

UNIVERSITY OF OKLAHOMA  
GRADUATE COLLEGE

DEEP LEARNING FOR SHALE SEM IMAGE ANALYSIS

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

AMANDA S. KNAUP  
Norman, Oklahoma  
2020

DEEP LEARNING FOR SHALE SEM IMAGE ANALYSIS

A THESIS APPROVED FOR THE  
MEWBOURNE SCHOOL OF PETROLEUM AND GEOLOGICAL ENGINEERING

BY THE COMMITTEE CONSISTING OF

Dr. Carl H. Sondergeld, Chair

Dr. Chandra S. Rai

Dr. Mark E. Curtis

Dr. Deepak Devegowda

© Copyright by AMANDA S. KNAUP 2020  
All Rights Reserved.

Dedicated to  
My advisors, friends, and family

# Table of contents

<b>TABLE OF CONTENTS .....</b>	<b>V</b>
<b>LIST OF TABLES .....</b>	<b>VIII</b>
<b>LIST OF FIGURES .....</b>	<b>X</b>
<b>ABSTRACT.....</b>	<b>XXII</b>
<b>CHAPTER 1:INTRODUCTION.....</b>	<b>1</b>
<b>1.1 Motivation and problem statement .....</b>	<b>1</b>
<b>1.2 Scope of the thesis .....</b>	<b>6</b>
<b>1.3 Organization of the thesis.....</b>	<b>7</b>
<b>CHAPTER 2:BACKGROUND RESEARCH AND LITERATURE REVIEW .....</b>	<b>9</b>
<b>2.1 Shale Microstructure Analysis.....</b>	<b>9</b>
<b>2.2 Machine Learning.....</b>	<b>11</b>
2.2.1 Machine Learning Classifications .....	14
2.2.2 Machine Learning Algorithms .....	17
2.2.3 Model Evaluation During Training .....	26
<b>2.3 Image Preparation/Preprocessing .....</b>	<b>27</b>
<b>2.4 Machine Learning Metrics and Analysis .....</b>	<b>28</b>
2.4.1 Classification.....	28
2.4.2 Image Segmentation .....	31

2.5 GPU Computing.....	34
<b>CHAPTER 3: METHODOLOGY .....</b>	<b>34</b>
<b>3.1 Equipment and Materials .....</b>	<b>35</b>
3.1.1 SEM Sample Preparation .....	35
3.1.2 SEM and Image Acquisition .....	36
3.1.3 Image Classification Computer .....	37
3.1.4 Image Segmentation Computer .....	38
<b>3.2 Evaluation of image quality .....</b>	<b>39</b>
3.2.1 Dataset preparation.....	39
3.2.2 Model Training.....	41
<b>3.3 Image Classification.....</b>	<b>43</b>
3.3.1 Dataset preparation.....	43
3.3.2 Model Training.....	48
<b>3.4 Image Segmentation .....</b>	<b>48</b>
3.4.1 Data Preparation .....	48
3.4.2 Model Training.....	56
<b>CHAPTER 4: RESULTS AND DISCUSSION .....</b>	<b>60</b>
<b>4.1 Evaluation of Image Quality .....</b>	<b>60</b>
<b>4.2 Image Classification.....</b>	<b>63</b>
4.2.1 Method 1 .....	63
4.2.2 Method 2 .....	67
<b>4.3 Image Segmentation .....</b>	<b>69</b>
4.3.1 Method 1 .....	69

4.3.2	Method 2 .....	71
4.3.3	Method 3 .....	73
4.3.4	Method 4 .....	85
<b>CHAPTER 5: CONCLUSIONS .....</b>		<b>88</b>
5.1	Summary .....	88
<b>REFERENCES.....</b>		<b>90</b>
<b>APPENDIX A .....</b>		<b>96</b>

## List of Tables

Table 2.1 – Table of common activation functions used in Neural Networks .....	25
Table 3.1 – Computer Specifications for Image Classification .....	38
Table 3.2 – Computer Specifications for Image Segmentation .....	38
Table 3.3 – GPU comparison.....	39
Table 3.4 – Number of images per classification for evaluating image quality .....	40
Table 3.5 – Separation of data during training .....	42
Table 3.6 – Percent representation of mineral components of Eagle Ford and Woodford images in Method 2 dataset.....	50
Table 3.7 – FTIR Minerology comparison of images taken at same depth.....	50
Table 4.1 – Metrics calculated from the confusion matrix .....	63
Table 4.2 – Evaluation metric calculated from confusion matrix.....	66
Table 4.3 – Evaluation metric calculated from confusion matrix for models 1-3 .....	67
Table 4.4 – Model mean intersection over union comparison between each model tested .....	72
Table 4.5 – Intersection over union per class for each model .....	72
Table 4.6 – Intersection over Union per class.....	74
Table 4.7 – Model predicted porosity and organic matter compared to lab measurements .....	83
Table 4.8 – Intersection over Union metric values for model comparison in mineral segmentation .....	86
Table A.5.1 – Model training parameters for Image Quality .....	96



Table A.5.2 – Initial training parameters used in Formation Classification with seven formations .....	96
Table A.5.3 – Parameters used in Model training .....	97

## List of Figures

Figure 1.1 – Map of US shale plays in the lower 48 states (EIA, 2016) .....	2
Figure 1.2 – Map of Canada and Lower 48 current and prospective shale plays (EIA, 2011) .....	3
Figure 1.3 – Big Data flow model showing how multi-structured data works together to create a data set ready for processing (Cameron 2014) .....	5
Figure 2.1 – (a) Schematic diagram of the FIB-SEM dual beam system. The ion beam (I-Beam) is used to remove material from the surface to create a cross-sectional view that can be imaged with the electron beam (E-Beam). (b) shows a BSE image of a cross-sectioned shale taken with the E-Beam (Curtis et al. 2010).....	10
Figure 2.2 – The procedures of a FIB-SEM reconstruction. (left) a stack of images procured with a FIB-SEM, (middle) a reconstructed volume using image stack, (right) segmentation of images to produce quantitative results for modeling .....	11
Figure 2.3 – Turning Test in which person C attempts to guess if A or B is the Human.	12
Figure 2.4 – Organization of a perceptron. When the retina “views” an image, where certain neurons are activated. These neurons that are activated are known as connections and are transmitted to a projection area. The perceptron will produce a response from the association area where the responses will give feedback back to the association area (modified from Rosenblatt 1958) .....	13
Figure 2.5 – How supervised machine learning works. Step 1) A machine learning method is provided a dataset with labels; the algorithm learns from the dataset. Step 2) The machine is then fed unseen data with unknown labels and labels the new data appropriately (NewTechDojo 2018). .....	15

Figure 2.6 – How unsupervised learning works. Step 1) the machine learning algorithm is provided uncategorized data where the model determines if there are patterns within the data. Step 2) the user observes and learns from the data that the algorithm matched (NewTechDojo 2018). ..... 16

Figure 2.7 – An example of a Decision Tree in determining if a person is fit or not (Kulkarni 2017). The root node is to determine if the person’s age is <30. If yes, then take the right branch. This branch wants to know if the person eats a lot of pizza. If yes, they are considered unfit and if no, then they are considered fit. .... 18

Figure 2.8 – Representation of a biological neuron (Blaus 2014). The dendrites of the neuron cell receive the data form the previous neuron as input (i.e. the neural network layer receives input data from previous layer). The data is processed in the cell body (the layer performing its calculation). If the nucleus data reaches a certain threshold, the neuron will fire an output signal to the next cell (if the layer reaches activation threshold it will sends data onto next layer) . .... 20

Figure 2.9 – Logical computations of neurons. (a) shows when neuron A is activated then neuron C will be activated. (b) shows that when neurons A and B are both activated then neuron C will activate. (c) shows when either neurons A or B are activated then C will activate. (d) shows that when neurons A activate and not B then C will activate (Géron 2019) ..... 21

Figure 2.10 – Example of how CNN layers are processed from the input layers. Each point in the convoluntional layer is associated with a rectangle of the previous layer. (a) represents how the first data point in the first hidden layers is determined with a 5x5 filter. (b) shows how when the 5x5 filter is moved over by one pixel, it creates the second data point in the first hidden

layer. This process is repeated across the entire image to complete the first hidden layer (Nielsen 2019). ..... 22

Figure 2.11 – Example of feature map generation. A vertical filter is used to create Feature Map 1 where as a horizontal filter is used to create Feature Map 2 (Géron 2019)..... 24

Figure 2.12 – Example of how K-fold cross validation is separated out for training and testing (Ren et al. 2019). ..... 27

Figure 2.13 – Example of image augmentations that include rotation, scaling, and shearing (Sudhakar 2017). ..... 28

Figure 2.14 – Representation of a confusion matrix for two classes predicting yes and no. The red box indicated the values used in calculation of recall and the dashed green box is used in the calculation of precision (Drakos 2018). ..... 29

Figure 2.15 – Confusion matrix for multiclass classification. The diagonal represents the true positive value correctly classified and values not along the diagonal are considered false positive or false negative. Precision is determined across each row then averaged while recall is determined along each column the averaged. .... 31

Figure 2.16 – Computing the intersection over union metric as the intersection of the ground truth and the predicted image over the union of both the ground truth and the predicted image (Rosebrock 2016). ..... 32

Figure 2.17 – In this example of how intersection and union are determined, (a) the expected output is the segmentation of the lady vs. the background. (b) the model prediction. (c) The intersection is the overlapping area between the true segmentation and the predicted segmentation. (d) The union is determined between both the combination of both the ground

truth labels and the predicted labels (Jordan 2018). The results from these images are used to determine the intersection of union metric. .... 33

Figure 2.18 – GPU vs CPU (Boston)..... 34

Figure 3.1 – (a) Model 1060™ Fischione mill used in sample preparation. (b) depiction of how sample is milled using ion beam. Two ion guns blast the surface of the sample while the sample rotates to create an even surface (Curtis 2017)..... 36

Figure 3.2 – FEI Helios Nanolab™ 650 DualBeam™ Scanning Electron Microscope... 37

Figure 3.3 – Examples of poor imaging. (a) poor brightness/contrast, (b) charging, (c) contamination of the sample, (d) curtaining effect, (e) out of focus, (f) redeposition, (g) shading, and (h) astigmatism..... 40

Figure 3.4 – An illustration of the architecture of AlexNet, explicitly showing the delineation of responsibilities between two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers (Krizhevsky et al. 2017). Each box is a computational layer that has a height, width and number of feature maps in each layer. CNNs used in classification also include dense layers or classification layers to provide prediction results..... 41

Figure 3.5 – Example of sub-tile generation from SEM images taken from the Marcellus formation..... 43

Figure 3.6 – Distribution of image counts per formation. For Eagle Ford and Vaca Muerta, thermal maturity values were obtained and allowed for separation into oil and gas windows. .... 44

Figure 3.7 – MAPS data consists of hundreds of tiles that are stitched together. These tiles are separated further into sub-tiles for training. A typical tile size is 2048x3027 pixels but

the sub-tile in this report is 512x512 or 128x128 pixels used for classification and segmentation respectively. (a) shows the typical size for a MAPS image with a black box indicating the size of a tile, (b) shows a tile taken from the MAPS image with the size of 512x512 and 128x128 pixels, (c) shows a 512x512 pixel image used for classification and (d) shows a 128x128 pixel image used for segmentation. .... 45

Figure 3.8 – Example of image transformations. (a) Original SEM sub-tile, (b)-(c) reflections across X- and Y-axis, (d) rotation between -3 and 3 degrees, (e)-(f) shearing in X and Y directions, (g)-(h) translate image along X and Y axis, respectively -5 to 5 pixels, (i)-(j) upscaled image to 120% and 80%, respectively along the X-axis, (k)-(l) upscaled image to 120% and 80%, respectively along the Y-axis..... 46

Figure 3.9 – Image datasets generated using image augmentations. Model 1 was the dataset consisting of no additional augmentations. Model 2 utilized the dataset consisting of 5x augmentations meaning each image had up to five augmentations. Model 3 utilized the dataset consisting of 10x augmentations meaning each image had up to ten augmentations..... 47

Figure 3.10 – Example of a Wolfcamp training image generated using method 1. This image was labeled using MATLAB. (a) is the SEM image used as the expected input, (b) labeled images as the expected output..... 49

Figure 3.11 – SEM segmentation dataset example image from the Woodford formation with respective elemental data accompanied by the hand drawn label mask used in training. .... 51

Figure 3.12 – SEM segmentation dataset example image from the Eagle Ford formation with respective elemental data accompanied by the hand drawn label mask used in training. .... 52

Figure 3.13 – Example of how images are labeled in ilastik. (a) the user will draw lines on areas of interest that will be used as input for the random forest model. In this case, the user

draws a red line as training data for pore/organics and a green line to represent minerals. (b)

After the model has run, this is the predicted output. .... 54

Figure 3.14 – Two examples of Eagle Ford images used in segmentation separating out pore/organic, silicates, carbonates and other heavy minerals. On the left is the SEM images with its respective labeled images on the right. .... 56

Figure 3.15 – U-Net CNN architecture. The blue box represents a multi-channel feature map where the number of channels is located above the box in gray. The x-y size of the feature map is located to the left of the box. The arrows between the boxed represent different operations performed between feature maps. (Ronneberger et al. 2015)..... 57

Figure 3.16 – Model architectures tested during Method 2 segmentation. (a) Model 1, (b) Model 2, (c) Model 3, (d) Model 4, and (e) Model 5, Models 4 and 5 are modeled after the U-Net architecture. Models 1-3 (a-c) are considered fully convolutional networks as they provide connections between convolutional layers. Each convolutional layer (large black box) has a filter (blue small box) associated with each layer that traverses the image by filter to produce individual pixels for the following layer. Models 4-5 (d-e) are fully convolutional networks but provide local connections of data from the beginning of the model due to the concatenation or merging of layers. .... 58

Figure 3.17 – Model used in generating pore and organic classification. The layer is the input layer requiring an image of 128x128x1, i.e. a grayscale image. A filter size of 1x1x1 is read to the first convolutional layer. The first convolutional layer is of size 128x128x16 which corresponds to a multi-layered image generated from 16 separate filters. The next convolutional layer has 32 filters that generates 32 images at a size of 128x128. And the pattern continues throughout the rest of the layers. For both layers with 32 filters, they are merged at the end to

provide a single output. The last layer consists of a 128x128x3 image corresponding to the probability masks of each pixel corresponding to each class. .... 59

Figure 3.18 – Models used in Method 4 to segment pores/organics, silicates, carbonates, and other heavy minerals. (a) displays model 1 the filter size (blue box) 1x1 pixels across all five convolutional layers and an output layer of size 128x128x4 pixels. (b) model 2 utilizes a 7x7 filter to generate layer Conv. 2, a 3x3 pixel filter to generate layer Conv. 3 and a 7x7 filter used in a Transpose convolutional layer to up sample the feature image to generate a feature map size of 128x128. .... 60

Figure 4.1 – Prediction results from the image quality dataset. (a) displays a ‘good’ image that was correctly classified as ‘good’ with the network classifying with as 74.5% probability. (b) displays a ‘bad’ image that was correctly classified as ‘bad’ with a 100% probability. (c) is an example of a ‘good’ image that was incorrectly classified. (d) shows an example of an incorrect classification. In this case the image was actually incorrectly labeled in the dataset. The network was able to detect the lower portion of the image is out-of-focus which is considered a win by the network. .... 62

Figure 4.2 – Confusion matrix for evaluation of image quality for determining accuracy, precision, recall and F1 score. The accuracy is defined on the confusion matrix as 95.4% which indicates the classifier is correct 95.4% of the time. The model has a recall of 94.8% and 92.2% for the classes ‘bad’ and ‘good’ respectively (shown in green in the right column), with an average recall of 94.8% indicating a high sensitivity to classifying correctly. The model has a precision of 96.4% and 93.3% for classes ‘bad’ and ‘good’ respectively (shown in green on the bottom row), and an average precision of 94.8% indicating that the model is precise in correctly classifying the data. .... 63



Figure 4.3 – Prediction results in 7 formation classification. (a) displays a correct prediction of a Marcellus predicted with a 96% probability. (b) displays an Eagle Ford tile correctly classified as Eagle Ford. Due to the low accuracy, this model was not always correct and as a result, you should expect to see incorrect predictions. An example of this is shown in (c) where the model incorrectly classified Marcellus as Woodford with 94.2% probability..... 65

Figure 4.4 – Confusion matrix in 7 formation classification. The accuracy was calculated to be 77.5%. The far-right column is the recall for individual classes, with an overall recall of 78% indicates how specific the model is on correctly classifying the formation. The last row is the precision for each formation, with an overall precision of 76.8%. This is an indication of how precise the model is on correctly classifying the formations..... 66

Figure 4.5 – Prediction results taken from Model 3, which was confirmed to be the best model. (a) the model accurately predicted this Marcellus tile with a 77% probability of being Marcellus and (b) the model accurately predicted Eagle Ford Oil tile as Eagle Ford Oil with 55% probability..... 68

Figure 4.6 – Initial segmentation Results Woodford. (a) the SEM image segmented by the CNN U-Net on MATLAB. (b) The network was able to accurately segment mineral but had difficulty differentiating organics and pores. This example does not contain pyrite..... 70

Figure 4.7 – Example CNN U-Net segmentation results using MATLAB. (a) Input SEM image. (b) Segmentation results shows the network was able to accurately able to segment minerals but had difficulty in differentiating between pore and organics in the image. The network was also not able to identify pyrite in the image. .... 71

Figure 4.8 – Segmentation results produced from tested models. (a) shows the Eagle Ford SEM image with (b) the labeled image. Model 1’s predicted image is shown in (c), Model 2 is

shown in (d), Model 3 is shown in (e), Model 4 is shown in (f) and Model 5 is shown in (g). Note that in (d) the that model was able to label pores to some degree whereas models 1-4 could not accurately identify not just pore but also k-feldspar. Due to the differences in network architecture, it is important to the note that the each model tested will have different predictions.

..... 73

Figure 4.9 – Unseen testing set used in model evaluation. (a) SEM image, (b) model predicted image, (c) Ilastik labeled image used for reference, (d) class representation for the pore class as a pixel probability map, and (e) the pixel probability map for organics with red being 100% probability being the given class and purple having the least probability of being the given class..... 75

Figure 4.10 – Unseen Woodford image taken several years prior to the Woodford MAPS training set and segmented with a convolutional neural network. The network was able to accurately segment pores and organics and also able to identify inorganic and mixed pores located in the upper left corner. .... 76

Figure 4.11 – Eagle Ford SEM image segmented using a convolutional neural network. The Network did fairly well in this image. It was able to accurately segment the pores and pore backs in the top right corner of the image..... 77

Figure 4.12 – Marcellus formation SEM image segmented using a convolutional neural network. The network was able to accurately label pores and organics in the image. The network was also able to identify, and label clay minerals located within the organics. .... 77

Figure 4.13 – Vaca Muerta SEM image segmented using a convolutional neural network. The network was able to accurately differentiate between the organics and the minerals within the clay cluster in the middle of the image. .... 78

Figure 4.14 – Osage Formation (STACK) SEM image segmented using a convolutional neural network. The network was able to segment the nano-pores in the images but incorrectly labeled them as organic. More training data would resolve this issue in the network. .... 78

Figure 4.15 – Vaca Muerta (oil window) MAPS image used for petrophysical comparison to lab measurements. This is an example of 1 out 144 tiles from a MAPS image on what is considered a representative elementary area..... 79

Figure 4.16 – (a) Marcellus 6XXX.10’ example tile from MAPS tile set used in estimating petrophysical properties using Representative Elementary Area. (b) This image was segmented using a convolutional neural network and able to accurately identify clay minerals within the organics and identify organics within the pyrite framboid..... 80

Figure 4.17 – (a) Marcellus 6XXX.35’ tile from MAPS tile set used in estimating petrophysical properties using Representative Elementary Area. (b) This image was segmented using a convolutional neural network and able to accurately identify the cracks within the organics located in the top right corner of the image..... 81

Figure 4.18 – (a) Marcellus 6XXX.62’ tile from MAPS tile set used in estimating petrophysical properties using Representative Elementary Area. (b) This image was segmented using a convolutional neural network and able to accurately differentiate organics and clay in the middle of the image. .... 82

Figure 4.19 – 3D FIB-SEM segmentation of the Eagle Ford. (a) shows the gray scale 3D representation, (b) the Total Organic Matter, (c) combined Organic Matter and Porosity, (d) total porosity, (e) pockets of connected porosity shown within the 3D scan..... 84

Figure 4.20 – Pore size distribution derived from 3D segmentation of FIB-SEM scan... 85

With great success with Method 3, we wanted to fine tune the model by testing filter size on applications to mineralogy. Results from this method are shown in **Figure 4.22**, **Figure 4.23** and **Figure 4.24**. From these images, we see that Model 3 with a 7x7 filter does a better job at smoothing and differentiating between grain boundaries. In **Figure 4.22**, overall Model 1 appears to have better labeling and smoothing of the grain boundaries. In addition, it appears that the network did better in correcting silicate labels from the true label as compared to model 2. In **Figure 4.22**, we see similar results to **Figure 4.22**, where Model 1 appears to perform well and even do better than human labeling of the image. Model 1 also appears to begin to label some of the minerals within the organics whereas the hand labeled images incorrectly labeled it as organic. This shows promise in accurately identifying mineral groups in SEM images. However, in **Figure 4.21**, we see that both models are struggling to identify the mineral structure as carbonate. This is especially the case along the grain/organic boundaries..... 85

Figure 4.22 – Prediction results from unseen test set for models 1 and 2. (a) Displays prediction results from model 1 and (b) prediction results from model 2 with a filter size 7x7. From both of these models, we actually see the predicted results are actually better than their hand labeled images, especially along the boundary..... 86

Figure 4.23 – Prediction results testing models 1 and 2. (a) Displays prediction results from model 1 with a filter size of 1x1 and (b) displays prediction results from model 2 with a filter size 7x7. From both of these models, we actually see the predicted results are actually better than their hand labeled images by providing smoother boundaries between mineral grains. .... 87

Figure 4.24 – Example of what the convolutional neural network is confusing. . (a) Displays prediction results from model 1 with a filter size of 1x1 and (b) displays prediction

results from model 2 with a filter size 7x7. This result shows that Model 2 is the best when identifying large areas of carbonates. More training data and network modification will help in correcting incorrect labeling such as these models. .... 88

Figure A.5.1 – AlexNet architecture showing delineation of responsibilities between multiple GPUs during the training process. Each GPU communicates with the other GPUs on specific layers. This is shown in between the second and third layer and between the fully connected layers at the end of the network (modified from Krizhevsky et al. 2017; Knaup et al. 2019) ..... 97

## **Abstract**

The microstructure of unconventional reservoir rocks not only controls the storage and transport of hydrocarbons but also controls the mechanical properties of the shale. Scanning Electron Microscopy (SEM) has been valuable in understanding the microstructure of reservoir rocks. However, quantitative image analysis has been proven to be difficult. There are many limitations to image analysis that produce significant errors in determining areal porosity and organic matter content within shales. Current methods in building a suitable database for statistical analysis is time intensive, requires a trained technician, and cannot deal with the thousands of images already collected. This research evaluates the application of machine learning, more specifically Deep Learning, to reduce the time required to analyze SEM images from days, for a single large-area high-resolution MAPS area to a matter of seconds for a single image.

The objective of the initial work presented was to determine if there were significant microstructural differences between different formations that could be captured by computer software. In order to avoid acquiring large amounts of data required for training a network from scratch, the technique called transfer learning was applied to the pretrained convolutional neural network (CNN) AlexNet (Krizhevsky et al. 2017). This technique allows a user to re-teach the pre-trained network to focus on a new dataset than it was originally trained on. The dataset used comprised of 27,000 images (each 512x512 pixels) from 18 different formations spanning range of maturities. Results from this study generated probabilities of classification in association with different formations. Images with higher probability to other formations other than the intended label suggests there are microstructural similarities between formations. This work proved that convolutional neural networks can learn to identify features from the shale microstructure with

an accuracy of 92%. As a result, this method was applied for classifying image quality with reasonable accuracy of 95% accuracy.

