Organization and Introduction

Advanced Computer Vision

Niels Landwehr

Agenda

- Organizational Overview
- Introduction to Computer Vision

Agenda

- Organizational Overview
- Introduction to Computer Vision

Welcome

- **Research group „Data Science“ at the Institute for Computer Science**

  Headed by Prof. Dr. Niels Landwehr

  Research and teaching in machine learning and its applications

  Contact:
  <landwehr@uni-hildesheim.de>
  Building J, Room J 103 on main campus

- Teaching of the group this semester:
  - Lecture: Machine Learning
  - Lab: Programming Machine Learning
  - **Lecture: Advanced Computer Vision**
  - Seminars Data Analytics I/II/III

Organization: Degree Programmes

- **Lecture „Advanced Computer Vision“: 2+2 SWS, 6 CPs**

- Within the international Msc degree „Data Analytics“:
  - Module „Advanced Computer Vision“
  - Methodological specialization within „Data Analytics“ programme

- Within the Msc degrees „Informationsmanagement und Informationstechnologie“ and „Angewandte Informatik“
  - Module „Advanced Computer Vision“
  - Optional choice („Wahlmodul Informatik“)

- Other degrees: please check degree regulations

Organization: Lecture

- The lecture and the tutorials will be completely in English
- Lecture takes place every Tuesday 12:15 – 13:45 in room B 0.37 (Samelson campus)

- **Learnweb page of course: https://www.uni-hildesheim.de/learnweb2023/course/view.php?id=2988**
  - Information about organization, dates,...
  - Lecture slides
  - Weekly exercise sheets
  - Please check the Learnweb frequently!

Organization: Tutorials

- The lecture will be accompanied by weekly exercises (some theoretical, some practical programming exercises)
- Tutorial takes place every Friday 12:15 – 13:45 in seminar room G 009 (main campus)
- Exercise sheets will be posted each Tuesday on the Learnweb. You should complete the exercise sheet and upload the solution in the Learnweb until the following Tuesday
- In the tutorial on Friday, you will present your solutions to the exercise handed in on Tuesday and there is room to discuss questions

Prerequisites for Course

- **Prerequisites for course**

- The course deals with deep learning for computer vision
  - Some prior knowledge in machine learning is very much recommended
  - In particular, advantageous to have completed the lecture "Machine Learning" or at least take it simultaneously
  - Knowledge in deep learning is helpful but not mandatory
  - Lecture will include some review of central machine learning and deep learning concepts
  - For programming exercises, familiarity with Python is required

Questions?

Agenda

- Organizational Overview
- Introduction to Computer Vision

Computer Vision: Definition

- **What is computer vision?**

> “Computer vision describes the automatic deduction of the structure and the properties of a (possibly dynamic) three-dimensional world from either a single or multiple two-dimensional images of the world"
>
> Vishvjit S. Nalwa: A guided tour of computer vision. Addison-Wesley, 1993

> “Computer vision is an exciting new research area that studies how to make computers efficiently perceive, process, and understand visual data such as images and videos. The ultimate goal is for computers to emulate the striking perceptual capability of human eyes and brains, or even to surpass and assist the human in certain ways”"
>
> Microsoft Research

Computer Vision: Definition

- **Computer vision: analyze visual data in order to extract semantic information**

- Visual data can be
  - a single image, multiple images, or video
  - images can be grayscale, color („RGB“), or multispectral/hyperspectral
  - in some cases, also depth information can be available (e.g. from 3D time-of-flight cameras or laser scanners)

- Semantic information can be
  - which category of image is this (classification, e.g. cat versus dog)
  - localize and classify multiple objects in image (e.g. cat here, dog there)
  - extract relationships between objects
  - estimate shapes of objects
  - in general, any information a human observer could deduct from image

Computer Vision Versus Computer Graphics

- Computer vision and computer graphics solve inverse problems
  - Computer vision: analyze visual data to infer semantic content
  - Computer graphics: generate visual data from semantic (physical) model
- Example:
  - Computer vision: infer 3D-model of human movement from 2D-video
  - Computer graphics: animate human character from 3D-model

Visual data (2D-image)

Computer vision

Computer graphics

Semantic model (3D)

