Segmentation and Classification of Porous Media X-ray Images using Convolutional Neural Networks

Efim Lavrukhin, Gerke Kirill[b,], Sizonenko Timothey[a,b], Korost Dmitry[a,b], Tarasenko Sergey[c]

[a] *Moscow, Russia*
[b] *Some Street*
[c] *Independent Researcher*

Abstract

Write abtract of the paper

Keywords: porous medium, image processing, convolutional neural network, image segmentation, segmentation networks

---

1. Introduction

2. Background and previous works

3. Methodology

3.1. Image processing applications for porous medium analysis

3.2. Convolutional neural networks for segmentation: breif review and benefits

3.3. Classifiers

4. Experiments

4.1. Porous medium specs

4.2. Our model

Key features of U-net architecture:

- Fully-convolutional

*Corresponding author
Email address: `cheshik@.com` (Gerke Kirill)

- Encoder-decoder acrchitecture with skip-connections
- Max pooling layres for comression in encoder part
- Transposed convolutional layers + concatenation with skip-connected encoder feature map for uncompression in decoder part
- Convolutional layers only with 3x3 filters

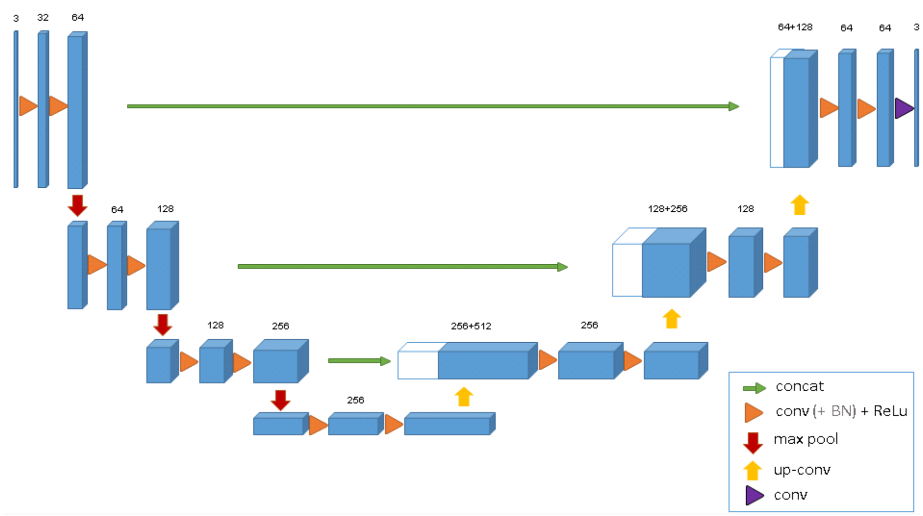

Figure 1: Unet architecture: ToDo

We used Unet architecture with some small features:

- Number of conv filters is multiplied by 32(insted of 64 in original article)
- Padding to conv filters, so network do not compress output
- ELU activation functions

4.3. Learning process

We using 2-d image patches for model training. We chose patch size = 64 and minibatch size = 32, so out model get float 4-d tensor of size $32 \times 1 \times 64 \times 64$.

Pixel intensivity scaled to $[-0.5, 0.5]$ interval. Model returns probabilities for each pixel, model output has size 4096×2(besause on the top of model used 1-d softmax activation), that output could be reshaped to size $32 \times 2 \times 64 \times 64$.

In each epoch only $\frac{1}{5}$ of minibatches used for training, because our trainig set is large enougth and full processing will be very time expensive. We do not use data augmentation for the same reason.

We used combination of cross-entropy and smoothed IOU as loss function:

$$L(x, y) = \frac{1}{N} \sum_{i=1}^{N} \sum_{x,y \in I_i, M_i} \frac{1}{WH} \Big(y \log p(x) + \alpha \log \frac{p(x)y + \varepsilon}{p(x) + y - p(x)y + \varepsilon} \Big) \quad (1)$$

with $\varepsilon = 10^{-5}$.

We used Adam optimizer with initial learning rate $= 10^{-3}$ and multiplicative learning rate decay to 10^{-5} until final epoch.

