Salt Identification Using Image Segmentation

Dylan Bailey

Process

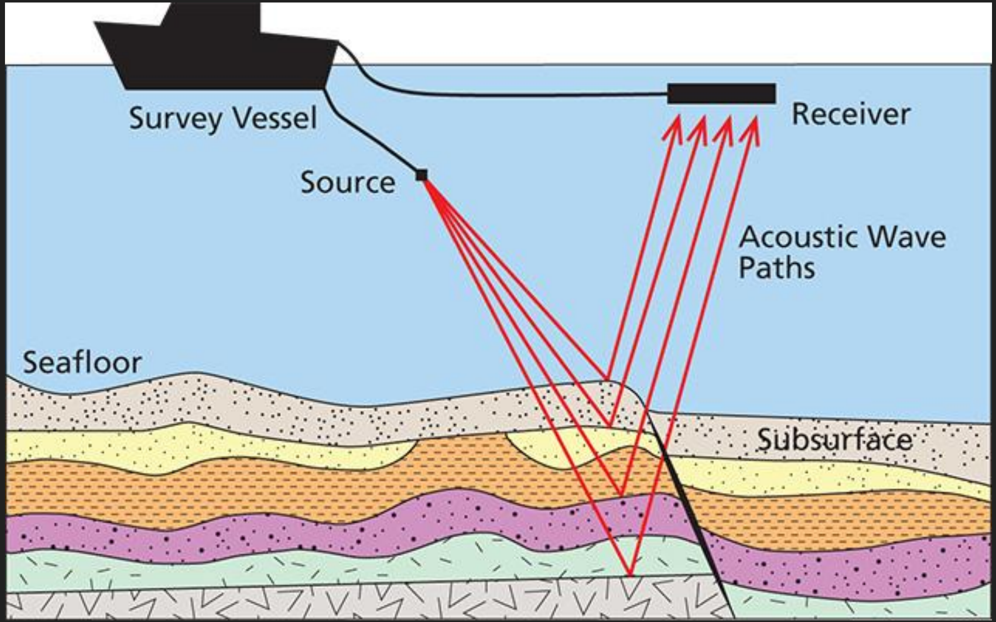
1. Problem and Challenges
2. What is seismic reflection?
3. How and why is it used?
4. Semantic Segmentation
5. Modeling (U-Net)
6. Results
7. Future

The Problem

- Identifying the amount of salt in seismic images
- Seismic data interpreters are used to interpreting 2D or 3D images that have been heavily processed
- We are going to use machine learning to speed up the process and take out any human bias

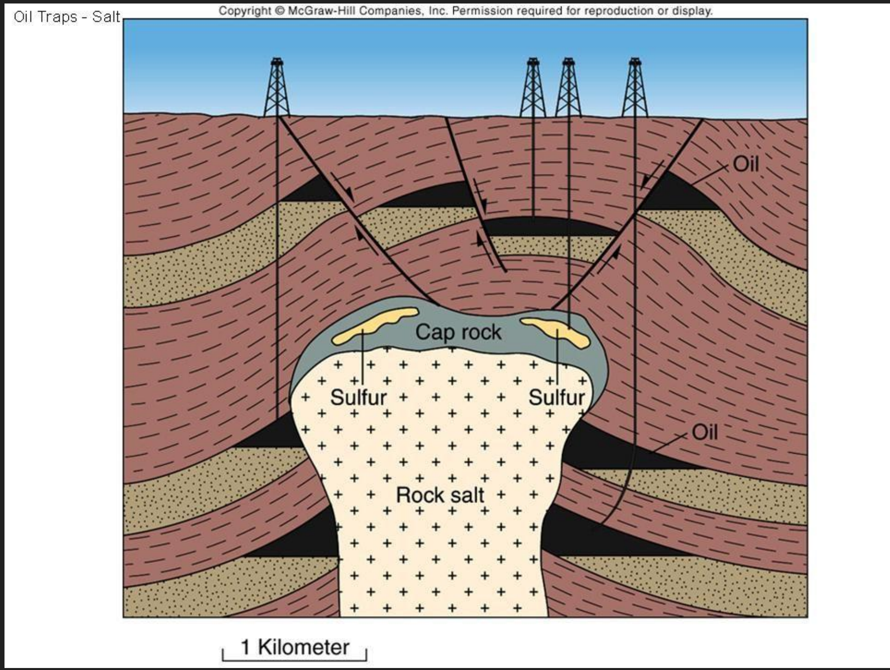
Seismic Reflection

- A method of exploring Earth's crust through artificial waves
- These waves fan out from the source and depending on what they hit are reflected at different speeds



