Imperial College London

ACSE 9 - Presentation
September 10, 2019

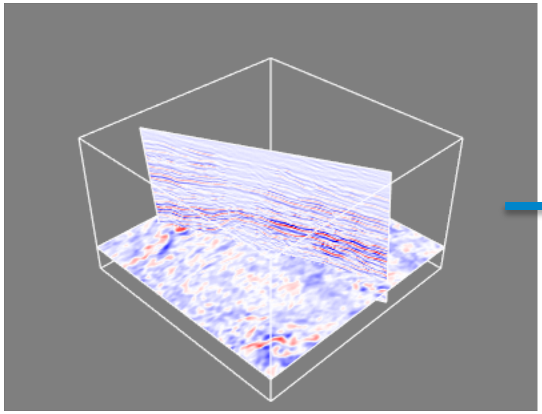
Automatic Seismic Interpretation Techniques

By Tayfun Karaderi

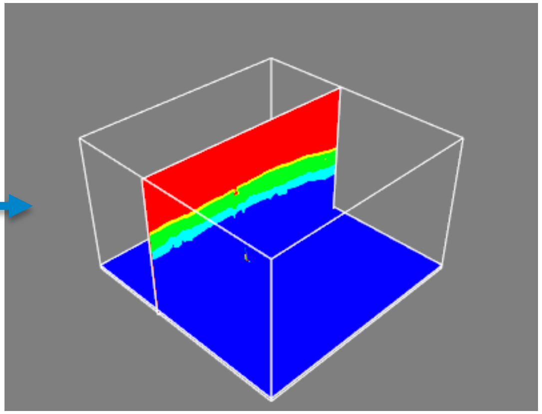
This project was performed as part of a summer internship at the Ikon Science Ltd in 2019.

Supervisors:
Dr. Ehsan Naeini, Prof. Olivier Dubrule, Lukas Mosser

Semantic Segmentation



Original Seismic Amplitudes Cube

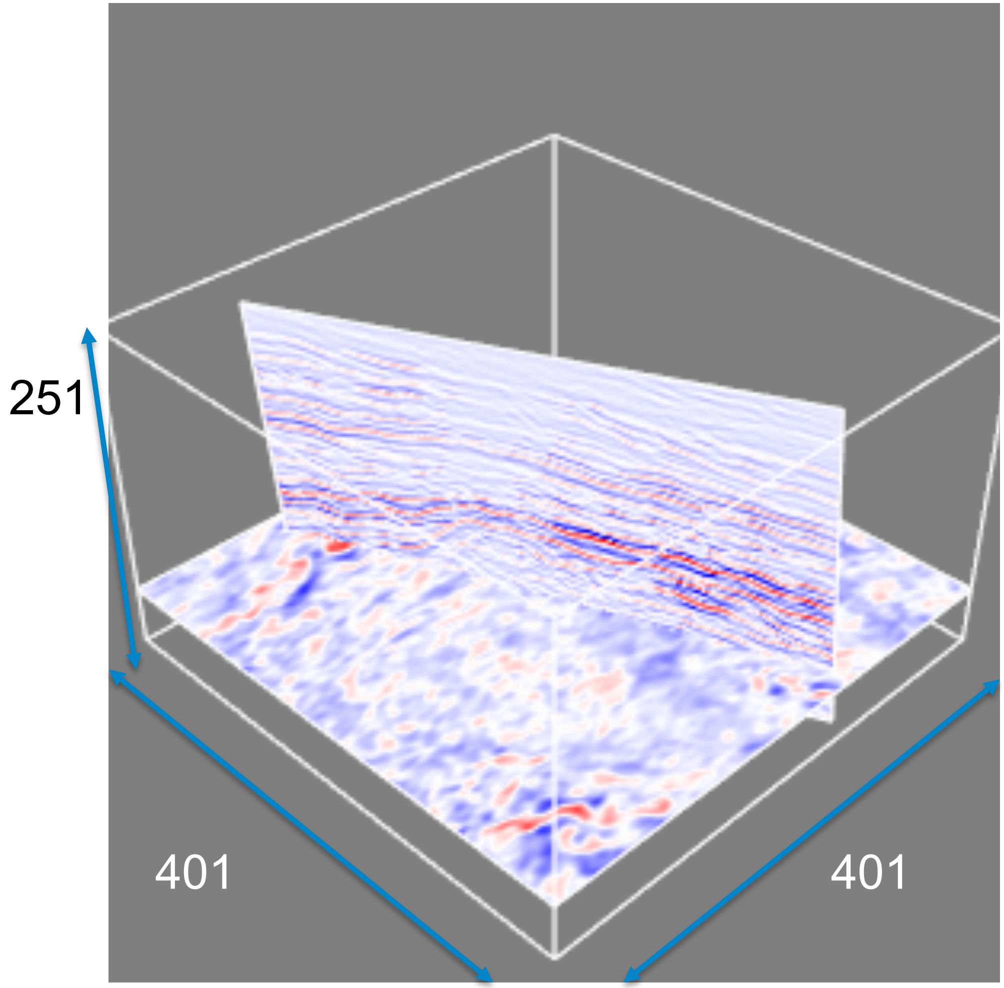


Segmentation Map

- This is a computer vision task where the aim is for each pixel to predict which category the pixel belongs to.
- We explore application of various semantic segmentation techniques to seismic datasets.

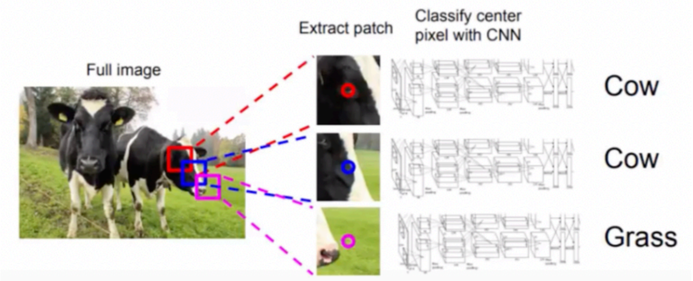
DataSet

- We make use of the seismic amplitudes cube obtained from the Forties oil field in North Sea.
- Dataset contains 401 inline slices, 401 crossline slices, 251 vertical slices.
- Contains five different classes corresponding to five different geological units of Overburden, Sele, Forties, Lista, and Underburden.

