Image recognition with machine learning methods - Introduction, challenges and example applications

XLI Heidelberg Physics Graduate Days

Heidelberg, October 12th, 2018

Agenda

- Part I:
  - Image recognition as a business task
  - Recap of machine learning concepts
    - Classical approach
    - Deep Learning: Convolutional Neural Networks
- Part II:
  - Project examples:
    - Machine engineering sector
    - Image classification for the manufacturing industry
  - Demo

d-fine

Image recognition as a business task

Selected applications and projects

Image pixels vs. semantic concepts - How to recognize an object?

The problem at hand

» Can a computer tell me whether this is a bike or a car?
» Can it detect where a particular object is located?
» How can I teach the computer to make that distinction?

The challenge

» Infinitely large number of "manifestations"
» Finding all representations is not feasible with classical programming approaches
» Computers do not generalize concepts! (in general)

The approach

» Construct a simple but flexible system that is not explicitly programmed but works out the commonalities and differences between different image manifestations ("learn the features")
» Train the system with many image examples for which you know the desired outcome
» Exploit the system's flexibility during the training: Have the system adjust its parameters until as much images as possible are classified correctly

A Machine Learning system is NOT "prepared" for all conceivable cases of a problem – Instead it is flexible enough to adapt itself to solve it in an optimal way – This is a complex task and requires LOTS of data!

All images on this slide are taken from pexels.org

Selected real life applications – Driver Assistance Systems

The problem

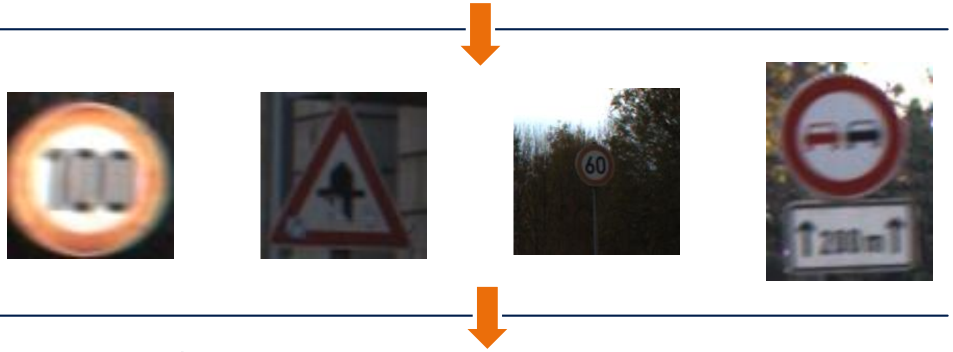
» The DAS must recognize the speed limit of 50 km/h reliably

Challenge

» Handling of many different visual conditions (fog, lighting, occlusions, etc.)

» Leads to: high variability although it is the same sign

» Unknown location of sign: Object Detection

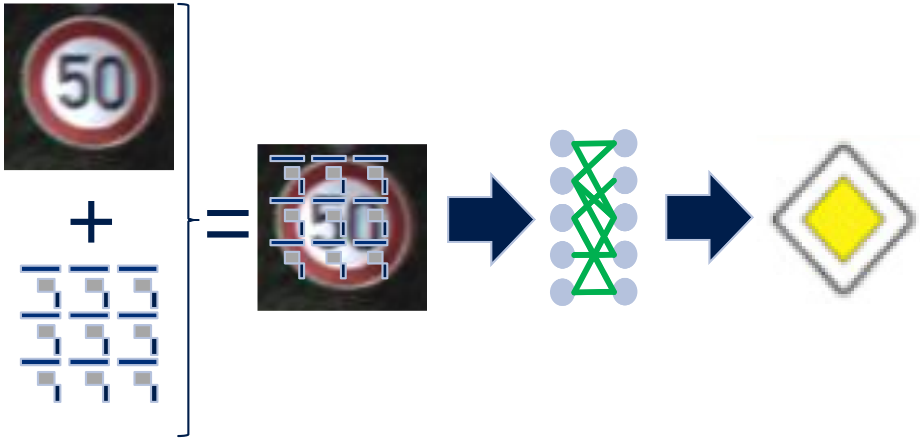


Current research

» Combination with document / text analytics methods (e.g. recognition of limited applicability)

» “Reading” of traffic signs (Interpretation of non StVO-conforming signs)

» Analysis of systematic errors and vulnerabilities for manipulation („Adversarial Perturbations“) („Fooling“ of neural networks with targeted physical manipulation of images [2]).



[1] Image source: The German Traffic Sign Detection Benchmark

[2] Robust Physical-World Attacks on Deep Learning Visual Classification, IEEE Conference on Computer Vision and Pattern Recognition 2018

Real-World examples from our consulting practice
Deep Neural Networks enhance quality control in inspection machines

Image recognition with Convolutional Neural Networks for a large provider of inspection machines

Project Goal

- » **Improved error detection** during visual inspection of medical containers
  - – Evaluation of different DL-architectures (using TensorFlow) and AI benchmarks
  - – Implementation of successful methods

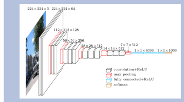
Challenges

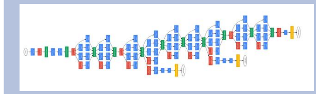
- » Selection and intelligent initialization of the **network architecture**
- » **Avoidance of overfitting** and handling of **asymmetrical data basis** (very few error samples)
- » Integration of highly complex system components into the existing architecture

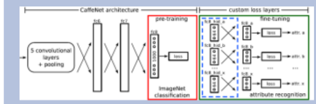
Solution

- » Setup of a highly efficient **training pipeline** for comparison of different DL-methods
- » Application of **data augmentation, Dropout / Batch Normalization**; benchmark against classical ML
- » **Systematic tests** of recognition performance /throughput for compliance with operational requirements