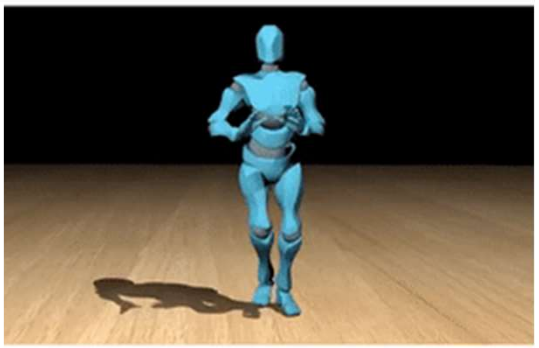
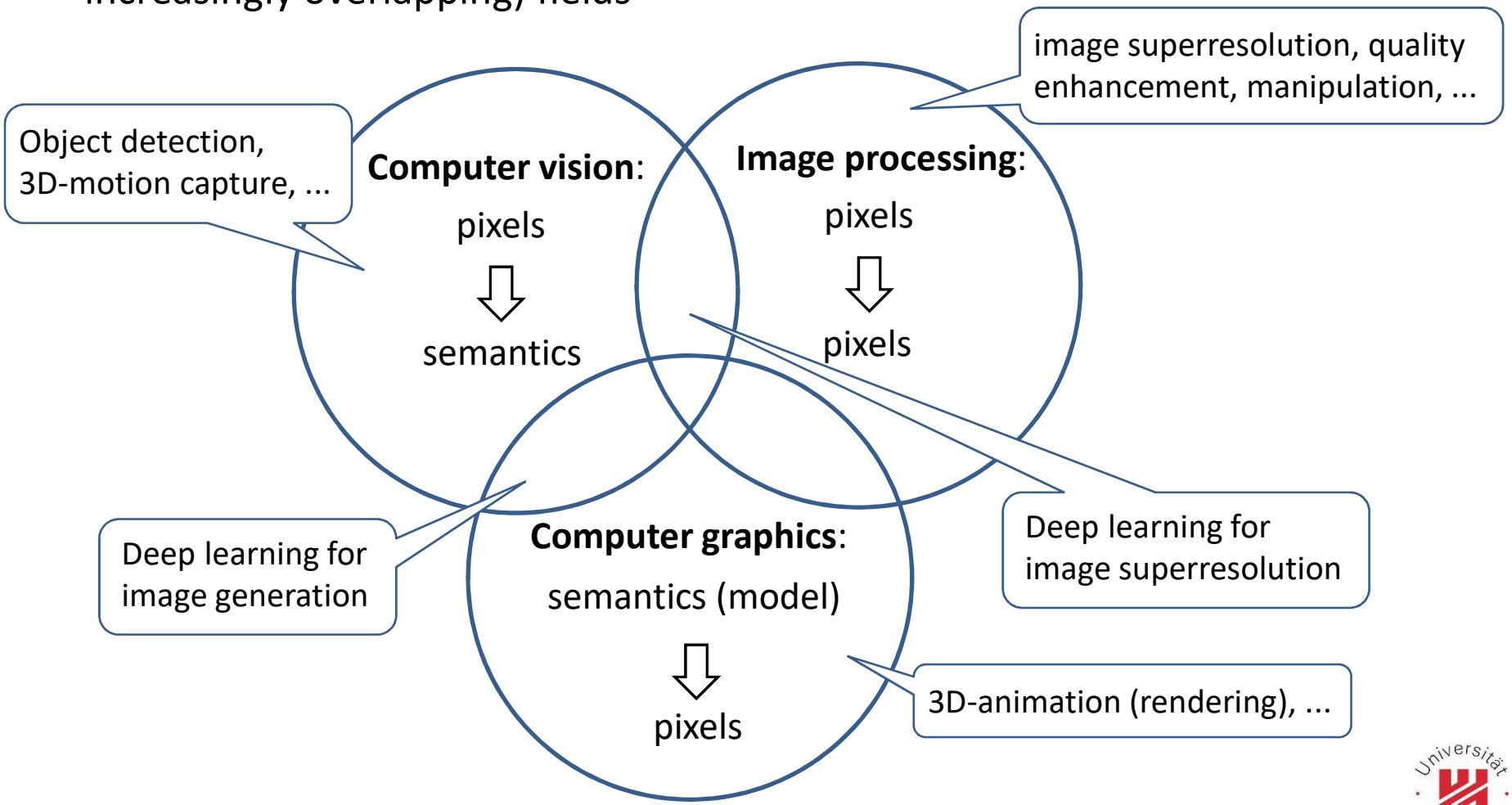


Image: Stanford University / Adobe (3D motion estimate)

Computer Vision, Image Processing and Computer Graphics

- Computer vision, image processing and computer graphics are related (and increasingly overlapping) fields



Computer Vision Techniques for Image Generation and Image Manipulation

- Techniques originally developed in computer vision are being increasingly used also for image generation and image processing

E.g. generation of photorealistic images of faces using deep neural networks

Karras et al., 2018 “A Style-Based Generator Architecture for Generative Adversarial Networks”

E.g. image superresolution using deep neural networks

low-res image

super resolution

Dai et al. 2019, “Second-order Attention Network for Single Image Super-Resolution”

Computer Vision: Applications

- Computer vision has many practical applications
  - Automotive: autonomous driving, car safety, assistant systems, ...
  - Robotics: localizing objects, inferring 3D geometry of scene, ...
  - Medical domains, e.g. analyzing x-ray or MRI images

https://www.zf.com/mobile/en/technologies/domains/autonomous_driving/autonomous_driving.html

Why is Computer Vision Difficult?

