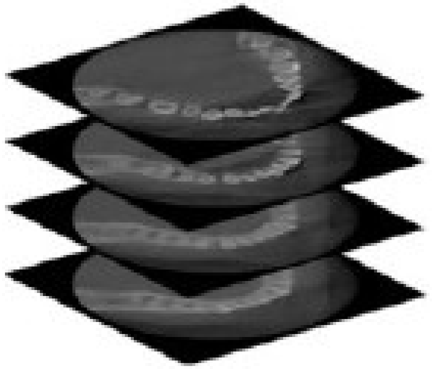
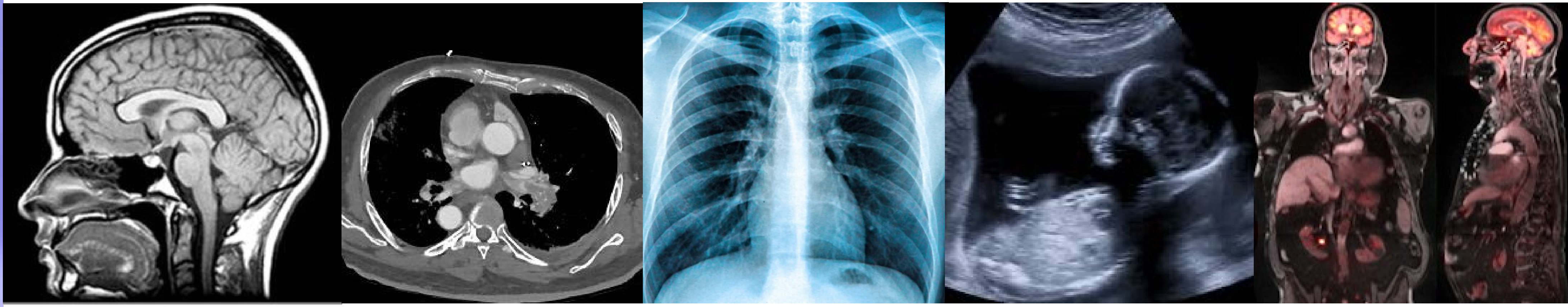
Convolutional Neural Networks for MRI and CT DICOM Files

Juan Castillo

Introduction:

- DICOM files are used for MRI, CT, PET, SPECT, ultrasound and standard X-Ray.
- They can be used to store 2D and 3D information and include patient infos and all scan parameters.
- A single scan can have a size of up to 1 GiB.
- Python libraries for DICOM:
    - Pydicom (preprocessing)
    - IPVolume(visualization)

