Imperial College London

Machine Learning for Imaging

Logistics

Syllabus

- Introduction
- Image segmentation
- Image registration
- Self-supervised learning
- Inverse problems
- Causality in imaging
- Trustworthy ML in imaging
- Advanced topics

People

Ben Glocker

Daniel Rueckert

Mélanie Roschewitz

Fabio De Sousa Ribeiro

Schedule

Course runs for 8 weeks (weeks 2 – 9)

Week 2	3	4	5	6	7	8	9	
Friday, Jan 19	Friday, Jan 26	Friday, Feb 2	Friday, Feb 9	Friday, Feb 16	Friday, Feb 23	Friday, Mar 1	Friday, Mar 8	
Lecture 14:00 – 16:00	Lecture 14:00 – 16:00	Lecture 14:00 – 16:00	Lecture 14:00 – 16:00	Lecture 14:00 – 16:00	Lecture 14:00 – 16:00	Lecture 14:00 – 16:00	Talk 14:00	Rev 15:00
			COURSEWORK					

- **One assessed coursework (mini-project)**
- Tutorials and CW are hands-on programming exercises in Python

Tools

https://scientia.doc.ic.ac.uk/2324/modules/70014/

- Scientia (lecture notes, tutorial files, links)
- Panopto (recordings)
- Ed Discussion (offline Q&A)
- Microsoft Teams (hybrid live sessions)
- Google Colab (programming tutorials, coursework)

Weekly tasks

Join the live sessions

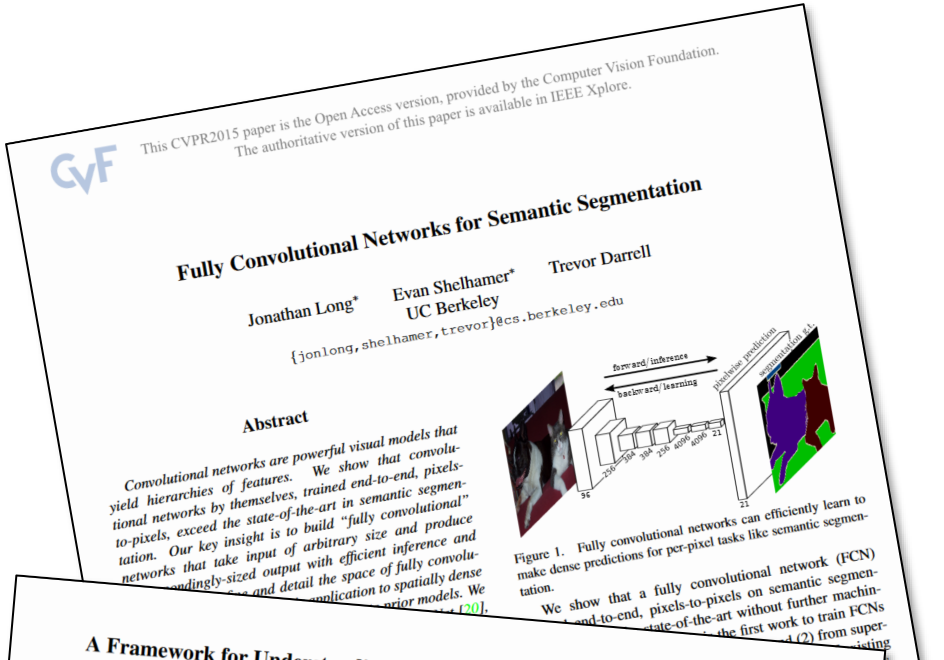
- **Fridays, 14:00 – 16:00, LT308**: Lectures, Q&A, quizzes, invited talk

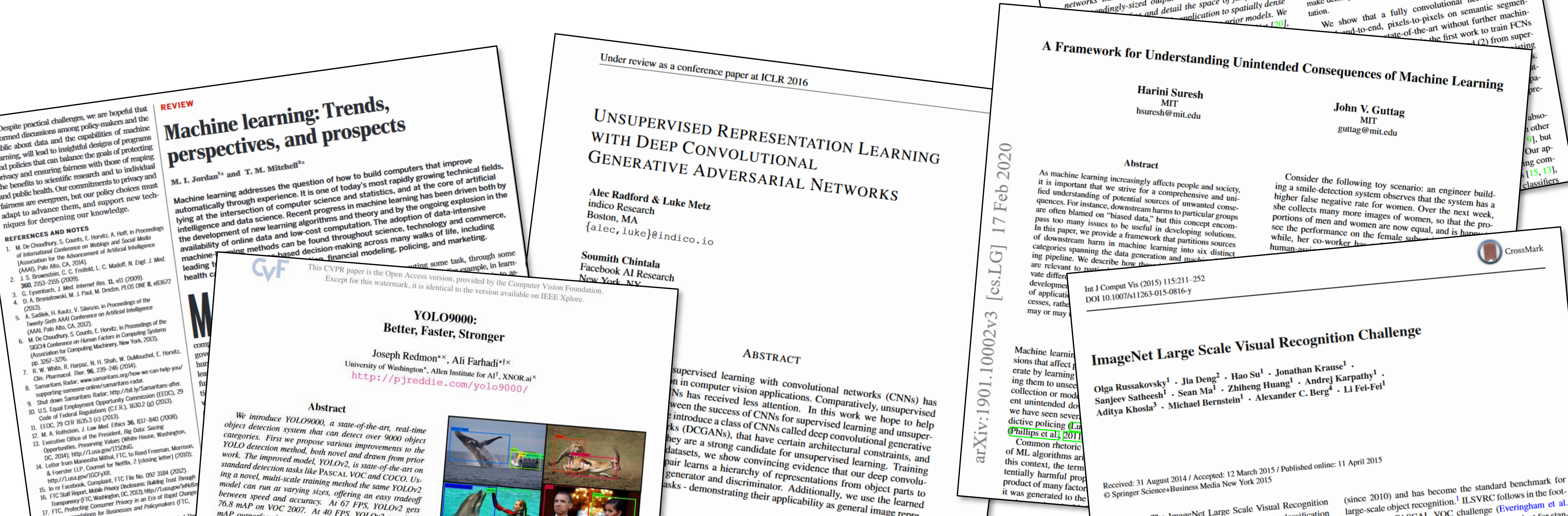
During the week

- programming tutorials (with support on-demand)
- ask any questions on Ed Discussion

Weekly papers

- Optional, complementary reading material
- Aim for **high-level understanding**



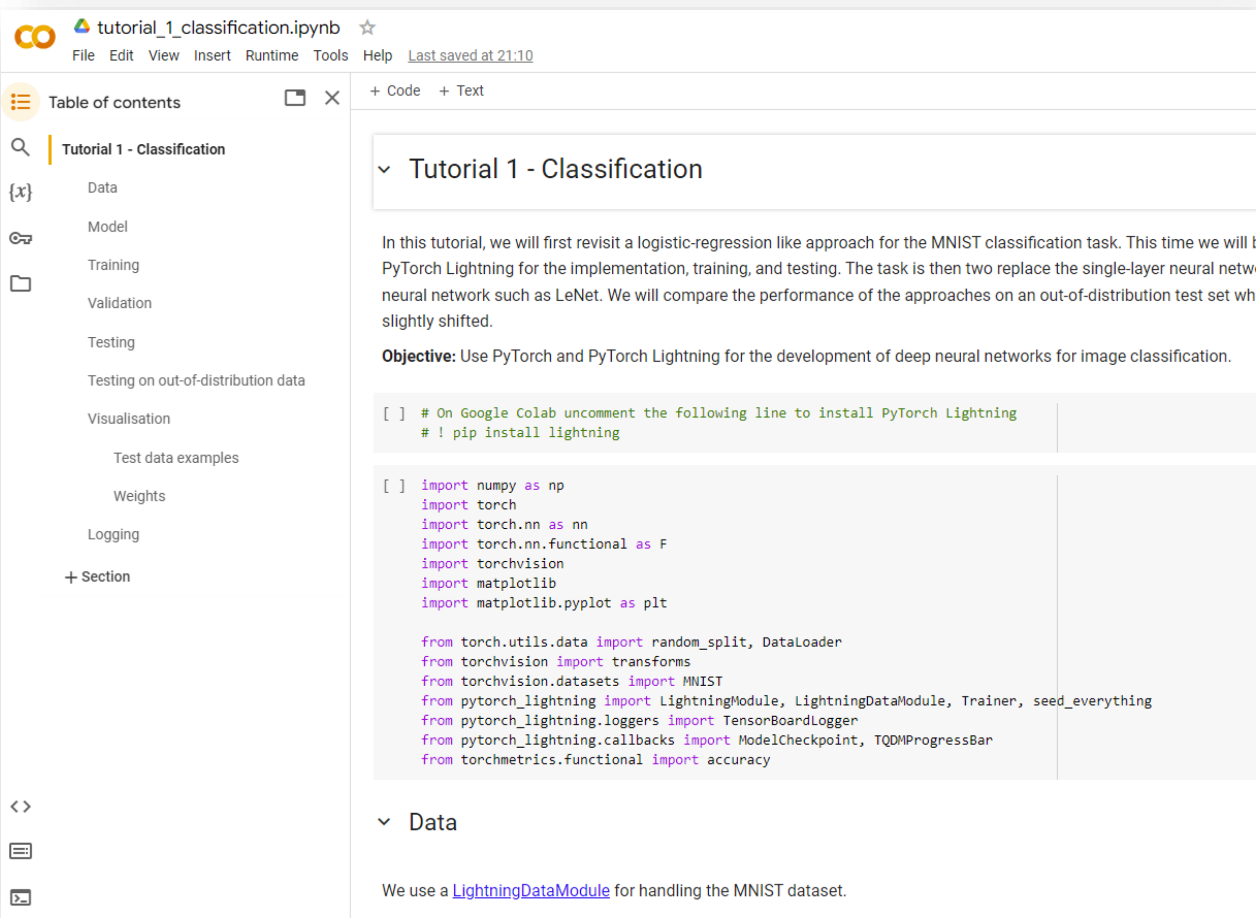


Tutorials

You will need a Python programming environment

- **Google Colab - recommended**
  (free but limited access to GPUs)
- **Locally on your own machine**
  (instructions on Scientia)
- **Locally/remotely using DoC machines**
  (instructions on Scientia)

Tutorial support on-demand via Teams



Coursework

Self-supervised learning for image classification

- mini-project on deep learning training strategies
- work in groups of two

Start: Friday, February 9 (week 5)

End: Thursday, Feburary 22 (week 7)

Further reading

- **Pattern Recognition and Machine Learning**
  C. Bishop. Springer, 2006.
- **Deep Learning: Foundations and Concepts**
  C. Bishop, H. Bishop. Springer, 2023.
- **Deep Learning**
  I. Goodfellow, Y. Bengio, A. Courville. MIT Press, 2016.
- **Mathematics for Machine Learning**
  M. Deisenroth, A. Faisal, C. Ong. Cambridge University Press, 2020
- **Medical Image Analysis**
  A. Frangi, J. Prince, M. Sonka. Elsevier, 2023

Imperial College London

Machine Learning for Imaging

Introduction

Ben Glocker

b.glocker@imperial.ac.uk

Image source: pixabay.com, CC0 Creative Commons

Where was this image taken?

Image source: pixabay.com, CC0 Creative Commons

What animals are in this image?

Image source: pixabay.com, CC0 Creative Commons

Are there birds in this image?

Image source: pixabay.com, CC0 Creative Commons

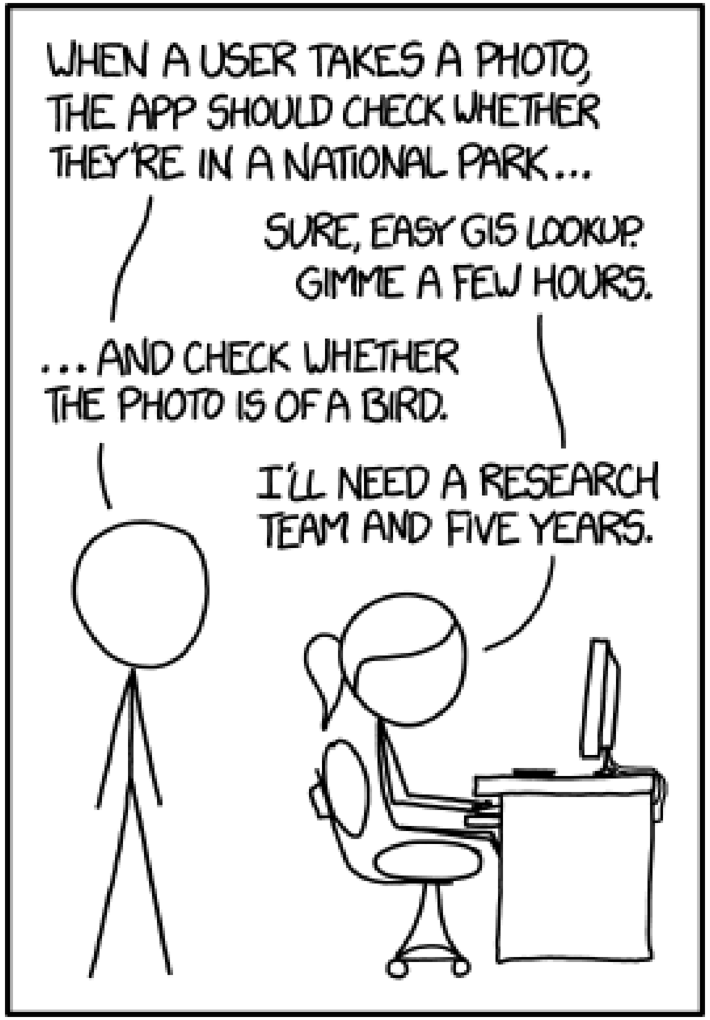
Where are the birds in this image?

Image source: pixabay.com, CC0 Creative Commons

What type of bird is in this image?

Image source: pixabay.com, CC0 Creative Commons

Source: https://xkcd.com/1425/



IN CS, IT CAN BE HARD TO EXPLAIN THE DIFFERENCE BETWEEN THE EASY AND THE VIRTUALLY IMPOSSIBLE.

