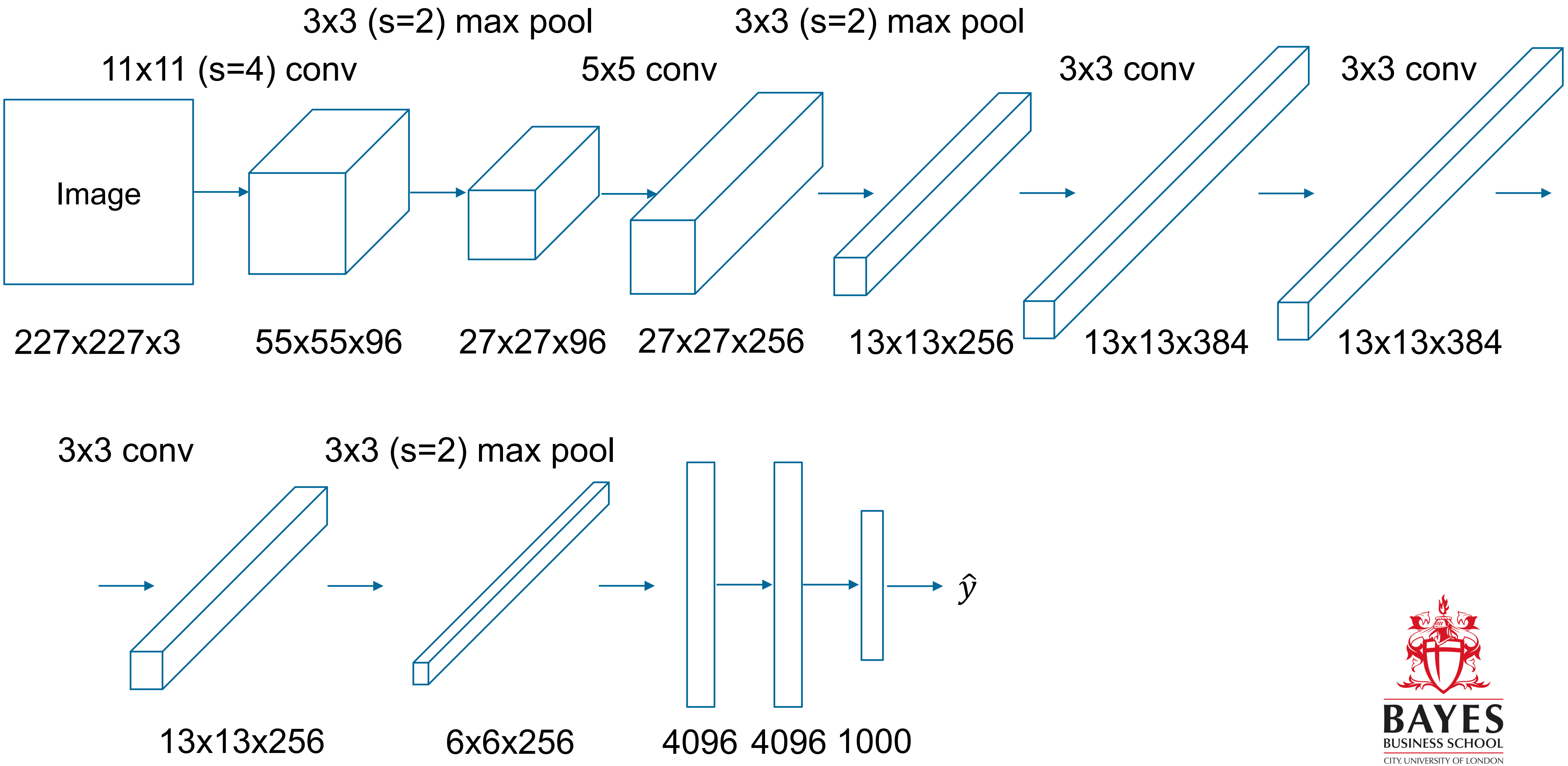
BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

AlexNet

- Winner of the 2012 challenge, by Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton
- Similar to LeNet, but much deeper
- Adds multiple convolutional layer before a pooling layer
- Using ReLU
- Regularization with dropout on the final two layers + data augmentation

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

AlexNet



3x3 (s=2) max pool
3x3 (s=2) max pool
11x11 (s=4) conv
5x5 conv
3x3 conv
3x3 conv
Image
227x227x3
55x55x96
27x27x96
27x27x256
13x13x256
13x13x384
13x13x384
3x3 conv
3x3 (s=2) max pool
$\hat{y}$
13x13x256
6x6x256
4096
4096
1000

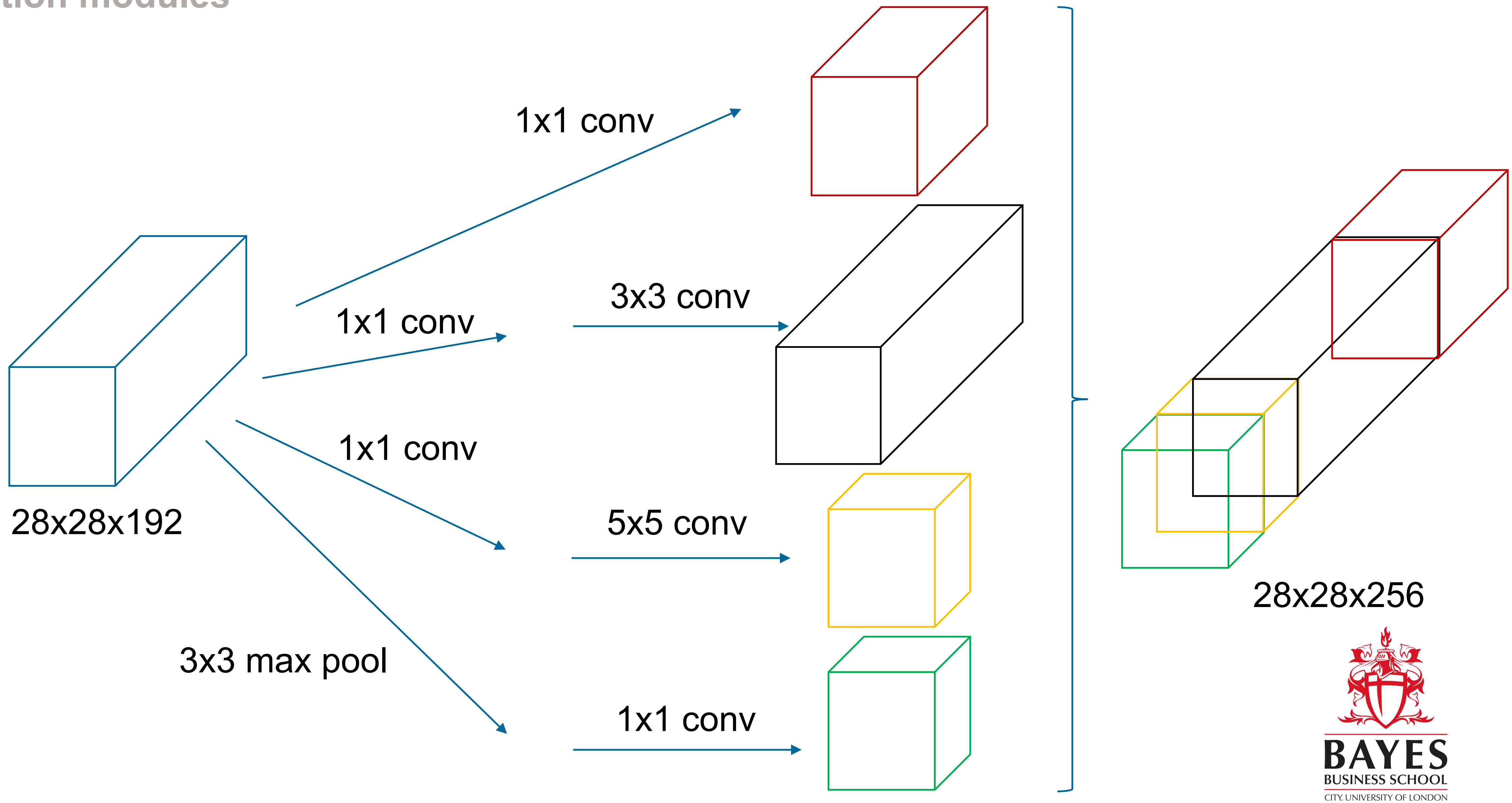
Inception modules

- General idea: to achieve higher performance networks are being made deeper and deeper, with negative effects on computation and overfitting
- Would be more effective to make the network sparser, but numerical computations slow
- Inception modules as a way to trade-off: exploit sparsity, but also current hardware
- The module name is actually inspired by this:

