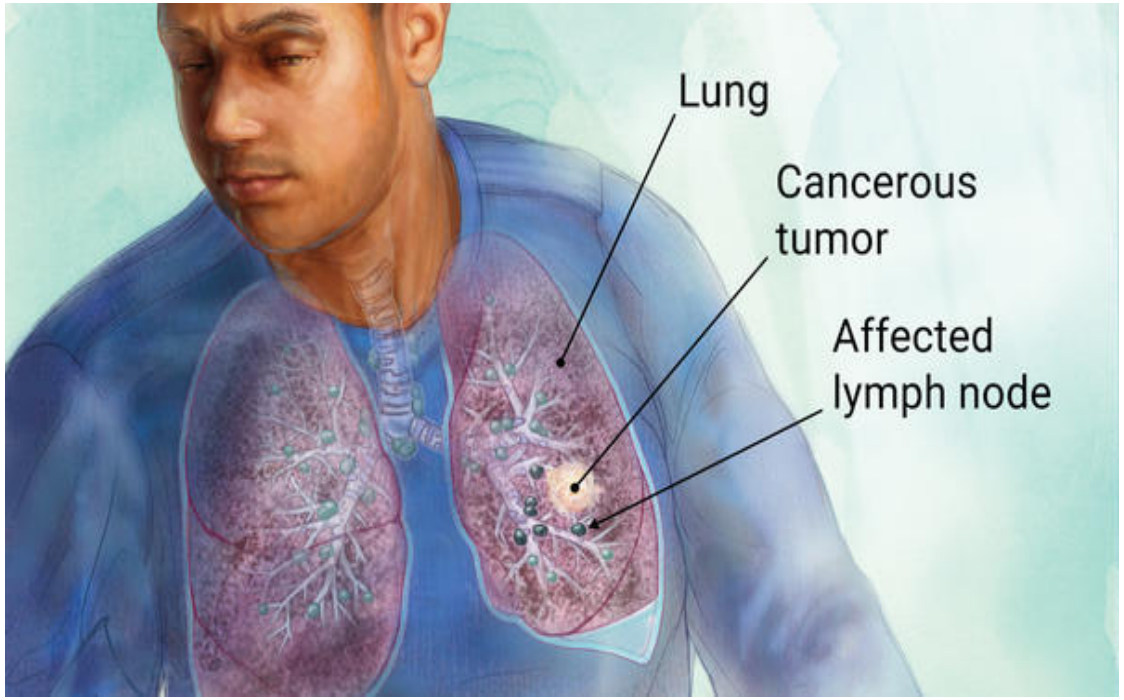
LUNG CANCER DETECTION

Advanced Data Mining Classification and Prediction:

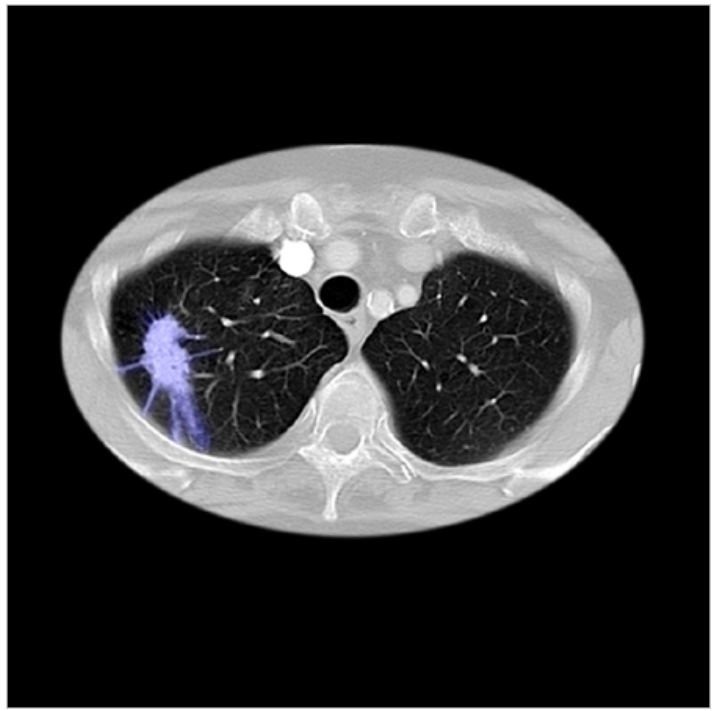
convolutional networks for Biomedical image segmentation

