MEDICAL IMAGE SEGMENTATION IN MRI SCAN IMAGES

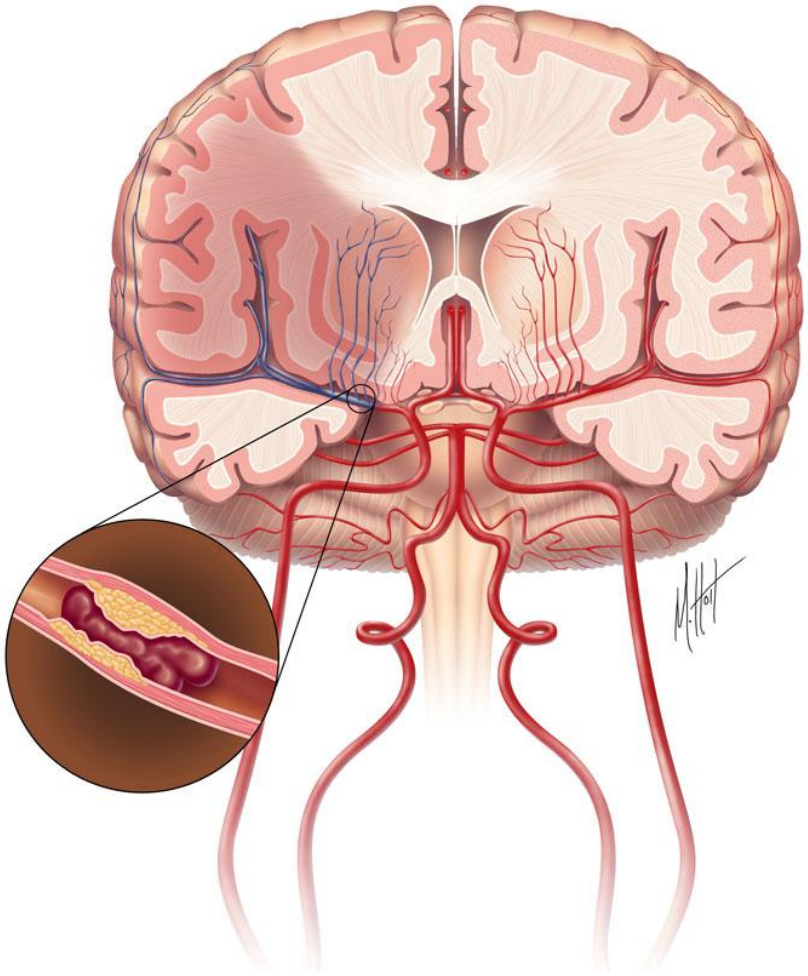
Graduation Project

Cem Yusuf Aydoğdu, 150120251

Advisor: Assoc. Prof. Dr. Gözde ÜNAL

Contents

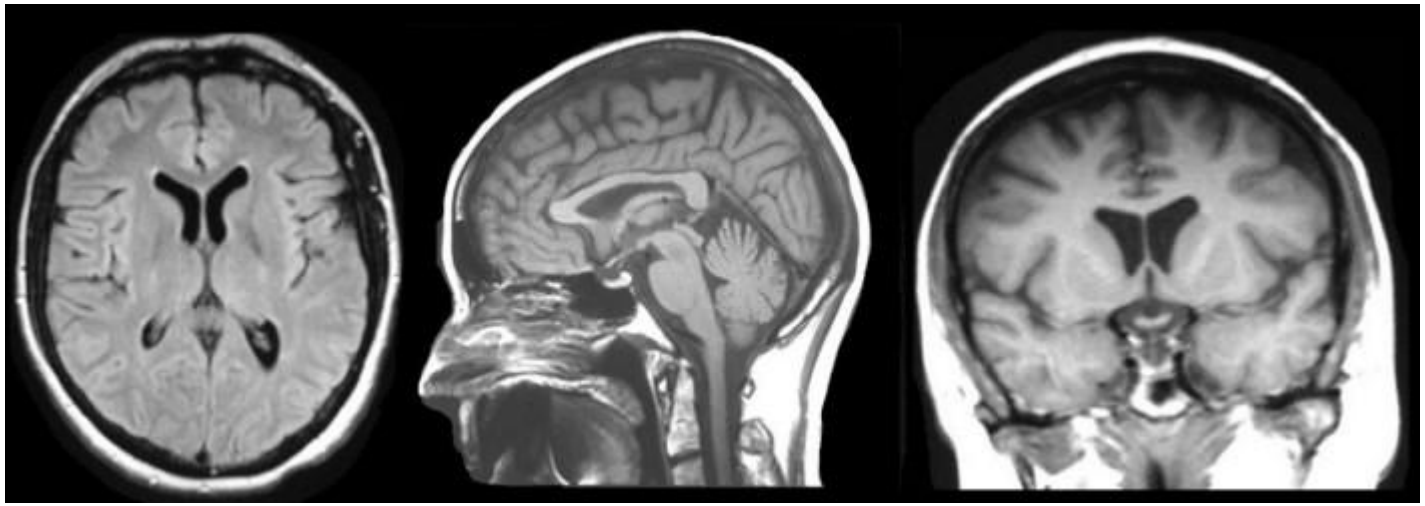
Problem - Ischemic Stroke



- Effects 15 million individuals each year
- Second most common mortality factor among elders
- Fifth most common mortality factor among youngers