Source: knowyourmeme.com

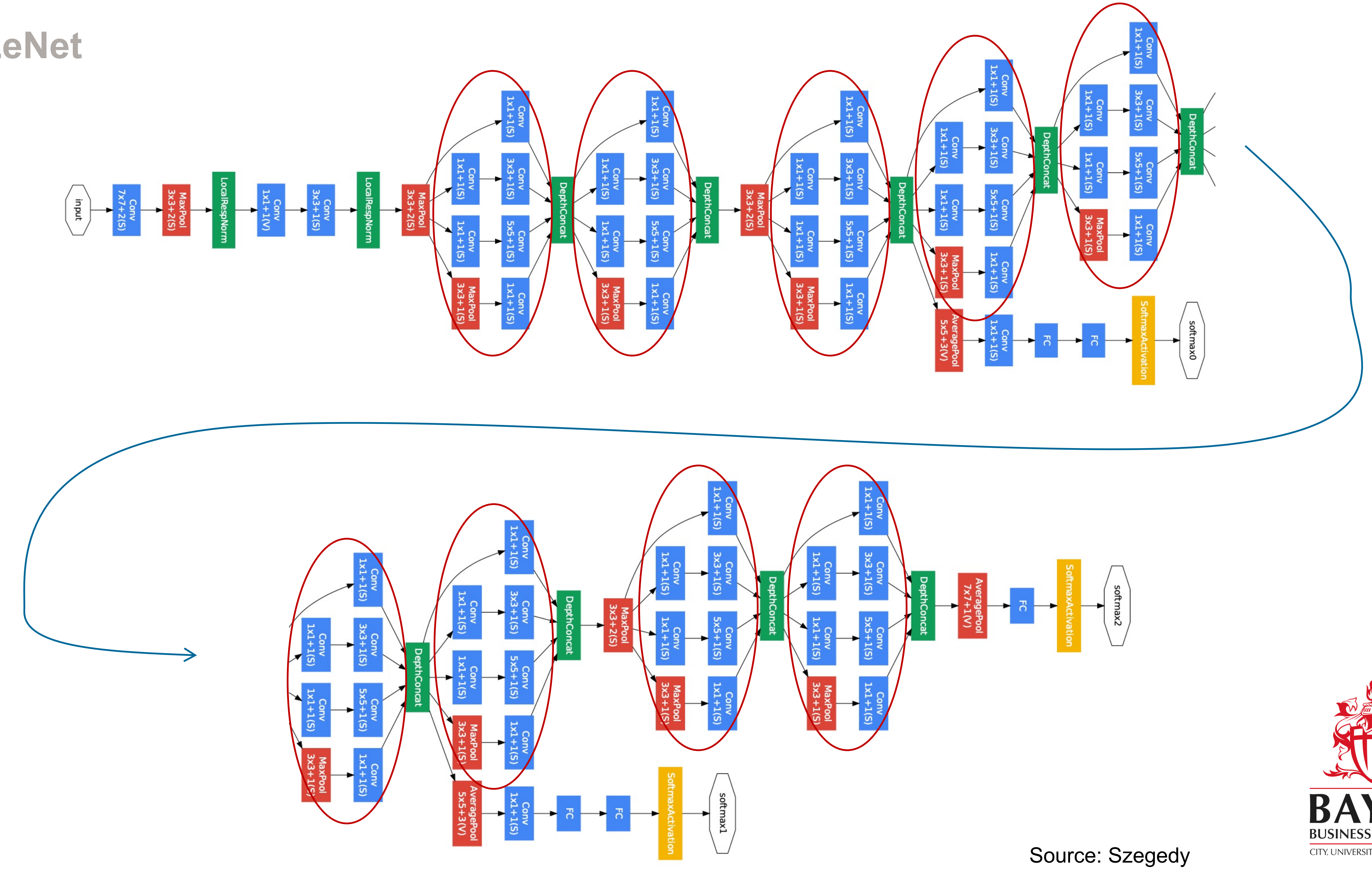
BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Inception modules



BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

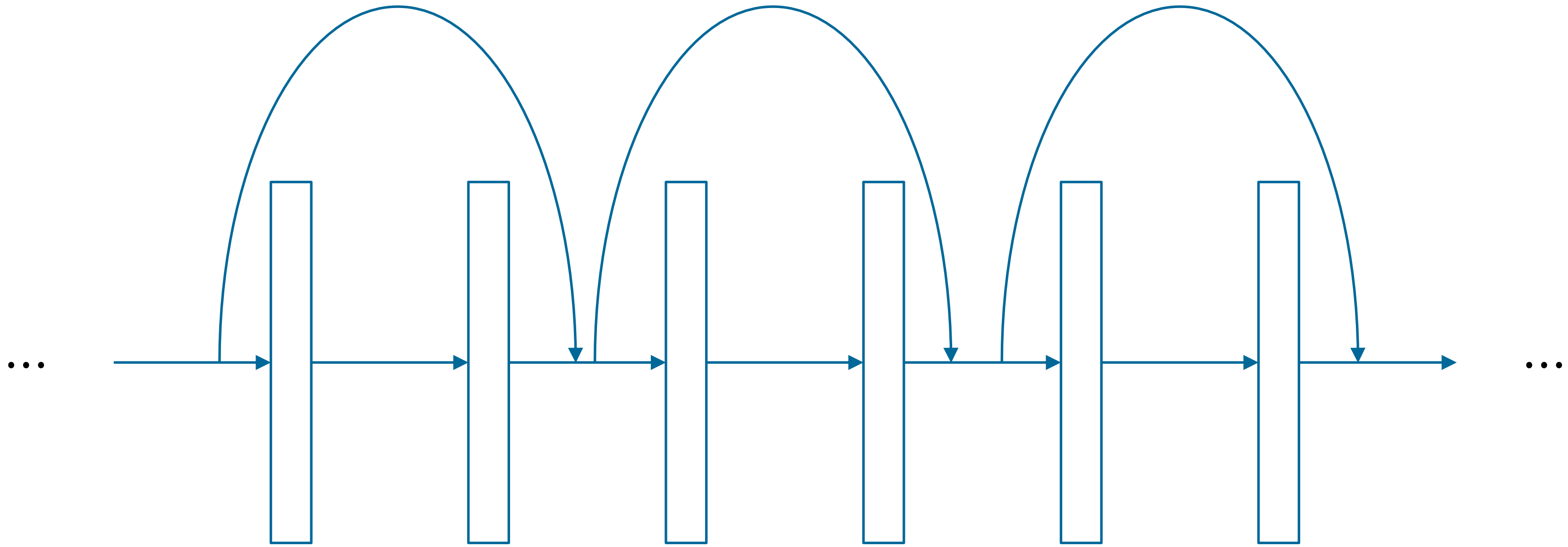
GoogLeNet



Source: Szegedy

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

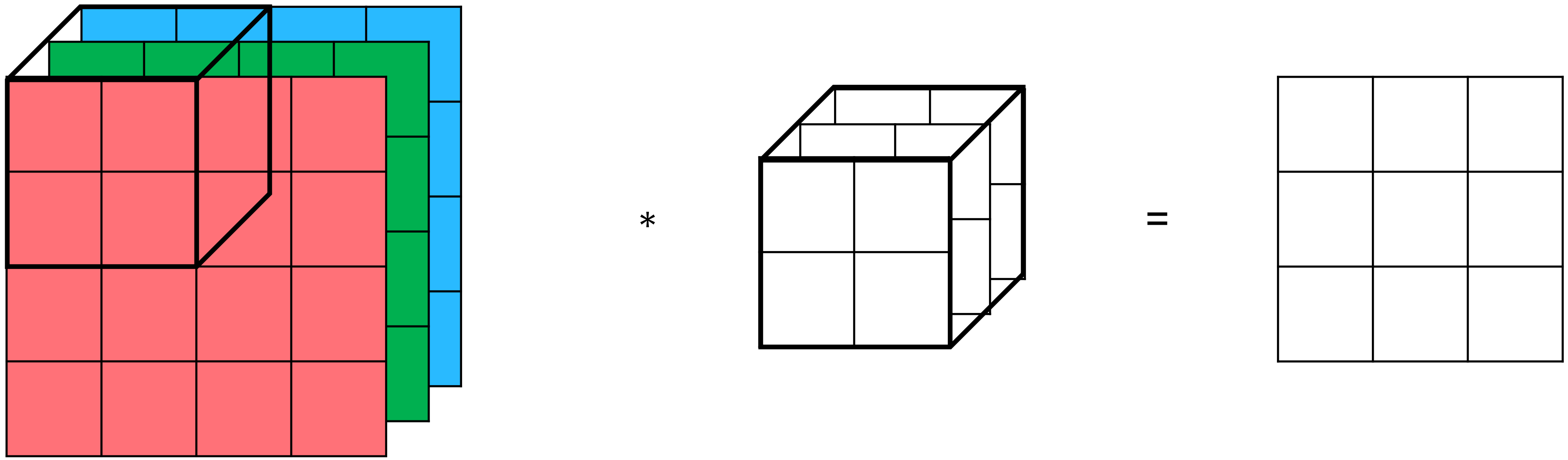
ResNets – using skip connections



MobileNets

- Idea: network that is fast to use even with limited hardware (e.g., mobile phone)
  - Note: training can still be long, as the phone will use a pre-trained model

- “Normal” convolution:

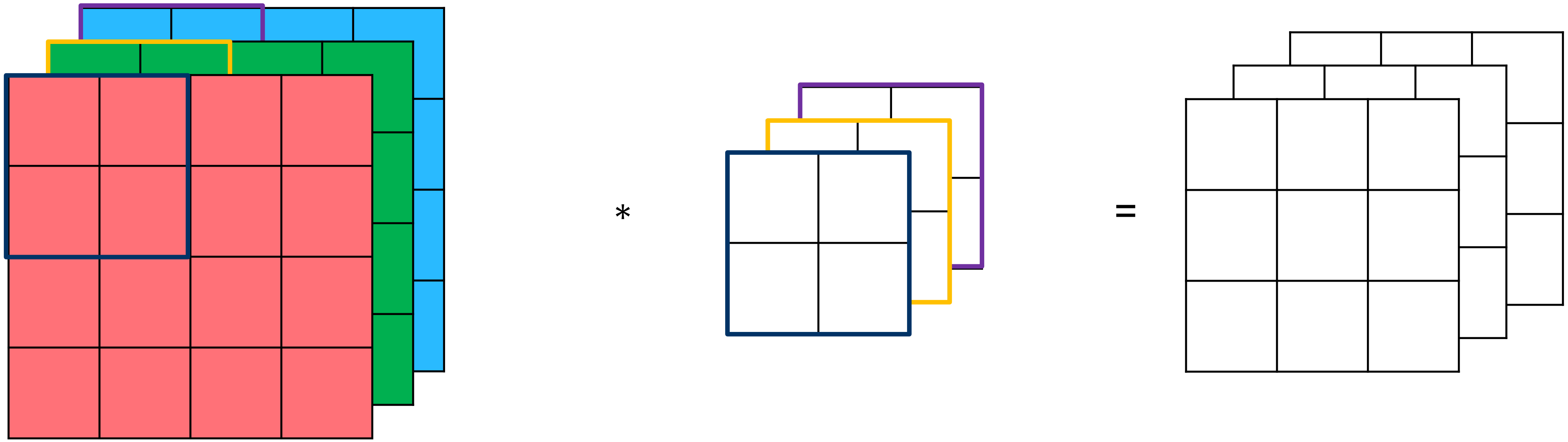


4x4x3 * 2x2x3 = 3x3

MobileNets

- Idea: network that is fast to use even with limited hardware (e.g., mobile phone)
  - Note: training can still be long, as the phone will use a pre-trained model

- “Depthwise separable” convolution:



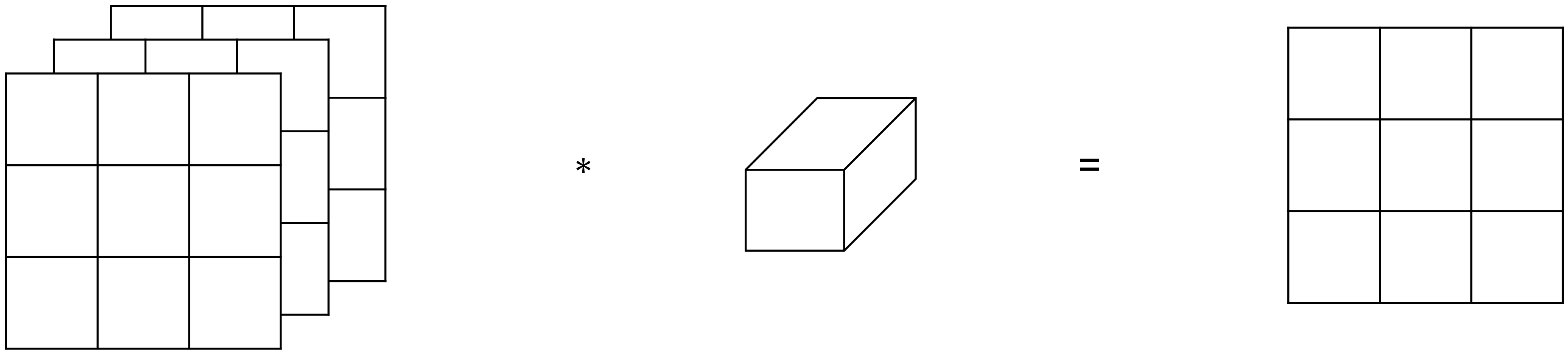
4x4x3 | 3 Filters: 2x2x1 | 3 outputs: 3x3

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

MobileNets

- Idea: network that is fast to use even with limited hardware (e.g., mobile phone)
    - Note: training can still be long, as the phone will use a pre-trained model

- “Depthwise separable” convolution:



3 outputs: 3x3 1x1x3 3x3

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Transfer learning

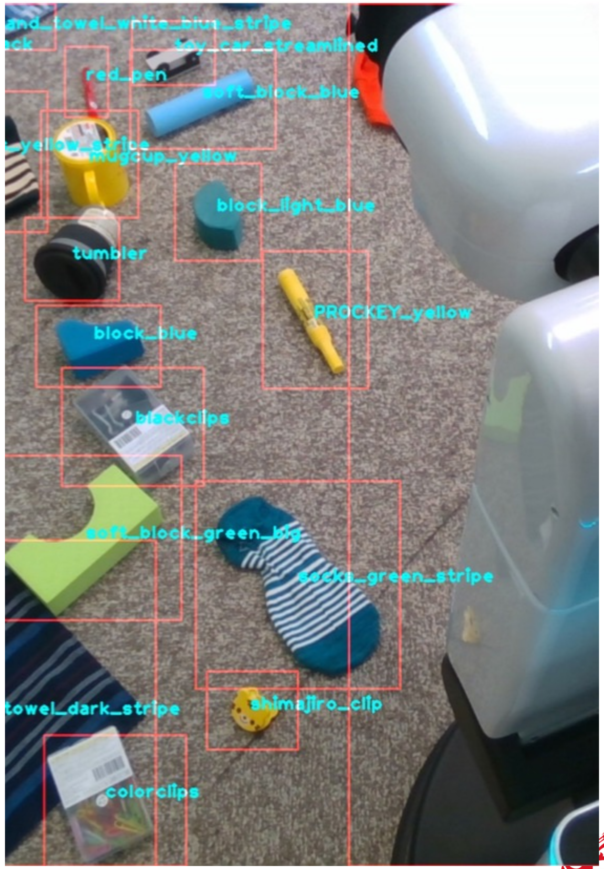
Transfer learning

- Say you want to create a neural network, which your cleaning robot can use to classify objects on the floor

- Instead of developing a new neural network, you decide to use a winner of an ImageNet competition. After all, they are pretty good at classifying many different objects