- **Why is computer vision difficult?**
- Task in computer vision is to go from low-level visual information (pixels in images) to high-level semantic information

„A cat!“

This image by Nikita is licensed under CC-BY 2.0

```
[[105 112 108 111 104  99 106  99  96 103 112 119 104  97  93  87]
 [ 91  98 102 106 104  79  98 103  99 105 123 136 110 105  94  85]
 [ 76  85  90 105 128 105  87  96  95  99 115 112 106 103  99  85]
 [ 99  81  81  93 120 131 127 100  95  98 102  99  96  93 101  94]
 [106  91  61  64  69  91  88  85 101 107 109  98  75  84  96  95]
 [114 108  85  55  55  69  64  54  64  87 112 129  98  74  84  91]
 [133 137 147 103  65  81  80  65  52  54  74  84 102  93  85  82]
 [128 137 144 140 109  95  86  70  62  65  63  63  60  73  86 101]
 [125 133 148 137 119 121 117  94  65  79  80  65  54  64  72  98]
 [127 125 131 147 133 127 126 131 111  96  89  75  61  64  72  84]
 [115 114 109 123 150 148 131 118 113 109 100  92  74  65  72  78]
 [ 89  93  90  97 108 147 131 118 113 114 113 109 106  95  77  80]
 [ 63  77  86  81  77  79 102 123 117 115 117 125 125 130 115  87]
 [ 62  65  82  89  78  71  80 101 124 126 119 101 107 114 131 119]
 [ 63  65  75  88  89  71  62  81 120 138 135 105  81  98 110 118]
 [ 87  65  71  87 106  95  69  45  76 130 126 107  92  94 105 112]
 [118  97  82  86 117 123 116  66  41  51  95  93  89  95 102 107]
 [164 146 112  80  82 120 124 104  76  48  45  66  88 101 102 109]
 [157 170 157 120  93  86 114 132 112  97  69  55  70  82  99  94]
 [130 128 134 161 139 100 109 118 121 134 114  87  65  53  69  86]
 [128 112  96 117 150 144 120 115 104 107 102  93  87  81  72  79]
 [123 107  96  86  83 112 153 149 122 109 104  75  80 107 112  99]
 [122 121 102  80  82  86  94 117 145 148 153 102  58  78  92 107]
 [122 164 148 103  71  56  78  83  93 103 119 139 102  61  69  84]]
```

Computer only „sees“ big grid of numbers (pixel values)

J. Johnson, 2019

- **„Semantic gap“**: pixel values determine semantic content, but relationship is not at all straightforward
- The semantic class „cat“ can be represented by very different grids of pixels

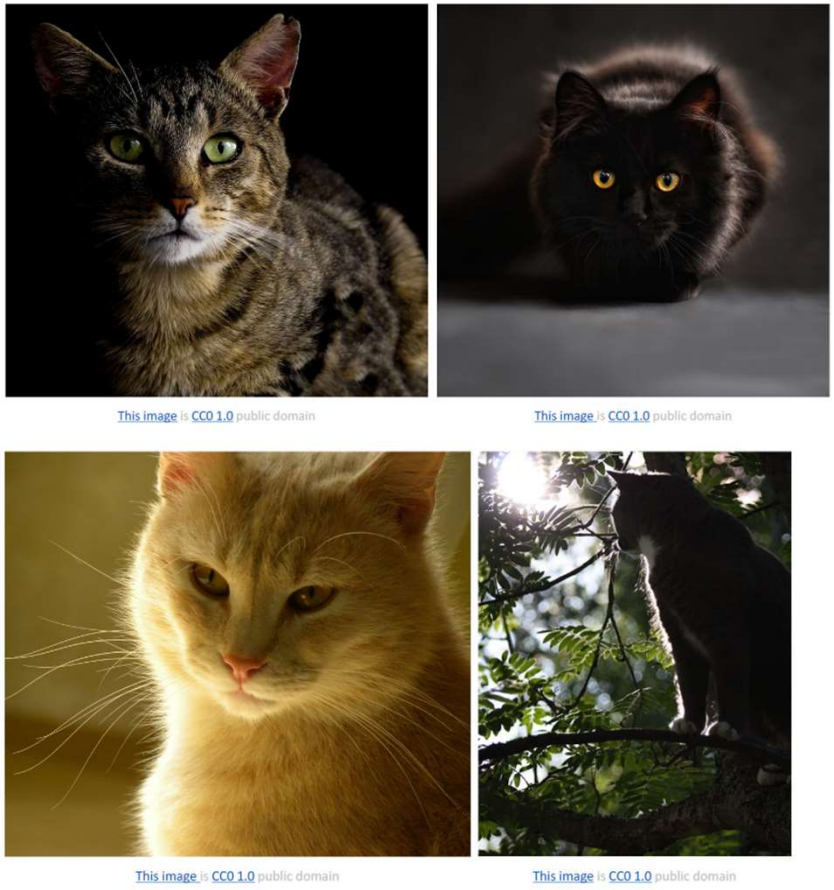
Challenges: Intraclass Variation

- **Challenges:** intraclass variation (not all objects in class look the same)