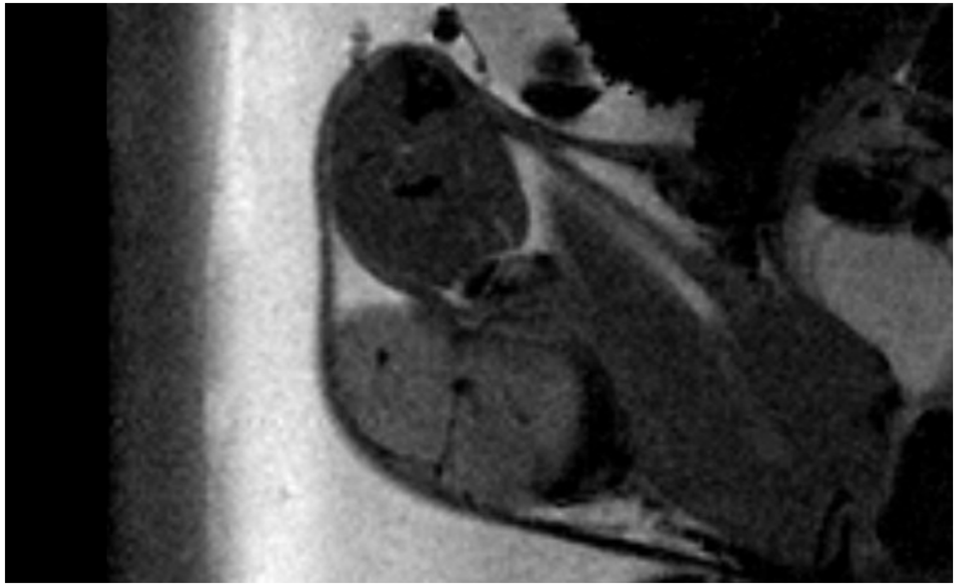




Dataset of MRI rat tumor scans

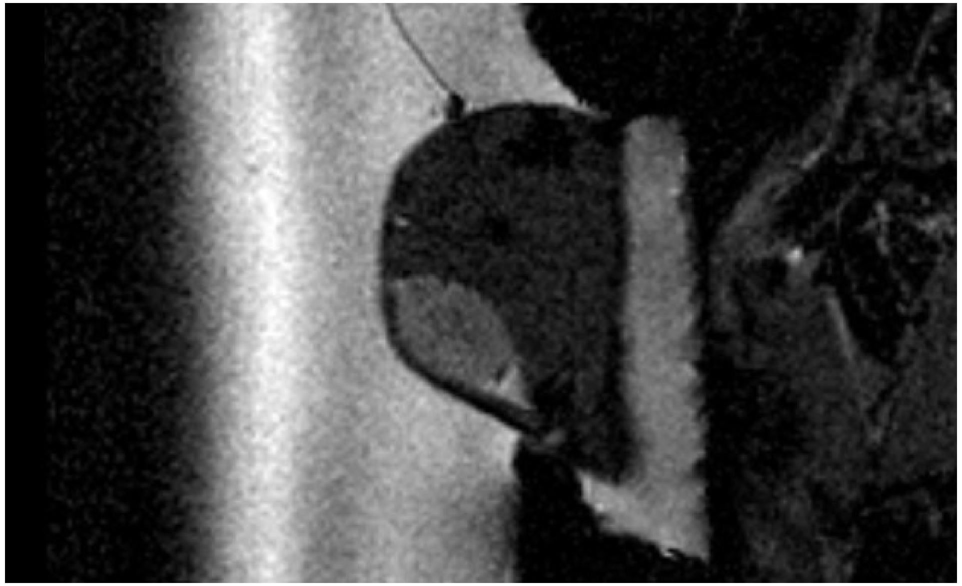
BN-175 tumor model:

- In the MRI it is darker and has necrotic parts (black spots)
- 51 FFE scans



R1 tumor model:

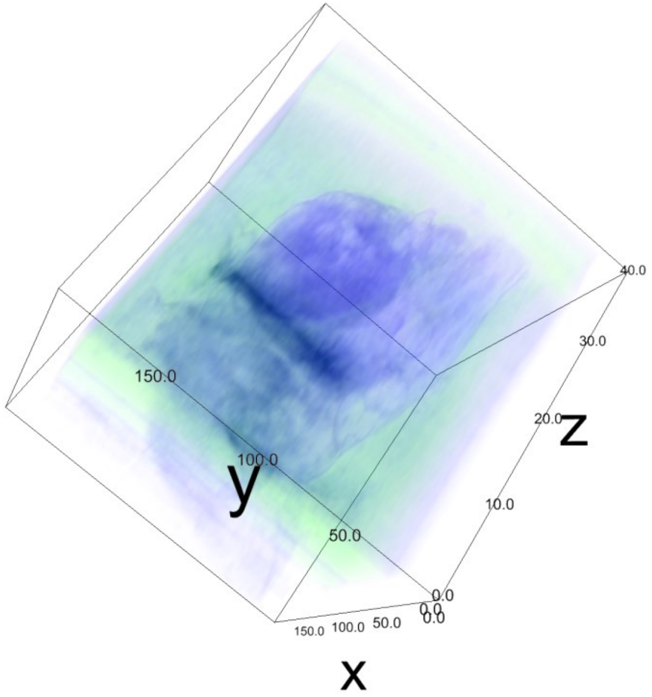
- In the MRI it is brighter and more uniform
- 20 FFE scans



