

# **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	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

Despite practical challenges, we are hopeful that informed discussions among policy-makers and the public about data and the capabilities of machine learning, will lead to insightful designs of programs and policies that can balance the goals of reaping the benefits to scientific research and to individual and public health. Our commitments to privacy and fairness are evergreen, but our policy choices must adapt to advance them, and support new techniques for deepening our knowledge.

**REFERENCES AND NOTES**

1. M. De Choudhury, S. Counts, E. Horvitz, A. Hoff, in Proceedings of International Conference on Weblogs and Social Media (Association for the Advancement of Artificial Intelligence (AAAI), Palm Ilets, CA, 2014).
2. J. S. Brownstein, C. C. Treille, L. C. Madoff, N. Engl. J. Med. 360, 2153–2155 (2009).
3. G. Eyherarach, J. Med. Internet Res. 11, e11 (2009).
4. D. Broniatowski, M. J. Paul, M. Dretke, PLOS ONE 8, e63672 (2013).
5. A. Sadak, H. Kautz, V. Silenzio, in Proceedings of the Twenty Sixth AAAI Conference on Artificial Intelligence (AAAI, Palo Alto, CA, 2012).
6. M. De Choudhury, S. Counts, E. Horvitz, in Proceedings of SIGCHI Conference on Human Factors in Computing Systems (Association for Computing Machinery, New York, 2013), pp. 3257–3276.
7. R. W. Pharmacol. Ther. 96, 239–246 (2004).
8. Samaritan Radar, [www.samaritans.org/how-can-we-help-you/someone-some-time/samaritans-radar](http://www.samaritans.org/how-can-we-help-you/someone-some-time/samaritans-radar); <http://tinyurl.com/Samaritan-after-911>.
9. Shut down Samaritan Radar: <http://tinyurl.com/Samaritan-after-911>.
10. U.S. Equal Employment Opportunity Commission (EEOC), 29 CFR 1625.3 (e) (2013).
11. EEOC, 29 CFR 1625.3 (e) (2013).
12. M. A. Rothman, J. Law Med. Ethics 36, 837–840 (2008).
13. Executive Office of the President, Big Data Seizing Opportunities: Preserving Values While Harnessing Data (White House, Washington, DC, 2014). <http://bigdata.gov/factsheet>.
14. Letter from Manisha Mittal, FTC, to Reed Freeman, Morrison & Foerster LLP, Counsel for Netlife, 2 (closing letter) (2010); <http://USA.gov/DOJ/FTC/Netlife> (2012).
15. In re Facebook, No. 15-cv-00001, complaint, FTC v. Facebook, Inc., 2015 U.S. Dist. LEXIS 147000 (D.C. 2015); <http://USA.gov/DOJ/FTC/Facebook>.
16. FTC Staff Report, Mobile Privacy Disclosures: Building Trust Through Transparency (FTC, Washington, DC, 2013); <http://USA.gov/DOJ/FTC/MobilePrivacy>.
17. FTC, Protecting Consumer Privacy in an Era of Rapid Change (since 2010) and has become the standard benchmark for large-scale object recognition.<sup>1</sup> ILSVRC follows in the footsteps of the PASCAL VOC challenge (Everingham et al., 2010).

**REVIEW**

## Machine learning: Trends, perspectives, and prospects

M. I. Jordan<sup>1\*</sup> and T. M. Mitchell<sup>2\*</sup>

Machin learning addresses the question of how to build computers that improve automatically through experience. It is one of today's most rapidly growing technical fields, lying at the intersection of computer science and statistics, and at the core of artificial intelligence and data science. Recent progress in machine learning has been driven both by the development of new learning algorithms and theory and by the ongoing explosion in the availability of online data and low-cost computation. The adoption of data-intensive learning methods can be found throughout science, technology and commerce, including machine learning-based decision-making across many walks of life, including financial modeling, policing, and marketing.

**REFERENCES AND NOTES**

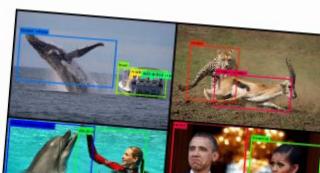
1. M. De Choudhury, S. Counts, E. Horvitz, A. Hoff, in Proceedings of International Conference on Weblogs and Social Media (Association for the Advancement of Artificial Intelligence (AAAI), Palm Ilets, CA, 2014).
2. J. S. Brownstein, C. C. Treille, L. C. Madoff, N. Engl. J. Med. 360, 2153–2155 (2009).
3. G. Eyherarach, J. Med. Internet Res. 11, e11 (2009).
4. D. Broniatowski, M. J. Paul, M. Dretke, PLOS ONE 8, e63672 (2013).
5. A. Sadak, H. Kautz, V. Silenzio, in Proceedings of the Twenty Sixth AAAI Conference on Artificial Intelligence (AAAI, Palo Alto, CA, 2012).
6. M. De Choudhury, S. Counts, E. Horvitz, in Proceedings of SIGCHI Conference on Human Factors in Computing Systems (Association for Computing Machinery, New York, 2013), pp. 3257–3276.
7. R. W. Pharmacol. Ther. 96, 239–246 (2004).
8. Samaritan Radar, [www.samaritans.org/how-can-we-help-you/someone-some-time/samaritans-radar](http://www.samaritans.org/how-can-we-help-you/someone-some-time/samaritans-radar); <http://tinyurl.com/Samaritan-after-911>.
9. Shut down Samaritan Radar: <http://tinyurl.com/Samaritan-after-911>.
10. U.S. Equal Employment Opportunity Commission (EEOC), 29 CFR 1625.3 (e) (2013).
11. EEOC, 29 CFR 1625.3 (e) (2013).
12. M. A. Rothman, J. Law Med. Ethics 36, 837–840 (2008).
13. Executive Office of the President, Big Data Seizing Opportunities: Preserving Values While Harnessing Data (White House, Washington, DC, 2014). <http://bigdata.gov/factsheet>.
14. Letter from Manisha Mittal, FTC, to Reed Freeman, Morrison & Foerster LLP, Counsel for Netlife, 2 (closing letter) (2010); <http://USA.gov/DOJ/FTC/Netlife> (2012).
15. In re Facebook, No. 15-cv-00001, complaint, FTC v. Facebook, Inc., 2015 U.S. Dist. LEXIS 147000 (D.C. 2015); <http://USA.gov/DOJ/FTC/Facebook>.
16. FTC Staff Report, Mobile Privacy Disclosures: Building Trust Through Transparency (FTC, Washington, DC, 2013); <http://USA.gov/DOJ/FTC/MobilePrivacy>.
17. FTC, Protecting Consumer Privacy in an Era of Rapid Change (since 2010) and has become the standard benchmark for large-scale object recognition.<sup>1</sup> ILSVRC follows in the footsteps of the PASCAL VOC challenge (Everingham et al., 2010).

## YOLO9000: Better, Faster, Stronger

Joseph Redmon<sup>\*x</sup>, Ali Farhadi<sup>\*t,x</sup>  
University of Washington<sup>x</sup>, Allen Institute for AI<sup>t</sup>, XNOR.ai<sup>x</sup>  
<http://pjreddie.com/yolo9000/>

### Abstract

We introduce YOLO9000, a state-of-the-art, real-time object detection system that can detect over 9000 object categories. First we propose various improvements to the YOLO detection method, both novel and drawn from prior work. The improved model, YOLOv2, is state-of-the-art on standard detection tasks like PASCAL VOC and COCO. Using a novel, multi-scale training method the same YOLOv2 model can run at varying sizes, offering an easy tradeoff between speed and accuracy. At 67 FPS, YOLOv2 gets 76.8 mAP on VOC 2007. At 40 FPS, YOLOv2 gets 76.8 mAP on COCO 2014.



Under review as a conference paper at ICLR 2016

## UNSUPERVISED REPRESENTATION LEARNING WITH DEEP CONVOLUTIONAL GENERATIVE ADVERSARIAL NETWORKS

Alec Radford & Luke Metz  
indico Research  
Boston, MA  
[{alec,luke}@indico.io](mailto:{alec,luke}@indico.io)

Soumith Chintala  
Facebook AI Research  
New York, NY

### ABSTRACT

supervised learning with convolutional networks (CNNs) has received less attention. In this work we hope to help introduce a class of CNNs for supervised learning and unsupervised learning called deep convolutional generative models (DCGANs), that have certain architectural constraints, and they are a strong candidate for unsupervised learning. Training datasets, we show convincing evidence that our deep convolutional generator and discriminator, learn a hierarchy of representations from object parts to whole objects. We also show that our learned representations are general enough to be used for other tasks - demonstrating their applicability as general image representations.

arXiv:1901.10002v3 [cs.CV] 17 Feb 2020

## ImageNet Large Scale Visual Recognition Challenge

Olga Russakovsky<sup>1</sup> · Jia Deng<sup>2</sup> · Hao Su<sup>1</sup> · Jonathan Krause<sup>1</sup> · Sanjeev Satheesh<sup>1</sup> · Sean Ma<sup>3</sup> · Zhiheng Huang<sup>1</sup> · Andrej Karpathy<sup>1</sup> · Aditya Khosla<sup>3</sup> · Michael Bernstein<sup>1</sup> · Alexander C. Berg<sup>4</sup> · Li Fei-Fei<sup>1</sup> · Phillips et al. [2011]

Received: 31 August 2014 / Accepted: 12 March 2015 / Published online: 11 April 2015  
© Springer Science+Business Media New York 2015

## A Framework for Understanding Unintended Consequences of Machine Learning

Harini Suresh  
MIT  
[hsuresh@mit.edu](mailto:hsuresh@mit.edu)

John V. Guttag  
MIT  
[guttag@mit.edu](mailto:guttag@mit.edu)

### Abstract

As machine learning increasingly affects people and society, it is important that we strive for a comprehensive and unbiased understanding of potential sources of unwanted consequences. For instance, downstream harms to particular groups are often blamed on "biased data," but this concept encompasses too many issues to be useful in developing solutions. In this paper, we provide a framework that partitions sources of downstream harm in machine learning into six distinct categories spanning the data generation and machine learning pipeline. We describe how these categories are relevant to particular applications and how they may or may not contribute to unintended consequences of machine learning.

Int J Comput Vis (2015) 115:211–252  
DOI 10.1007/s11263-015-0816-y

## Fully Convolutional Networks for Semantic Segmentation

Jonathan Long<sup>\*</sup> · Evan Shelhamer<sup>\*</sup> · Trevor Darrell  
UC Berkeley  
[{jonlong,shelhamer,trevor}@cs.berkeley.edu](mailto:{jonlong,shelhamer,trevor}@cs.berkeley.edu)

### Abstract

Convolutional networks are powerful visual models that yield hierarchies of features. We show that convolutional networks by themselves, trained end-to-end, pixels-to-pixels, exceed the state-of-the-art in semantic segmentation. Our key insight is to build "fully convolutional" networks that take input of arbitrary size and produce a spatially-dense output with efficient inference and minimal storage. This allows us to fully utilize the space of fully convolutional networks and detail the space of spatially dense prior models. We show that a fully convolutional network (FCN) can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation, end-to-end, pixels-to-pixels on semantic segmentation datasets without further machine learning. This is the first work to train FCNs from scratch, without pre-training, and to do so without further machine learning.

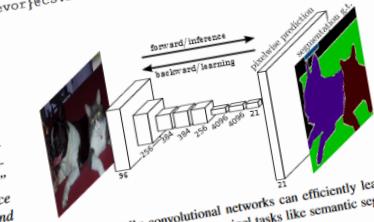


Figure 1. Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

We show that a fully convolutional network (FCN) can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation. This is the first work to train FCNs end-to-end, pixels-to-pixels on semantic segmentation datasets without further machine learning. We show that a fully convolutional network (FCN) can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation.

Consider the following toy scenario: an engineer building a smile-detection system observes that the system has a higher false negative rate for women. Over the next week, she collects many more images of women, so that the proportions of men and women are now equal, and is happy to see the performance on the female subjects has improved while her co-worker has not.



The ImageNet Large Scale Visual Recognition Classification

(since 2010) and has become the standard benchmark for large-scale object recognition.<sup>1</sup> ILSVRC follows in the footsteps of the PASCAL VOC challenge (Everingham et al., 2010).

# 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

The screenshot shows a Jupyter Notebook interface with the following details:

- Title:** tutorial\_1\_classification.ipynb
- Table of Contents:** Tutorial 1 - Classification, expanded to show:
  - Data
  - Model
  - Training
  - Validation
  - Testing
  - Testing on out-of-distribution data
  - Visualisation
  - Test data examples
  - Weights
  - Logging
- Code Cell:** [ ] # On Google Colab uncomment the following line to install PyTorch Lightning  
# ! pip install lightning
- Code Cell:** [ ] import numpy as np  
import torch  
import torch.nn as nn  
import torch.nn.functional as F  
import torchvision  
import matplotlib  
import matplotlib.pyplot as plt
- Code Cell:** [ ] from torch.utils.data import random\_split, DataLoader  
from torchvision import transforms  
from torchvision.datasets import MNIST  
from pytorch\_lightning import LightningModule, LightningDataModule, Trainer, seed\_everything  
from pytorch\_lightning.loggers import TensorBoardLogger  
from pytorch\_lightning.callbacks import ModelCheckpoint, TQDMProgressBar  
from torchmetrics.functional import accuracy
- Section:** Data
- Note:** We use a [LightningDataModule](#) for handling the MNIST dataset.

