ARAMIS LAB
BRAIN DATA SCIENCE

www.aramislab.fr

www.clinica.run

AI4Health School 2021

Practical session DL4MI

Deep learning for computer-aided diagnosis from images

Application to Alzheimer's disease

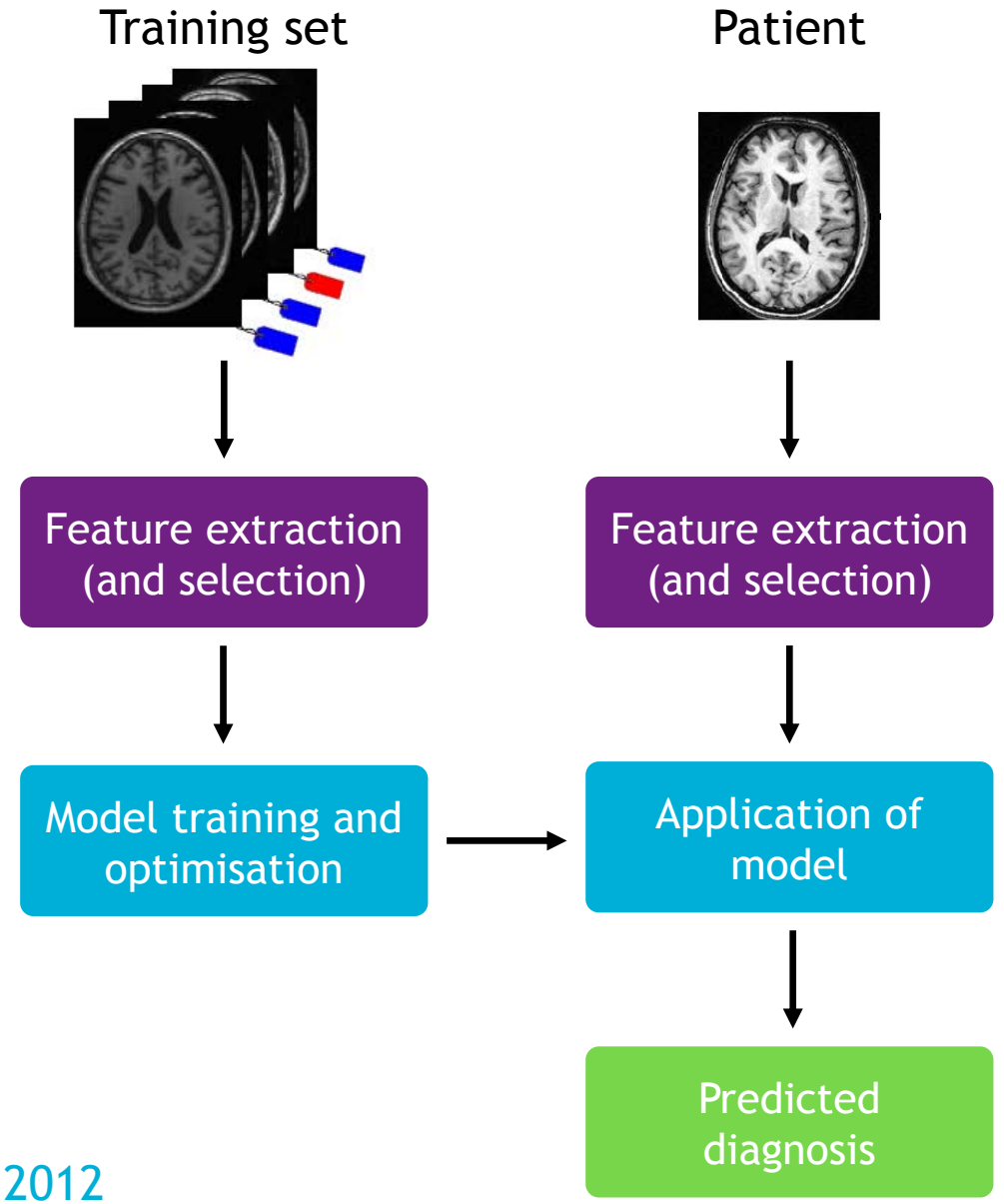
Ninon Burgos, with Simona Bottani, Mauricio Diaz Melo, Johann Faouzi and Elina Thibeau-Sutre

Aramis Lab, Paris Brain Institute, France

Clinical pattern recognition

Basic elements of a machine learning classification pipeline

- Training data set
- Feature extraction from raw data and dimensionality reduction
- Model training and optimisation
- Application to test data

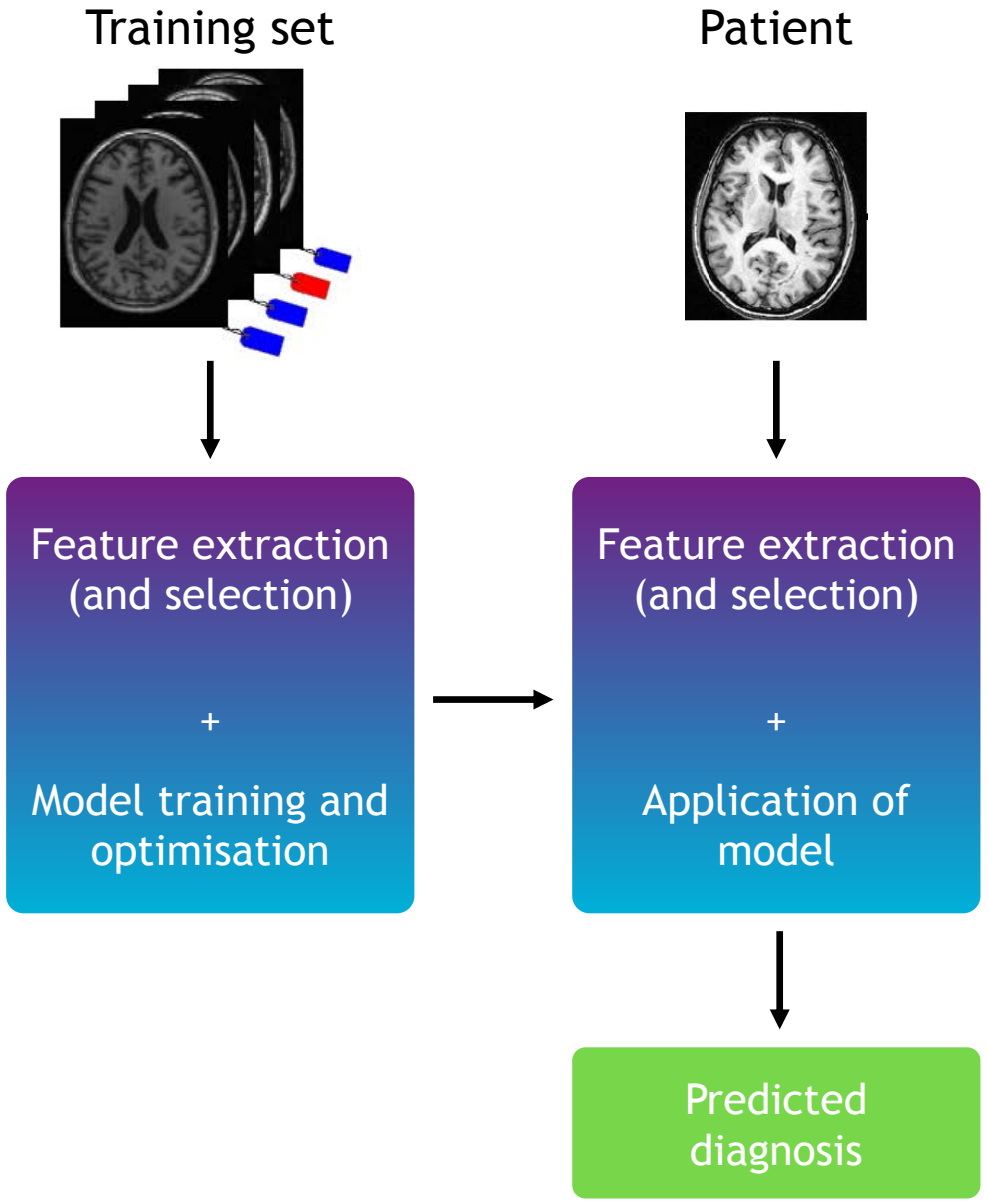


Klöppel et al., NeuroImage, 2012

Clinical pattern recognition

Basic elements of a deep learning classification pipeline

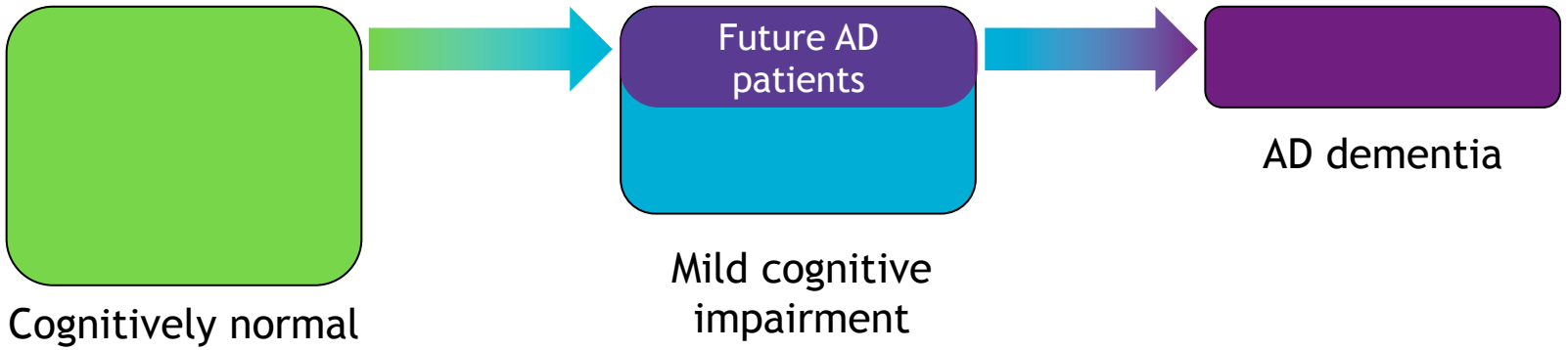
- Training data set
- Feature extraction from raw data and dimensionality reduction
- Model training and optimisation
- Application to test data



Use case: Alzheimer's disease (AD)

What is Alzheimer's disease?

