

# Introduction

**A. Maier, V. Christlein, K. Breininger, Z. Yang, L. Rist, M. Nau, S. Jaganathan, C. Liu, N. Maul, L. Folle,  
K. Packhäuser, M. Zinnen**

Pattern Recognition Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg

April 15, 2024



## Who are we? - Lab Members



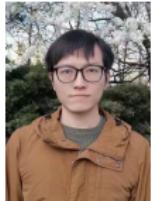
Andreas  
Maier



Zijin  
Yang



Alexander  
Barnhill



Chang  
Liu



Leonhard  
Rist



Merlin  
Nau



Noah  
Maul



Mathias  
Zinnen



Srikrishna  
Jaganathan



Kai  
Packhäuser

## Who are we? - Student Members



Lisa  
Schmidt



Majid  
Sharghi



Chengze  
Ye



Teena  
Tom Dieck



Leyi  
Tang



Supraja  
Ramesh



Karlo  
Gabriel  
Fonseca  
Yakovenko



Jingyi  
Yao



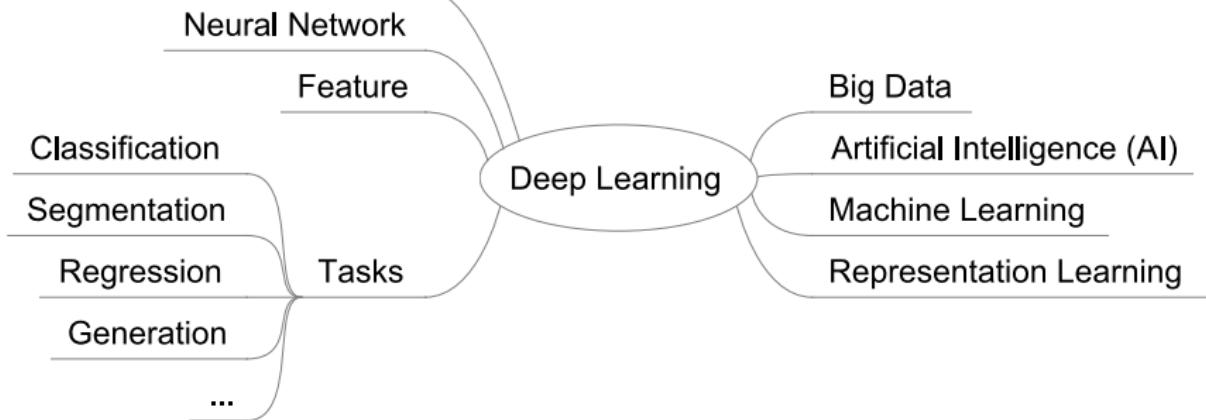
Anna-  
Sophie  
Stephan



Philip  
Wagner

# Deep Learning – Buzzwords

Supervised vs. unsupervised



## Outline

### Motivation

### Machine Learning and Pattern Recognition

### Perceptron

### Organizational Matters



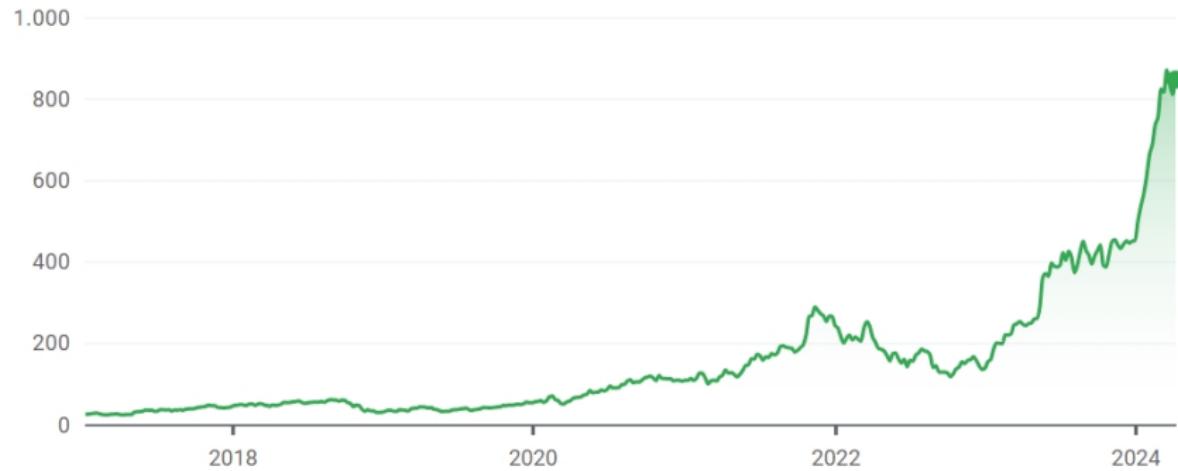
**FAU**

FRIEDRICH-ALEXANDER-  
UNIVERSITÄT  
ERLANGEN-NÜRNBERG  
SCHOOL OF ENGINEERING

# Motivation



## NVIDIA Stock Market



Source: <https://www.google.com/finance/quote/>

# The Big Bang of Deep Learning

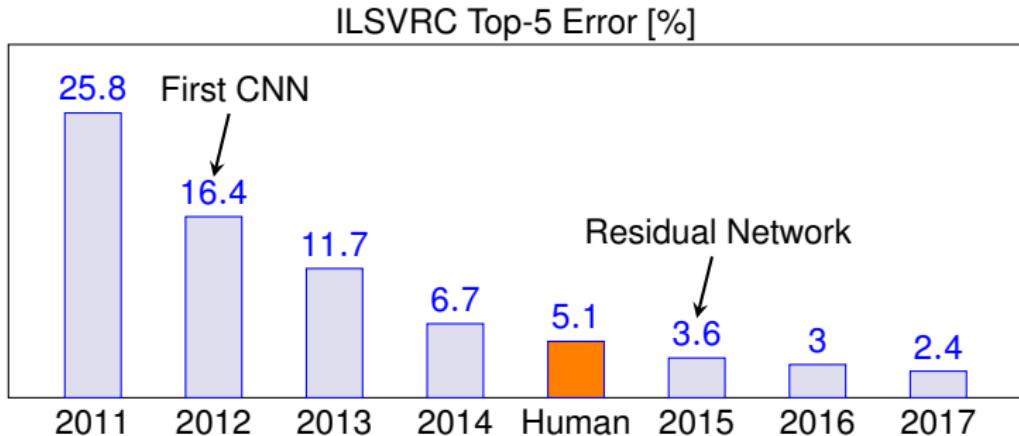


## ImageNet [8] Dataset

- $\approx$  14 mio. images, labeled into  $\approx$  20.000 **synonym sets**
- ImageNet Large Scale Visual Recognition Challenge using  $\approx$  1000 classes
- Images downloaded from the Internet, **single** label per image
- **2012: Breakthrough** by Krizhevsky et al. [10]

Source: Krizhevsky et al. 2012

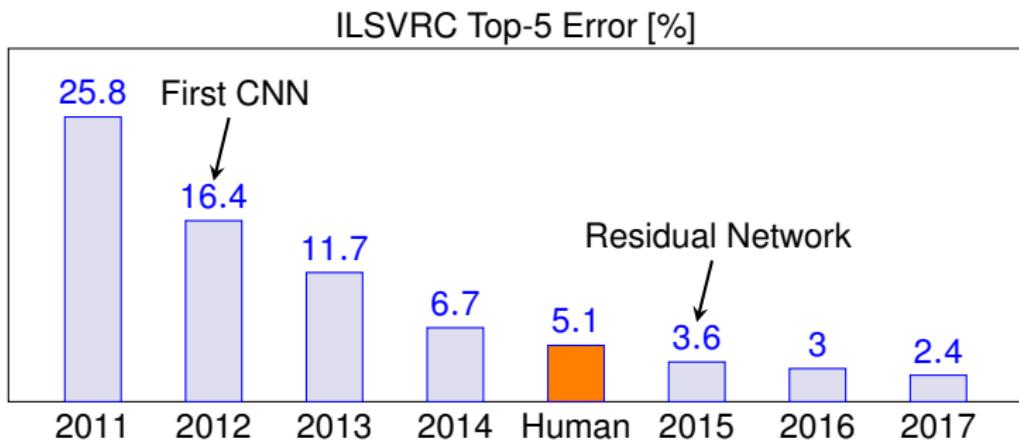
# ImageNet Large Scale Visual Recognition Challenge



- First CNN approach now famous as **AlexNet** [10]

Source: image-net.org, Russakovsky et al. 2015

# ImageNet Large Scale Visual Recognition Challenge

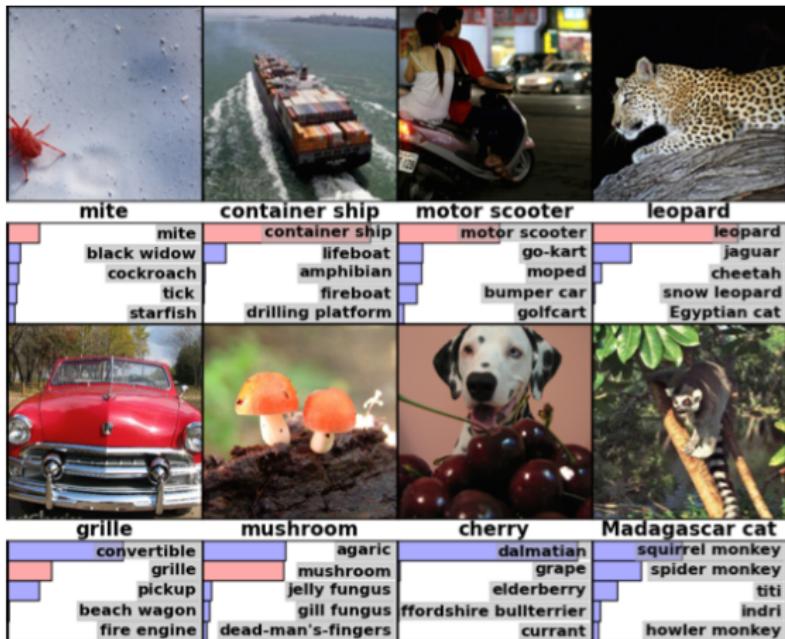


- First CNN approach now famous as **AlexNet** [10]
- “Superhuman” should be Super-Karpathy-an performance



Source: image-net.org, Russakovsky et al. 2015

# ImageNet Large Scale Visual Recognition Challenge



Source: Krizhevsky et al. 2012

## Deep Learning Users

**NETFLIX**

**DAIMLER**

**IBM**

**xerox**



**Microsoft**



 **Lunit**



**SIEMENS**

**Google**

 **DeepMind**



**SAMSUNG**

## Playing Go

- 1997: Deep Blue beats Garry Kasparov
- Go as a next challenge
- Large branching factor



Source: <https://commons.wikimedia.org/wiki/File:FloorGoban.jpg>

## Playing Go

- 1997: Deep Blue beats Garry Kasparov
- Go as a next challenge
- Large branching factor
- 2016: AlphaGo [16] beats a professional



Source: <https://commons.wikimedia.org/wiki/File:FloorGoban.jpg>

## Playing Go

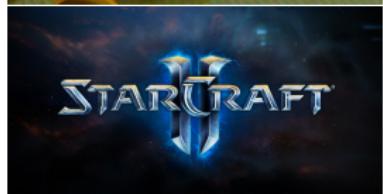
- **1997:** Deep Blue beats Garry Kasparov
- **Go** as a next challenge
- Large branching factor
- **2016:** AlphaGo [16] beats a professional
- **2017:** AlphaGoZero [1] surpasses every human in Go by self-play
- **2017:** AlphaZero [2] generalizes to a number of other board games



Source: <https://commons.wikimedia.org/wiki/File:FloorGoban.jpg>

## Playing Go

- **1997:** Deep Blue beats Garry Kasparov
- **Go** as a next challenge
- Large branching factor
- **2016:** AlphaGo [16] beats a professional
- **2017:** AlphaGoZero [1] surpasses every human in Go by self-play
- **2017:** AlphaZero [2] generalizes to a number of other board games
- **2019:** AlphaStar beats professional StarCraft players



Source: <https://commons.wikimedia.org/wiki/File:FloorGoban.jpg>

# Google DeepDream

Attempt to understand the inner workings of the network: What it "dreams" about when presented with images

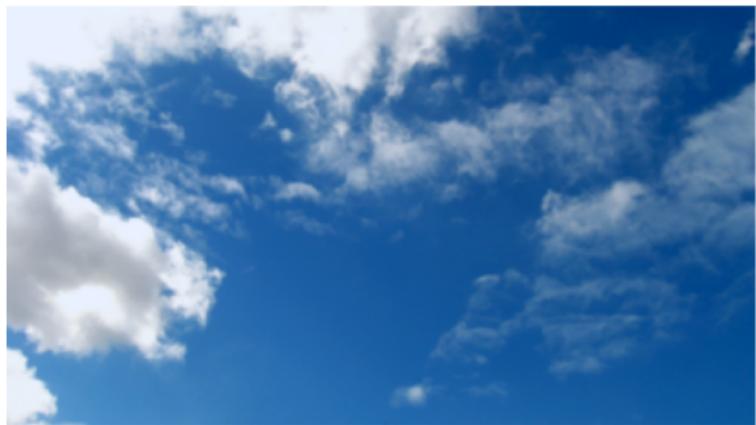
## Idea:

- Arbitrary image or noise as input
- Instead of adjusting network parameters, tweak image towards high activations
- Different layers enhance different features (low or high level)



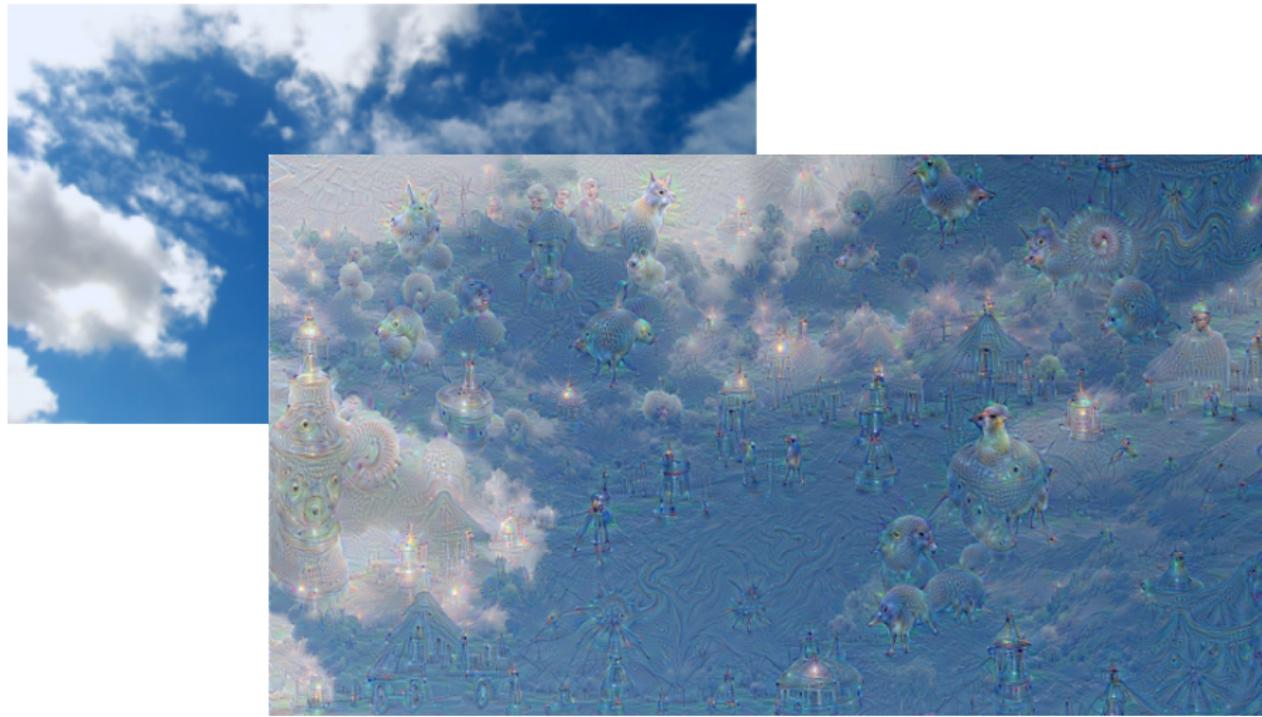
Source: <https://research.googleblog.com>

## Google DeepDream



Source: <https://research.googleblog.com>

# Google DeepDream



Source: <https://research.googleblog.com>

# Google DeepDream

Looking for new animals in the clouds



"Admiral Dog!"



"The Pig-Snail"



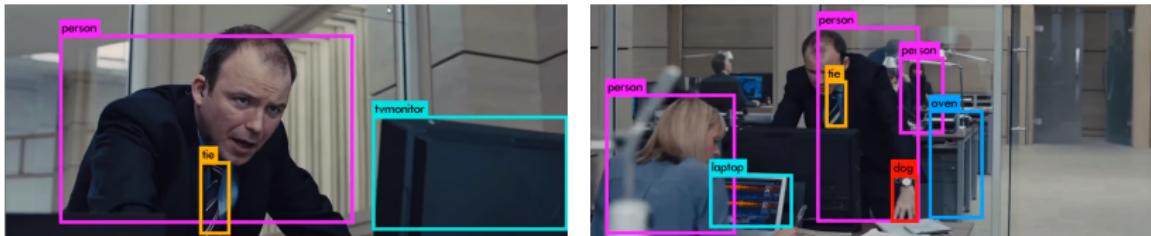
"The Camel-Bird"



"The Dog-Fish"

Source: <https://research.googleblog.com>

## Real-Time Object Detection: YOLO, YOLO9000, YOLOv3 [11]–[13]



Click for video

- YOLO: You only look once
- Prior systems → Use classifiers at multiple locations and scales
- YOLO → Simultaneous regression of bounding box and label
- FAST: 40-90 frames/second on a NVIDIA Titan X

Source: [www.youtube.com, Redmon and Farhadi 2016](https://www.youtube.com/watch?v=9JzXWVgkQHc)

# Every Day Use



# Siri

## Siri: Speech Interpretation and Recognition Interface



"Hey Siri, call Mom"

You can activate Siri and make your request all at once  
— without using the Home button.\*

Source: [www.apple.com/ios/siri/](http://www.apple.com/ios/siri/)

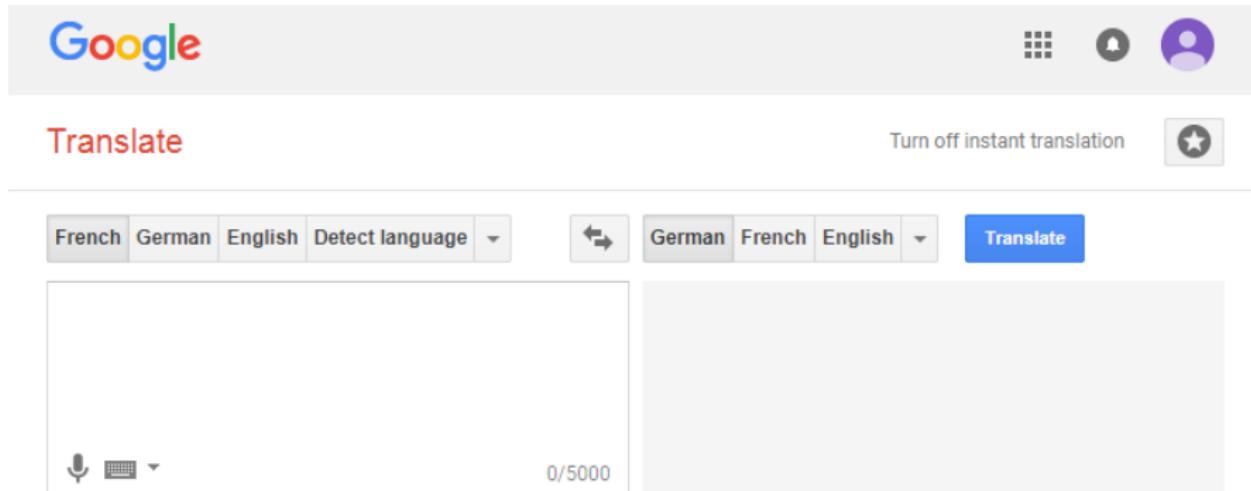
# Google Echo & Amazon Alexa Voice Service

W H A T   I S  
ECHO DOT?



Source: [www.amazon.com](http://www.amazon.com)

# Google Translate



The screenshot shows the Google Translate homepage. At the top left is the Google logo. To its right are three icons: a grid, a bell, and a user profile. Below the logo, the word "Translate" is written in red. To the right of "Translate" are two buttons: "Turn off instant translation" and a star icon. Below this is a horizontal bar with language selection boxes. The first box contains "French", "German", "English", and "Detect language". The second box contains "German", "French", "English", and a dropdown arrow. To the right of these boxes is a blue "Translate" button. Below the language boxes are two large input fields. The left field has a microphone and keyboard icon with a dropdown arrow, and the text "0/5000". The right field is currently empty. At the bottom of the page, there is a text input field with placeholder text "Type text or a website address or translate a document." and a "Source: translate.google.de" link.

**NEXT TIME  
ON DEEP LEARNING**



**FAU**

FRIEDRICH-ALEXANDER-  
UNIVERSITÄT  
ERLANGEN-NÜRNBERG  
SCHOOL OF ENGINEERING

# Introduction - Part 2

**A. Maier, V. Christlein, K. Breininger, Z. Yang, L. Rist, M. Nau, S. Jaganathan, C. Liu, N. Maul, L. Folle,  
K. Packhäuser, M. Zinnen**

Pattern Recognition Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg

April 15, 2024



# Research at the Pattern Recognition Lab

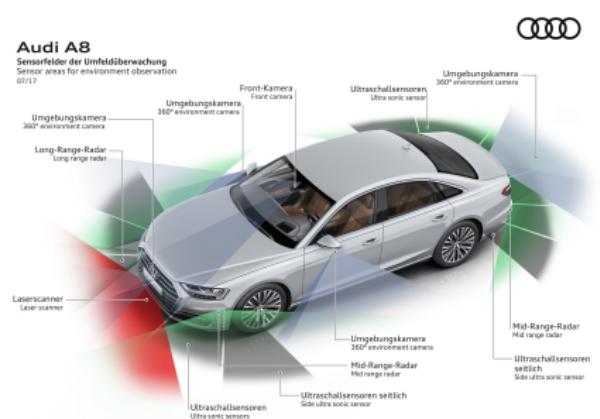


# Assisted and Automated Driving

## Goal

Find new ways to train and update deep learning mechanisms in environments with high safety requirements

- Assisted and automatic driving relies on sensor data
- Cameras to detect dynamic objects, driving lanes and free space
- Detection and segmentation tasks → deep learning



Source: Audi AG

# Assisted and Automated Driving

- Currently: neural networks trained and thoroughly tested before deployment
  - Requires huge amounts of manually labeled data
- Regular test drives cannot verify system reliability in all traffic scenarios



Click for video

## Assisted and Automated Driving

- Currently: neural networks trained and thoroughly tested before deployment
- Requires huge amounts of manually labeled data
- Regular test drives cannot verify system reliability in all traffic scenarios
- **Challenge:** New ways to test algorithms in simulated environments and utilize data collected in production cars equipped with appropriate hardware



Click for video

# Smart Devices

## Problem statement

Renewable energy power  $\neq$  energy demand

- Underproduction → backup power plants
- Overproduction → energy lost
- Real-Time-Pricing to match energy demand and supply
- Needs *smart devices* to shift workload automatically



# Smart Devices

## Goal

Establish energy equilibrium by predicting energy consumption

- Example: Interrupt fridge cooling cycle when price is high, start washing machine when price is low
- Dependencies between tasks, user information and action necessary (e.g., washer/dryer)
- Task: Identify time-shiftable loads and assess appropriate time frame
- Approach: Train **recurrent neural networks** to identify usage patterns and dependencies between devices

# Cloud Detection for Power Forecast [4]

## Goal

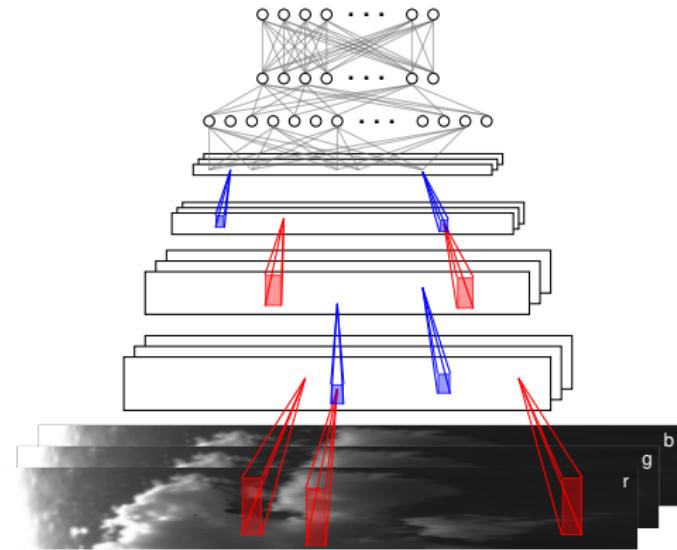
Power forecast for solar power plants with a high temporal and spatial resolution

## Approach

1. Monitor the sky
2. Detect clouds
3. Estimate the cloud motion
4. Establish power forecasts



## Cloud Detection for Power Forecast [4]



Input: Sky moving towards the sun

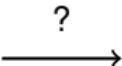
Output: Clear Sky Index = values betw. 0 (overcast sky) to 1 (clear sky)

# Writer Recognition

## Goal

Writer identification with limited training data (few pages per writer)

If we desire to  
desire to secure  
rising prosperity  
for war.



Also The idea  
and Europe but  
from Asia country



Необходимо для  
один параллельный  
и одна из них  
вопросов для уча-  
стников соревнований.