Applications

- transportation (autonomous cars, airplanes, drones)
- healthcare (diagnostic imaging, surgical robots)
- disaster response (search and rescue)
- manufacturing (industrial robots)
- military (autonomous weapons)
- entertainment (games, sports)
- agriculture (satellite imaging)
- surveillance (policing)
- material sciences
- astronomy
- forensics
- security
- biology
- art
- ...

Common image analysis tasks

- Image classification
- Semantic segmentation
- Image registration, motion estimation
- Object detection, localisation, recognition, tracking
- Image captioning (image-to-text)
- Image synthesis / generation (text-to-image)
- Image restoration, inpainting, style transfer, super-resolution
- …

Common image analysis tasks

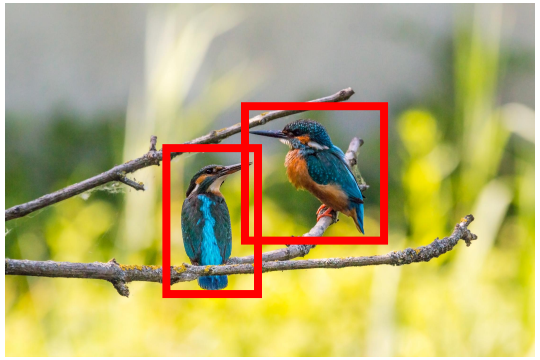
Image Classification

Output: Category (e.g., “bird”)

Object Detection

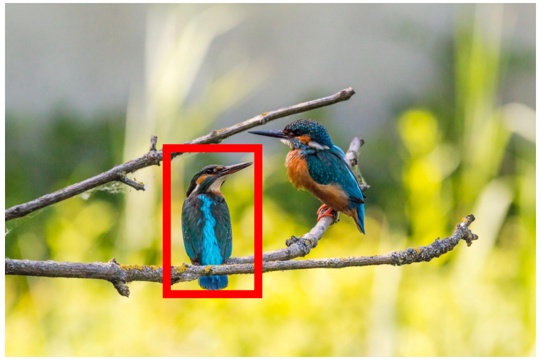
Output: Coordinates (e.g., centroid)

Object Localisation



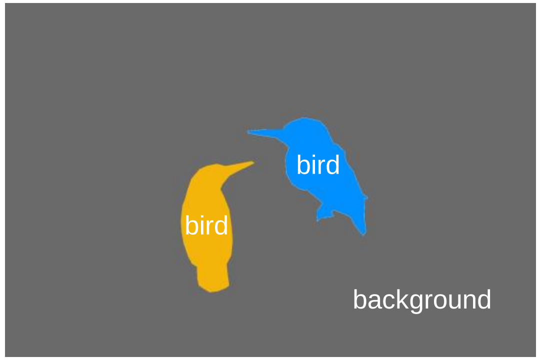
Output: Coordinates (e.g., bounding box)

Object Recognition



Output: Category (e.g., “kingfisher”)

Semantic Segmentation



Output: Labelmap

Image Captioning

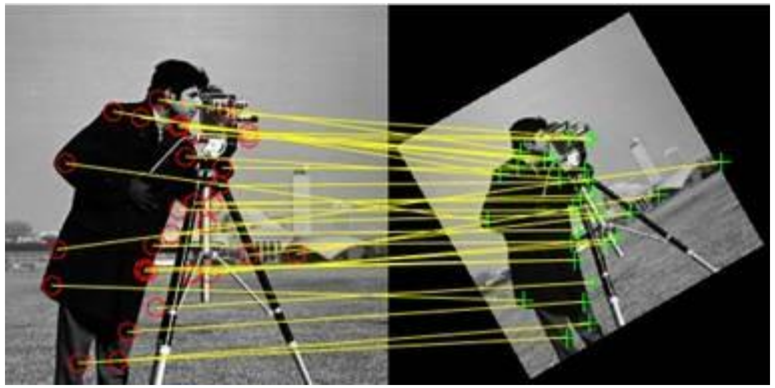
Output: Text
(e.g., “two birds sitting on a branch”)

Common image analysis tasks

Image Registration

Output: Transformation

Point Correspondences



Output: Pairs of Coordinates

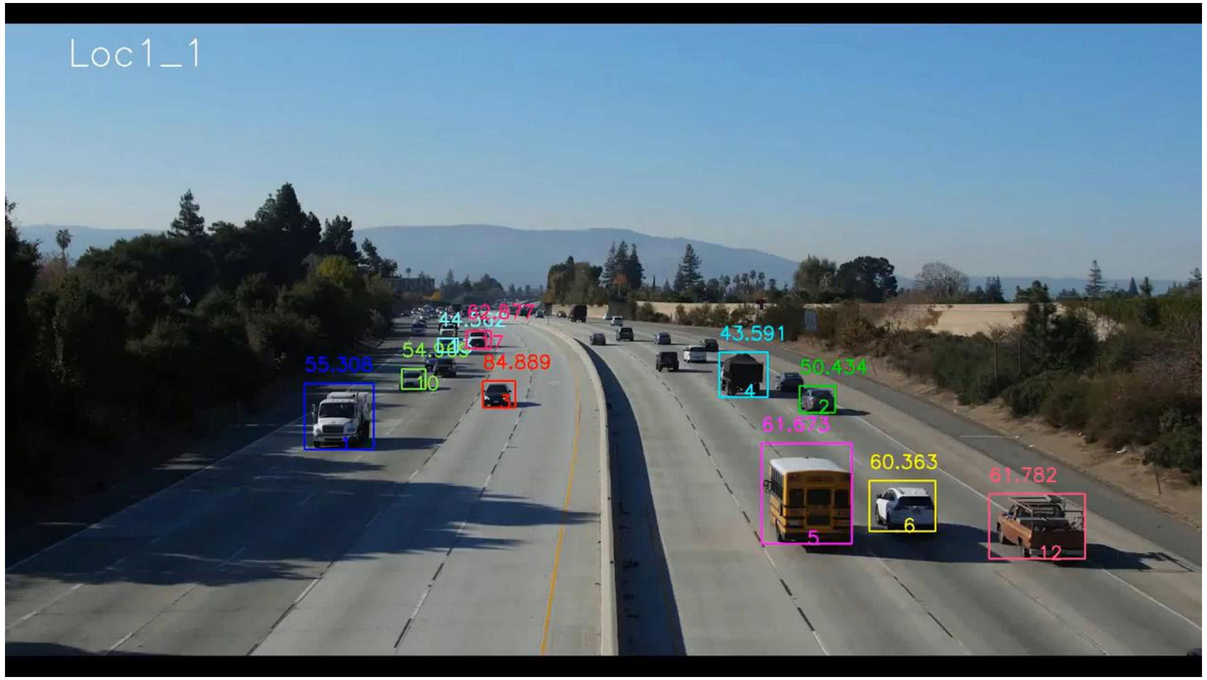
Source: https://uk.mathworks.com/discovery/image-registration.html

Common image analysis tasks

Panoramic Image Stitching

Common image analysis tasks

Object Tracking



Source: https://youtu.be/_i4numqiv7Y

Face Tracking

Source: https://youtu.be/ohmajJTcpNk

Image restoration

Source: https://en.wikipedia.org/wiki/Image_restoration_by_artificial_intelligence | https://github.com/megvii-research/NAFNet

Image synthesis

Source: https://thispersondoesnotexist.com/

Image synthesis

“An oil painting of a cowboy herding giant insects on the moon”

Midjourney 5.1

DALL·E 3

Image synthesis

“A painting of a giraffe surfing in the style of Jackson Pollock”

Midjourney 5.1

DALL·E 3

Inpainting

“A mecha robot sitting on a bench”



Source: https://huggingface.co/runwayml/stable-diffusion-inpainting

Style transfer

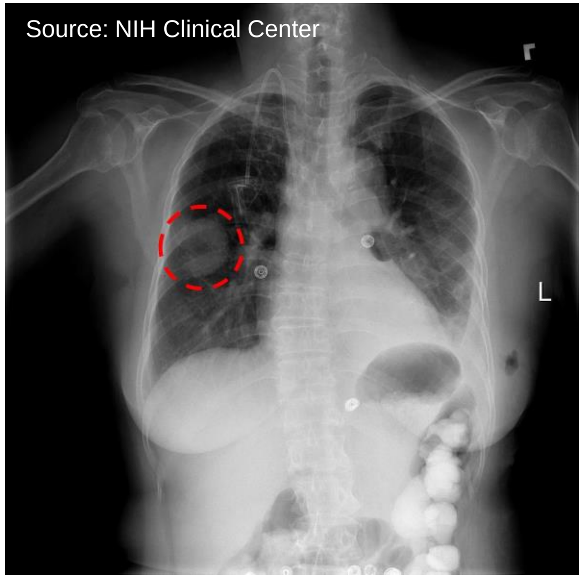
Source: Gatys et al. 2016. [Image Style Transfer Using Convolutional Neural Networks]

Medical imaging applications

Image Classification

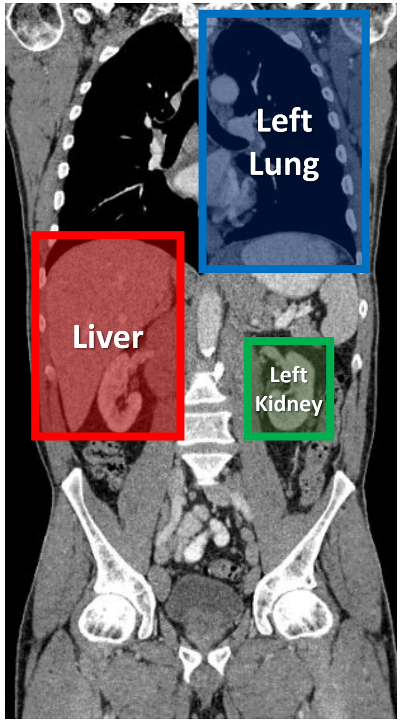
Output: cancer / no cancer

Object Detection



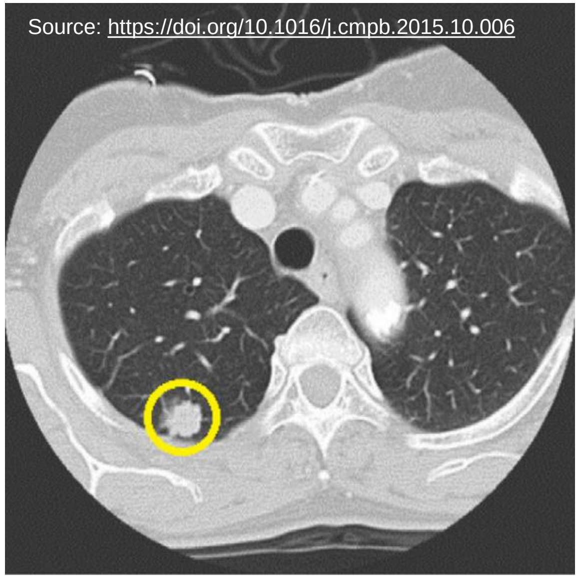
Output: presence and location of a mass

Object Localisation



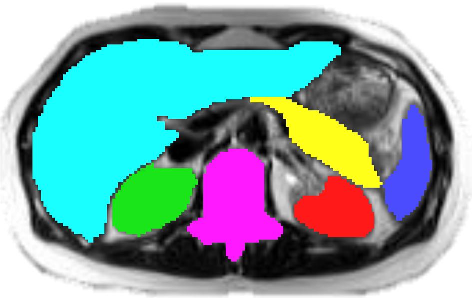
Output: organ bounding boxes

Object Recognition



Output: benign / malignant

Semantic segmentation



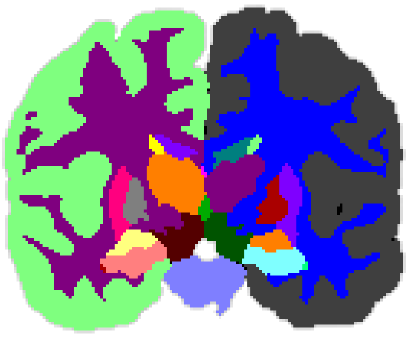
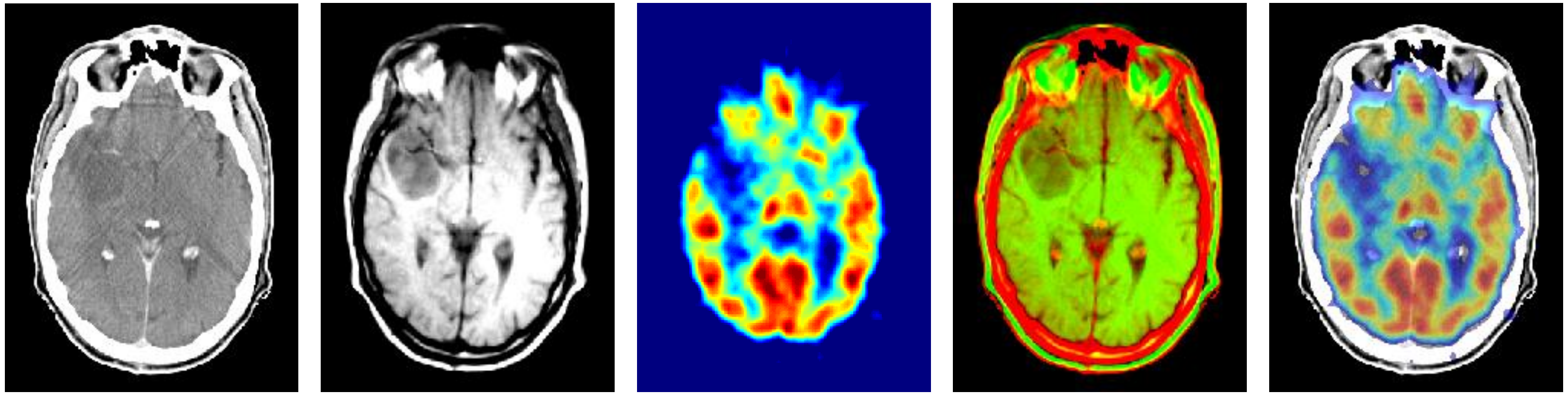
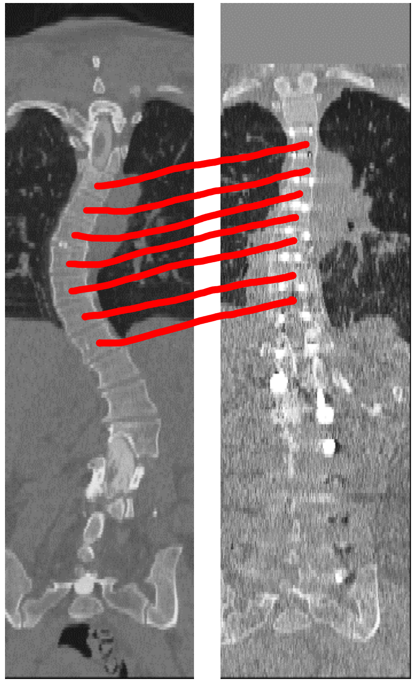


