Understanding Semantic Segmentation with UNET

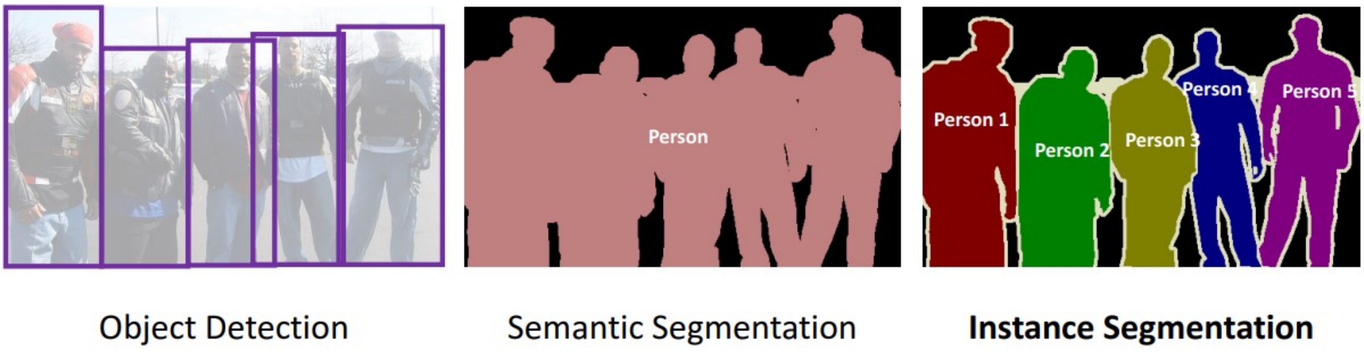
A Salt Identification Case Study

Harshall Lamba Follow
Feb 17 · 15 min read

Table of Contents:

1. Introduction

Computer vision is an interdisciplinary scientific field that deals with how computers can be made to gain high-level understanding



Object Detection vs Semantic Segmentation vs Instance Segmentation

In this post we will learn to solve the Semantic Segmentation problem using Fully Convolutional Network (FCN) called UNET.

4. Applications

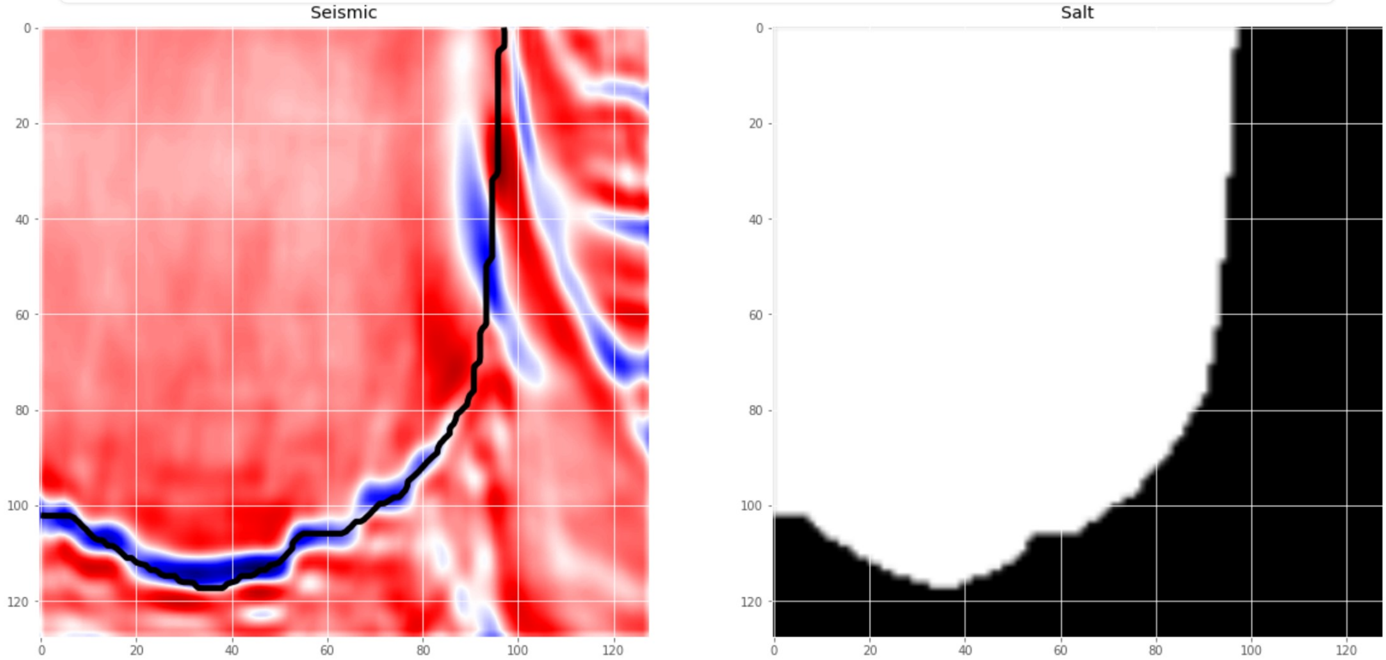
If you are wondering, whether semantic segmentation is even useful or not, your query is reasonable. However, it turns out that a lot of complex tasks in Vision require this fine grained understanding of images. For example:

a. Autonomous vehicles

Autonomous driving is a complex robotics tasks that requires perception, planning and execution within constantly evolving environments. This task also needs to be performed with utmost precision, since safety is of paramount importance. Semantic Segmentation provides information about free space on the roads, as well as to detect lane markings and traffic signs.

Source: https://www.youtube.com/watch?v=ATlcEDSPWXY

b. Bio Medical Image Diagnosis

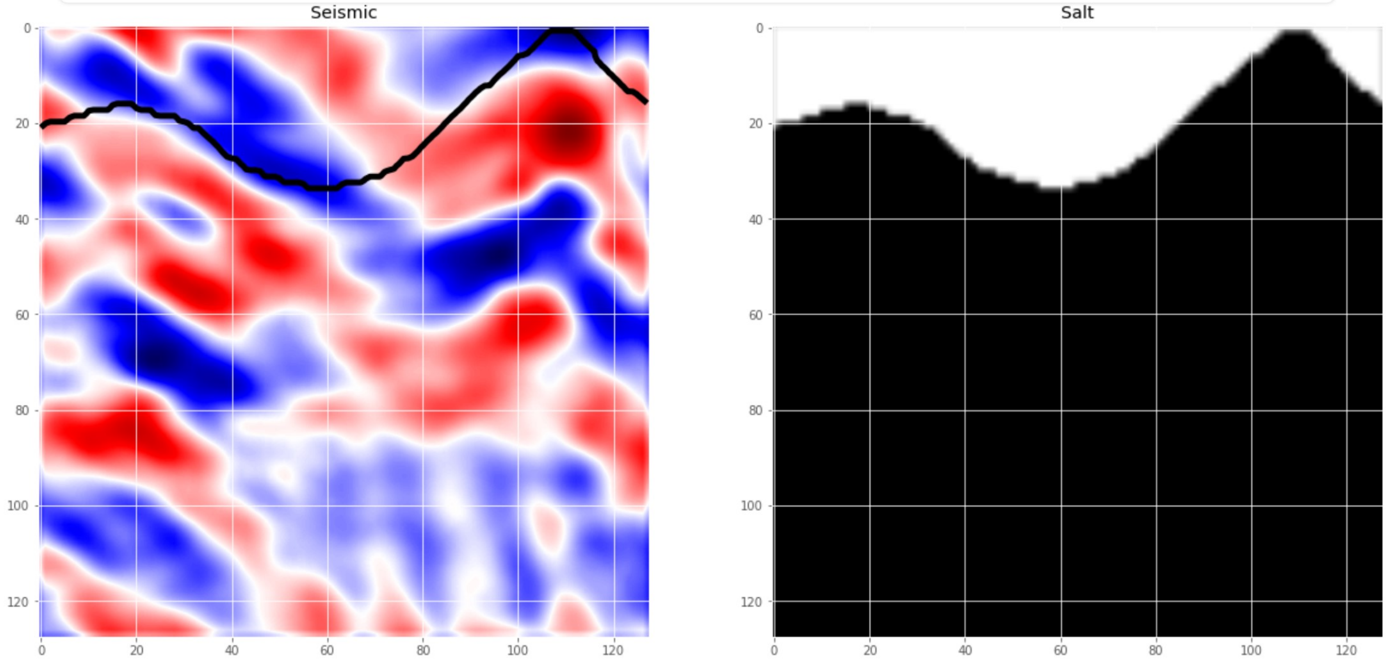