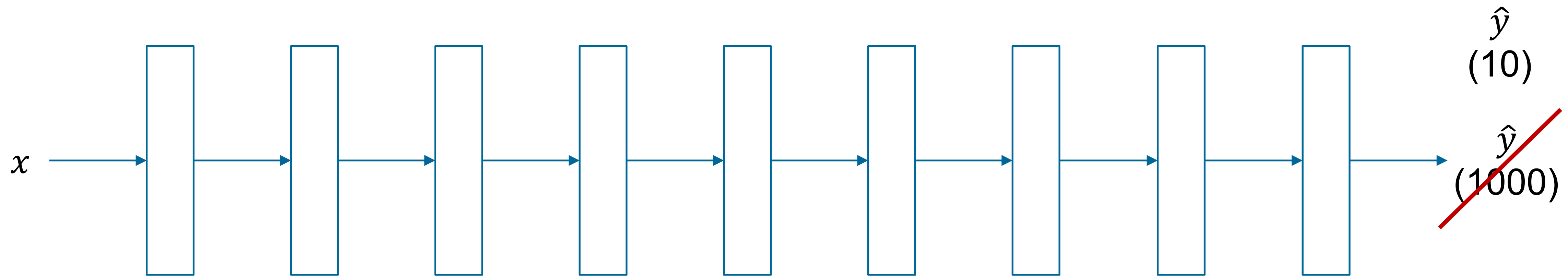
- Going through the list of objects in ImageNet, you realize there are no classes capturing dirty socks or similar items

- But you believe that such items, while not contained in the original network’s training, share the same low-level features as other items that are found there



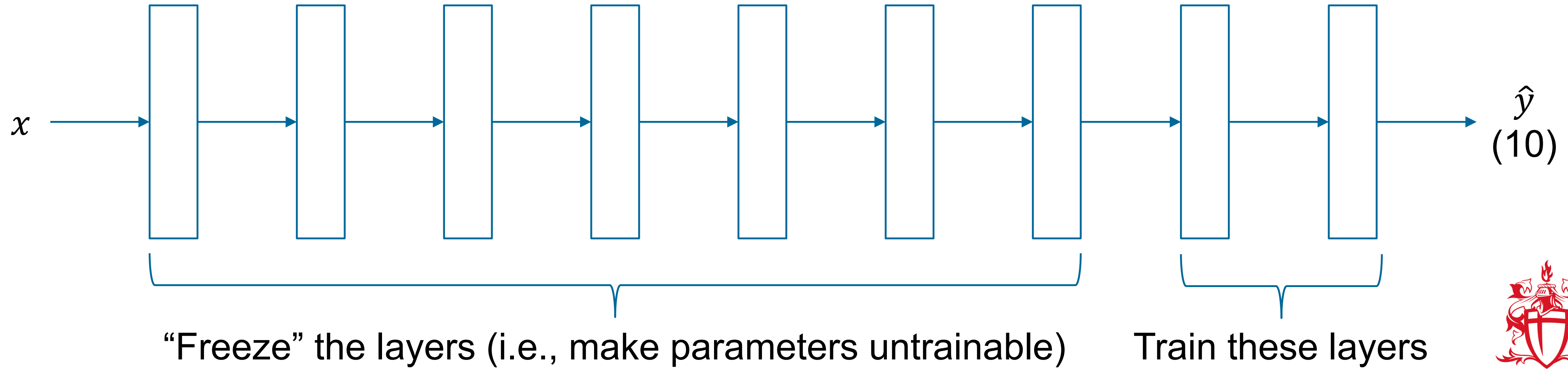
Repurposing a neural network

- Naïve approach: take the existing (trained) neural network
- Adjust the output layer
- Train some more with your data set



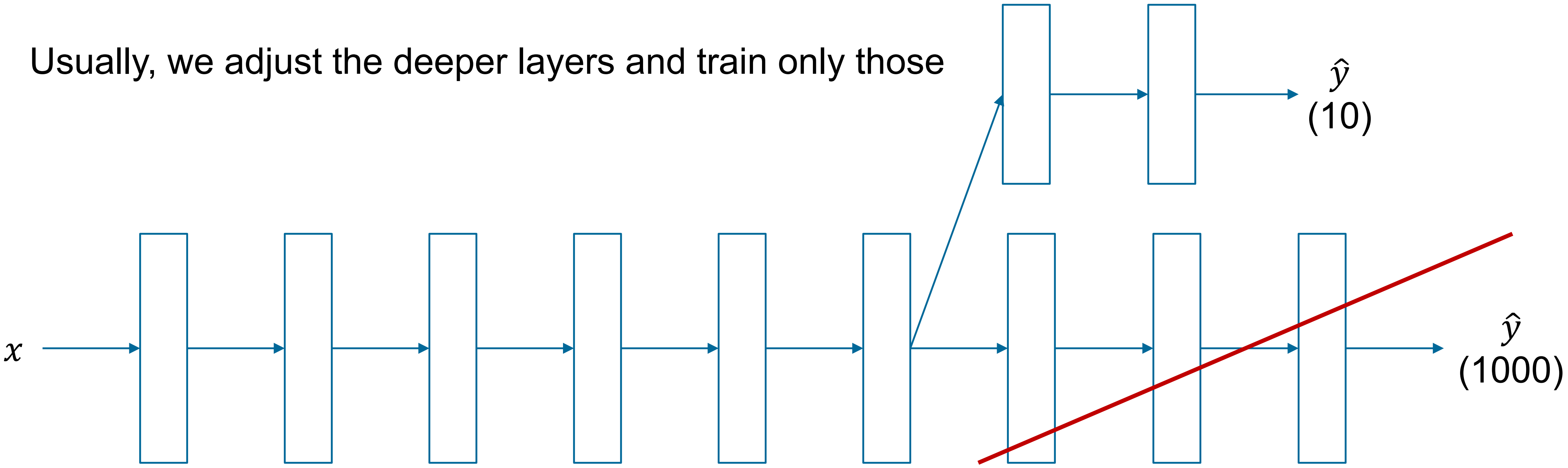
Difference between low-level and high-level features

- Problem with the previous approach: training may be very slow
- But: early layers capture low-level features that are unlikely to be different
- Deeper layers capture high-level features that are likely to be different



Difference between low-level and high-level features

- We can go further, by adjusting some of the layers to fit better with our context
- Usually, we adjust the deeper layers and train only those

