Computing Methods for Experimental Physics and Data Analysis

Data Analysis in Medical Physics

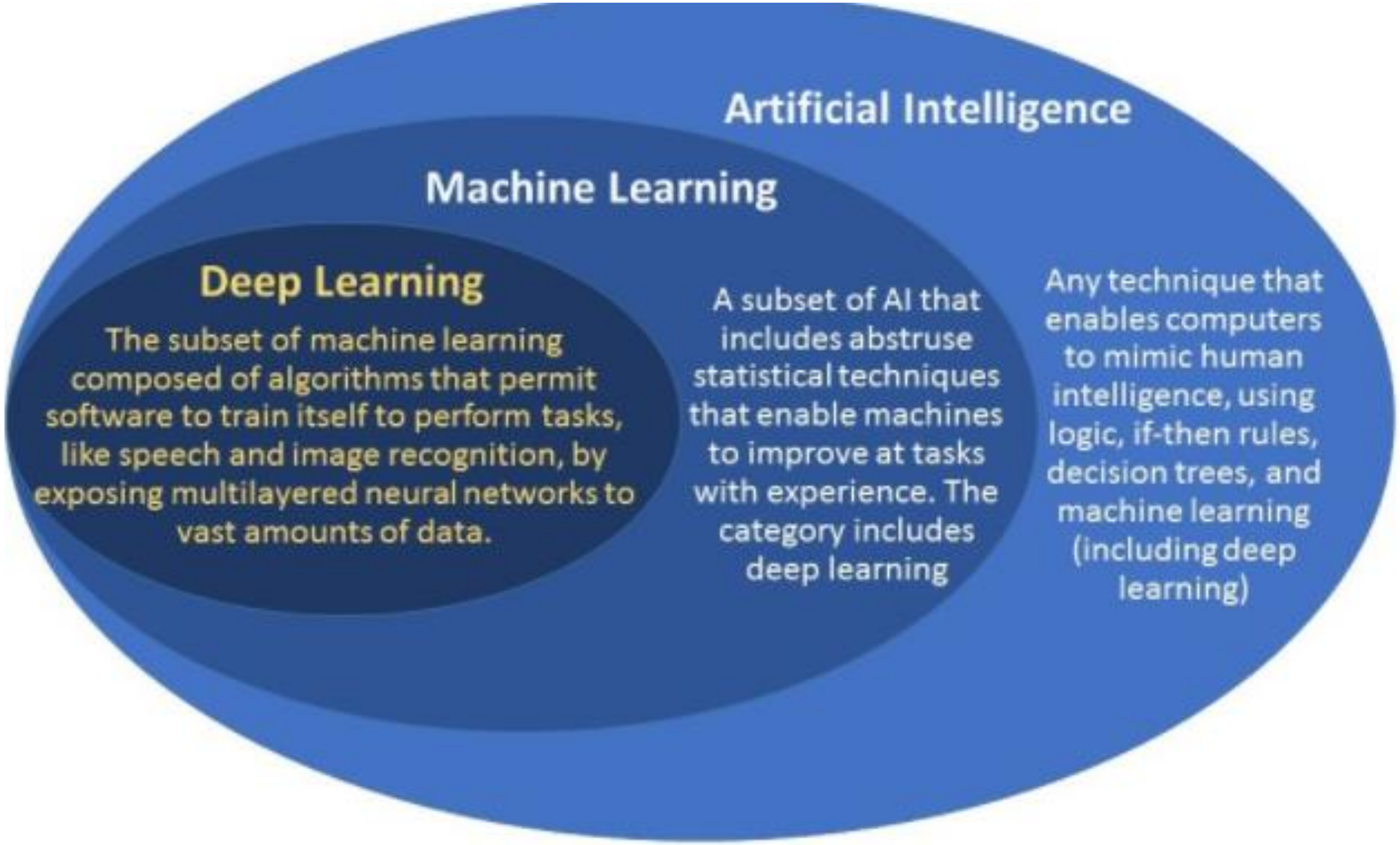
Lecture 8: DL applications on medical images: CNN for image categorization; CAE for image segmentation, and course summary

Alessandra Retico

alessandra.retico@pi.infn.it

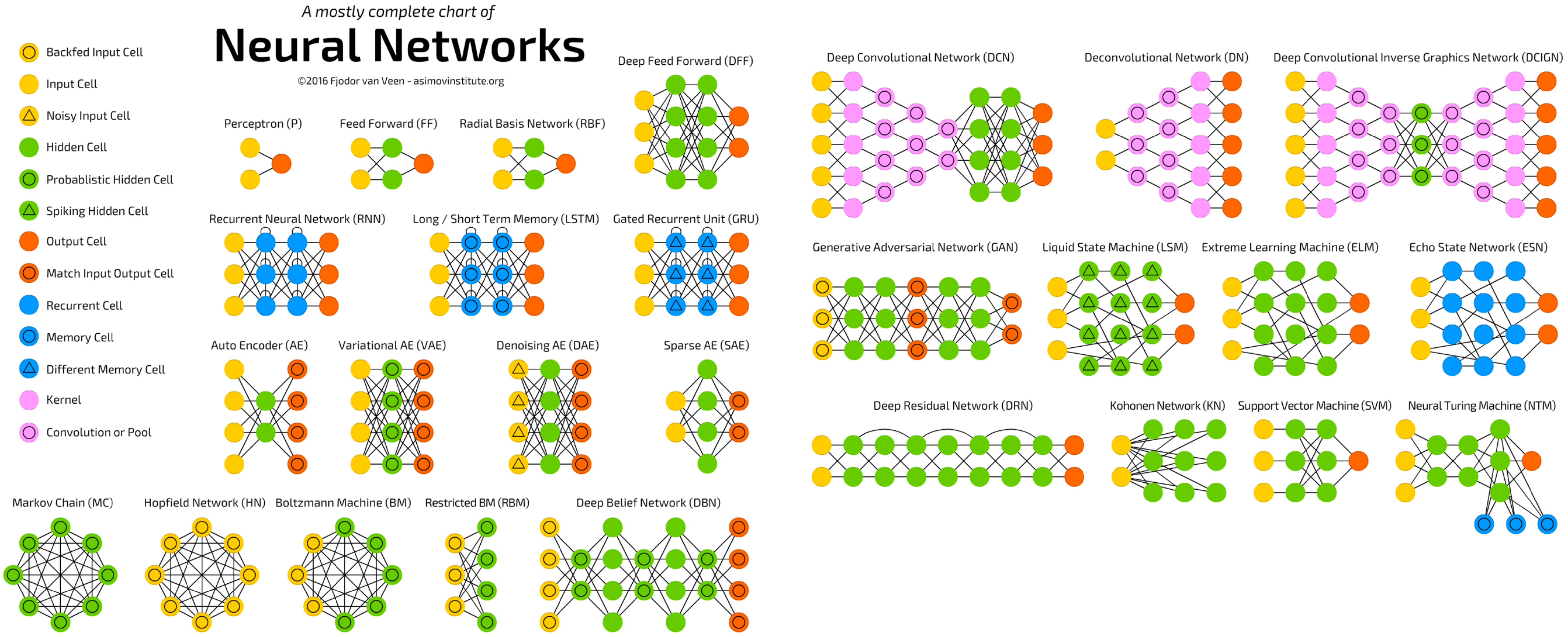
INFN - Pisa

Artificial Intelligence, Machine Learning, Deep Learning



https://www.deeplearningitalia.com/una-panoramica-introduttiva-su-deep-learning-e-machine-learning/

Neural network Zoo

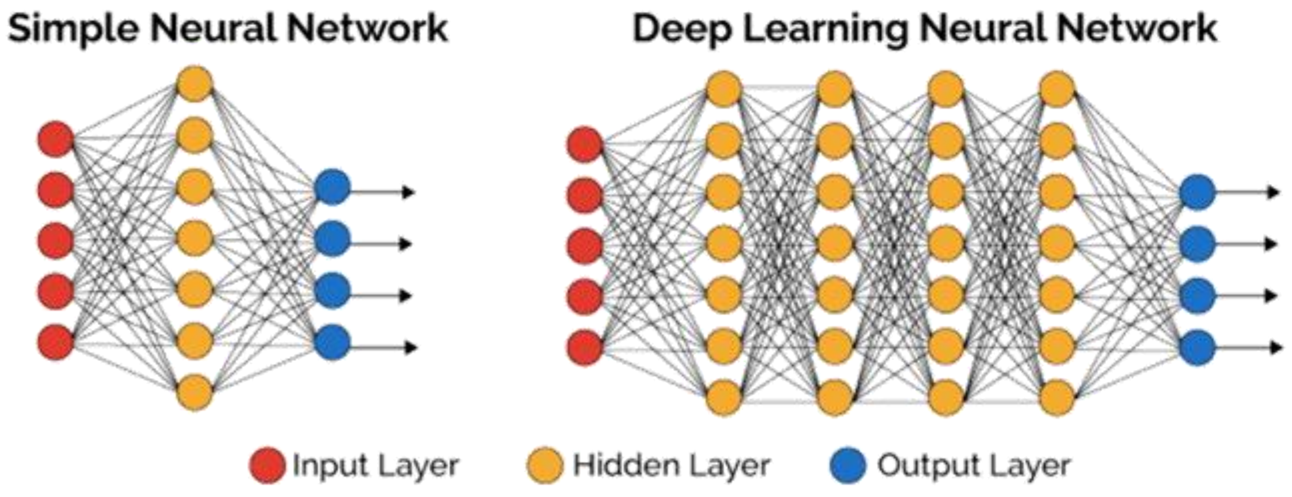


http://www.asimovinstitute.org/neural-network-zoo/

Deep Neural Networks

Deep Learning (DL) means using a neural network with several layers of nodes between input and output

DL models are a family of parametric models which learn non-linear hierarchical representations



$$a_L(\mathbf{x};\boldsymbol{\Theta}) = h_L(h_{L-1}(\ldots(h_1(\mathbf{x},\boldsymbol{\theta}_1),\boldsymbol{\theta}_{L-1}),\boldsymbol{\theta}_L)$$

input

parameters of the network

non-linear activation function

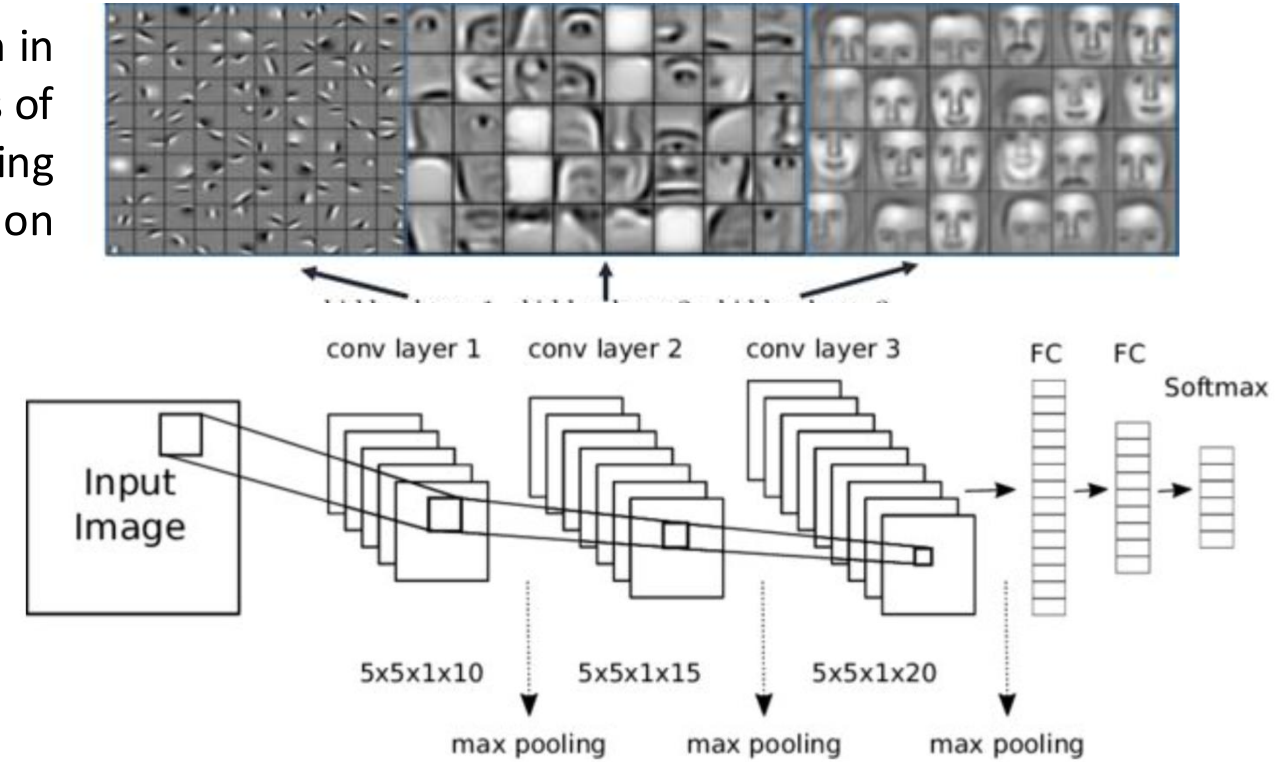
parameters of layer *L*

Deep Neural Networks

- Deep neural networks are generally better than other ML methods on images
- The series of layers between input and output compute relevant features automatically in a series of stages, just as our brains seem to do.

