Project: Capstone Project 2: Milestone Report 1

Salt Identification in subsurface of reservoir

1. Introduction

Seismic data is collected using reflection seismology, or seismic reflection. The method needs a controlled seismic source of energy (such as compressed air or a seismic vibrator, and sensors record the reflection from rock interfaces within the subsurface). The recorded data is then processed to generate a 3D view of earth's interior. Reflection seismology is similar to X-ray, sonar and echolocation.

A seismic image is produced from imaging the reflection coming from rock boundaries. The seismic image shows the boundaries between different rock types. In theory, the strength of reflection is directly proportional to the difference in the physical properties on either sides of the interface. Seismic images do not provide any more information about the rock themselves except illustrating rock boundaries. Some rocks are easy to identify while some are difficult.

There are several areas of the world where there are vast quantities of salt in the subsurface. One of the challenges of seismic imaging is to identify the part of subsurface which is salt. The presence of salt makes it both simple and difficult to identify. Salt density is usually 2.14 g/cc which is lower than most surrounding rocks. The lower density of salt leads to the faster seismic velocity (4.5 km/s) than its surrounding rock surfaces. Usually salt is an amorphous rock without much internal structure. This means that there is typically not much reflectivity inside the salt, unless there are sediments trapped inside it. So, this difference in seismic velocity does not only generates a sharp reflection at the salt-sediment interface but also cause of some problems with seismic imaging.

The aim of this project is to build a good model to identify if a subsurface target is salt or not.

Data source: All data are come from kaggle dataset (total data is around 200 MB).

https://www.kaggle.com/c/tgs-salt-identification-challenge/data

2. Dataset description

The data is a set of images chosen at different locations and random in the subsurface. The images are 101 x 101 pixels and each pixel is classified as either salt or sediment. In addition, the depth of the imaged location is provided for each image.

In train.csv file, it provided file id and run-length encoded format (rle_mask) for each image in training set (Table 1). Run-length encoding (RLE) is a very simple form of lossless data compression in which runs of data (sequences in which the same data value occurs in many consecutive data elements) are stored as a single data value and count, rather than as the original run.

Table1: First 5 columns of train.csv file

id	**rle_mask**
575d24d81d	NaN
a266a2a9df	5051 5151
75efad62c1	9 93 109 94 210 94 310 95 411 95 511 96 612 96 712 97 812 98 913 98 1015 97 1116 97 1216 98 1316 99 1416 8786
34e51dba6a	48 54 149 54 251 53 353 52 455 51 557 50 659 49 762 47 864 46 966 45 1068 44 1171 42 1273 41 1376 39 1478 38 1581 36 1683 35 1785 34 1888 32 1990 31 2092 30 2195 28 2297 27 2399 26 2501 25 2602 25 2704 24 2806 23 2907 23 3009 22 3110 22 3212 21 3313 21 3414 21 3516 20 3617 20 3718 20 3819 20 3921 19 4022 19 4123 19 4225 18 4326 18 4428 17 4529 17 4631 16 4733 15 4834 15 4936 14 5038 13 5140 12 5242 11 5344 8

In our study, rle_mask can be NaN (no salt) or list of number in type of string. RLE values write down "only the white" pixels and always in pairs: (start_Pixel, number_Of_Pixels). So, the first value is "index", the second is "count", the third is "index" again, the fourth is "count", and so on. To count the number of white pixels, get only the numbers at even positions. To know where each sequence of pixels start, get only the odd numbers. Note that NaN value of rle_mask means that there is no salt in this image. So, there were 1562 seismic images in training set without salt (rle_mask = NaN).

In depth.csv file, it stored depth value in feet of all image (including both training and test set). All data were collected in from 50 to 959 feet. All image id of data are unique. Note that the value of the depth corresponds to the center of the image.