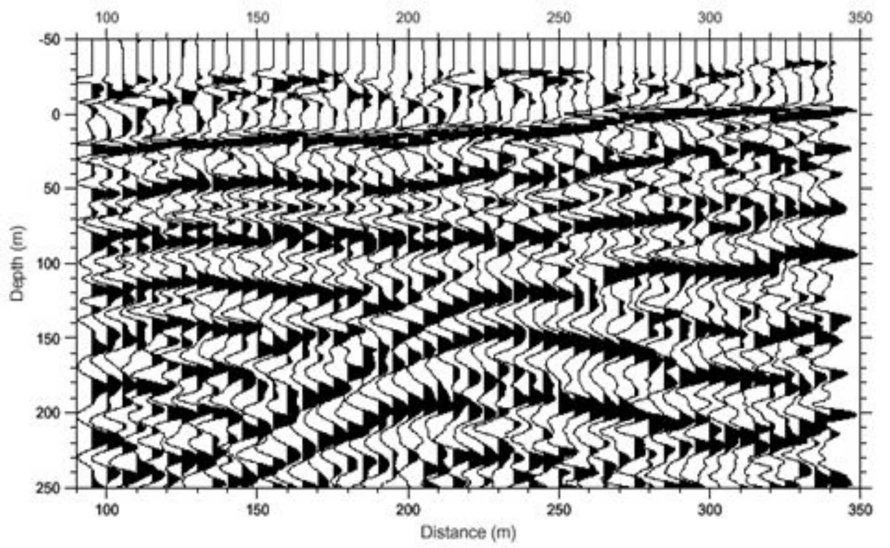
Why is this Important?

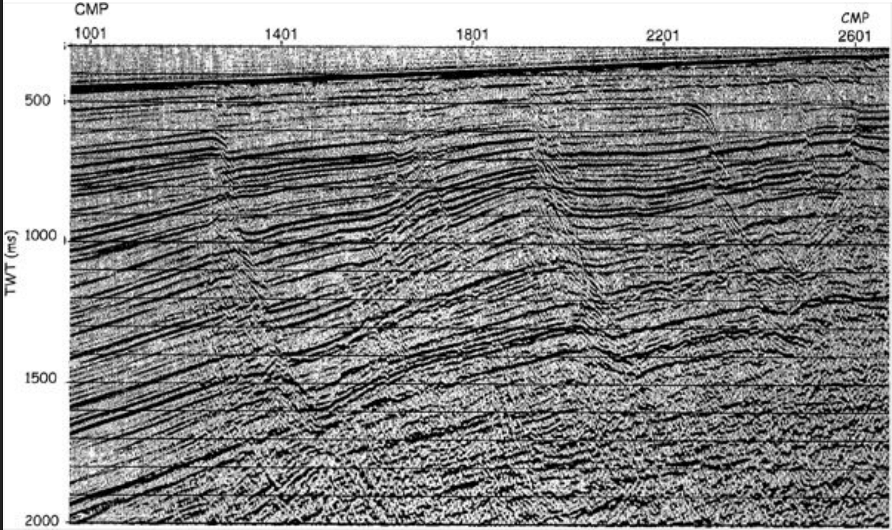
- Oil traps form where low density salt is squeezed upwards through tectonic forces
- These salt formations create reservoirs for hydrocarbons
- Finding these formations involves effectively recognizing how they will appear in a seismic image