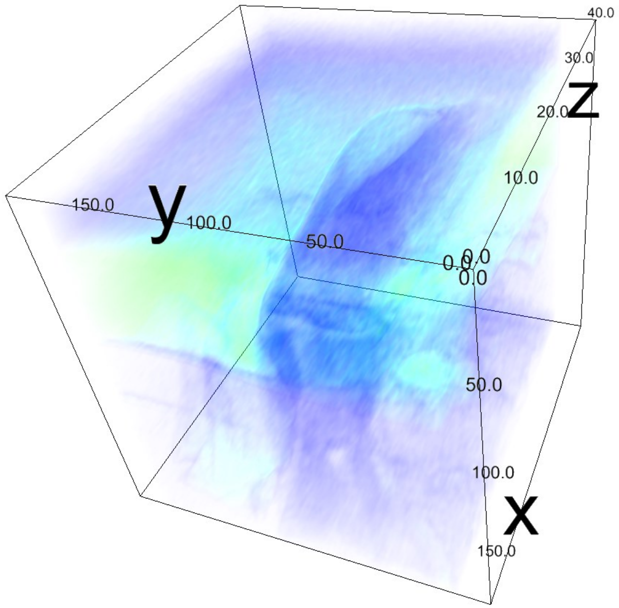
3D visualization of rat MRIs with IPVolume

All voxel information of all scans is merged to a list(all scans) of arrays(one scan) of arrays(one slice) of arrays(one line) of floats(one voxel)

BN-175 tumor model:



R1 tumor model:



Convolutional neural networks (CNNs) with rat models

Data preprocessing:

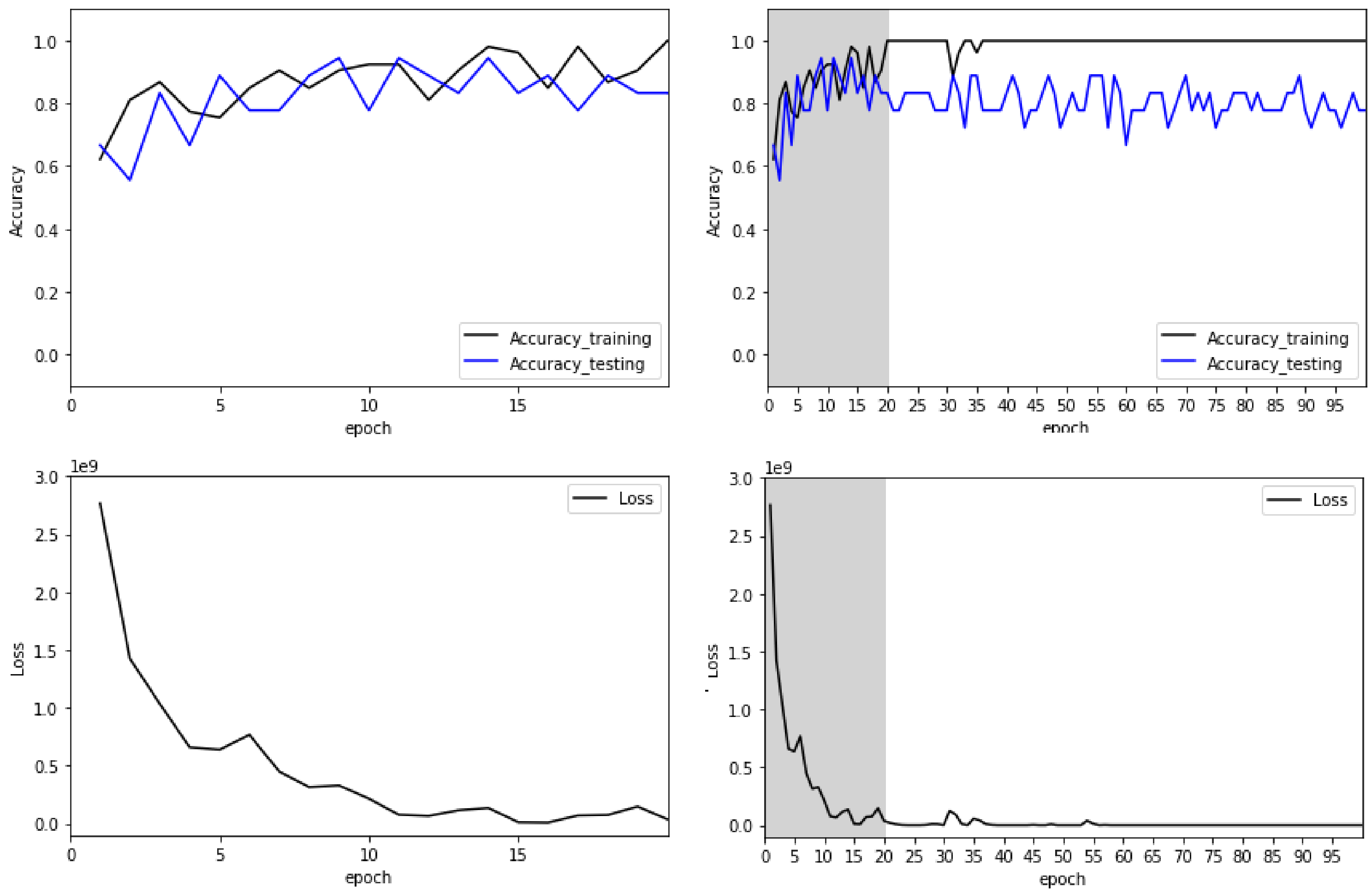
- Create pixel arrays of DICOM scans with **pydicom** and **numpy**
- Resize pixel arrays to 50x50x20 volumes with **cv2** and **math**
- Combine resized arrays for all scans in one list and save as .npy file