Image registration

Multi-modal Image Fusion

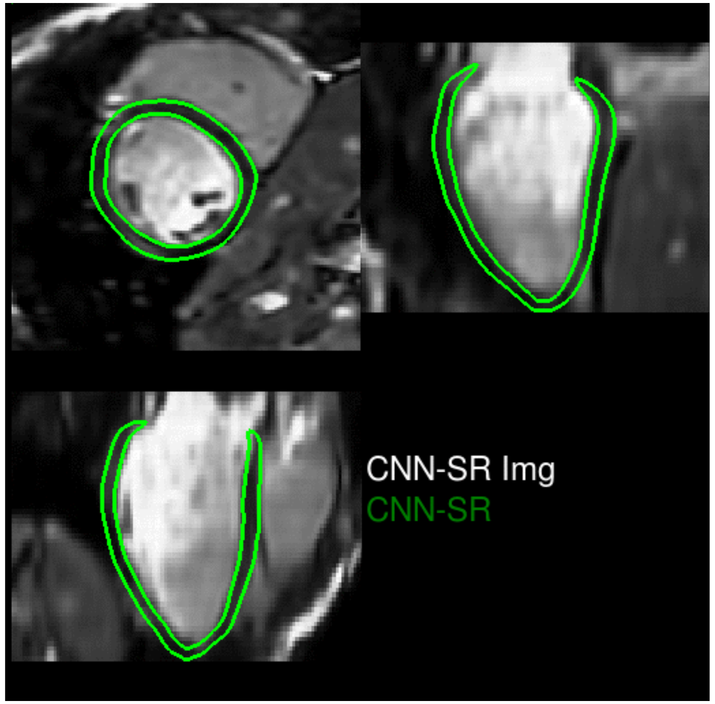


Longitudinal Comparison



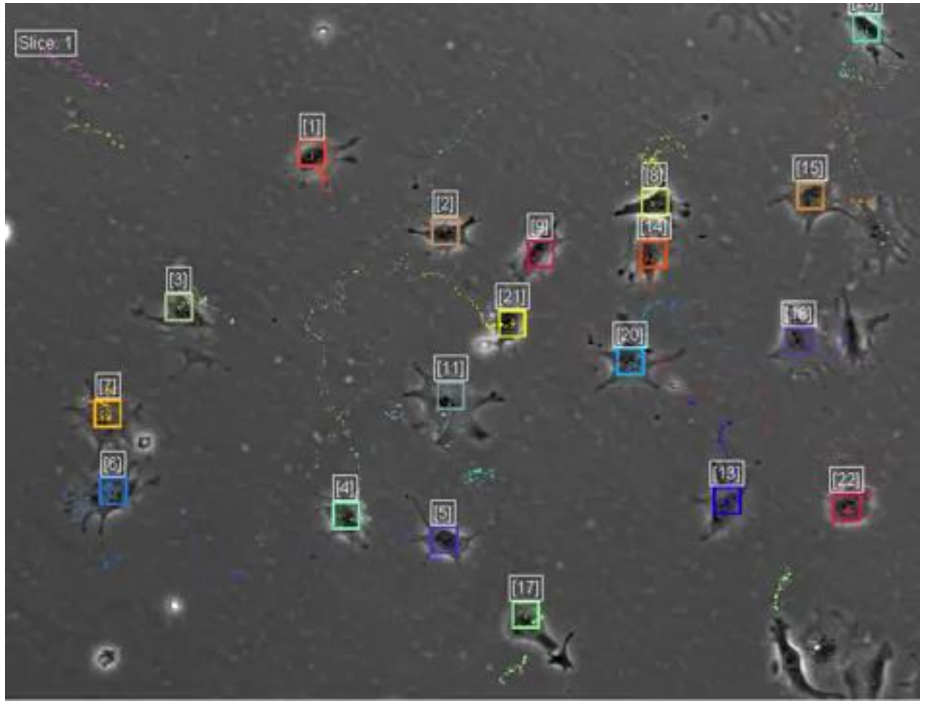
Motion estimation & object tracking

Cardiac Motion



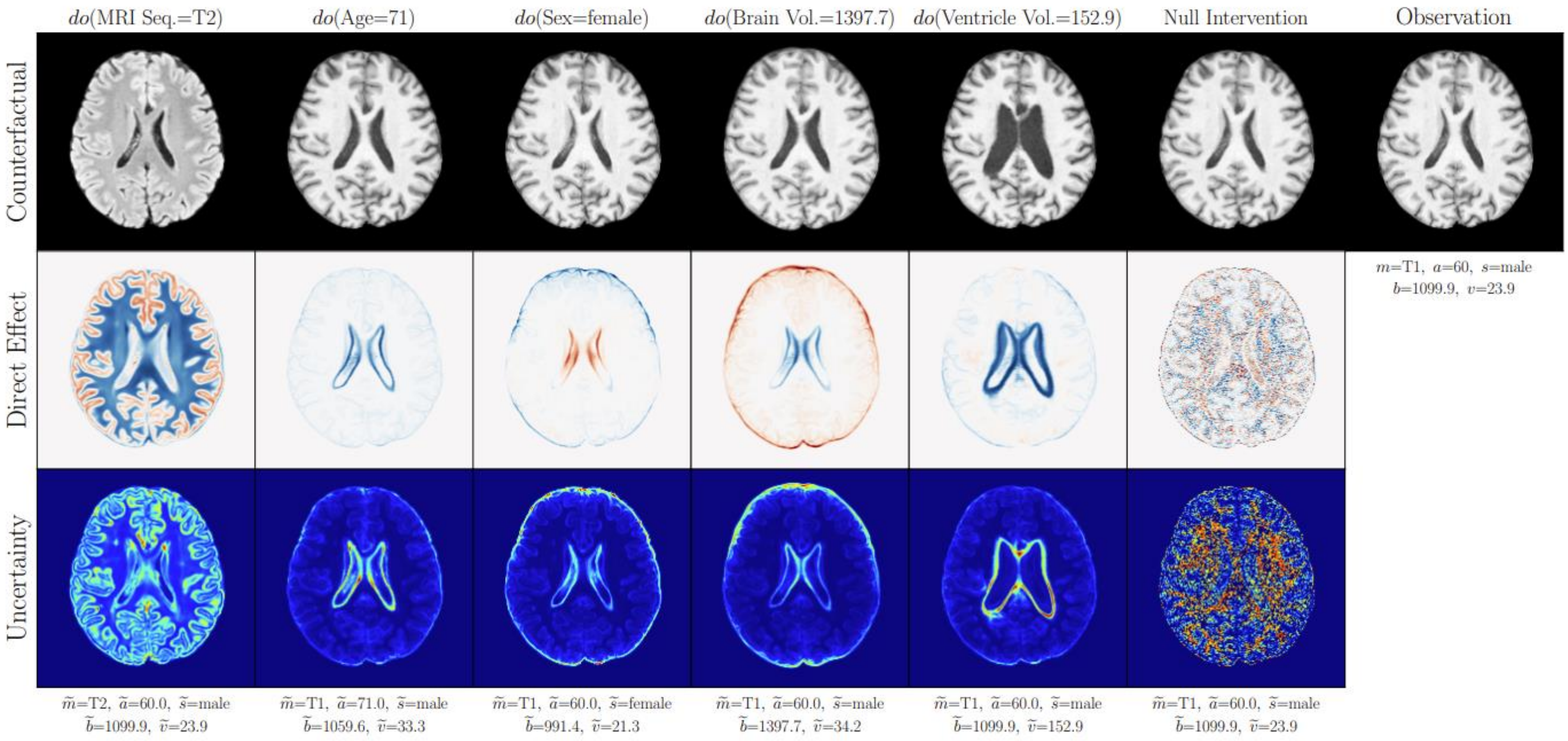
Source: Oktay et al. MICCAI 2016

Cell Tracking



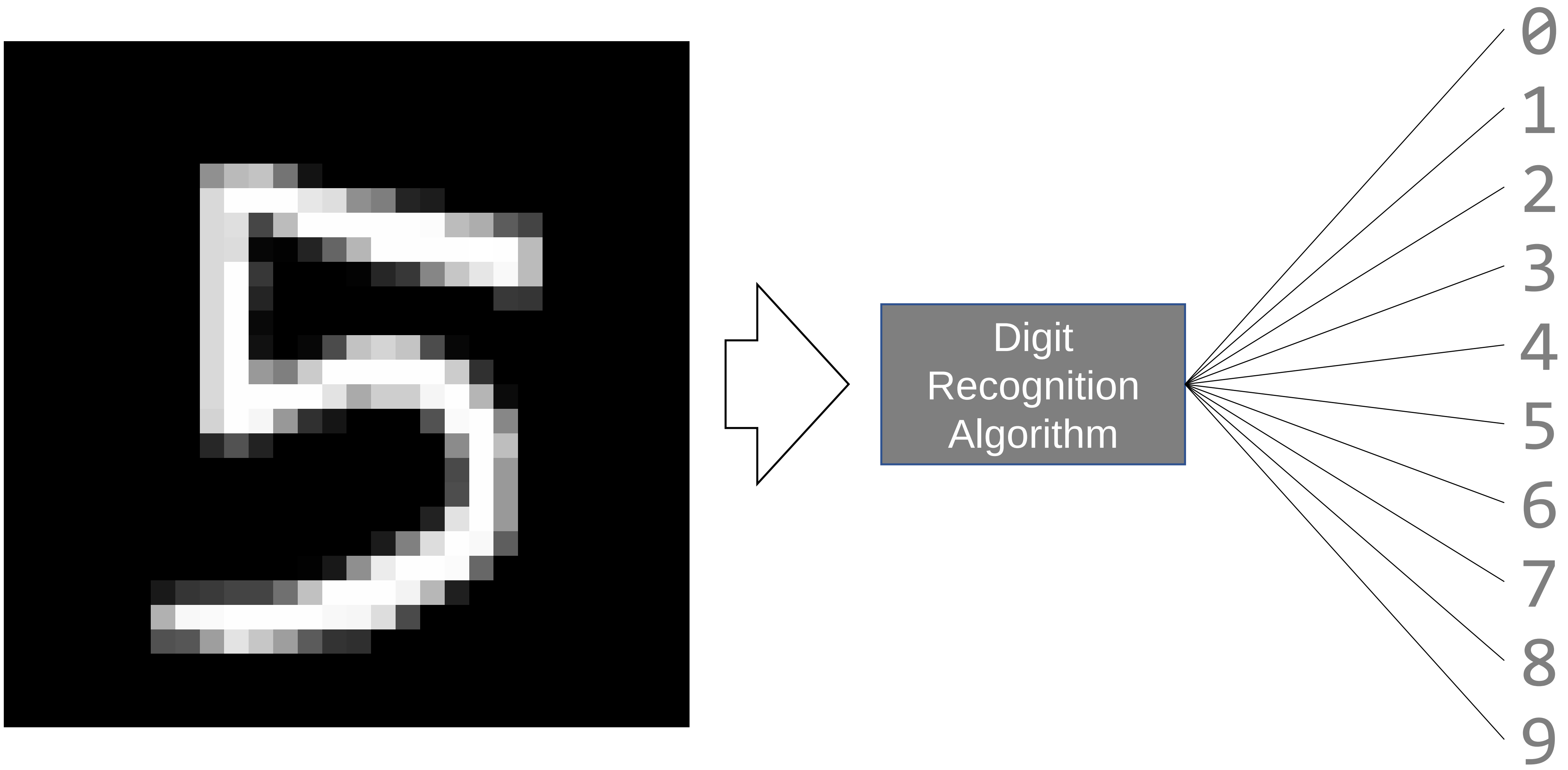
Source: https://youtu.be/SLYgvHzAm2w

Counterfactual image synthesis

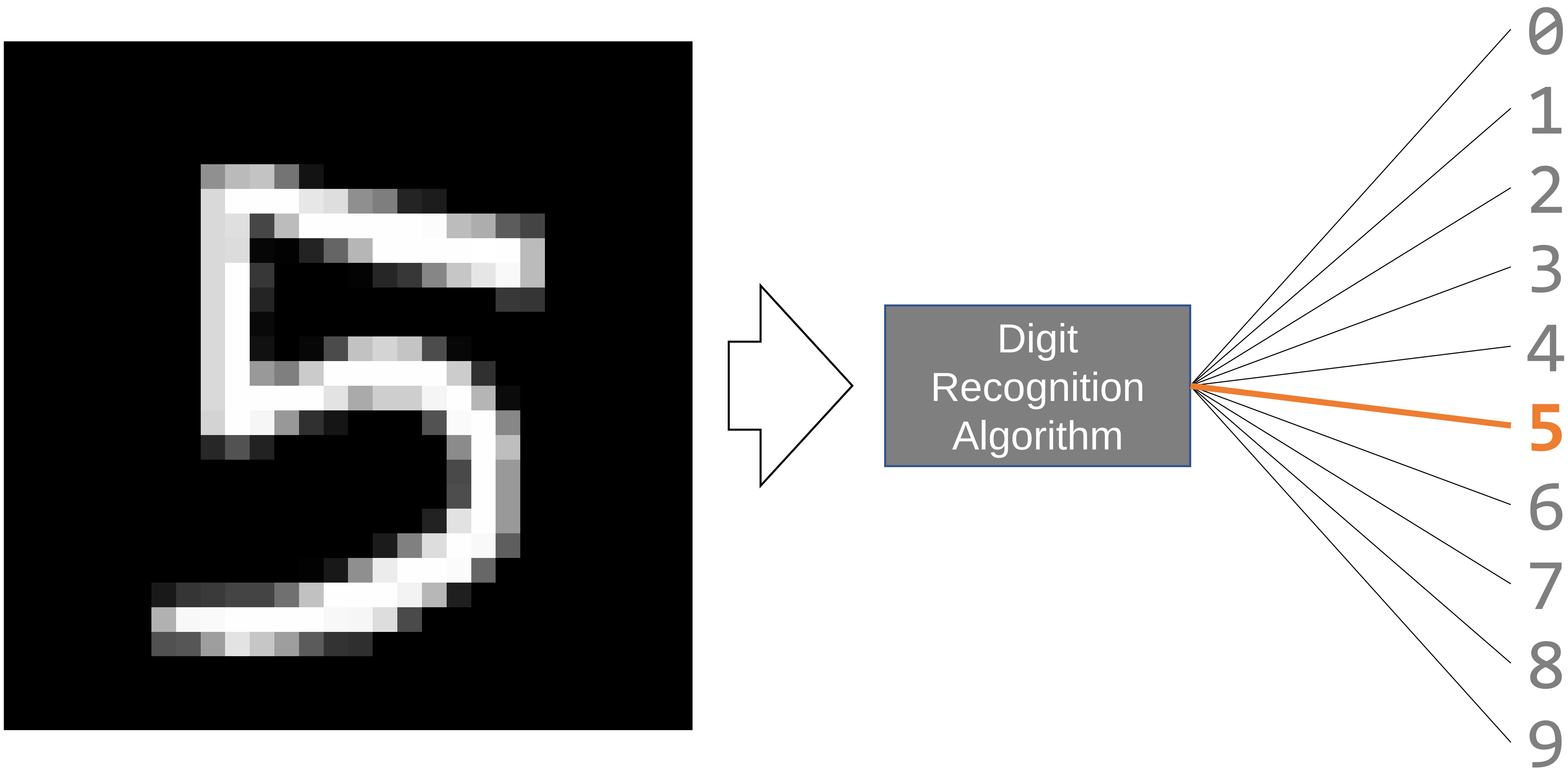


Source: Ribeiro et al. 2023. [High Fidelity Image Counterfactuals with Probabilistic Causal Models]