J. Johnson, 2019

Challenges: Illumination, Perspective

- **Challenges:** Illumination, perspective



J. Johnson, 2019

Even for the same object,
changes in perspective
or illumination completely
change the raw pixel values

Challenges: Background Clutter

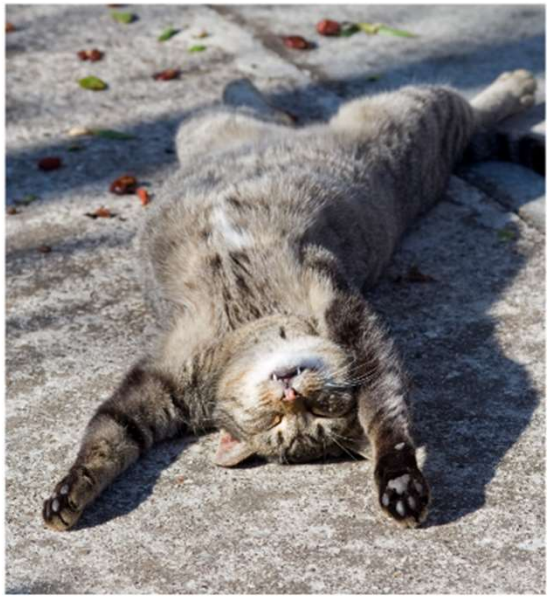
- **Challenges:** Background clutter

This image is CC0 1.0 public domain

J. Johnson, 2019

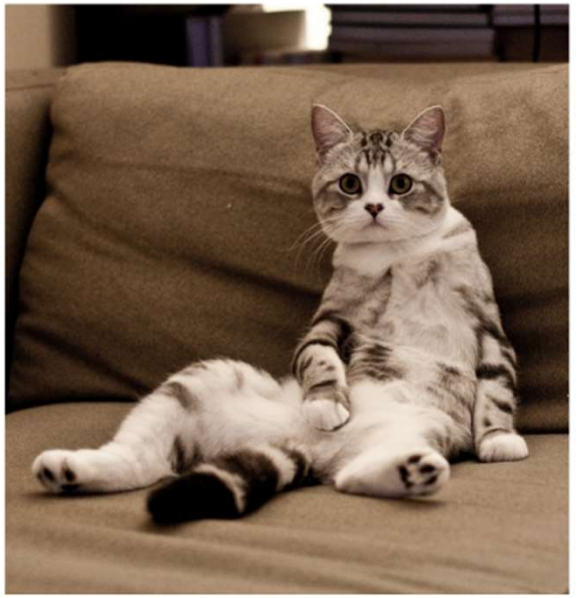
Challenges: Deformation

- **Challenges:** Deformation



This image by Umberto Salvagnin is licensed under CC-BY 2.0

This image by sare bear is licensed under CC-BY 2.0



This image by Tom Thai is licensed under CC-BY 2.0

J. Johnson, 2019

Challenges: Occlusions

- **Challenges:** Occlusions

This image is CC0 1.0 public domain

This image is CC0 1.0 public domain

This image by jonsson is licensed under CC-BY 2.0

J. Johnson, 2019

Computer Vision Versus Human Vision

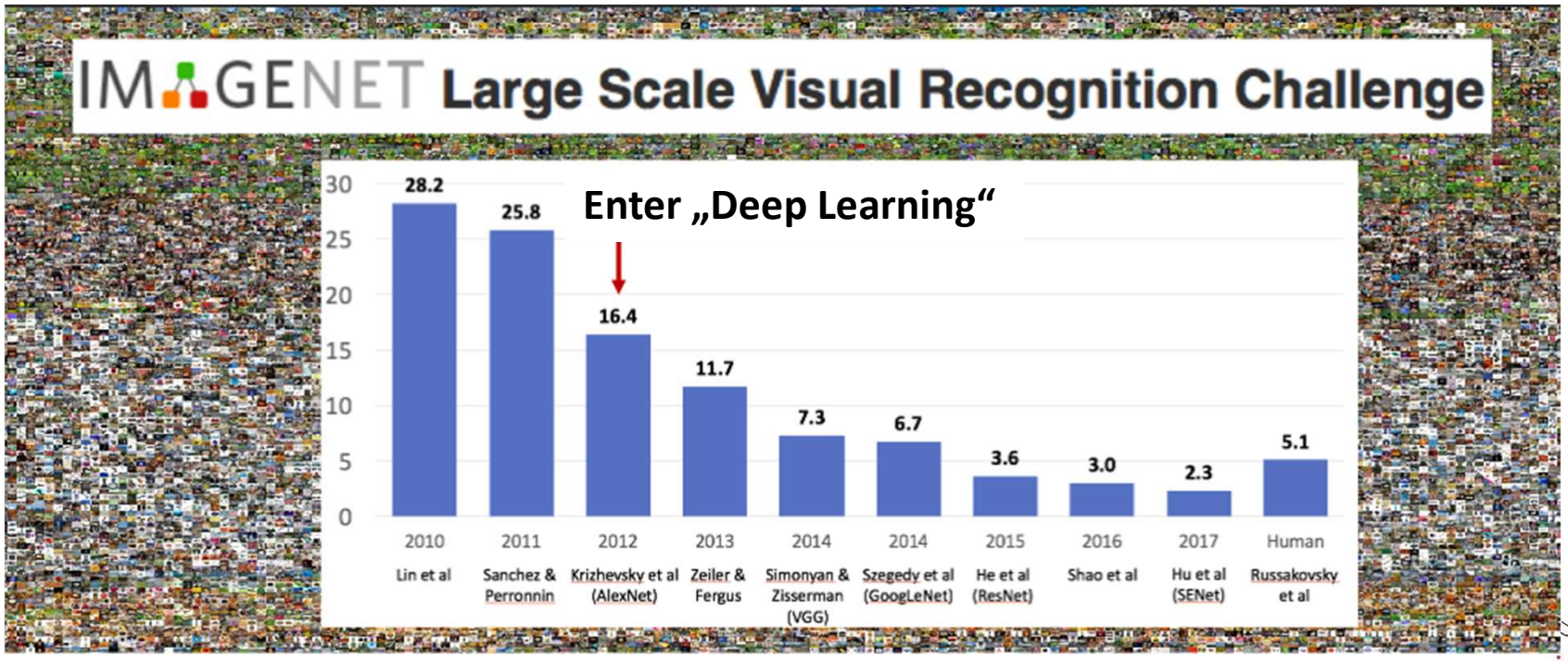
- Humans have striking perceptual abilities:
  - Despite the outlined challenges, we can recognize objects and understand semantic content of images reliably and within fractions of a second
  - Important for success/survival (evolutionary speaking)
  - Learned skill: exposed to constant flow of visual information from birth
- Historically speaking, computers have for a long time been far inferior to humans at solving visual problems
  - Very difficult to go from pixel information to semantic image content
  - Robust solutions have only been available for very narrow, tightly controlled environments