Deep neural networks learn in multiple levels of representation, corresponding to different levels of abstraction

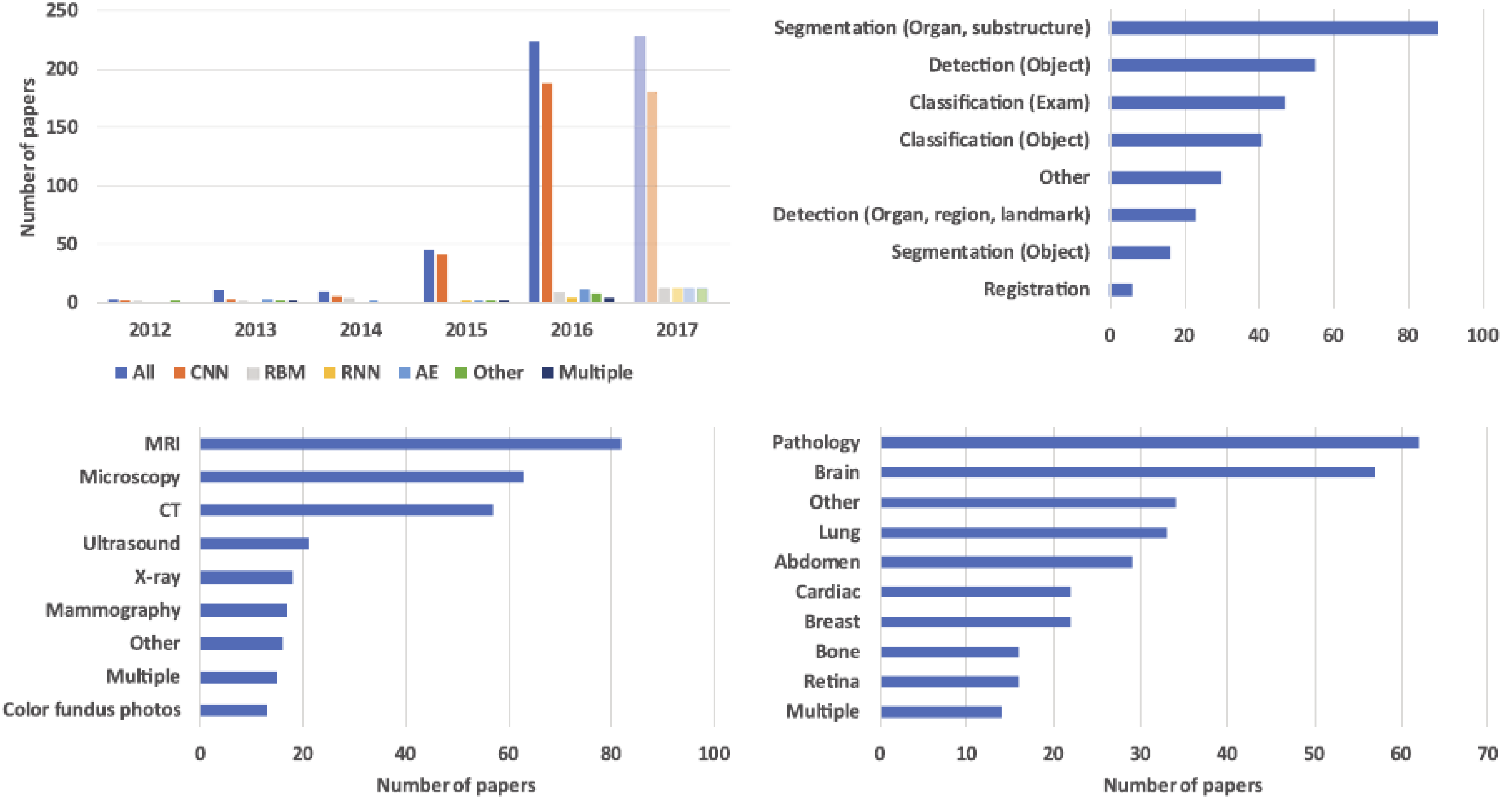
https://towardsdatascience.com



See demo code on L8_code/demo_train_CNN.m, demo_visualizing_CNN_layers.mlx

Deep Learning has become very popular in Medical Imaging

G. Litjens et al., A survey on in deep learning medical image analysis, Medical Image Analysis 42 (2017) 60–88



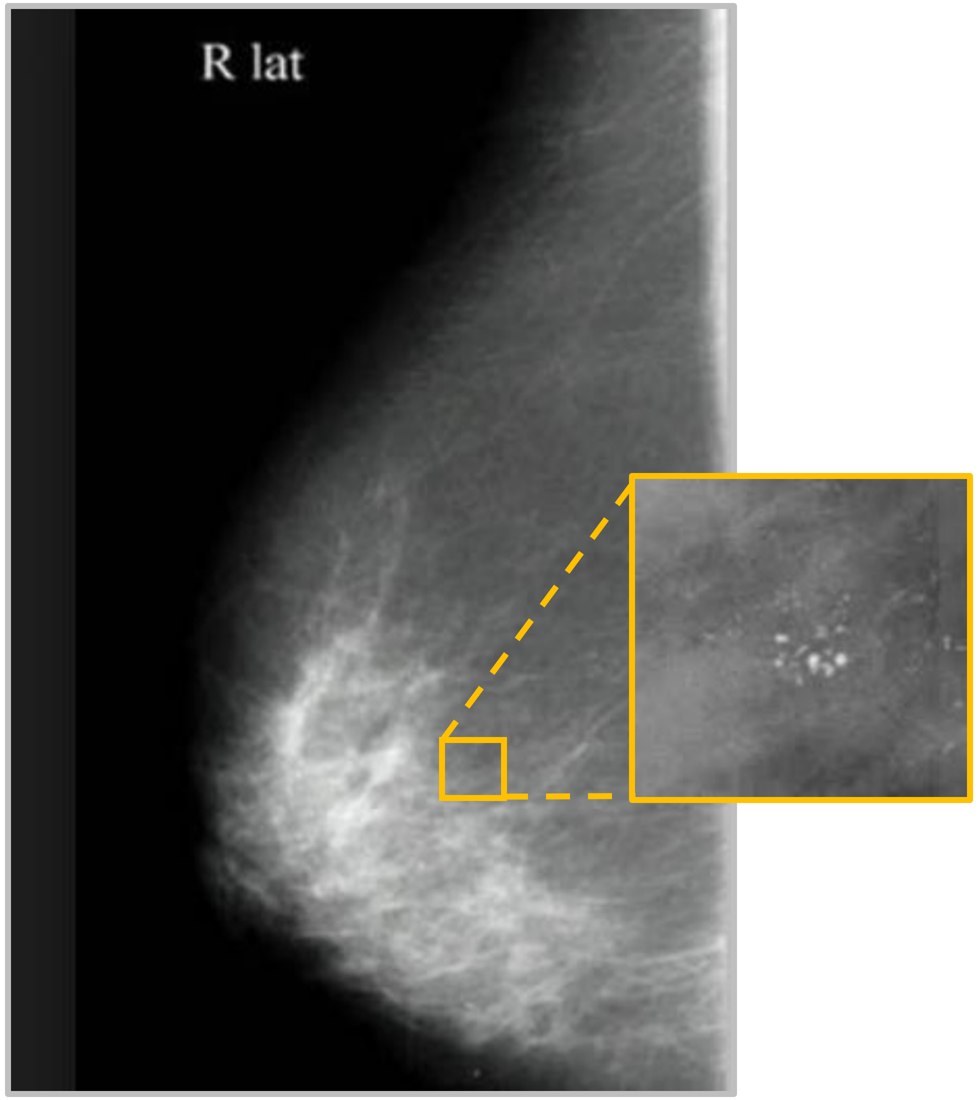
Breakdown of scientific papers by publication year, task addressed, imaging modality, and application area

CNN for image classification

CNN can be used for direct image classification:

- We can distinguish image portions of in mammograms containing either microcalcification clusters or normal tissue
- We can use a trained CNN to identify and localize regions suspected of containing a microcalcification clusters
- Thus we can build a:

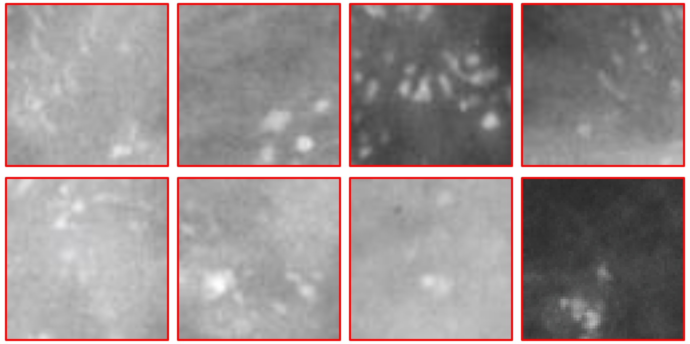
➔ Decision Support System (DSS) for microcalcification detection



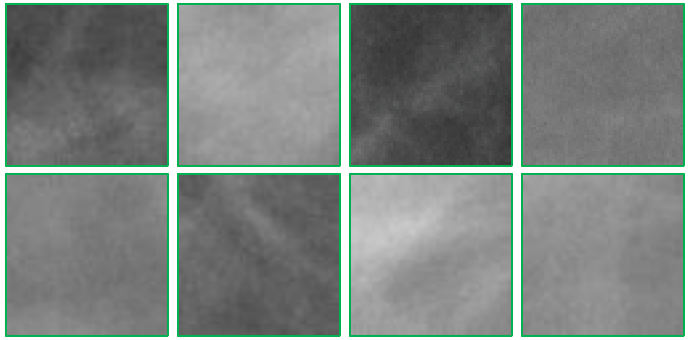
CNN for image classification

Input data

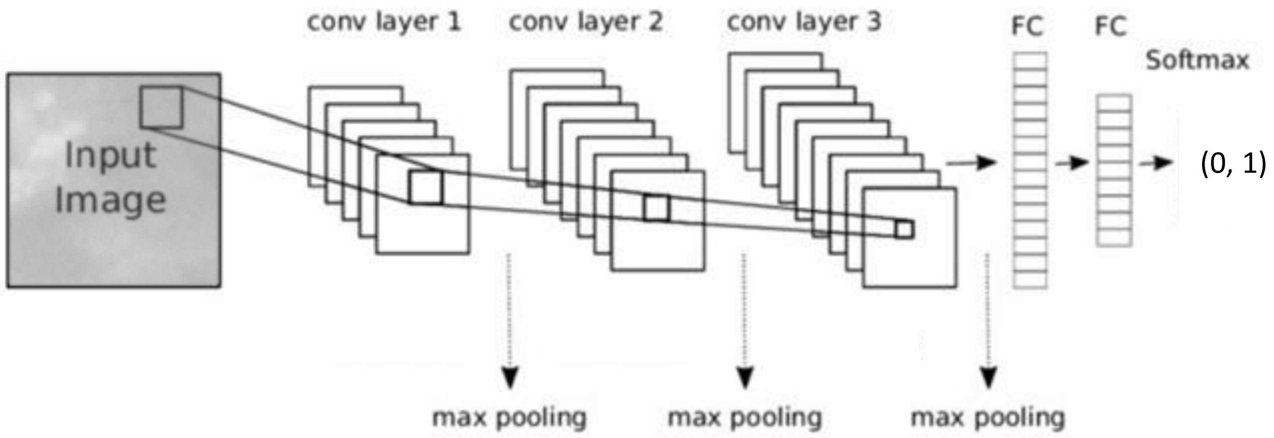
Label 1: images with microcalcifications