In addition to classification, CNNs can be applied to individual pixels within an image for classification. This is known as image segmentation. The focus in this topic is the identification and quantification of discrete objects such as pores, grains, organics, etc. applied directly to SEM images. When the model was applied to a large-area, high resolution maps with a large enough representative area (REA), it can provide representative and accurate results of area porosity and organic matter content (OM), consistent with lab measured porosity and TOC values. Accuracy of segmentation range from 92-99% for intersection over union metric (IoU) when classifying pore, OM and mineral content. Direct inspection of the images when compared to data generated using the Ilastik software proved to surpass the random forest method by more accurately defining boundaries between labels. The model was trained using Woodford images but was able to be successfully applied to images from other formations such as the Marcellus, Vaca Muerta, and Eagle Ford shales in addition to the Osage formation in the STACK play. This method was then expanded to identify carbonates, silicates, and other heavy minerals in addition to pore and organics. A sensitivity study was done in order to determine the best model. The sensitivity study was done to determine whether deeper or shallower models performed better with the data, more or less convolutional layers in the model, or a narrower or wider model performed better with the data, more or less filters per convolutional layer. This research shows that applications of CNNs to shales can quickly and accurately provide results in identifying similar formations in addition to features of interest.

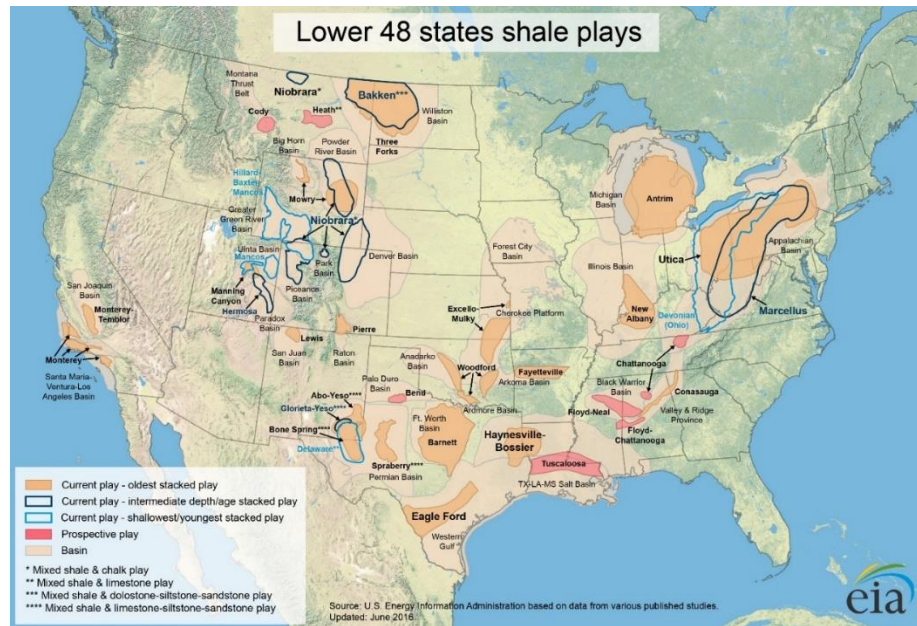
# Chapter 1: Introduction

## 1.1 Motivation and problem statement

Organic rich shales have typically been treated as source rock and seals for conventional reservoirs. In recent years, tight reservoirs and organic rich shales have emerged as a major source of production within the United States. The constancy in shale production is associated with the size of the resources available, which extend nearly 500,000 square miles, as well as improvements in technology which allow for the development of these resources at a lower cost (EIA 2020). The distribution and location of current and prospective shale formations within the United States can be seen in **Figure 1.1** and within North America in **Figure 1.2**. For the contiguous American shale plays, the technically recoverable crude oil is estimated at 246 BBL and technically recovered dry natural gas is 2,341 Tcf (EIA 2020). Of these values, approximately 174 billion barrels of crude oil and 1,611 Tcf of natural gas are technically recoverable from tight/shale reservoirs (EIA 2020).

Compositionally, there is no difference between the oil and gas of conventional and unconventional reservoirs, they differ primarily in the method of extraction. The choice of method is mainly controlled by the microstructure of the reservoir. Conventional reservoir rocks have favorable storage porosity and permeability which allow the hydrocarbons to be easily stored and produced, often flowing naturally. However, an unconventional reservoir is typically the source rock for conventional reservoirs; characterized by ultra-low permeability, low porosity and organic matter. Unconventional reservoirs have hydrocarbons that have never migrated, they are still sequestered in the host shale. Due to the tight nature of unconventional reservoirs, it becomes imperative to understand the microstructural properties, such as mechanical properties, in order to be able effectively produce from them by being able to predict how the formation will react during the production and completion processes.





**Figure 1.1 – Map of US shale plays in the lower 48 states (EIA, 2016)**

In unconventional reservoirs such as tight-sands, shales, and coalbed methane formations, petroleum resides in the micro- and nano-pores, as well as being absorbed within the organics (Ambrose et al. 2010). Most of the current investigation surrounding these pores has been made using indirect measurements at a macroscopic scale to quantify pore volume and describe pore structure (Curtis et al. 2010). Nuclear magnetic resonance (NMR) spectroscopy, mercury injection capillary pressure (MICP), and Brunauer-Emmett-Teller (BET) surface area analysis, are the just some of the most common indirect measurement methods for measuring porosity and characterizing pore structure on a macroscopic scale. While these methods are very effective in characterizing the pore structure, they do not generate a detailed image of the pore structure.

Standard scanning electron microscopy (SEM) and focused ion beam SEM (FIB-SEM) produces 2D and 3D images of pores at the nanometer scale. In SEM, a 2-D image is generated by bombarding the surface of a sample with a beam of electrons and then detecting the various signals produced by the interaction of the beam with the sample’s surface (Goldstein 2003; Reimer 1998;

Rodriguez et al. 2014). The SEM also provides surface topography through secondary electrons (SE) and atomic composition through backscattered electrons (BSE). In addition, using a different type of detector, elemental composition can be determined using energy dispersive x-ray spectroscopy (EDS). This detector allows for spot analysis in the mapping of elements across the surface. FIB-SEM uses an ion beam to slowly mill away the surface, while a series of successive images are taken in order to generate a stack of images for a 3D representation. Through a combination of these methods, we gain a 2D representation of the rock's matrix with mineralogical information. And, by using FIB-SEM imaging to reconstruct a 3D representation of the matrix, we can visually understand the microstructure and its connectivity at a very fine scale.



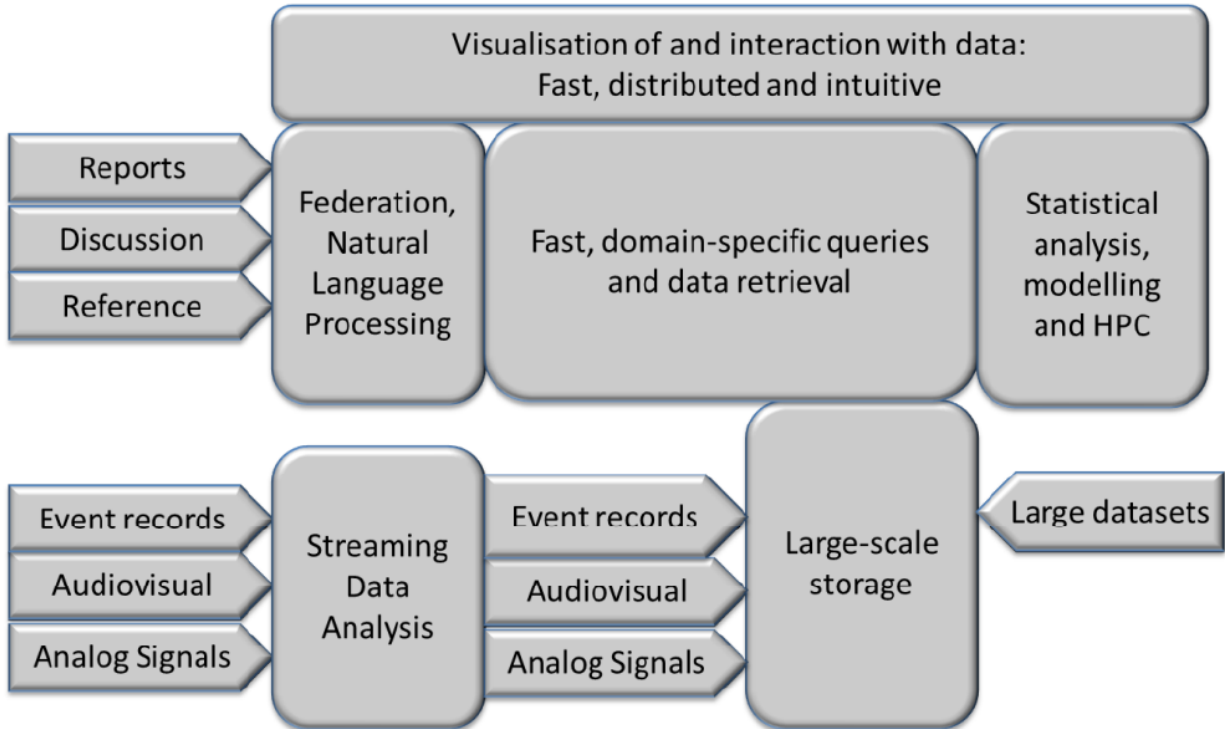
**Figure 1.2 – Map of Canada and Lower 48 current and prospective shale plays (EIA, 2011)**

SEM imaging provides detailed surface and topographic information and 3D reconstructions capture connectivity; however, some of the main disadvantages are the size and

cost of the instrument as well as the resolution of the instrument. Sample preparation is particularly important in as it could result in imaging artifacts if done incorrectly.

In the context of this study, we are interested in determining pore size distributions, organic matter content, and mineralogy based on a single SEM image. This procedural method is called digital rock physics (DRP), and it investigates and calculates petrophysical properties through image analysis, when these properties would typically be derived through core analysis (Kalam 2012). The idea behind DRP is to provide high resolution representations of the complex pore geometry for other forms of analysis. In any DRP workflow, the first step prior to analysis is the segmentation of images, based not just on simple grayscale thresholding, but on multi-band thresholding (Andrä et al. 2013; Iassonov et al. 2009). Improper segmentation can result in misleading outcomes or even entirely inaccurate representations. Current methods of segmentation are time intensive and are done manually with the help of commercial software.

Due to the growing amount of data in the oil and gas industry, there has been a push to automate processes and apply new technologies to handle large data sets. This trend is also known as ‘Big Data’ or ‘Big Data analytics’. A typical dataset involved with big data embodies six main characteristics: volume, variety, velocity, veracity, value and complexity (Mohammadpoor et al. 2018; Ishwarappa et al. 2015). Much of the data provided includes unstructured data, which is data that is unorganized or may be text heavy and multi-structured data. Multi-structured data is typically a result of multiple people or machine interactions. An example of how multi-structured data might be used within a company is shown in **Figure 1.3**.



**Figure 1.3 – Big Data flow model showing how multi-structured data works together to create a data set ready for processing (Cameron 2014)**

Due to the massive amount of data continually generated on a daily basis, and the complex nature of the issues that are required to be solved within the oil and gas industry, current methods of analysis are prohibitively time and effort consuming. Artificial intelligence (AI) applied in the oil and gas industry has provided cost saving measures by quickly and robustly analyzing the different problems. Machine learning (ML) and AI have provided a means to create tools that enable the development of assets quickly with minimal downtime. Examples of groups within a company that would benefit from these workflows include reservoir engineering, geology, petrophysics, geophysics, production, and operations.

With the development of new technologies, image analysis has become easier thanks to deep learning (DL) and artificial neural networks (ANN). These methods, if trained properly, can provide for rapid and simple analysis of the image data. In many forms of image analysis, such as

formation classification and feature extraction, an ANN is trained to learn specific details about the image. Currently, work is being done to determine petrophysical properties from images. In the future, the hope is to predict petrophysical properties such as mineralogy and Young's modulus, as well as porosity and organic content, from images of the microstructure.

## **1.2 Scope of the thesis**

The goal of this thesis is to present a comprehensive method for evaluation of SEM image quality, shale play identification, and an automated and robust method of segmentation of shale SEM images in order to identify areal porosity, organic matter and mineralogy. SEM imaging analysis can be directly correlated to measurements made on core. Conventional methods require core plugs which can be expensive to obtain, whereas SEM images can be obtained more cheaply from cuttings and less consolidated samples. Initial work began with the evaluation of image quality and shale play identification to determine if convolutional neural networks could identify the differences in grayscale images for the image quality question and identify plays based purely on microstructure.

In the evaluation of image quality, the focus was to determine whether an SEM image can be classified as 'good' or 'bad'. The class of bad would include images that contained artifacts from imaging, such as curtaining, brightness/contrast, and charging, or indications of poor sample preparation. The purpose of this step was a proof of concept to determine whether or not images taken from different laboratories met the standards that are needed in following models of segmentation and play identification. The shale identification portion of this thesis was done as a proof of concept to determine what can be learned from shale microstructure and to see if a model could learn the differences that are apparent to humans upon visual inspection.

Initial success with the classification of images, allowed for a narrowing of the scope of the project from single image classification to classification of single pixels within an image, also known as segmentation or feature extraction. In providing a method of feature extraction from SEM images, this thesis addresses the automation of image segmentation of large areas, high resolution MAPS images, 3-D volumes from FIB-SEM, and micro-CT images, in order to generate volumetric measurements. Earlier works include the results of Tran (2017), Curtis et al. (2014) and Goergen et al. (2014).

The objectives of this thesis are to: 1) automate feature extraction of SEM images by being able to process a large number of images quickly, and 2) work with data that might not be of the highest quality, (i.e. noisy, presence of artifacts, poor contrast, etc.). Current methods of SEM image analysis can be costly because they require a skilled technician, which can create user bias in the analysis, so the downsides are two-fold. Another prohibitive consideration is the time-consuming nature of the work, which can limit the number of images that can be examined. The method proposed in this thesis allows for the segmentation of a large-area, high resolution MAPS image taken at 10 nm resolution, up to  $\sim 1.8 \text{ mm}^2$  in area, in order to provide a quantitative analysis of areal porosity and organic matter content with lab comparison measurements.

### **1.3 Organization of the thesis**

This thesis is presented in five chapters and is organized as follows:

- Chapter 2 introduces background research in understanding shale microstructure analysis, SEM in shale analysis, and machine learning, (or more specifically, convolutional neural networks)

- Chapter 3 reviews the equipment, methodologies and machine set up for running convolutional neural networks
- Chapter 4 reviews the results acquired by classification and segmentation of shale SEM images using convolutional neural networks
- Chapter 5 concludes the investigation with takeaways and a review of the most significant findings of this project

## **Chapter 2: Background Research and Literature Review**

### **2.1 Shale Microstructure Analysis**

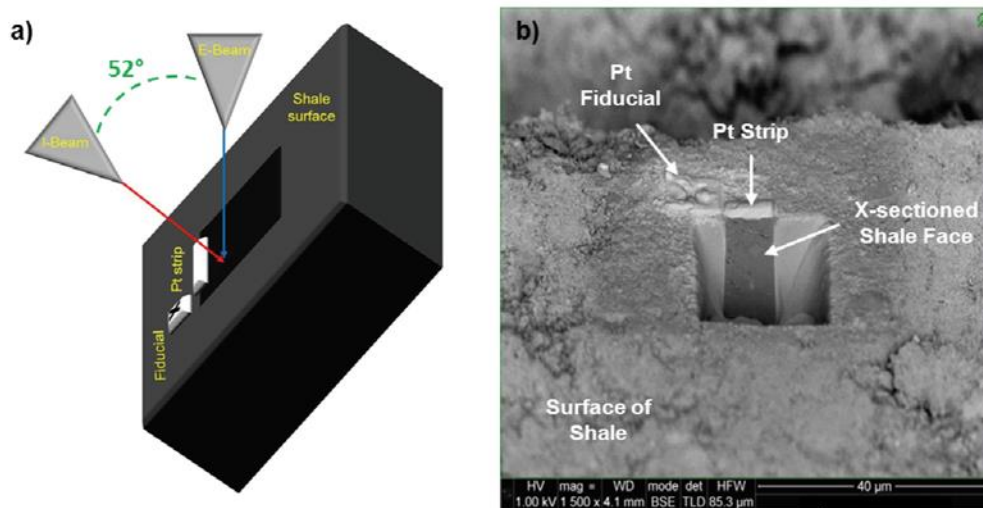
Shale microstructure is complex in nature but is essential to understand for shale production. Shale reservoirs are difficult to characterize due to the significant portion of their total porosity resides at the nano-scale (Clarkson et al. 2013). Many early studies viewed shales and other fine-grained rocks exclusively as seals for conventional reservoirs due to their low permeability and source rocks for other reservoirs (Brace 1980, Saif et al. 2017). For conventional reservoirs, the API method of measuring porosity is well documented and is determined through the combination of bulk density and grain density.

Both permeability and porosity are controlled in shale by microstructure. Knowledge of pore size leads to an understanding of fluid flow through the reservoir. Current and most popular macroscopic techniques for the measurement of porosity includes mercury injection capillary pressure (MICP) and nuclear magnetic resonance (NMR) relaxation spectroscopy. More recently, the use of focused ion beam (FIB) and scanning electron microscopy (SEM) have allowed to visualize the micro- and nano-scale structures in 3D.

Scanning electron microscopy, transmission electron microscopy, and optical microscopy have been used to identify lithology, pore space, interconnectivity of pores, and grain size within a rock matrix (Prasad 2001; Sok et al. 2010; Knackstedt et al. 2013). These techniques allow for mapping of large areas; however, these methods are limited in resolution. Image acquisition using SEM resolves small features, but with a favorable resolution comes a limited field of view.



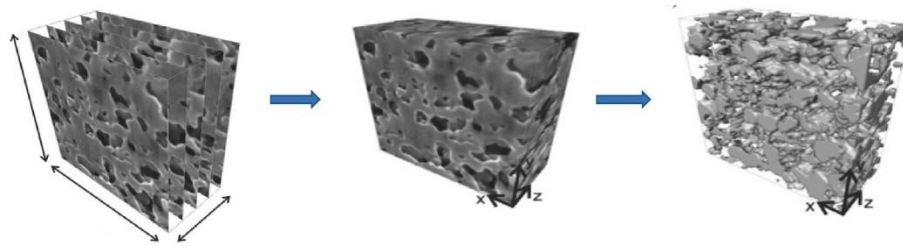
Using a combination of focused ion beam SEM, or FIB-SEM, one is able to investigate the microstructure at micro- and nano-scale in 3D. Traditional optical methods such as SEM images can reveal the surface and 2-D morphologies. Disadvantages to this method includes the destructive nature of gathering images for the 3-D representation but with high resolution of up to 2nm/pixel. **Figure 2.1**, shows a schematic representation on how images are gathered in FIB-SEM imaging.



**Figure 2.1 – (a) Schematic diagram of the FIB-SEM dual beam system. The ion beam (I-Beam) is used to remove material from the surface to create a cross-sectional view that can be imaged with the electron beam (E-Beam). (b) shows a BSE image of a cross-sectioned shale taken with the E-Beam (Curtis et al. 2010).**

X-Ray computer tomography (CT) is another method for gathering 3-D representations of shales. Current technologies allow for the resolution to be in the micrometer range. The technique is specifically called microtomography and is referred to as micro CT or  $\mu$ CT.  $\mu$ CT is a non-destructive method of imaging and is a combination of a series of 2-D x-ray images that are rotated around a single axis and reconstructed to form a single 3-D image. A voxel, or volume pixel, is associated with the x-ray absorption at the single point (Knackstedt et al. 2004; Monteiro et al. 2017; Herman 2009; Saif et al. 2017).

In the case of shales, a 3-D approach facilitates obtaining important information on spatial distributions of organic matter, pore distributions, pore shape, connectivity etc. The scanning electron microscope is a powerful tool that produces images by scanning a samples surface with a focused beam of electrons. These electrons interact with the atoms of the surface of the sample to provide a wide range signals that contain information about the topography and composition of the surface of the sample.

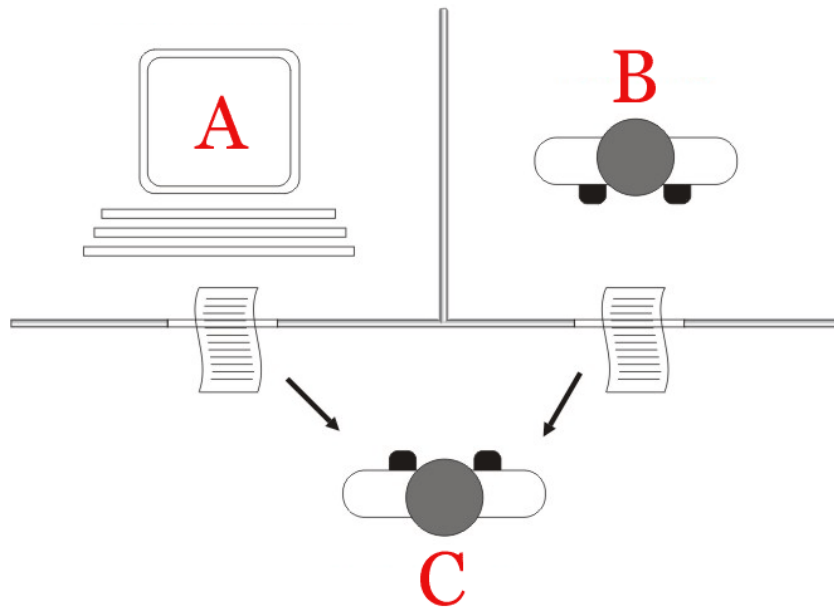


**Figure 2.2 – The procedures of a FIB-SEM reconstruction. (left) a stack of images procured with a FIB-SEM, (middle) a reconstructed volume using image stack, (right) segmentation of images to produce quantitative results for modeling**

## **2.2 Machine Learning**

Machine Learning (ML) has become an important tool in modern research and analysis. It is a method of data analysis that allows a user to automate model building and detect meaningful patterns in the data. It stems from artificial intelligence (AI) which is based on the idea of allowing a computer to mimic human behavior and thinking (SAS, 2019). Advances in hardware and computing technologies have allowed machine learning to advance. These algorithms automatically build mathematical models using training data to make decisions based on what was provided. The basis of ML has been largely modeled after the brain and how neurons in our central nervous system interact. Although machine learning has been around since the 50s, recent advances in computational power and algorithms have made advances in AI more notable.

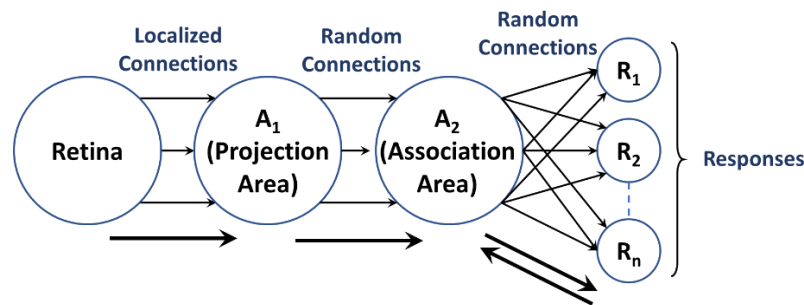
One of the first tests of AI was done by Alan Turing in 1950. This test, called the “Turing Test”, and it was done to determine if a computer could obtain real intelligence. The test was to see if the computer could learn to fool the human into believing the computer was a human. An interrogator would question both the human and the computers labeled X and Y and they would not know which was labeled which. Based on the interview, the interrogator would be asked to determine the label of the computer and the human. An example of how the test was conducted is shown in **Figure 2.3**.