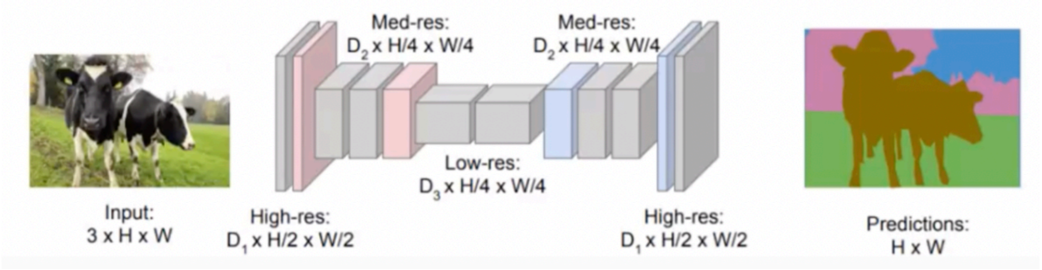


Semantic Segmentation

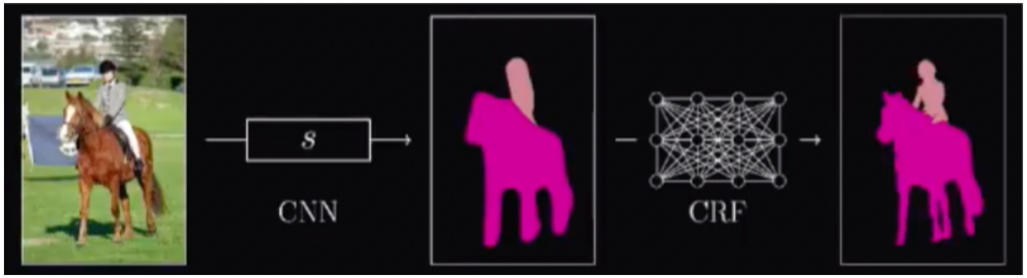
Sliding Window Classifier (SWC)



Fully Convolutional Networks (FCN)

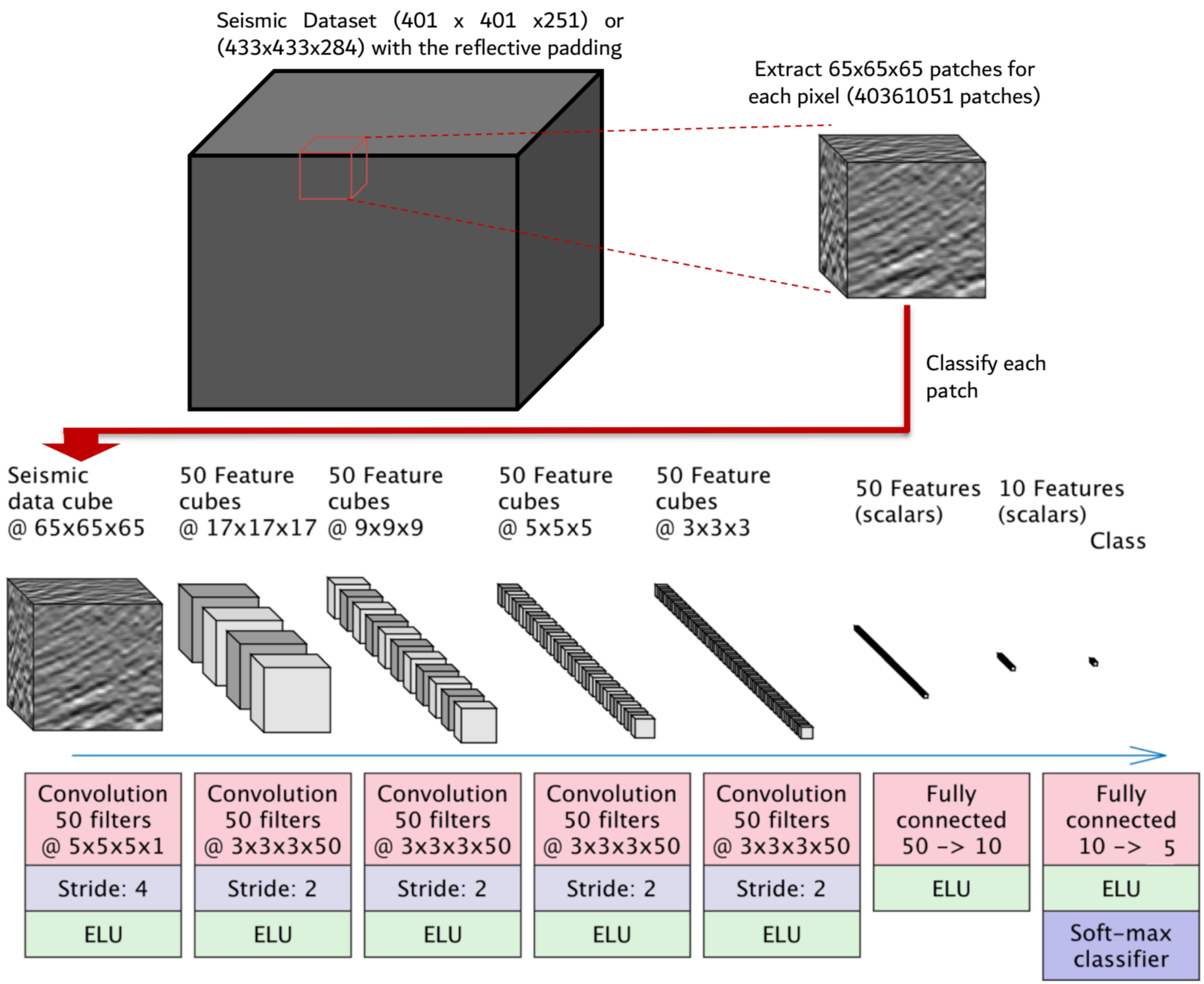


Conditional Random Fields (CRFs)



Sliding Window Classifier (SWC)

- We use a sliding window classifier approach that extracts 3D (65x65x65) patches around each pixel in the training set.
- Reflective padding (of size 32) is applied to the original amplitudes cube in order to be able to extract patches around every pixel.
- Training details–training dataset: 8 inline slices, optimizer: Adams, Batch size: 32, Epochs: 10, loss: cross-entropy. 50% of training and validation data is used per epoch and we shuffle the training/validation set at the end of each epoch.