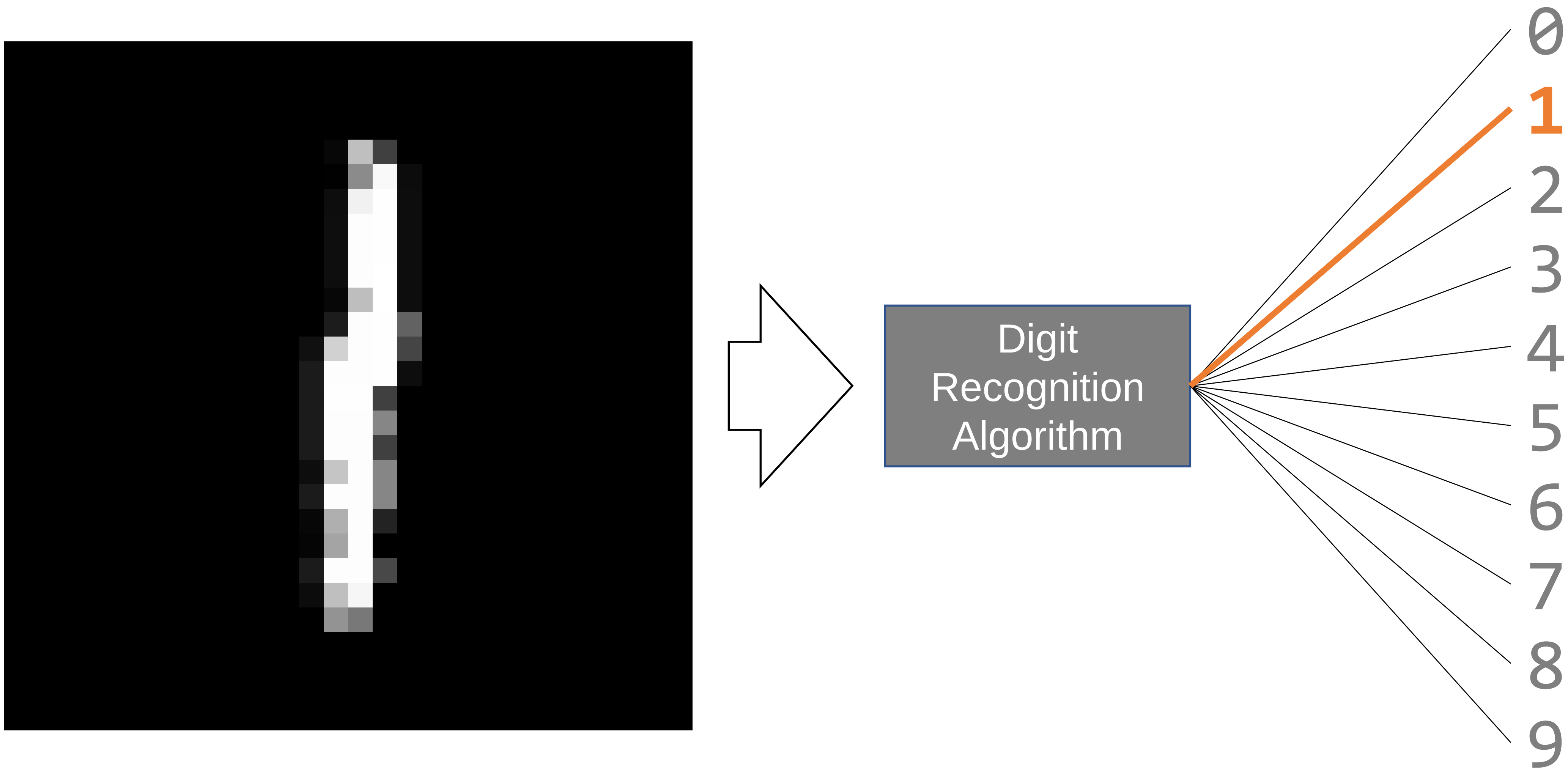
Example: Digit recognition



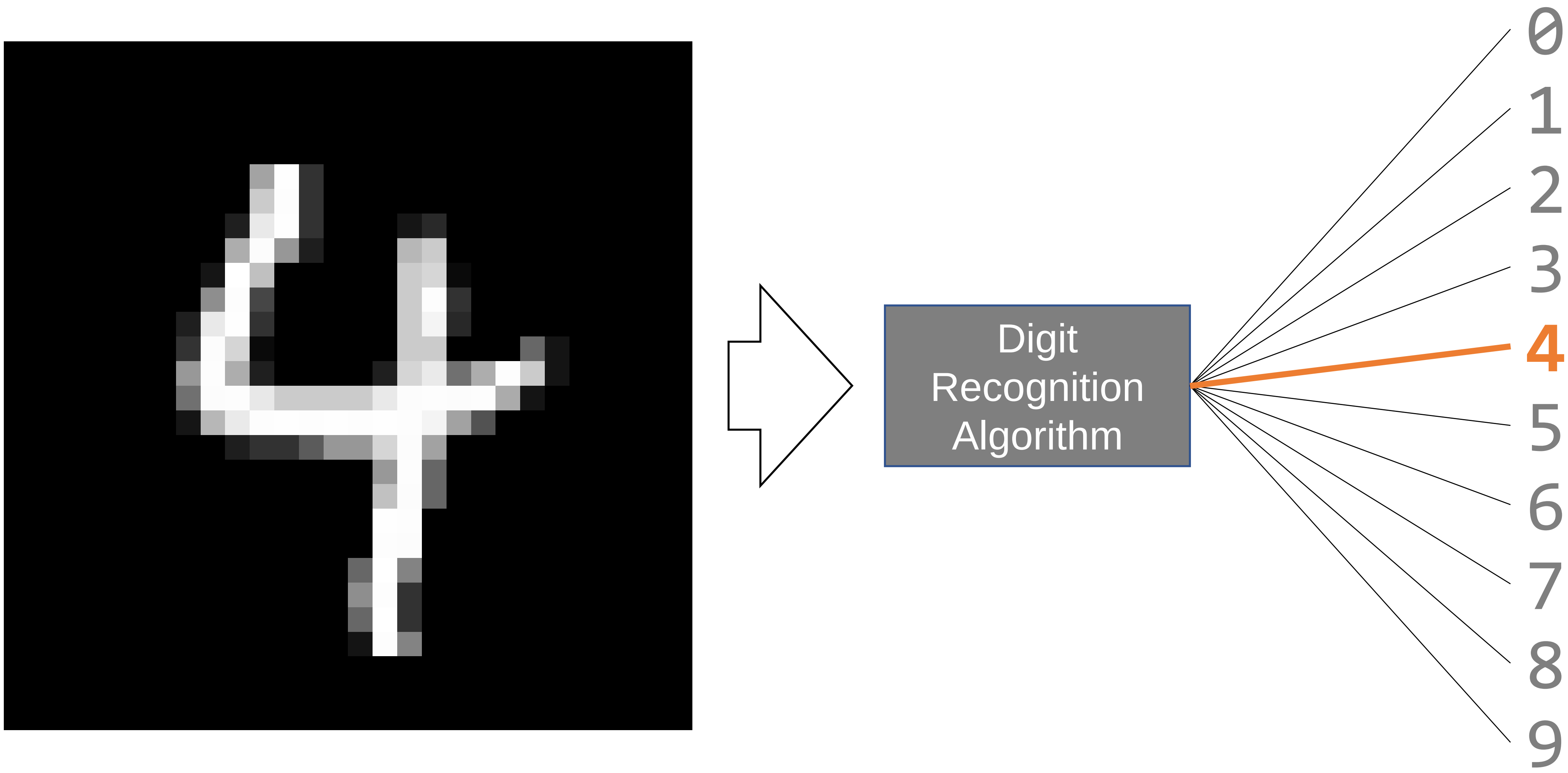
Example: Digit recognition



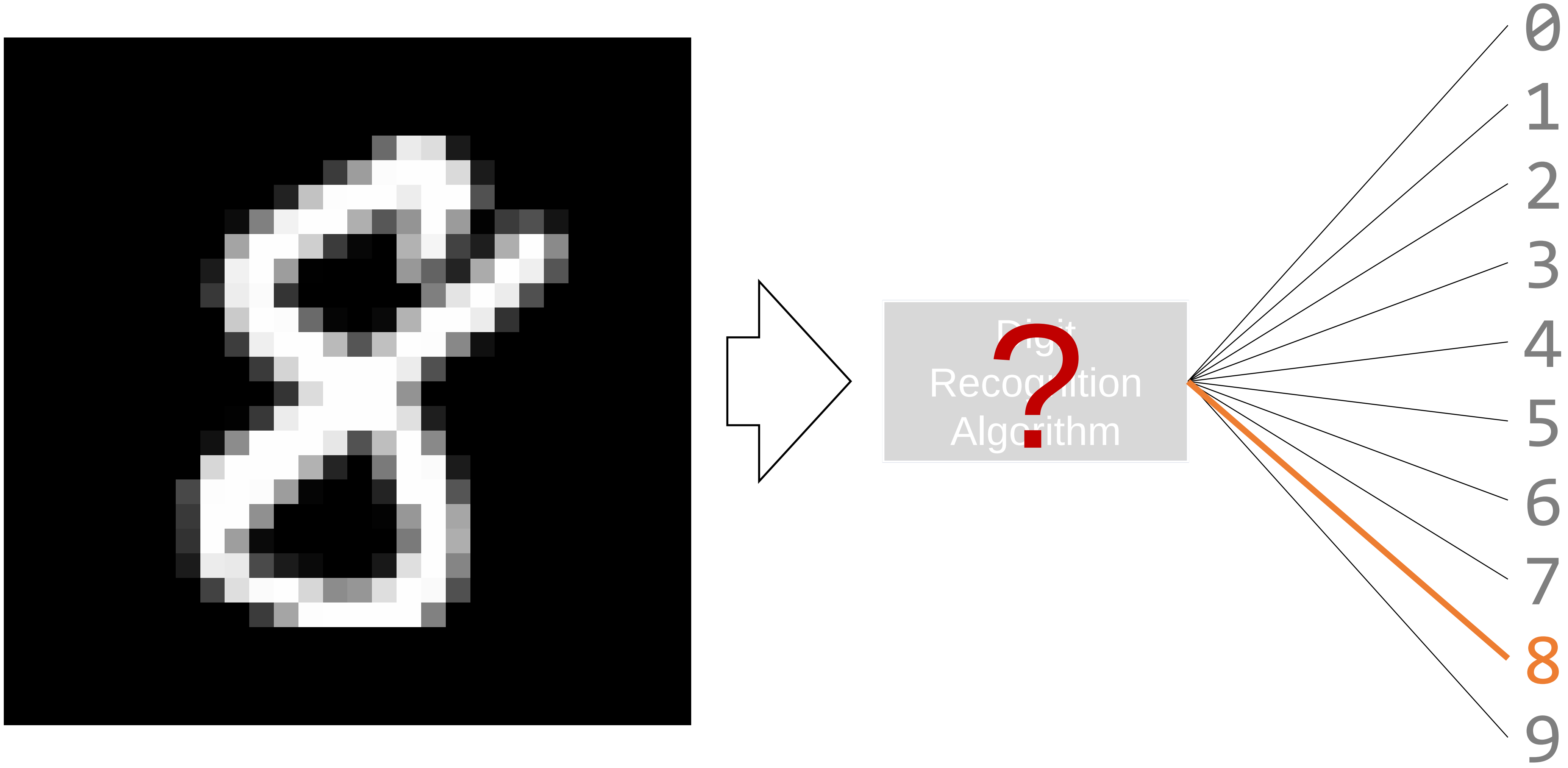
Example: Digit recognition



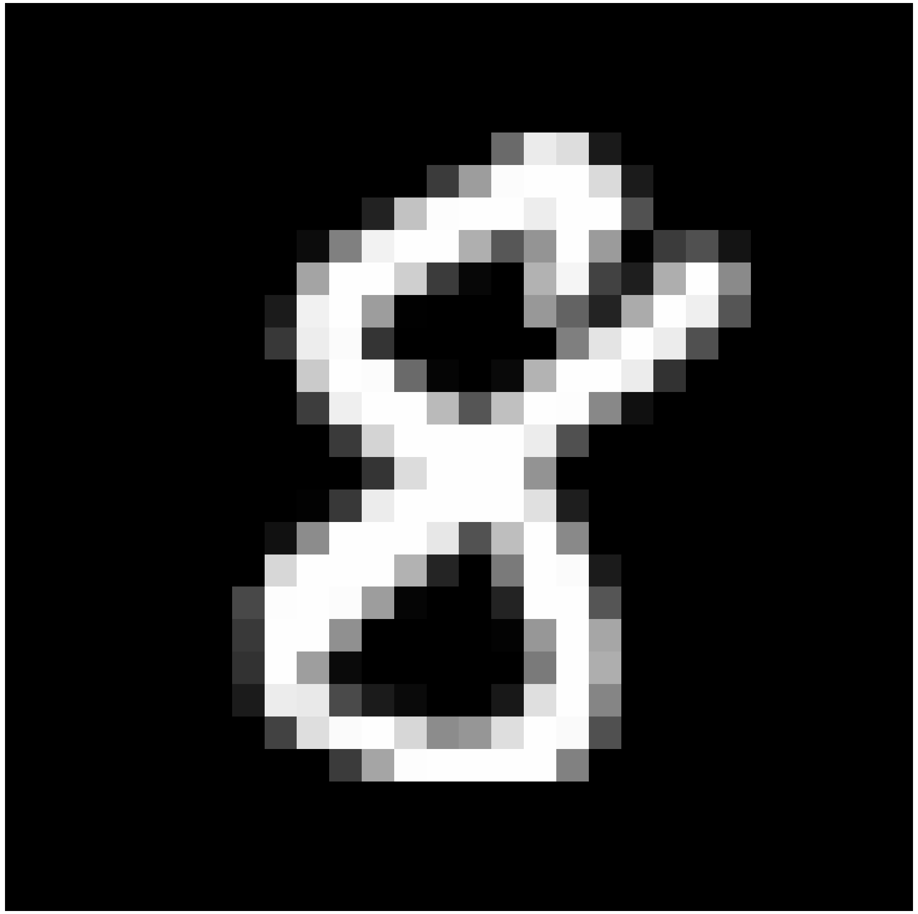
Example: Digit recognition



Example: Digit recognition

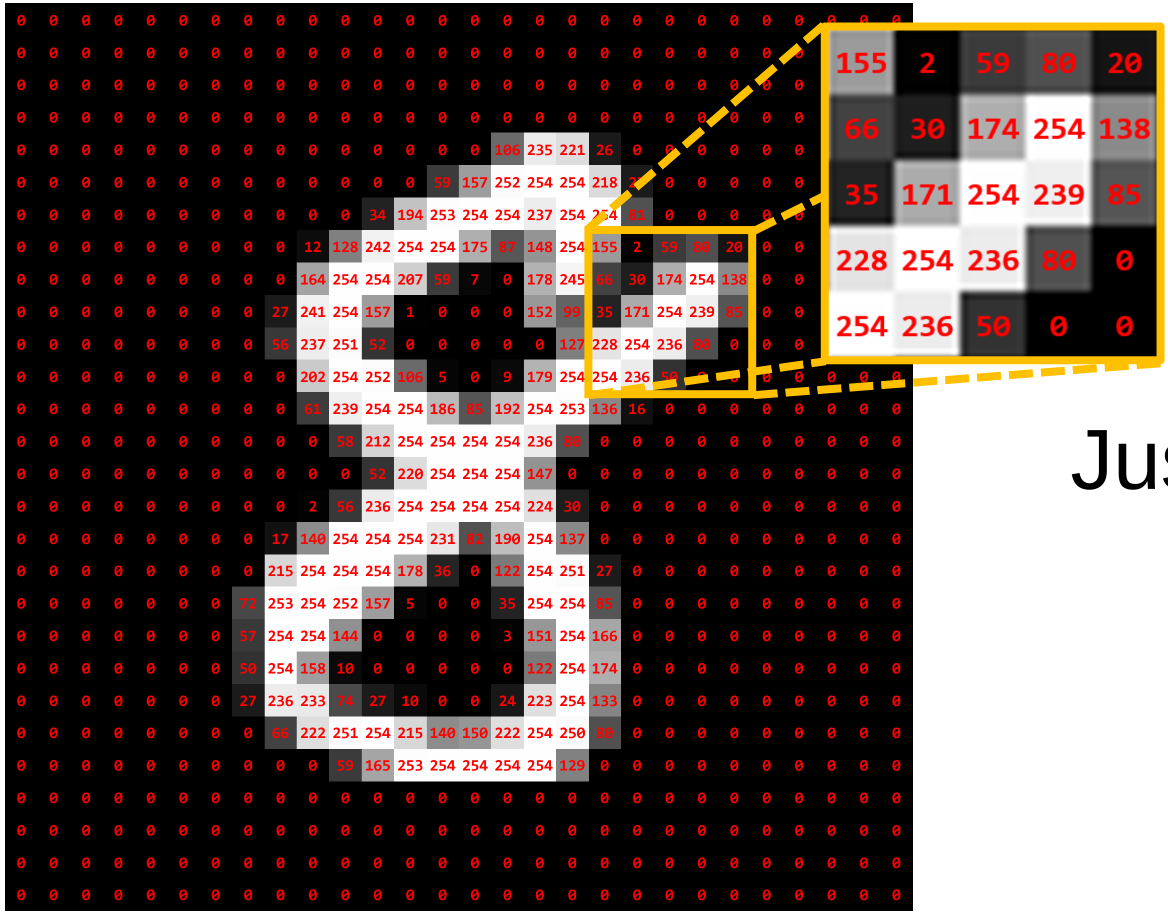


Example: Digit recognition