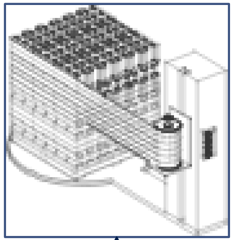
TensorFlow training and test application







- » Implementation of different Deep Network architectures und training methods
- » Training & test of recognition performance
- » Analysis and comparison of approaches
- » ***Deployment of successful networks for usage in the productive application***



0,240223	0,373932	0,272828	0,370835	0,274199	0,270655	0,190182	0,443651
0,305001	0,207674	0,457662	0,07377	0,324191	0,78958	0,667146	0,622
0,491249	0,709508	0,273823	0,129238	0,096895	0,688673	0,359714	0,217979
0,89068	0,213714	0,685433	0,652688	0,716151	0,476408	0,134205	0,695035
0,694536	0,32809	0,831618	0,721933	0,791843	0,466265	0,195058	0,934757
0,858783	0,769162	0,796223	0,185682	0,474272	0,583952	0,593843	0,854703
0,774046	0,693664	0,832288	0,545018	0,102674	0,118859	0,407129	0,742597
0,173326	0,928554	0,701729	0,702977	0,705125	0,76469	0,984143	0,824379
0,677896	0,611145	0,927865	0,282646	0,692481	0,168465	0,651464	0,250175
0,929921	0,066496	0,110062	0,020356	0,368053	0,165102	0,881439	0,918489

Proprietary machine application

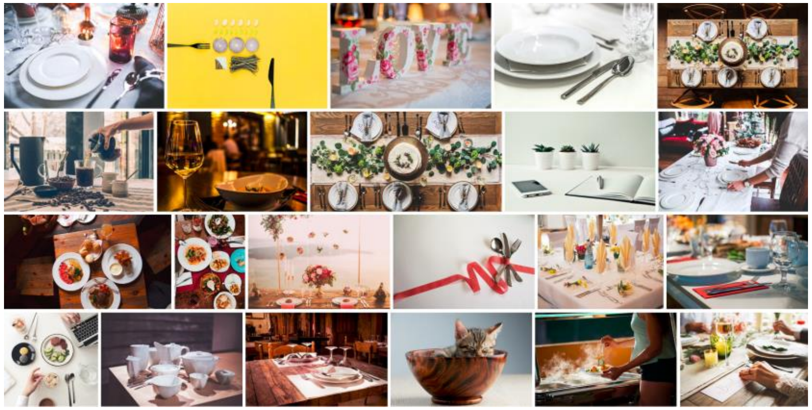
- » Read image data
- » Call DNN-classifier
- » Evaluate model response
- » Return result to machine control

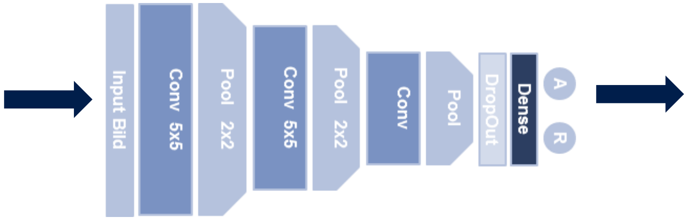
d-fine

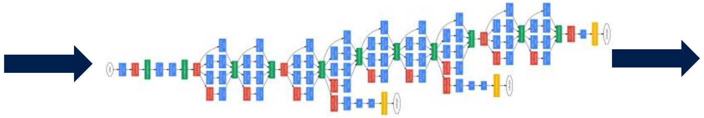
Real-World examples from our consulting practice
Deep Neural Networks enable reliable separation of complex patterns

Product identification with Deep Neural Networks for a large ceramics manufacturer

Project Goal	» Implementation of a Deep-Learning based prototype system for **detection and identification of products** in customer images and in the company‘s image database. » Evaluation of image-recognition for other applications and draft of a roadmap for the implementation
Challenges	» Selection of suitable algorithms and creation of a sufficiently **large training data basis** » Design of methods for **object detection** in complex images and optimization of the recognition » Quantification of added value and effort for the extension of the system to other applications
Solution	» Setup of a **standardized training pipeline** for pre-trained and specifically optimized DNNs » Systematic extension of the data basis; functional upgrades of network architecture and training method » Workshops with business divisions and digital unit for evaluation of the identified use cases » Compilation and reconciliation of the implementation roadmap







All images on this slide are either taken from pexels.org or are the property of d-fine.

d-fine

Recap of the underlying Machine Learning concepts

Terminology, methods and tools: A brief overview

d-fine

Machine Learning – Teaching a system to extract information from data

Typical Fields of Application

» **Industry**
- › Monitor equipment (predictive maintenance)
- › Optimize production processes
- › Analyze factors influencing quality & yield