# 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

# Machine Learning for Imaging

## Introduction

Ben Glocker

[b.glocker@imperial.ac.uk](mailto:b.glocker@imperial.ac.uk)



Image source: pixabay.com, CC0 Creative Commons

# Where was this image taken?



# What animals are in this image?



# Are there birds in this image?



# Where are the birds in this image?



# What type of bird is in this image?



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



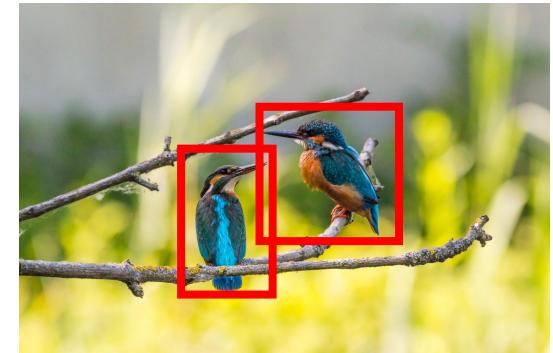
Output: Category (e.g., “bird”)

Object Detection



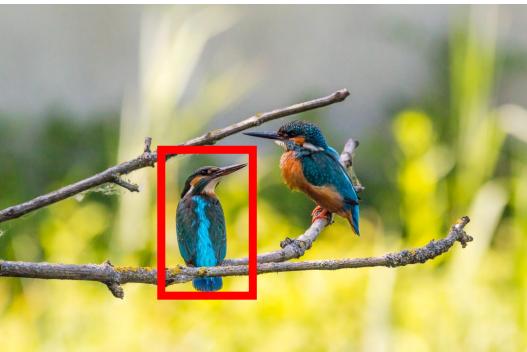
Output: Coordinates (e.g., centroid)

Object Localisation



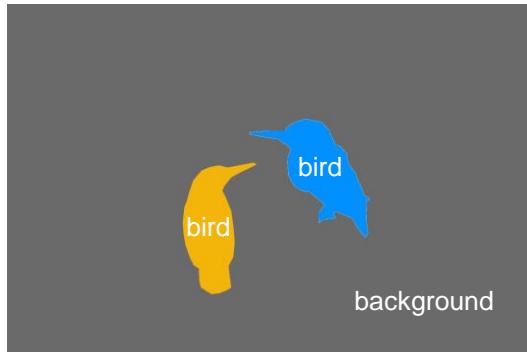
Output: Coordinates (e.g., bounding box)

Object Recognition



Output: Category (e.g., “kingfisher”)

Semantic Segmentation



Output: Labelmap

Image Captioning



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

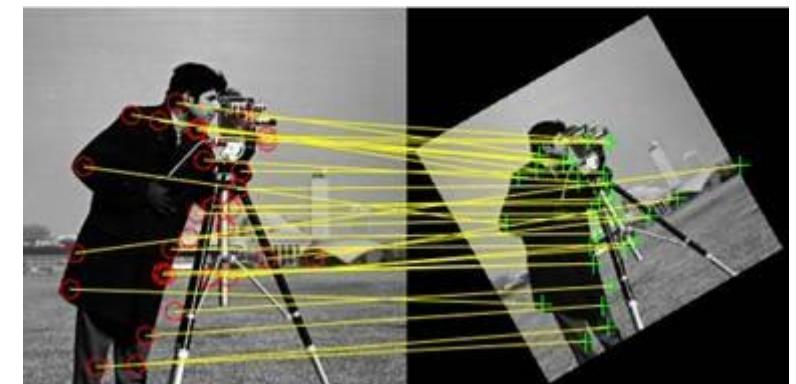
# Common image analysis tasks

Image Registration



Output: Transformation

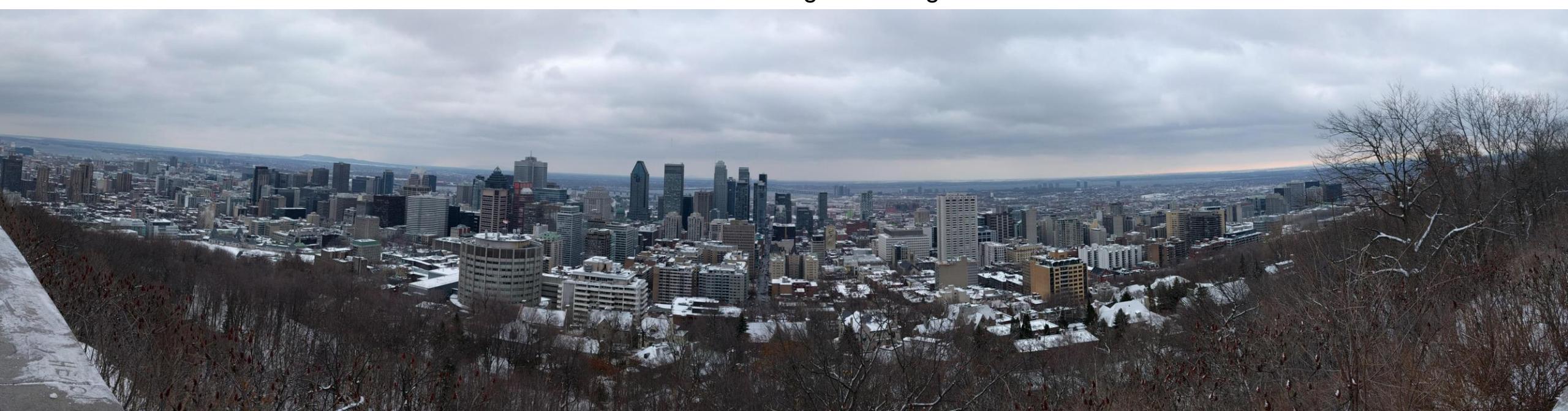
Point Correspondences



Output: Pairs of Coordinates

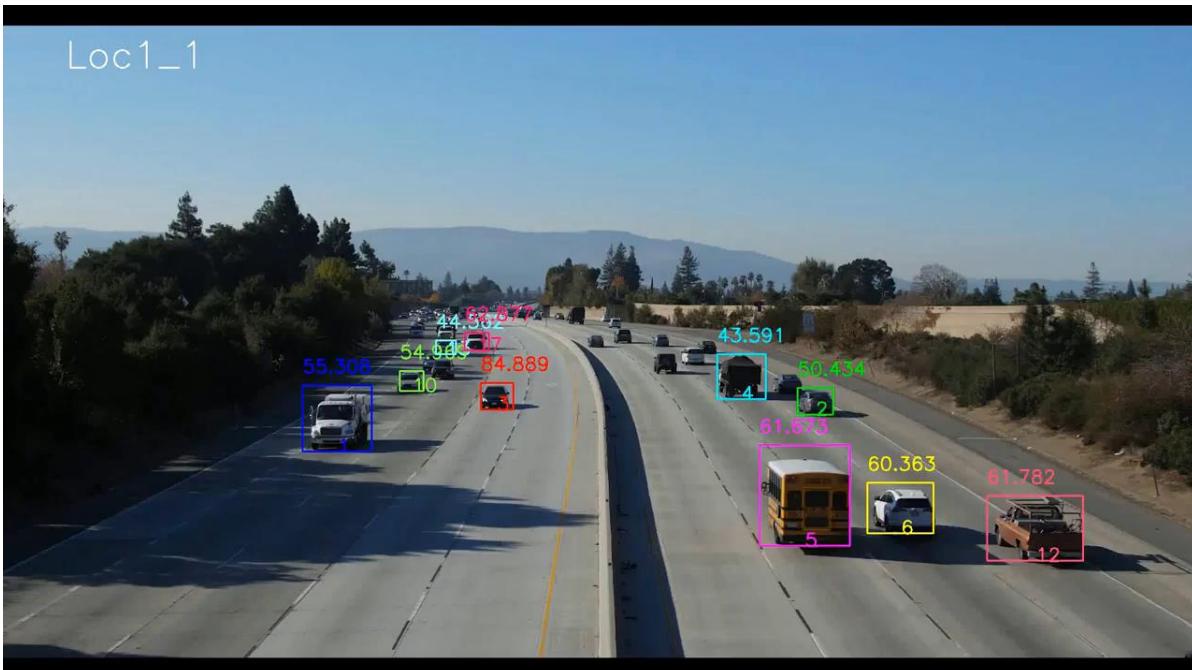
# Common image analysis tasks

Panoramic Image Stitching



# Common image analysis tasks

Object Tracking



Source: [https://youtu.be/\\_i4numqiv7Y](https://youtu.be/_i4numqiv7Y)

Face Tracking

Face2Face: Real-time Face Capture  
and Reenactment of RGB Videos

*Justus Thies<sup>1</sup>, Michael Zollhöfer<sup>2</sup>,  
Marc Stamminger<sup>1</sup>, Christian Theobalt<sup>2</sup>,  
Matthias Nießner<sup>3</sup>*

<sup>1</sup>University of Erlangen-Nuremberg

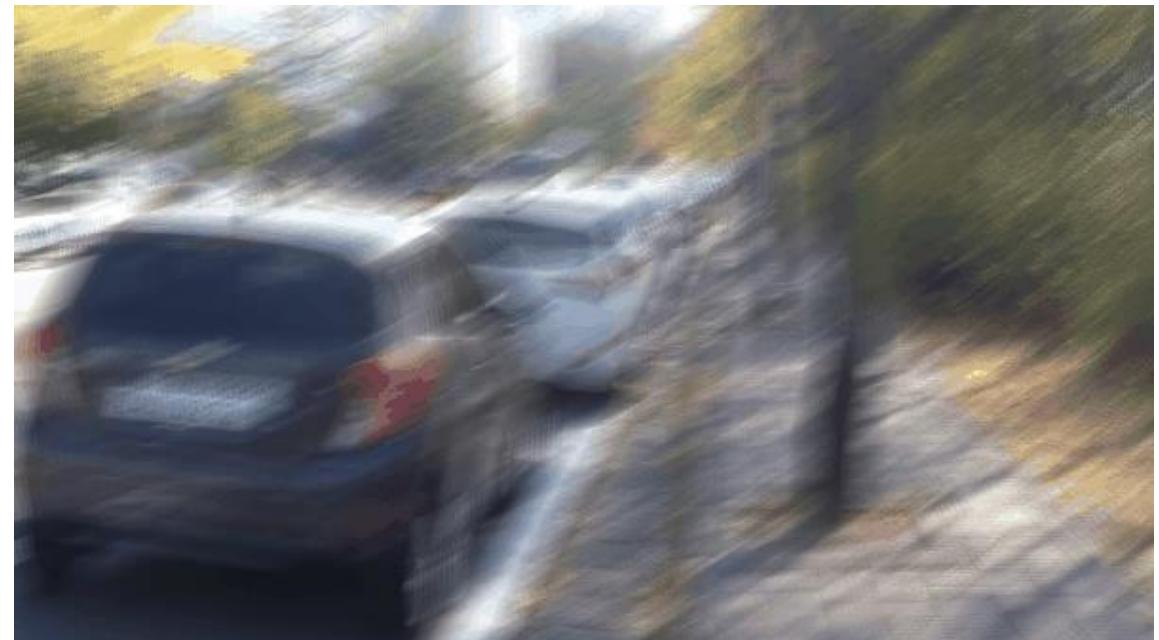
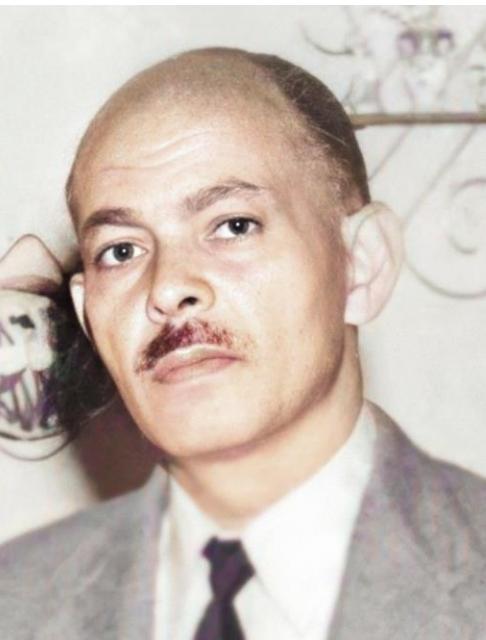
<sup>2</sup>Max-Planck-Institute for Informatics

<sup>3</sup>Stanford University

CVPR 2016 (Oral)

Source: <https://youtu.be/ohmajJTpNk>

# Image restoration



# Image synthesis



# 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”**



# Style transfer



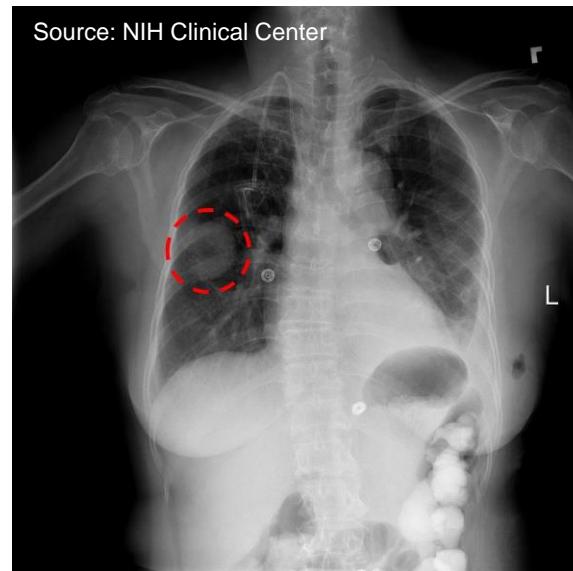
# Medical imaging applications

Image Classification



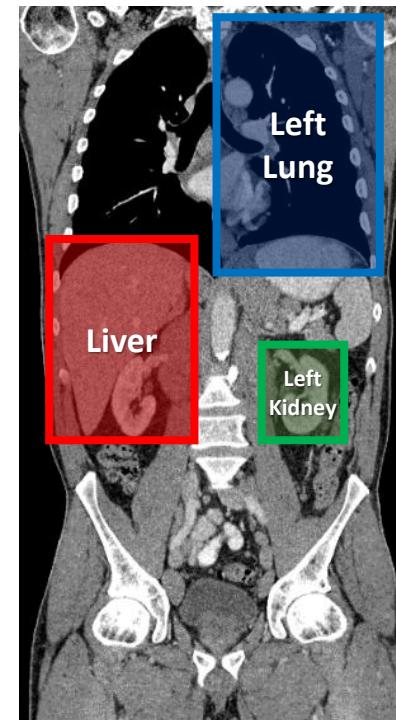
Output: cancer / no cancer

Object Detection



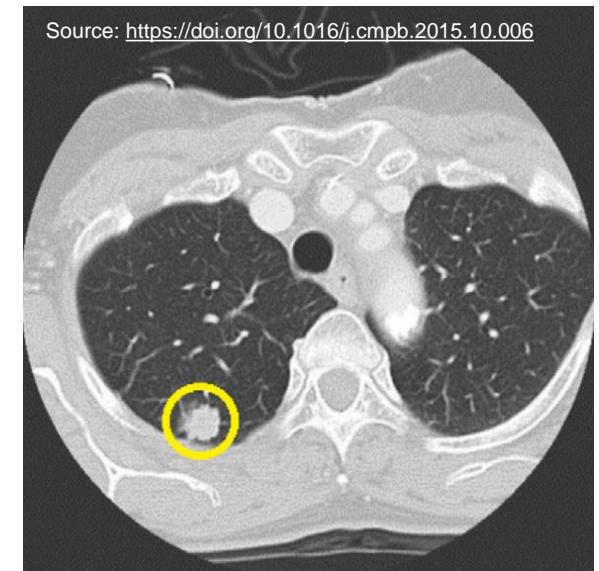
Output: presence and location of a mass

Object Localisation



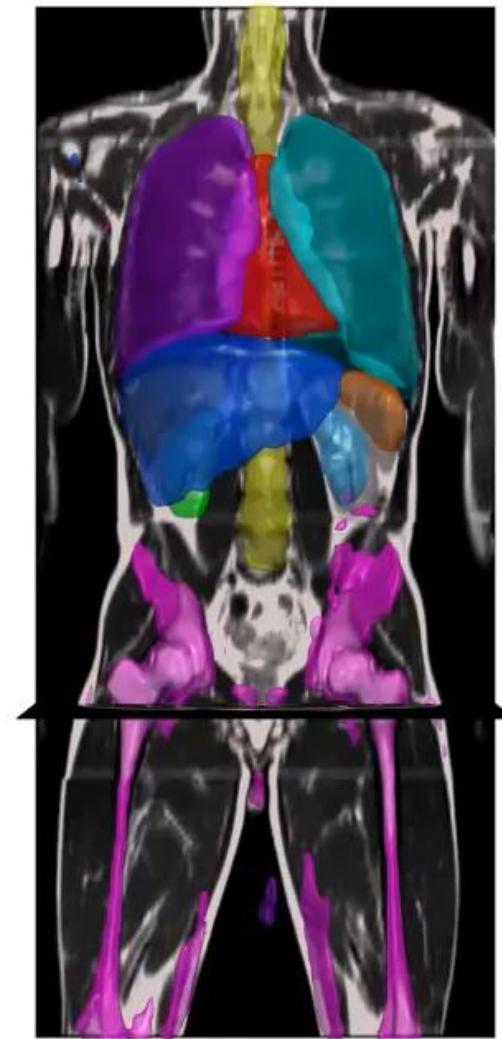
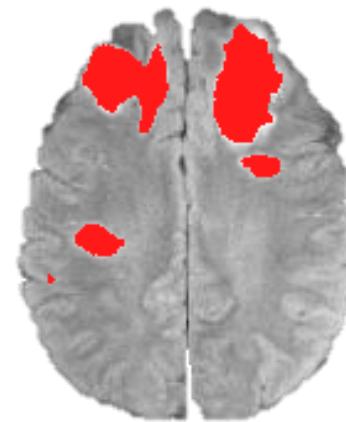
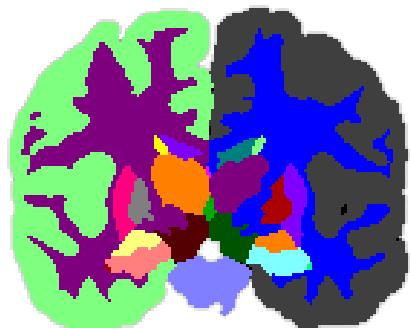
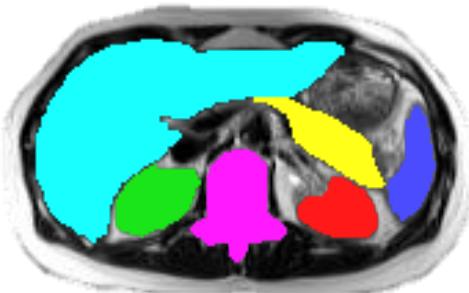
Output: organ bounding boxes