ANYU ZHANG

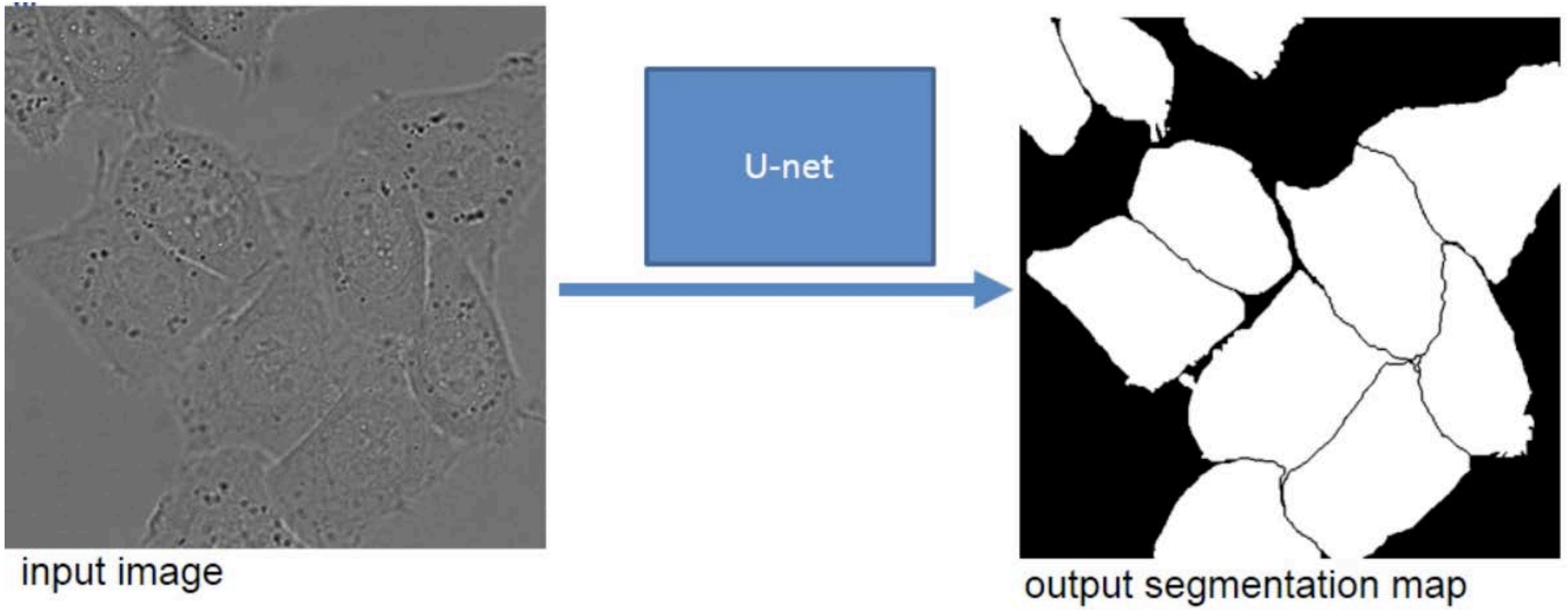
LUNG CANCER



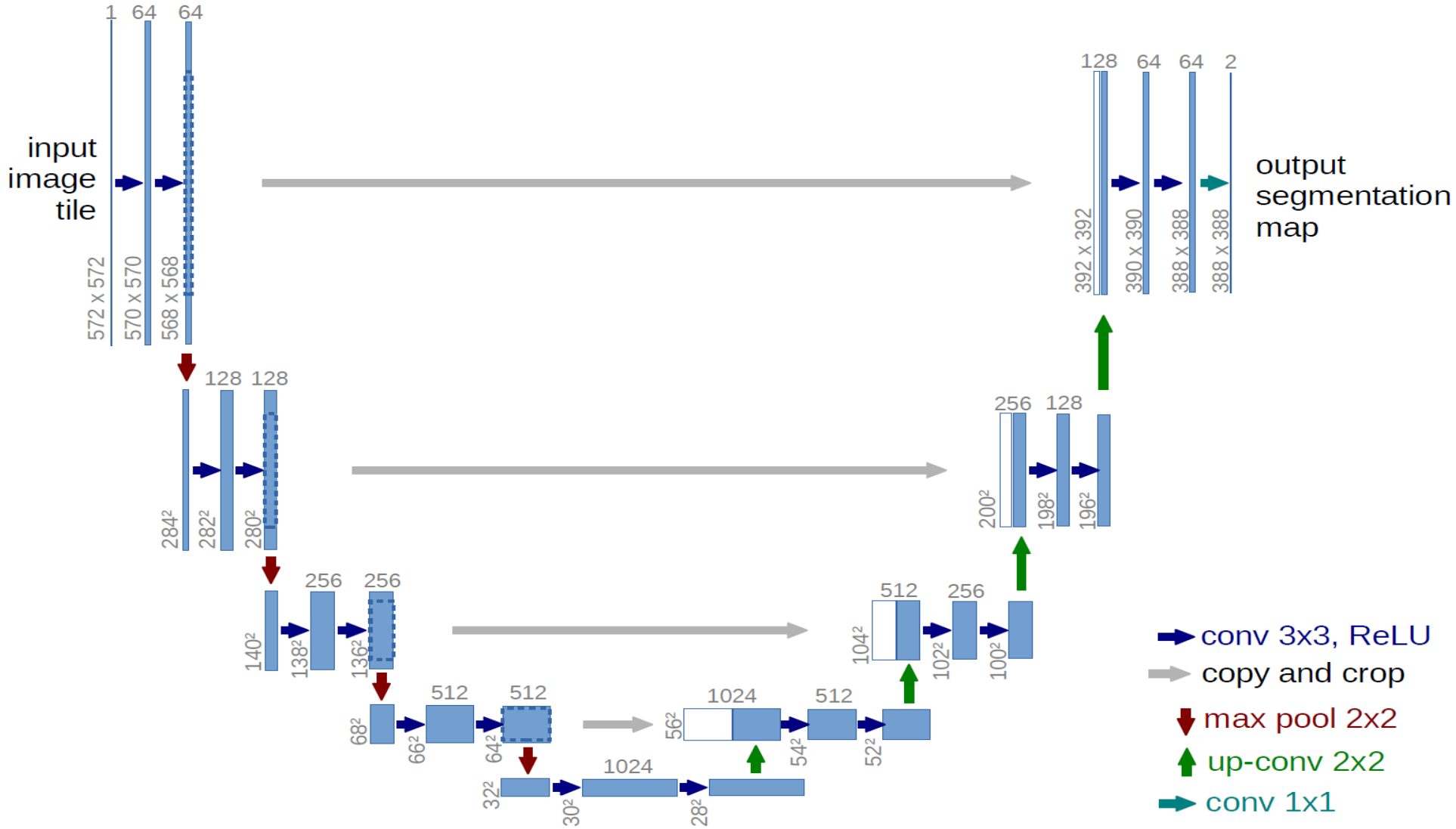
LUNG CANCER



CT scan segmentation

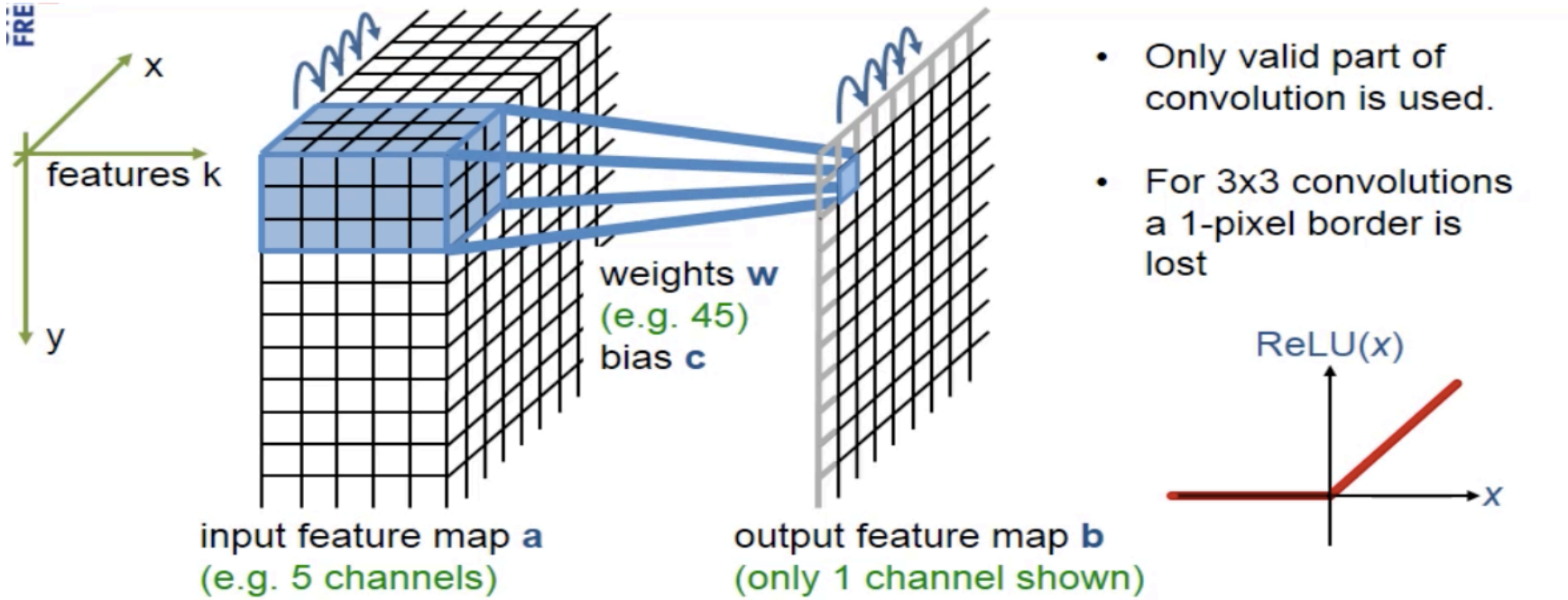


Why U-net but not normal CNN ?



