Image Segmentation Using Deep Learning

Tensorflow and Keras Implementation in Python

Ahad Momin

Hildebrand Department of Petroleum and Geosystems Engineering,

The University of Texas at Austin

Professional Research Practicum, University of New South Wales Sydney

Supervisors: Dr. Masa Prodanovic and Dr. Christoph Arns

May 19, 2020

Codes and Workflow:

https://github.com/AhadMomin/2D-and-3D-UNet-for-Image-Segmentation-in-Digital-Rock-Physics/tree/master/workflows

TABLE OF CONTENTS

INTRODUCTION

The deep learning methods in the field of Digital Rock Petrophysics (DRP) are quickly becoming an attractive technology to process natural rock images. These digitized images are used in many engineering calculations, including those for fluid properties, transport mechanics, and reservoir characterization. In a standard DRP workflow, segmentation is an essential step to characterize several distinct phases and pore space of the rock sample after the 3D image is acquired and reconstructed from pore-scale X-ray tomographic imaging. The 3D segmented image is then used in numerical analysis of fluid flow properties, transport mechanics, and other properties of interest.

In DRP, there are image segmentation techniques ranging from simple thresholding to multi marker-based watershed and active contours algorithms. The methods are substantially reviewed in [8]. There are also many software's which can perform segmentation using these methods, such as Mango. Mango is a software developed by the Australian National University to perform segmentation using watershed and active contour algorithms, but it is an expensive tool. However, all methods are judgment-based and require user input which introduces bias. These conventional methods are also time consuming, require expertise, and make it difficult to differentiate distinct phases with similar intensities and colors. The binary segmentation (pore and mineral) is usually considered, but it strongly impacts porous media analysis. Hence, multi-phase segmentation is usually required.

Deep learning convolution neural networks (CNN) have proved successful in image segmentation in the biomedical field [3,9] and in DRP [7,5,11]. The benefit of CNN is its ability to learn from patterns, textures, and features instead of color differences. Recent publications have proven considerable segmentation accuracy using 2D and 3D *SegNet*, *UNe*t, and *U-ResNet* CNN architectures [7,5,11].

The current challenges with CNN are the lack to ground truth (GR) for training. The labels (segmented images) are manually obtained from conventional methods which already involve user bias. This issue is still unaddressed and not within the scope of this project. CNN usually requires large datasets and, due to usually limited datasets available, there can be a need for data augmentation. The data itself requires preprocessing to make sure the results are consistent and model sensitivities are controlled.

In this project, I have built and tested 2D and 3D *UNet*, *U-ResNet*, and *ResUnet* using Keras and Tensorflow implementation in Python and trained with datasets obtained from Professor Christoph Arns. Intersection over union (IOU) was used as a metric to see the performance of each model on each phase of the predicted image. For measuring the physical accuracy of the models, the Euler number for each phase was computed. The 2D *UNe*t, *U-ResNet*, and *ResUNet* are trained on 134 million pixels and 3D *UNet*, *U-ResNet* and *ResUNet* are trained on 37 million pixels. Due to parallel computing issues with the Tensorflow package, the image size to train on 3D models was reduced. From multiple tests using different datasets, the models are working with about 99% accuracy on validation dataset with 2D while obtaining reasonable accuracy with 3D due to time and computing constraints. There is some significant prediction error with running 3D models on phase 2 and phase 4 of our current dataset which will be addressed and fixed in the near future. Data augmentation was not considered but it would have been helpful with successful parallel computation. However, the models can be used with cloud and we can train with larger datasets. Mango software was used to obtain segmented labels with preprocessed anisotropic diffusion filter and edge enhancement applied to the images. This makes the model prone to user bias. However, one simple method of addressing user bias in multi-mineral segmentation is to simply train any one of the CNN to large and varied datasets enough to average out bias in segmentation [11].

The models will be very useful in deep learning research in DRP and can ultimately be trained with diverse and large data to be used as tools for segmentation. There are several advance models that have been developed today in the medical field which can be applied to DRP for more efficient and faster computations. The datasets used in this project from Professor Christoph Arns are available on Digital Rock Portal to help researchers to continue work and collaborate. This project is supervised by both Professor Christoph Arns and Professor Masa Prodanovic.

The models were initially made from scratch and trained and tested using a benthimer sandstone dataset from the Digital Rock Portal with multi-phase labels. After receiving feedback from Javier Santos and extensive research, the models were improved to load the data properly and obtain better accuracy and results. The improved models were then tested on the images obtained from Professor Arns and significant results were obtained with mean IOU and accuracy of about 99% on the validation dataset. For 3D models, the phase 2 and phase 4 predictions were significantly different. Some interesting results showed how 2D predictions had higher IOU but the physical accuracy, like Euler number, was significantly different. In all measures, the *ResUNet* model seemed to perform the best in both 2D and 3D over other models as well as the *U-ResNet* for 3D. The biggest challenge in this process was to train the models and work remotely in Austin on the *ct00* computer in Sydney. Because of parallel computing and time constraints, the image size to train 3D models was reduced. However, the models can be trained on large image sizes if they are compiled successfully on all GPUs. The GPUs used in this project were three Nvidia Titan Xp processing units.

Motivation

CNNs have played significant roles in DRP from predicting permeability and other porous media properties [2,10]. In recent publications, various networks have been tested on different samples [7,5,11]. However, there are only few works that have been performed with 3D segmentation in DRP and data augmentation has been the necessary step in most cases due to computational limitations. However, we have enough data with enough diversity to train the network without any data augmentation. The computational resource was available, but there were constraints due to some technical issues with remote connection and Tensorflow API. However, I was still able to train the network and obtain results for quality assessment. Usually, 2D slice wise segmentation with CNN gives considerably good results but inherently fails to capture context in adjacent slices, most importantly connectivity, which is important information for prediction of segmentation maps. 3D CNNs address this by capturing the entire volumetric information with 3D kernels and thus increase the performance of the predictions.

Although the generalization capability of trained CNN is difficult question to address, the trained models have shown great results with unseen images and can be further trained and used for further research into developing a segmentation tool. Nevertheless, if the models are trained enough with large and diverse datasets with continuously newly developed segmentation CNNs as in medical field, such as *HRNetV2* [15], using controlled parameters, this could result in a very efficient 2D and 3D segmentation tool.

DATASETS AND WORKFLOW

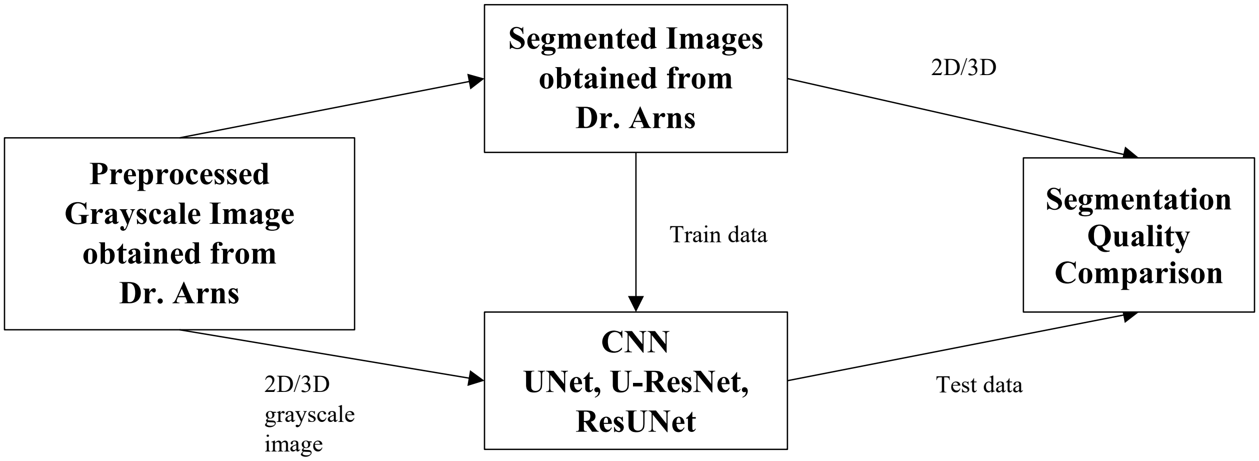

Segmented Images obtained from Dr. Arns
2D/3D
Preprocessed Grayscale Image obtained from Dr. Arns
Train data
Segmentation Quality Comparison
CNN UNet, U-ResNet, ResUNet
2D/3D grayscale image
Test data

Fig. 1 – Deep learning workflow

The workflow of this project is to use grayscale and segmented data from Professor Christoph Arns and split it into testing and training. The CNN model is then trained and tested for segmentation quality assessment. My intention is to establish a comparison between *UNet*, *U-ResNet*, and *ResUNet* CNNs for each 2D and 3D to see how each network performs and how generalized the model becomes after it is trained. The gray scale images are already preprocessed with anisotropic diffusion and edge enhancement. The workflow is shown in Figure 1.

The data that is used in this model consist of Castlegate, Leopard with 4 phases and Bentheimer with 3 phases. The 2D *UNet*, *U-ResNet*, and *ResUNet* are fed with total of 2,400 512^2 images and 3D *UNet*, *U-ResNet*, and *ResUNet* with total of 18 128^3 images. For 2D *UNet* and *U-ResNet*, the grayscale images are converted from netcdf files to tiff file format and then applied with min-max normalization to set all pixel values between 0 to 1. The segmented phases are divided into 4 categories and converted into binary. For 2D *UNet*, *U-ResNet*, and *ResUNet*, the gray scale and their constituent segmented images as well as the 4 separated phases are shown below for both 2D and 3D models in Figures 2 and 3.

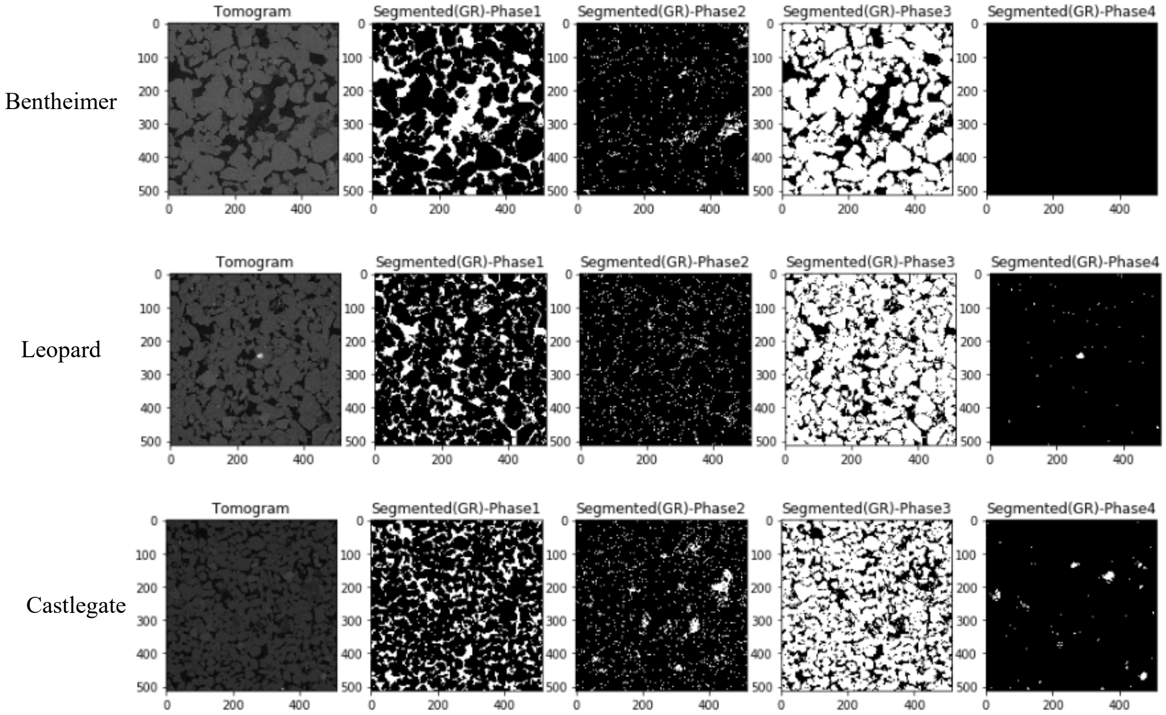


Fig. 2 – 512^2 images with 4 categorical phases used in 2D UNet, U-ResNet, and ResUNet

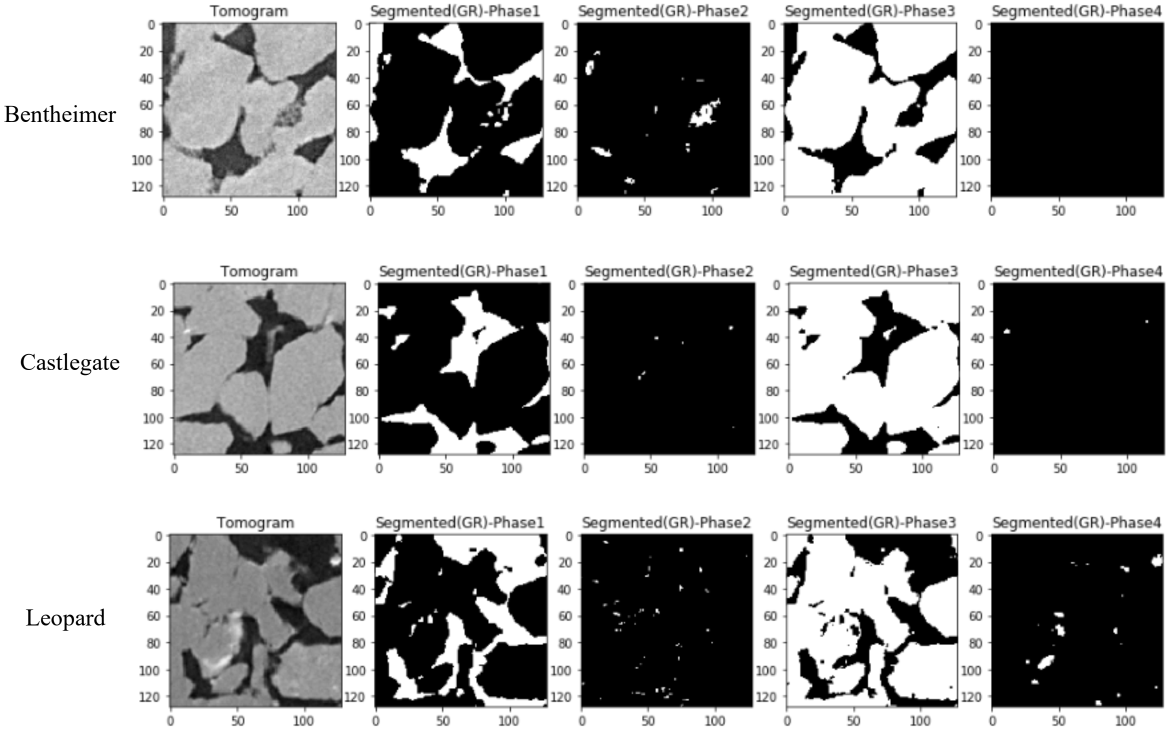


Fig. 3 – 128^3 images with 4 categorical phases used in 3D UNet, U-ResNet, and ResUNet

The dataset for 2D *UNet*, *U-ResNet*, and *ResUNet* is split into 80% training and 20% in validation. The dataset for 3D *UNet*, *U-ResNet*, and *ResUNet* is split into 90% training and 10% (2 images: Leopard and Castlegate) in validation. The tomograms are converted to float 32 and segmented images are 8 bits. The testing data of unseen Bentheimer, Leopard and Castlegate of same sizes are used with a total of 10 images for 2D and 2 images for 3D for testing evaluation. The 2D models are tested on 2 images (Castlegate and Leopard) and 3D models are also tested on 2 images (Benthimer and Castlegate).

Balanced weights are applied on each class to obtain balanced distribution of labels, which produces weights to equally penalize under or over-represented classes in the training set.

CNN ARCHITECTURES

Basics

CNNs are a part of deep learning methods and are widely used in image characterization problems. The convolution layers extract important features from the input layers. These convolution layers are integrated in a neural network which helps to translate features obtained from the previous layers to the given output phases. The CNN networks used in this project are *UNet*, *U-ResNet*, and *ResUNet* for both 2D and 3D image segmentation. The basic layers in CNN are

1) Input Layer: Grayscale images data which is inputted in the CNN.
2) Convolutional Layer: Input image is convolved with filters to generate new maps.
3) ReLU Layer: Activation function for convolution to introduce nonlinearity in the system. The ReLU layer applies the function $f(x) = max(0, x)$ to all values in the input volume. The logic behind ReLU is that this layer changes all the negative values to zero.
4) Max Pooling Layer: Down-sampling layer which reduces image size by summarizing data by choosing local maximum sliding window across feature maps.
5) Up Sample and Transpose: Up-sampling layers which increase the image size with concatenation from saved feature maps during max pooling.
6) Softmax Layer: Another activation function which produces a class-by-class probability distribution such that the total sum of outputs is equal to one.

1) UNet

I adopted *UNet* from biomedical segmentation works. In most segmentation tasks, basic *UNet* is adapted and modified. The 2D and 3D *UNet* network used in the project is illustrated in Figures 4 and 5. The network is developed with encoding and decoding process. The encoding consists of input layer, convolution layer followed by ReLU activation, batch Normalization, and max pooling operation. The image size is reduced in the process while the filters (feature channels) are doubled. The decoding consists of up convolution transpose layer, convolution layer followed by ReLU activation, batch Normalization and softmax activation layer before the output layer. The *UNet* transfers the entire encoding feature map (before max pooling operation) to decoding process to help up sample back to original image size; this process is called concatenation. The *UNet* network is described in [3,9].

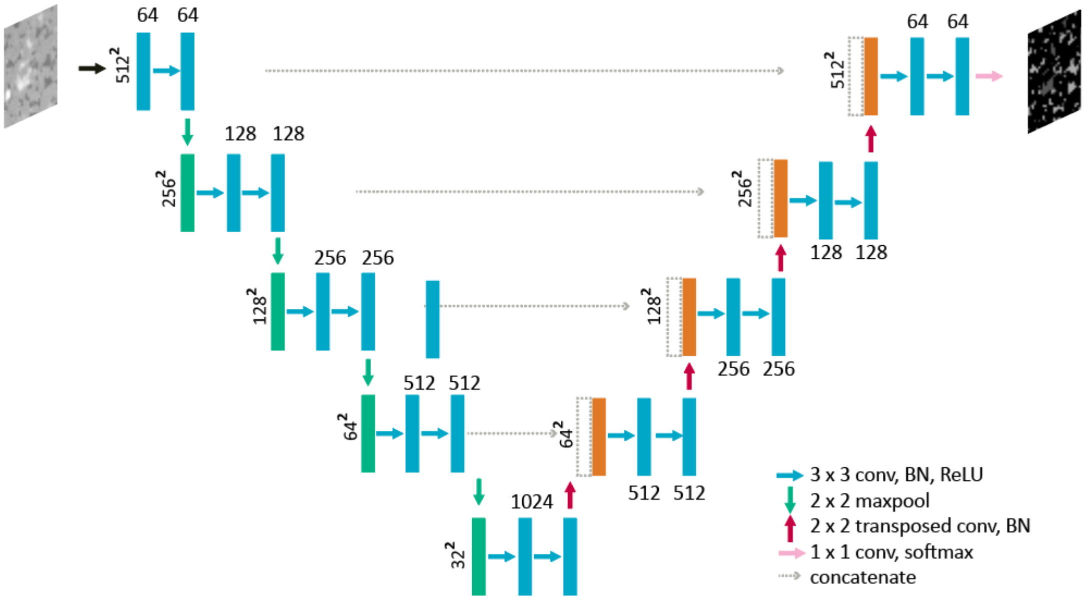


Fig. 4 – 2D UNet Architecture

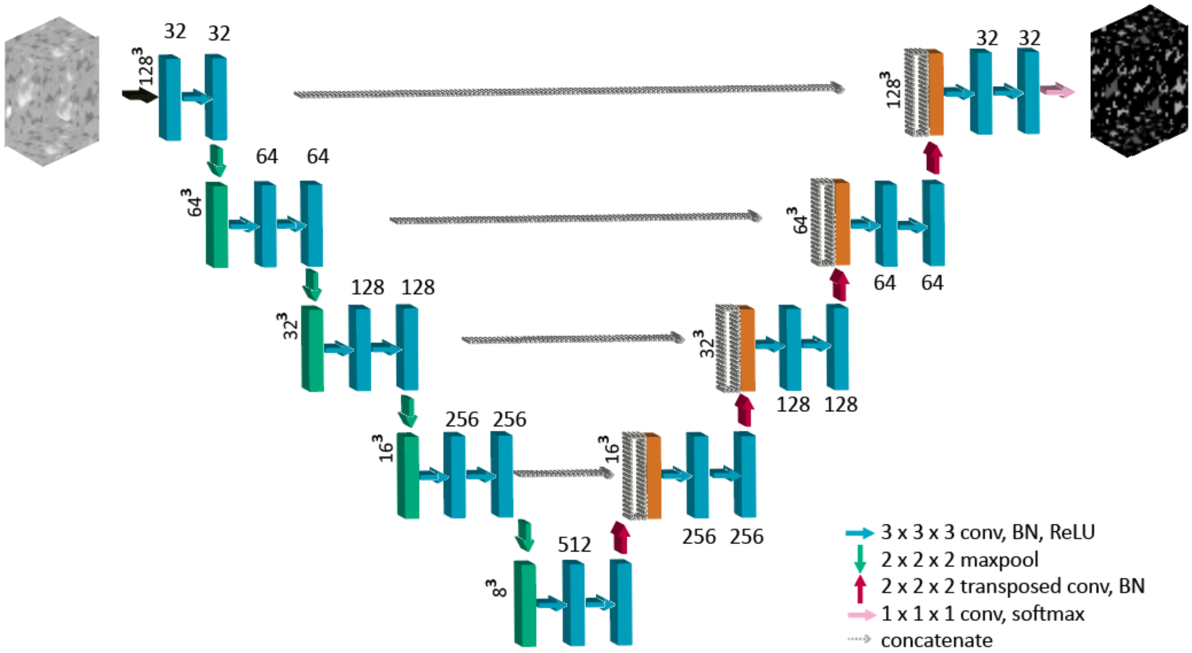


Fig. 5 – 3D UNet Architecture

2) U-ResNet

U-Resnet is a hybrid network obtained from *ResNet* and *UNet.* It is a combination of the short and long skip connections and concatenation that permit the networks to scale well with increasing complexity. There are three convolution blocks applied in each layer with residual convolution block added after first convolution before max pooling to make the network deep. A recent publication [11] has shown that *U-ResNet* has outperformed other networks. The 2D and 3D *U-ResNet* network is illustrated in Figures 6 and figure 7. The *U-ResNet* network is described in [11].