Arquitecture of the CNN:
(convolution box 3x3x3)

- Layer 1:
  - 32 nodes
  - Activation function: relu
- Layer 2:
  - 64 nodes
  - Activation function: relu
- Layer 3:
  - 1024 nodes (fc)
  - Activation function: relu
- Output layer:
  - 2 outputs (2 classes)
  - Activation function: softmax
  - Dropout: 20%

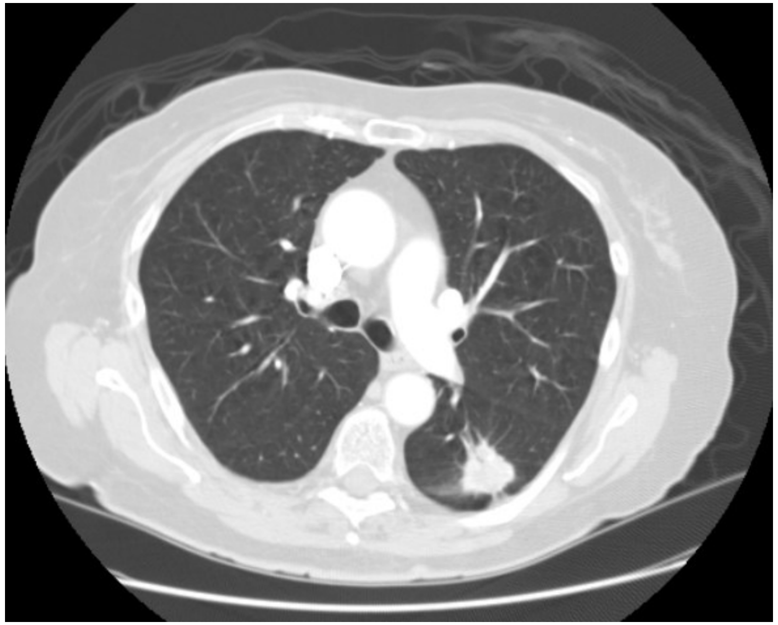
CNN results rat models

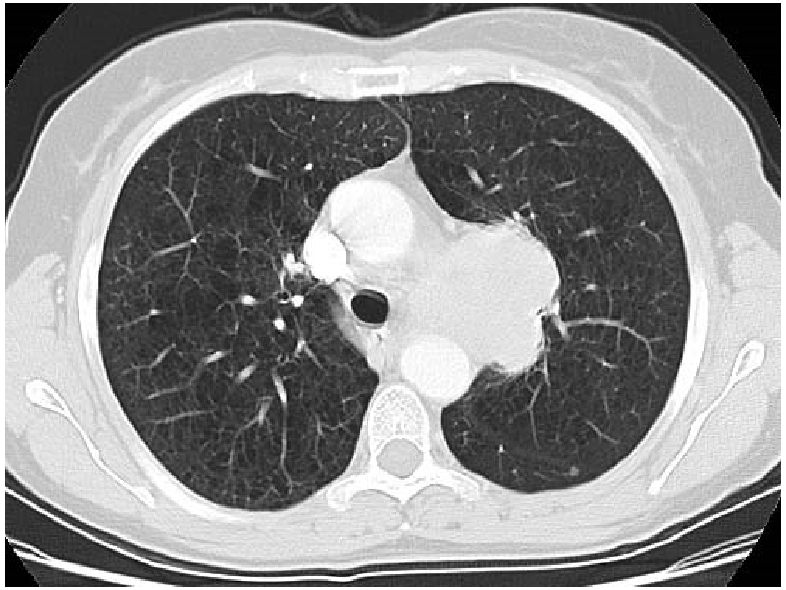
Epochs: 15, Loss: 10986040, Accuracy_training: 96%, Accuracy_testing: 94%
Epochs: 100, Loss: 0, Accuracy_training: 100%, Accuracy_testing: 77%