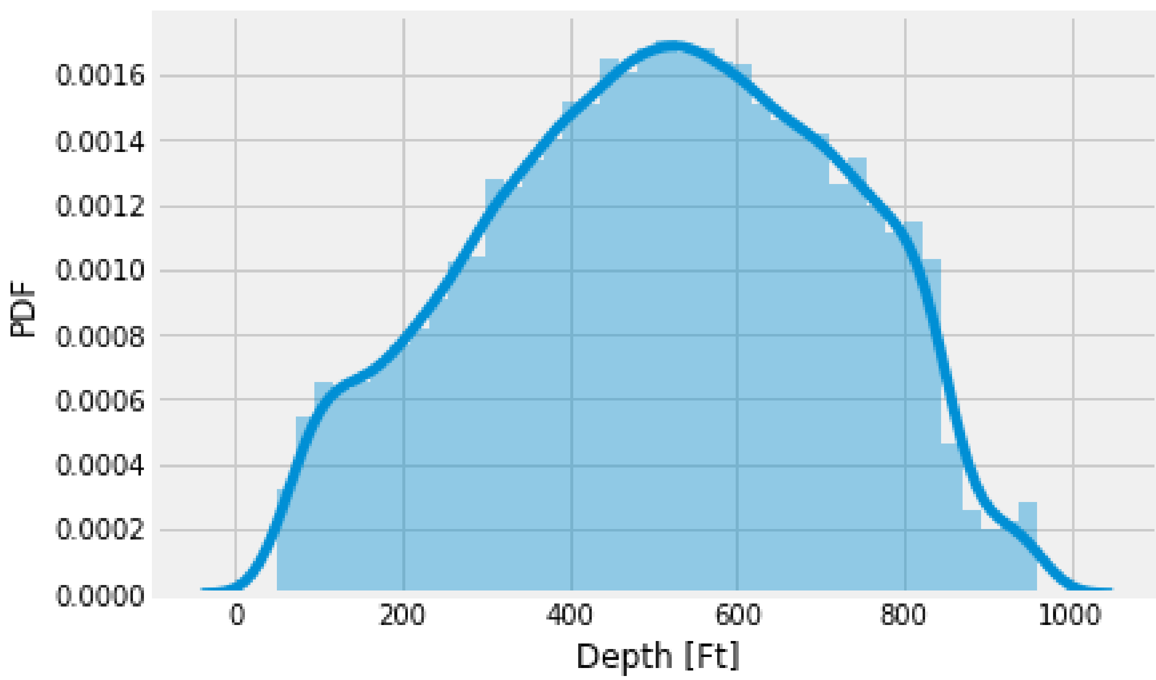
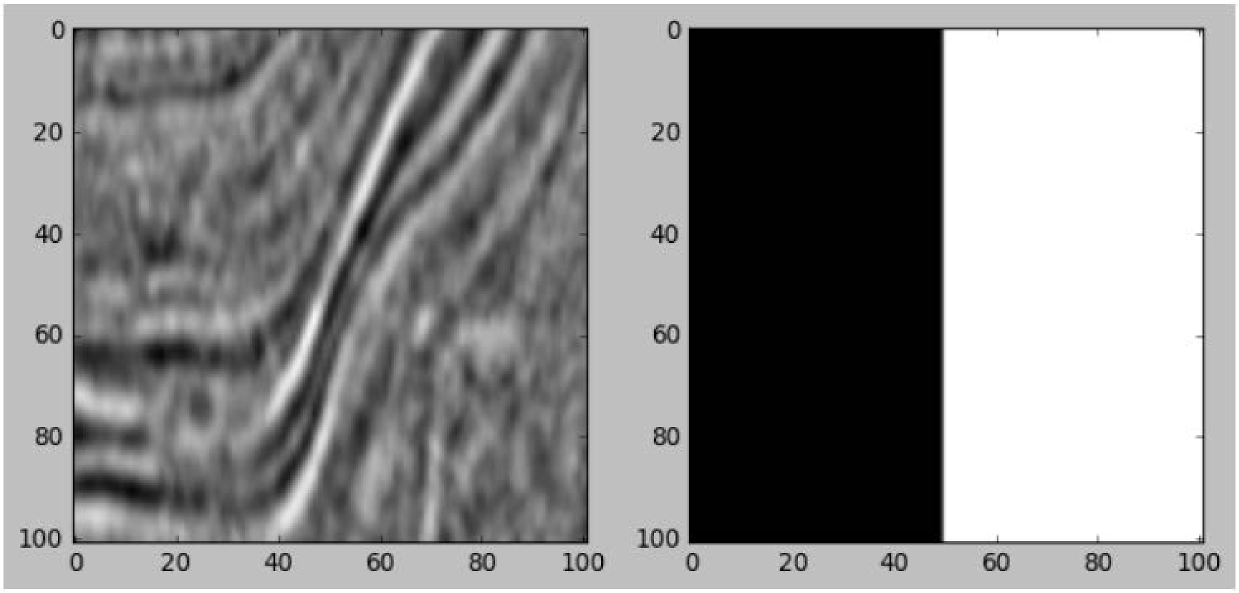


Figure 1: Depth distribution of all subsurface

There were 22000 seismic images in this dataset. Training set had 4000 seismic images and 4000 mask images. It indicated clearly the presence salt when looking at mask image (Figure 2). The white area shows the salt occurrence in the seismic image. For test set, they had 22000 seismic images that need to determine whether the presence of salt is.