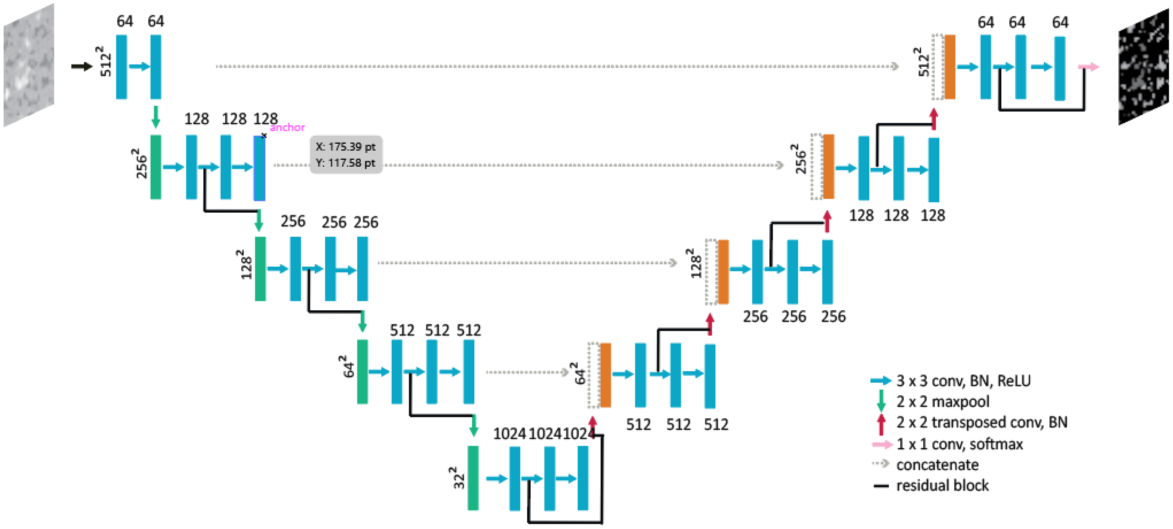


Fig. 6 – 2D U-ResNet Architecture

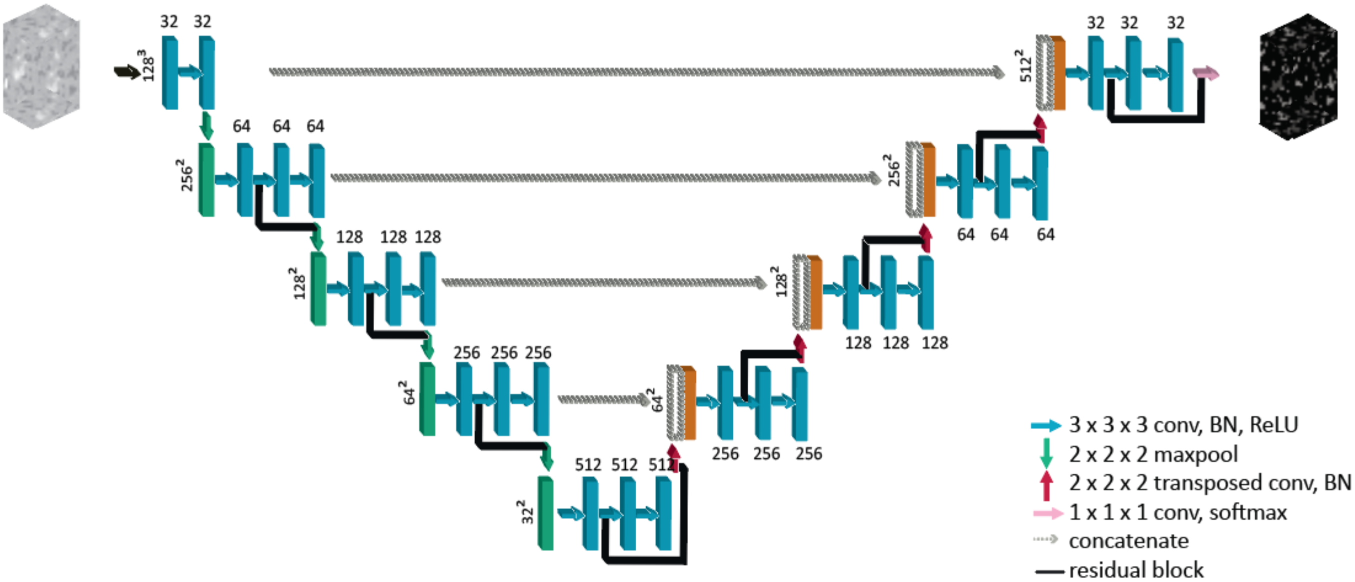


Fig. 7 – 3D U-ResNet Architecture

3) ResUNet

ResUNet is a hybrid network obtained from U-*ResNet* and *UNet. ResUNet* is also a combination of long skip connections and concatenation that permit the network to scale well with increasing complexity. The difference is that there are two convolution blocks in one layer with batch normalization and activation applied before the convolution. There is no max pooling layer but an addition layer where the residual convolution block is added from input. Like *UNet*, *ResUNet* transfers the entire encoding feature map from the addition layer to decoding process to help up sample back to the original image size. The network is shorter compared to *UNet* and *U-ResNet* used in this project. Figures 8 and 9 shows 2D and 3D *ResUNet* used in this study.

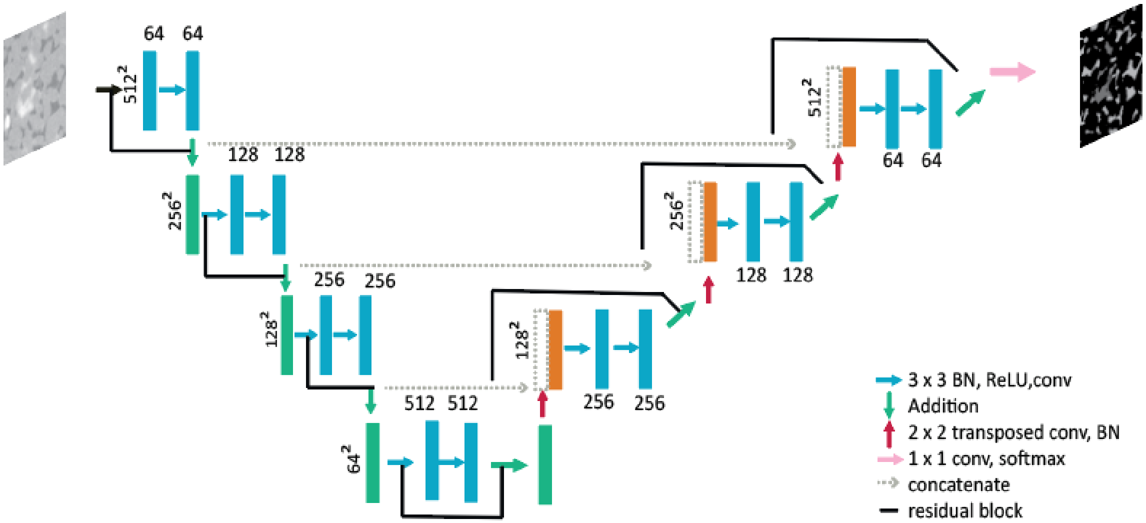


Fig. 8 – 2D ResNet Architecture

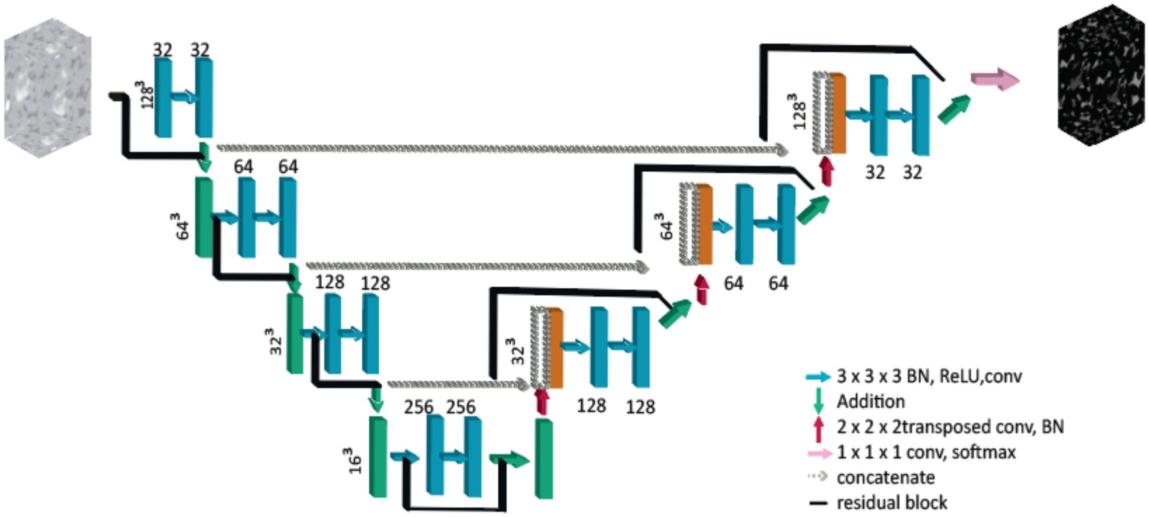


Fig. 9 – 3D U-ResNet Architecture

Loss function, Activation Layer, and Parameters

Cross entropy and softmax layer are used as loss function and last activation layer respectively for all 2D and 3D models.

Softmax squashes a vector in the range (0, 1) and all the resulting elements add up to 1. It is applied to the output scores (si). As elements represent a class or phase C, they can be interpreted as class probabilities. Cross entropy XE is calculated between the one-hot mineral label vectors, m and f(si). Softmax and cross entropy loss is computed using Equations 1 and 2, respectively. In Keras, it is referred to ‘binary_crossentropy.’

$$f(si) = \frac{e^{sj}}{\sum_{j}^{C} e^{sj}} \qquad \textit{Eq. 1}$$

$$Cross\ entropy\ loss\ = \sum_{i=1}^{N} mlog(f(si)) \qquad \textit{Eq. 2}$$

For 2D I used 50 epochs with batch size of 6 and for 3D I used 375 epochs with batch size of 1. The learning rate is reduced if the accuracy is not reduced and the best model is saved with maximized loss.

Metrics and Accuracy Measures

For metric, I used intersection of union (IOU) for all 2D and 3D models.

Mean Intersection-Over-Union is a common evaluation metric for semantic image segmentation, which first computes the IOU for each semantic class and then computes the average over classes. IOU is defined in Equation 3.

$$IOU = \frac{(true\ positive)}{(true\ positive + false\ positive + false\ negative)} \qquad Eq.\ 3$$

True-positives are those pixels that belong to the class and are correctly predicted as the class, false-negatives are those pixels that belong to the class but are incorrectly predicted as a different class, and false-positives are those pixels that belong to a different class but are predicted as the class. The IOU score is calculated for each class separately and then averaged over all classes to provide a global mean IOU score of our semantic segmentation prediction. The model during training minimizes the false positive and maximizes the true positive.

The IOU is a value between 0 and 1, where a larger value indicates a more accurate segmentation. The mean IOU is then the mean value across all the classes in the dataset.

For physical accuracy measure, I am calculating Euler number which is a measure of the topology of an image. It is defined as the total number of objects in the image minus the number of holes in those objects. I am using 8 connected neighborhoods. Euler number is calculated for each phase and a comparison is conducted between all phases of each models on different sandstones.

RESULTS AND SEGMENTATION QUALITY COMPARISON

The prediction results were generated from all models for validation and testing dataset. The IOU and Euler number are compared in validation dataset and testing dataset to perform the segmentation quality assessment. The 3D models had significant problems as the output was non-binary image for all phases and phase 2 and phase 4 were specifically had lower IOU which skewed the mean IOU. To tackle that for now I used reasonable thresholding to make images binary to compute Euler number. Table 1 shows the thresholds used for all 3D model for prediction values on both validation and testing dataset.

Figures 11-15 show the visuals of prediction and ground truth generated with the corresponding histogram for all models for validation dataset. Figures 18-23 show the visuals of prediction and ground truth generated with the corresponding histogram for all models for testing dataset Figures 16-17 show the Euler number comparison for all models for each phase in each sandstone on validation dataset. Figures 24-25 show the Euler number comparison for all models for each phase in each sandstone on testing dataset. Tables 2 and 3 shows the IOU comparison of all models on validation and testing datasets, respectively.

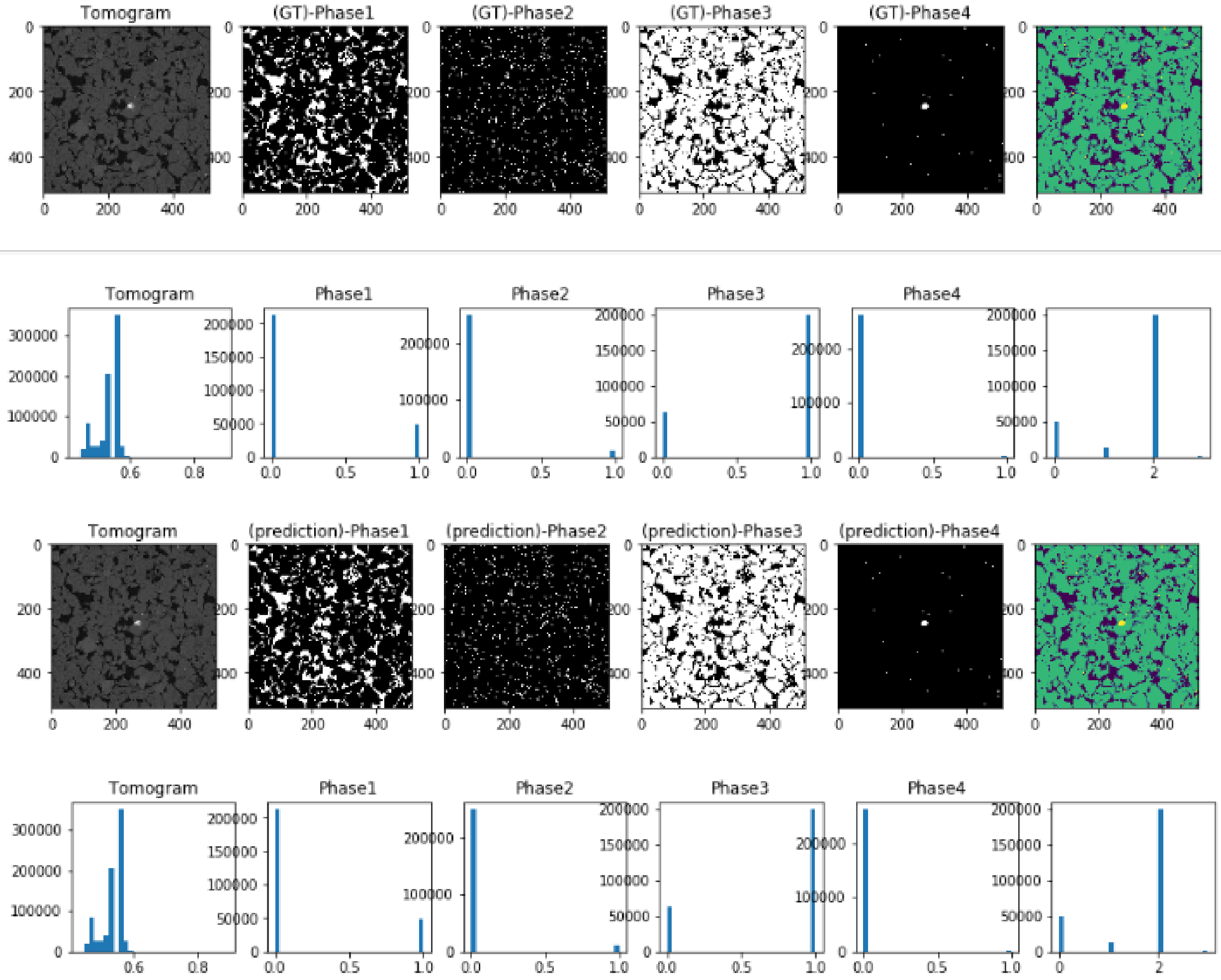
Phase	Threshold
1	> 0.5999
2	> 0.0699
3	> 0.5
4	> 0.029599

Table 1 – 3D Prediction Value Thresholds

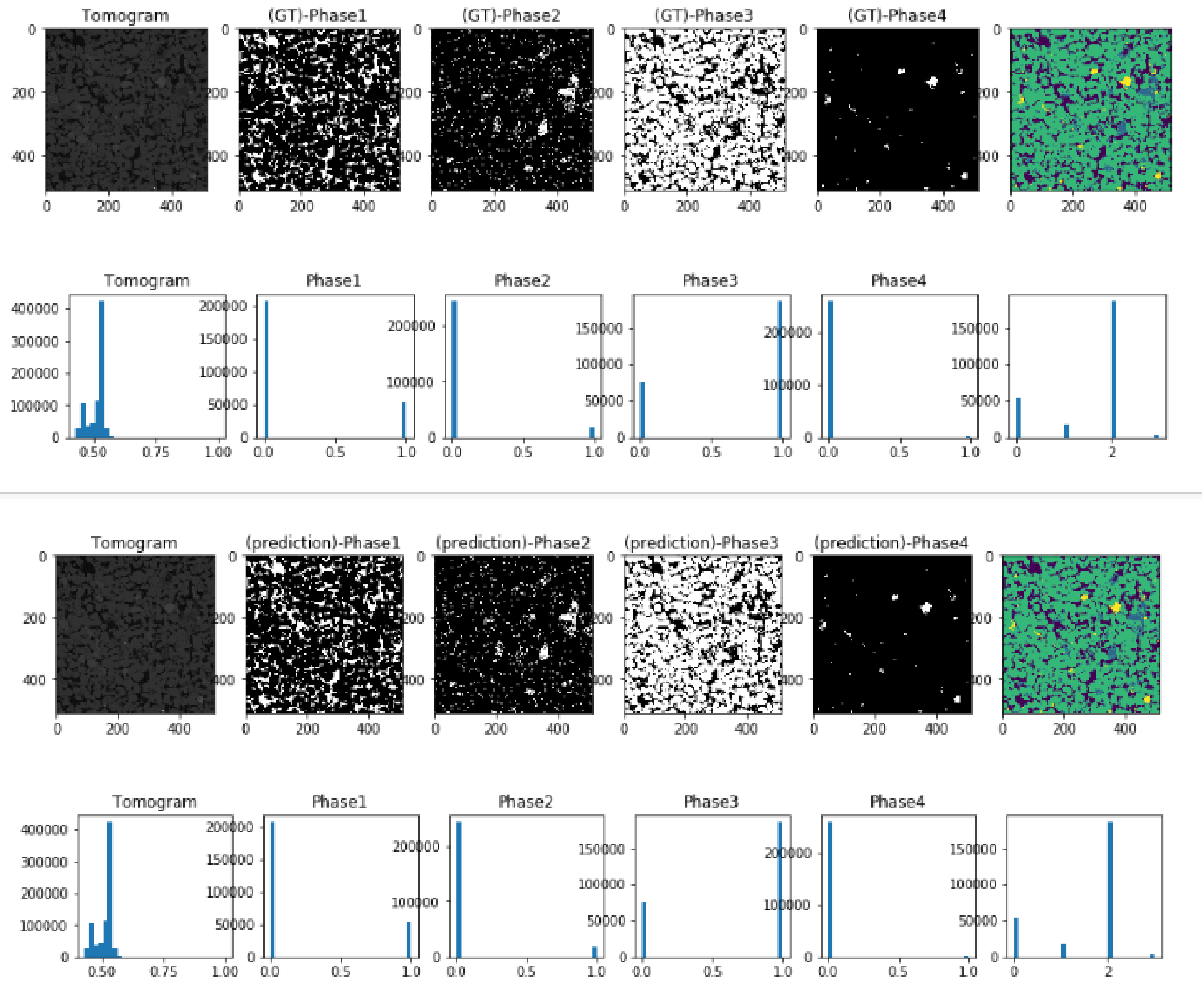
Validation Dataset

2D U-ResNet

Leopard



Castlegate


Tomogram
(GT)-Phase1
(GT)-Phase2
(GT)-Phase3
(GT)-Phase4
0
200
400
Tomogram
Phase1
Phase2
Phase3
Phase4
400000
300000
200000
100000
150000
50000
0.50
0.75
1.00
0.0
0.5
1.0
2
(prediction)-Phase1
(prediction)-Phase2
(prediction)-Phase3
(prediction)-Phase4