E.g. recognizing scanned digits:
constant lighting and viewing angle

- However, relatively recently (last 5-10 years) advances in modern machine learning have dramatically advanced the state of the art

Example: Improving Accuracy on ImageNet Benchmark

- Example: improving accuracy of computer vision methods on ImageNet
  - large benchmark data set to track progess in the field of computer vision
  - Accuracy has improved dramatically, to superhuman performance (in this narrow, well-defined classification task)



Machine Learning for Computer Vision

- Almost all state-of-the-art computer vision methods are data-driven methods
  - Collect **data** about the computer vision problem to solve
  - Use **machine learning** to train a model that solves the actual problem

- Simple example: image classification, cats versus dogs

Image: P. Radhakrishnan

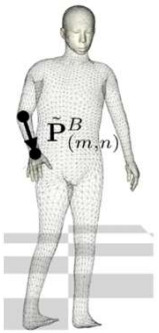
$$f_\theta(\ \ ) = \text{"cat"}$$

Machine Learning for Computer Vision

- Almost all state-of-the-art computer vision methods are data-driven methods
  - Collect **data** about the computer vision problem to solve
  - Use **machine learning** to train a model that solves the actual problem

- More complex example: 3D human pose estimation (simplified)

Training Data:
captured human
3D poses
(„PanOptic Studio“)

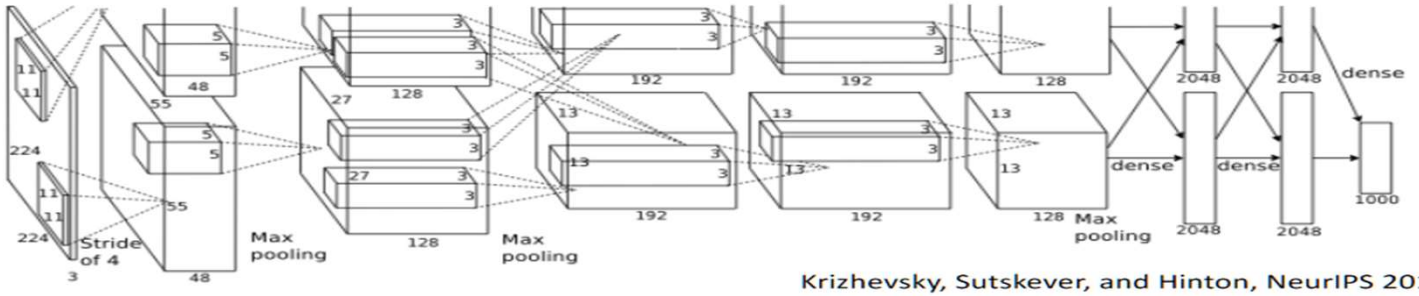


machine learning → model $f_{\boldsymbol{\theta}}$

$$f_{\boldsymbol{\theta}}(\ \ ) = "\ \ "$$

Deep Convolutional Neural Networks

- Most state-of-the-art computer vision is based on a class of machine learning methods called deep convolutional neural networks
  - Large, complex models whose architecture is tailored towards image data
  - Inputs and intermediate representations take the form of 3D tensors that reflect the 2D-arrangement of pixels plus a dimension for color channels or more generally channels of feature activations
  - Models typically have millions of parameters that are estimated on large-scale training data



