Image Analysis (02502)

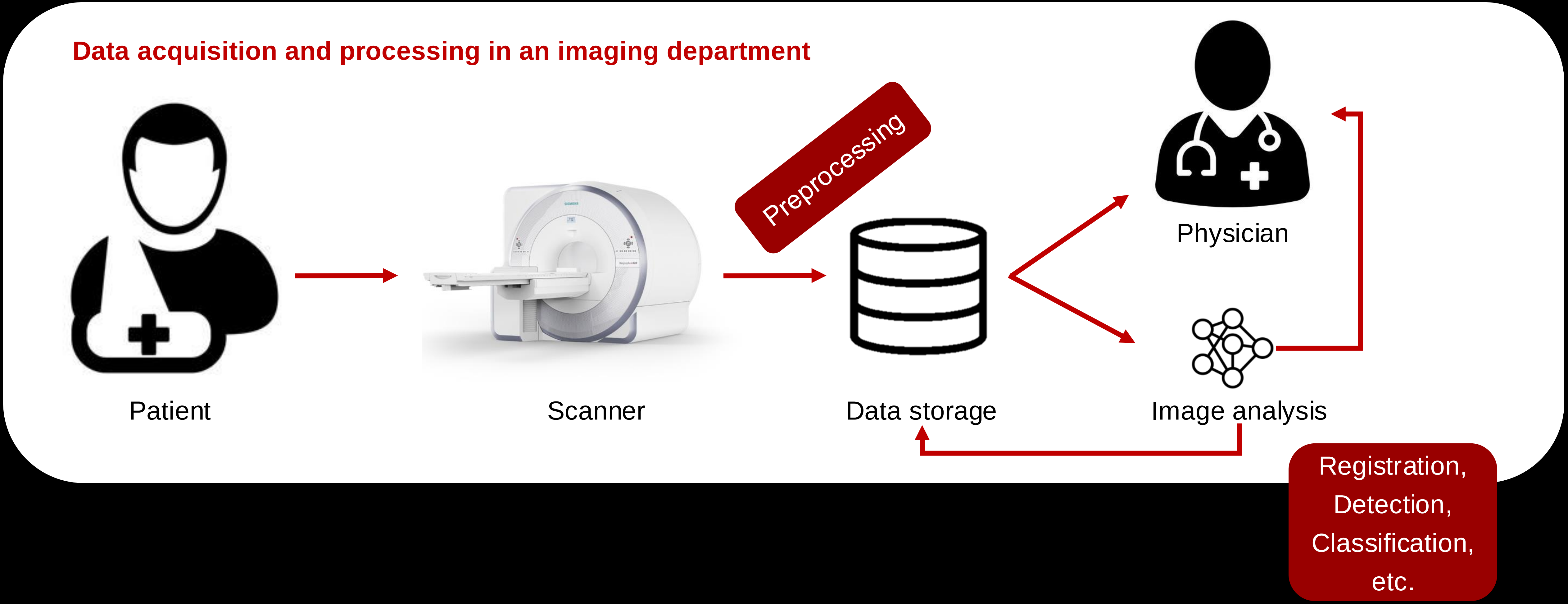
Advanced Topics

Claes Ladefoged, PhD

Claes N. Ladefoged

- MSc from Computer Science KU
- PhD in Medicine from SUND
- Head of AI Research at Rigshospitalet
- Associate Professor, DTU Compute

Overview



Data acquisition and processing in an imaging department

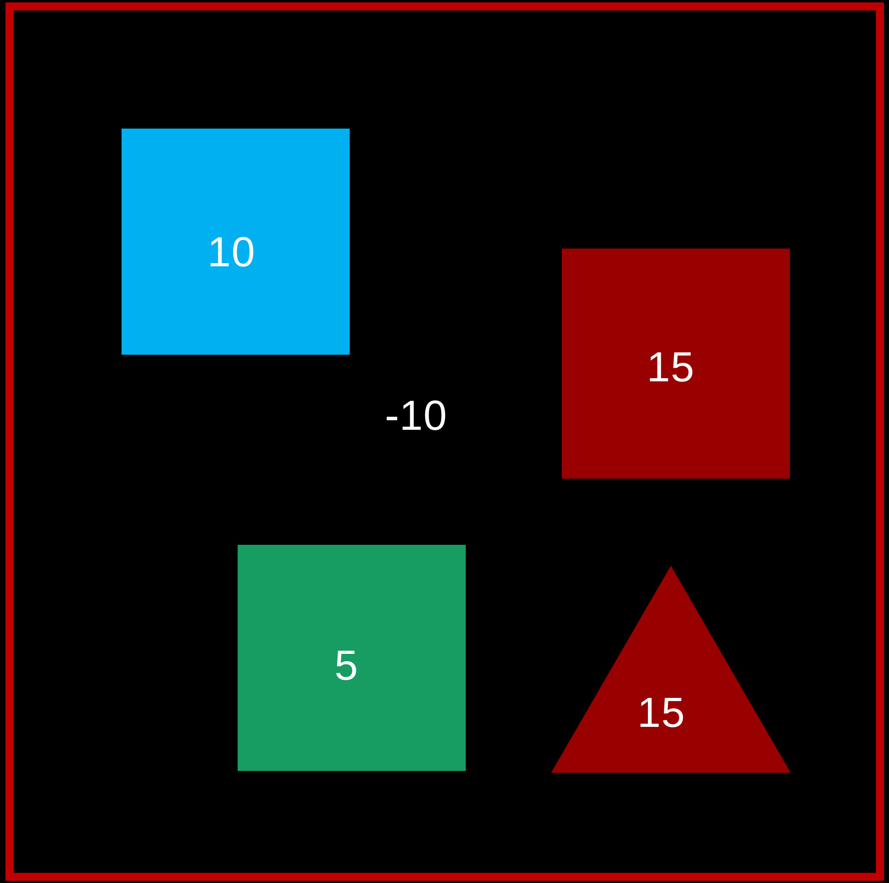
Preprocessing

- Data compression
- Intensity normalization
- Intensity augmentation
- Intensity mapping
- Filtering

Data compression

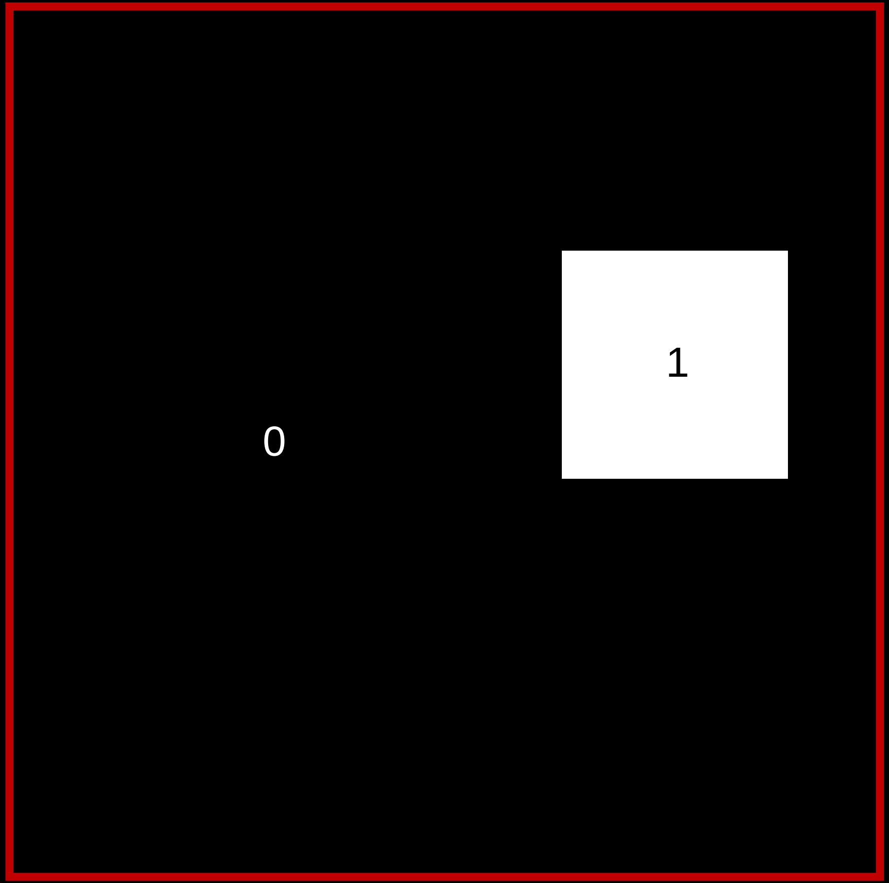
Task:
Store using fewest number of possitive digits

Image



1024 x 1024

Label

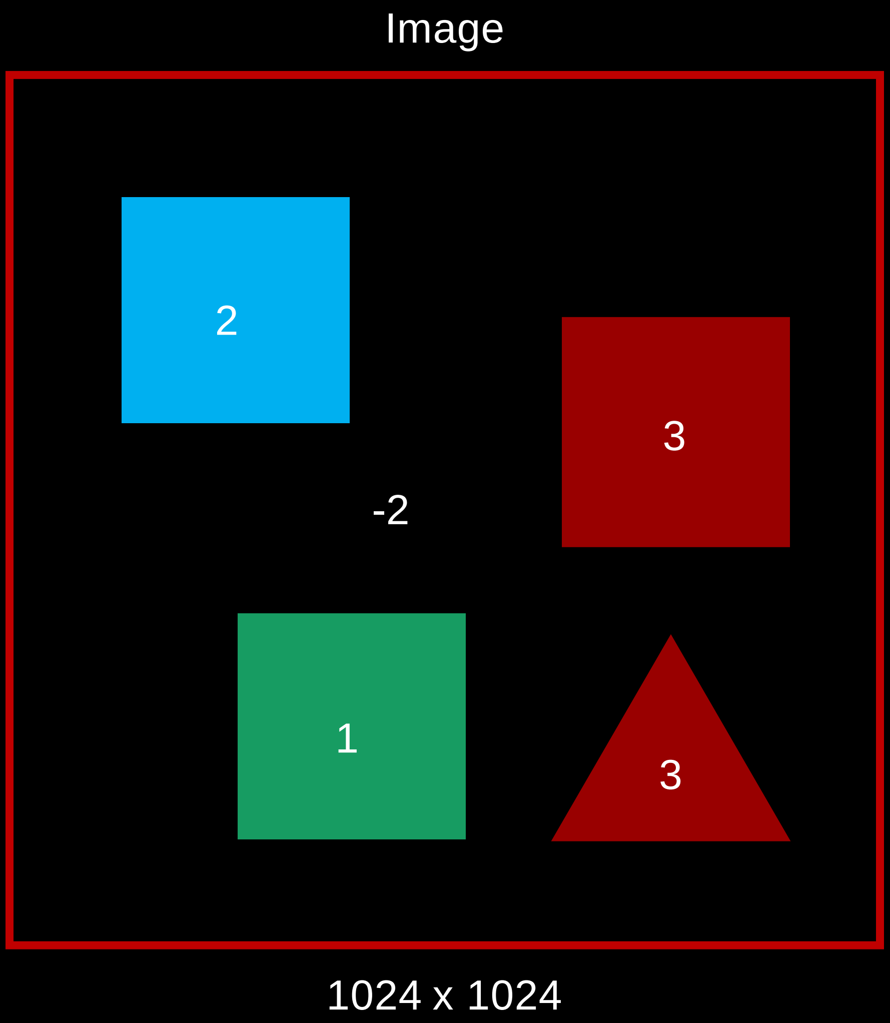


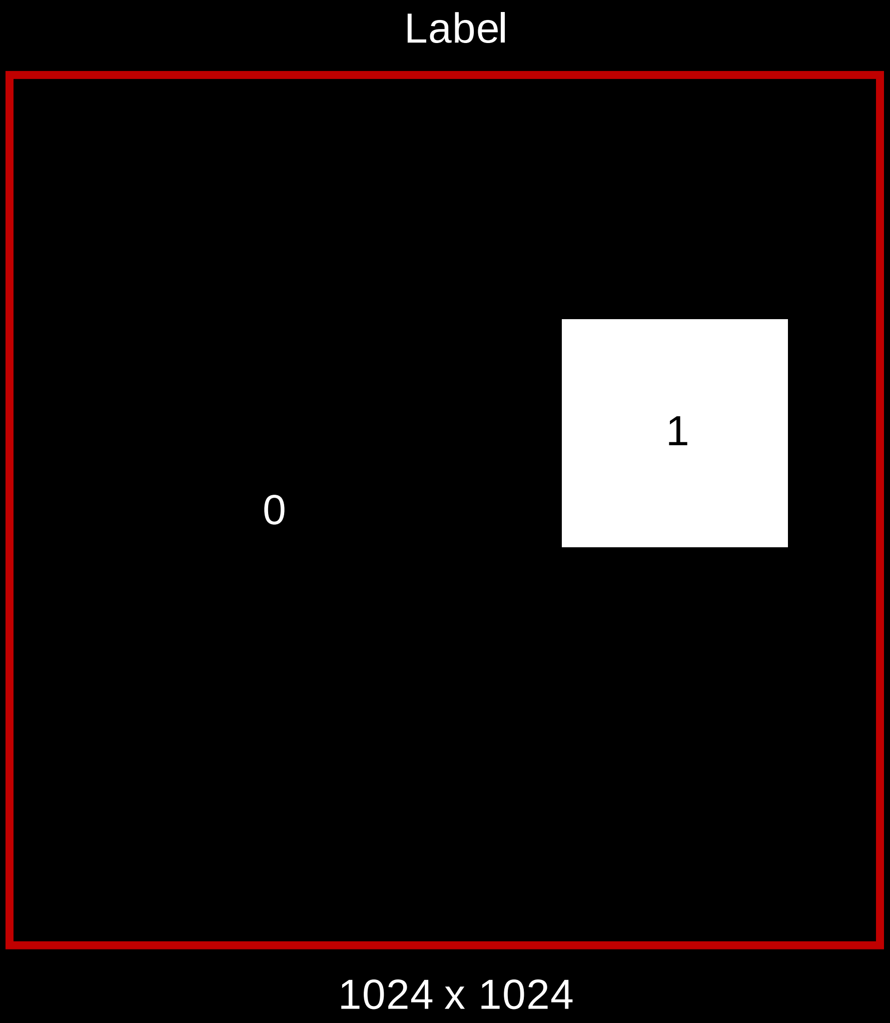
1024 x 1024

Data compression

Task:
Store using fewest number of possitive digits

/ 5



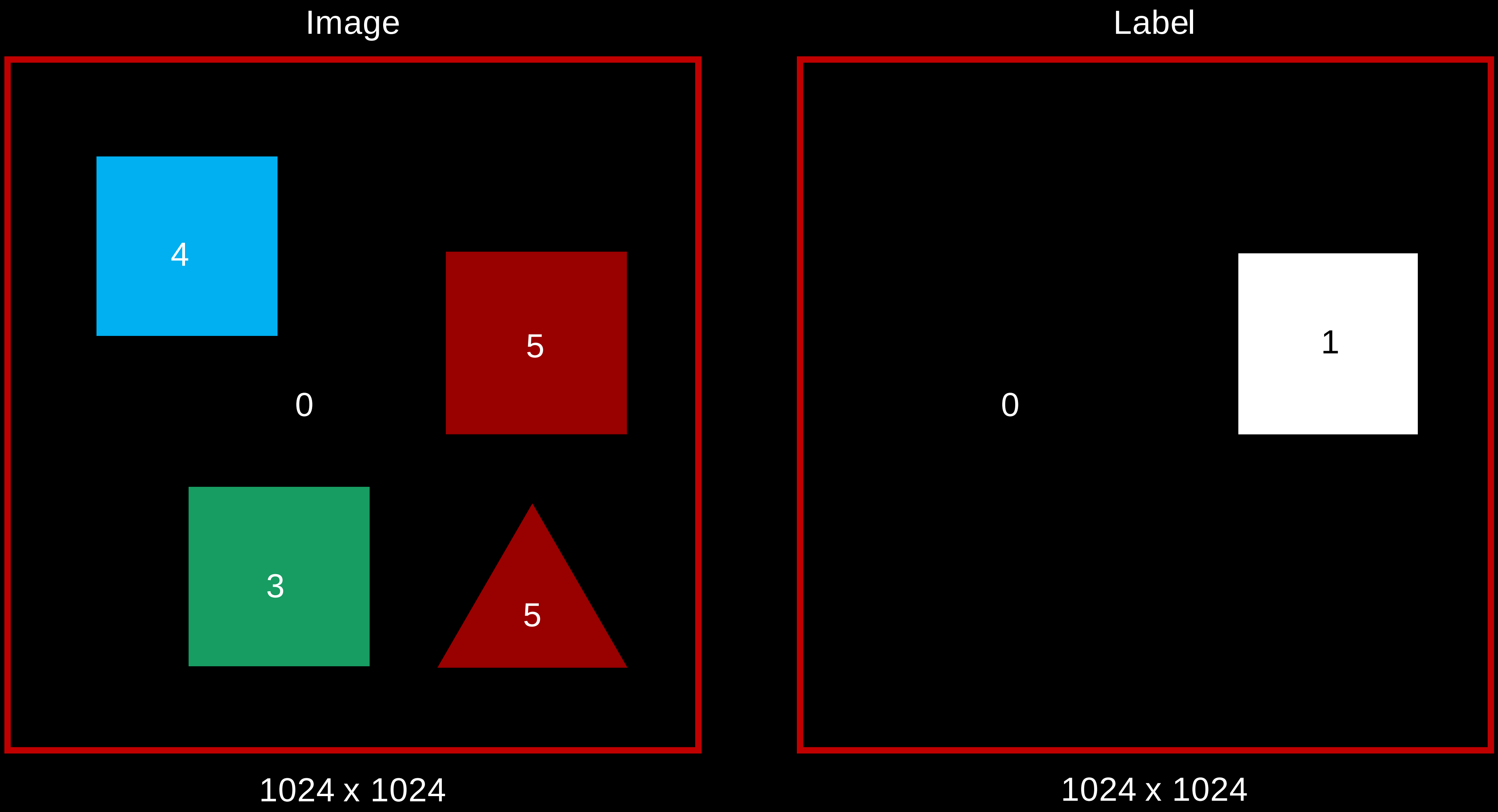


Data compression

Task:
Store using fewest number of possitive digits

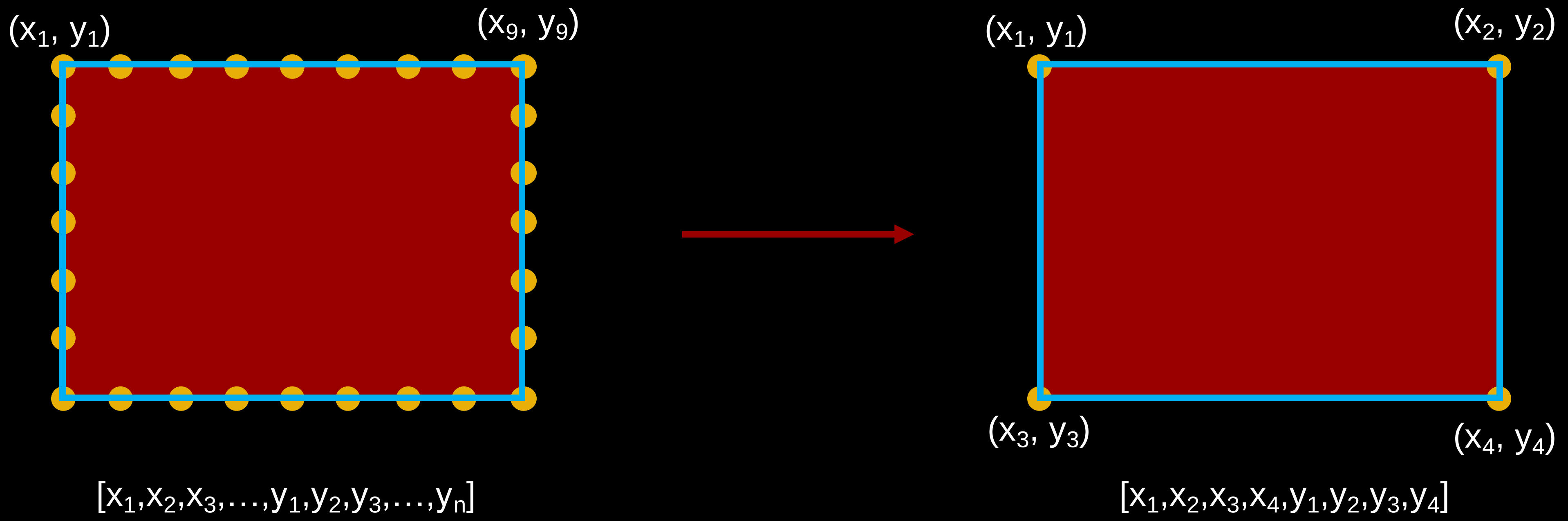
/ 5

+2



Data compression

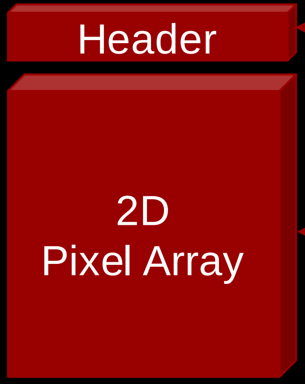
- Representation of outlines


(x_1, y_1)
(x_9, y_9)
$[x_1,x_2,x_3,\ldots,y_1,y_2,y_3,\ldots,y_n]$
(x_1, y_1)
(x_2, y_2)
(x_3, y_3)
(x_4, y_4)
$[x_1,x_2,x_3,x_4,y_1,y_2,y_3,y_4]$

Data compression

- CT values are usually defined in [-1024;3071] HU
- Values are usually stored as unsigned integer
- Large part of the volume is air (-1024 HU)

$$F(x) = ax + b$$



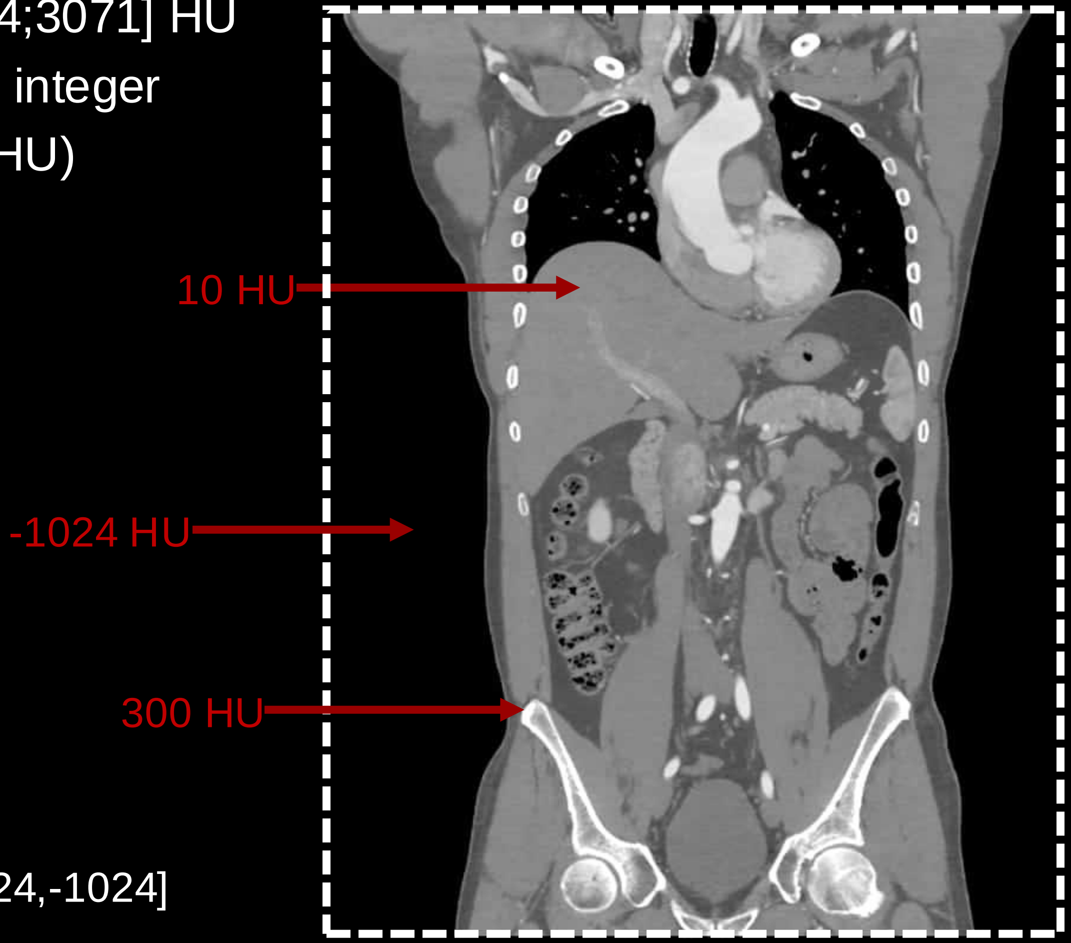
Header

2D
Pixel Array

Slope (a): 1
Intercept (b): -1024

Stored values:
[0,0,0,1034,…,1324,0,0]

Read values:
[-1024,-1024,-1024,10,…,300,-1024,-1024]



Quiz 1

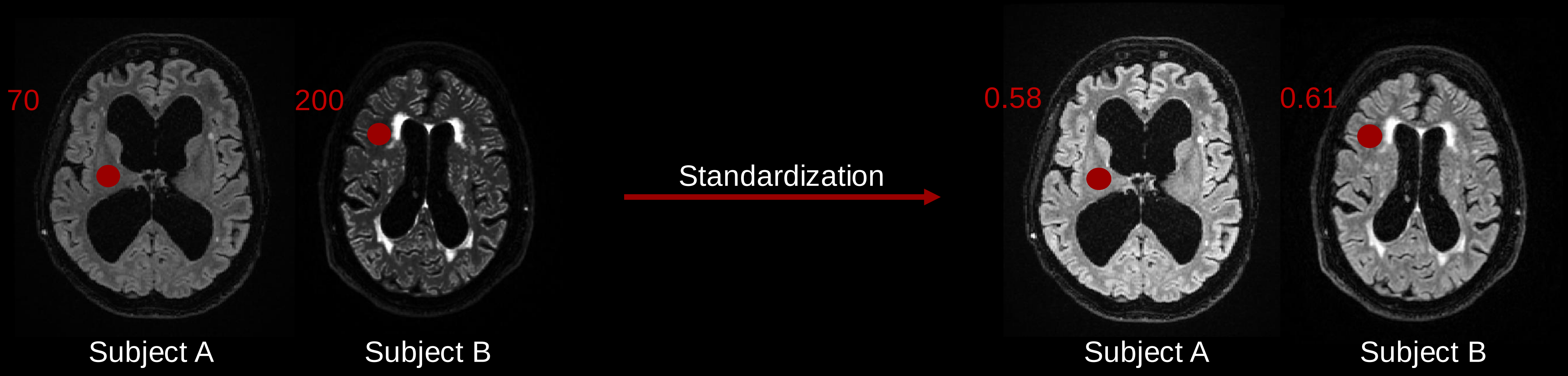
- An image containing values ranging from 0 to 52,427 needs to be stored in DICOM format
- The DICOM file has to be in the type SHORT (max value = 32,767)
- What can the slope and intercept be?
  - Slope 1.4 and intercept 1
  - Slope 1.6 and intercept 0
  - Slope 1 and intercept -19,660

menti.com

Code: **1339 2939**

Intensity normalization

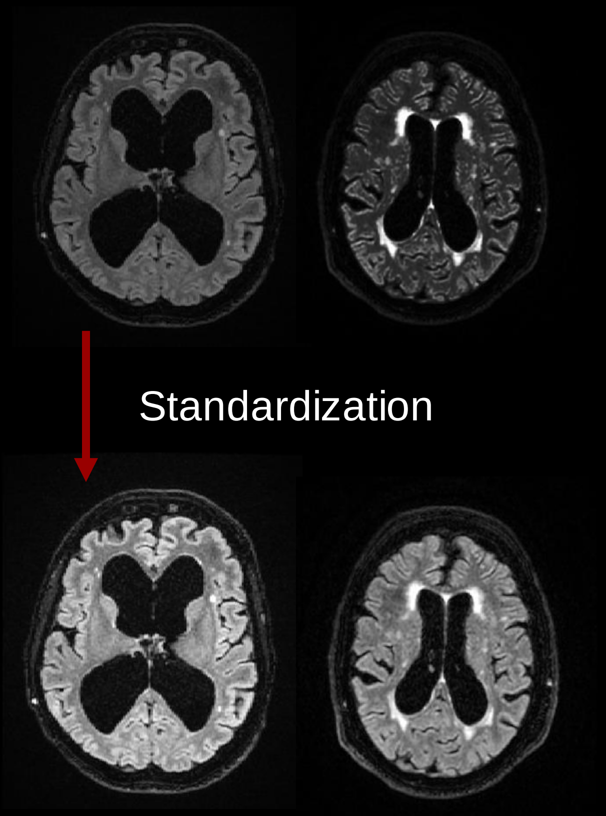
- Conventional MRI intensites (T1-w, T2-w, PD, FLAIR) are acquired in arbitrary units



Quantification analysis:
Incomparable results

Quantification analysis:
Comparable results

Intensity normalization



Standardization

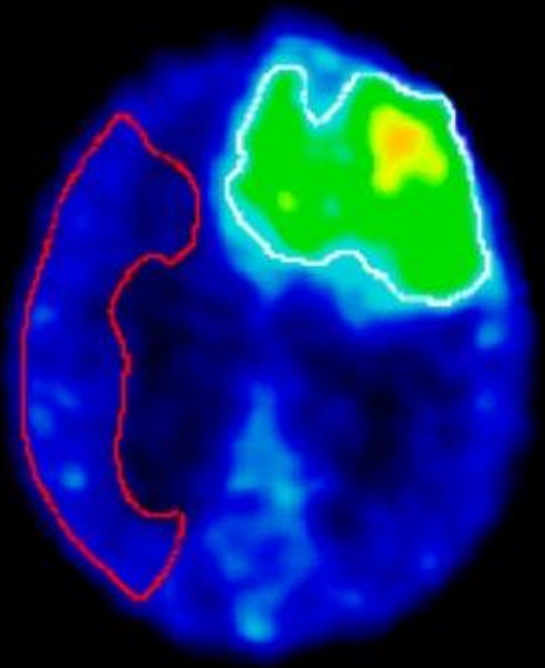
Some available mapping functions:

- Min-max scaling $$g(x,y) = \frac{f(x,y) - v_{min}}{v_{max} - v_{min}}$$
- Histogram stretching $$g(x,y) = \frac{v_{max,d} - v_{min,d}}{v_{max} - v_{min}}(f(x,y) - v_{min}) + v_{min,d}$$
- Z-normalization $$g(x,y) = \frac{f(x,y) - \mu}{\sigma}$$

Be aware when high intensity areas are present!

Z-normalization is the de-facto standard for most MRI-based preprocessing
What about images with non-arbitrary units (CT, PET)?

Intensity normalization

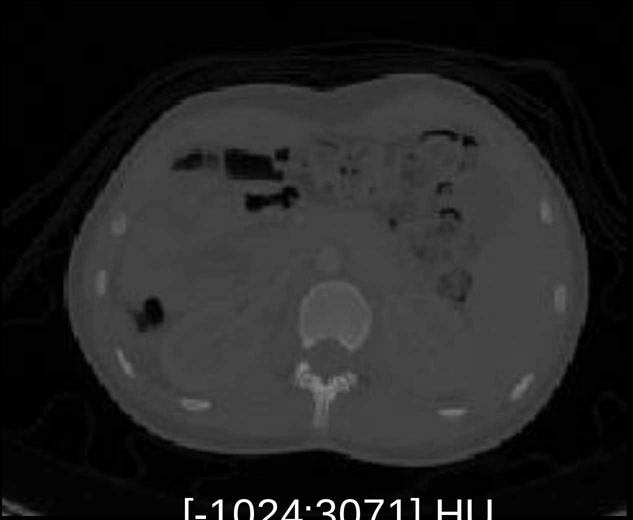


Normalize relative to a reference region before scaling

Examples:

- Background region in brain
- Liver region in whole-body imaging

Intensity mapping



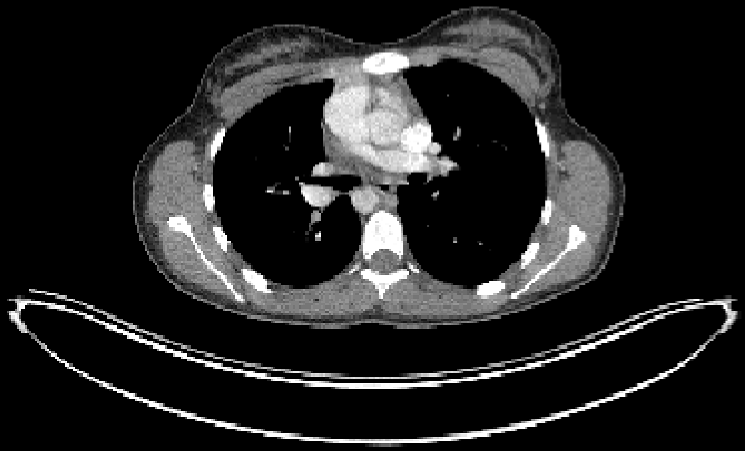
[-1024;3071] HU

E.g. by histogram stretching or intensity rescaling:

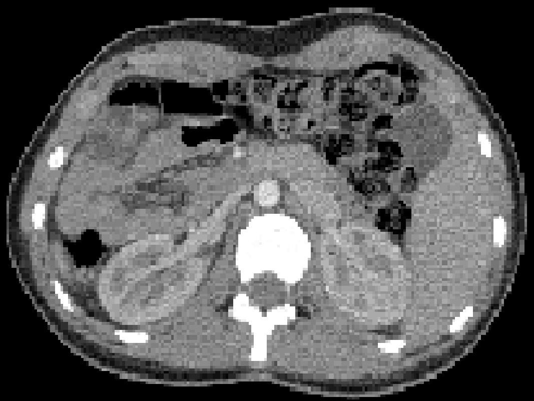
Each image is mapped from v_{min} and v_{max} to $v_{min,d}$ and $v_{max,d}$ (often 0-255) using:

$$g(x,y) = \frac{f(x,y) - v_{min}}{v_{max} - v_{min}} * (v_{max,d} - v_{min,d}) + v_{min,d}$$

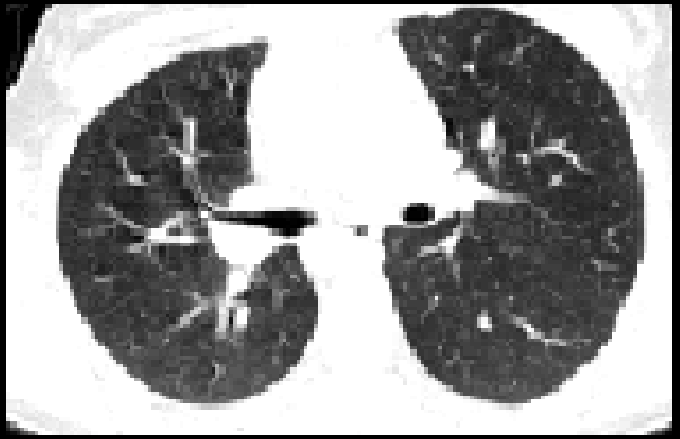
followed by clamping values outside the range



[-150;250] HU

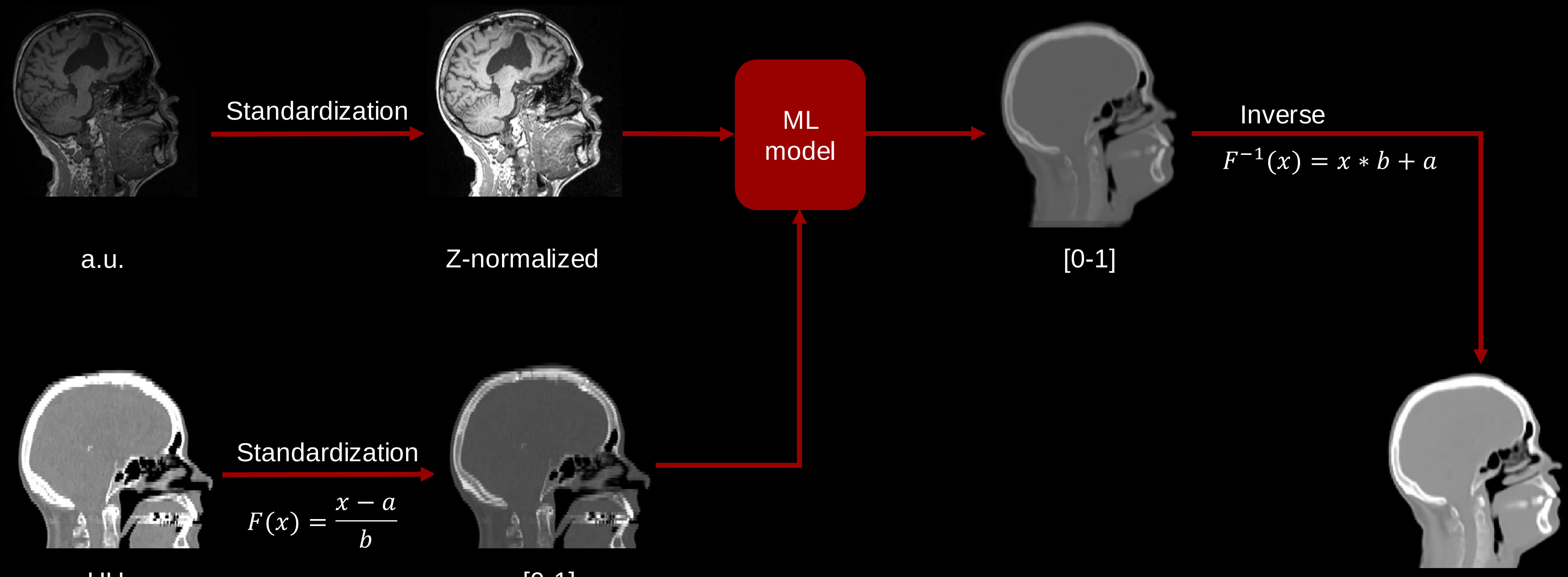


[-150;250] HU



[-1000;0] HU

Intensity normalization



Standardization

a.u.