Sample data point and corresponding label

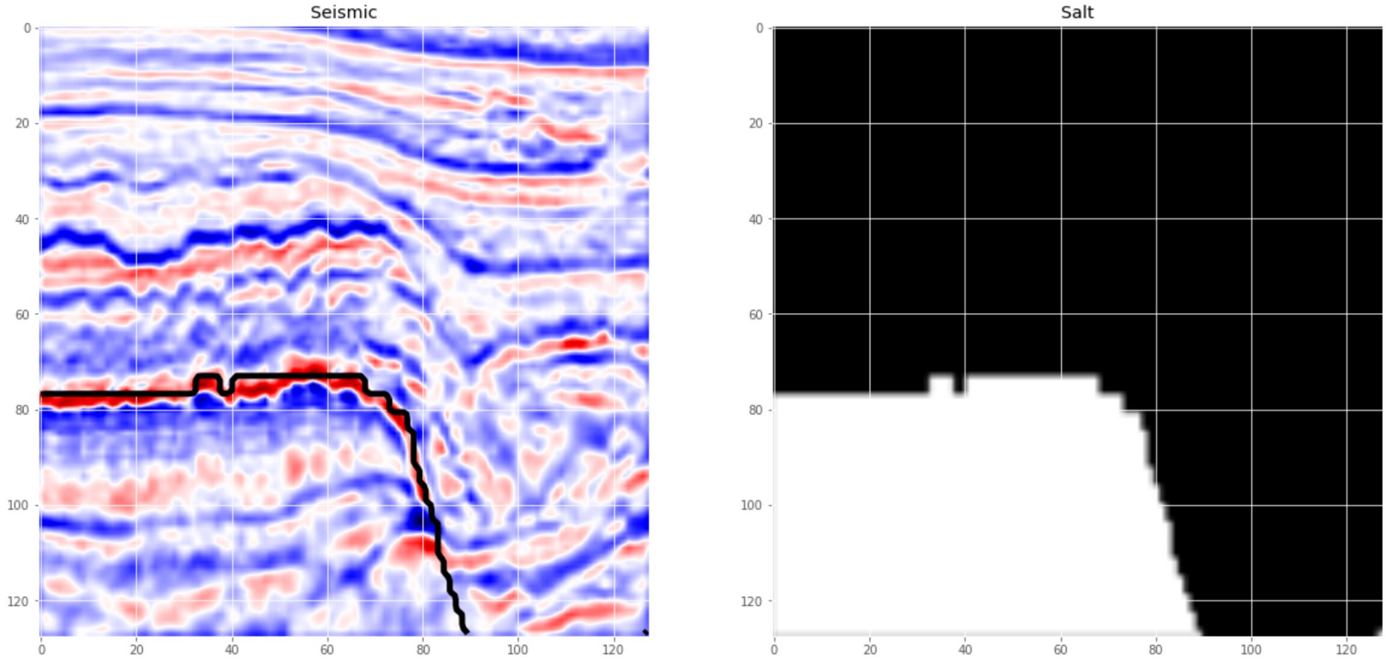
The image on left is the seismic image. The black boundary is drawn just for the sake of understanding denoting which part contains salt and which does not. (Of course this boundary is not a part of the original image)

The image on the right is called as the mask which is the ground truth label. This is what our model must predict for the given seismic image. The white region denotes salt deposits and the black region denotes no salt.