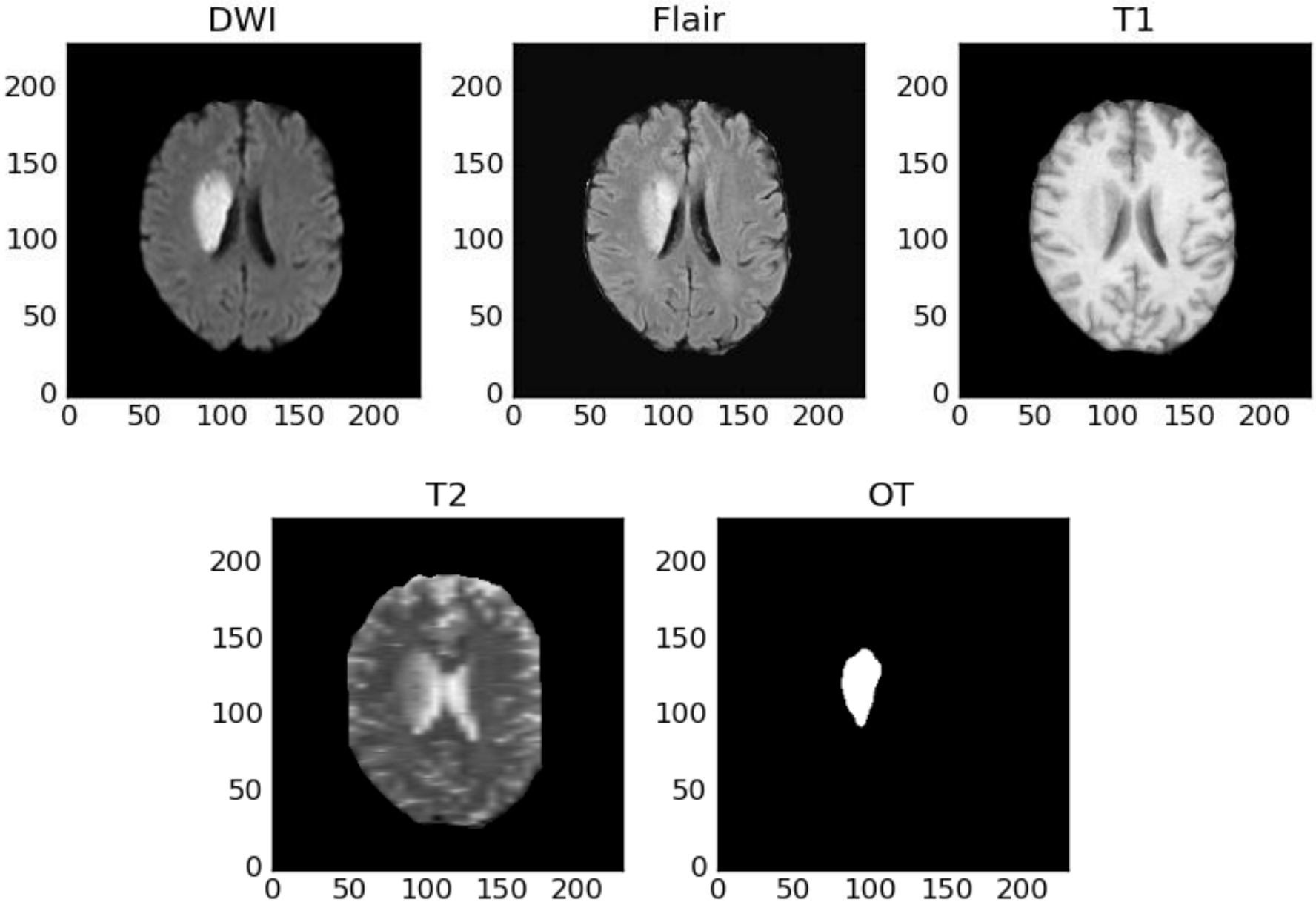
Magnetic Resonance Imaging

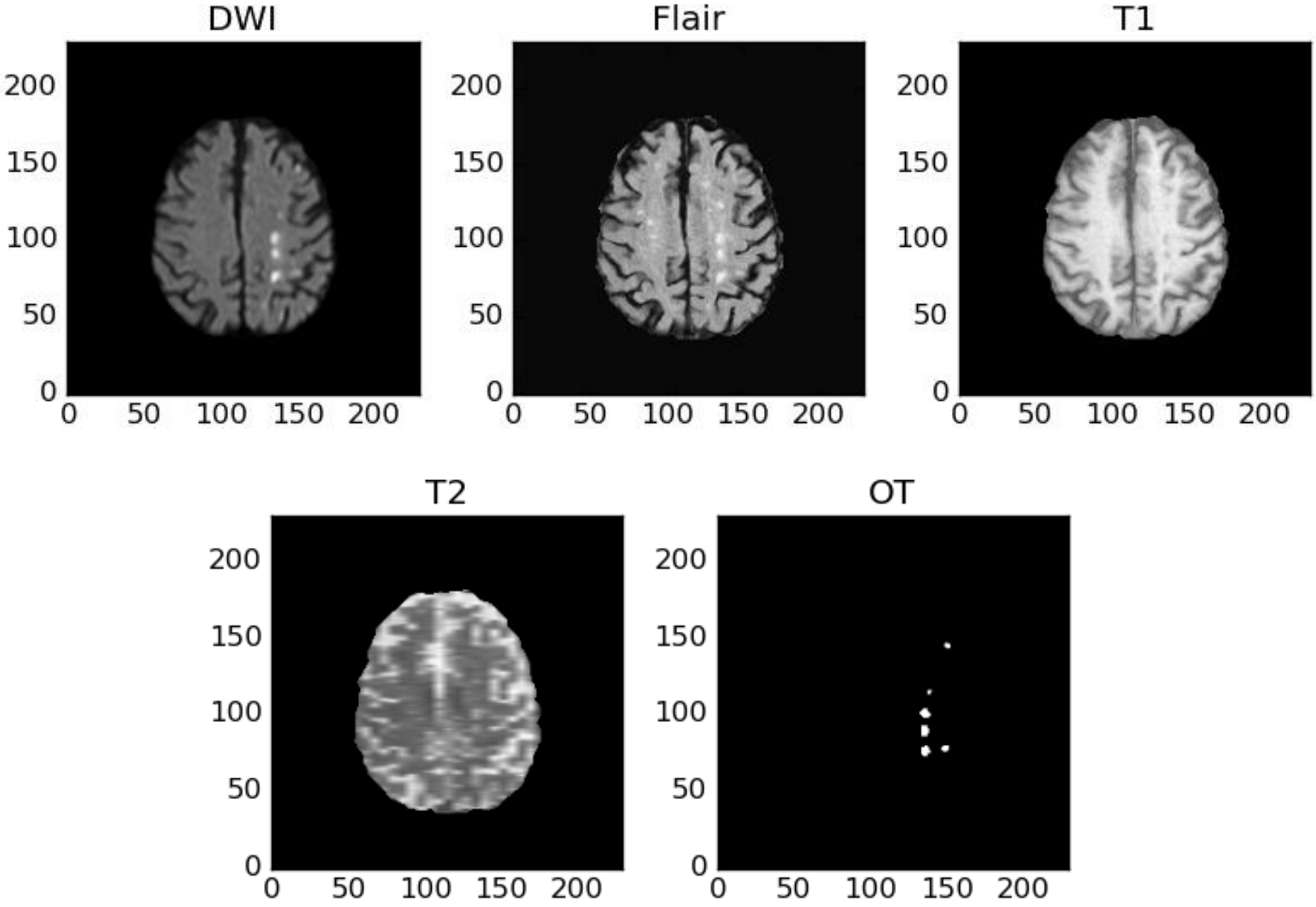
- A non-invasive imaging technique
- Used in diagnosing ischemic stroke disease
- Various modalities to distinguish distinct tissue types



Data Source of the Project

- Obtained from "2015 MICCAI Ischemic Stroke Segmentation Challenge"
- 3D MRI volumes from 26 patients
- Four different modalities (Flair, T1, T2, DWI) and one ground-truth (OT) volumes for each patient
- For each patient, all volumes are pre-registered