Z-normalized

ML model

[0-1]

Inverse

$$F^{-1}(x) = x * b + a$$

HU

Standardization

$$F(x) = \frac{x - a}{b}$$

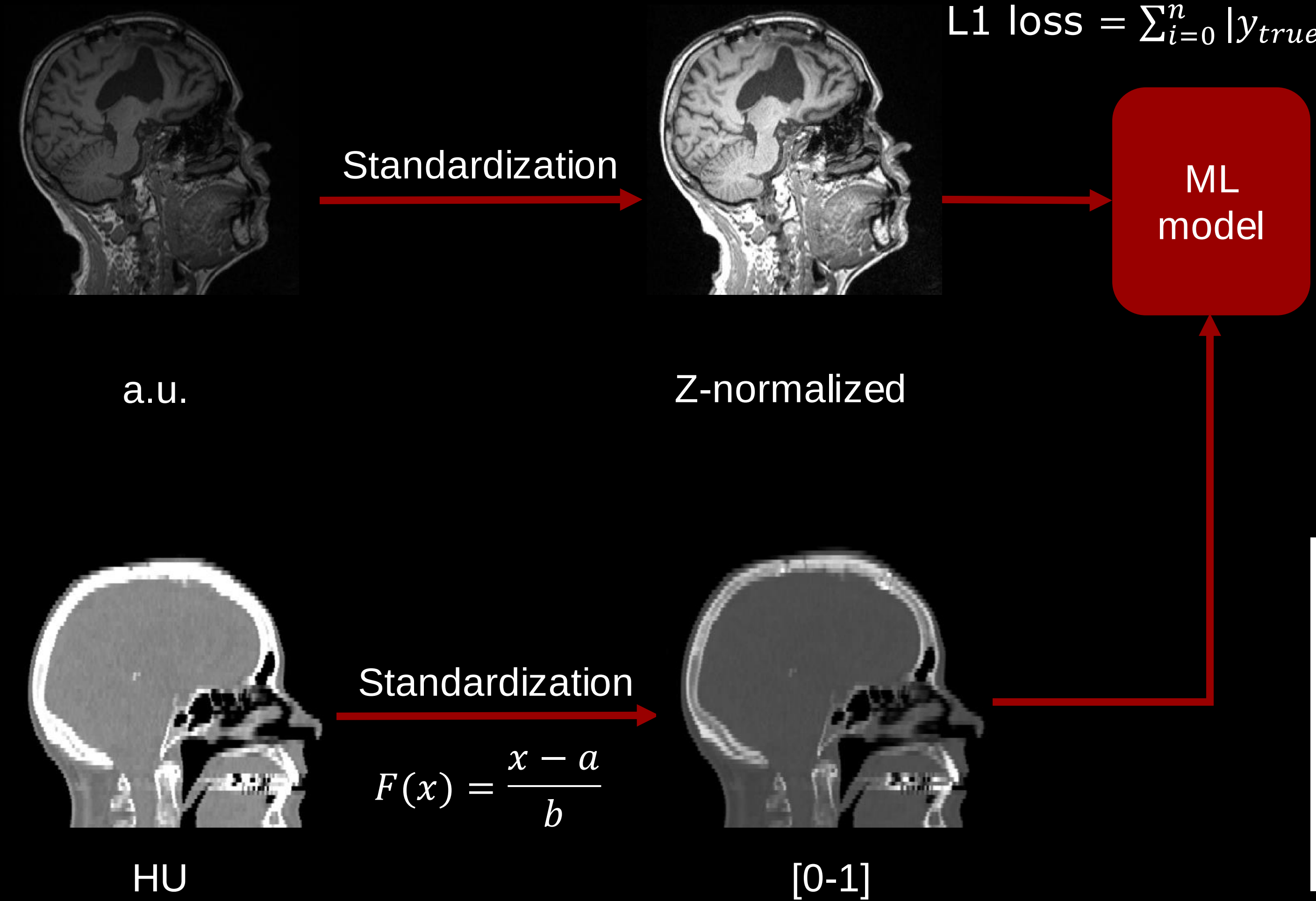
HU

[0-1]

Intensity normalization

	Min-max
500 vs 600 HU	0.05
-50 vs 50 HU	0.05

$$\text{L1 loss} = \sum_{i=0}^{n} |y_{true} - y_{predicted}|$$



Standardization

ML model

a.u.

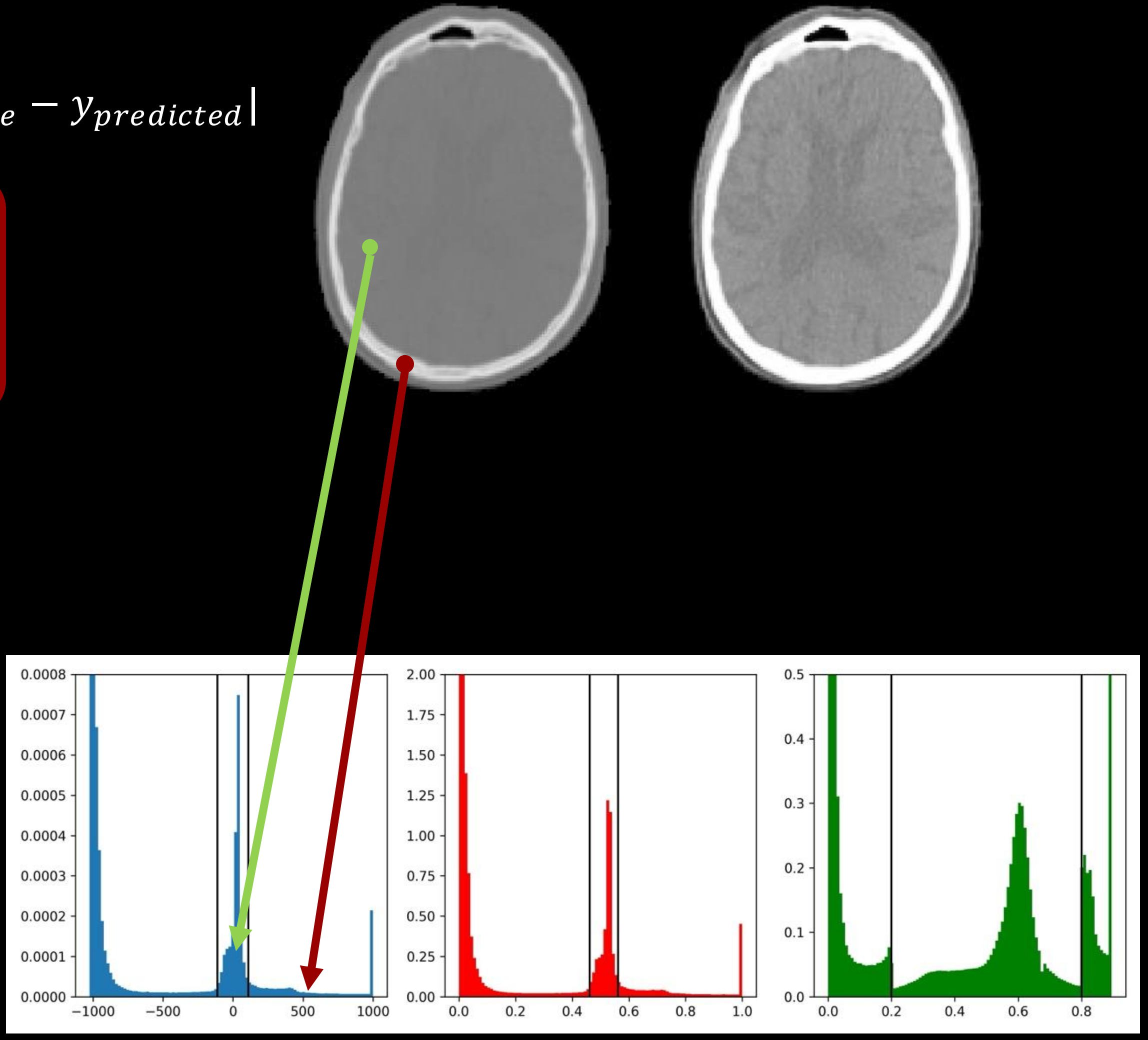
Z-normalized

Standardization

$$F(x) = \frac{x-a}{b}$$

HU

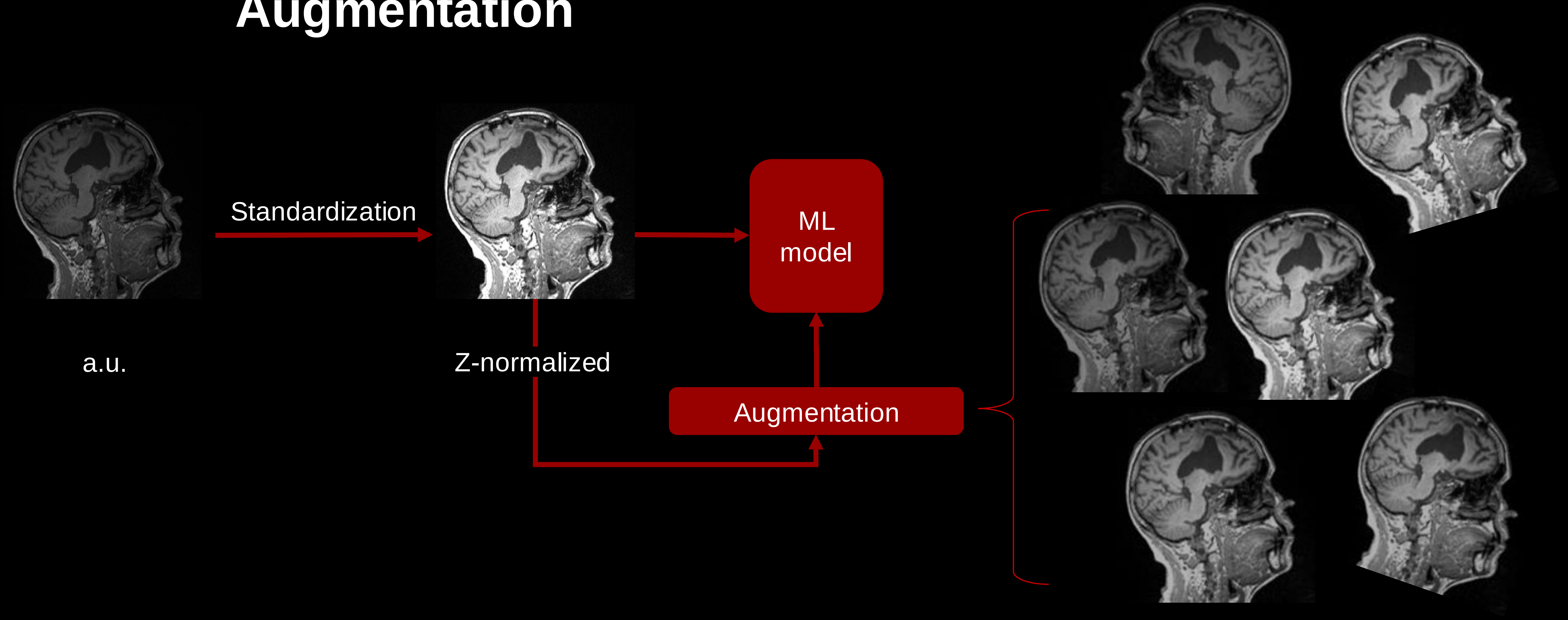
[0-1]



Min-max standardization with a=-1000 b=2000

Stretched with intervals [-1000, -100, 100, 1000]

Augmentation



a.u.

Standardization

Z-normalized

ML model

Augmentation

Quiz 2

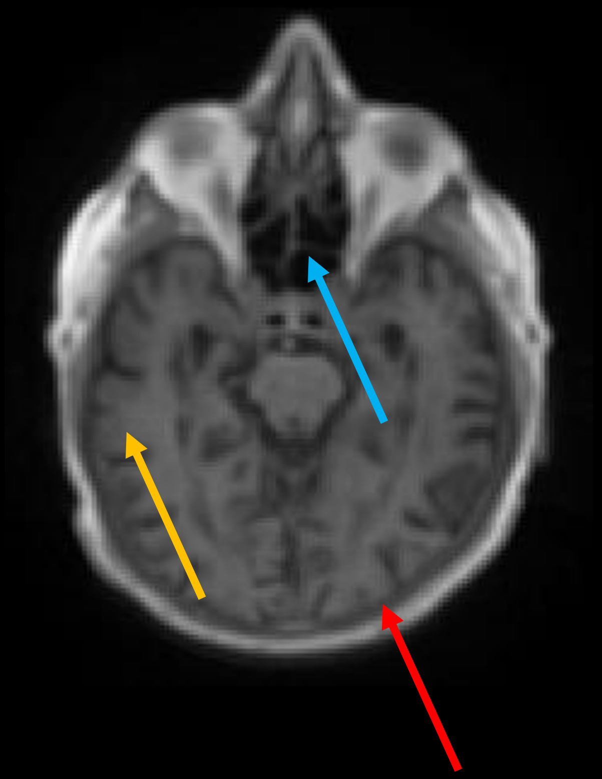
- A model is trained to predict the percieved age of a patients' brain given an MRI
- The model was trained with data containing ages of 18 to 99, so was scaled using:

$$g(x,y) = \frac{f(x,y) - v_{min}}{v_{max} - v_{min}} * (v_{max,d} - v_{min,d}) + v_{min,d}$$

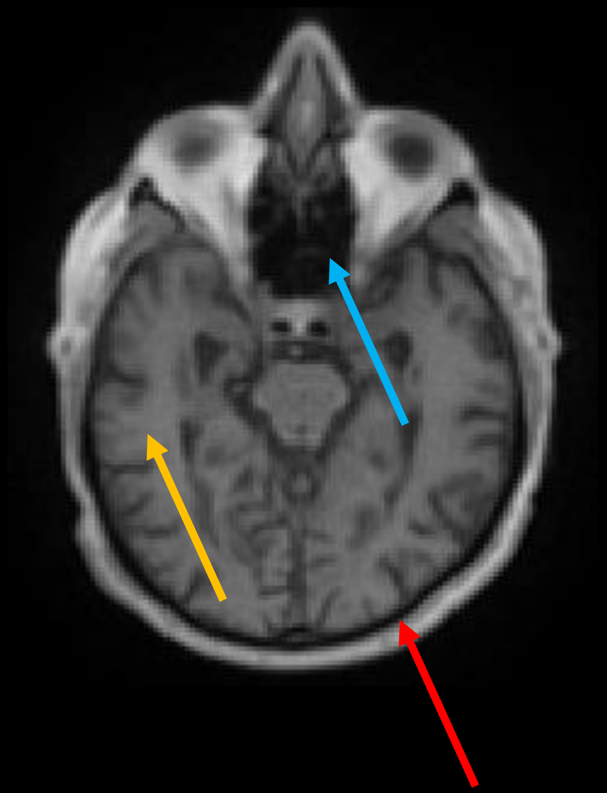
  where (v_{min}, v_{max}) = (18, 99) and $(v_{min,d}, v_{max,d})$ = (0, 1)
- The model predict 0.78 for a given MRI. What is the predicted age (in years) of the patient?
  - 63
  - 70
  - 81
  - 95

Segmentation of air regions

UTE$_{TE1}$



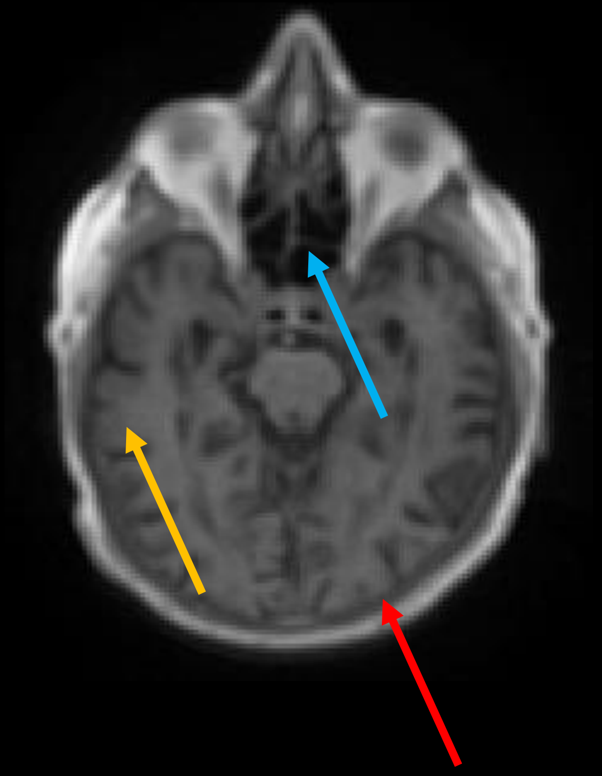
UTE$_{TE2}$



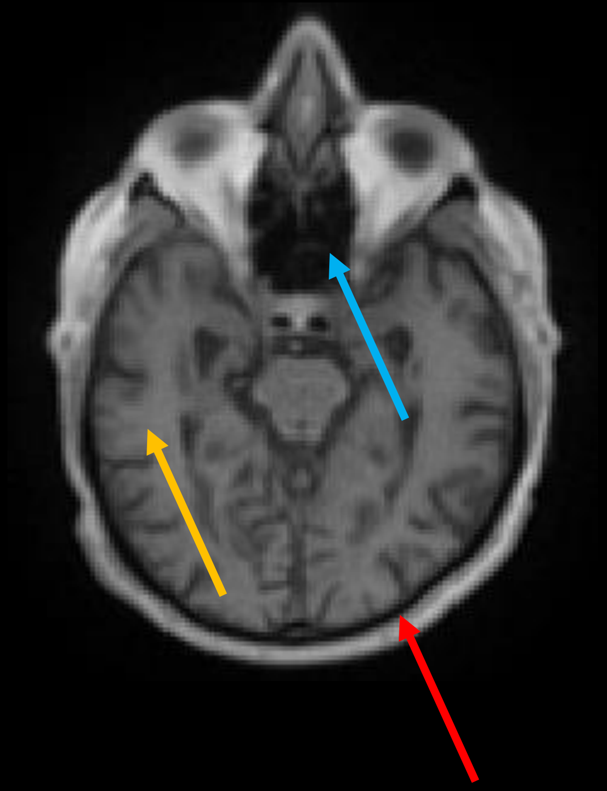
- Two MR images acquired with different echo times TE1 << TE2
- Different intensities are expected in bone but not in air and tissue