Seismic Dataset (401 x 401 x251) or (433x433x284) with the reflective padding
Extract 65x65x65 patches for each pixel (40361051 patches)
Classify each patch
Seismic data cube @ 65x65x65
50 Feature cubes @ 17x17x17
50 Feature cubes @ 9x9x9
50 Feature cubes @ 5x5x5
50 Feature cubes @ 3x3x3
50 Features (scalars)
10 Features (scalars)
Class
Convolution 50 filters @ 5x5x5x1
Stride: 4
ELU
Convolution 50 filters @ 3x3x3x50
Stride: 2
ELU
Convolution 50 filters @ 3x3x3x50
Stride: 2
ELU
Convolution 50 filters @ 3x3x3x50
Stride: 2
ELU
Convolution 50 filters @ 3x3x3x50
Stride: 2
ELU
Fully connected 50 -> 10
ELU
Fully connected 10 -> 5
ELU
Soft-max classifier

Sliding Window Classifier (SWC)

	Size of input image n	Number of input channels	f	p	s	Size of output image (n+2p-f)/s+1	Number of output channels or filters	Number of output neurons	Size of Filter + 1	Number of Parameters
Conv1	65	1	5	2	4	17	50	245650	126	6300
Conv2	17	50	3	1	2	9	50	36450	1351	67550
Conv3	9	50	3	1	2	5	50	6250	1351	67550
Conv4	5	50	3	1	2	3	50	1350	1351	67550
Conv5	3	50	3	1	2	2	50	400	1351	67550
	Size of input							Number of output neurons		
FC1	400							50	401	20050
FC2	50							10	51	510
Softmax	10							5	11	55
							Total Neurons	290165	Total Parameters	297115

Table: The CNN model used as the classifier of the SWC approach is shown in the table. Batch normalization was used between each convolutional layer as well as a dropout rate of 0.2.

Imperial College London

ACSE 9 - Presentation
September 10, 2019

Sliding Window Classifier (SWC) Results

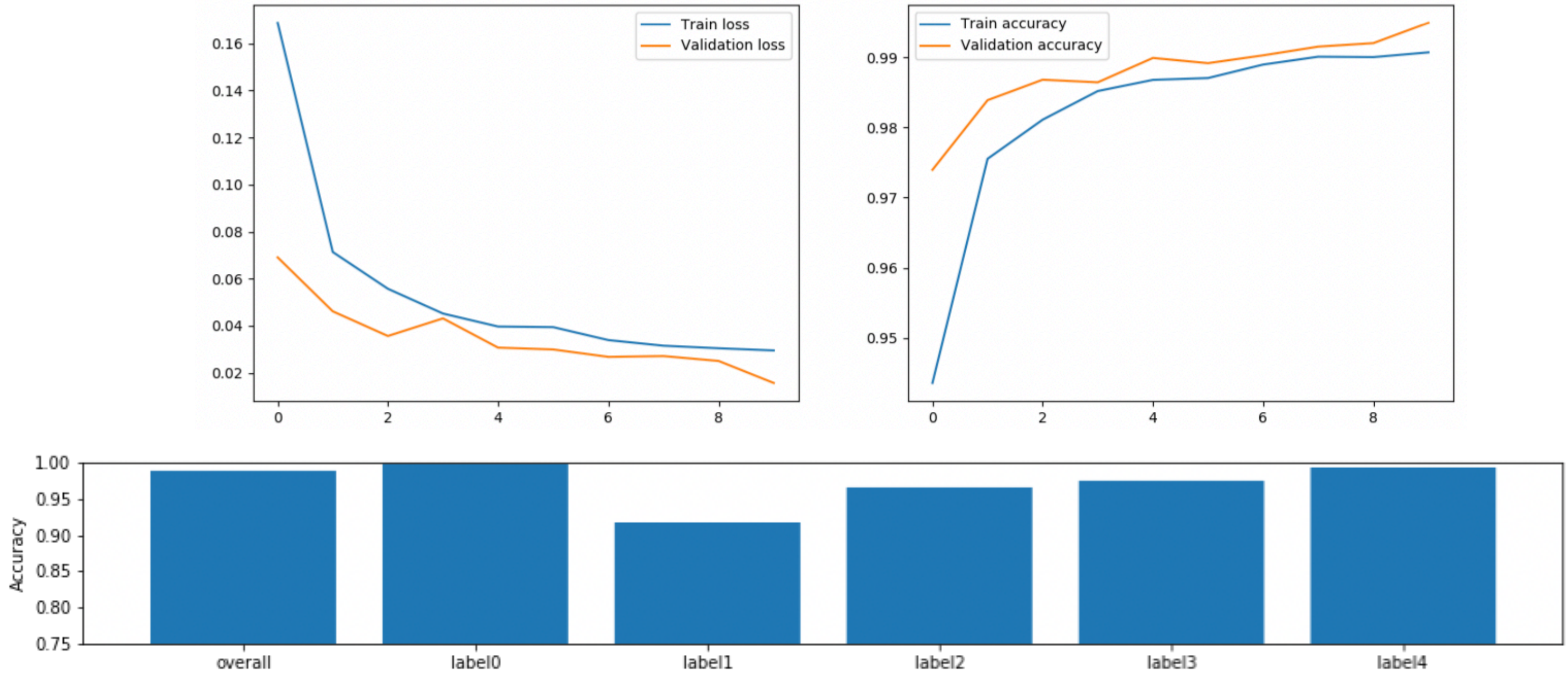
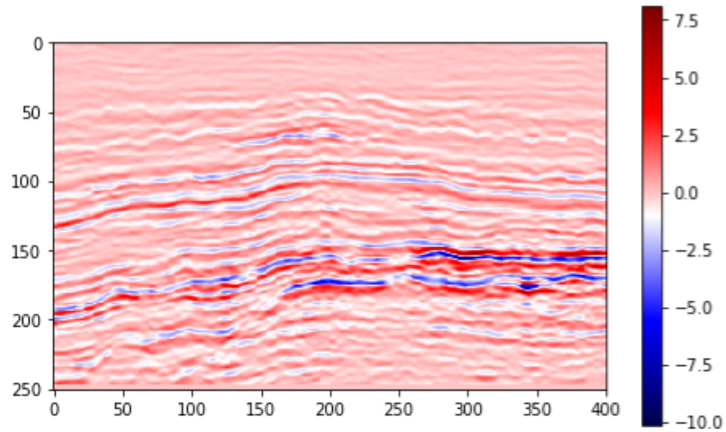
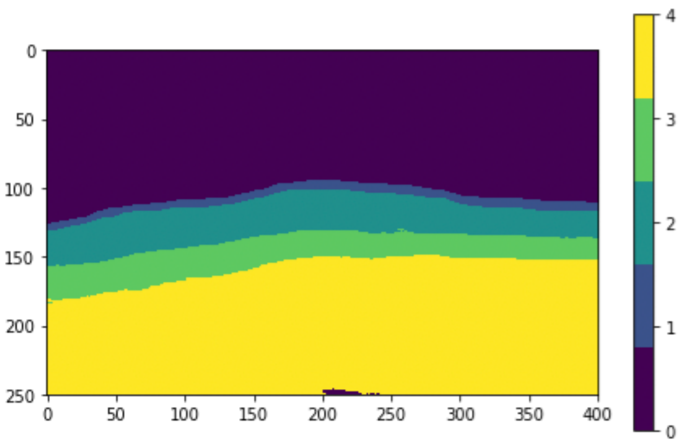