Object Recognition



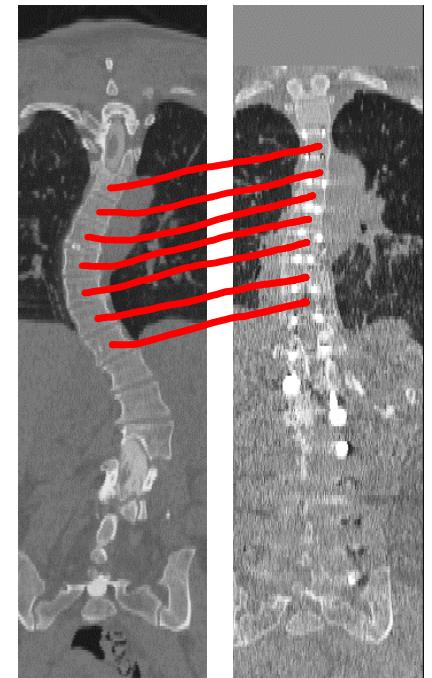
Output: benign / malignant

# Semantic segmentation

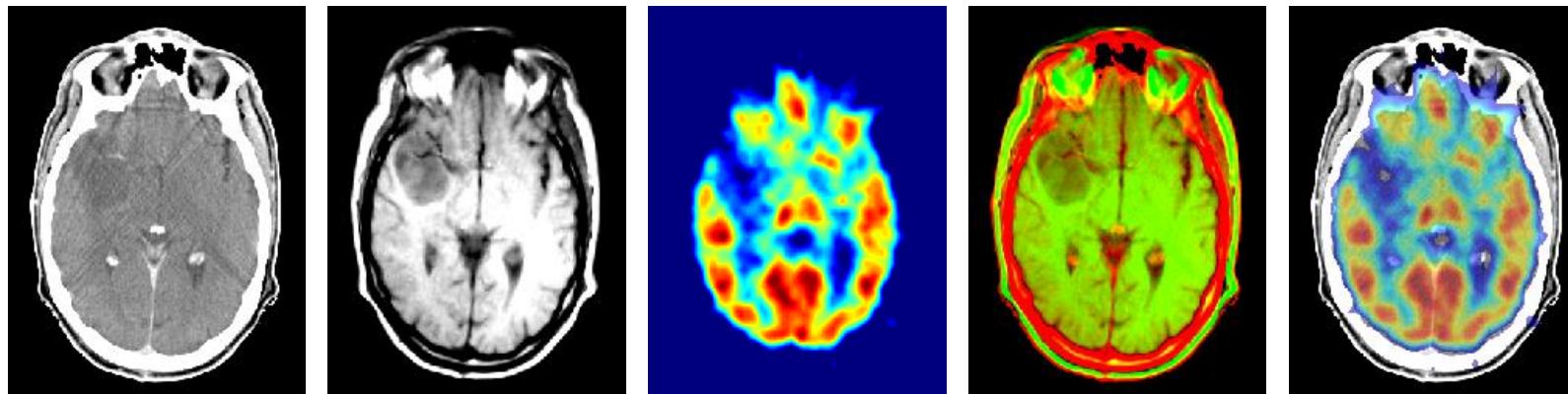


# Image registration

Longitudinal Comparison

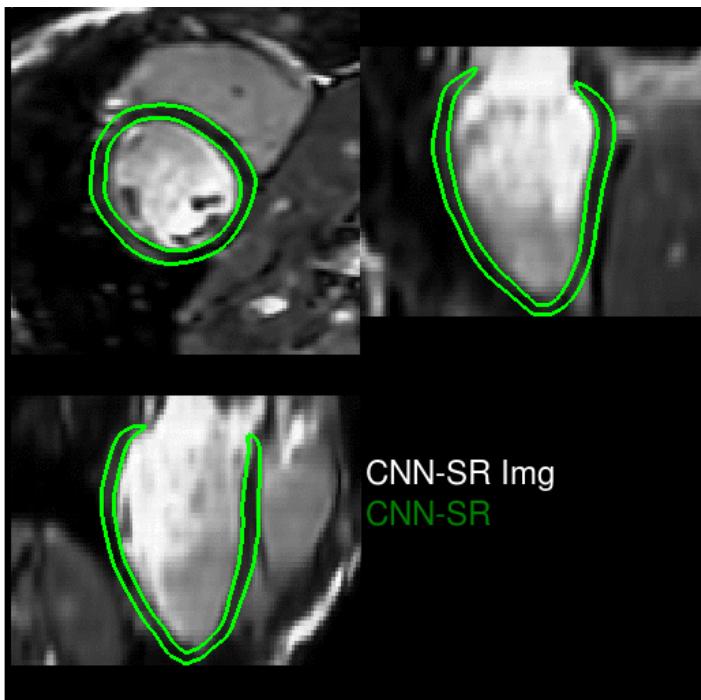


Multi-modal Image Fusion



# Motion estimation & object tracking

Cardiac Motion



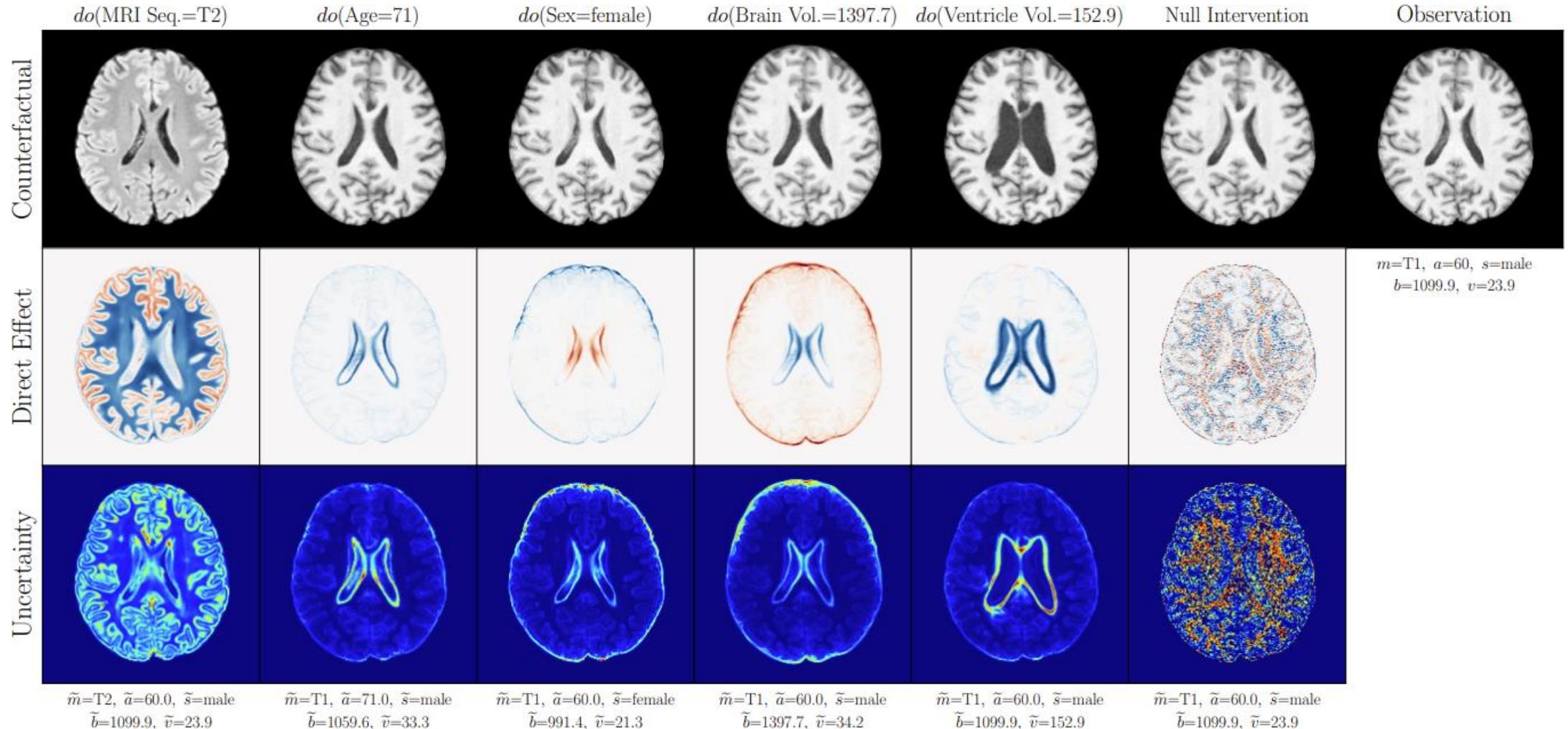
Source: Oktay et al. MICCAI 2016

Cell Tracking

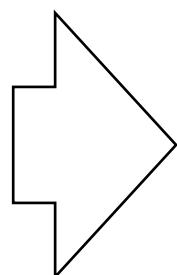
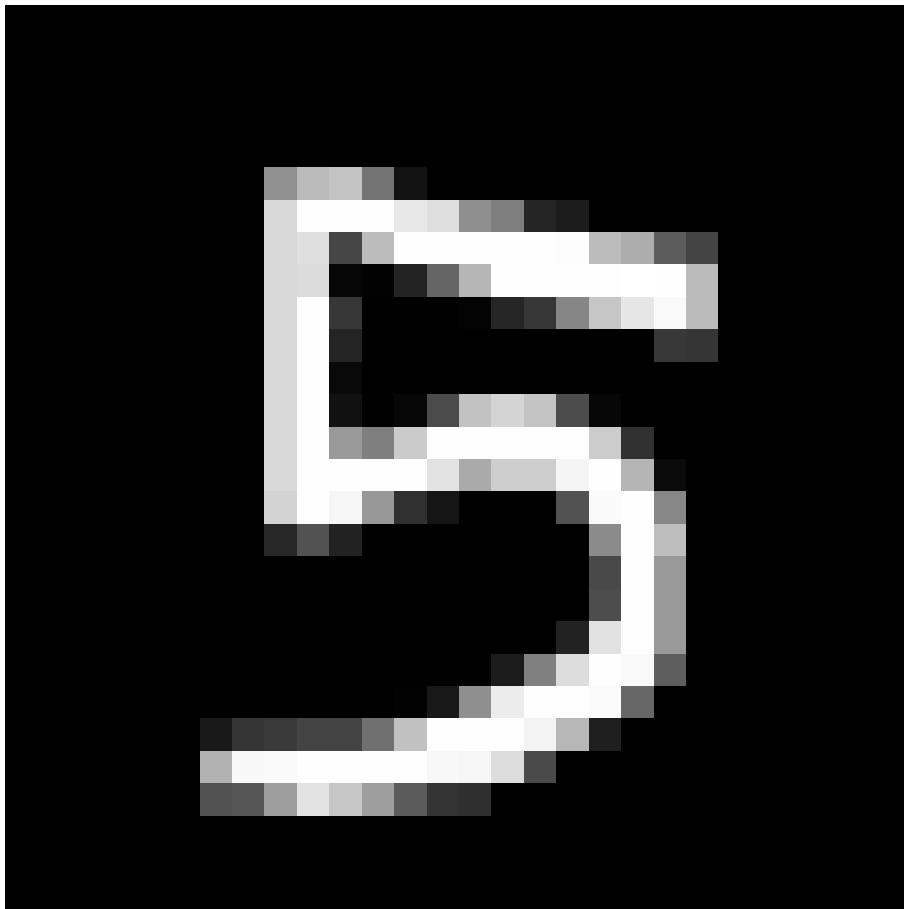