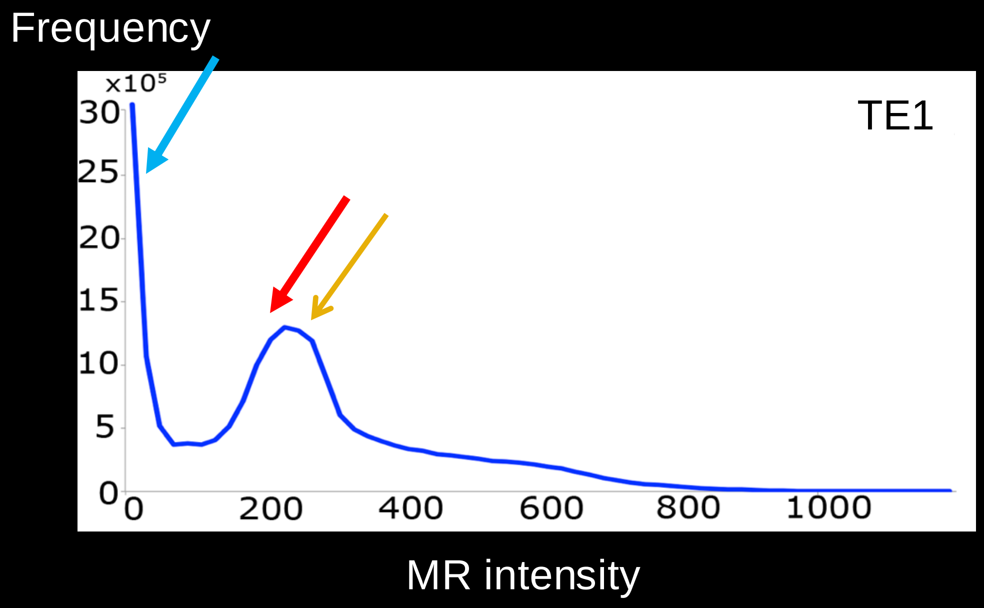
Segmentation of air regions

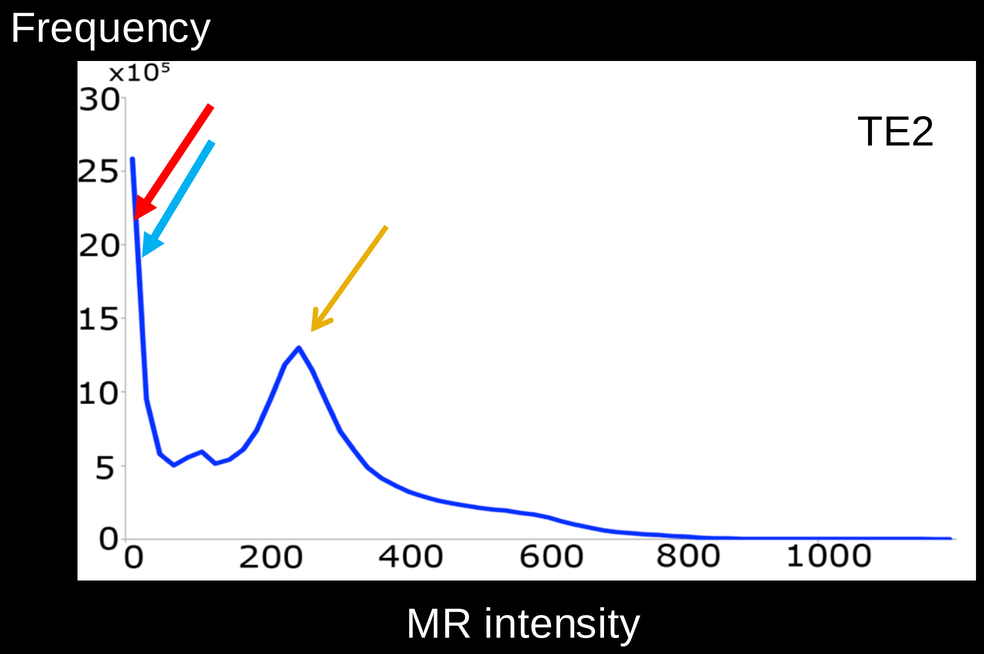
UTE$_{TE1}$



UTE$_{TE2}$

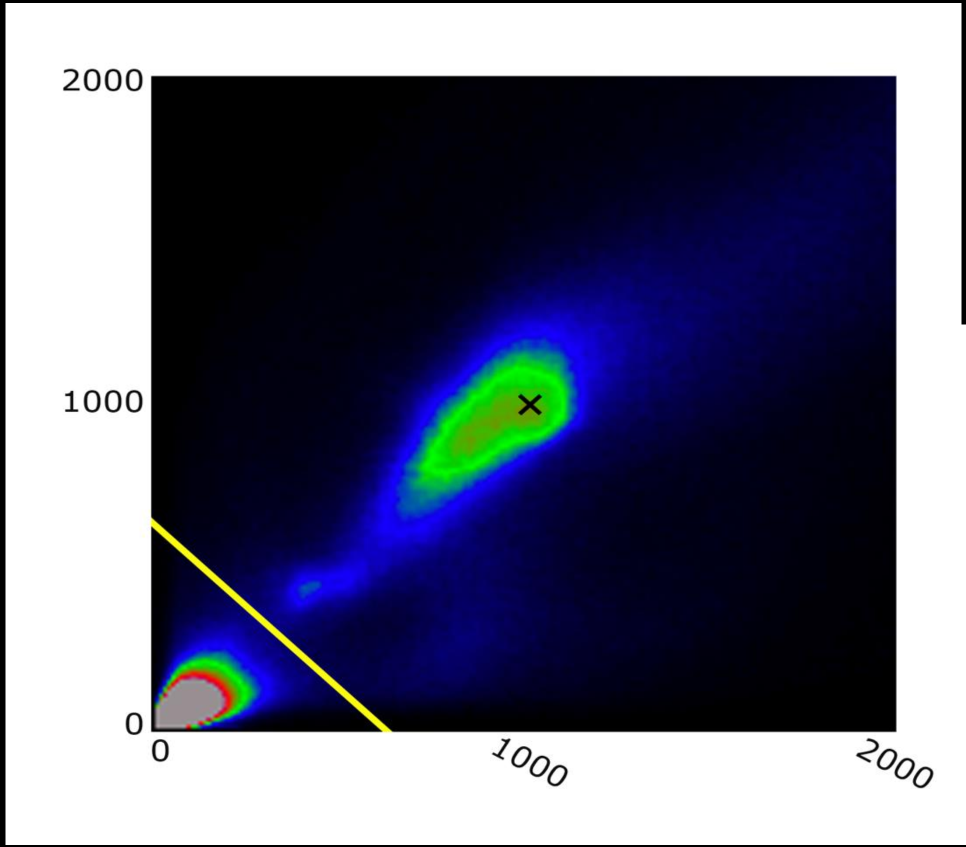


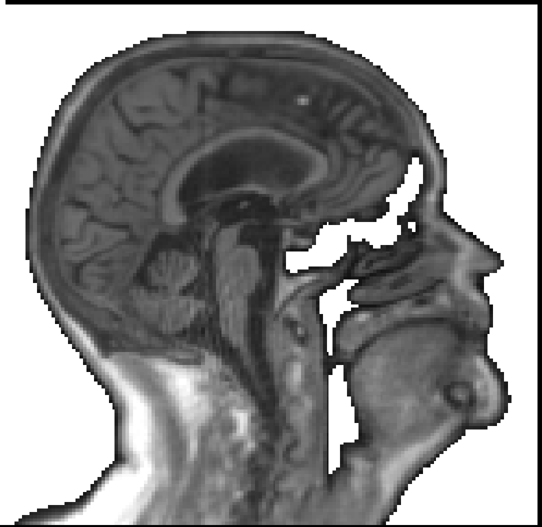


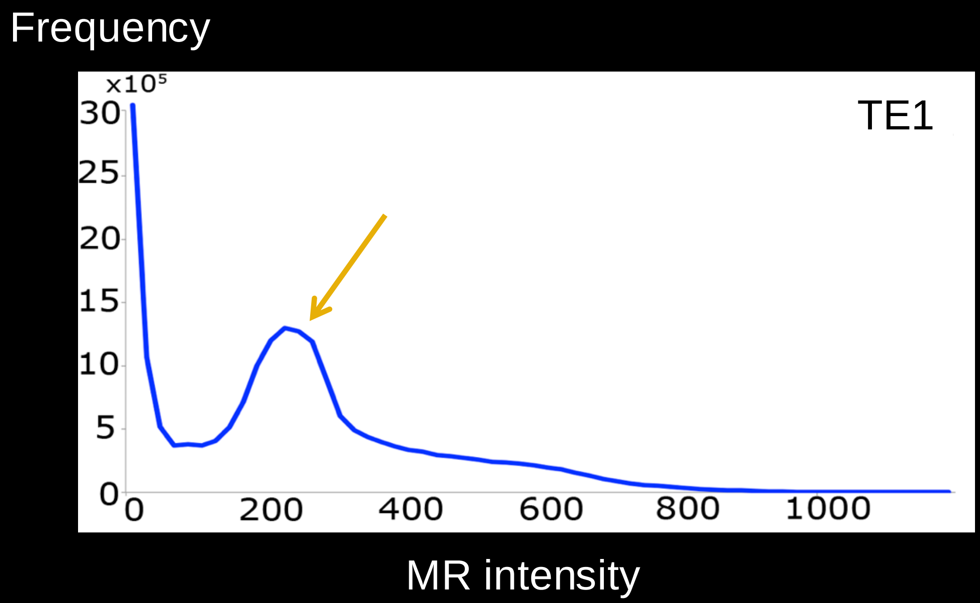


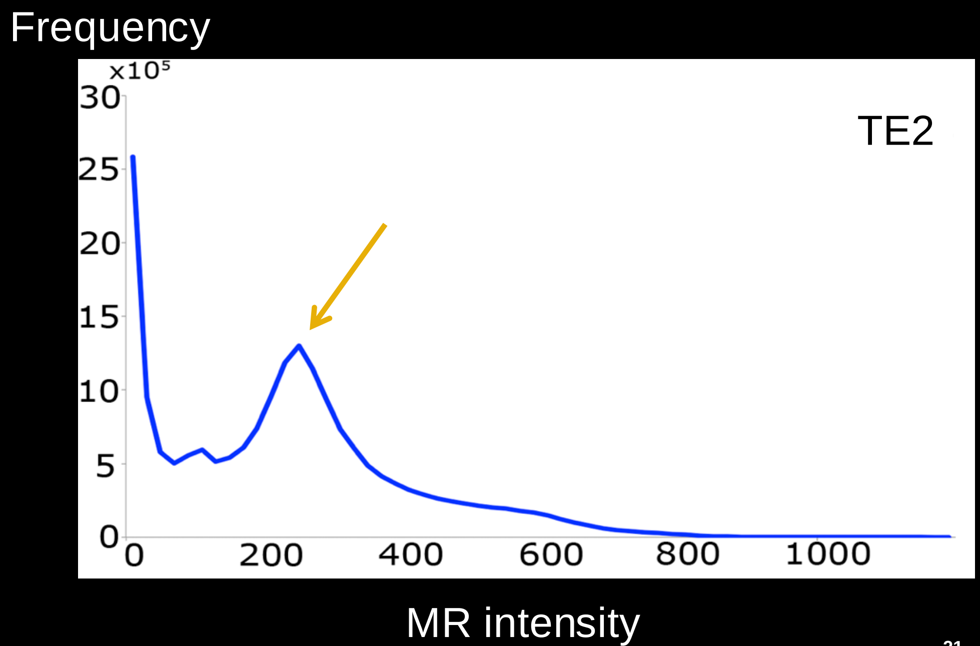
Segmentation of air regions

Normalized Joint histogram



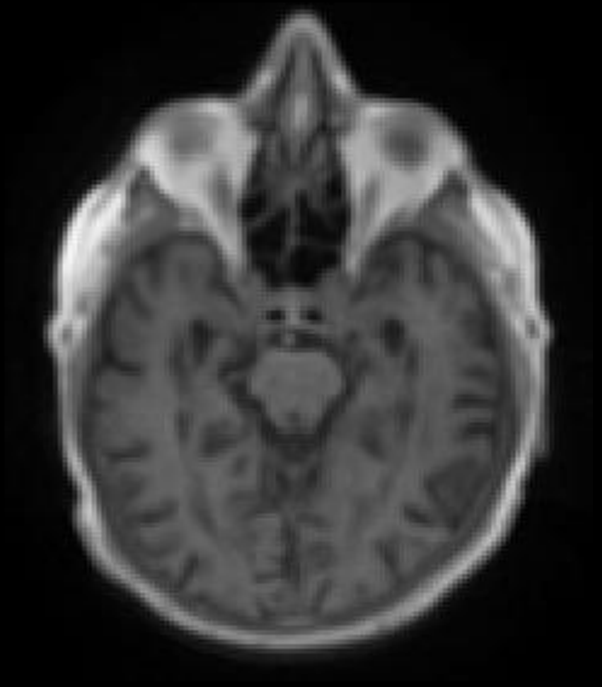




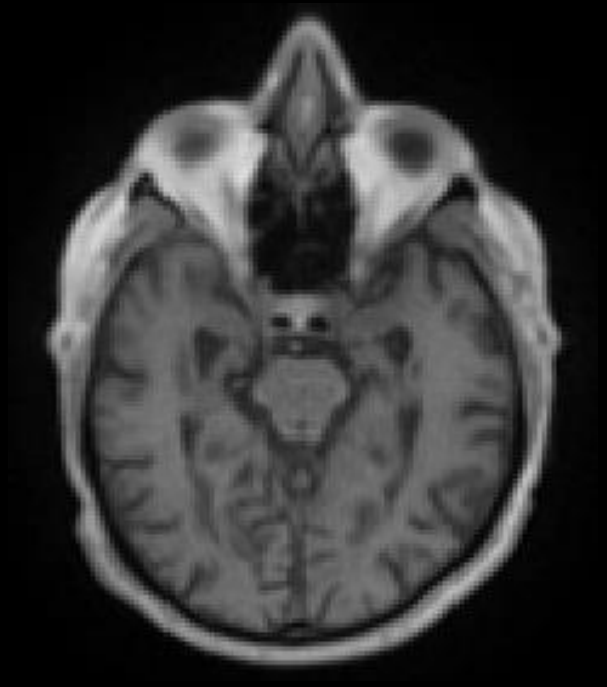


Intensity mapping

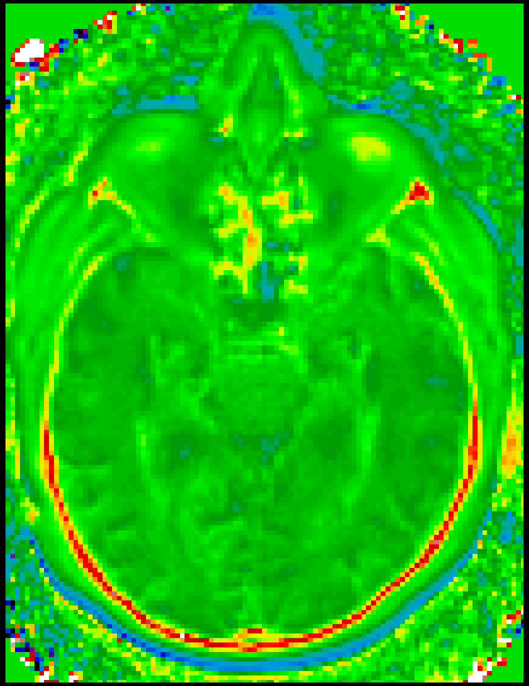
$\mathrm{UTE}_{\mathrm{TE1}}$



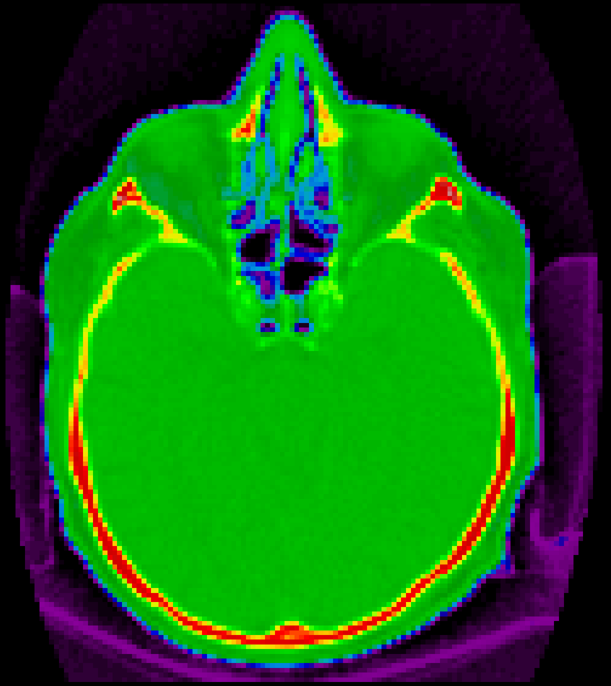
$\mathrm{UTE}_{\mathrm{TE2}}$



R_2^*

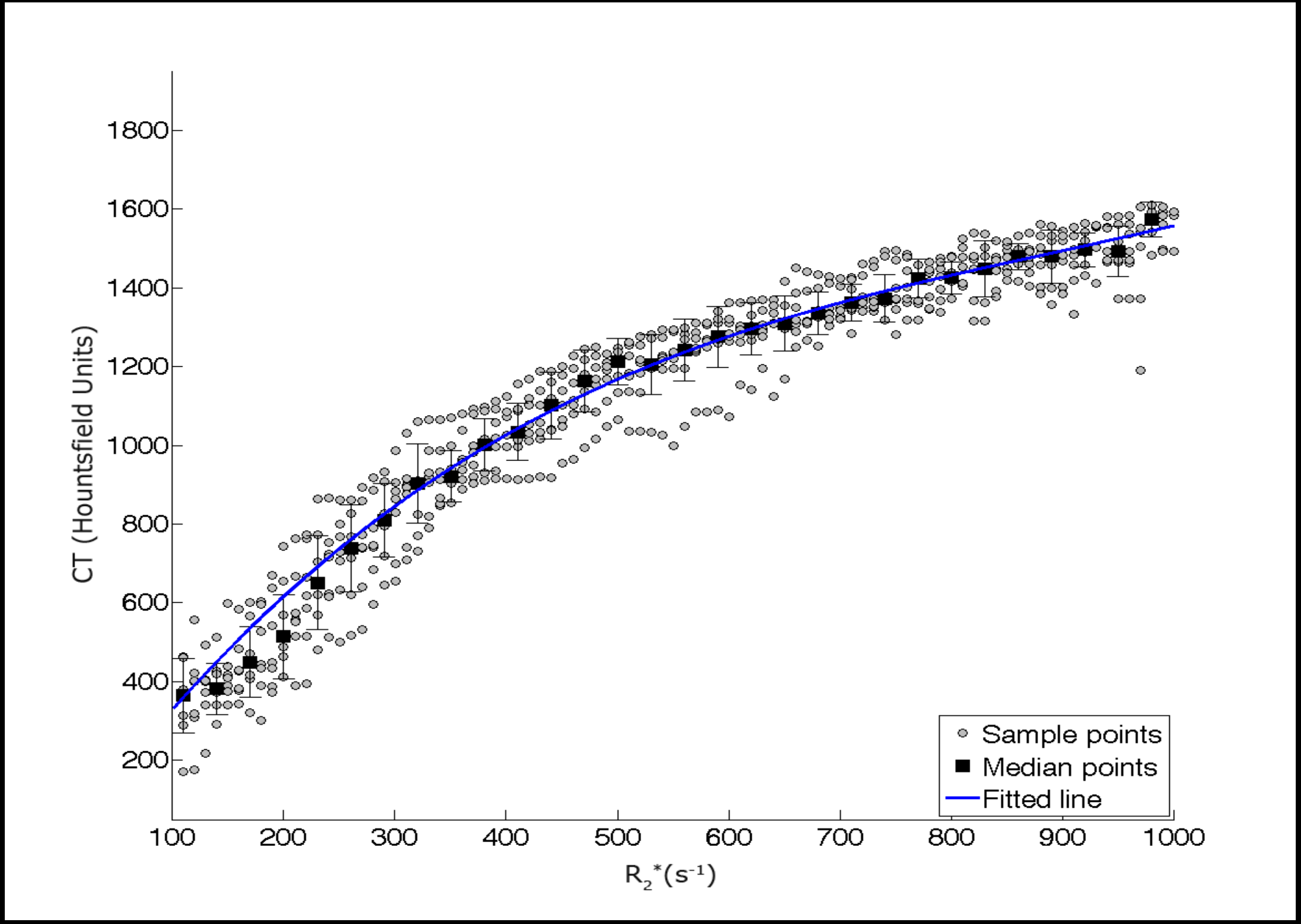


CT

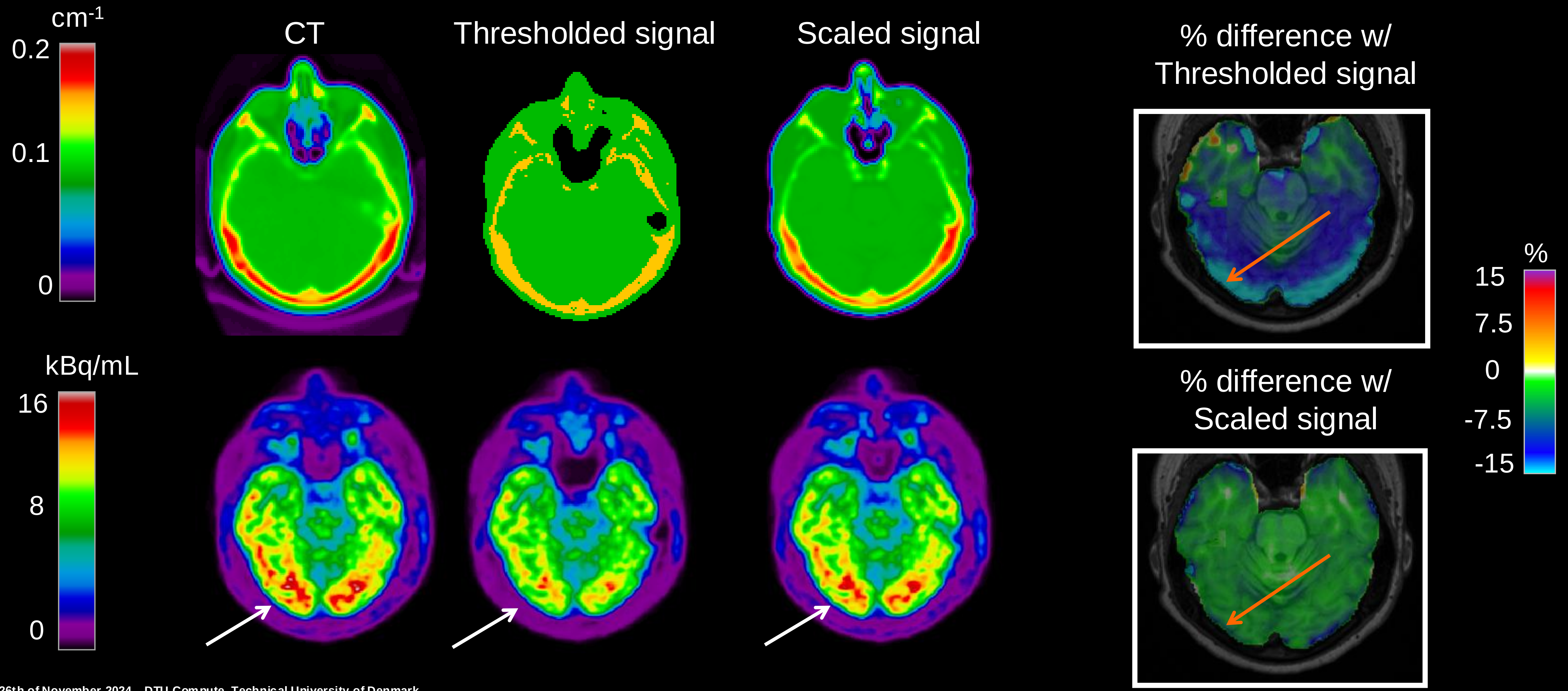


$$R_2^* = \frac{\ln(UTE_{TE1}) - \ln(UTE_{TE2})}{TE2 - TE1}$$

Intensity mapping



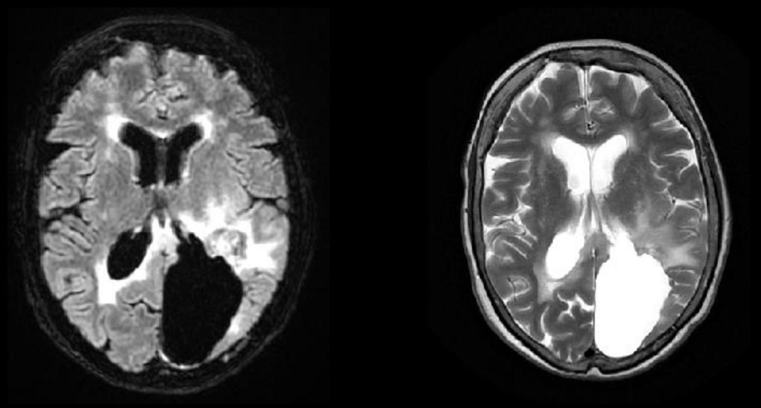
Intensity mapping



Registration

- Interpolation
- Intra subject registration
  - Same session
  - Between sessions
- Inter subject registration

Interpolation



0.5x0.5x0.6 mm3 0.5x0.5x4.4 mm3

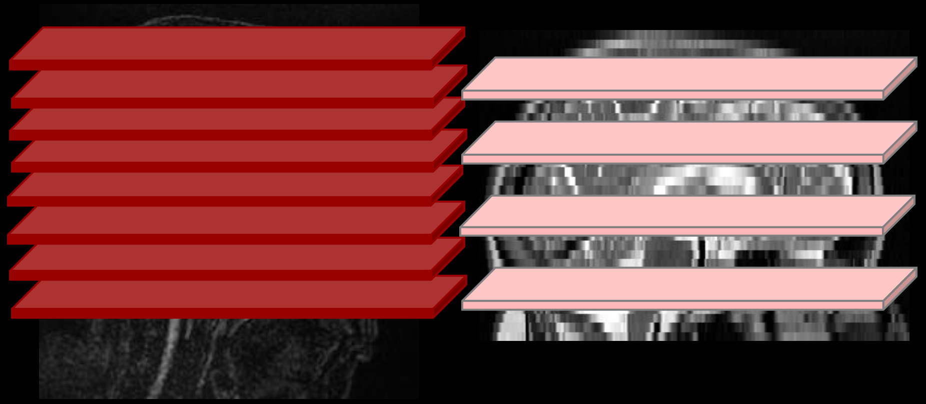
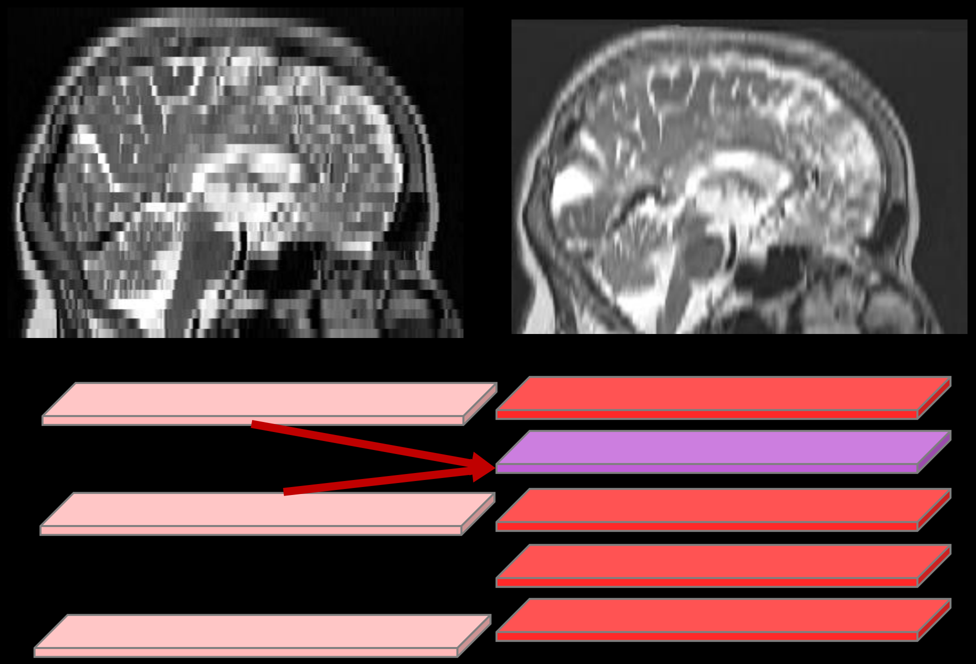
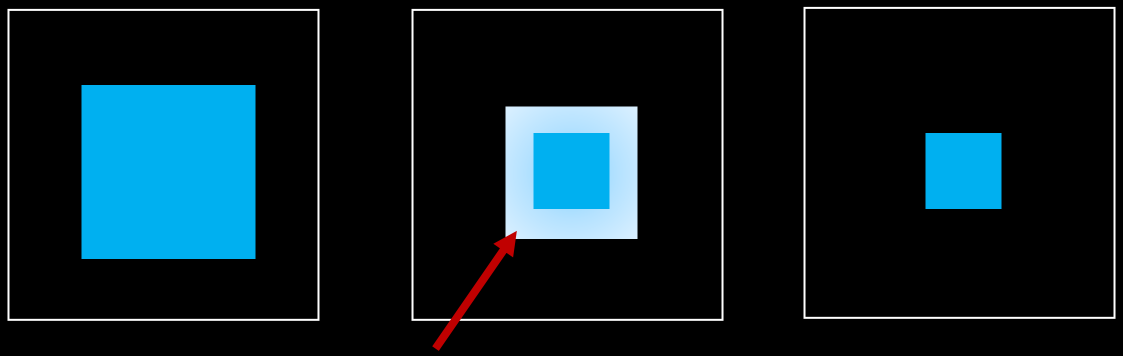


Image interpolation → Trilinear (or similar)



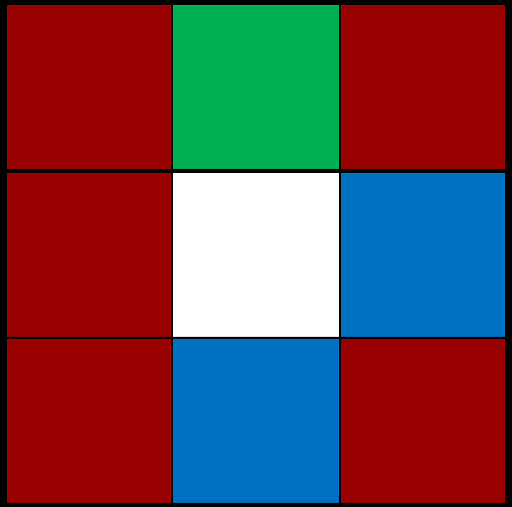
Label interpolation → Nearest Neighbour



Nearest neighbour ensures integer (e.g. 0 and 1) values

Quiz 3

- In a 4-connectivity setting, what would the color of the white center pixel be assigned when using nearest neighbour interpolation?
  - – Green
  - – Blue
  - – Red



Different transformations:

Translation

Rotation

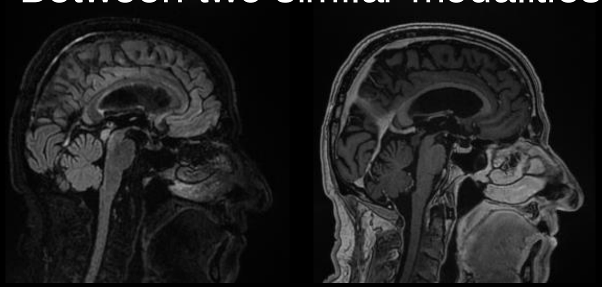
Scaling

Sheering

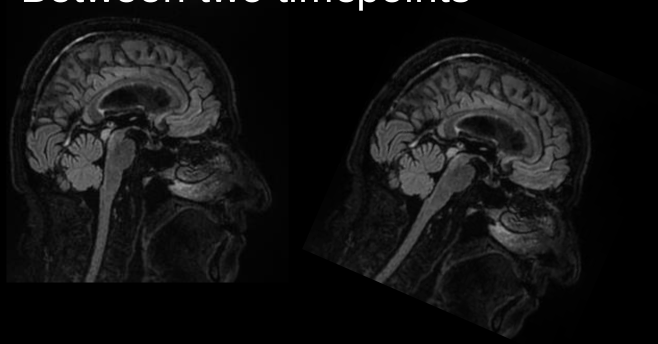
Registration

- Intra subject

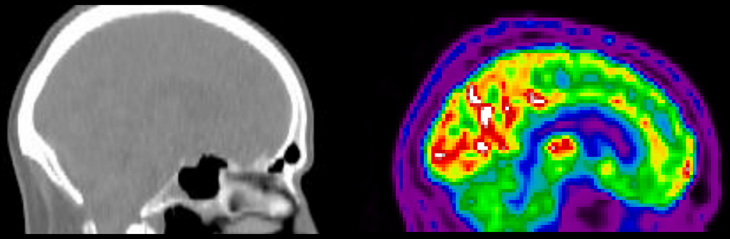
Between two similar modalities



Between two timepoints



Between two different modalities

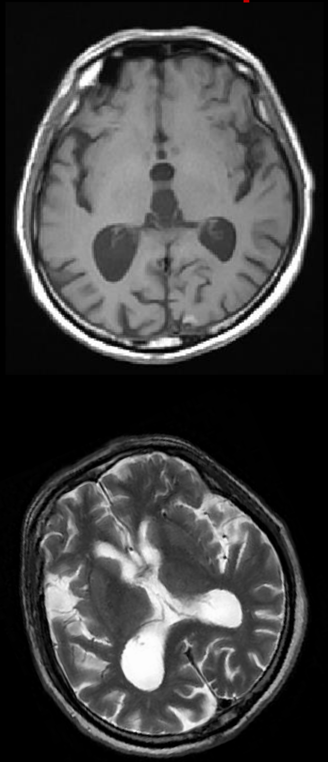


Translation and rotation are used for intra subject registration

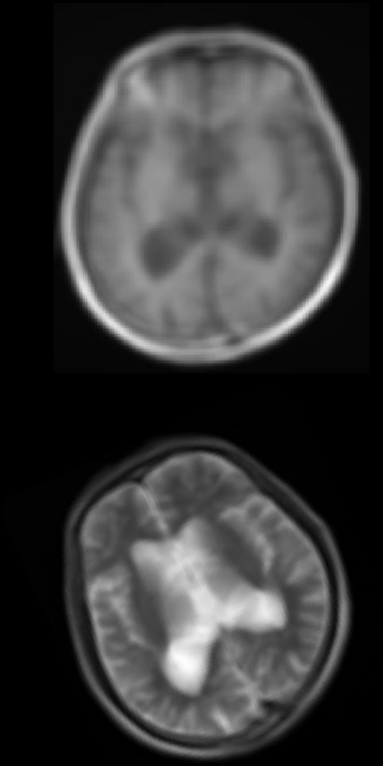
Scaling mainly used for inter subject registration

Registration

Global step:

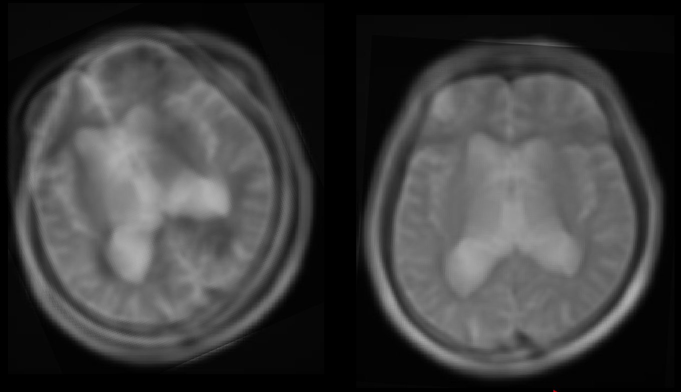


Search for overlap at low-to-high resolution

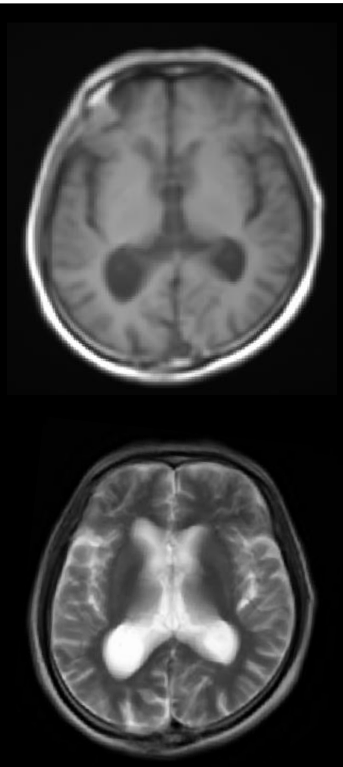


8mm

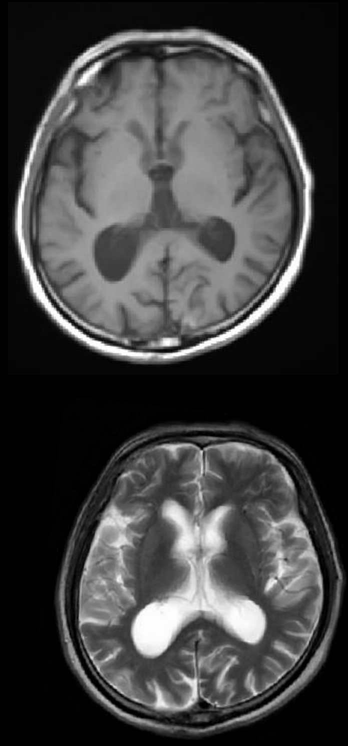
Local optimization step:



Course search grid to find optimal translation and rotation



4mm



2mm

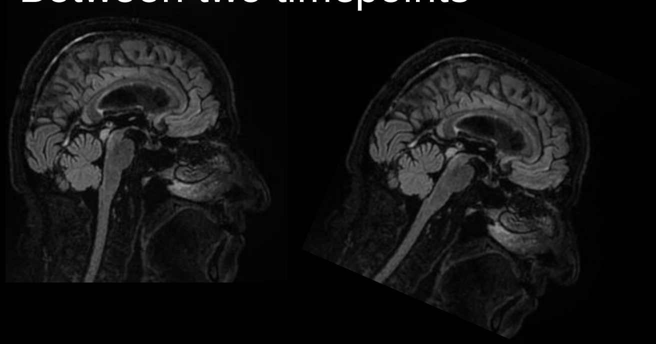
Registration

Similar modality cost function:

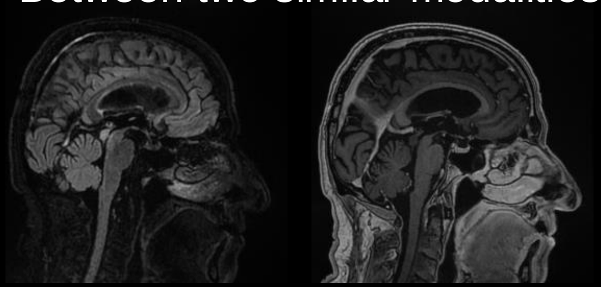
Least squares

Normalized correlation

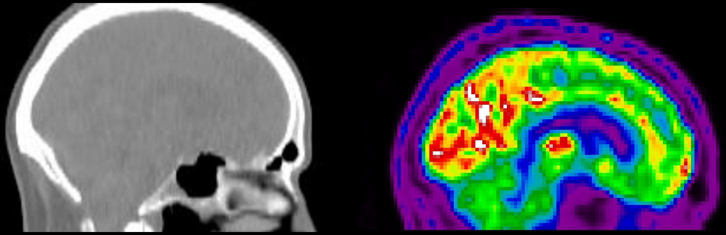
Between two timepoints



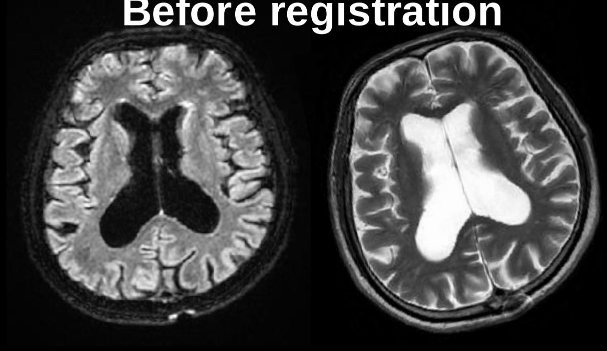
Between two similar modalities



Between two different modalities



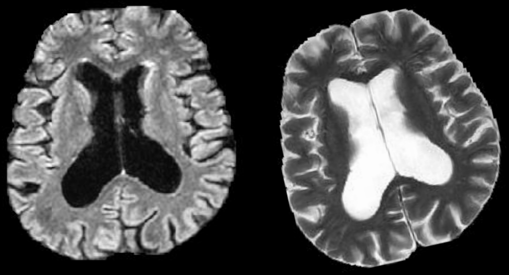
Before registration



Target

Moving

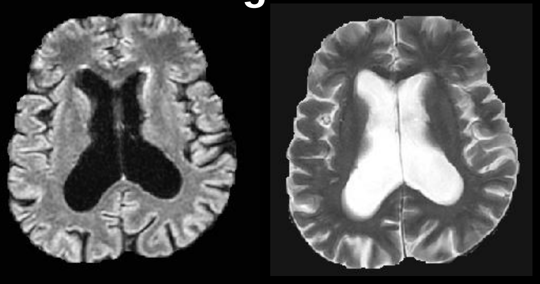
Brain extraction



Target

Moving

After registration



Target

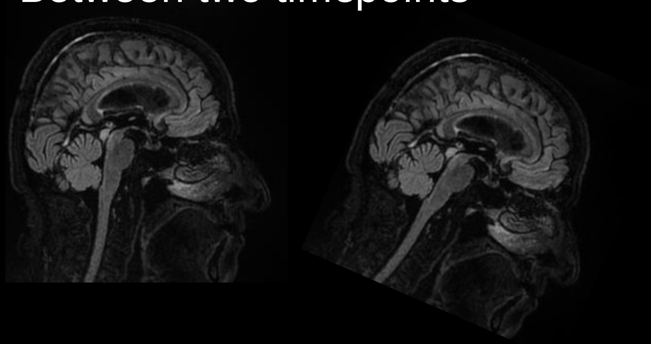
Moving

Registration

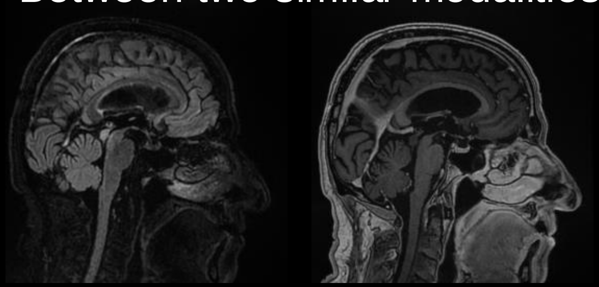
Different modality cost-function:

Mutual information

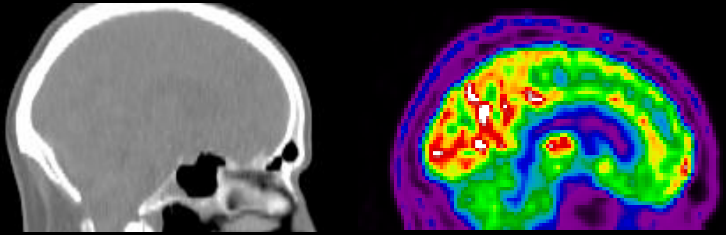
Between two timepoints



Between two similar modalities

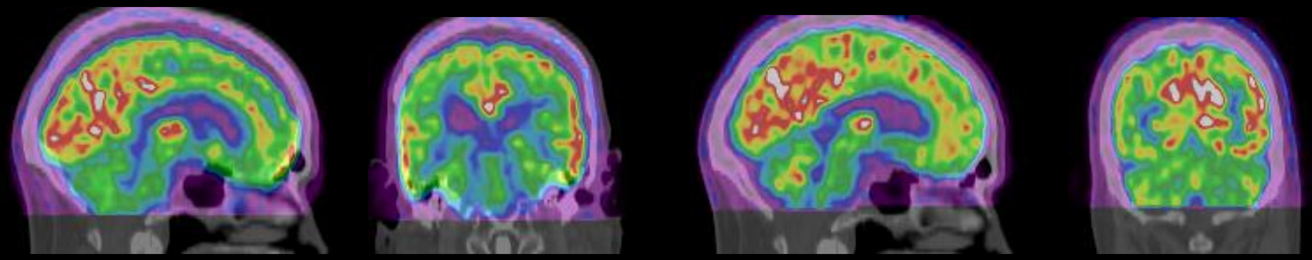


Between two different modalities



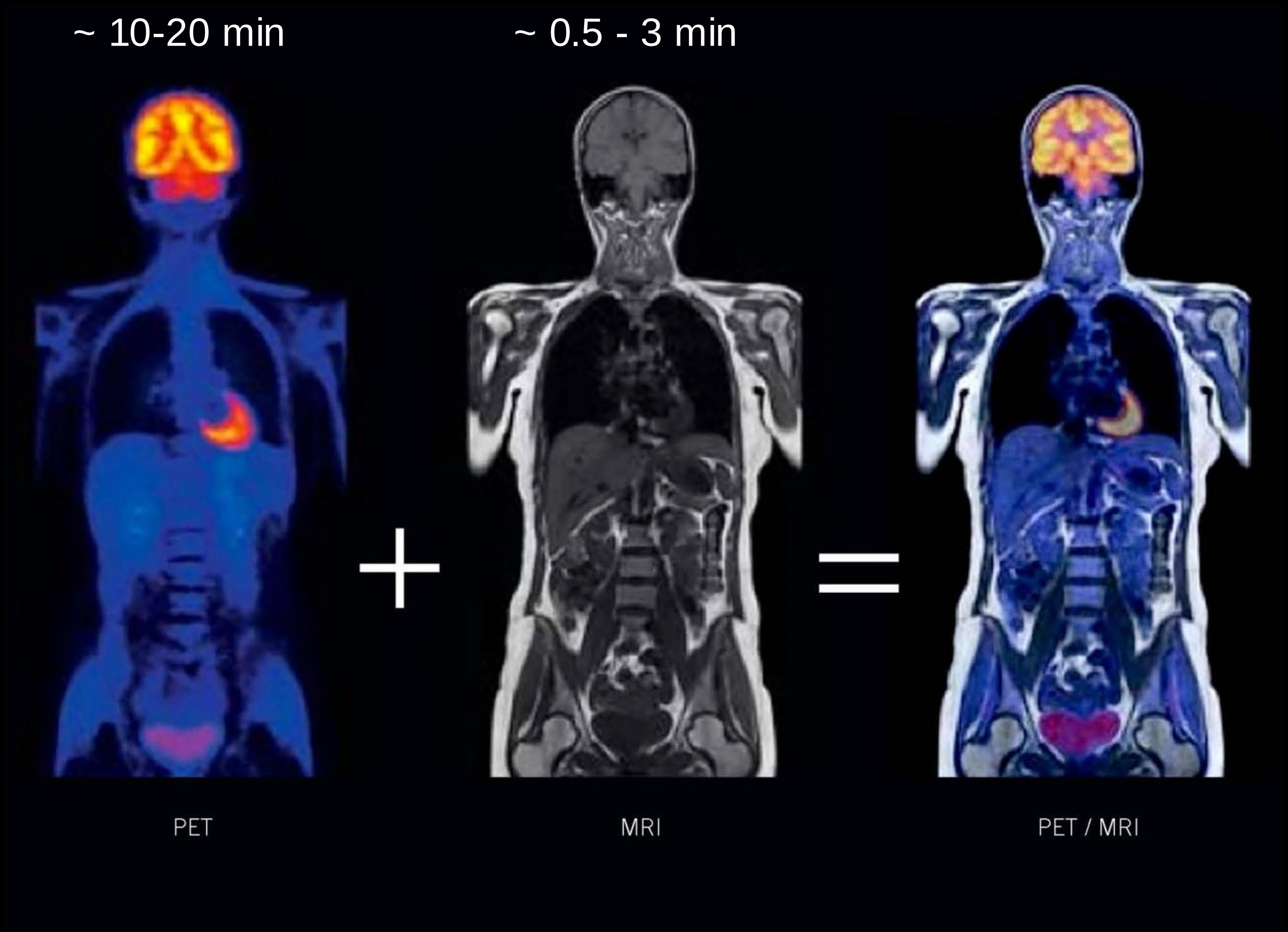
Before registration

After registration



Sagittal Coronal Sagittal Coronal

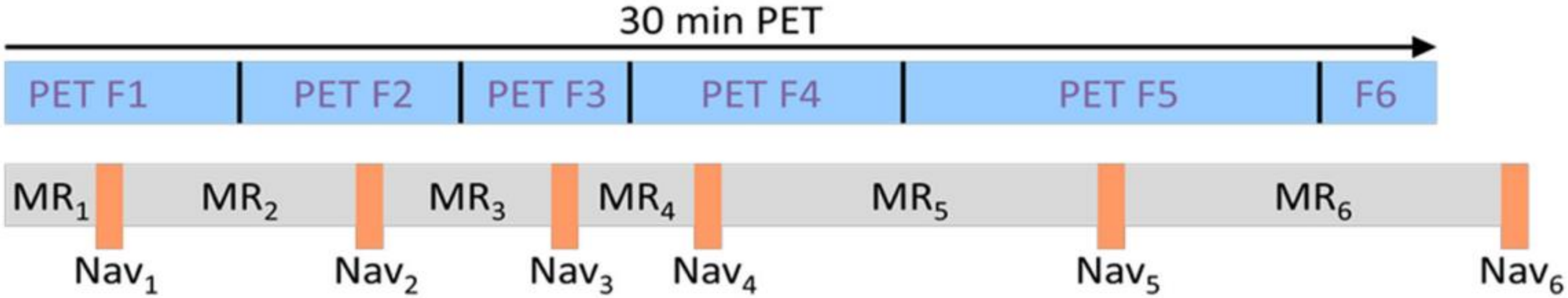
Design a motion-compensated PET/MRI system