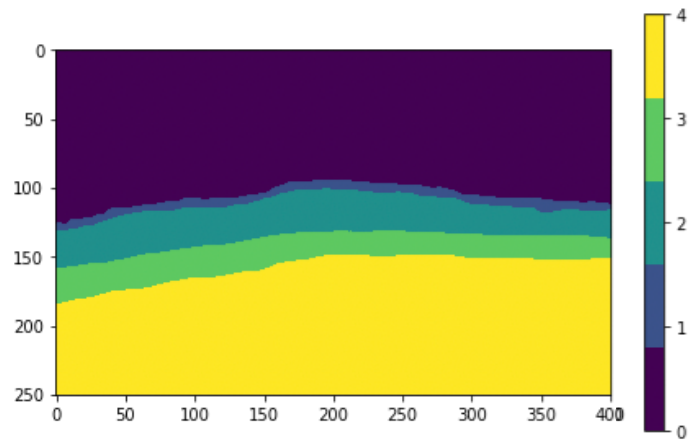
Figure: Overall pixel-wise accuracy on the SWC predicted seismic cube is 0.9894 and the IoU accuracy is 0.9426.



Seismic amplitudes (76th inline)



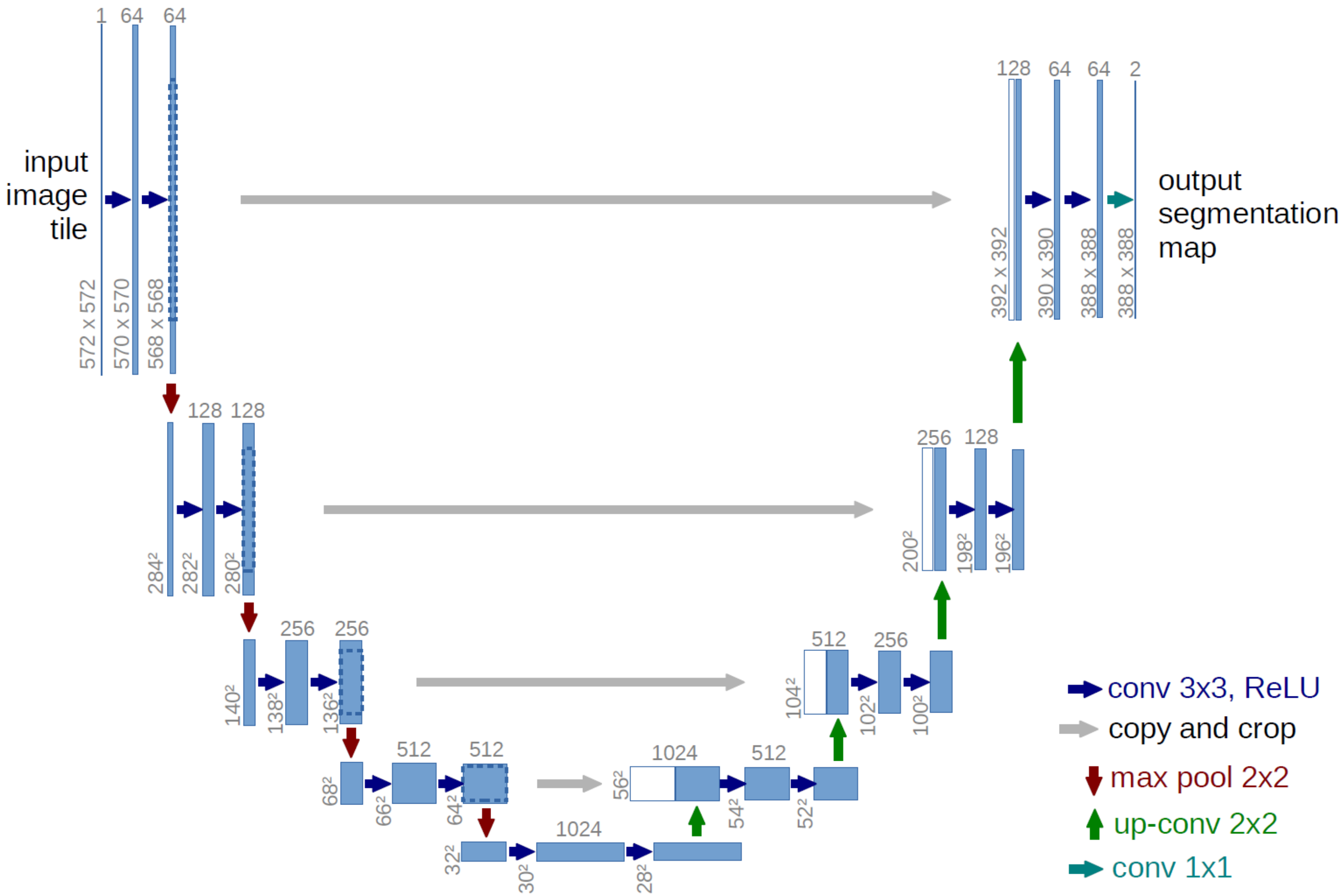
Segmentation Map from SWC



Ground truth

Fully Convolutional Networks (FCNs)

- FCNs are considered to be the more elegant than SWCs as they avoid redundant computation of low-level filters many times on pixels in overlapping patches.
- We use Auto-Encoder and U-Net architectures as our FCNs.
- For training and validation sets we use 8 and 2 inline/crossline slices respectively. We apply horizontal flips to the training data to increase its size by two fold.