Bentheimer

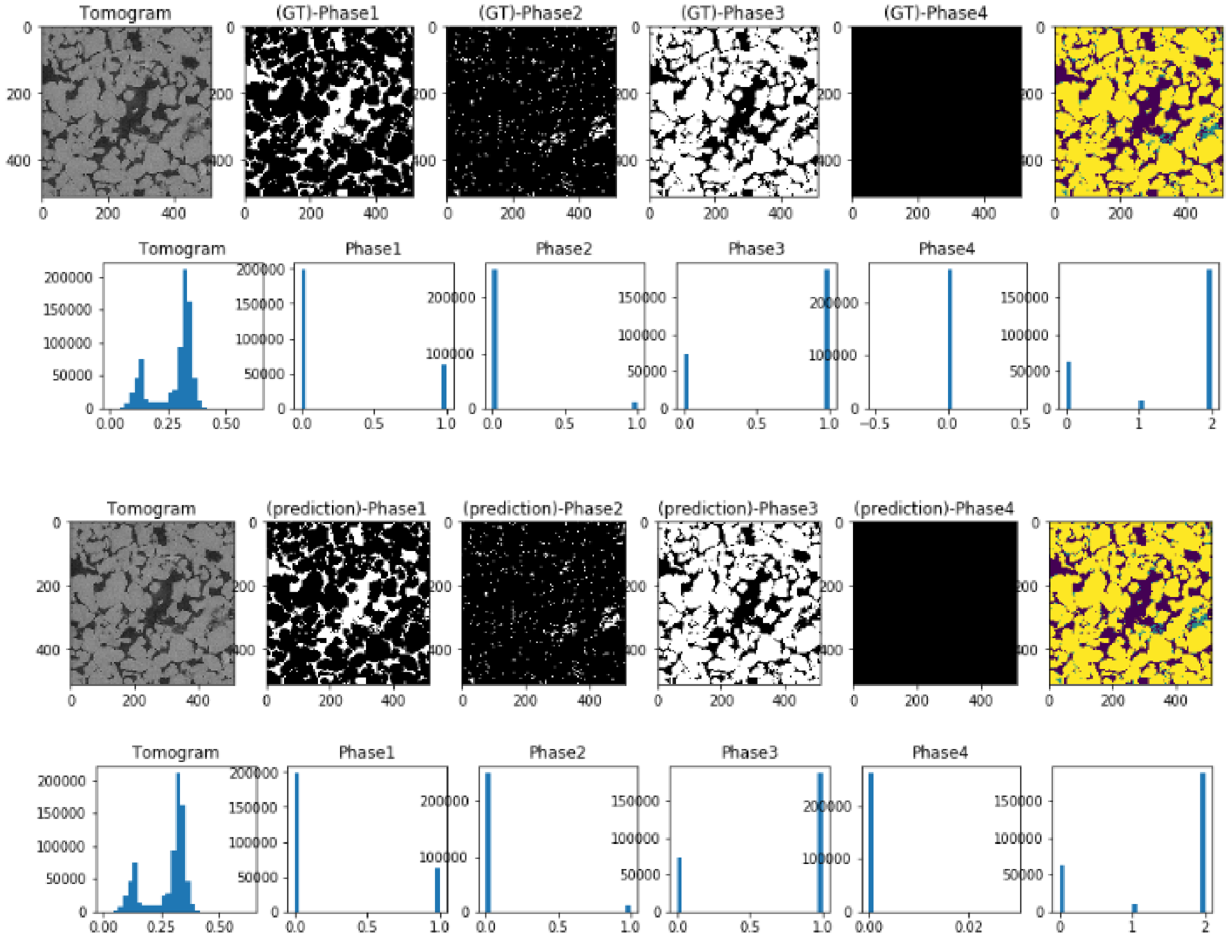
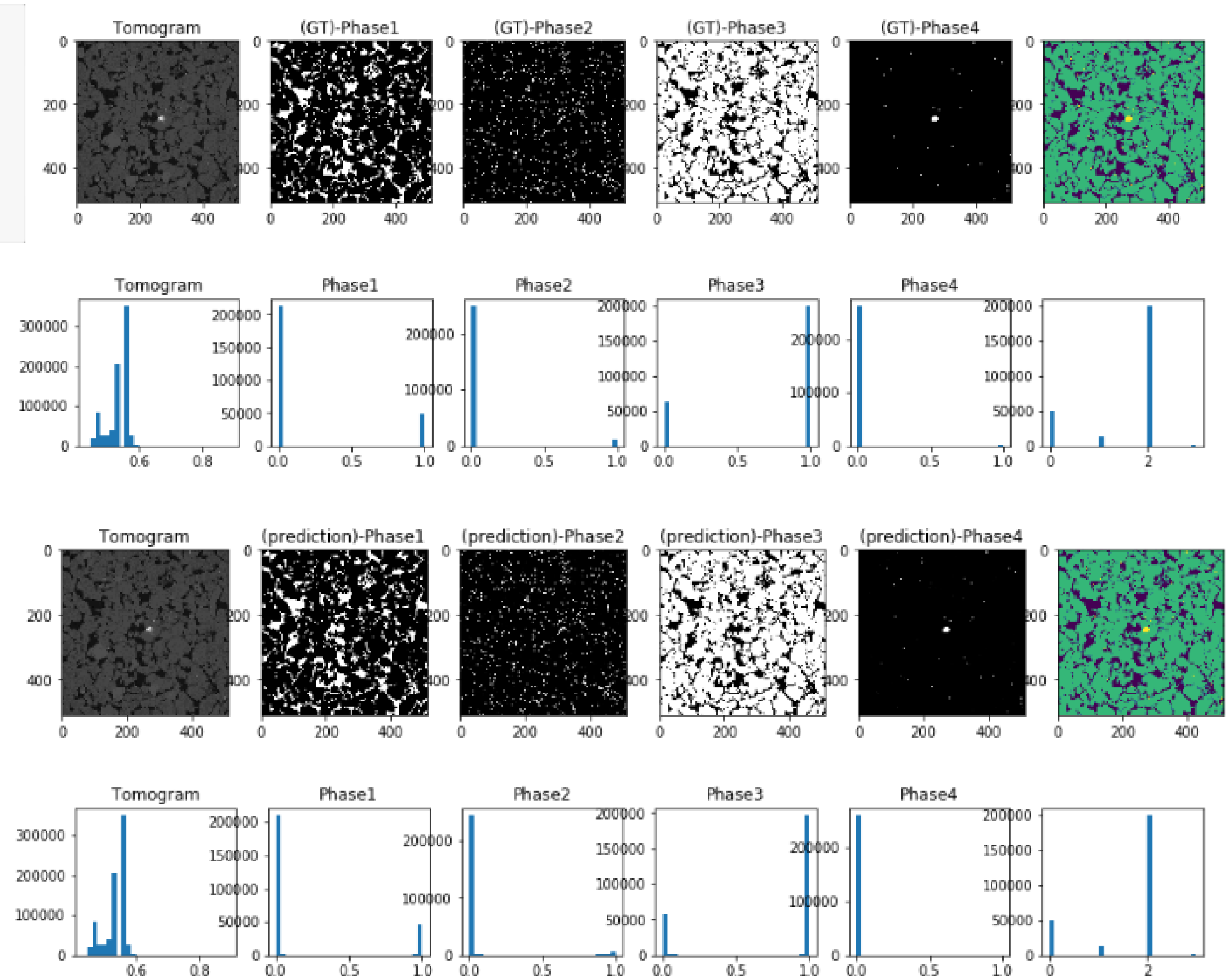


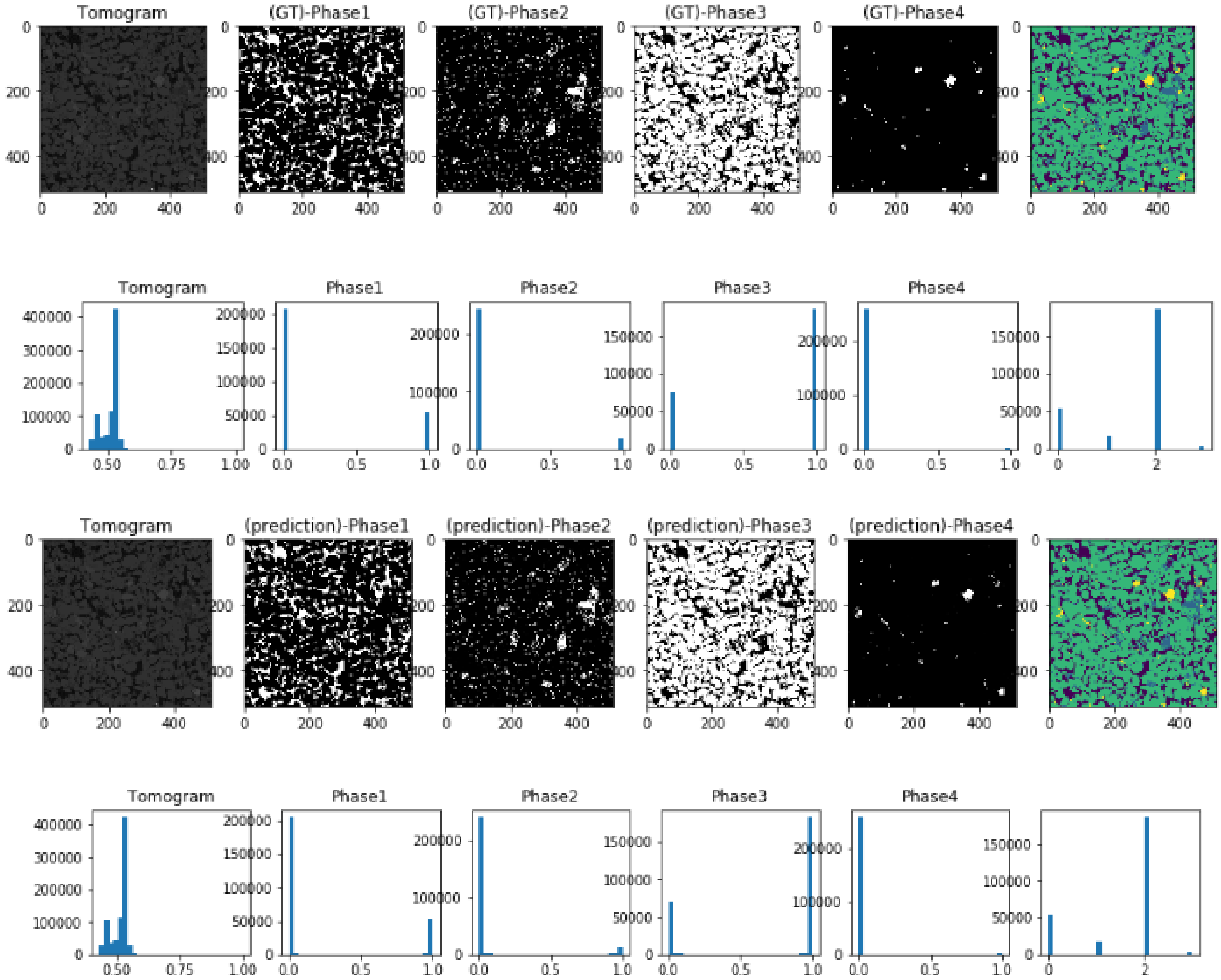
Fig. 10 – 2D U-ResNet Prediction vs Ground Truth

2D UNet

Leopard



Castlegate


Tomogram
(GT)-Phase1
(GT)-Phase2
(GT)-Phase3
(GT)-Phase4
0
200
400
Tomogram
Phase1
Phase2
Phase3
Phase4
400000
300000
200000
100000
150000
50000
0.50
0.75
1.00
0.0
0.5
1.0
2
Tomogram
(prediction)-Phase1
(prediction)-Phase2
(prediction)-Phase3
(prediction)-Phase4
Tomogram
Phase1
Phase2
Phase3
Phase4

Bentheimer

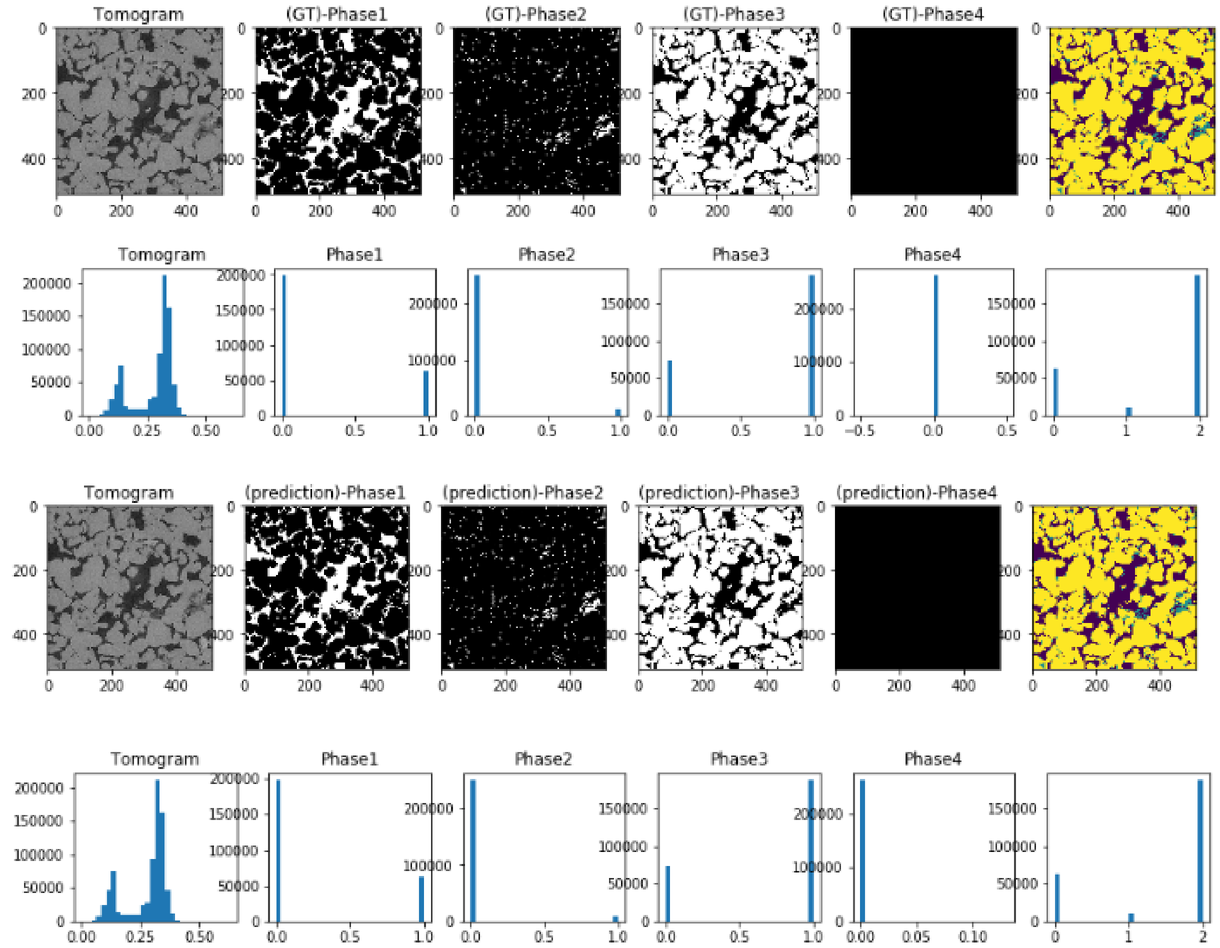
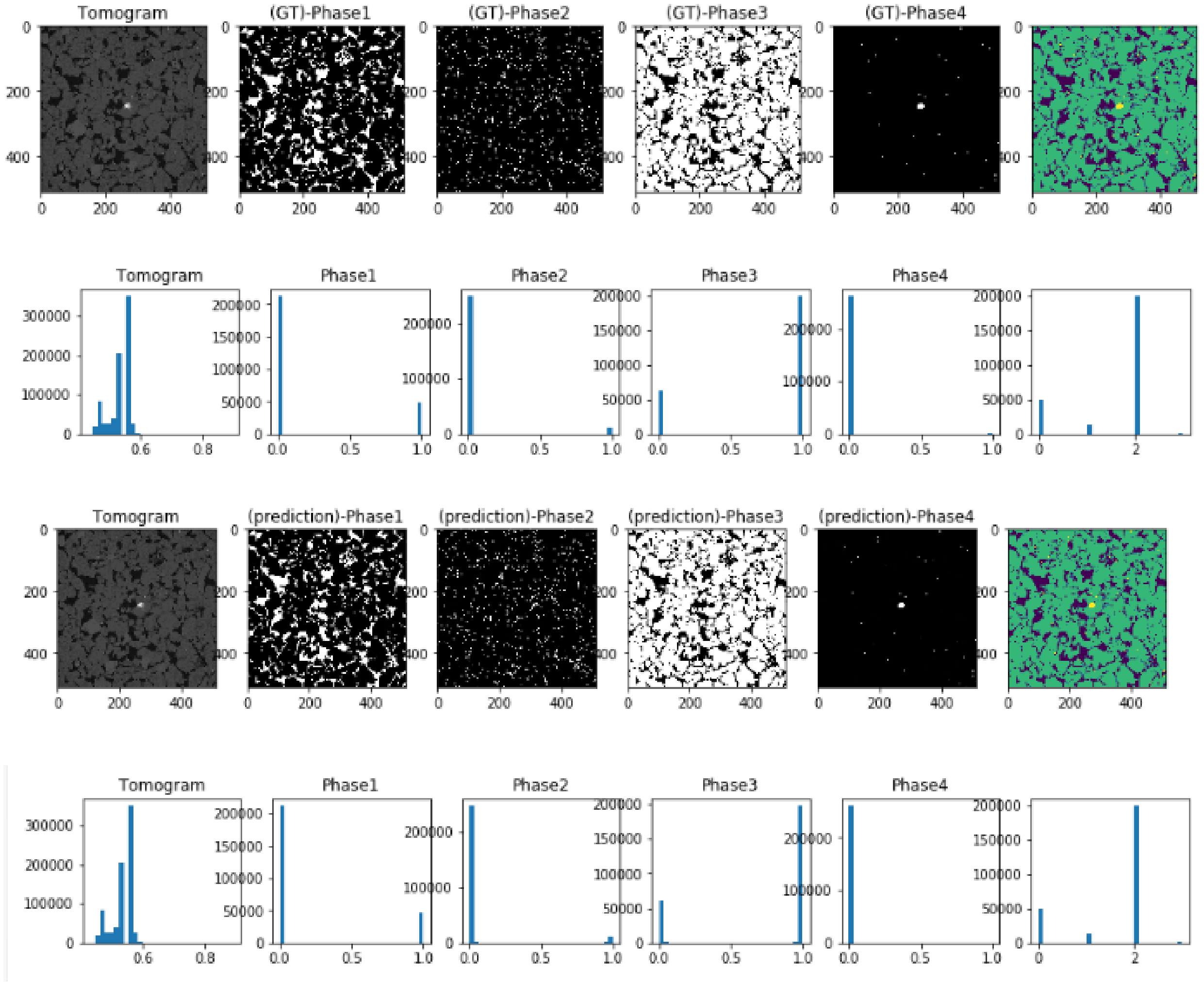


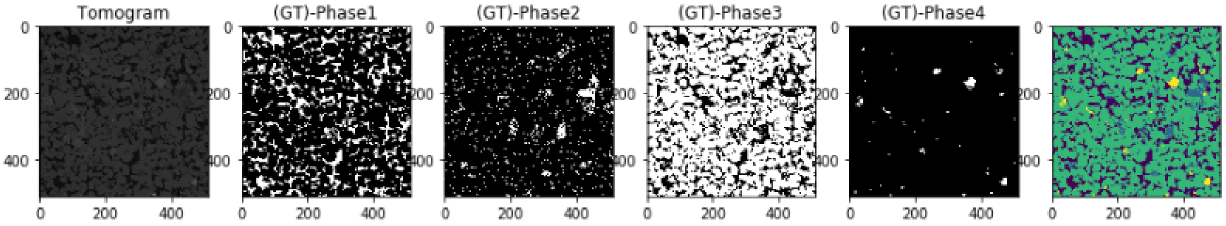
Fig. 11 – 2D UNet Prediction vs Ground Truth

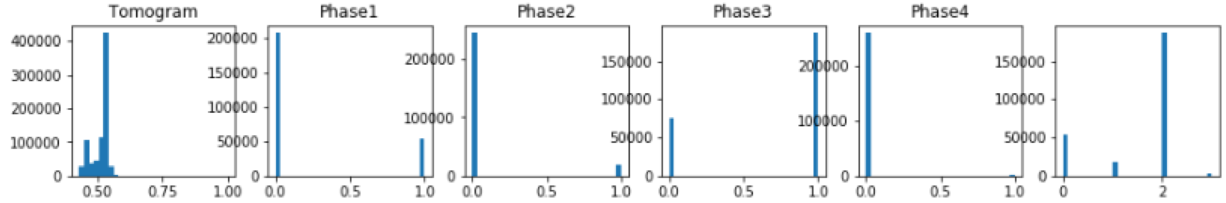
2D ResUNet

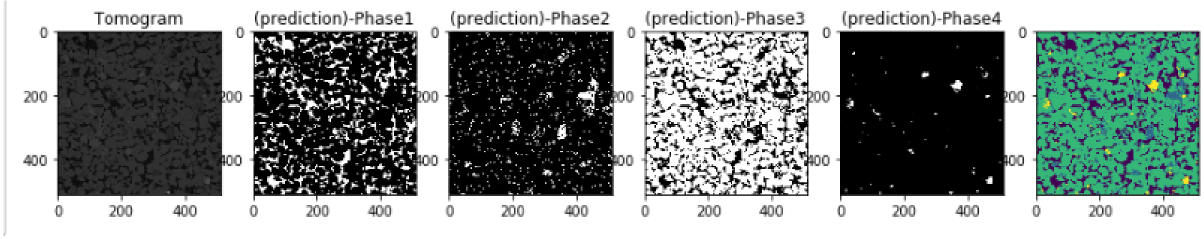
Leopard



Castlegate


Tomogram
(GT)-Phase1
(GT)-Phase2
(GT)-Phase3
(GT)-Phase4
0
200
400
0
200
400


Tomogram
Phase1
Phase2
Phase3
Phase4
400000
300000
200000
100000
0
0.50
0.75
1.00
200000
150000
100000
50000
0
0.0
0.5
1.0
0
2