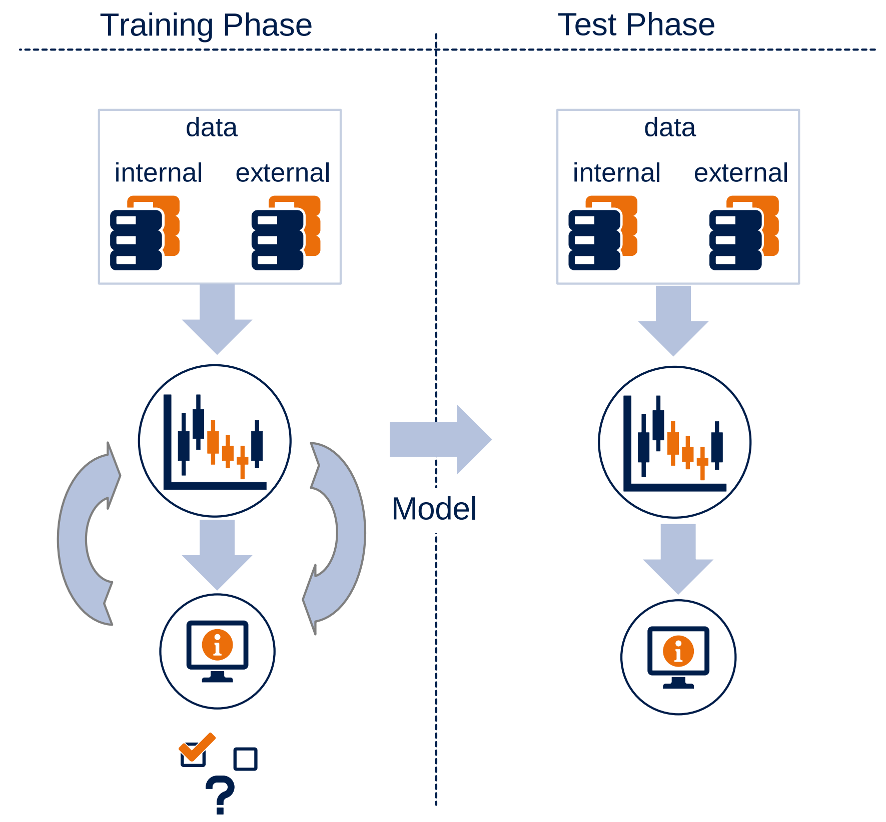
» **Logistics**
- › Real time control and optimization of supply chain elements

» **Marketing**
- › Segmentation and characterization of customer base (potential assessment)

» **Finance**
- › Fraud detection (payments)

» **Healthcare**
- › Oncology diagnosis (IBM / Google)

Machine Learning Approach



Use available data as examples to teach a system to extract information from unknown input data.

Machine Learning – Teaching a system to extract information from data

Typical Fields of Application

» **Industry**
- › Monitor equipment (predictive maintenance)
- › Optimize production processes
- › Analyze factors influencing quality & yield

» **Logistics**
- › Real time control and optimization of supply chain elements

» **Marketing**
- › Segmentation and characterization of customer base (potential assessment)

» **Finance**
- › Fraud detection (payments)

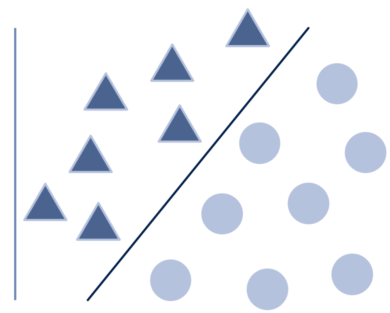
» **Healthcare**
- › Oncology diagnosis (IBM / Google)

Machine Learning Methods

Supervised Learning

» **Typical Tasks**
- › Classification & Regression

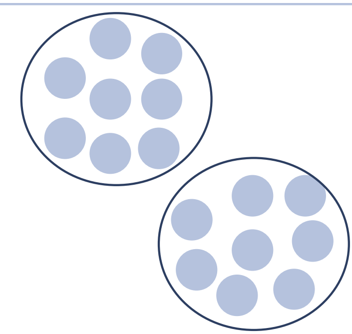
» **Typical Algorithms**
- › Decision Trees
- › Random Forests
- › Gradient boosted Trees
- › **Deep Neural Networks**



Unsupervised Learning

» **Typical Tasks**
- › Cluster-Analysis
- › Dimensional reduction
- › Anomaly-Detection

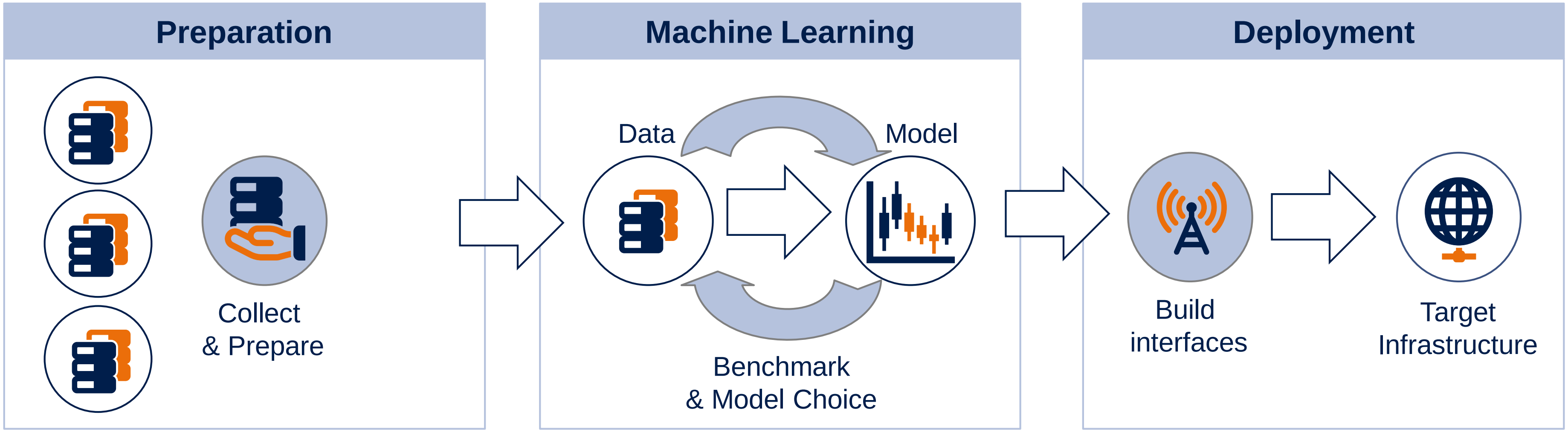
» **Typical Algorithms**
- › kMeans, DBSCAN
- › PCA, t-SNA
- › IsolationForests



There is a plethora of tools and applications to query your data and get the most out of them - Choosing the right tool for the specific problem at hand requires a lot of experience!

d-fine

Training a model is just one of three major building blocks, preparation and deployment are often neglected, but time-consuming and crucial for success.