**Figure 2.3 – Turing Test in which person C attempts to guess if A or B is the Human**

This breakthrough in understanding if computers can think gave way to Arthur Samuels AI program of checkers. In 1952, Samuel wrote the first computer learning program that taught a machine to play checkers better than himself and today he is considered one of the pioneers of machine learning (Puget 2016). The Samuel Checkers-playing Program is considered to be one of the first self-learning programs and it also demonstrates some of the fundamental concepts of AI. In Samuel’s 1959 paper on machine learning using the game checkers, he discusses different

procedures used in the development of his program, such as the Neural Network approach. According to Samuel, this approach involves a reward-and-punishment routine by inducing behavior by randomly assigning values to nodes similar to backpropagation in neural networks that will be discussed in later chapters. The second approach he discusses in the paper is using a highly organized network. Although the organized network is considered more efficient than the Neural Network approach, it becomes inefficient for new applications as it requires reprogramming for each new application. Samuels machine-learning program was one of the earliest examples of non-numerical computation (Wiederhold et al. 1990).



**Figure 2.4 – Organization of a perceptron. When the retina “views” an image, where certain neurons are activated. These neurons that are activated are known as connections and are transmitted to a projection area. The perceptron will produce a response from the association area where the responses will give feedback back to the association area (modified from Rosenblatt 1958)**

The next advance in ML was the creation of the perceptron by Frank Rosenblatt at the Cornell Aeronautical Laboratory in 1958. Rosenblatt states that a perceptron is designed to illustrate some of the fundamental properties of intelligent systems, i.e. the nervous system in biological systems. The organization of the perceptron model, shown in **Figure 2.4**, follows such that an optical pattern affects the retina of a sensory unit which is assumed to respond as an all-or-nothing basis. Once these responses are activated, they are then transmitted to a set of cells known as a projection area. Several connections received by this area are algebraically summed to be greater than or equal to a certain threshold. Once the threshold has been reached, the connection

fires off to send an output response. During the creation of this algorithm, its original uses were intended to be a machine, rather than a program, for image recognition. Its first implementation was in the software for IBM 704, where it was implemented as “Mark 1 perceptron” (Foote 2019).

While the perceptron seemed promising as image recognition and object detection, it struggled with new problems especially in detecting faces which inherently stalled much of the research in this topic. As a result, it was several years before the resurgence of neural networks in the 1990s and advancements in computational power did neural networks become successful.

Following the Samuel Checkers-playing program and the struggle of the perceptron model, in 1967, the “nearest neighbor” algorithm, also known as k-nearest neighbor algorithm, was written that allowed computers to gain basic pattern recognition (Foote 2019). This method is used for both classification and regression and is based on feature similarity. Typical problems included the traveling salesman problem and optimization for shortest path problems.

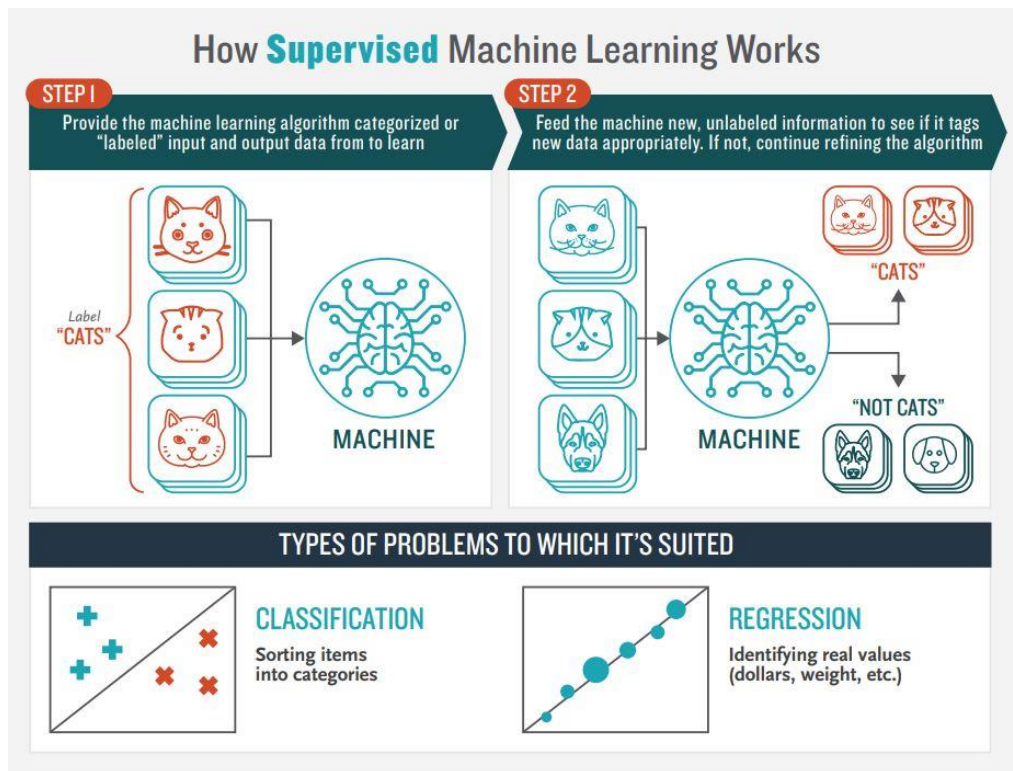
### ***2.2.1 Machine Learning Classifications***

Machine Learning tasks are classified into different categories such as supervised learning, unsupervised learning, and reinforcement learning. In the following sections we will discuss the uses of each type of learning.

#### **2.2.1.1 Supervised Learning**

In supervised learning, the algorithm used is trained on labeled data and maps inputs to desired outputs (Osisanwo et al. 2017). This is the most common learning method used by most ML applications. Supervised learning is often used in classification and regression problems since the goal is to have the program learn from a classification system already created and minimize the error with respect to the given inputs (Ayodele 2010; NewTechDojo 2018). This method is

highly dependent on pre-determined classifications which can be influenced by biases in the labeling of the data. A workflow of the supervised learning process is shown in **Figure 2.5**. A significant portion of the process comes with data pre-processing and model train set definitions. In addition, all of these are affected by the Algorithm Selection, which will be discussed later.

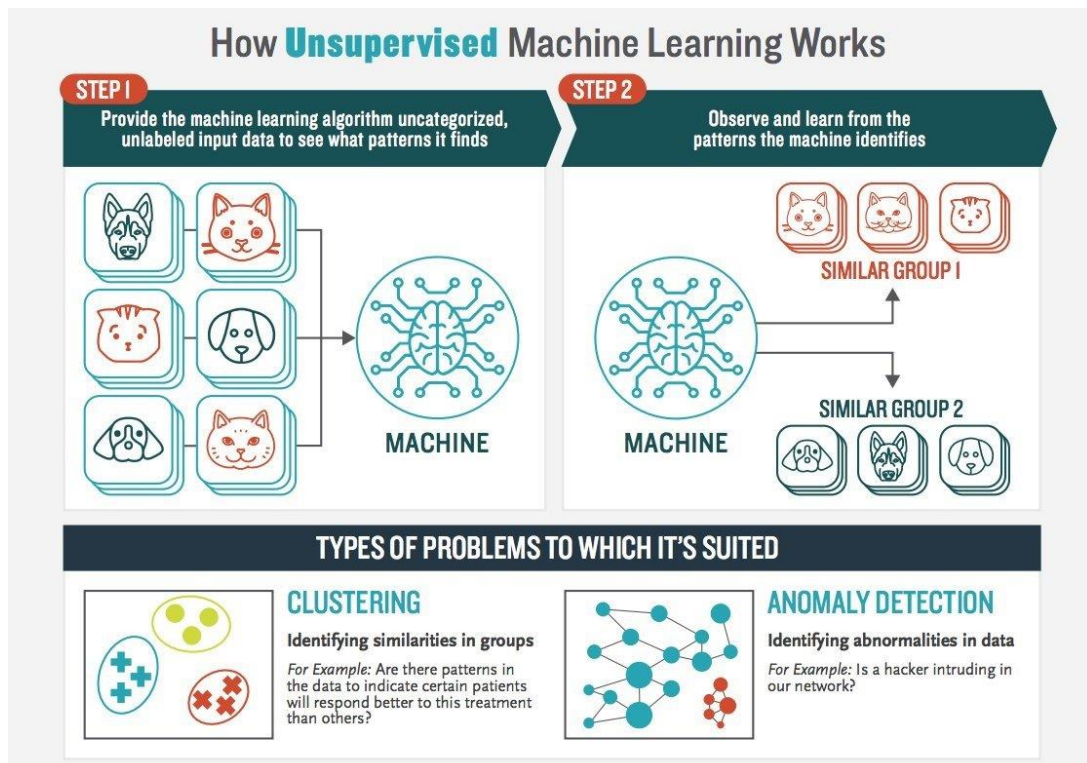


**Figure 2.5 – How supervised machine learning works. Step 1) A machine learning method is provided a dataset with labels; the algorithm learns from the dataset. Step 2) The machine is then fed unseen data with unknown labels and labels the new data appropriately (NewTechDojo 2018).**

Typical machine learning algorithms include decision trees, naïve Bayes classification, Support Vector Machine (SVM), random forest, linear regression, ordinary least squares regression, logistic regression, and neural networks just to name a few. We see that for supervised learning it is common to have training data labeled for classification. These methods take in a sequence of desired inputs. The goal of these methods is to produce the correct output given a new and unseen input (Gharhramani 2004).

## 2.2.1.2 Unsupervised Learning

Unlike supervised learning, unsupervised learning's goal is to learn or detect patterns that were not labeled. For this we are interested in seeing if a model can detect patterns within the data without giving explicit labels on the data. There are two main types of unsupervised learning, including clustering and association (NewTechDojo 2018). In the categorizing approach, the ultimate goal is to teach the computer by not giving explicit examples as you normally would in supervised learning. This can be considered a form of reinforcement learning as often the computer can be rewarded for taking certain actions and punished for others (Ayodele 2010).



**Figure 2.6 – How unsupervised learning works. Step 1) the machine learning algorithm is provided uncategorized data where the model determines if there are patterns within the data. Step 2) the user observes and learns from the data that the algorithm matched (NewTechDojo 2018).**

In unsupervised clustering, we are interested in finding similarities in the data. The assumption is that within the clusters, the data will match reasonably well with other data points

within the cluster. This method works well when it is provided with enough data. Unsupervised learning can also experience overfitting if not carefully applied to the data. The ultimate goal of unsupervised learning is to help model or even structure the data to help the user understand more about the data (NewTechDojo 2018). Unlike supervised learning, unsupervised learning does not have correct answers and these algorithms are used to present the structure of the data for easy analysis in the future. Examples of unsupervised learning are clustering and dimension reduction. This can be done using Principal Component Analysis (PCA), K-means clustering, and Independent Component Analysis (ICA).

### **2.2.1.3 Reinforcement Learning/Semi-Supervised Learning**

Reinforcement learning is similar to both supervised and unsupervised learning. In reinforcement learning, the model interacts with its environment by producing certain actions (Ghahramani 2004). Some forms of reinforcement learning can be used in unsupervised cases in categorizing of data. The interactions with the environment can be rewarded or punished based on the results. The ultimate goal is to minimize the punishments the machine receives.

### **2.2.2 *Machine Learning Algorithms***

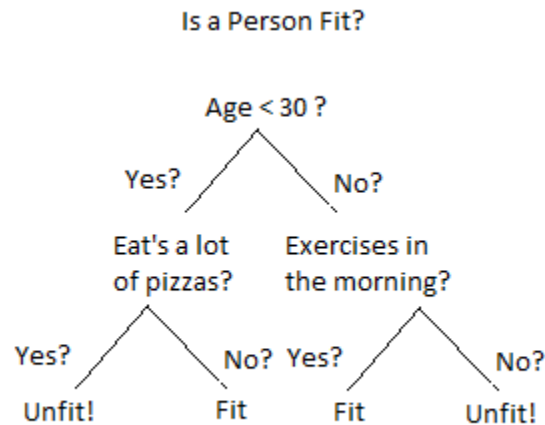
Machine learning comprises of many different algorithms that are classified as supervised, unsupervised or reinforcement learning. The following section discusses various types of ML algorithms used in this project.

#### **2.2.2.1 Decision Trees and Random Forest**

Decision tree is a machine learning model that creates a list of branches from observations in the data to the targeted value in the leaves. This can be used to visually and explicitly represent



decisions and decision making (Gupta 2017). In the process of creating a decision tree, the root node has a condition that splits the tree into branches. These branches either have another node or what splits to another branch or it does not split and becomes a leaf where it has been classified.



**Figure 2.7 – An example of a Decision Tree in determining if a person is fit or not (Kulkarni 2017). The root node is to determine if the person’s age is <30. If yes, then take the right branch. This branch wants to know if the person eats a lot of pizza. If yes, they are considered unfit and if no, then they are considered fit.**

In classification using Decision trees, the data is continuously split according to certain parameters. An example is shown in **Figure 2.7**. Here the purpose is to determine whether a person is considered fit. To start off, the root node initially determines the age of the person; whether or not they are younger than 30. If the answer is yes, take the left node, and if the person is older than 30, take the right node. Say the person is 35, we take the right node. From here, this brings us to the first branch, determining whether or not this person exercises in the morning or not. In our case, we will say that they do exercise in the morning, so we take the left node and determine that they are fit.

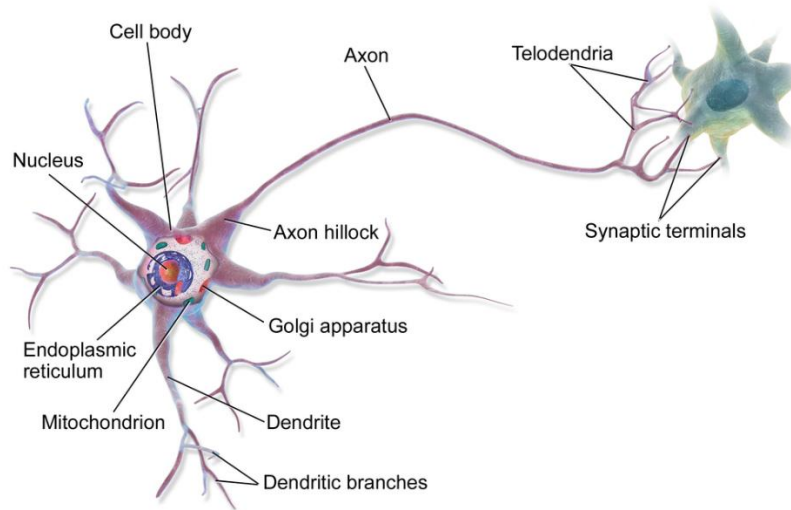
Random forest is similar to decision trees. In fact, random forest is made up of a collection of decision trees. When decision trees get too deep, meaning they have too many

branches downward, overfitting of the data tends to happen (NewTechDojo 2018). In the process of creating a random forest model, multiple decision trees are grown to prevent overfitting of the data. This method provides an average of the different trees which helps to reduce the variance in the data. The two main concepts that give the name random is that there is random sampling of the data during the training process as well as a random subset of features are used when considering the splitting of nodes.

Both decision trees and random forest are great in accurately providing classification and regression. Random forest is extremely flexible with the data and returns a high accuracy and also can maintain a high accuracy when a large proportion of the data is missing. When implemented correctly, these methods provide easy and accurate results. One disadvantage of random forest is that it is computationally expensive for large forests.

#### **2.2.2.2 Neural Networks**

Artificial Neural Networks (ANNs) are a machine learning model that are inspired by the networks of biological neurons found in brains. They also constitute the core that makes up Deep Learning (Géron 2019). These models are very versatile and provide a means to tackle complex tasks such as image classification, speech recognition, text recognition, image segmentation etc. In 1943, McCulloch and Pitts presented a simplified computational model of how biological neurons interact with each other and the mapping of nervous activity and neural events through logical calculus. This is an example of the early success of ANN development.

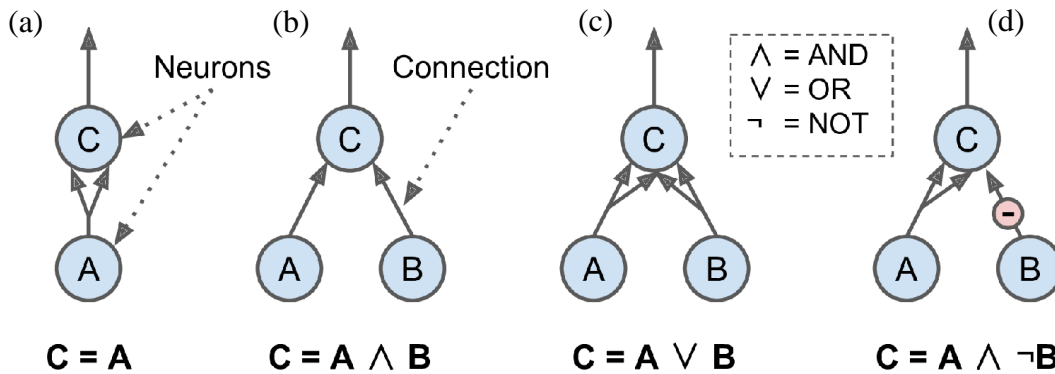


**Figure 2.8 – Representation of a biological neuron (Blaus 2014).** The dendrites of the neuron cell receive the data form the previous neuron as input (i.e. the neural network layer receives input data from previous layer). The data is processed in the cell body (the layer performing its calculation). If the nucleus data reaches a certain threshold, the neuron will fire an output signal to the next cell (if the layer reaches activation threshold it will sends data onto next layer) .

When understanding how neurons interact with one another in a neural network, it becomes important to understand how they are related to their biological counterparts. The main components consist of the cell body and an axon. The cell body is where most of the complex actions occur in the neuron. Once the nucleus has reached its threshold requirement, the neuron activates and sends the data to the next neuron. The nervous system is highly organized and each of these neurons can be connected to thousands of other neurons (Géron 2019).

This first example of an artificial neuron was created by McCulloch and Pitts (1943) and was considered a binary model. This model was activated once a certain number of inputs were activated. They were able to create networks where logical expressions, such as and, or, and not, are able to be expressed. An example is shown in **Figure 2.9a-d**. **Figure 2.9a** shows that if neuron A gets activated then Neuron C will also be activated. **Figure 2.9b** shows that when both neurons A and B are activated then C will activate. In **Figure 2.9c**, it shows that when either A or B are

activated then C will activate, and the last case, **Figure 2.9d**, shows that when A is activated and B is not then C will activate.



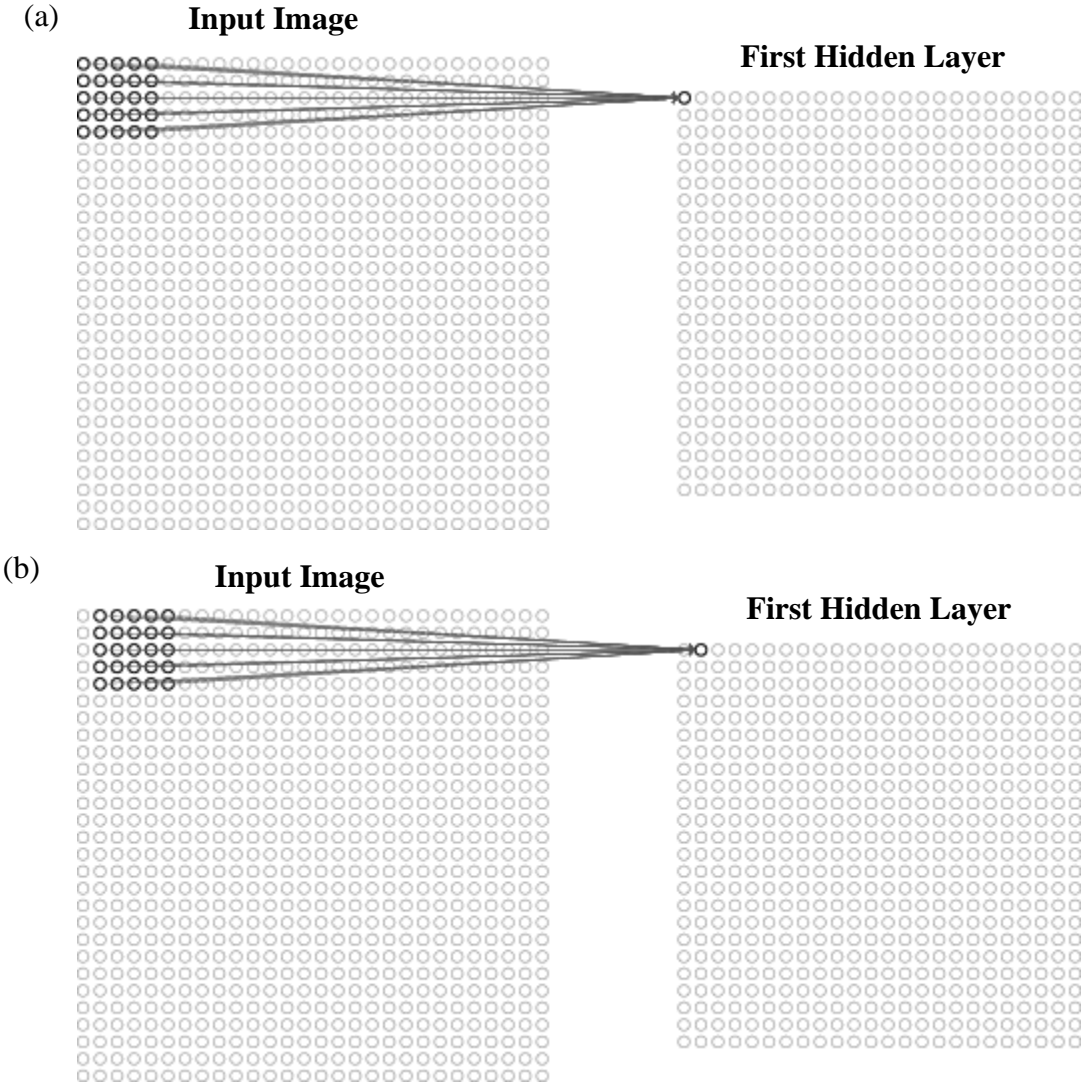
**Figure 2.9 – Logical computations of neurons.** (a) shows when neuron A is activated then neuron C will be activated. (b) shows that when neurons A and B are both activated then neuron C will activate. (c) shows when either neurons A or B are activated then C will activate. (d) shows that when neurons A activate and not B then C will activate (Géron 2019)

As mentioned previously, the perceptron, created by Rosenblatt (1957) is one of the simplest ANN architectures. Shown in **Figure 2.4**, displays how multiple neurons will connect from the retina to the predicted responses. These neurons are typically organized into multiple layers where neurons of one layer only connect to the layers of the preceding layer.

### 2.2.2.2.1 Convolutional Neural Networks

Convolutional Neural Networks (CNN)s, also known as convnets are a specific type of ANN that have demonstrated excellent performance in the classification of images (LeCun et al. 1989). They are considered a regularized version of a multilayered perceptron. In designing a CNN, there are multiple hidden layers. These layers focus on the bulk of the calculations within the network. In addition to convolutional layers, layers such as pooling, padding, and fully connected layers are used in creating the ANN classifier.

Convolutional layers are the building blocks to CNNs. Unlike ANNs, where each datapoint (pixel in the case of images) is associated to a neuron, each neuron in the preceding layer is connected to only the neurons associated with a rectangle in the previous layer. Using this method allows the network to focus on low level features in the first layers then assemble them to higher level features in the following layers (Géron 2019).