File ID: a266a2a9df



File ID: 34e51dba6a

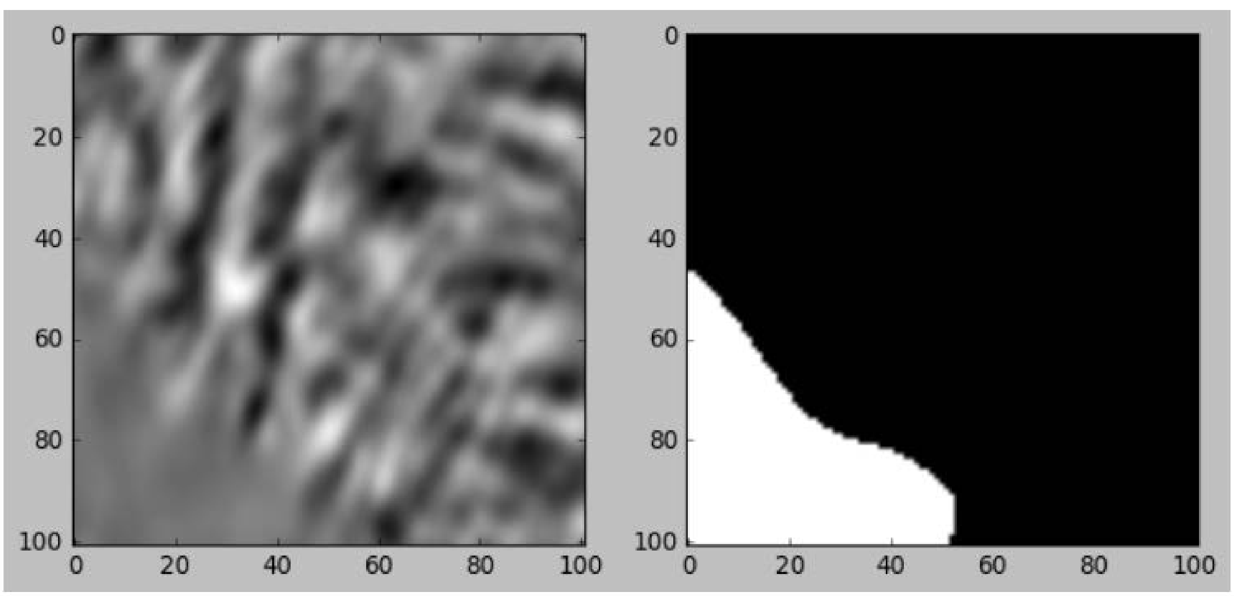


Figure 2: Seismic image and its mask of some figures in training set

3. Initial findings from exploratory analysis

Value of skewness (-0.12) and Q-Q plot in Figure 3 are clearly indicated that the depth distribution of all data is not followed the Gaussian distribution. Similarly, the depth distribution for training and test set are also not distributed normally.