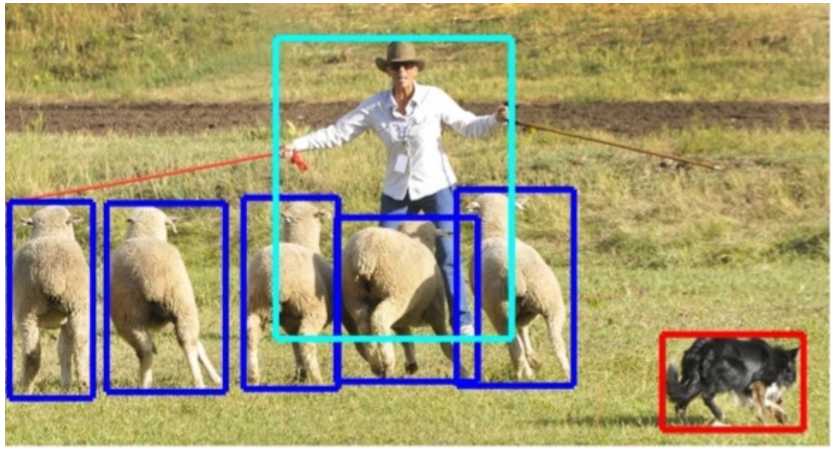


Beyond classification – object detection

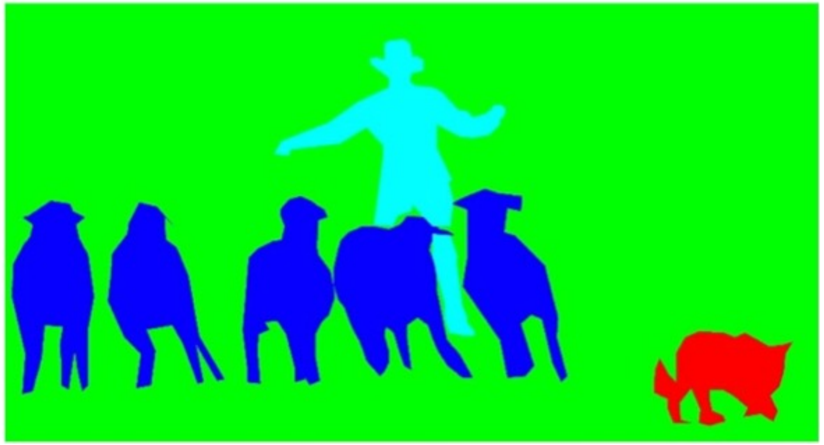
Typical computer vision problems

Image classification

Object detection



Semantic segmentation

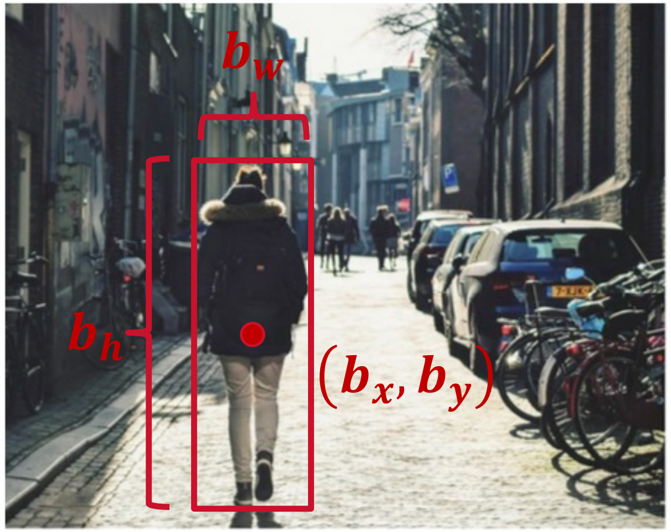


Neural style transfer

Source: Lin, reiinakano.com

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Before detection: classification + localization



A CNN

Softmax

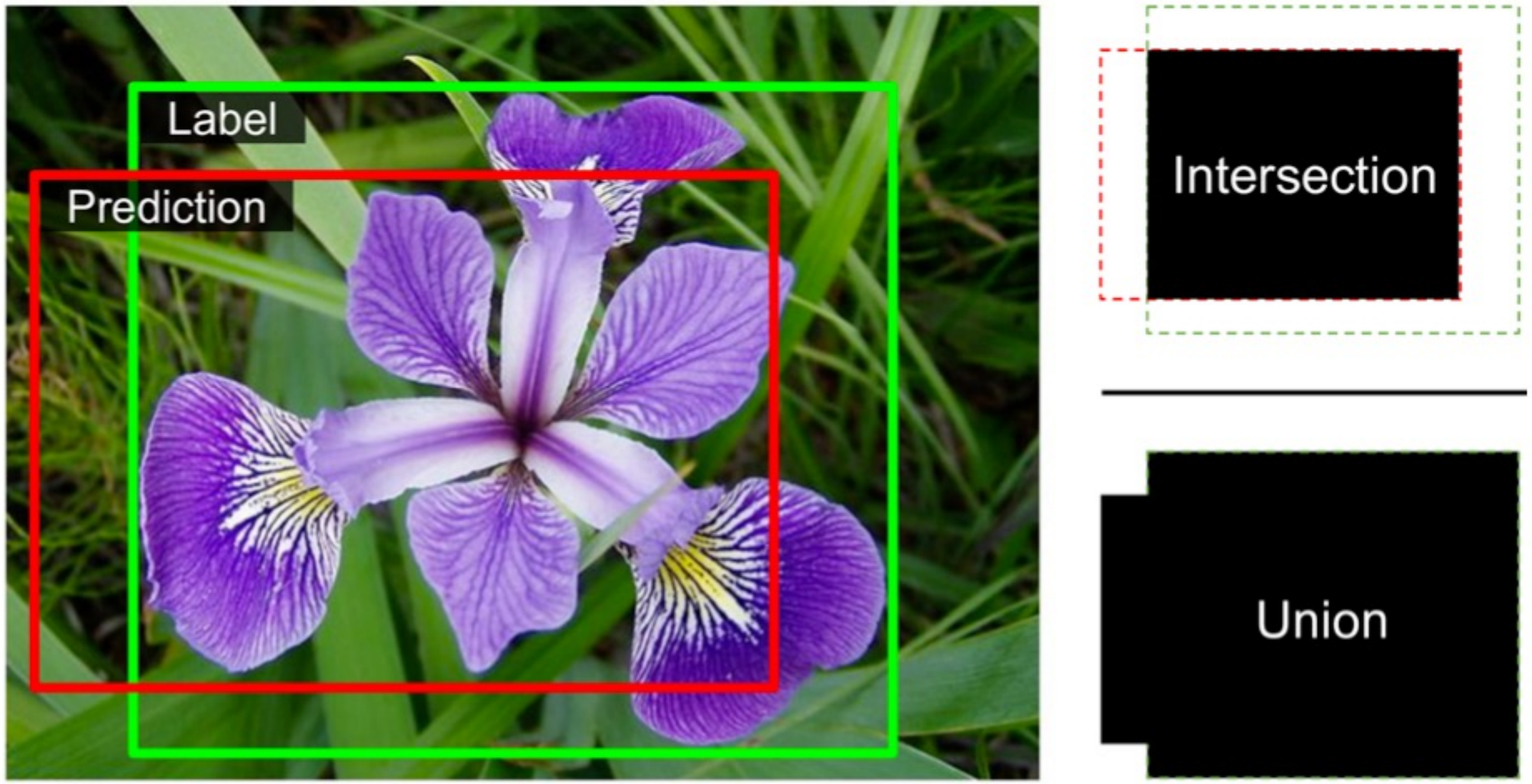
$$\hat{y}_1 = \begin{cases} 1 - person \\ 2 - car \\ 3 - background \end{cases}$$