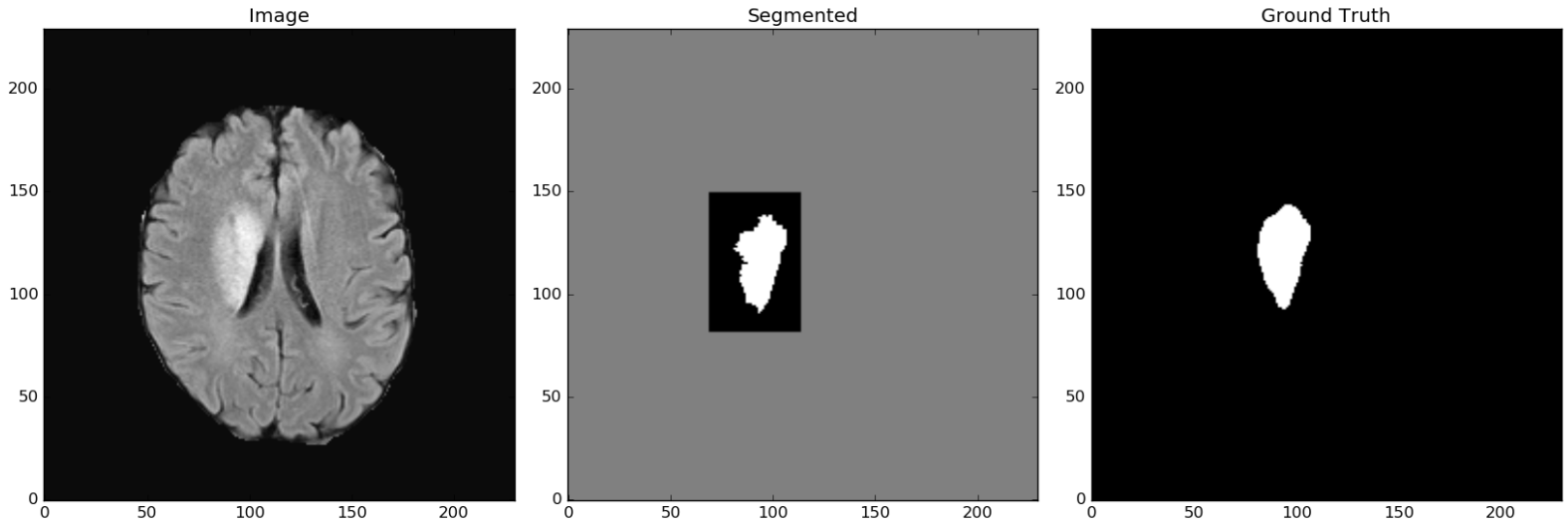

DWI
Flair
T1
T2
OT
0
50
100
150
200


DWI
Flair
T1
T2
OT
0
50
100
150
200

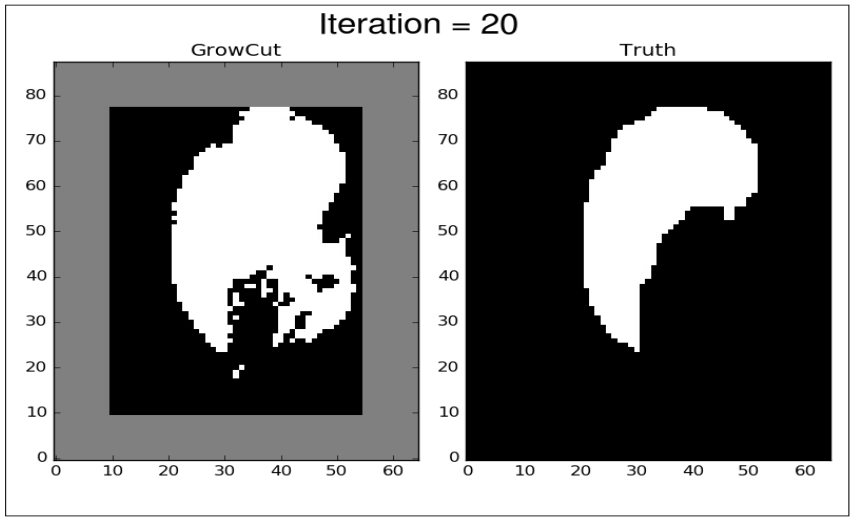
Solution in Computer Vision

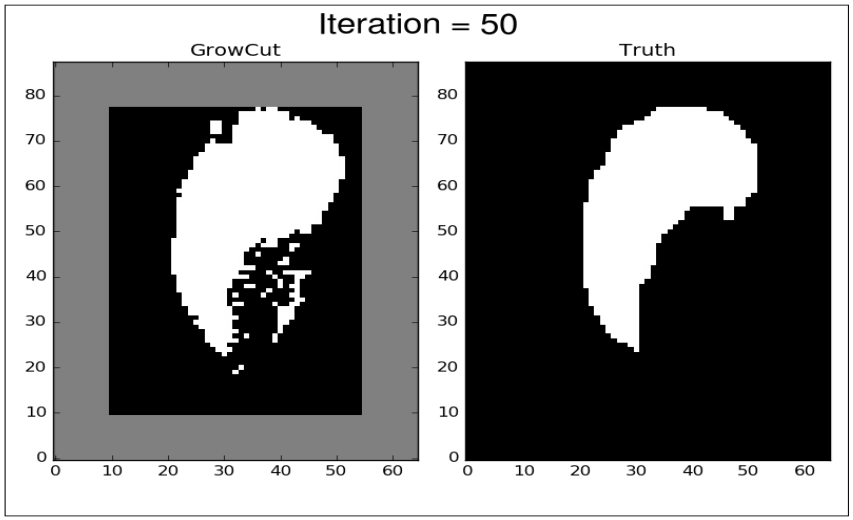
GrowCut

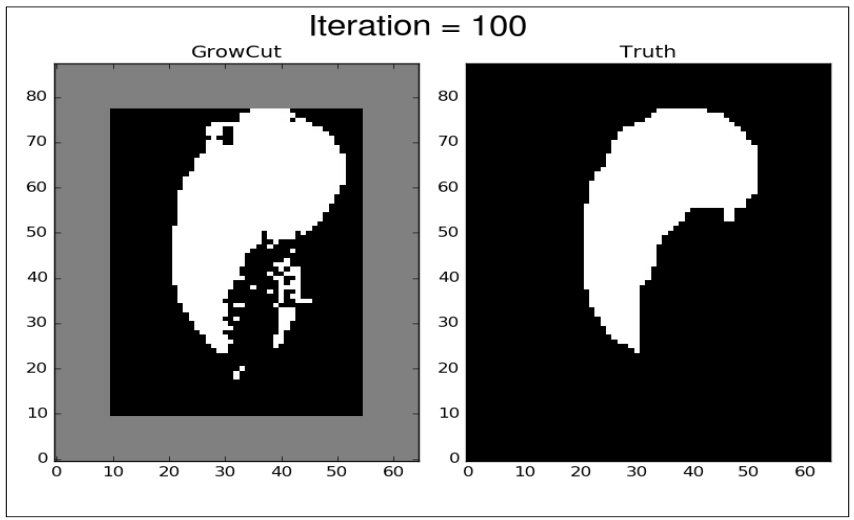
- A semi-automatic segmentation algorithm, based on "cellular automata"
- Foreground and background seeds are manually initialized
- Works in multidimensional data



Results From GrowCut



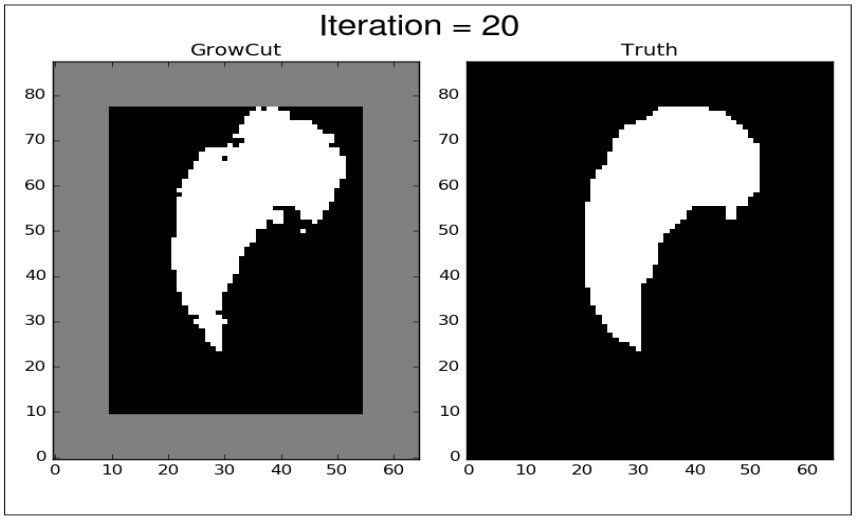


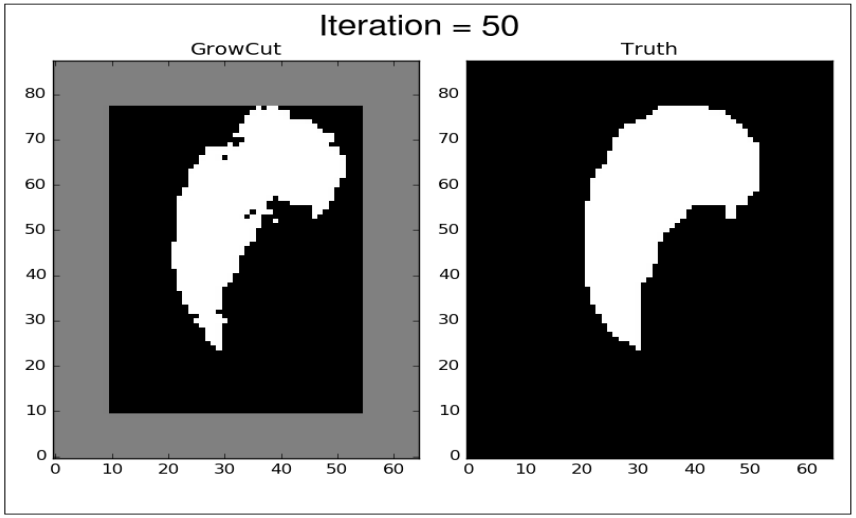


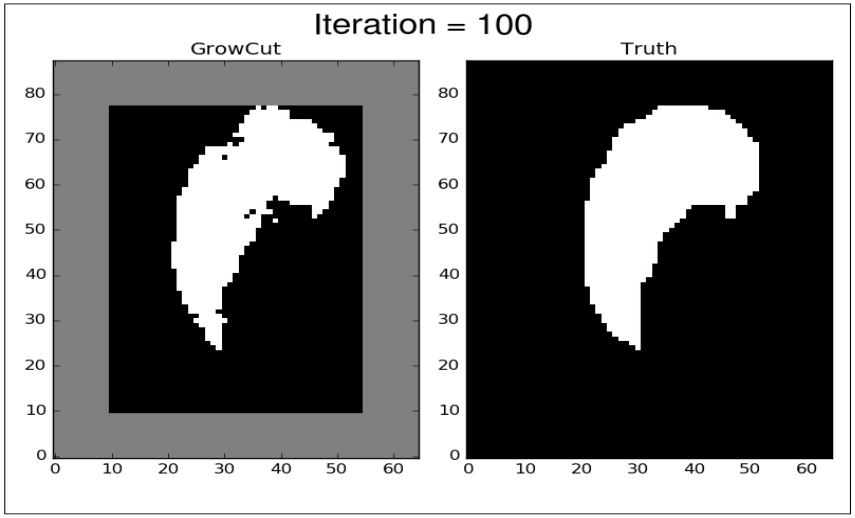
$str_{foreground} = 1$

$str_{background} = 1$

Results From GrowCut







$str_{foreground} = 1$

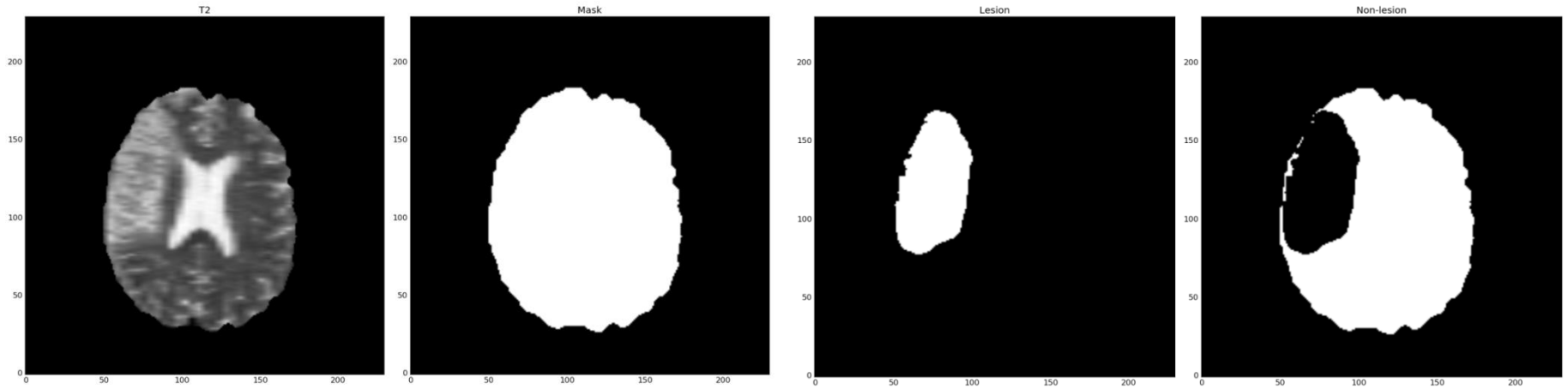
$str_{background} = 1.05$

Solution In Machine Learning

Dataset Generation:

1. Volumes of each patient are masked
2. Random samples from each patient are extracted
3. All dataset is normalized

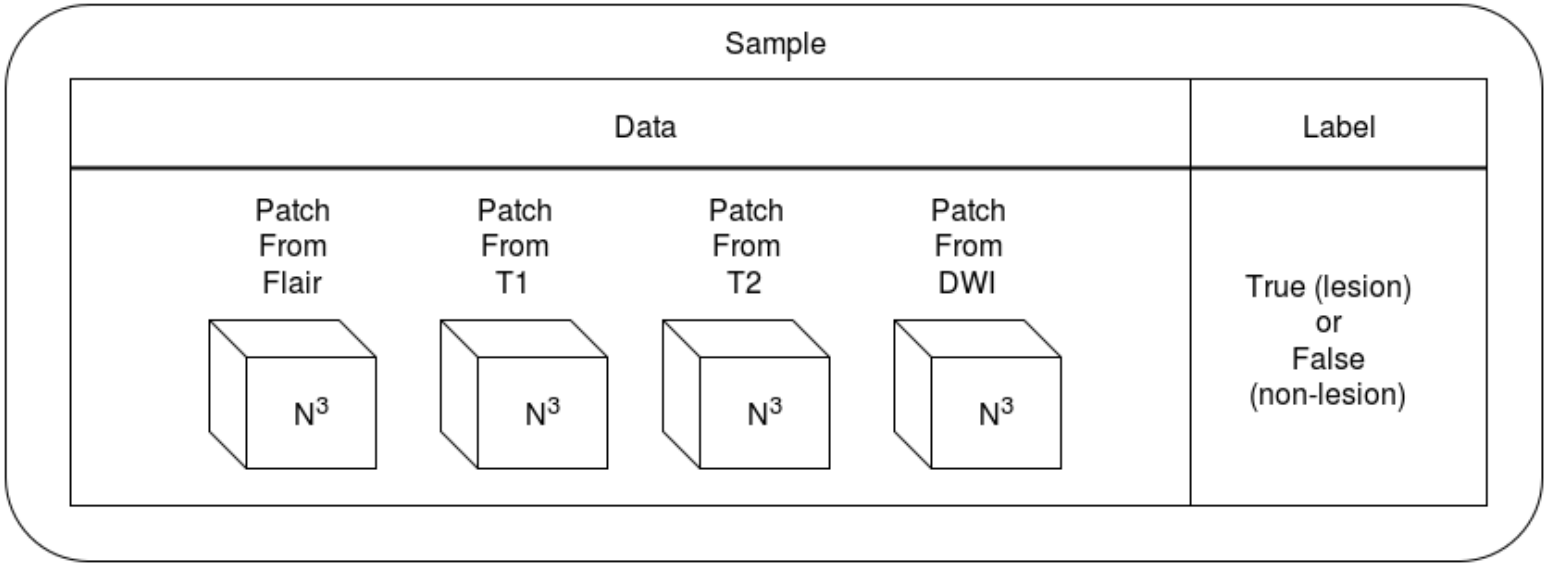
Masking:



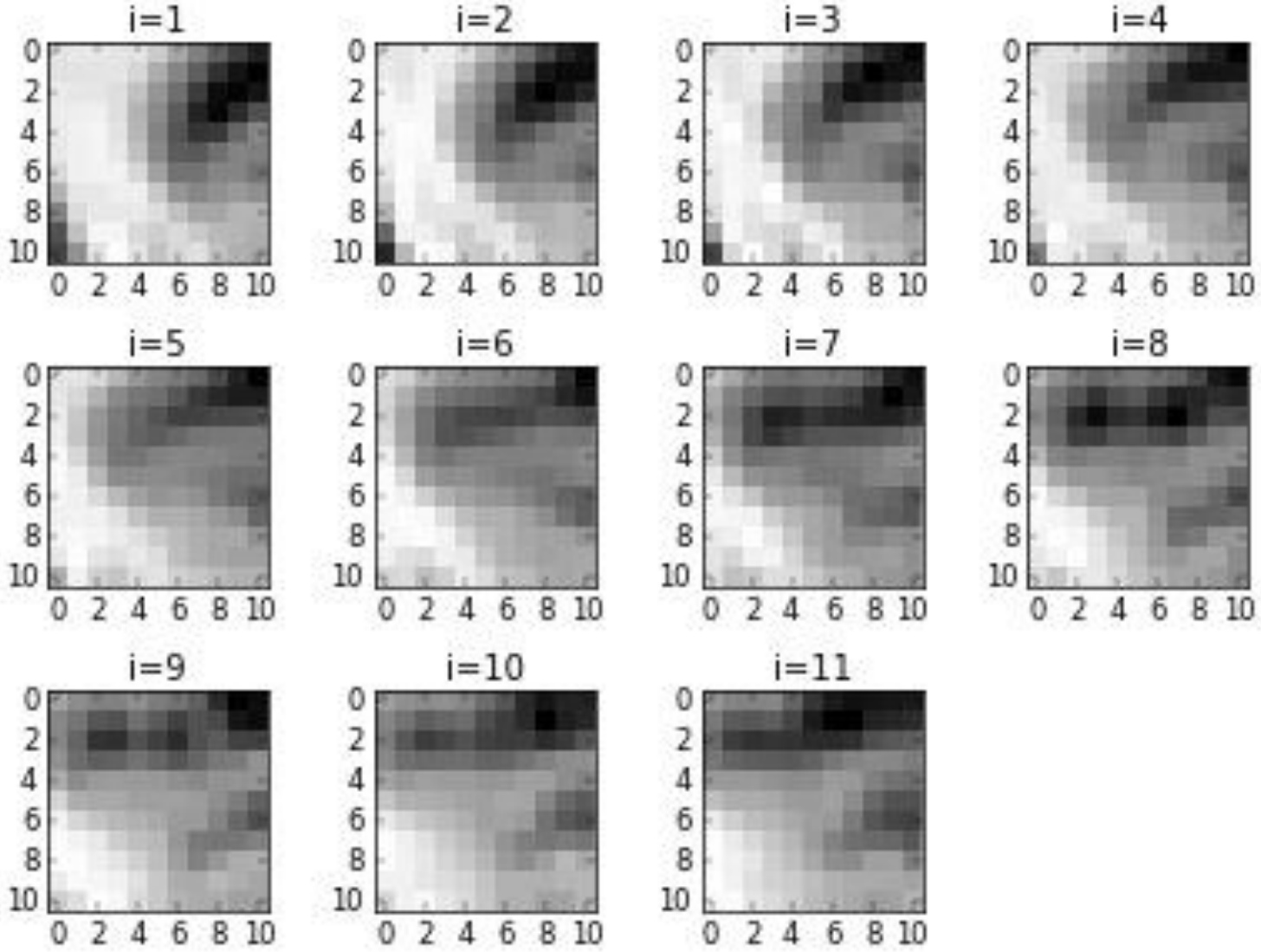
Brain Mask Lesion Non-lesion

Creating Samples:

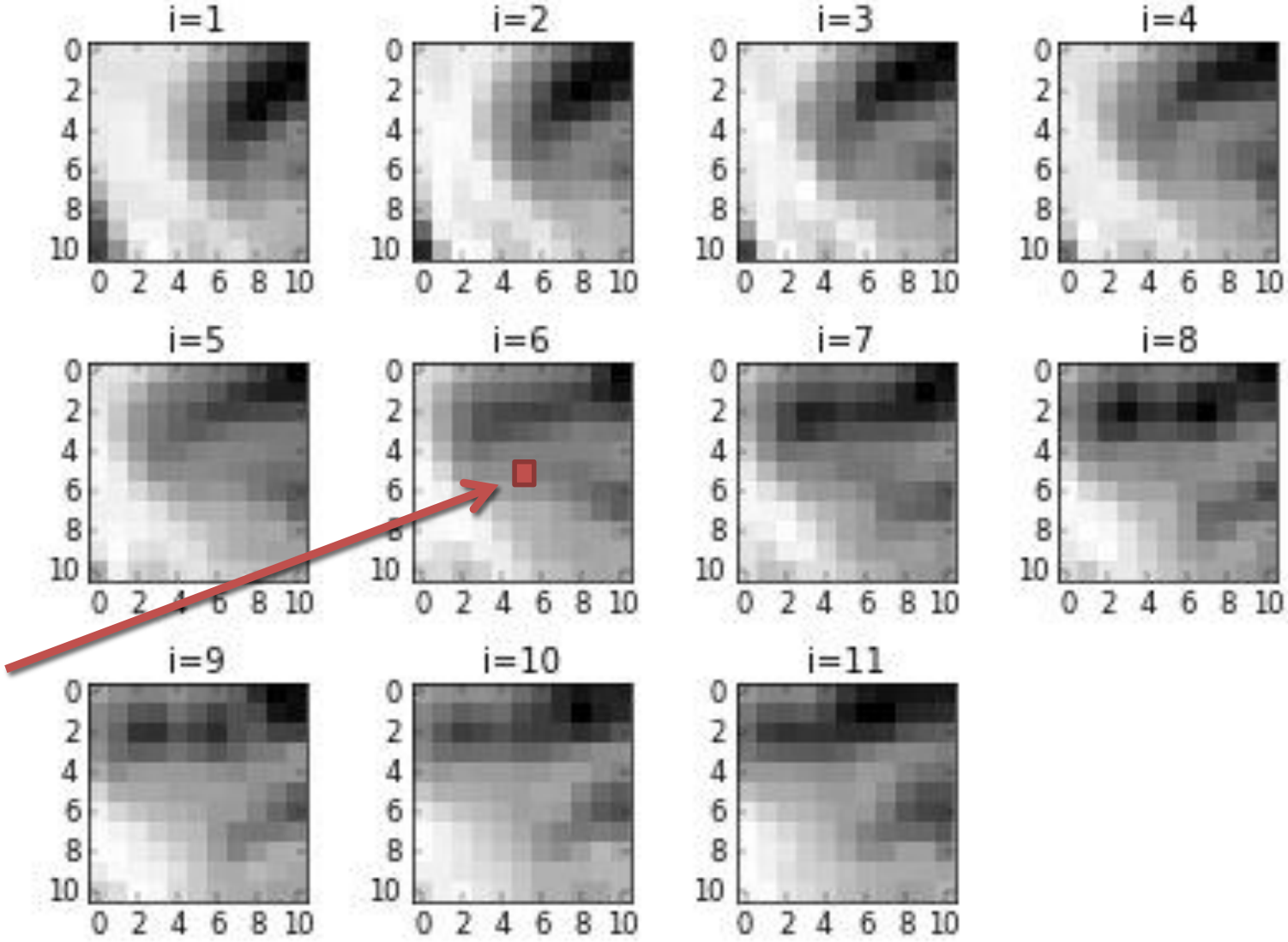
- 300 lesion and 300 non-lesion labeled samples from each patient
- Each sample contains four patches (11x11x11 cube) and a corresponding label
- 600 samples from each patient, 600*26 = 15600 samples in total