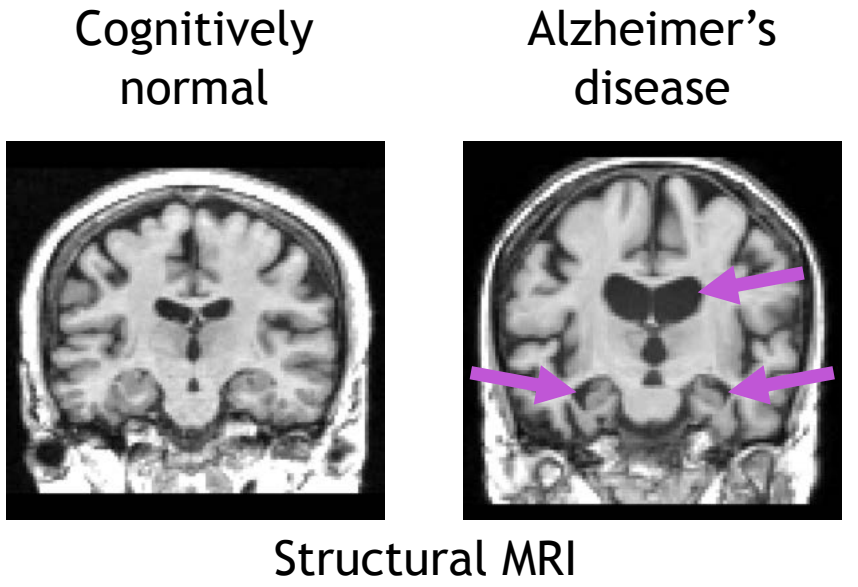
- Most common cause of dementia
- Disorder caused by abnormal brain changes
- Trigger decline in cognitive abilities, severe enough to impair daily life
- Affect behaviour, feelings and relationships
- Progressive disease



Use case: Alzheimer's disease (AD)

AD-related markers

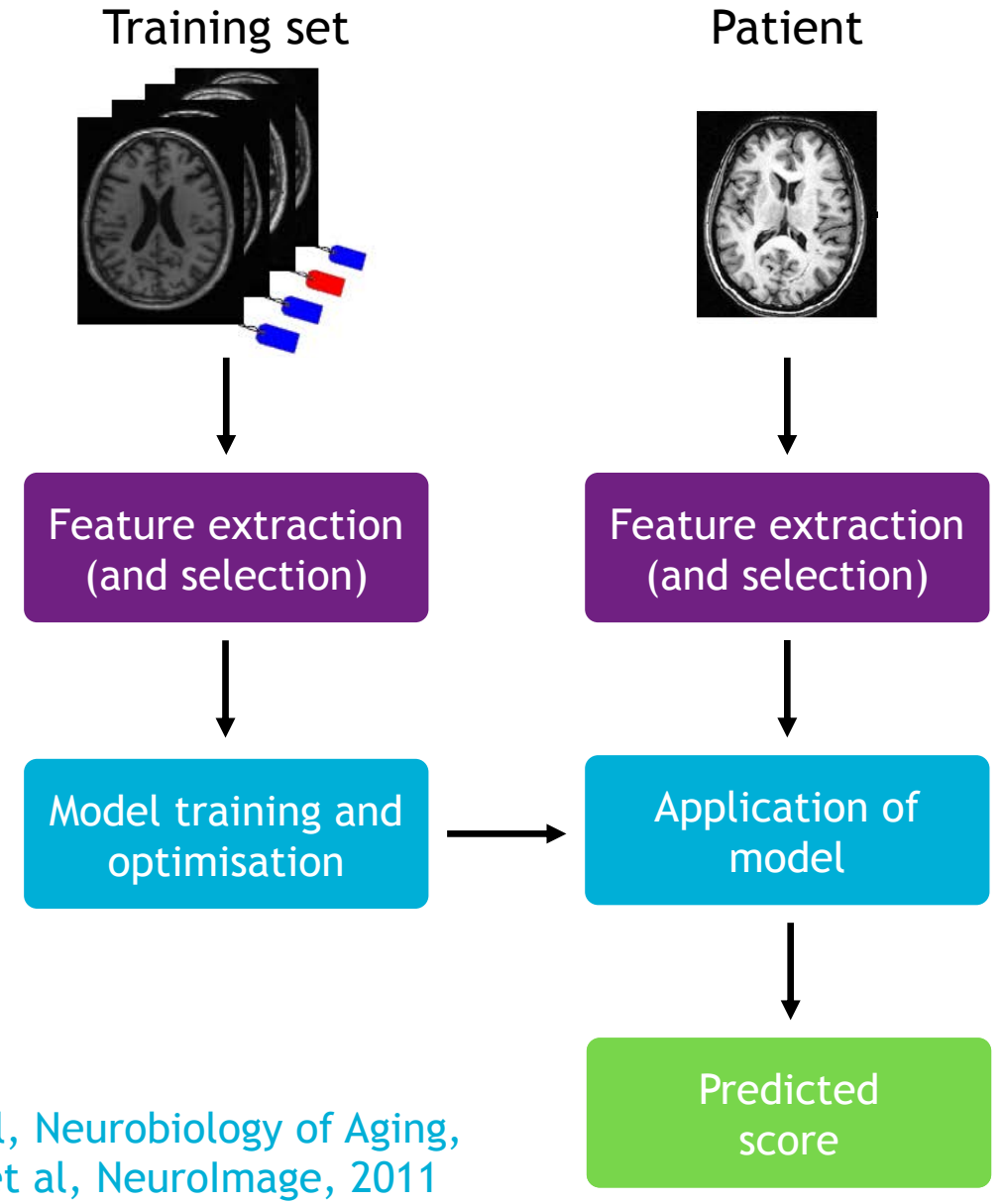
- **Clinical/cognitive tests**
  - Neuropsychological testing of cognitive functions (memory, language, etc.)
- **Structural MRI**
  - Atrophy
- **FDG PET**
  - Hypometabolism
- **CSF Aß42, CSF tau, amyloid PET, tau PET, diffusion MRI, etc.**



Clinical pattern recognition

Use case: Alzheimer's disease (AD)

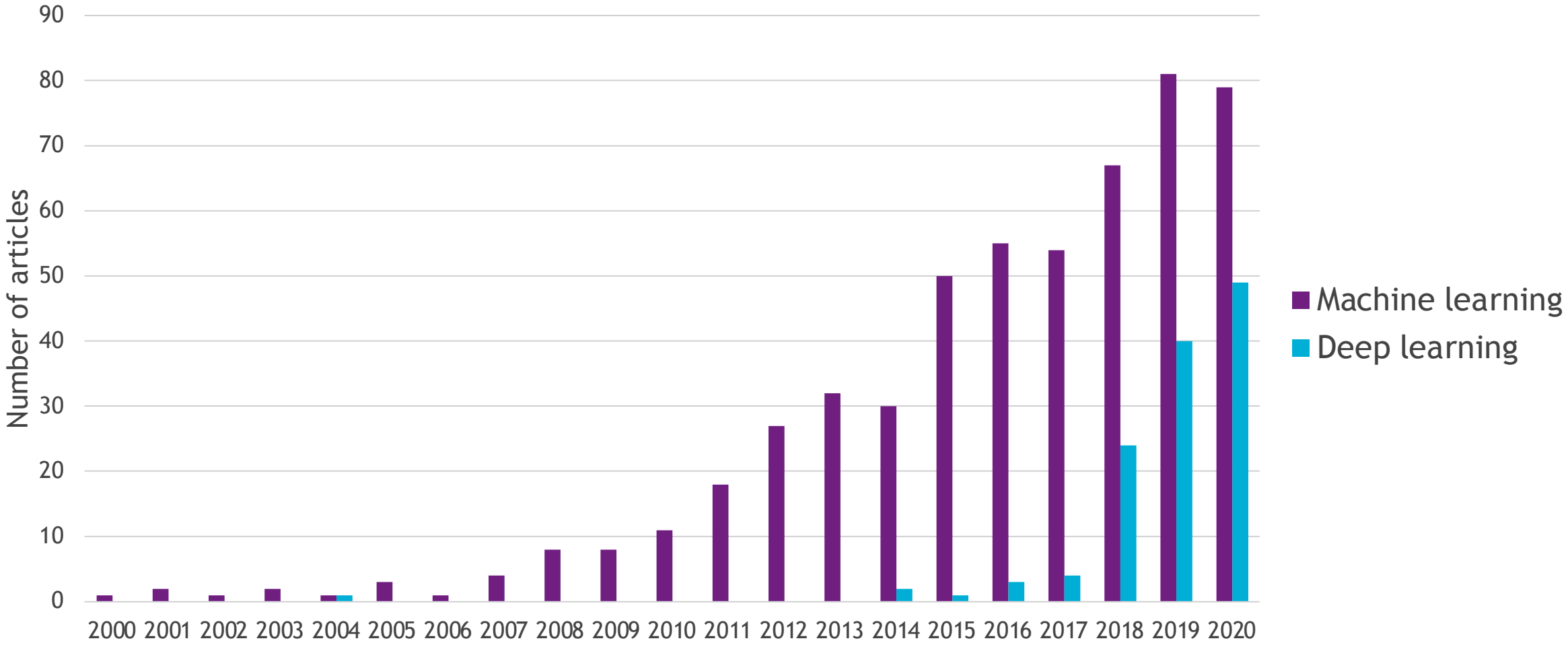
- **Classification**
  - Controls vs AD patients
  - Stable vs progressive mild cognitive impairment (MCI)
- **Regression**
  - Time of onset
  - Future clinical score



Klöppel et al., NeuroImage, 2008; Davatzikos et al, Neurobiology of Aging, 2008; Zhang et al, NeuroImage, 2011; Cuingnet et al, NeuroImage, 2011

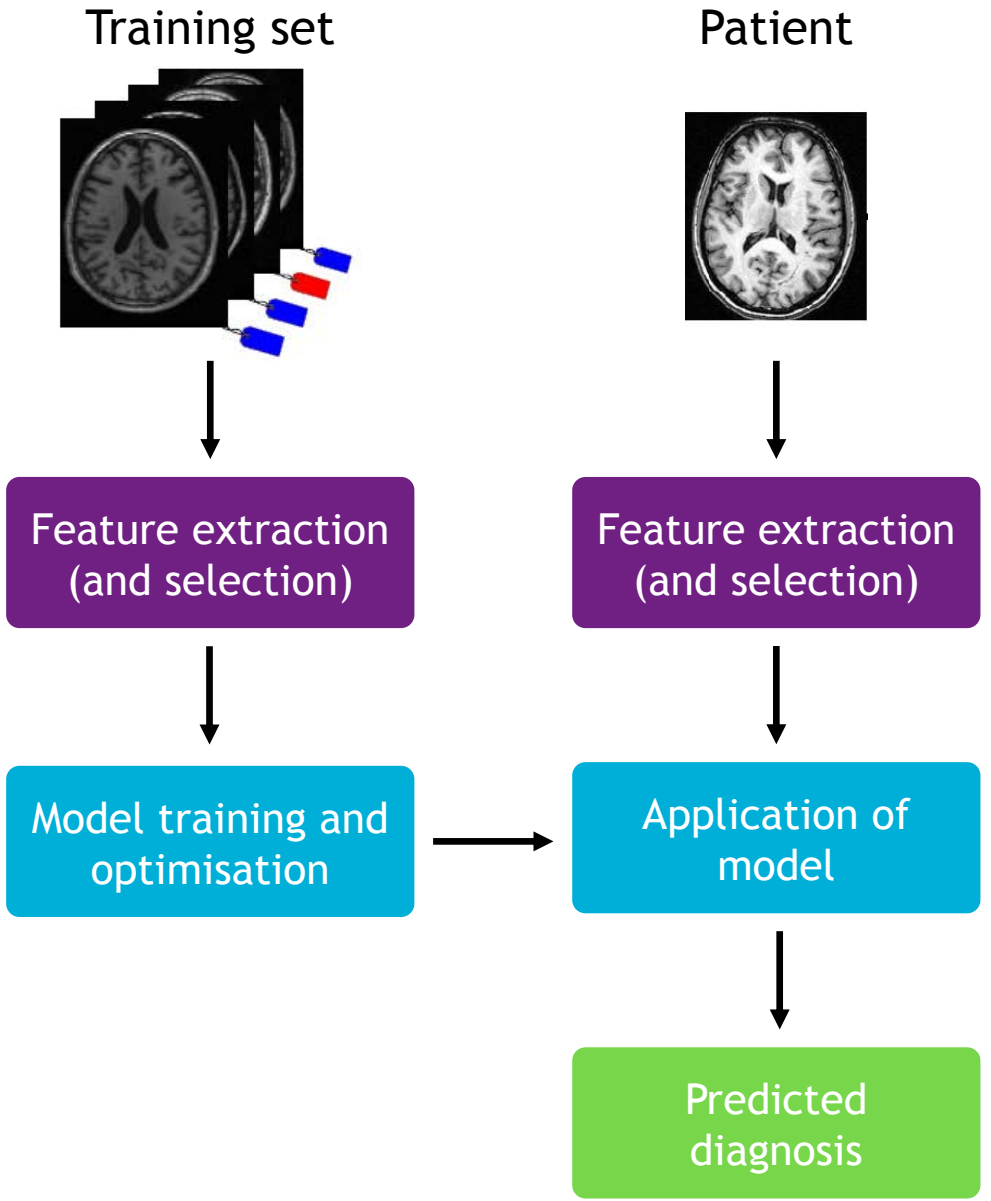
ML/DL for Alzheimer's diagnosis & prognosis