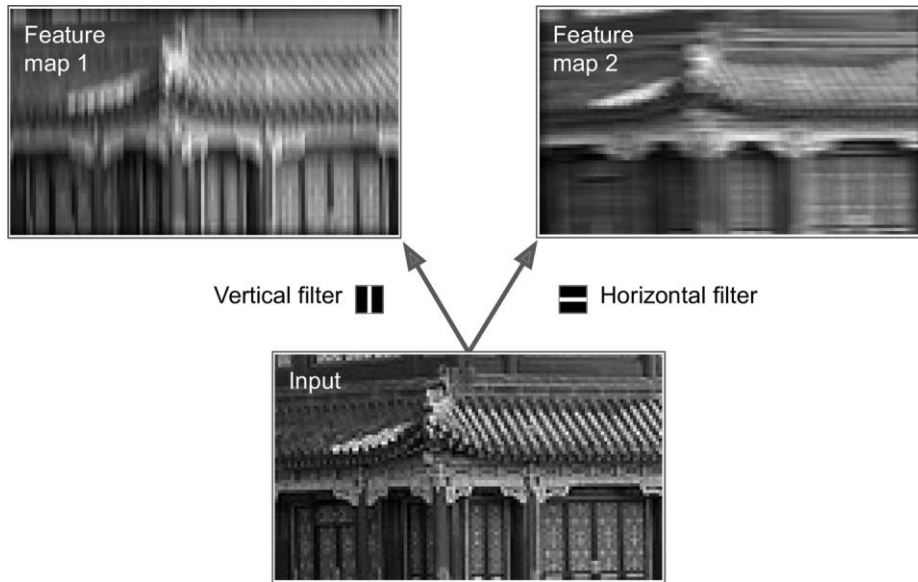
**Figure 2.10 – Example of how CNN layers are processed from the input layers. Each point in the convolutional layer is associated with a rectangle of the previous layer. (a) represents how the first data point in the first hidden layers is determined with a 5x5 filter. (b) shows how when the 5x5 filter is moved over by one pixel, it creates the second data point in the**

**first hidden layer. This process is repeated across the entire image to complete the first hidden layer (Nielsen 2019).**

Filters in convolutional layers are associated matrix functions where the filter is applied across the image to produce results. An example is shown in **Figure 2.10** where the input image, 28x28 pixels, is applied with a 5x5 filter that is moved across the first five layers of the image and applying matrix multiplication to generate the first row in the first hidden layer. Once the filter has reached the end of the row, the filter will shift down one row and repeat the process until the 5x5 filter has seen every pixel in the input image. This process will generate a 24x24 image in the first hidden layer (convolutional layer). For each neuron in the input, the equation below represents how each neuron in the first hidden layer is calculated.

$$\sigma \left( b + \sum_{r=0}^4 \sum_{c=0}^4 w_{r,c} a_{(j+r),(k+c)} \right)$$

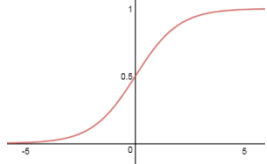
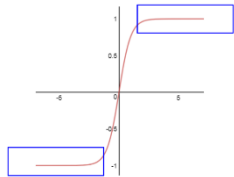
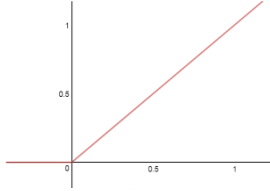
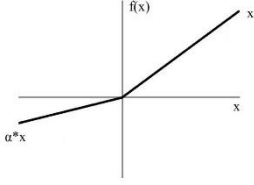
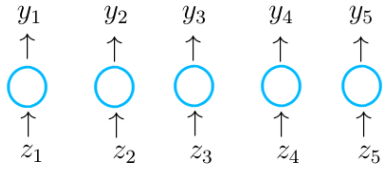
In the hidden layer calculation,  $\sigma$  represents the activation function, which will be discussed later in the chapter.  $b$  is the shared value for the bias term, which is similar to an intercept value in regression.  $a_{(j+r),(k+c)}$  is the input patch in the image where  $j$  and  $k$  are the location in the input image, and  $w_{r,c}$  is the 5x5 array of weights that are used in generating the new 24x24 image. The weights used in generating this layer indicate that all the neurons that were created in this first hidden layer will detect the same feature, such as a horizontal line or a vertical line. This will create a feature map of the input image (Nielsen 2015; Géron 2019). An example of a feature map generated in a convolutional layer is shown in **Figure 2.11**. Using the equation mentioned above, Feature map 1 and Feature map 2 are generated using a vertical and horizontal filter, respectively.



**Figure 2.11 – Example of feature map generation. A vertical filter is used to create Feature Map 1 where as a horizontal filter is used to create Feature Map 2 (Géron 2019).**

Activation functions are used as a gateway to proceed with a convolutional filter. This is based on whether the neuron’s input is relevant for the model’s prediction (MissingLink.ai 2020). Many neural networks used a non-linear activation function since they allow for more complex mapping between the inputs and outputs of the data. This is important when analyzing datasets that include text, images, video and audio. Some of the most common non-linear activation functions are sigmoid, tanh, Rectified Linear Unit (ReLU), Leaky ReLU, and softmax. A table containing their function and their respective graphs is shown in **Table 2.1**.

**Table 2.1 – Table of common activation functions used in Neural Networks**

Activation Function	Equation	Plot
Sigmoid	$\text{sigmoid}(x) = \frac{e^x}{1 + e^x}$	
Tanh	$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$	
Rectified Linear Unit (ReLU)	$f(x) = \max(0, x)$	
Leaky ReLU	$f(x) = \max(0.1x, x)$	
Softmax	$y_i = \frac{e^{z_i}}{\sum_{i=0}^m e^{z_i}}$	

Loss functions are used during training of a model to optimize the algorithm. The goal of the loss function is to minimize the error generated during training and to determine how well a model is performing given specific data as the price paid for the inaccuracy of a prediction (Mahendru 2019). Some common loss functions used in machine learning are binary cross entropy and categorical cross entropy.

Binary cross entropy is used in two-class classification, such as determining if an image is a cat or a dog. The definition of entropy is generally used to indicate disorder or uncertainty.



This type of loss is used for classification models which provides a probability,  $p$ , for a certain class,  $y$ . The equation is shown below:

$$L = -y \cdot \log(p) - (1 - y) \cdot \log(1 - p) = \begin{cases} -(\log(1 - p)), & \text{if } y = 0 \\ -\log(p), & \text{if } y = 1 \end{cases}$$

This is also called Log-Loss where the probability,  $p$ , is determined using the activation function for final classification.

In the case of multi-class cross entropy, it is calculated similarly to binary cross-entropy. It will compare the distributions of the predictions, i.e. the activation function in the output layer, to the probability of the true class. For the loss, an input vector  $X_i$  and its corresponding target vector  $Y_i$  are used in the equation below:

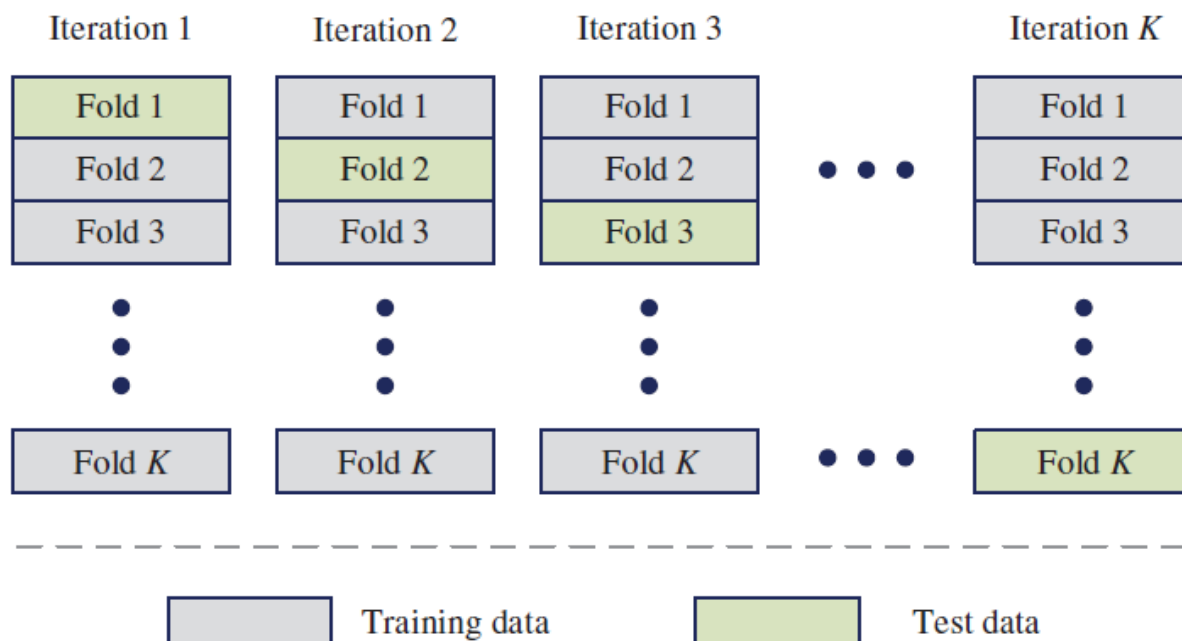
$$L(X_i, Y_i) = - \sum_{j=1}^c y_{ij} \cdot \log(p_{ij})$$

where  $Y_i$ , in a target vector in  $(y_{i1}, y_{i2}, \dots, y_{ic})$  where  $c$  is the total number of classes and  $p_{ij}$  is the probability that the  $i^{\text{th}}$  element is in class  $j$ .  $y_{ij}$  is defined below:

$$y_{ij} = \begin{cases} 1, & \text{if } i^{\text{th}} \text{ element is in class } j \\ 0, & \text{otherwise} \end{cases}$$

### **2.2.3 Model Evaluation During Training**

During model training, it is important to test the model with an unbiased way to determine the model skill on unseen data and to prevent overfitting of the data. The purpose of separating the dataset into training, validation, and testing is to provides an unbiased evaluation of the model during the training phase (Brownlee 2017). The validation set is used in evaluating the model as it is being trained on the training dataset to help tune the model's parameters. The test data is a training, set



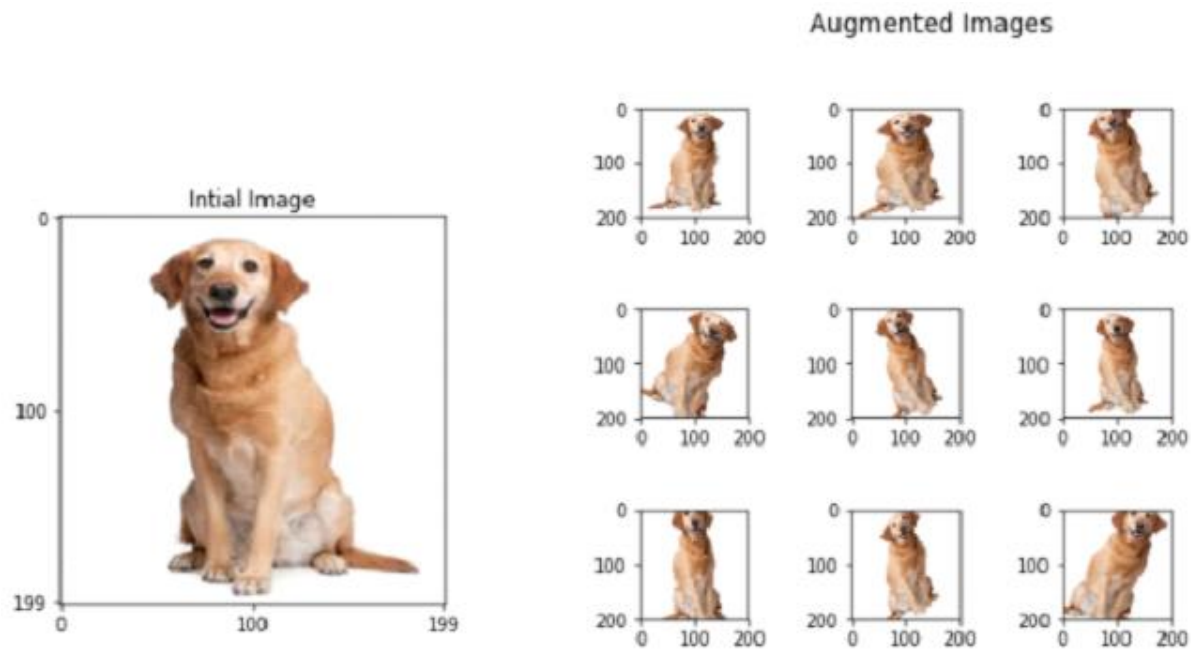
**Figure 2.12 – Example of how K-fold cross validation is separated out for training and testing (Ren et al. 2019).**

A popular method to use during training is K-Fold cross-validation. This method is a resampling procedure used in model evaluation on a limited data sample (Brownlee 2019). K-Fold Cross Validation uses a single parameter,  $k$ , to indicate the number of subsets of data that it will be separated into. It is a popular method as it generally results in a less biased model than a test/train split of the data. To begin, the data is randomly shuffled, before being separated into  $k$ -partitions of the data. An example of the  $k$ -fold process is shown in **Figure 2.12**. For each unique set,

### 2.3 Image Preparation/Preprocessing

Image preparation is key when beginning training. Some pre-trained networks require specific input sizes while others allow you to adjust the input size to fit the needs of your images. In addition, Image augmentation is a technique that allows a user to increase the dataset. Some

parameters that are generally used include, zoom, shear, rotation, translation, and reflection. An example is shown in **Figure 2.13**.



**Figure 2.13 – Example of image augmentations that include rotation, scaling, and shearing (Sudhakar 2017).**

## **2.4 Machine Learning Metrics and Analysis**

Understanding how well the model has performed is important but understanding how each metric applied to different scenarios is important to having a valuable model. This section of the thesis will discuss typical metrics used in machine learning classification and image segmentation.

### **2.4.1 Classification**

One of the most common and easiest to understand metrics in classification is the confusion matrix. It is mainly used in classification problems where the output is typically one or more classes and used when the true values are known. It is called a confusion matrix to show

where the model is confusing the data. This is a matrix that described the performance of a machine learning model on testing data (Drakos 2018). Accuracy, precision, recall and F<sub>1</sub> scores can be calculated using the values found in confusion matrices. True positive values (TP) and True Negative (TN) are when both the actual label and the predicted label of the image are the same. False positive (FP) and False Negative (FN) are when the actual label and the predicted label are not the same. In **Figure 2.14**, TP is determined when the actual label ‘yes’ is accurately predicted ‘yes’ and TN values are determined when ‘no’ is accurately predicted ‘no’.

	Actual = Yes	Actual = No
Predicted = Yes	TP	FP
Predicted = No	FN	TN

**Figure 2.14 – Representation of a confusion matrix for two classes predicting yes and no. The red box indicated the values used in calculation of recall and the dashed green box is used in the calculation of precision (Drakos 2018).**

Accuracy is the most straight forward measure of classifiers and is defined below:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

It is defined as the sum of the true positives and true negatives over the sum of all responses and is a value between 0 and 1. Unbalanced dataset may cause higher accuracies when in reality it should not be. As an example, you are interested in determining the difference between cats and dogs. In your data set you have 90 images as dogs and 10 images as cat. If you were to predict

that every image you see is a dog, then the accuracy would be 0.9 or 90%. As a result, accuracy is a good measure when the classes within a dataset are balanced (Drakos 2018).

Recall or sensitivity or True Positive Rate (TPR) is defined as the number of items that are correctly classified that are identified as positive (Drakos 2018). The red box in **Figure 2.14** shows the values that are used in calculating recall. The equation is shown below:

$$recall (R) = \frac{TP}{TP + FN}$$

Recall tells the most about a model's performance with respect to how many were incorrectly classified. It is not so much about correctly guessing but more about determining which answers were correctly classified as true and yes.

Precision is the number of items that are correctly identified as positive out of total items identified as positive (Drakos 2018). The green dashed box in **Figure 2.14** shows the values from the confusion matrix that are used in calculating precision. The equation is shown below:

$$precision (P) = \frac{TP}{TP + FP}$$

In the case of precision, it is more about being precise and about how correct the answer was to the real response. The main goal of precision is to minimize the number of false negatives by having recall as close to 100% as possible, however, to minimize the number of false positives, then the goal would be to make precision as close to 100%.

F1 Score is a single-number evaluation metric that makes comparing models easier. Just like in regression, when comparing multiple models adjusted  $R^2$  is the single-number evaluation metric that considers the number of variables in your model,  $F_1$  score allows you to quickly compare classifier models. This value is comprised of both precision (P) and recall (R). The equation is shown below:

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} = \left( \frac{2}{R^{-1} + P^{-1}} \right)$$

For multiclass classification, the actual values are still represented by columns and the predicted values are represented by rows. For a single class, the true positive is the diagonal position, the false positive is the sum of the column excluding the diagonal value, and the false negative value is the sum of the row excluding the diagonal. Precision and recall are all calculated similarly as above for each class then averaged to generate overall values. F<sub>1</sub> score is calculated the same as binary classification.

		Actual		
		Yes	Maybe	No
Predicted	Yes	True Positive (TP)	False Positive (FP)	False Positive (FP)
	Maybe	False Positive (FP)	True Positive (TP)	False Positive (FP)
	No	False Positive (FP)	False Positive (FP)	True Positive (TP)

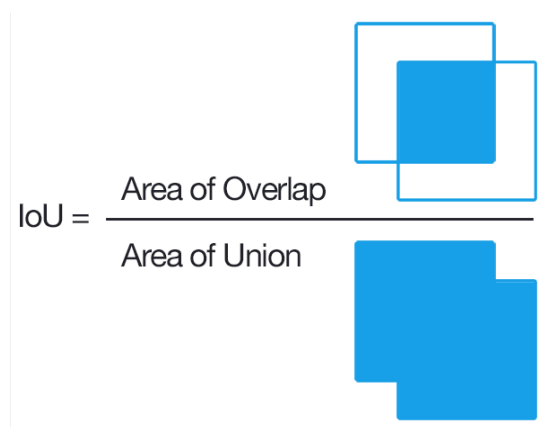
**Figure 2.15 – Confusion matrix for multiclass classification. The diagonal represents the true positive value correctly classified and values not along the diagonal are considered false positive or false negative. Precision is determined across each row then averaged while recall is determined along each column the averaged.**

### 2.4.2 Image Segmentation

In image segmentation, a pixel is assigned a probability of belonging in a specific class (Higham 2018). There are three main metrics used in segmentation, pixel accuracy, intersection

over union, also known as the Jaccard Index, and the Dice Coefficient, also known as the  $F_1$  score. Using a combination of these three metrics, you will know if the model is performing well.

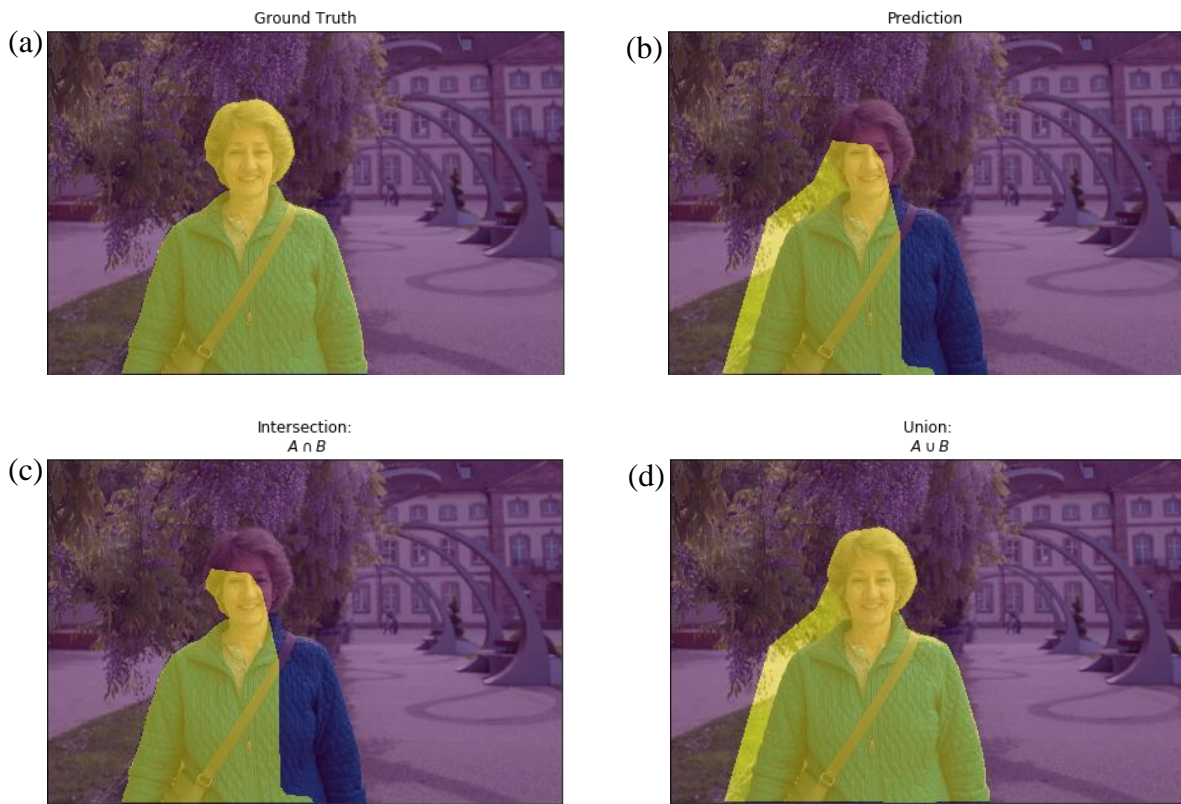
Pixel accuracy is the percentage of pixels in the image that are correctly classified. In calculation of pixel accuracy, it is common to report the accuracy for each class as well as globally. The equation used in this method is essentially comparing binary representation of a single instance by comparing the pixels that are true positive (TP) and true negative (TN) in their segmentation over both true positive values and true negative values along with the false positive (FP) and false negative values (FN) according to their targets masked image. This may seem easy to apply but by itself and an unbalanced dataset, the value can be misleading when some classes have small representation within the image (Tiu 2019; Jordan 2018).



**Figure 2.16 – Computing the intersection over union metric as the intersection of the ground truth and the predicted image over the union of both the ground truth and the predicted image (Rosebrock 2016).**

The metric intersection over union and mean intersection over union are used as a pixel class comparison and as a model comparison. Intersection over union is done using an unseen but labeled dataset to a segmentation model. The true class is compared to the predicted class by taking the overlapped area of the true class and the predicted class over the combination of both the true and the predicted areas (**Figure 2.16**). An example is shown in **Figure 2.17** where the intersection

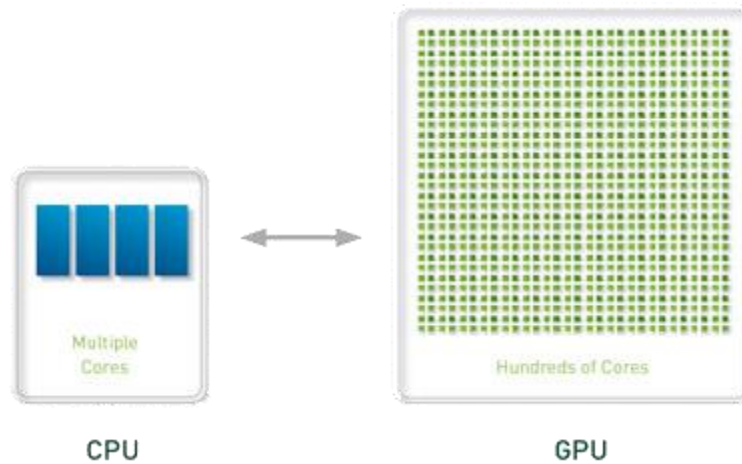
is defined by the pixels found in both the ground truth and the predicted image and the union is defined by the pixels in both ground truth and predicted images.