- An example patch:



- An example patch:



Normalization:

- Normalization is performed in each patch of each sample

- Mean substraction and standard deviation scaling applied to each patch separately

Normalization:

```
Before normalization
sample 0
min: 456.879303  mean: 741.500305  max: 902.608398  std: 99.483292
min: 353.000000  mean: 912.804688  max: 1414.000000  std: 235.895309
min: 510.000000  mean: 973.979736  max: 1552.000000  std: 244.382370
min: 370.397888  mean: 512.687256  max: 611.376587  std: 48.541573
sample 1
min: 499.460815  mean: 883.014954  max: 1264.290894  std: 132.276749
min: 324.000000  mean: 666.930908  max: 1084.000000  std: 152.074860
min: 822.000000  mean: 1656.366699  max: 2031.000000  std: 184.562317
min: 206.757553  mean: 421.092590  max: 703.650696  std: 110.297859
sample 2
min: 564.177551  mean: 779.269470  max: 938.853821  std: 65.423935
min: 247.000000  mean: 637.012756  max: 1117.000000  std: 173.408554
min: 828.000000  mean: 1311.475586  max: 1566.000000  std: 111.665344
min: 270.885315  mean: 394.815552  max: 523.309814  std: 46.288769
```

Normalization:

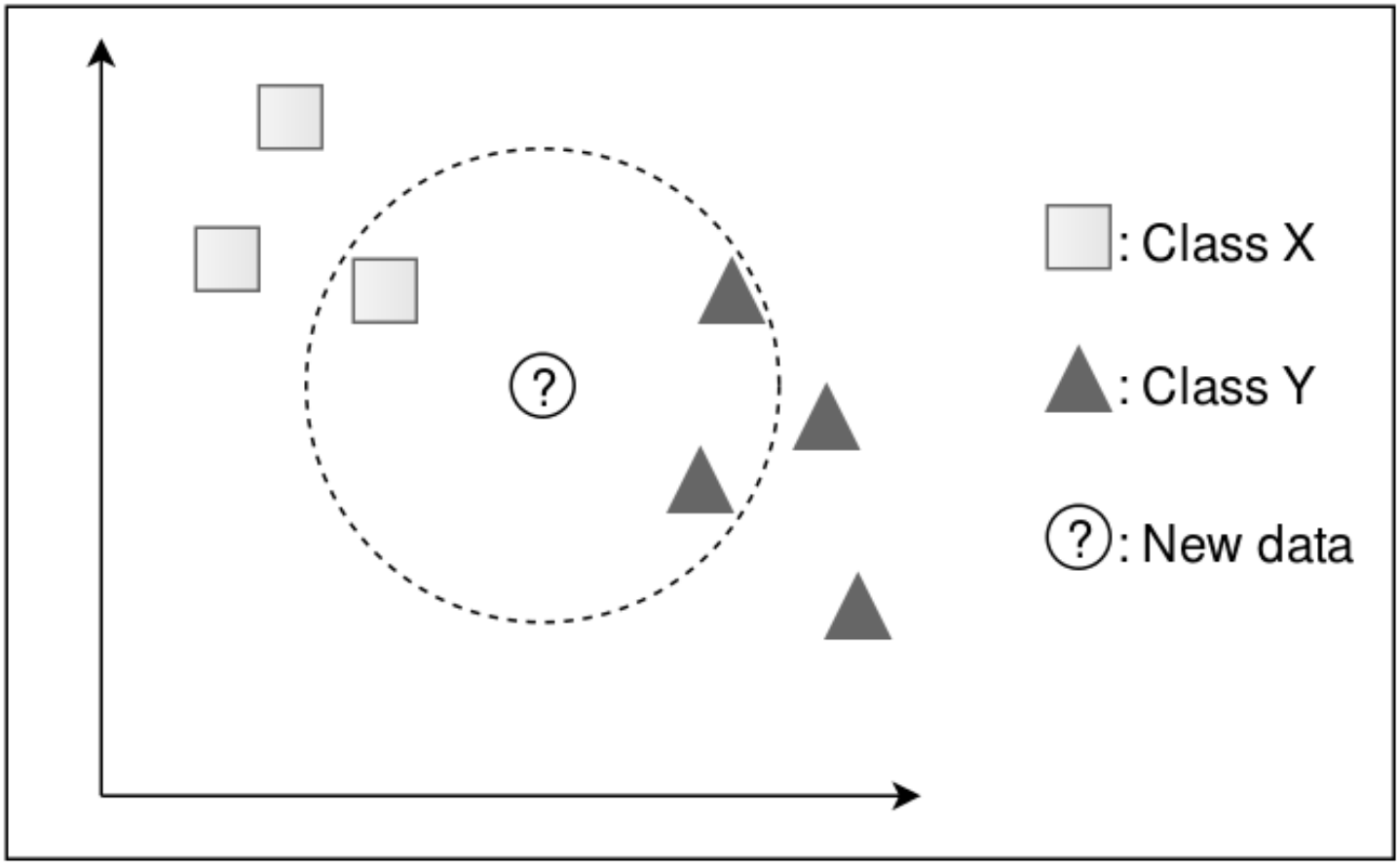
```
After normalization
sample 0
min:-2.860993  mean:-0.000001  max:1.619449  std:1.000000
min:-2.373106  mean:-0.000000  max:2.124651  std:1.000000
min:-1.898581  mean:-0.000000  max:2.365229  std:1.000000
min:-2.931289  mean:-0.000001  max:2.033089  std:1.000000
sample 1
min:-2.899634  mean:-0.000001  max:2.882411  std:1.000000
min:-2.255014  mean:-0.000000  max:2.742525  std:1.000000
min:-4.520786  mean:-0.000000  max:2.029847  std:1.000000
min:-1.943238  mean:0.000000  max:2.561773  std:1.000000
sample 2
min:-3.287664  mean:-0.000001  max:2.439235  std:1.000000
min:-2.249098  mean:0.000000  max:2.767956  std:1.000000
min:-4.329683  mean:-0.000000  max:2.279350  std:1.000000
min:-2.677328  mean:0.000000  max:2.775928  std:1.000000
```

Application Of Machine Learning Models

- Following learning based models are applied:
  - K-Nearest Neighbors
  - Linear Classifier
  - Support Vector Machine
  - Random Forests
  - 3D Convolutional Neural Networks (CNN)
- Models are evaluated with 5-fold cross validation
- Sensitivity, specificity and accuracy are used as performance metrics