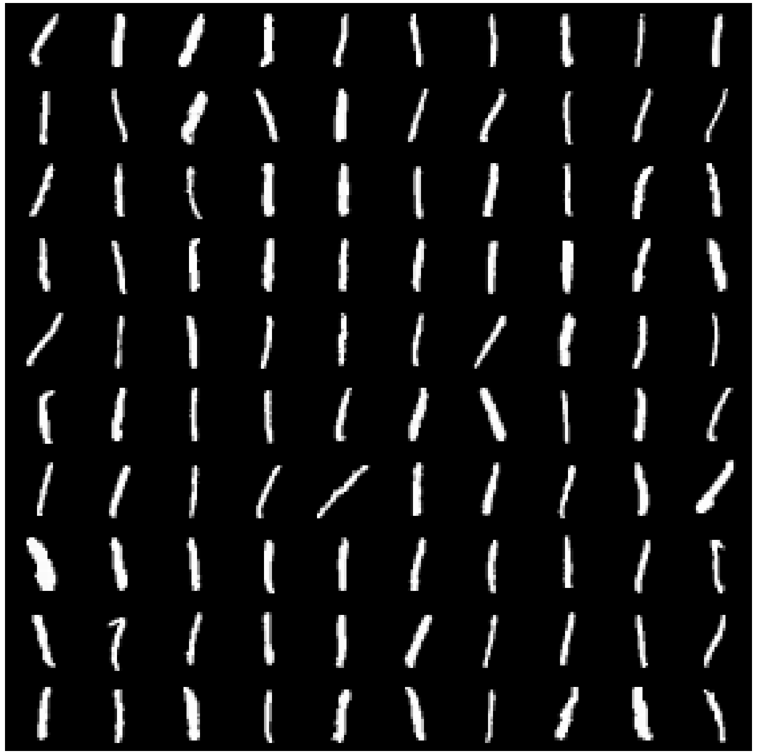
What is an image?

Example: Digit recognition



Just an array of numbers

Example: Digit recognition

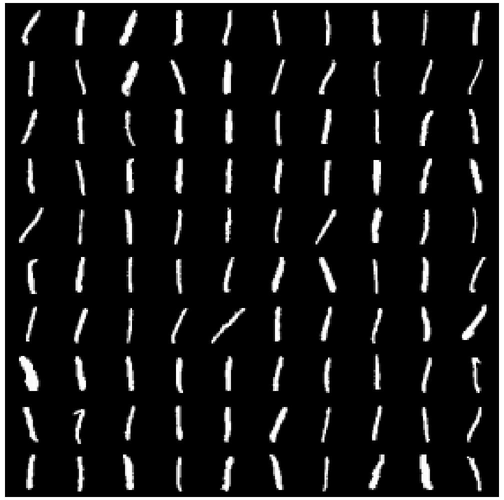


ones

eights

Example: Digit recognition

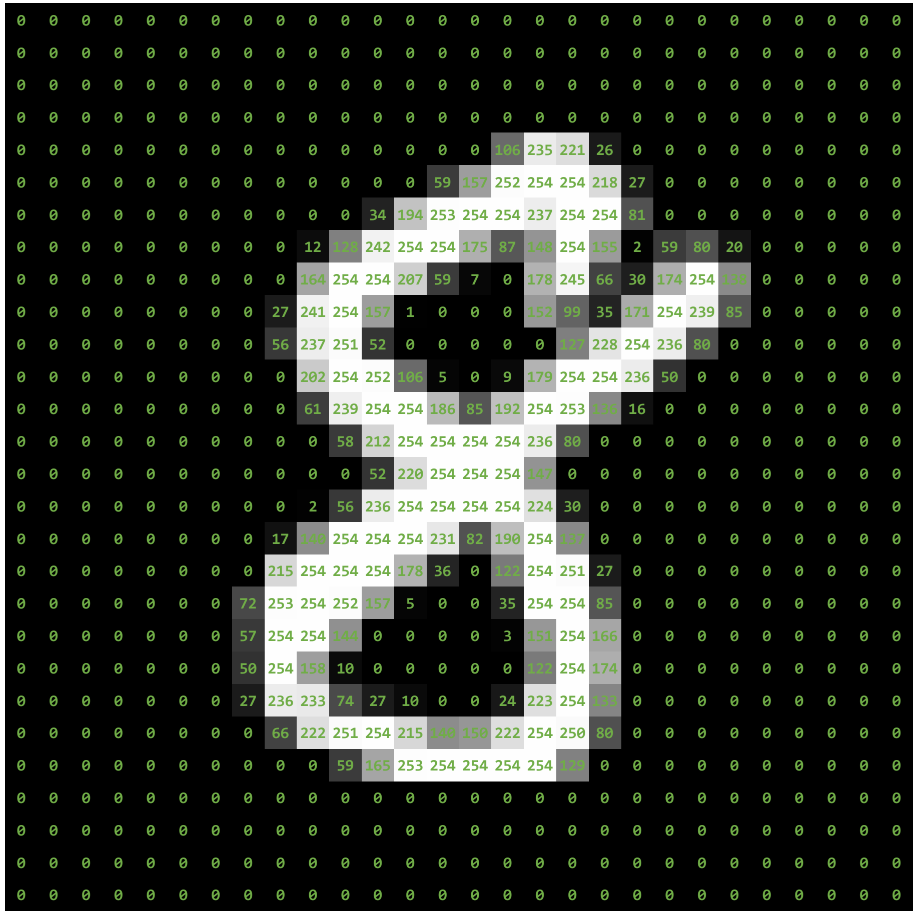
- Can we devise a simple algorithm to distinguish between digits one and eight?
- How about counting the number of “white” pixels?