Source: <https://youtu.be/SLYgvHzAm2w>

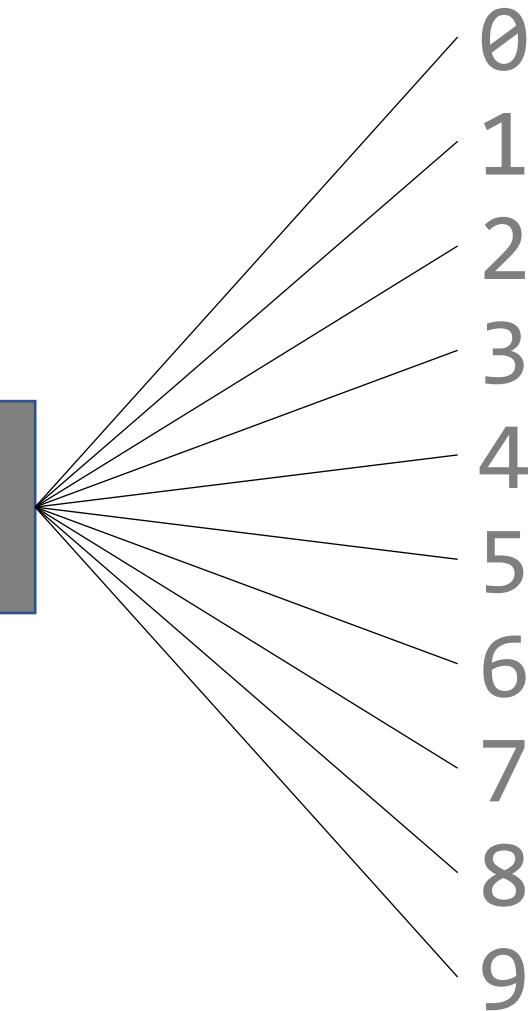
# Counterfactual image synthesis



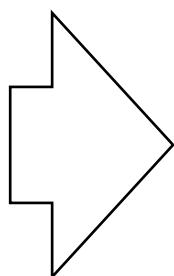
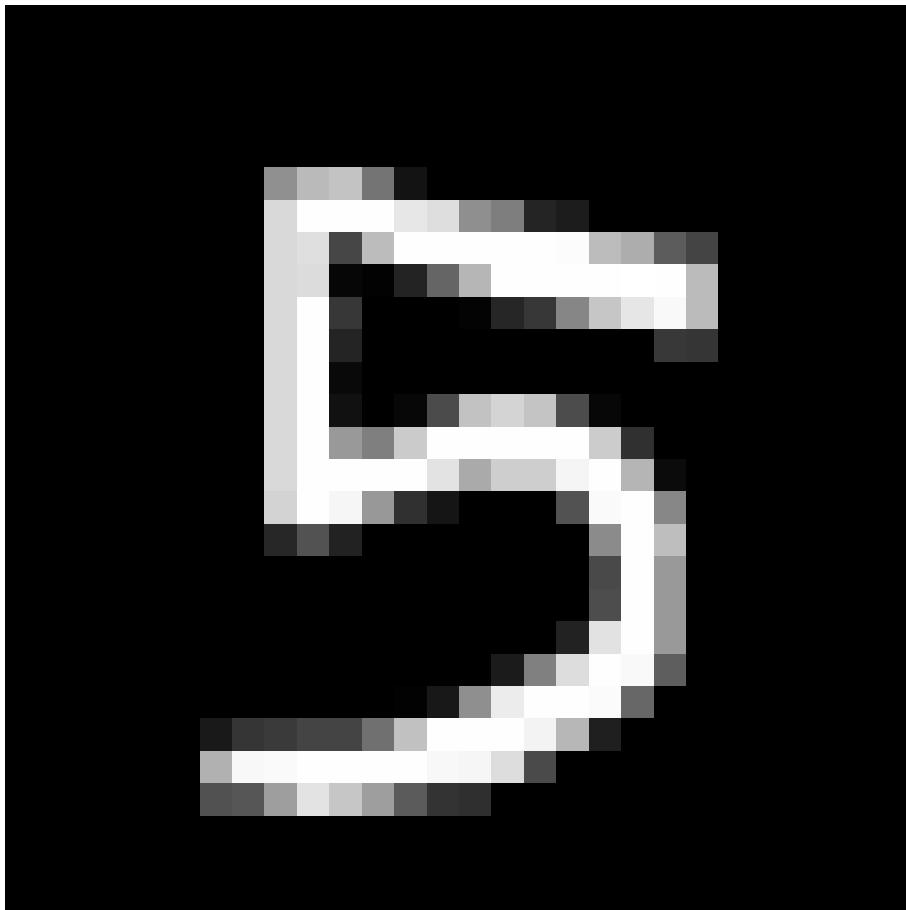
# Example: Digit recognition



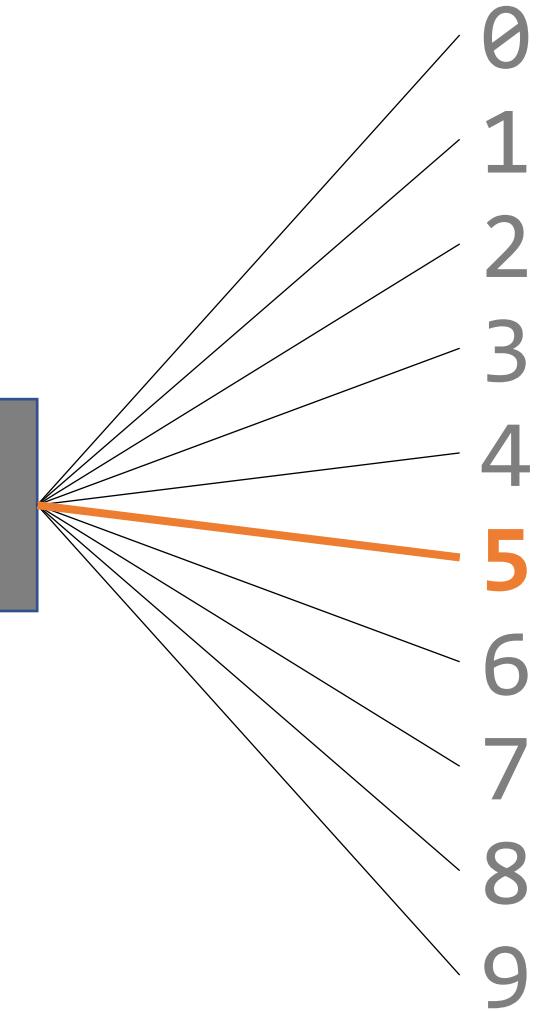
Digit  
Recognition  
Algorithm



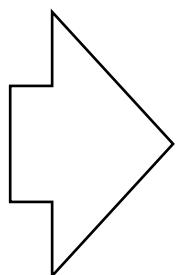
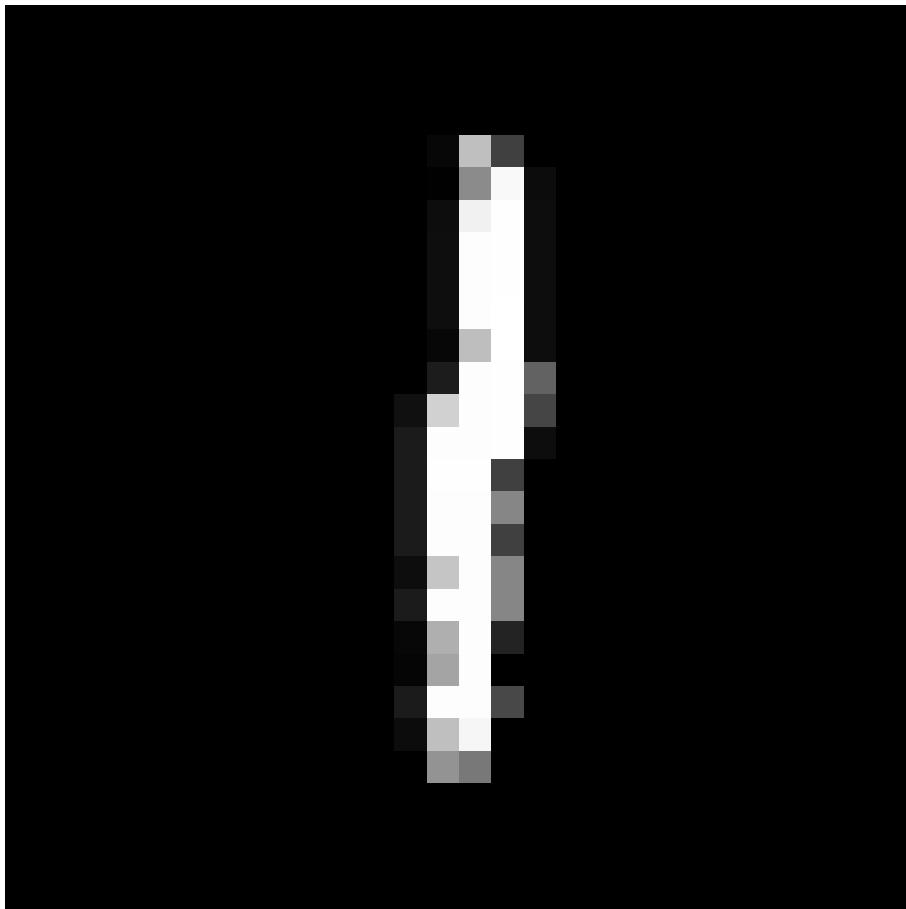
# Example: Digit recognition



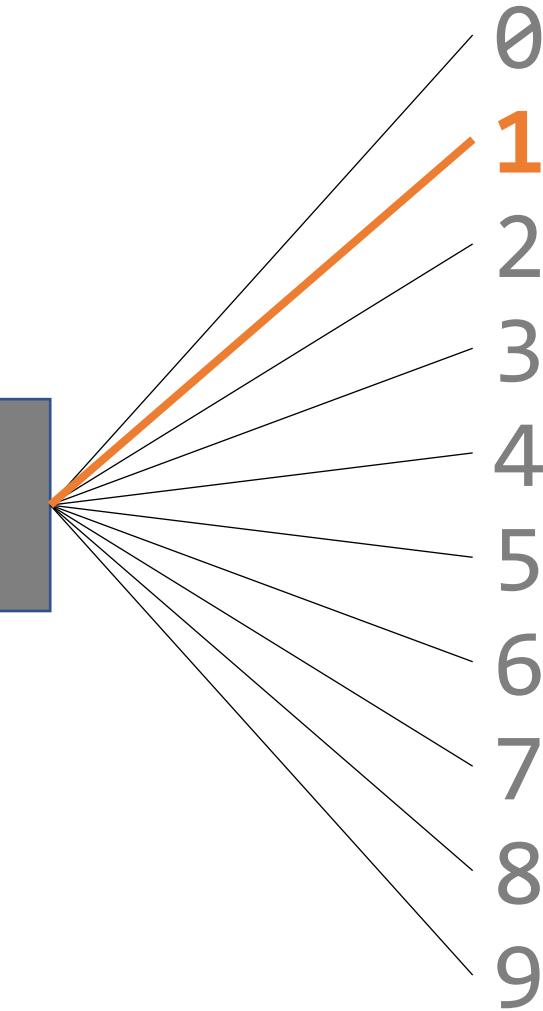
Digit  
Recognition  
Algorithm



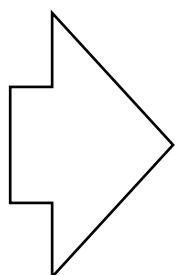
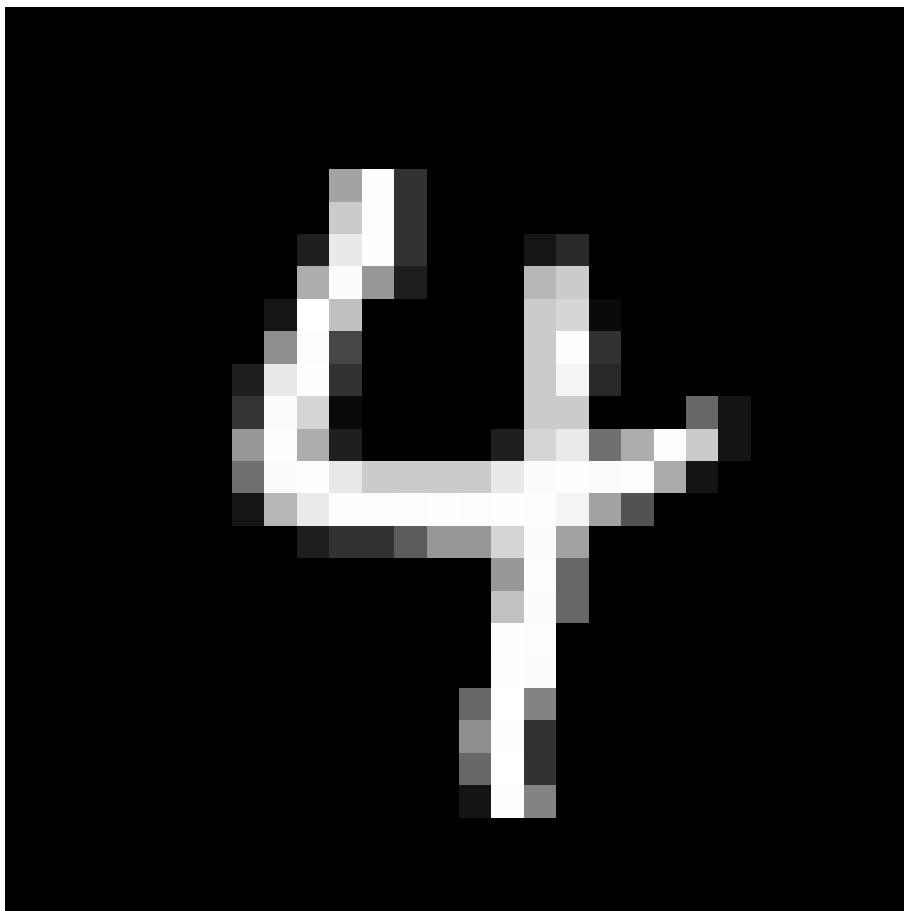
# Example: Digit recognition



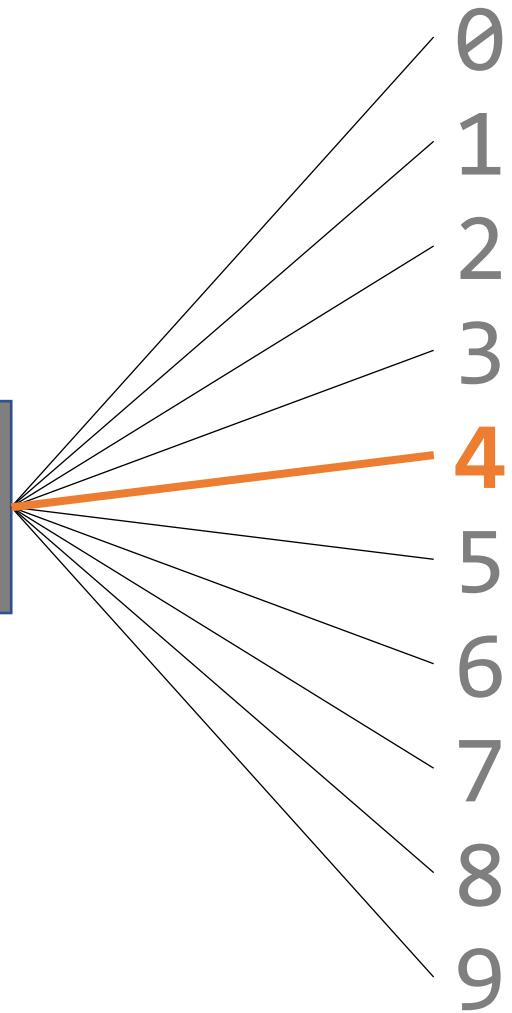
Digit  
Recognition  
Algorithm



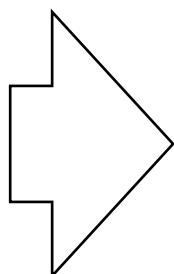
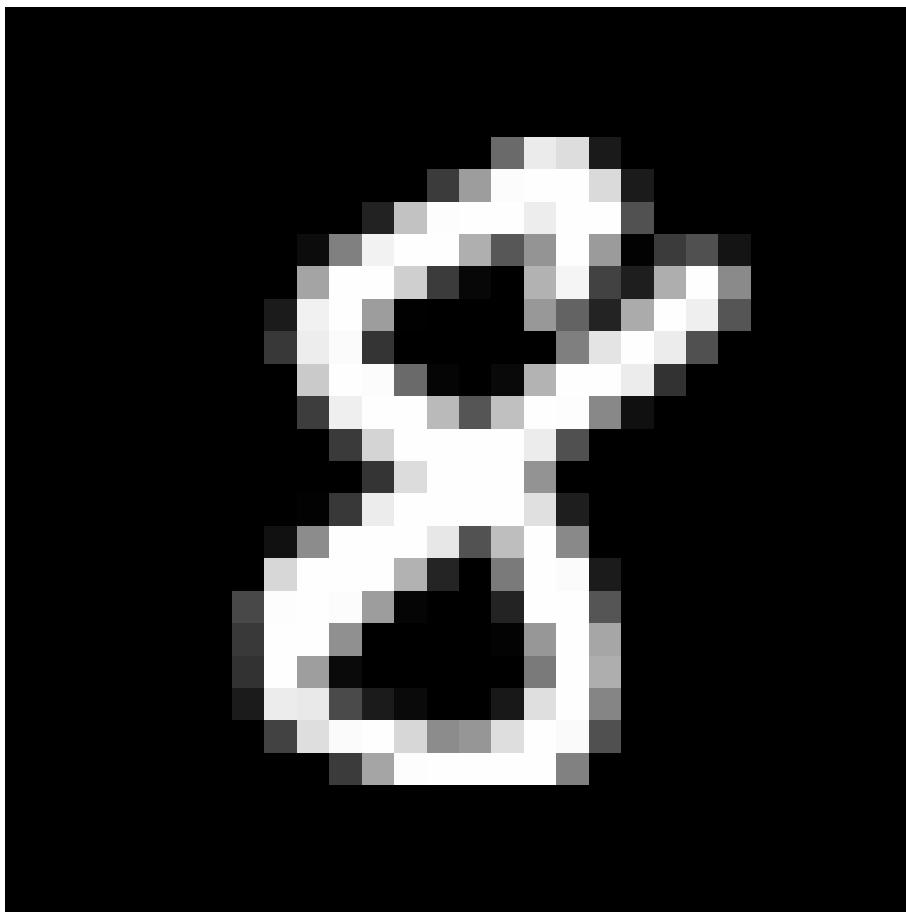
# Example: Digit recognition