**Figure 2.17 – In this example of how intersection and union are determined, (a) the expected output is the segmentation of the lady vs. the background. (b) the model prediction. (c) The intersection is the overlapping area between the true segmentation and the predicted segmentation. (d) The union is determined between both the combination of both the ground truth labels and the predicted labels (Jordan 2018). The results from these images are used to determine the intersection of union metric.**



## 2.5 GPU Computing



**Figure 2.18 – GPU vs CPU (Boston)**

GPU computing makes use of a graphic processing unit (GPU) that is used to speed up computations by allowing parallel processing. These devices are utilized by a computer by offloading the more computationally expensive calculations from the central processing unit (CPU) to the GPU. GPUs were initially created for computer graphic used in video games but have since been proven critical in providing acceleration in calculations as compared to CPU processing. Shown in **Figure 2.18**, a typical CPU can contain anywhere from four to eight cores for processing while a GPU has hundreds of smaller cores. While GPUs can process much faster than CPUs due to GPU-parallelism, they are not as versatile when it comes to managing inputs and outputs of the computer (OmniSci 2020). GPUs are best suited for highly repetitive and parallelized computing task such as machine learning, financial simulations, etc.

GPU computing has allowed for an increase in processing speeds by allowing GPU-parallelization. NVIDIA offers a parallel-programing model, NVIDIA® CUDA®, that offers support in correctly applying GPU-parallelization to certain NVIDIA CUDA enable GPUs.

## Chapter 3: Methodology

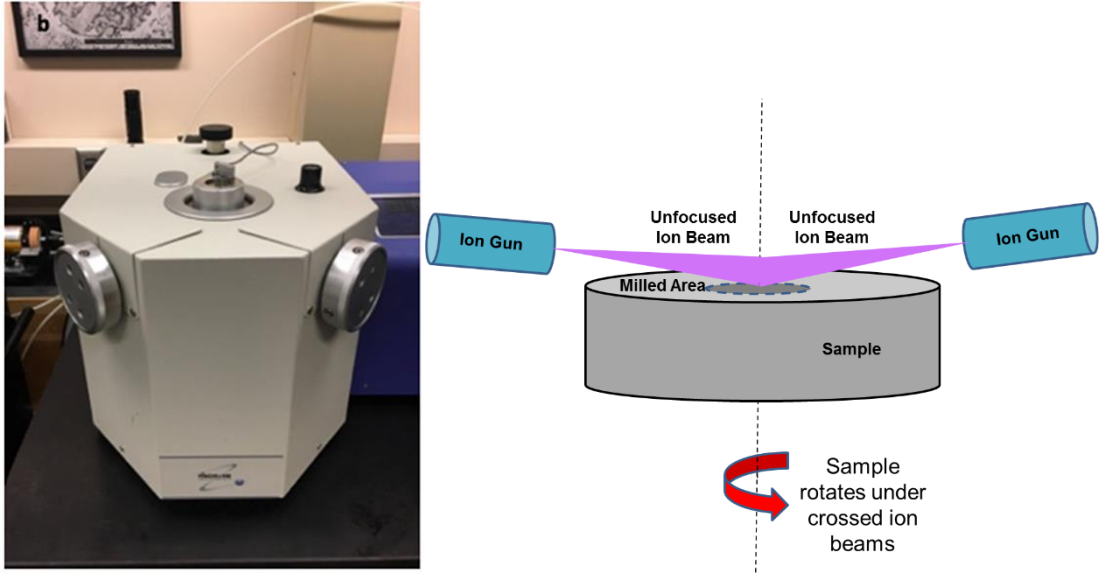
Chapter 3 describes the details of the equipment and procedures used in this study. There are two main topics: Shale Image Classification and Shale Image Segmentation

### **3.1 Equipment and Materials**

#### **3.1.1 SEM Sample Preparation**

SEM samples were selected from core at desired depths from multiple formations. The samples were prepared perpendicular to the bedding plane. These samples were polished by hand starting at 120 grit emery paper to initially smooth the surface before polishing with 400, 600, and 800 grit paper. This sequence was done in successive order to ensure no notable scratches or artifacts are seen on the surface during imaging. A Fischione™ Model 1060 argon ion mill (**Figure 3.1a**) is used to further ensure the sample surface is smooth. A two-step procedure is used in the milling process where the surface is milled for three hours at 5kV focusing at the center of the sample with a high incidence angle, followed by an additional nine hours at 6 kV covering the surrounding region.

The milled sample is then covered lightly in Au/Pd using a Desk V DENTON vacuum to sputter coat the surface in three-second increments for approximately ten seconds total. This is also done to reduce charging effects on the surface. If the sample is known to have additional charging issues during images, a copper tape is applied before sputter coating the sample surface.



**Figure 3.1 – (a) Model 1060™ Fischione mill used in sample preparation. (b) depiction of how sample is milled using ion beam. Two ion guns blast the surface of the sample while the sample rotates to create an even surface (Curtis 2017)**

### **3.1.2 SEM and Image Acquisition**

A FEI Helios Nanolab™ 650 DualBeam™ machine was used in image acquisition. A concentric backscatter detector (CBS) at 2 kV accelerating voltage is used during imaging and is sensitive to atomic number of atoms. As a result, heavier elements, i.e. pyrite, rutile etc., have a higher atomic number and will appear lighter in contrast to pores and organics, which have zero or low atomic number. During the segmentation portion of this thesis, an Oxford XMAX 20 mm<sup>2</sup> detector was used in acquiring elemental maps for mineral labeling.



**Figure 3.2 – FEI Helios Nanolab™ 650 DualBeam™ Scanning Electron Microscope**

FEI MAPS software was used in the generation of large area, high resolution images used in both segmentation and formation classification. Each MAP was taken at 10 nm resolution and 500  $\mu\text{m}$  wide and contained approximately 200-1,000 individual images, or tiles, stitched together to generate the large-area images. These MAPS varied in length but varied up to 1mm. Unfiltered tiles were used in both the segmentation and formation classification.

### ***3.1.3 Image Classification Computer***

A windows server machine was initially used for image classification. This computer contains two NVIDIA Quadro K6000 GPUs that each provide computing power of 5,196 GFLOPs single precision and 1,732 GFLOPs double precision. Gigaflops (GFLOPs) are measure of floating-point operations a processor can perform per second. Initial work with image classification used MATLAB version R2017b – R2020a.

**Table 3.1 – Computer Specifications for Image Classification**

	Parameters
OS	Windows Server 2016
Memory	96 GB
CPU	Intel® Xeon® Processor E5-2690 @ 2.90GHz × 16 (x2)
GPU	NVIDIA Quadro K6000 (x2)
CUDA Cores	5760
Virtual Memory (GB)	24
Single Precision Compute Power (GFLOPs)	10,390

**3.1.4 Image Segmentation Computer**

A Unix box using Ubuntu 18.04.03 was used in segmentation work. The switch from Windows to Linux was done due to the slower communication time between MATLAB to the GPUs in Windows.. Computer specs are shown in **Table 3.2**. Initial work utilized five NVIDIA GeForce® 1080 Ti GPUs where each GPU provided 10,609 GFLOPS in single precision and 332 GFLOPS double precision processing power. Later work in segmentation utilized two additional NVIDIA Titan RTX GPUs where each GPU provided 12,442 GFLOPS in single precision and 389 GFLOPS in double precision processing power. A comprehensive table of GPU comparisons is shown in **Table 3.3**.

**Table 3.2 – Computer Specifications for Image Segmentation**

	Parameters
OS	Ubuntu 18.04 64-bit
Memory	64 GB
CPU	Intel® Core™ i7-7700K CPU @ 4.20GHz × 8
GPU	NVIDIA GeForce 1080 Ti (x5) NVIDIA Titan RTX (x2)
CUDA Cores	27,136
Tensor Cores	1,152
Virtual Memory (GB)	103
Single Precision Compute Power (GFLOPs)	89,320
Tensor TFLOPs	260

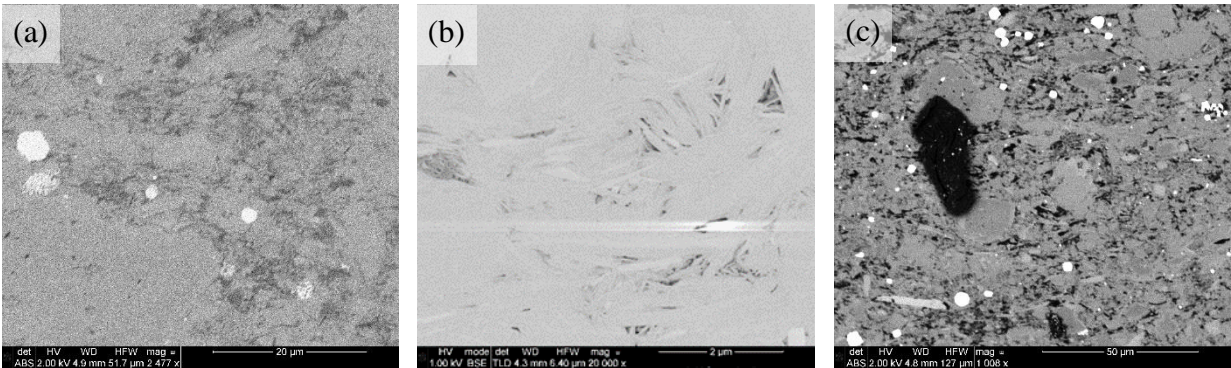
**Table 3.3 – GPU comparison**

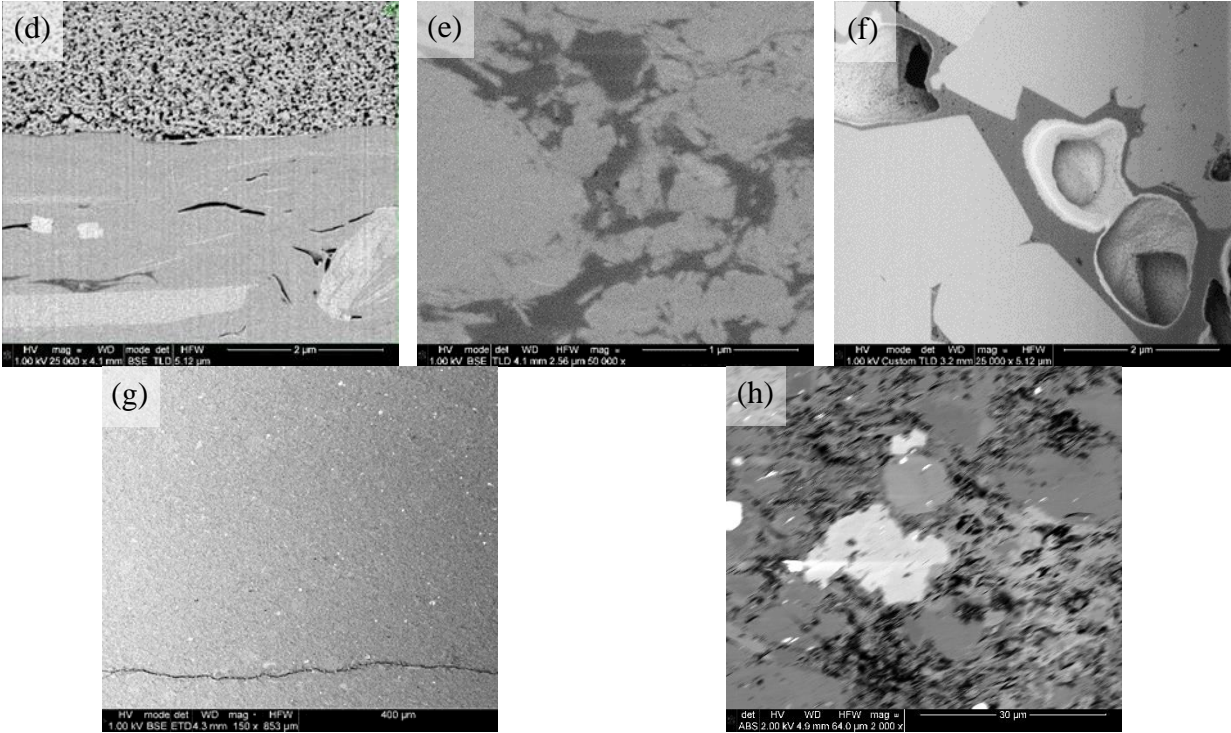
NVIDIA Model	Memory Size (GB)	Memory Clock Speeds (MT/s)	CUDA Cores	Tensor Cores	Single Precision (GFLOPs)	Double Precision (GFLOPs)	Half Precision (GFLOPs)
GeForce 1080 Ti	11	11,000	3,584	–	10,609	332	166
Titan RTX	24	14,000	4,608	576	12,442	389	24,884
Quadro K6000	12	–	2,880	–	5,196	1,732	–

## 3.2 Evaluation of image quality

### 3.2.1 Dataset preparation

The purpose of this portion of the project was to determine the image quality, i.e. was it sufficient for further processing for play identification and image segmentation. Image sub-tiles were taken from SEM images to generate the dataset. **Table 3.4** shows the number of images used in each classification. Examples of poor SEM images are shown in **Figure 3.3** and some of examples include poor brightness and contrast, charging of the sample, contamination, curtaining, poor focus, redeposition, shading and astigmatism.





**Figure 3.3 – Examples of poor imaging. (a) poor brightness/contrast, (b) charging, (c) contamination of the sample, (d) curtaining effect, (e) out of focus, (f) redeposition, (g) shading, and (h) astigmatism.**

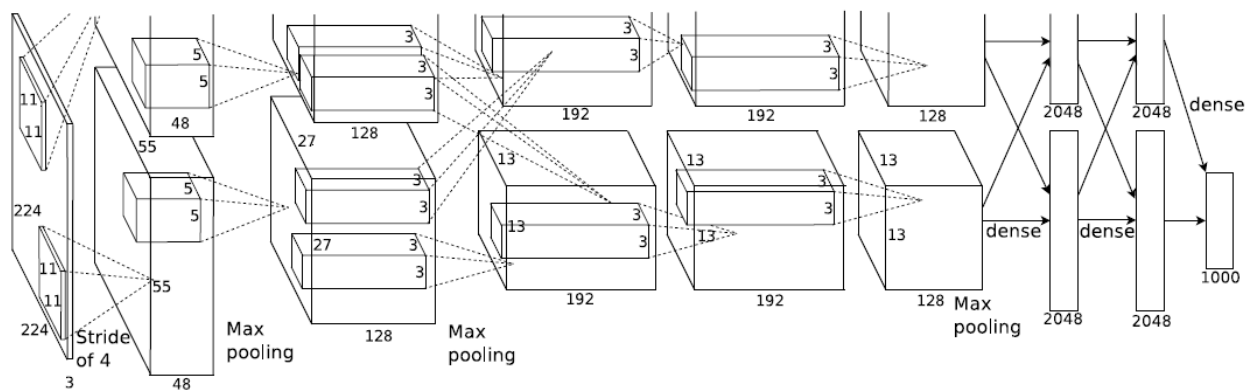
SEM images at varying resolutions were gathered and used to generate the training dataset. For the purposed of image classification, a pre-trained CNN called AlexNet was used. Transfer learning was used due to the limited number of images in our dataset. Image input size for AlexNet is 227x227 pixels (Krizhevsky et al. 2017). The images were tiled to 455x455 pixels before being resized to 227x227 for use in the pre-trained network AlexNet through the Deep Learning package on MATLAB.

**Table 3.4 – Number of images per classification for evaluating image quality**

Classification	No. of Images (455x455 pixels)
Good	833
Bad	1,935

### 3.2.2 Model Training

In model training, supervised learning, more specifically transfer learning, was used on the pretrained CNN AlexNet through the Deep Learning package in MATLAB version R2017b (Krizhevsky et al. 2017). This network was initially trained and applied to the ImageNet database. This database contains more than 15 million images which consist of everyday objects such as pens, pencils, coffee mugs, cars, planes, etc. Transfer learning is used to train the last layers of a CNN to identify a different set of images. This process uses the parameters determined in the initial training of the ImageNet database, and fine tuning those parameters and retraining the final layers to better suit the needs of the new dataset.



**Figure 3.4 – An illustration of the architecture of AlexNet, explicitly showing the delineation of responsibilities between two GPUs. One GPU runs the layer-parts at the top of the figure while the other runs the layer-parts at the bottom. The GPUs communicate only at certain layers (Krizhevsky et al. 2017). Each box is a computational layer that has a height, width and number of feature maps in each layer. CNNs used in classification also include dense layers or classification layers to provide prediction results.**

AlexNet is a CNN created by Alex Krizhevsky and competed in the ImageNet Large Scale Visual Recognition Challenge in 2012. This network achieved a top-5 error of 15.3% meaning the percentage of test images for which the correct label is not among the top 5 considered the most probable. The architecture consists of eight layers, five convolutional layers followed by three fully-connected layers. At the time of initial training by Krizhevsky, GPU computing was still



taking off. He used two GTX 580 GPUs that have 3GB of memory each. Training on a single with 3GB of memory limits the size of the network that can be trained.

In initial training of AlexNet, the number of training parameters was over 1.2 million which is too big to fit on a single GPU. **Figure 3.4**, shows how cross-GPU parallelization was utilized in model training. This method allowed for half of the kernels on each GPU where each GPU can only communicate at certain layers. The first convolutional layer filters the 227x227x3 image with kernel sizes of 11x11x3 sliding across the image with a stride of 4 pixels. This data is then fed to the second layer where it filters the output of the first convolutional layer with a kernel size of 5x5x48, followed by the third, fourth, and fifth layers connected to the outputs of the previous layers with a kernel size of 3x3x192. These convolutional layers are then connected to the fully connected layers to produce the predicted output. **Table A.5.1** in the appendix shows the training parameters used in transfer learning of AlexNet.

During model training, the data was separated into a training, validation and testing set, with 80% of the images in training, 10% in validation and 10% in testing. **Table 3.5** shows the number of images used in each dataset.

**Table 3.5 – Separation of data during training**

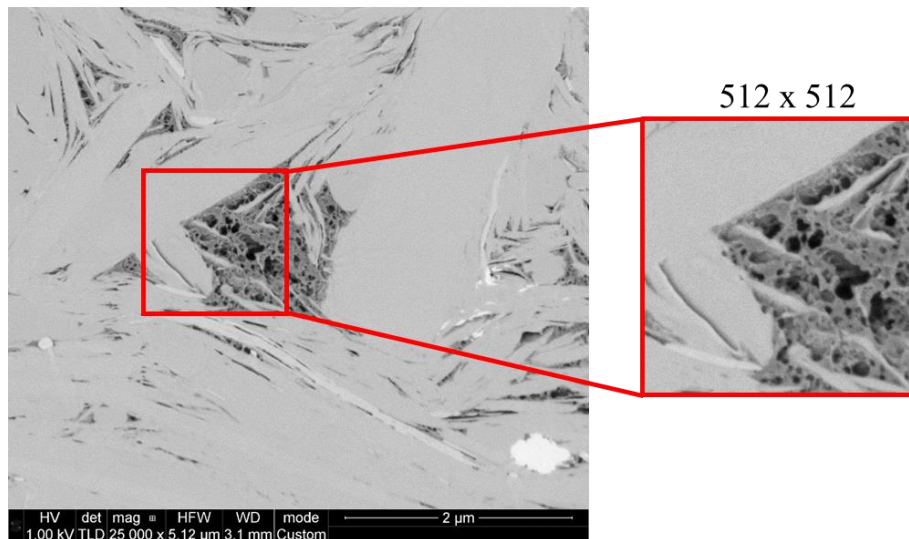
	Percentage, %	No. of Images (455x455 px)
Training	80	2,254
Validation	10	281
Testing	10	283

### 3.3 Image Classification

#### 3.3.1 Dataset preparation

##### 3.3.1.1 Method 1

Initial dataset creation was done in a similar manner to evaluate image quality. Images from the Unconventional Shale Consortium database were used. SEM images taken at varying resolutions were used in generating this dataset. Sub-tiles of these images were generated at a 512x512 pixel size. An example image and a sub-tile taken from the Marcellus formation is shown in **Figure 3.5**. A total of 13,756 sub-tiles, 1,500–2,500 per formation, were generated using this method and we applied it to seven formations, Eagle Ford, Haynesville, Lower Bakken, Marcellus, Niobrara, Wolfcamp, and Woodford.

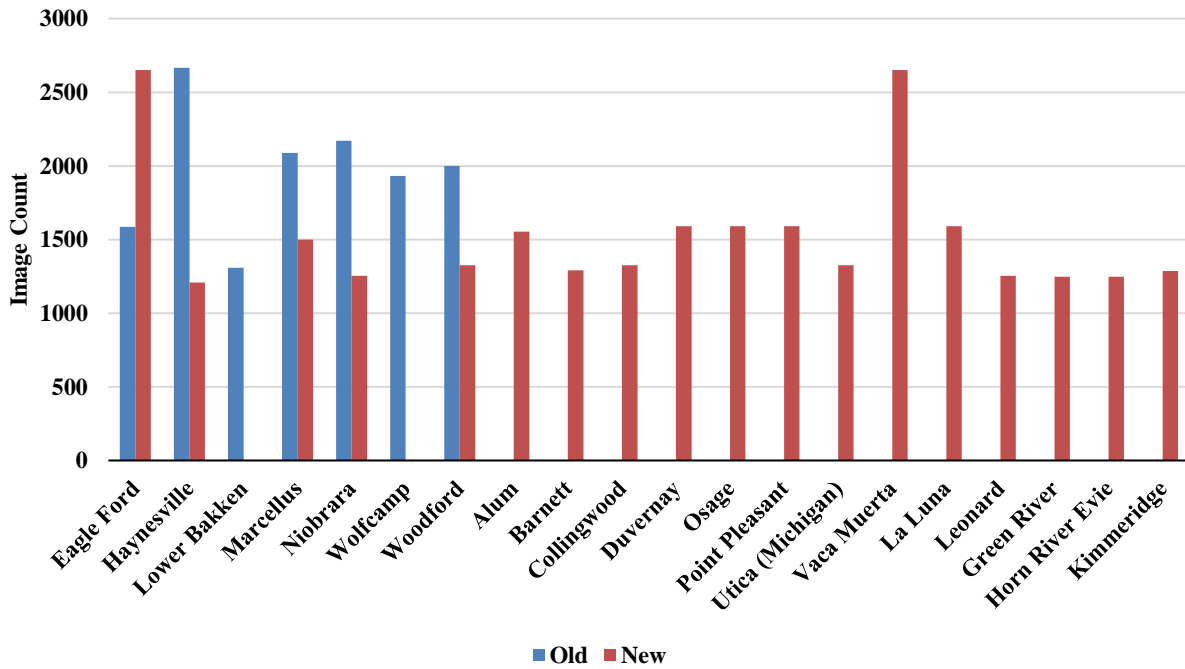


**Figure 3.5 – Example of sub-tile generation from SEM images taken from the Marcellus formation.**

##### 3.3.1.2 Method 2

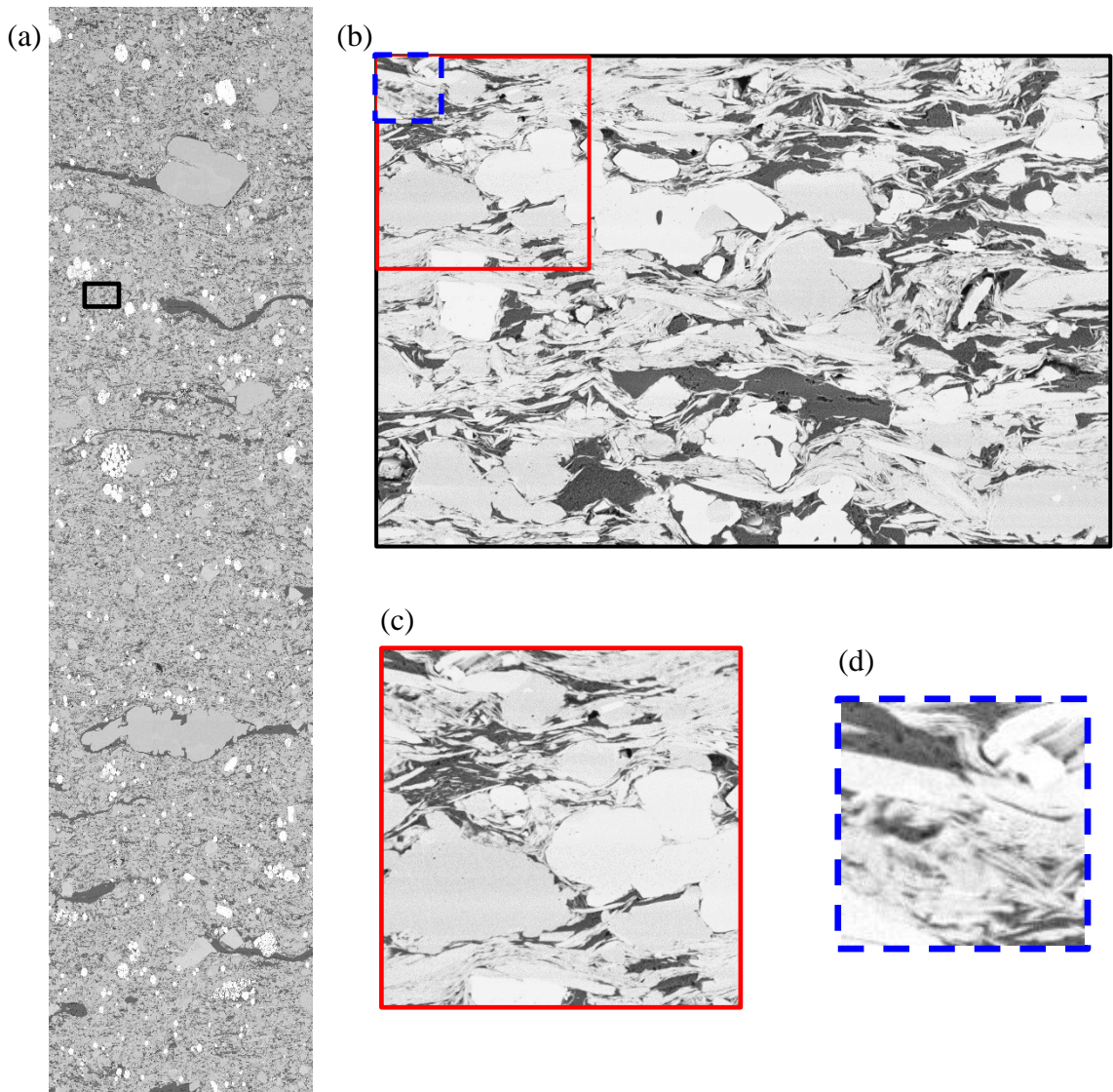
Based on the results from Method 1 (shown in Chapter 4, page 63), a different method of image gathering was used. Over 28,000 SEM sub-tiles from 18 different formations were taken

from a MAPs dataset. For two of the 18 formations, Eagle Ford and Vaca Muerta, thermal maturity values were obtained which allowed for the separation into oil and gas windows. In **Figure 3.6**, shows the distribution of sub-tiles per class where each class has a range of 1,200–1,600 images for the new dataset.