A very active field of research



DL for Alzheimer's diagnosis & prognosis

- Publicly accessible data
- Data organisation
- Image pre-processing
- Network types
- Network building blocks



Publicly accessible data

Dementia

- Image and Data Archive (https://ida.loni.usc.edu)

- Open Access Series of Imaging Studies (www.oasis-brains.org)

Other conditions

- BraTS (http:/braintumorsegmentation.org)
- IXI (https://brain-development.org/ixi-dataset)
- etc.

Data organisation and curation


```
094_S_4089
...
├── Accelerated_SAG_IR-SPGR
├── AV45_Coreg,_Avg,_Standardized_Image_and_Voxel_Size
...
├── Average_DC
├── Axial_DTI
├── Axial_FLAIR
├── Axial_T2_Star
├── Calibration_Scan
├── Coreg,_Avg,_Standardized_Image_and_Voxel_Size
...
├── Eddy_current_corrected_image
├── EPI_current_corrected_image
├── Fractional_Ansio.
├── HarP_135_final_release_2015
├── HHP_6_DOF_AC-PC_registered_MPRAGE
├── MT1__GradWarp__N3m
├── Sag_IR-SPGR
│   ├── 2011-06-29_14_37_16.0
│   ├── 2011-10-18_12_15_56.0
│   │   └── S125692
│   │       ├── ADNI_094_S_4089_MR_Sag_IR-SPGR__br_raw_20111019095510271_80_S125692_I261478.dcm
│   │       ├── ADNI_094_S_4089_MR_Sag_IR-SPGR__br_raw_20111019095512256_62_S125692_I261478.dcm
│   │       ├── ...
│   ├── 2011-12-14_15_53_24.0
│   ├── 2012-08-15_14_00_36.0
│   └── 2013-09-25_14_14_23.0
├── Sag_IR-SPGR_REPEAT
├── Spatially_Normalized,_Masked_and_N3_corrected_T1_image
└── T2-weighted_trace
```

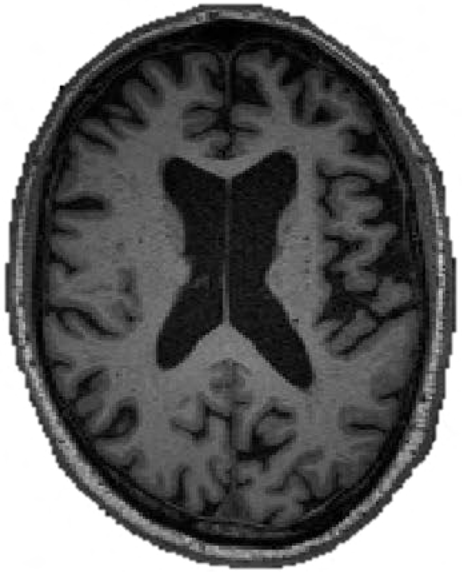
```
sub-ADNI094S4089
├── ses-M00
│   ├── anat
│   │   └── sub-ADNI094S4089_ses-M00_T1w.nii.gz
│   └── dwi
│       ├── sub-ADNI094S4089_ses-M00_acq-axial_dwi.bval
│       ├── sub-ADNI094S4089_ses-M00_acq-axial_dwi.bvec
│       └── sub-ADNI094S4089_ses-M00_acq-axial_dwi.nii.gz
│   ├── pet
│   │   ├── sub-ADNI094S4089_ses-M00_task-rest_acq-av45_pet.nii.gz
│   │   └── sub-ADNI094S4089_ses-M00_task-rest_acq-fdg_pet.nii.gz
│   └── sub-ADNI094S4089_ses-M00_scans.tsv
├── ses-M03
├── ses-M12
└── ses-M24
```

BIDS
BRAIN IMAGING DATA STRUCTURE

http://bids.neuroimaging.io
Gorgolewski et al., Nature Scientific Data, 2016

Image preprocessing

Original T1-weighted MR image



Correction of intensity inhomogeneities + affine registration to template

Non-rigid registration to template + skull stripping

Grey matter segmentation

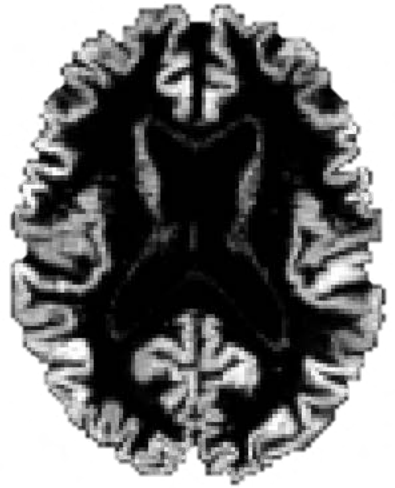


Image preprocessing

Statistical Parametric Mapping (SPM)

- www.fil.ion.ucl.ac.uk/spm
- **Modalities:** Structural and functional MRI, PET, SPECT, EEG, MEG
- **Features:** preprocessing, modelling, statistical inference, voxel-based morphometry, connectivity analysis

Frackowiak, Friston, Frith, Dolan, and Mazziotta, editors. Human Brain Function. Academic Press USA, 1997

FreeSurfer

- https://surfer.nmr.mgh.harvard.edu
- **Modalities:** Structural, functional, diffusion MRI
- **Features:** skullstripping, registration, cortical surface reconstruction, segmentation, longitudinal processing, fMRI analysis, tractography

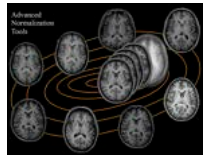
Fischl, NeuroImage, 2012

FMRIB Software Library (FSL)

- https://fsl.fmrib.ox.ac.uk
- **Modalities:** Structural, functional, diffusion MRI
- **Features:** brain extraction, segmentation, registration, tractography, longitudinal analysis, statistical analysis

Jenkinson et al., NeuroImage, 2012

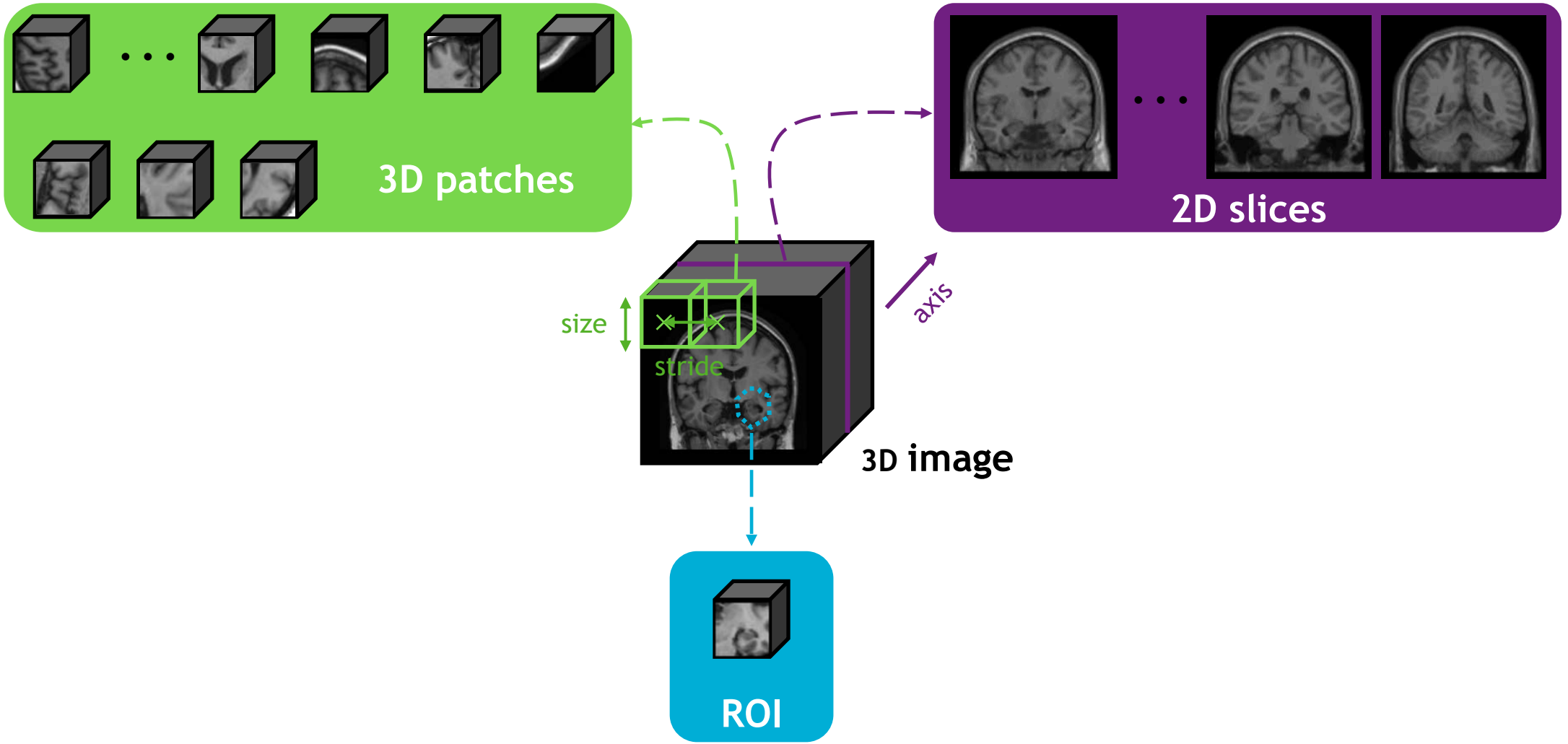
Advanced Normalization Tools (ANTs)