4.4. Experimantal set-up

We handle image stacks in folowing way:

1. Split each 2-d image to minimal overlapping patches, feed patches to our model and get probabilities for each pixel.
2. Patches with probabilities binarized with some threshold so we get segmentation mask that contains our class labels.
3. 2-d mask for source image is assembled from probabilities patches.

We choose 3 rock types: carbonate, soil and sanstone. We have 4 carbonate, 4 soil and 3 sandstone stacks. We choose one of stacks from each category as trained(carbonate 1, soil 1 and sandstone 1). Then we train 3 models on each trained stack independenlty, 3 models on each pair of trained stacks and one model on both trained stacks. As result we have 7 models to compare.

Each models have training set with equal number of images, so models that trained on single stack type get full stack as input, models that trained on two stack types get a half of eash stack as input and model that trained on all three stack types get third part of stacks.

We measure different metrics(logloss, IOU, accuracy, precision, recall, PR-AUC) with diffent thresholds on each of 11 stacks.

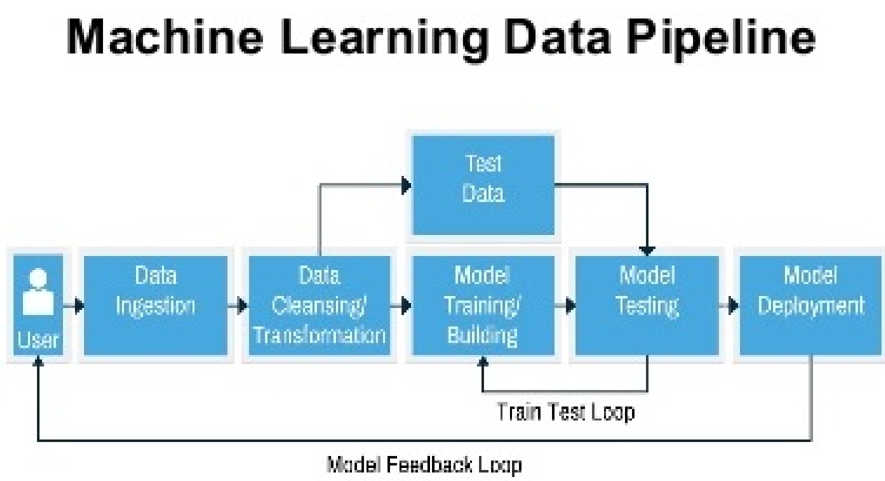

Figure 2: Trainig pipeline: ToDo

5. Results

In out experiments we used 6 metrics: accuracy measures number of right predicted pixels, precision measures number of right predicted positive class pixels relative to number of total emount of predicted positive pixels,s logloss is a metric, that takes into account predicted propapiblities of each pixels, recall measures number of right predicted positive class pixels relative to number of total emount of true positive pixels, PR-AUC is area under precission-recall curve, IOU is relation between intersection of predicted and true positive labels and union of predicted and true positive labels. We measure 6 metrics for each pair of used model and predicted stack.

5.1. Segmentation threshold

At first, we need to determine probabilty threshold in order to get binary segmentation. It seems to be a dificult task, but we traced the dependence of our metric from thresholds and found wide large plateau on this graphics. So we could take any threshold and get same quality of segmentation.

5.2. Visual segmentation results

In this section we show visual results of segmentation. As we can see, our model is uncertaion about border pixels, so amount of false-negative error is much more that amount of false-positive error(false-negative pixels are colored in blue, false-positive in red). Also there is another artifacts: some models classify bright solid clusters as pores(because there is no such clusters in trainig set of that models).

5.3. Numerical segmentation results

We create large table with measurements, so we could draw some conclusion.

- Specific models that trained on some stack type shows excelent perfomance on stacks from same type, but could do bad predictions on stacks from another type.
- Aggregate models show good perfomance on all stacks, so their outperform specific models in average, but lose on stacks with type that native for specific model.

6. Discussion

Overview of results and applicability of the results.

7. Conclusion

General conclusion and future work.