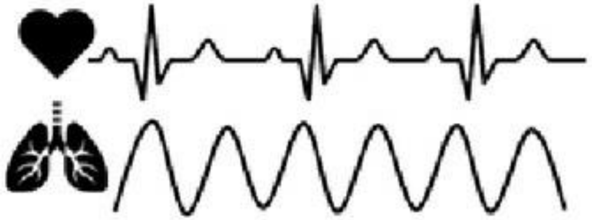
Registration

- Intra-scan motion correction usually requires sensors

Part of the acquisition



Wearable sensors



External sensors

Registration

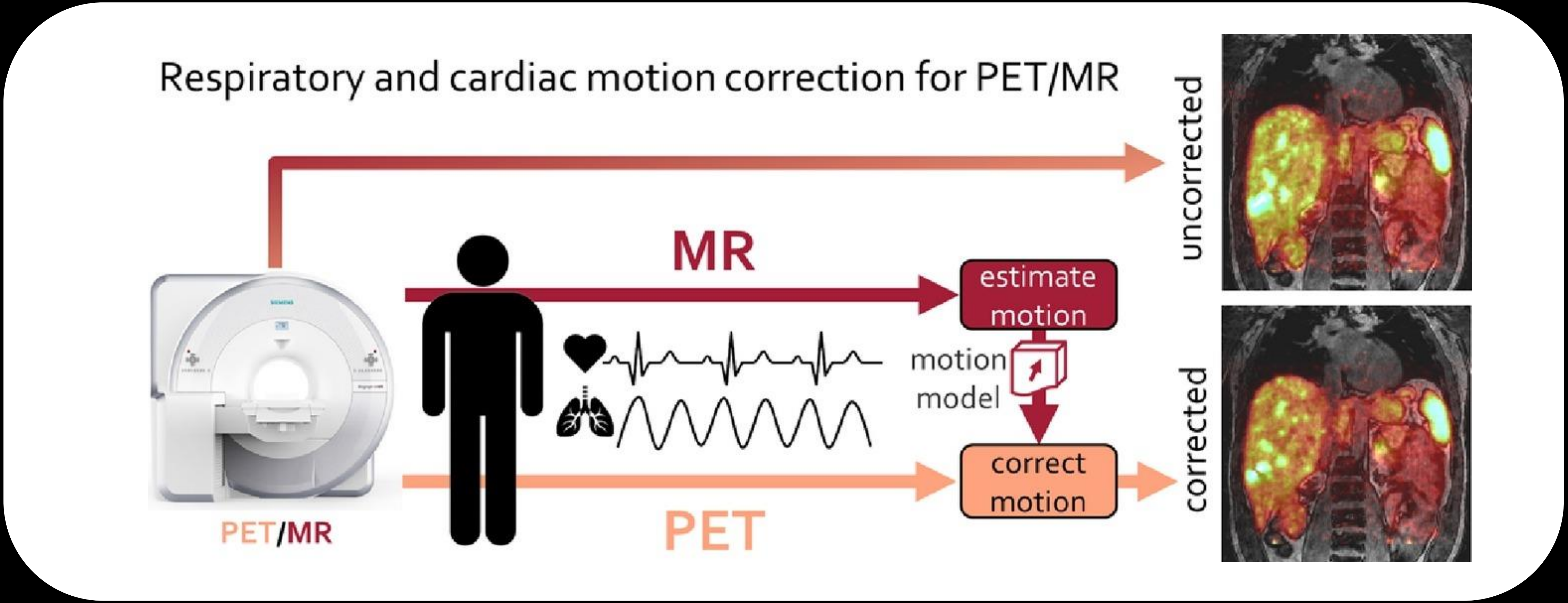


Figure: https://doi.org/10.1016/j.media.2017.08.002

I^{MRI} is target MRI

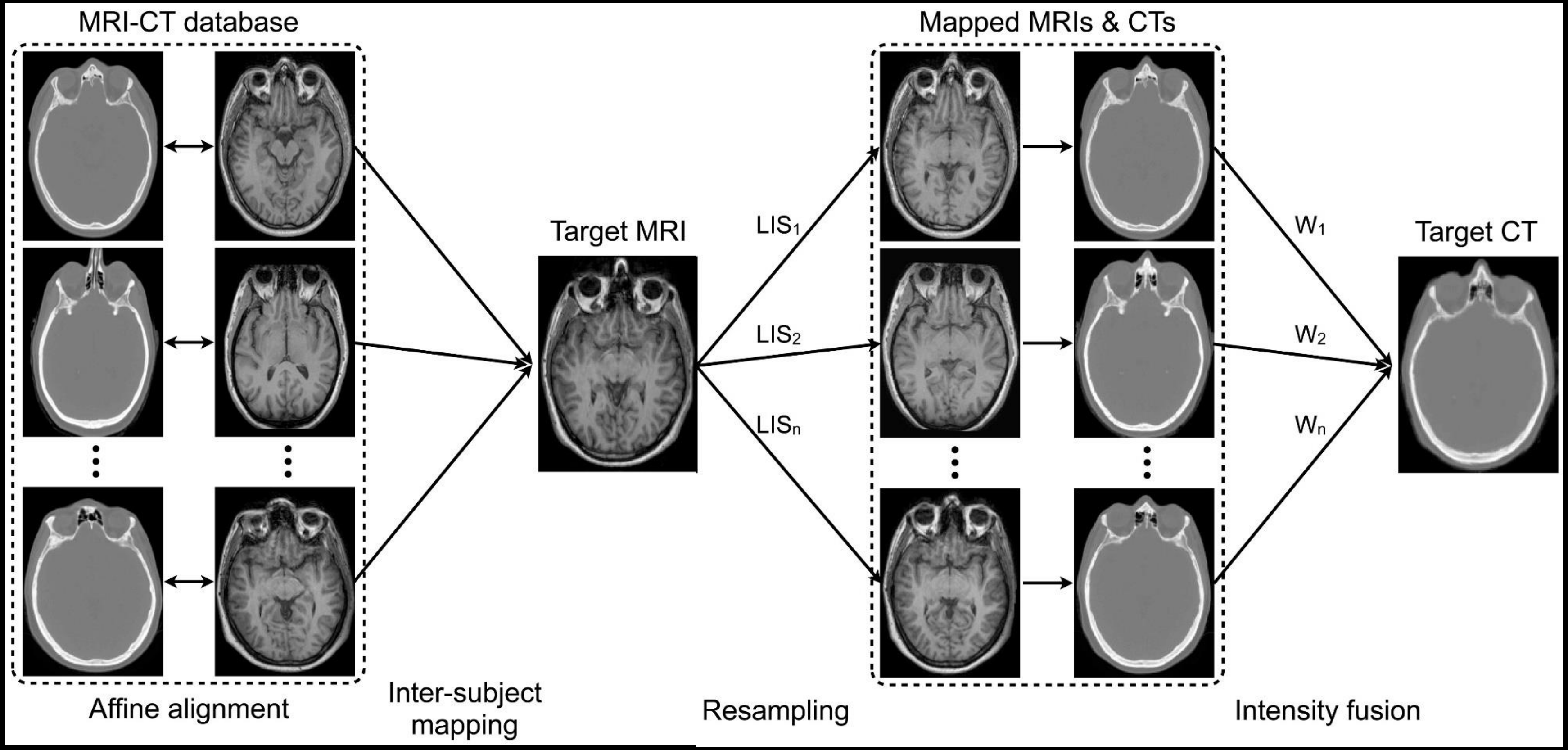
J_n^{MRI} is warped atlas n

$\bar{I}$ is mean of I

$\sigma(I)$ is standard deviation of I

Registration

- Goal is to obtain a synthetic CT based on a patient’s own MRI



Simplest solution:
Find best matching warped MRI

$$NCC_n = \frac{1}{N}\frac{\langle I^{MRI} - \overline{I^{MRI}}, J_n^{MRI} - \overline{J_n^{MRI}} \rangle}{\sigma(I^{MRI})\sigma(J_n^{MRI})}$$

More complex solution:

1. For each voxel, extract patch and compute local NCC (LNCC)
2. Rank the patches based on their LNCC
3. Fuse the CT values based on their ranks
   (higher rank = higher weight)

I^{MRI} is target MRI

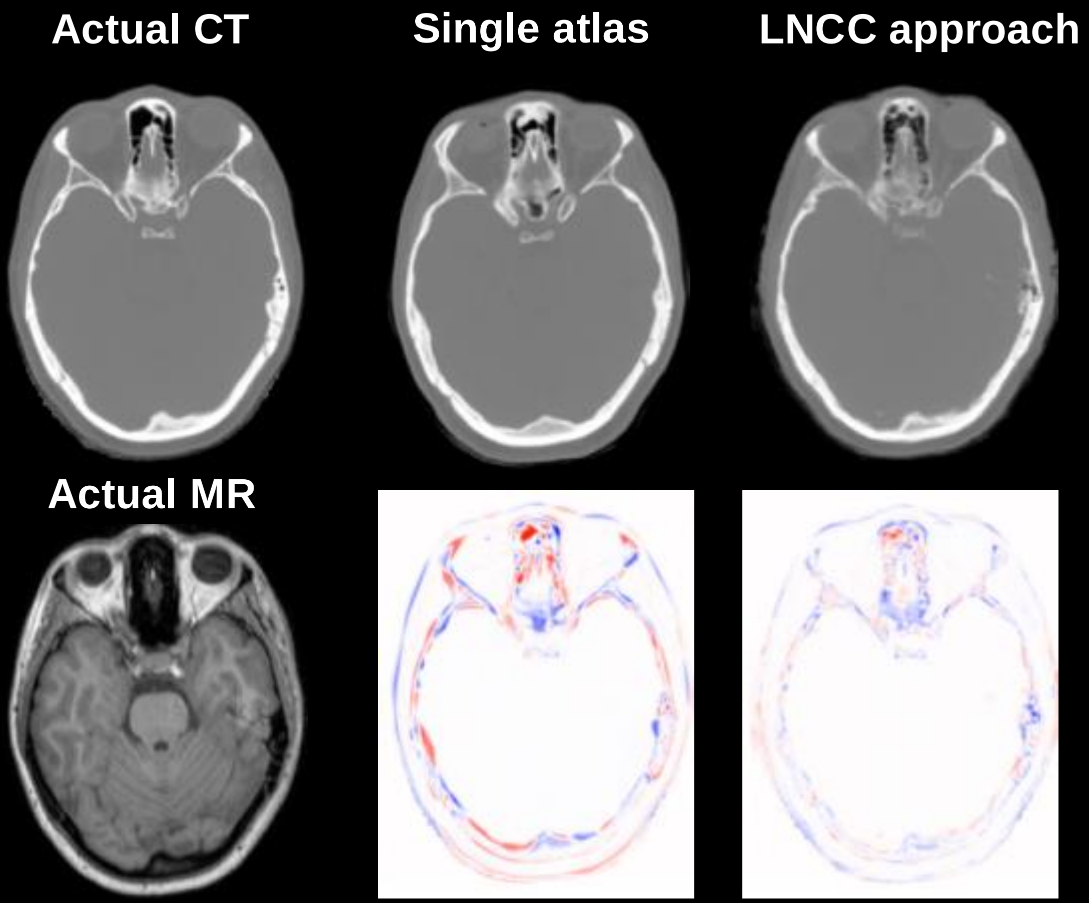
J_n^{MRI} is warped atlas n

$\bar{I}$ is mean of I

$\sigma(I)$ is standard deviation of I

Registration

- Goal is to obtain a synthetic CT based on a patient’s own MRI



Simplest solution:
Find best matching warped MRI

$$NCC_n = \frac{1}{N}\frac{\langle I^{MRI} - \overline{I^{MRI}}, J_n^{MRI} - \overline{J_n^{MRI}} \rangle}{\sigma(I^{MRI})\sigma(J_n^{MRI})}$$

More complex solution:

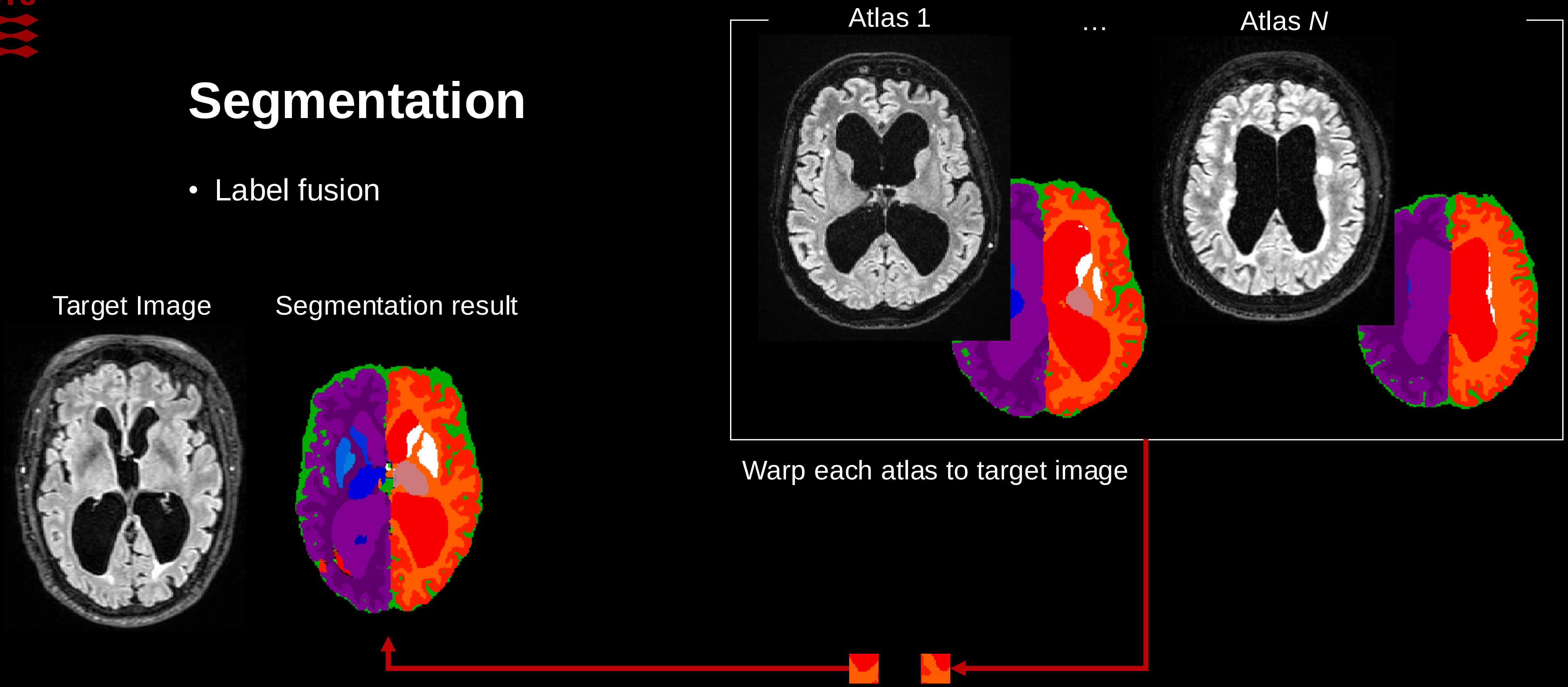
1. For each voxel, extract patch and compute local NCC (LNCC)
2. Rank the patches based on their LNCC
3. Fuse the CT values based on their ranks
(higher rank = higher weight)

Detection

- Segmentation
- Detection
- Tracking

Segmentation

- Label fusion



Target Image

Segmentation result

Atlas 1

…

Atlas *N*

Warp each atlas to target image

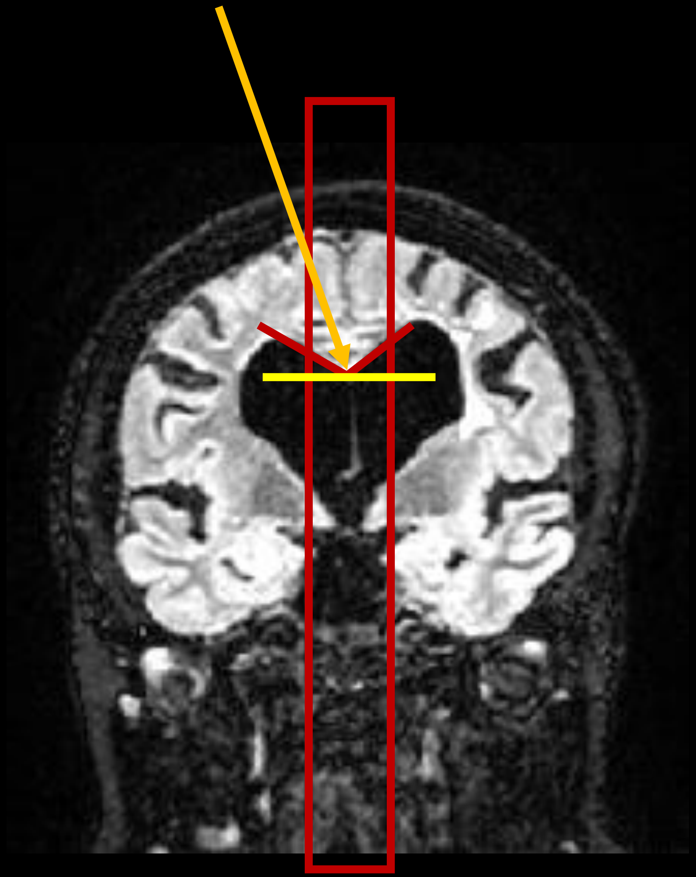
Fuse labels to final class (e.g. by majority voting) for each patch

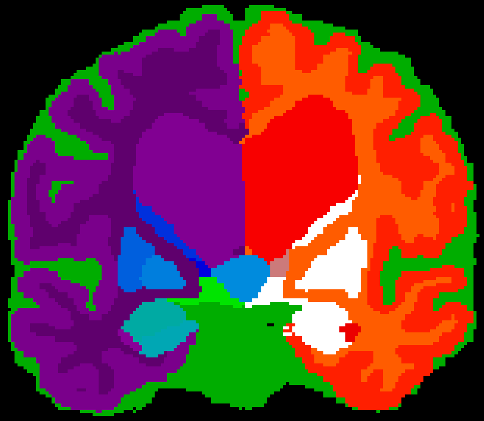
Quiz 4

- The 10 estimates for a class label are found after registration.
  - [1, 5, 2, 1, 2, 5, 4, 5, 2, 2]
- Using majority voting, what is the final predicted class?
  - Answers:
    - 1
    - 2
    - 4
    - 5

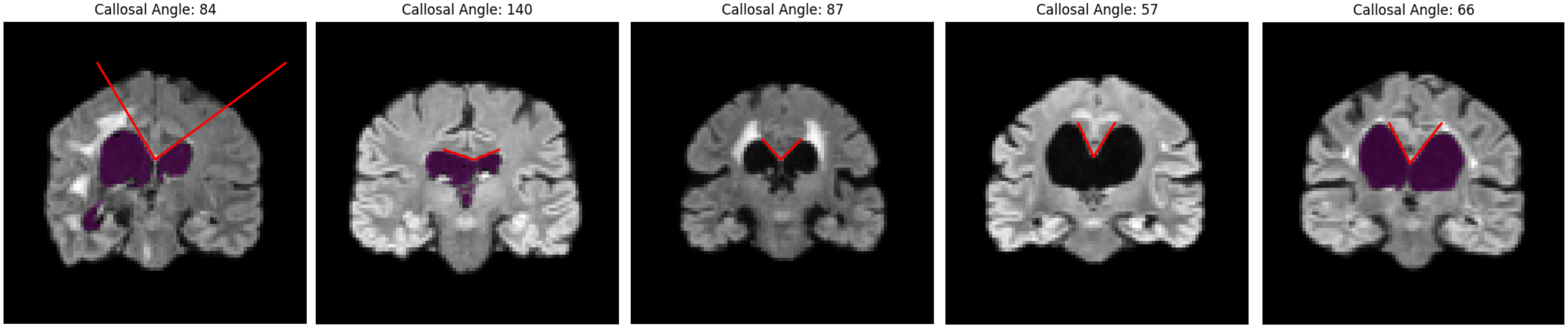
Detection

- Determine the Callosal angle
- Steps
  1. Align MRI to standard space to select standard center slice
  2. Determine first row without brain tissue in center columns
  3. Fit a line to brain tissue points for each side
  4. Determine angle between lines



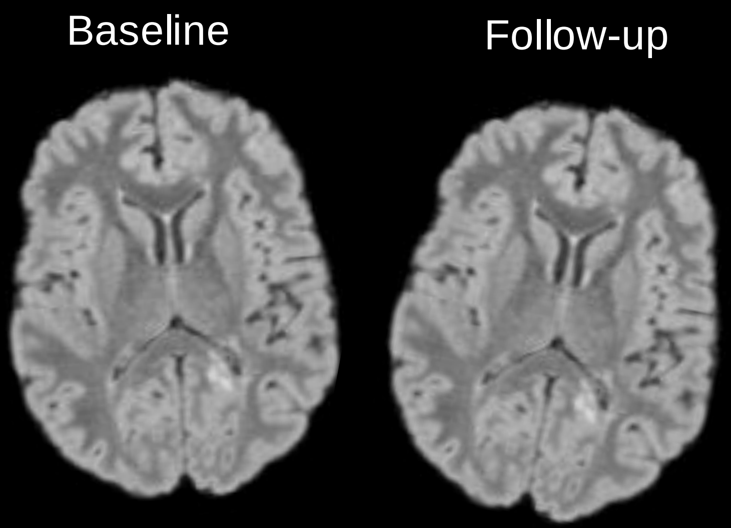


Detection


Callosal Angle: 84
Callosal Angle: 140
Callosal Angle: 87
Callosal Angle: 57
Callosal Angle: 66

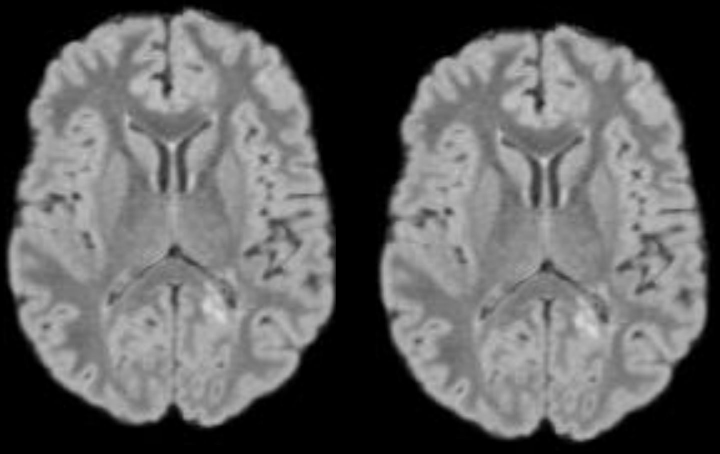
Tracking

- Tracking of objects over time to detect progression



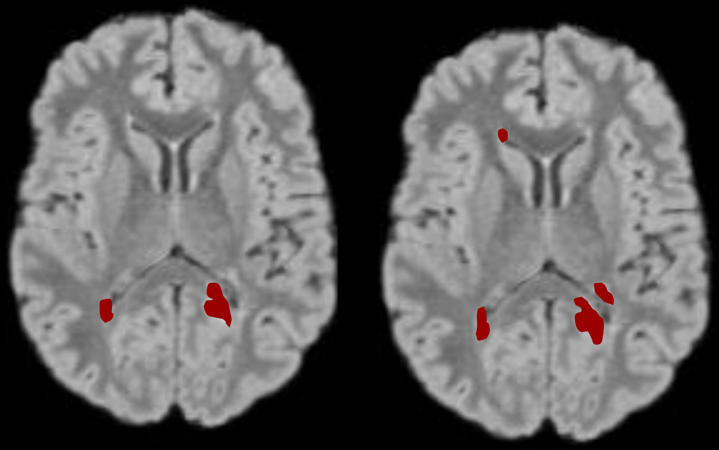
Step 1

Register images



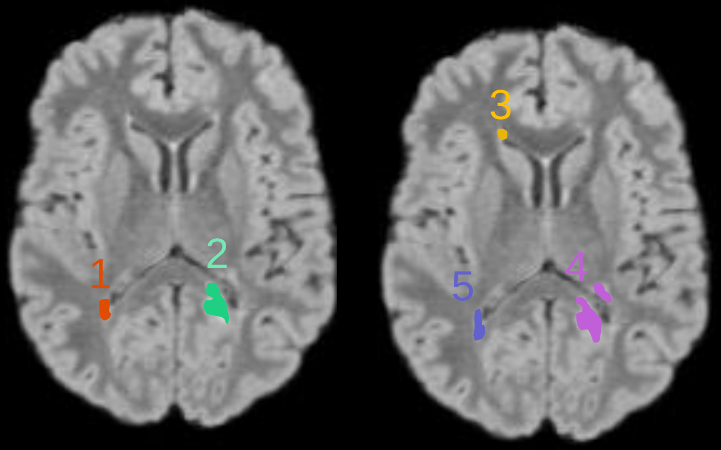
Step 2

Segment lesions



Step 3

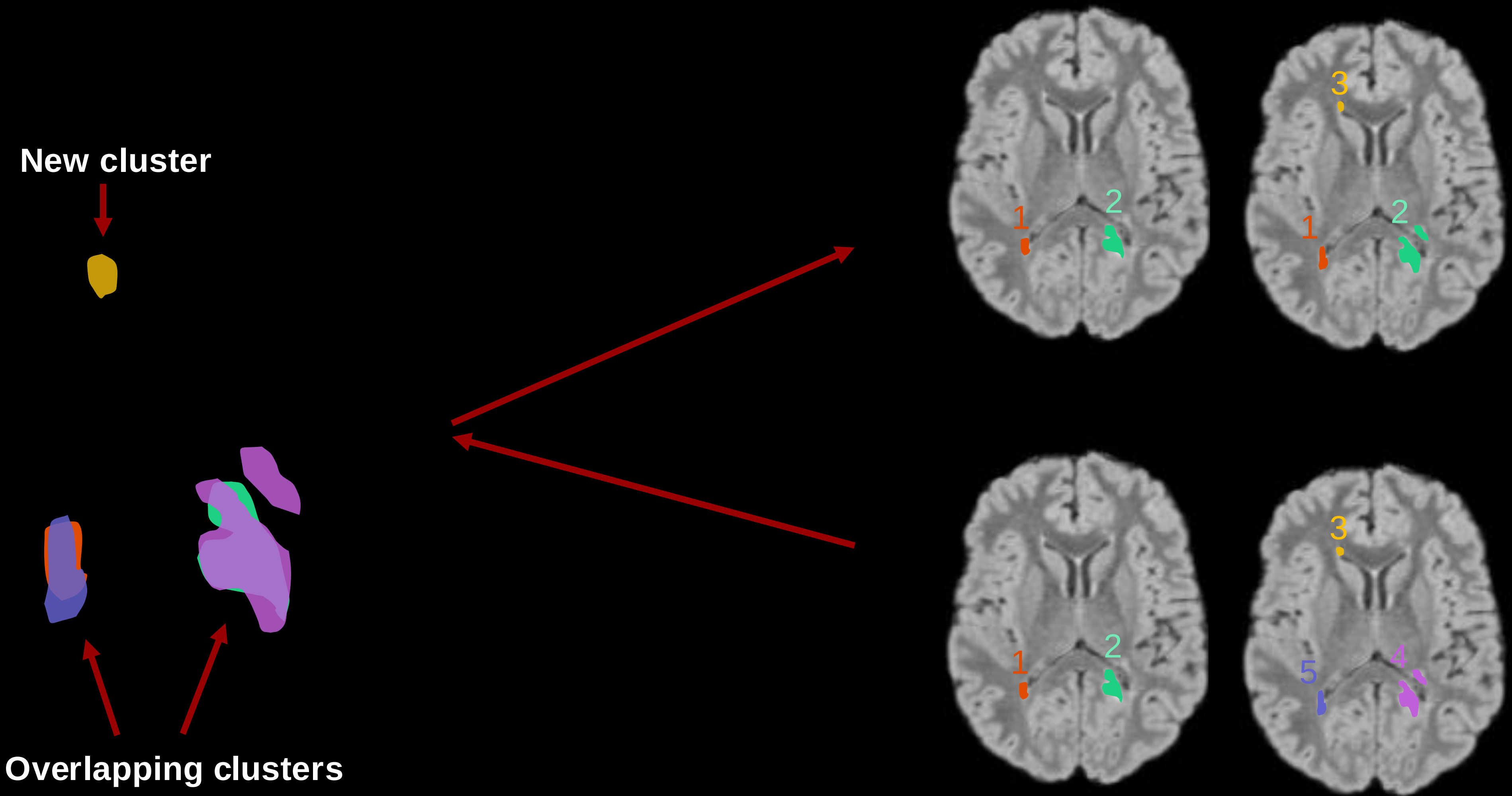
Connected component analysis



Tracking

Step 4

Global remapping

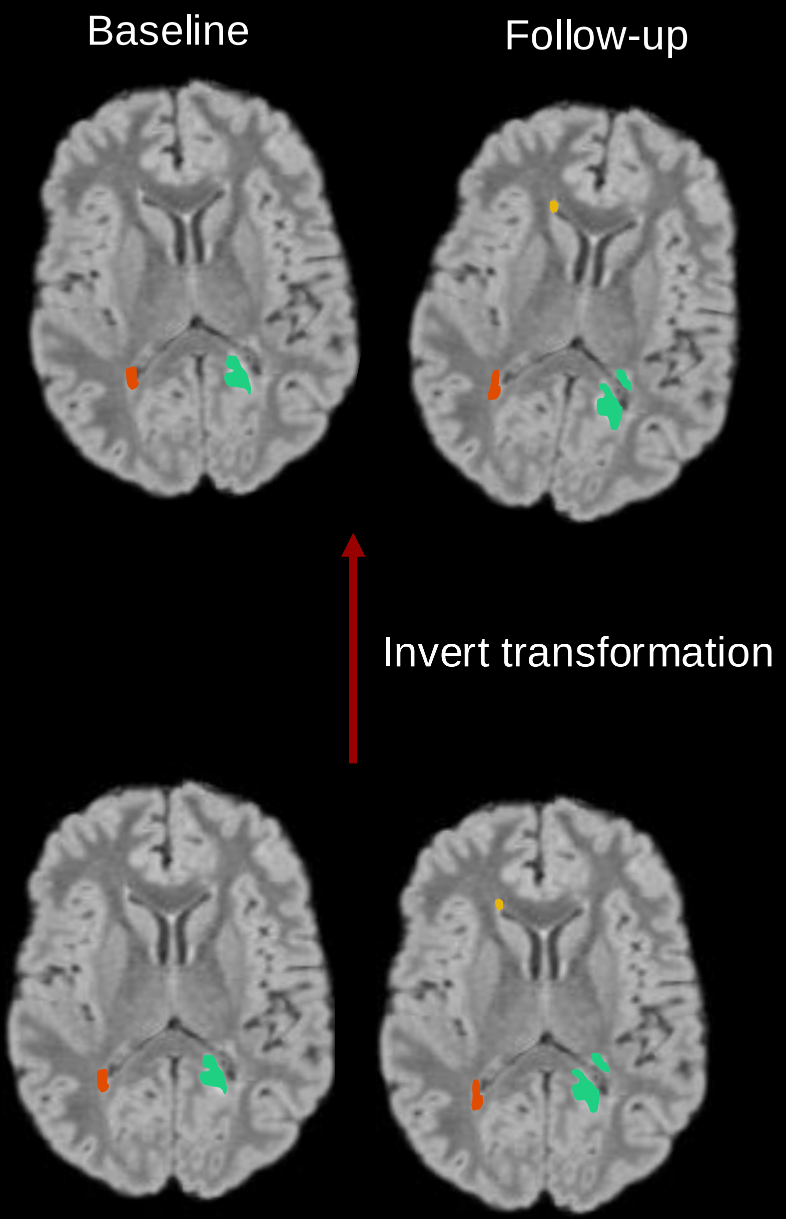


New cluster

Overlapping clusters

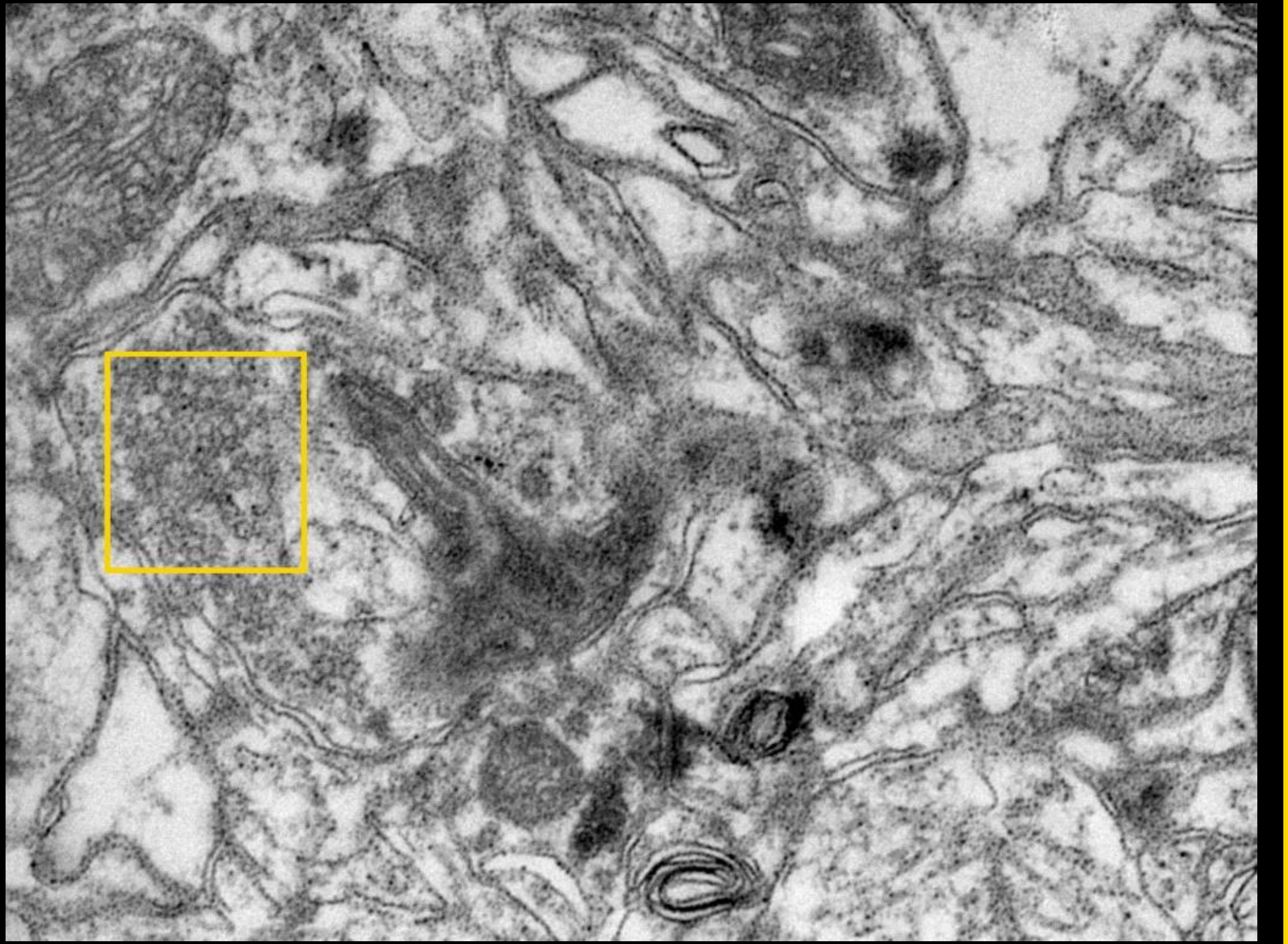
Tracking

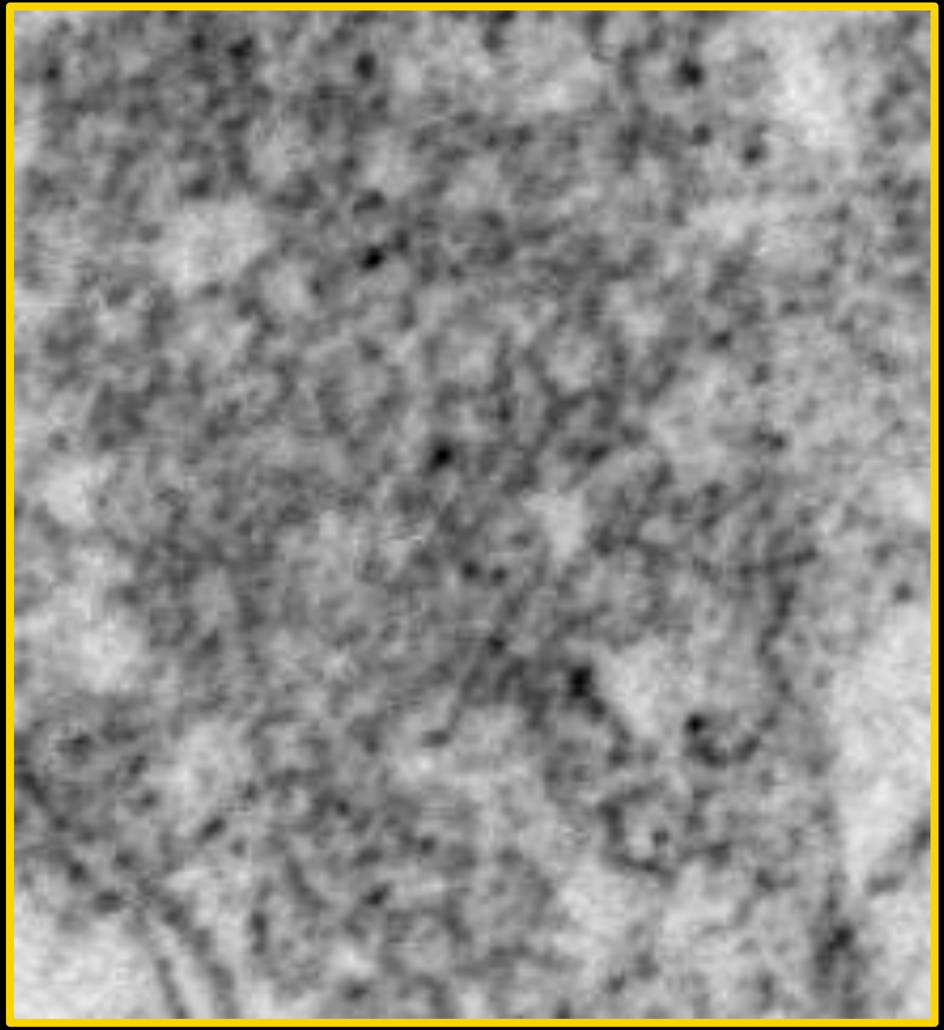
- Tracking of objects over time to detect progression



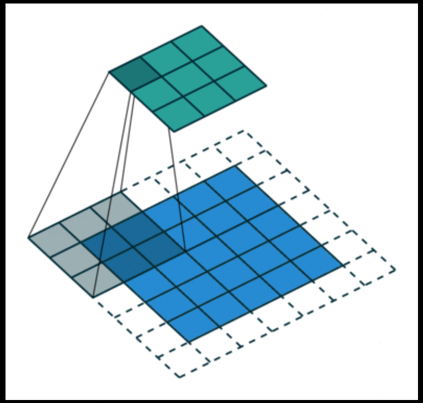
Classification (and more)

- Template matching
- Feature engineering
- Random Forest
- Active Shape Models
- Active Contours