Multi-Regression

$$\hat{y}_2 = \begin{pmatrix} b_x \\ b_y \\ b_w \\ b_h \end{pmatrix}$$

Source: Géron

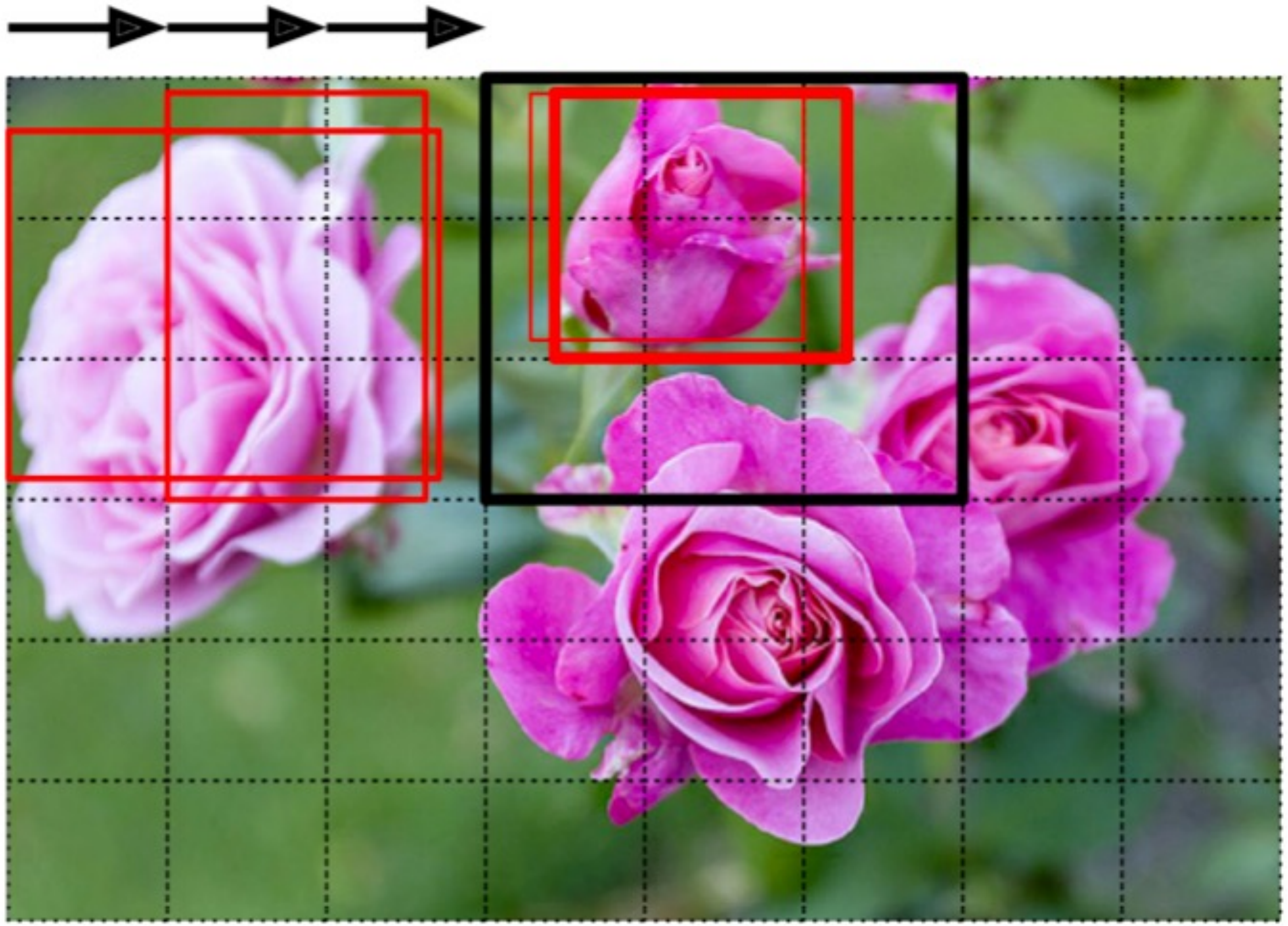
Learning bounding boxes



Intersection over Union (IoU) as the standard metric for bounding boxes
E.g., if IoU ≥ 0.6, consider the bounding box as correctly predicted

Source: Géron

Object detection with sliding windows



Source: Géron

Fully convolutional networks and YOLO

- FCN: Replace dense layers at the top of CNN by convolutional layers
  - instead of having to process parts of the image, the whole image will be processed at once
  - each cell of the final convolutional layer contains the output corresponding to one part (e.g., probability of object, class probabilities, bounding box coordinates)

- YOLO – You Only Look Once
  - Five bounding boxes per grid, each with a probability of containing an object and 20 box-independent class probabilities
  - Predict bounding box coordinates relative to grid cell positions
  - Use five representative “anchor boxes” based on training set, and only predicts how actual bounding boxes have to be rescaled relatively
  - Extremely fast: https://www.youtube.com/watch?v=MPU2HistivI

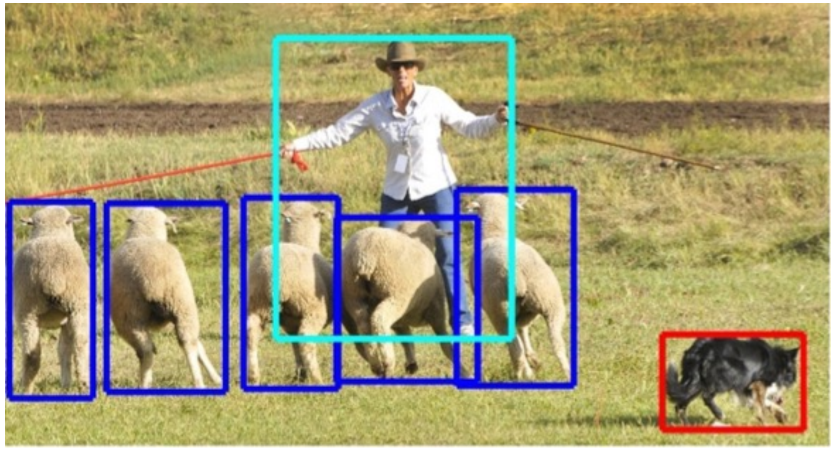
BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Beyond classification – semantic segmentation