U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

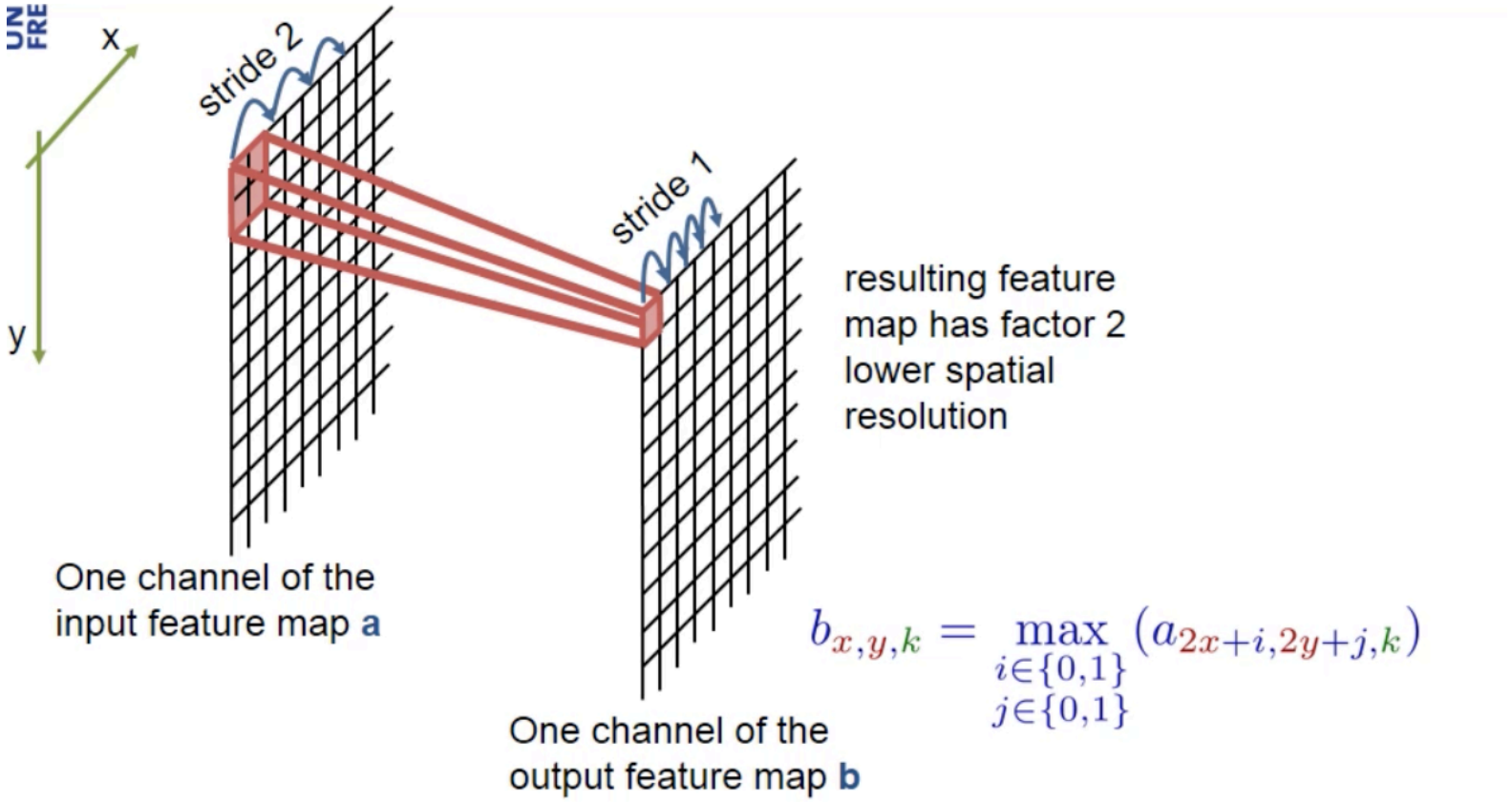
3X3 CONVOLUTION+RELU



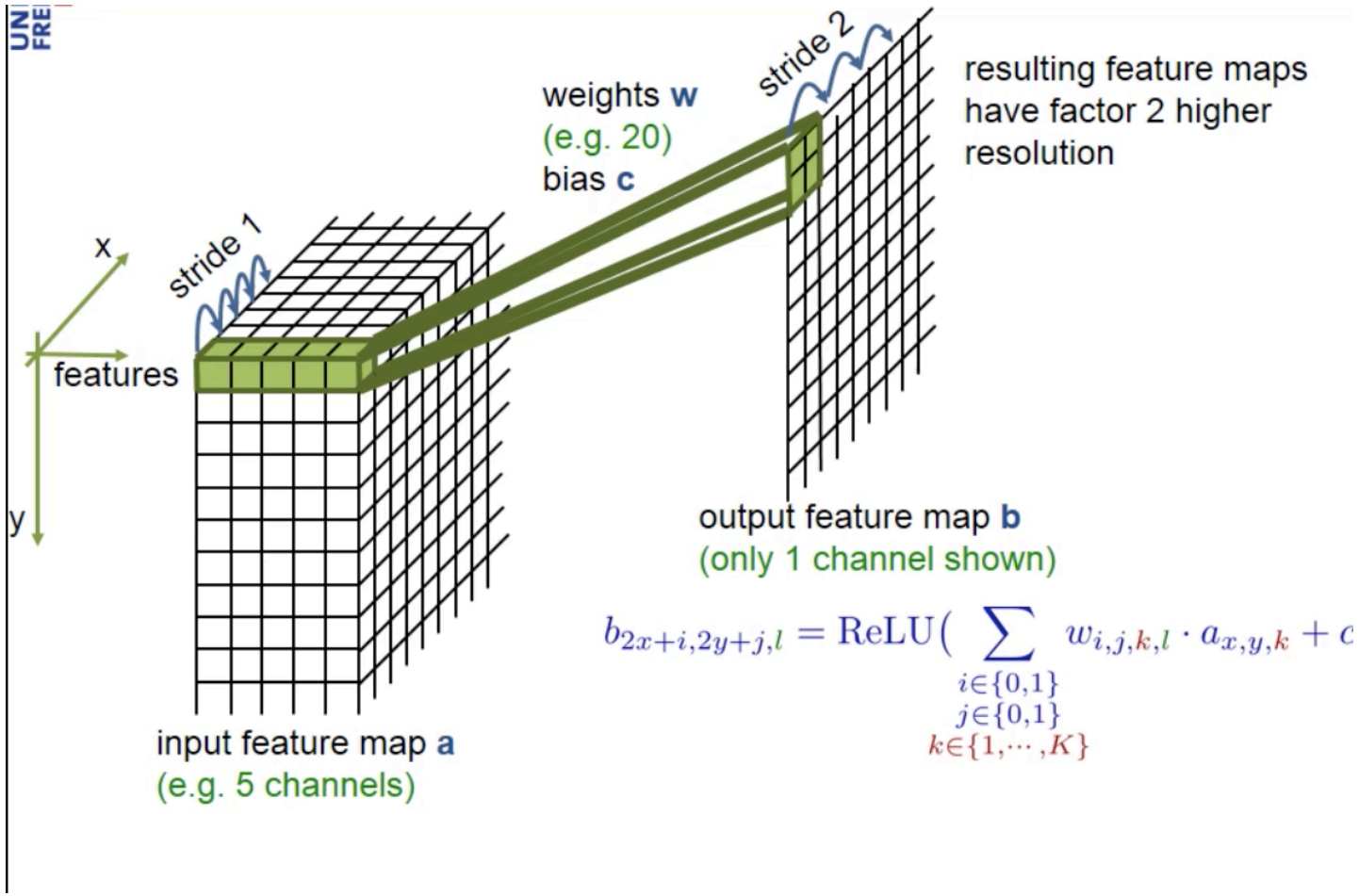
- Only valid part of convolution is used.
- For 3x3 convolutions a 1-pixel border is lost

$$b_{x,y,l} = \mathrm{ReLU}\Big(\sum_{\substack{i\in\{-1,0,1\}\\ j\in\{-1,0,1\}\\ k\in\{1,\dots,K\}}} w_{i,j,k,l} \cdot a_{x+i,y+j,k} + c_l \Big)$$