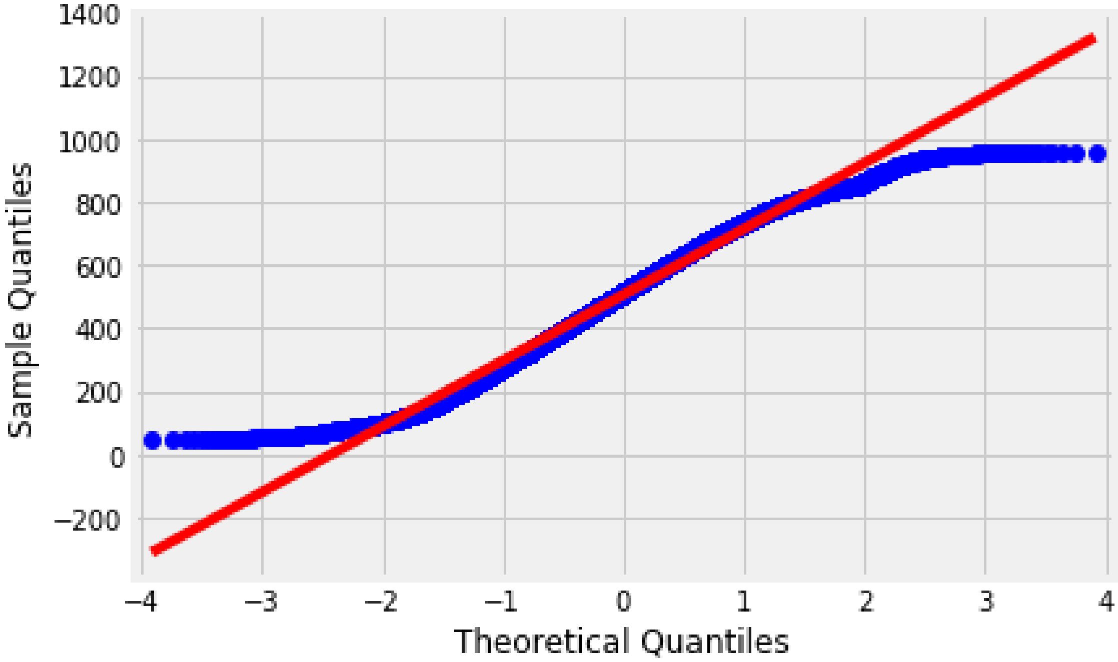
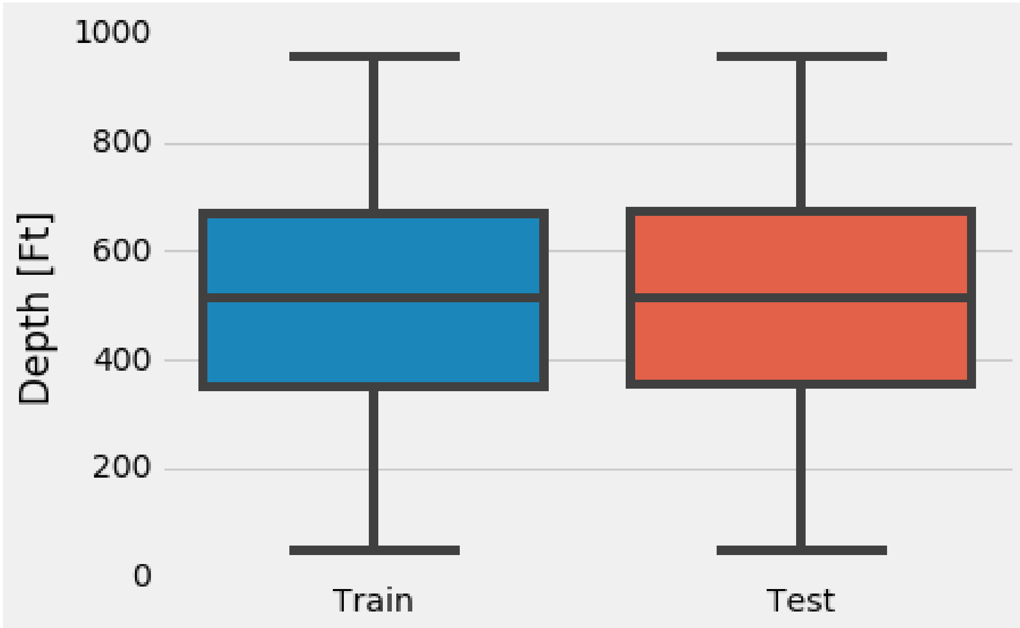


Figure 3: Q-Q plot of depth distribution for all data

Figure 4 shows the box plot and histogram of depth distribution for training and test set. They seem to have the same distribution. The Kolmogorov-Smirnov test was used to determine whether depth distributions of train and test set are identical. Null hypothesis that depth of train and test set have the same distribution. I got p-value is 0.736. So, null hypothesis can not be rejected. It can be considered that train and test set have the same distribution of depth.