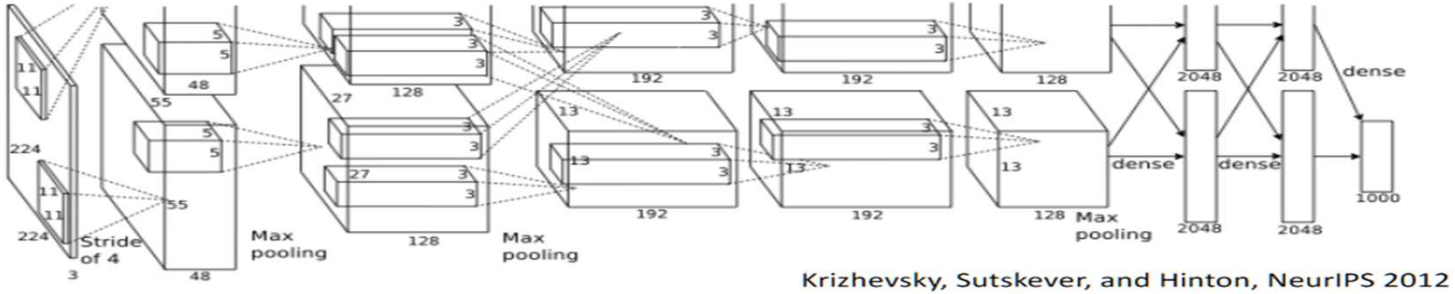
Krizhevsky, Sutskever, and Hinton, NeurIPS 2012

- Recently, Transformer-based architectures have been proposed as a possible alternative...

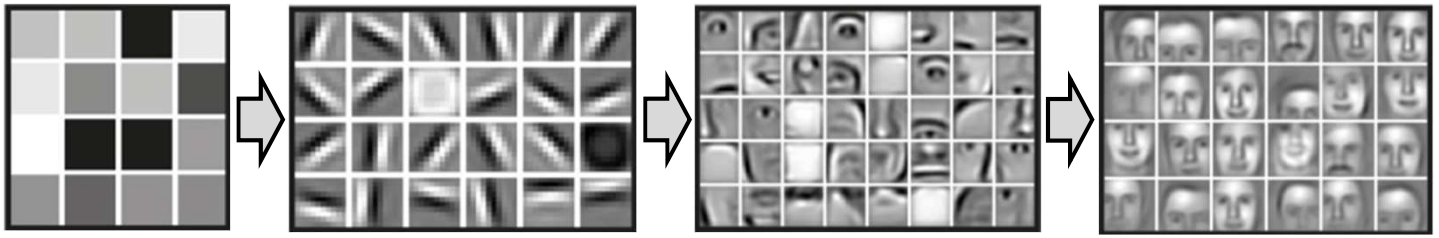
Deep Neural Networks and Feature Learning

- One reason for the effectiveness of deep neural networks in computer vision problems is that they can learn hierarchies of increasingly complex features to go from raw pixels to semantic image content
- Also called „deep learning“ or „representation learning“

Example: face recognition



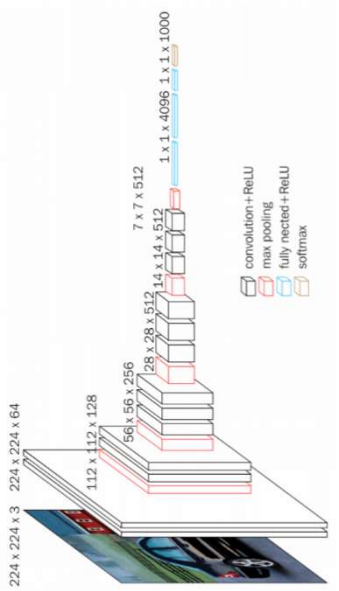
Krizhevsky, Sutskever, and Hinton, NeurIPS 2012



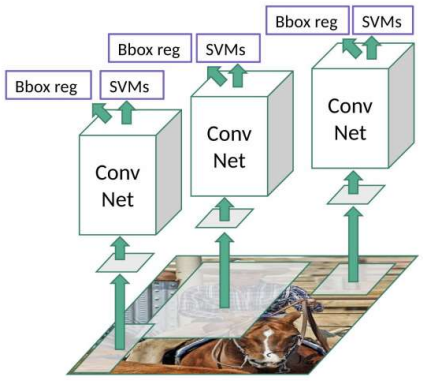
Pixels **Edges, simple patterns** **Face parts** **Faces**