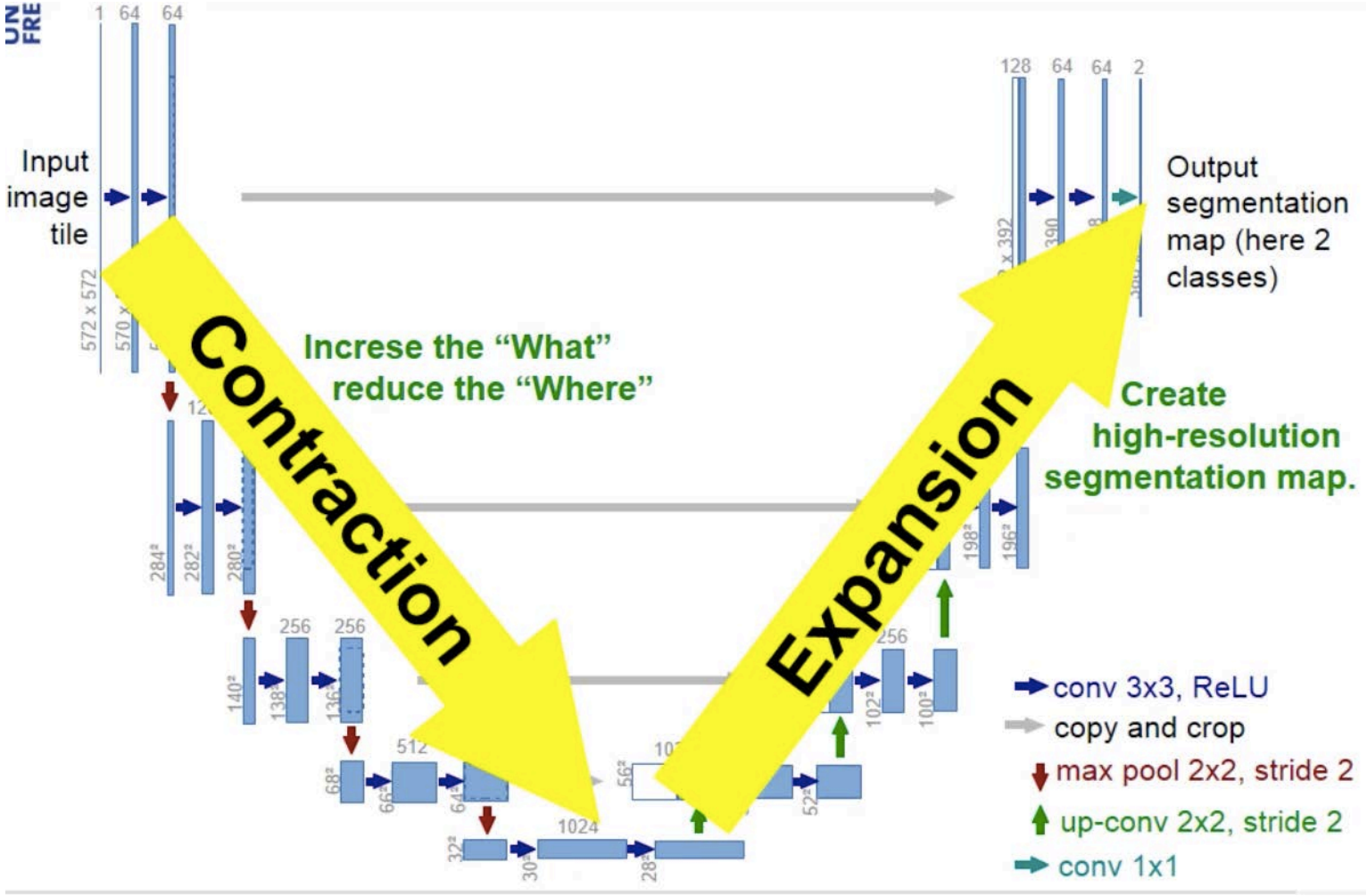
2X2 MAX POOLING



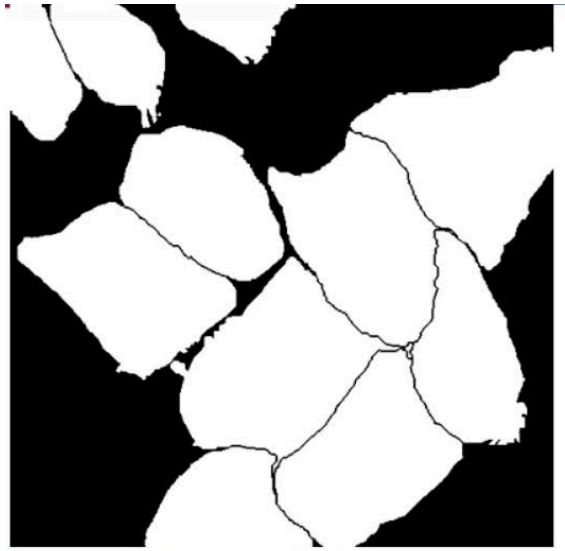
2X2 UP-CONVOLUTION



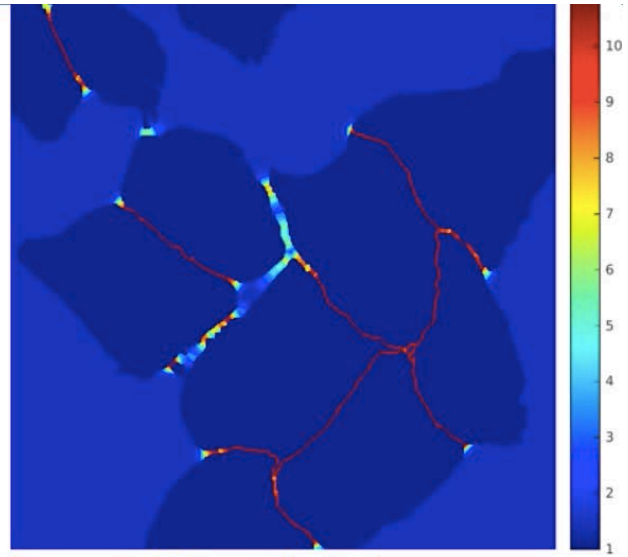
U-NET ARCHITECTURE



BOUNDARY LOSS



Segmentation mask for training (inserted background between touching objects)



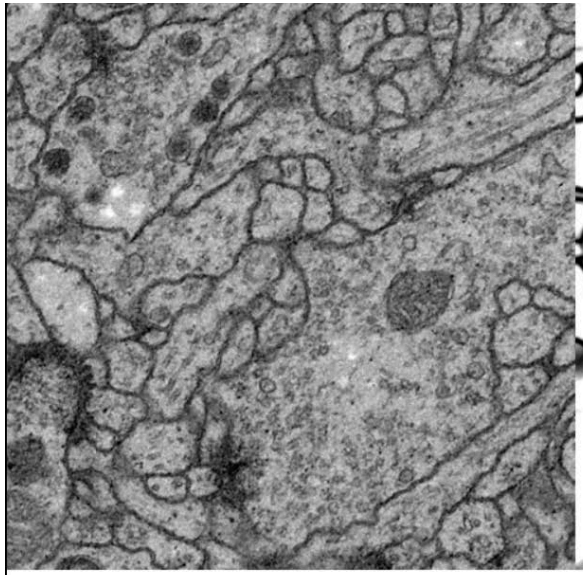
Loss weight for each pixel

The separation border is computed using morphological operations. The weight map is then computed as

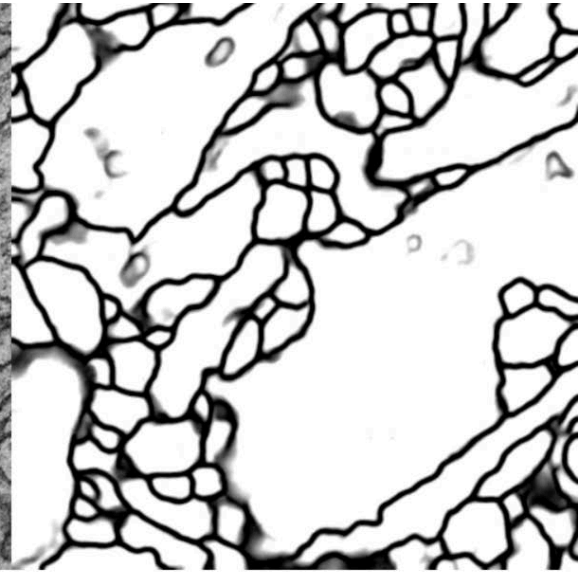
$$w(\mathbf{x}) = w_c(\mathbf{x}) + w_0 \cdot \exp\left(-\frac{(d_1(\mathbf{x}) + d_2(\mathbf{x}))^2}{2\sigma^2}\right) \quad (2)$$

where $w_c : \Omega \to \mathbb{R}$ is the weight map to balance the class frequencies, $d_1 : \Omega \to \mathbb{R}$ denotes the distance to the border of the nearest cell and $d_2 : \Omega \to \mathbb{R}$ the distance to the border of the second nearest cell. In our experiments we set $w_0 = 10$ and $\sigma \approx 5$ pixels.

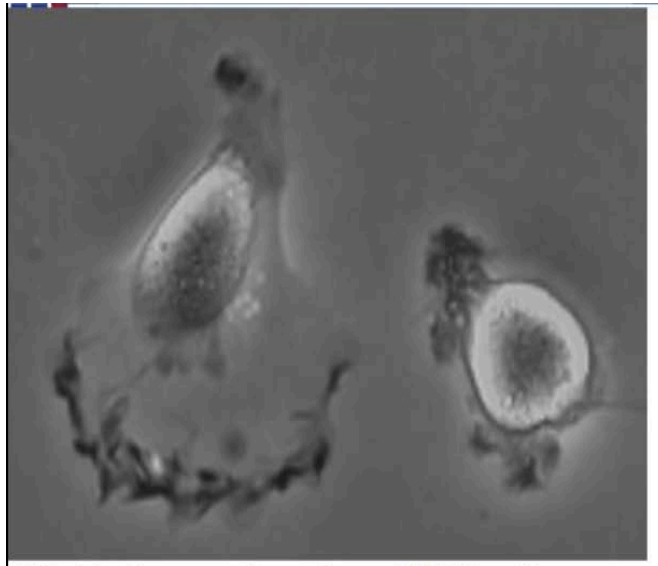
U-NET COMPUTATIONAL EXAMPLES



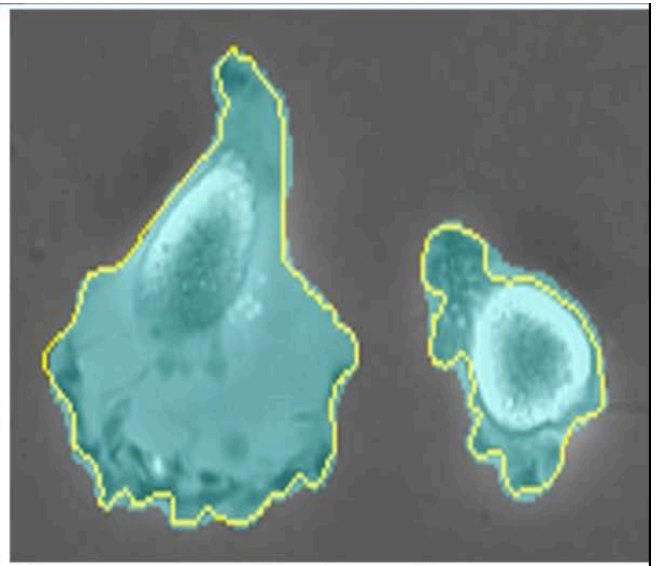
Input image



Our result: 0.000353 warping error
(**New best score** at submission march 6th, 2015)
Sliding-window CNN: 0.000420
Training time: 10h, Application: 1s per image

