

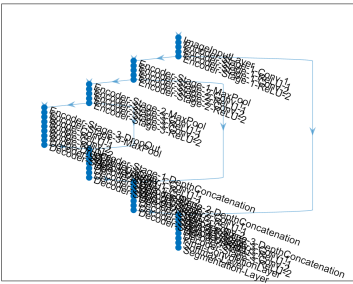
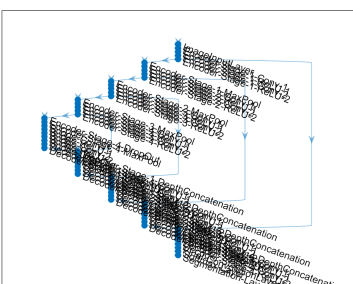
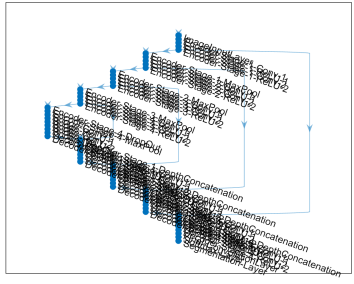
Team SPECT

Mei Li Luisa Cham Perez A01139386

Ana Lucía Soria Cardona A00827570

Graciela Alejandra Rincón López A00826575

Creating a U-net Network

Case 1	Case 2	Case 3
Image size: 480x640x3 Classes: 5 Encoder Depth:3	Image size: 480x60x3 Classes: 3 Encoder Depth: 4	Image size: 480x60x3 Classes: 5 Encoder Depth:3
		

Training Data

<div>Case 1</div> <div>Learning Rate: 1e-2</div> <div>Max Epochs: 30</div>	<div>Layers: [58x1 nnet.cnn.layer.Layer]</div> <div>Connections: [61x2 table]</div> <div>InputNames: {'ImageInputLayer'}</div> <div>OutputNames: {'Segmentation-Layer'}</div> <div>Training on single CPU.</div> <div>Initializing input data normalization.</div> <table><thead><tr><th>Epoch</th><th>Iteration</th><th>Time Elapsed (hh:mm:ss)</th><th>Mini-batch Accuracy</th><th>Mini-batch Loss</th><th>Base Learning Rate</th></tr></thead><tbody><tr><td>1</td><td>1</td><td>00:00:03</td><td>75.57%</td><td>2.4341</td><td>0.0100</td></tr><tr><td>6</td><td>6</td><td>00:00:18</td><td>86.77%</td><td>NaN</td><td>0.0100</td></tr></tbody></table> <div>Training finished: Training loss is NaN.</div> <div>Warning: Training stopped at iteration 6 because training loss is NaN. Predictions using the output network might contain NaN values.</div>	Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate	1	1	00:00:03	75.57%	2.4341	0.0100	6	6	00:00:18	86.77%	NaN	0.0100												
Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate																										
1	1	00:00:03	75.57%	2.4341	0.0100																										
6	6	00:00:18	86.77%	NaN	0.0100																										
<div>Case 2</div> <div>Learning Rate: 1e-5</div> <div>Max Epochs: 30</div>	<div>OutputNames: { Segmentation-Layer }</div> <div>Training on single CPU.</div> <div>Initializing input data normalization.</div> <table><thead><tr><th>Epoch</th><th>Iteration</th><th>Time Elapsed (hh:mm:ss)</th><th>Mini-batch Accuracy</th><th>Mini-batch Loss</th><th>Base Learning Rate</th></tr></thead><tbody><tr><td>1</td><td>1</td><td>00:00:02</td><td>78.79%</td><td>1.5972</td><td>1.0000e-05</td></tr><tr><td>10</td><td>10</td><td>00:00:27</td><td>82.45%</td><td>1.3881</td><td>1.0000e-05</td></tr><tr><td>20</td><td>20</td><td>00:00:55</td><td>87.99%</td><td>1.0551</td><td>1.0000e-05</td></tr><tr><td>30</td><td>30</td><td>00:01:23</td><td>91.65%</td><td>0.8483</td><td>1.0000e-05</td></tr></tbody></table> <div>Training finished: Max epochs completed.</div>	Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate	1	1	00:00:02	78.79%	1.5972	1.0000e-05	10	10	00:00:27	82.45%	1.3881	1.0000e-05	20	20	00:00:55	87.99%	1.0551	1.0000e-05	30	30	00:01:23	91.65%	0.8483	1.0000e-05
Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate																										
1	1	00:00:02	78.79%	1.5972	1.0000e-05																										
10	10	00:00:27	82.45%	1.3881	1.0000e-05																										
20	20	00:00:55	87.99%	1.0551	1.0000e-05																										
30	30	00:01:23	91.65%	0.8483	1.0000e-05																										

Case 3

Learning Rate: 1e-5

Max Epochs: 50

Training on single CPU.