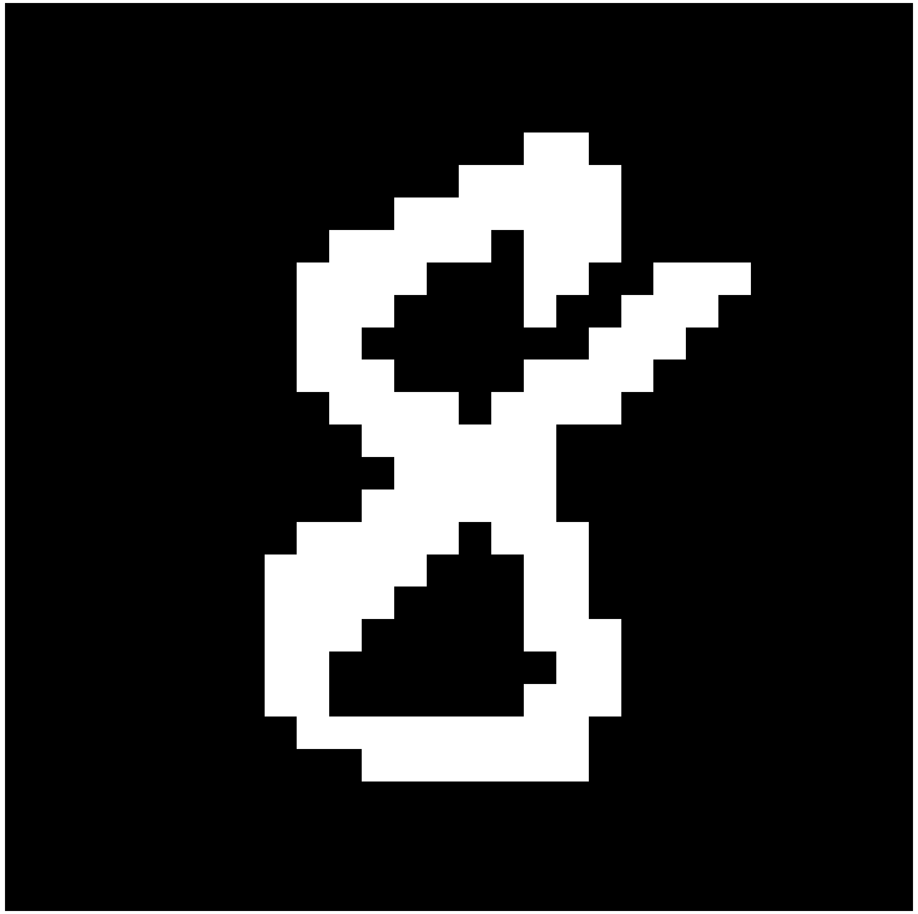
ones

eights

Example: Digit recognition

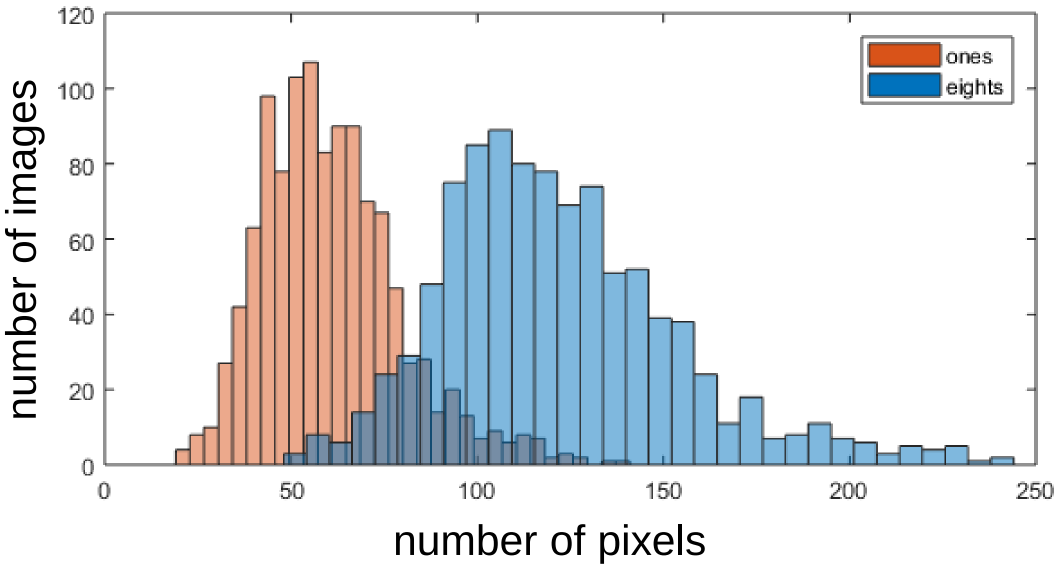


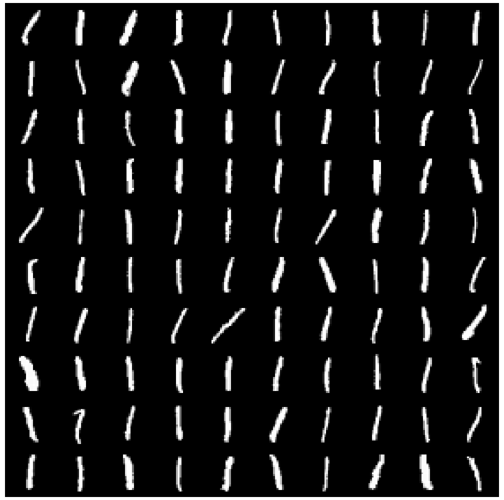
$> 127 =$



Example: Digit recognition

- Can we devise a simple algorithm to distinguish between digits one and eight?
- How about counting the number of “white” pixels?
- A simple “threshold” algorithm might work well...



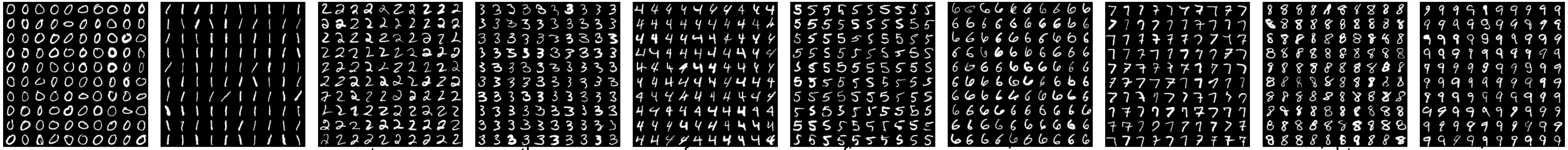


ones

eights

Example: Digit recognition

- Can we devise a simple algorithm to distinguish between digits one and eight?
- How about counting the number of “white” pixels?
- A simple “threshold” algorithm might work well...

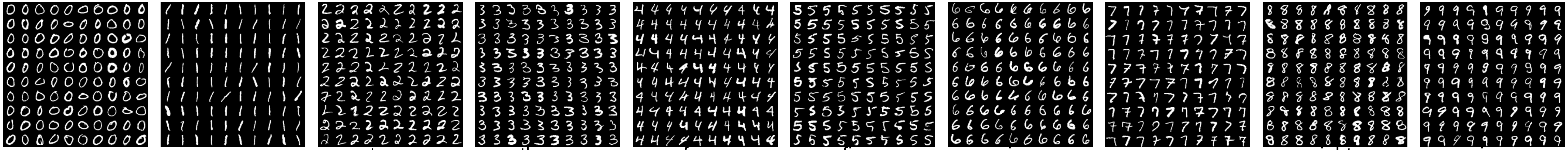


zeros ones twos threes fours fives sixes sevens eights nines

The MNIST Database: http://yann.lecun.com/exdb/mnist/

Example: Digit recognition

- Can we devise a simple algorithm to distinguish between digits one and eight?
- How about counting the number of “white” pixels?
- A simple “threshold” algorithm might work well...except...
- Devising an explicit algorithm based on simple rules is difficult!

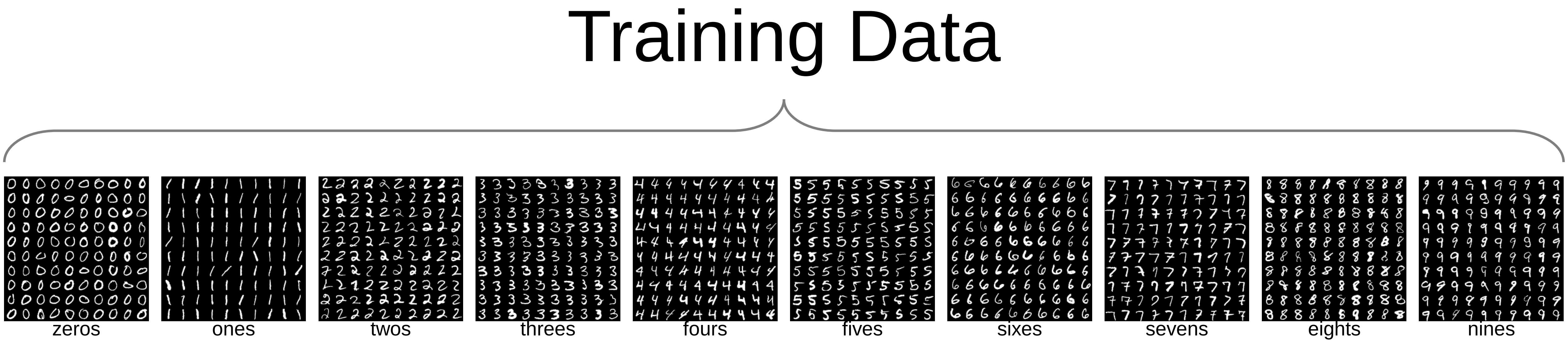


zeros ones twos threes fours fives sixes sevens eights nines

The MNIST Database: http://yann.lecun.com/exdb/mnist/

Example: Digit recognition

- How about learning the (possibly complex) rules from data?
- Assume we are given (a lot of) examples...



The MNIST Database: http://yann.lecun.com/exdb/mnist/

Example: Digit recognition

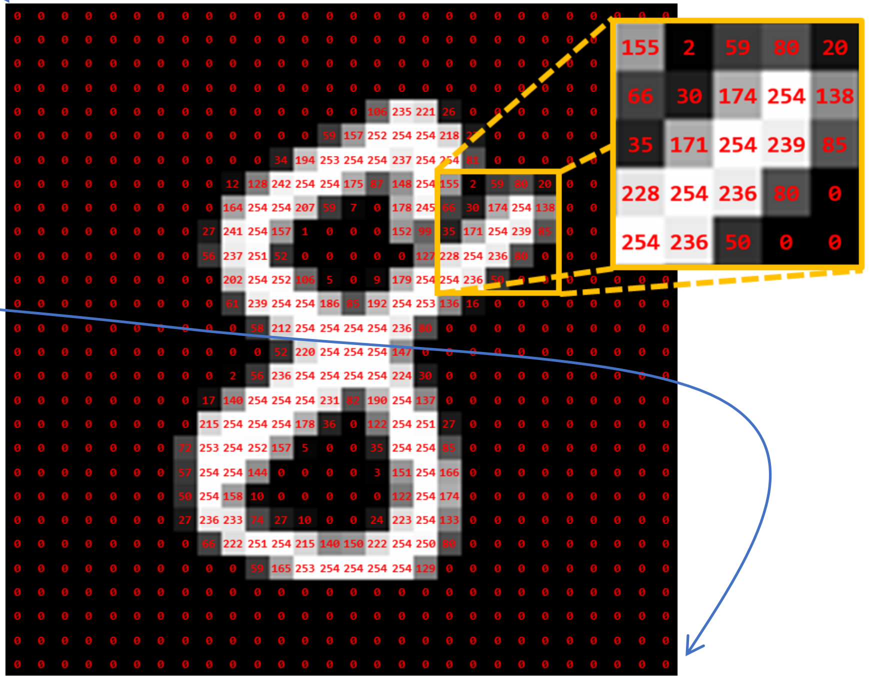
- Linear model for digit recognition

$$h_\Theta(\mathbf{x}) = \frac{1}{1 + e^{-\Theta^\top \mathbf{x}}}$$

Logistic regression

- Feature vector $\mathbf{x} = ?$
- How about raw pixel values?

$$\mathbf{x} = [x_1, x_2, \dots, x_n]^\top$$



$$\Theta^\top \mathbf{x} = \Theta_0 x_0 + \Theta_1 x_1 + \Theta_2 x_2 + \cdots + \Theta_n x_n$$

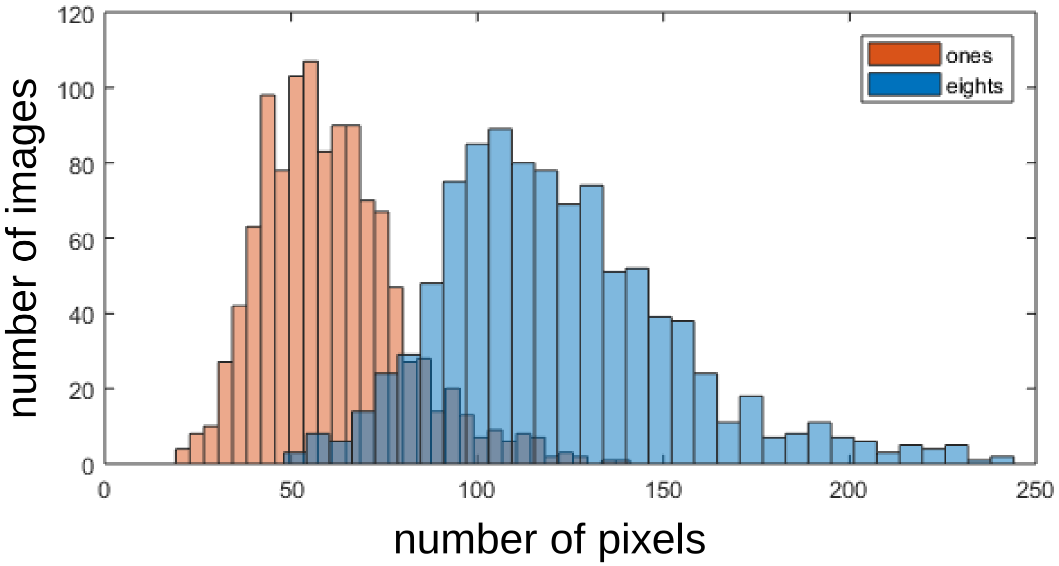
$x_0 = 1$

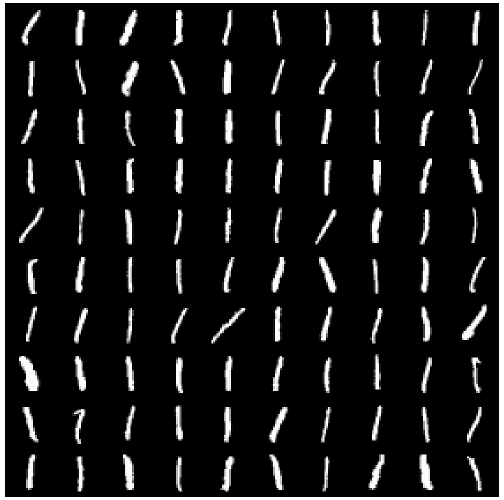
Many parameters!*

*785 (including bias) in case of 28-by-28 pixels.