Preparation

» **Understand your problem**
› Bring together business- and ML-experts to describe added value

» **Obtain access to data**
› Convince all stakeholders, encourage open discussion

» **Clean and validate data**
› Understand semantics and quality, remove artefacts

Machine Learning

» **Adopt an iterative approach**
› Increase complexity step-by-step, from simple to complex models

» **Always evaluate quantitatively**
› Non-trivial to get right, random independent splits

» **Discuss intermediate results**
› let others challenge the outcome, be wary of too good results

Deployment

» **Interfaces to existing systems**
› Robust interfaces to systems and processes, real-time requirement

» **Support in stabilization phase**
› Model deployment process, continuous data delivery

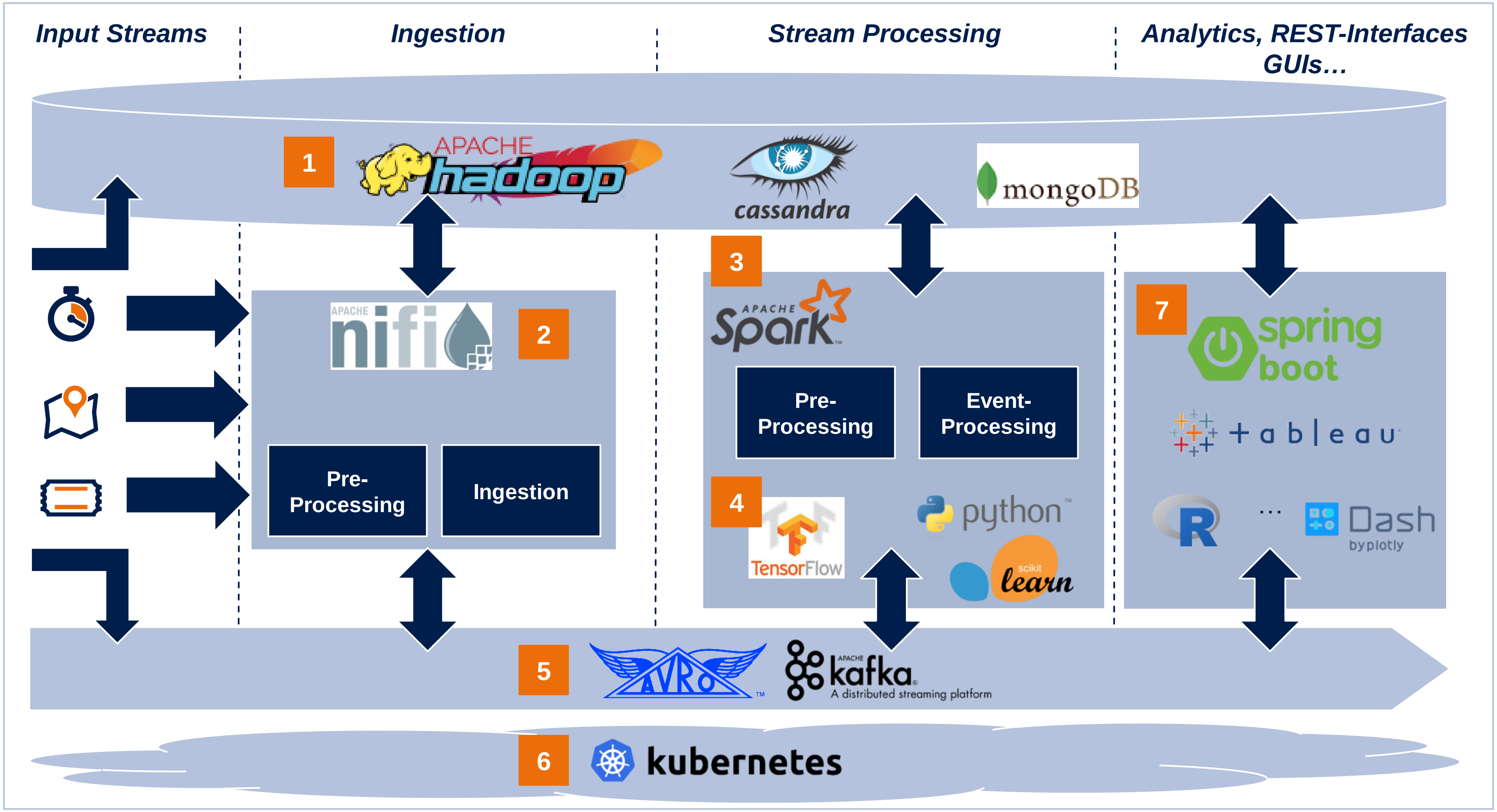
» **Include the users**
› Workshops and coordinated handover

A successful ML-project builds upon bringing together the *domain knowledge* of the involved business divisions and the specific expert know-how on *data science and machine learning*.

d-fine

Popular standard components enable re-use of data across projects, facilitating preparation and deployment as well as speeding up release cycles.