**Figure 3.6 – Distribution of image counts per formation. For Eagle Ford and Vaca Muerta, thermal maturity values were obtained and allowed for separation into oil and gas windows.**

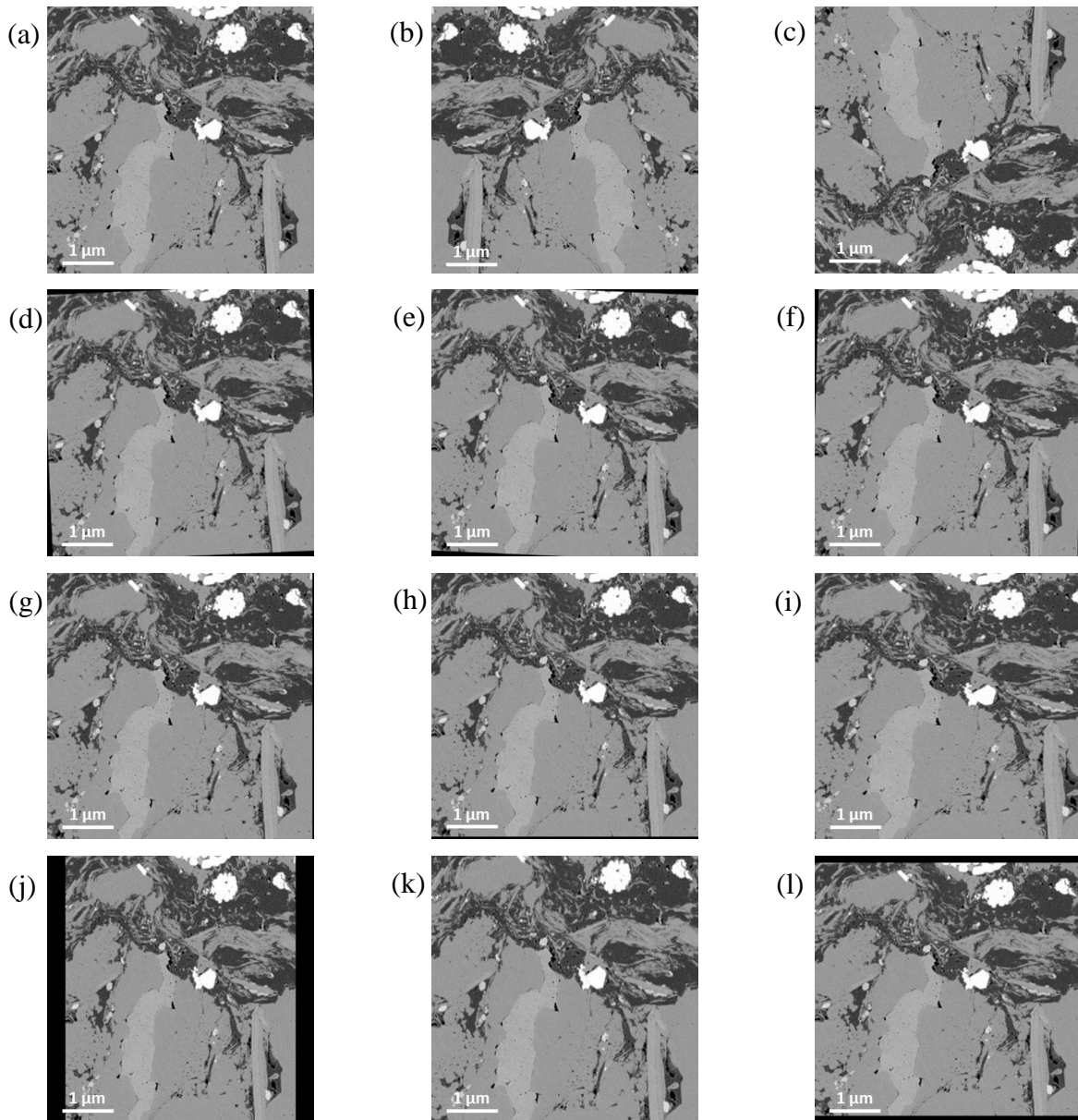
As opposed to Method 1, these images were taken directly from MAPS image tiles imaged at 10nm/pixel resolution and at a size of 3072x2048 pixels. These MAPS tiles were tiled further to create sub-tiles at 512x512 pixels. An example of this image separation is shown in **Figure 3.7**. What is considered A tile is considered a single 3072x2048 pixel image that is used in the MAPS images, and a sub-tile is what has been reduced further to desired size 512x512 pixels. This value of 512x512 was decided upon to try and capture the local microstructure and the surrounding features.



**Figure 3.7 – MAPS data consists of hundreds of tiles that are stitched together. These tiles are separated further into sub-tiles for training. A typical tile size is 2048x3027 pixels but the sub-tile in this report is 512x512 or 128x128 pixels used for classification and segmentation respectively. (a) shows the typical size for a MAPS image with a black box indicating the size of a tile, (b) shows a tile taken from the MAPS image with the size of 512x512 and 128x128 pixels, (c) shows a 512x512 pixel image used for classification and (d) shows a 128x128 pixel image used for segmentation.**

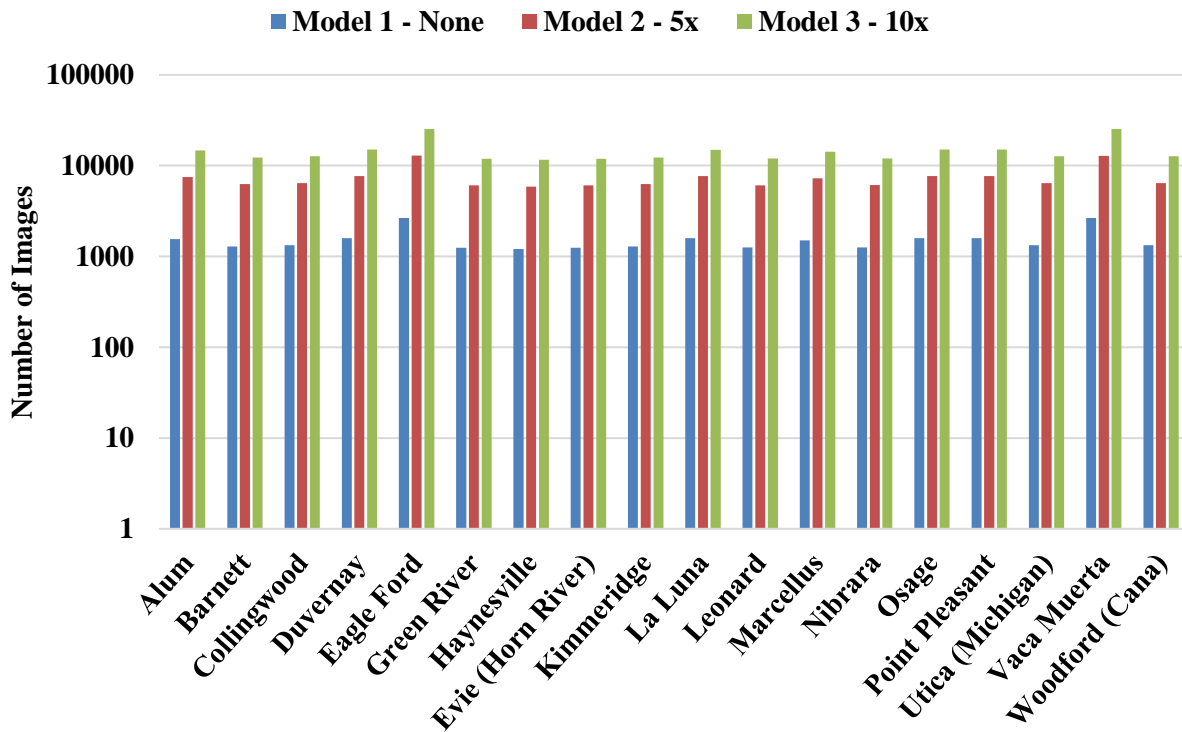
The sub-tiles were down-sampled using nearest neighbor's interpolation to fit AlexNet imaging requirements. This reduced the resolution from 10nm to 56 nm. Although a significant amount of the information about micro- and nano- pores will be lost when the images are resized,

we are interested in the overall texture of the 512x512 images. Future work will be to determine if reducing image resolution will affect training.



**Figure 3.8 – Example of image transformations. (a) Original SEM sub-tile, (b)-(c) reflections across X- and Y-axis, (d) rotation between -3 and 3 degrees, (e)-(f) shearing in X and Y directions, (g)-(h) translate image along X and Y axis, respectively -5 to 5 pixels, (i)-(j) upscaled image to 120% and 80%, respectively along the X-axis, (k)-(l) upscaled image to 120% and 80%, respectively along the Y-axis**

Overfitting of the data is the result of the network focusing on features of specific images. In the case of shales, the position of the mineral grains, the formation of the organics and the pore structure are descriptors of the shale itself. In order to prevent overfitting, we applied random applications of transformations to the sub-tiles. These transformations include rotation and shearing in the horizontal and vertical direction, translation in the horizontal and vertical directions, scaling, and reflection in the vertical or horizontal planes. Examples of these transformations are shown in **Figure 3.8**.



**Figure 3.9 – Image datasets generated using image augmentations. Model 1 was the dataset consisting of no additional augmentations. Model 2 utilized the dataset consisting of 5x augmentations meaning each image had up to five augmentations. Model 3 utilized the dataset consisting of 10x augmentations meaning each image had up to ten augmentations.**

In addition to expanding the dataset, we wanted to test how augmentation would affect training. Two additional models were where created using 5x augmentations and 10x augmentations,

meaning for every image, up to five or ten images were created using a combination of augmentations. In **Figure 3.9**, shows the number of images per formation that are in each dataset.

### **3.3.2 Model Training**

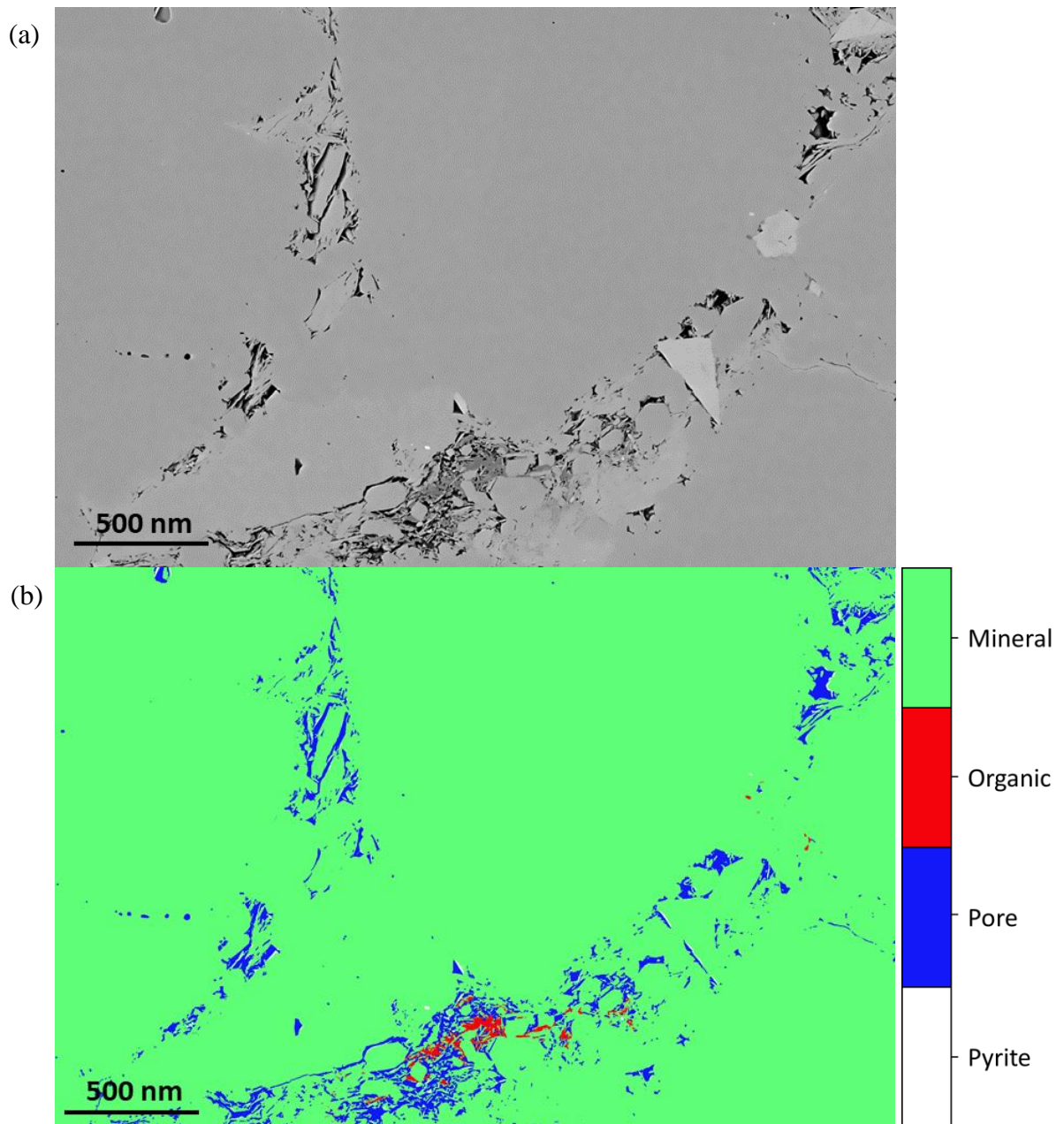
Using the AlexNet architecture, we utilized five GTX 1080 Ti GPUs simultaneously using the Deep Learning package MATLAB version R2018b to test training parameters. The model trained within 24 hours using 53,045 GFLOPS single precision processing power. An example of the network used is shown in **Figure A.5.1** for multi-GPU processing, and **Table A.5.2** shows the training parameters used in transfer learning of AlexNet located in the appendix.

## **3.4 Image Segmentation**

### **3.4.1 Data Preparation**

#### **3.4.1.1 Method 1**

Image segmentation is a computer vision task where a network assigns a class label to a set of pixels. The first step in being able to train a model is the generation of the training set. Initial segmentation work was done using MATLAB and the Image Segmenter App. A total of 16 images from the Wolfcamp formation, at a size of 3072 x 2048 pixels at 10 nm resolution, were labeled as mineral, pore, organic or pyrite. An example of a training image is shown in **Figure 3.10** with **3.9a** as an expected input SEM and **3.9b** as the expected output.



**Figure 3.10 – Example of a Wolfcamp training image generated using method 1. This image was labeled using MATLAB. (a) is the SEM image used as the expected input, (b) labeled images as the expected output.**

### **3.4.1.2 Method 2**

Method 2 focused on providing mineralogy from SEM imaging. The dataset contained 10 classes such as calcite, clay, dolomite, heavy minerals, k-feldspar, organics, plagioclase, pores,



pyrite, and silica, all determined through elemental maps. The dataset used in the initial image segmentation included a multi-layered SEM image with respective Energy Dispersive X-ray Spectroscopy (EDS) elemental data as expected input, and a hand-drawn labeled image as the expected output. The initial dataset consisted of 319 images at 128x128 pixels, taken at 10nm resolution. This included 208 images from a Woodford MAPS tile image and 121 images from an Eagle Ford MAPS tile image. An example of both a Woodford and Eagle Ford training labeled image is shown in **Figure 3.11** and **Figure 3.12** respectively.

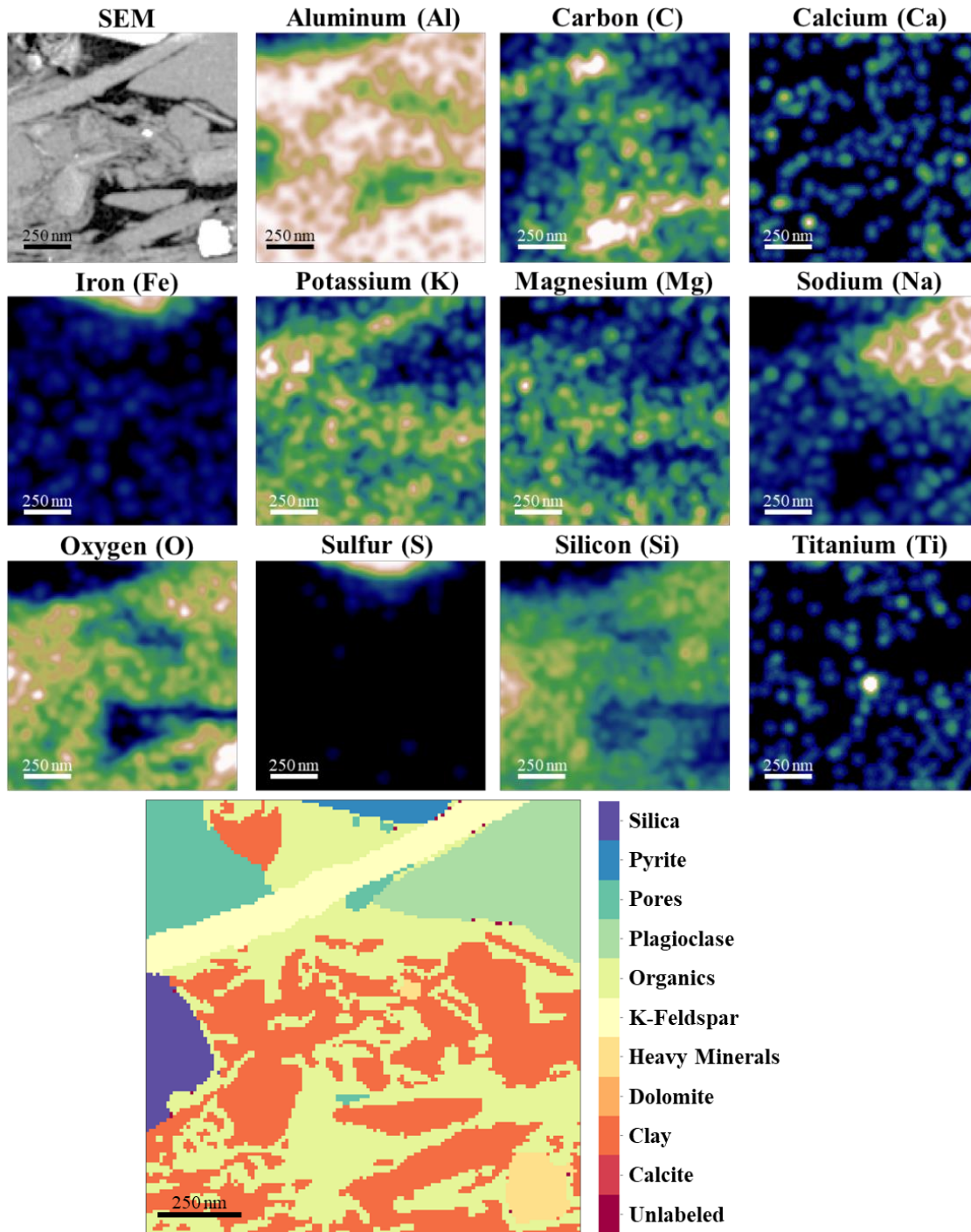
**Table 3.6 – Percent representation of mineral components of Eagle Ford and Woodford images in Method 2 dataset**

Component	Eagle Ford (%)	Woodford (%)
Calcite	55.5	0.2
Clay	25.5	30.6
Dolomite	0.2	13.3
Rutile	0.2	0.6
K-Feldspar	0	3.4
Organics	13.5	29
Plagioclase	2.1	3.6
Pores	0.5	7.4
Pyrite	0.7	5.6
Silica	1.9	6.3

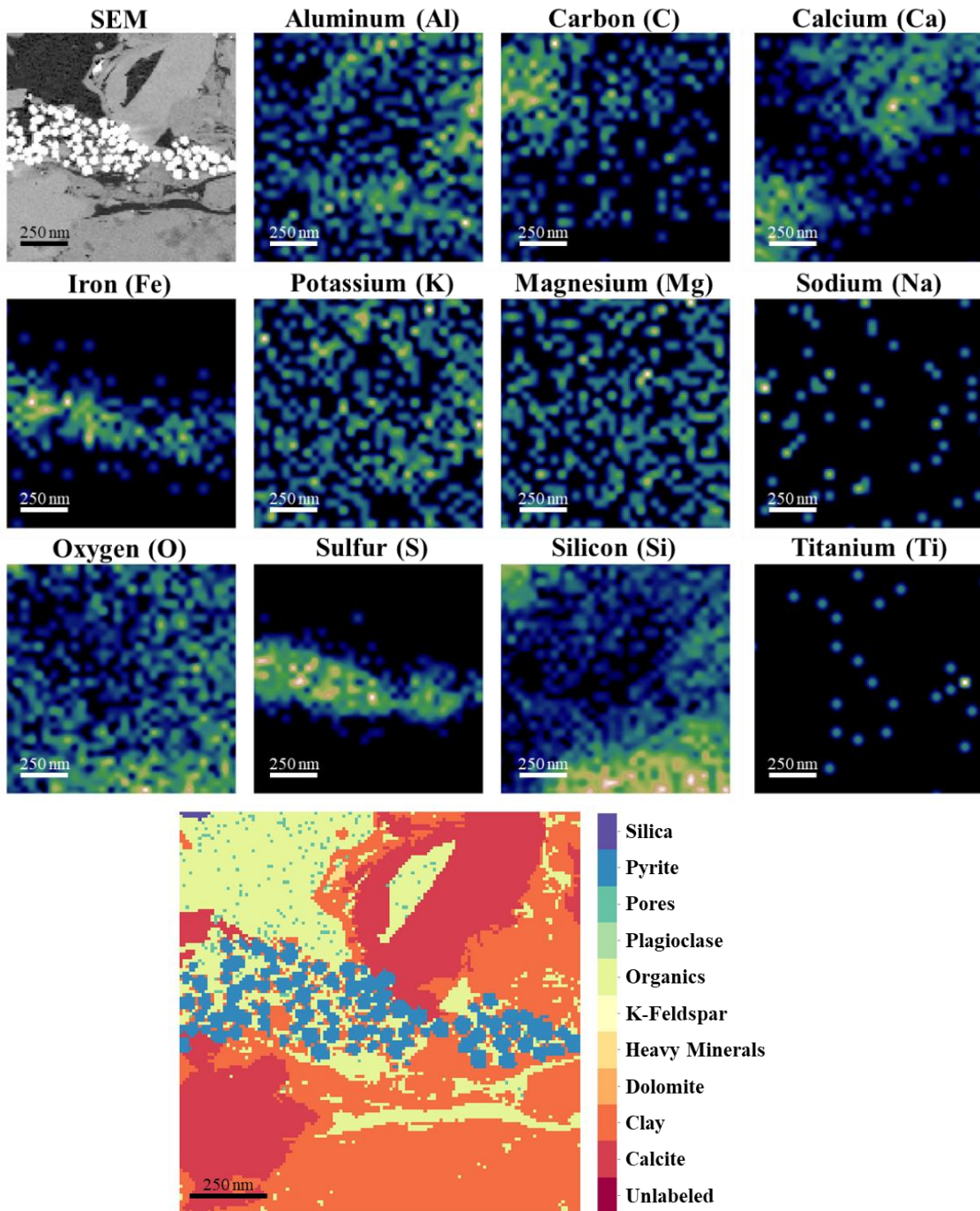
**Table 3.7 – FTIR Minerology comparison of images taken at same depth**

	Eagle Ford (wt.%)	Woodford (wt.%)
Calcite	61	0
Clay	12	55
Dolomite	5	3
Heavy Minerals	14	4
K-Feldspar	0	15
Plagioclase	5	3
Pyrite	0	2
Silica	5	17

Labeled images were hand- drawn using Gimp 2.9 by overlaying elemental maps to determine elemental makeup of the mineral grains. The distribution of pixels across the entire image set is shown in **Table 3.6** and the FTIR minerology is shown in **Table 3.7**.