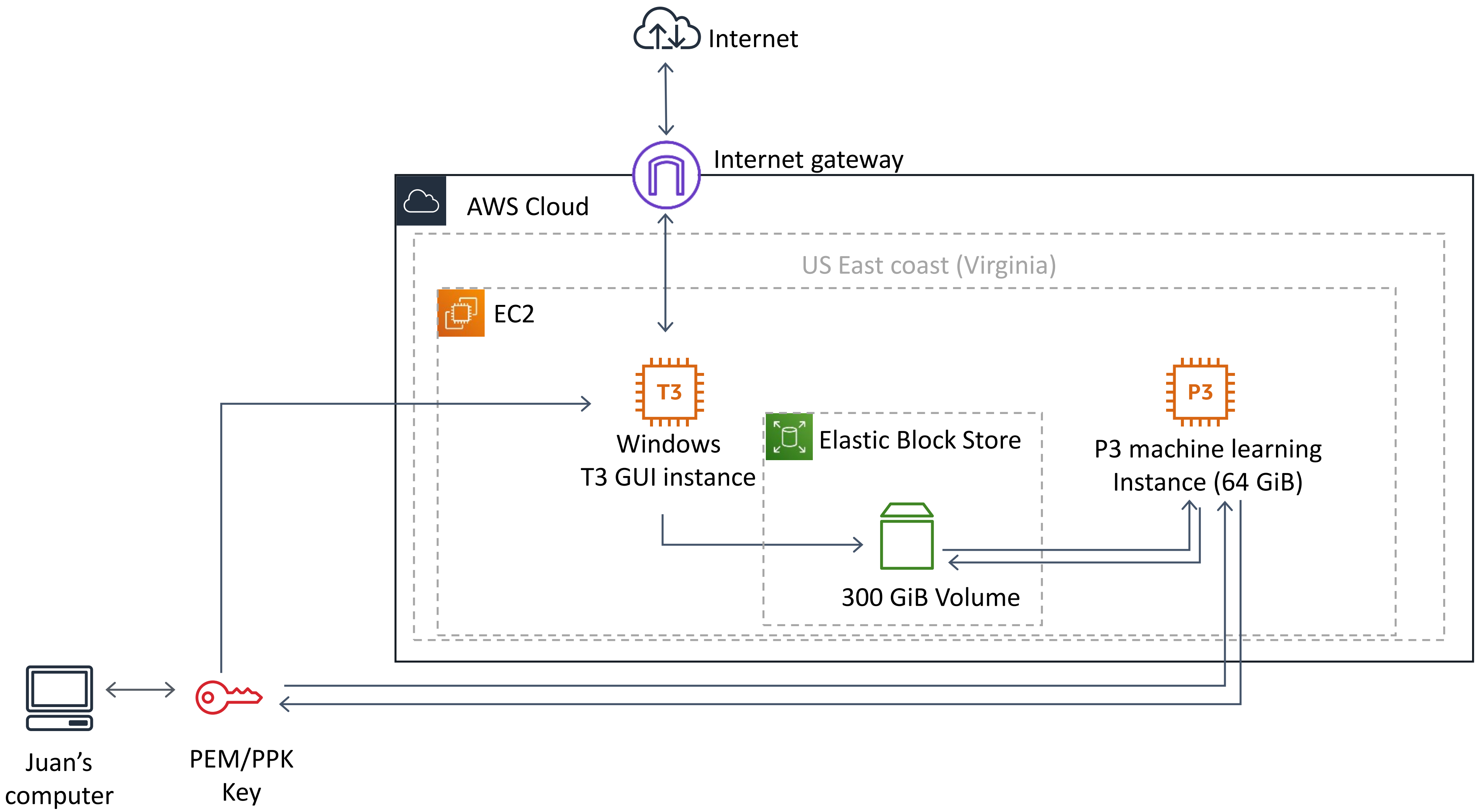
Human lung cancer dataset

- LIDC-IDRI Data set of CT scans of Patients with lung cancer and/or lung metastases:
- 1007 Patients.
- Each scan ca. 100-500 MB.
- For some Patients, also X-Ray and Survey scans (all also DICOM) included.
- Cleaning up necessary.
- Not possible to download and work up on local computer.
- Solution:
    - **Amazon AWS**