Initializing input data normalization.

Epoch	Iteration	Time Elapsed (hh:mm:ss)	Mini-batch Accuracy	Mini-batch Loss	Base Learning Rate
1	1	00:00:02	92.21%	1.0396	1.0000e-05
10	10	00:00:27	86.42%	1.6232	1.0000e-05
20	20	00:00:53	91.81%	1.0619	1.0000e-05
30	30	00:01:20	94.87%	0.7159	1.0000e-05
40	40	00:01:46	95.16%	0.6701	1.0000e-05
50	50	00:02:12	95.52%	0.6206	1.0000e-05

Training finished: Max epochs completed.

What is the most appropriate number of max epochs you can use? Why?

From the three cases done, the most appropriate number of epochs used were 50 epochs, and a learning rate of 1e-5. This number of epochs allowed the program to continue optimizing the accuracy of the mini-batch.

How did the learning rate affect the accuracy?

The learning rate was the first parameter to be edited. In case 1 the rate was 1e-2, and for the second case it was decreased to 1e-5. In the first case the training was stopped at the 6th iteration because of NaN values, while in the second case the training was completed to the maximum indicated epochs and an accuracy of 92% was achieved.

Thus it can be concluded that a decrease in the learning rate allowed for better results.

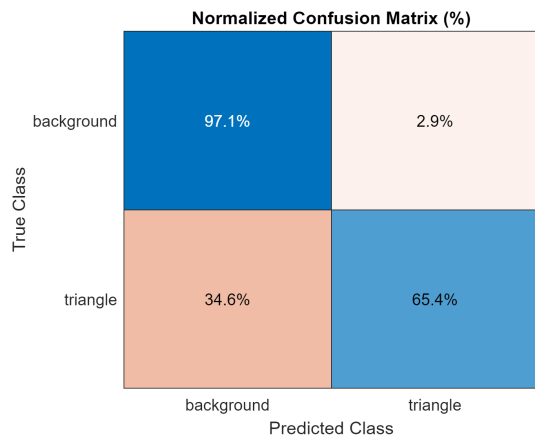
2nd part

Testing trained U-Net

Evaluating semantic segmentation results

- * Selected metrics: global accuracy, class accuracy, IoU, weighted IoU, BF score.
- * Processed 0 images.
- * Processed 29 images.
- * Processed 81 images.
- * Processed 100 images.
- * Finalizing... Done.
- * Data set metrics:

GlobalAccuracy	MeanAccuracy	MeanIoU	WeightedIoU	MeanBFScore
0.95618	0.81227	0.68139	0.92956	0.60433



ans =

2×3 [table](#)

	Accuracy	IoU	MeanBFScore
triangle	0.6537	0.40797	0.41135
background	0.97083	0.95482	0.7973

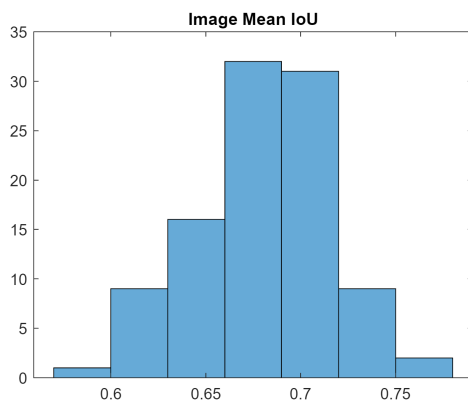
ans =

ans =

2×2 [table](#)

	triangle	background
triangle	3092	1638
background	2849	94821

>>



Include your histogram in the report and answer: what was the most common mean IoU through the images?

Between 0.66 and 0.69

Lowest IoU

st Image vs. Truth vs. Prediction. IoU = 0.59



Best IoU

Test Image vs. Truth vs. Prediction. IoU = 0.75495