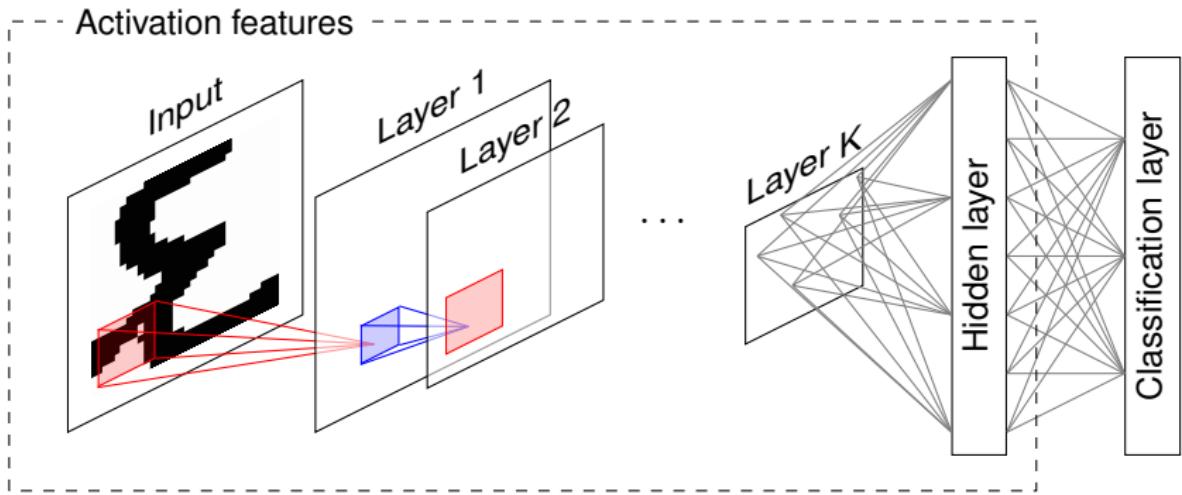
يجب أن يكون انتظار العان  
النتائج أو ظهور النتائج  
يمكن المفهوم .



Source: ICDAR'13 dataset, QUWI'15 dataset, freepik.com

## Writer Recognition using CNN Activation Features [6]

Use Neuronal Network for feature extraction



# Medical Applications



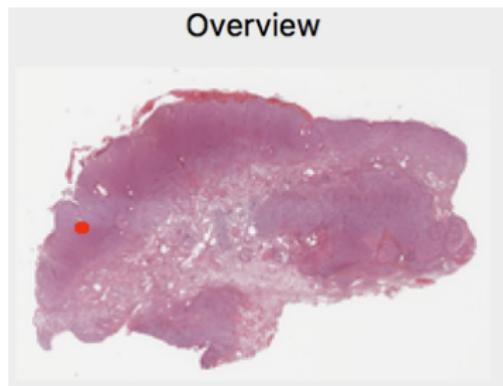
# Cell Classification for Tumor Diagnostics [3]

## Goal

Identify cells undergoing mitosis to assess tumor proliferation and aggressiveness in histological images

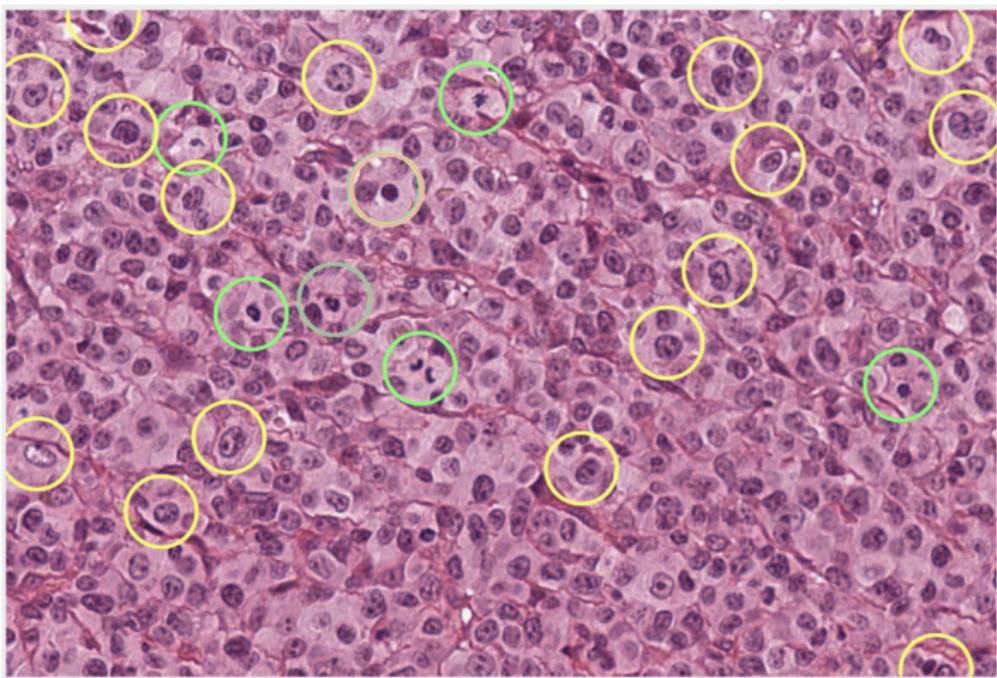
## Challenge

- Histological images: large number of cells
- Full annotations not feasible
- Sparse annotations
- Cells vary significantly in size/shape/etc



Source: Aubreville et al. 2017

## Cell Classification for Tumor Diagnostics [3]

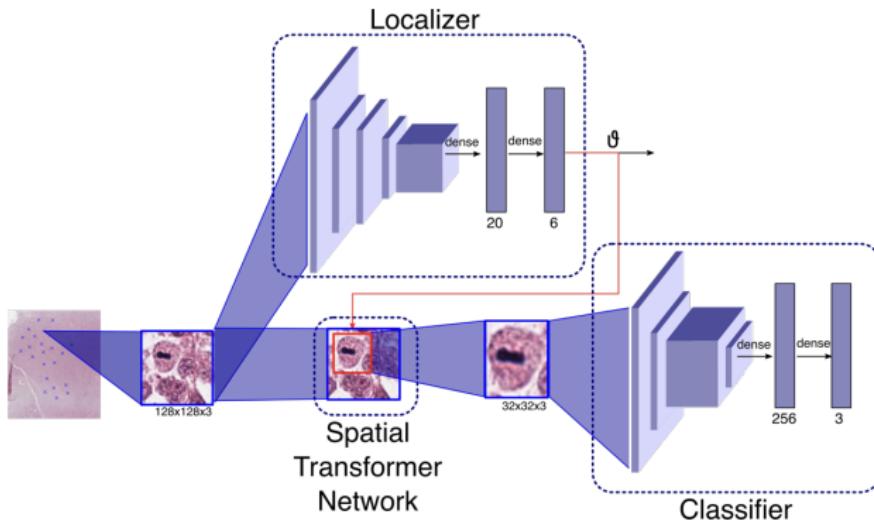


Source: Aubreville et al. 2017

# Cell Classification for Tumor Diagnostics [3]

## Approach

Use *spatial transformer networks* (STNs) to learn affine transformation **and** classification



Source: Aubreville et al. 2017

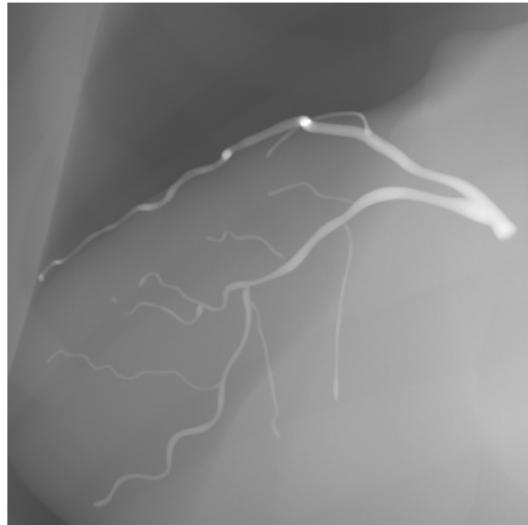
# Defect Pixel Interpolation

## Goal

- Reconstruction of coronaries based on truncated X-ray images
- Create “virtual” digital subtraction angiography

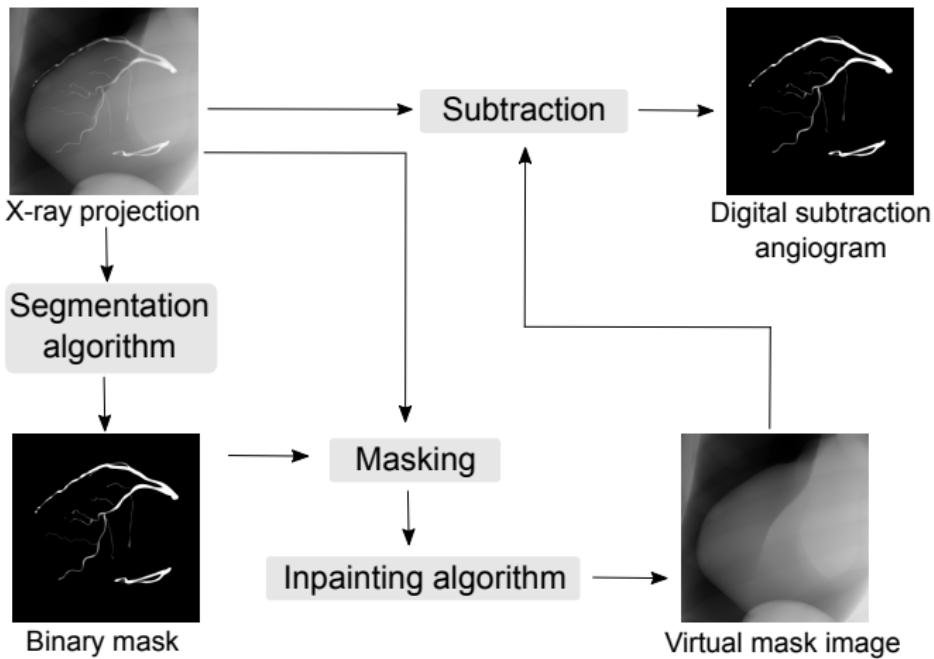
## Approach

1. Segment coronary vessels
2. Mask fluoroscopic image
3. Inpaint using U-net
4. Subtract inpainted image to get untruncated data



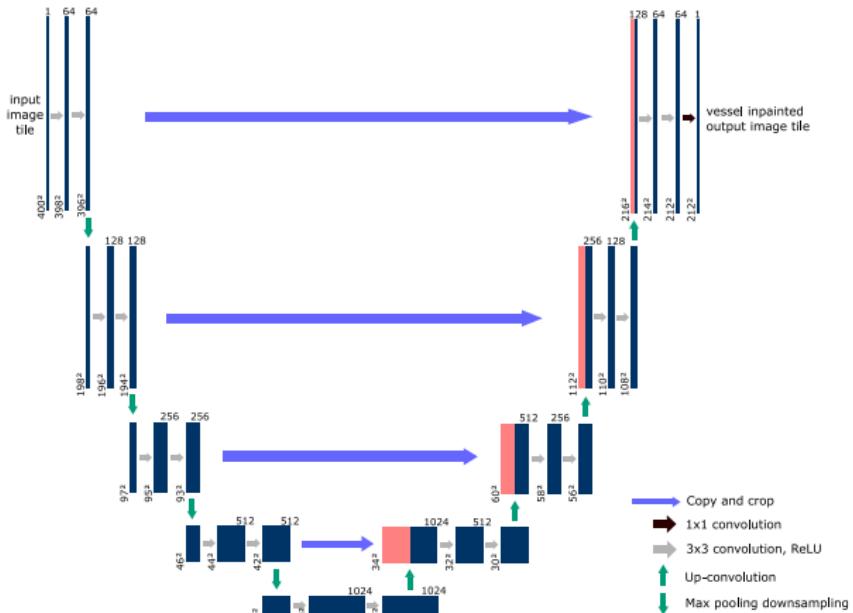
# Defect Pixel Interpolation

## Processing pipeline



## Defect Pixel Interpolation

## Deep learning for inpainting



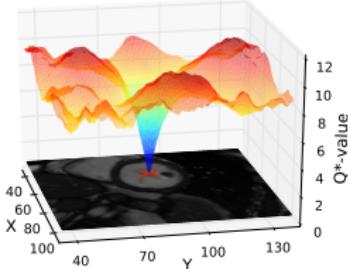
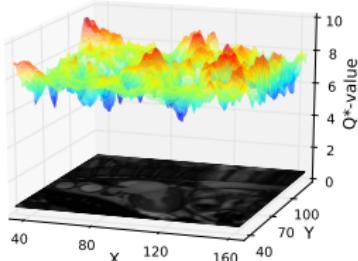
# Organ Search [7]