Label 0: images of normal tissue



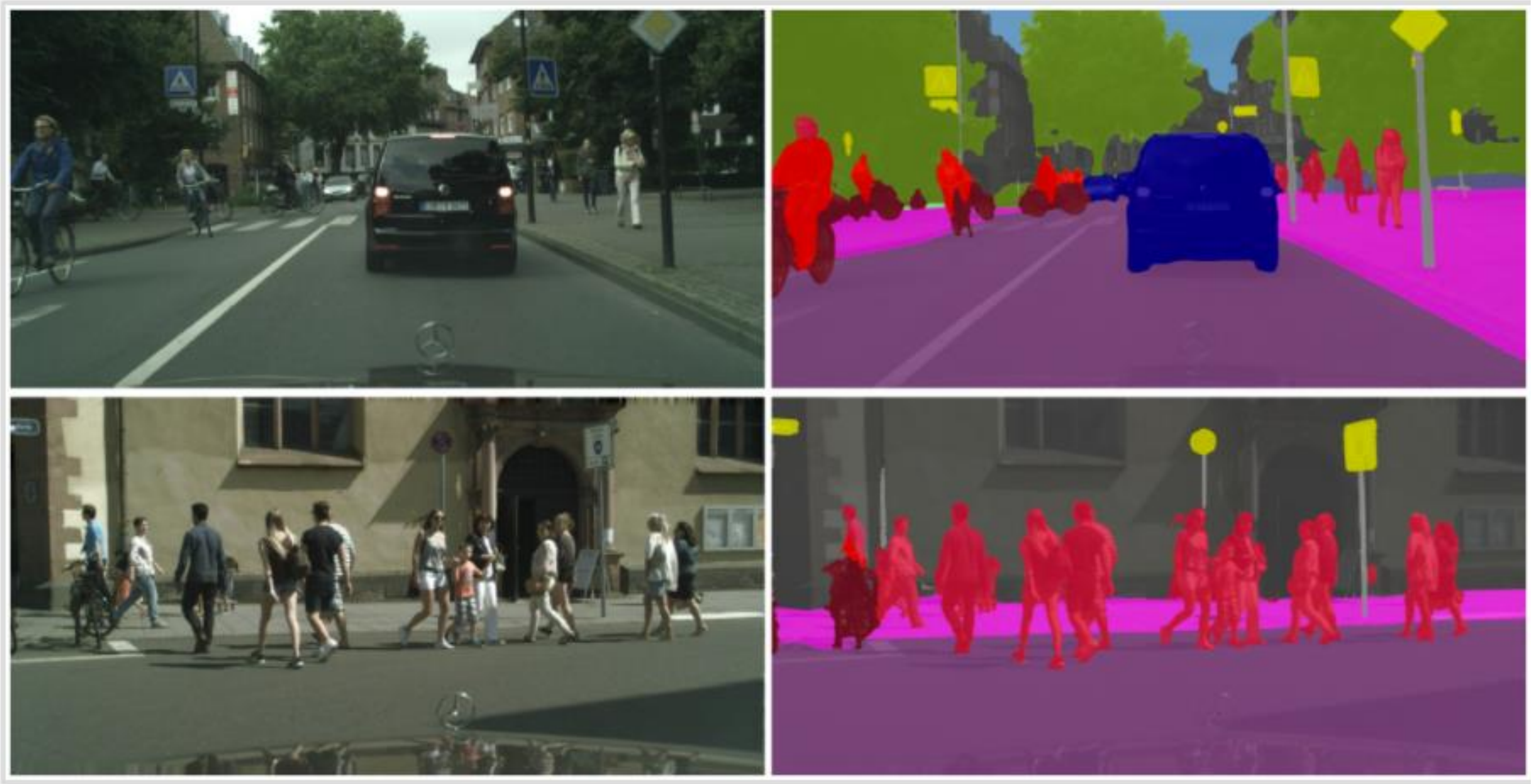
CNN classifier



Exercise: https://github.com/retico/cmepda_medphys/L7_code/Hands-on/DL_classification_microcalcifications.ipynb

See demo code: https://github.com/retico/cmepda_medphys/L8_code/demo_train_CNN.mlx, demo_visualizing_CNN_layers.mlx
Data available on https://drive.google.com/drive/folders/1r3BW9clkwoS-Sq-enGGBL3LyC1gszSh7?usp=drive_link

Semantic segmentation



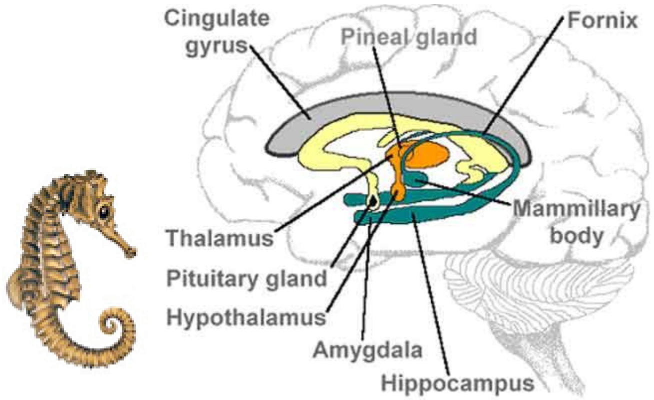
Semantic segmentation is the process of classifying each pixel belonging to a particular label.
It does not distinguish different instances of the same object.

https://vladlen.info/publications/feature-space-optimization-for-semantic-video-segmentation/

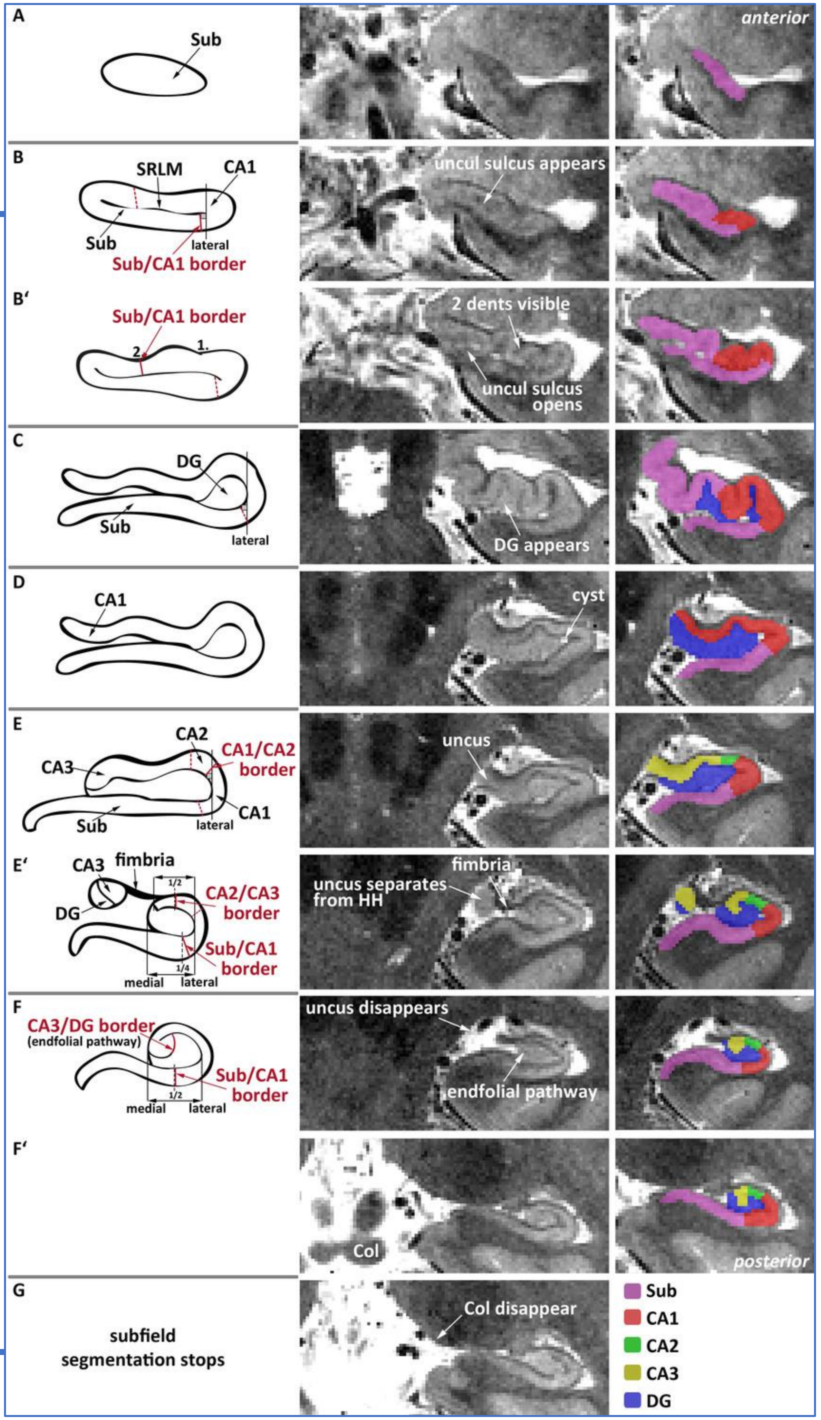
Medical image segmentation

- Image segmentation: the image is partitioned into its parts or regions (segments)
- Segmentation of tissues, organs, lesions or other **regions of interest (ROIs)** out of a medical image is often a useful step to extract meaningful information related to shape or texture of the object of interest

For example, in the study of Alzheimer's Disease (AD), the hippocampus is one of the first regions of the brain to suffer damage; memory loss and disorientation are included among the early AD symptoms.



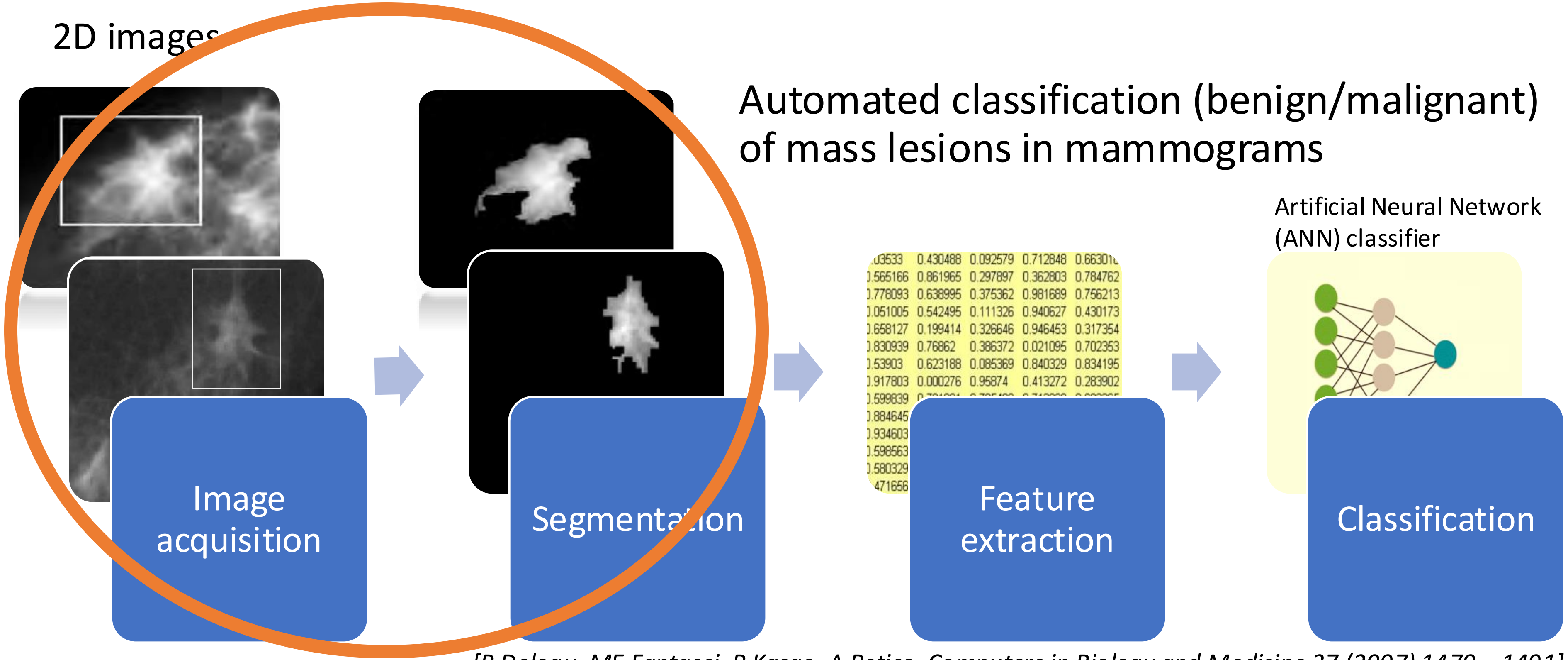
D. Berron et al., A protocol for manual segmentation of medial temporal lobe subregions in 7 Tesla MRI, NeuroImage: Clinical 15(C) 2017