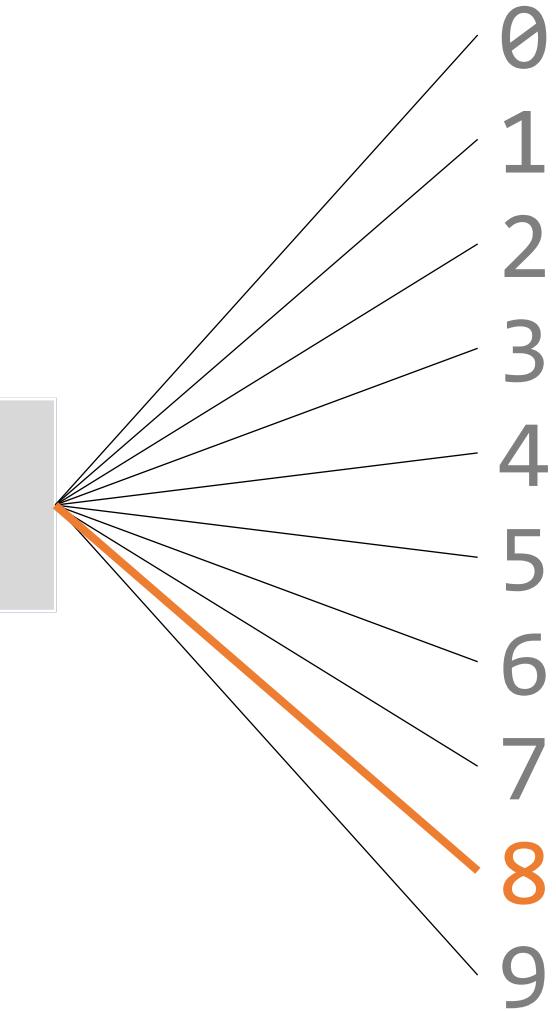
Digit  
Recognition  
Algorithm



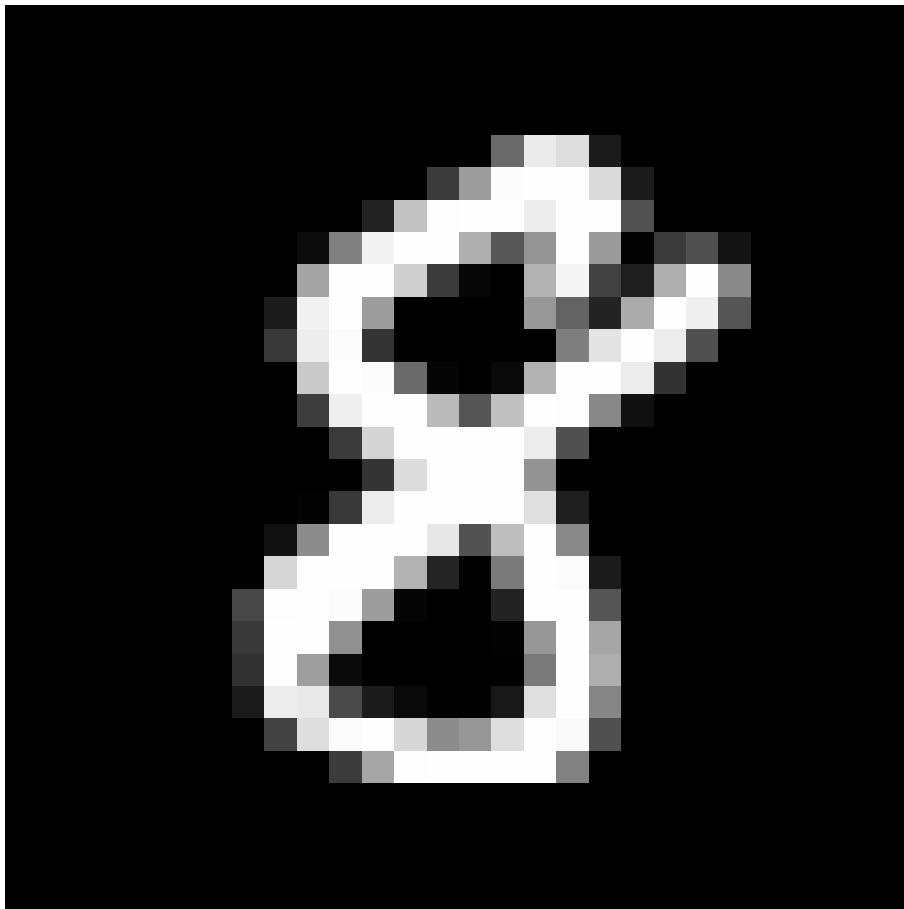
# Example: Digit recognition



Digit  
Recognition  
Algorithm

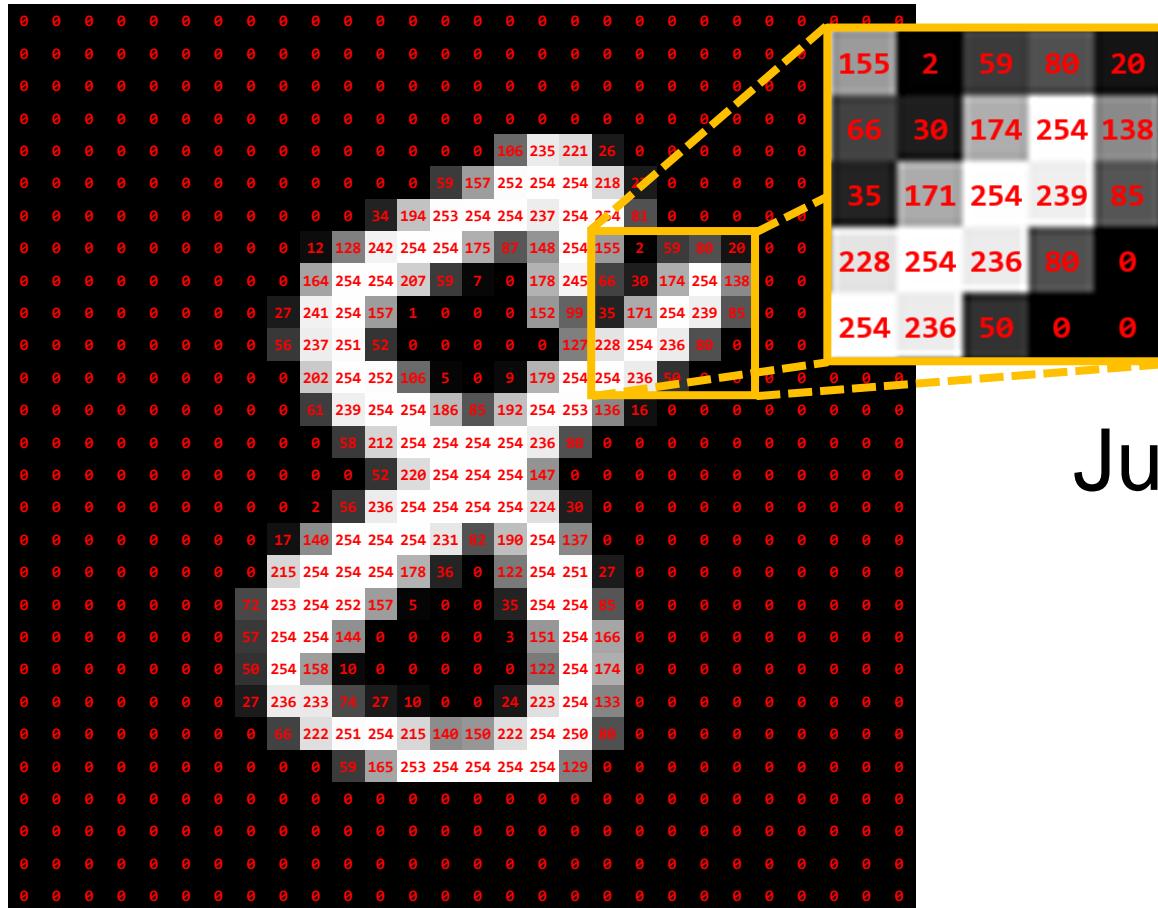


# Example: Digit recognition



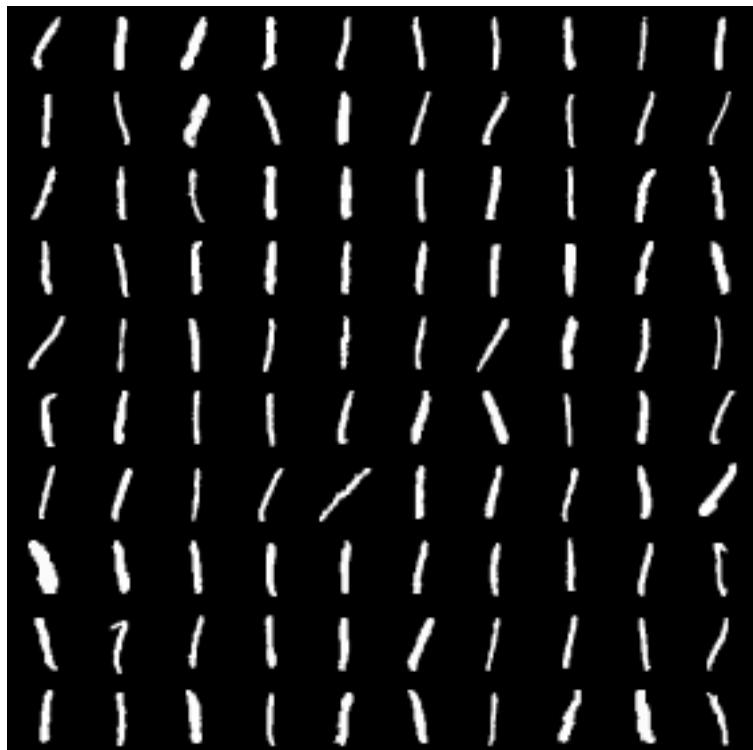
What is an image?

# Example: Digit recognition



Just an array of numbers

# Example: Digit recognition



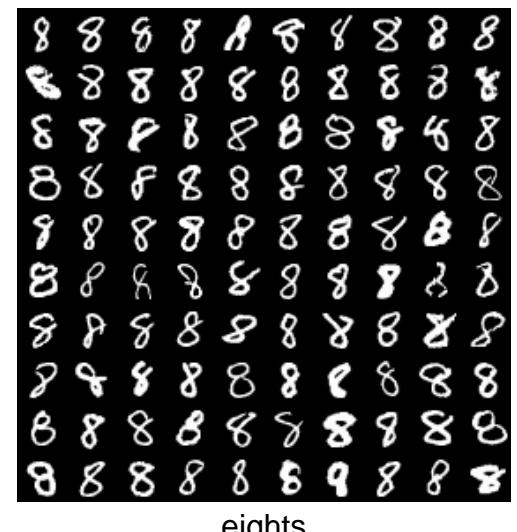
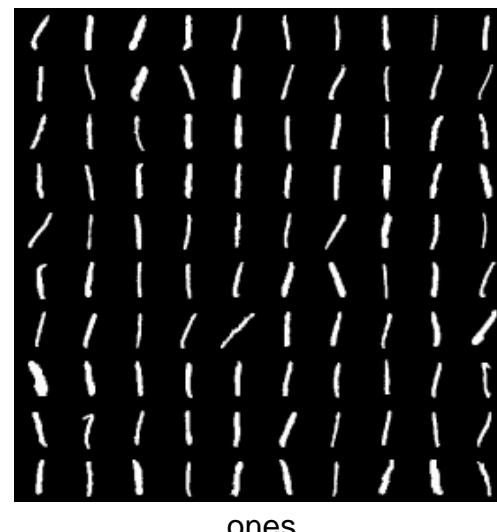
ones