## Goal

Locate anatomic structures automatically

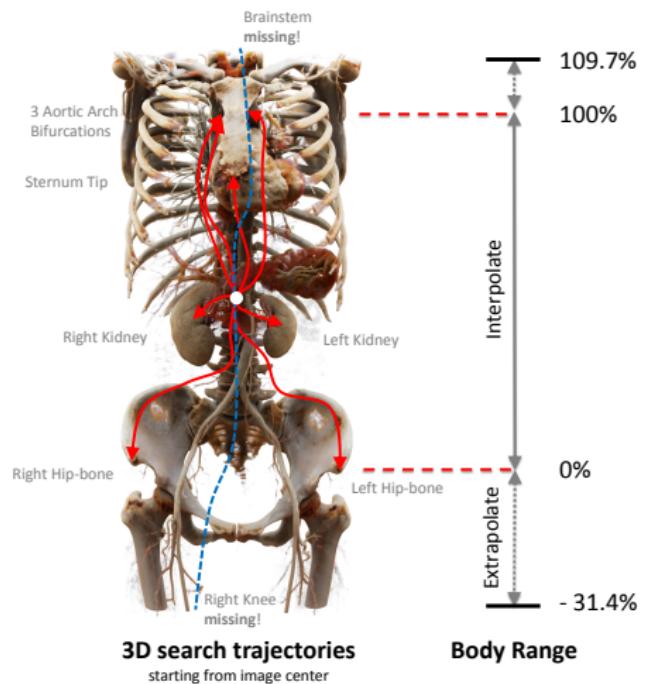
## Approach

- Deep reinforcement learning
- Learn strategies how to search objects  
→ Learn optimal shortest search through image volume to different landmarks
- Hierarchical approach to improve speed and robustness



Source: Ghesu et al. 2016, Ghesu et al. 2017

## Organ Search [7]



## Organ Search [7]



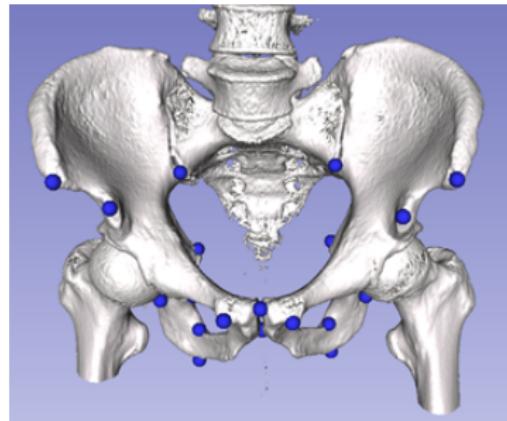
# X-ray-transform Invariant Anatomical Landmark Detection

## Goal

- Detect landmarks in X-ray images
- Knowing correspondences enables symbolic reconstruction
- Classic computervision reconstruction

## Challenge

- Transmission imaging
- Overlap/superposition of structures
- High variance due to projection
- Artifacts e.g. interventional devices



Source: Bier et al. 2018

## X-ray-transform Invariant Anatomical Landmark Detection

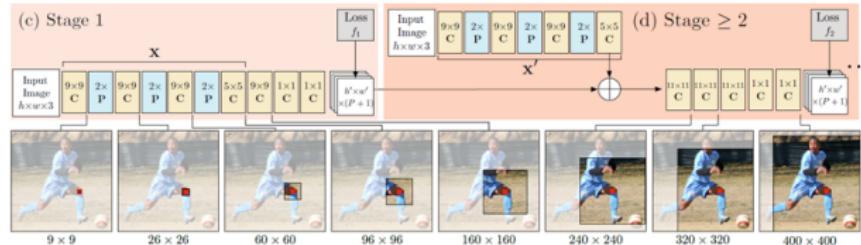
## Approach: Convolutional Pose Machine (CPM) [17]

- Sequential prediction framework to detect landmarks

→ Yields 2D belief maps

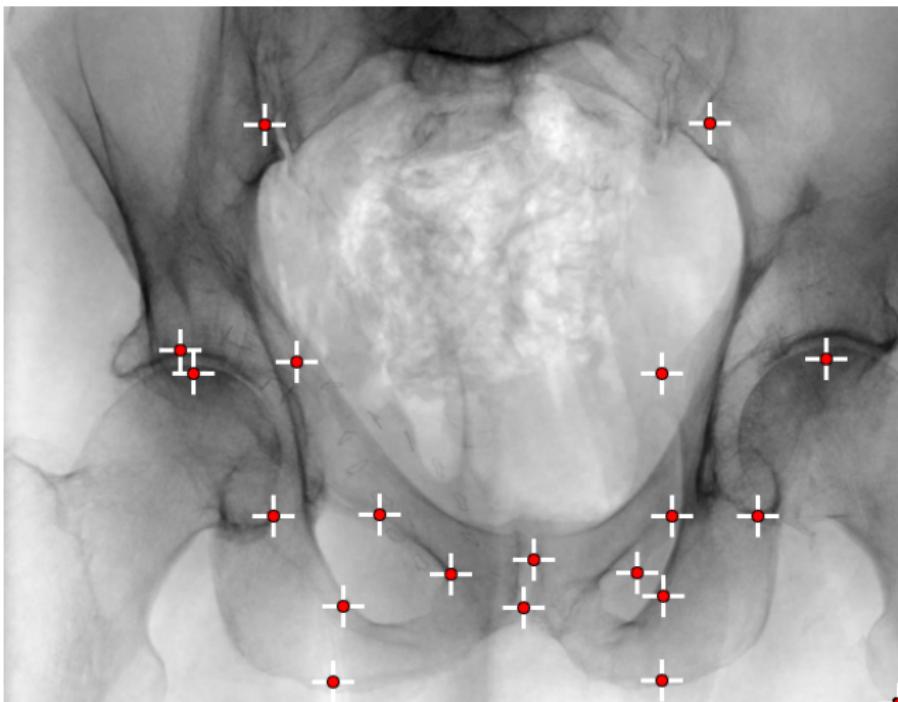
# Properties

- Large receptive fields enable learning of configurations
  - Estimation is refined over stages



Source: Wei et al., 2016

## X-ray-transform Invariant Anatomical Landmark Detection

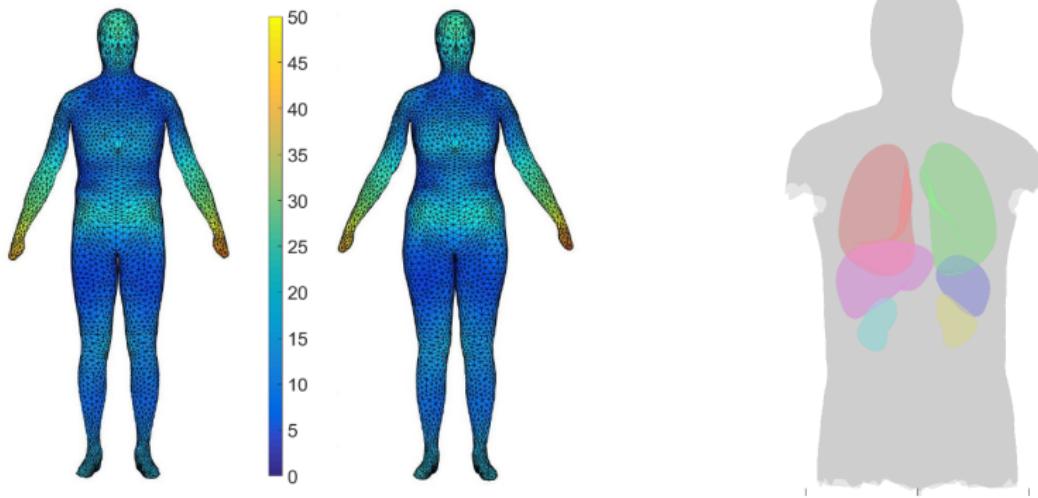


Source: Bier et al. 2018

# Organ Prediction

## Goal

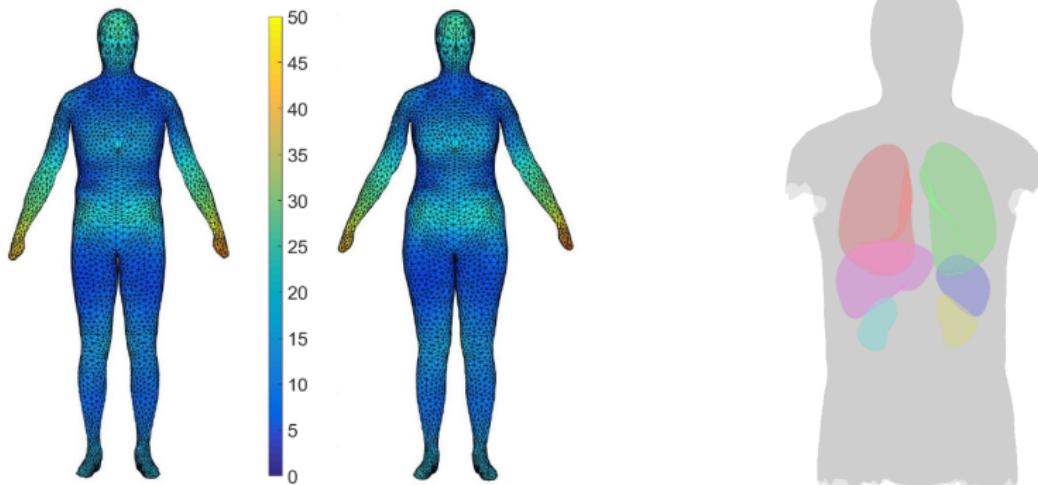
Estimation of body and organ shapes based on patient's height and weight for X-ray exposure estimation.



# Organ Prediction

## Goal

Estimation of body and organ shapes based on patient's height and weight for X-ray exposure estimation.



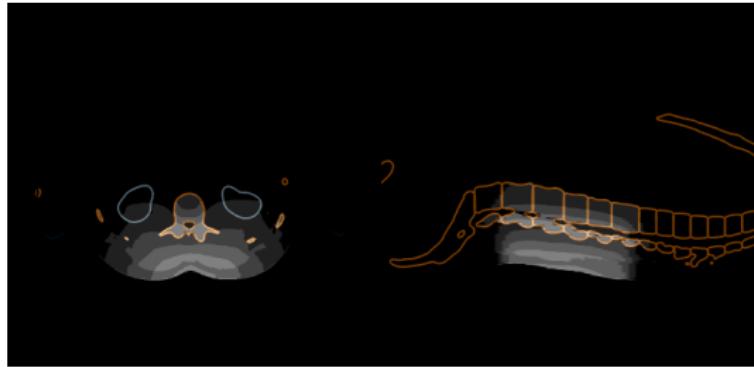
Could we achieve more if we had old CT data of a patient?

# Action Learning for 3D Point Cloud Based Organ Segmentation

**Goal: Versatile organ segmentation for:**

- Use it in computer aided diagnosis
- Treatment planning
- Dose management

**Dose estimation in interventions with overlays**

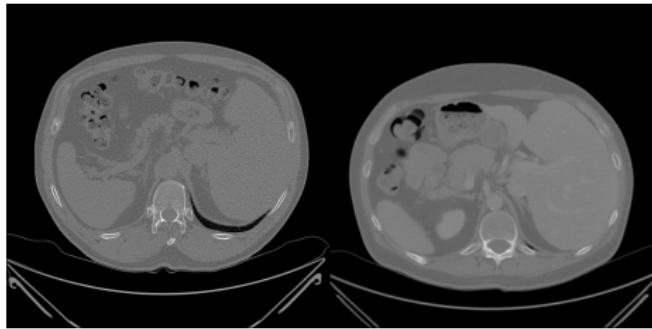


# Action Learning for 3D Point Cloud Based Organ Segmentation

## Challenges for clinical applications

- Robustness w.r.t.
  1. Individual anatomy
  2. Scan protocols
- Time constraints

Pre-operative CT (left) and contrast enhanced CT (right)