Example: Digit recognition

- Can we reduce the number of features?
- Counting the number of “white” pixels



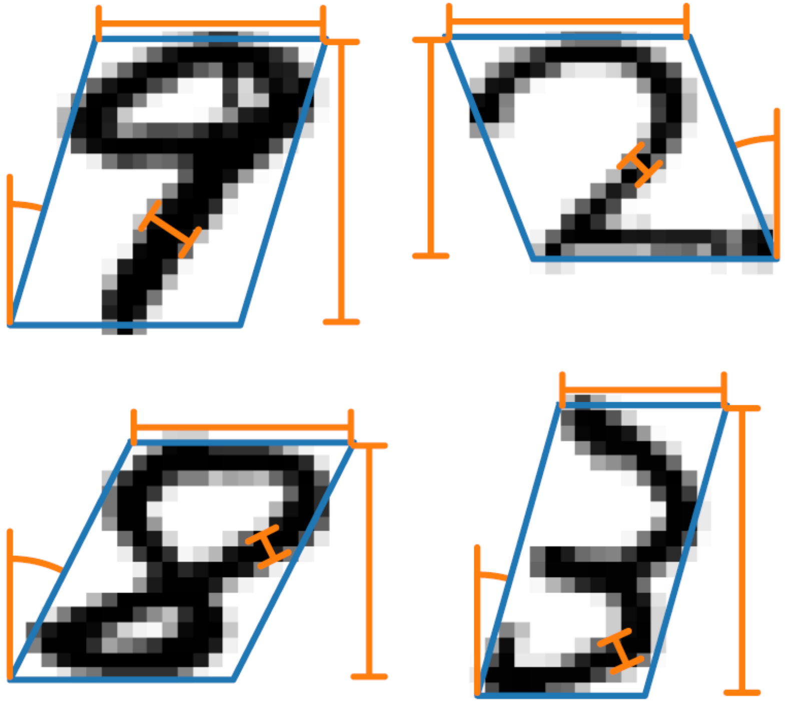


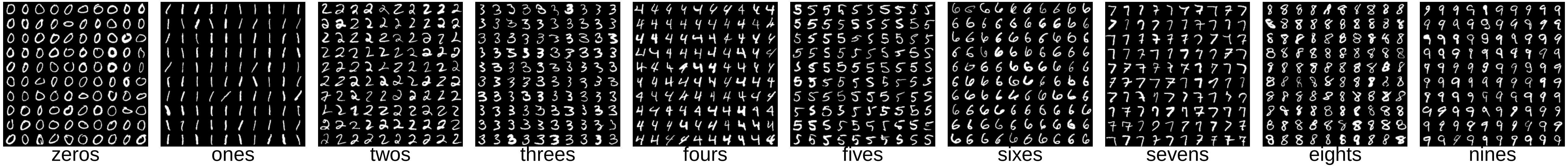
ones

eights

Example: Digit recognition

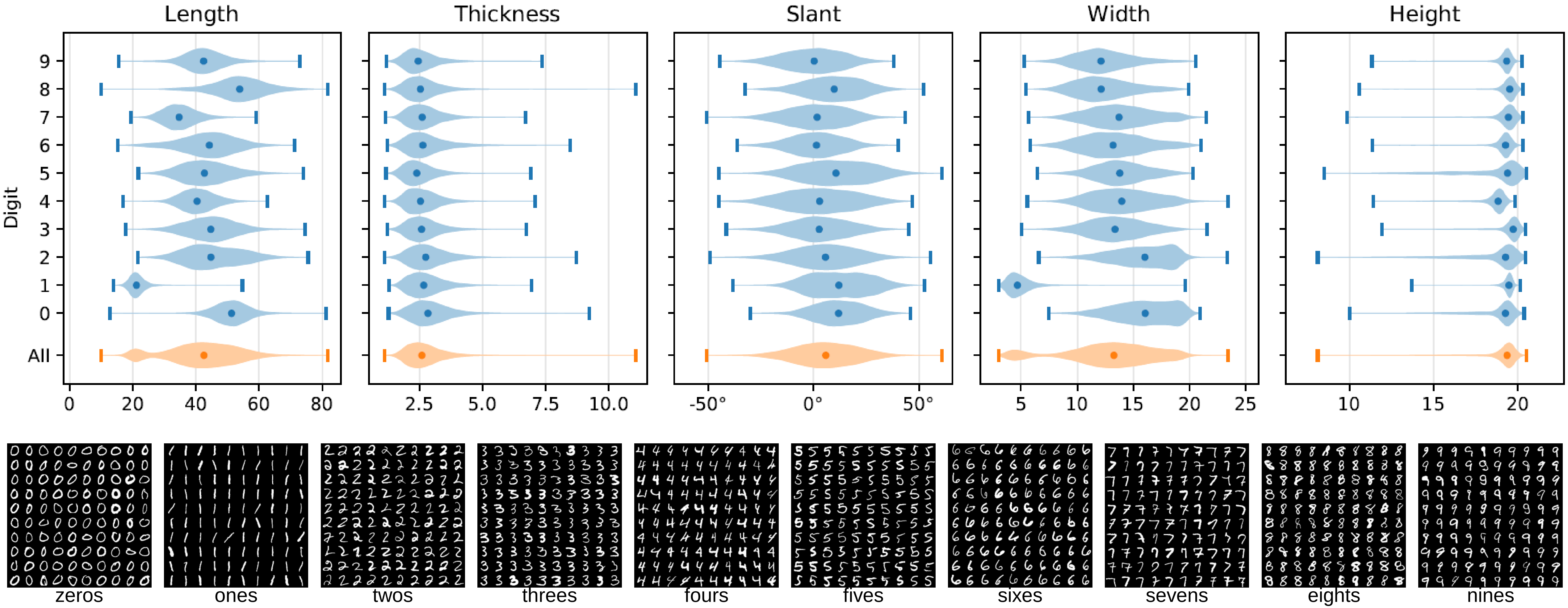
- What else could we “measure”?
- Digit width, height, slant, thickness, length...





Castro et al. 2018. https://arxiv.org/abs/1809.10780

Example: Digit recognition



Castro et al. 2018. https://arxiv.org/abs/1809.10780

Example: Digit recognition

- Linear model for digit recognition $h_\Theta(\mathbf{x}) = \dfrac{1}{1 + e^{-\Theta^\top \mathbf{x}}}$
- Feature vector $\mathbf{x} = ?$

Logistic regression

$$\mathbf{x} = [\text{length}, \text{thickness}, \text{slant}, \text{width}, \text{height}]^\top$$

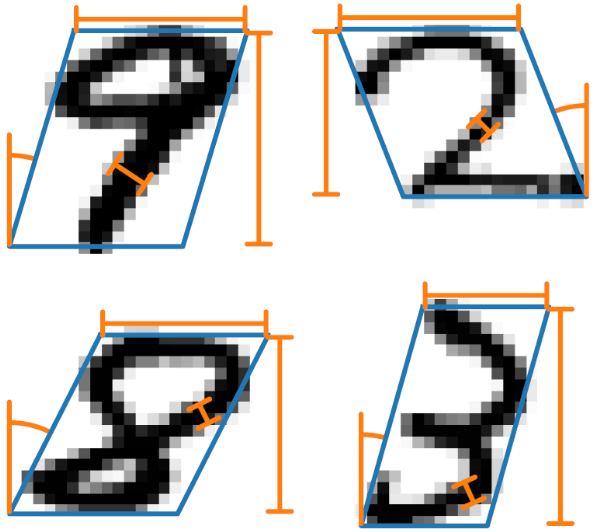
$$\Theta^\top \mathbf{x} = \Theta_0 + \Theta_1 \text{length} + \Theta_2 \text{thickness} \ldots$$

$$\ldots + \Theta_3 \text{slant} + \Theta_4 \text{width} + \Theta_5 \text{height}$$

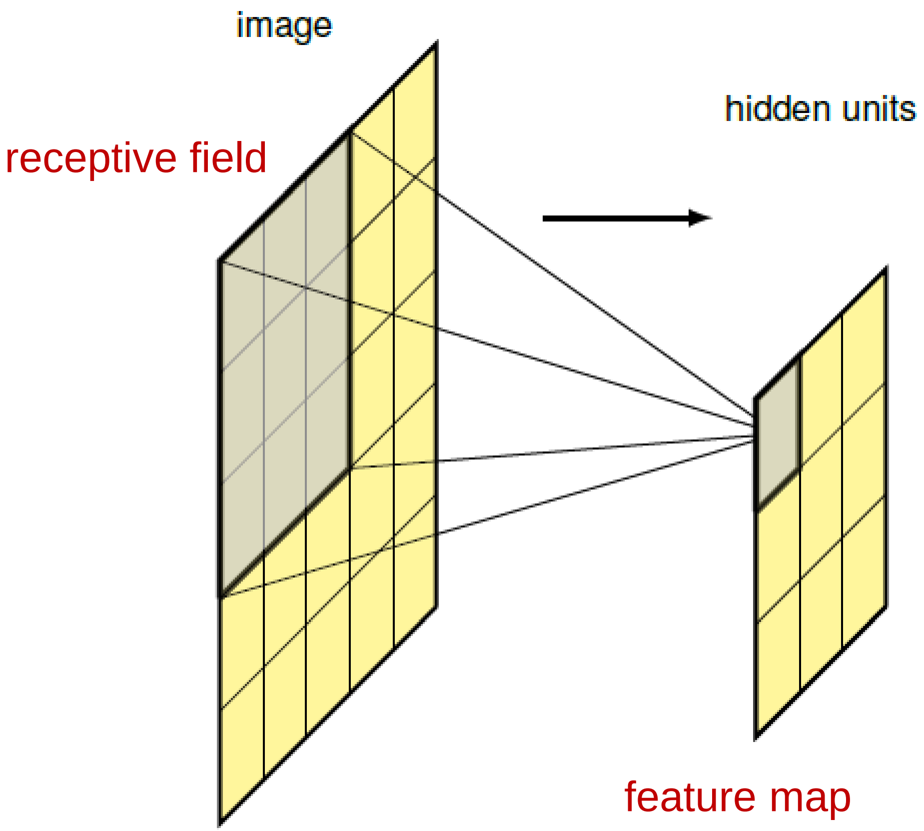
Tutorial 0 - Introduction

Feature learning

- Hand-crafting the right set of features can be challenging
- We might not even know what features to extract
- Can we learn the features itself?



Convolutional filters



filter / kernel

0.4	1.7	0.9
2.3	−2.1	4.0
−1.4	0.7	2.1

weights

$$z = \mathrm{ReLU}(\mathbf{w}^{\top}\mathbf{x} + w_0)$$

activation

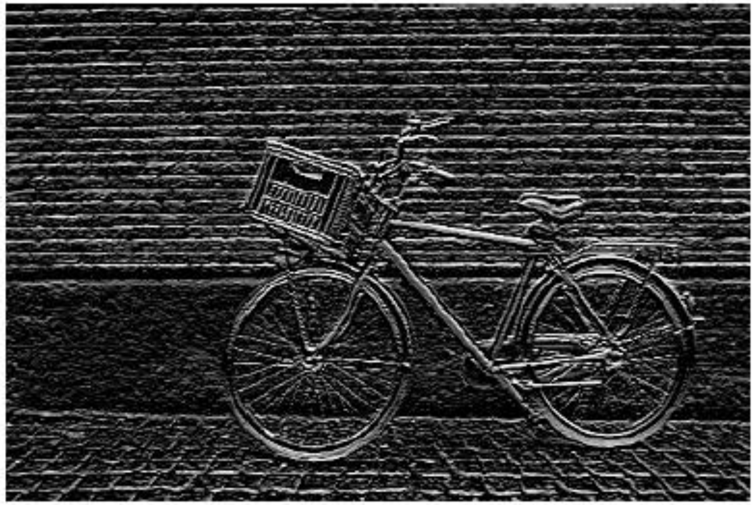
activation function

Bishop & Bishop. Deep Learning: Foundations and Concepts. 2023

Feature detectors

−1	0	1
−1	0	1
−1	0	1

−1	−1	−1
0	0	0
1	1	1

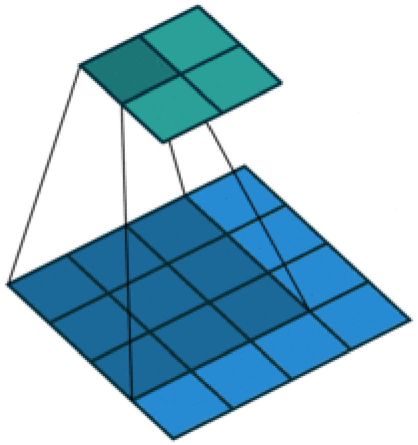


Convolutional filters are translation equivariant

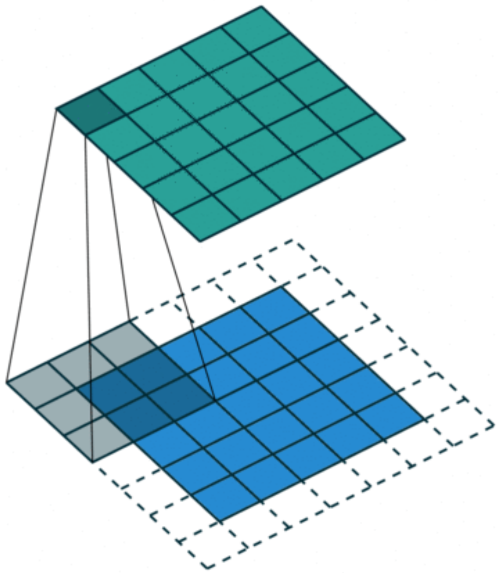
Padding & strides

output size $\left\lfloor \frac{M + 2P - K}{S} \right\rfloor + 1$

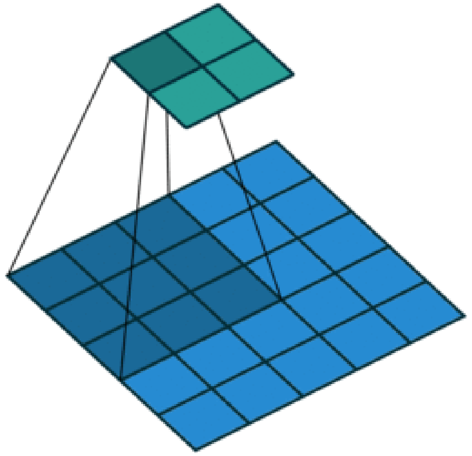
M=input size; K=kernel size; P=padding; S=stride