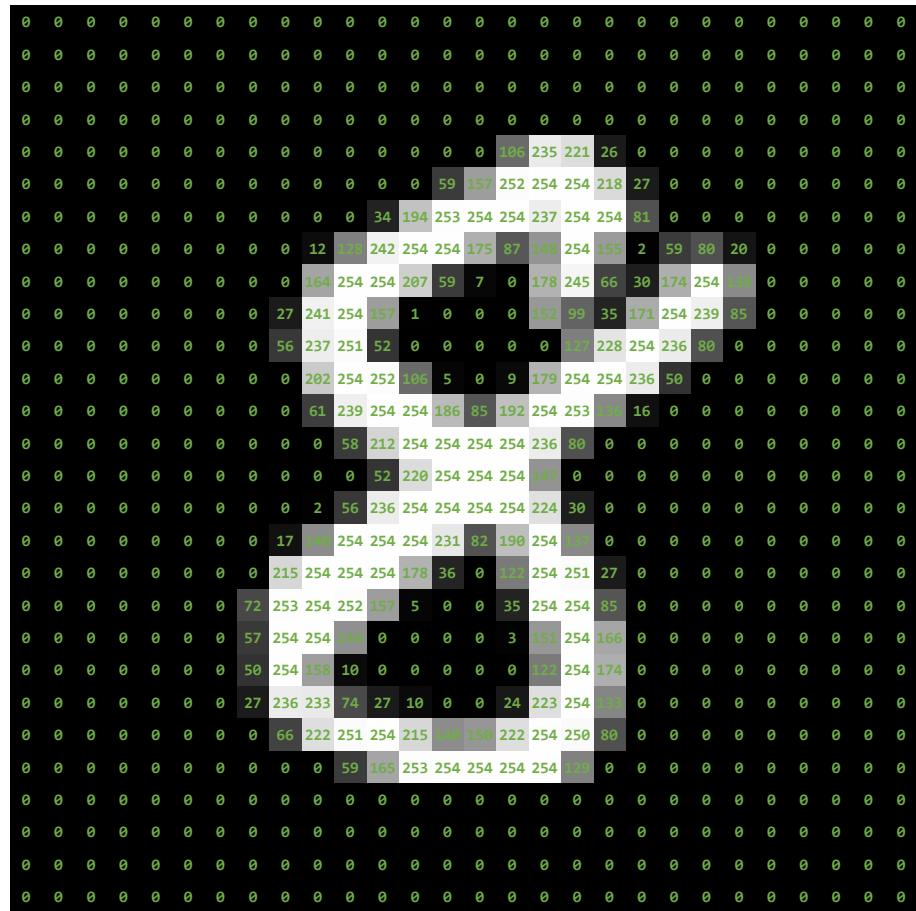
eights

# Example: Digit recognition

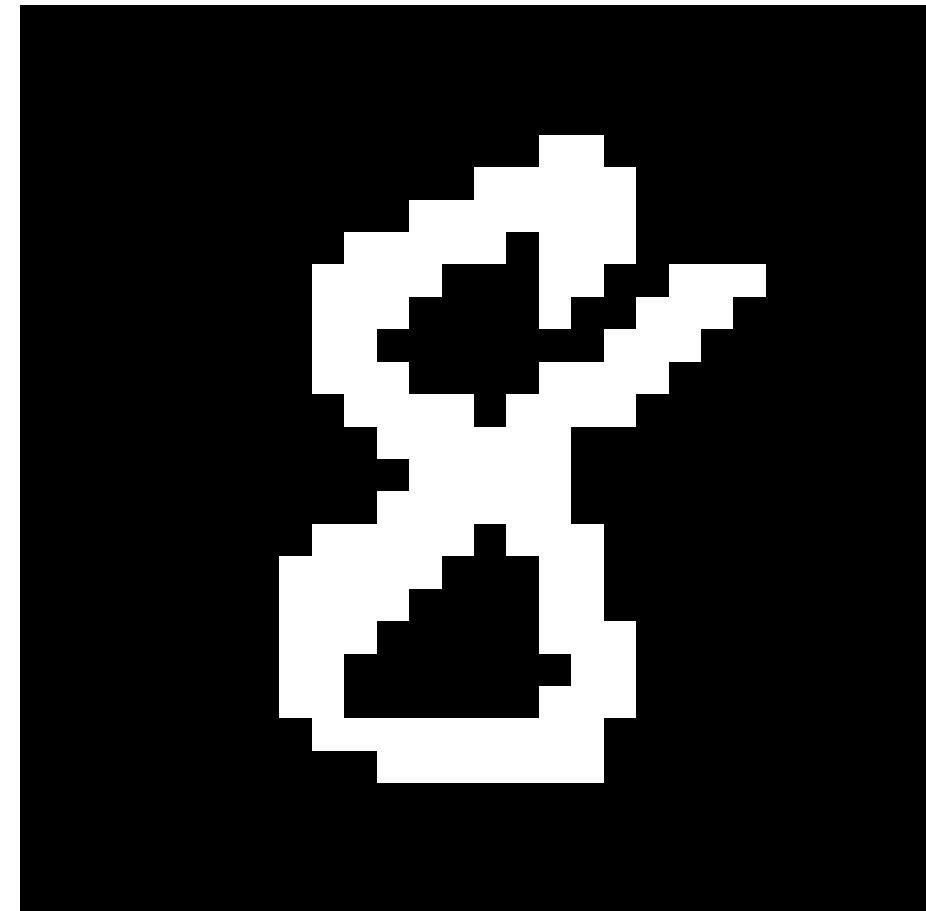
- Can we devise a simple algorithm to distinguish between digits one and eight?
- How about counting the number of “white” pixels?



# Example: Digit recognition

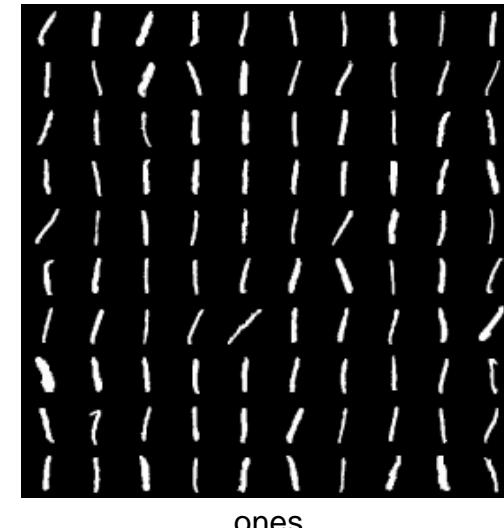
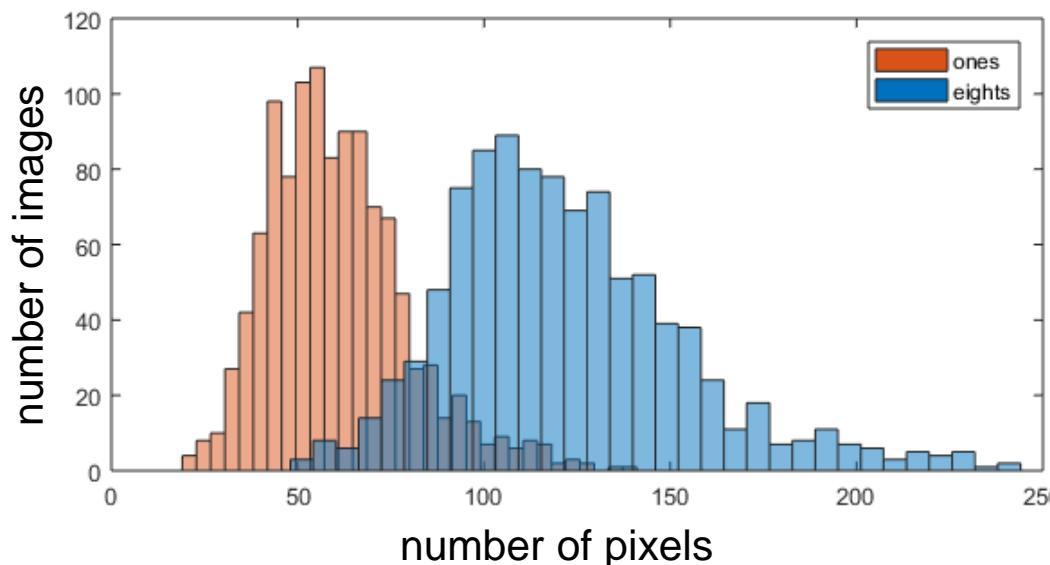


$> 127 =$

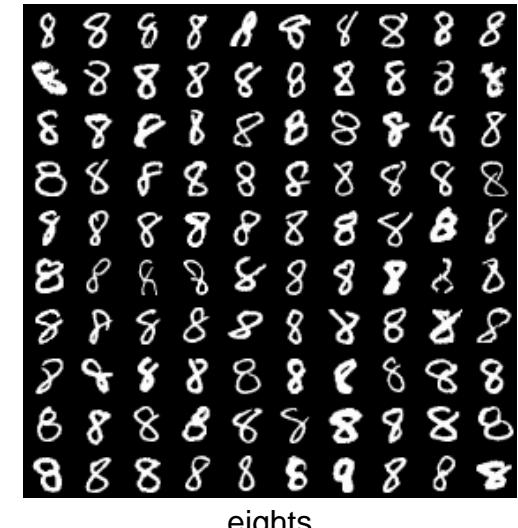


# Example: Digit recognition

- 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...



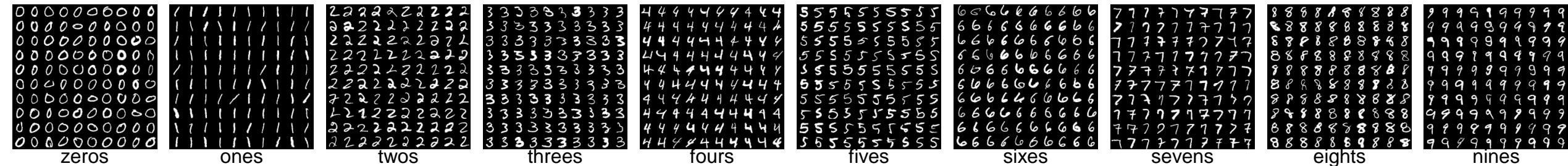
ones



eights

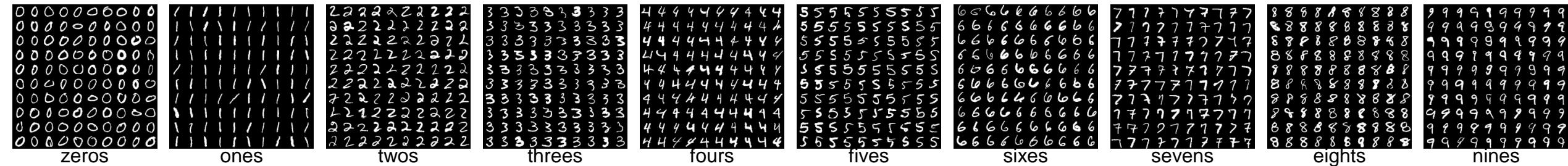
# Example: Digit recognition

- 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...



# 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!**



# Example: Digit recognition

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

## Training Data

000000000000	1111111111	222222222222	333333333333	444444444444	555555555555	666666666666	777777777777	888888888888	999999999999
000000000000	1111111111	222222222222	333333333333	444444444444	555555555555	666666666666	777777777777	888888888888	999999999999
000000000000	1111111111	222222222222	333333333333	444444444444	555555555555	666666666666	777777777777	888888888888	999999999999
000000000000	1111111111	222222222222	333333333333	444444444444	555555555555	666666666666	777777777777	888888888888	999999999999
000000000000	1111111111	222222222222	333333333333	444444444444	555555555555	666666666666	777777777777	888888888888	999999999999
000000000000	1111111111	222222222222	333333333333	444444444444	555555555555	666666666666	777777777777	888888888888	999999999999
000000000000	1111111111	222222222222	333333333333	444444444444	555555555555	666666666666	777777777777	888888888888	999999999999
000000000000	1111111111	222222222222	333333333333	444444444444	555555555555	666666666666	777777777777	888888888888	999999999999
000000000000	1111111111	222222222222	333333333333	444444444444	555555555555	666666666666	777777777777	888888888888	999999999999
000000000000	1111111111	222222222222	333333333333	444444444444	555555555555	666666666666	777777777777	888888888888	999999999999
zeros	ones	twos	threes	fours	fives	sixes	sevens	eights	nines

# Example: Digit recognition

- Linear model for digit recognition
- Feature vector  $\mathbf{x} = ?$
- How about raw pixel values?

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

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

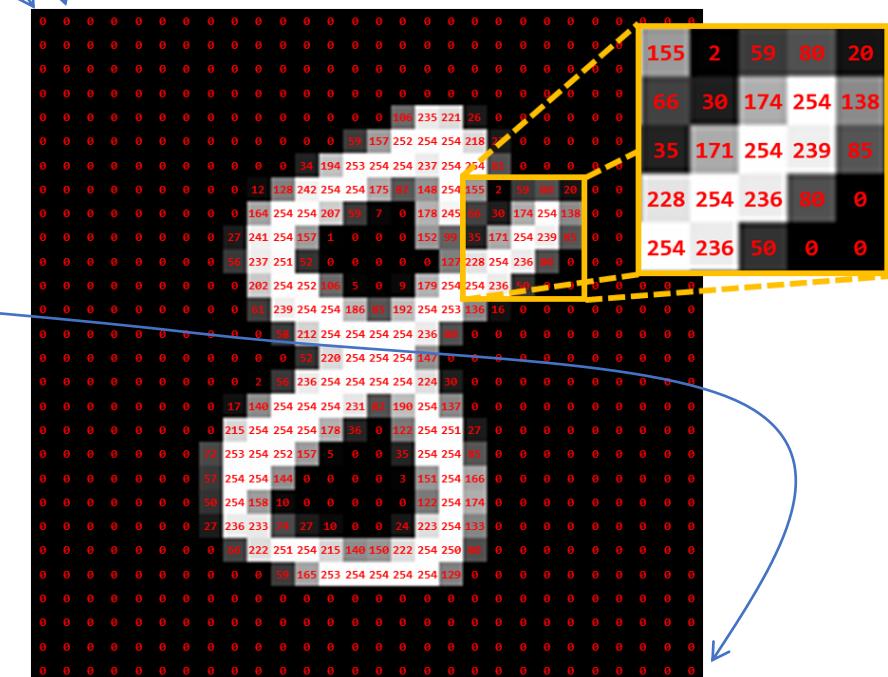
$$x_0 = 1$$

Many parameters!\*

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

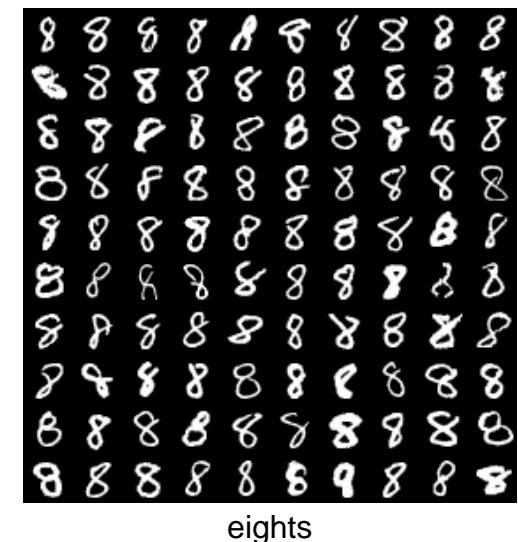
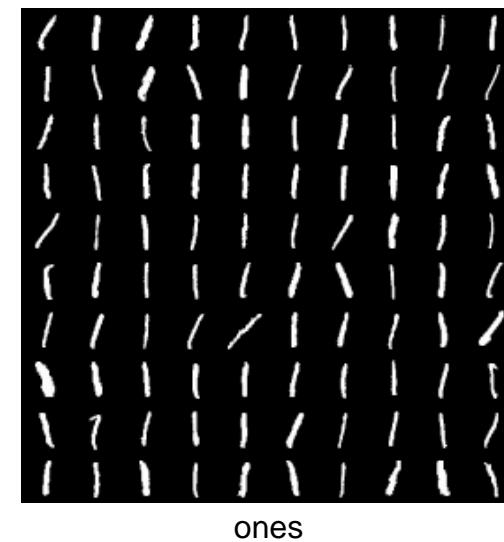
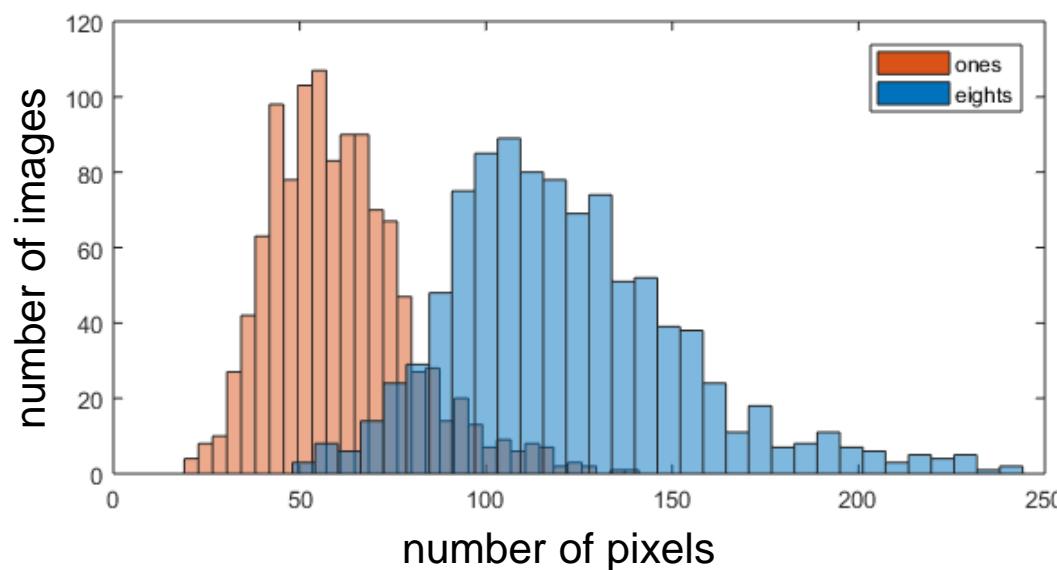
$$h_{\Theta}(\mathbf{x}) = \frac{1}{1 + e^{-\Theta^T \mathbf{x}}}$$