# Action Learning for 3D Point Cloud Based Organ Segmentation

- Reinforcement learning
- Predict the transformation at given state

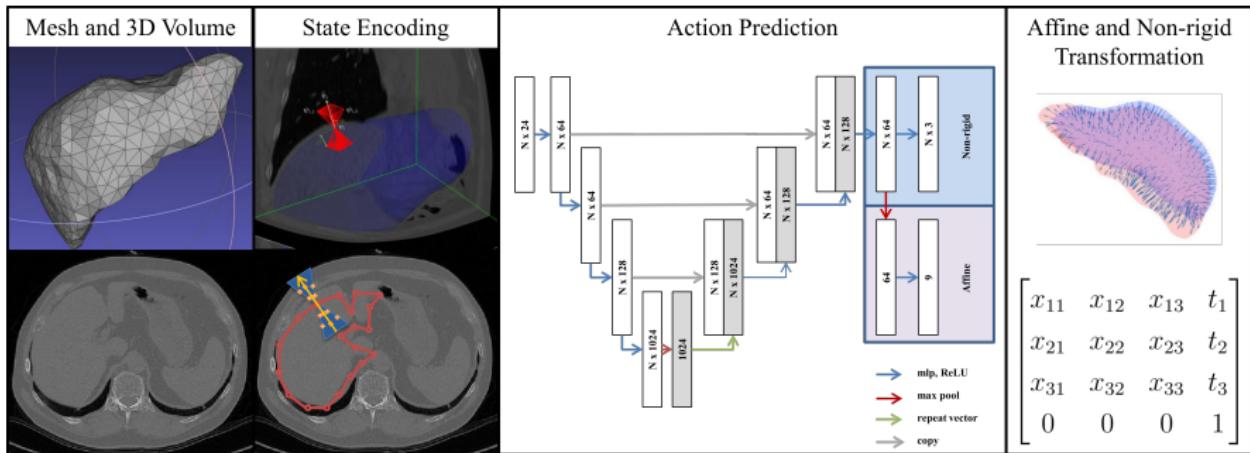
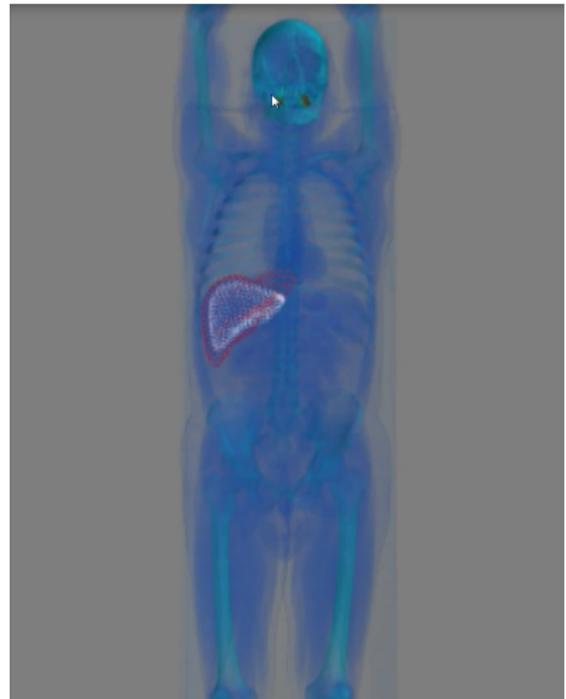


Figure: Action prediction pipeline for 3D point cloud based organ segmentation

Source: Zhong et al. 2018

# Action Learning for 3D Point Cloud Based Organ Segmentation

- Runtime:
  1. **0.3 - 2.6s per volume**
  2. **50 - 100 speedup** from U-net [5]
- Very accurate
- Robust to:
  1. scan protocol
  2. contrast agent
  3. organ initialization



Source: Zhong et al. 2018

**NEXT TIME  
ON DEEP LEARNING**



**FAU**

FRIEDRICH-ALEXANDER-  
UNIVERSITÄT  
ERLANGEN-NÜRNBERG  
SCHOOL OF ENGINEERING

# Introduction - Part 3

**A. Maier, V. Christlein, K. Breininger, Z. Yang, L. Rist, M. Nau, S. Jaganathan, C. Liu, N. Maul, L. Folle,  
K. Packhäuser, M. Zinnen**

Pattern Recognition Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg

April 15, 2024



# Limitations



## Image Captioning



Figure: "baseball player is throwing ball in game."



Figure: "girl in pink dress is jumping in air."



Figure: "man in black shirt is playing guitar."

Source: <http://cs.stanford.edu/people/karpathy/deepimagesent>

## Image Captioning & Generating



GPT4: "An athletically built man with a serious expression stands in a well-lit room with a wooden ceiling. He is wearing a blue t-shirt with the phrase 'SAVE THE CHUBBY UNICORNS' and an image of a rhinoceros. The man is taking a selfie with a modern smartphone, capturing his reflection in a mirror. In the background, one can see gym equipment including dumbbells and a weight rack. There is also a hanging chandelier with a unique design, illuminating the room"

Source: [https://twitter.com/maier\\_ak/status/1713563211574751494](https://twitter.com/maier_ak/status/1713563211574751494)

## Image Captioning & Generating

Now DALL-E3, "Create an image for this prompt in different styles"



Figure: "version 1."



Figure: "version 2."



Figure: "version 3."

Source: [https://twitter.com/maier\\_ak/status/1713563211574751494](https://twitter.com/maier_ak/status/1713563211574751494)

## Image Captioning & Generating

To DALL-E3, "Can you create a few pixel art versions of this?"



Figure: "pixel version 1."



Figure: "pixel version 2."



Figure: "pixel version 3."

Source: [https://twitter.com/maier\\_ak/status/1713563211574751494](https://twitter.com/maier_ak/status/1713563211574751494)

## Image Captioning & Generating

To DALL-E3, "Show the scene in the style of a massive online roleplaying game"



Figure: "warcraft version 1."



Figure: "warcraft version 2."



Figure: "warcraft version 3."

Source: [https://twitter.com/maier\\_ak/status/1713563211574751494](https://twitter.com/maier_ak/status/1713563211574751494)

## Image Captioning & Generating

To DALL-E3, "Make a few versions that show the scene in Lego Style"



Figure: "lego version 1."



Figure: "lego version 2."



Figure: "lego version 3."

Source: [https://twitter.com/maier\\_ak/status/1713563211574751494](https://twitter.com/maier_ak/status/1713563211574751494)

## Challenges with Training Data

- Deep learning applications often rely on **huge**, manually-annotated data sets
- Hard to obtain, time-consuming, expensive, ambiguous
- To err is human: Mislabeled ground-truth annotation
  - May cause a significant drop in performance

## Challenges with Training Data

- Deep learning applications often rely on **huge**, manually-annotated data sets
- Hard to obtain, time-consuming, expensive, ambiguous
- To err is human: Mislabeled ground-truth annotation
  - May cause a significant drop in performance
- Question: How far can we get with simulations?

## Generating Synthetic Data

- Sample from trained latent diffusion model

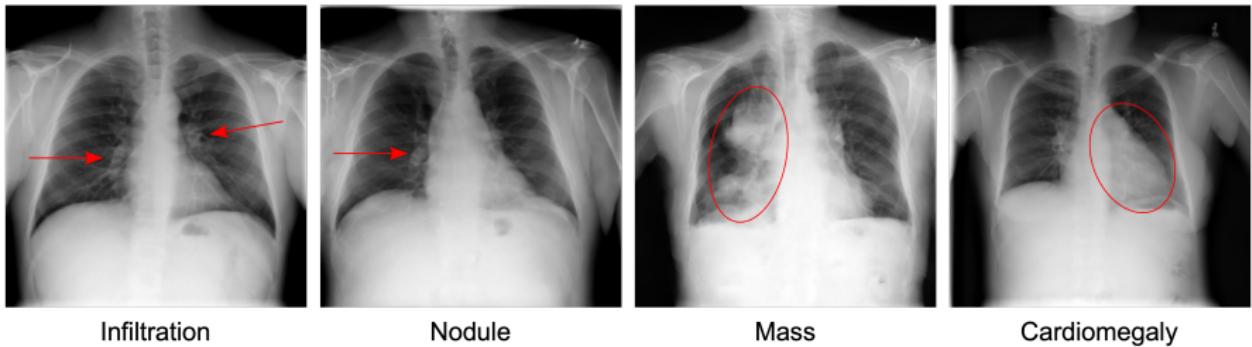


Figure: four chest X-ray images sampled from a trained latent diffusion model. Image generation was done in a conditional way to produce images of specific abnormality classes. The induced abnormality patterns in the synthetic images are highlighted with red arrows and circles.

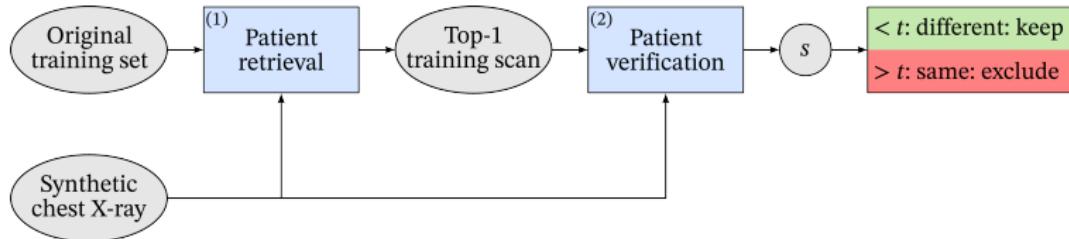
Source: Packhäuser et al. IEEE ISBI 2023

# Memorization problem for diffusion models



- Problem: models in practice tend to reproduce patterns/complete images of the used training set.
- Proposed method: uses a pre-trained patient retrieval model to compare a created synthetic scan to all training images (see pipeline below)

## Privacy-enhancing Image Sampling Strategy



## Challenges with Trust and Reliability

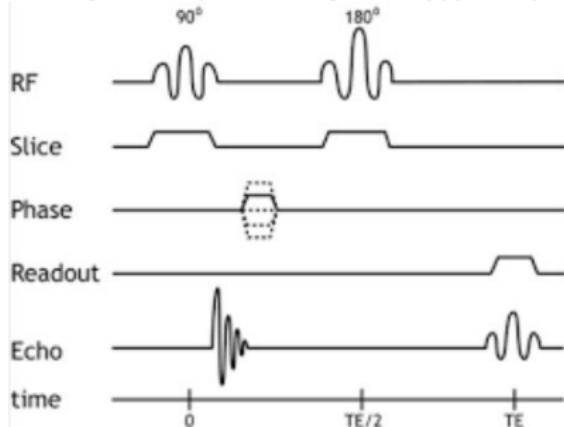
- Verification is mandatory for high risk applications
- End-to-end learning prohibits verification of parts
- Largely unsolved

## Challenges with Trust and Reliability

- Verification is mandatory for high risk applications
- End-to-end learning prohibits verification of parts
- Largely unsolved
- Possible solution: Reformulate classical algorithms

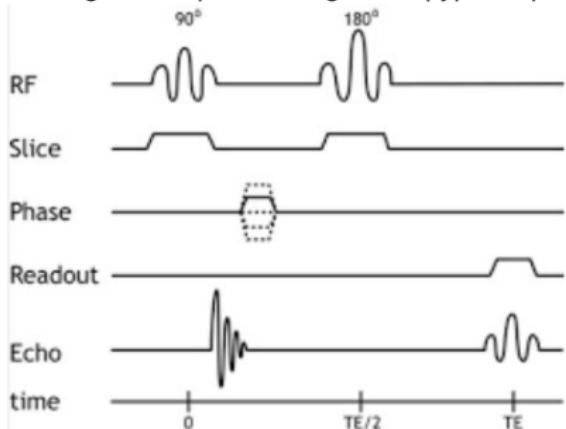
# Large Language Models for MRI Scanners

to GPT: Can you implement the sequence shown in the given sequence diagram in pypulseq?



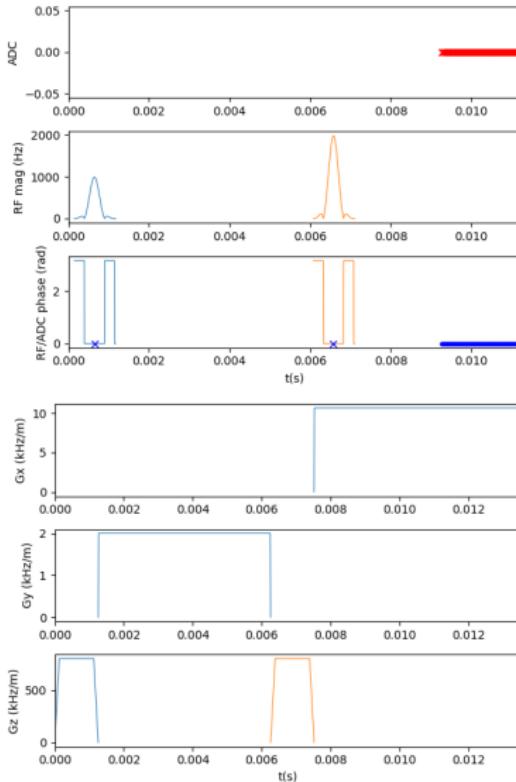
# Large Language Models for MRI Scanners

to GPT: Can you implement the sequence shown in the given sequence diagram in pypulseq?