- Training details – optimizer: Adams, Batch size: 4, Epochs: 100, loss: categorical cross-entropy. We weight the loss for each class proportionally to the class proportion in the training dataset.


1
64
64
input image tile
572 x 572
570 x 570
568 x 568
128
128
284²
282²
280²
256
256
140²
138²
136²
512
512
68²
66²
64²
1024
32²
30²
28²
1024
512
56²
54²
52²
512
256
104²
102²
100²
256
128
200²
198²
196²
128
64
64
2
392 x 392
390 x 390
388 x 388
388 x 388
output segmentation map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

Fully Convolutional Networks (FCNs) Results

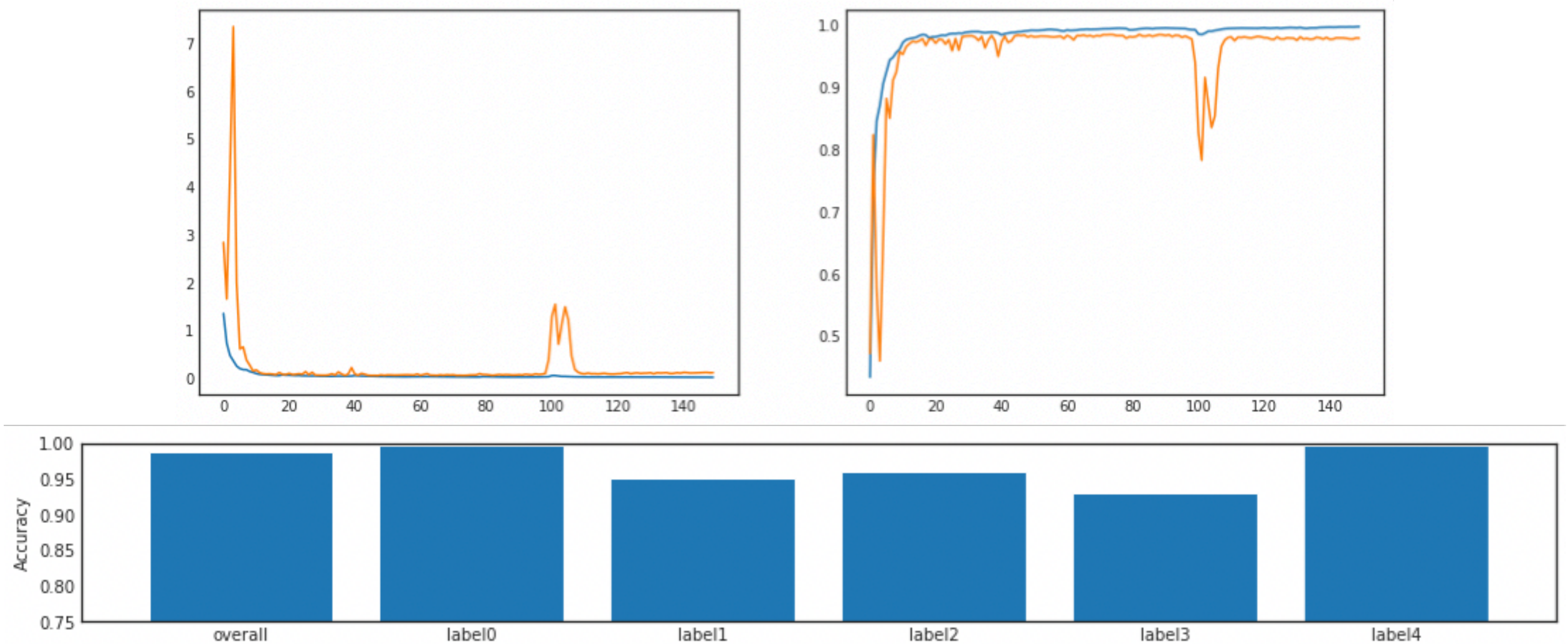
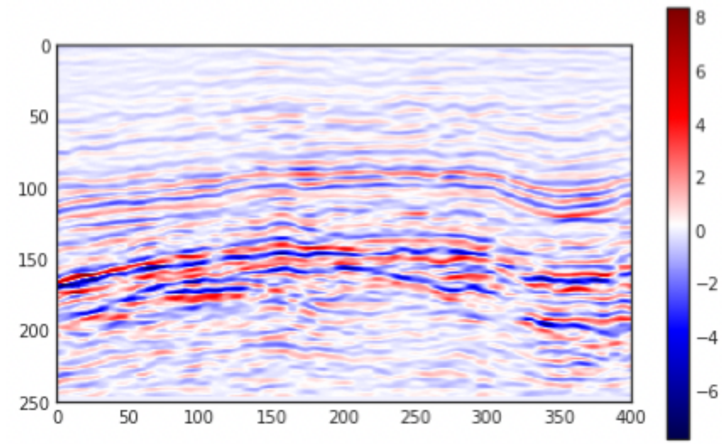
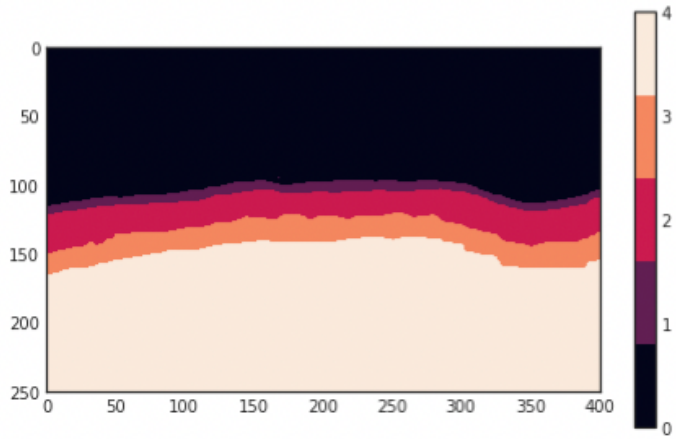