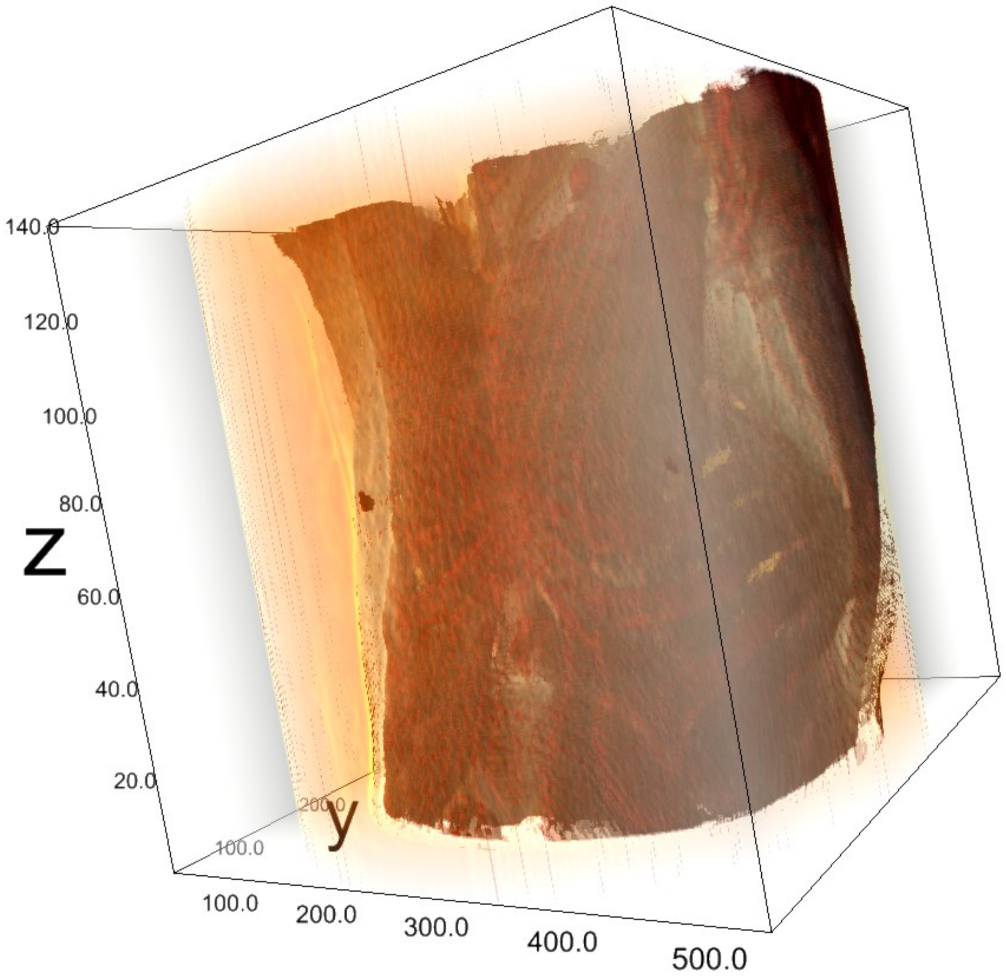


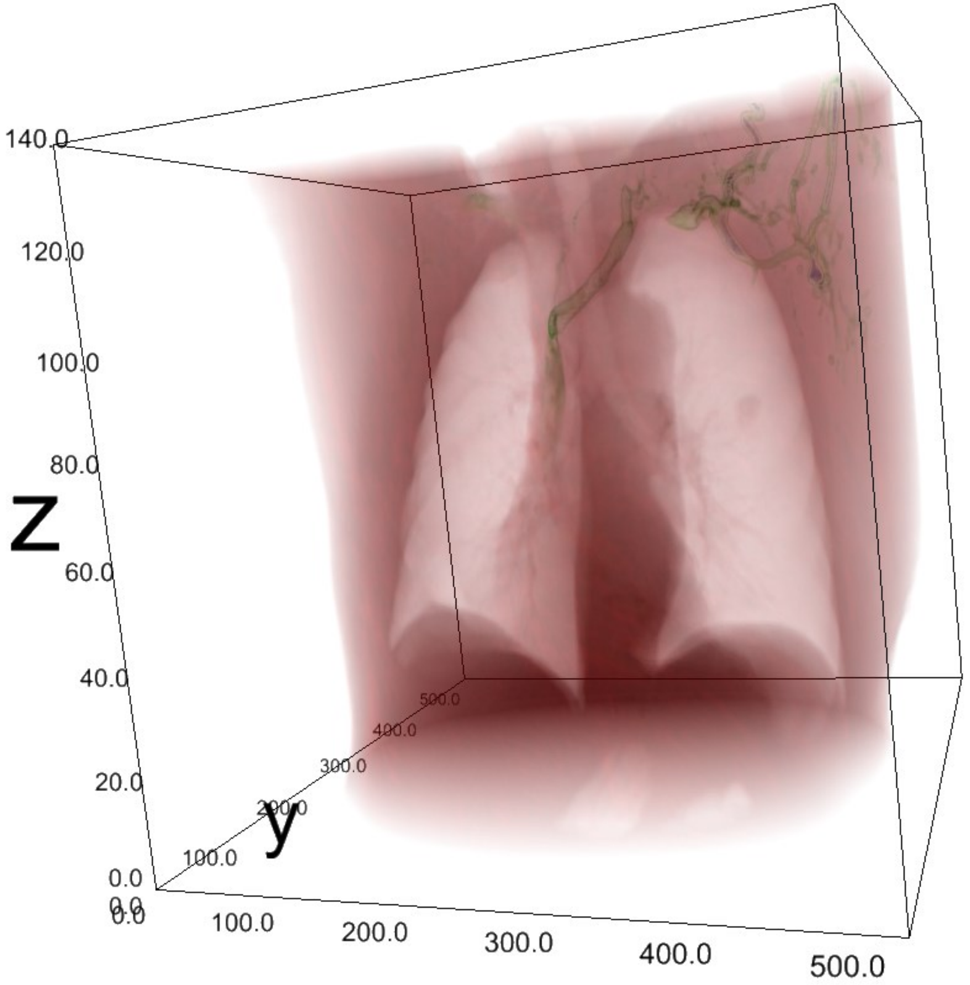


AWS architecture



3D visualization of human CTs with IPVolume





CNN with human data set

Arquitecture of the CNN (same as with rats):
(convolutions 3x3x3)

- Layer 1:
  - 32 nodes
  - Activation function: relu
- Layer 2:
  - 64 nodes
  - Activation function: relu
- Layer 3:
  - 1024 nodes (fc)