Tomogram
(prediction)-Phase1
(prediction)-Phase2
(prediction)-Phase3
(prediction)-Phase4
0
200
400
0
200
400

Bentheimer

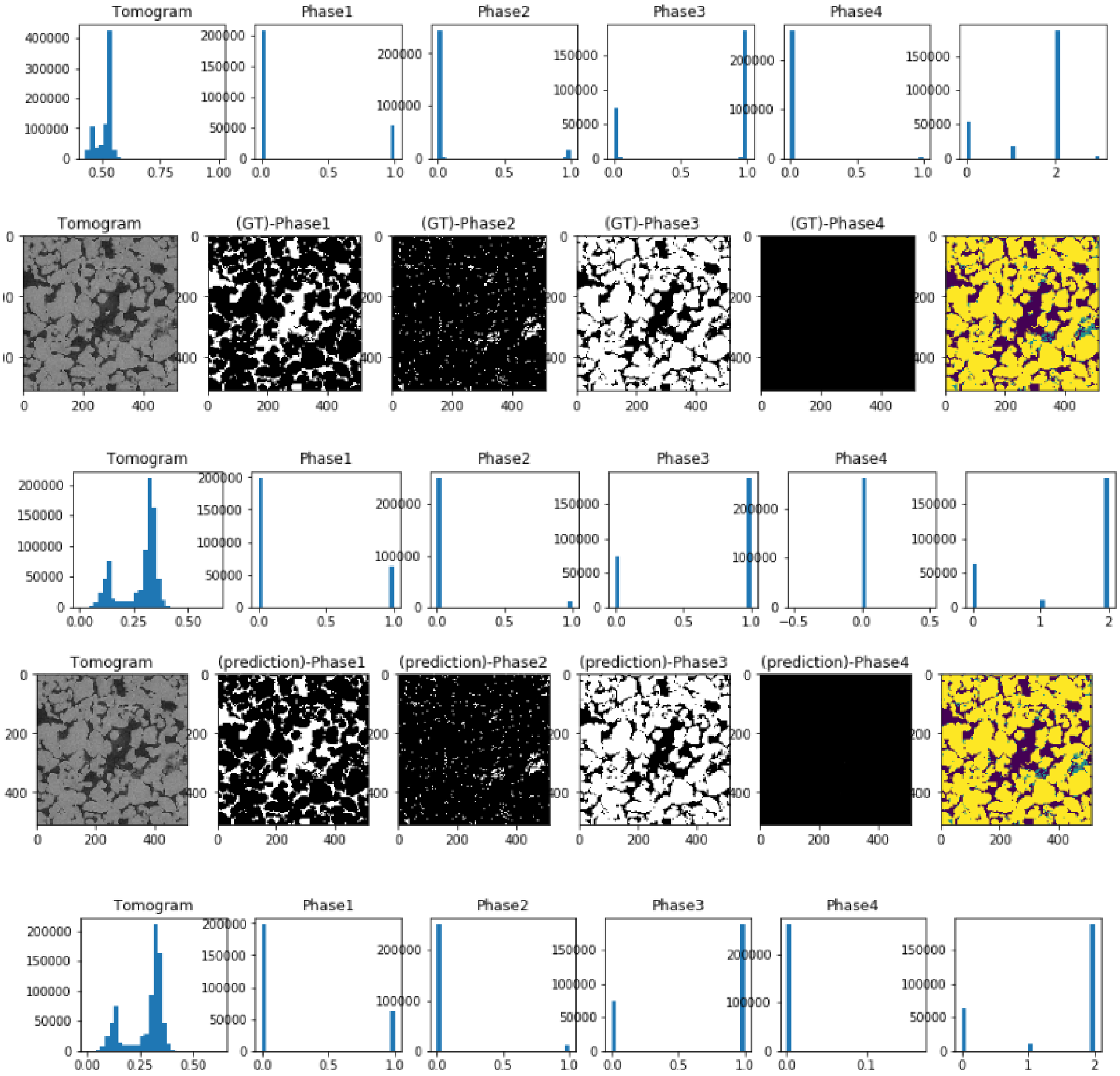
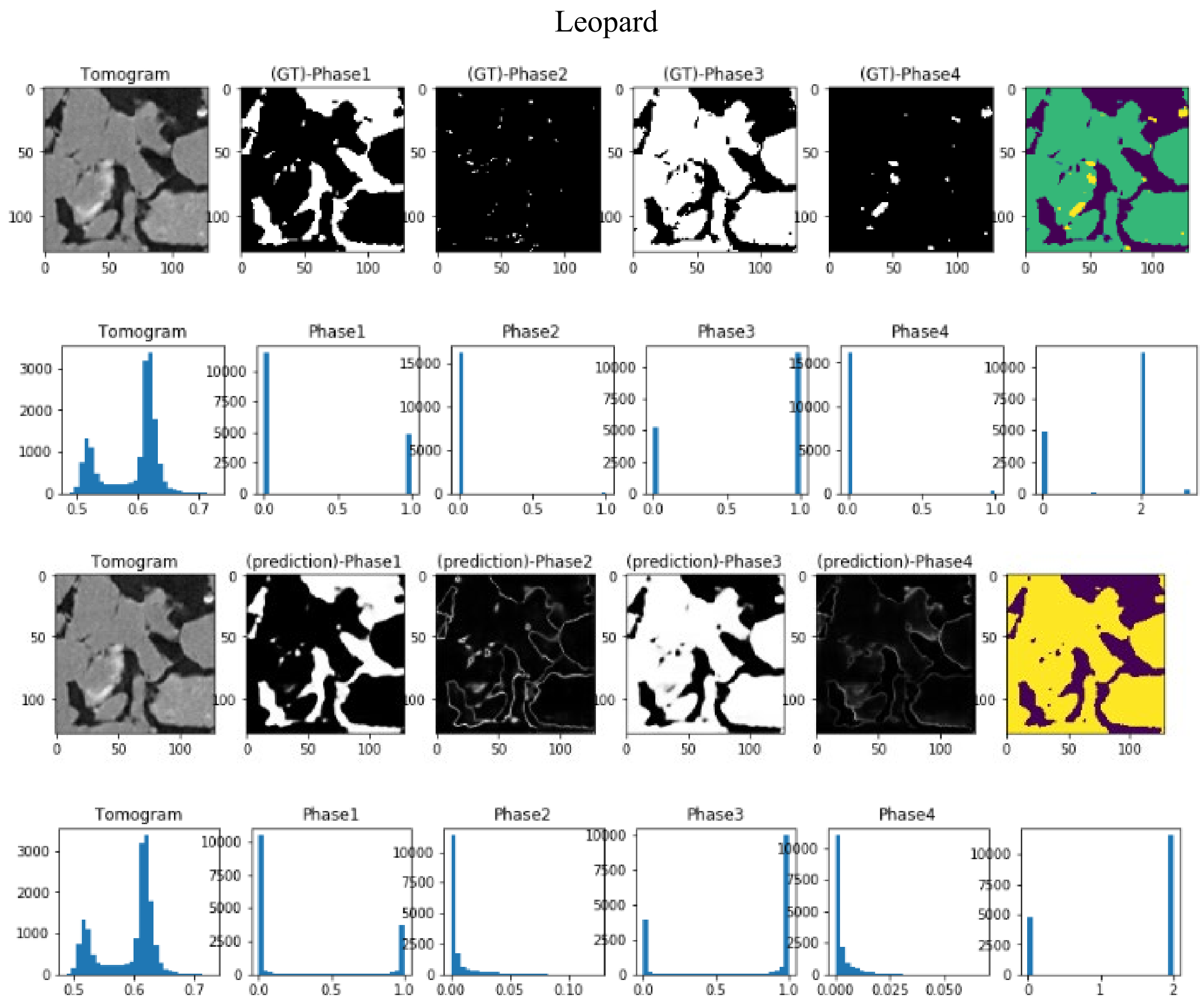


Fig. 12 – 2D ResUNet Prediction vs Ground Truth

3D U-ResNet


Leopard
Tomogram
(GT)-Phase1
(GT)-Phase2
(GT)-Phase3
(GT)-Phase4
0
50
100
Tomogram
Phase1
Phase2
Phase3
Phase4
0
1000
2000
3000
2500
5000
7500
10000
15000
0.5
0.6
0.7
0.0
0.5
1.0
2
Tomogram
(prediction)-Phase1
(prediction)-Phase2
(prediction)-Phase3
(prediction)-Phase4
Tomogram
Phase1
Phase2
Phase3
Phase4
0.00
0.05
0.10
0.000
0.025
0.050
1

Castlegate

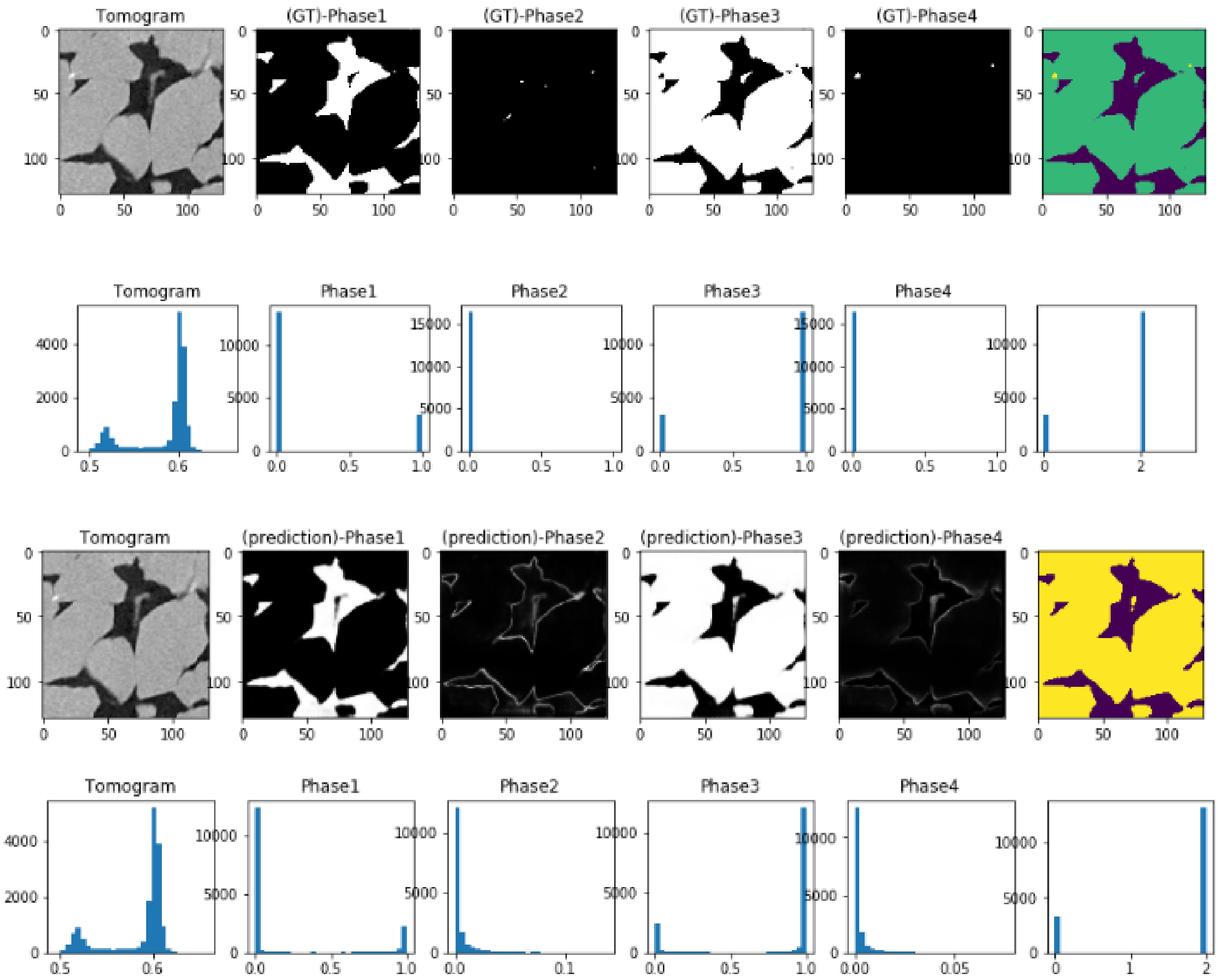
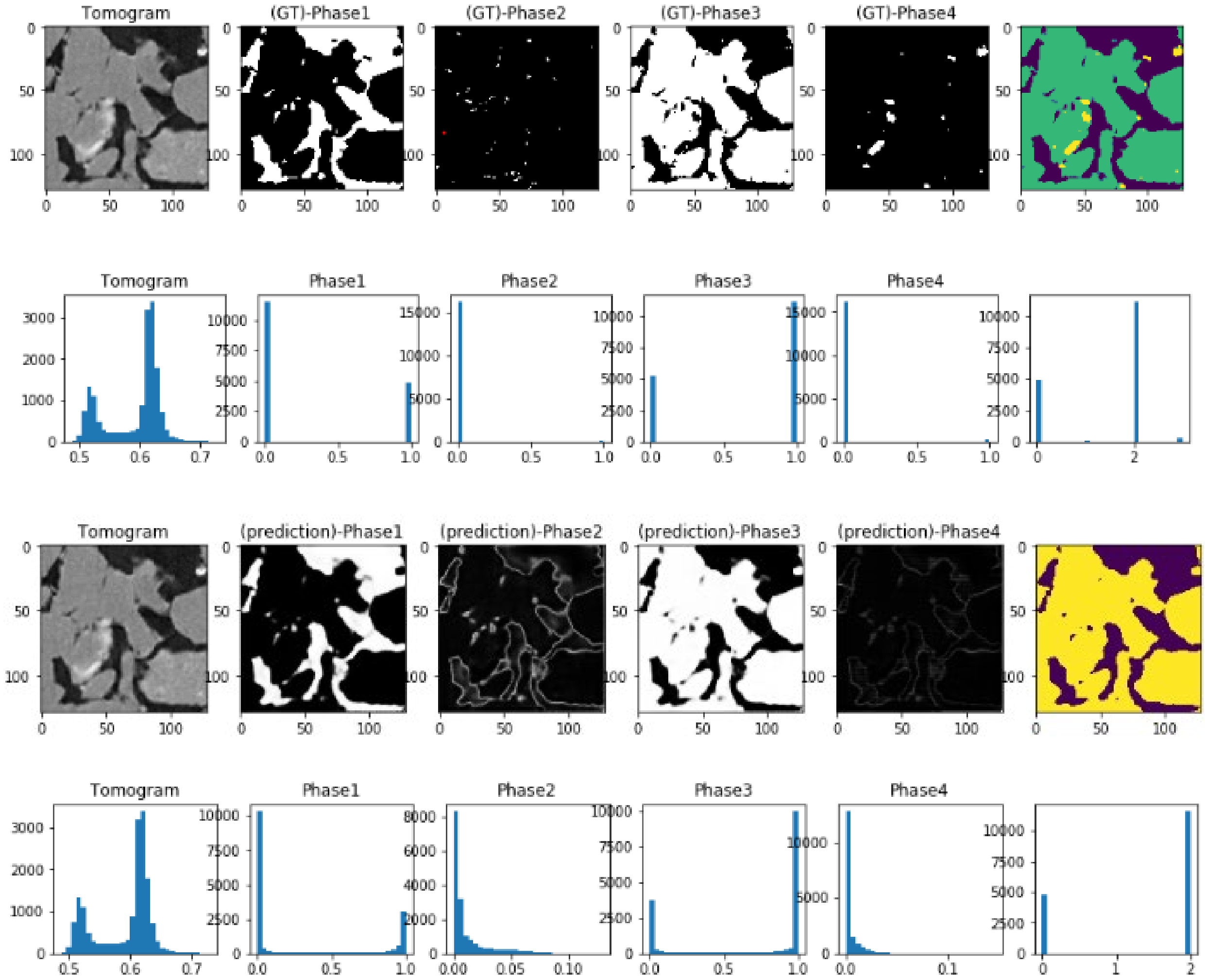


Fig. 13 – 3D U-ResNet Prediction vs Ground Truth

3D UNet

Leopard



Castlegate

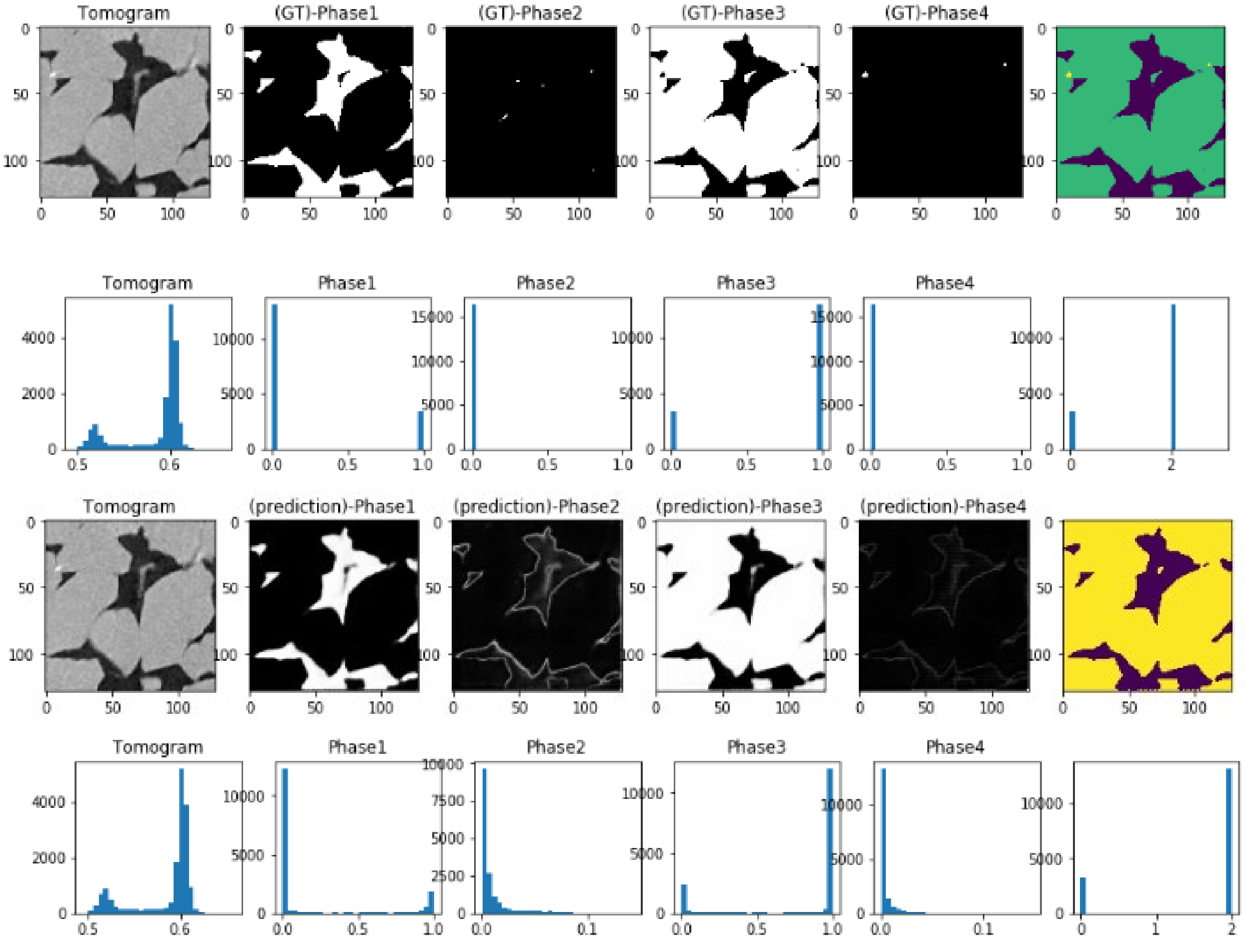
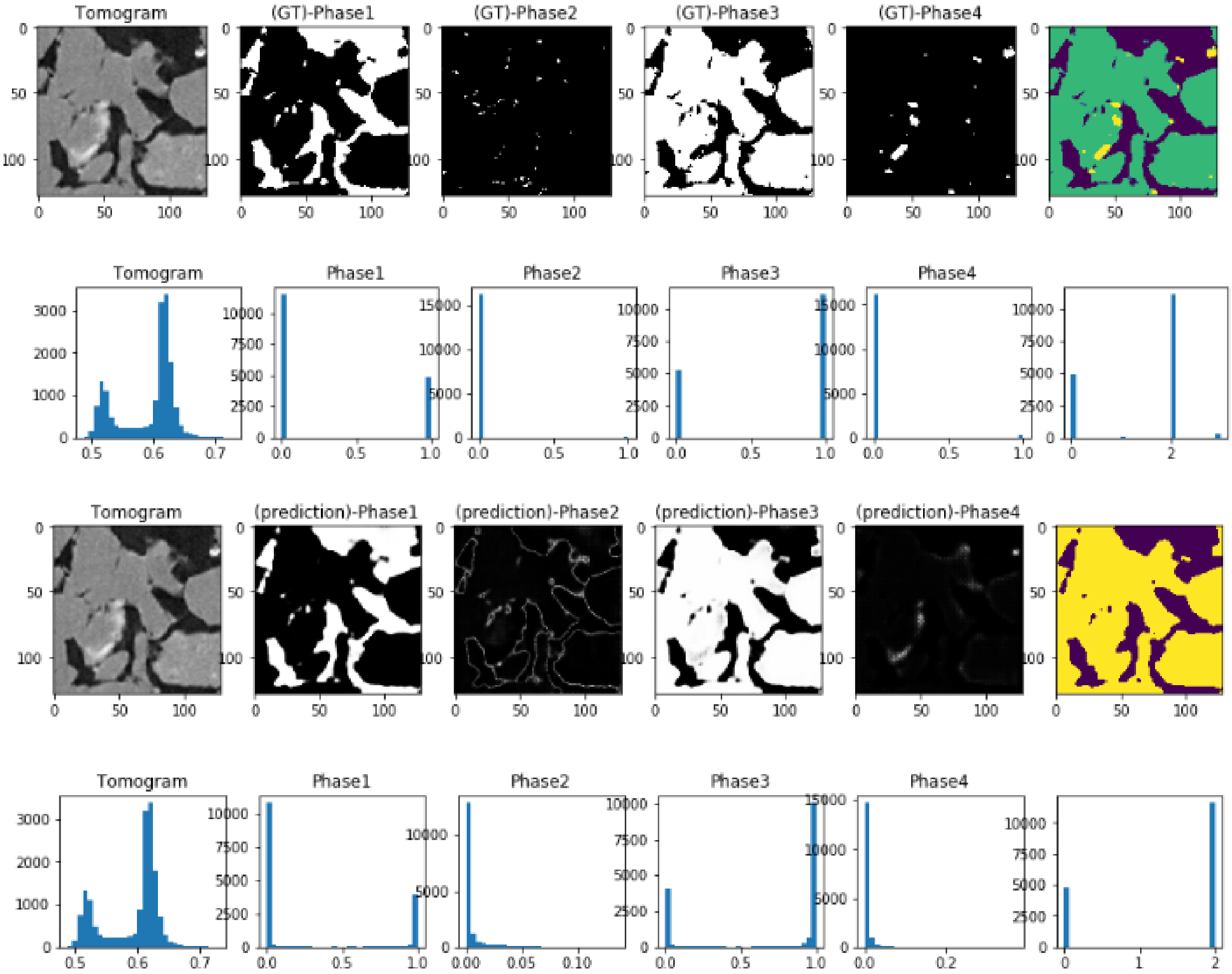