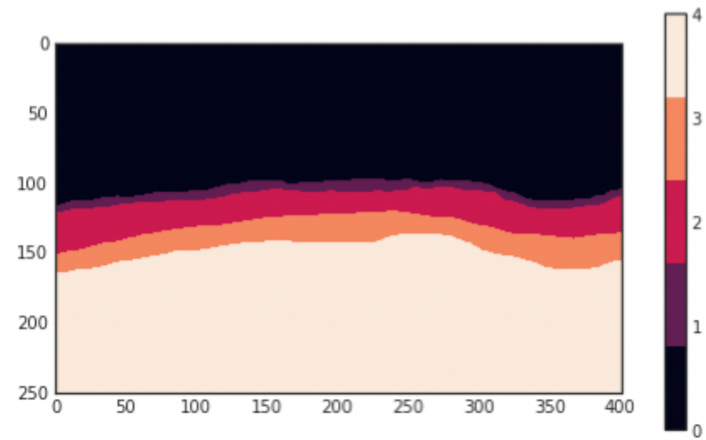
Figure: Overall pixel-wise accuracy on the U-Net predicted seismic cube is 0.9870 and the IoU accuracy is 0.9345.



Seismic amplitudes (330th crossline)



Segmentation Map from U-Net



Ground truth

Comparison of FCNs and SWCs

FCN

- Number of neurons in the model: 5×10^6
- Number of patches required to label the entire seismic cube: 401
- Time required to predict the entire seismic volume: 15 seconds
- Accuracy (pixelwise/IoU) of prediction: 0.9870/0.9345

SWC

- Number of neurons in the model: 3×10^5
- Number of patches required to label the entire seismic cube: 4×10^7
- Time required to predict the entire seismic volume: 13 hours
- Accuracy (pixelwise/IoU) of prediction: 0.9894/0.942

Conditional Random Fields (CRFs)

- Many state-of-the-art segmentation algorithms also include a CRF model within their pipelines. We consider CRF models as post-processing tools to structure the output space of the FCN/SWC model.
- We use fully-connected CRFs with gaussian edge potential [efficient inference implementation by Krahembuhl et al. 2011]

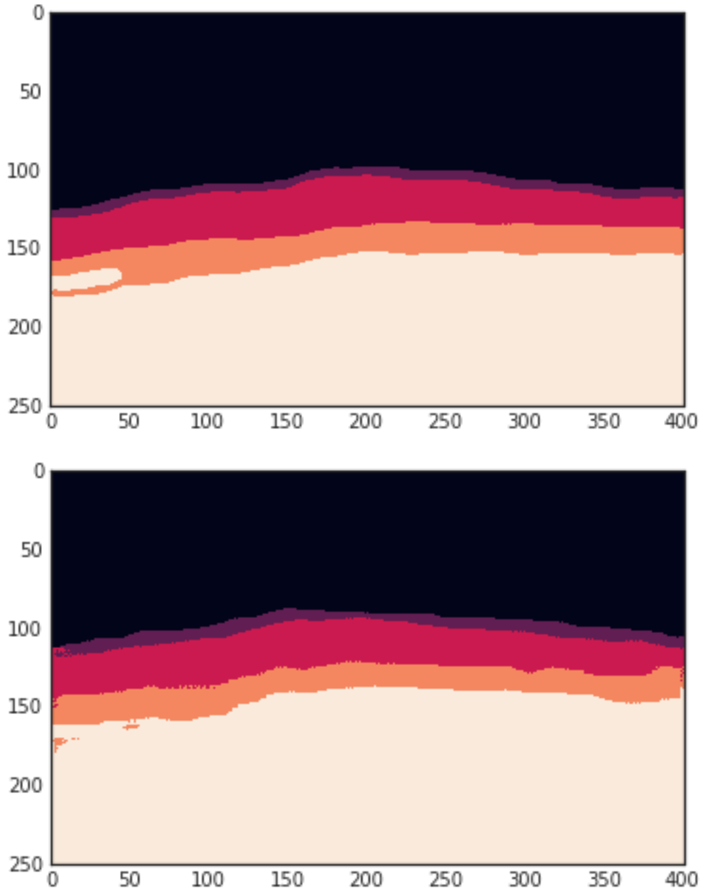
- The CRF model considers the Gibbs energy:

$$E(\boldsymbol{x}) = \sum_i \psi_u(x_i) + \sum_{i<j} \psi_p(x_i, x_j),$$

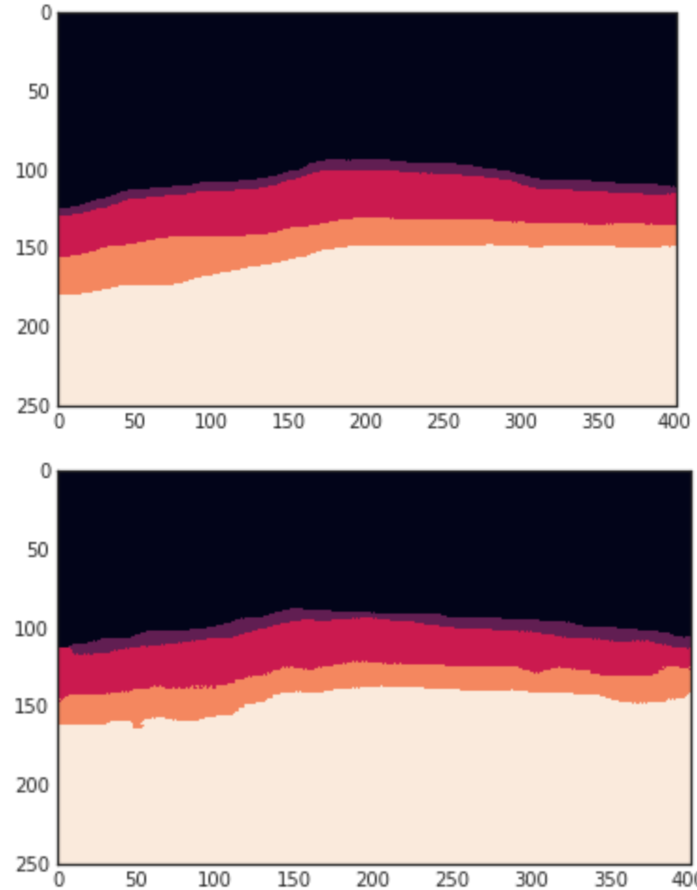
where, ψ_u is the unary-potential term which describes the cost of assigning a certain label to a given pixel (obtained from the FCN/SWC predictions) and ψ_p is the pairwise gaussian potential term where,

$$\psi_p(x_i, x_j) = \delta(x_i, x_j) k(\boldsymbol{f}_i, \boldsymbol{f}_j), \ and\ k(\boldsymbol{f}_i, \boldsymbol{f}_j) = w \exp\left(-\frac{|\boldsymbol{f}_i - \boldsymbol{f}_j|^2}{2\theta^2}\right).$$

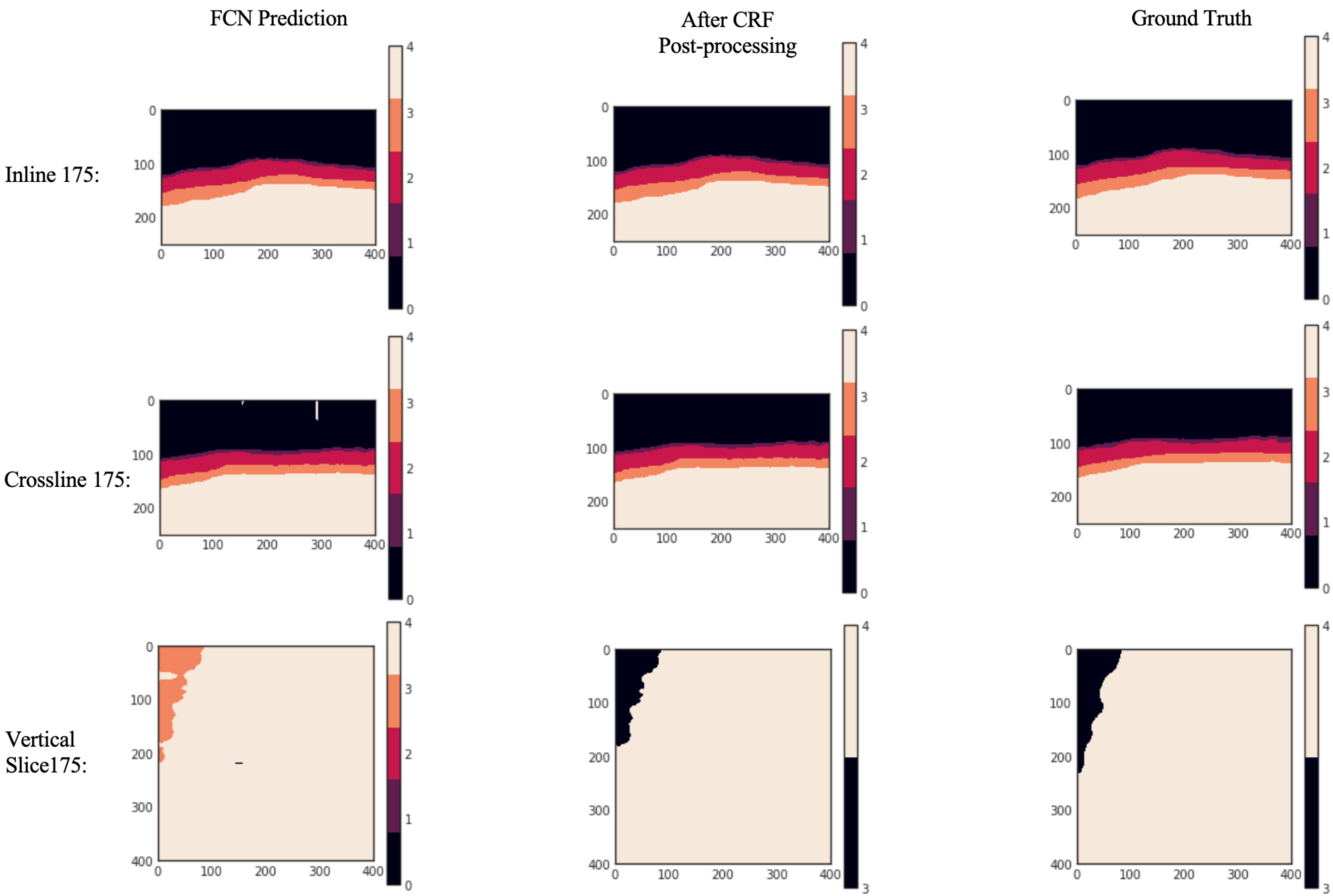
Conditional Random Fields (CRFs) Results