References

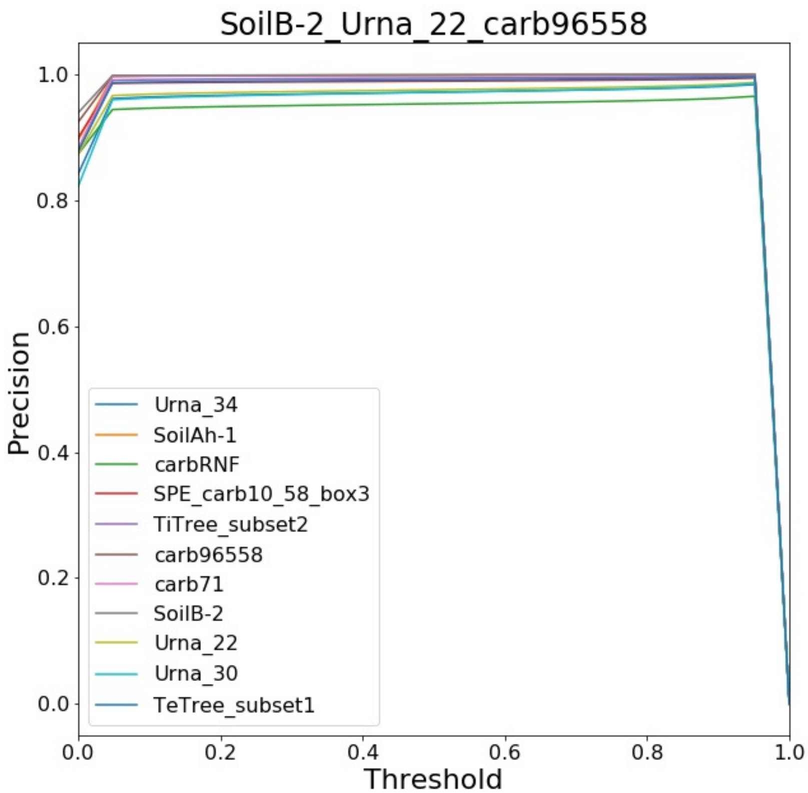

Figure 3: Dependency of metrics from threshold

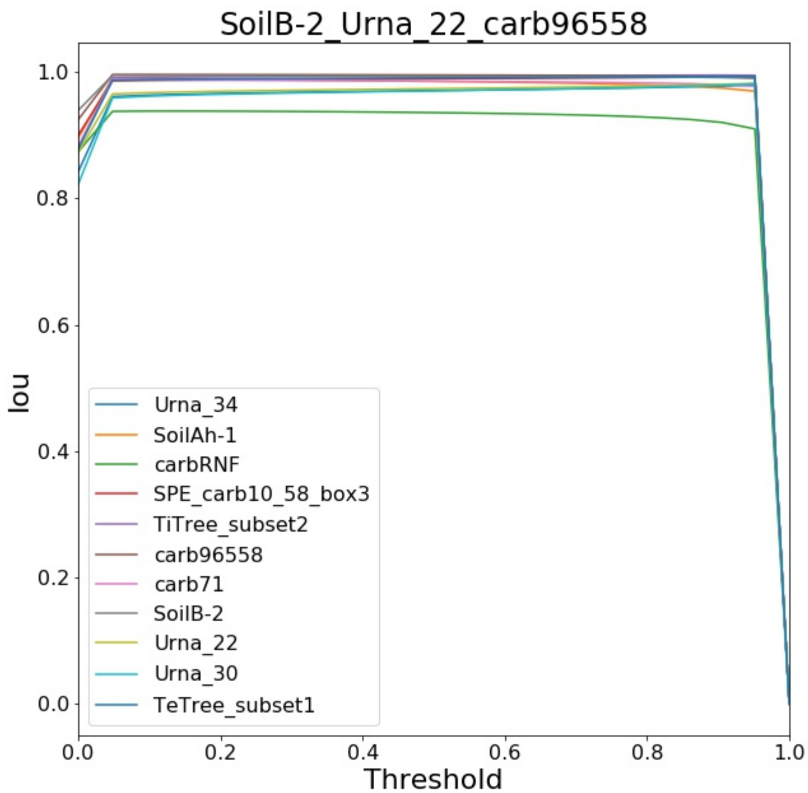

Figure 4: Dependency of metrics from threshold

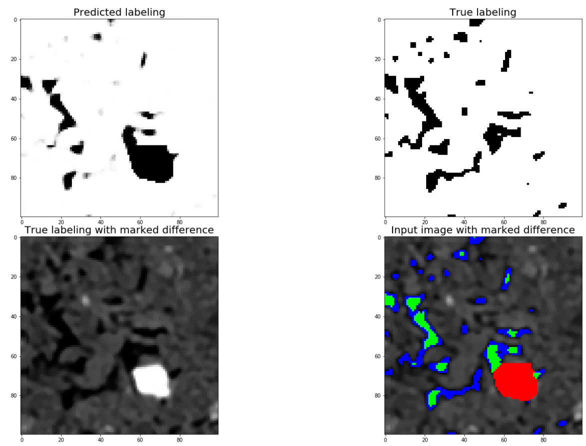

Figure 5: Model trained on carb96558 and predicts Urna22

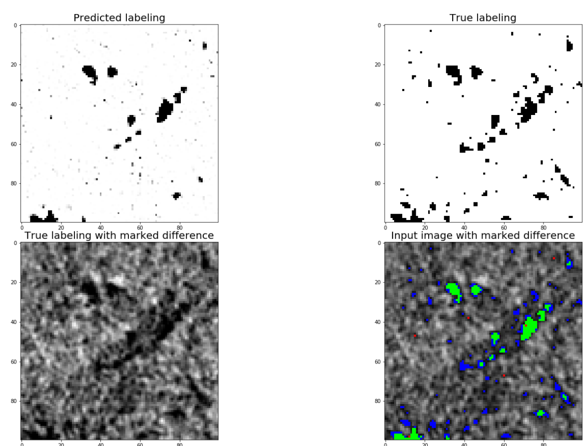