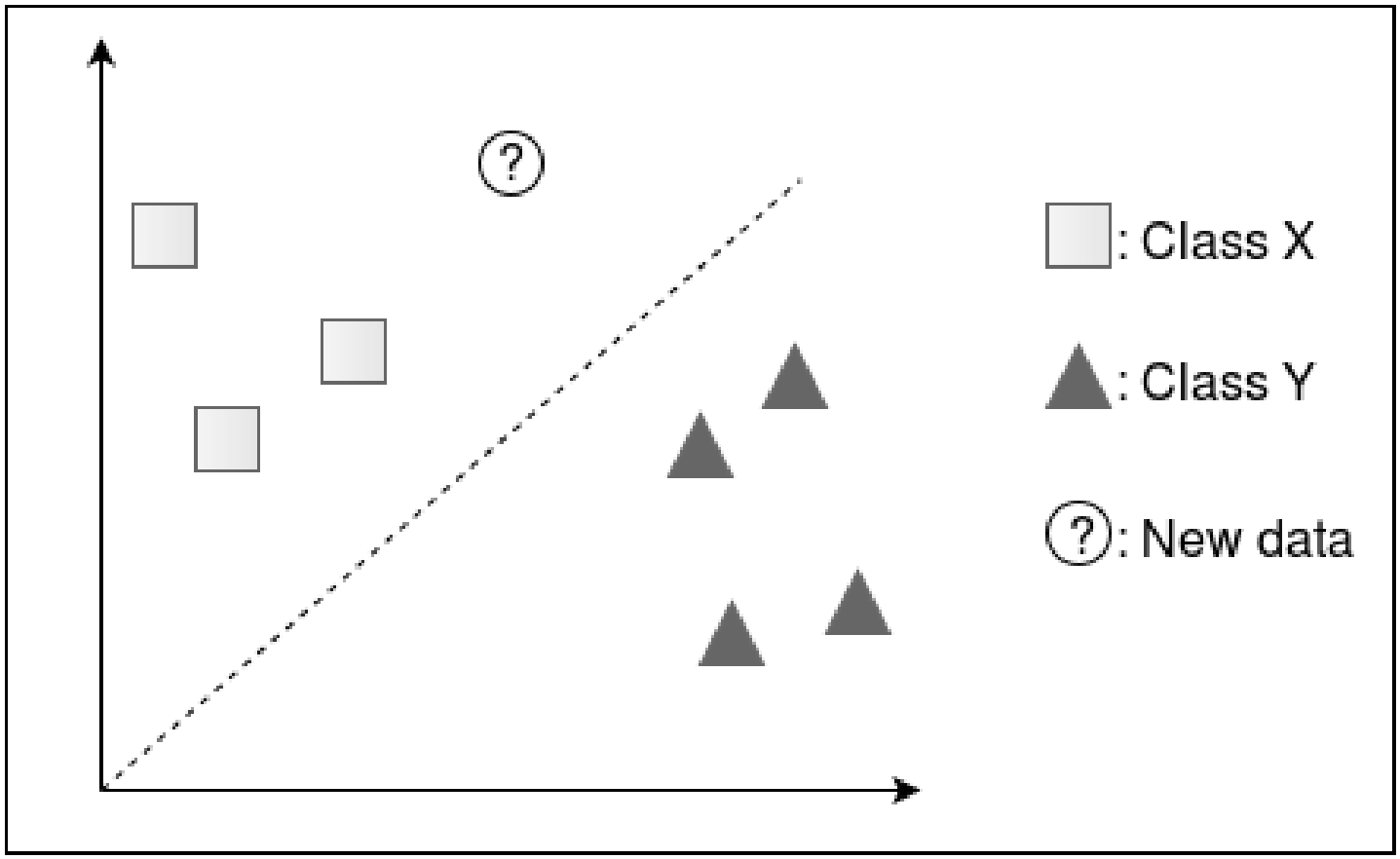
Learning Based Models

- K-Nearest Neighbors Classification



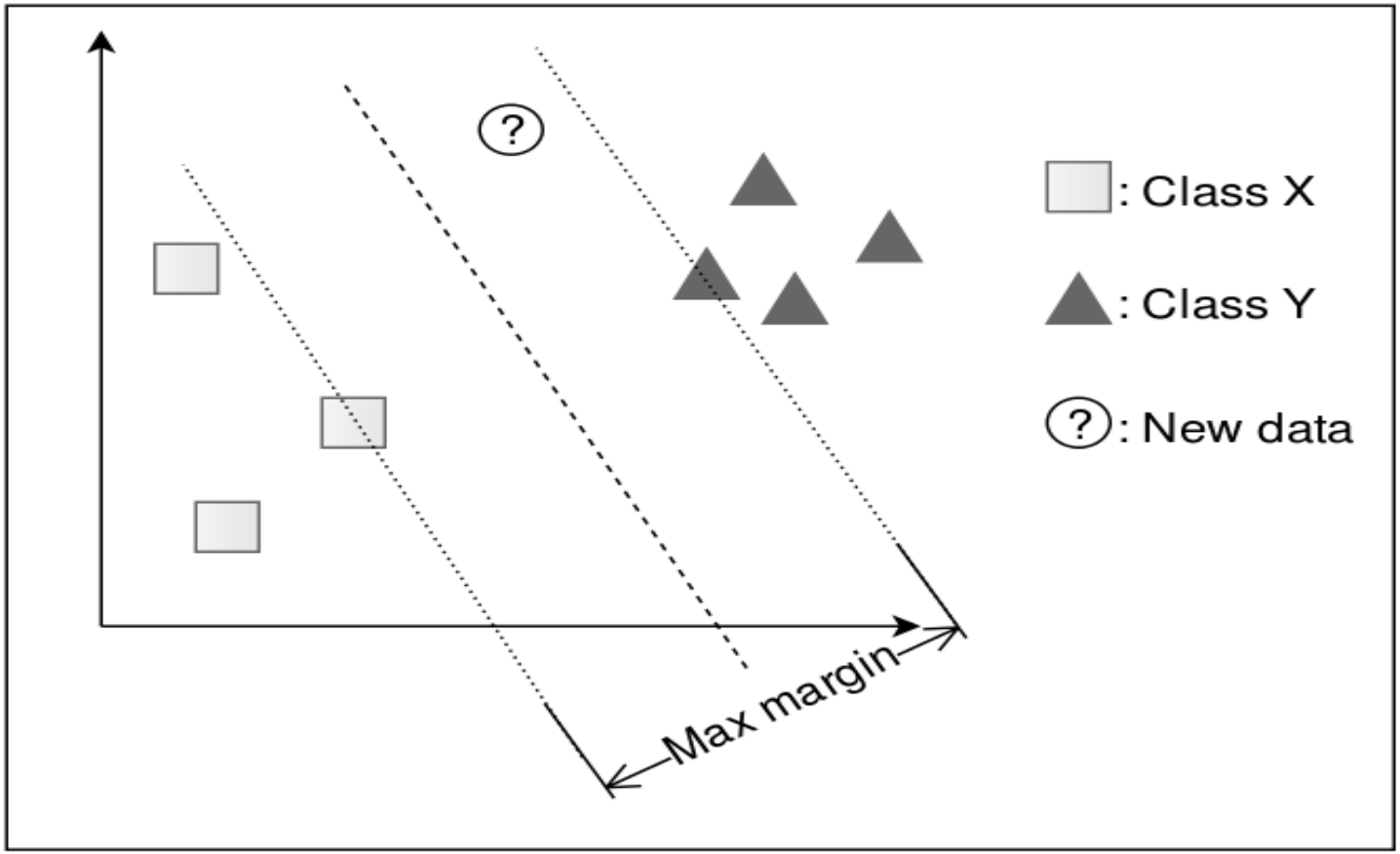
Learning Based Models

- Linear Classifier



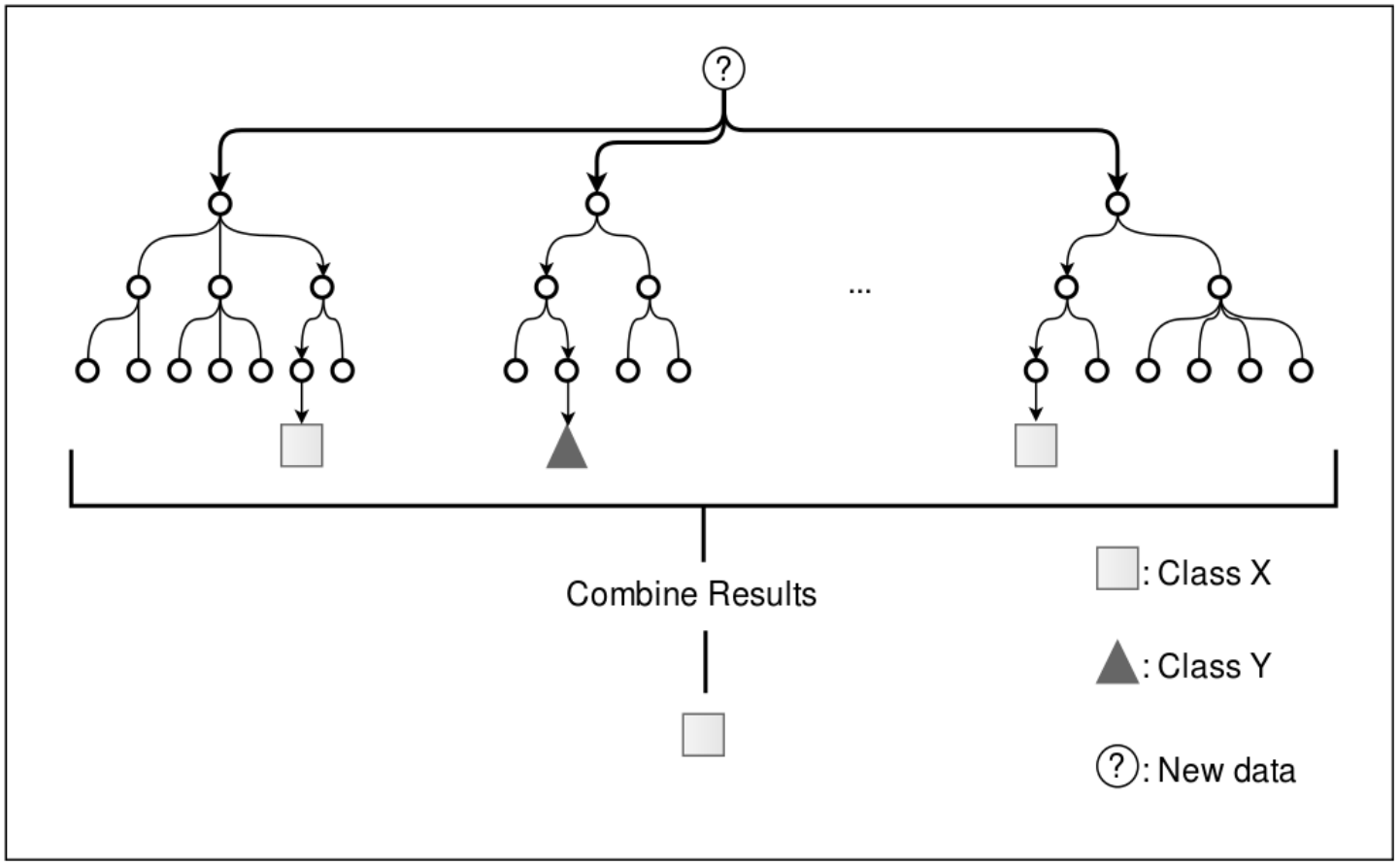
Learning Based Models

- Support Vector Machine



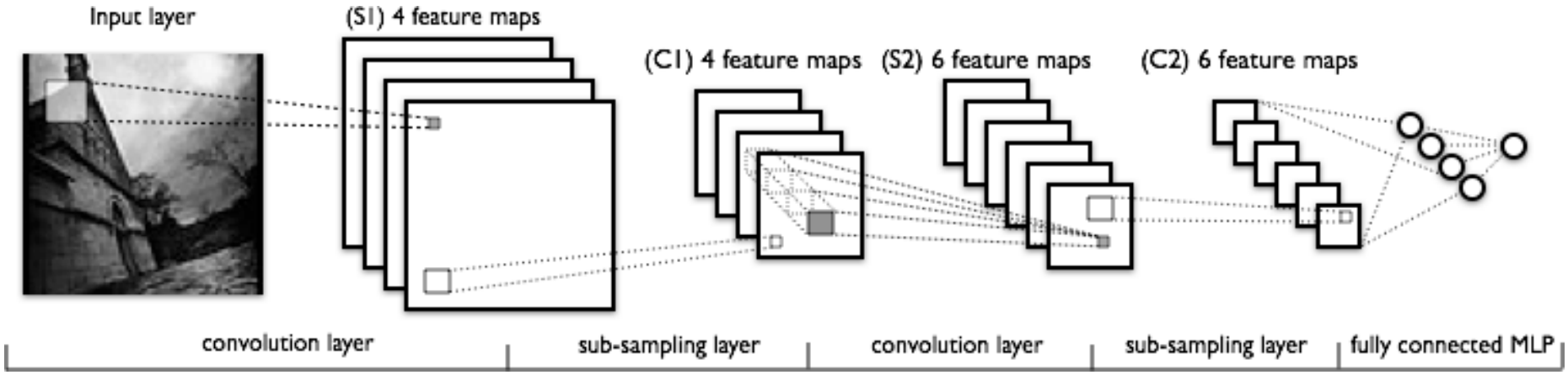
Learning Based Models

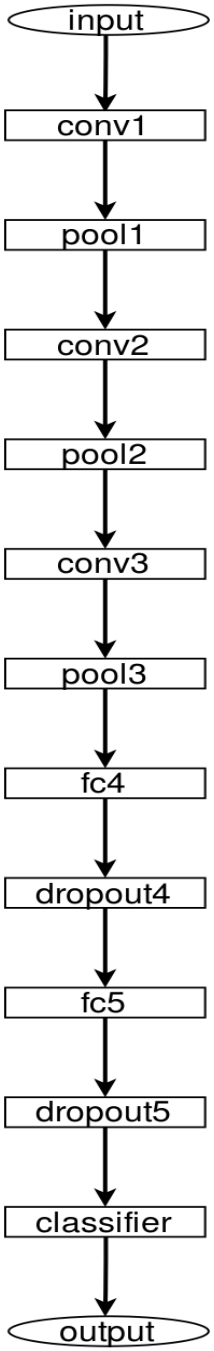
- Random Forest



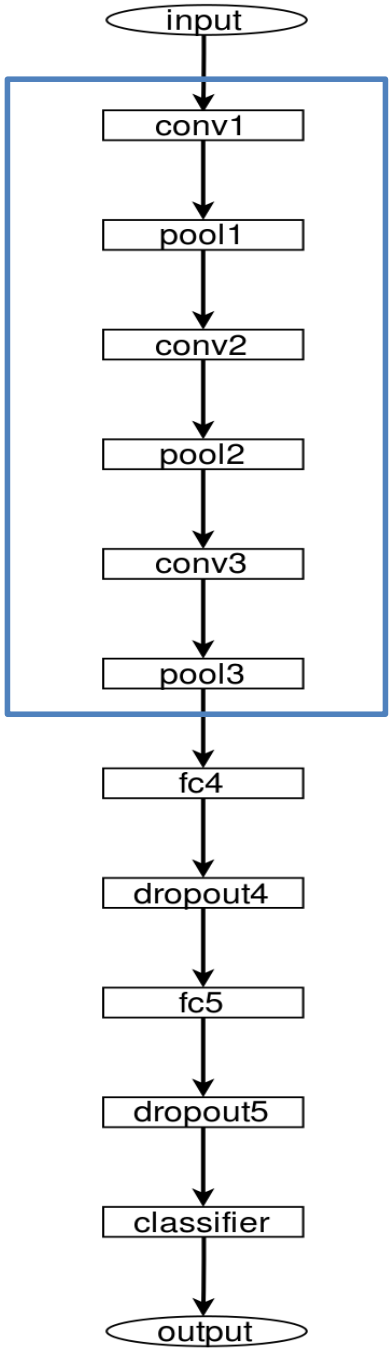
Learning Based Models

- 3D Convolutional Neural Network




input
conv1
pool1
conv2
pool2
conv3
pool3
fc4
dropout4
fc5
dropout5
classifier
output

Convolutional & pooling layers

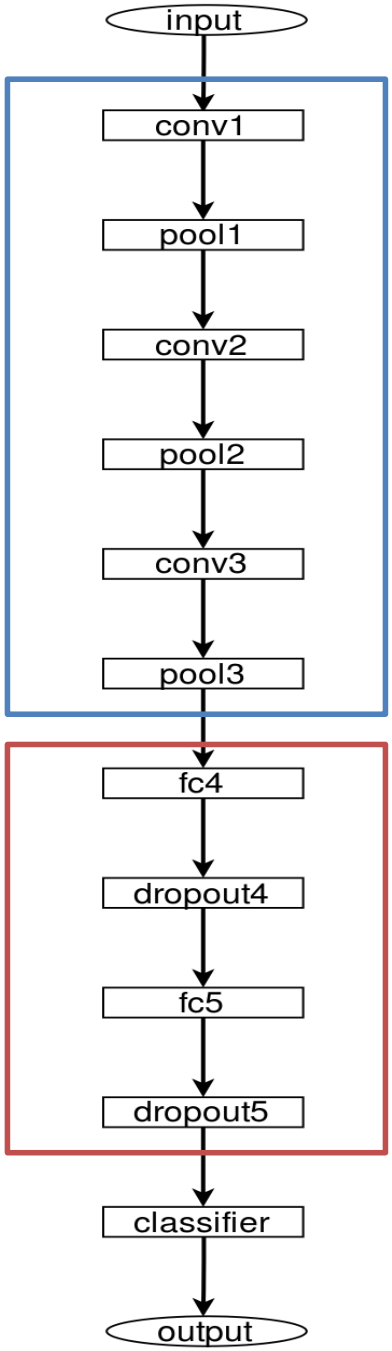


Convolution Layers