Typical image analysis pipeline for assisted diagnosis

Example: 1) Object segmentation; 2) Hand-crafted feature extraction; 3) Machine Learning classification

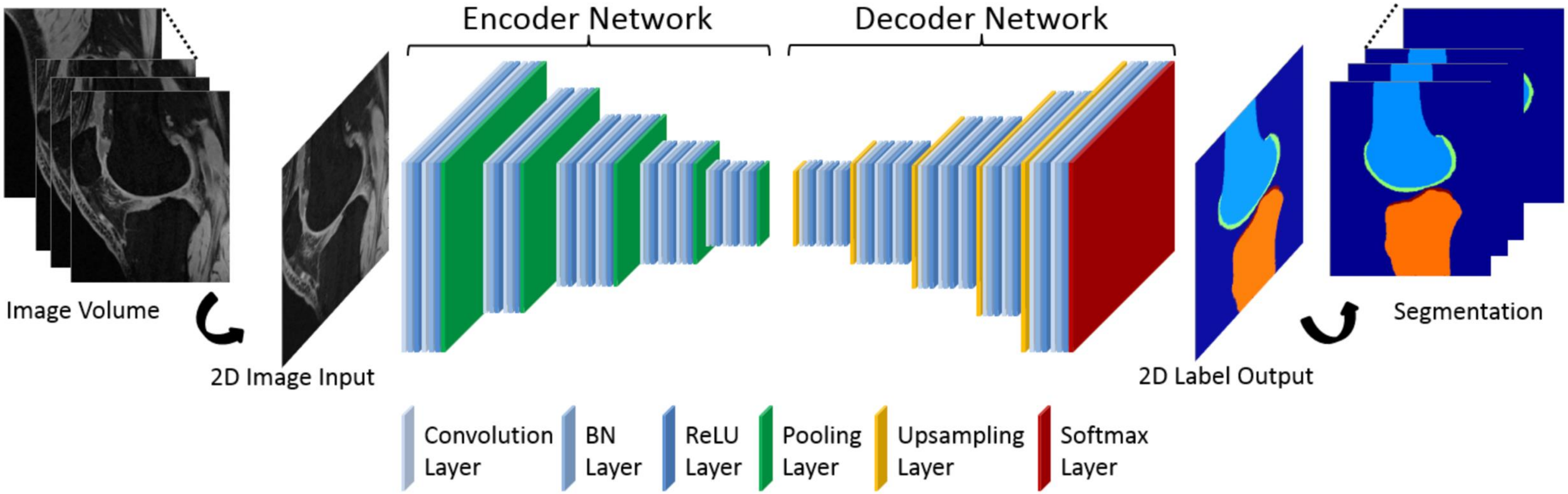
Automated classification (benign/malignant) of mass lesions in mammograms



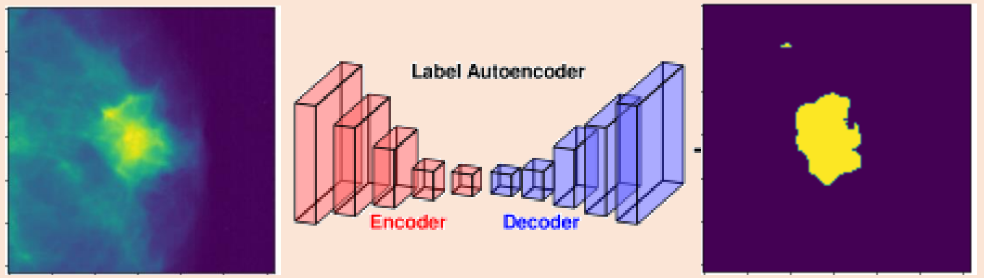
[P Delogu, ME Fantacci, P Kasae, A Retico, Computers in Biology and Medicine 37 (2007) 1479 – 1491]

See demo code (L5_code/mass_segmentation): mass_segment.m
Data available on https://drive.google.com/drive/folders/1r3BW9clkwoS-Sq-enGGBL3LyC1gszSh7?usp=drive_link

Convolutional Auto-Encoders for image segmentation



Liu et al, Deep Convolutional Auto-Encoder and 3D Deformable Approach for Tissue Segmentation in Magnetic Resonance Imaging, Proc. Intl. Soc. Mag. Reson. Med. 25, 2017

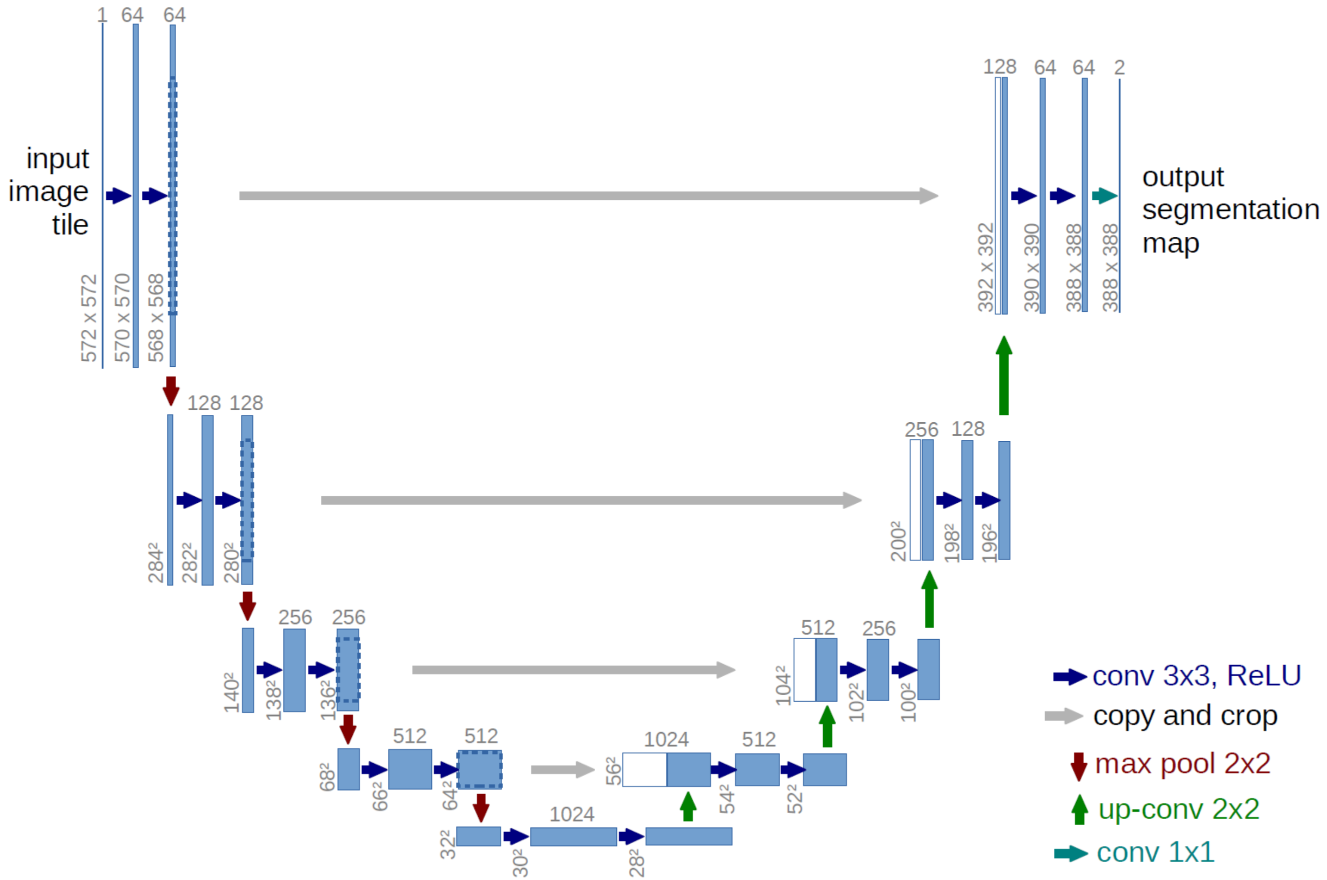


See L8_code/DL_lesion_segmentation_CAE_Unet.ipynb
on https://github.com/retico/cmepda_medphys/

U-net

U-Net: Convolutional Networks for Biomedical Image Segmentation
Olaf Ronneberger, Philipp Fischer, Thomas Brox
Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, LNCS, Vol.9351: 234--241, 2015, available at arXiv:1505.04597

Winner of [ISBI Challenge: Segmentation of neuronal structures in EM stacks]()



https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/

3D nnUNet by Fabian Isensee
https://github.com/MIC-DKFZ/nnUNet

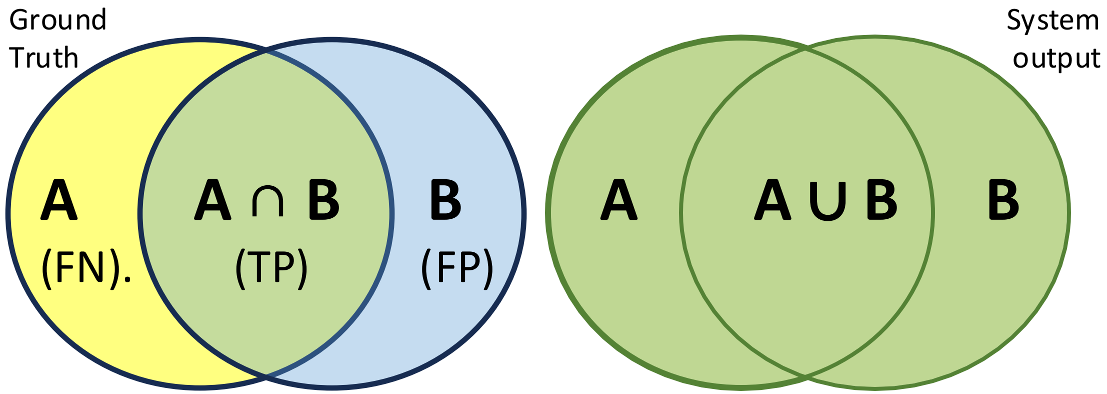
Challenges in data collection

- Precise object localization is hard to annotate
- Annotating every pixel is extremely time consuming
- Common solution is to define a segmentation mask:
  - annotate relevant objects (foreground)
  - mark rest as “other” (background)

Segmentation similarity measures

- **Jaccard similarity coefficient: Intersection over Union**

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} \qquad 0 \le J(A,B) \le 1.$$



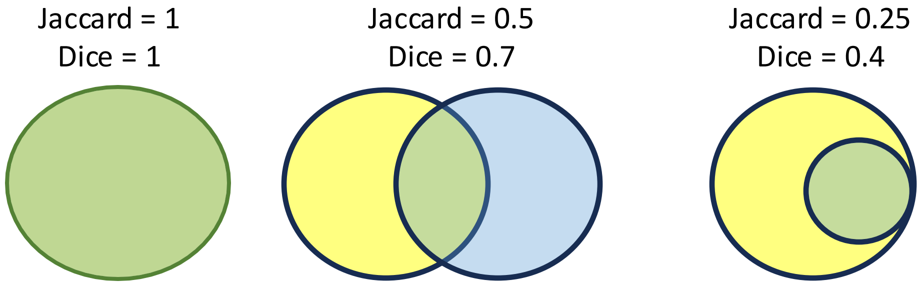
- **Sørensen-Dice similarity coefficient**

$$DSC = \frac{2 \times |A \cap B|}{|A| + |B|}$$

where $|A|$ and $|B|$ are the cardinalities of the two sets (i.e. the number of elements in each set).
The Sørensen index equals twice the number of elements common to both sets divided by the sum of the number of elements in each set.
When applied to boolean data, using the definition of true positive (TP), false positive (FP), and false negative (FN), it can be written as:

$$DSC = \frac{2TP}{2TP + FP + FN}.$$

It is different from Jaccard index which only counts true positives once in both the numerator and denominator.