Schematic Architecture of a fully fledged analytics pipeline – Not all components are always required !



d-fine

Classical Machine Learning vs. Deep Neural Networks

A comparison for image recognition applications

Image recognition is an excellent example for a machine learning problem that benefits highly from end-to-end learning approaches

Fundamental Challenge:

» The large number of pixels in an image mandates some form of intermediate representation.

Basic solution approaches:

Feature representation

» The image is represented by manually defined feature encoding

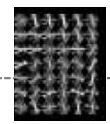
» The recognition model does only take into account the feature representation

Object type ?

Input

1. Feature encoding

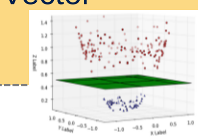
Example: **HOG** (Histogram of oriented Gradients)



Features $\begin{pmatrix} 1 \\ \vdots \\ d \end{pmatrix}$

2. Learning

Example: **SVM** (Support Vector Machine)



Car!

Output

End-to-end learning

» Features are part of the learning process, i.e. are determined by the model

» Control is limited by design; user intervention e.g. via choice of input data

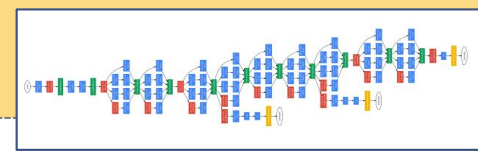
Object type ?

Input

Learning (end-to-end)

Example: *CNN (Convolutional Neural Network)*

Inception v3



Bike!

Output

Approaches based on feature representation require human effort in form of feature encoding while end-to-end models are designed to work completely autonomously.

d-fine

Image Recognition with Classical Machine Learning

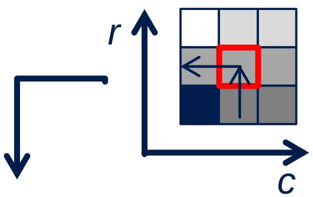
Feature Extraction and Classification

An exemplified classical image recognition pipeline – part I
Feature extraction with histograms of oriented gradients (HOGs)

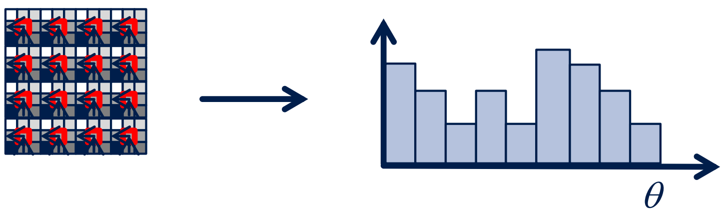
» The idea underlying the HOG-transformation is to characterize an image by the ***local distribution*** of intensity gradients without knowing the whole "gradient vector field".

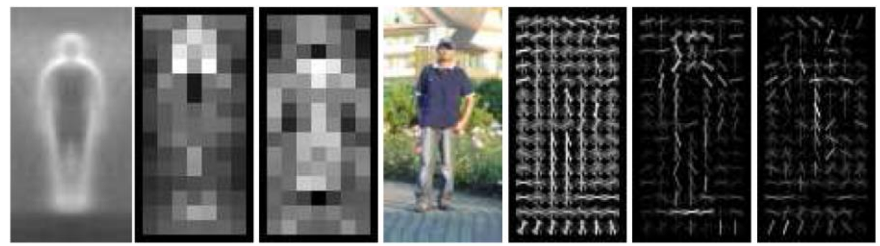
» Gradients are measured for all pixels in a given image region (a cell) and "bucketed" to a polar coordinate histogram(1):

$$I_x(r,c) = I(r,c+1) - I(r,c-1) \quad \text{and} \quad I_y(r,c) = I(r-1,c) - I(r+1,c)\ .$$



$$\mu = \sqrt{I_x^2 + I_y^2} \quad \text{and} \quad \theta = \frac{180}{\pi}\left(\tan_2^{-1}(I_y, I_x) \mod \pi\right)$$





» 2x2 cells is combined into blocks, whereby each pixel contributes according to its position within the block. A normalization on block level enhances contrast invariance.(1)

» The histograms accumulated in this way provide a (much) lower dimensional representation of the image and can be used in combination with classical machine learning approaches

Manually extracted feature vectors provide a low dimensional abstraction of the image.

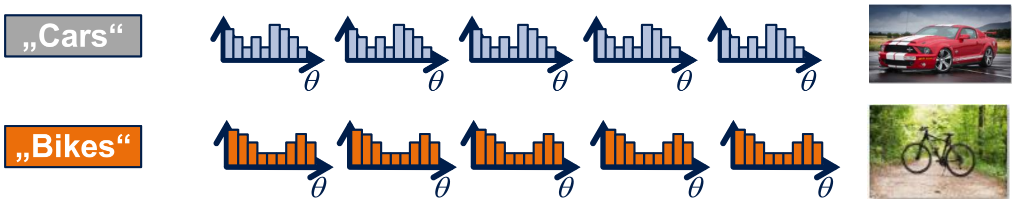
(1) For details on HOG transformation see: Dalal, Triggs IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, pages 886–893, June 2005

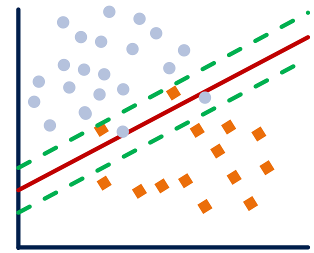
d-fine

An exemplified classical image recognition pipeline – part II
Separating bikes from cars with a Support Vector Machine

» A (linear) SVM in its simplest form is used to try to (linearly) separate two classes of objects.