activations: leaky_relu
regularization: L2
kernel size : 3^3
feature maps : 32, 64, 128

Pooling Layers
kernel size : 2^3
strides:1

Convolutional & pooling layers

Fully connected & dropout layers



Convolution Layers
activations: leaky_relu
regularization: L2
kernel size : 3^3
feature maps : 32, 64, 128

Pooling Layers
kernel size : 2^3
strides:1

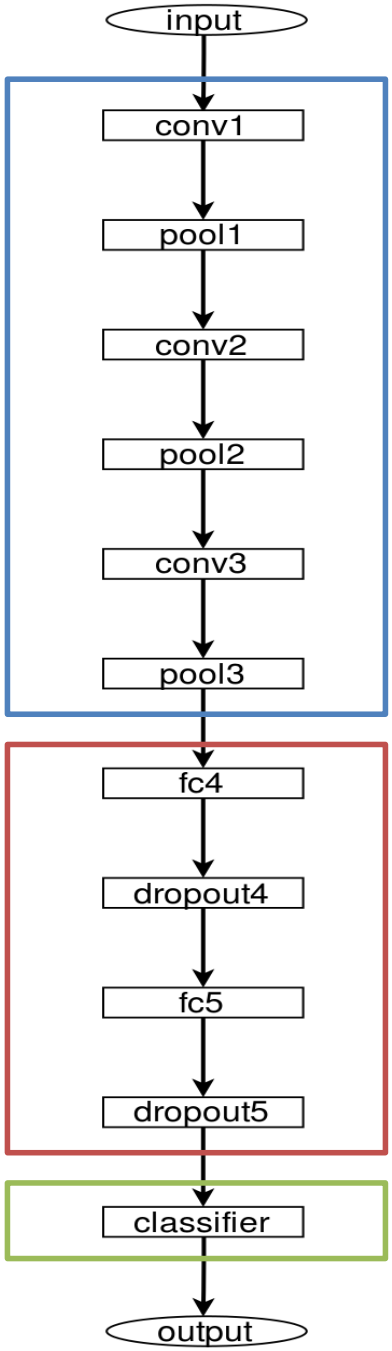
Fully Connected Layers
activations: leaky_relu
neuron counts: 256, 512

Dropout Layers
keep probability: 75%

Convolutional & pooling layers

Fully connected & dropout layers

Classifier layer



Convolution Layers
activations: leaky_relu
regularization: L2
kernel size : 3^3
feature maps : 32, 64, 128