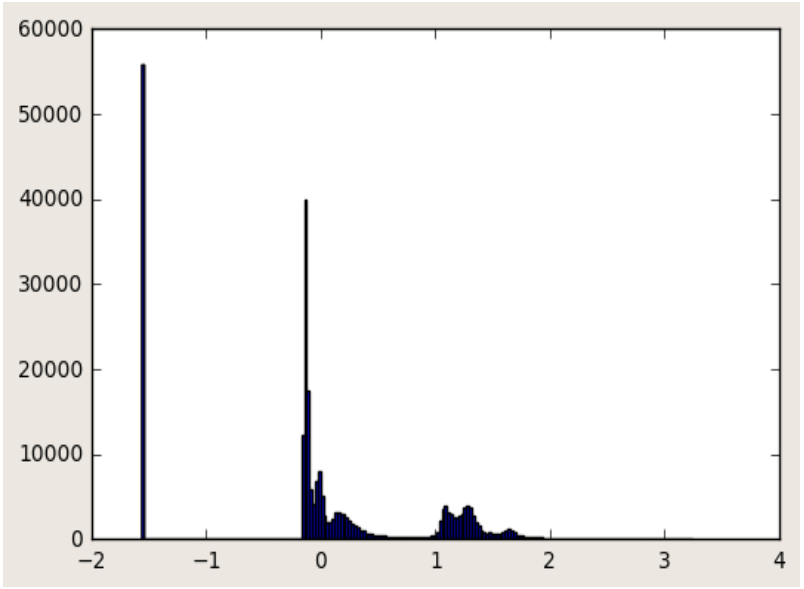


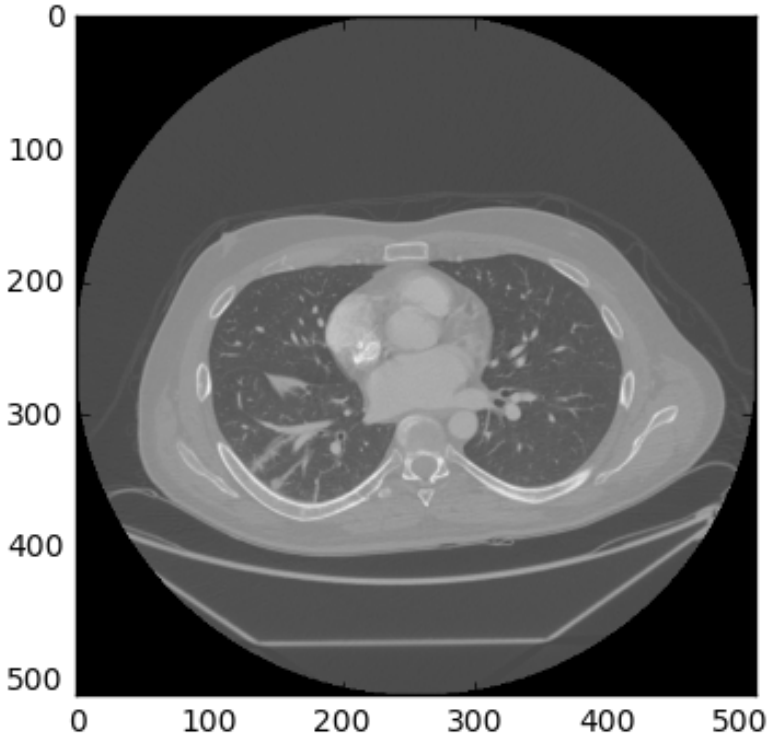
Glioblastoma-astrocytoma U373 cells on a polyacrylimide substrate
Phase contrast microscopy



Cyan: segmentation by u-net
Yellow borders: our manual ground truth

preprocessing






```
img = imgs_to_process[i]
#Standardize the pixel values
mean = np.mean(img)
std = np.std(img)
img = img-mean
img = img/std
plt.hist(img.flatten(),bins=200)
```

The underflow peak near -1.5 is the black out-of-scanner part of the image. The peaks around 0.0 are the background and lung interior and the wide clumps from 1.0 to 2.0 are the non-lung-tissue and bone. The structure of this histogram varies throughout the data set. Two images are shown below that are typical of the data set.

THE HU OF COMMON SUBSTANCES

The **Hounsfield scale** /ˈhaʊnzˌfiːld/ or **CT numbers**, named after Sir Godfrey Newbold Hounsfield, is a quantitative scale for describing radiodensity.

By comparison, conventional X-ray images are two-dimensional projections of the true three-dimensional anatomy, i.e. radiodensity shadows.

Substance	HU
Air	–1000
Lung	–500
Fat	–100 to –50
Water	0
CSF	15
Kidney	30
Blood	+30 to +45
Muscle	+10 to +40
Grey matter	+37 to +45
White matter	+20 to +30
Liver	+40 to +60
Soft Tissue, Contrast	+100 to +300
Bone	+700 (cancellous bone) to +3000 (cortical bone)

IMAGE COORDINATE SYSTEM

- The image coordinate system describes how an image was acquired with respect to the anatomy. Medical scanners create regular, rectangular arrays of points and cells which start at the upper left corner. The *i* axis increases to the right, the *j* axis to the bottom and the *k*axis backwards.
- In addition to the intensity value of each voxel *(i j k)* the origin and spacing of the anatomical coordinates are stored too.
- The origin represents the position of the first voxel (0,0,0) in the anatomical coordinate system, e.g. (100mm, 50mm, -25mm)
- The spacing specifies the distance between voxels along each axis, e.g. (1.5mm, 0.5mm, 0.5mm)
- The right 2D example shows the meaning of origin and spacing:

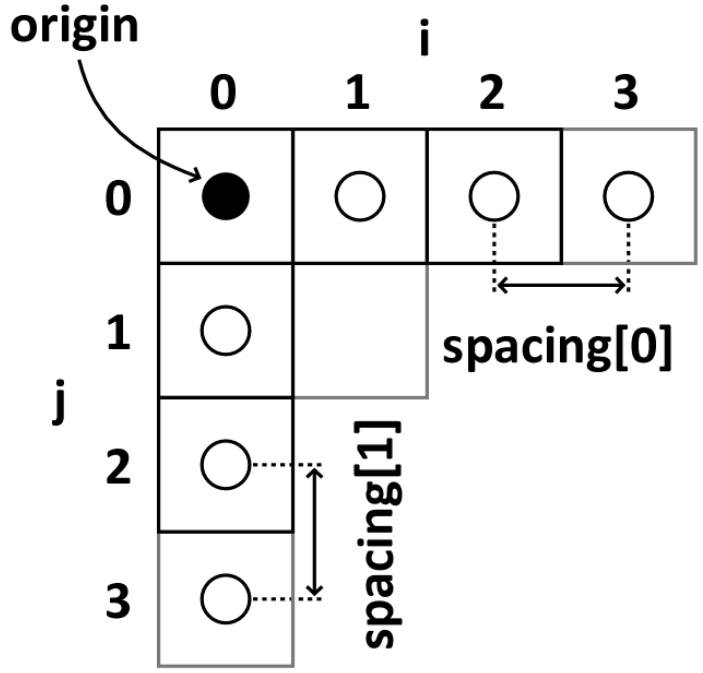


IMAGE TRANSFORMATION

The transformation from an image space vector $(i\ j\ k)'$ to an anatomical space vector $\vec{x}$ is an affine transformation, consists of a linear transformation A followed by a translation $\vec{t}$.

$$\vec{x} = A(i \quad j \quad k)' + \vec{t}$$

The transformation matrix A is a 3×3 matrix and carries all information about space directions and axis scaling.