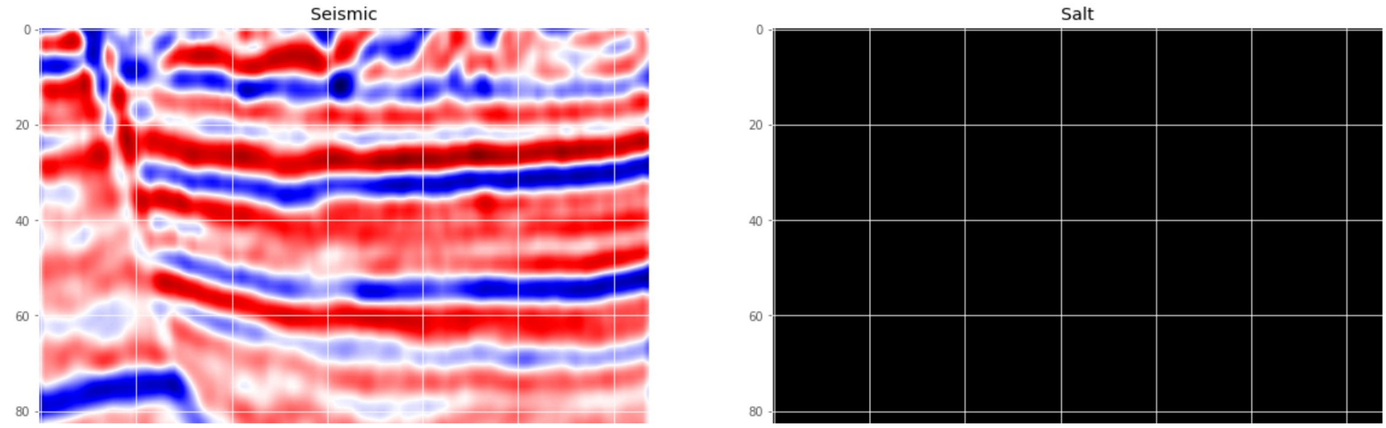
Let’s look at a few more images:



Sample data point and corresponding label



Sample data point and corresponding label



Notice that if the mask is entirely black, this means there are no salt deposits in the given seismic image.

Clearly from the above few images it can be inferred that its not easy for human experts to make accurate mask predictions for the seismic images.

7. Understanding Convolution, Max Pooling and Transposed Convolution

Before we dive into the UNET model, it is very important to understand the different operations that are typically used in a Convolutional Network. Please make a note of the terminologies used.

i. Convolution operation

There are two inputs to a convolutional operation

i) A 3D volume (input image) of size (nin x nin x channels)

ii) A set of 'k' filters (also called as kernels or feature extractors) each one of size (f x f x channels), where f is typically 3 or 5.

The output of a convolutional operation is also a 3D volume (also called as output image or feature map) of size (nout x nout x k).