  - Activation function: relu
- Output layer:
  - 2 outputs (2 classes)
  - Activation function: softmax
  - Dropout: 20%

Feature engeneering:

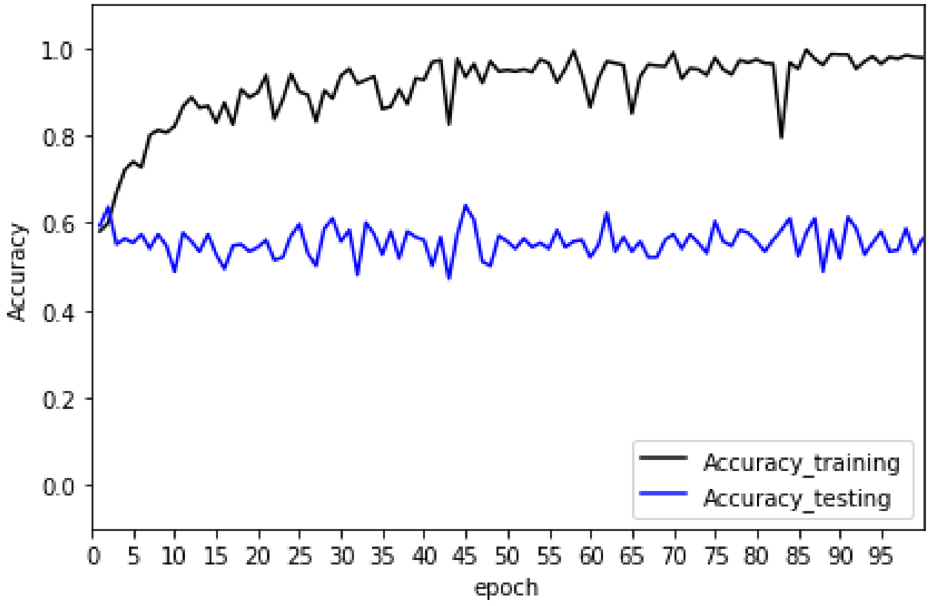
- Original data processing similar as for rat models.
- Resizing: 50x50x20 to 100x100x50.
- Data was split in two categories:
  - A: 0-1 nodules with diameter >= 3 mm
  - B:1-25 nodules with diameter >= 3 mm