**Figure 3.11 – SEM segmentation dataset example image from the Woodford formation with respective elemental data accompanied by the hand drawn label mask used in training.**

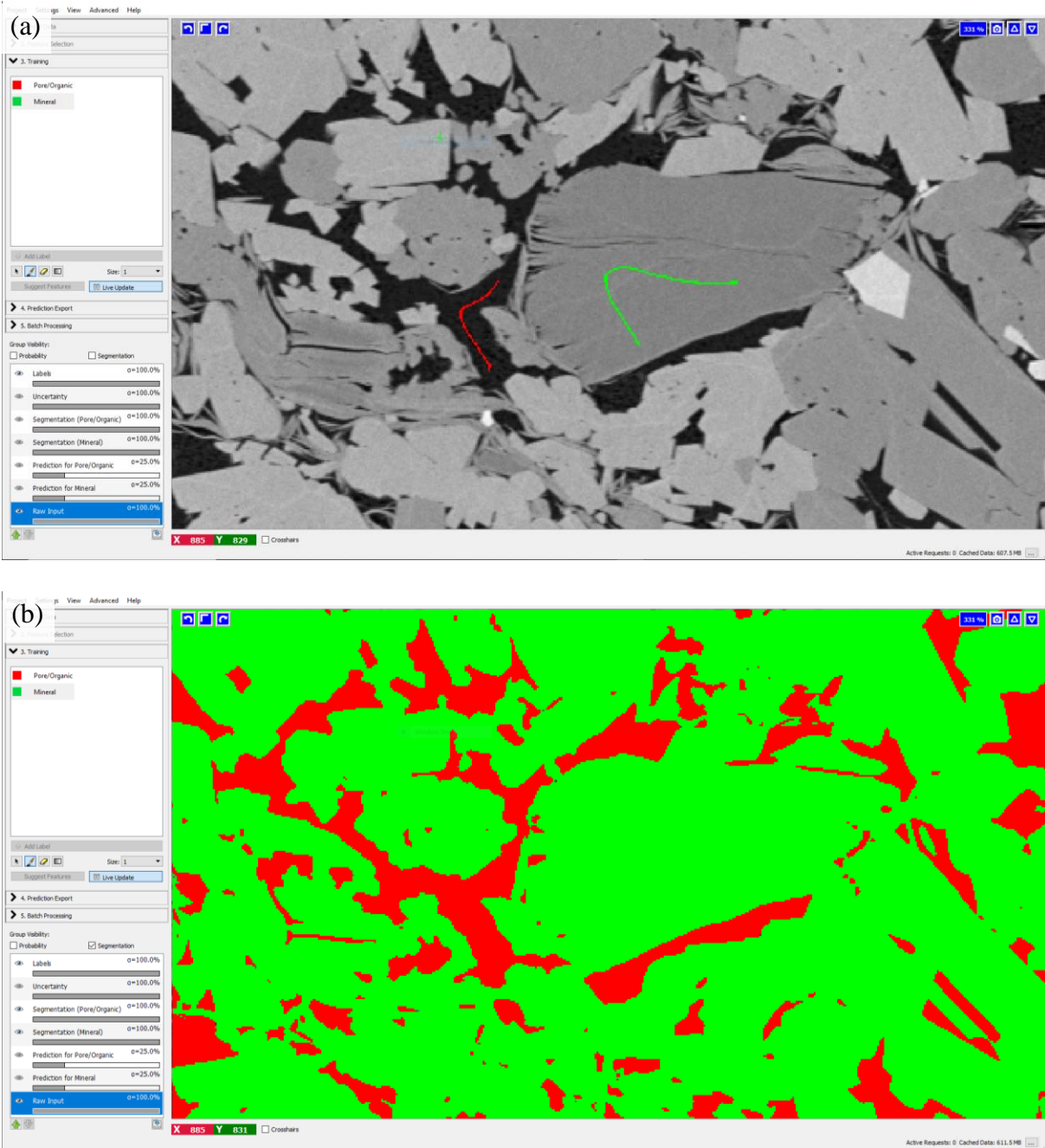


**Figure 3.12 – SEM segmentation dataset example image from the Eagle Ford formation with respective elemental data accompanied by the hand drawn label mask used in training.**

### 3.4.1.3 Method 3

The labeling process of method 2 was very time consuming. Based on the results determined in Method 2 (discussed in Chapter 4), the decision was made to reduce the number of classes from ten to three, i.e. pore, organic, and mineral. The idea was to generate a model that will accurately determine porosity and organic matter, which then can be masked to help determine only mineralogy (Method 4).

Since, labeling by hand was time intensive, image label generation was done using an open source software created by the NIH. The interactive learning and segmentation toolkit (ilastik) use a random forest classifier to labeled images through user interactions. While this method has been proven to be successful and much faster than hand labeling individual images, this method can be slow when processing multiple images; each image requires some user input. As a result, this method was a great method in generating a usable training set quickly and effectively as compared to the previous method. An example of the user interface and how labeled images were generated is shown in **Figure 3.13**.



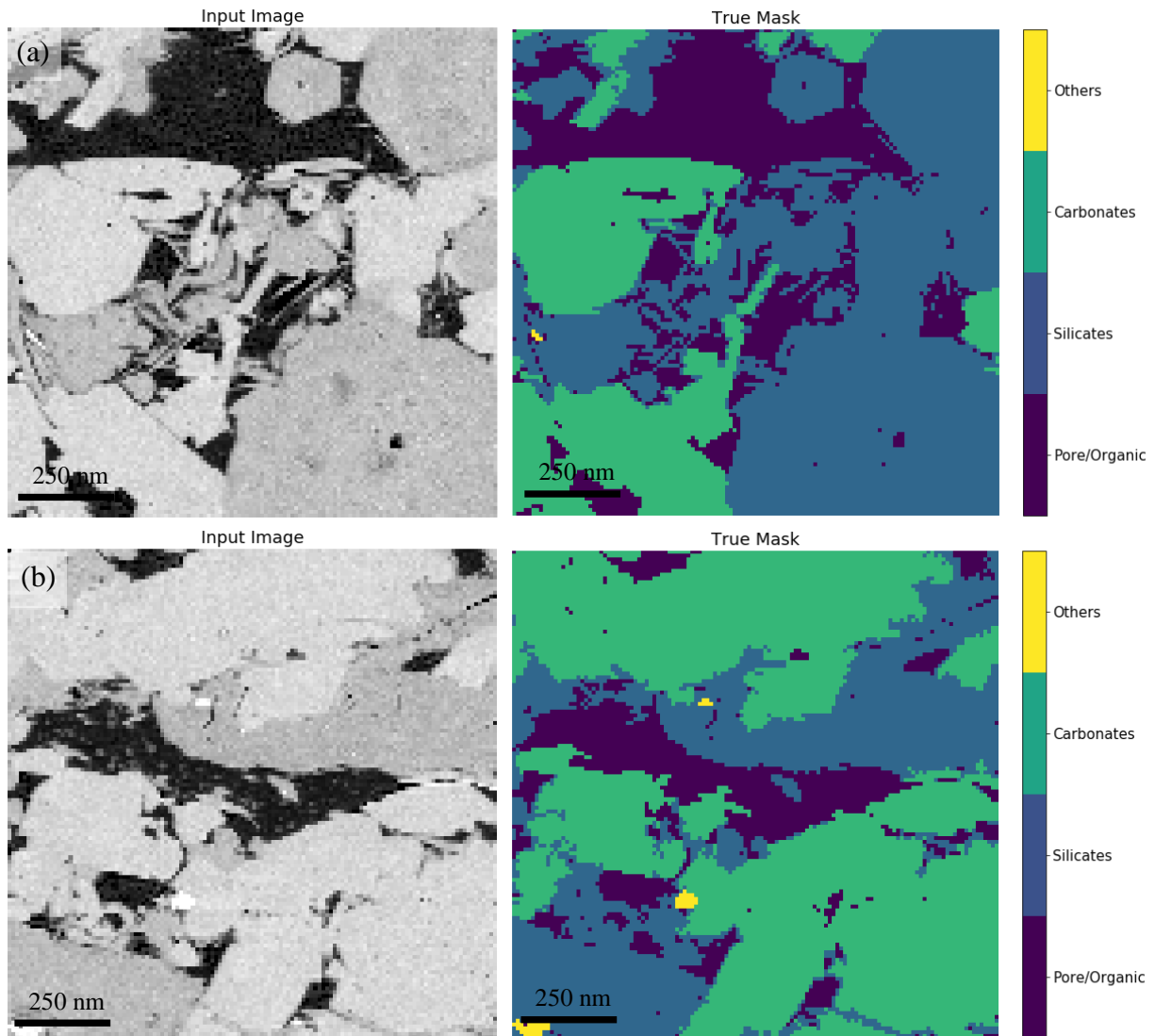
**Figure 3.13 – Example of how images are labeled in ilastik. (a) the user will draw lines on areas of interest that will be used as input for the random forest model. In this case, the user draws a red line as training data for pore/organiacs and a green line to represent minerals. (b) After the model has run, this is the predicted output.**

A Woodford MAPS image dataset was used in generating a training set. A total of three MAPS tiles were used in generating a labeled training set. After an ilastik labeled image was

generated from a MAPS tile using the random forest classifier, the 2048x3072 image was tiled across the image to generate ~1,400 sub-tiles at 128x128 pixels.

#### **3.4.1.4 Method 4**

Based on the results from Method 3, the original dataset from Method 2 was applied to the network that was proven to be successful in Method 3 (Model 5). Due to the class disparity within Method 2, some classes were combined to six classes: pore/ organics (combined), silicates, carbonates (calcite and dolomite), clays, feldspars, and other high-density minerals such as rutile, apatite and pyrite. An example of an Eagle Ford image is shown in **Figure 3.14**.



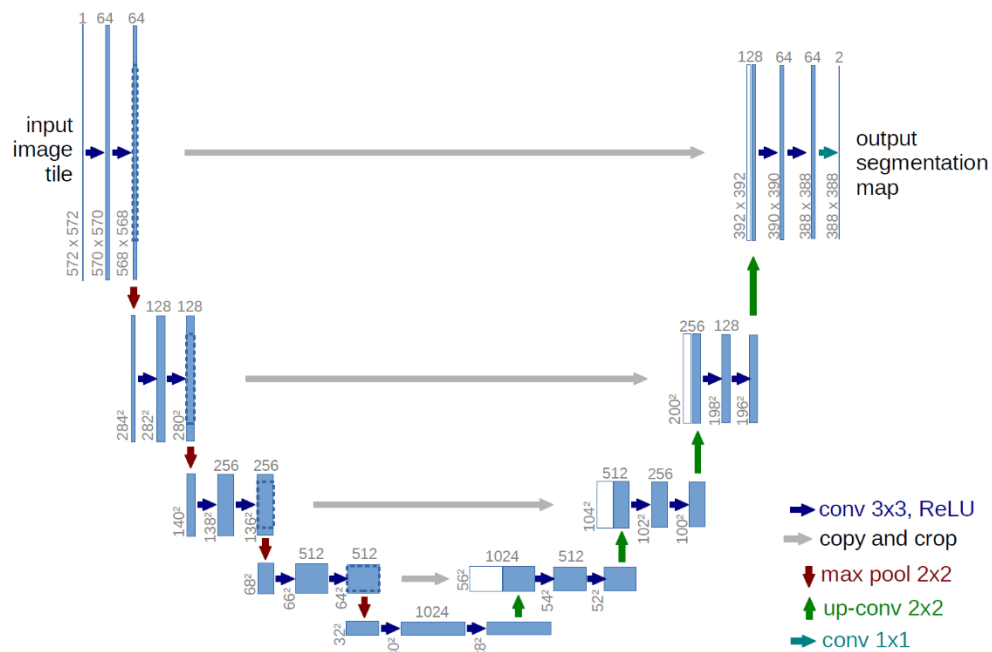
**Figure 3.14 – Two examples of Eagle Ford images used in segmentation separating out pore/organic, silicates, carbonates and other heavy minerals. On the left is the SEM images with its respective labeled images on the right.**

### 3.4.2 Model Training

#### 3.4.2.1 Method 1

This method was done in MATLAB using a pretrained network U-Net known for image segmentation. The U-Net architecture is a network that provides end-to-end training from very

few training images (Ronneberger et al. 2015). This network is modeled after a fully convolutional network with 23 convolutional layers and is trained to segment cell walls. This model provides an up-sampling of the data to allow propagation of important information to the higher resolution layers. In turn, this generates the U-shape of the network. Ronneberger et al. (2015) attempted to make maximal use of the NVIDIA Titan GPU (6 GB) memory by using larger image input size over a smaller image and larger batch size.



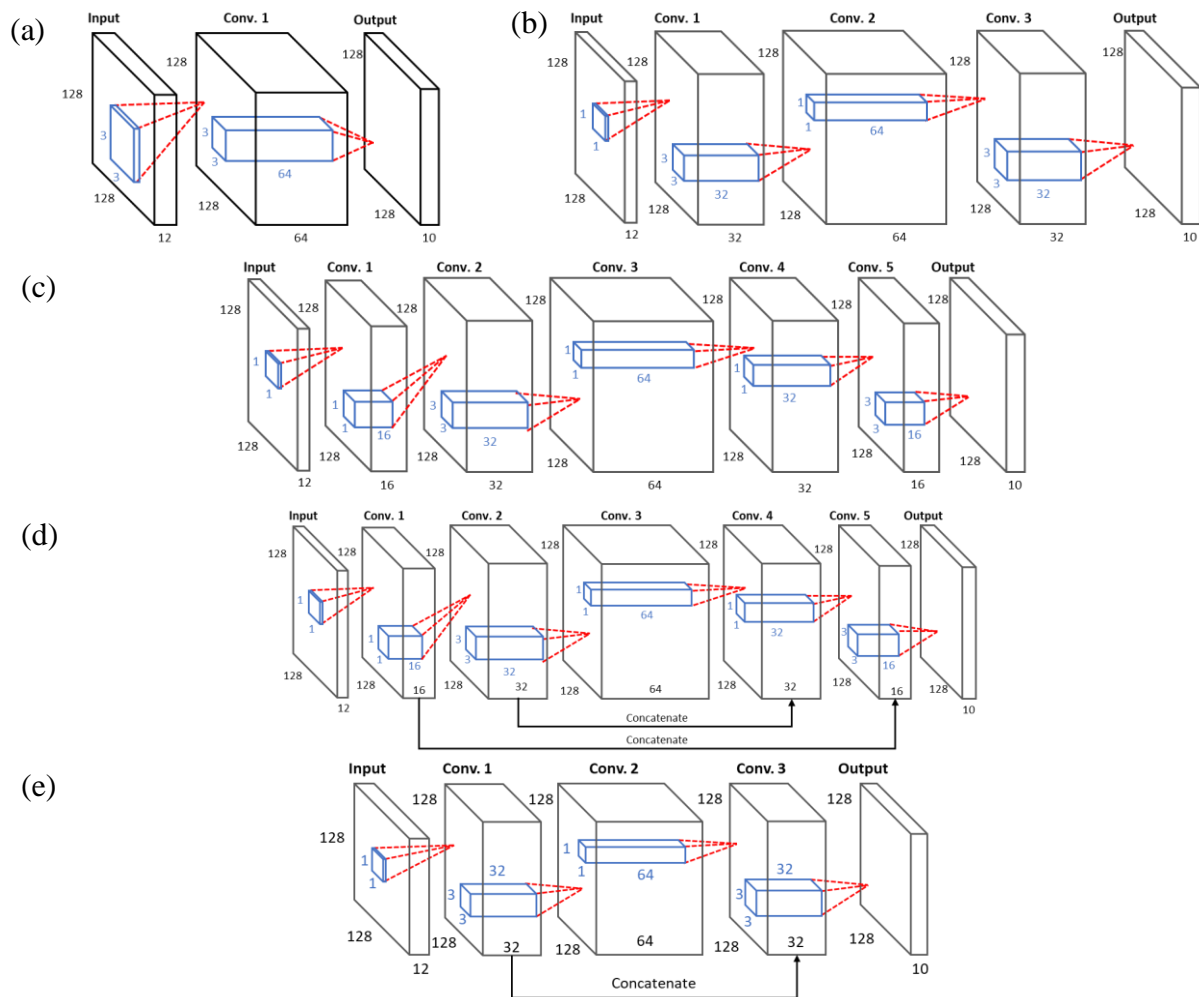
**Figure 3.15 – U-Net CNN architecture. The blue box represents a multi-channel feature map where the number of channels is located above the box in gray. The x-y size of the feature map is located to the left of the box. The arrows between the boxed represent different operations performed between feature maps. (Ronneberger et al. 2015)**

### 3.4.2.2 Method 2

Based on Method 1’s success with MATLAB (discussed in Chapter 4), the next step was to transfer over to Python 3.6 and Keras using TensorFlow 1.14 for mineralogical segmentation. Due to the success of U-Net in MATLAB, five custom models were tested to determine which



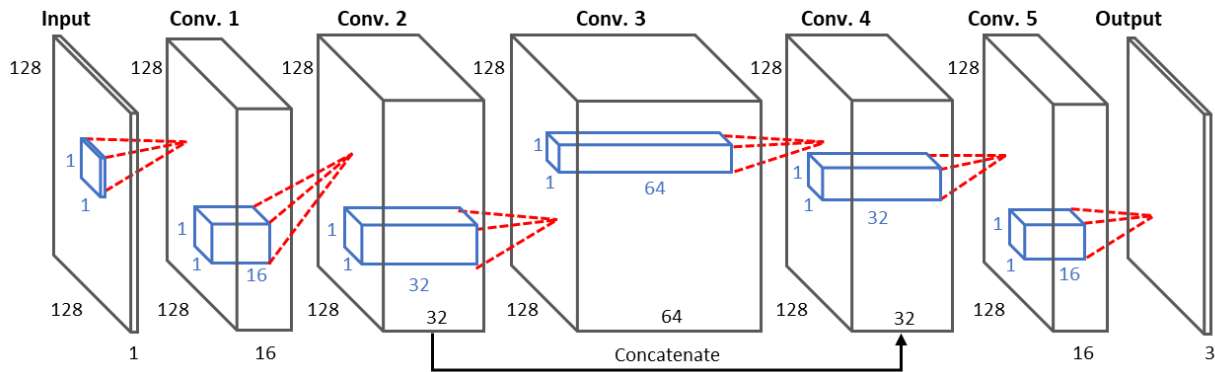
model would produce the best results. The models are shown in **Figure 3.16**, where models 4 (3.15d) and 5 (3.15e) are modeled after a U-Net and models 1–3 (3.15a–c) are fully convolutional networks. Each model was trained on a single GeForce 1080 Ti GPU and training took less than 10 minutes.



**Figure 3.16 – Model architectures tested during Method 2 segmentation. (a) Model 1, (b) Model 2, (c) Model 3, (d) Model 4, and (e) Model 5, Models 4 and 5 are modeled after the U-Net architecture. Models 1-3 (a-c) are considered fully convolutional networks as they provide connections between convolutional layers. Each convolutional layer (large black box) has a filter (blue small box) associated with each layer that traverses the image by filter to produce individual pixels for the following layer. Models 4-5 (d-e) are fully convolutional networks but provide local connections of data from the beginning of the model due to the concatenation or merging of layers.**

### 3.4.2.3 Method 3

With some success in Method 2, the decision was to reduce class sizes from the 10 mineral components down to three to focus on determining porosity and organic matter content derived from SEM image analysis. The results from Method 2 are discussed in Chapter 4. Based on this success, this was heading in the right direction. The model (**Figure 3.17**) used here was a modification after Model 5 in method 3. As compared to Model 5, this network added two additional convolutional layers.