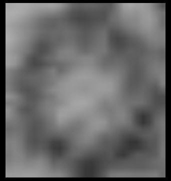


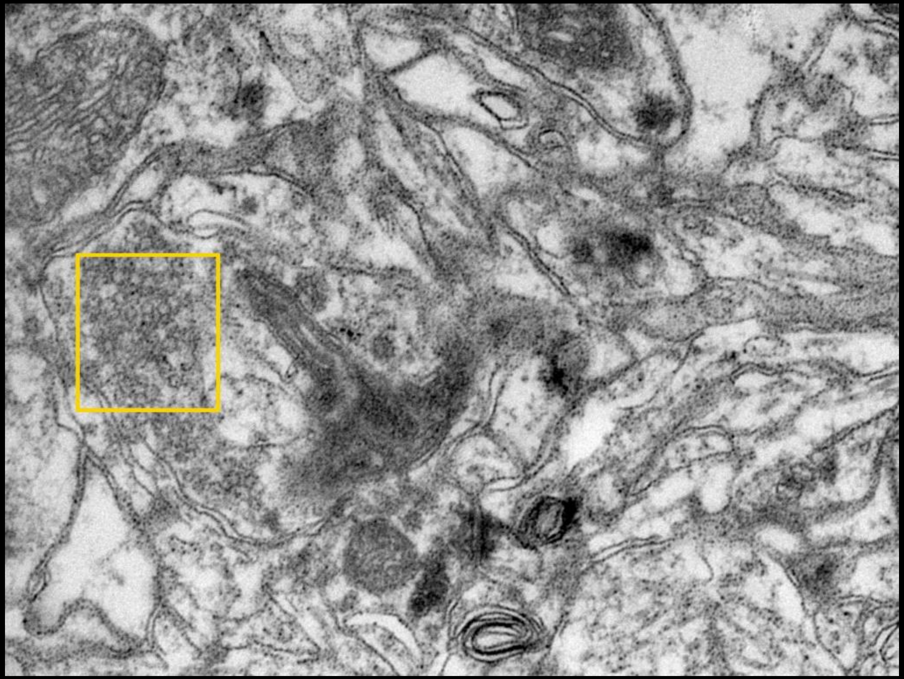
Template matching

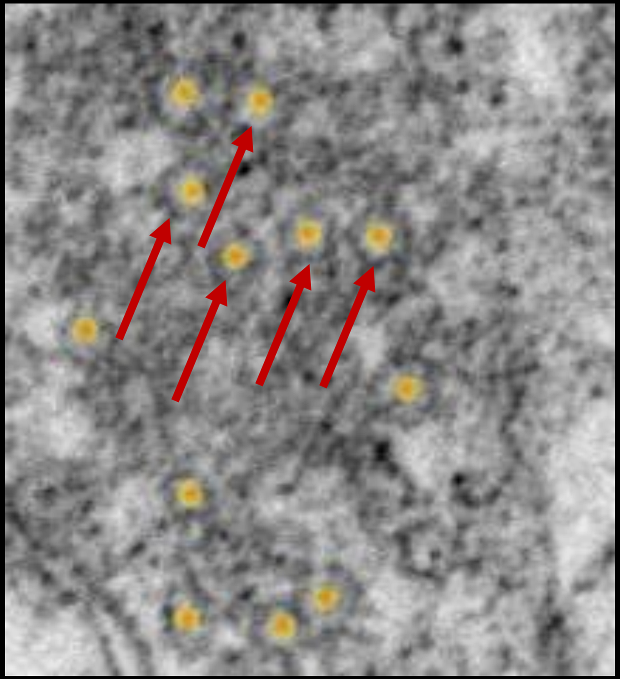


$$g(x,y) = \sum_{j=-R}^{R} \sum_{i=-R}^{R} h(i,j) \cdot f(x+i, y+j)$$

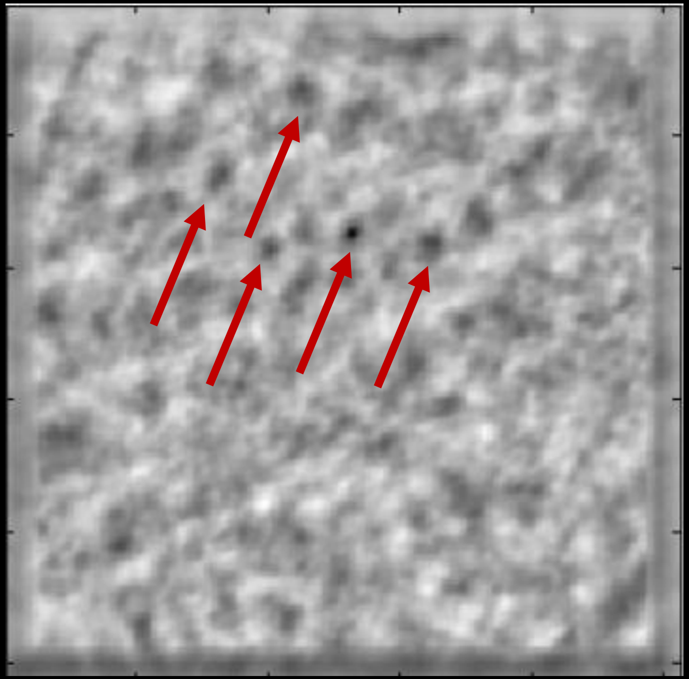
Examples of h:







Reference

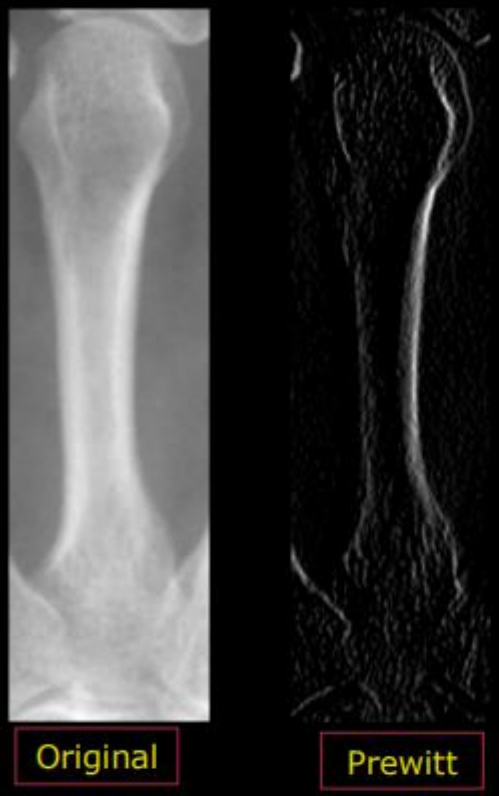


Resulting g

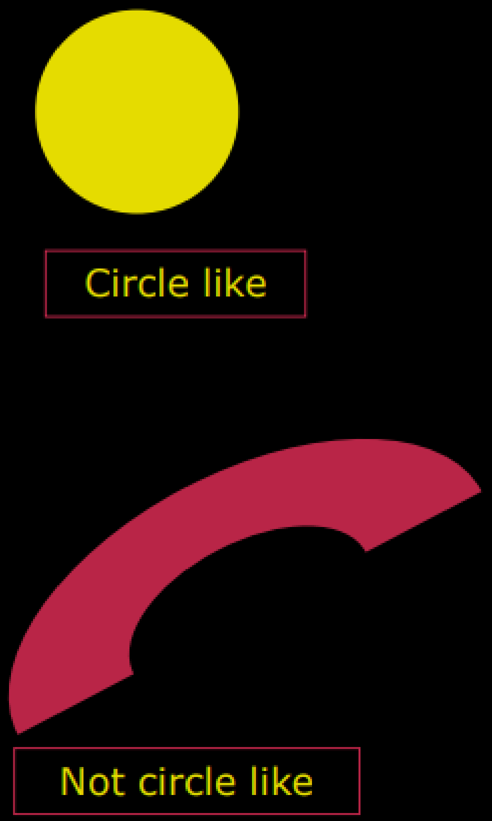
Feature engineering

What is relevant to know about this image to classify each voxel/pixel?

- Edges?
- Shapes?



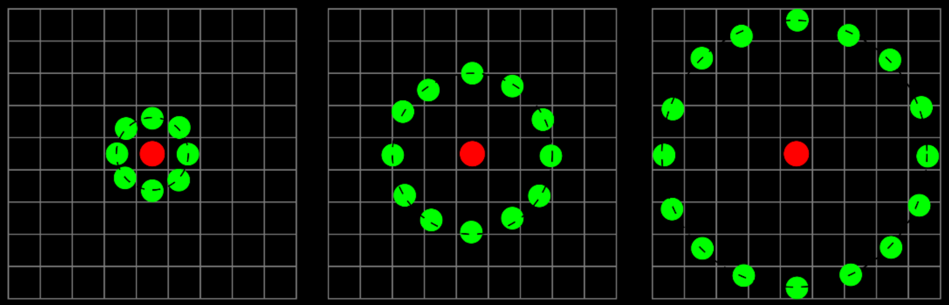
Week #4, Filtering



Week #5, Blob features

Feature engineering – Local Binary Patterns

Tunable parameters include radius (distance between center and points) and number of points on grid



Read points

6	8	15
15	10	17
10	9	20

Compare to center

0	0	1
1	0	1
1	0	1

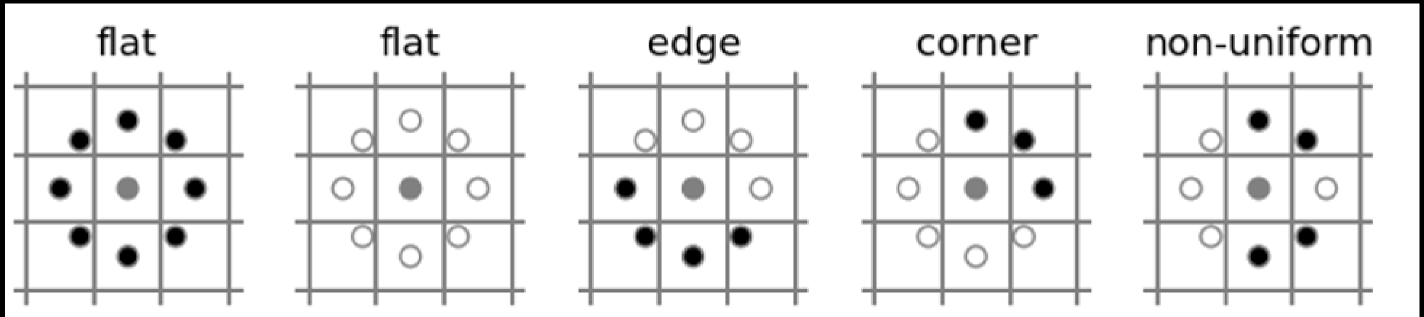
Binary

00111011

Decimal

59

$0x2^7 + 0x2^6 + 1x2^5 + 1x2^4 + 1x2^3 + 0x2^2 + 1x2^1 + 1x2^0$



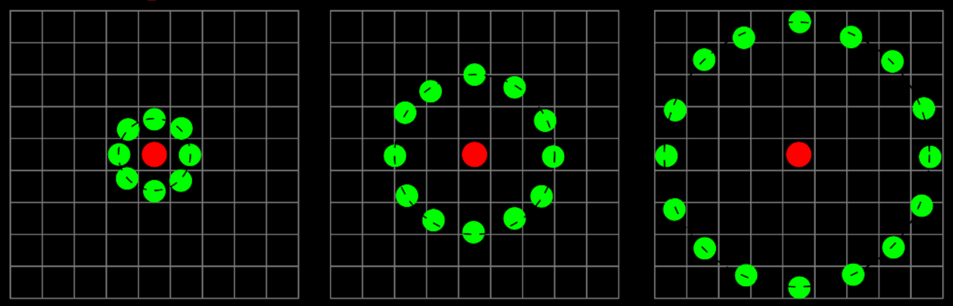
Quiz 5

- Given the read matrix, what is the calculated LBP value
  - – 163
  - – 167
  - – 171
  - – 180

Read matrix

6	4	6
15	5	4
10	9	3

From previous slide:



Read points

6	8	15
15	10	17
10	9	20

Compare to center

0	0	1
1	0	1
1	0	1

Binary

00111011

Decimal

59

$0x2^{7} + 0x2^{6} + 1x2^{5} + 1x2^{4} + 1x2^{3} + 0x2^{2} + 1x2^{1} + 1x2^{0}$

Quiz 5

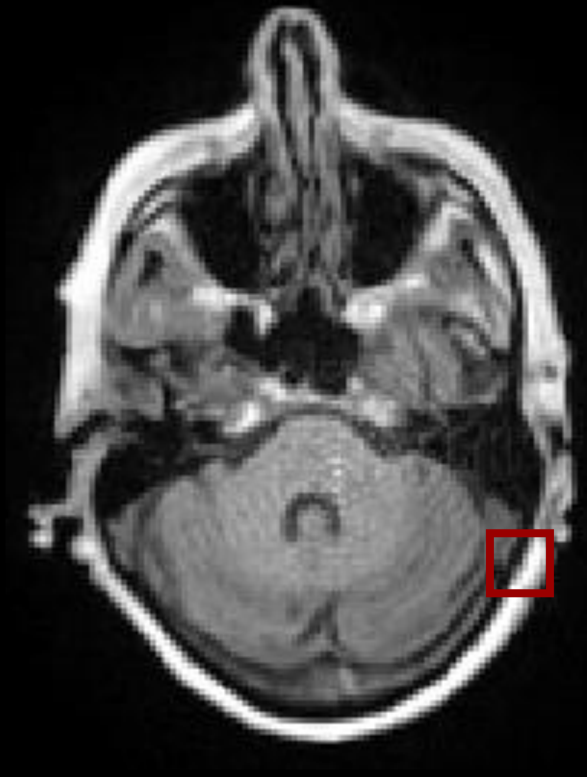
- Given the read matrix, what is the calculated LBP value
  - 163
  - 167
  - 171
  - 180

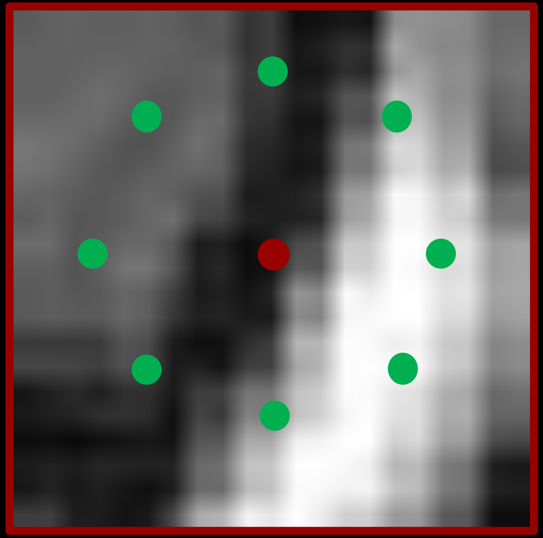
Read matrix

6	4	6
15	5	4
10	9	3

$$1\ 0\ 1\ 0\ 0\ 1\ 1\ 1 = 128 + 32 + 4 + 2 + 1 = 167$$

Feature engineering – Local Binary Patterns





Read points

30	20	35
28	10	41
15	37	45

Compare to center

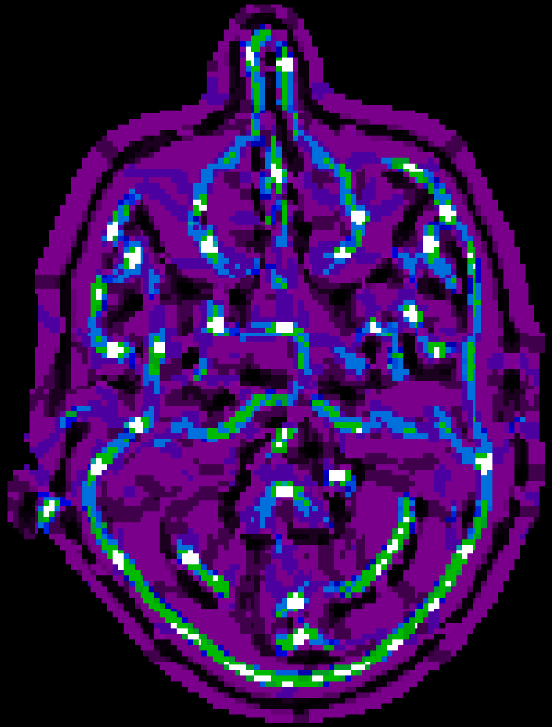
1	1	1
1	0	1
1	1	1

Binary

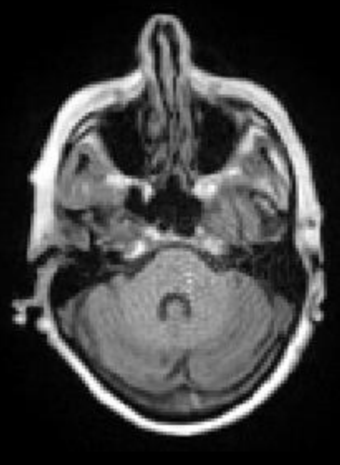
11111111

Decimal

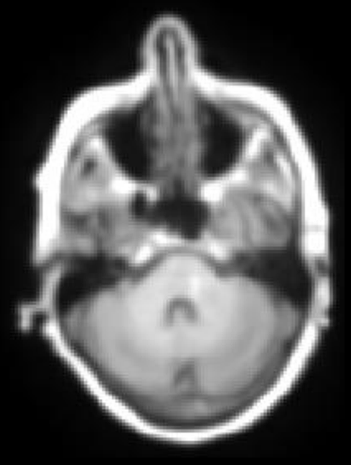
255



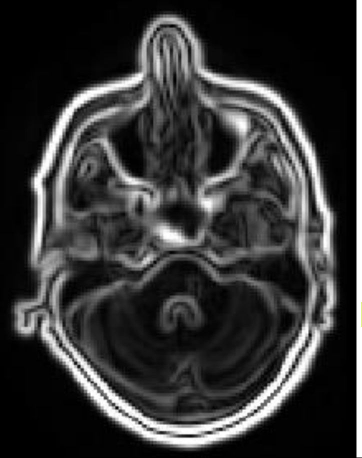
Feature engineering



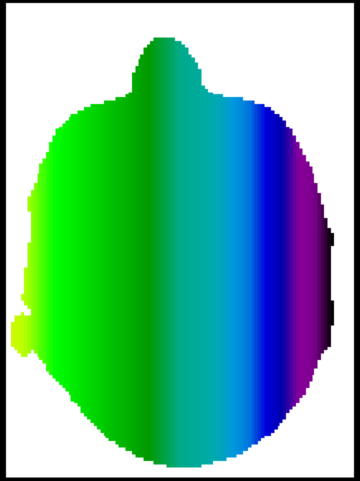
Normalized



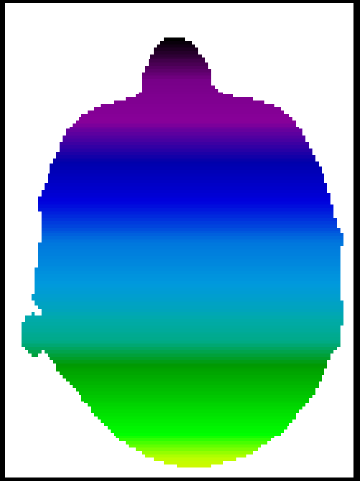
Blurred



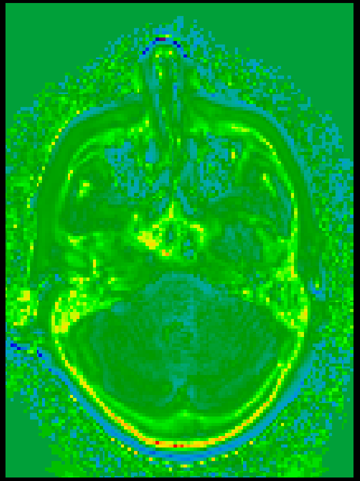
Gradient magnitude



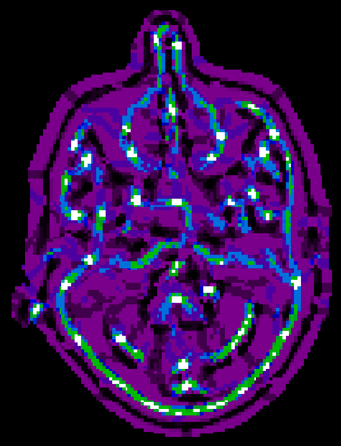
Spatial-x



Spatial-y



R2*



LBP

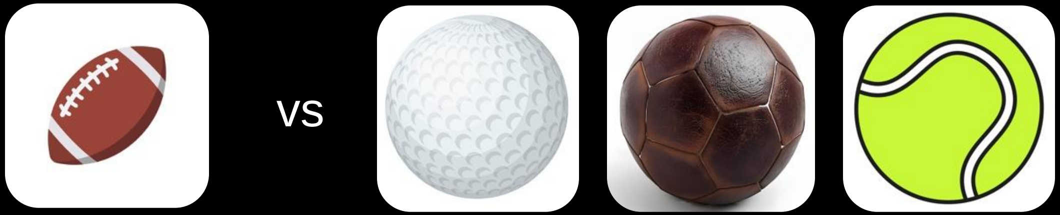
How to we combine these into a voxel classification model?

Which features are relevant for image classification?

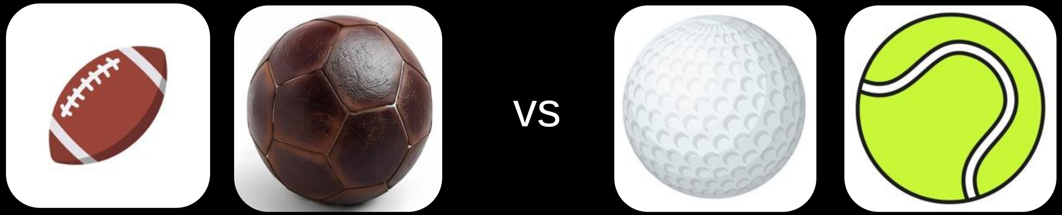
Go pick up the golf ball

Image features

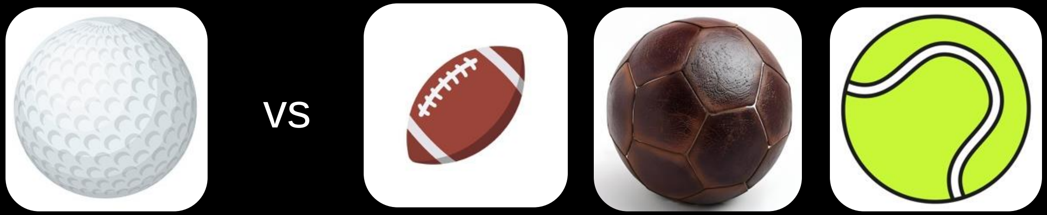
Roundness



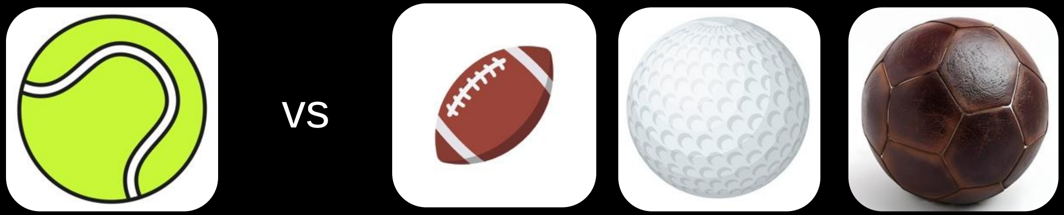
Size (Largest diameter > 10 cm)



Convex (yes or no)



Color (is_yellow)



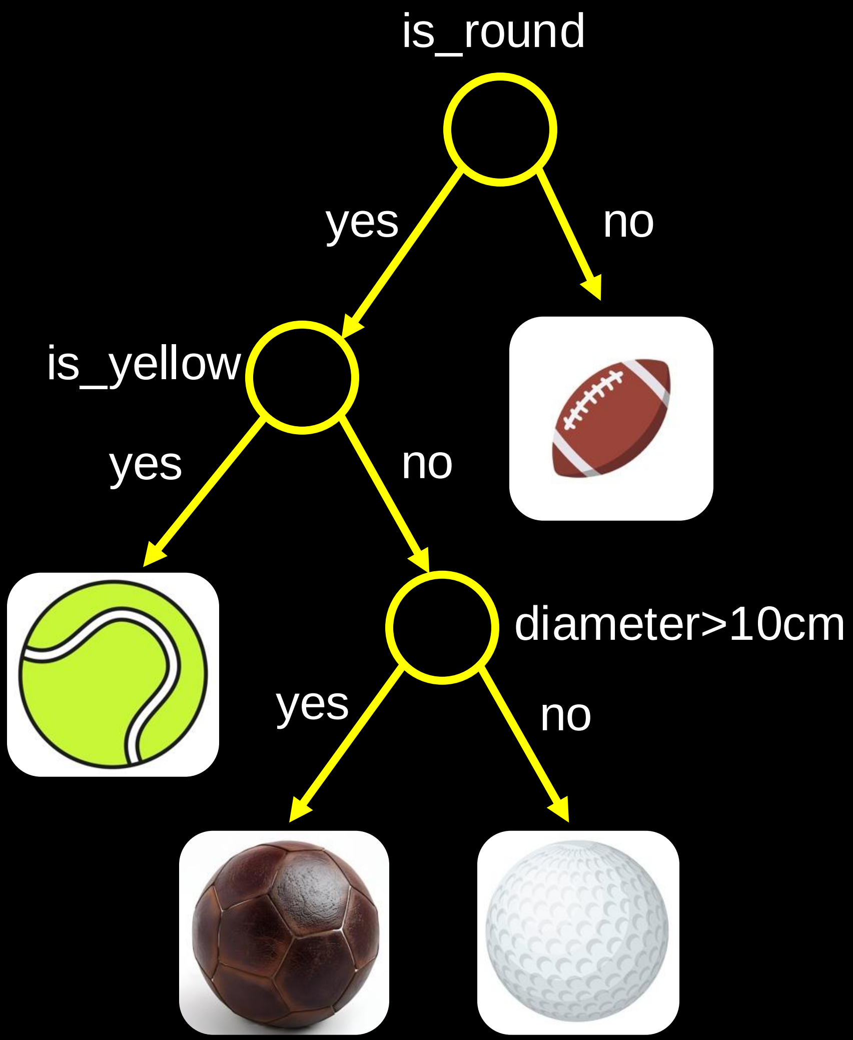
From features to decision trees

Available features:

- Roundness (is_round)
- Color (is_yellow)
- Size (diameter>10cm)

Rules:

- Yes means you go left
- Leaves cannot be empty



Build your own decision tree

Rules:

- Yes means you go left
- Leaves cannot be empty

GROUP A (Birthdate is odd)

HAS_FUR

EATS_MEAT

HAS_MANE

LIVES_IN_WATER

HAS_TEETH

GROUP B (Birthdate is even)

BUILDS_DAM

ATTACKS_HUMANS

HAS_TEETH

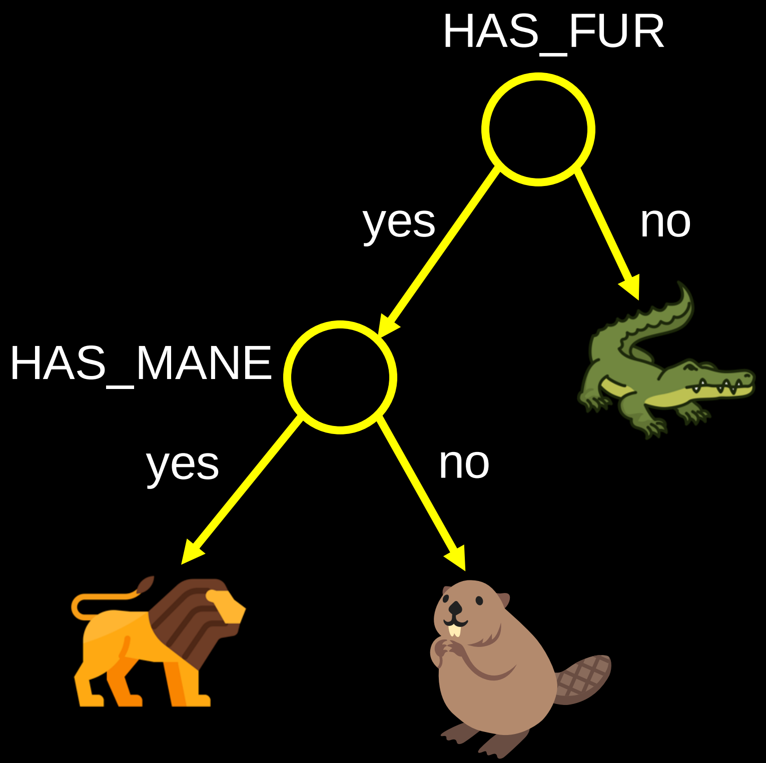
LIVES_IN_WATER

Build your own decision tree

~~LIVES_IN_WATER~~ ~~HAS_TEETH~~

HAS_FUR ~~EATS_MEAT~~ HAS_MANE

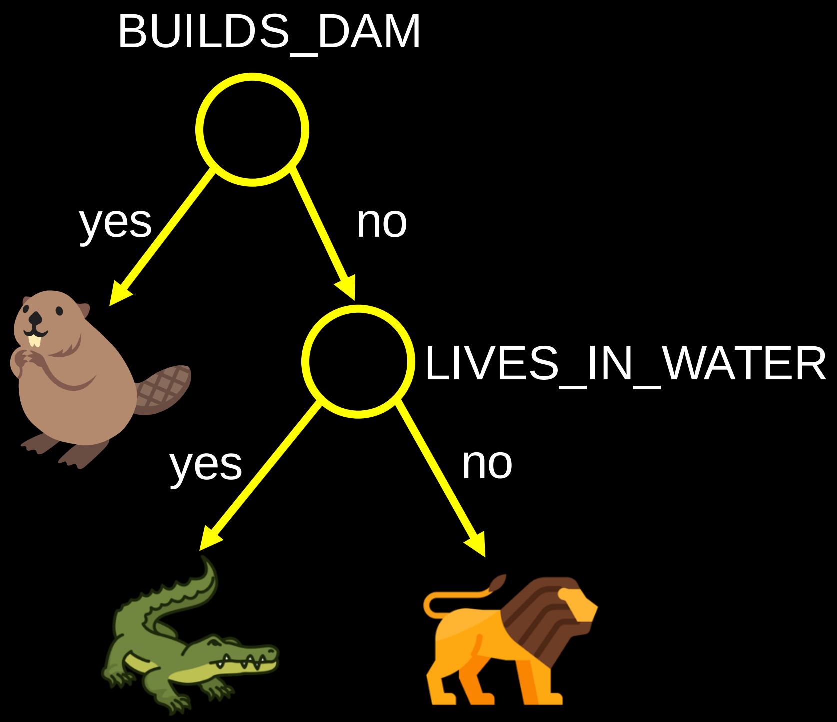
GROUP A



BUILDS_DAM ~~ATTACKS_HUMANS~~

~~HAS_TEETH~~ LIVES_IN_WATER

GROUP B



FROM TREES TO A (RANDOM) FOREST

Random Forest

Variables

Data points

Observations (Training data)

[id]	[x_0, x_1, …]	[y]
[0]	[YES, YES, …]	🦫
[1]	[YES, NO, …]	🦁
	…	
	…	
[n]	[NO, YES, …]	🦁

Random Forest

Observations (Training data)

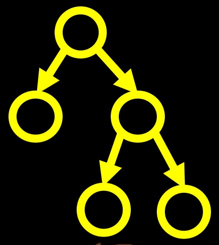
[id]	[x_0, x_1, …	]	[y]
[0]	[YES, YES, …	]	🦫
[1]	[YES, NO, …	]	🦁
	…		
	…		
[n]	[NO, YES, …	]	🦁

- Why random?
  - Each tree sees a random subset of the variables
  - Each tree sees a random subset of the data points with replacement (Bootstrap)
- Multiple trees make a forest

Majority voting of result 🦁

[id] [x_0, x_4]

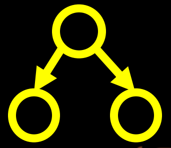
[3
2
3
1
6
8]



🦁

[id] [x_5, x_9]

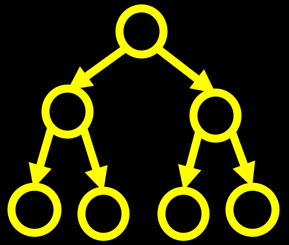
[0
2
1
3
7
5]



🦁

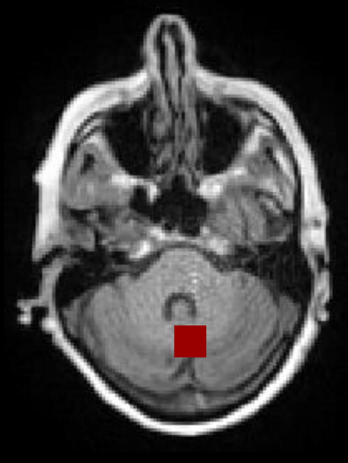
[id] [x_2, x_6, x_{11}]

[2
5
5
9
2
0]

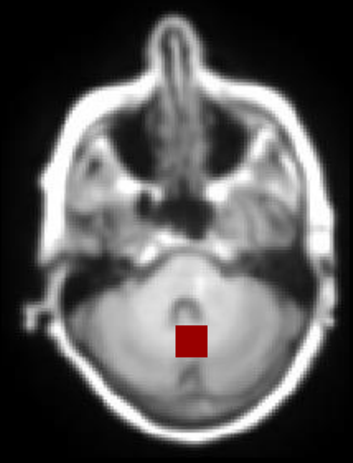


🐊

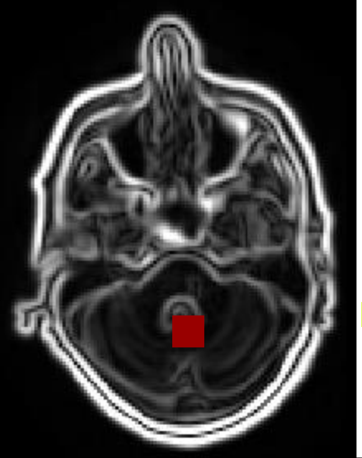
Feature engineering



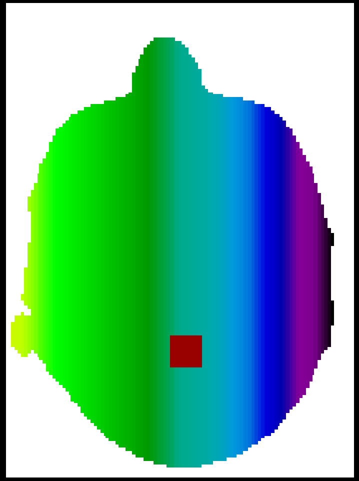
Normalized



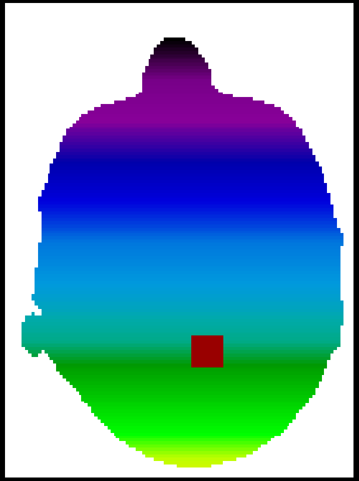
Blurred



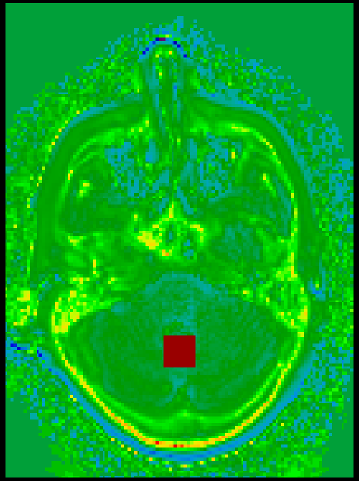
Gradient magnitude



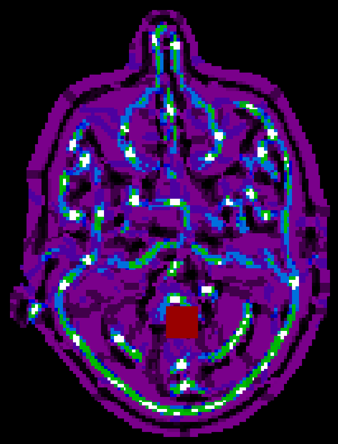
Spatial-x



Spatial-y



R2*

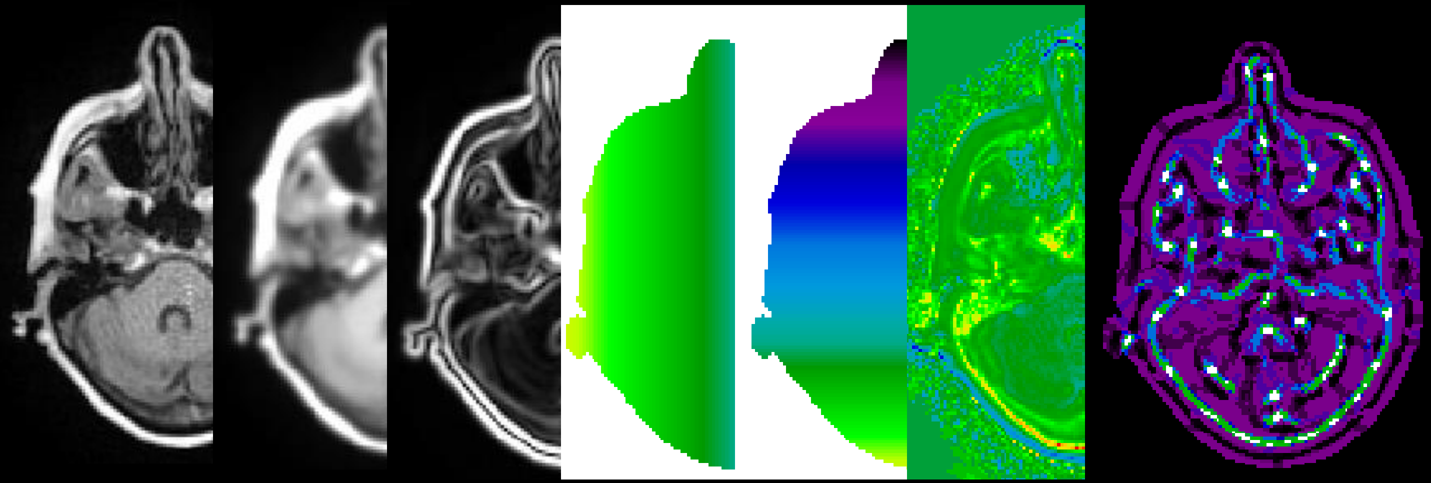


LBP

How to we combine these into a voxel classification model?