- http://stnava.github.io/ANTs
- **Modalities:** Structural, functional, diffusion MRI, PET
- **Features:** bias field correction, registration, segmentation, cortical thickness estimation

Avants et al., Frontiers in Neuroinformatics, 2014

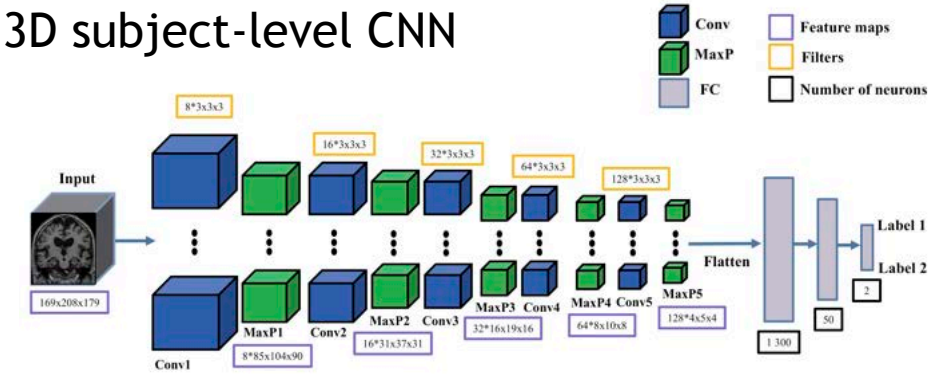
Image preprocessing



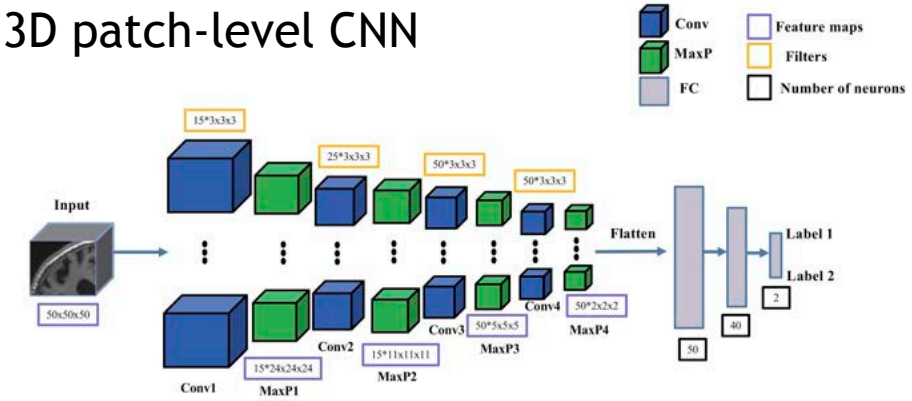
Classifiers

Convolutional neural networks

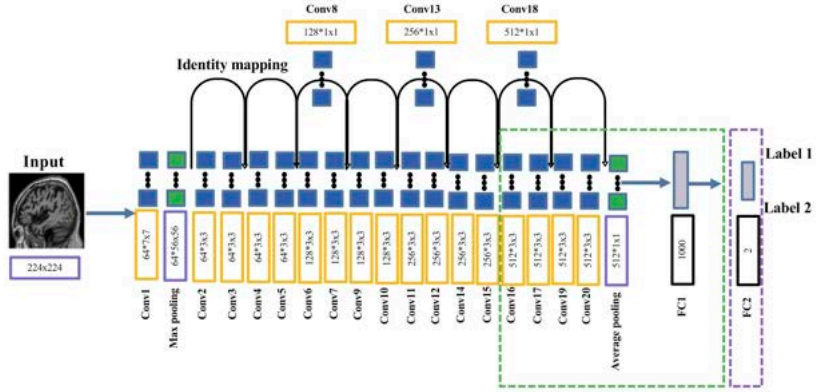
3D subject-level CNN



3D patch-level CNN

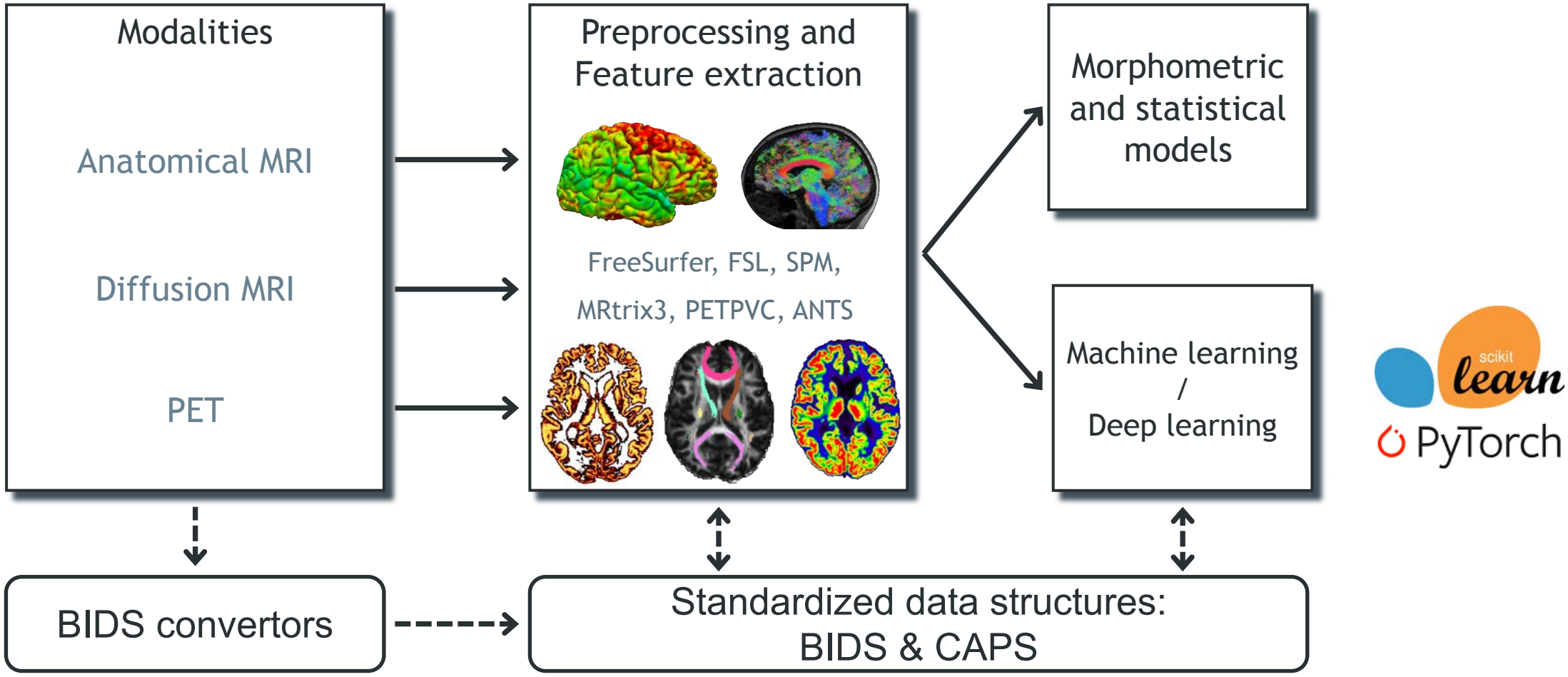


2D slice-level CNN



Wen, Thibeau-Sutre et al., Medical Image Analysis, 2020

Software platform for clinical neuroimaging studies



ClinicaDL

Clinica + PyTorch =

ClinicaDL

Prepare your imaging data

- `clinicadl preprocessing` - Preprocessing pipelines
  - `t1-linear` - Linear processing of T1w MR images: affine registration to the MNI standard space
  - `t1-extensive` - 'Extensive' processing of T1w MR images: non linear registration to the MNI standard space
- `clinicadl quality_check` - Quality control of preprocessed data: use a pretrained network [Fonov et al., 2018] to classify adequately registered images.
- `clinicadl extract` - Prepare input data for deep learning with PyTorch
- `clinicadl quality_check` - Evaluate registration quality

Train & test your classifier

- `clinicadl train` - Train with your data and create a model
- `clinicadl classify` - Classify one image or a list of images with your previously trained model

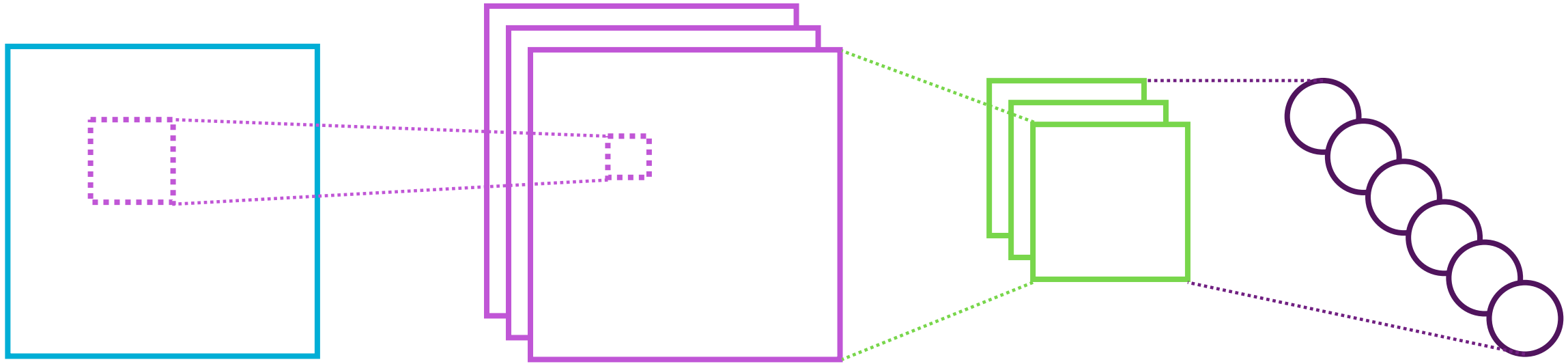
Utilitaries

- `clinicadl generate` - Generate synthetic data for functional tests
- `clinicadl tsvtool` - Handle TSV files for metadata processing and data splits