Challenges

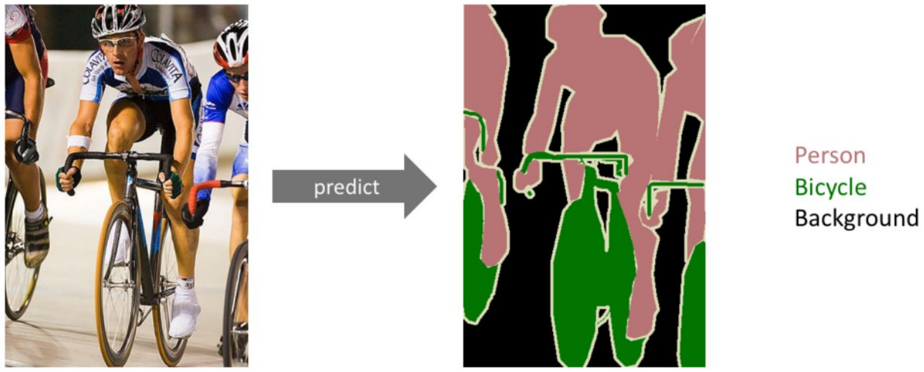
- Real world
  - Seismic images are messy
  - “Noise” from waves and artificial sources create disturbances
  - Lower resolution at deeper depths
- Machine Learning
  - Speed
  - Low resolution images
  - Complicated subsurface structures





Semantic Segmentation

- Semantic segmentation is using computer vision to label each pixel of an image with a corresponding class
- Unlike other forms of image segmentation the output itself is a high resolution image
- Used for:
  - Autonomous vehicles
  - Bio-medical imaging
  - Geo-Sensing
  - Precision Agriculture

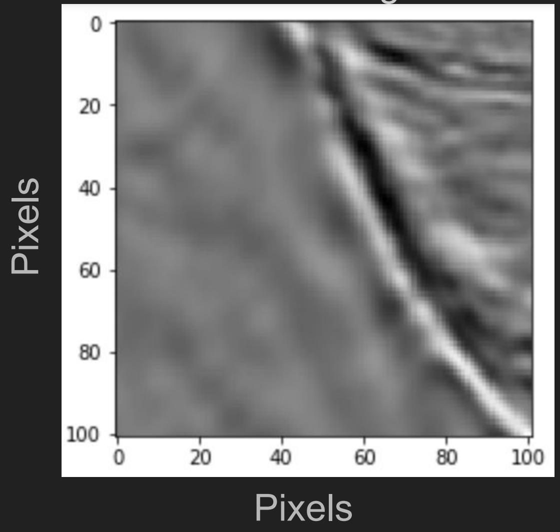


Semantic Segmentation

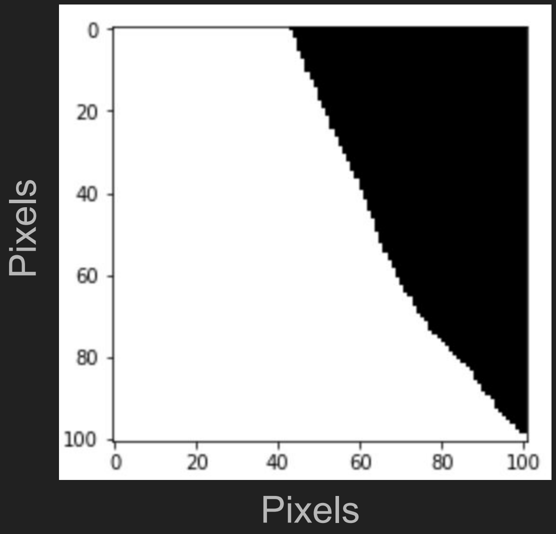
Data

- Training data: 4,000 images paired with masks
- Testing data: 18,000 images
- Images 101x101 pixel size