Fig. 14 – 3D UNet Prediction vs Ground Truth

3D ResUNet

Leopard



Castlegate

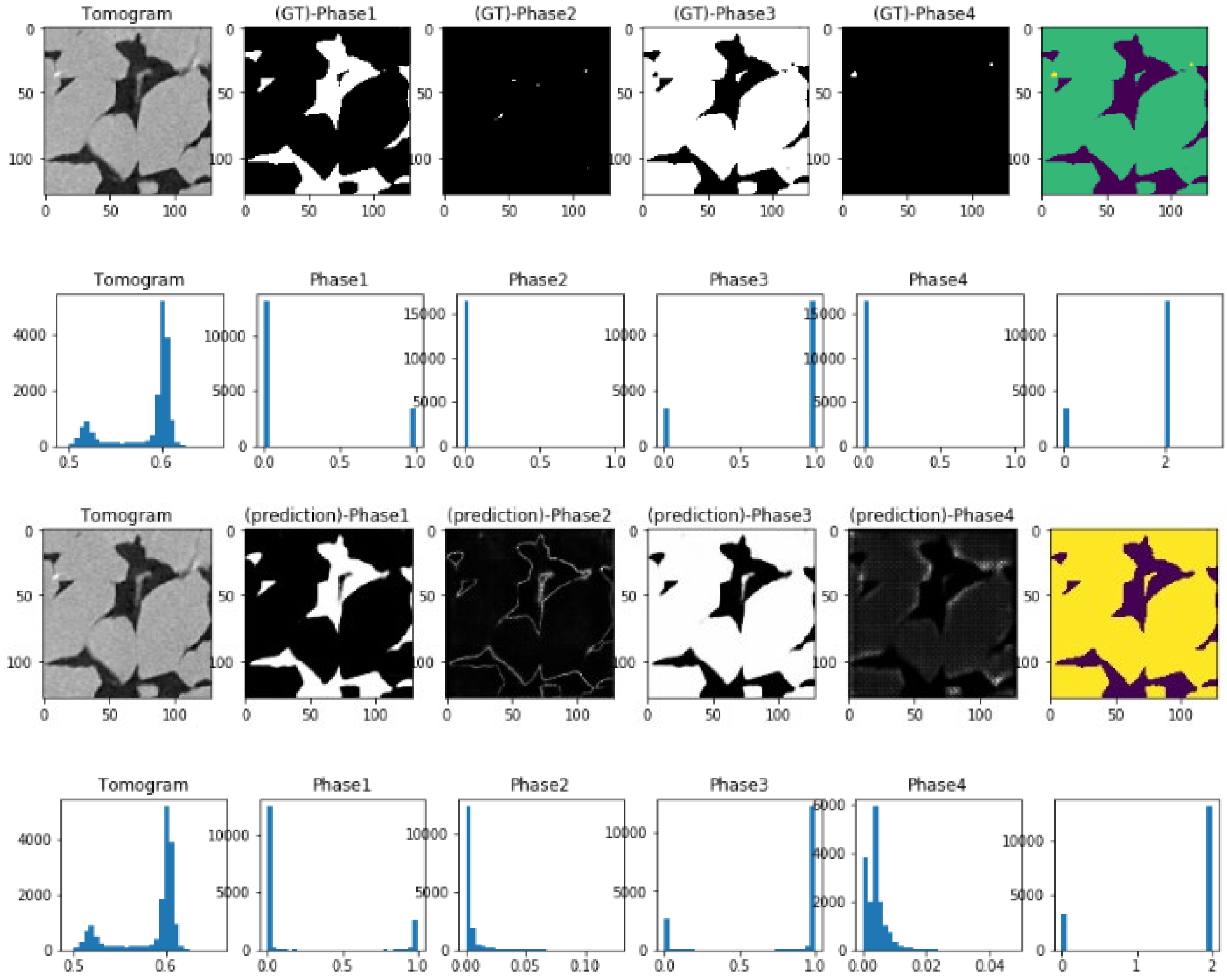
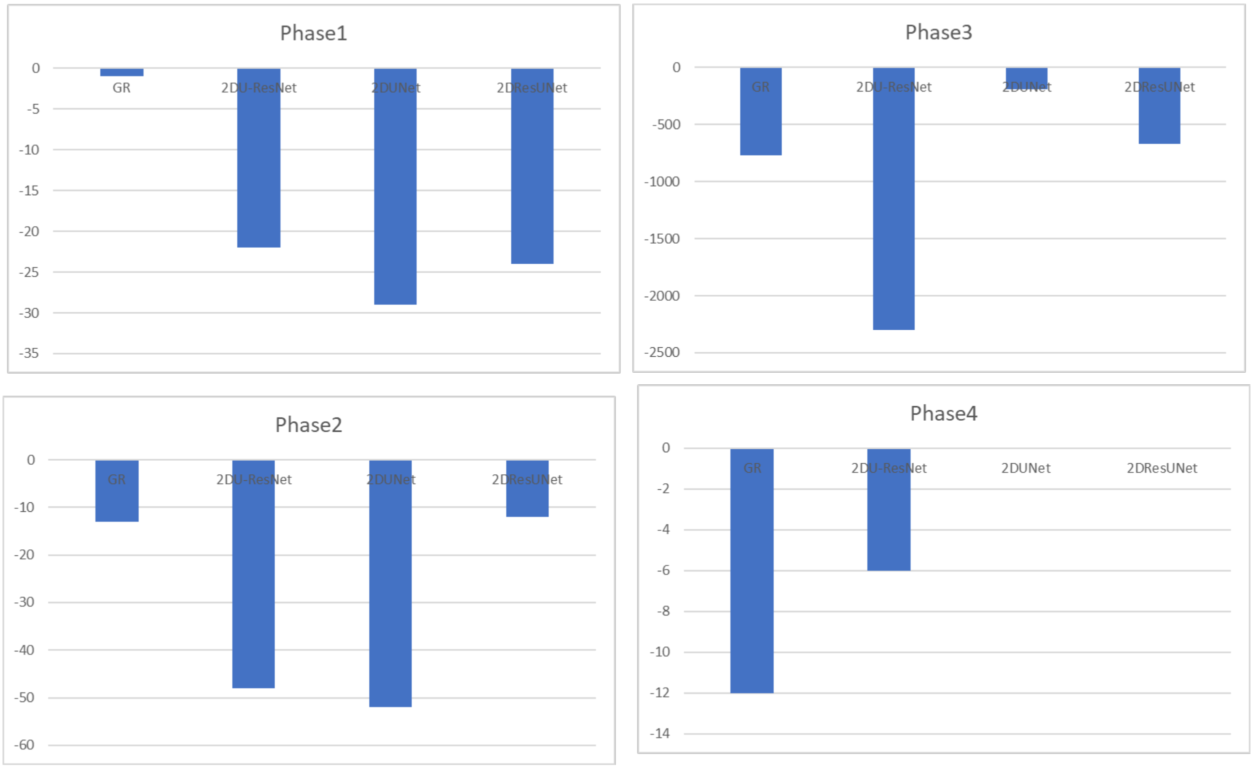


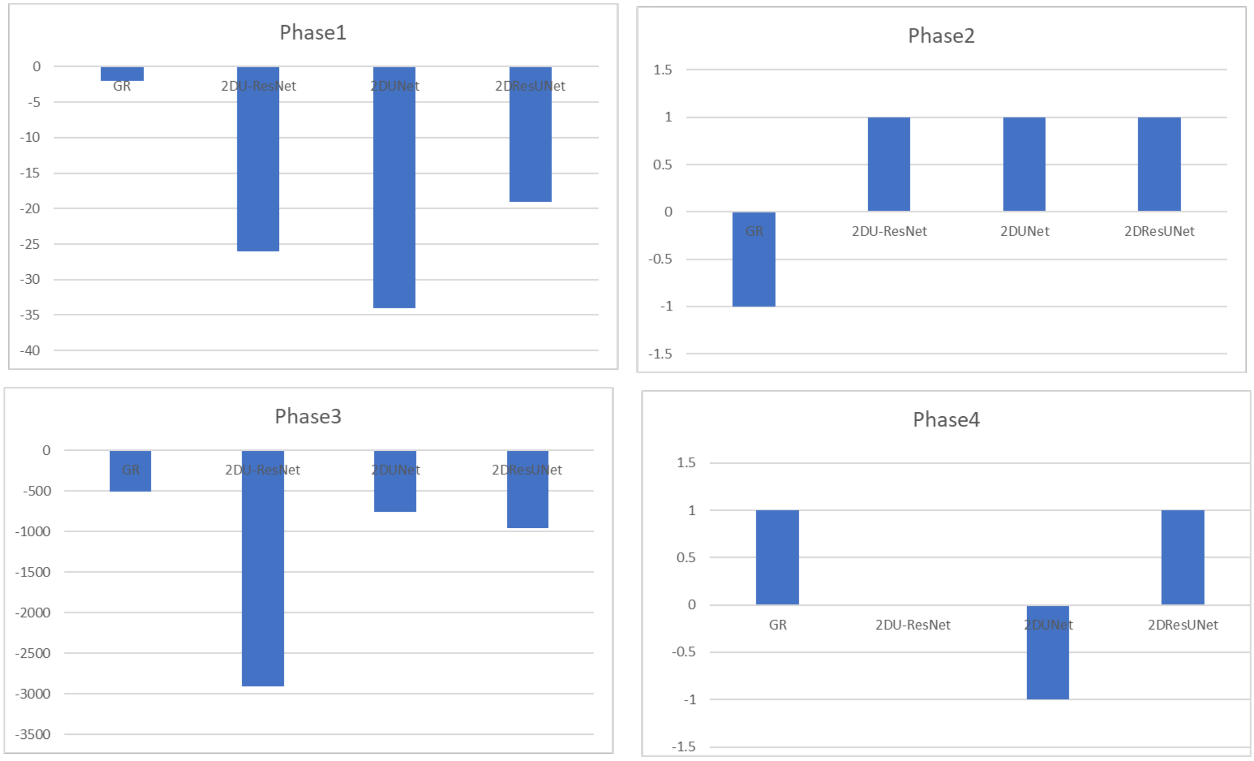
Fig. 15 – 3D ResUNet Prediction vs Ground Truth

Validation Euler Comparison

2D Castlegate



2D Leopard


Phase1
0
-5
-10
-15
-20
-25
-30
-35
-40
GR
2DU-ResNet
2DUNet
2DResUNet
Phase2
1.5
1
0.5
0
-0.5
-1
-1.5
GR
2DU-ResNet
2DUNet
2DResUNet
Phase3
0
-500
-1000
-1500
-2000
-2500
-3000
-3500
GR
2DU-ResNet
2DUNet
2DResUNet
Phase4
1.5
1
0.5
0
-0.5
-1
-1.5
GR
2DU-ResNet
2DUNet
2DResUNet

2D Bentheimer

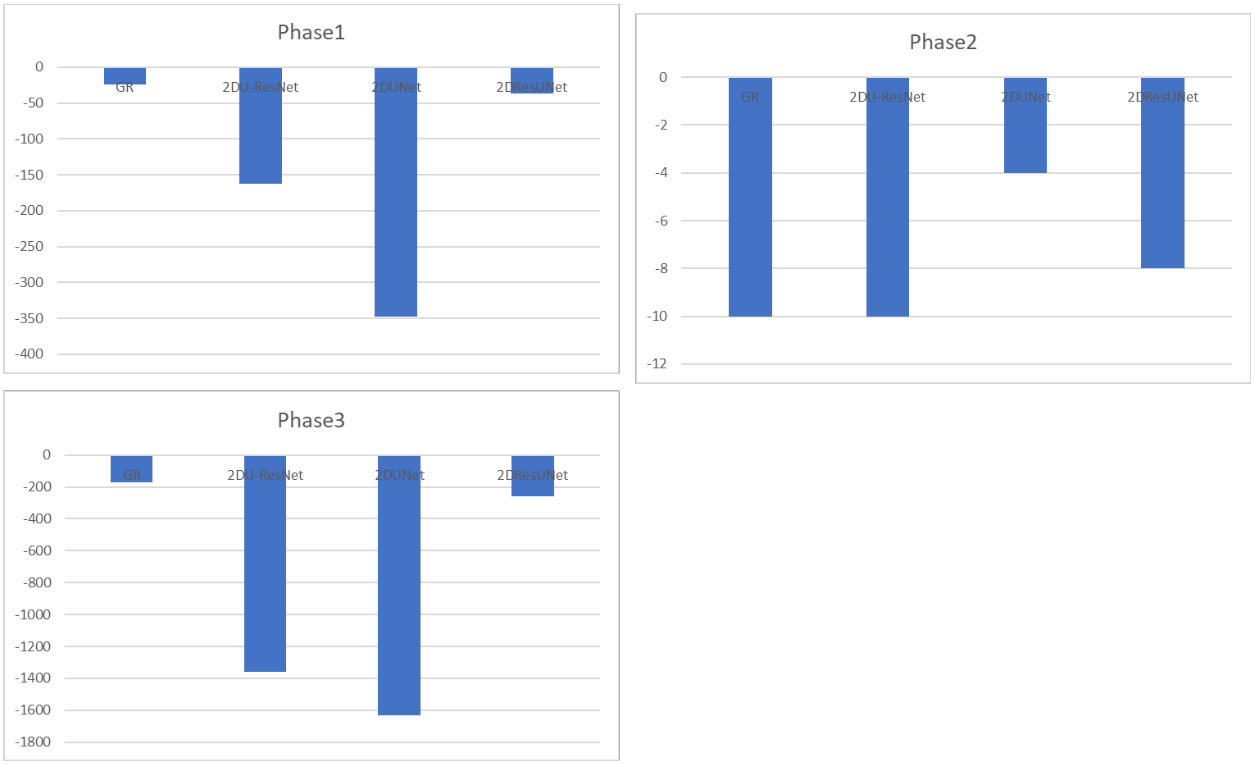
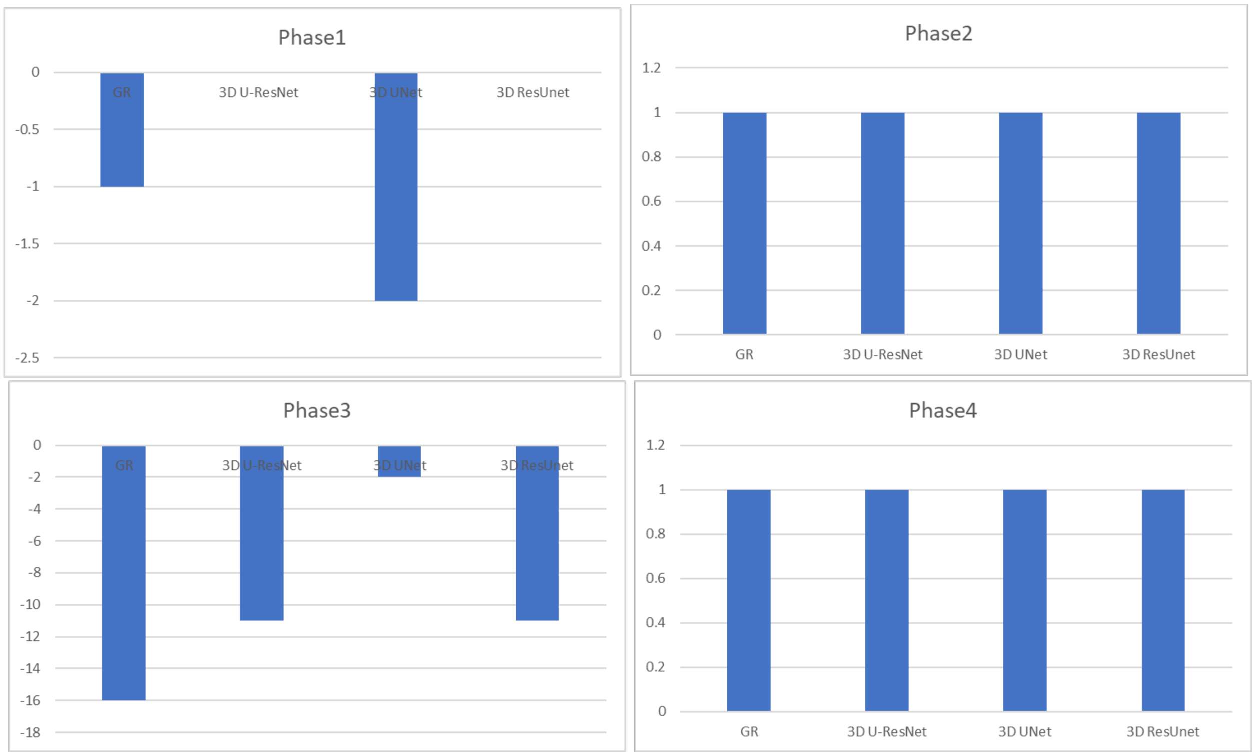


Fig. 16 – 2D Validation Euler Comparisons

3D Leopard


Phase1
0
-0.5
-1
-1.5
-2
-2.5
GR
3D U-ResNet
3D UNet
3D ResUnet
Phase2
1.2
1
0.8
0.6
0.4
0.2
0
GR
3D U-ResNet
3D UNet
3D ResUnet
Phase3
0
-2
-4
-6
-8
-10
-12
-14
-16
-18
GR
3D U-ResNet
3D UNet
3D ResUnet
Phase4
1.2
1
0.8
0.6
0.4
0.2
0
GR
3D U-ResNet
3D UNet
3D ResUnet

3D Castlegate

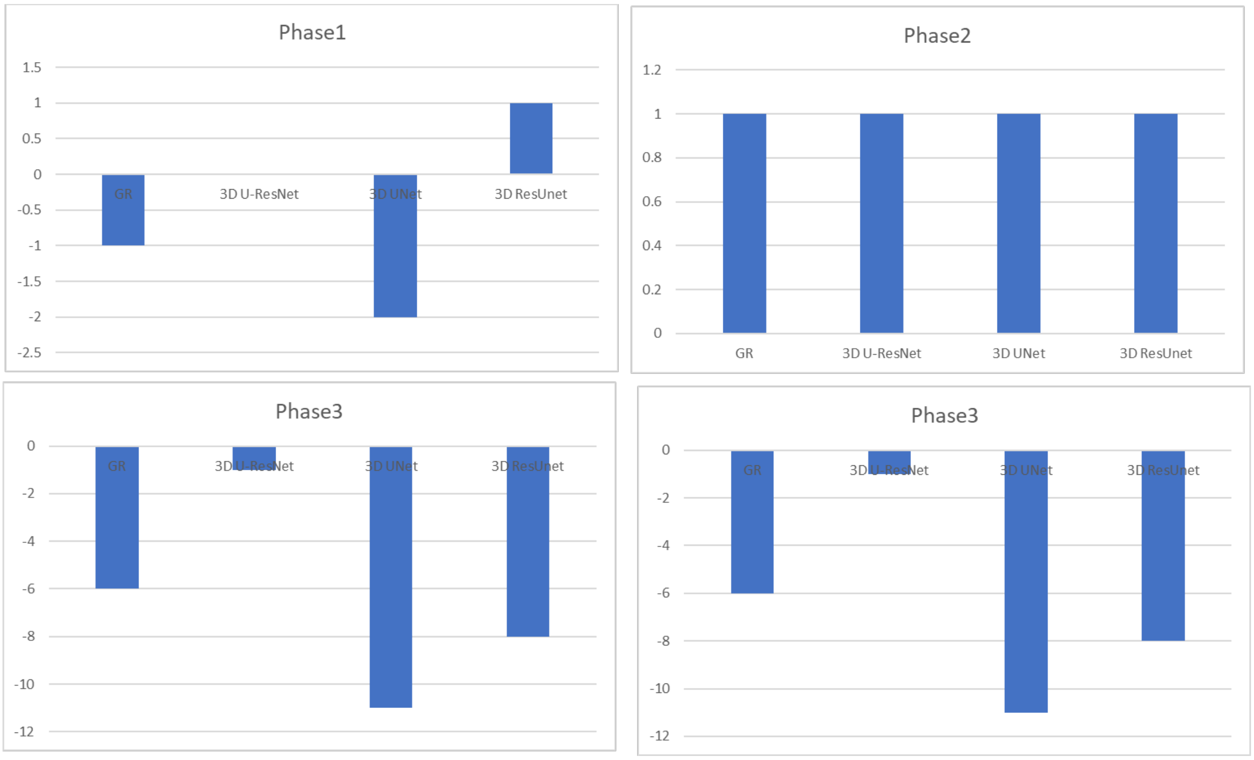


Fig. 17 – 3D Validation Euler Comparisons

	Mean IOU		Threshold Adjusted (Mean IOU)
	Training	Validation	Validation
2D *U-ResNet*	0.9996	0.9994	
2D *UNet*	0.9653	0.9702	
2D *ResUNet*	0.9842	0.9853	
3D *U-ResNet*	0.6166	0.4761	0.931
3D *UNet*	0.5548	0.4586	0.922
3D *ResUNet*	0.5527	0.4997	0.9377