https://github.com/aramis-lab/AD-DL

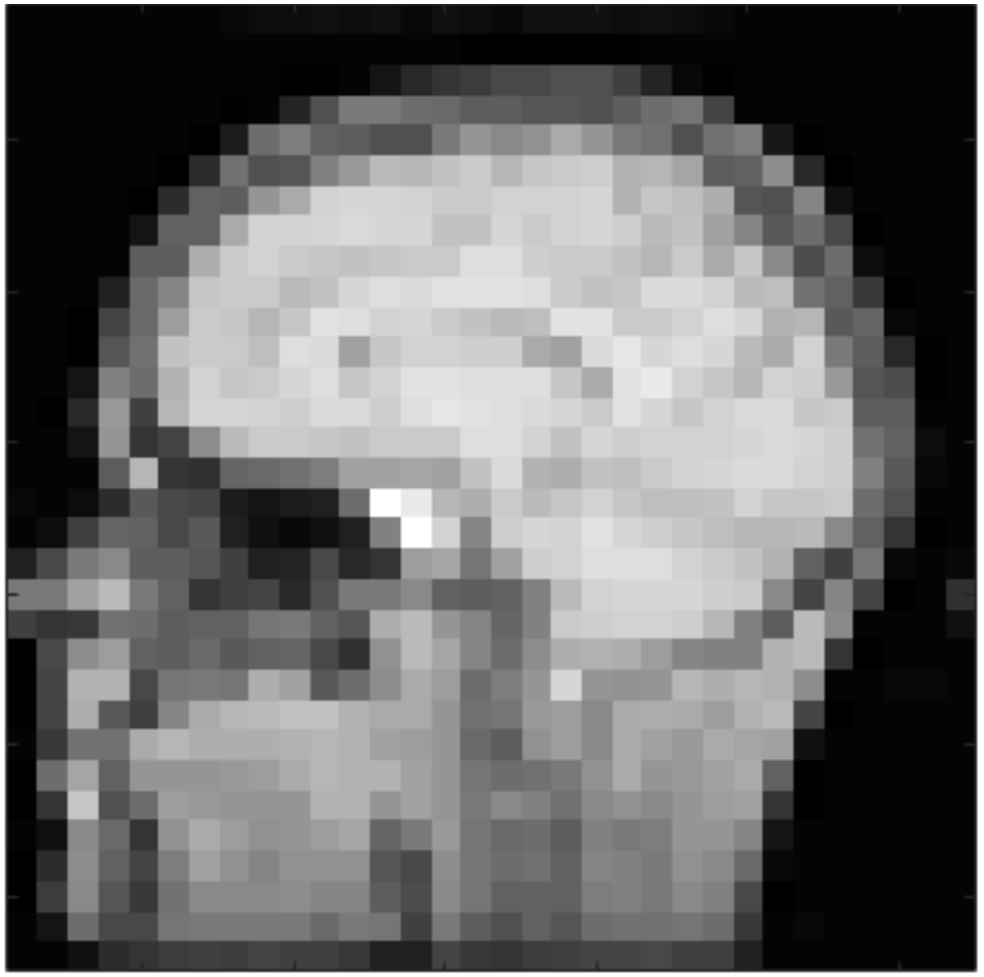
Convolutional neural networks

CNNs for classification



Using an image as input of a neural network

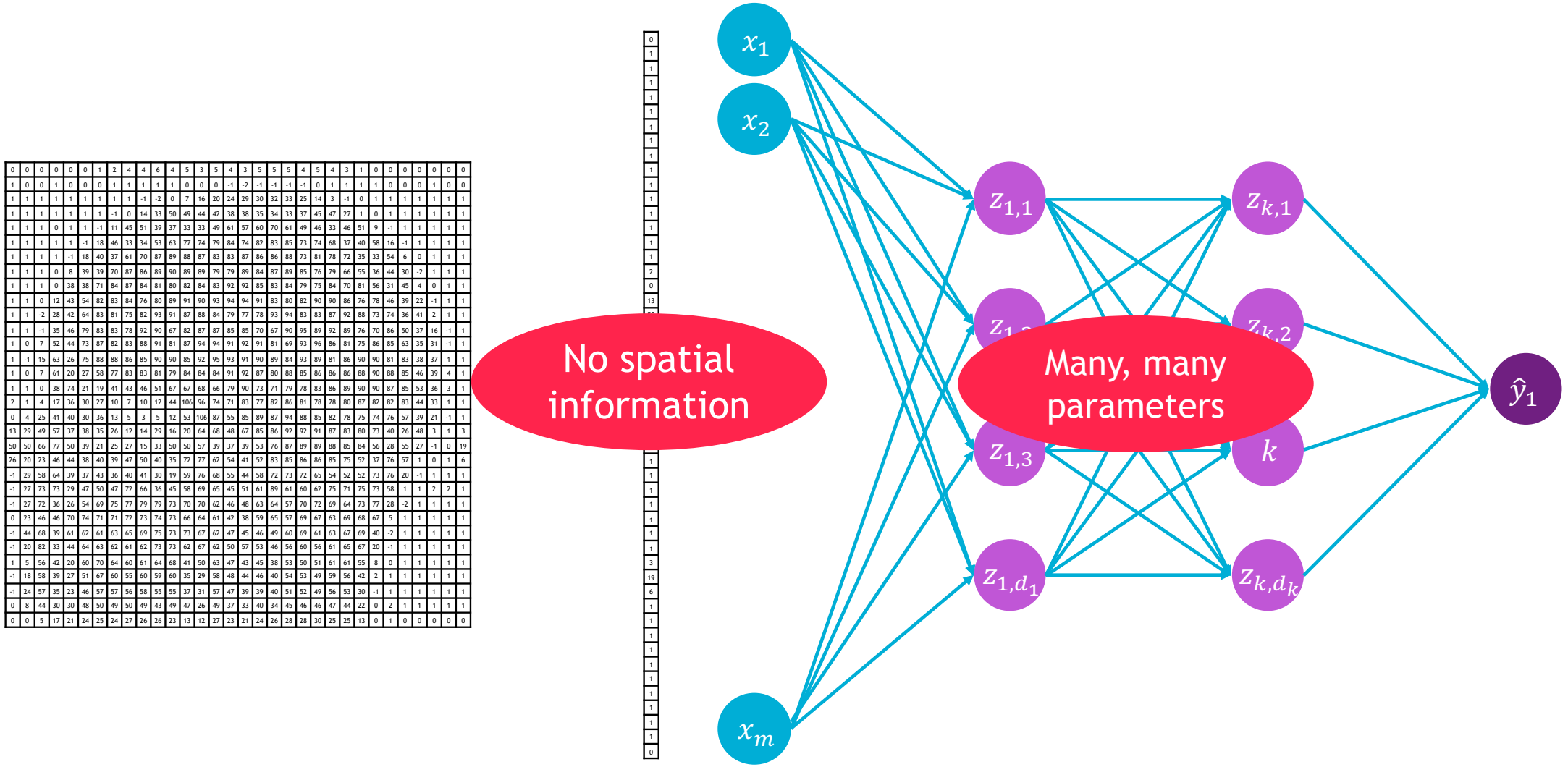
Image = matrix of numbers



0	0	0	0	0	0	1	2	4	4	6	4	5	3	5	4	3	5	5	5	4	5	4	3	1	0	0	0	0	0	0	0
1	0	0	1	0	0	0	1	1	1	1	1	0	0	0	-1	-2	-1	-1	-1	-1	0	1	1	1	1	0	0	0	1	0	0
1	1	1	1	1	1	1	1	1	-1	-2	0	7	16	20	24	29	30	32	33	25	14	3	-1	0	1	1	1	1	1	1	1
1	1	1	1	1	1	1	-1	0	14	33	50	49	44	42	38	38	35	34	33	37	45	47	27	1	0	1	1	1	1	1	1
1	1	1	0	1	1	-1	11	45	51	39	37	33	33	49	61	57	60	70	61	49	46	33	46	51	9	-1	1	1	1	1	1
1	1	1	1	1	-1	18	46	33	34	53	63	77	74	79	84	74	82	83	85	73	74	68	37	40	58	16	-1	1	1	1	1
1	1	1	1	-1	18	40	37	61	70	87	89	88	87	83	83	87	86	86	88	73	81	78	72	35	33	54	6	0	1	1	1
1	1	1	0	8	39	39	70	87	86	89	90	89	89	79	79	89	84	87	89	85	76	79	66	55	36	44	30	-2	1	1	1
1	1	1	0	38	38	71	84	87	84	81	80	82	84	83	92	92	85	83	84	79	75	84	70	81	56	31	45	4	0	1	1
1	1	0	12	43	54	82	83	84	76	80	89	91	90	93	94	94	91	83	80	82	90	90	86	76	78	46	39	22	-1	1	1
1	1	-2	28	42	64	83	81	75	82	93	91	87	88	84	79	77	78	93	94	83	83	87	92	88	73	74	36	41	2	1	1
1	1	-1	35	46	79	83	83	78	92	90	67	82	87	87	85	85	70	67	90	95	89	92	89	76	70	86	50	37	16	-1	1
1	0	7	52	44	73	87	82	83	88	91	81	87	94	94	91	92	91	81	69	93	96	86	81	75	86	85	63	35	31	-1	1
1	-1	15	63	26	75	88	88	86	85	90	90	85	92	95	93	91	90	89	84	93	89	81	86	90	90	81	83	38	37	1	1
1	0	7	61	20	27	58	77	83	83	81	79	84	84	84	91	92	87	80	88	85	86	86	86	88	90	88	85	46	39	4	1
1	1	0	38	74	21	19	41	43	46	51	67	67	68	66	79	90	73	71	79	78	83	86	89	90	90	87	85	53	36	3	1
2	1	4	17	36	30	27	10	7	10	12	44	106	96	74	71	83	77	82	86	81	78	78	80	87	82	82	83	44	33	1	1
0	4	25	41	40	30	36	13	5	3	5	12	53	106	87	55	85	89	87	94	88	85	82	78	75	74	76	57	39	21	-1	1
13	29	49	57	37	38	35	26	12	14	29	16	20	64	68	48	67	85	86	92	92	91	87	83	80	73	40	26	48	3	1	3
50	50	66	77	50	39	21	25	27	15	33	50	50	57	39	37	39	53	76	87	89	89	88	85	84	56	28	55	27	-1	0	19
26	20	23	46	44	38	40	39	47	50	40	35	72	77	62	54	41	52	83	85	86	86	85	75	52	37	76	57	1	0	1	6
-1	29	58	64	39	37	43	36	40	41	30	19	59	76	68	55	44	58	72	73	72	65	54	52	52	73	76	20	-1	1	1	1
-1	27	73	73	29	47	50	47	72	66	36	45	58	69	65	45	51	61	89	61	60	62	75	71	75	73	58	1	1	2	2	1
-1	27	72	36	26	54	69	75	77	79	79	73	70	70	62	46	48	63	64	57	70	72	69	64	73	77	28	-2	1	1	1	1
0	23	46	46	70	74	71	71	72	73	74	73	66	64	61	42	38	59	65	57	69	67	63	69	68	67	5	1	1	1	1	1
-1	44	68	39	61	62	61	63	65	69	75	73	73	67	62	47	45	46	49	60	69	61	63	67	69	40	-2	1	1	1	1	1
-1	20	82	33	44	64	63	62	61	62	73	73	62	67	62	50	57	53	46	56	60	56	61	65	67	20	-1	1	1	1	1	1
1	5	56	42	20	60	70	64	60	61	64	68	41	50	63	47	43	45	38	53	50	51	61	61	55	8	0	1	1	1	1	1
-1	18	58	39	27	51	67	60	55	60	59	60	35	29	58	48	44	46	40	54	53	49	59	56	42	2	1	1	1	1	1	1
-1	24	57	35	23	46	57	57	56	58	55	55	37	31	57	47	39	39	40	51	52	49	56	53	30	-1	1	1	1	1	1	1
0	8	44	30	30	48	50	49	50	49	43	49	47	26	49	37	33	40	34	45	46	46	47	44	22	0	2	1	1	1	1	1
0	0	5	17	21	24	25	24	27	26	26	23	13	12	27	23	21	24	26	28	28	30	25	25	13	0	1	0	0	0	0	0