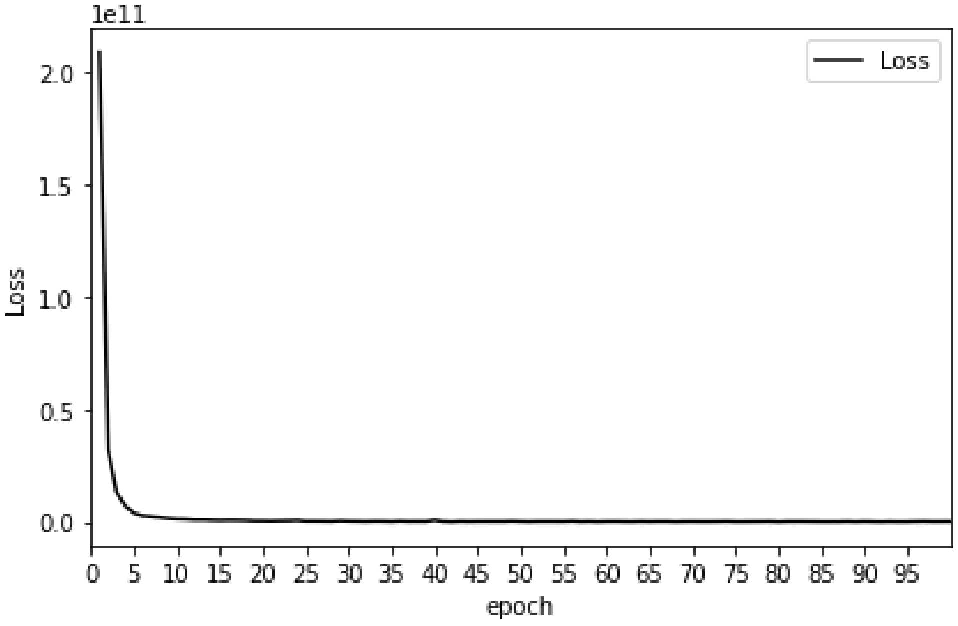
Problems:

- Each resizing step time consuming (several hours)
- Modeling time consuming.
- No good accuracy could be achieved (max 60%)
- Images too large compared to areas with relevant information. (Possible solution: Masking)

CNN results human data set

Scan resizing: 50x50x20
Epochs: 20, Loss: 458617109, Accuracy_training: 90%, Accuracy_testing: 54%
Epochs: 100, Loss: 160776626, Accuracy_training: 97%, Accuracy_testing: 56%





Work in progress / still to do:

- Implement YOLO algorithm to actually localize the tumor in rat models.
- Preselect slices that contain cancerous nodules in human CTs and only train with them.
- Introduce masking to human CTs CNN and retrain model.
- Expand human model to other disseases (COVID-19?)
- If possible, deploy models and/or build dashboard.

References:

- https://pythonprogramming.net/cnn-tensorflow-convolutional-nerual-network-machine-learning-tutorial/
- https://www.youtube.com/watch?v=CPZ5ihaNfJc&t=1259s
- https://github.com/pydicom/pydicom
- https://github.com/maartenbreddels/ipyvolume
- https://wiki.cancerimagingarchive.net/display/Public/LIDC-IDRI

Thank you for your attention!