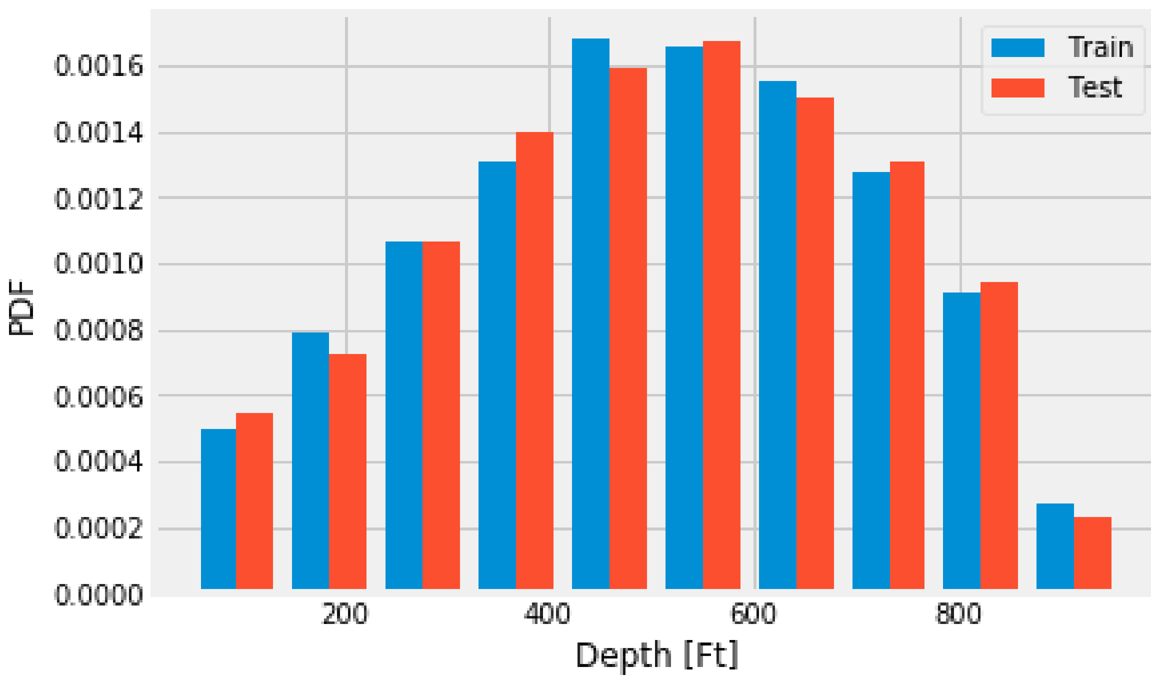


Figure 4: Depth distribution for training and test set (boxplot and histogram plot)

4. Build model

Given a seismic image and its mask (Figure 2), it can see that there are two possible values for each pixel in mask image (white is salt and black is rock without salt). It is become easy to identify the presence of salt when mask image was generated. It needs to create a model which will be able to automatically extract information from seismic image and generate the corresponding mask image.

Therefore, this problem is related to image segmentation which is the process of dividing an image into multiple parts. U-Net is a generic deep-learning solution which could work well with a segmentation task. So, I decided U-Net algorithm to solve the problem in this study.

Generally, the U-net architecture is synonymous with an encoder-decoder architecture. It includes a contracting path and an expansive path, which gives it the U-shaped architecture. The contracting path is a typical convolutional network that consists of repeated application of convolutions, each followed by a rectified linear unit (ReLU) and a max pooling operation. During the contraction, the spatial information is reduced while feature information is increased. The expansive pathway combines the feature and spatial information through a sequence of up-convolutions and concatenations with high-resolution features from the contracting path.

a. Accuracy metric

In this study, the mean average precision at different intersection over union (IoU) thresholds was considered as the accuracy metric. The IoU of a set of object pixels and a set of true object pixels is defined as:

$$IoU(A,B) = \frac{A \cap B}{A \cup B}$$

The metric sweeps over a range of IoU thresholds, at each point calculating an average precision value. The threshold values range from 0.5 to 0.95 with a step size of 0.05. For example, at a threshold of 0.5, a predicted object is considered a "hit" if its intersection over union with a ground truth object is larger than 0.5.

At each threshold value t, a precision value is calculated based on the number of true positives (TP), false negatives (FN), and false positives (FP) resulting from comparing the predicted object to all ground truth objects:

$$\frac{TP(t)}{TP(t)+FP(t)+FN(t)}$$

A true positive is counted when a single predicted object matches a ground truth object with an IoU above the threshold. A false positive indicates a predicted object had no associated ground truth object. A false negative indicates a ground truth object had no associated predicted object.