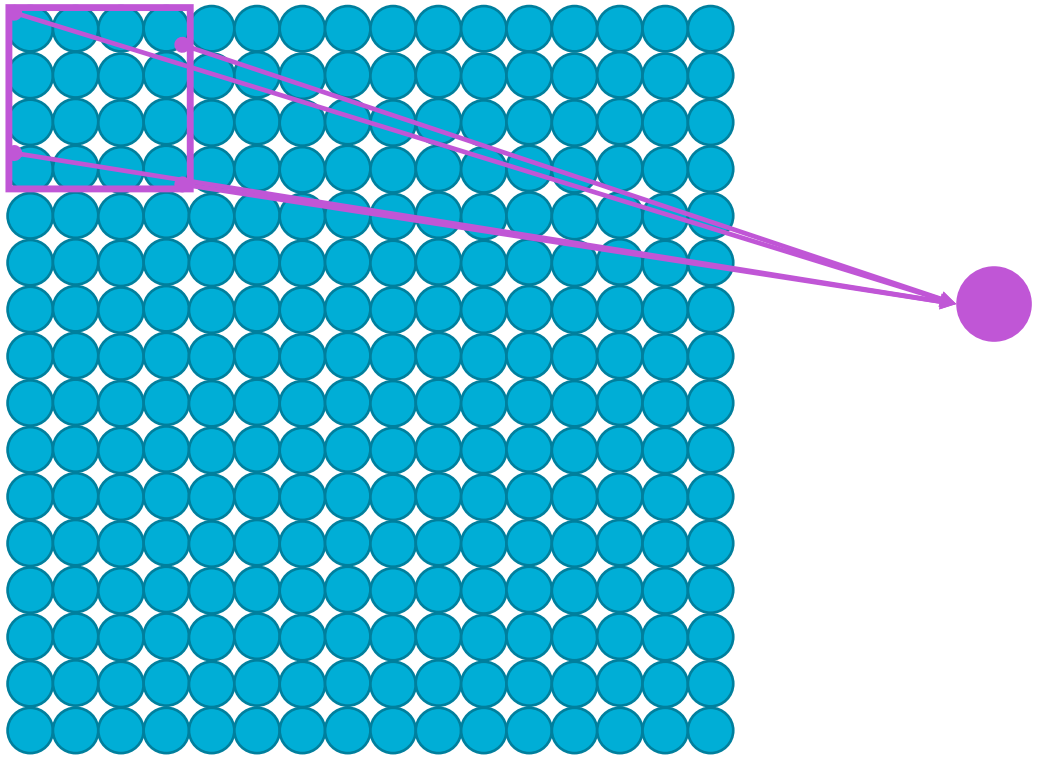
Using an image as input of a neural network

Fully connected neural network



Using an image as input of a neural network

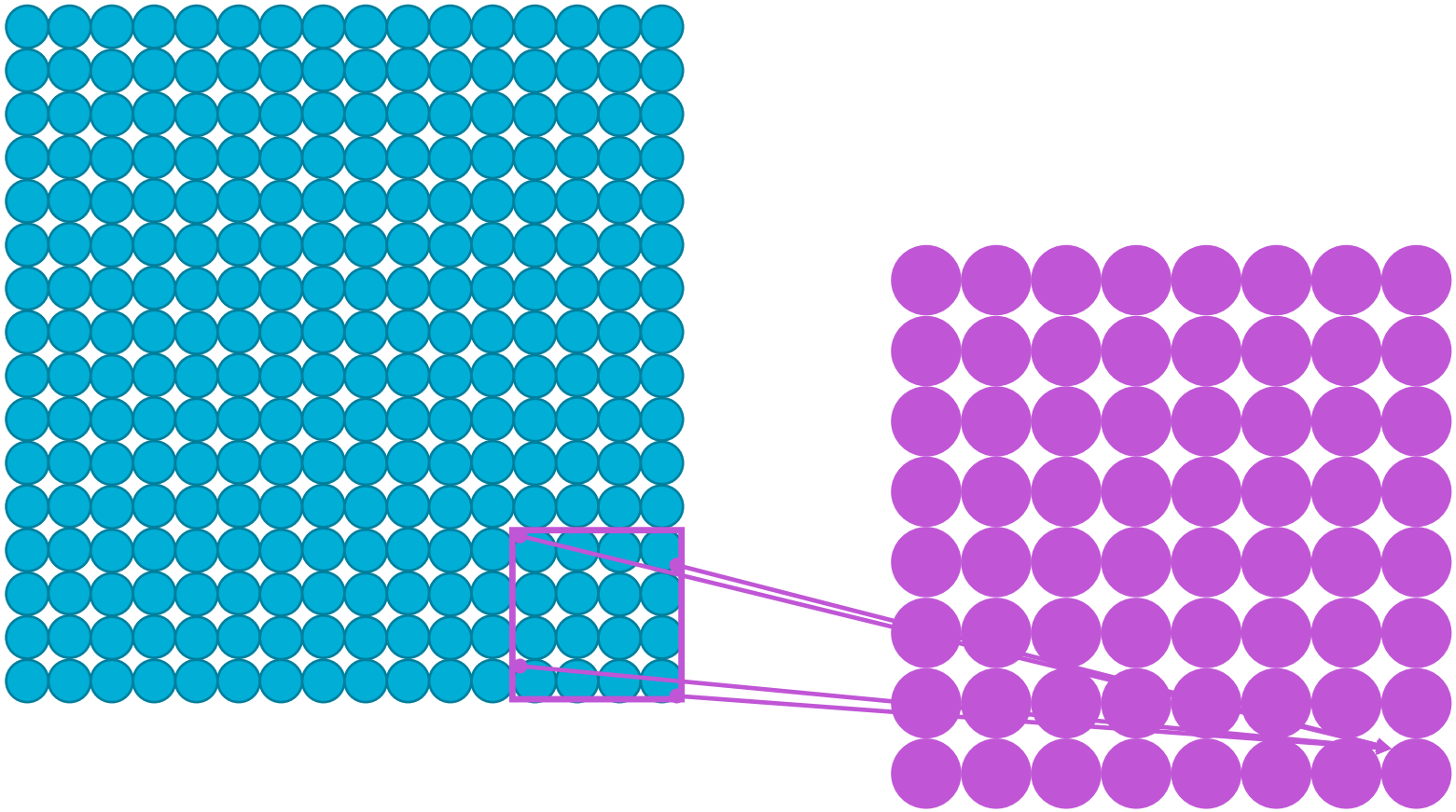
Using spatial features



Idea: connect patches of input to neurons in hidden layer

Using an image as input of a neural network

Using spatial features



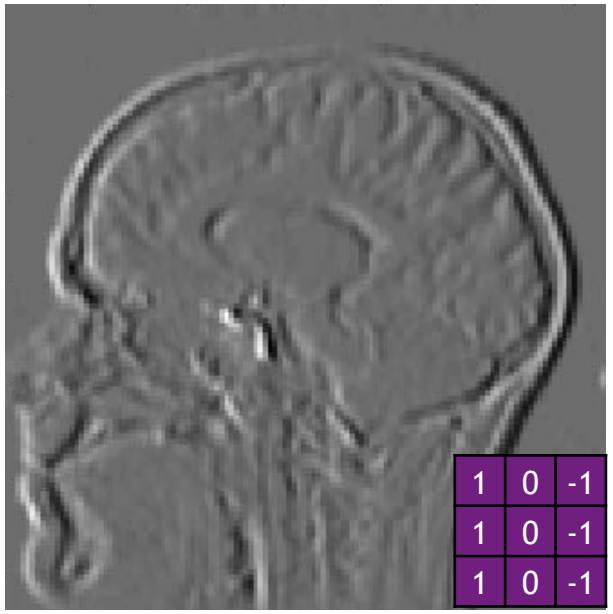
- Slide patch window across input image
- Weight pixels inside the patch
- Apply weighted summation

→ Convolution

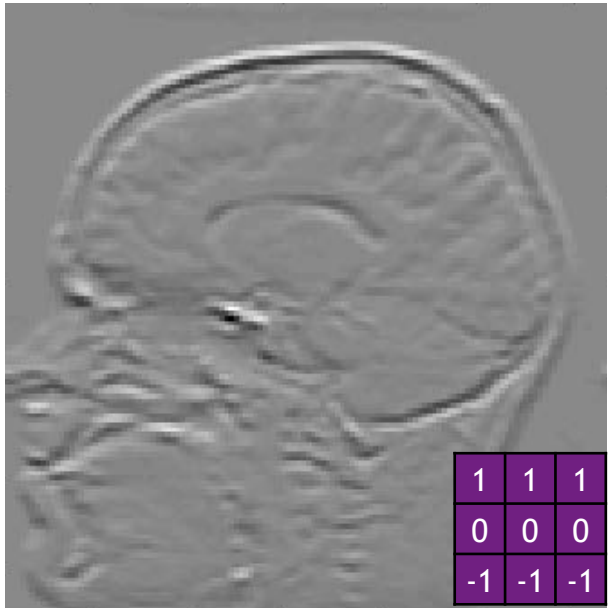
Using an image as input of a neural network