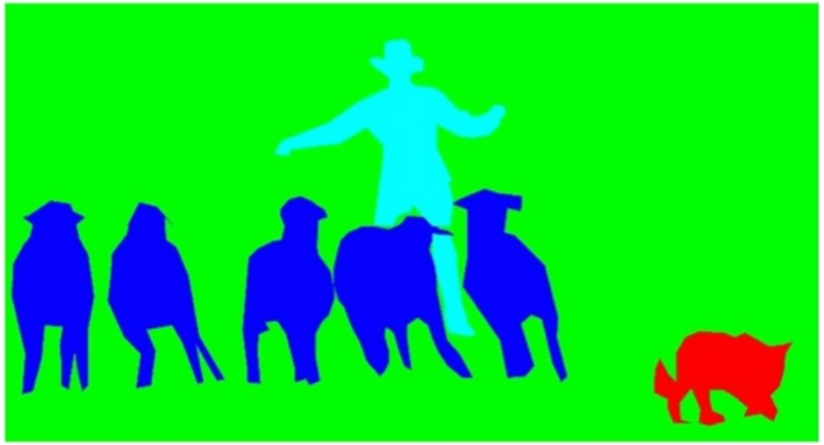
Typical computer vision problems

Image classification

Object detection



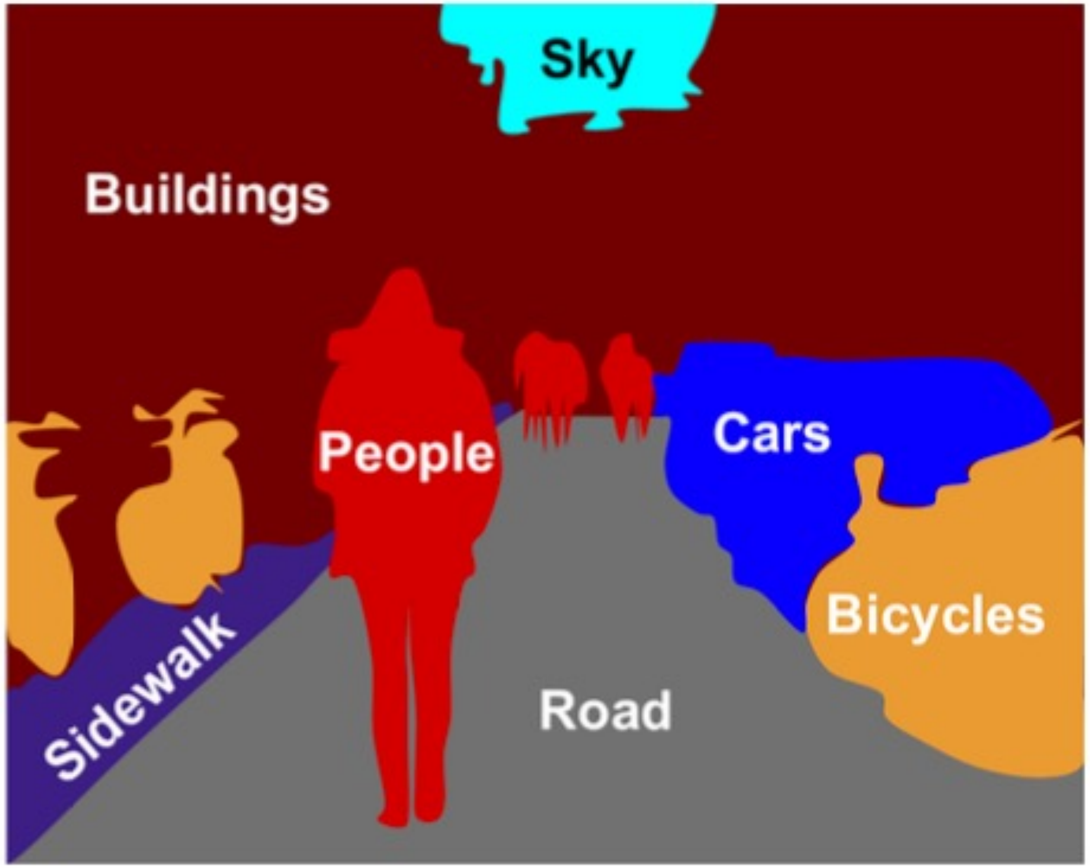
Semantic segmentation



Neural style transfer

Source: Lin, reiinakano.com

Objective: classify each pixel



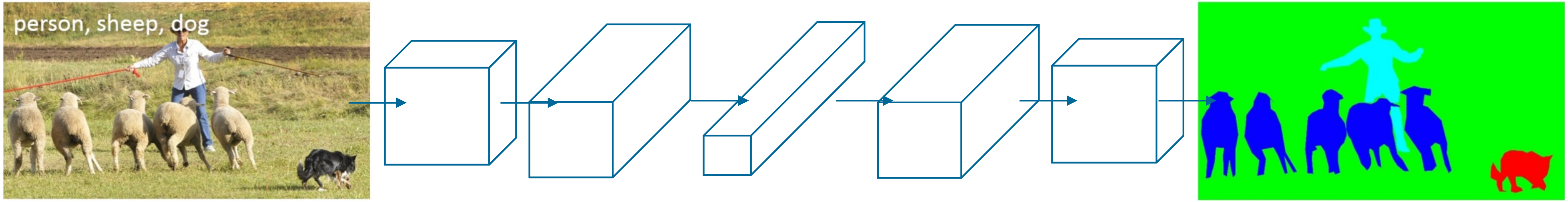
Source: Géron

Semantic segmentation versus object detection

- More accurate, as we don’t rely on (rectangular) bounding boxes

- For training and testing, we need images where every pixel is labeled
  - there are some tools that help, but this is still more tedious than drawing bounding boxes

The general approach



BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Upsampling with transpose convolutions

$$\begin{bmatrix}1&3&1&2\\6&1&5&4\\5&4&2&5\\3&3&1&2\end{bmatrix} * \begin{bmatrix}1&2\\2&1\end{bmatrix} = \begin{bmatrix}20&12&19\\22&21&22\\22&15&16\end{bmatrix}$$

$$\begin{bmatrix}1&3&1\\6&1&5\\5&4&2\end{bmatrix} *^T \begin{bmatrix}1&2\\2&1\end{bmatrix} = \begin{bmatrix}1&2&3&6&1&2\\2&1&6&3&2&1\\6&12&1&2&5&10\\12&6&2&1&10&5\\5&10&4&8&2&4\\10&5&8&4&4&2\end{bmatrix}$$

Transpose convolution with stride 2

Upsampling with transpose convolutions

	1		3		1	
	6		1		5	
	5		4		2	

*

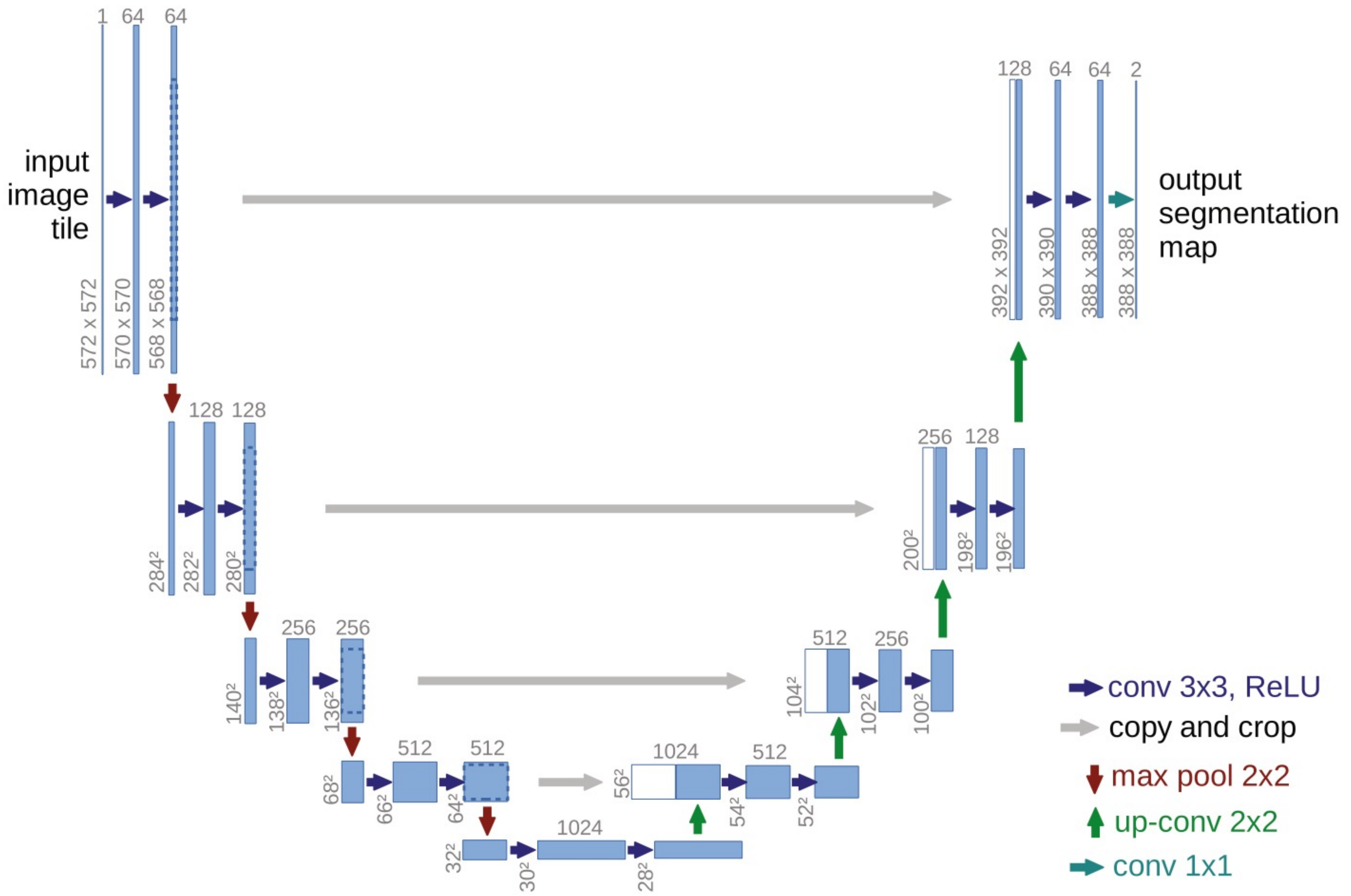
1	2
2	1

=

1	2	3	6	1	2
2	1	6	3	2	1
6	12	1	2	5	10
12	6	2	1	10	5
5	10	4	8	2	4
10	5	8	4	4	2

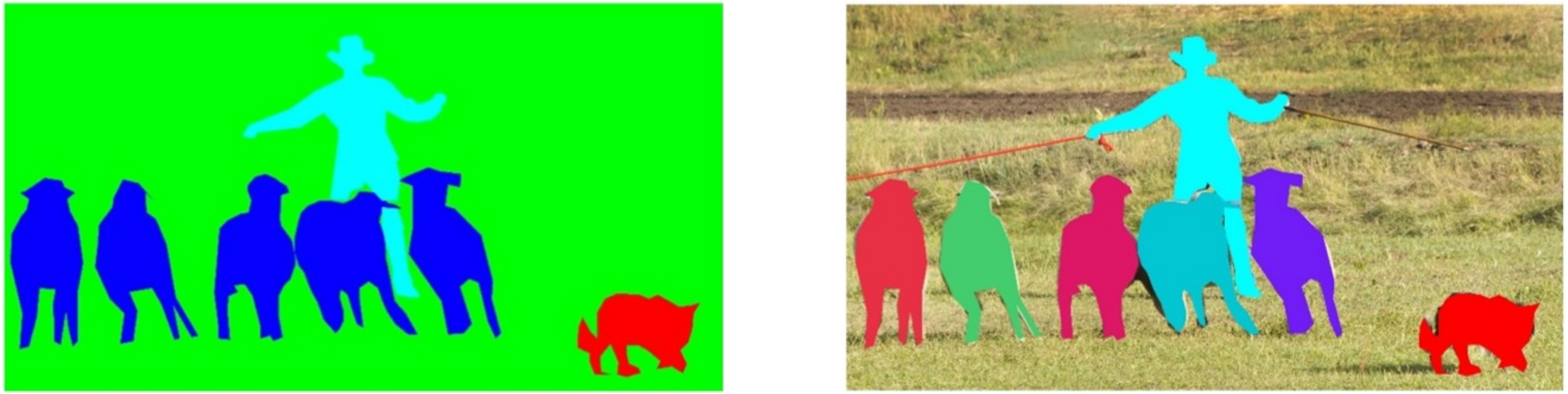
BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