» Assume you’re given two class labels $y \in \{-1,1\}$ equivalent to {“car”, “bike”} and a set of labelled feature vectors $\{(\mathrm{y}_1, \vec{x_1}), \ldots, (\mathrm{y}_\mathrm{n}, \vec{x_n})\}$ telling to which class each $\vec{x_i}$ belongs.





» Task: Find an n-1-dimensional (hyper-) plane described by the vector $\vec{w}$ that separates the classes minimizing the following loss function (called Hinge loss):

$\vec{w} \times \vec{x_i} - b \geq 0$ for „cars“

$\vec{w} \times \vec{x_i} - b < 0$ for „bikes“

$$argmin_w \left[\frac{1}{n}\sum_{i=1}^{n} \max(0, 1 - y_i \times (\vec{w} \times \vec{x_i} - b)\right] + \lambda \times \|\vec{w}\|^2$$

» This optimization problem is convex in $\vec{w}, b$ and thus has a unique solution, that can be found by quadratic programing approaches or directly in the primal formulation with sub-gradients .

The support vector machine tries to separate the low dimensional representation of the images.
It can be generalized to more than two classes and augmented to nonlinear separable features.

(1) For more on SVMs for separation of non-linear separable data and multiple classes see e.g. (Machine Learning - A Probabilistic Perspective, Kevin P. Murphy, MIT Press 2012)

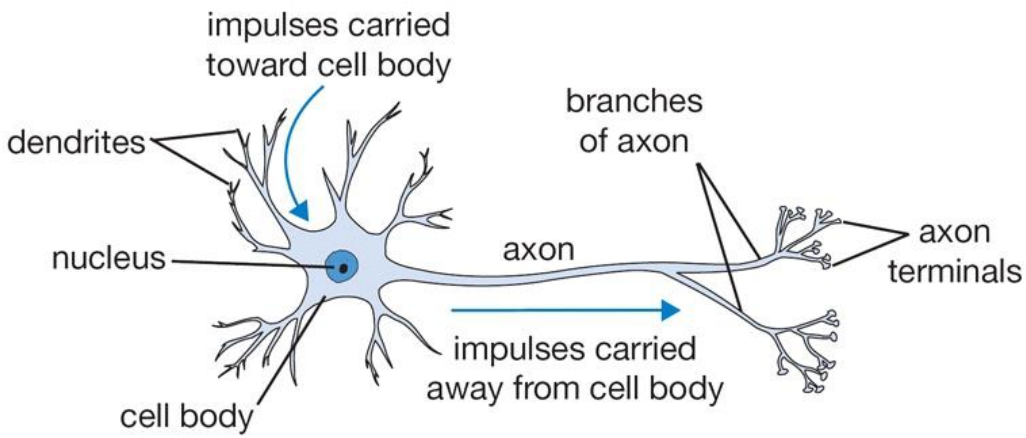
d-fine

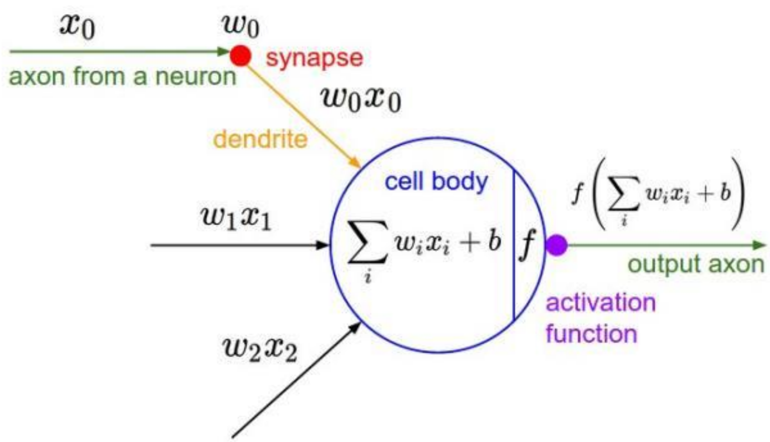
Image Recognition with (Deep) Neural Networks

Feature Extraction and Feature Learning

(Deep) neural networks

» Neural networks can be understood as a mathematical model of neurons in brains[1] (all we know this model is to simplistic to account for the processes in a real brain)





» Neurons “fire” electrical impulses along so-called axons to other neurons, thereby producing a dense and complicated web of interacting units

» In the mathematical model of a neuron the signal x from another neuron undergoes first an affine transformation (the parameters of those are called weights), which is the input to a non-linear function called an activation function

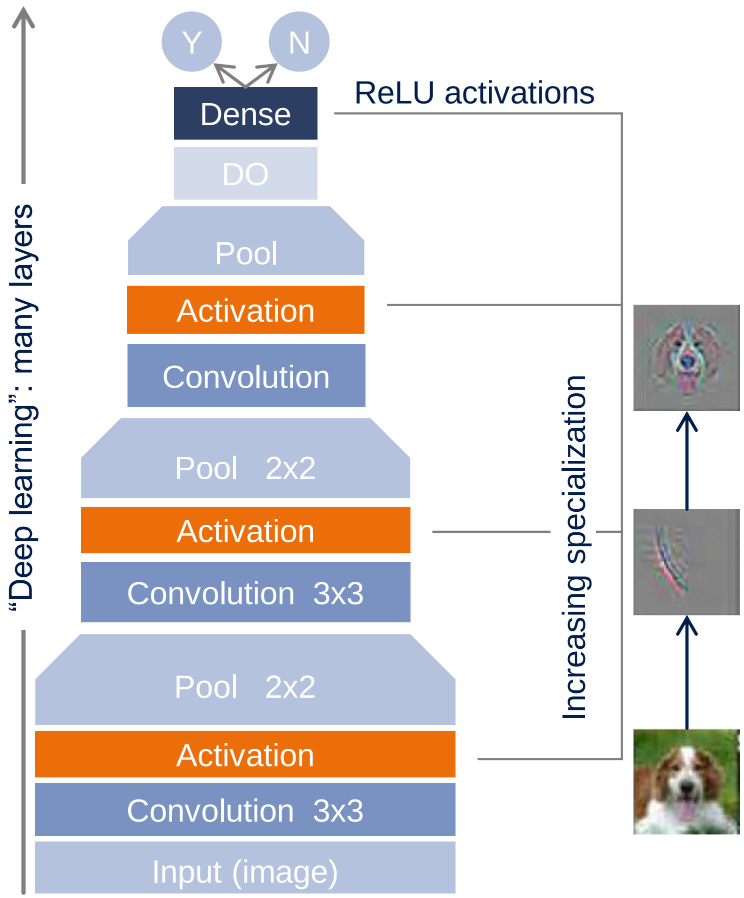
» By building a network of such mathematical neurons one obtains a so-called neural network