Before CRF



After CRF


FCN Prediction
After CRF
Post-processing
Ground Truth
Inline 175:
Crossline 175:
Vertical
Slice175:

Ensemble Approach

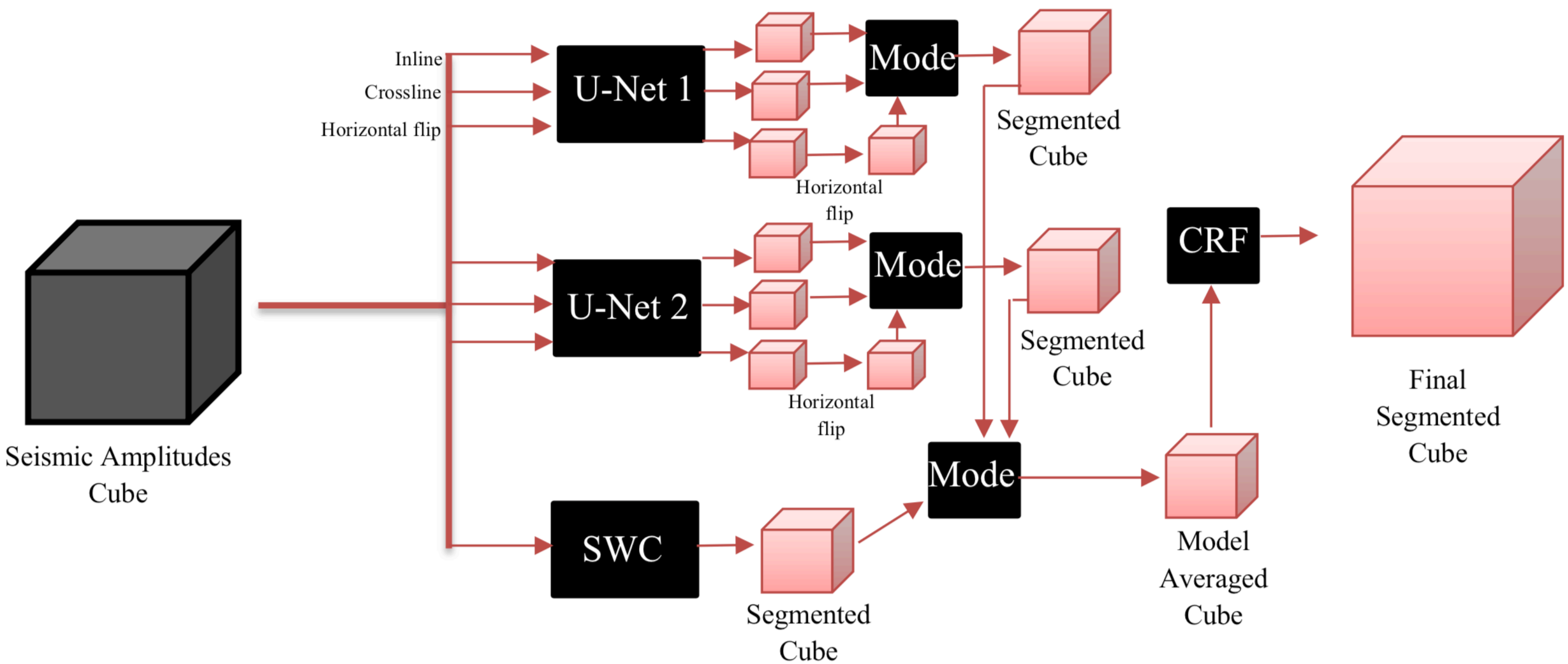


Figure: Ensemble approach used. We feed the original amplitudes cube to the U-Net models (model 3 and model 4 in our FCN_Notebook.ipynb) as inline slices, crossline slices and after horizontal flipping. We get three predictions for the cube segmentation map. We model average (by taking the mode) those predictions to get a better cube segmentation map. For the SWC, we feed in the patches extracted around each pixel to obtain a cube segmentation map. We model average (by taking the mode) the three segmentation maps obtained from U-Net1, U-Net2 and SWC. Then, we apply the CRF model on this model averaged segmented cube.

Software

Two main notebooks for training, predicting, CRF-Postprocessing, visualization using FCN and SWC approaches:

- FCN_Notebook.ipynb
- SWC_Notebook.ipynb

Two notebooks for model averaging and 3d-visualization:

- model_average.ipynb
- 3d_visualize.ipynb

Conclusions - Future Work

It was observed that when training with small datasets SWC approach has an advantage. This ability of SWC to extract patches and generate large training datasets enables it to overfit significantly less to the training data and hence achieve slightly higher validation/testing accuracies. However, patch extraction results in long prediction times and FCN approaches are usually many thousands of times faster than SWCs for making predictions. On the other hand, FCNs achieved higher training accuracies than the SWCs and when trained with larger datasets they achieved higher validation/testing accuracy than the SWC. The CRF post-processing was observed to be very promising when prediction accuracies were low where the overall pixel-wise accuracies could be improved significantly. However, for high accuracy prediction (e.g. pixel-wise accuracies > 98.5%), the CRF model often only barely improve the overall accuracy (by often < 0.1%).

Possible Future work:

- CRFs as Recurrent Networks (zheng et al. 2015, Arnab et al. 2018)
- Adversarial training to learn more complex CRF potentials (Luc et al. 2012)

Questions ?