$\vec{t}$ is a 3×1 vector and contains information about the geometric position of the first voxel.

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix} \begin{pmatrix} i \\ j \\ k \end{pmatrix} + \begin{pmatrix} t_1 \\ t_2 \\ t_3 \end{pmatrix}$$

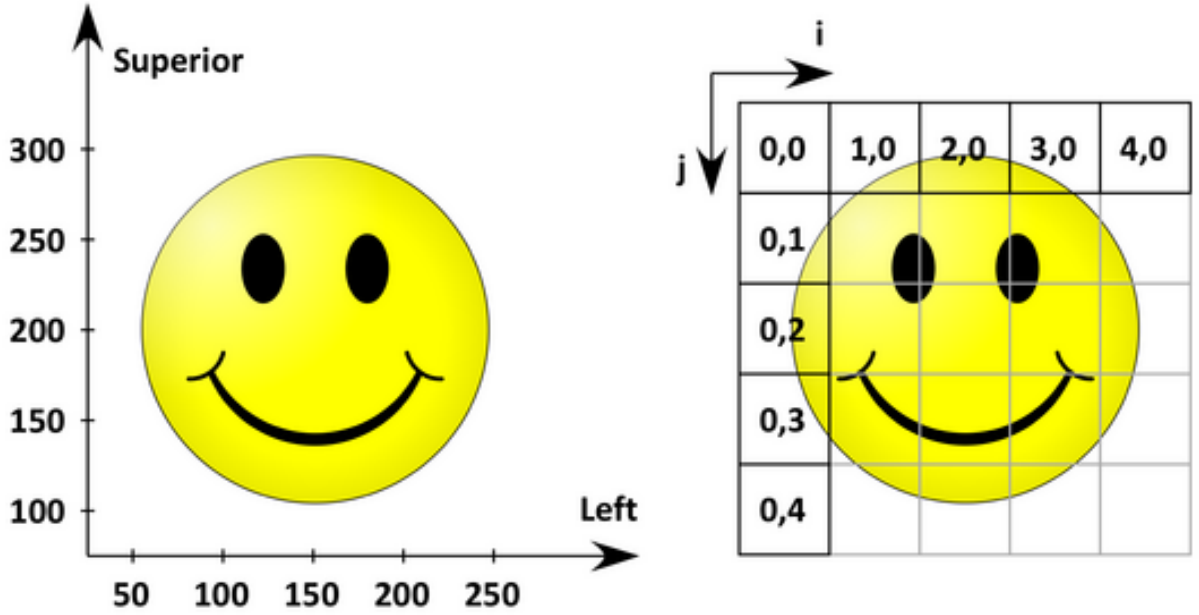
The last equation shows that the linear transformation is performed by a matrix multiplication and the translation by a vector addition. To represent both, the transformation and the translation, by a matrix multiplication an augmented matrix must be used. This technique requires that the matrix A is augmented with an extra row of zeros at the bottom, an extra column-the translation vector-to the right, and a '1' in the lower right corner. Additionally all vectors have to be written as homogeneous coordinates, which means that a '1' is augmented at the end.

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ 1 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & A_{13} & t_1 \\ A_{21} & A_{22} & A_{23} & t_2 \\ A_{31} & A_{32} & A_{33} & t_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ k \\ 1 \end{pmatrix}$$

Depending on the used anatomical space (LPS or RAS) the 4×4 matrix is called **IJKtoLPS**- or **IJKtoRAS**-matrix, because it represents the transformation from IJK to LPS or RAS.

2D EXAMPLE OR CALCULATING AN *IJTOLS*-MATRIX

- The following figure shows the anatomical space with a L(P)S basis on the left and the corresponding image coordinates on the right.



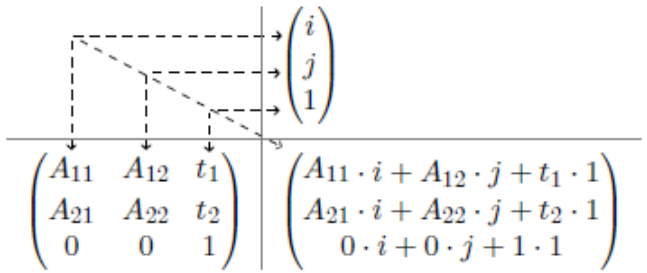
The origin (the coordinates of the first 'pixel' in anatomical space) is *(50 mm, 300 mm)* and the spacing (the distance between two pixels) is *(50 mm, 50 mm)*.

The origin (the coordinates of the first 'pixel' in anatomical space) is *(50 mm, 300 mm)* and the spacing (the distance between two pixels) is *(50 mm, 50 mm)*.

As this is a 2D example A is a 2×2 matrix and $\vec{t}$ a 2×1 vector. Therefore the equation of the affine transformation is:

$$\begin{pmatrix} L \\ S \\ 1 \end{pmatrix} = \begin{pmatrix} A_{11} & A_{12} & t_1 \\ A_{21} & A_{22} & t_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} i \\ j \\ 1 \end{pmatrix}$$

By multiplying the **IJtoLS**-matrix and the vector of the right side, the following product will be obtained:



$$\begin{pmatrix} A_{11} & A_{12} & t_1 \\ A_{21} & A_{22} & t_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} A_{11} \cdot i + A_{12} \cdot j + t_1 \cdot 1 \\ A_{21} \cdot i + A_{22} \cdot j + t_2 \cdot 1 \\ 0 \cdot i + 0 \cdot j + 1 \cdot 1 \end{pmatrix}$$

The last equation and the matrix product show that a total of 6 unknown variables $(A_{11}, A_{12}, A_{21}, A_{22}, t_1, t_2)$ have to be determined. The knowledge of origin and spacing however allows the following relations between image and anatomical space:

$$\begin{pmatrix} L \\ S \end{pmatrix} \equiv \begin{pmatrix} i \\ j \end{pmatrix} \qquad \begin{pmatrix} 50 \\ 300 \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 0 \end{pmatrix} \qquad \begin{pmatrix} 100 \\ 300 \end{pmatrix} \equiv \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad \begin{pmatrix} 50 \\ 250 \end{pmatrix} \equiv \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Thus, at least six equations can be derived:

$$\begin{aligned} 50 &= A_{11} \cdot 0 + A_{12} \cdot 0 + t_1 \cdot 1 \\ 300 &= A_{21} \cdot 0 + A_{22} \cdot 0 + t_2 \cdot 1 \\ 100 &= A_{11} \cdot 1 + A_{12} \cdot 0 + t_1 \cdot 1 \\ 300 &= A_{21} \cdot 1 + A_{22} \cdot 0 + t_2 \cdot 1 \\ 50 &= A_{11} \cdot 0 + A_{12} \cdot 1 + t_1 \cdot 1 \\ 250 &= A_{21} \cdot 0 + A_{22} \cdot 1 + t_2 \cdot 1 \end{aligned}$$

As mentioned above, the translation $\vec{t}$ contains the information about the geometric position of the first pixel and is therefore equivalent to the origin. This result is also confirmed by the first equations.

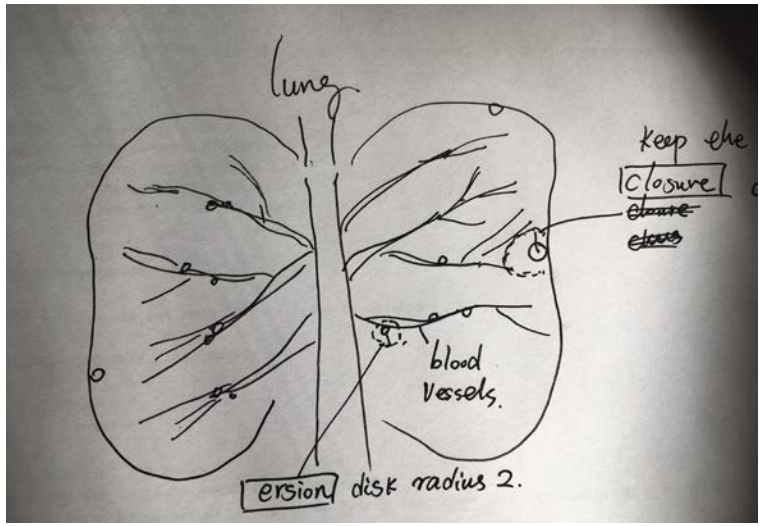
The solution of the other equations leads to the following **IJtoLS**-matrix:

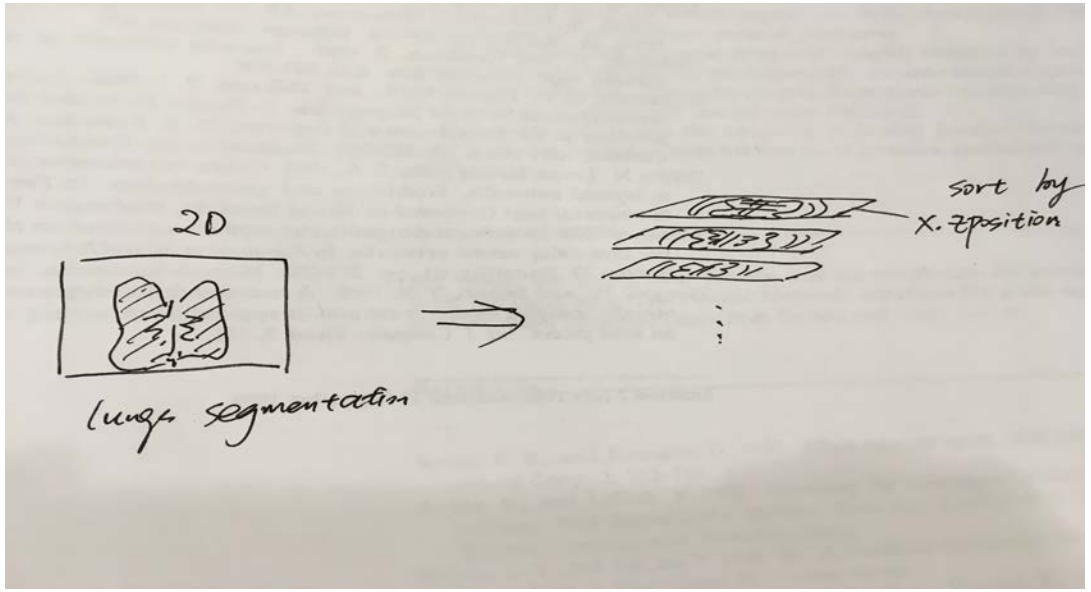
$$IJtoLS = \begin{pmatrix} 50 & 0 & 50 \\ 0 & -50 & 300 \\ 0 & 0 & 1 \end{pmatrix}$$

In the event that a R(A)S basis was used, just the left and anterior axis of the anatomical space are flipped, and the image coordinate system appears in the same way as in the L(P)S case.

DATA PREPROCESSING: CODES

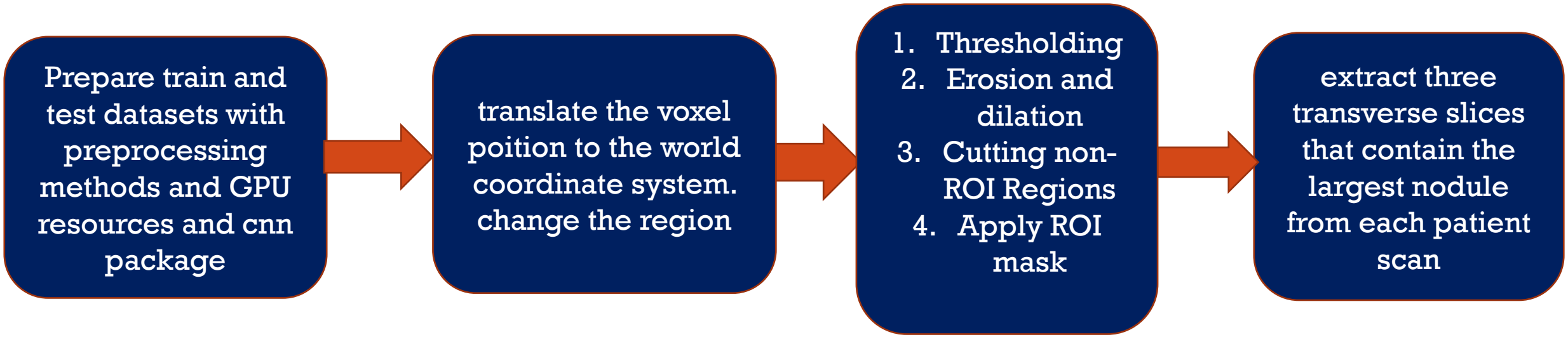
- Python: 3D lungs construction and nodules generation
- Open jupyter notebook
- R: introduction of medical image processing





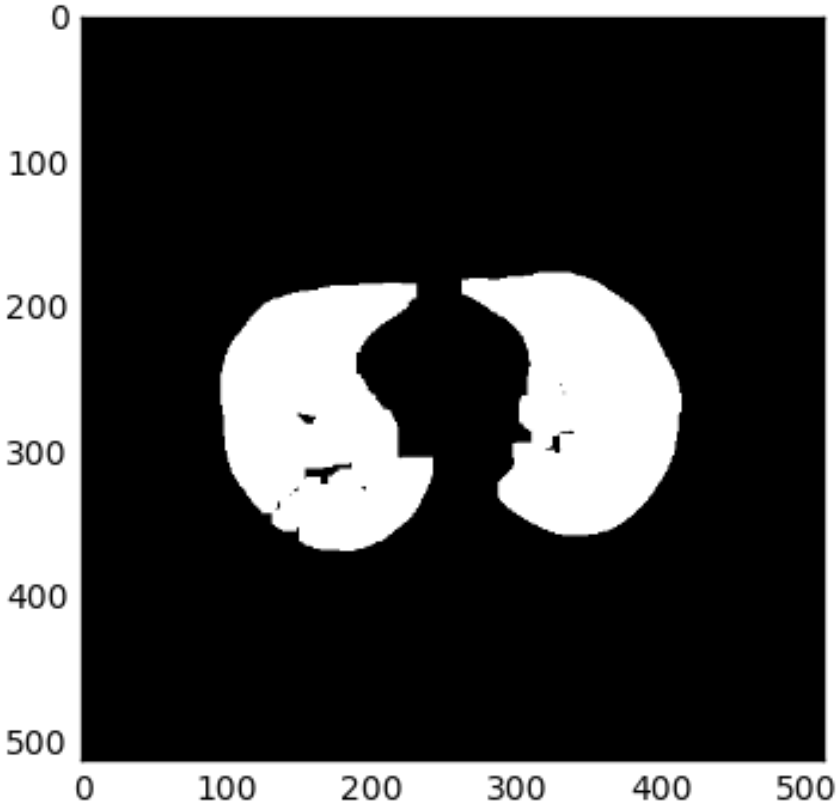
PROJECT SCHEME PART I

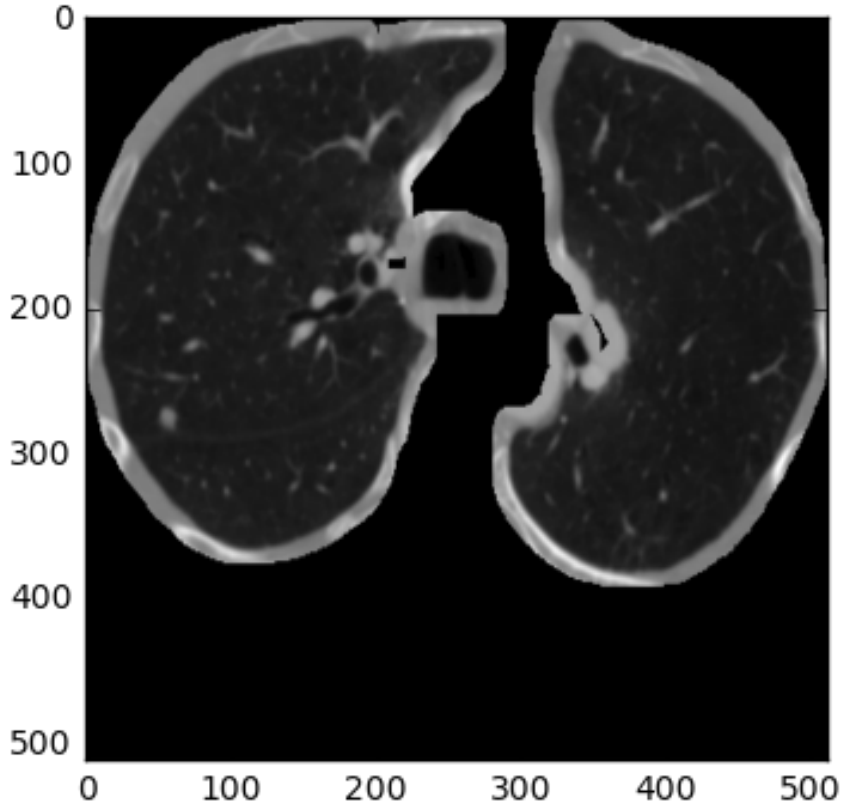
- We will train a network to segment out potentially cancerous nodules and then use the characteristics of that segmentation to make predictions about the diagnosis of the scanned patient within a 12 month time frame.



output will be two files for each patient scan:
a set of images and a set of corresponding nodule masks.