Logistic regression



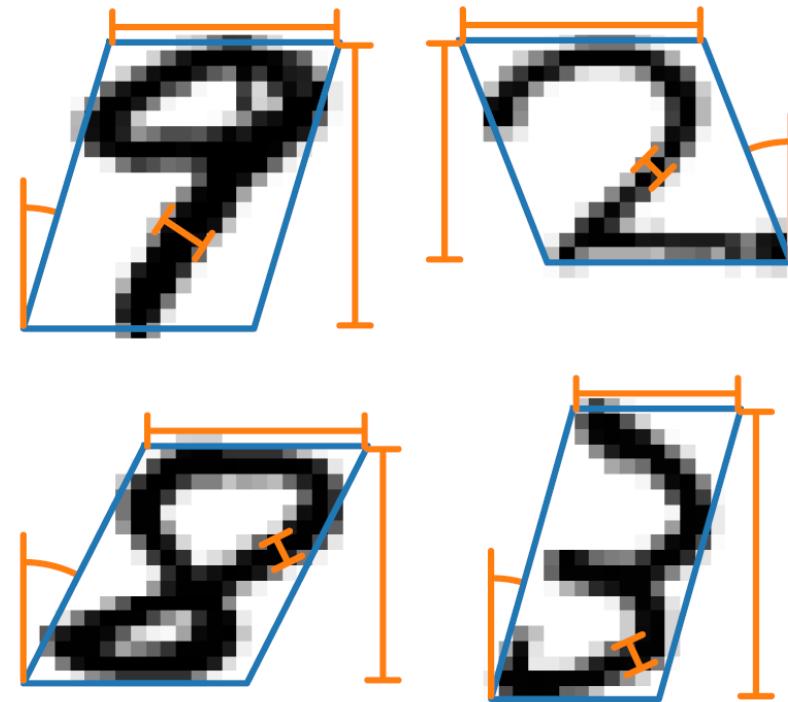
# Example: Digit recognition

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

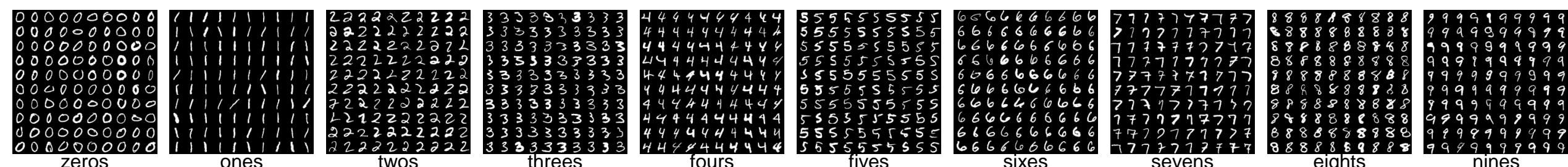
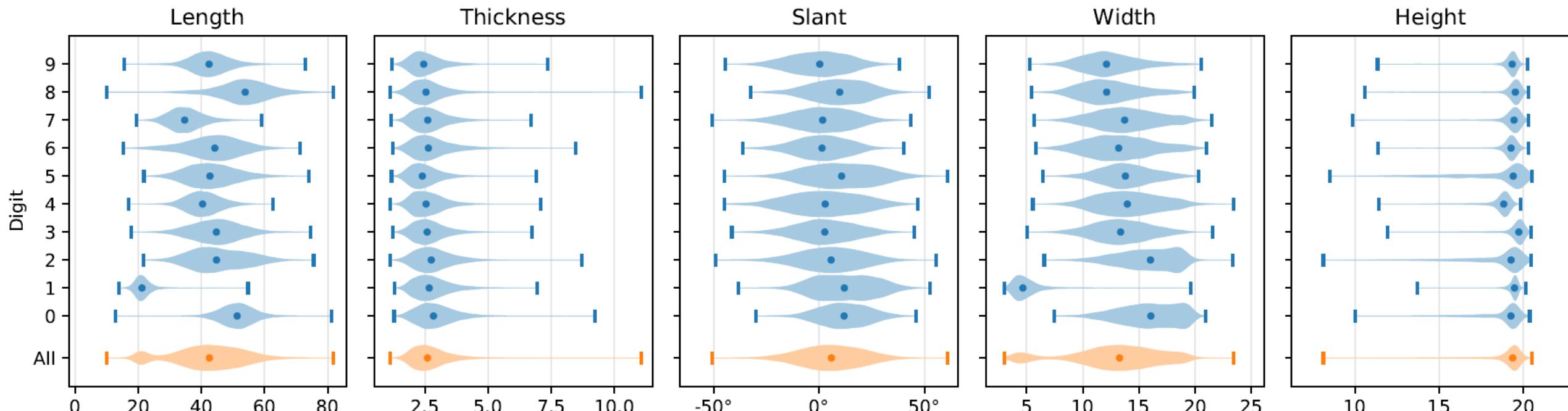


# Example: Digit recognition

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



# Example: Digit recognition



# Example: Digit recognition

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

Logistic regression

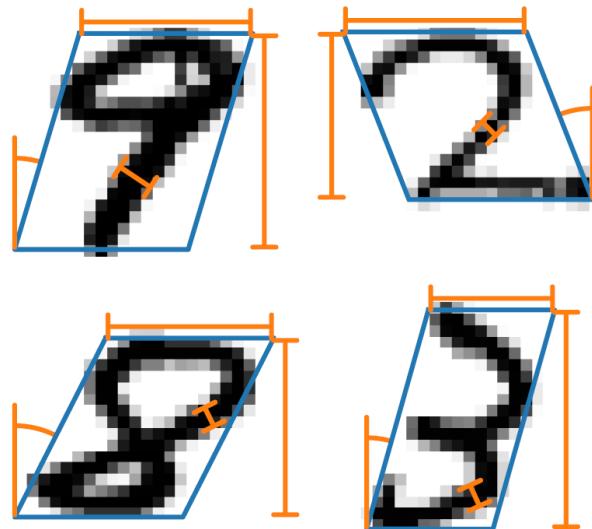
$$\mathbf{x} = [\text{length}, \text{thickness}, \text{slant}, \text{width}, \text{height}]^T$$

$$\begin{aligned}\Theta^T \mathbf{x} &= \Theta_0 + \Theta_1 \text{length} + \Theta_2 \text{thickness} \dots \\ &\dots + \Theta_3 \text{slant} + \Theta_4 \text{width} + \Theta_5 \text{height}\end{aligned}$$

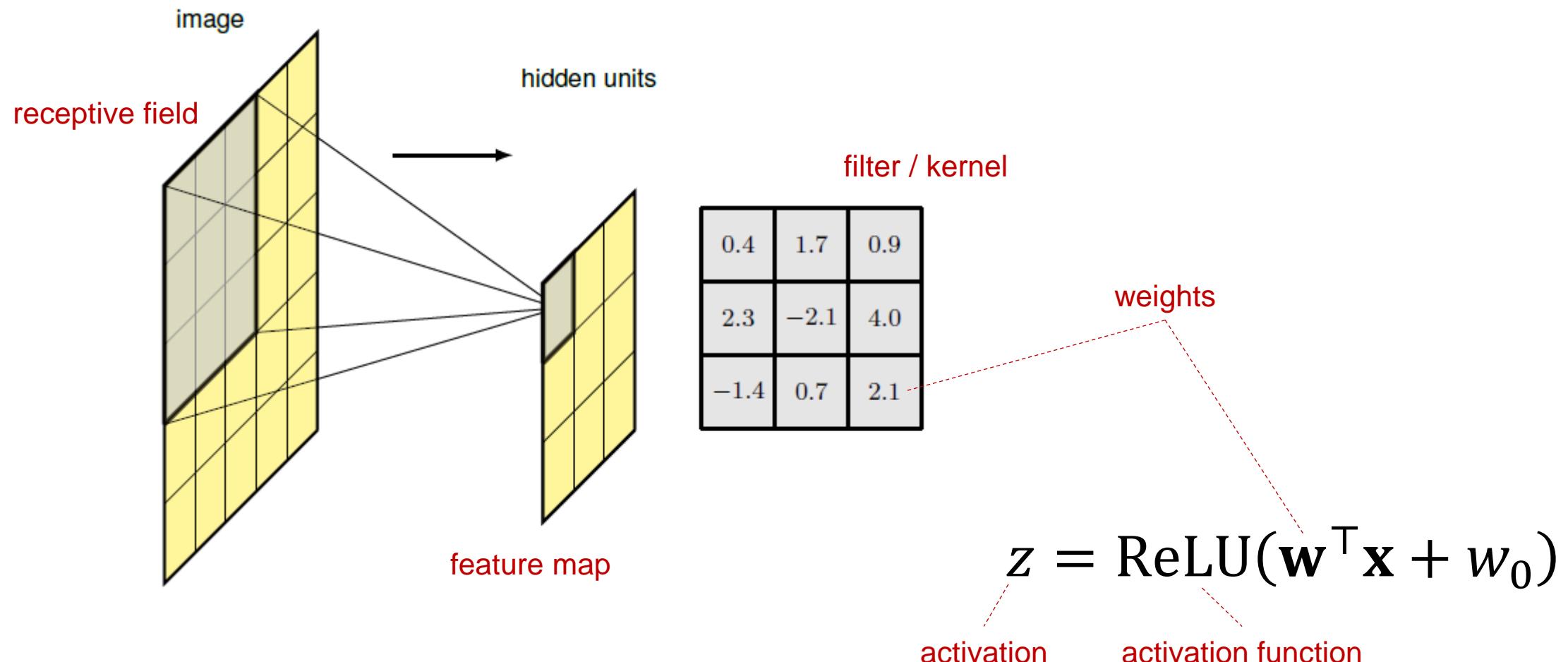
# Tutorial 0 - Introduction

# 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?



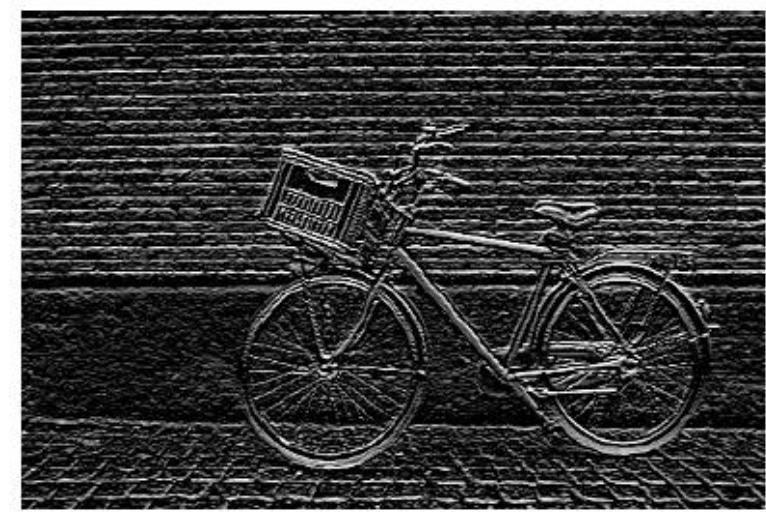
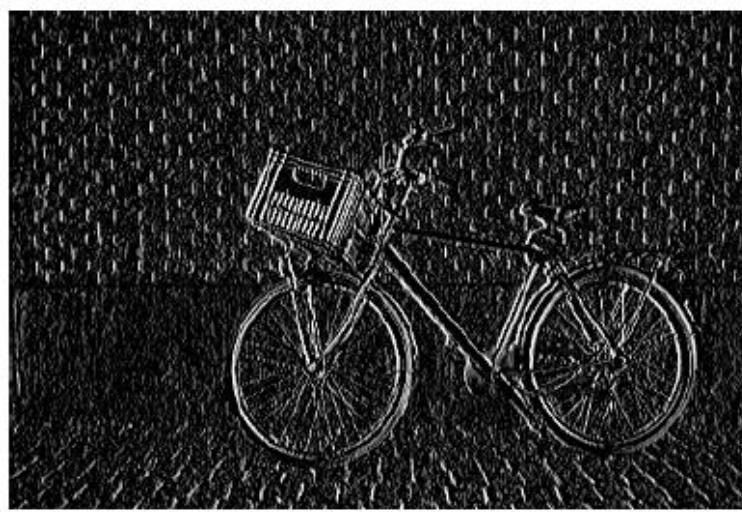
# Convolutional filters



# Feature detectors

-1	0	1
-1	0	1
-1	0	1

-1	-1	-1
0	0	0
1	1	1



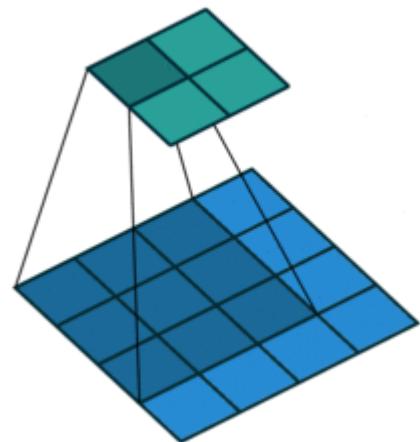
Convolutional filters are translation equivariant

# Padding & strides

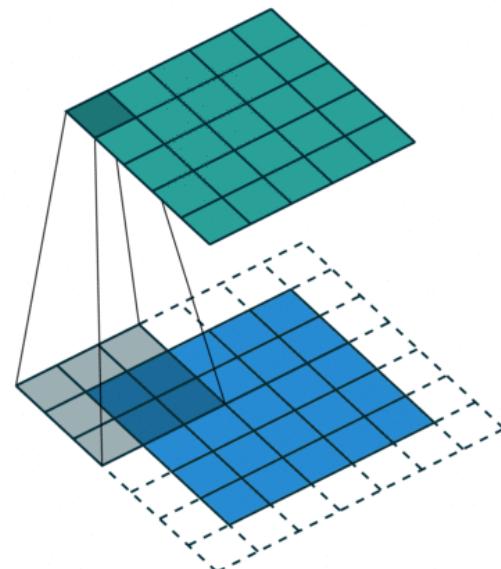
output size

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

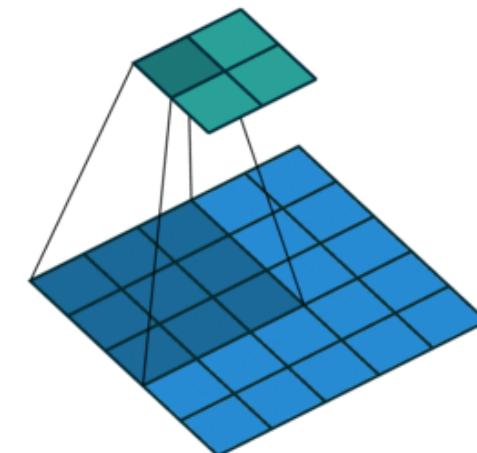
$M$ =input size;  $K$ =kernel size;  $P$ =padding;  $S$ =stride