Figure 6: Model trained on carb96558 and predicts carbRNF, good prediction

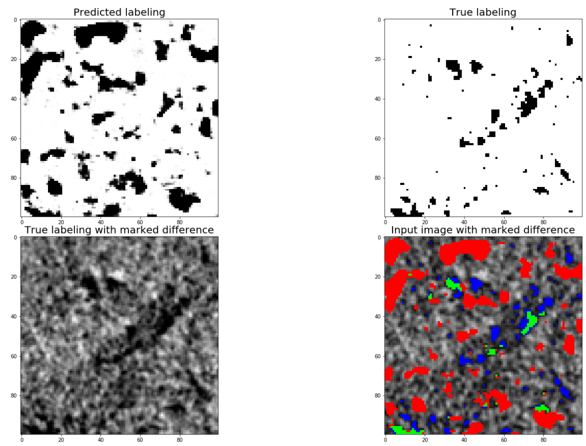

Figure 7: Model trained on SoilB-2 and predicts carbRNF, bad prediction

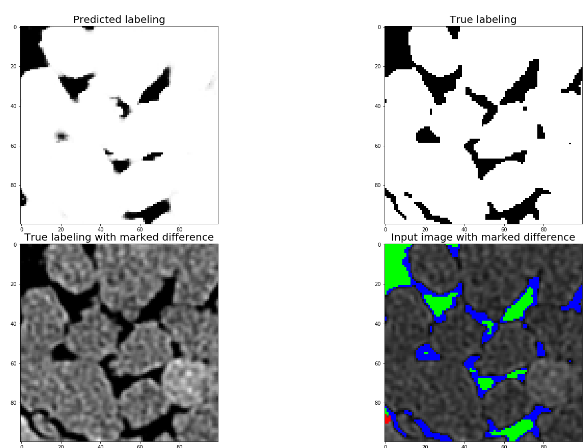

Figure 8: Model trained on carb96558 and SoilB-2 and predicts Urna34, unconfident model