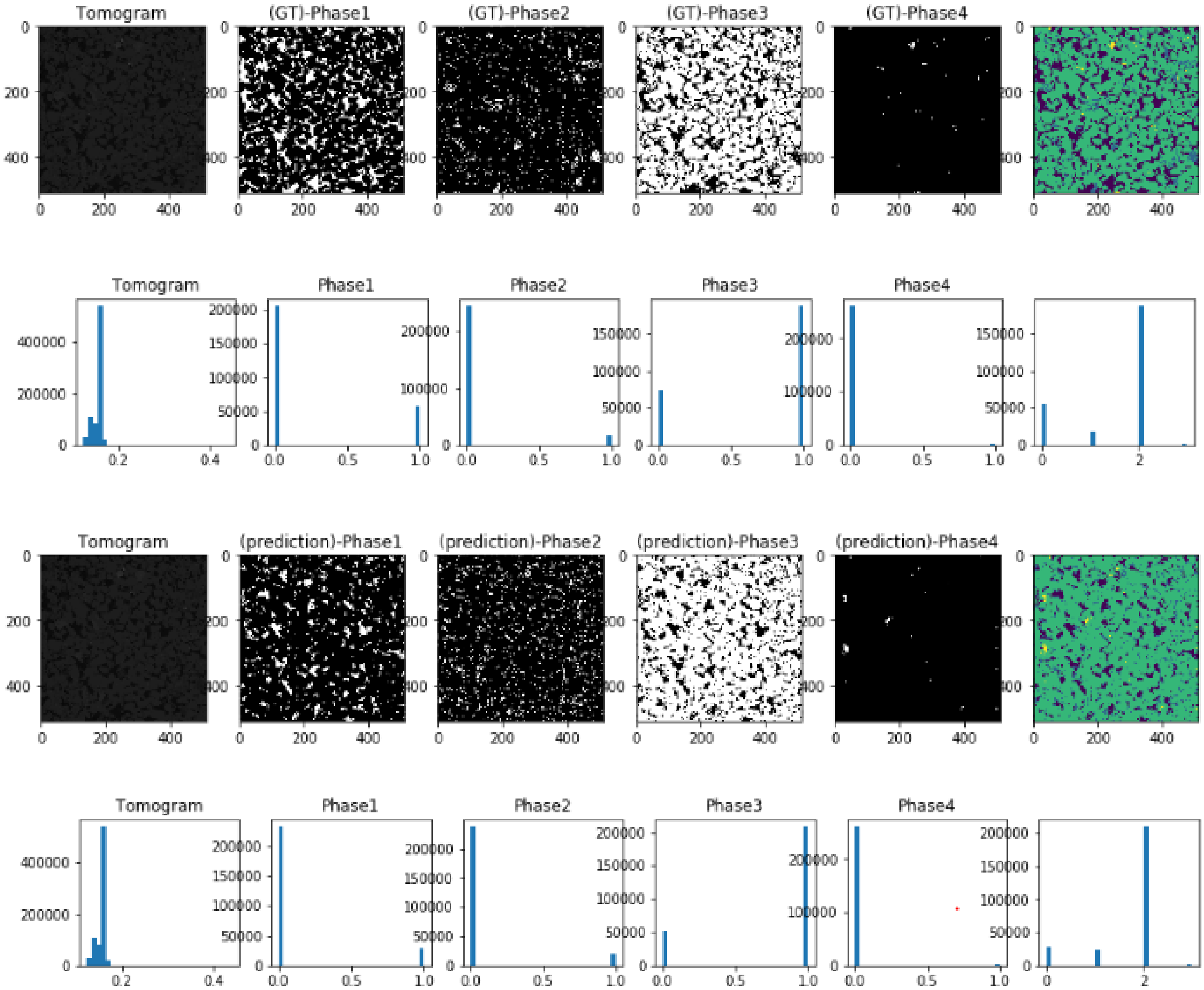
Table 2 – Validation Mean IOU

By examining the prediction results from validation datasets, the 2D models showed about 0.96 to 0.999 IOU on validation datasets. Note: for 3D models, the IOU for validation is low due to significant problems of prediction with phase 2 and phase 4 which skewed the mean IOU, but with adjusted threshold its around 0.92 to 0.93 IOU. The results were very interesting for Euler number for 2D models. All 2D models for Bentheimer, Leopard and Castlegate sandstones fails to predict accurate Euler number for phase 1 (porosity) when compared to the ground truth (GT), while 2D *ResUNe*t outperforms all other models in other phases and is consistent with the ground truth. The Euler number for 3D models comes very consistent with ground truth as 3D convolutions proves to preserve the connectivity. However, since we are using 128^2 image size and using thresholds to adjust phase 3 and phase 4, it is not a very fair comparison. However, 3D *UNet* and 3D *ResUNet* performs better then 3D *UNet*. With more training datasets, the results would be much better.

Testing Dataset

2D U-ResNet

Castlegate



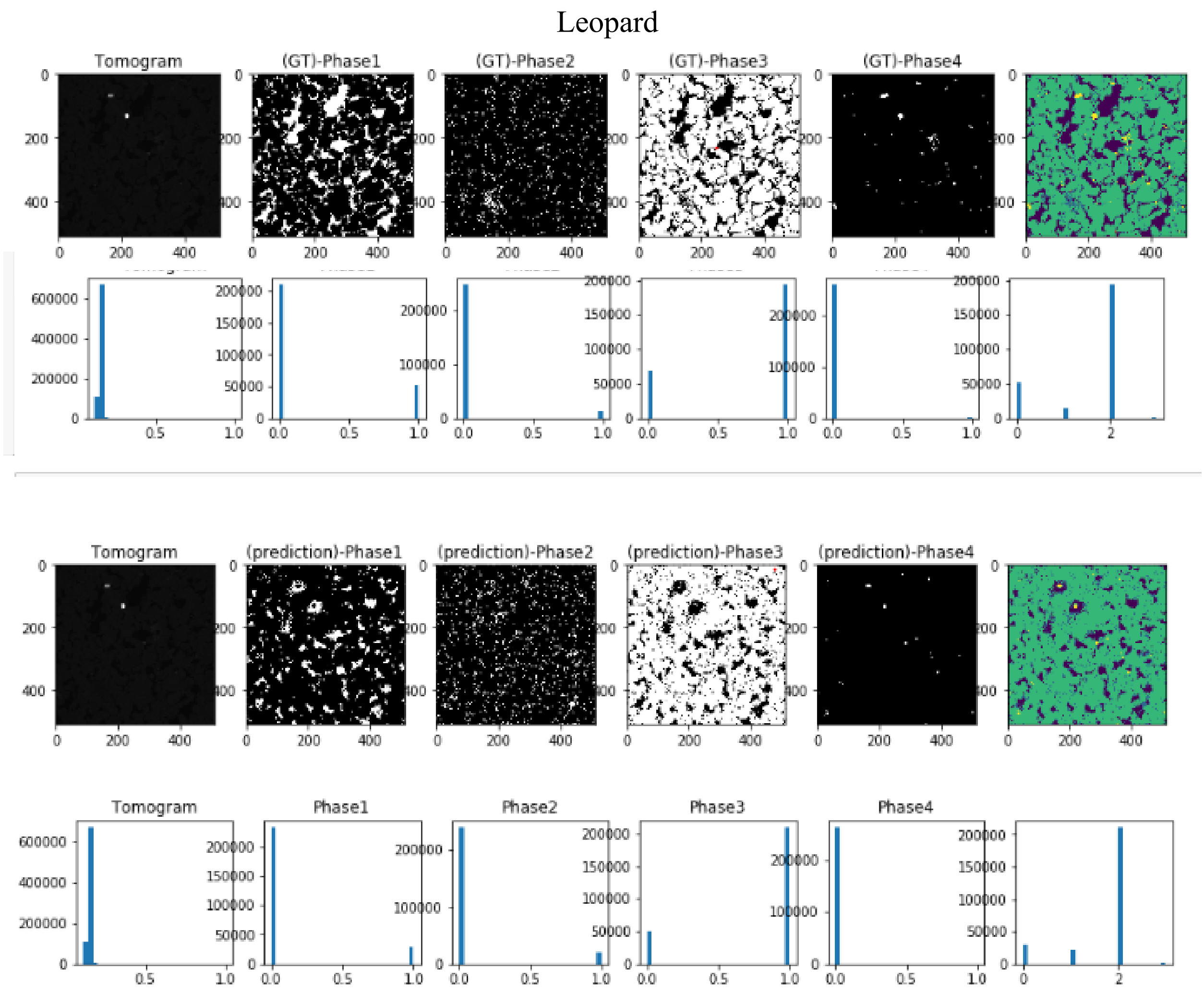
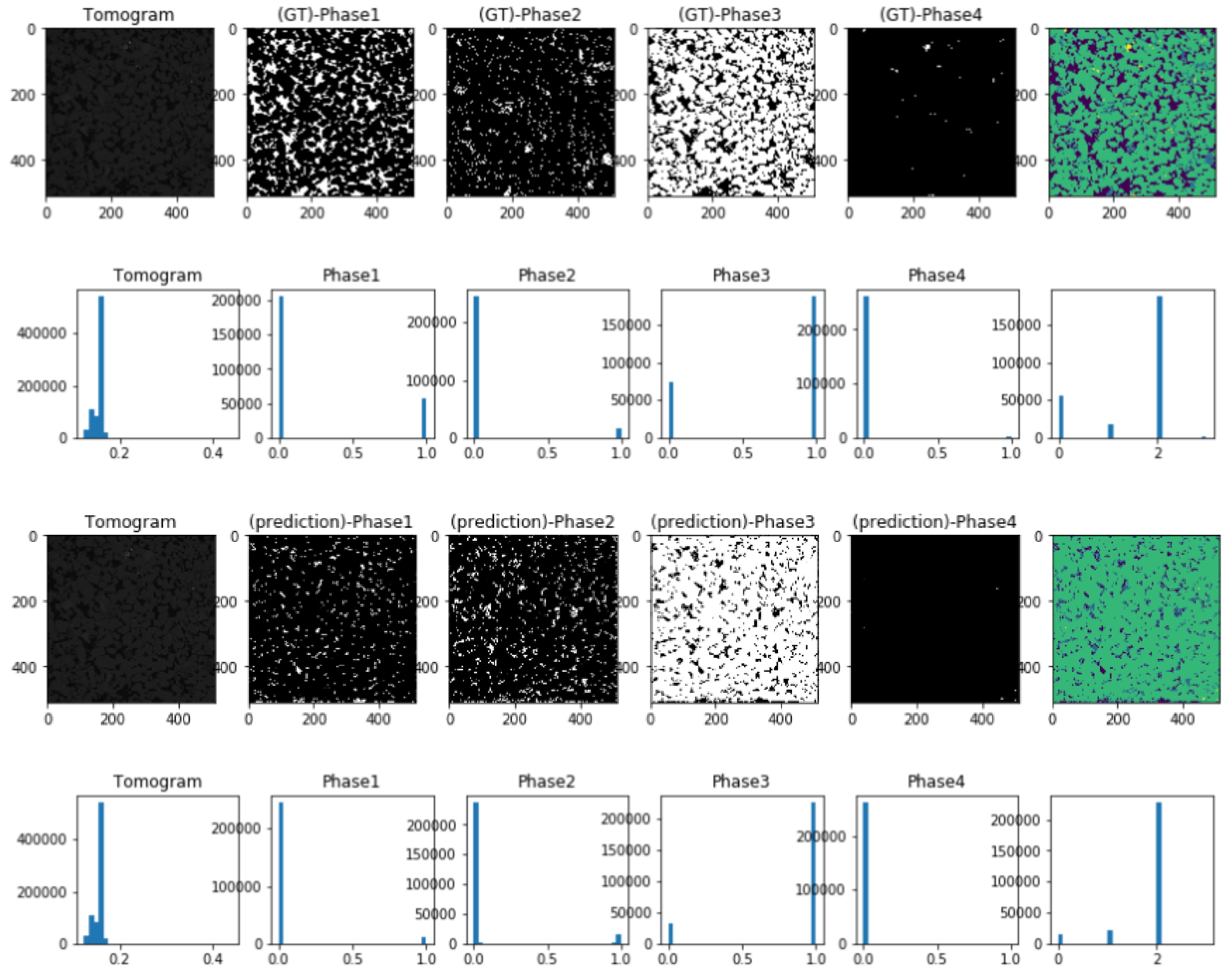


Fig. 18 – 2D U-ResNet Prediction vs Ground Truth

2D UNet

Castlegate



Leopard

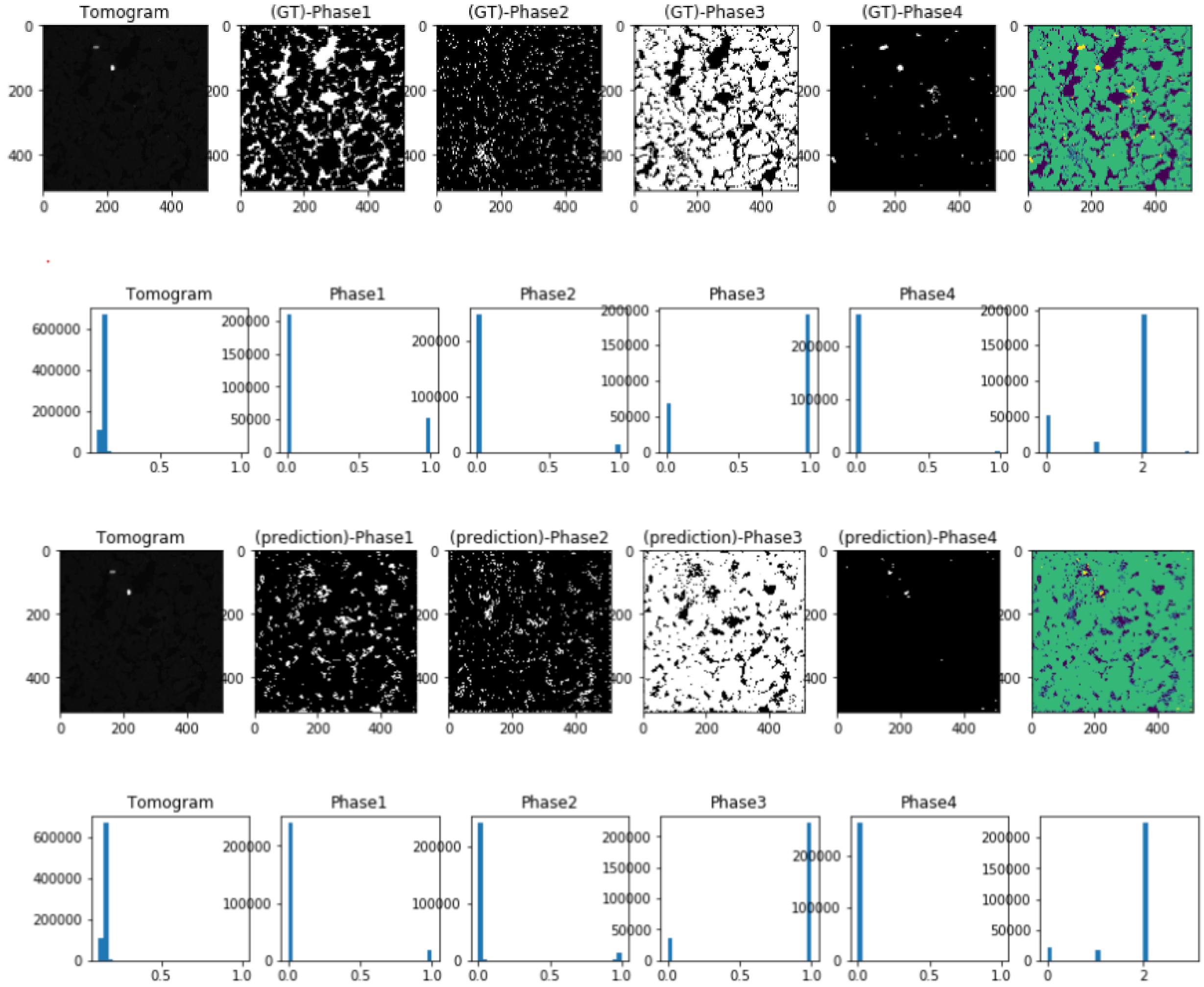
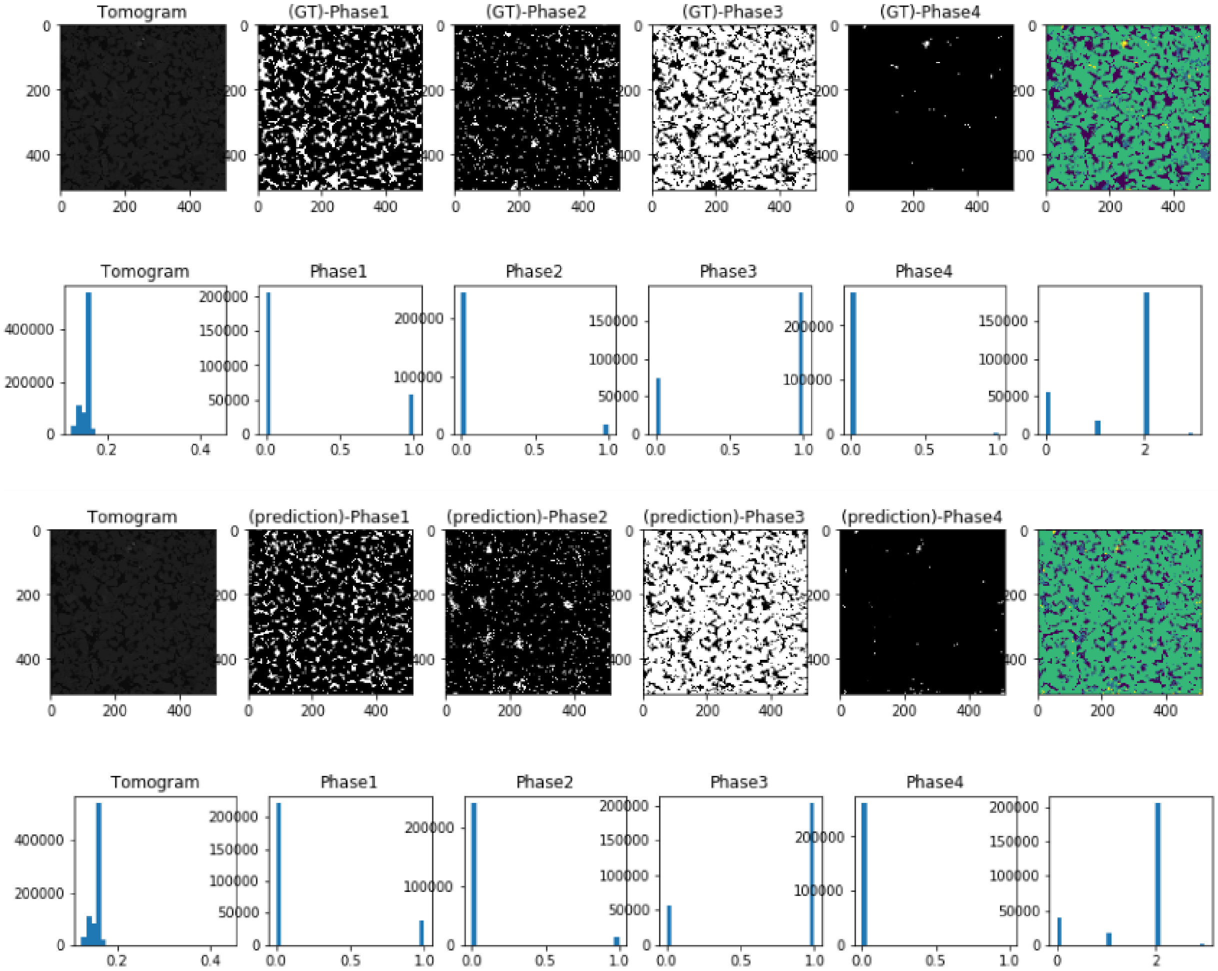