Both **Jaccard = 1** and **Dice = 1** indicate a perfect segmentation

DL vs. traditional ML approaches

- Deep Neural Networks are competing with traditional handcrafted feature extraction + ML approaches to Medical Image Analysis, fostering a more

  $\longrightarrow$ ***data driven decision making***

  - **Pros:**
    - No prior selection of problem-related features => no loss of information
  - **Cons:**
    - **Larger annotated** data samples are necessary
    - **Deep Neural Networks are black boxes: which image features are relevant for discrimination?**

Data augmentation (flip, rotate, scale images to augment data sets)

Model interpretability, explainable AI

Reproducibility in Medical Imaging Analysis studies

- Different analysis pipelines with the same purpose may rely both on different principles and on different algorithm implementations
- Extensive tests should be performed to assess the **reliability** of the features/measures they produce, before the output of SW packages for the analysis of medical data is used to infer results in the field of clinical research
- To estimate the precision of a SW package, a reference “gold standard” is necessary
- In most cases “gold standard” measures on large samples are not available, nevertheless, we can evaluate:
  - the robustness of a SW pipeline (**intra-method agreement**)
  - the reproducibility of the same measure across different SW pipelines (**inter-method agreement**)
- **The lack of intra-method and inter-method agreement can produce inconsistent results in medical imaging studies**

The *gold standard (ground truth)* problem

- Expert radiologists :
  - They quite often do not agree with each other on the presence of lesions or on the lesion relevance
  - Their manual segmentations will not be highly reproducible in inter- and intra-reader analyses

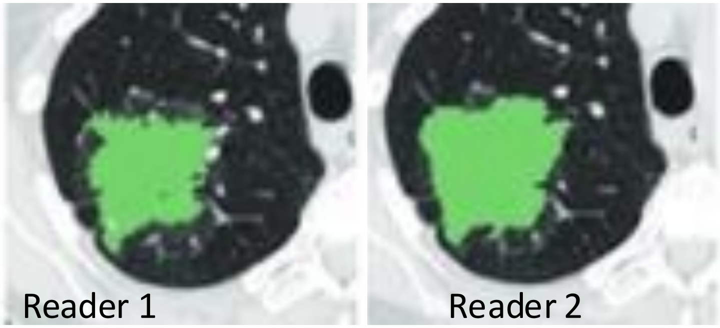
- The most common strategy to create a gold standard is to form a panel of experts to jointly find a solution in doubtful cases

- Problems when performing algorithm training and validation:
  - the algorithm performance may drastically change according to the gold standard

- A possible alternative approach is to rely on **synthetic datasets…**

Limited availability of annotated data: small dataset for ML/DL training

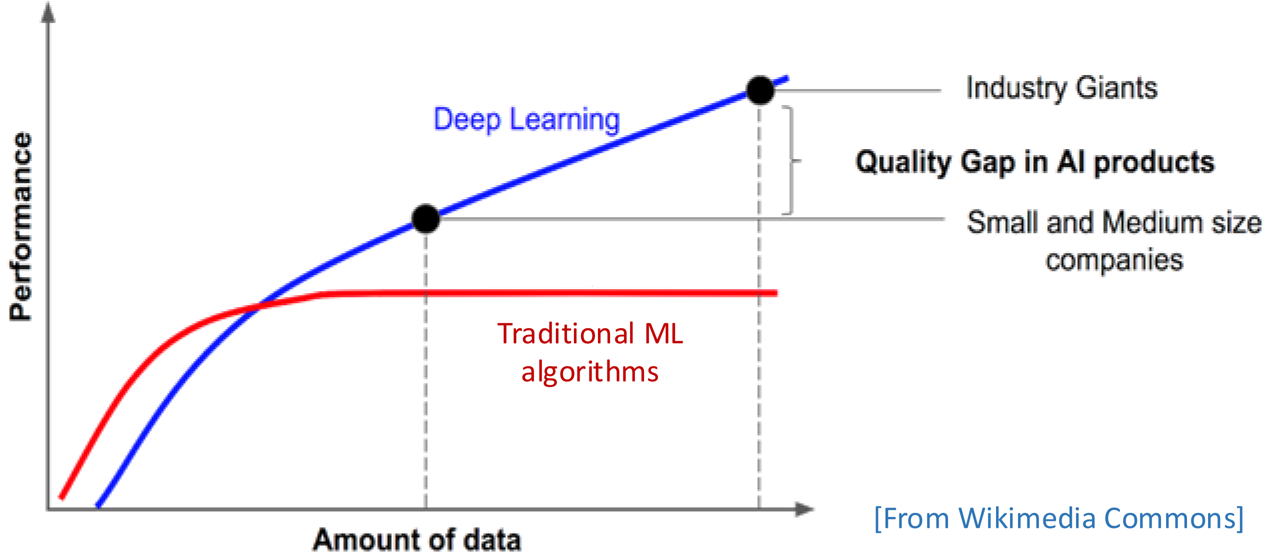
- **Data annotation by human experts is an extremely time-consuming task**, which typically requires:
  - the collection of additional information from other storing systems,
  - expertise in segmenting meaningful regions in images,
  - specific knowledge to assign class labels.
- Moreover, segmentation of organs or lesions (i.e. **voxel-wise annotation**) are affected by inter- and intra-reader variability.



In most cases, medical data samples available for training ML models are of limited size: **small datasets**

Performance of ML algorithms vs. sample size

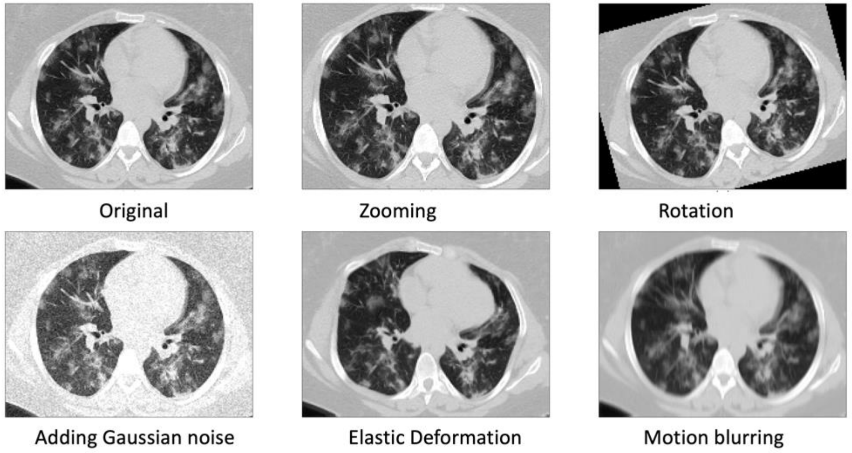
- Traditional ML models can perform even better than DL ones for small sample sizes
- DL models definitely outperform traditional ones in case of large and meaningful data samples



[From Wikimedia Commons]

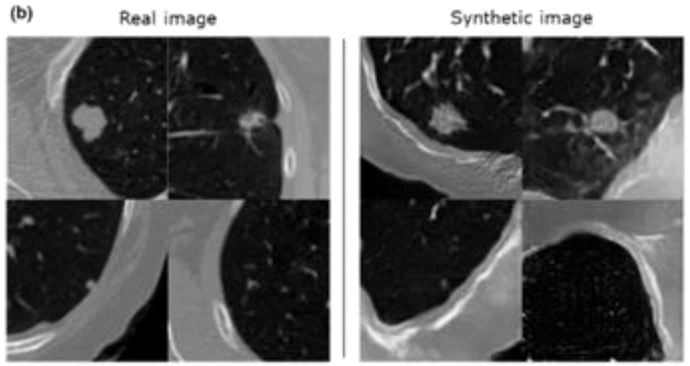
Strategies to mitigate the "small data" problem

➔ Data augmentation with traditional techniques



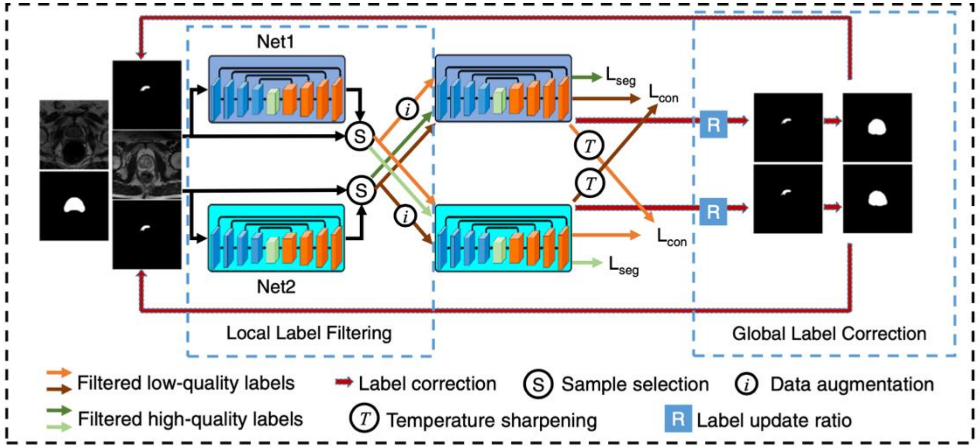
[Lizzi F et al, Quantification of pulmonary involvement in COVID-19 pneumonia…, *IJCARS*, 17(2), 229–237 (2022)]

➔ Data augmentation via synthetic data generation



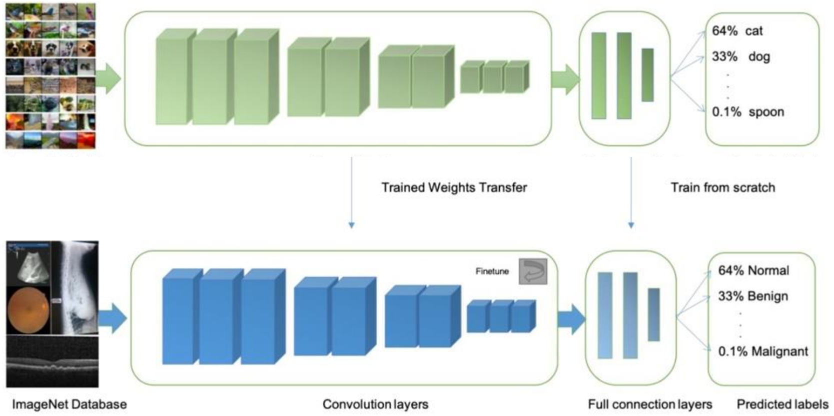
[Chlap P et al, A review of medical image data augmentation techniques for deep learning applications. *Journal of Medical Imaging and Radiation Oncology*, *65*(5), 545–563 (2021)]

➔ Automated/semi-automated annotation



[Wang et al, Annotation-efficient deep learning for automatic medical image segmentation. *Nature Communications*, *12*(1), 1–13 (2021)]

➔ Transfer learning



[Xu et al, Current status and future trends of clinical diagnoses via image-based deep learning. *Theranostics*, *9*(25), 7556–7565 (2019)]