[0.65, 0.61, 0.5, 5, -10, 0.25, 231, …]

Feature engineering



Reference value for each voxel

Repeat for all voxels:

[0.65, 0.61, 0.5, 5, -10, 0.25, 231, …]
[0.45, 0.66, 0.4, 6, -12, 0.24, 251, …]
…
…
[0.87, 0.41, 0.1, 2, 25, 0.55, 131, …]

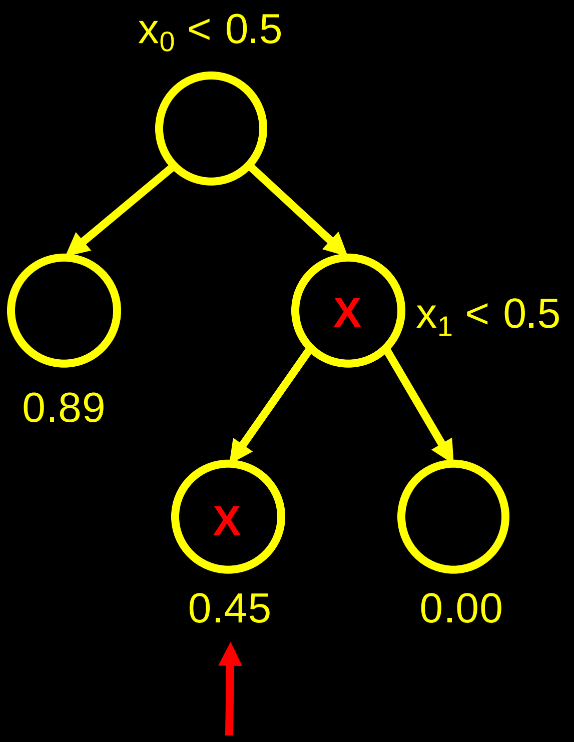
Normalize to 0-1 range

[0.61, 0.59, 0.63, 0.5, 0.11, 0.25, 0.88, …] [0.00]
[0.41, 0.65, 0.45, 0.6, 0.08, 0.24, 0.95, …] [0.89]
…
…
[0.81, 0.38, 0.12, 0.2, 0.31, 0.55, 0.45, …] [0.45]

Random Forest

[id]	[x_0, x_1, …]	[y]
[0]	[0.61, 0.59, 0.63, 0.5, 0.11, 0.25, 0.88, …]	[0.00]
[1]	[0.41, 0.65, 0.45, 0.6, 0.08, 0.24, 0.95, …]	[0.89]
	…	
	…	
[n]	[0.81, 0.38, 0.12, 0.2, 0.31, 0.55, 0.45, …]	[0.45]

Decision tree



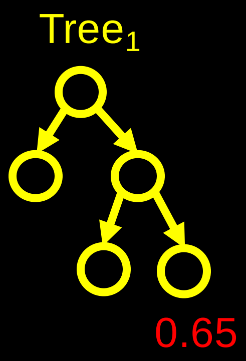
- Multiple trees make a forest
- Why random?
  - Each tree sees a random data sample with replacement (Bootstrap)
  - Each tree sees a random subset of the variables

New datapoint: **[0.65, 0.33, …,]**

Mean of results (Aggregating): $\bar{y} = \frac{1}{n}\sum_i \bar{y_i} = \frac{1}{3}(0.65 + 0.61 + 0.78) = \underline{0.68}$

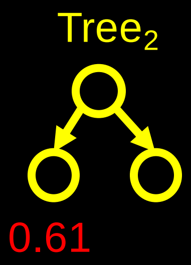
[id] [x_0, x_4]

[3
2
3
1
6
8]



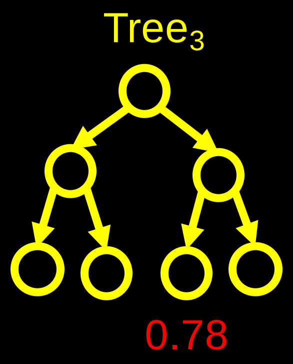
[id] [x_5, x_9]

[0
2
1
3
7
5]



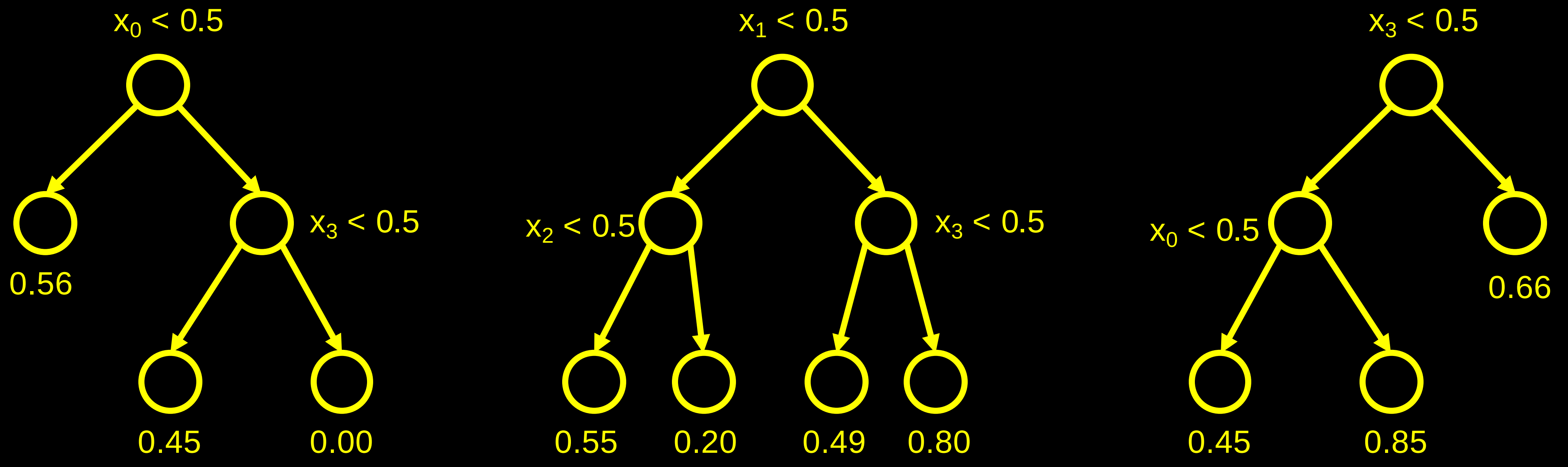
[id] [x_2, x_6, x_{11}]

[2
5
5
9
2
0]



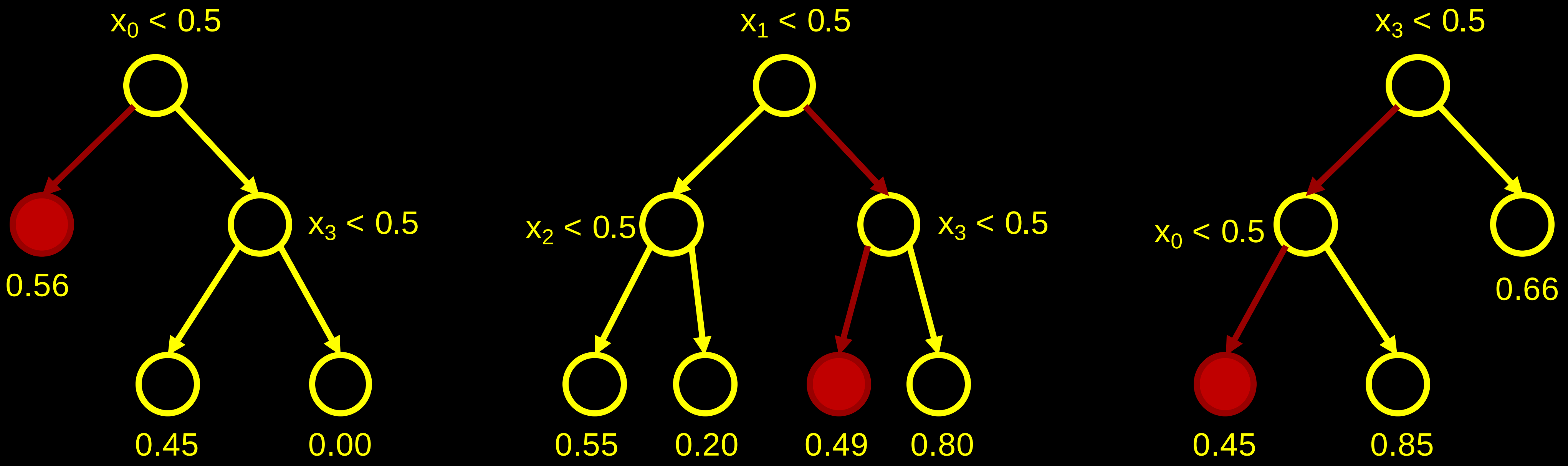
Quiz 6

- Using the 3 trained trees below, what is the predicted value after aggregating the output?
- Input data: [0.49, 0.56, 0.99, 0.32]
- Options:
  - – 0.45
  - – 0.50
  - – 0.80
  - – 0.48



Quiz 6

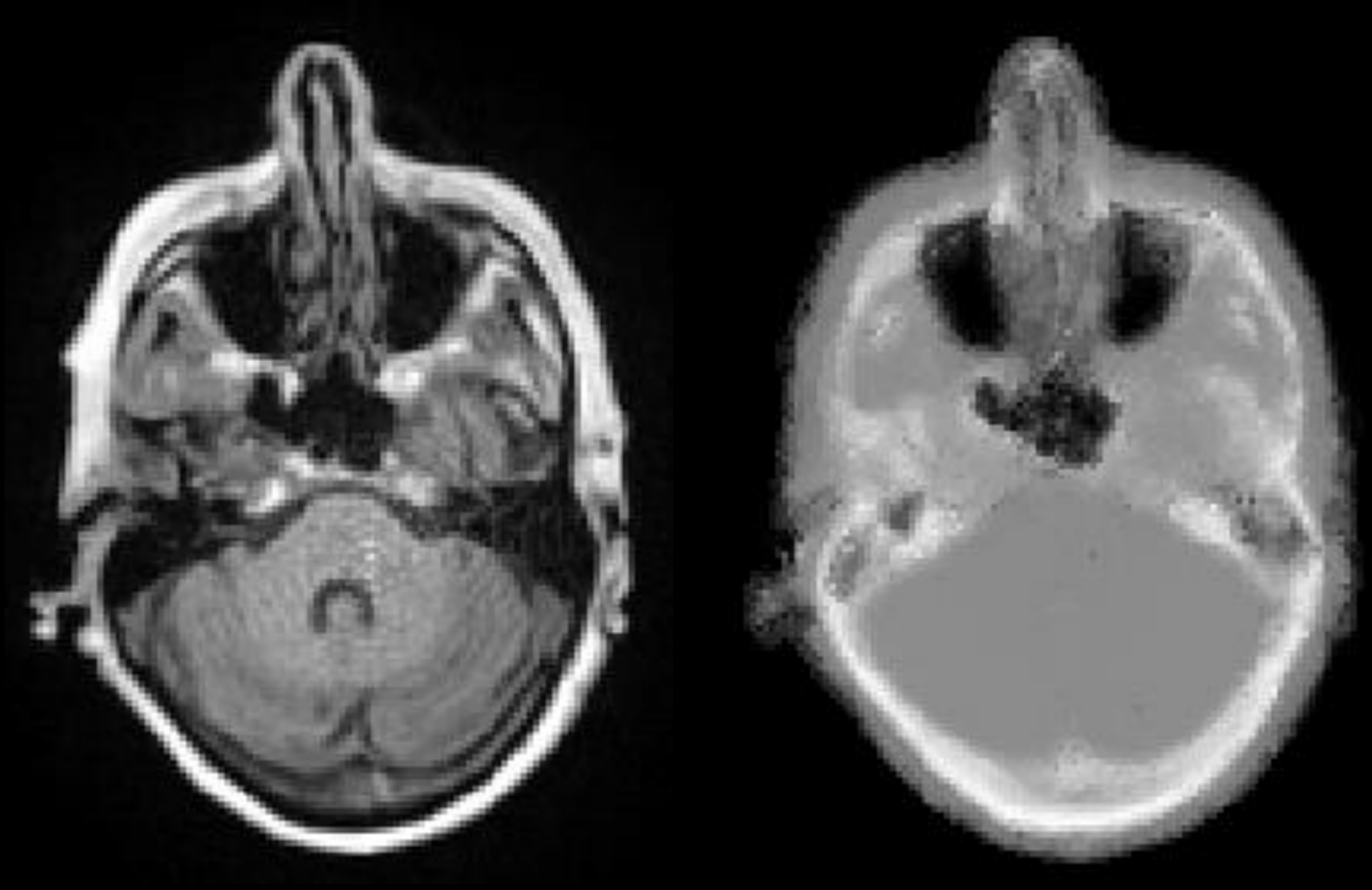
- Using the 3 trained trees below, what is the predicted value after aggregating the output?
- Input data: [0.49, 0.56, 0.99, 0.32]
- Options:
  - 0.45
  - 0.50
  - 0.80
  - 0.48



$$(0.56 + 0.49 + 0.45)/3 = 0.5$$

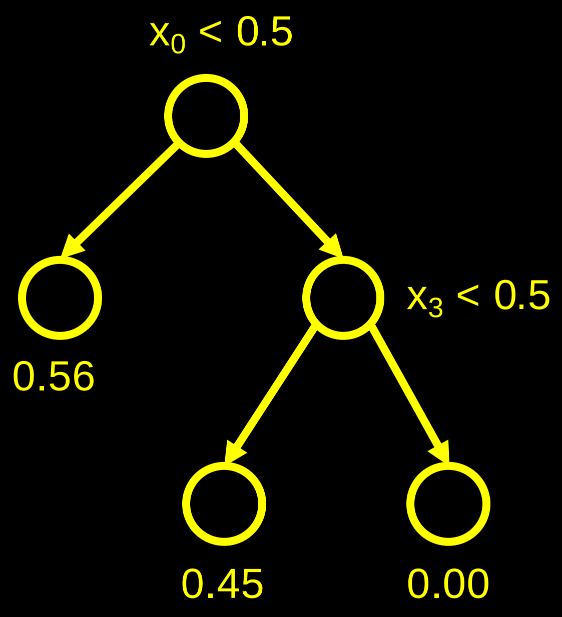
Random forest

- Example output:
  - 100 trees
  - n=25 patients
  - Features from
    - Original and filtered images
    - Edge enhanced
    - R2*
    - LBP
- Trained with RandomForestRegressor from sklearn



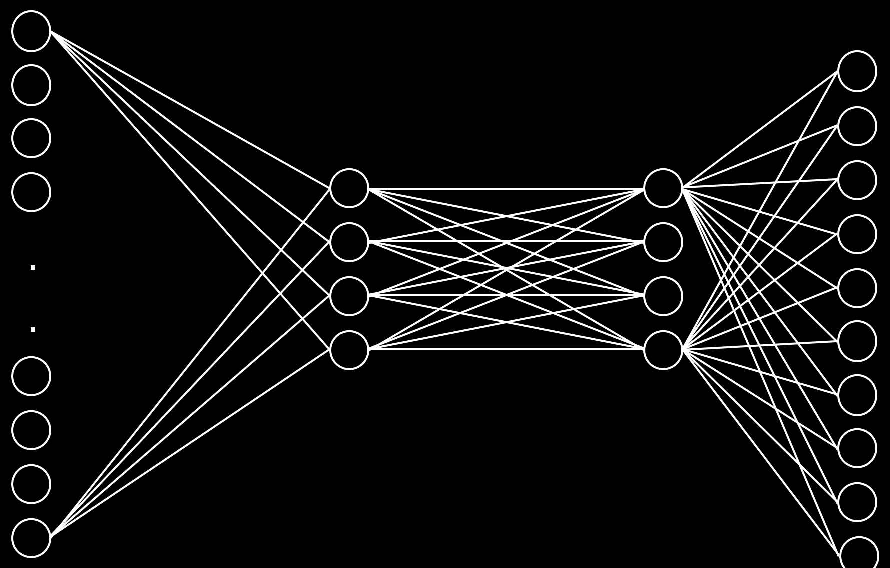
Increasing complexity..

Random Forest



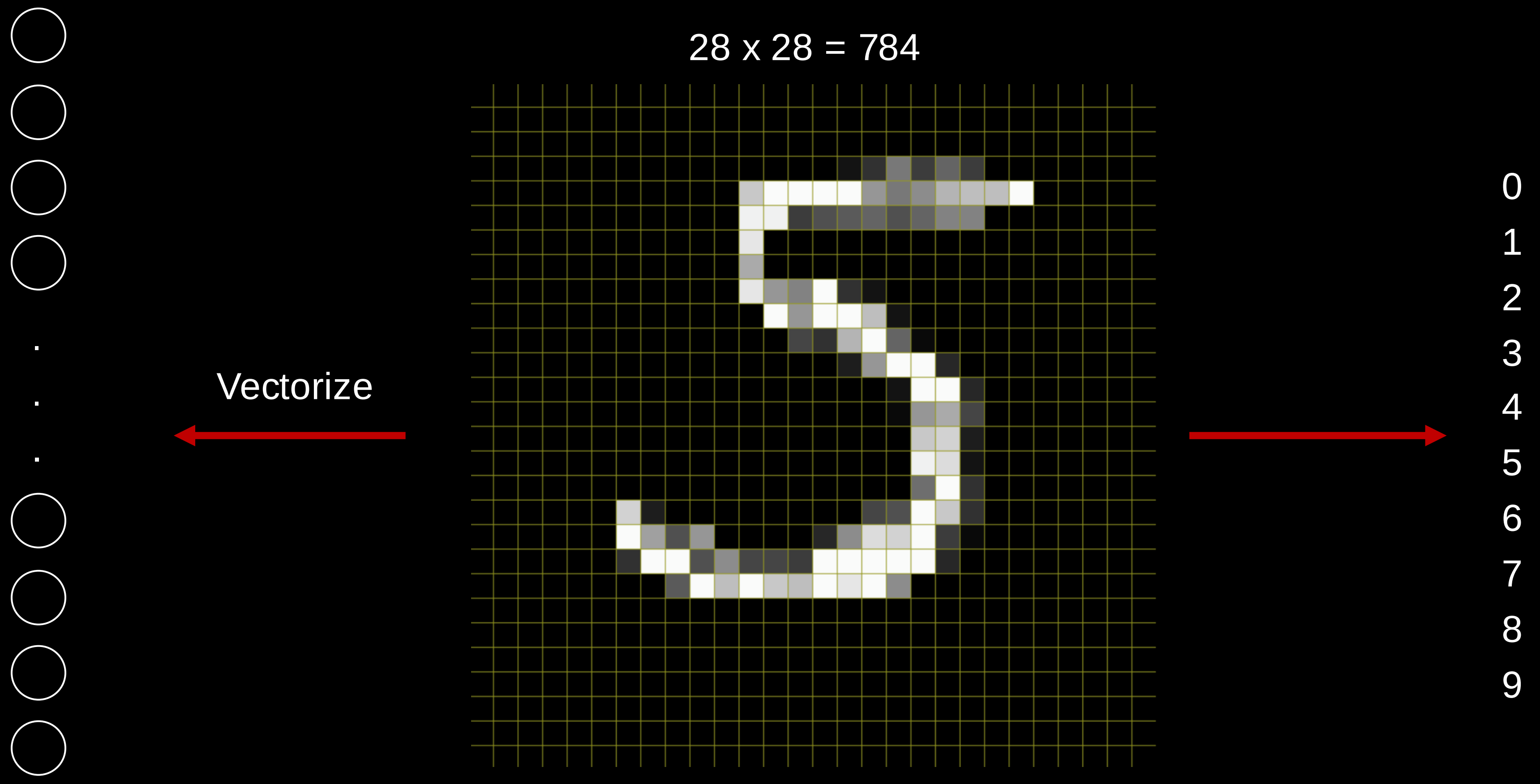
(Potential for) high level of interpretability

Neural network

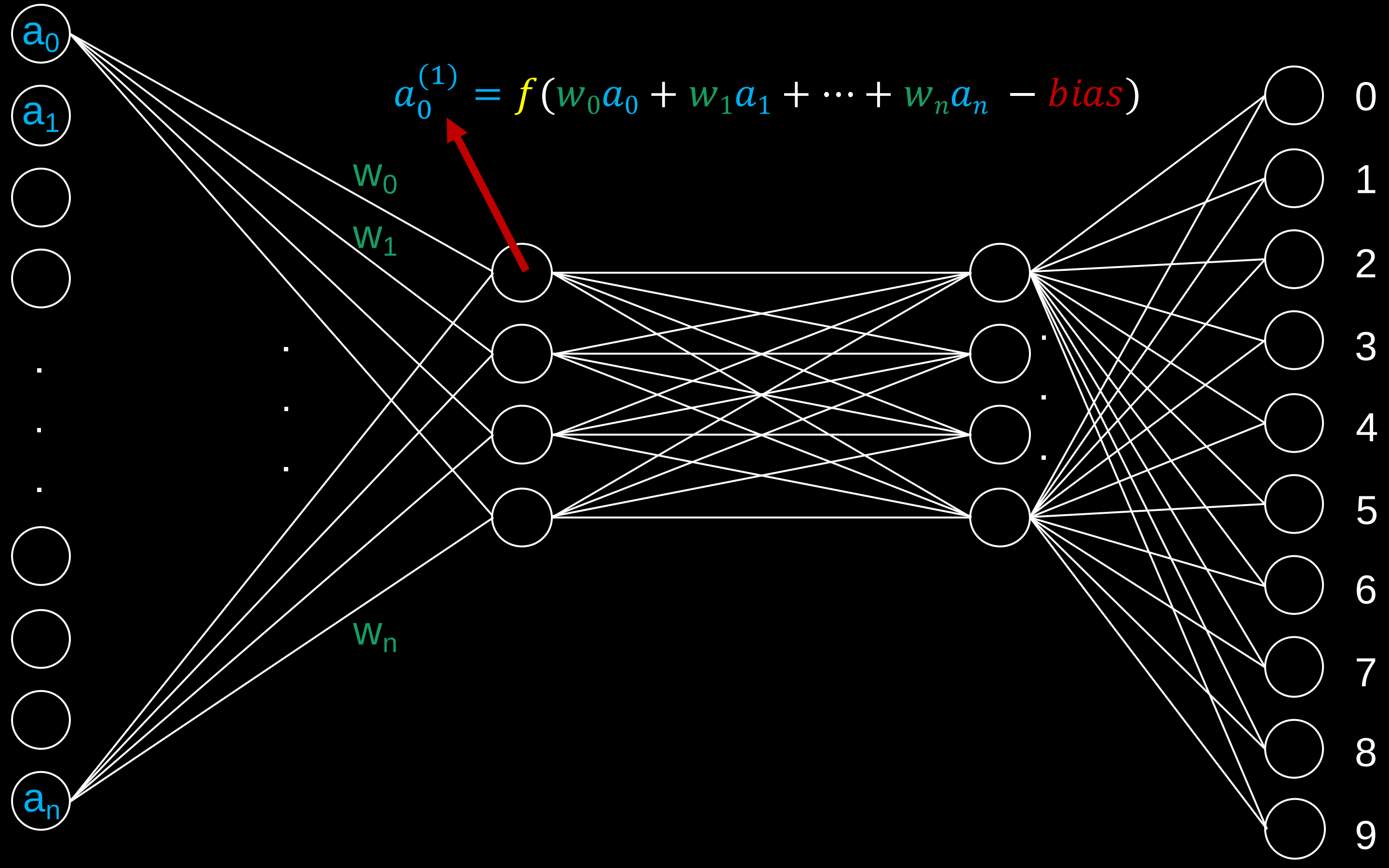


Limited interpretability

Neural Networks



Neural Networks



$$a_0^{(1)} = f(w_0 a_0 + w_1 a_1 + \cdots + w_n a_n - bias)$$

Input layer

Hidden layers

Output layer

- Each neuron contain a value, its ”activation”
  - The values in the input are the pixel values
  - The value at the last output layer represents the likelihood of that digit
  - f is an activation function (e.g. sigmoid)

# weights: 784x4+4x4+4x10

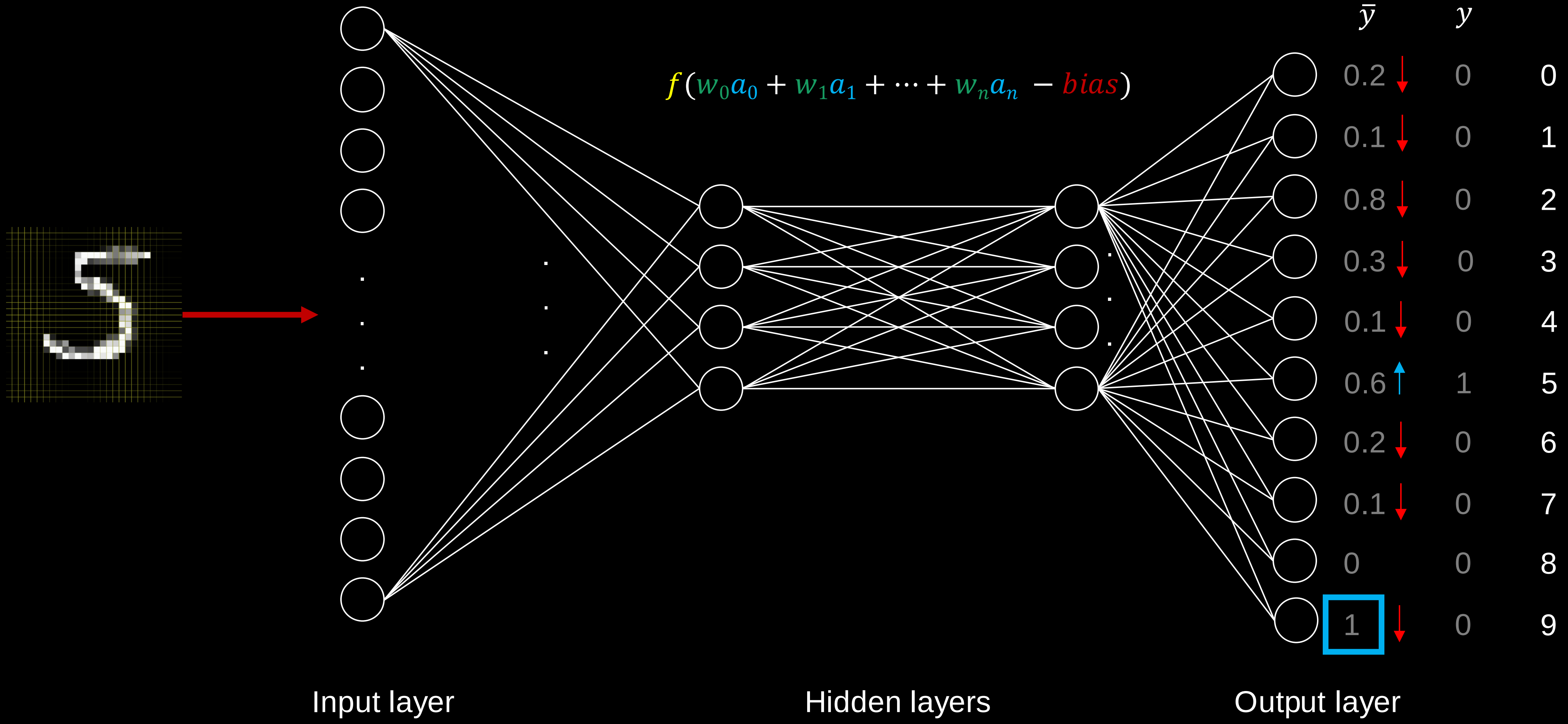
# biases: 4 + 4 + 10

Total parameters: 3,210

Neural Networks

”Cost” of the difference:

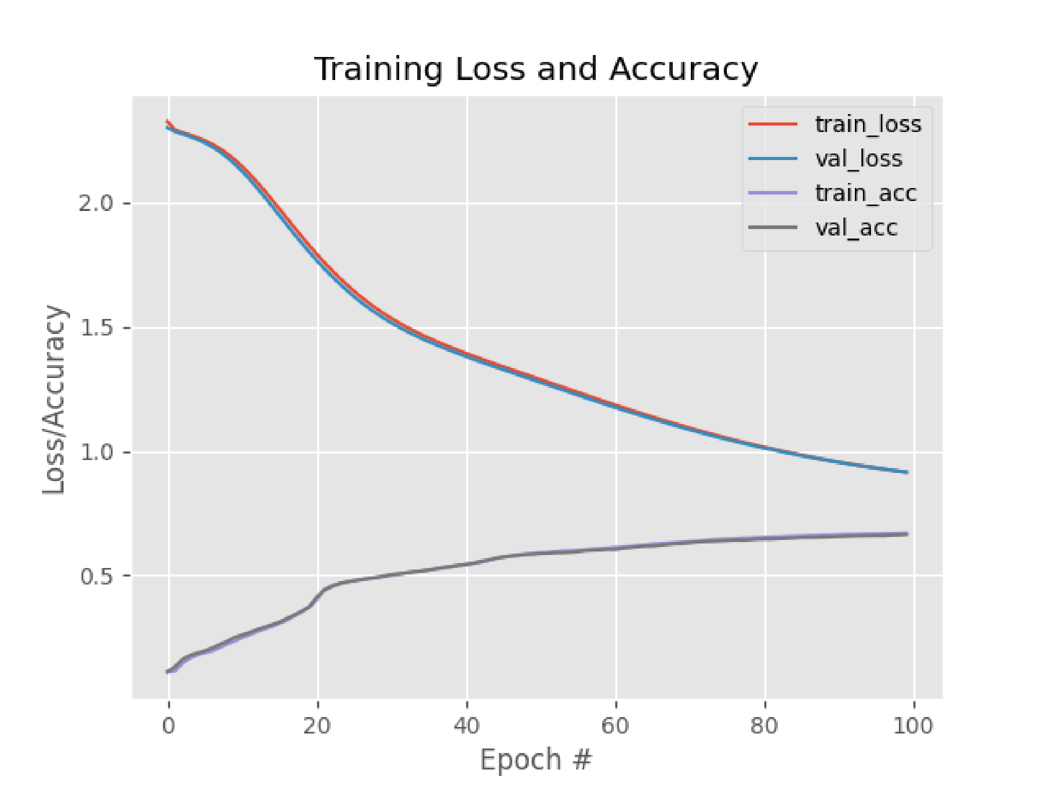
$$\sum(\bar{y}-y)^2$$



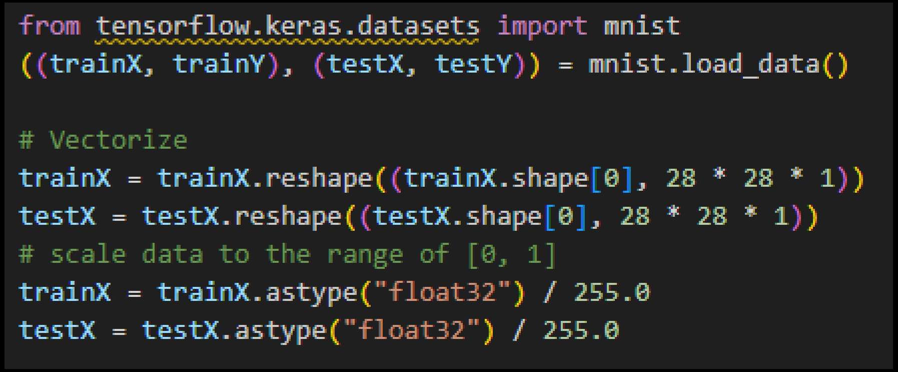
$$f(w_0a_0 + w_1a_1 + \cdots + w_na_n \; - bias)$$

$\bar{y}$	y	
0.2 ↓	0	0
0.1 ↓	0	1
0.8 ↓	0	2
0.3 ↓	0	3
0.1 ↓	0	4
0.6 ↑	1	5
0.2 ↓	0	6
0.1 ↓	0	7
0	0	8
1 ↓	0	9

Neural Networks

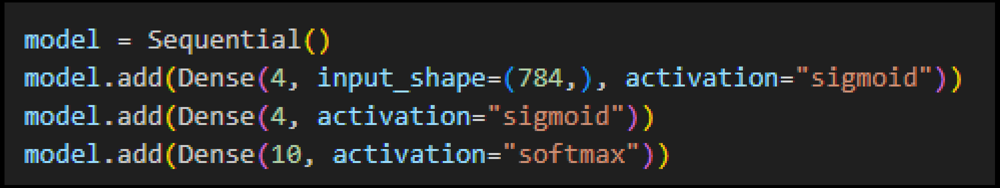


Load and prepare data



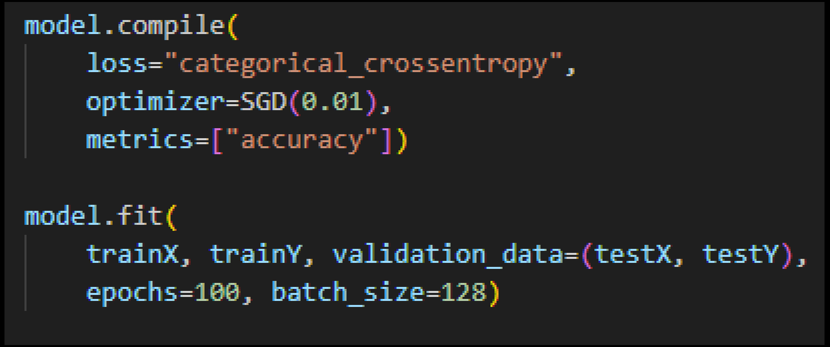
```
from tensorflow.keras.datasets import mnist
((trainX, trainY), (testX, testY)) = mnist.load_data()

# Vectorize
trainX = trainX.reshape((trainX.shape[0], 28 * 28 * 1))
testX = testX.reshape((testX.shape[0], 28 * 28 * 1))
# scale data to the range of [0, 1]
trainX = trainX.astype("float32") / 255.0
testX = testX.astype("float32") / 255.0
```