Fig. 19 – 2D UNet Prediction vs Ground Truth

2D ResUNet

Castlegate



Leopard

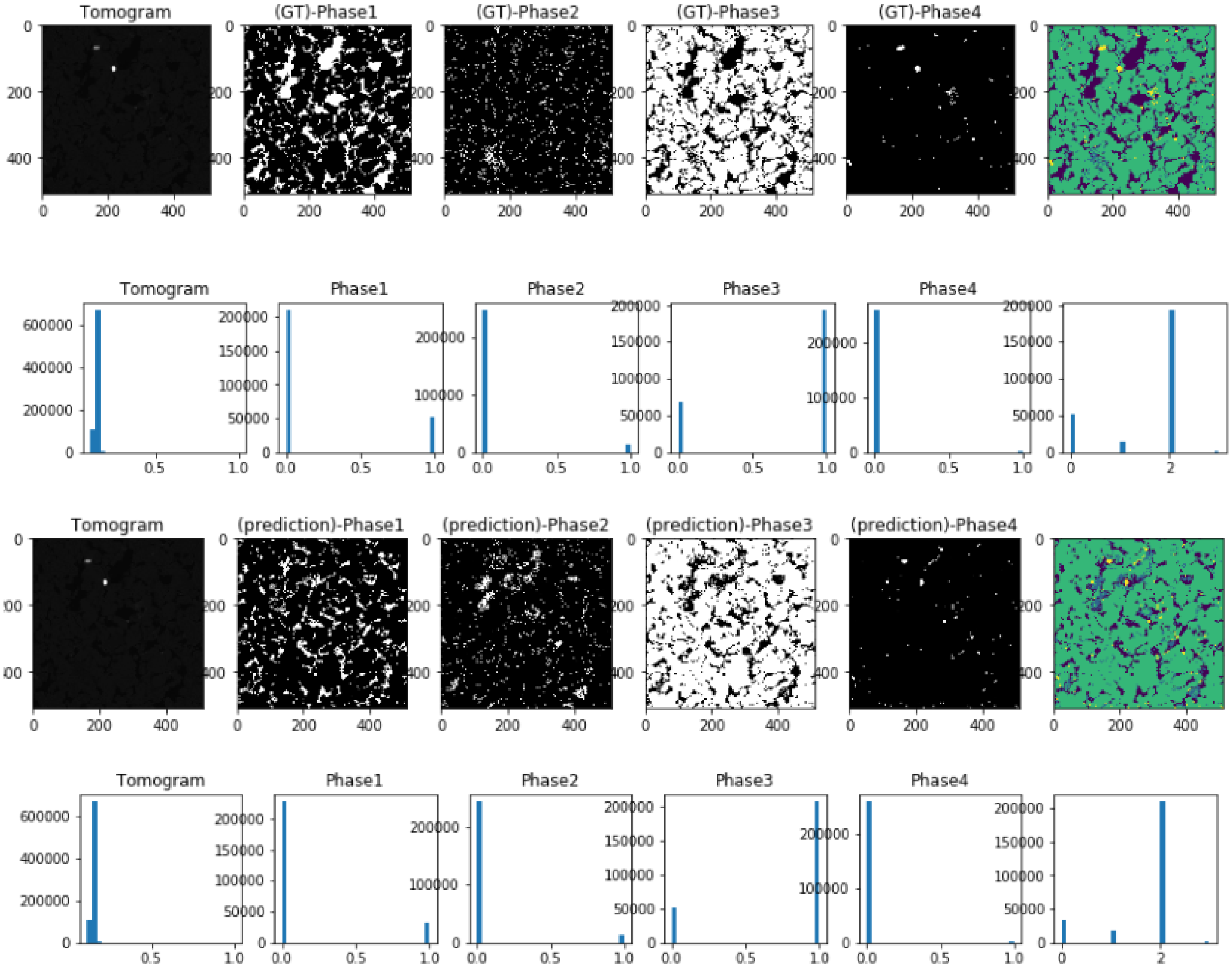
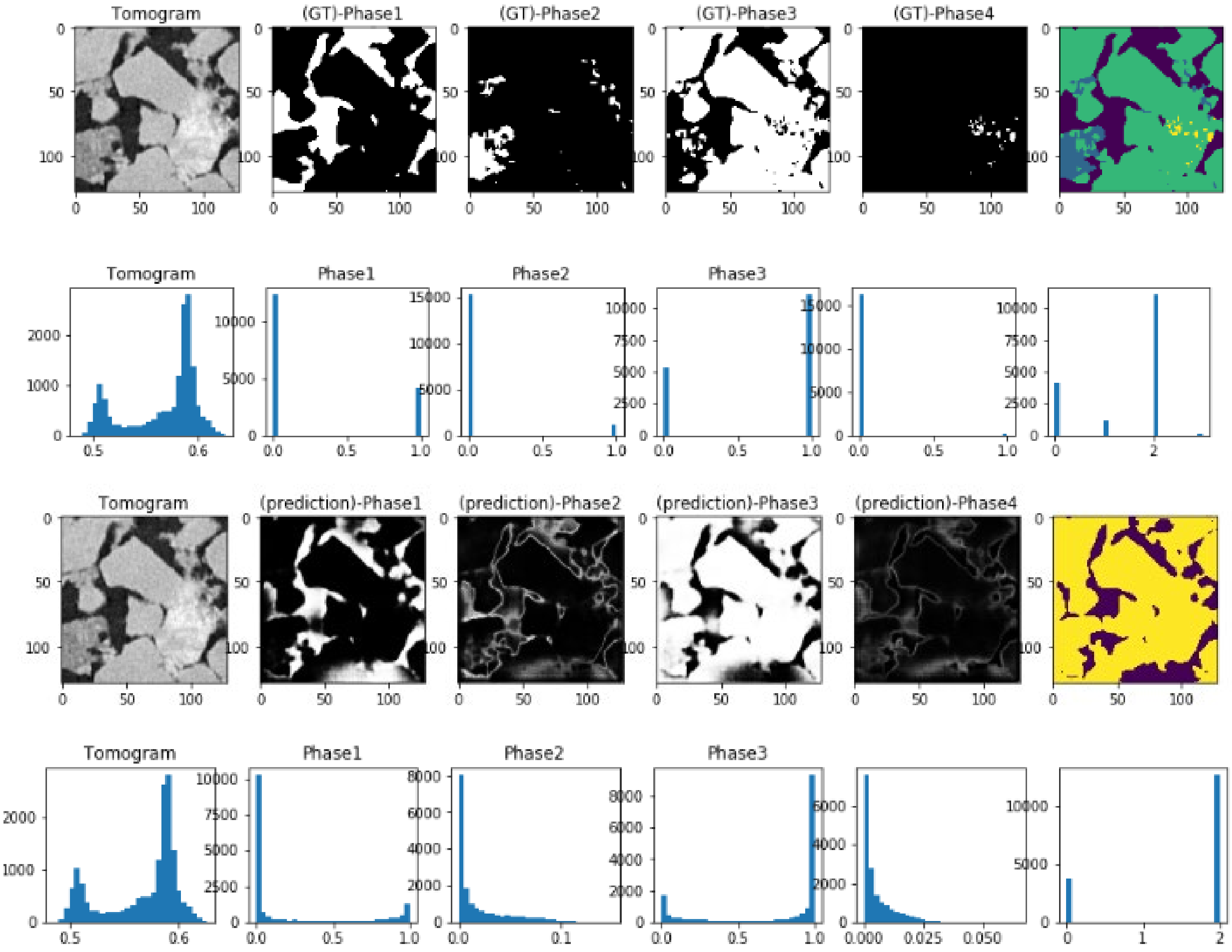


Fig. 20 – 2D ResUNet Prediction vs Ground Truth

3D U-ResNet

Bentheimer



Leopard

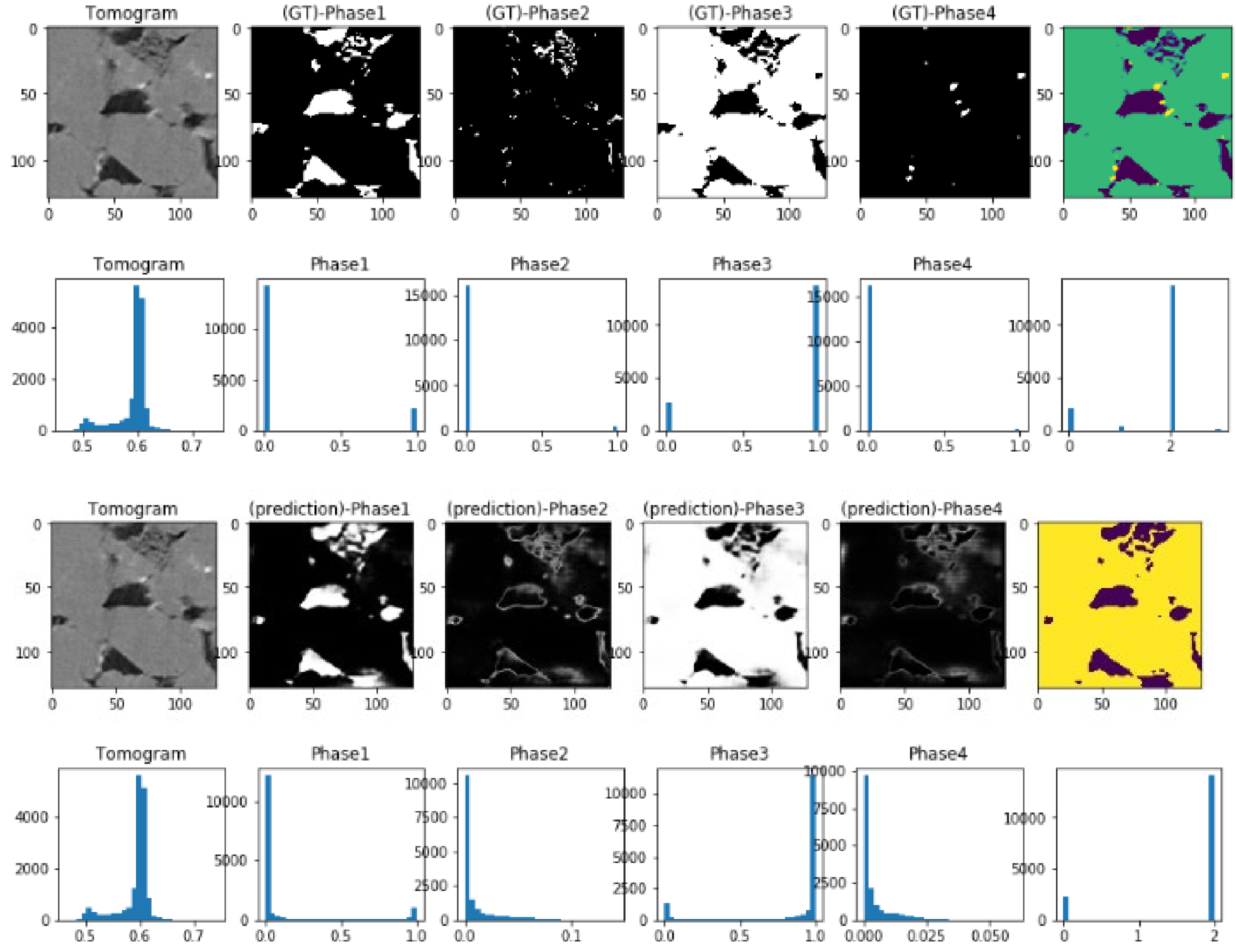
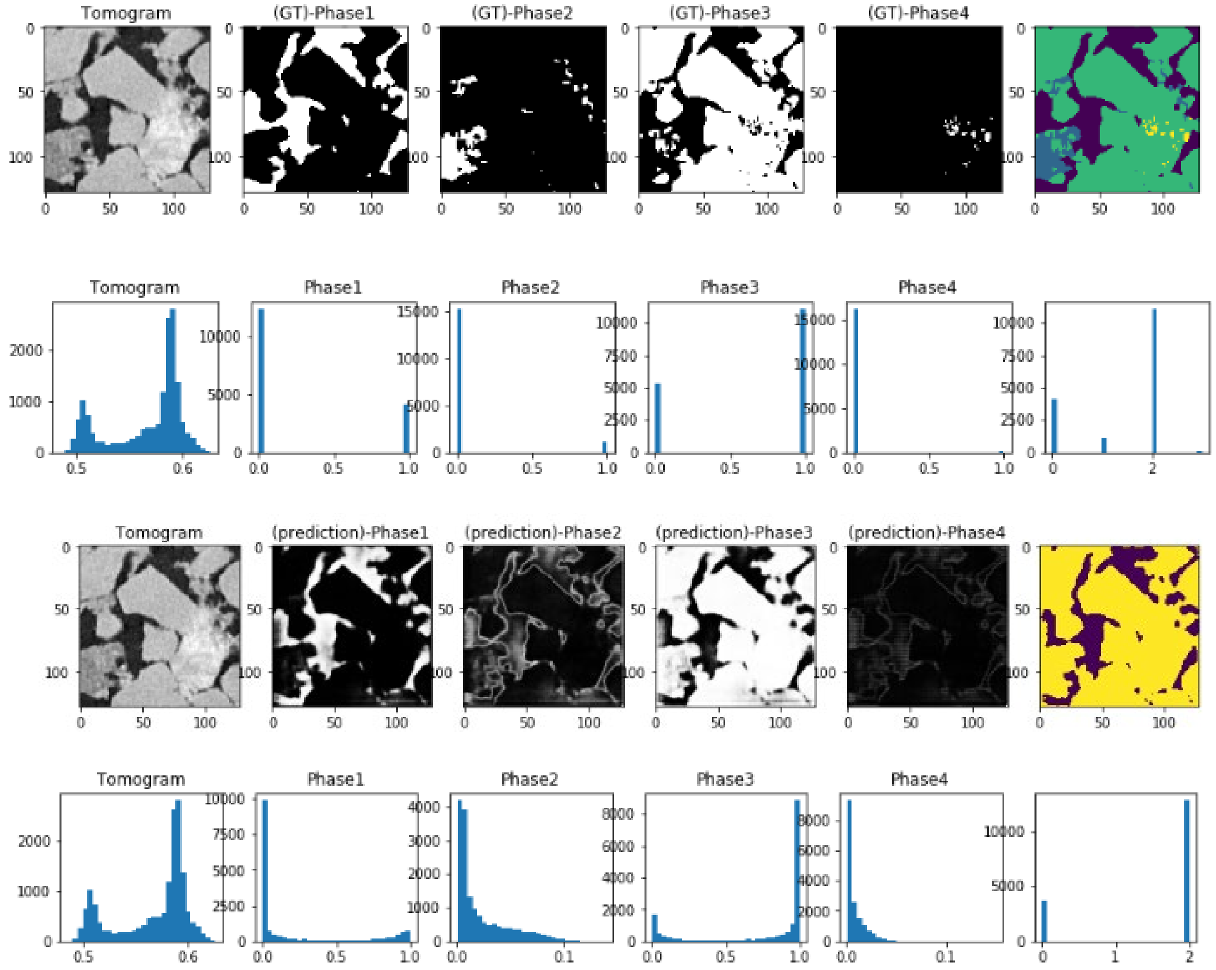


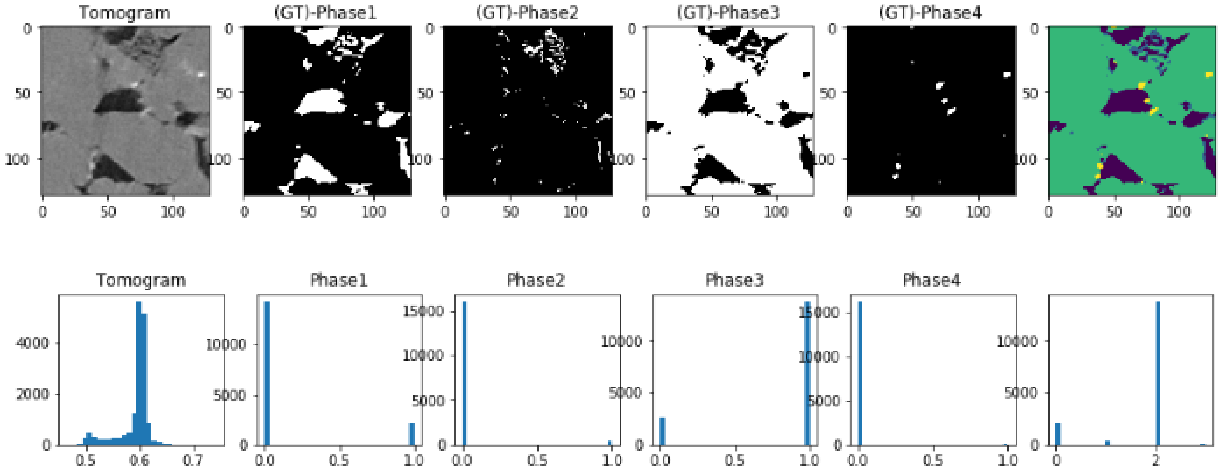
Fig. 21 – 2D U-ResNet Prediction vs Ground Truth

3D UNet

Bentheimer



Leopard



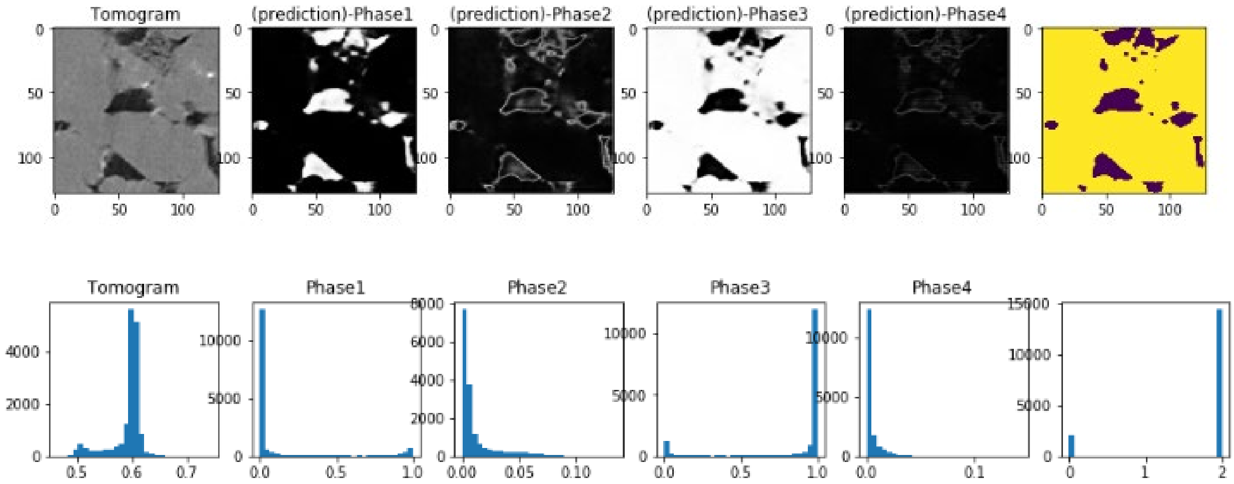
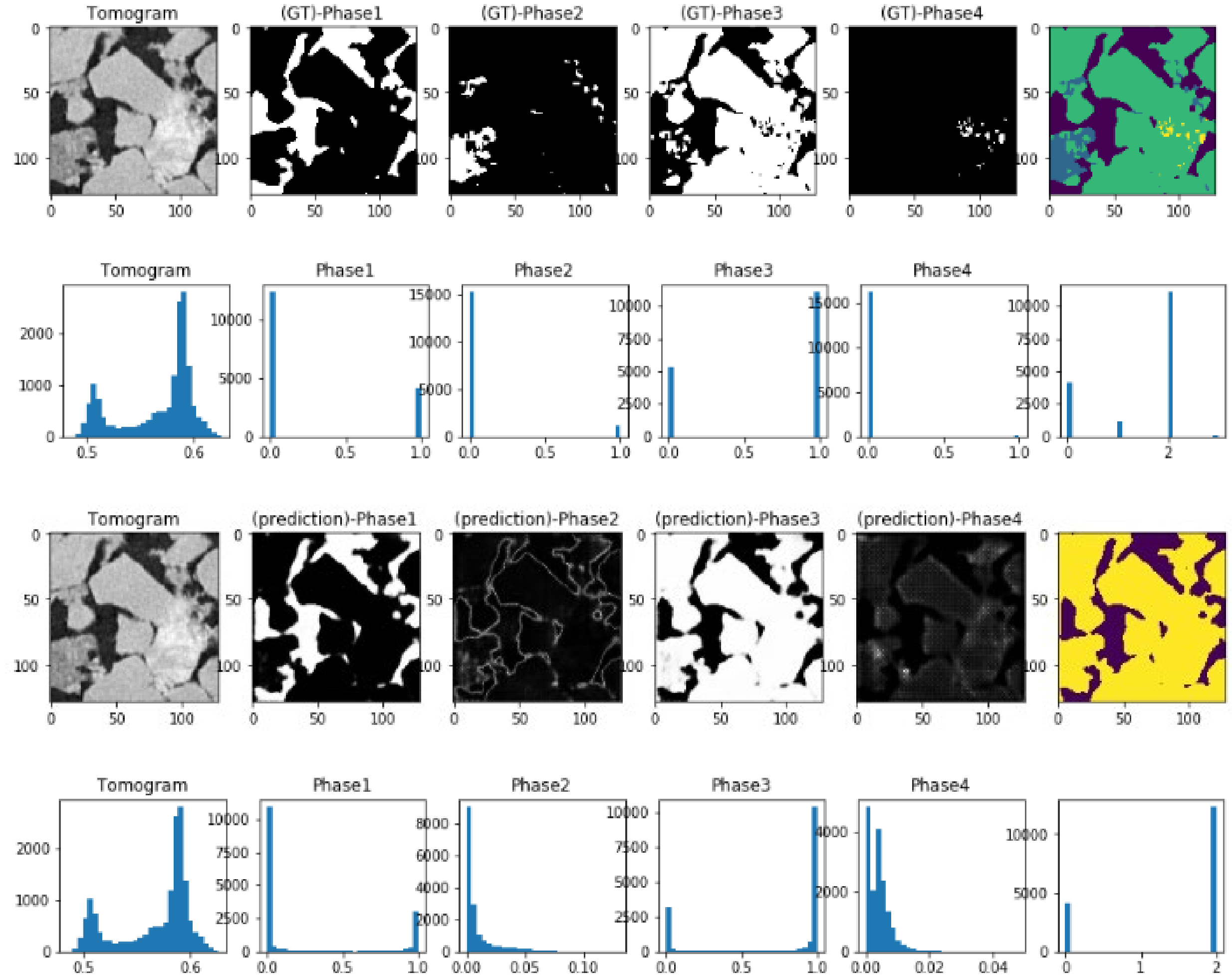