The average precision of a single image is then computed as the average of the above precision values at each IoU threshold:

$$\frac{1}{|thresholds|}\sum_{t}\frac{TP(t)}{TP(t)+FP(t)+FN(t)}$$

b. Model parameters

Training set was splitted into train and validation set with validation and train ratio of 0.3. The U-net model was built with following information:

Weights initialization: random normal

Learning rate: 1e-4

Optimizer: ‘Adam’

Loss function: binary crossentrpoy

Batch size: 32

Dropout ratio: 0.5

Filter sizes: 3x3 and 2x2

Number of epochs for training: 60

5. Results

After all images in training set were pre-processed and splitted, the U-net algorithm started training my model. Figure 5 indicated the change of accuracy metric (IoU) in respect with number of epoch. Note that 60 epochs were chosen in this study. It was seen that the IoU score increases quickly at initial epochs. After 40 epochs, there was slow increasing of IoU score. In addition, the value of loss function was also presented in Figure 6. Loss is a function of the predicted vs the actual target values. It is a measure of how good a prediction model does in terms of being able to predict the expected outcome. Model weights get updated during training to minimize the loss on the training set. Fundamentally, loss on training set approaches 0 with increasing number of epochs.

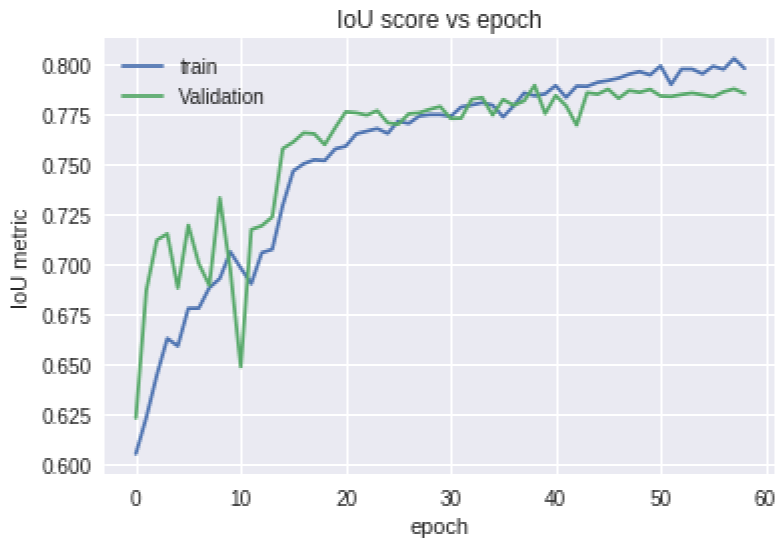


Figure 5: IoU score vs epoch

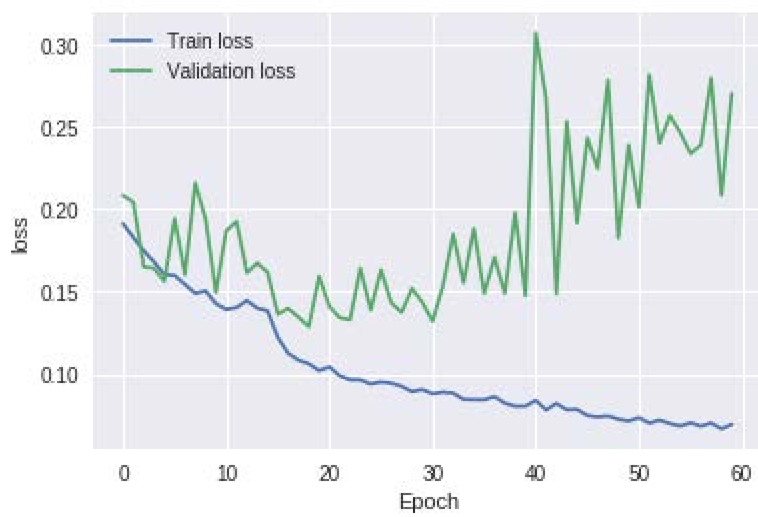


Figure 6: Loss function vs epoch

After training a model, the best IoU scores was recorded and used for predicting the mask images for test set. Let's observe some predicted mask from this model for validation set in Figure 7 (first image is seismic, next is "truth" mask and last is predicted mask). There are a good agreement between the "truth" mask and the predicted one. It indicated that the U-net model is a good selection to solve problem of image processing for seismic identification of subsurface.