[1]See e.g. Bengio, Yoshua, et al. "Towards biologically plausible deep learning." *arXiv preprint arXiv:1502.04156* (2015 for a discussion.

d-fine

Convolutional Neural Networks are highly effective at end-to-end learning for image recognition tasks and have dramatically improved results

Structure



Key building blocks

Convolution

Groups of linear filters (defined via parameters w)

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Example: edge detection

Other common sizes: 5x5, 7x7

Non-linear activation

Operation on filter outputs (separation of convolutions)

Pooling

Area reduction (avg, max pool)

Strengths

- » Ideal for highly complex recognition tasks (photos)
- » CNNs replace manually calibrated representations
- » Unbeatable edge for highly complex tasks

Challenges

- » High effort for training, criticality of start parameters
- » Specific problems (overfitting, adversarial examples, etc.)
- » Non convex optimization problem / local extrema

CNNs represent the most powerful methodical route for image recognition known today, however pose serious challenges that need to be addressed carefully.

Szegedy et al. Going Deeper with Convolutions arXiv:1409.4842

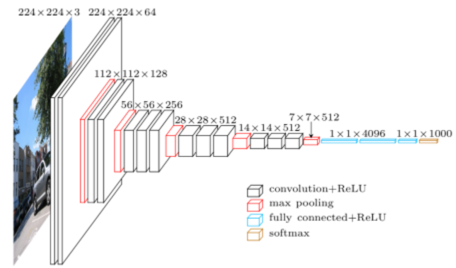
Convolutional Neural Networks form a hierarchical representation, which makes them extremely powerful in tasks that fully leverage this potential.

Design criteria for the network architecture

» Increase network depth (stepwise)

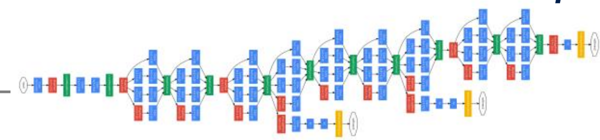
» Restrict number of parameters in a smart way (tricky task – see e.g. *Inception* Architecture)

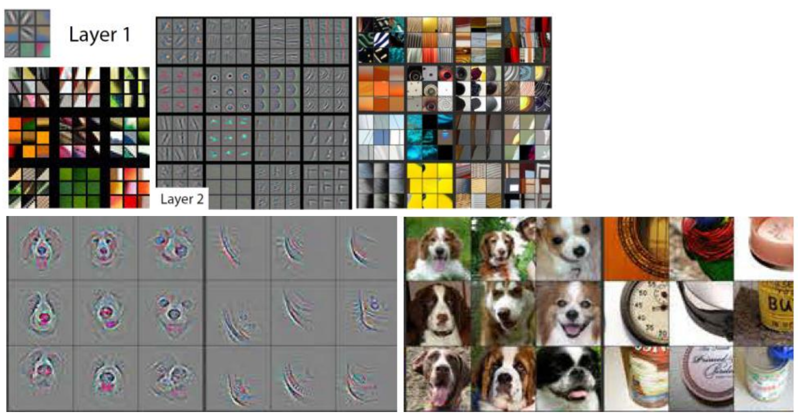
» Make optimal use of computing resources (GPU RAM)



VGG16

Inception v3





Hierarchical Representation

» With increasing *layer depth* more complex structures composed of simpler ones are represented

» The *activations* of the *neurons* become more and more specific to a particular *pattern*

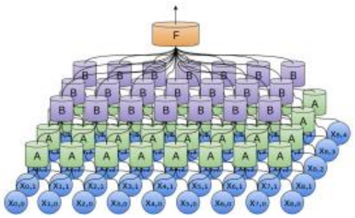
Few basic building blocks are combined into a complex model, resulting in a very powerful, hierarchical representation, which would be infeasible to define manually.

Image sources: (1) https://blog.heuritech.com/2016/02/29/a-brief-report-of-the-heuritech-deep-learning-meetup-5/ (2) Szegedy et al. Going Deeper with Convolutions arXiv:1409.4842
(3) Visualizing and Understanding Convolutional Networks, Matthew D Zeiler, Rob Fergus (https://arxiv.org/abs/1311.2901)

d-fine

There are several DNN architectures for (un-)supervised learning tasks

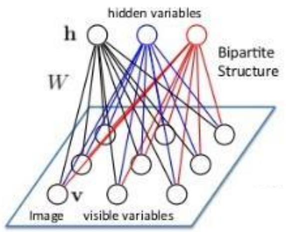
Convolutional (CNN)



Mainly used for static data such as images

- The static length input features are “convoluted”
- Each layer learns a more abstract representation of the data
- Equivalent to renormalization group flow in physics

Boltzmann machines (and variants)

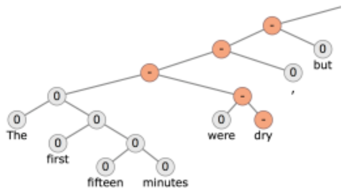


Mainly used for static data

- The neural net consists of a visible and hidden part
- The hidden part can learn arbitrary complex representations of the inputs
- Comparable to HMMs

Architectures can be 100s of layers deep!