Fig. 22 – 3D UNet Prediction vs Ground Truth

3D ResUNet

Bentheimer



Leopard

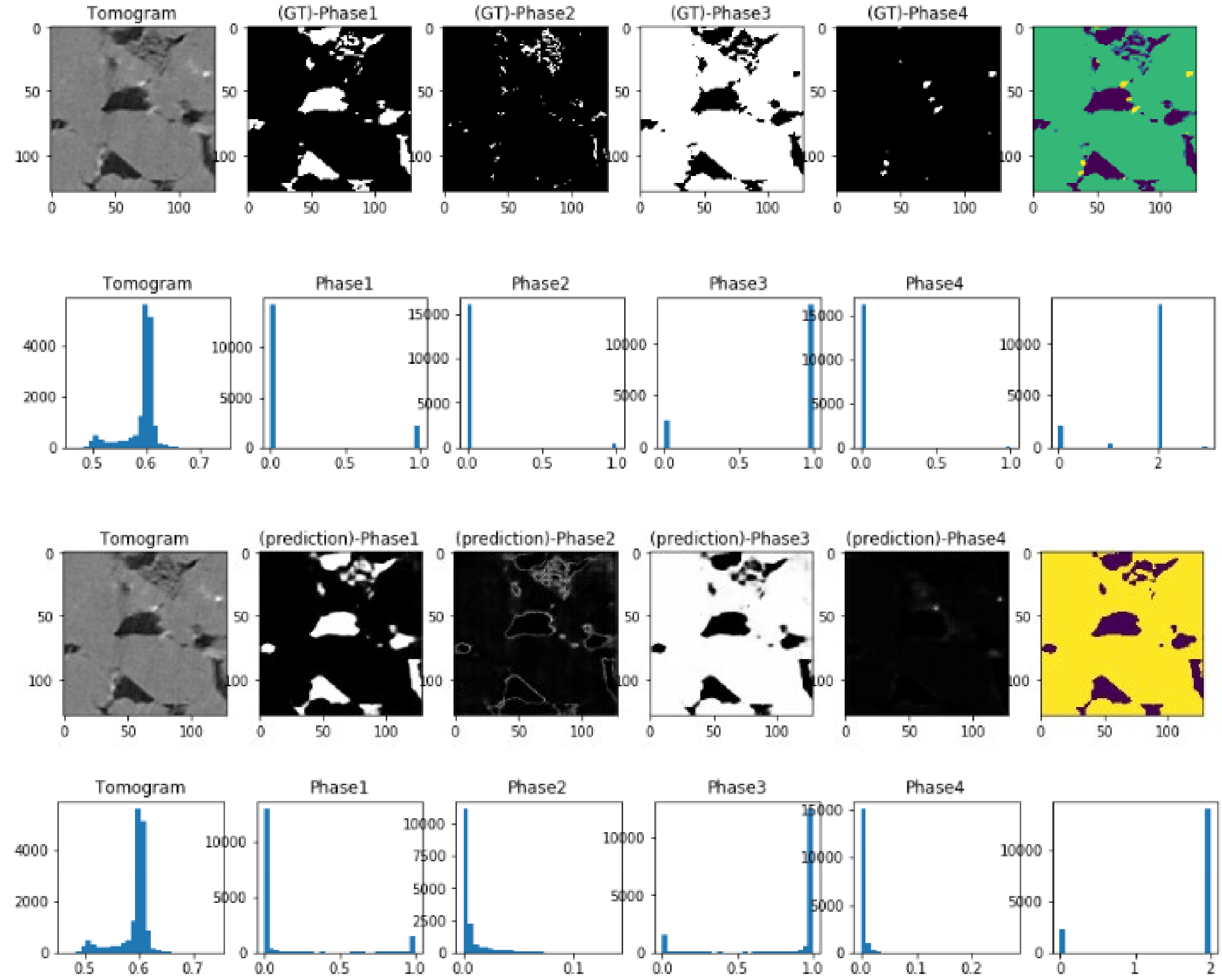
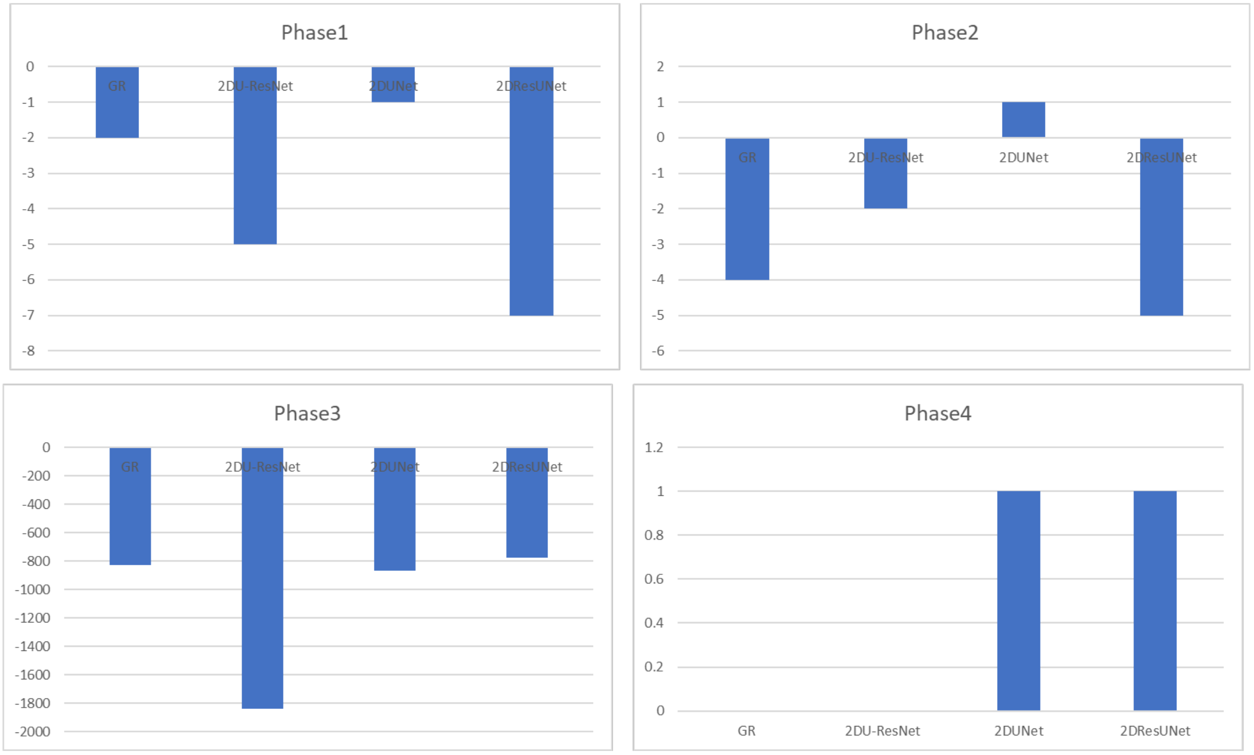


Fig. 23 – 3D ResUNet Prediction vs Ground Truth

Testing Euler Comparison

2D Castlegate



2D Leopard

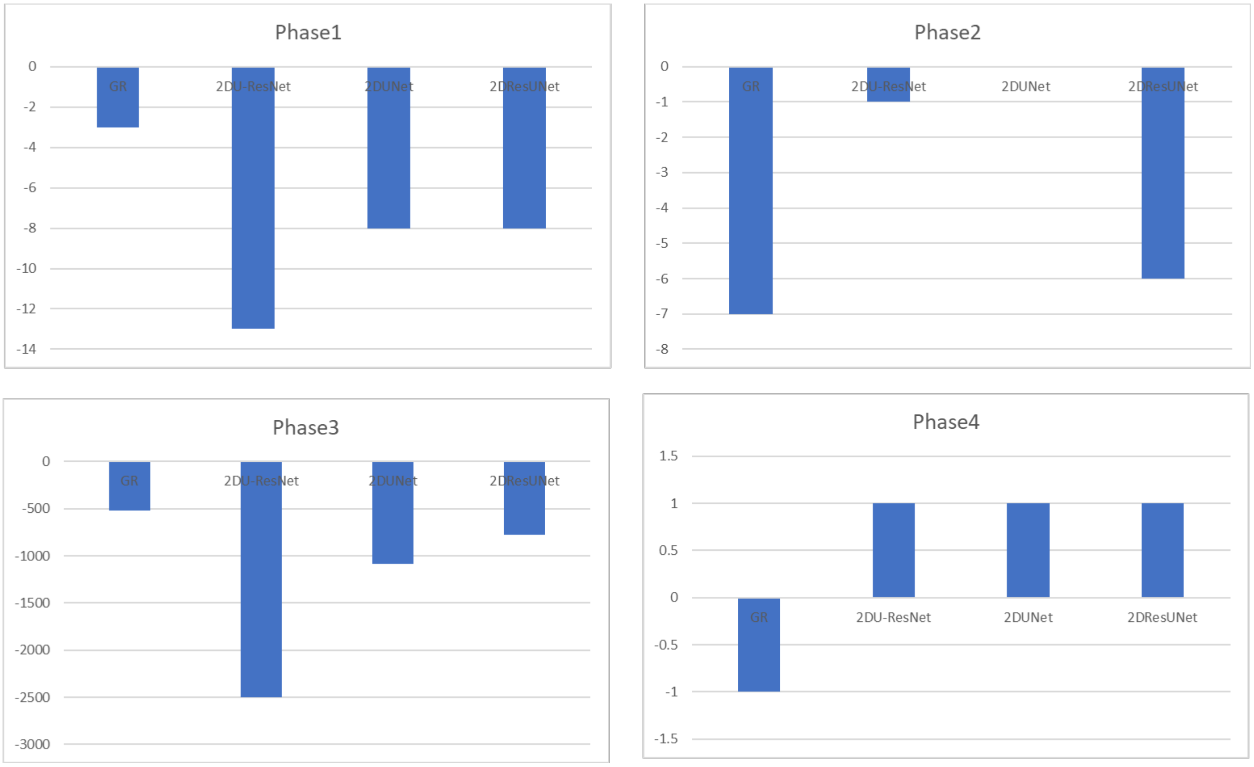
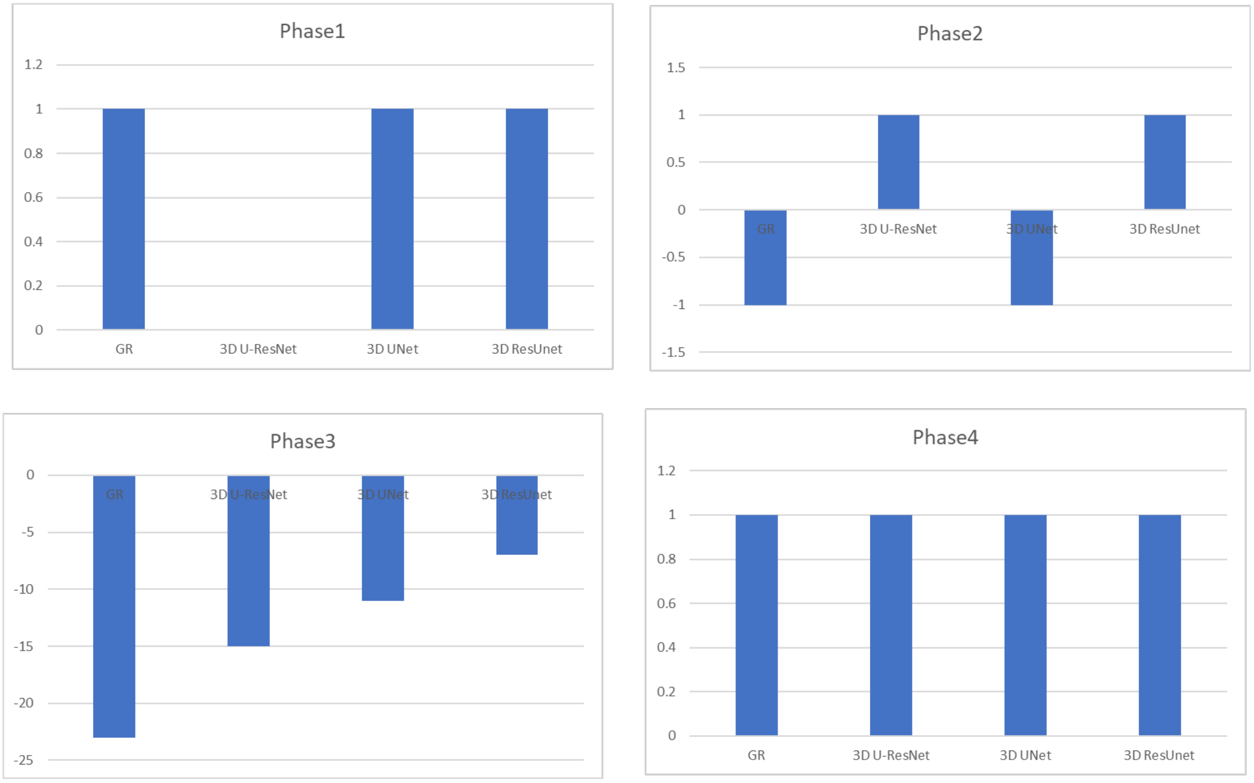


Fig. 23 – 2D Testing Euler Comparisons

3D Bentheimer



3D Leopard

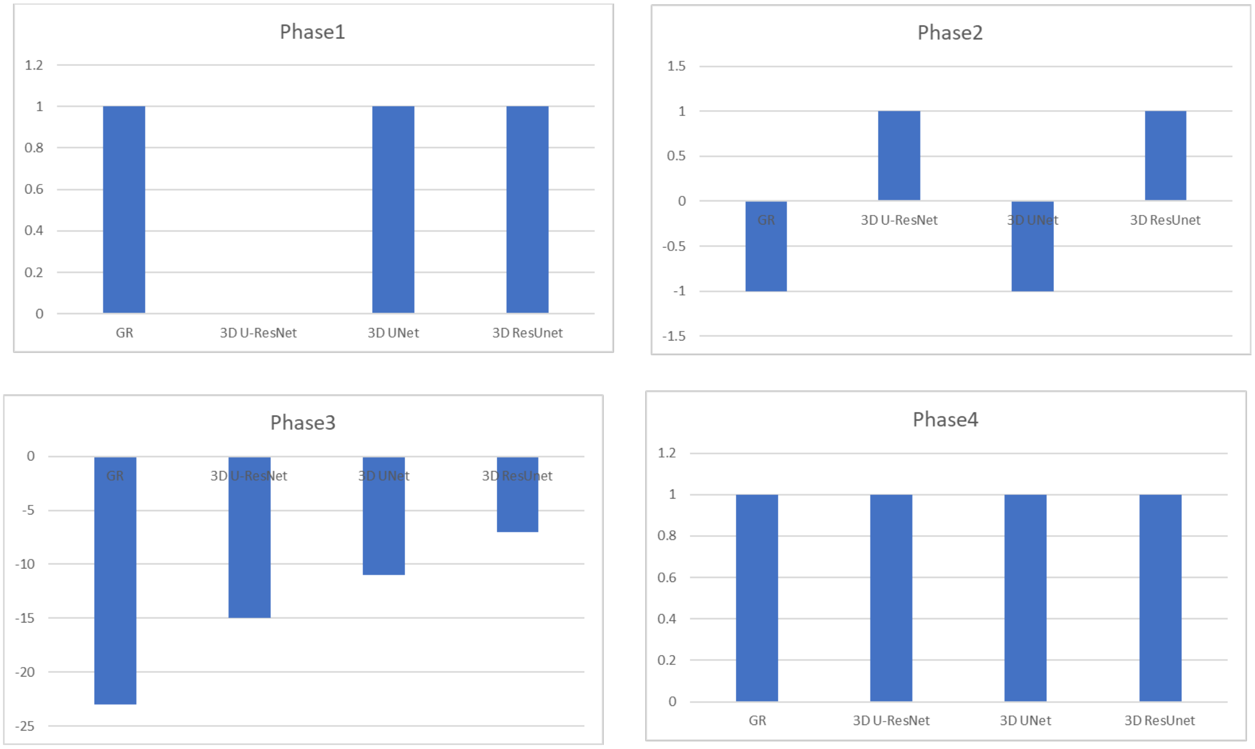


Fig. 24 – 3D Testing Euler Comparisons

	Mean IOU
	Testing
2D *U-ResNet*	0.8373
2D *UNet*	0.8079
2D *ResUNet*	0.8243
3D *U-ResNet*	0.8845
3D *UNet*	0.894
3D *ResUNet*	0.93

Table 3 – Testing Mean IOU

By examining the prediction results from testing datasets, the 2D models showed about 0.80 to 0.83 mean IOU. Note: for 3D models, the IOU for testing is adjusted with threshold applied, which gave us better results than 2D ranging from 0.88 to 0.93. Again, since we have 128 x 128 images in 3D, it's not a fair comparison. The results are very interesting for Euler number for 2D models. All 2D models for Leopard and Castlegate sandstones fails to predict accurate Euler number for phase 1(porosity) when compared to the ground truth (GT), while 2D *ResUNe*t outperforms all other models in other phases and is consistent with the ground truth. The Euler number for 3D models comes very consistent with ground truth as 3D convolutions proves to preserve the connectivity. However, since we are using 128^2 image size and using thresholds to adjust phase 3 and phase 4, it is not a very fair comparison. Like I mentioned before there is significant problem in the prediction results for the 3D models and will be fixed it in the future. However, 3D *UNet* and 3D *ResUNet* performs better then 3D *UNet.*

CONCLUSION AND DISCUSSION

Image segmentation is a user bias task in DRP workflow which utilizes more time and is less efficient. Deep learning CNNs have changed the game in recent years to do fast segmentation without user bias. In this study I trained 2D and 3D *UNet*, *U-ResNet* and *ResUNet* semantic segmentation models on Bentheimer, Leopard and Castlegate sandstones. The goal was to see how each model predict and become generalized when shown to unseen images. The 2D *UNe*t, *U-ResNet*, and *ResUNet* are trained on 134 million pixels and 3D *UNet* and *U-ResNet* are trained on 37 million pixels. Due to parallel computing issues with the Tensorflow package, the image size to train on 3D models was reduced. For 3D models, the phase 2 and phase 4 predictions were significantly different. Some interesting results showed how 2D predictions had higher IOU but the physical accuracy, like Euler number, was significantly different. In all measures, the *ResUNet* model seemed to perform the best in both 2D and 3D over other models as well as the *U-ResNet* for 3D. The results of this study are not limited to sandstone samples, but the network models can be used for any samples. My next step is to correct the 3D models and reevaluate the results.

ACKNOWLEDGMENTS

This was a very intensive project from preparing data to obtaining results. For this project, I want to acknowledge Professor Christoph Arns and Professor Masa Prodanovic for their support and providing me with this research opportunity. I would also like to thank Javier Santos for providing me with resources and advice.

REFERENCES

[1]Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M. Tensorflow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16), Savannah, GA, USA, 2–4 November 2016; pp. 265–283.

[2]Alqahtani, N.; Armstrong, R.T.; Mostaghimi, P. Deep learning convolutional neural networks to predict porous media properties. In Proceedings of the SPE Asia Pacific Oil and Gas Conference and Exhibition, Brisbane, Australia, 23–25 October 2018.

[3]Çiçek, Ö.; Abdulkadir, A.; Lienkamp, S.S.; Brox, T.; Ronneberger, O. 3D U-Net: Learning dense volumetric segmentation from sparse annotation. In International Conference on Medical Image Computing and Computer-Assisted Intervention; Springer: Cham, Switzerland, 2016; pp. 424–432.

[4]Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Martinez-Gonzalez, P.; Garcia-Rodriguez, J. A survey on deep learning techniques for image and video semantic segmentation. Appl. Soft Comput. 2018, 70, 41–65.

[5]Igor Varfolomeev, Ivan Yakimchuk, and Ilia Safonov. An application of deep neural networks for segmentation of microtomographic images of rock samples. Computers, 8(4), 2019.

[6]Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv 2015, arXiv:1502.03167.

[7]Karimpouli, S.; Tahmasebi, P. Segmentation of digital rock images using deep convolutional autoencoder networks. Comput. Geosci. 2019, 126, 142–150.

[8]Pavel Iassonov, Thomas Gebrenegus, and Markus Tuller. Segmentation of x-ray computed tomography images of porous materials: A crucial step for characterization and quantitative analysis of pore structures. Water Resources Research, 45(9), 2009.

[9]Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical Image Computing and Computer-Assisted Intervention; Springer: Cham, Switzerland, 2015; pp. 234–241.

[10]Serveh Kamrava, Pejman Tahmasebi, and Muhammad Sahimi. Linking morphology of porous media to their macroscopic permeability by deep learning. Transport in Porous Media, pages 1–22, 2019.

[11]Wang, Ying Da & Shabaninejad, Mehdi & Armstrong, Ryan & Mostaghimi, Peyman. (2020). Physical Accuracy of Deep Neural Networks for 2D and 3D Multi- Mineral Segmentation of Rock micro-CT Images.

[12]Z. Zhang, Q. Liu and Y. Wang. Road Extraction by Deep Residual U-Net. IEEE Geoscience and Remote Sensing Letters, vol. 15, no. 5, pp. 749-753, May 2018.

[13]Shibuya, N. (2017, November 13). *Up-sampling with Transposed Convolution*. Medium. https://medium.com/activating-robotic-minds/up-sampling-with-transposed-convolution-9ae4f2df52d0

[14]Tiu, E. (2019, August 9). *Metrics to Evaluate your Semantic Segmentation Model*. Medium. https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2

[15]*Semantic Segmentation*. (n.d.) Papers with Code. https://paperswithcode.com/task/semantic-segmentation/latest?fbclid=IwAR2FLPK7cwwJ-_4CVKWYVMG1cgcl-yehFzB3z8ZzTZeK04YyKdtuydiPmuo