Different Network Architectures for Different Tasks

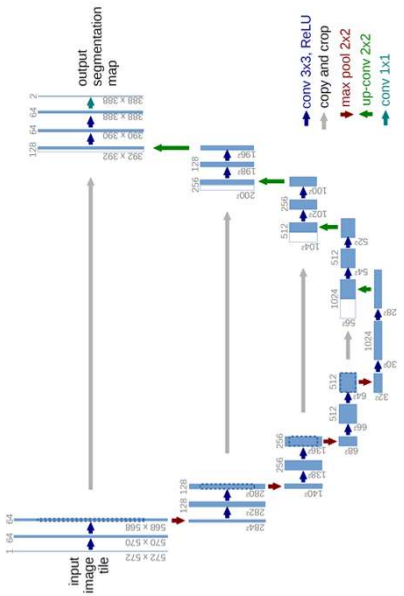
- Many domain-specific and task-specific deep neural network architectures, often build up from common building blocks



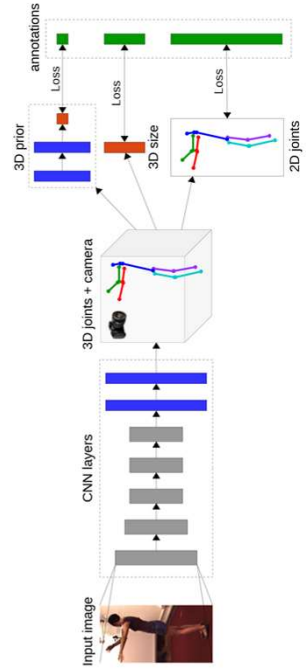
classification



object detection



segmentation



3D-pose estimation

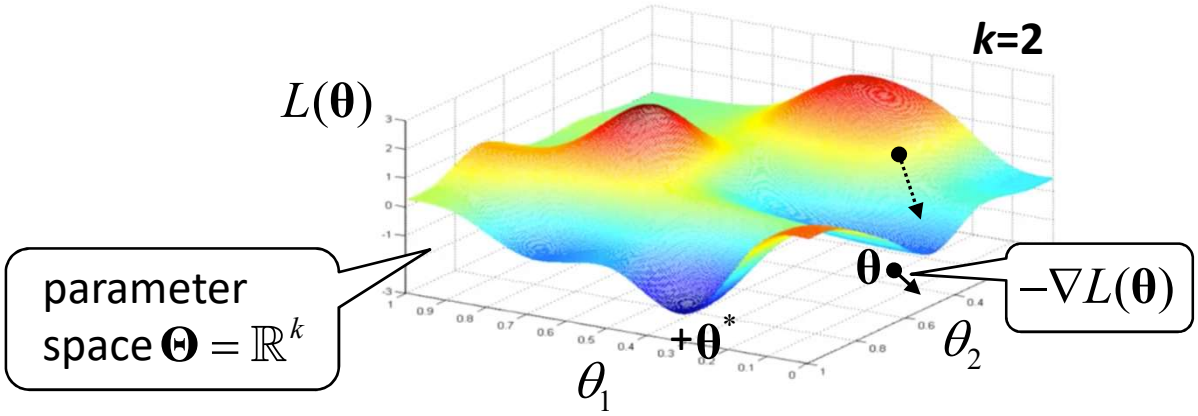
Training Deep Neural Networks

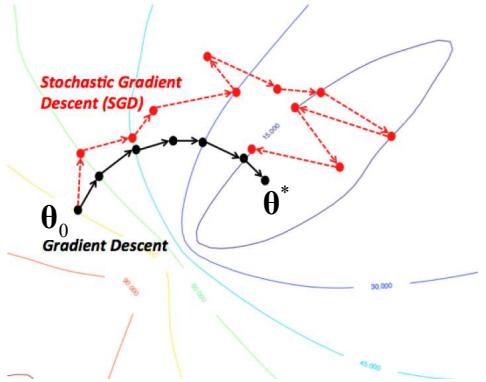
- Training deep neural networks from data: stochastic optimization

$$\text{Optimization: } \boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

$$L(\boldsymbol{\theta}) = \underbrace{\frac{1}{n}\sum_{i=1}^{n} \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i)}_{\text{Loss}} + \underbrace{\lambda R(\boldsymbol{\theta})}_{\text{Regularizer}}$$

Minibatch gradient descent. Obtain $\nabla L(\boldsymbol{\theta})$ by automatic differentiation





Deep Learning: Software and Toolkits

- A major contributor to the success of deep neural networks has been the availability of well-designed and mature software packages to implement, train and deploy deep neural networks

- **Modularity**: quickly build new models from existing building blocks, can quickly reuse and adapt existing models, no need to „reinvent the wheel“
- **Software quality**: well-supported, well-documented toolkits with a vibrant developer community
- **Efficiency**: extremely efficient implementations (e.g. CUDA/CuDNN: low-level primitives for neural networks written and optimized by GPU manufacturers)