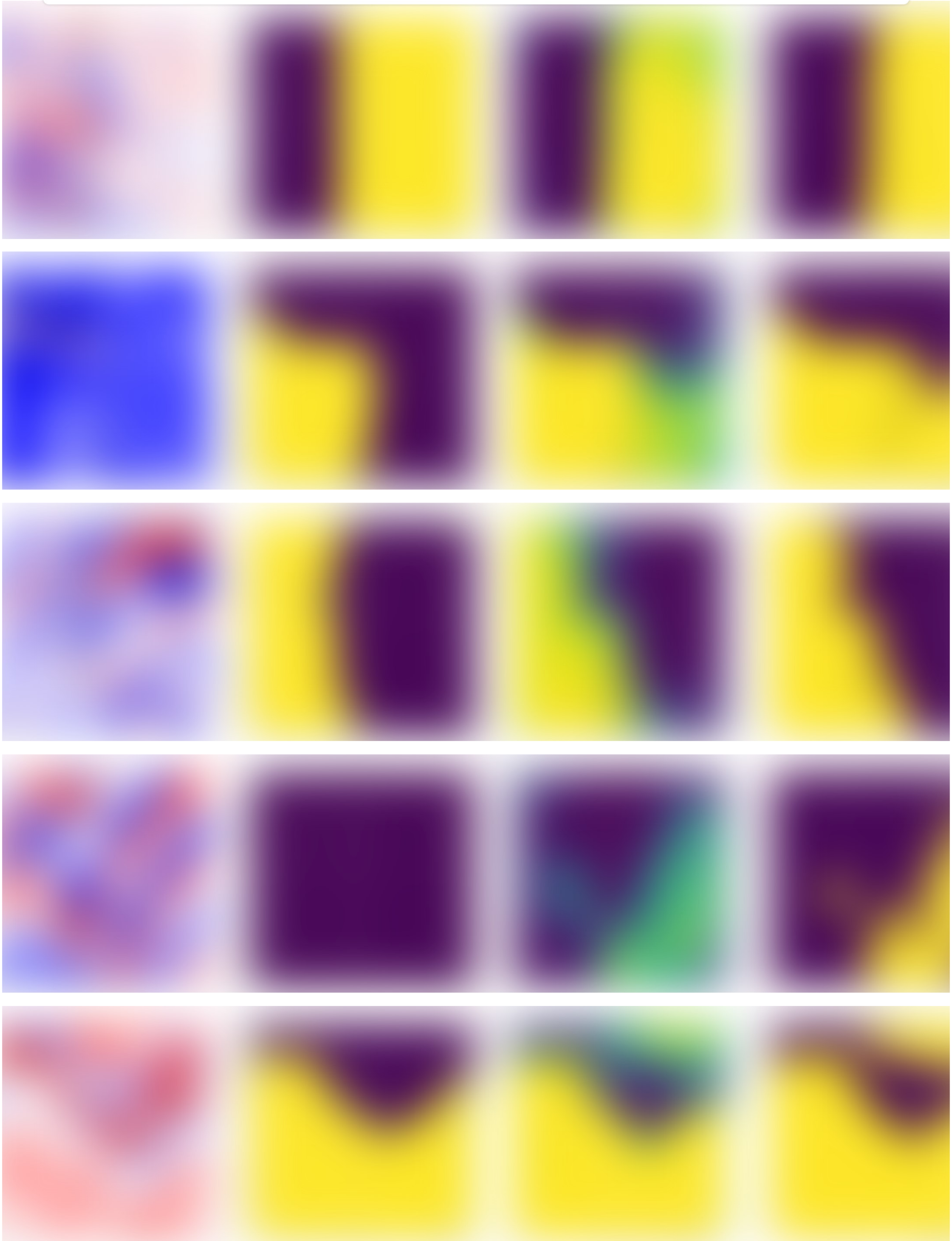
The relationship between nin and nout is as follows:

$$| n_{in} + 2p - k |$$

Detailed UNET Architecture

- 2@Conv layers means that two consecutive Convolution Layers are applied
- c1, c2, …. c9 are the output tensors of Convolutional Layers
- p1, p2, p3 and p4 are the output tensors of Max Pooling Layers
- u6, u7, u8 and u9 are the output tensors of up-sampling (transposed convolutional) layers
- The left hand side is the contraction path (Encoder) where we apply regular convolutions and max pooling layers.
- In the Encoder, the size of the image gradually reduces while the depth gradually increases. Starting from 128x128x3 to 8x8x256
- This basically means the network learns the “WHAT” information in the image, however it has lost the “WHERE” information
- The right hand side is the expansion path (Decoder) where we apply transposed convolutions along with regular convolutions
- In the decoder, the size of the image gradually increases and the depth gradually decreases. Starting from 8x8x256 to 128x128x1
- Intuitively, the Decoder recovers the “WHERE” information (precise localization) by gradually applying up-sampling
- To get better precise locations, at every step of the decoder we use skip connections by concatenating the output of the transposed convolution layers with the feature maps from the Encoder at the same level:

Results on Validation Set



set which implies the model suffers from overfitting. One obvious reason could be the small number images used to train the model.

10. Conclusion

Thank you for interest in the blog. Please leave comments, feedback and suggestions if you feel any.

Full code on my GitHub repo *here*.

11. References

- https://arxiv.org/abs/1505.04597
- https://www.depends-on-the-definition.com/unet-keras-segmenting-images/
- https://towardsdatascience.com/up-sampling-with-transposed-convolution-9ae4f2df52d0

Machine Learning Deep Learning Semantic Segmentation Unet Convolutional Network