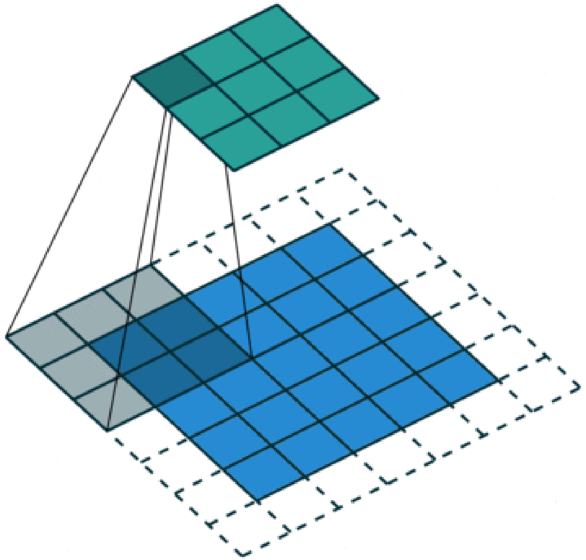
no padding, unit strides



padding



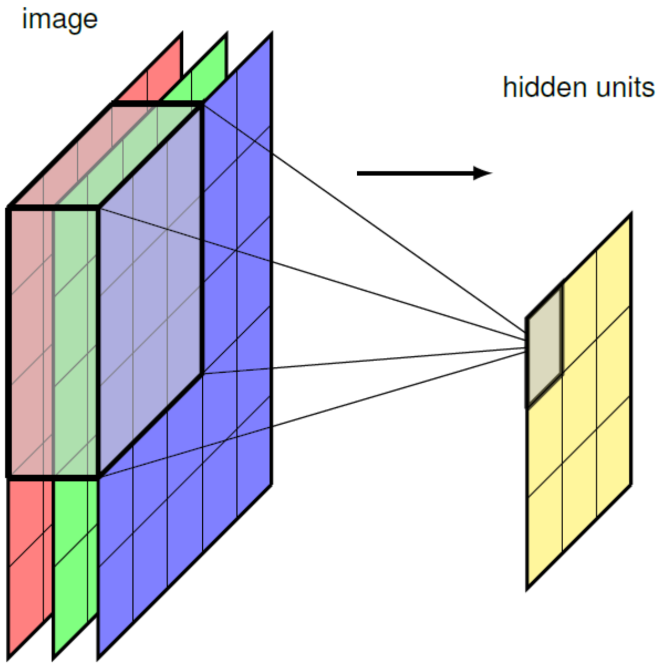
non-unit strides

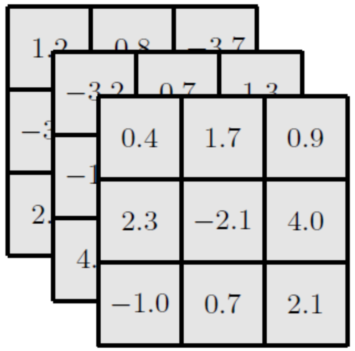


padding, non-unit strides

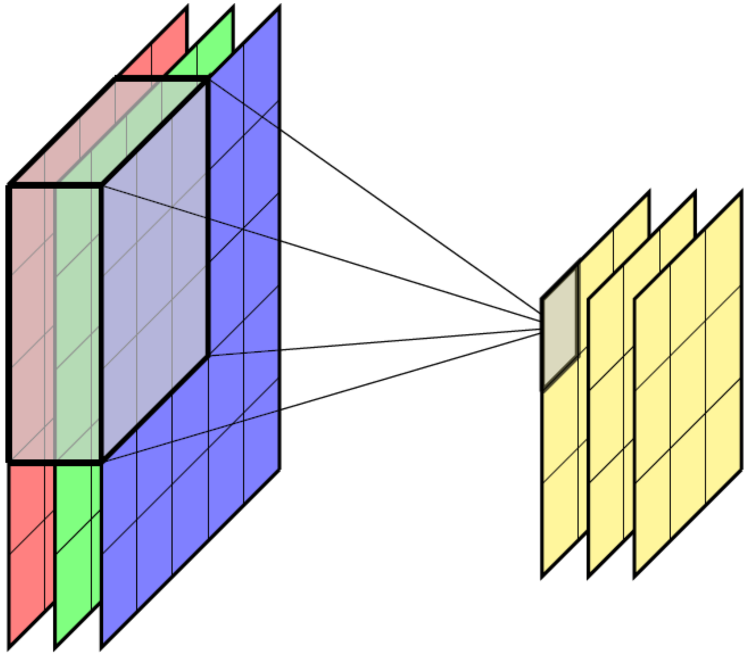
Source: https://github.com/vdumoulin/conv_arithmetic

Multi-dimensional convolutions



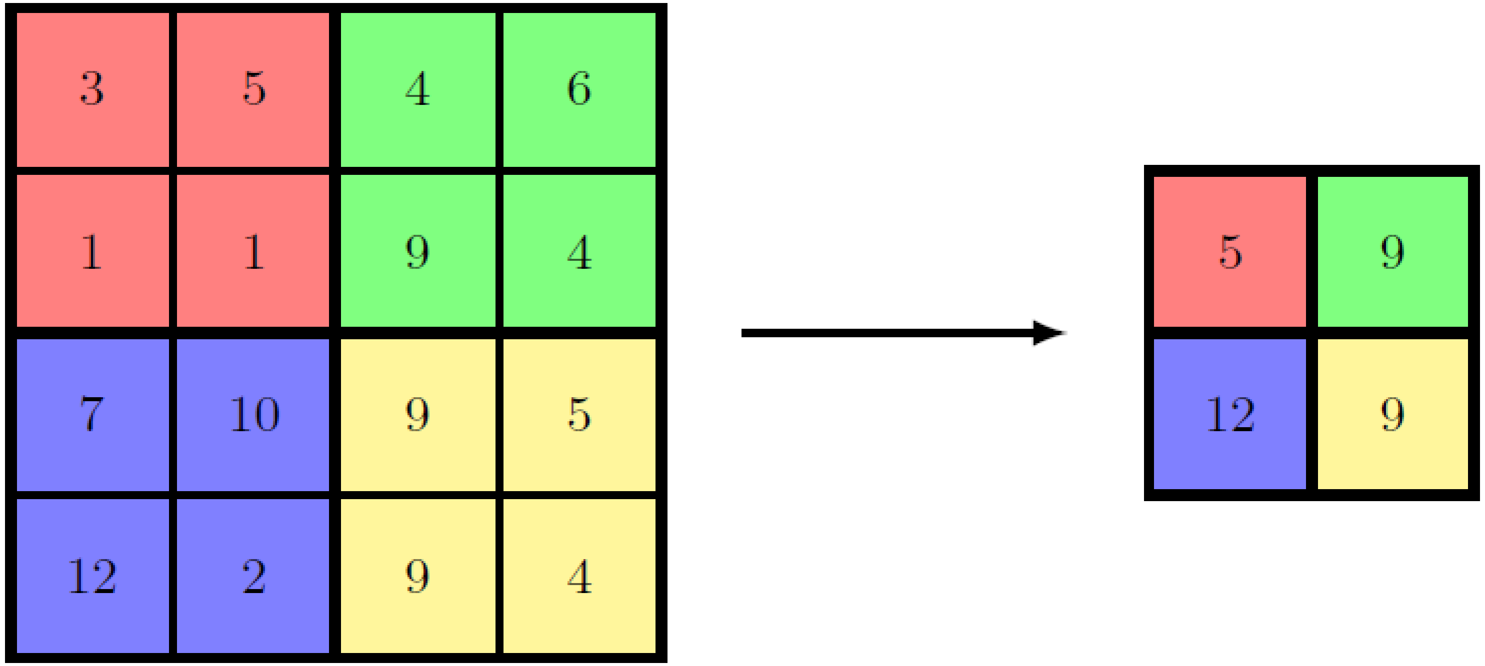


multiple input channels



multiple output channels

Pooling



Pooling yields some local translation invariance

Convolutional network architecture

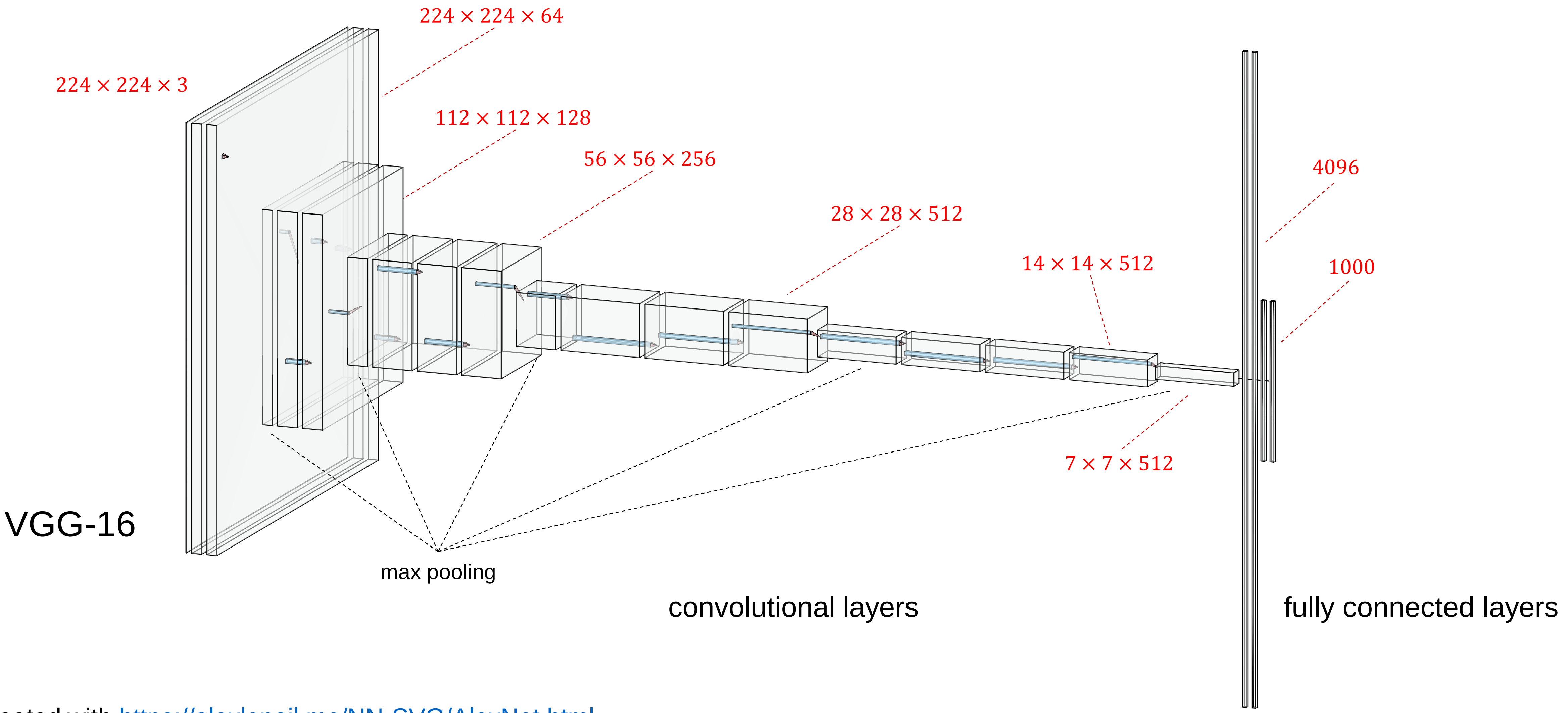
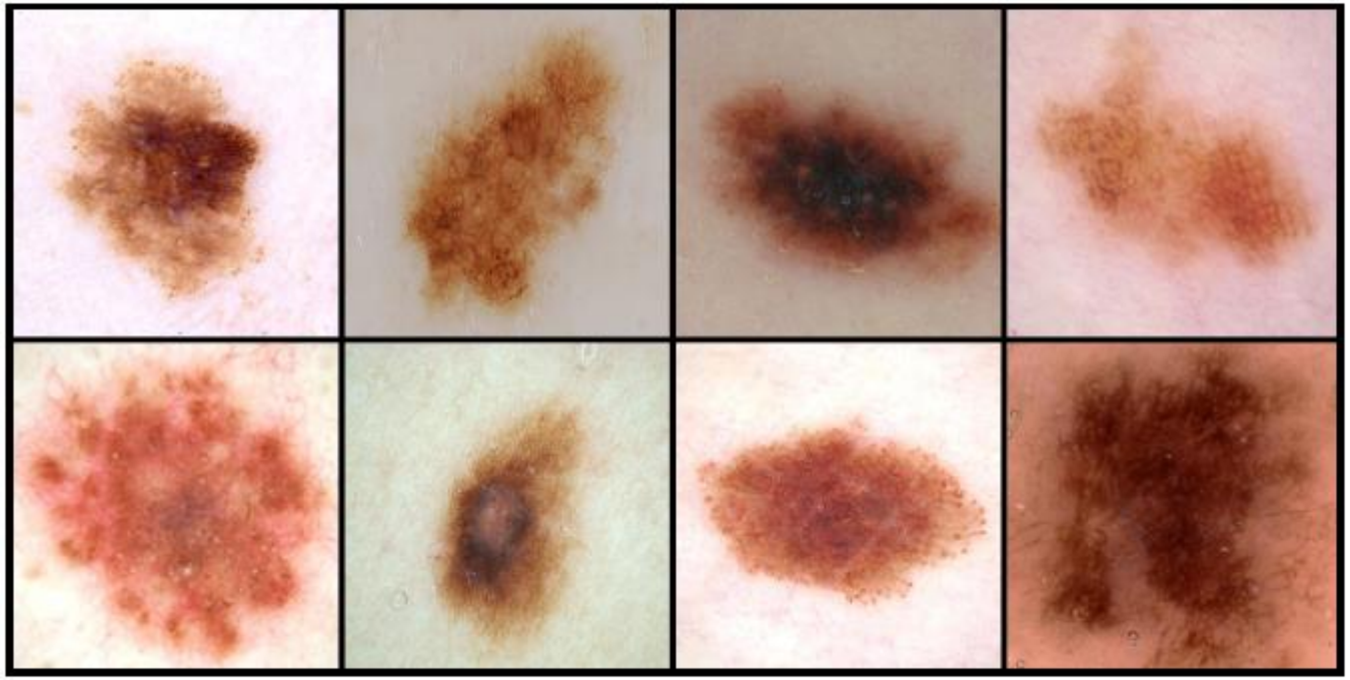


Figure created with https://alexlenail.me/NN-SVG/AlexNet.html

Example: Skin Cancer Diagnosis

- Image classification
- Supervised learning
- Transfer learning



Tutorial 1 - Classification