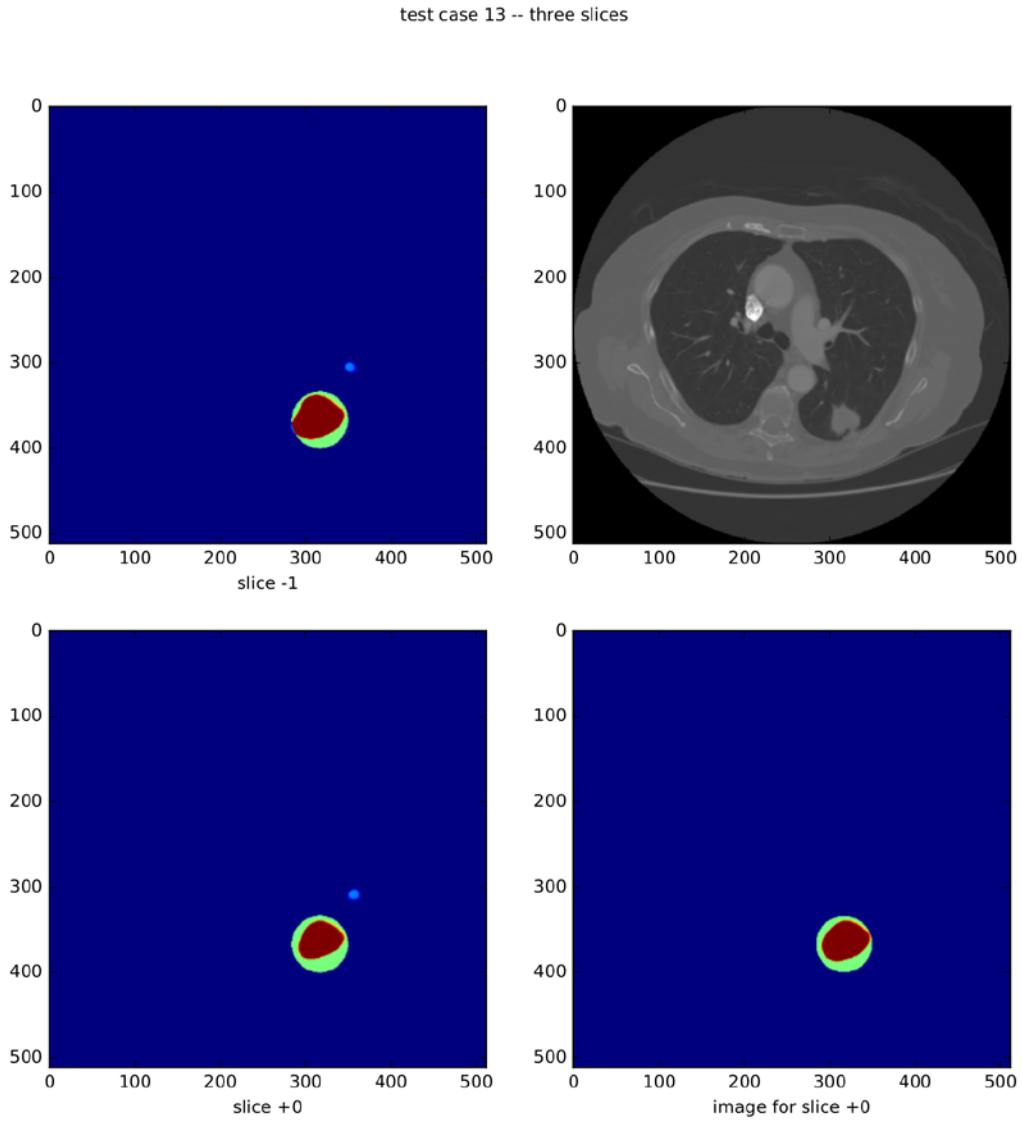
ROI REGION AND MASK APPLIED



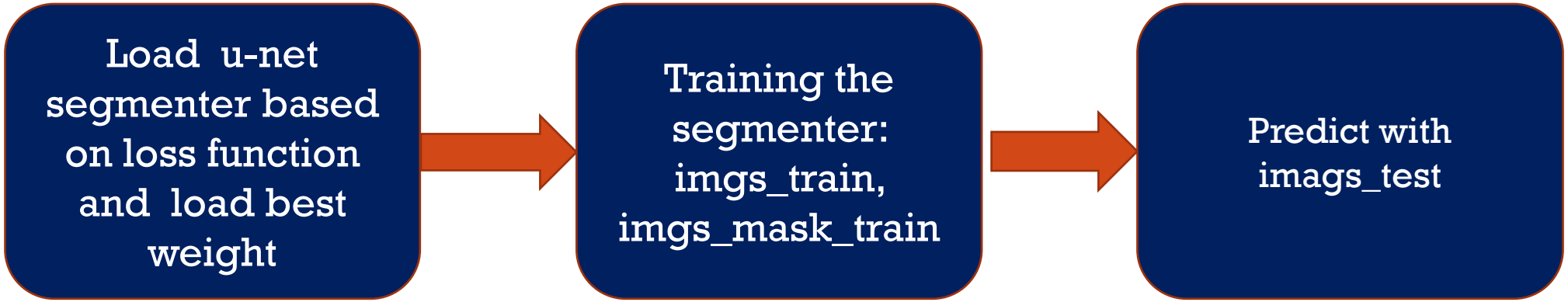


LUNA MASK

- An example segmentation is given here for the three slices taken from a patient scan. The perfect circle is the "true" node mask from the LUNA annotationc.csv file, and the red is the predicted node region from the segmenter. The original image is given in the top right.



PROJECT SCHEME PART II



- Here, imgs_train is the original images after resize, rescale and change the region.
- And the imgs_mask_train are the imgs generated by extract three transverse slices that contain the largest nodule from each patient scan

IMPROVEMENT OF U-NET PREDICTION

- There are many ways we can improve the U-net model.
- 1.increase datasets:

Larger dataset can give us better u-net model

2. add more mask images to input dataset

But this will need a lot of more time.

3. change the scheme of finding and create mask

Which can be improved based on many medical reports

OLD VERSION CLASSIFIER BASED ON FEATURES

- As we can always make classification based on features of pictures.
- For this problem we can create a lot of features and use cancer label to make classification models.
- Features like bellow:
- avgArea = totalArea / numNodes
- weightedX = weightedX / totalArea
- weightedY = weightedY / totalArea
- numNodesperSlice = numNodes*1. / nslices

RANDOM FOREST AND XGBOOST

```
Random Forest
             precision    recall  f1-score   support
  No Cancer       0.81      0.98      0.89       463
     Cancer       0.17      0.02      0.03       107
avg / total       0.69      0.80      0.73       570
('logloss', 0.52600332670816652)
XGBoost
             precision    recall  f1-score   support
  No Cancer       0.83      0.86      0.84       463
     Cancer       0.27      0.21      0.24       107
avg / total       0.72      0.74      0.73       570
('logloss', 0.5700685138621493)
Predicting all positive
             precision    recall  f1-score   support
  No Cancer       0.00      0.00      0.00       463
     Cancer       0.19      1.00      0.32       107
avg / total       0.04      0.19      0.06       570
('logloss', 28.055831025357818)
Predicting all negative
             precision    recall  f1-score   support
  No Cancer       0.81      1.00      0.90       463
     Cancer       0.00      0.00      0.00       107
avg / total       0.66      0.81      0.73       570
('logloss', 6.4835948671148085)
```

RESOURCES

- 1. https://www.kaggle.com/c/data-science-bowl-2017#tutorial
- 2. https://lmb.informatik.uni-freiburg.de/people/ronneber/**u-net**/
- 3. https://www.kaggle.com/c/ultrasound-nerve-segmentation/forums/t/21358/0-57-deep-learning-keras-tutorial
- 4. https://luna16.grand-challenge.org/

THANK YOU!