U-Net



Source: Ronneberge

Going further: instance segmentation



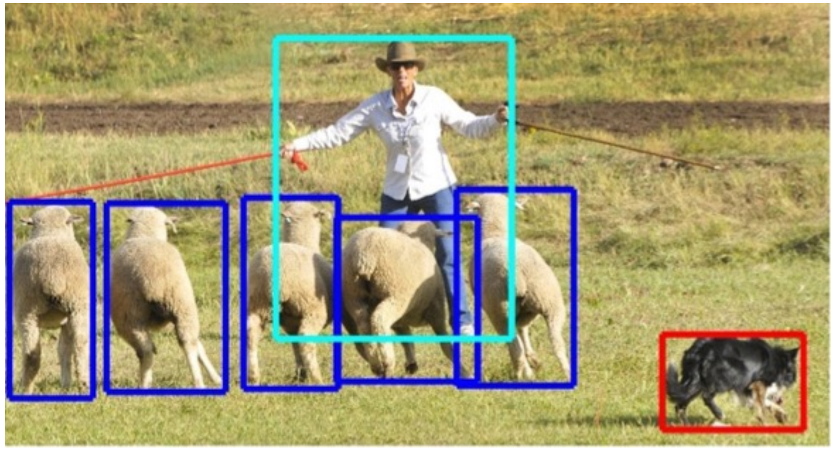
See Lin et al., 2014, Microsoft COCO: Common Objects in Context:
https://arxiv.org/pdf/1405.0312.pdf

Beyond classification – neural style transfer

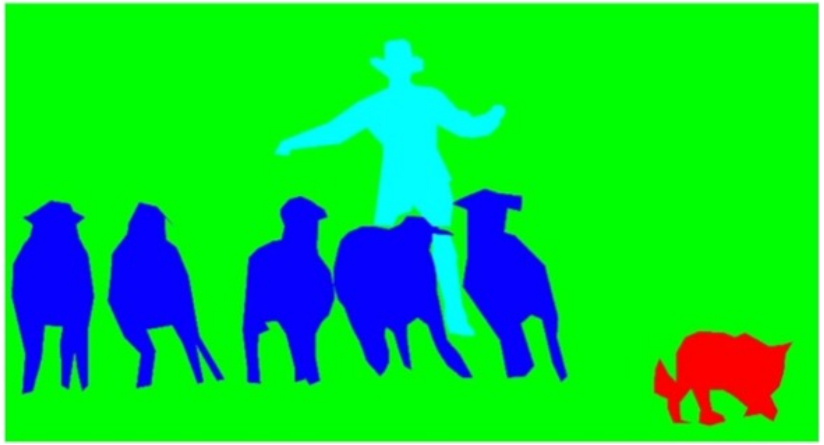
Typical computer vision problems

Image classification

Object detection



Semantic segmentation



Neural style transfer

Source: Lin, reiinakano.com

BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Neural style transfer

+

=

- Idea: create an image that has a similar representation to both the content image and the style image
  - Use a pre-trained CNN
  - Start with a randomly generated picture
  - Consider some layer and how active it is, given the content image. Consider also how active it is, given the generated image. The difference is the “content cost”.
  - Consider some layer and how correlated its different activations are, given the style image. Consider the same for the generated image. The difference is the “style cost”
  - We adjust the output image to minimize content and style costs together
- Applications:
  - Artificial artwork
  - Image enhancements

Source: Lin, wikipedia.com

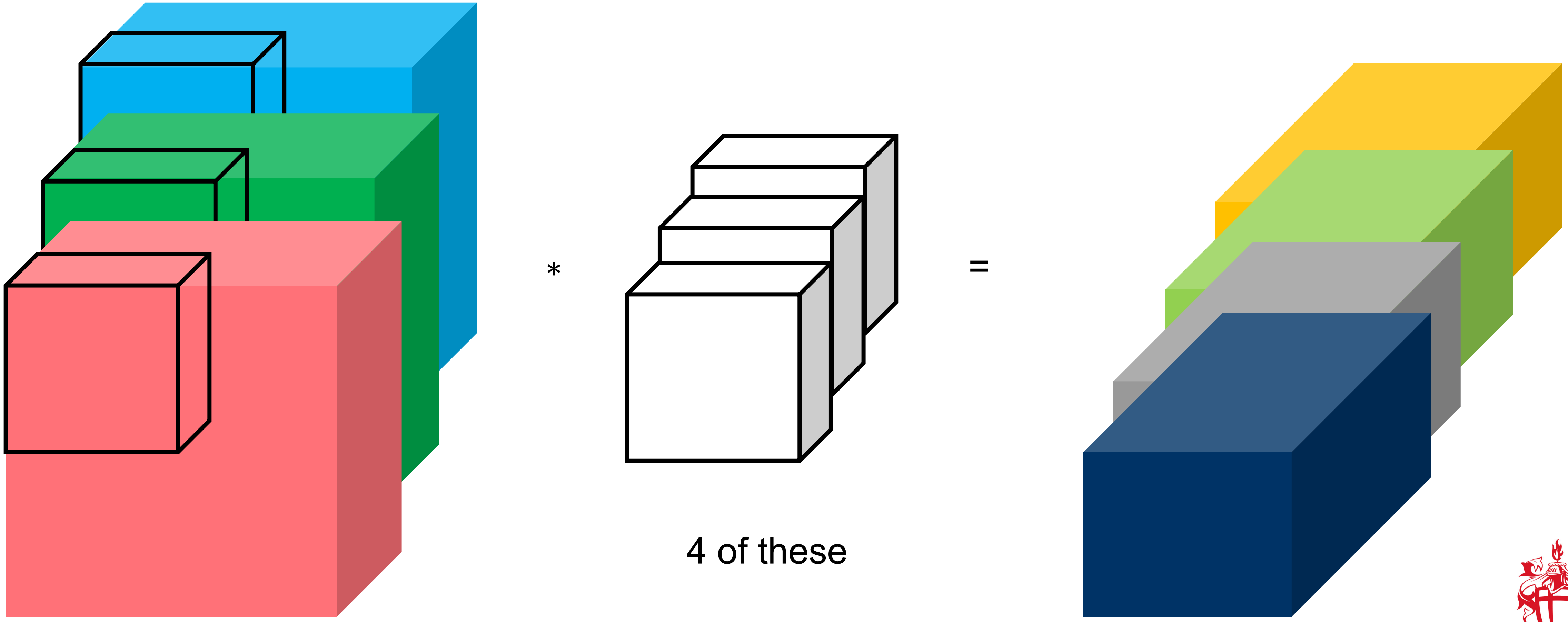
BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

Convolutions of 1D and 3D data

Applications

- 1D: analyze sequence data
  - E.g., audio
- 3D
  - analyze three-dimensional images, e.g., MRIs
  - analyze videos (time is the third dimension)

3D-convolution of 3D-data



BAYES BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON

See you in class!

Sources

- Bhaskhar, 2021, Introduction to Deep Learning: https://cs229.stanford.edu/syllabus.html
- DeepLearning.AI, n.d.: deeplearning.ai
- Dieleman, 2020, Lecture 3: Convolutional Neural Networks: https://storage.googleapis.com/deepmind-media/UCLxDeepMind_2020/L3%20-%20UUCLxDeepMind%20DL2020.pdf
- Géron, 2019, Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow
- Goodfellow, Bengio, Courville, 2016, The Deep Learning Book: http://www.deeplearningbook.org
- Liang, 2016, Introduction to Deep Learning: https://www.cs.princeton.edu/courses/archive/spring16/cos495/
- Lin et al., 2014, Microsoft COCO: Common Objects in Context: https://arxiv.org/pdf/1405.0312.pdf
- Ronneberger et al., 2015, U-Net: Convolutional Networks for Biomedical Image Segmentation: https://link.springer.com/content/pdf/10.1007/978-3-319-24574-4_28.pdf
- Szegedy et al., 2015, Going Deeper with Convolutions: https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Szegedy_Going_Deeper_With_2015_CVPR_paper.pdf

BAYES
BUSINESS SCHOOL
CITY, UNIVERSITY OF LONDON