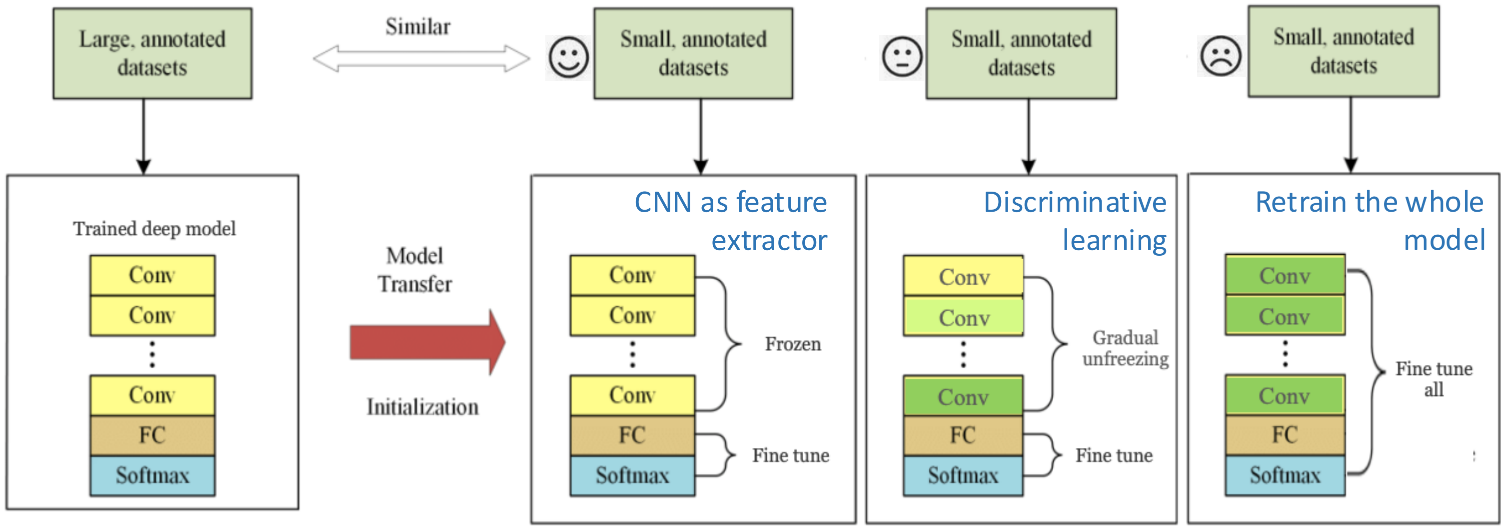
Transfer Learning

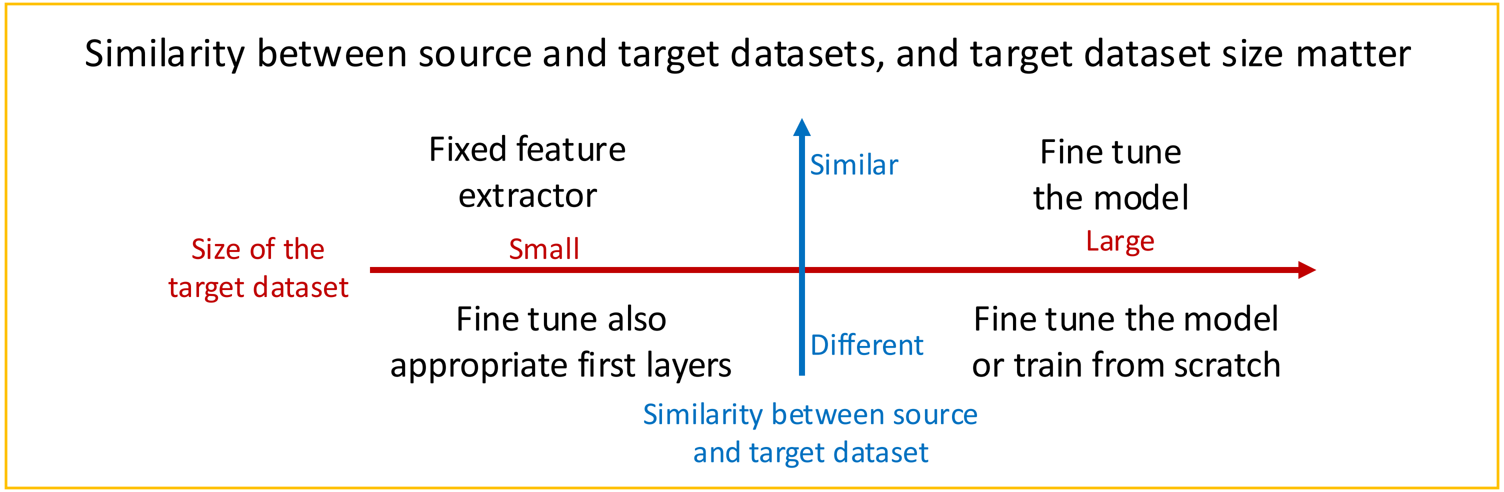
Different TL approaches can be implemented:

- **CNN as feature extractor**
  All but the last feed-forward layer(s) of the network are frozen. The only weights that are trained are those in the last layers.

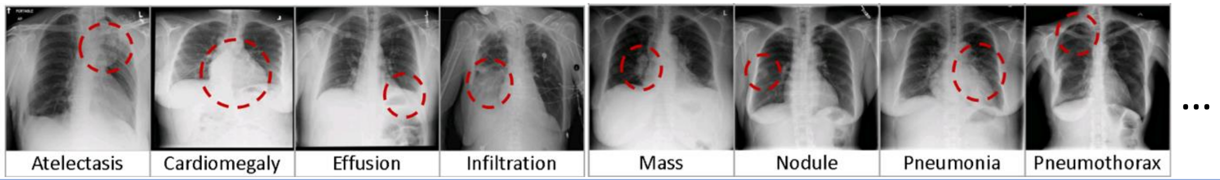
- **Discriminative learning rates with gradual unfreezing**
  The first layers of a network typically learn general features (e.g., lines, circles, colors, etc.). Thus, the weights in those layers should be changed less than the weights of the downstream layers which are more specialized in the target task.

- **Fine tune all CNN simultaneously**
  None of the weights are frozen. The pretrained network is used as a starting point.





Transfer learning



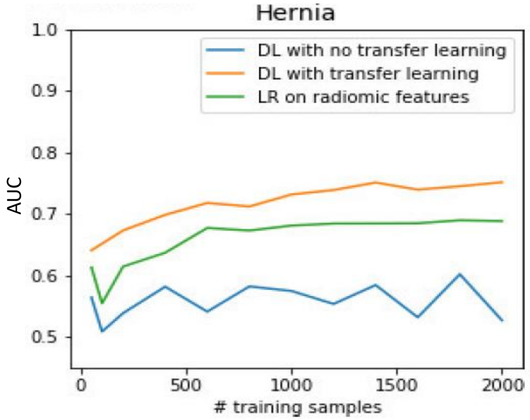
Example of implementation

Comparison of three different TL methods, using DenseNet121, and different training dataset sizes and different classification tasks.

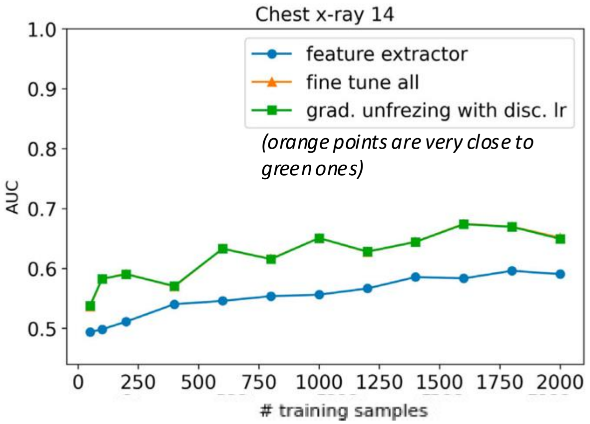
Results:

- Traditional ML can perform better that DL for small datasets; if DL is used, TL performs better.
- Fine-tune-all and gradual-unfreezing methods perform very similar, and they outperform using DL as feature extractor
- Features learned may not be as general as currently believed:
  - TL from models trained on images of the same modality and different anatomical site is equivalent to using ImageNet
- TL is useful for small datasets (N < 2000)

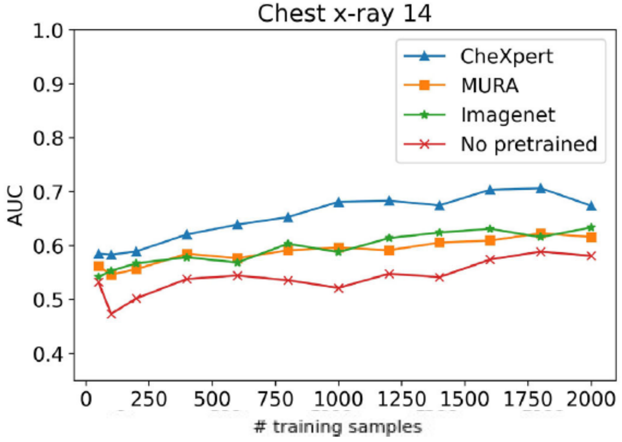
Traditional ML vs DL (w and w/o TL)



Different TL methods



Similarity between source and target datasets

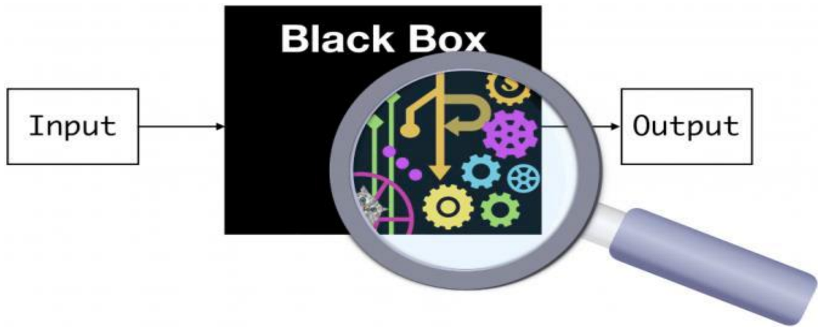


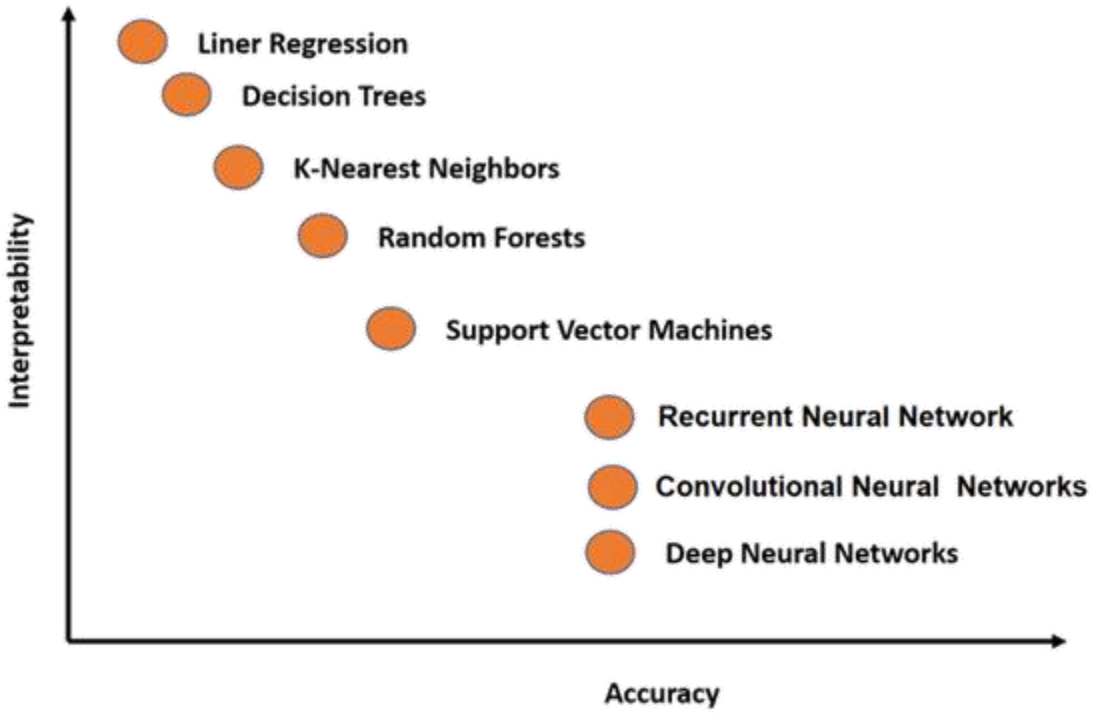
CheXpert: Chest X-ray images
MURA: Musculoskeletal RX images (elbow, finger, forearm, hand, humerus, shoulder, and wrist)
ImageNet: natural images

[Romero et al. Targeted transfer learning to improve performance in small medical physics datasets. *Medical Physics*, *47*(12), 6246–6256 (2020)]

Accuracy vs. explainability of ML models

Artificial intelligence is an extremely promising technology for the future.
To fully exploit its potential and minimize risks, it is necessary to demonstrate that AI can be trusted:
→ AI algorithms are required to be explainable





The models with the **best performance are often very complex,** and thus the motivations behind their responses are very difficult to explain.

Explainable AI **(XAI)** is a research field that aims to make AI results more understandable to humans.

XAI methods for tabular data

In the analysis of tabular features (e.g. Radiomic features) with ML/DL models, we can implement the:

Permutation feature importance

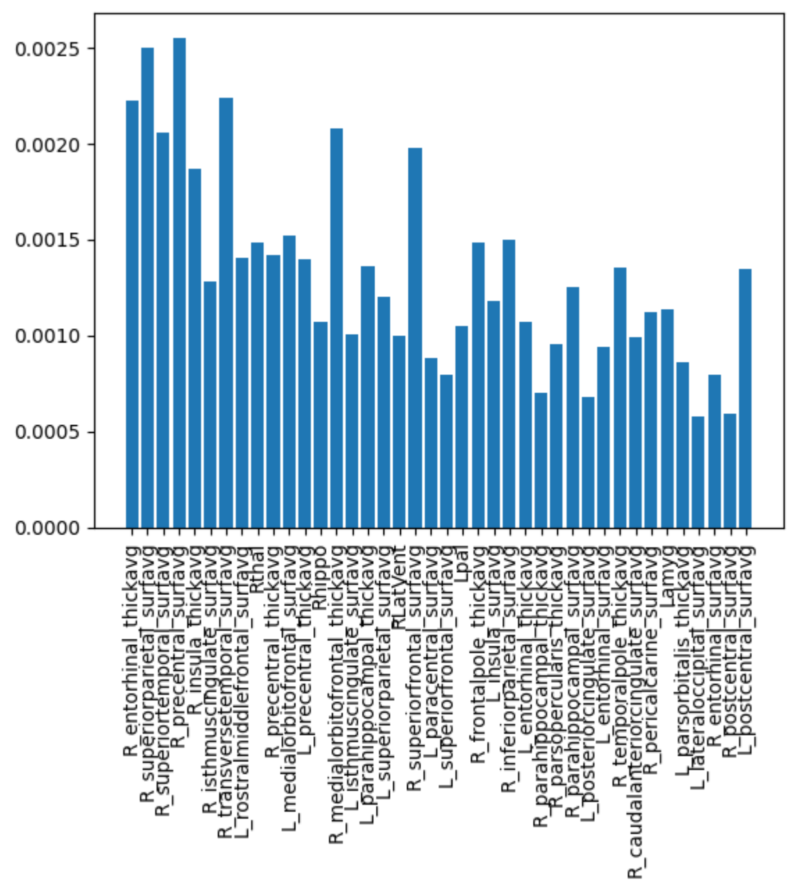
We measure the importance of a feature by calculating the increase in the model's prediction error after permuting the feature. A feature is "important" if shuffling its values increases the model error, because in this case the model relied on that feature for the prediction.