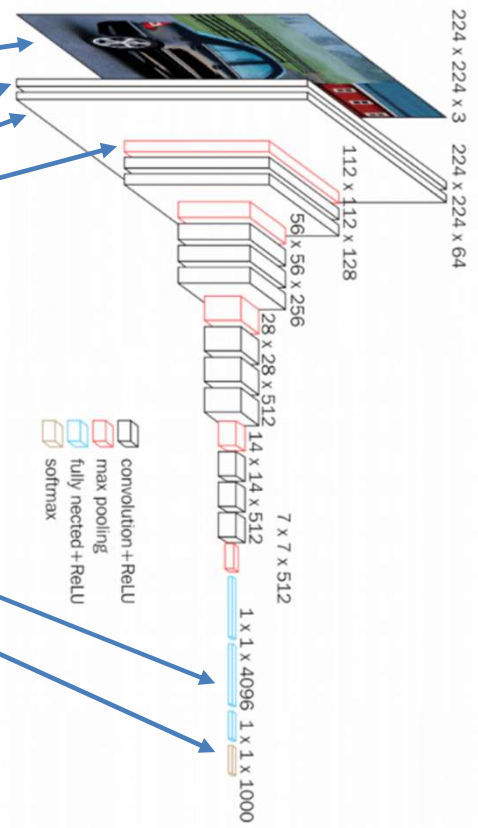
Deep Learning: Software and Toolkits

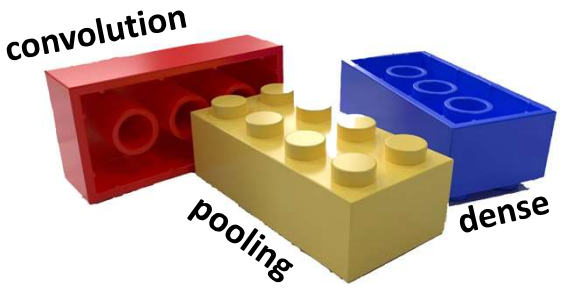
- Using deep learning toolkits, can design and build new models quickly

E.g. TF.Keras: high-level API for DNNs

```
inp = Input(shape=(224,224,3))
z = Conv2D(64, kernel_size=(3,3), activation="relu“,...)(inp)
z = Conv2D(64, kernel_size=(3,3), activation="relu“,...)(z)
z = MaxPooling2D(pool_size=(2,2), strides=(2,2))(z)
...
z = Dense(4096, activation=„relu“)(z)
out = Dense(1000, activation=„relu“)(z)

model = Model(inputs=[inp], outputs=[out])
model.compile(loss=..., optimizer=..., metrics=...)
model.fit(X, y, batch_size=..., num_epochs=...)
```





... as students, you still need to understand what is going on „under the hood“

Deep Learning Hardware

- Recent successes in machine learning in general and deep learning in particular have partially been driven by more data and more compute power
- Training of deep neural networks is computationally expensive: deep neural networks specifically rely on and profit from advances in hardware

- Training on highly parallel hardware: graphic processing units (GPUs) or specialized hardware such as „Tensor Processing Units“ (TPUs)
- Theoretical peak performance much higher than for (equally expensive) CPUs

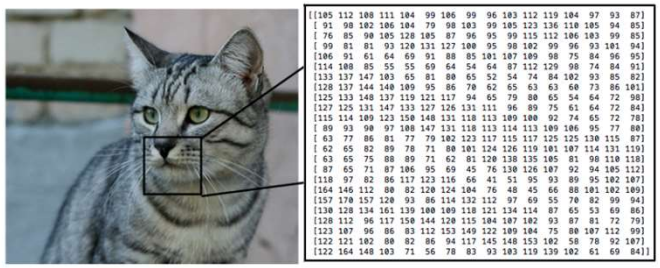
Example: RTX 3090 GPU, theoretical peak performance of tensor cores: 285 TFlop/s. In practice limited by memory bandwidth.

Example: „Emmy“ supercomputer (as of 11/2020 place 47 in ranking of most powerful clusters in the world). Theoretical peak performance: 8780 TFlop/s, approx. 3 TFlop/s per CPU.

Summary

- Computer vision: study of how to make computers perceive, process, and understand visual data such as images and videos
- Difficult problem due to semantic gap:

„A cat!“

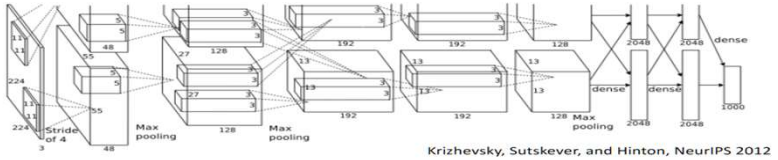


This image by Nikita is licensed under CC-BY 2.0

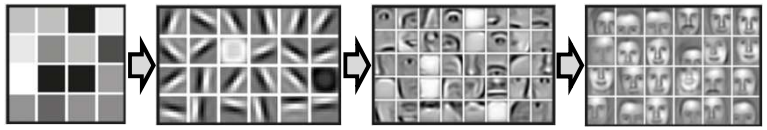
J. Johnson, 2019

Computer only „sees“ big grid of numbers (pixel values)

- State-of-the-art: deep neural networks that learn hierarchies of visual features



Krizhevsky, Sutskever, and Hinton, NeurIPS 2012



Pixels Edges, simple patterns Face parts Faces