Seismic Image

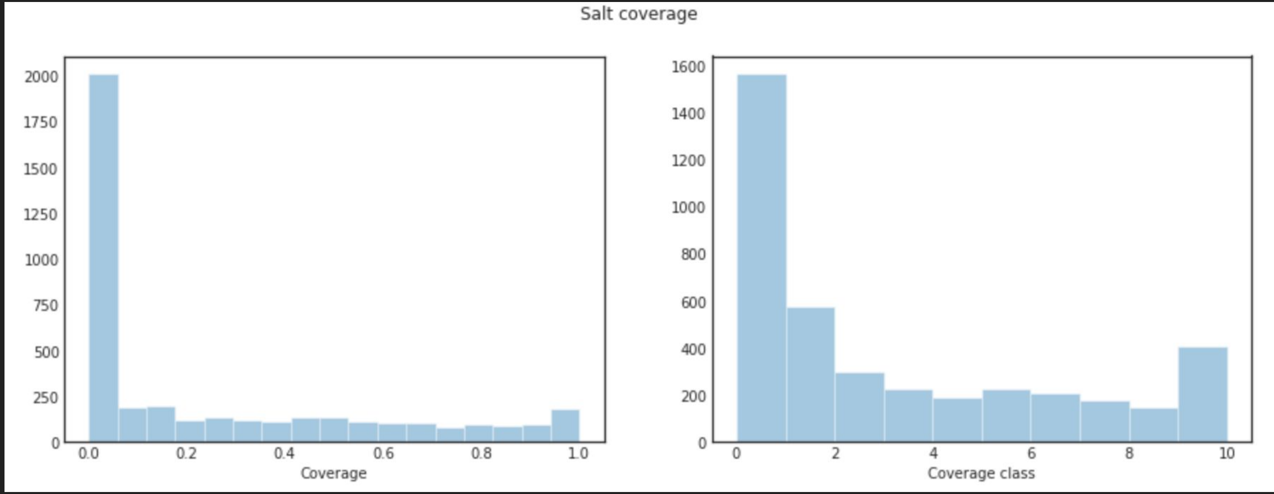


Salt Mask



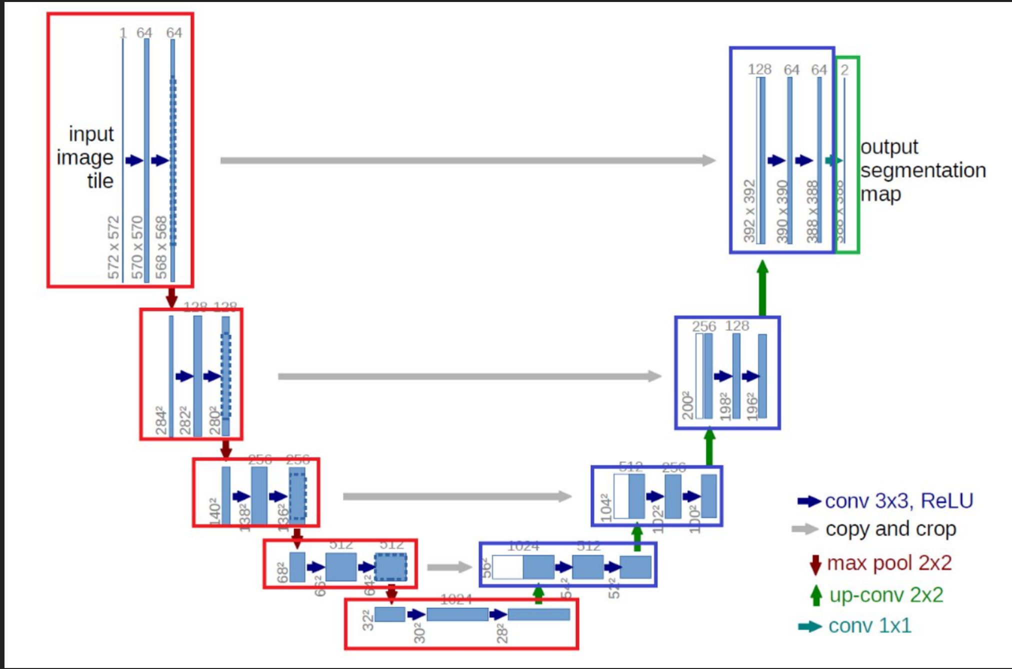
Pre-Processing

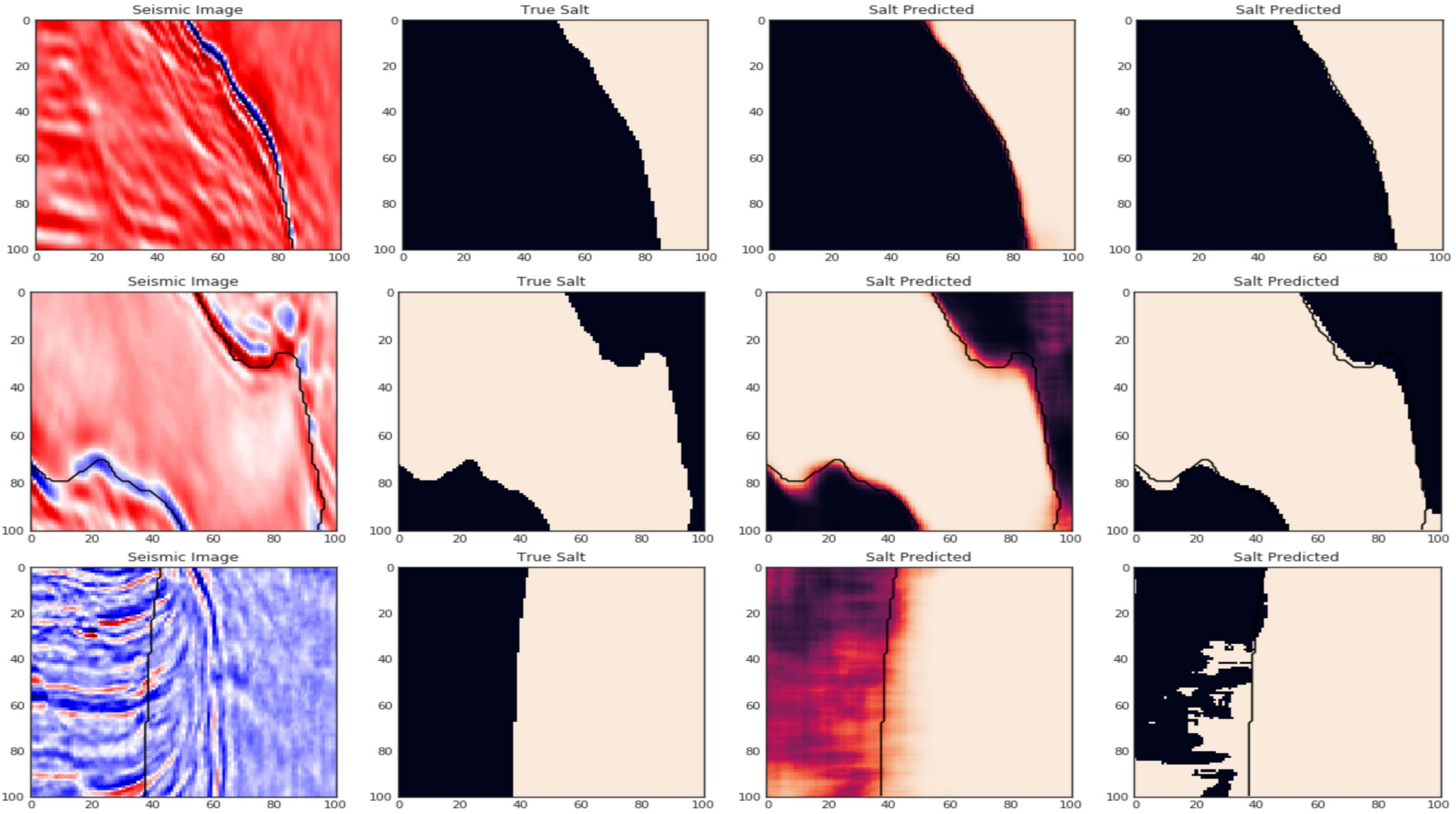
- Each image is turned into a numpy array and set to greyscale
- Network expects values between 0 and 1. Divide arrays by 255
- Augmentation
  - Each image is flipped from left to right
  - This creates more images for the training set



U-Net

- Multiclass classification model for images
- Encoding path (contracting)/decoding path (expanding) architecture
- Encoder consists of convolution blocks, max pooling with downsampling
- Decoder consists of upsampling, concatenation, and convolution layers,
- Final layer is a convolution to output predictions