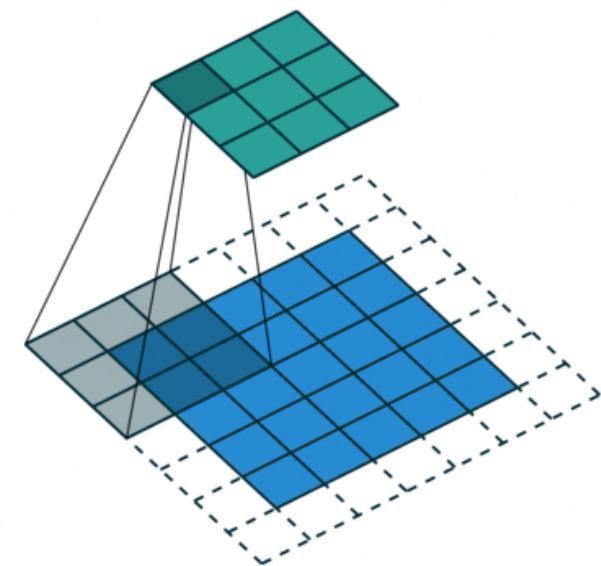
no padding, unit strides



padding

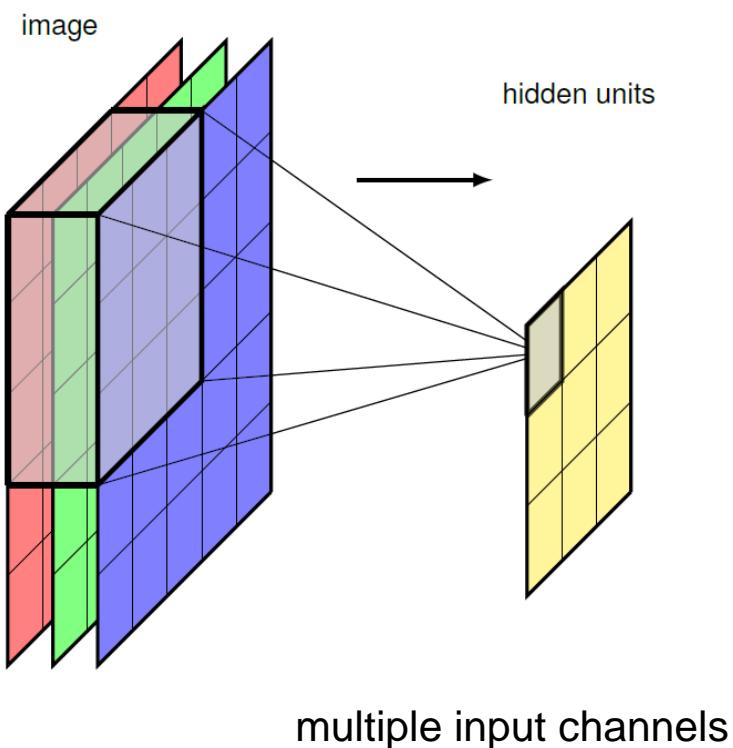


non-unit strides

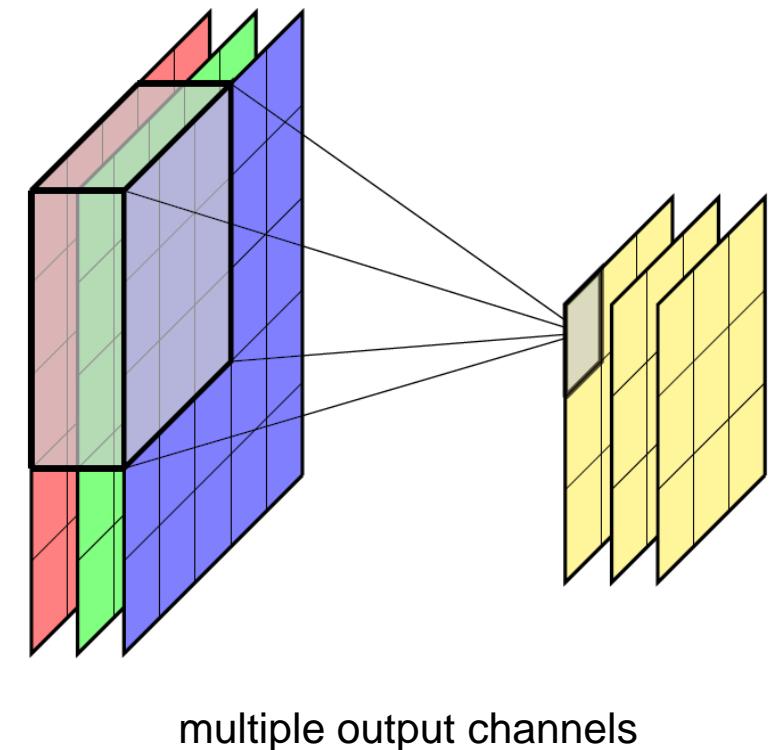


padding, non-unit strides

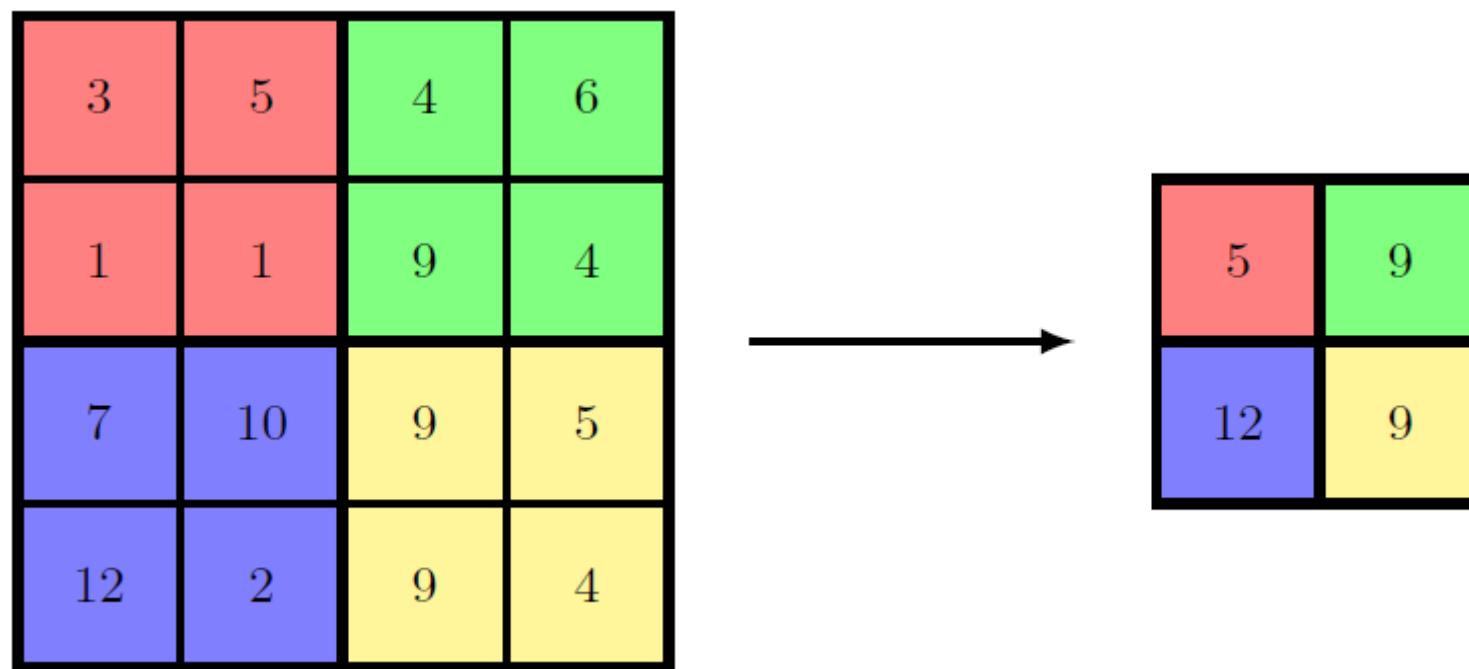
# Multi-dimensional convolutions



1.2	0.8	-3.7
-3.2	0.7	1.3
0.4	1.7	0.9
-1		
2.1	2.3	-2.1
4.	-1.0	4.0
	0.7	2.1

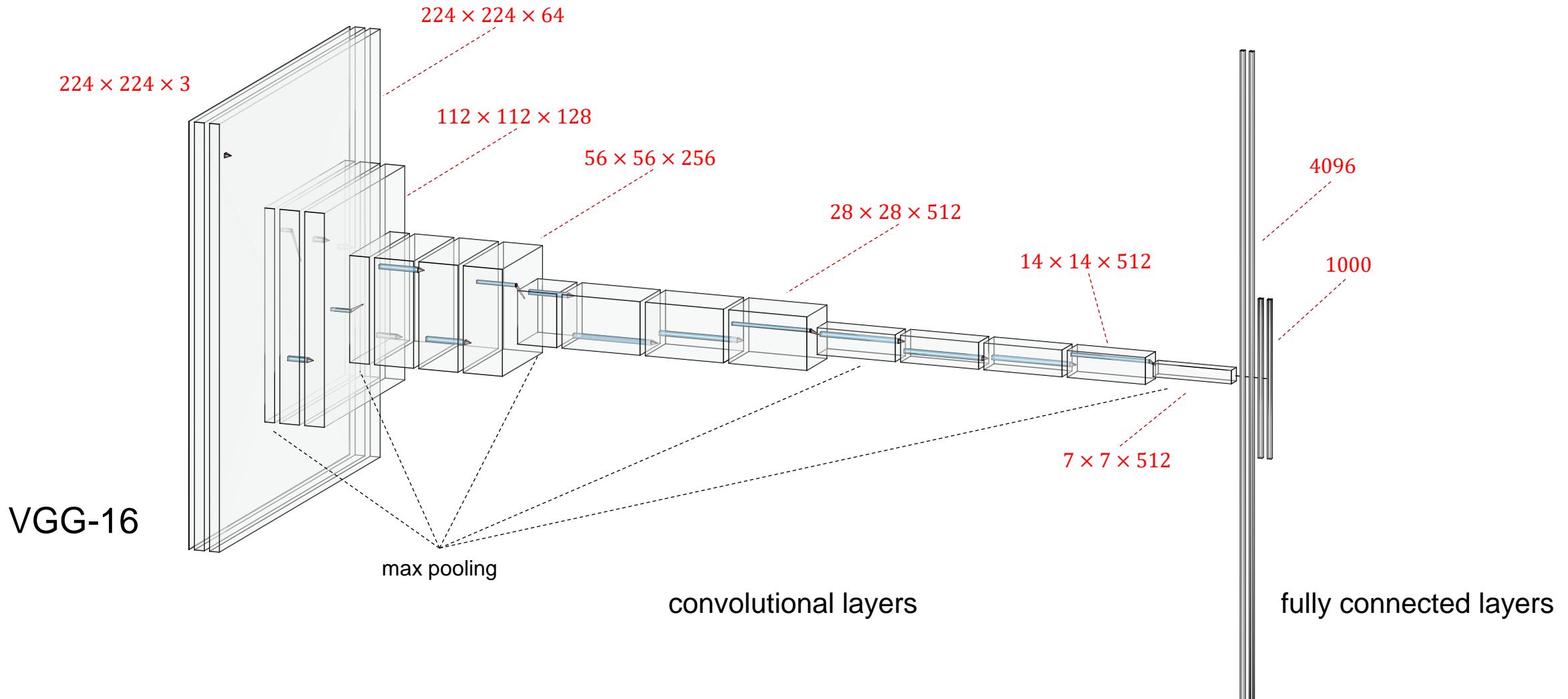


# Pooling



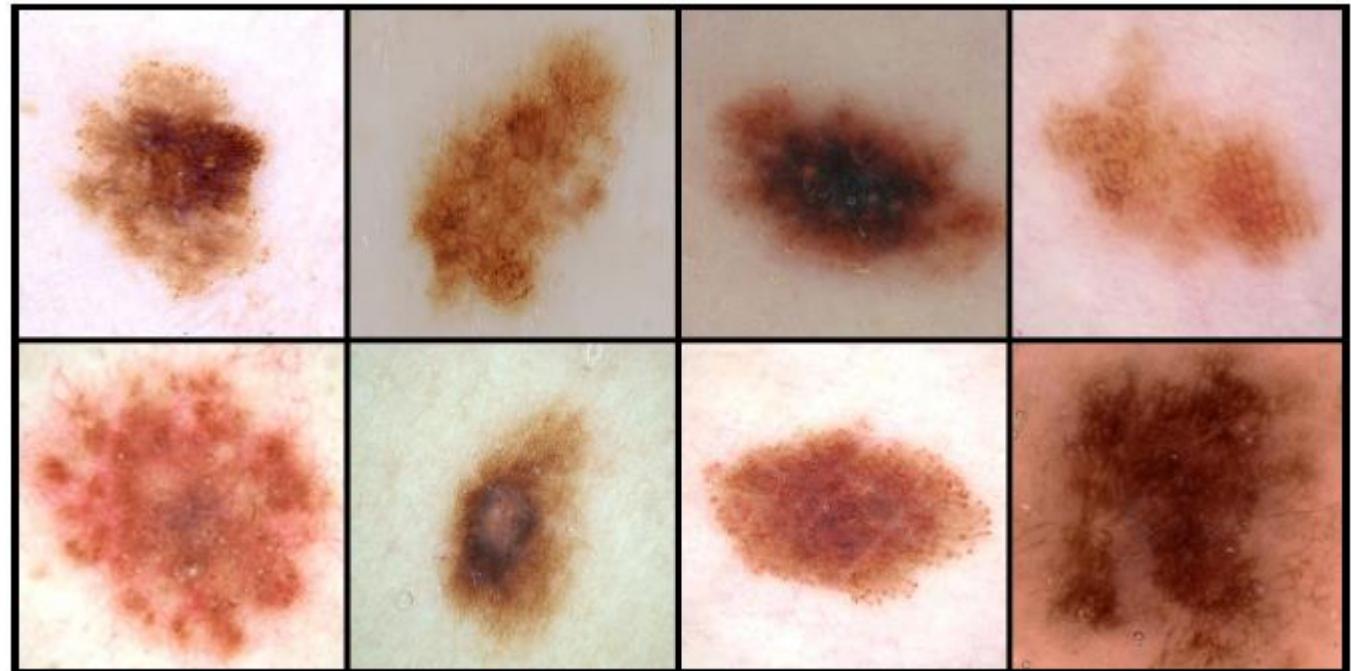
Pooling yields some local translation invariance

# Convolutional network architecture



# Example: Skin Cancer Diagnosis

- Image classification
- Supervised learning
- Transfer learning



# Tutorial 1 - Classification