Pooling Layers
kernel size : 2^3
strides:1

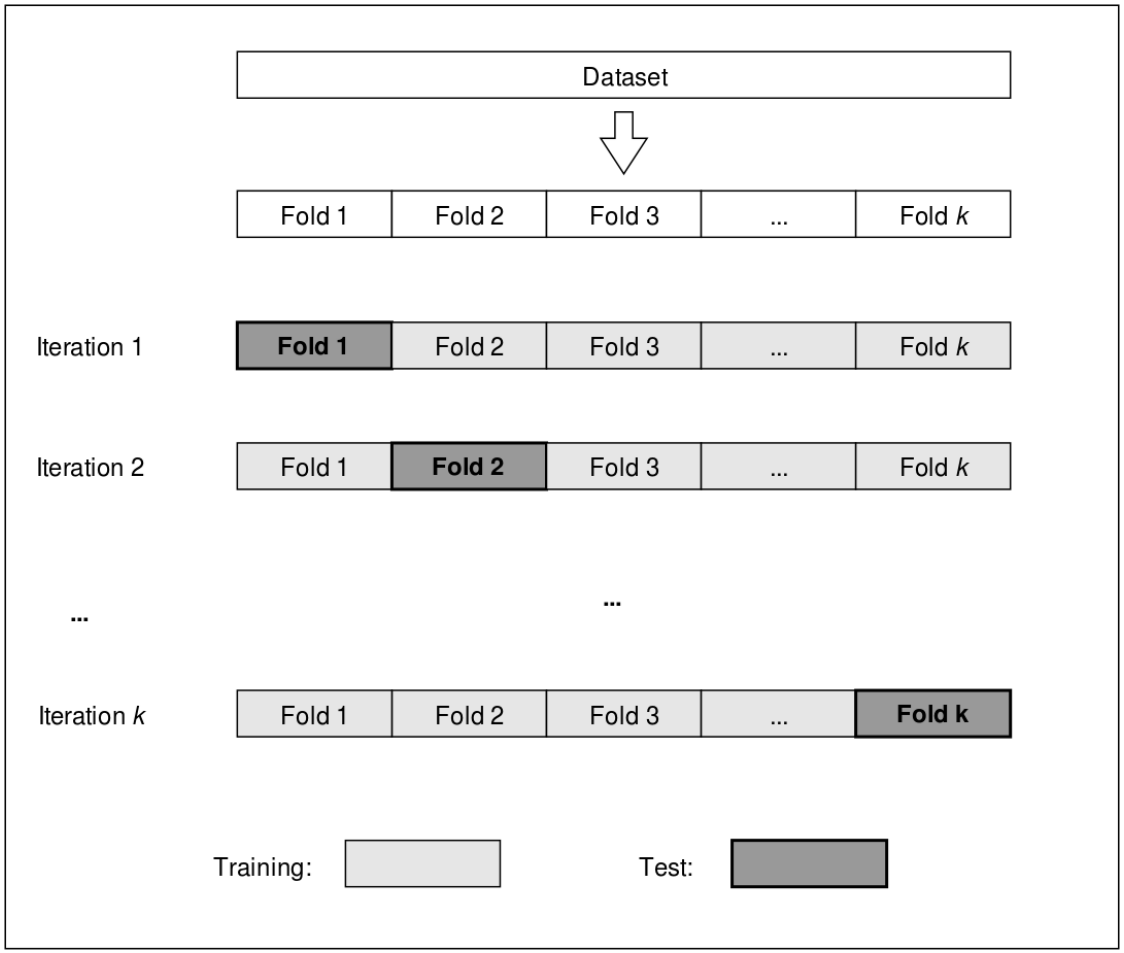
Fully Connected Layers
activations: leaky_relu
neuron counts: 256, 512

Dropout Layers
keep probability: 75%

Classifier Layer
activation: softmax
output count: 2

Model Evaluation

- K-Fold cross validation



Performance Metrics

Confusion Matrix		Actual Label	
		Positive	Negative
Predicted Label	Positive	TP	FP
	Negative	FN	TN

Performance Metrics

Confusion Matrix		Actual Label	
		Positive	Negative
Predicted Label	Positive	TP	FP
	Negative	FN	TN

$$Sensitivity = \frac{TP}{TP+FN}$$

Performance Metrics

Confusion Matrix		Actual Label	
		Positive	Negative
Predicted Label	Positive	TP	FP
	Negative	FN	TN

$$Sensitivity = \frac{TP}{TP+FN}$$

$$Specificity = \frac{TN}{TN+FP}$$

Performance Metrics

Confusion Matrix		Actual Label: Positive	Actual Label: Negative
Predicted Label	Positive	TP	FP
	Negative	FN	TN

$$Sensitivity = \frac{TP}{TP+FN}$$

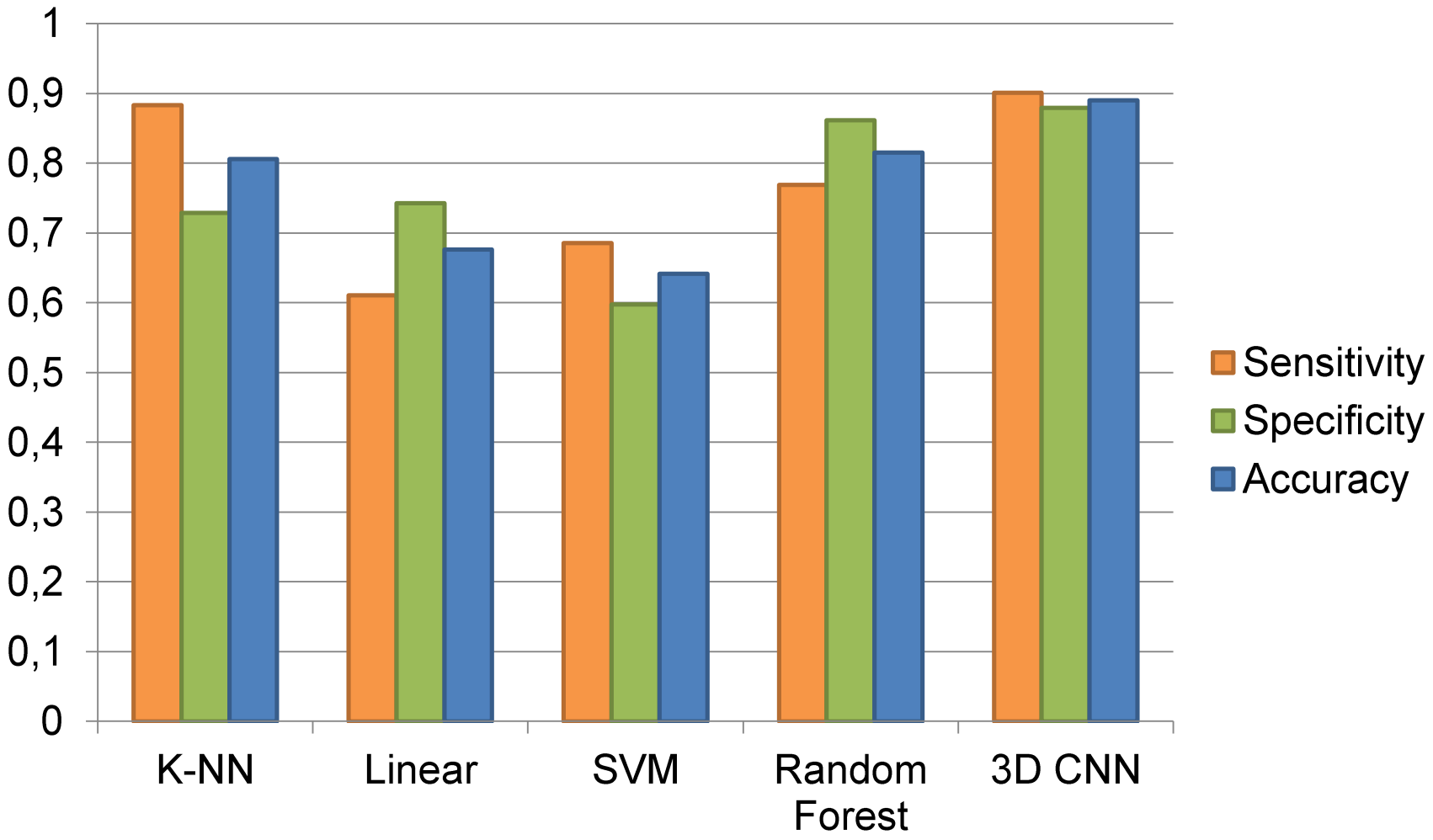
$$Specificity = \frac{TN}{TN+FP}$$

$$Accuracy = \frac{TN+TP}{TN+TP+FN+FP}$$

Results From Learning Based Models

Classifier	Confusion Matrix		Sensitivity	Specificity	Accuracy
K-Nearest Neighbors	6891	2116	0,8835	0,7287	0,8061
	909	5684			
Linear Classifier	4760	2006	0,6103	0,7428	0,6765
	3040	5794			
Support Vector Machine	5344	3143	0,6851	0,5971	0,6411
	2456	4657			
Random Forest	5994	1078	0,7685	0,8618	0,8151
	1806	6722			
3D Convolutional Neural Network	7029	942	0,9012	0,8792	0,8902
	771	6858			

Results From Learning Based Models



Conclusion

Problem is adressed as a:

Conclusion

Problem is adressed as a:

- Computer vision segmentation problem
  - GrowCut

Conclusion

Problem is adressed as a:

- Computer vision segmentation problem
  - GrowCut
- Machine learning classification problem
  - K-Nearest Neighbors
  - Linear Classifier
  - Support Vector Machine
  - Random Forests
  - 3D Convolutional Neural Networks (CNN)

Conclusion

Problem is adressed as a:

- Computer vision segmentation problem
  - GrowCut
- Machine learning classification problem
  - K-Nearest Neighbors
  - Linear Classifier
  - Support Vector Machine
  - Random Forests
  - 3D Convolutional Neural Networks (CNN)

Evaluation:

- Visual examination

Conclusion

Problem is adressed as a:

- Computer vision segmentation problem
  - GrowCut
- Machine learning classification problem
  - K-Nearest Neighbors
  - Linear Classifier
  - Support Vector Machine
  - Random Forests
  - 3D Convolutional Neural Networks (CNN)

Evaluation:

- Visual examination
- Sensitivity, specificity, accuracy metrics

Further Development

- Improvements in methods
- Advanced visualization
- Opinions from medical experts

Thank you for listening