GPT: Yes. (On the right)

- Still buggy, works only 50% of the time



# Future Directions



# Learning of Algorithms

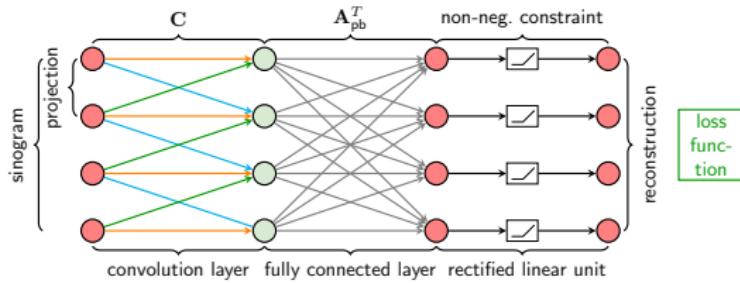
- Computed Tomography
- Efficient solution via filtered back-projection:

$$f(x, y) = \int_0^{\pi} p(s, \theta) * h(s)|_{s=x \cos \theta + y \sin \theta} d\theta$$

- Three steps:
  - Convolution along  $s$
  - Back-projection along  $\theta$
  - Suppress negative values

# Reconstruction Networks

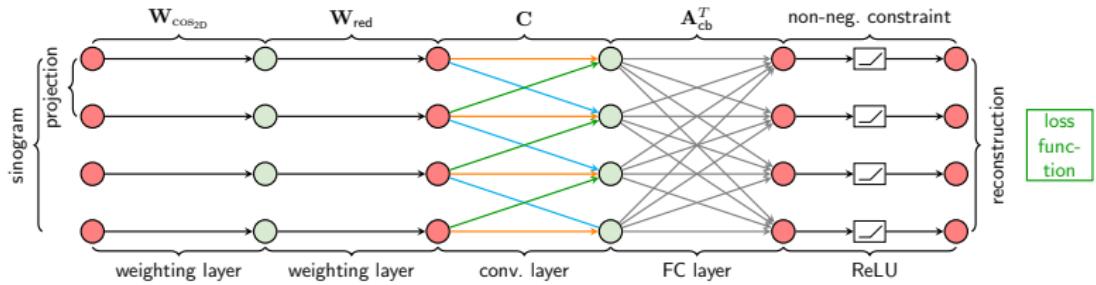
- All three steps can be modeled as a neural network:



- All weights are known from FBP

# Reconstruction Networks

- Reconstruction Networks can be expanded



- Embedding of "heuristics" for artifact reduction possible

## Application to Incomplete Scans [18]



Figure: Reconstruction with 360°

## Application to Incomplete Scans [18]

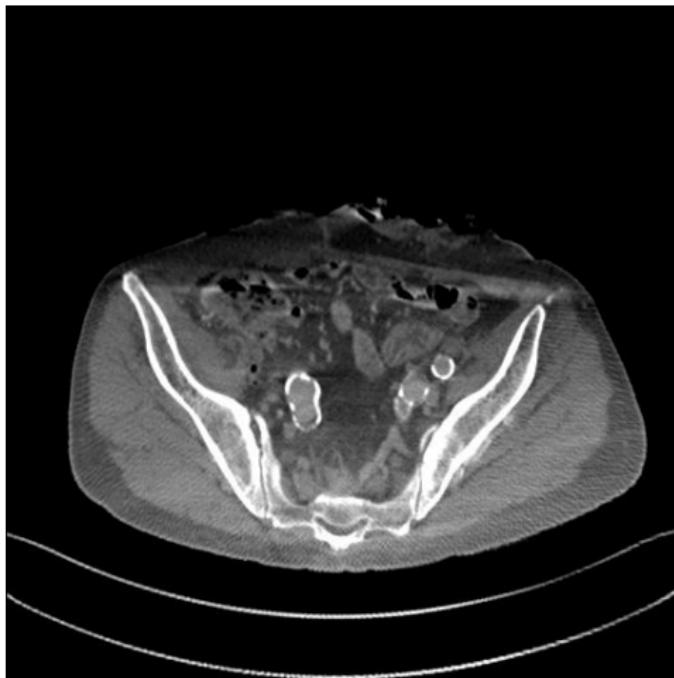


Figure: Reconstruction with 180° (FBP)

## Application to Incomplete Scans [18]



Figure: Reconstruction with  $180^\circ$  (NN)

## Application to Incomplete Scans [18]

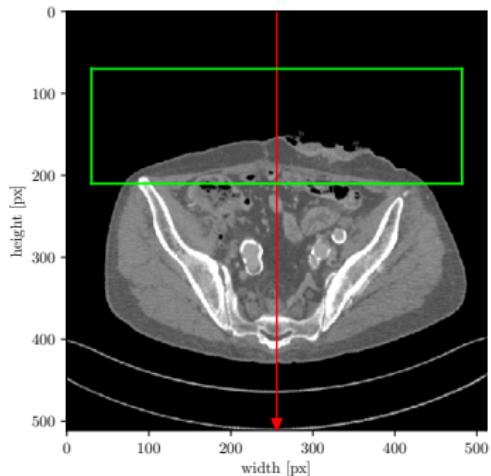


Figure: Location of the lineplot

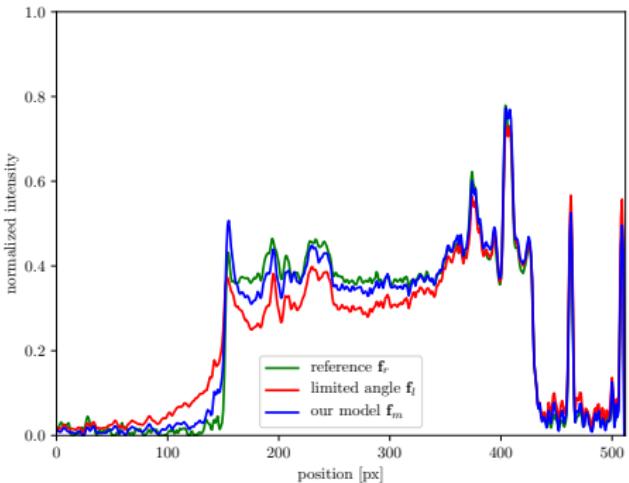


Figure: Lineplot

## Parker Weights

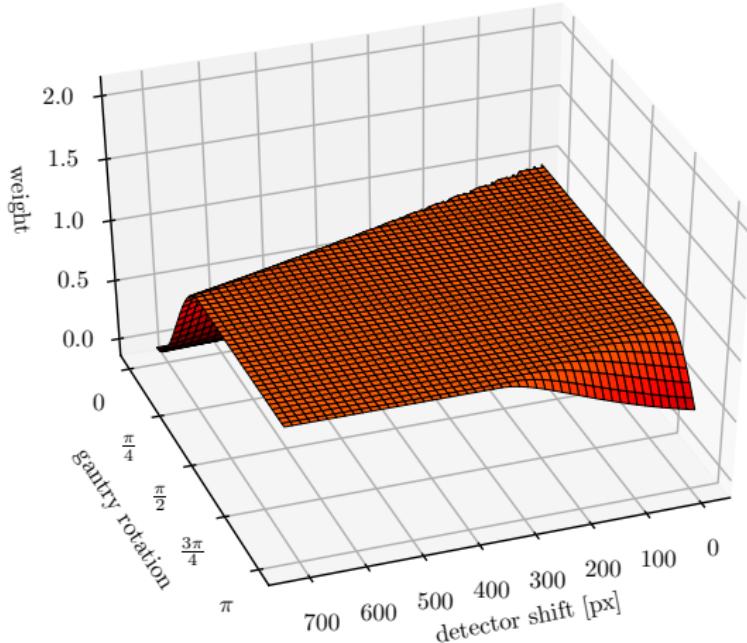


Figure: Parker weights before learning

# Parker Weights

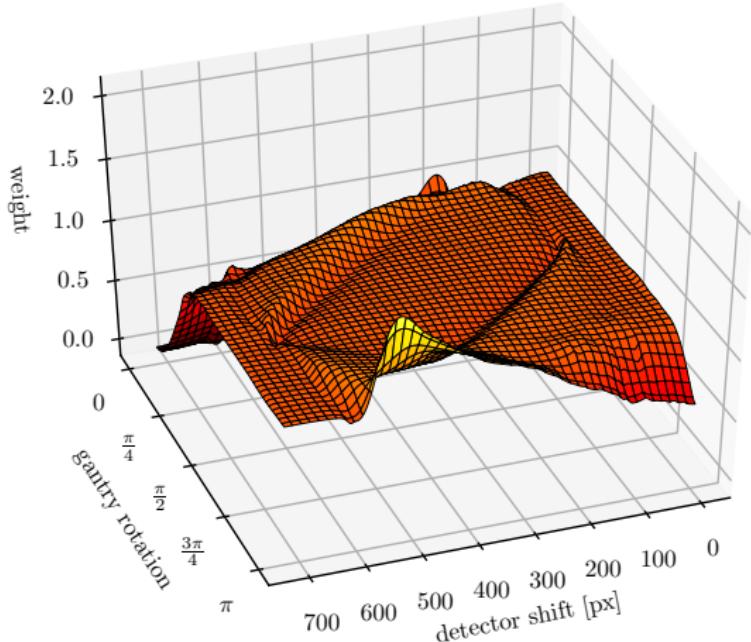
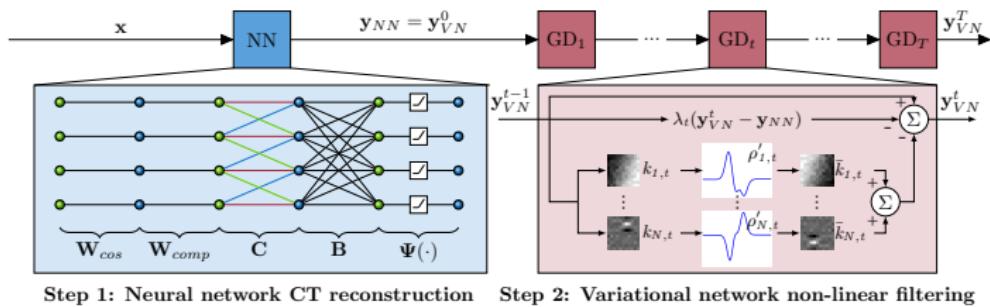


Figure: Parker weights after learning

## Further Extensions

- Add non-linear de-streaking and de-noising step:

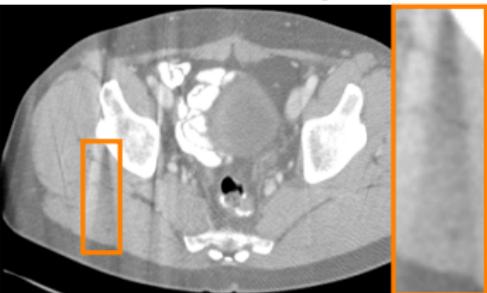


## Further Extensions

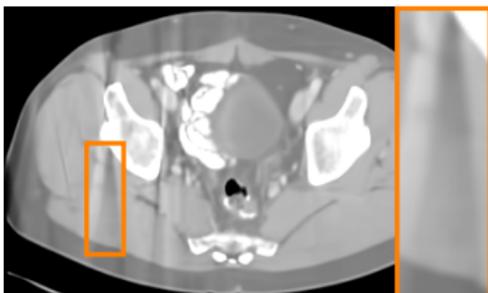
Full Scan Reference



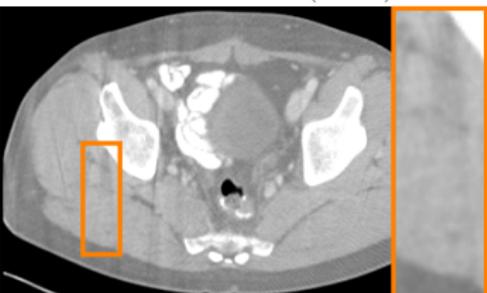
Neural Network Input



BM3D



Variational Network ( $k = 13$ )



**NEXT TIME**  
**ON DEEP LEARNING**



**FAU**

FRIEDRICH-ALEXANDER-  
UNIVERSITÄT  
ERLANGEN-NÜRNBERG  
SCHOOL OF ENGINEERING

# Introduction - Part 4

**A. Maier, V. Christlein, K. Breininger, Z. Yang, L. Rist, M. Nau, S. Jaganathan, C. Liu, N. Maul, L. Folle,  
K. Packhäuser, M. Zinnen**

Pattern Recognition Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg

April 15, 2024





**FAU**

FRIEDRICH-ALEXANDER-  
UNIVERSITÄT  
ERLANGEN-NÜRNBERG  
SCHOOL OF ENGINEERING

# Machine Learning and Pattern Recognition



## Terminology and Notation

Throughout these slides, we will use the following notation:

- Matrices: bold, uppercase, e.g.,  $\mathbf{M}$ ,  $\mathbf{A}$
- Vectors: bold, lowercase, e.g.,  $\mathbf{v}$ ,  $\mathbf{x}$
- Scalars: italic, lowercase, e.g.,  $y$ ,  $w$ ,  $\alpha$
- Gradient of a function:  $\nabla$ , partial derivative:  $\partial$

## Terminology and Notation

Throughout these slides, we will use the following notation:

- Matrices: bold, uppercase, e.g.,  $\mathbf{M}, \mathbf{A}$
- Vectors: bold, lowercase, e.g.,  $\mathbf{v}, \mathbf{x}$
- Scalars: italic, lowercase, e.g.,  $y, w, \alpha$
- Gradient of a function:  $\nabla$ , partial derivative:  $\partial$

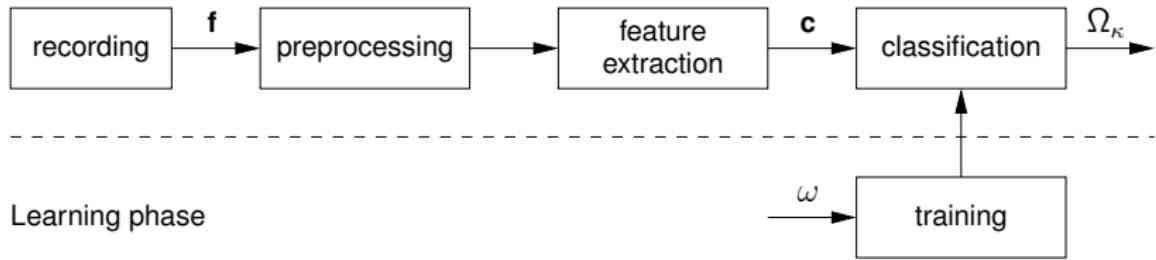
Notation regarding deep learning:

- Trainable parameters (“weights”):  $w$
- Features/input:  $\mathbf{x}$
- Ground truth label/target:  $y$
- Estimated output:  $\hat{y}$
- Index denoting iteration will be in superscript, e.g.,  $\mathbf{x}^{(i)}$

*The notation and the terminology will be further developed throughout the lecture.*

## “Classical” Image Processing Pipeline

Classification phase

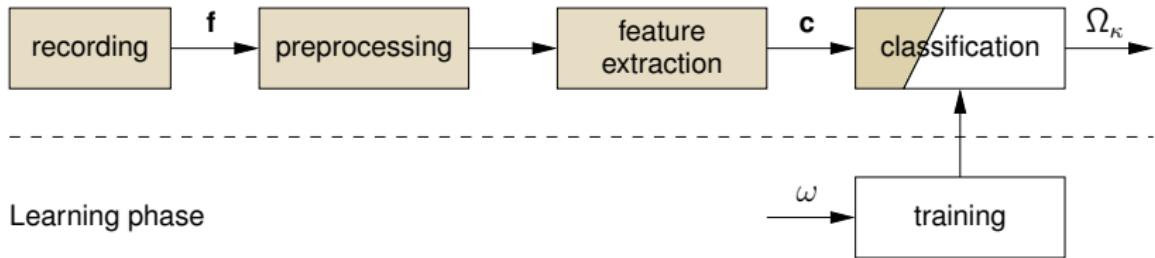


Learning phase

# “Classical” Image Processing Pipeline

Lecture Introduction to Pattern Recognition

Classification phase

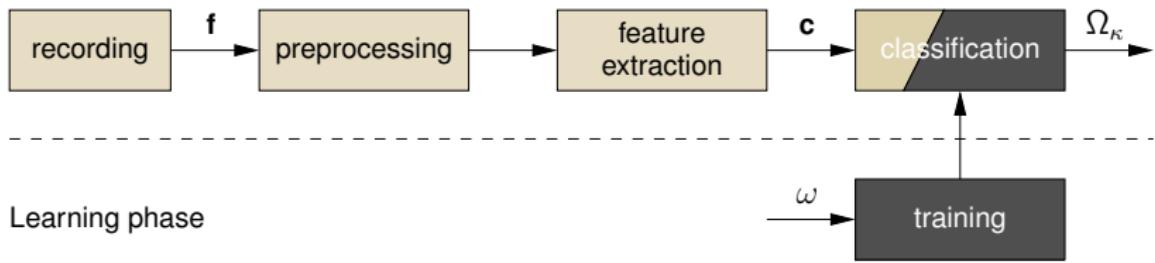


Learning phase

# “Classical” Image Processing Pipeline

Lecture Introduction to Pattern Recognition

Classification phase



Learning phase

Lecture Pattern Recognition

## “Classical” Image Processing Pipeline: Apple vs. Pears



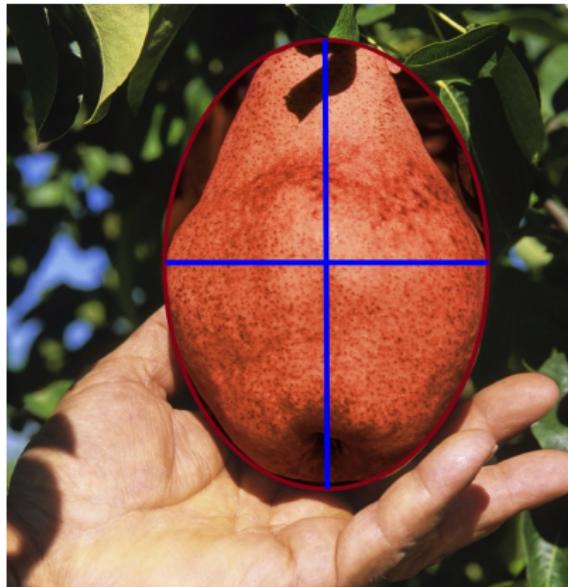
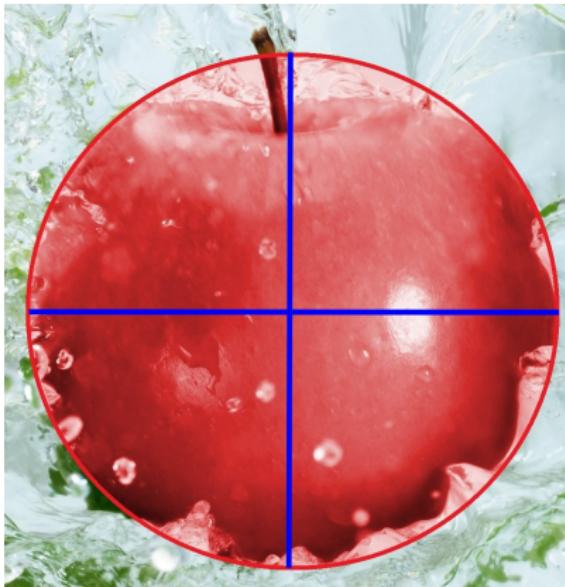
Source: <https://commons.wikimedia.org>

## “Classical” Image Processing Pipeline: Apple vs. Pears



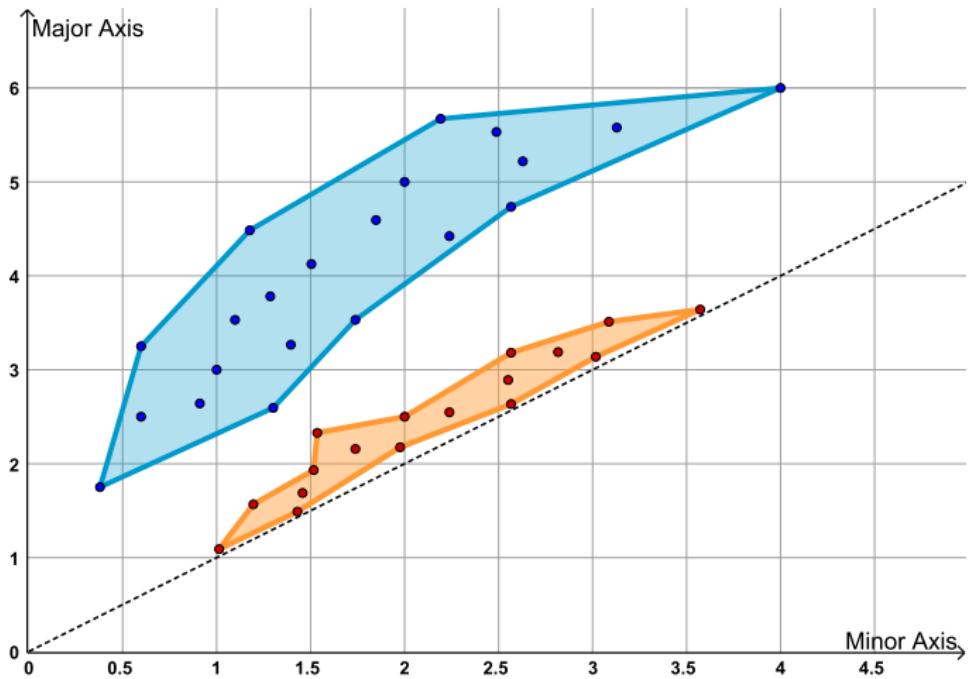
Source: <https://commons.wikimedia.org>

## “Classical” Image Processing Pipeline: Apple vs. Pears

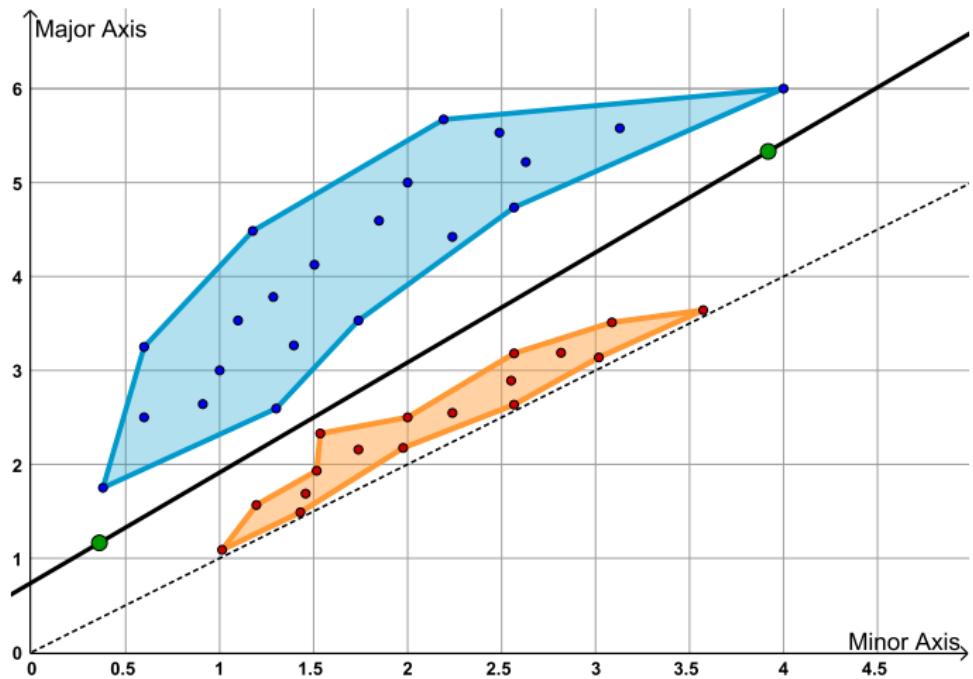


Source: <https://commons.wikimedia.org>

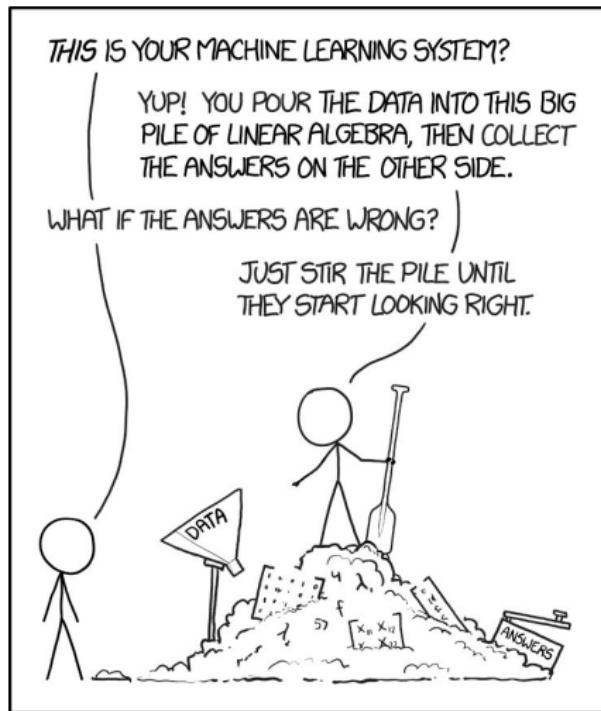
## “Classical” Image Processing Pipeline: Apple vs. Pears



## “Classical” Image Processing Pipeline: Apple vs. Pears



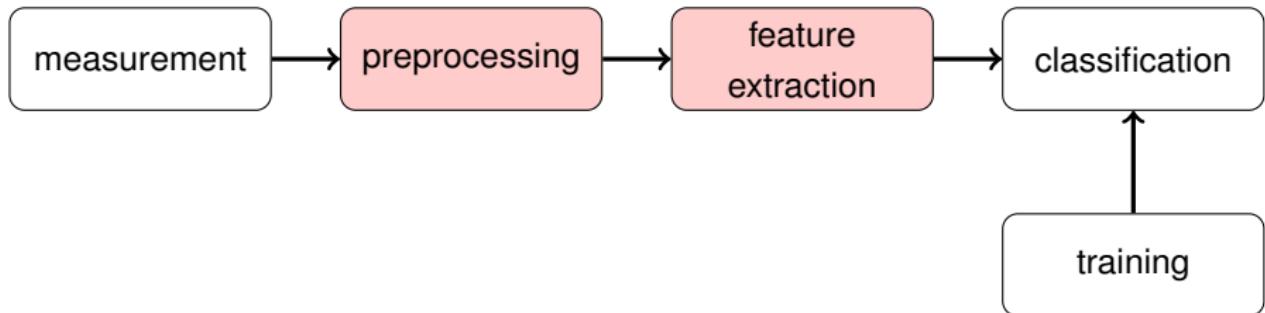
## Pipeline in Deep Learning



Source: <https://xkcd.com/1838/>

# Pipeline in Deep Learning

## Reminder



# Pipeline in Deep Learning

Now



# Postulates for Pattern Recognition

6 Postulates:

1. Availability of a **representative sample**  $\omega$  of **patterns**  ${}^i\mathbf{f}(\mathbf{x})$  for the given field of problems  $\Omega$

$$\omega = \{{}^1\mathbf{f}(\mathbf{x}), \dots, {}^N\mathbf{f}(\mathbf{x})\} \subseteq \Omega.$$

# Postulates for Pattern Recognition

6 Postulates:

1. Availability of a **representative sample**  $\omega$  of **patterns**  ${}^i\mathbf{f}(\mathbf{x})$  for the given field of problems  $\Omega$

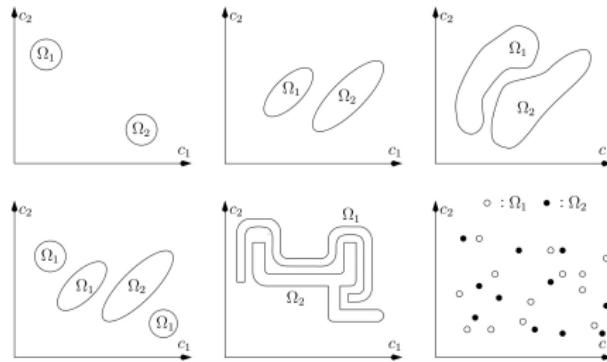
$$\omega = \{{}^1\mathbf{f}(\mathbf{x}), \dots, {}^N\mathbf{f}(\mathbf{x})\} \subseteq \Omega.$$

2. A (simple) pattern has **features**, which characterize its membership in a certain class  $\Omega_\kappa$ .

## Postulates for Pattern Recognition (cont.)

3. Compact domain of features of the same class; domains of different classes are (reasonably) separable.
- small **intra-class distance**
  - high **inter-class distance**

Example of an increasingly less compact domain in the feature space:



## Postulates for Pattern Recognition (cont.)

4. A (complex) pattern consists of **simpler constituents**, which have certain relations to each other. A pattern may be decomposed into these constituents.