Different filters = different feature maps

Original image



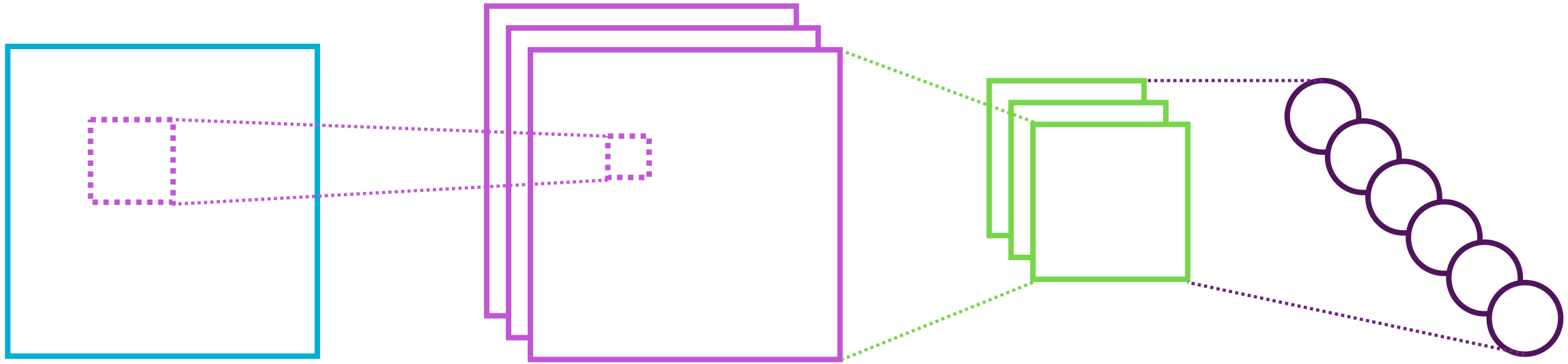
Vertical edge detection



Horizontal edge detection

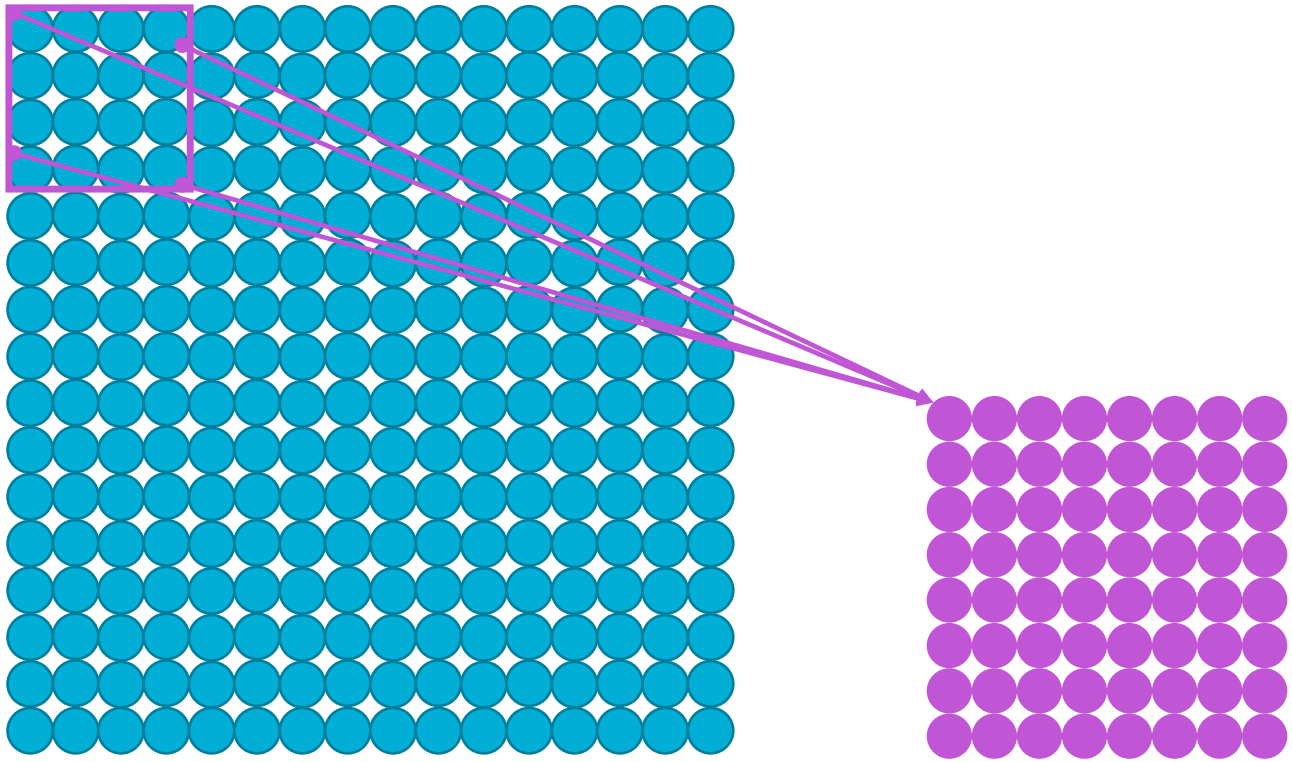
Convolutional neural networks

CNNs for classification



Convolutional neural networks

Convolutional layer



Neuron (p, q) in hidden layer
Filter size 4×4
Weights $w_{i,j}$

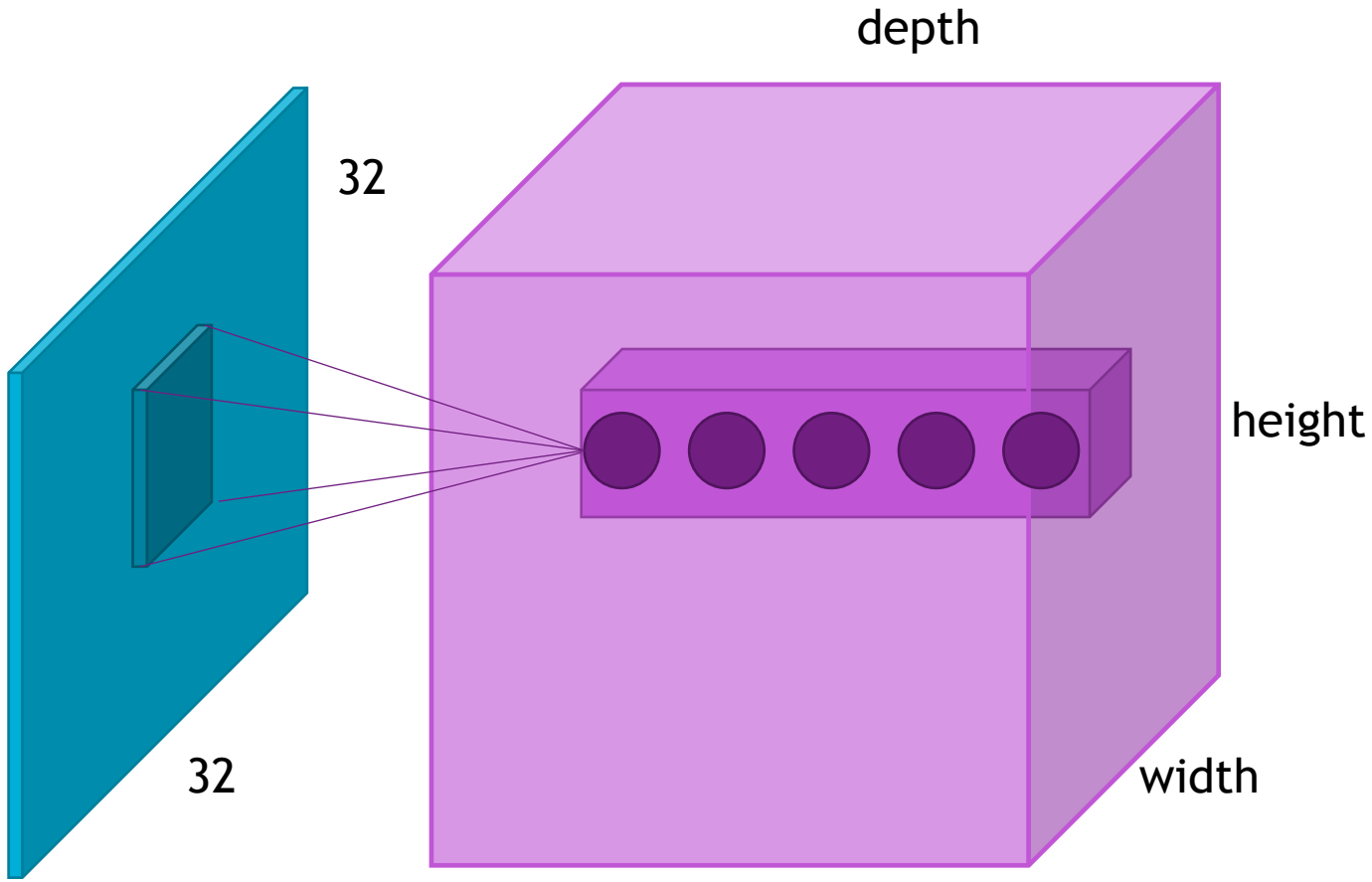
$$h\left(\sum_{i=1}^{4}\sum_{j=1}^{4} w_{i,j} x_{i+p,j+q} + b\right)$$

For a neuron in hidden layer:

- Take inputs from patch
- Compute weighted sum
- Apply bias
- Activate with non-linear function

Convolutional neural networks

Spatial arrangement of output volume



Layer Dimensions:

$h \times w \times d$

h & w = spatial dimensions

d = number of filters

Stride:

Filter step size

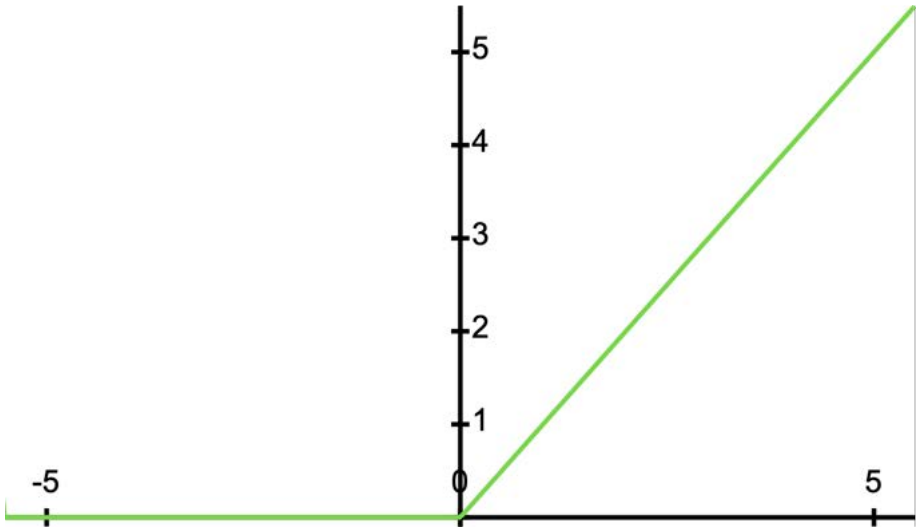
Receptive Field:

Locations in input image that a node is connected to

Convolutional neural networks

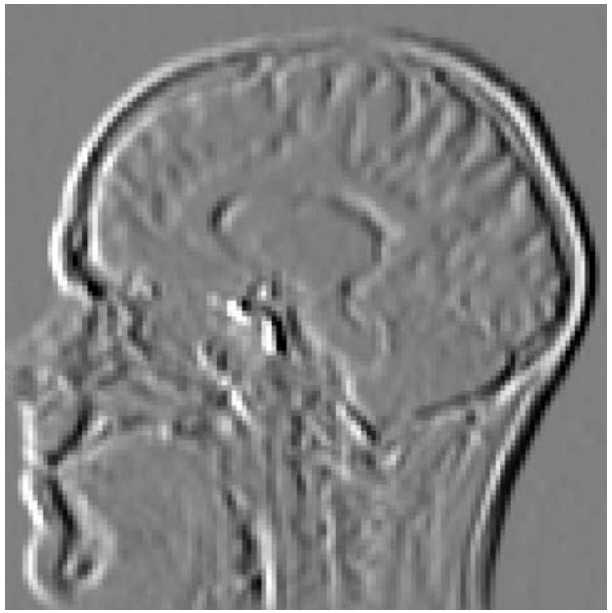
Introducing non-linearity

Rectified linear unit (ReLU)



$$h(z) = \max(0, z)$$

Input feature map



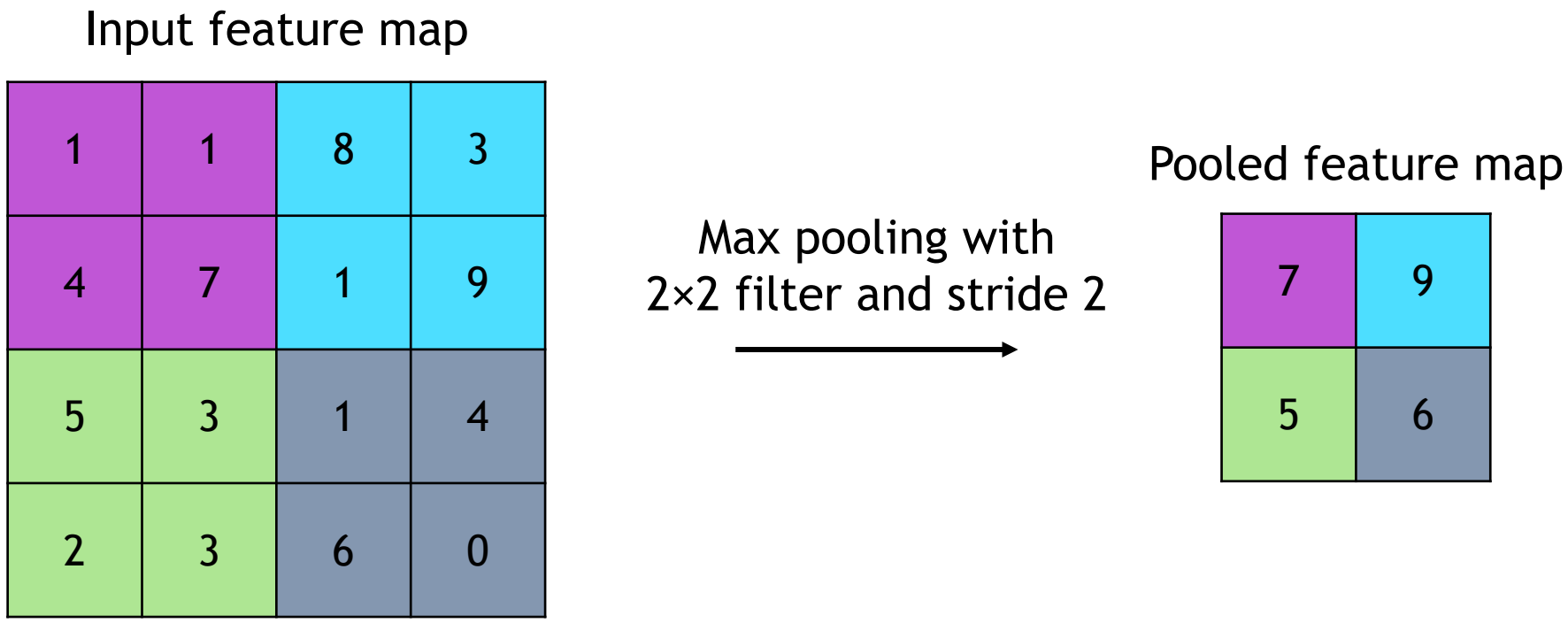
Rectified feature map

ReLU

Black: negative values - White: positive values

Pooling

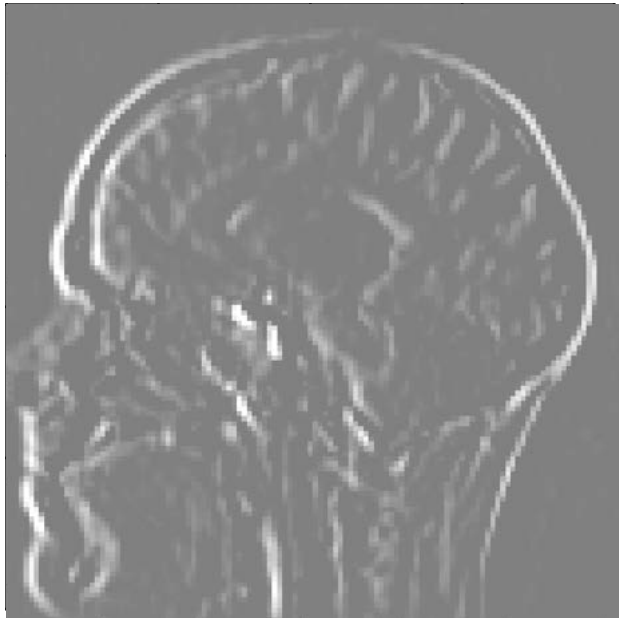
- **Reduce dimensionality while preserving spatial invariance**



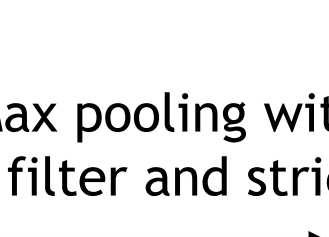
Pooling

- **Reduce dimensionality while preserving spatial invariance**

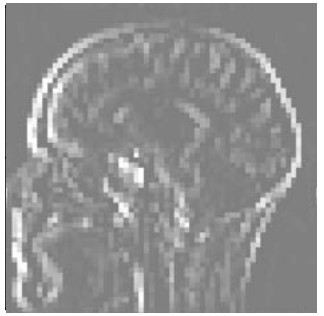
Input feature map



Max pooling with
2×2 filter and stride 2

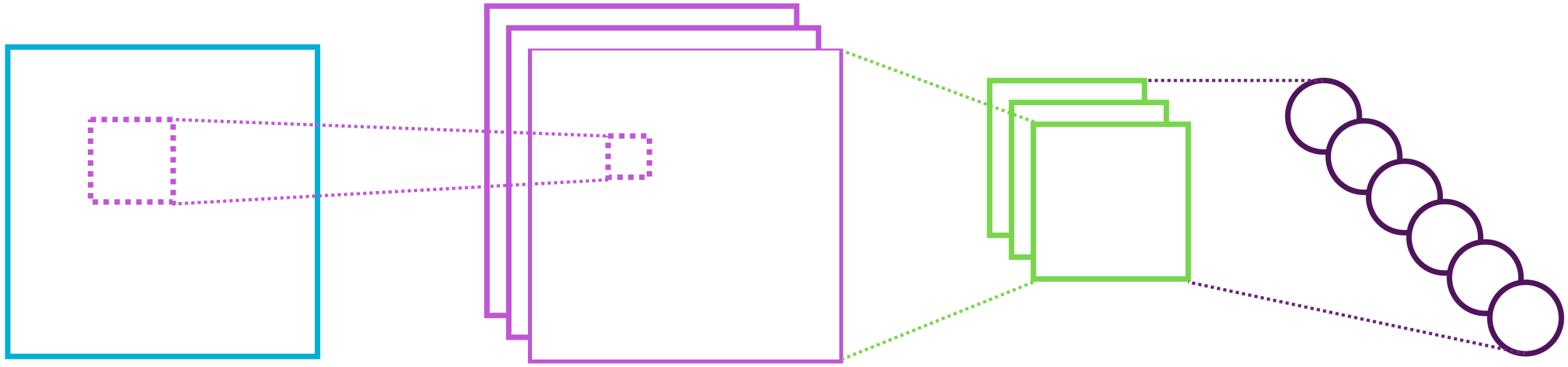


Pooled feature map



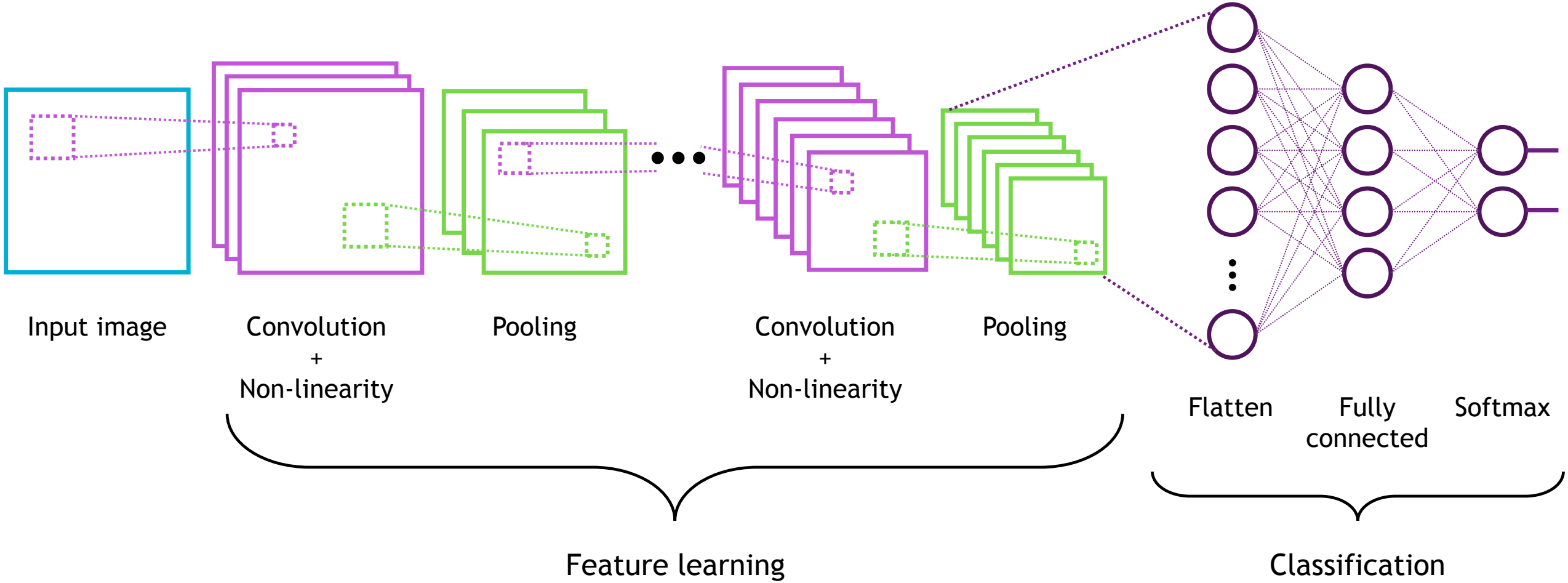
Convolutional neural networks

CNNs for classification



Convolutional neural networks

CNNs for classification



MIT Introduction to Deep Learning (introtodeeplearning.com)

ARAMIS LAB
BRAIN DATA SCIENCE

www.aramislab.fr

www.clinica.run

AI4Health School 2021

Practical session DL4MI

Deep learning for computer-aided diagnosis from images

Application to Alzheimer's disease

https://aramislab.paris.inria.fr/workshops/DL4MI