Input: Trained model $\hat{f}$, feature matrix X, target vector y, error measure $L(y, \hat{f})$

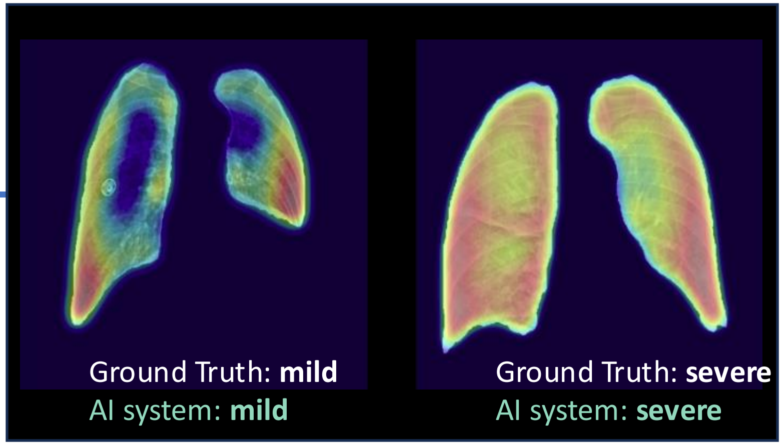
1.Estimate the original model error $e_{origin} = L\left(y, \hat{f}(X)\right)$ (e.g. mean squared error)

2.For each feature j∈{1,...,p} do:

1. Generate feature matrix X_{perm} by permuting feature j in the data X. This breaks the association between feature j and true outcome y.
2. Estimate $e_{perm} = L\left(y, \hat{f}\left(X_{perm}\right)\right)$ based on the predictions of the permuted data.
3. Calculate permutation feature importance as quotient $FI_j = e_{perm}/e_{origin}$ or difference $FI_j = e_{perm} - e_{origin}$.
4. Sort features by descending FI.



XAI methods for images

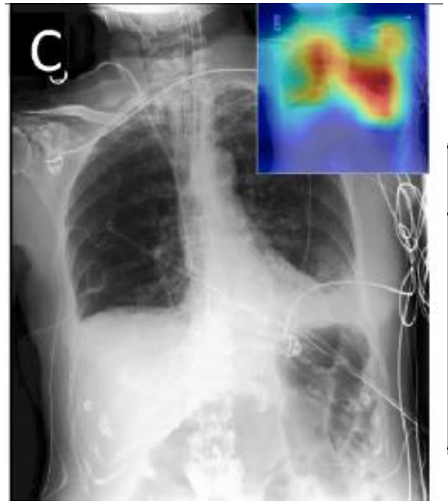


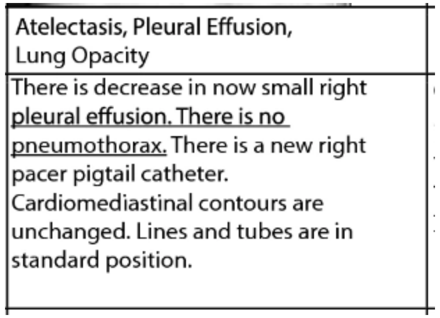
XAI techniques applied to medical imaging can be divided in three main categories:

1) **Visual explanation:**
- it is the most used form of XAI in medical imaging
- **Saliency maps** show the important parts of an image for a prediction.

1) **Textual explanation:**
- it provides textual description
- it can describe image characteristics of an image or even complete a radiological report.

1) **Example-based explanation:**
- it provides examples that are related to a certain data point.
- Example: prototype networks



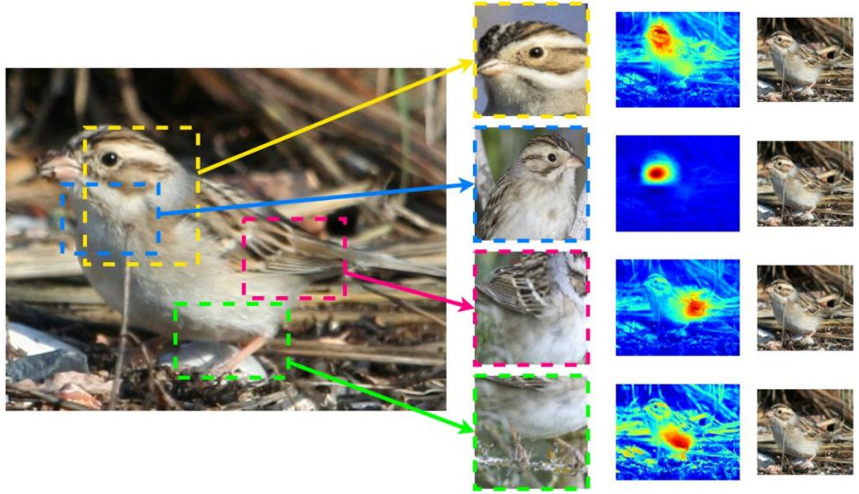


Atelectasis, Pleural Effusion, Lung Opacity

There is decrease in now small right pleural effusion. There is no pneumothorax. There is a new right pacer pigtail catheter. Cardiomediastinal contours are unchanged. Lines and tubes are in standard position.

Selivanov, A., Rogov, O.Y., Chesakov, D. et al. Medical image captioning via generative pretrained transformers. Sci Rep 13, 4171 (2023). https://doi.org/10.1038/s41598-023-31223-5

Chen C, Li O, Tao C, Barnett A, Su J, Rudin C. This Looks Like that: Deep Learning for Interpretable Image Recognition. In: Neural Information Processing Systems (NeurIPS); 2019



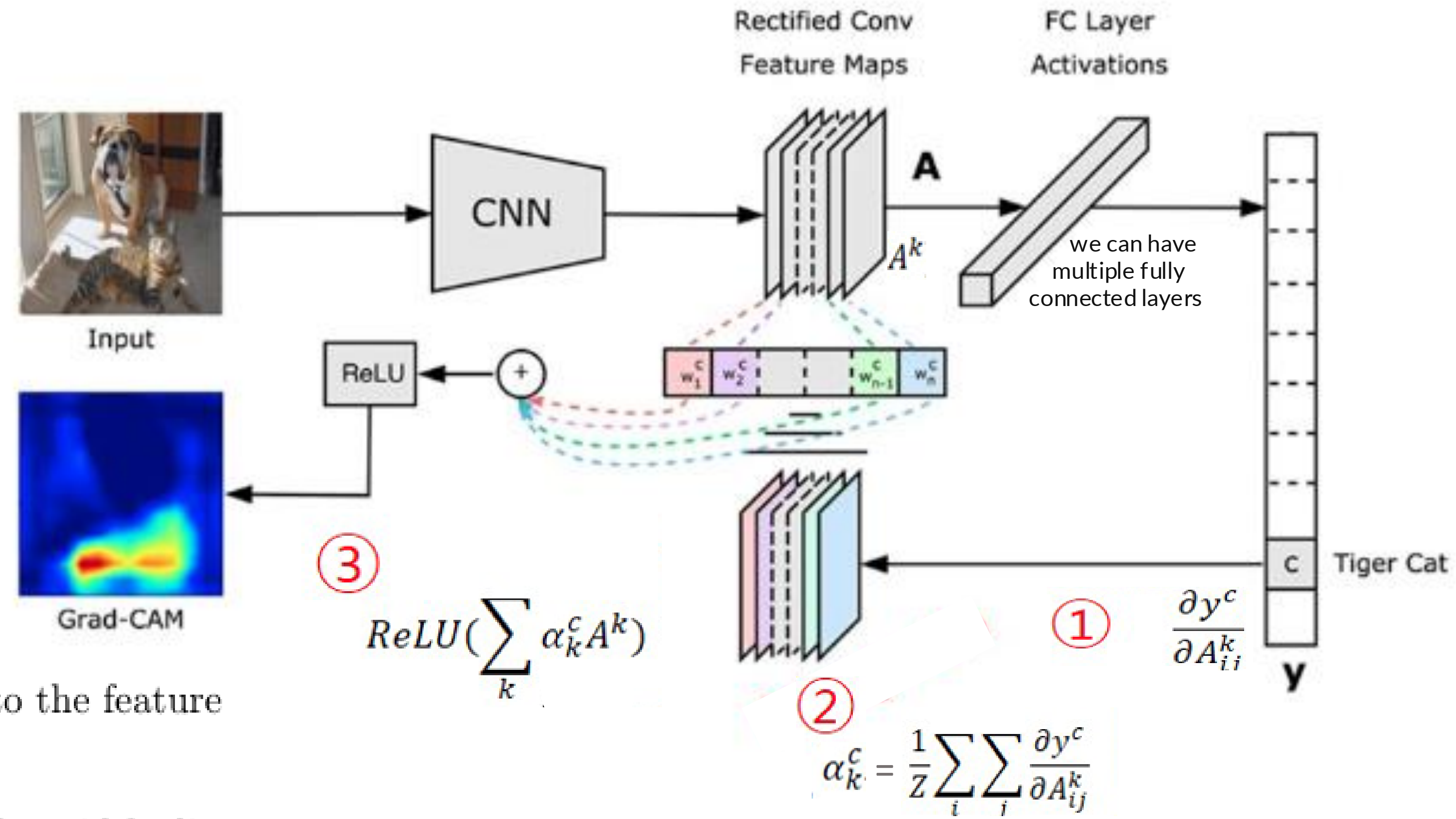
Gradient-based Class Activation Maps (**Grad-CAM**)

it is a method for producing heatmaps that is applied to an already-trained neural network after training is complete and the parameters are fixed.

It allows visualizing where a convolutional neural network model is looking when it takes a decision

→ can provide insight into model failures

It consists in 3 steps:



Step 1: Compute the gradient of y_c with respect to the feature map activations A^k of a convolutional layer, i.e. $\frac{\partial y^c}{\partial A^k}$

Step 2: Global average pool the gradients over the width dimension (indexed by i) and the height dimension (indexed by j) to obtain neuron importance weights α_k^c

Step 3: Perform a weighted combination of the feature map activations A^k where the weights are the α_k^c just calculated

https://glassboxmedicine.com/2020/05/29/grad-cam-visual-explanations-from-deep-networks/

MATLAB and Python

- MATLAB provides a flexible, two-way integration with other programming languages, allowing you to reuse legacy code
- Since the MATLAB release R2014, you can access Python functionality from MATLAB (you call Python functions and objects directly from MATLAB)
- To call Python modules in MATLAB, you must have a supported version of the reference implementation (CPython) installed on your system: https://it.mathworks.com/help/matlab/matlab_external/install-supported-python-implementation.html
- https://it.mathworks.com/support/requirements/python-compatibility.html

You can create a virtual environment for a specific version of Python using the Python venv module. You can also install the MATLAB Engine API for Python in a virtual environment

Install Supported Python Implementation

- **MATLAB 2023b supports Python versions 3.9, 3.10, 3.11.**
- MATLAB selects the version of Python based on your system path (you can specify the path to a specific python enviroment).
- To call a Python function, type py. in front of the module name and function name.
- Pass MATLAB data as arguments to Python function.
- MATLAB converts the data into types that best represent the data to the Python language.

https://it.mathworks.com/help/matlab/call-python-libraries.html

```
>> pyenv                 % to determine which Python version MATLAB is using
>> pyenv('Version','/usr/bin/python3’)   % to change default environment
                                           of Python interpreter
>> py.help('len’)        % To call a Python function, type py. in front of the
                           function name
>> L = py.os.listdir(".");
```

https://it.mathworks.com/help/matlab/matlab-engine-for-python.html?lang=en

```
>>> import matlab.engine
>>> eng = matlab.engine.start_matlab()
>>> M1 = eng.ones(3)
>>> eng.quit()
```

How to call Python from MATLAB

Configure Your System to Use Python

- **Python Support**
- To call Python® modules in MATLAB®, you must have a supported version of the reference implementation (CPython) installed on your system. Install a distribution, such as those found at https://www.python.org/downloads/. MATLAB does not support CPython versions installed from the Microsoft® store. For supported version information, see Versions of Python Compatible with MATLAB Products by Release. If you are on a Linux® or Mac platform, you already have Python installed. If you are on Windows®, you need to install a distribution, if you have not already done so. For more information, see Install Supported Python Implementation.
- To verify that Python is installed on your system, open the Python interpreter from your system prompt and call Python functions.
- By default, MATLAB selects the version of Python based on your system path. To view the system path in MATLAB, use the getenv('PATH') command. To determine which version MATLAB is using, call the pyenv function.

```
pe = pyenv; pe.Version

ans = "3.11"
```

- The value set by pyenv is persistent across MATLAB sessions. If you have multiple supported versions, use pyenv to display the version currently used by MATLAB. MATLAB automatically selects and loads a Python version when you type a Python statement. For example, to call *func_name*, type:

```
py.func_name
```

See https://github.com/retico/cmepda_medphys/L8_code/matlab_python /How_to_call_Python_from_MATLAB.mlx

How to call MATLAB from Python

Install MATLAB Engine API for Python

- To start the MATLAB® engine within a Python® session, you first must install the engine API as a Python package.

  https://it.mathworks.com/help/matlab/matlab_external/install-the-matlab-engine-for-python.html
  https://it.mathworks.com/help/matlab/matlab_external/system-requirements-for-matlab-engine-for-python.html

Verify Your Configuration

- Before you install, verify your Python and MATLAB configurations.
- Check that your system has a supported version of Python and MATLAB R2014b or later. For more information, see Versions of Python Compatible with MATLAB Products by Release.