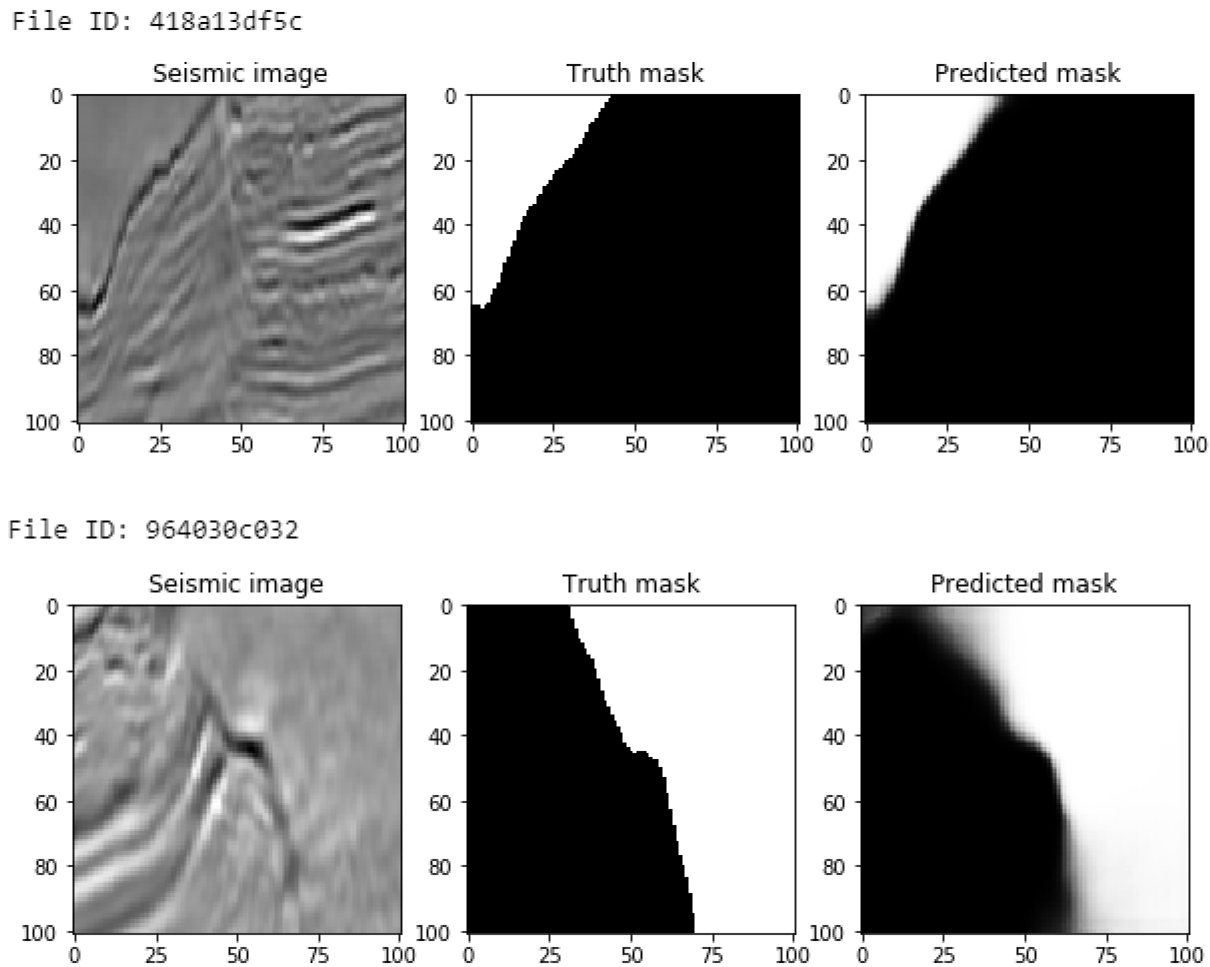


Figure 7: Prediction of the model for validation set

Figure 8 are some predicted masks for test set. Note that white color represents the occurrence of salt. Finally, the rle_mask values for all of test images were generated. The accuracy score of this model for test set is 0.795.