## Postulates for Pattern Recognition (cont.)

4. A (complex) pattern consists of **simpler constituents**, which have certain relations to each other. A pattern may be decomposed into these constituents.
5. A (complex) pattern  $f(\mathbf{x}) \in \Omega$  has a certain **structure**. Not any arrangement of simple constituents is a valid pattern. Many patterns may be represented with relatively few constituents.

## Postulates for Pattern Recognition (cont.)

4. A (complex) pattern consists of **simpler constituents**, which have certain relations to each other. A pattern may be decomposed into these constituents.
5. A (complex) pattern  $f(\mathbf{x}) \in \Omega$  has a certain **structure**. Not any arrangement of simple constituents is a valid pattern. Many patterns may be represented with relatively few constituents.
6. Two patterns are **similar** if their features or simpler constituents differ only slightly.



**FAU**

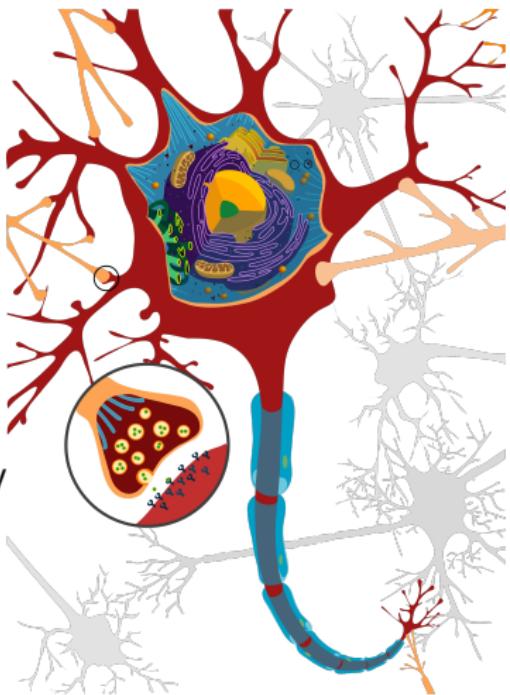
FRIEDRICH-ALEXANDER-  
UNIVERSITÄT  
ERLANGEN-NÜRNBERG  
SCHOOL OF ENGINEERING

# Perceptron



## Perceptron Biology - Neural Excitation (simplified)

- Neurons are **connected** by synapses / dendrites
- If the **sum** of incoming (excitatory and inhibitory) **activations** is large enough, an action potential is created
- The action potential activates synapses to other neurons, “transmitting” information
- All-or-none response: A **higher** stimulus does **not** cause a **higher** response → “binary classifier”



Source: <https://commons.wikimedia.org>

## Rosenblatt's Perceptron

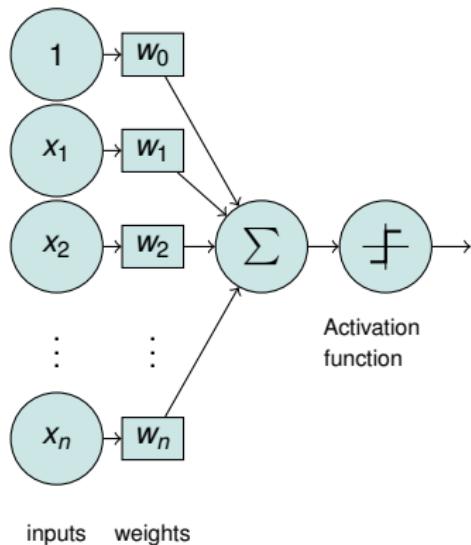
- In 1957, Frank Rosenblatt [14] invented the Perceptron
- Binary classification  $y \in \{-1, 1\}$ .
- It computes the function

$$\hat{y} = \text{sign}(\mathbf{w}^T \mathbf{x}),$$

where

$\mathbf{w} = (w_0, \dots, w_n)$ : set of weights  
 $(w_0 = \text{bias})$

$\mathbf{x} = (1, x_1, \dots, x_n)$ : input feature vector



## Perceptron Objective Function

Task: Find weights that minimize the distance of misclassified samples to the decision boundary

### Assumptions

- Let  $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)\}$  be a training data set
- Let  $\mathcal{M}$  be the set of misclassified feature vectors  $y_i \neq \hat{y}_i = \text{sign}(\mathbf{w}^\top \mathbf{x}_i)$  according to a given set of weights  $\mathbf{w}$
- Optimization problem:

$$\operatorname{argmin}_{\mathbf{w}} \quad \left\{ D(\mathbf{w}) = - \sum_{\mathbf{x}_i \in \mathcal{M}} y_i \cdot (\mathbf{w}^\top \mathbf{x}_i) \right\}$$

## Perceptron Objective Function – Observations

- Objective function depends on misclassified feature vectors  $\mathcal{M} \rightarrow$  iterative optimization
- In each iteration, the cardinality and composition of  $\mathcal{M}$  may change
- The gradient of the objective function is:

$$\nabla D(\mathbf{w}) = - \sum_{x_i \in \mathcal{M}} y_i \cdot \mathbf{x}_i$$

## Perceptron Training

- Strategy 1: Process all samples, then perform weight update
- Strategy 2: Take an update step right after each misclassified sample
- Update rule in iteration  $(k + 1)$  for the misclassified sample  $\mathbf{x}_i$  simplifies to:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} + y_i \cdot \mathbf{x}_i$$

- Optimization until convergence or for a predefined number of iterations

**NEXT TIME  
ON DEEP LEARNING**



**FAU**

FRIEDRICH-ALEXANDER-  
UNIVERSITÄT  
ERLANGEN-NÜRNBERG  
SCHOOL OF ENGINEERING

# Introduction - Part 5

**A. Maier, V. Christlein, K. Breininger, Z. Yang, L. Rist, M. Nau, S. Jaganathan, C. Liu, N. Maul, L. Folle,  
K. Packhäuser, M. Zinnen**

Pattern Recognition Lab, Friedrich-Alexander-Universität Erlangen-Nürnberg

April 15, 2024





**FAU**

FRIEDRICH-ALEXANDER-  
UNIVERSITÄT  
ERLANGEN-NÜRNBERG  
SCHOOL OF ENGINEERING

# Organizational Matters



## Grading

- Module consists of lecture **and** exercises (together 5 ECTS)
- 90 min. written exam in the semester break, determines grade
- Exercises are **optional**. 100% exercise completion = 10% grade when you pass the exam

## Exercise Content

- Python introduction
- Developing a neural network framework from scratch
  - Feed Forward Neural Networks
  - Convolutional Neural Networks
  - Regularization
  - Recurrent Networks
- Using the PyTorch framework
  - Large scale classification

## Exercise Requirements

- Basic knowledge of Python and Numpy
- Linear algebra, -
- Image processing, -
- Pattern recognition fundamentals
- Passion for coding
- Attention to detail
- Time

## How it works

- Five exercises throughout the semester
- Unit tests for all but last exercise
- Last exercise: PyTorch + Challenge
- Assistance during exercise sessions
- Personal demonstration of every exercise to get bonus points
- Exercise deadlines are announced in the respective exercise sessions

## Summary

- Deep learning more and more present in day to day life
- Huge support and interest from industry
- **Very** active area of research!
- Perceptron as binary classifier motivated by biological neurons

**NEXT TIME  
ON DEEP LEARNING**

## Next Lecture Block

- Extending the Perceptron to obtain a universal function approximator
- Gradient based training algorithm for these models
- Efficient automatic computation of gradients

## Comprehensive Questions

- What are the six postulates of pattern recognition?
- What is the Perceptron objective function?
- Can you name three applications successfully tackled by deep learning?

## Further Reading

- [Link](#) - Deep learning book
- [Link](#) - Research and publications at the Pattern Recognition Lab
- [Link](#) - Google Research Blog with posts on e.g. [Deep dream](#) or [Alpha Go](#)

**Questions?**



**FAU**

FRIEDRICH-ALEXANDER-  
UNIVERSITÄT  
ERLANGEN-NÜRNBERG  
SCHOOL OF ENGINEERING

# References



## References I

- [1] David Silver, Julian Schrittwieser, Karen Simonyan, et al. "Mastering the game of go without human knowledge". In: [Nature](#) 550.7676 (2017), p. 354.
- [2] David Silver, Thomas Hubert, Julian Schrittwieser, et al. "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm". In: [arXiv preprint arXiv:1712.01815](#) (2017).
- [3] M. Aubreville, M. Krappmann, C. Bertram, et al. "A Guided Spatial Transformer Network for Histology Cell Differentiation". In: [ArXiv e-prints](#) (July 2017). arXiv: 1707.08525 [cs.CV].
- [4] David Bernecker, Christian Riess, Elli Angelopoulou, et al. "Continuous short-term irradiance forecasts using sky images". In: [Solar Energy](#) 110 (2014), pp. 303–315.

## References II

- [5] Patrick Ferdinand Christ, Mohamed Ezzeldin A Elshaer, Florian Ettlinger, et al. "Automatic liver and lesion segmentation in CT using cascaded fully convolutional neural networks and 3D conditional random fields". In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer. 2016, pp. 415–423.
- [6] Vincent Christlein, David Bernecker, Florian Höning, et al. "Writer Identification Using GMM Supervectors and Exemplar-SVMs". In: Pattern Recognition 63 (2017), pp. 258–267.
- [7] Florin Cristian Ghesu, Bogdan Georgescu, Tommaso Mansi, et al. "An Artificial Agent for Anatomical Landmark Detection in Medical Images". In: Medical Image Computing and Computer-Assisted Intervention - MICCAI 2016. Athens, 2016, pp. 229–237.

## References III

- [8] Jia Deng, Wei Dong, Richard Socher, et al. "Imagenet: A large-scale hierarchical image database". In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on IEEE. 2009, pp. 248–255.
- [9] A. Karpathy and L. Fei-Fei. "Deep Visual-Semantic Alignments for Generating Image Descriptions". In: ArXiv e-prints (Dec. 2014). arXiv: 1412.2306 [cs.CV].
- [10] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: Advances in Neural Information Processing Systems 25. Curran Associates, Inc., 2012, pp. 1097–1105.

## References IV

- [11] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, et al. "You Only Look Once: Unified, Real-Time Object Detection". In: [CoRR abs/1506.02640](#) (2015).
- [12] J. Redmon and A. Farhadi. "YOLO9000: Better, Faster, Stronger". In: [ArXiv e-prints](#) (Dec. 2016). arXiv: 1612. 08242 [cs.CV].
- [13] Joseph Redmon and Ali Farhadi. "YOLOv3: An Incremental Improvement". In: [arXiv](#) (2018).
- [14] Frank Rosenblatt. [The Perceptron—a perceiving and recognizing automaton.](#) 85-460-1. Cornell Aeronautical Laboratory, 1957.
- [15] Olga Russakovsky, Jia Deng, Hao Su, et al. "ImageNet Large Scale Visual Recognition Challenge". In: [International Journal of Computer Vision](#) 115.3 (2015), pp. 211–252.

## References V

- [16] David Silver, Aja Huang, Chris J. Maddison, et al. "Mastering the game of Go with deep neural networks and tree search". In: Nature 529.7587 (Jan. 2016), pp. 484–489.
- [17] S. E. Wei, V. Ramakrishna, T. Kanade, et al. "Convolutional Pose Machines". In: CVPR. 2016, pp. 4724–4732.
- [18] Tobias Würfl, Florin C Ghesu, Vincent Christlein, et al. "Deep learning computed tomography". In:  
International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer International Publishing. 2016, pp. 432–440.