Install Engine API

- You can install the MATLAB Engine API for Python using the pip command or a Python setup script setup.py.
- **Install Using pip**
- Starting with MATLAB R2022b, you can use the pip command to install the API. Choose one of the following procedures and execute from the system prompt.
- To install from the MATLAB folder, on Windows® type:

  cd "*matlabroot*\extern\engines\python" python -m pip install .
- Install the engine API from https://pypi.org/project/matlabengine with the command:

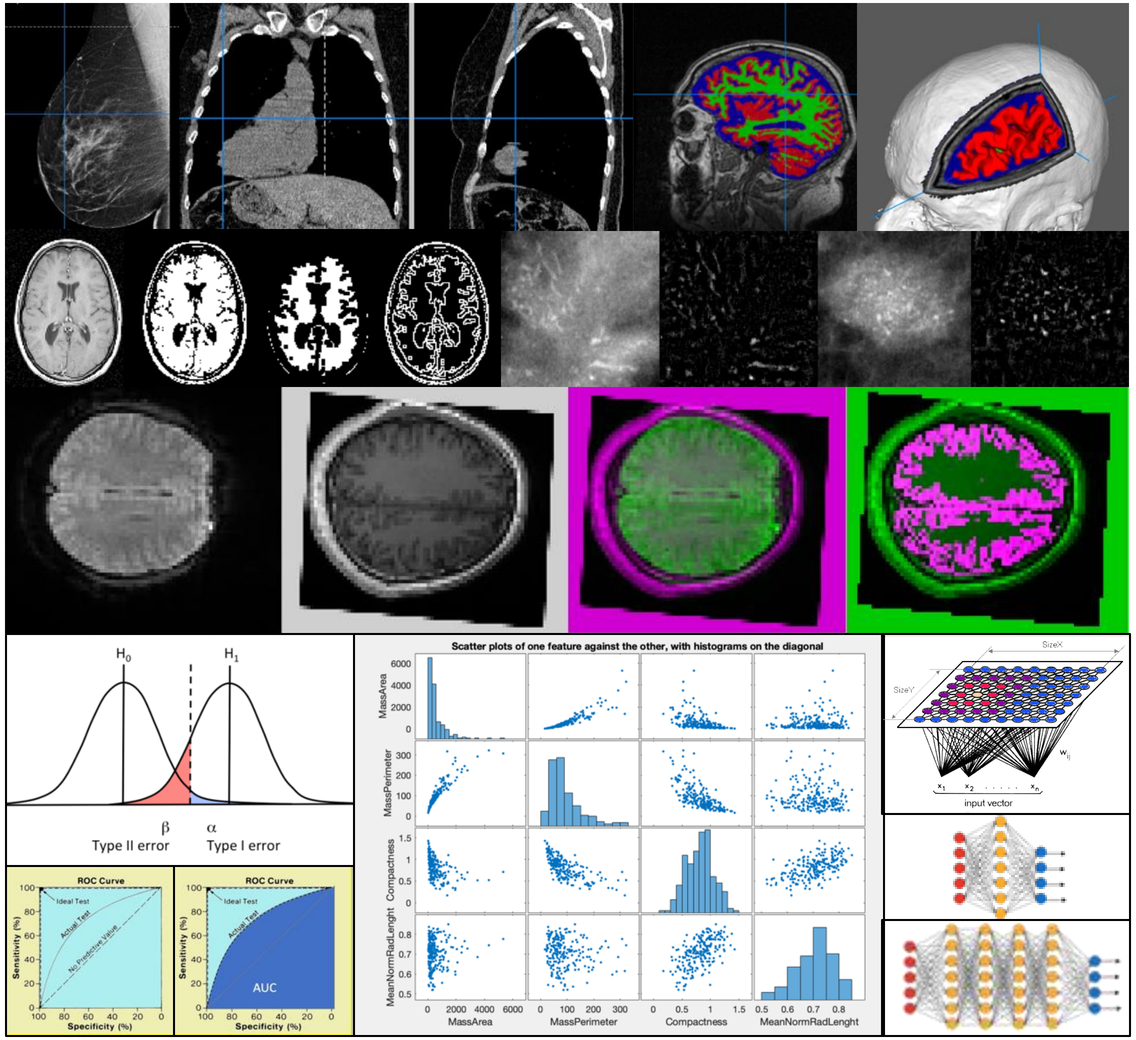
  python -m pip install matlabengine

See https://github.com/retico/cmepda_medphys/L8_code/matlab_python /How_to_call_MATLAB_from_Python.ipynb

Summary of the course module

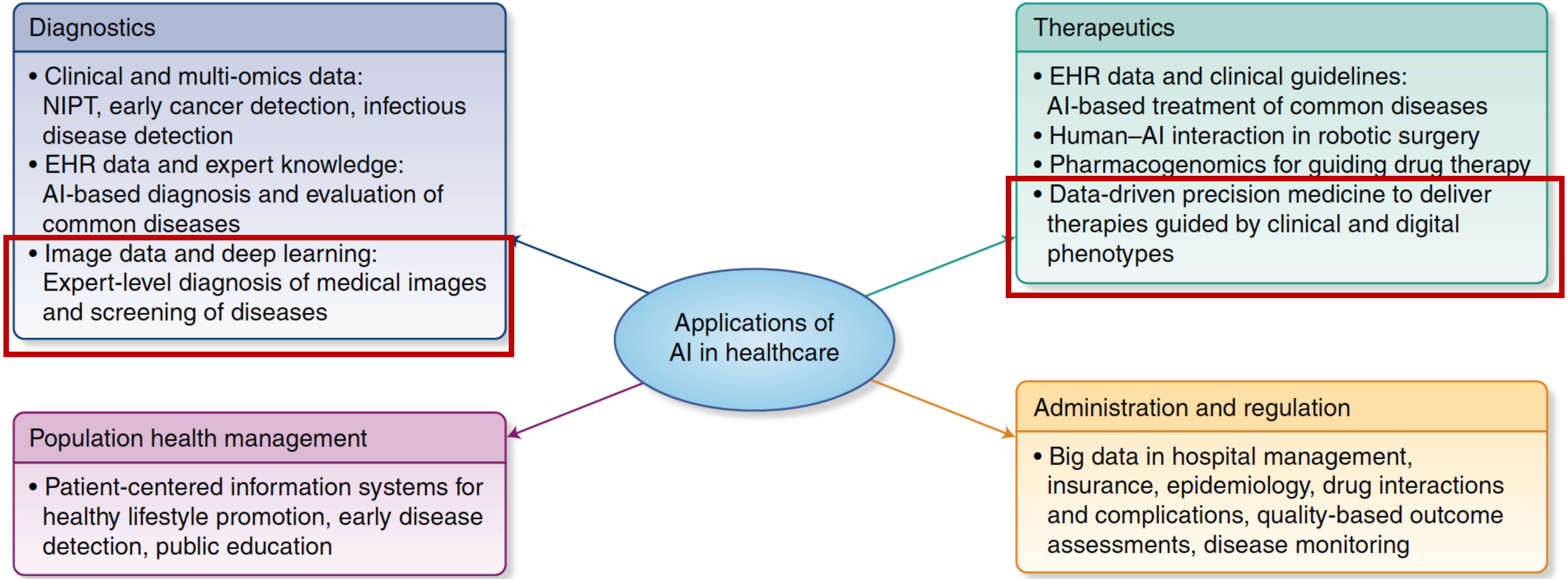
Medical data processing, feature extraction, feature/image classification:

- ✓ Handling standard-format medical data (DICOM), data anonymization, visualization
- ✓ Image filtering
- ✓ Deriving features form images, image segmentation, image coregistration
- ✓ Data quality control, outlier removal, dimensionality reduction
- ✓ Data exploration and analysis, supervised and unsupervised machine learning techniques
- ✓ Performance evaluation: metrics and cross-validation strategies
- ✓ Machine-learning and deep-learning tools for segmentation and classification



Artificial Intelligence applications in healthcare

J He et al., The practical implementation of artificial intelligence technologies in medicine, Nature Medicine ***25****, 30–36 (2019)*



Legend: HER, Electronic Health Records; NIPT, noninvasive prenatal test

For an overview of INFN activities in this field have a look at the Artificial Intelligence in Medicine (AIM) project, https://www.pi.infn.it/aim/

Concluding remarks

- Medical diagnostic imaging daily produces to an incredible amount of digital information
  - **not fully exploited neither for diagnosis nor for research!**

- Clinicians need to be supported by reliable, effective and easy-to-use tools for diagnosing and monitoring a wide range of disease conditions

- Large Consortia are sharing multimodal and multicenter data in different medical fields

- The Medical Imaging community still lacks:
  - automated data quality pipelines
  - data harmonization strategies both for longitudinal and multicenter studies
  - new computational approaches to process and to mine multimodal data (imaging, genetics, clinical, demographic, etc.)
  - **Reliable, validated and explainable** expert systems to support diagnosis and follow up of patients

References, sources and useful links

- References-Books
  - Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel: Backpropagation Applied to Handwritten Zip Code Recognition, Neural Computation, 1(4):541-551, Winter 1989
  - S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice-Hall
  - I. Goodfellow, Y. Bengio, A. Courville. Deep Learning, The MIT Press
  - A. Géron, Hands-On Machine Learning with Scikit-Learn and TensorFlow, O’Reilly
  - Python Data Science Handbook by Jake VanderPlas https://github.com/jakevdp/PythonDataScienceHandbook
- Sources
  - https://www.deeplearningitalia.com/una-panoramica-introduttiva-su-deep-learning-e-machine-learning/
  - http://www.asimovinstitute.org/neural-network-zoo/
  - https://towardsdatascience.com
  - https://it.mathworks.com/help/matlab/matlab-engine-for-python.html

References, sources and useful links

Papers

- Liu et al, Deep Convolutional Auto-Encoder and 3D Deformable Approach for Tissue Segmentation in Magnetic Resonance Imaging, Proc. Intl. Soc. Mag. Reson. Med. 25, 2017
- Vijay Badrinarayanan et. al 2017 “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation” https://arxiv.org/abs/1511.00561
- U-Net: Convolutional Networks for Biomedical Image Segmentation Olaf Ronneberger, Philipp Fischer, Thomas Brox, Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, LNCS, Vol.9351: 234--241, 2015, available at arXiv:1505.04597

Blogs

- https://towardsdatascience.com/master-the-coco-dataset-for-semantic-image-segmentation-part-1-of-2-732712631047
- https://neptune.ai/blog/image-segmentation-in-2020

Datasets

- https://archive.ics.uci.edu/ml/datasets/Image+Segmentation
- https://cocodataset.org/#home
- https://groups.csail.mit.edu/vision/datasets/ADE20K/
- https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/fg/

Repos

- https://lmb.informatik.uni-freiburg.de/people/ronneber/u-net/