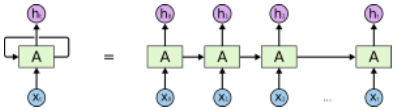
Recursive



Mainly used for hierarchical and tree-like data such as language

- The “input features” come in a natural hierarchy or tree-like structure
- The neural net is applied all along the hierarchy

Recurrent (RNN)



Mainly used for sequential data such as language and time series

- The “input features” are sequential and of different length
- The neural net is recurring
- The neural net can learn long term dependencies (c.f. LSTM, GRUs)

d-fine

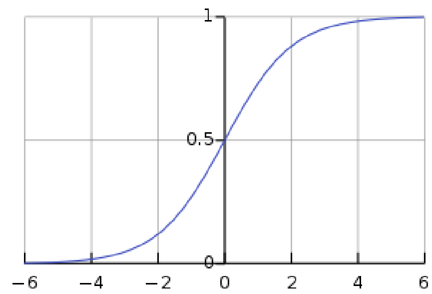
Deep neural networks face various challenges

» Vanishing gradient problem[1]

- Neural networks are usually trained by various incarnations of gradient descent, e.g.

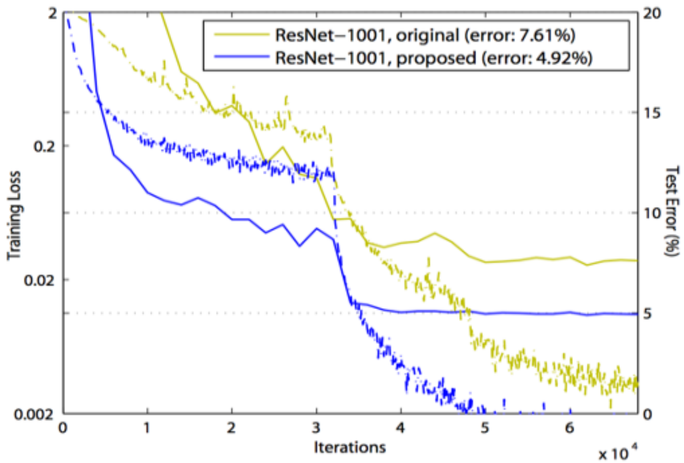
$$w_{i+1} = w_i - \gamma \cdot \left.\frac{\partial J(w)}{\partial w}\right|_{w_i}$$

- By the chain rule this leads to products of activation functions
- As most activation functions take values in $[-1, 1]$ these products become very small for deep networks



» Slow training

- To optimize a non-convex function with millions of terms and millions of variables is computationally very expensive
- Without special hardware the training of deep neural nets is not feasible



» Overfitting

- Deep neural networks typically have millions of free parameters
- Without care this leads to the overfitting phenomenon

» Lots of (labelled) data is needed for training

- Without expert knowledge, which could either be built into the topology of the net or into constraints on the weights and/or the objective function, lots of labelled data is needed to bring deep neural nets into a regime of good behaviour with respect to generalization

[1]Hochreiter, Sepp. "Untersuchungen zu dynamischen neuronalen Netzen."*Diploma, Technische Universität München* (1991): 91.

Summary & Take Home Messages

» Image recognition is a long standing problem, tackled in various ways in the past
  - Approaches using separate feature engineering & ML-tools perform reasonably when the problem is sufficiently simple
  - Deep Neural Networks are fundamentally different from classical ML - They not only “identify similar things” but - by finding their own features - learn “what makes things similar”
  - Deep Learning is a real game changer outperforming ANY classical methods also in complex settings!

» Deep Learning’s recent emergence as a useful concept is due to three factors
  - Abundance of available data (and in particular labelled data)
  - The enormous computing power of modern processor architectures (concentrated in particular on GPUs)
  - The development of new algorithms

» Deep Learning is not the “one size fits all” magic wand for image recognition
  - *There are ways to systematically confuse DNNs via intentional image manipulation (See e.g. (1))*
  - *DL can lead to misconception of your method’s recognition power if it is applied without proper understanding of your data and a without careful choice of the right method for the problem at hand.*
  - *Object detection, the exploitation of (textual) context and aspects like “one-shot-learning” in recognition tasks are active research areas where the human brain is still by far superior to the computer.*

(1) Robust Physical-World Attacks on Deep Learning Visual Classification, IEEE Conference on Computer Vision and Pattern Recognition 2018

d-fine

Contact

Dr Patrick Sudowe
Senior Consultant
Tel +49 89 7908617-0
Mobile +49 152 57975167
E-Mail Patrick.Sudowe@d-fine.de

Dr Alexander Arzhanov
Consultant
Tel +49 69 90737-0
Mobile +49 152 57975316
E-Mail Alexander.Arzhanov@d-fine.de

Dr Tassilo Christ
Senior Manager
Tel +49 89 7908617-111
Mobile +49 162 2630118
E-Mail Tassilo.Christ@d-fine.de

d-fine

Frankfurt
Munich
London
Vienna
Zurich

Headquarters

d-fine GmbH
An der Hauptwache 7
D-60313 Frankfurt/Main
Germany

Tel +49 69 90737-0
Fax +49 69 90737-200

www.d-fine.com

d-fine