Define model



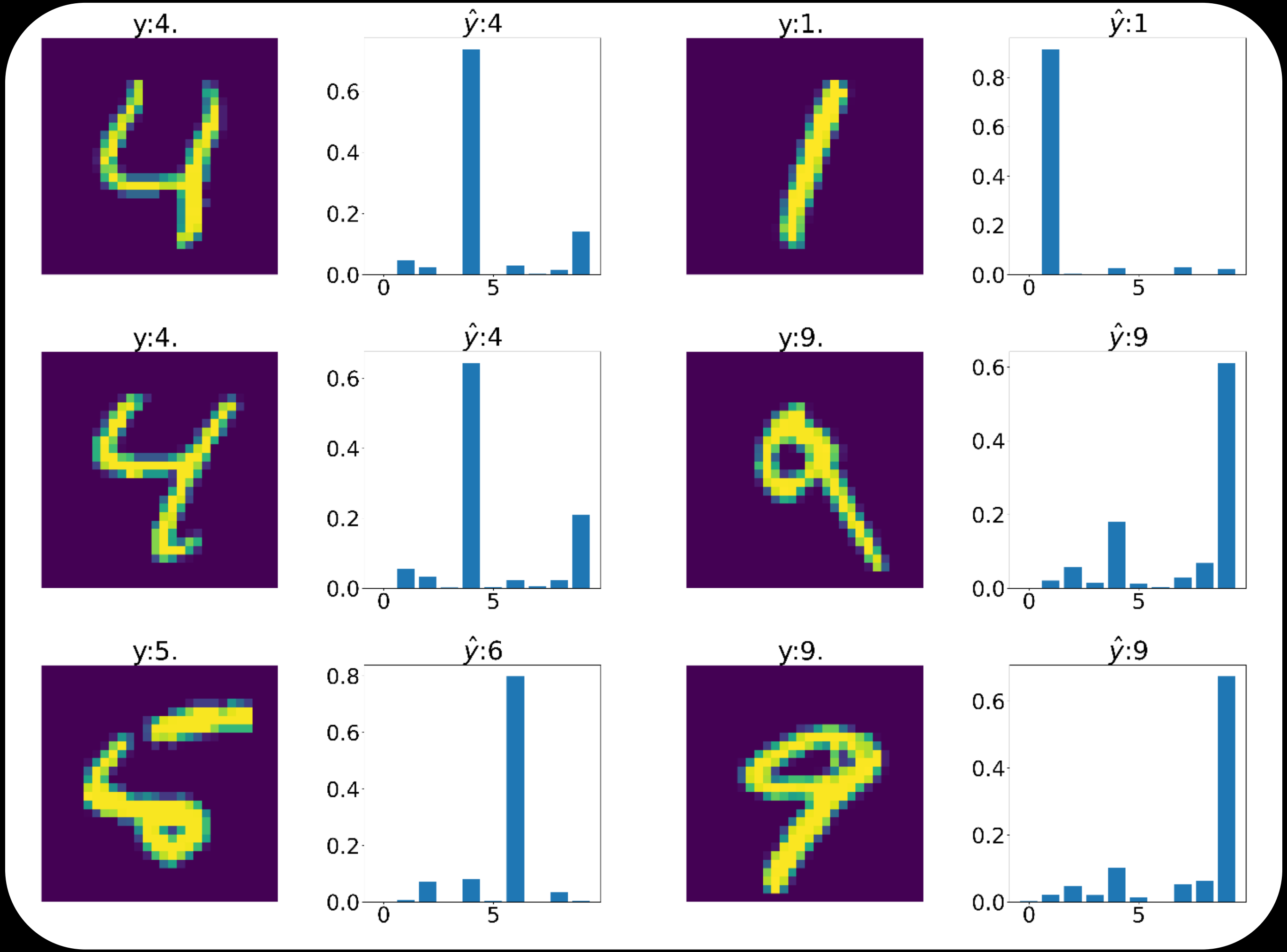
```
model = Sequential()
model.add(Dense(4, input_shape=(784,), activation="sigmoid"))
model.add(Dense(4, activation="sigmoid"))
model.add(Dense(10, activation="softmax"))
```

Train model

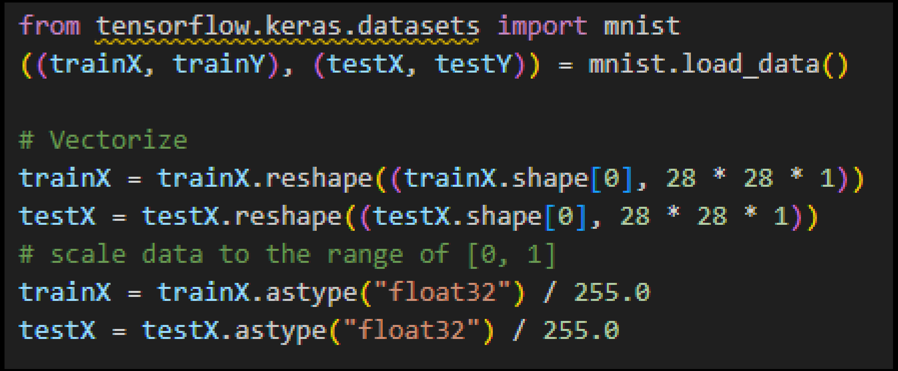


```
model.compile(
    loss="categorical_crossentropy",
    optimizer=SGD(0.01),
    metrics=["accuracy"])

model.fit(
    trainX, trainY, validation_data=(testX, testY),
    epochs=100, batch_size=128)
```

Neural Networks

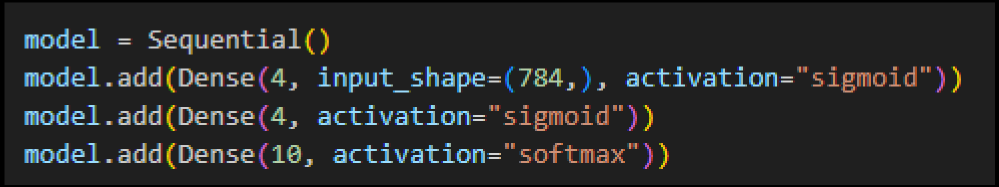


Load and prepare data



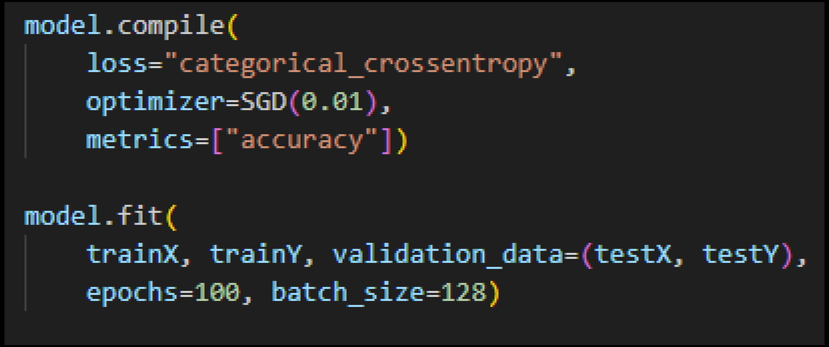
```
from tensorflow.keras.datasets import mnist
((trainX, trainY), (testX, testY)) = mnist.load_data()

# Vectorize
trainX = trainX.reshape((trainX.shape[0], 28 * 28 * 1))
testX = testX.reshape((testX.shape[0], 28 * 28 * 1))
# scale data to the range of [0, 1]
trainX = trainX.astype("float32") / 255.0
testX = testX.astype("float32") / 255.0
```

Define model



```
model = Sequential()
model.add(Dense(4, input_shape=(784,), activation="sigmoid"))
model.add(Dense(4, activation="sigmoid"))
model.add(Dense(10, activation="softmax"))
```

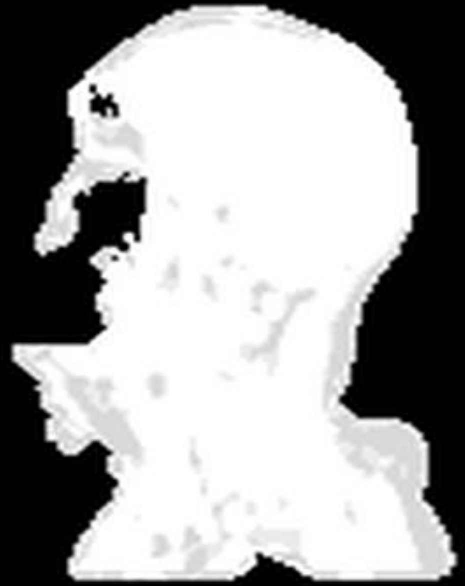
Train model



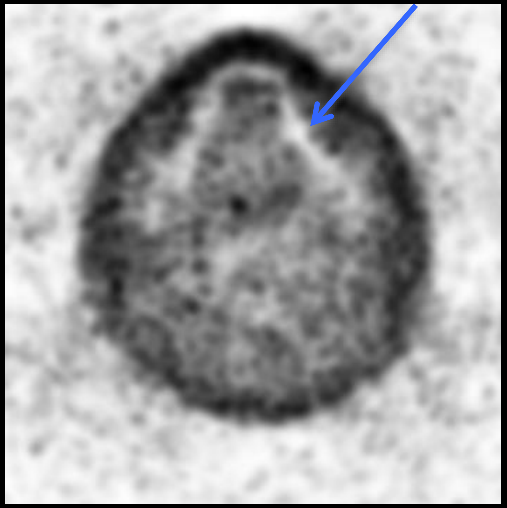
```
model.compile(
    loss="categorical_crossentropy",
    optimizer=SGD(0.01),
    metrics=["accuracy"])

model.fit(
    trainX, trainY, validation_data=(testX, testY),
    epochs=100, batch_size=128)
```

Active Shape Models – and more

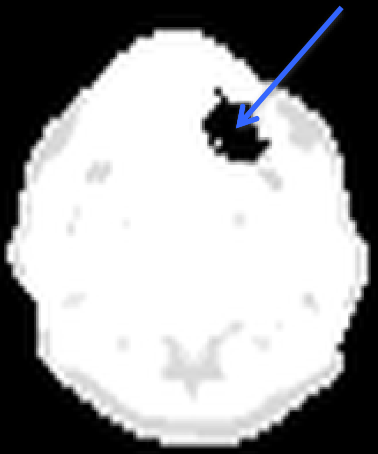
Motivation: Artifacts in umaps result in loss of quantitative accuracy



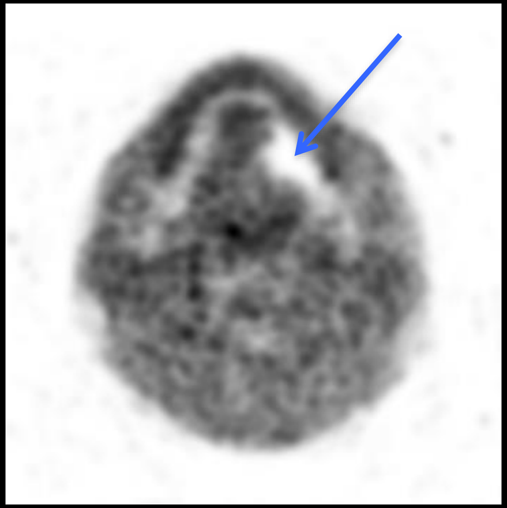
μ-map



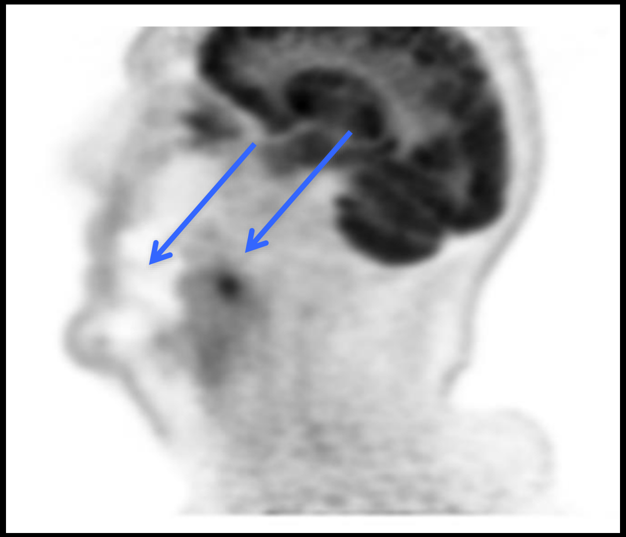
NAC-PET$_{MR}$



μ-map

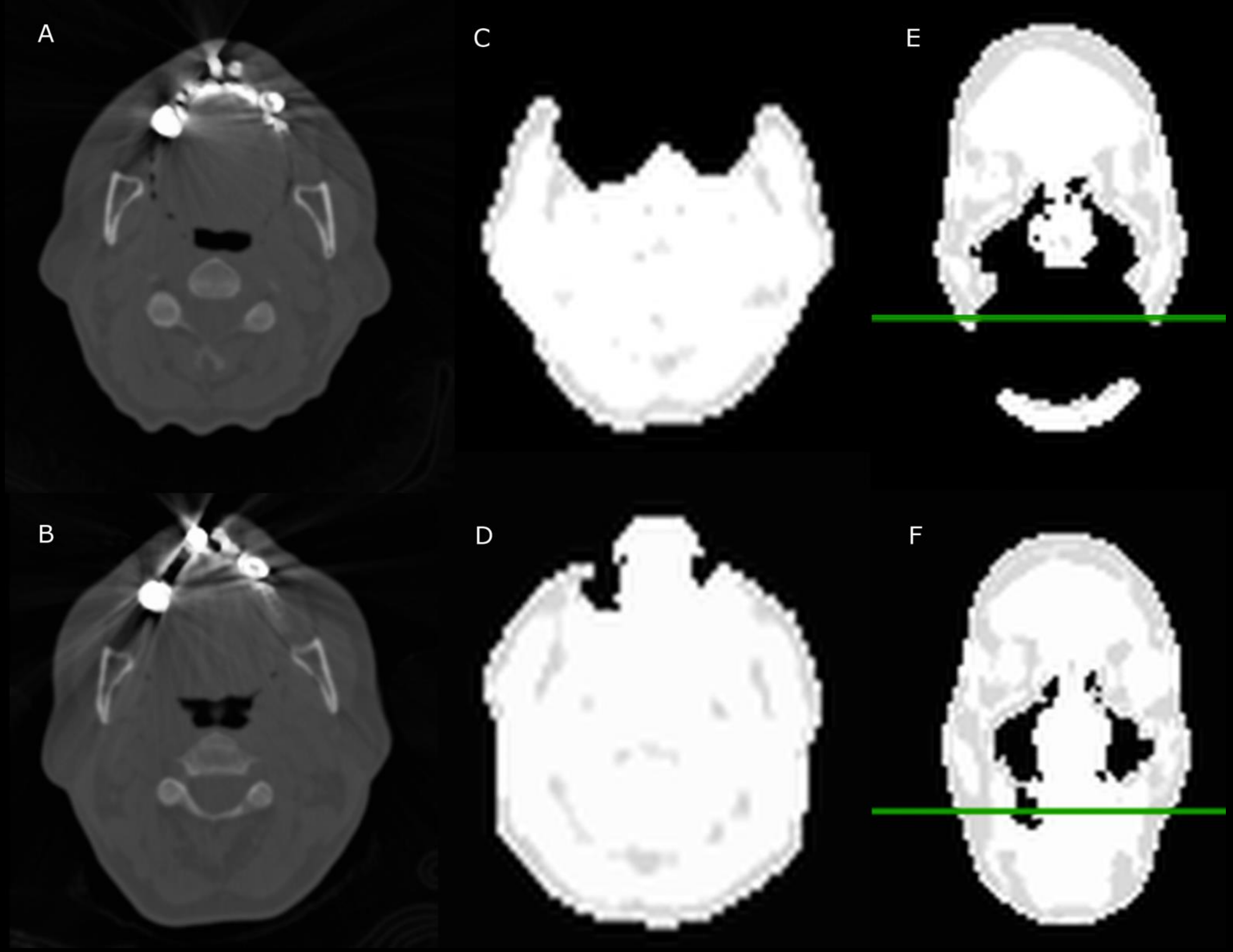


AC-PET$_{MR}$



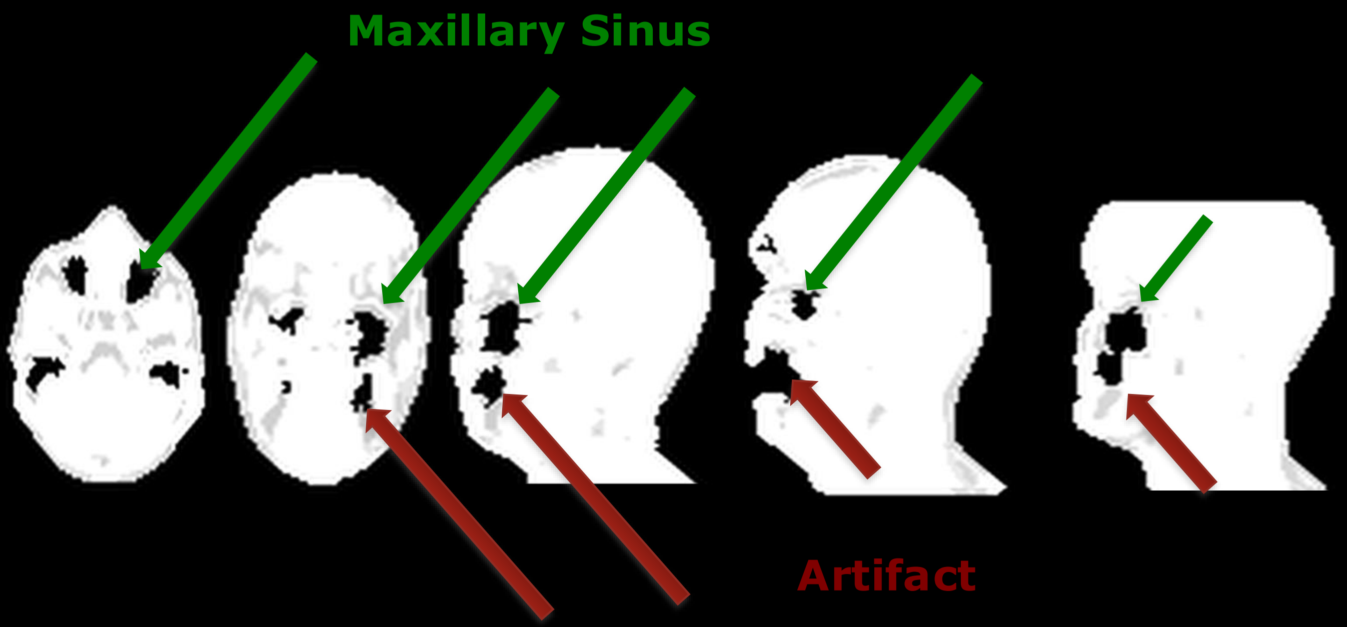
AC-PET$_{MR}$

Active Shape Models – and more



Artifacts can not be predicted just by knowing the amount of metal

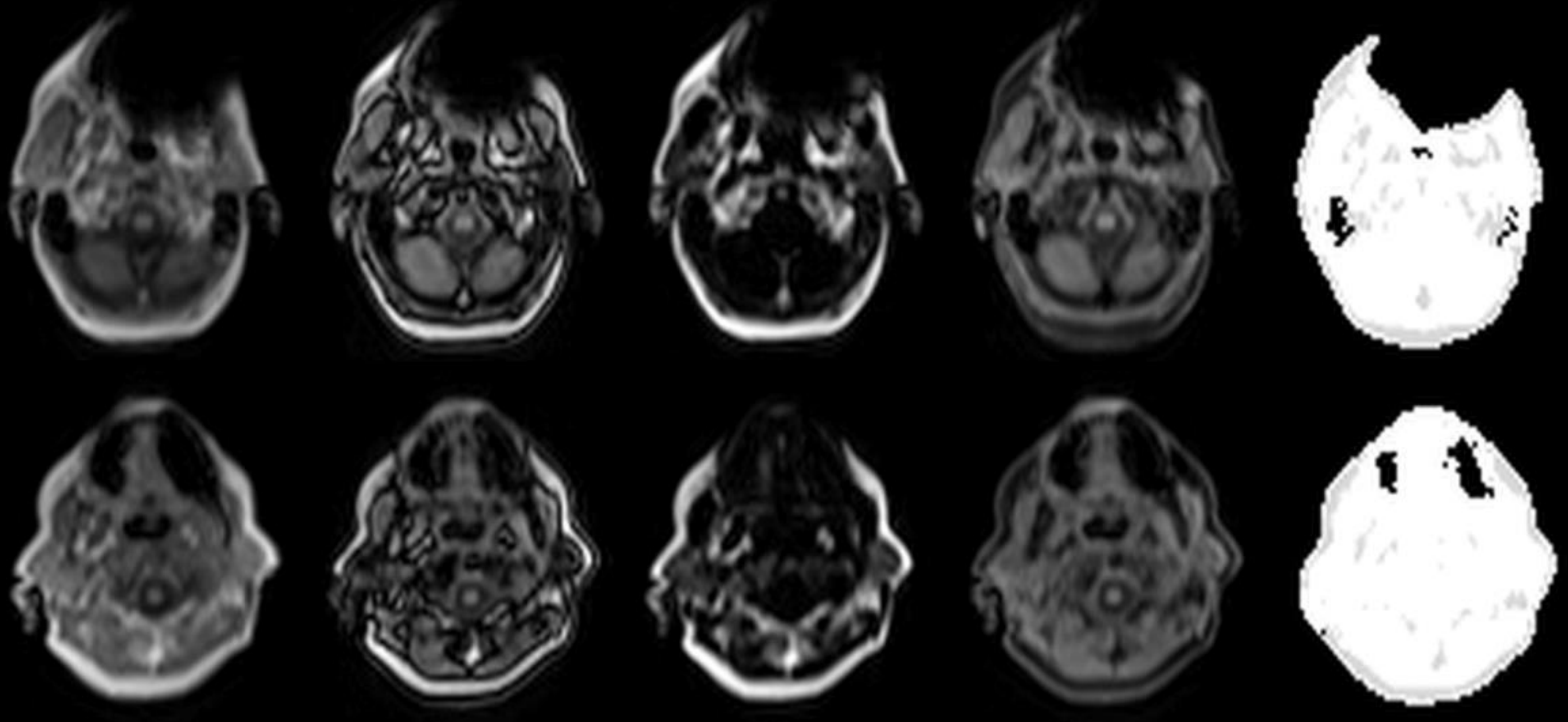
Active Shape Models – and more



Artifacts can be connected artificially with sinuses or background

Active Shape Models – and more

Outer holes = Signal voids breaching the anatomical surface



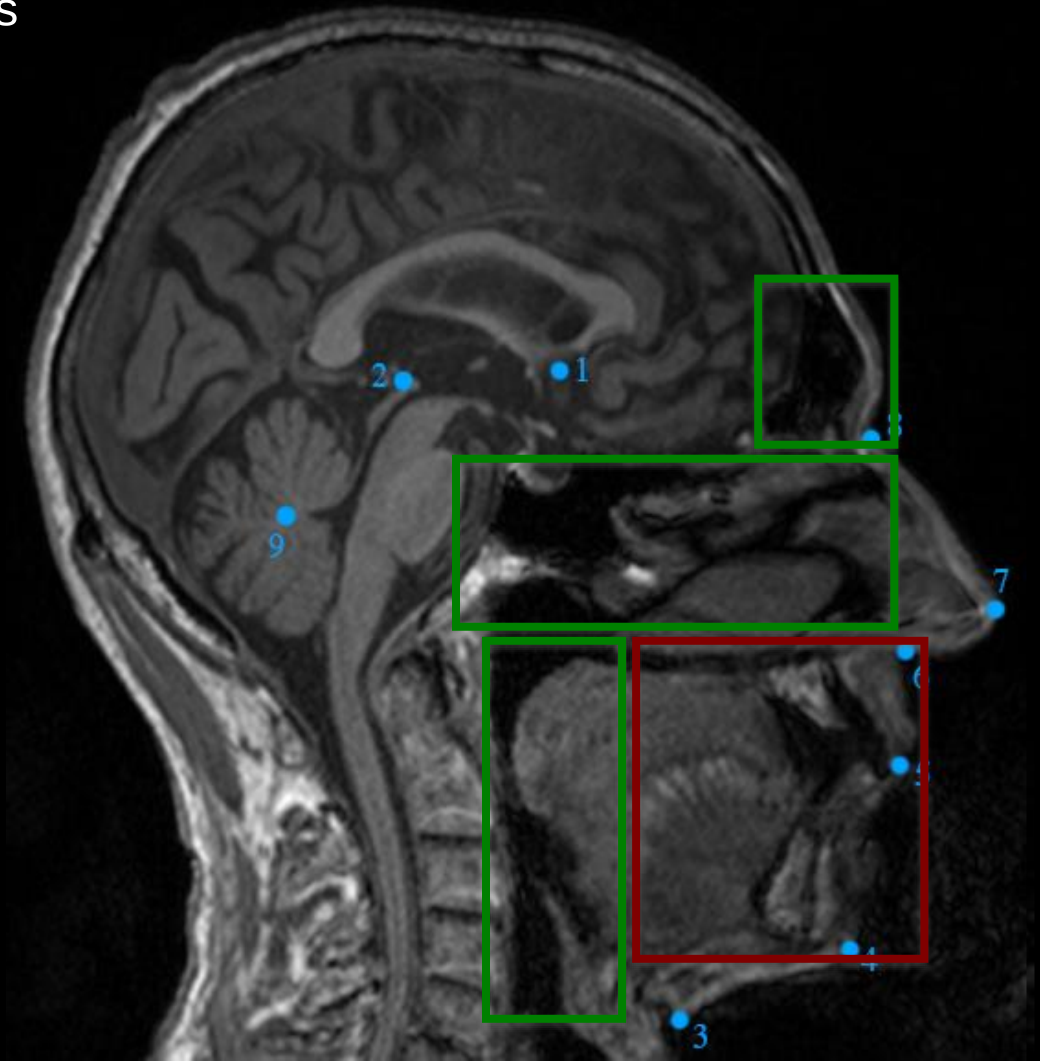
Inner holes = Signal voids within the anatomical surface

Active Shape Models – and more

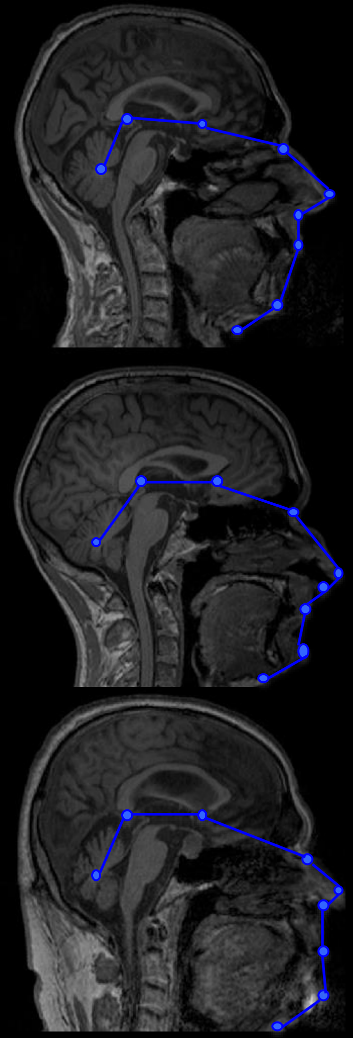
Artifacts can be separated from actual signal voids

How?

By the offset to a set of landmarks in 2D



Active Shape Models – and more



Procrustes analysis:

Transformation

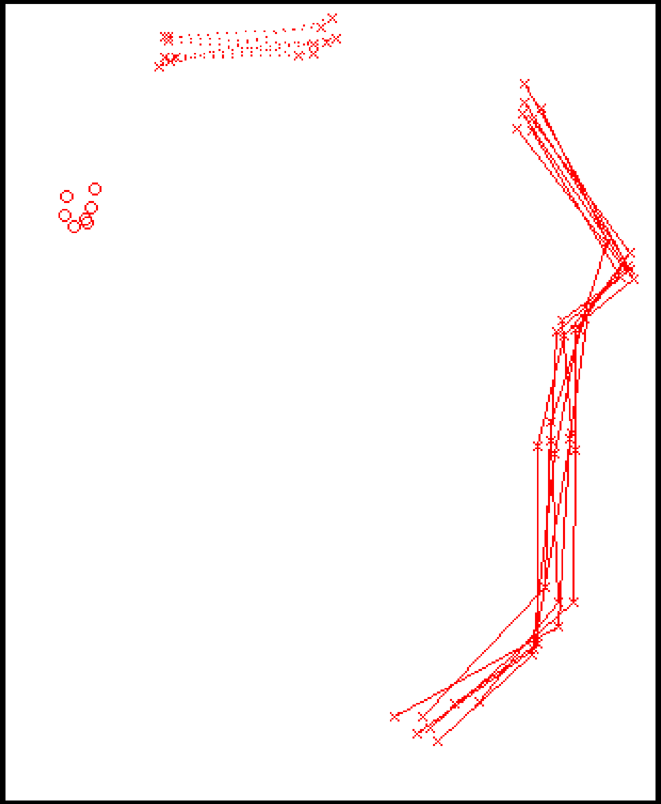
$$X_i \text{®} \; r_i X_i H_i + T_i$$

ρ : scaling
H : rotation

T : translation

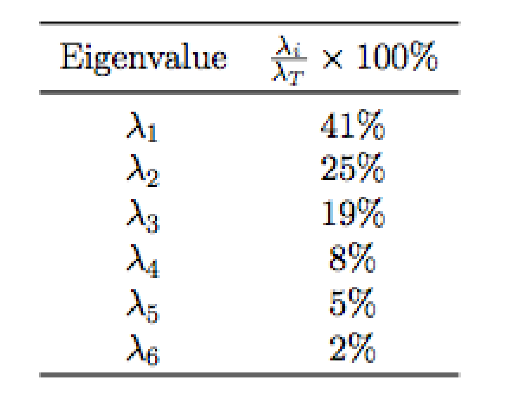
Minimization problem

$$\overset{s}{\underset{i<s}{\text{å}}} \left\| (r_i X_i H_i + T_i) - (r_s X_s H_s + T_s) \right\|_F^2$$

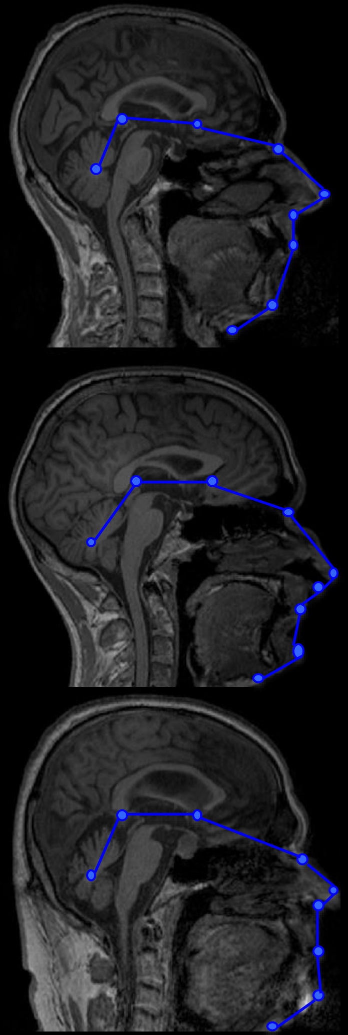


Landmarks on 7 patients

Active Shape Models – and more



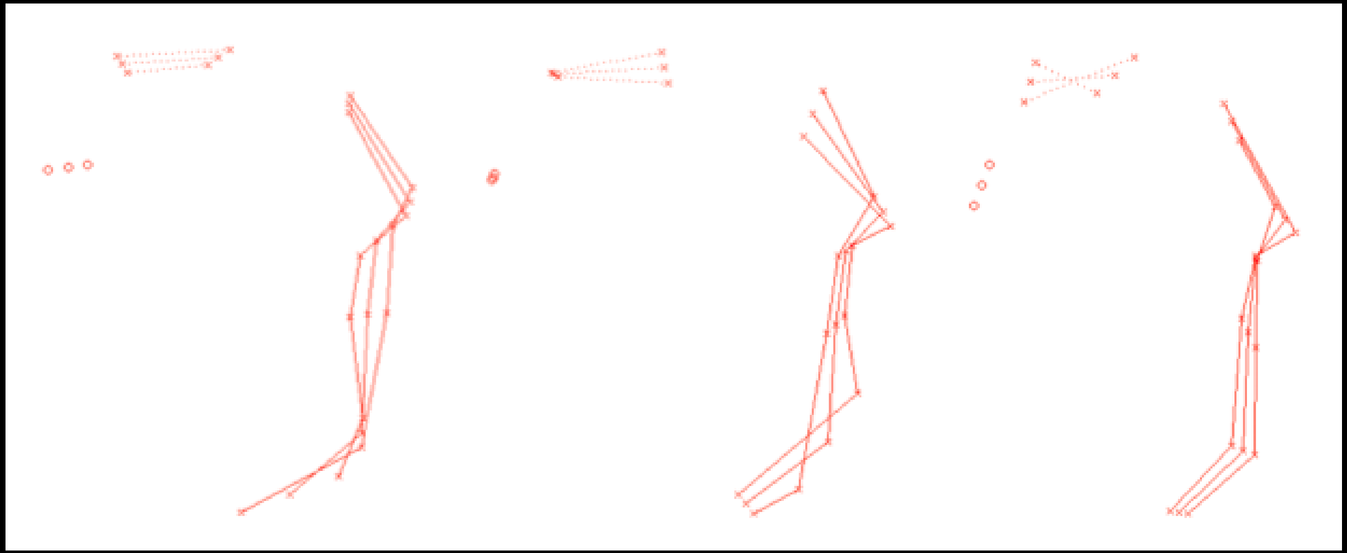
Eigenvalue	$\frac{\lambda_i}{\lambda_T} \times 100\%$
λ_1	41%
λ_2	25%
λ_3	19%
λ_4	8%
λ_5	5%
λ_6	2%



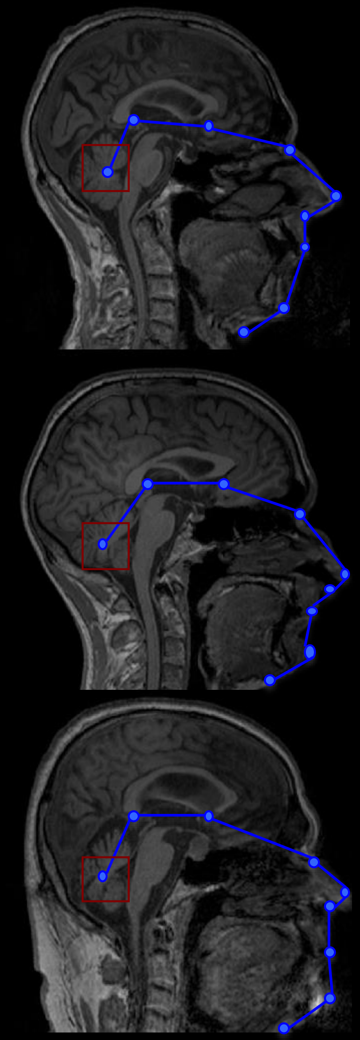
Mode 1: Mouth, horizontal & cerebellum

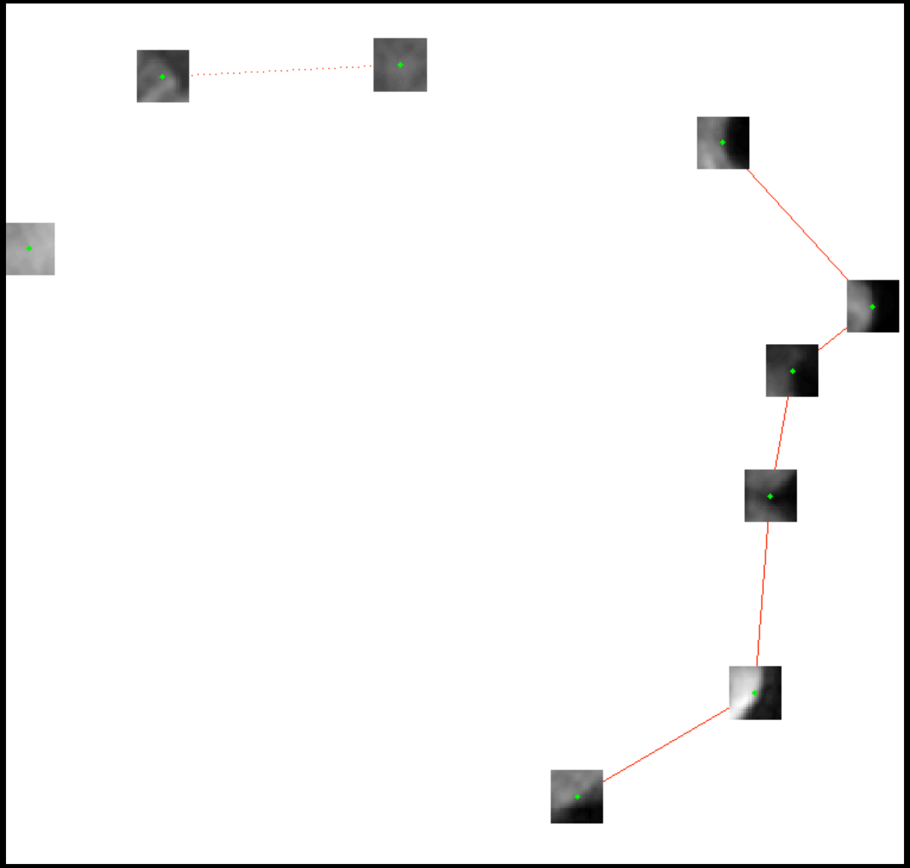
Mode 2: Chin

Mode 3: Aterior-posterior landmarks in respect to each other & cerebellum



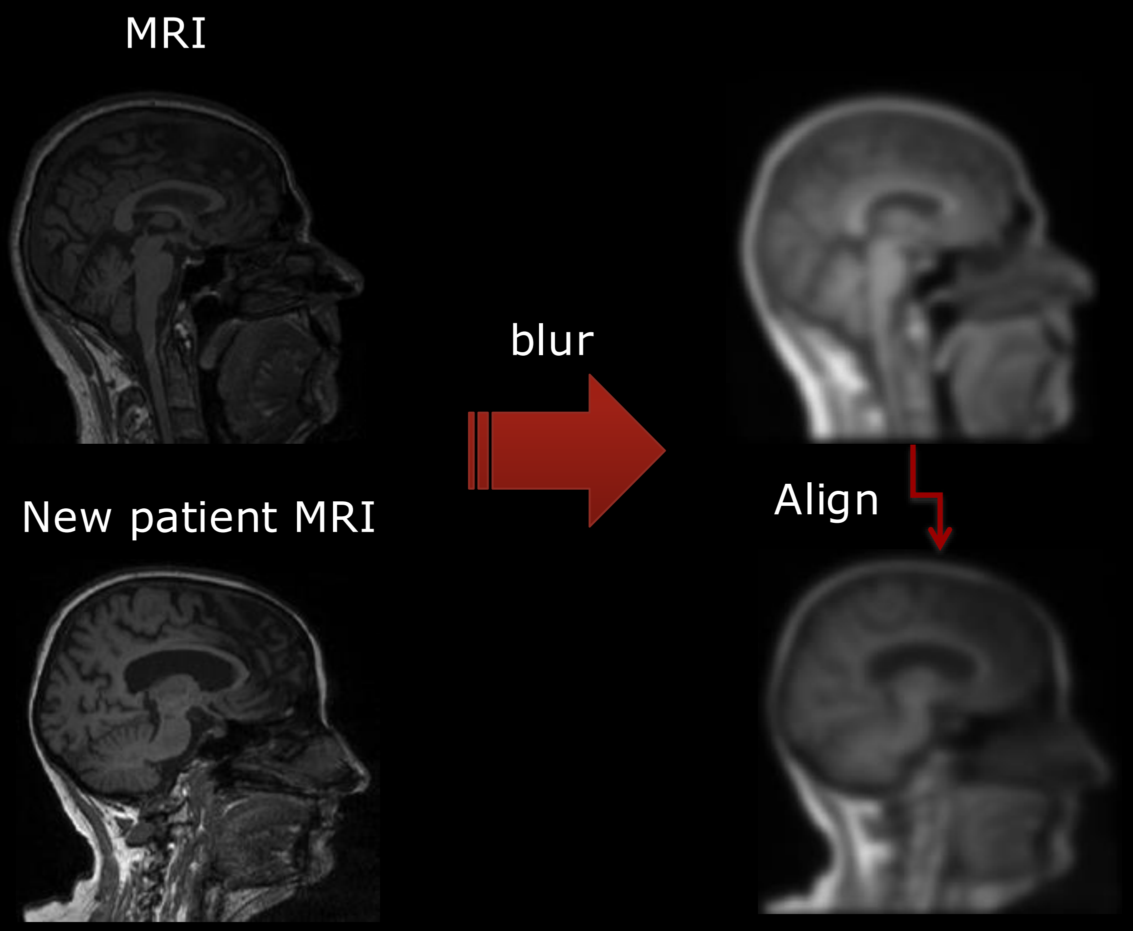
Active Shape Models – and more





Mean patches from 5 patients

Active Shape Models – and more



1. Blur MRI images from “atlas” patient and new patient
2. Align the two using rigid transformation
3. Apply the same transformation to the shape of the atlas patient