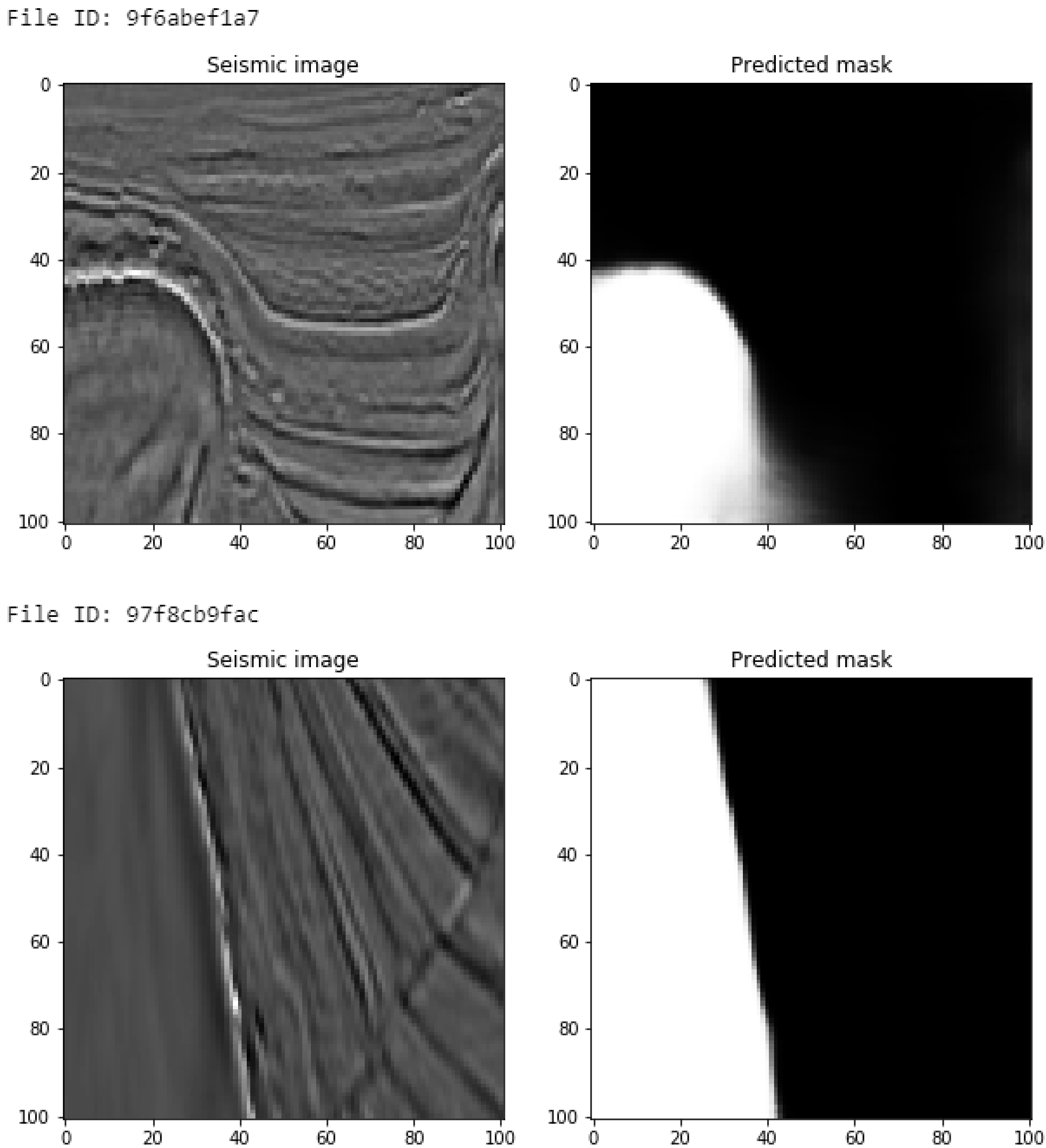


Figure 8: Prediction of mask images for test set

6. Conclusions

In summary, this report was used a convolutional neutral network (U-net) for identifying whether the salt presents in subsurface of reservoir. All data were obtained from kaggle website (provided by TGS company). The modeling results showed the advantages of U-net model for solving image processing problem. This algorithm was proved and applied in many practical application such as biomedical image segmentation, brain image segmentation … I think this model will be better if the amount of data in training set is increased. In this dataset, the ratio of train and test set is little small (around 0.22)

Even though current U-net model in this report works fairy well to predict the mask image, the improvement is still considered. For example, the size of image in data is 101x101. It could be work better if image size was converted into 128x128. Besides, other deep learning models (like ResNet34, fastai…) can be tried to compare the performance.