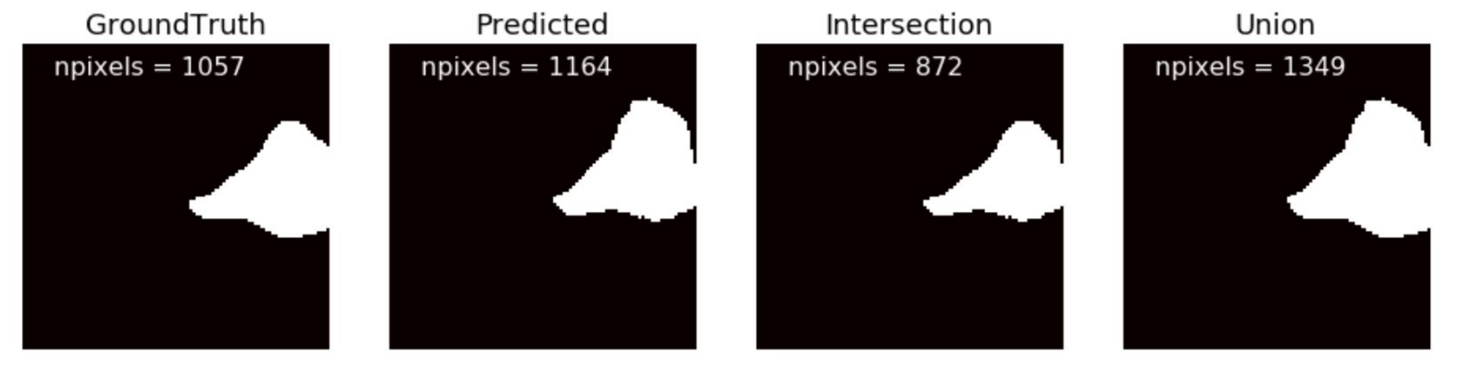



Seismic Image
True Salt
Salt Predicted
Salt Predicted
Seismic Image
True Salt
Salt Predicted
Salt Predicted
Seismic Image
True Salt
Salt Predicted
Salt Predicted

Intersection of Union Metric

$$IoU = \frac{|\text{mask} \cap \text{prediction}|}{|\text{mask} \cup \text{prediction}|}$$

- Comparing the original mask to the predicted mask
- Considers a set of pixels, the ones the model classifies as salt and compares it to to the true mask which has the true amount of pixels classified as salt



Results

Best Model: U-Net with Resnet. 300 epochs

Train: 82.10%

Test: 78.48%

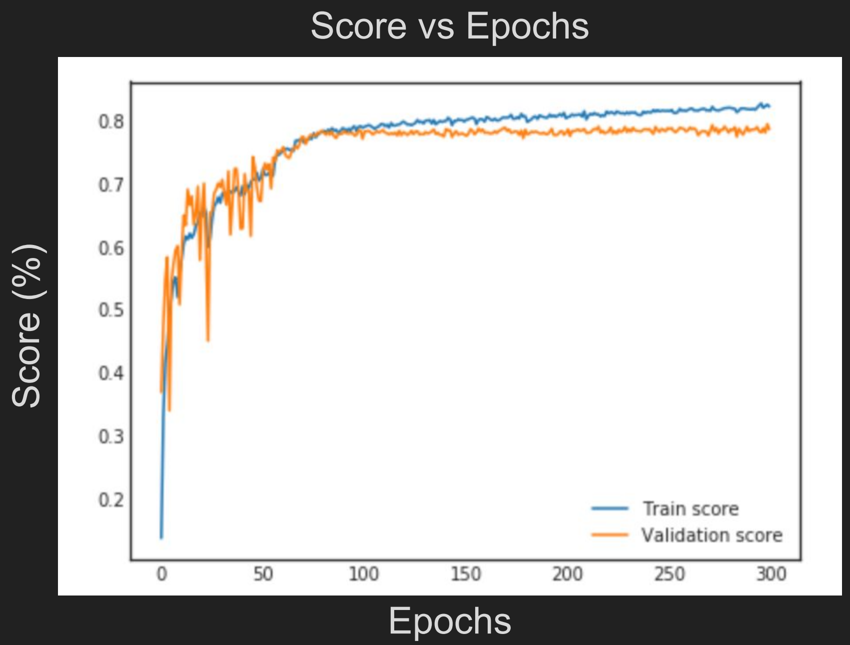
Kaggle: 81.79% (top 30%)

Boosts:

More Layers: 16 to 32 ~4%

Resnet increased score ~5%

Augmentation increased score ~3%



Future

1. Try other semantic segmentation models (i.e. R-CNN, FRRN)
2. Increase resolution of images
3. Use depth as a feature
4. Different versions of metric IOU (i.e. Dice loss, Lovasz loss)
5. More computing power
6. Grid searching over parameters
7. Use model on new data, not only seismic images. (i.e. radio frequencies, satellite images, bio-medical images)

Questions?