Cootes & Taylor, Comp. Vis. and Img. Under. 1995

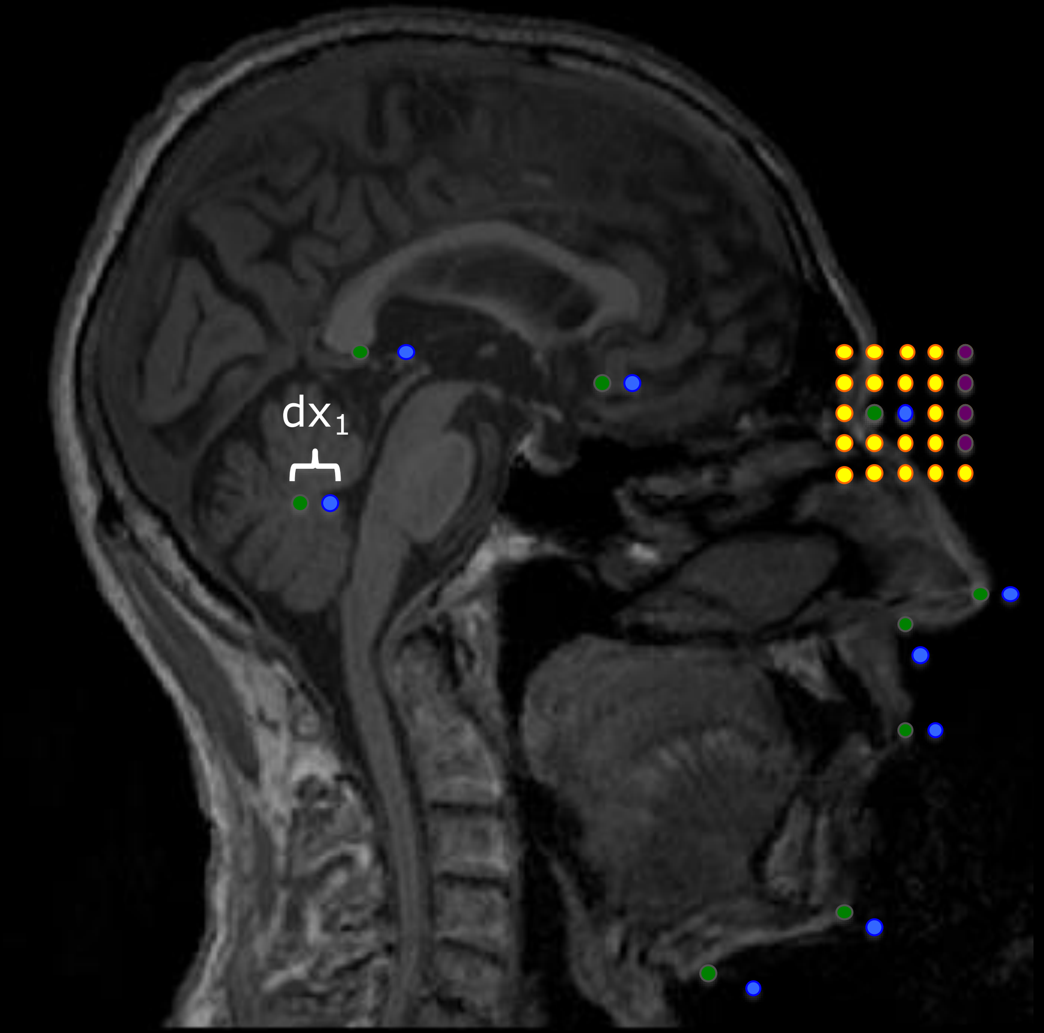
Active Shape Models – and more

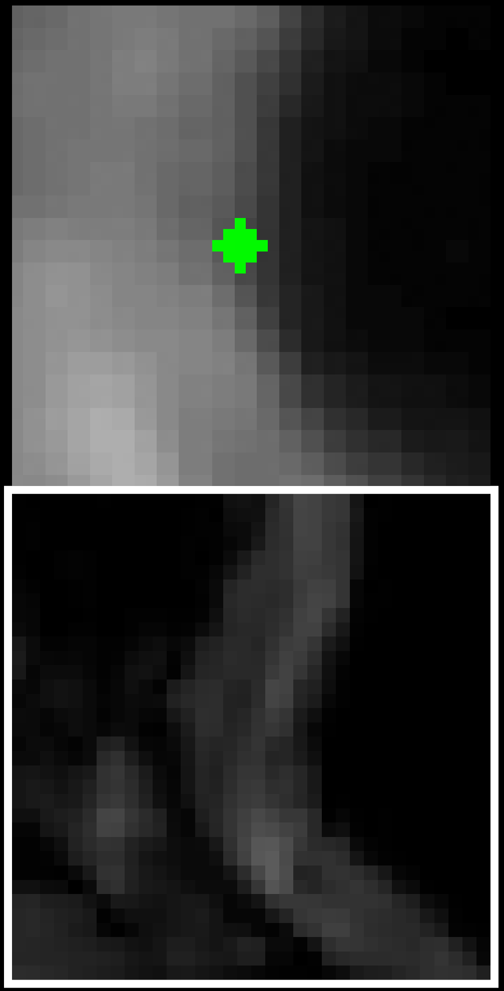
Offset to
mean shape:

$d\mathbf{x}=(dx_1,\dots,dx_n)$

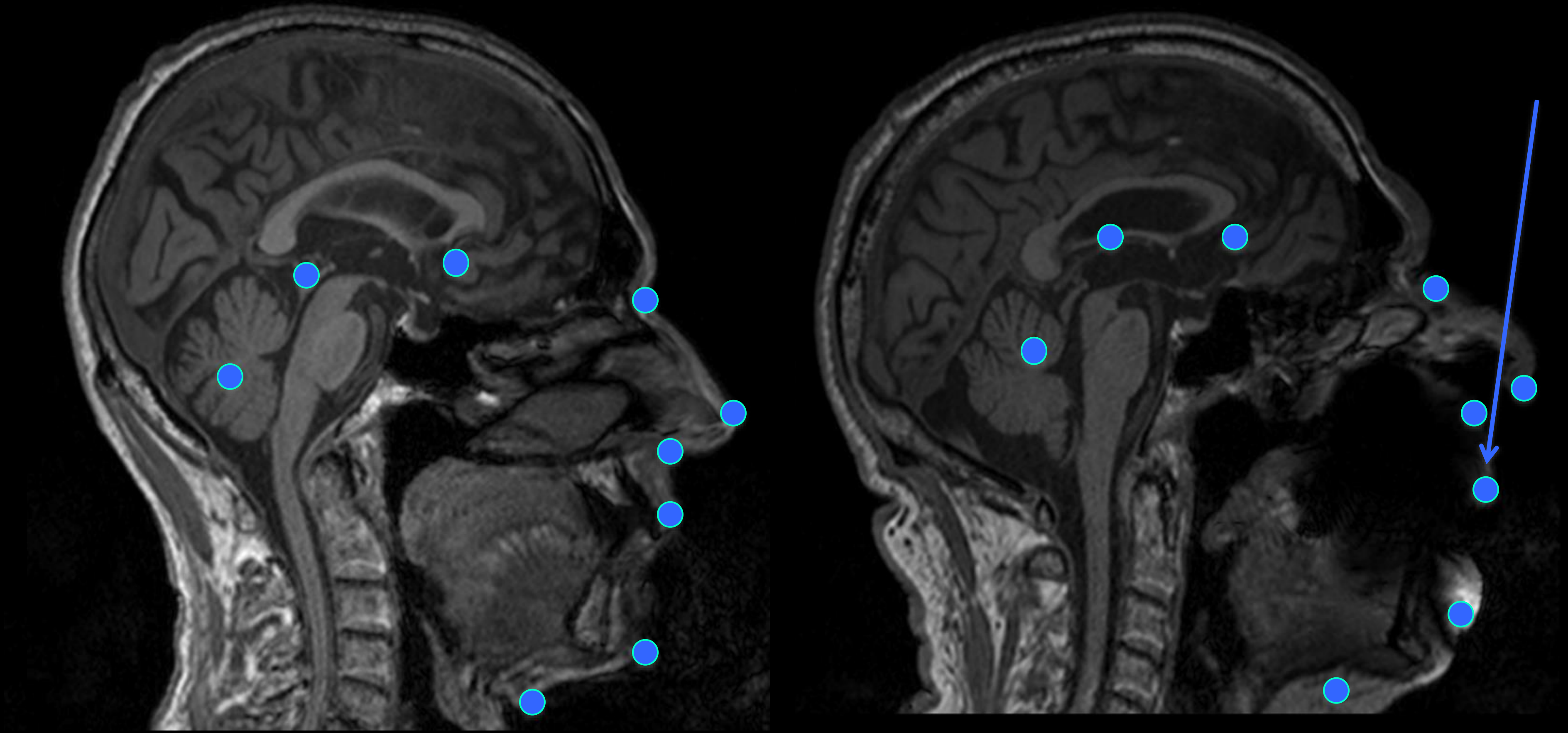
Projected to legal
shape space:

$d\mathbf{y}=\phi^T d\mathbf{x}$

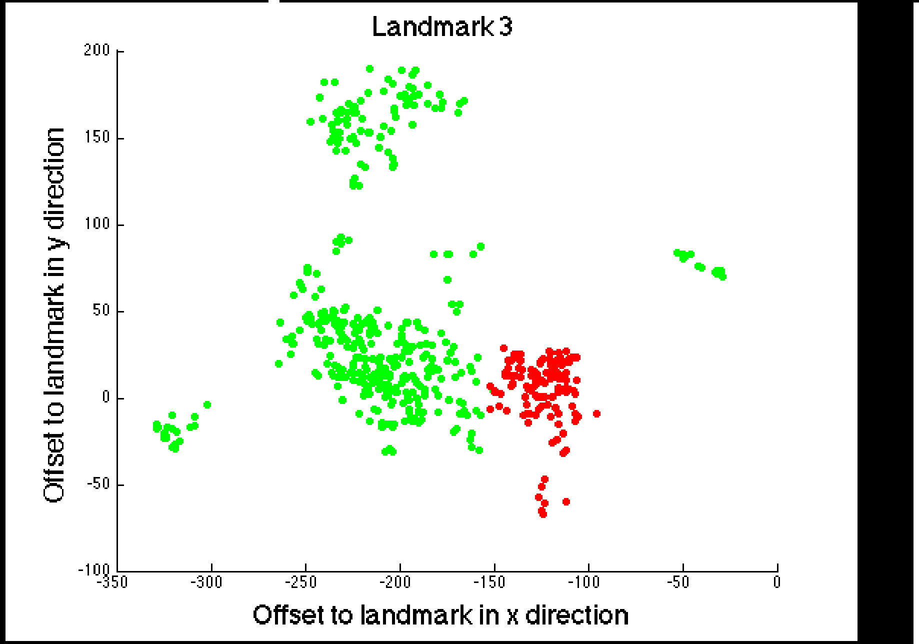


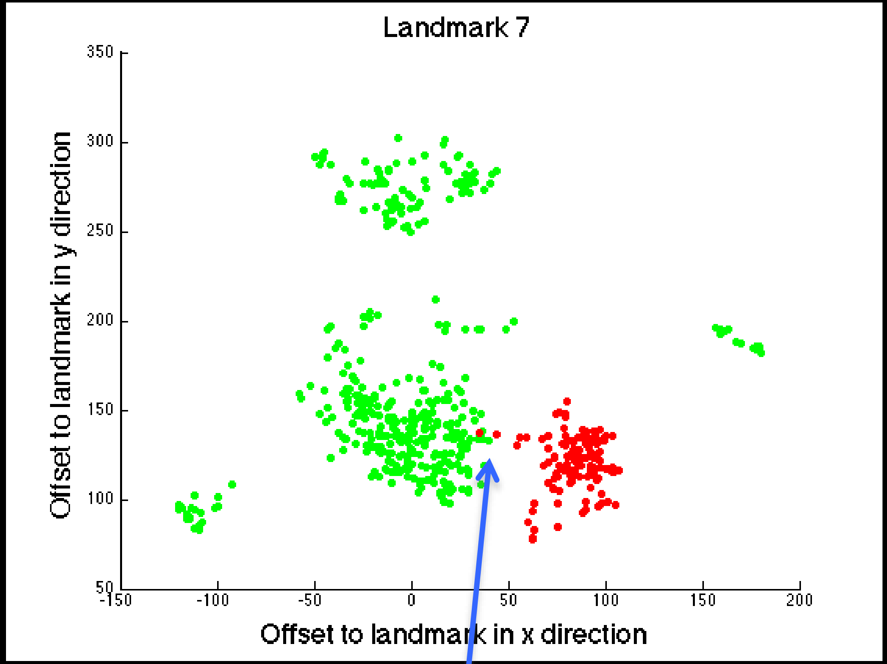


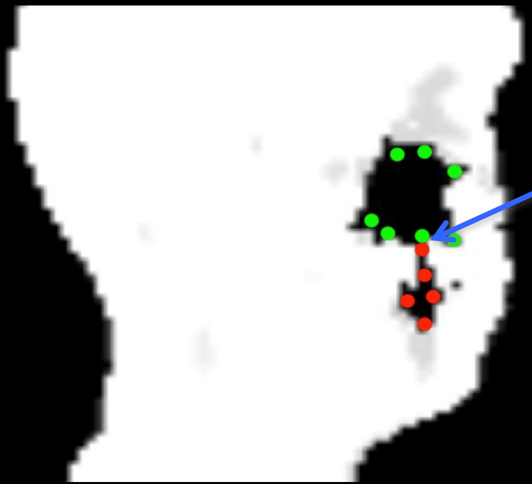
Active Shape Models – and more



Active Shape Models – and more





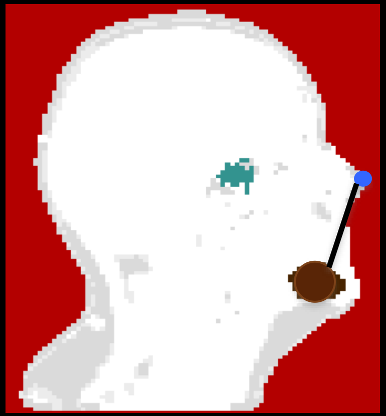


5 patients

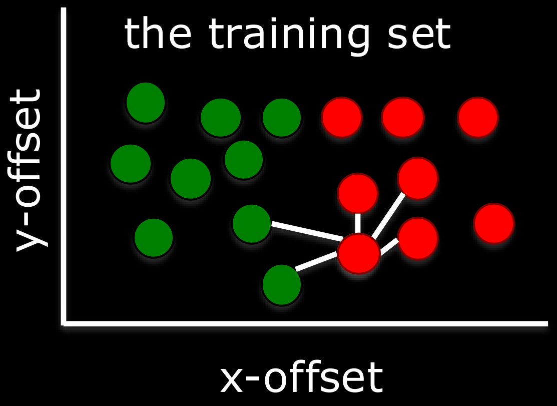
650 non-artifact pixels

210 artifact pixels

Active Shape Models – and more



Offsets to a landmark in the training set

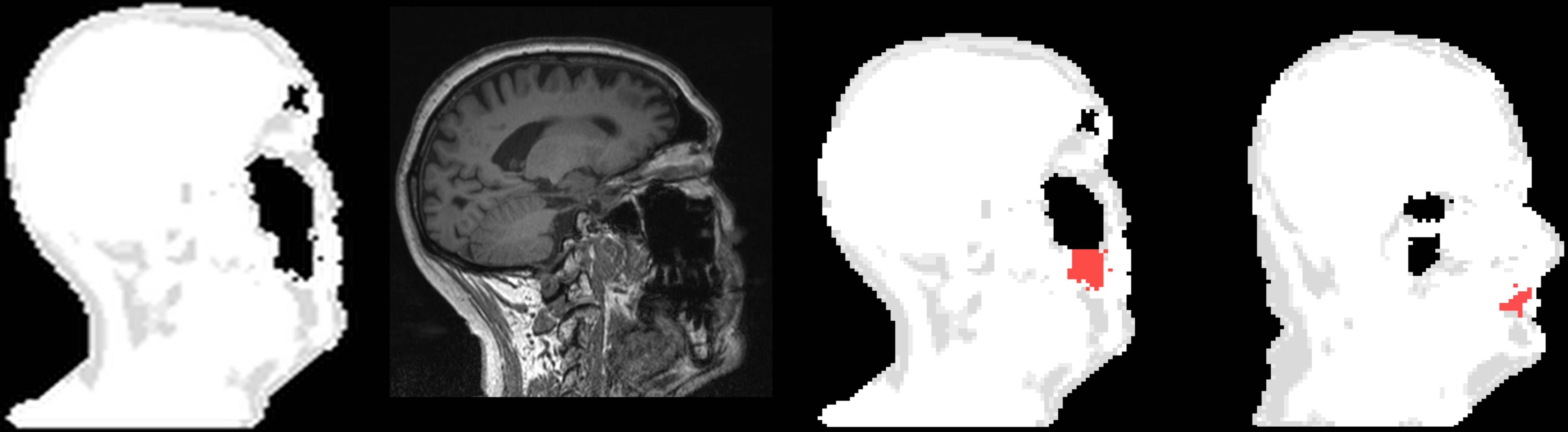


- ❑ Classify using kNN
  - For each pixel in a signal void
    - Find the offset to each landmark
      - Find 5-Nearest-Neighbors
      - Majority of neighbor-labels decides the landmark
    - Majority of landmark-labels decides the class

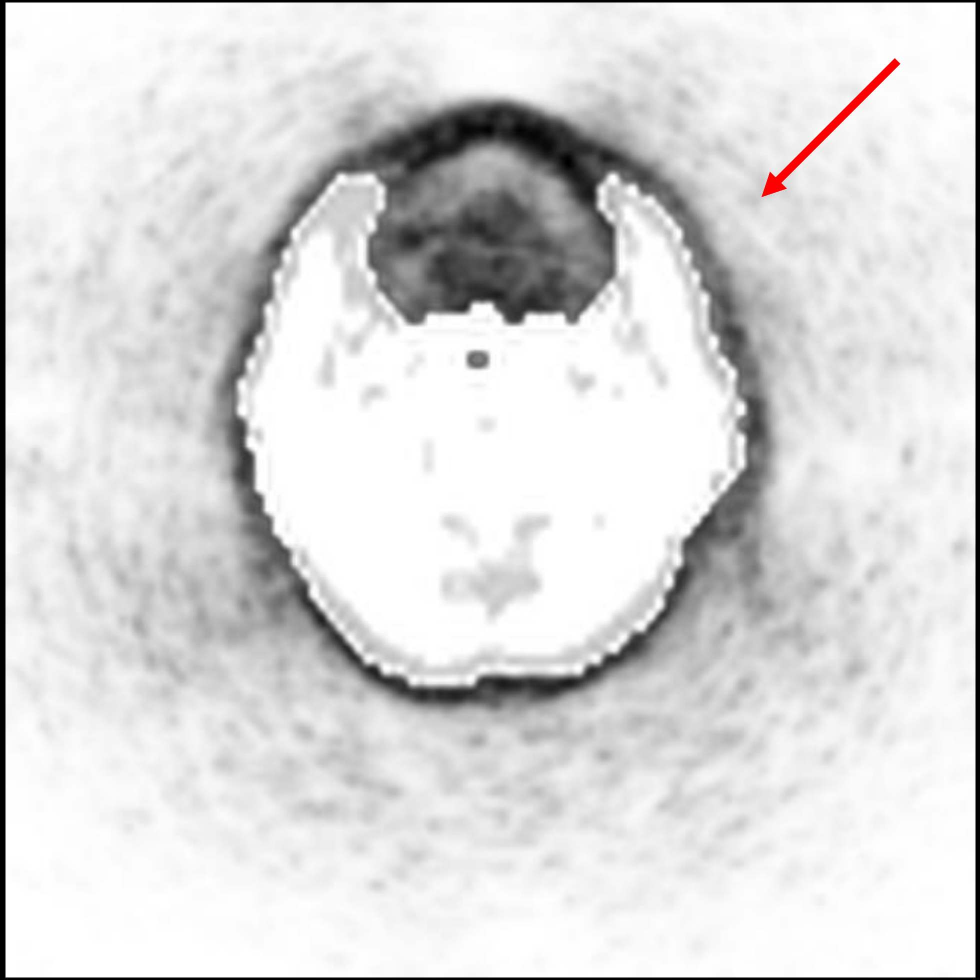
Classification: ●

Classify each pixel using kNN and a training set

Active Shape Models – and more



Active Contours: Chan Vese

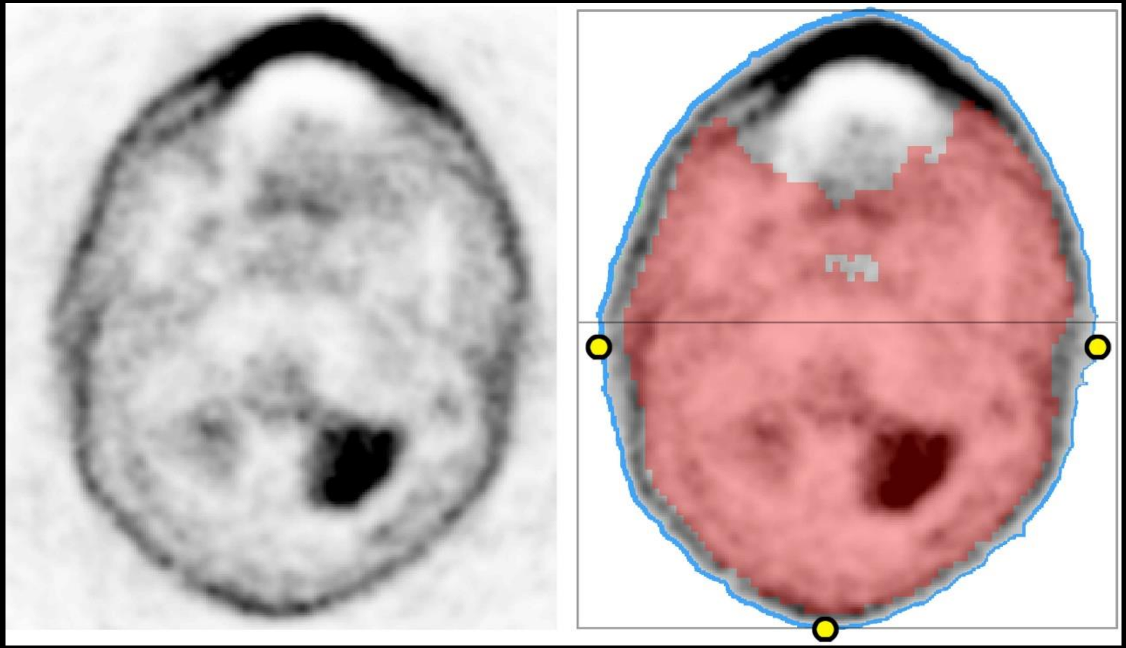


- ”Outer holes” cannot be corrected easily by MRI
- NAC-PET holds information about outer contour
- … but contains noise and needs to be delineated

Active Contours: Chan Vese

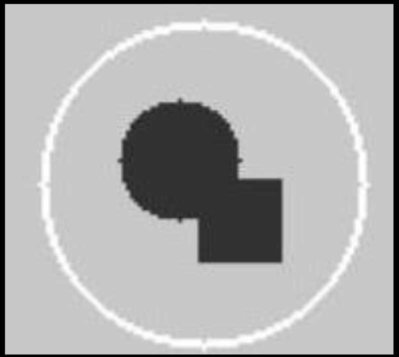
$$E_{CV}(\phi, c_i, c_o) = \mu \cdot Length(\phi) + \overbrace{\lambda_i \int_\Omega H(\phi)|u_0 - c_i|^2}^{\text{Inside contour}} + \overbrace{\lambda_o \int_\Omega H(-\phi)|u_0 - c_o|^2}^{\text{Outside contour}}$$

Length of contour   NAC-PET   Mean of areas



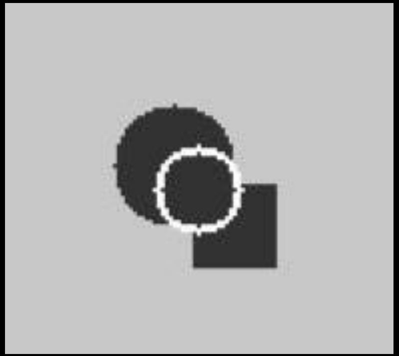
F(ϕ) > 0

F(-ϕ) ≈ 0



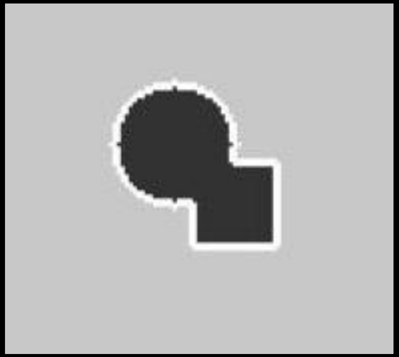
F(ϕ) ≈ 0

F(-ϕ) > 0

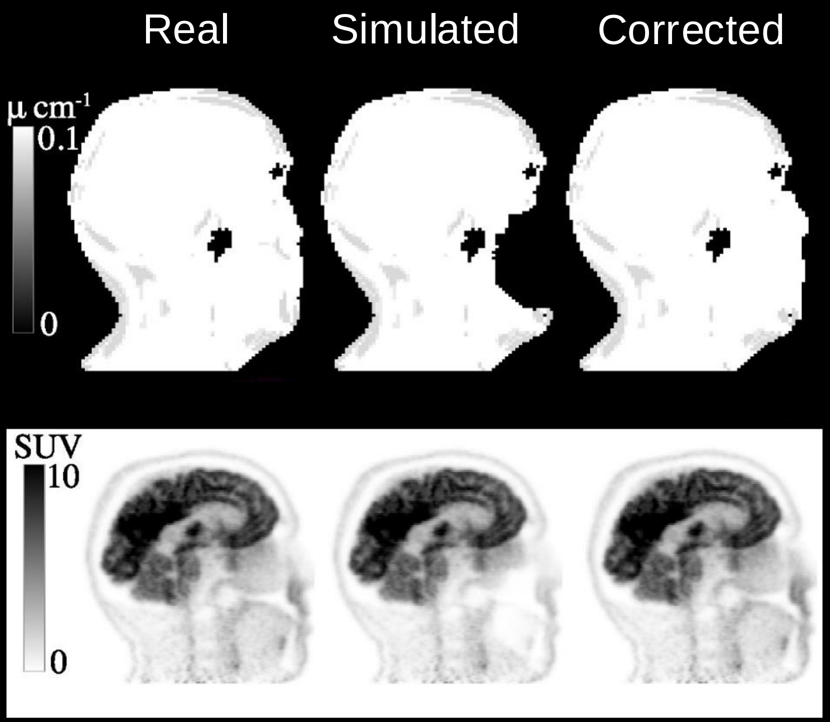


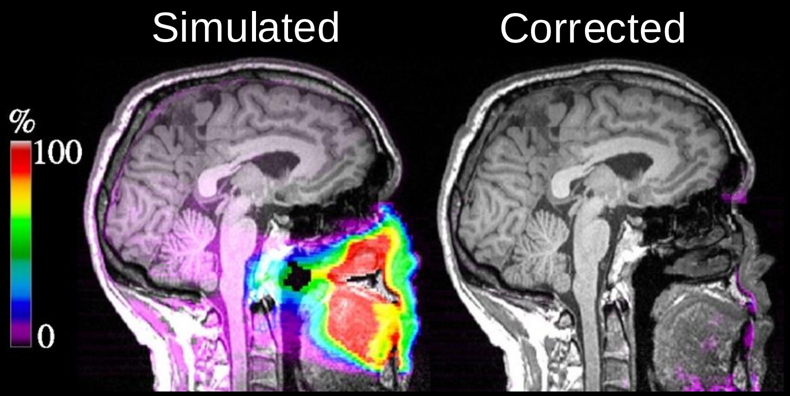
F(ϕ) ≈ 0

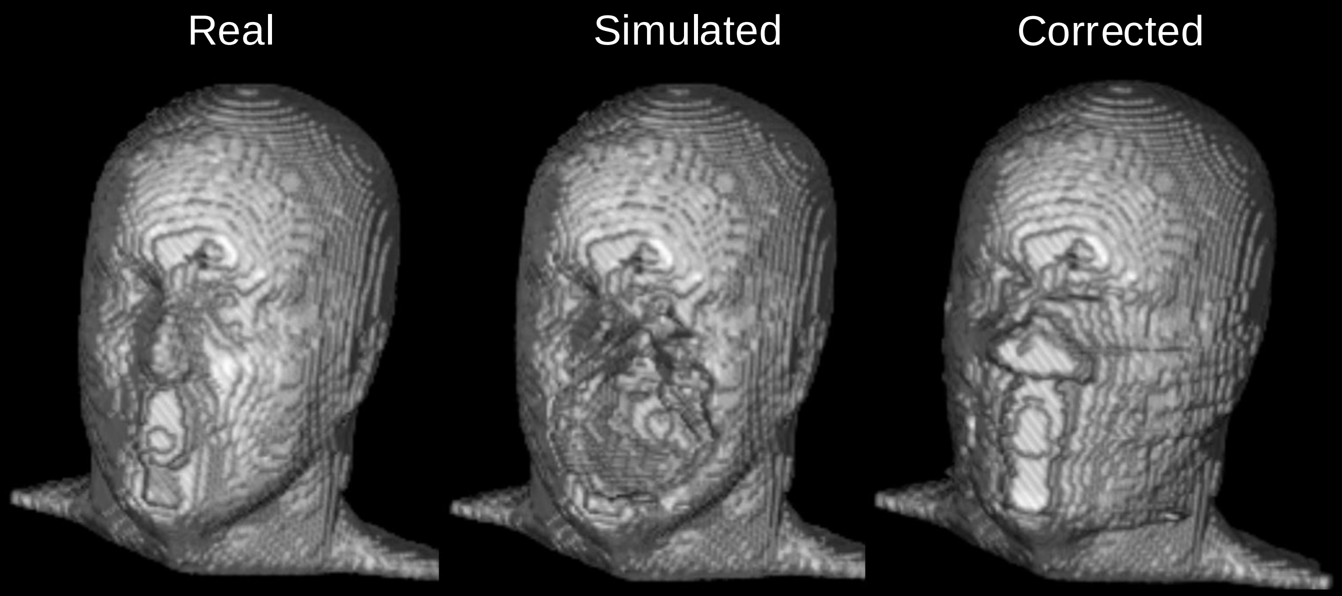
F(-ϕ) ≈ 0



Active Contours: Chan Vese

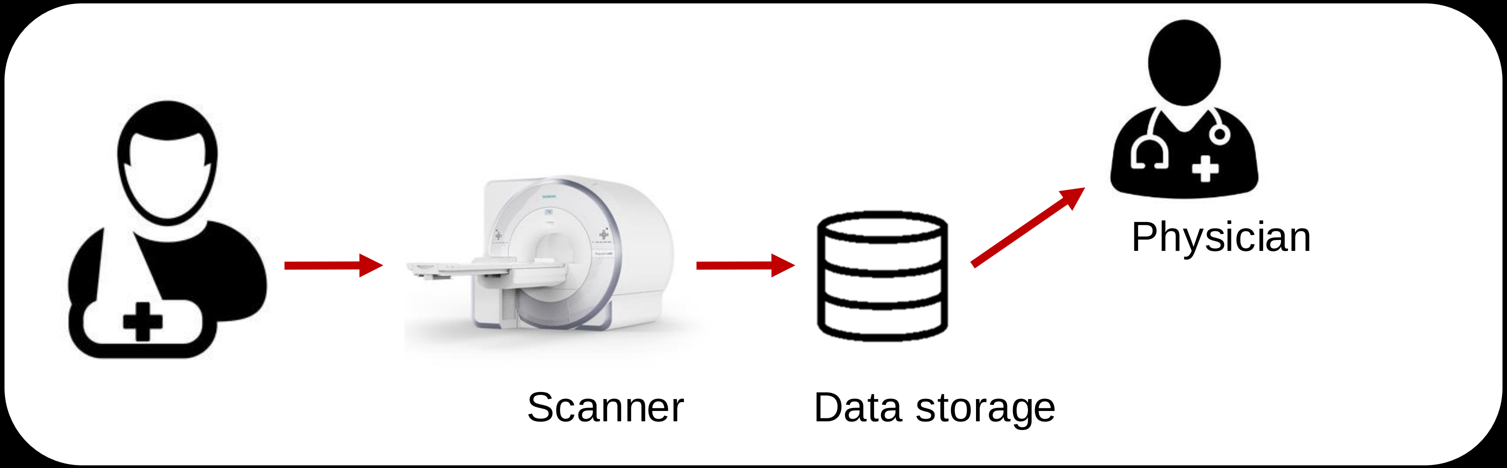






What did you learn today

- Many of the topics taught during this course can be useful for image analysis at an imaging department in a hospital
- Topics like preprocessing are always used before any imaging project
- Registration are used to align scans within a patient examination, and across examinations
- Simple tools are often wanted as it
  - Works well with limited data
  - Strengthens the explainability of a method


Physician
Scanner
Data storage