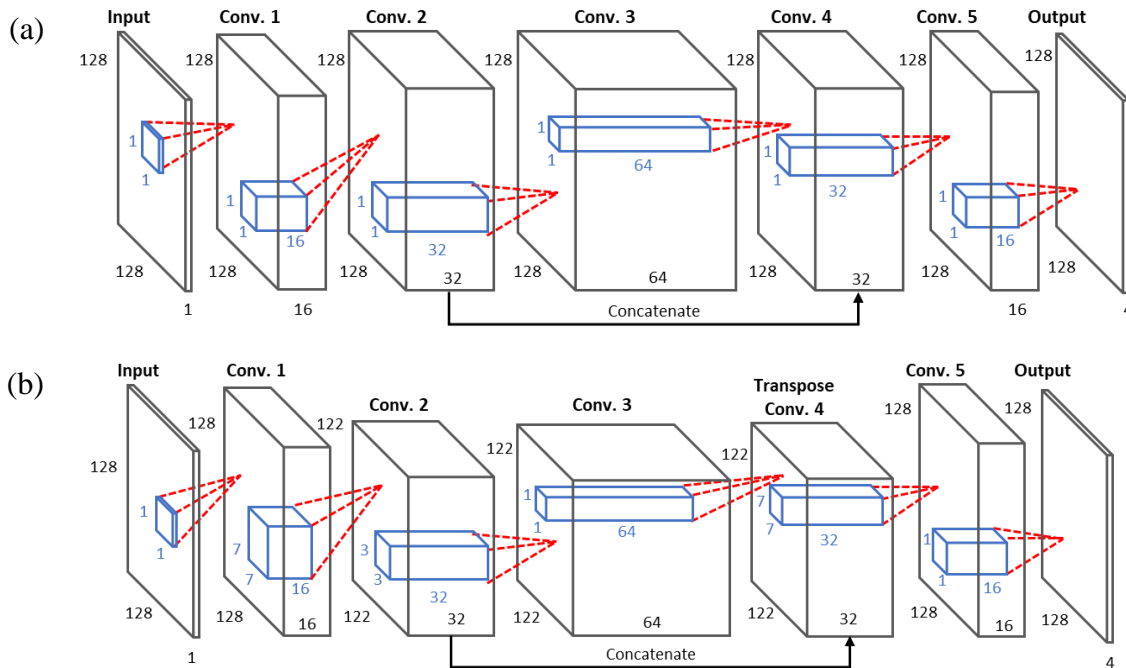


**Figure 3.17 – Model used in generating pore and organic classification. The layer is the input layer requiring an image of 128x128x1, i.e. a grayscale image. A filter size of 1x1x1 is read to the first convolutional layer. The first convolutional layer is of size 128x128x16 which corresponds to a multi-layered image generated from 16 separate filters. The next convolutional layer has 32 filters that generates 32 images at a size of 128x128. And the pattern continues throughout the rest of the layers. For both layers with 32 filters, they are merged at the end to provide a single output. The last layer consists of a 128x128x3 image corresponding to the probability masks of each pixel corresponding to each class.**

### 3.4.2.4 Method 4

With success in Method 3 (Chapter 4), a similar model was applied to the new data set with known success in three classes. Two models tested are shown in **Figure 3.18**. **Figure 3.17a** uses a kernel size of 7x7, whereas **3.17b** uses a kernel size of 1x1 for predicting the labels. The

purpose of this final method was to determine if kernel size has an effect on accurately predicting class labels during pixel labeling.



**Figure 3.18 – Models used in Method 4 to segment pores/organics, silicates, carbonates, and other heavy minerals. (a) displays model 1 the filter size (blue box) 1x1 pixels across all five convolutional layers and an output layer of size 128x128x4 pixels. (b) model 2 utilizes a 7x7 filter to generate layer Conv. 2, a 3x3 pixel filter to generate layer Conv. 3 and a 7x7 filter used in a Transpose convolutional layer to up sample the feature image to generate a feature map size of 128x128.**

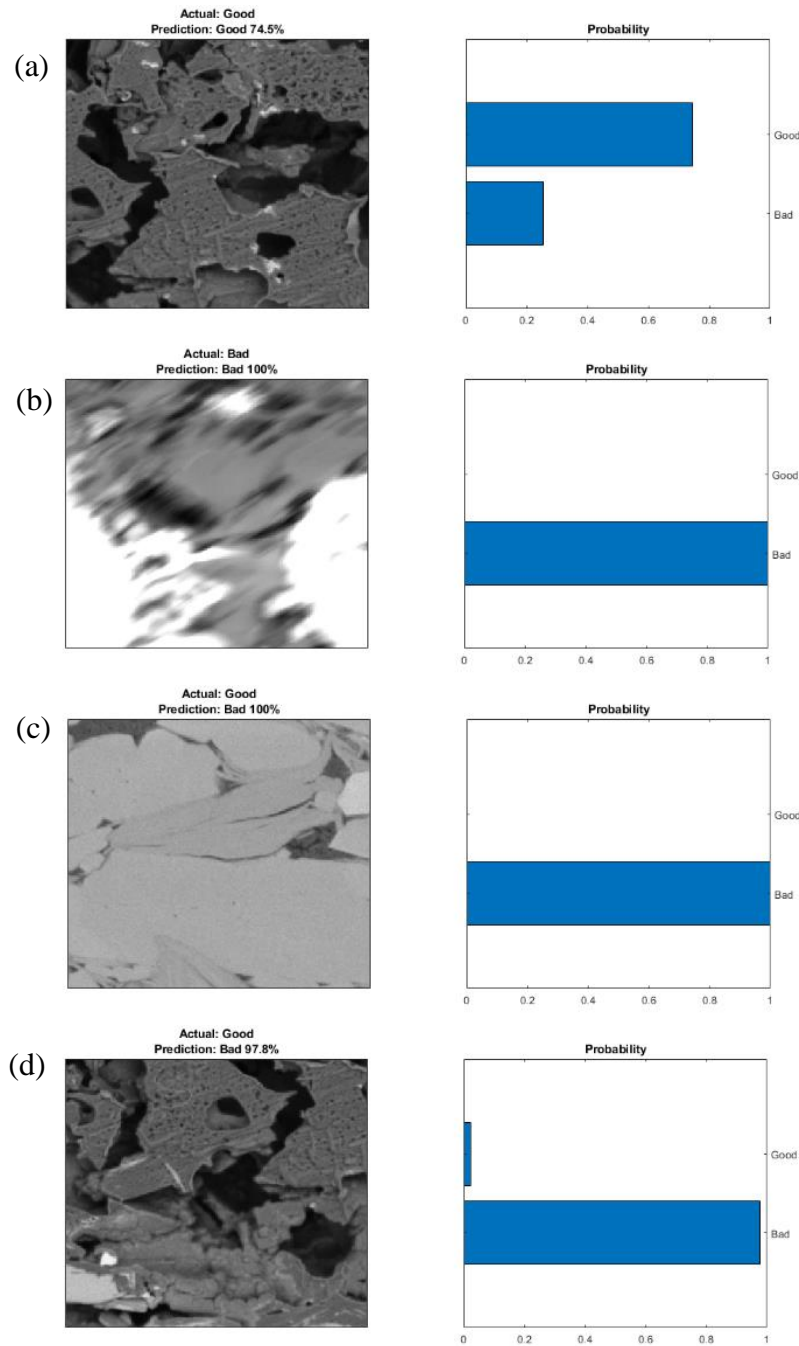
## Chapter 4: Results and Discussion

### 4.1 Evaluation of Image Quality

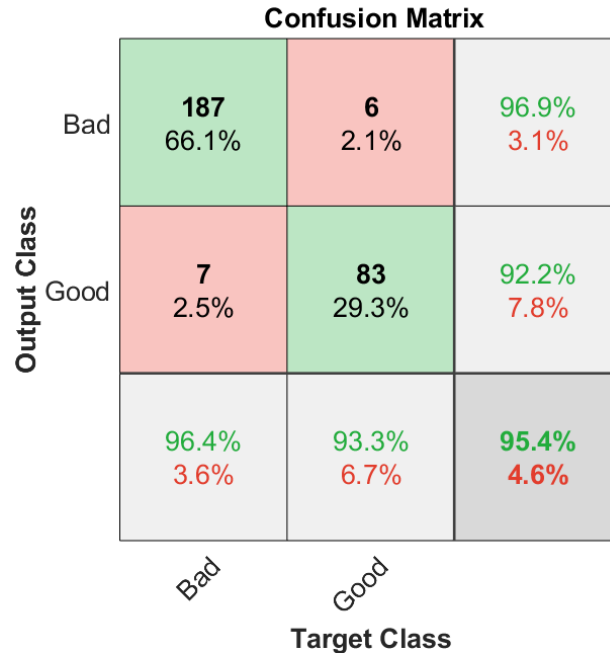
The evaluation of image quality becomes important to avoid user bias when assessing SEM images taken from multiple sources. Minor issues such as brightness and contrast can affect batch processing of the images. The purpose of this project was to provide a tool to accurately and quickly assess the quality of images. Using the pre-trained network ‘AlexNet’, accuracy in determining between the ‘good’ and ‘bad’ was 98%. Presented in **Figure 4.1** are

some of the predicted images from this model. A confusion matrix is shown in **Figure 4.2** and metrics calculated from the confusion matrix are shown in **Table 4.4**.

Based on the results determined by the network, the confusion matrix shows that seven images labeled as 'bad' were incorrectly labeled as 'good' and six images labeled as 'good' were incorrectly classified as 'bad'. Accuracy is 95.3% and is an indication on how often the classifier is correct. Precision is 94.8% which indicates that most images were classified correctly and how specific the model is predicting correct instances. Recall is 94.6% which indicates how sensitive the model is to detecting labels correctly.



**Figure 4.1 – Prediction results from the image quality dataset. (a) displays a ‘good’ image that was correctly classified as ‘good’ with the network classifying with as 74.5% probability. (b) displays a ‘bad’ image that was correctly classified as ‘bad’ with a 100% probability. (c) is an example of a ‘good’ image that was incorrectly classified. (d) shows an example of an incorrect classification. In this case the image was actually incorrectly labeled in the dataset. The network was able to detect the lower portion of the image is out-of-focus which is considered a win by the network.**



**Figure 4.2 – Confusion matrix for evaluation of image quality for determining accuracy, precision, recall and F1 score. The accuracy is defined on the confusion matrix as 95.4% which indicates the classifier is correct 95.4% of the time. The model has a recall of 94.8% and 92.2% for the classes ‘bad’ and ‘good’ respectively (shown in green in the right column), with an average recall of 94.8% indicating a high sensitivity to classifying correctly. The model has a precision of 96.4% and 93.3% for classes ‘bad’ and ‘good’ respectively (shown in green on the bottom row), and an average precision of 94.8% indicating that the model is precise in correctly classifying the data.**

**Table 4.1 – Metrics calculated from the confusion matrix**

	Metric, %
Accuracy	95.3
Precision	94.8
Recall	94.6
F1 Score	94.7

## 4.2 Image Classification

### 4.2.1 Method 1

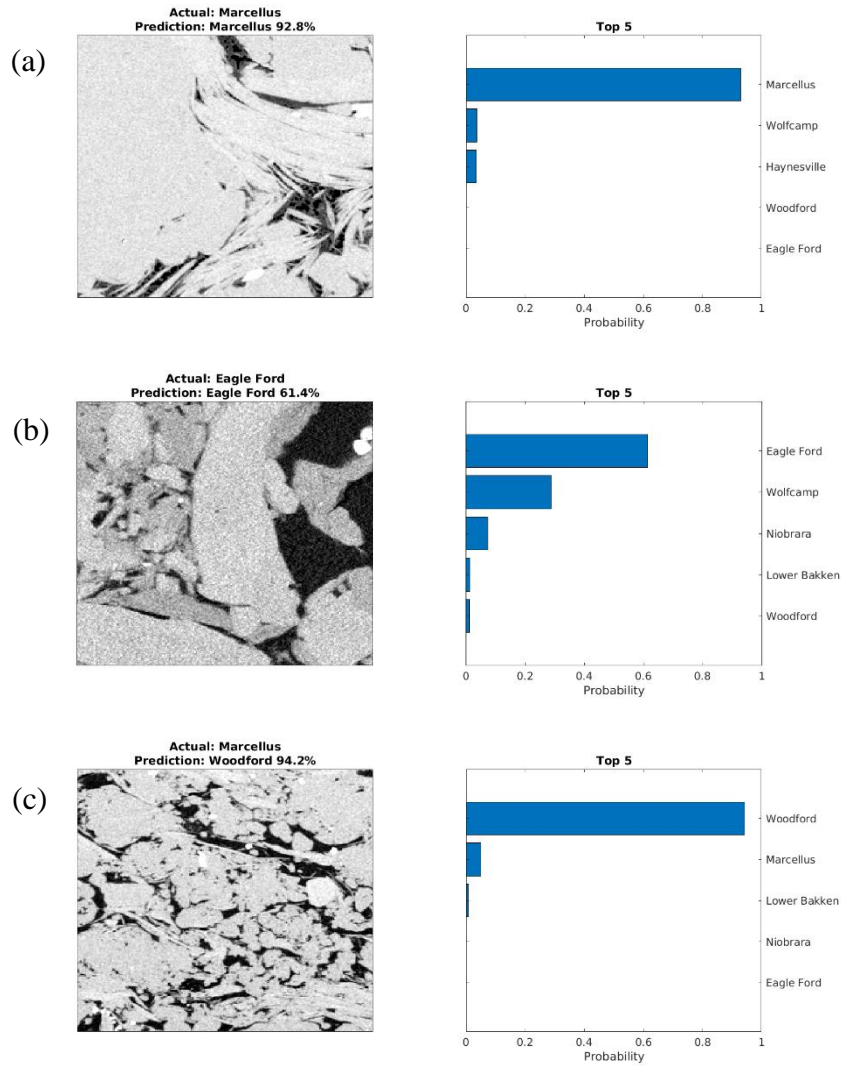
The CNN ‘AlexNet’ took approximately two hours to train using approximately 14,000 images from the original dataset. This model was run using the image classification computer,

and the specs of this computer are shown in **Table 3.1**. Accuracy was greater than 77%.

Examples of prediction results are shown in **Figure 4.3**. The confusion matrix is shown in

**Figure 4.4** and the evaluations metrics calculated from the confusion matrix are shown in **Table 4.2**.

Based on the results shown in the confusion matrix, the network had a hard time correctly identifying the Wolfcamp formation. The network predicted, seven Wolfcamp images as Niobrara, five as Marcellus, and four as Eagle Ford. The recall for each label is located in the right column, we see that the model has the hardest time at correctly identifying Niobrara with a recall of 66.2%. The precision for each column is located on the last row. Here we see that the model has a hard time with correctly labeling Wolfcamp images with a precision of 61.2%, although overall we see that the precision is 76.8%, this indicates that overall the model is mostly precise in correctly classifying.



**Figure 4.3 – Prediction results in 7 formation classification. (a) displays a correct prediction of a Marcellus predicted with a 96% probability. (b) displays an Eagle Ford tile correctly classified as Eagle Ford. Due to the low accuracy, this model was not always correct and as a result, you should expect to see incorrect predictions. An example of this is shown in (c) where the model incorrectly classified Marcellus as Woodford with 94.2% probability.**



Output Class	Target Class							
	Eagle Ford	Haynesville	Lower Bakken	Marcellus	Niobrara	Wolfcamp	Woodford	
Eagle Ford	32 9.2%	1 0.3%	2 0.6%	2 0.6%	5 1.4%	4 1.2%	2 0.6%	66.7% 33.3%
Haynesville	1 0.3%	57 16.4%	0 0.0%	5 1.4%	2 0.6%	2 0.6%	1 0.3%	83.8% 16.2%
Lower Bakken	0 0.0%	0 0.0%	23 6.6%	2 0.6%	1 0.3%	1 0.3%	0 0.0%	85.2% 14.8%
Marcellus	1 0.3%	5 1.4%	3 0.9%	40 11.5%	3 0.9%	5 1.4%	1 0.3%	69.0% 31.0%
Niobrara	6 1.7%	1 0.3%	3 0.9%	3 0.9%	43 12.4%	7 2.0%	2 0.6%	66.2% 33.8%
Wolfcamp	0 0.0%	3 0.9%	1 0.3%	0 0.0%	0 0.0%	30 8.6%	0 0.0%	88.2% 11.8%
Woodford	0 0.0%	0 0.0%	1 0.3%	1 0.3%	1 0.3%	0 0.0%	44 12.7%	93.6% 6.4%
	80.0% 20.0%	85.1% 14.9%	69.7% 30.3%	75.5% 24.5%	78.2% 21.8%	61.2% 38.8%	88.0% 12.0%	77.5% 22.5%

**Figure 4.4 – Confusion matrix in 7 formation classification. The accuracy was calculated to be 77.5%. The far-right column is the recall for individual classes, with an overall recall of 78% indicates how specific the model is on correctly classifying the formation. The last row is the precision for each formation, with an overall precision of 76.8%. This is an indication of how precise the model is on correctly classifying the formations.**

**Table 4.2 – Evaluation metric calculated from confusion matrix**

	Metric, %
Accuracy	77.5
Precision	76.8
Recall	78
F1 Score	77.4

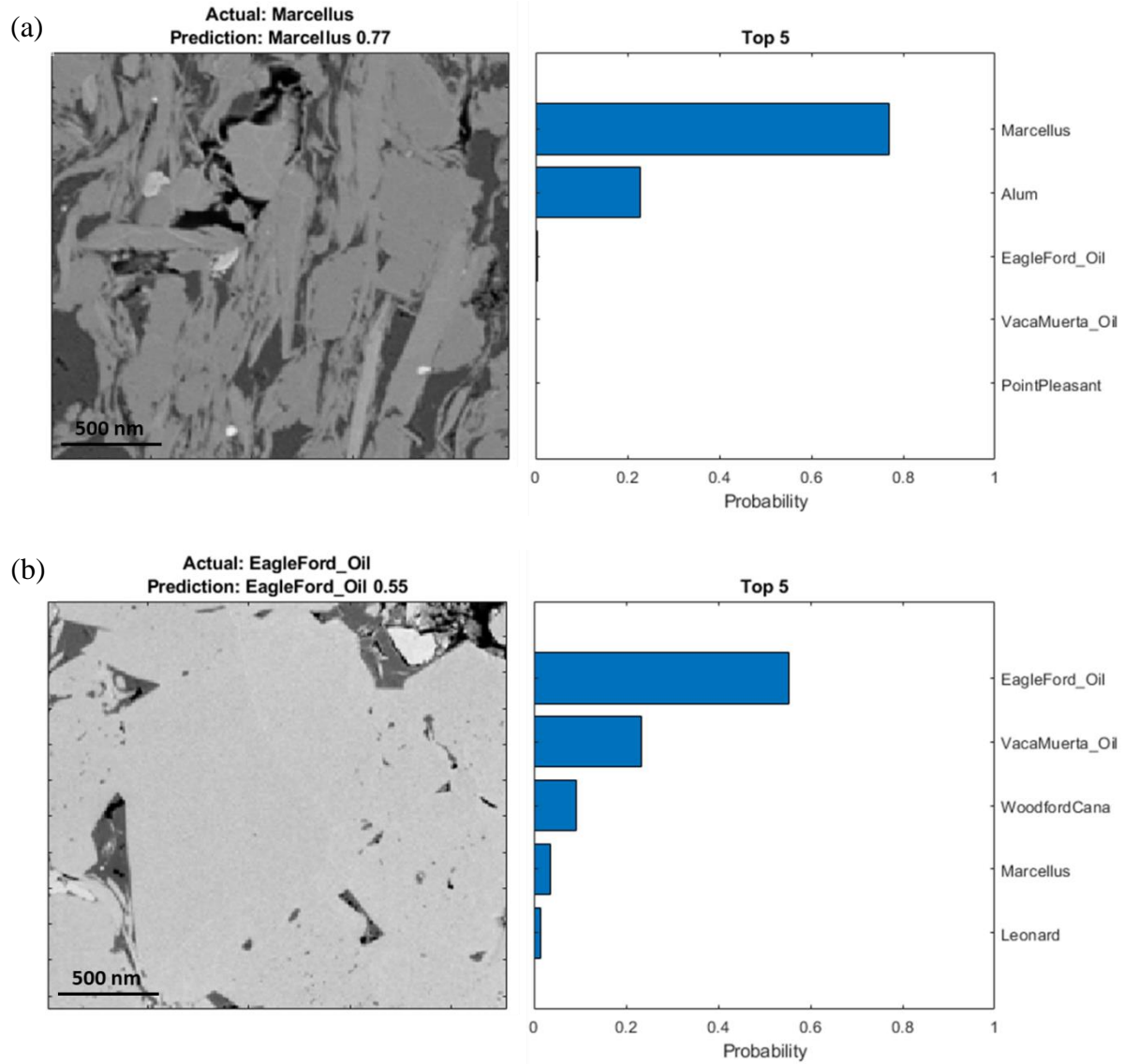
#### 4.2.2 Method 2

Due to success in Method 1 with seven formations, we decided to increase the number of formations. This dataset trained with 18 formations, such as Eagle Ford and Vaca Muerta, that had existing thermal maturity values to be separated into oil and gas windows. The confusion matrix is shown in **Table A.1** for model 1 using no augmentations, **Table A.2** for model 2 using 5x augmentations, and **Table A.3** for model 3 for 10x augmentations, located in the appendix. The evaluations metrics calculated from the confusion matrix are shown in **Table 4.3**

The best model, in this test was determined to be Model 3 with a F1 Score of 93.2%. Examples of prediction results taken from model 3 are shown in **Figure 4.5**. In **Figure 4.5b**, it is important to note the probabilities of the top 5 predictions. The model accurately predicted this Eagle Ford (Oil) tile with 55% accuracy, but the second prediction probability is Vaca Muerta Oil. This is be an indication that the microstructure is similar enough to be identified in the top 5 consistently.

**Table 4.3 – Evaluation metric calculated from confusion matrix for models 1-3**

	Model 1	Model 2	Model 3
Accuracy, %	90.4	87.8	92.8
Precision, %	90.4	89.3	93.6
Recall, %	90.6	88.1	92.8
F1 Score, %	90.4	88.7	93.2

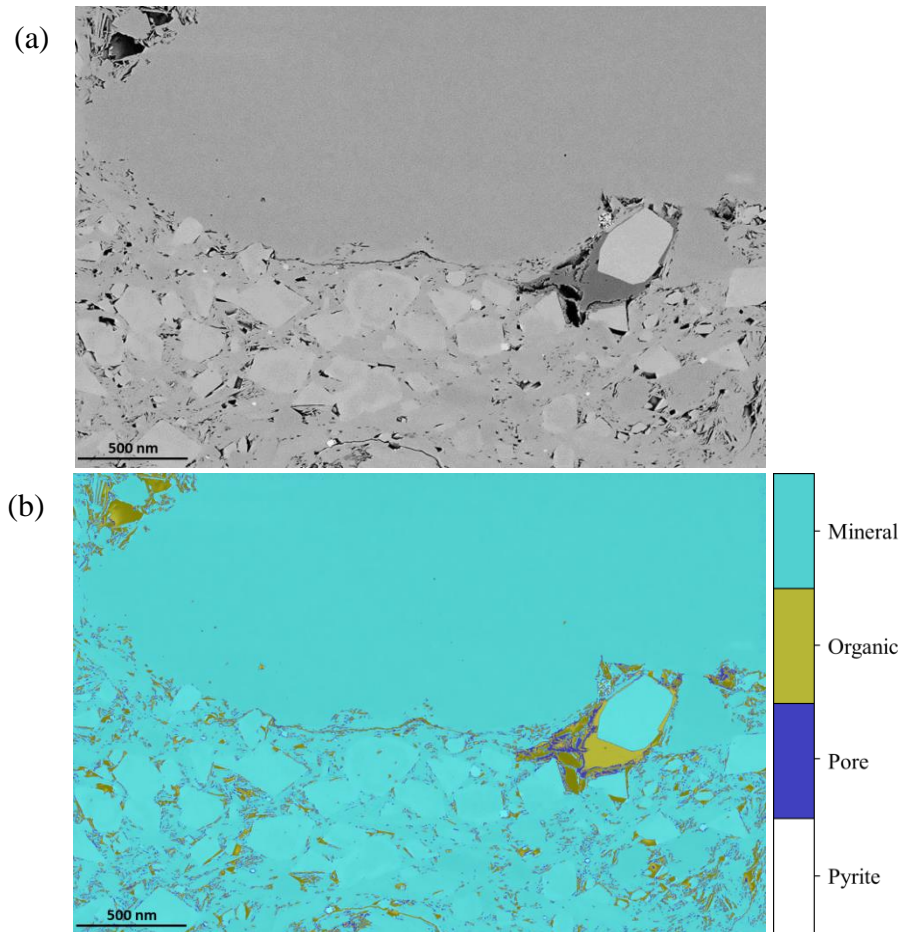


**Figure 4.5 – Prediction results taken from Model 3, which was confirmed to be the best model. (a) the model accurately predicted this Marcellus tile with a 77% probability of being Marcellus and (b) the model accurately predicted Eagle Ford Oil tile as Eagle Ford Oil with 55% probability.**

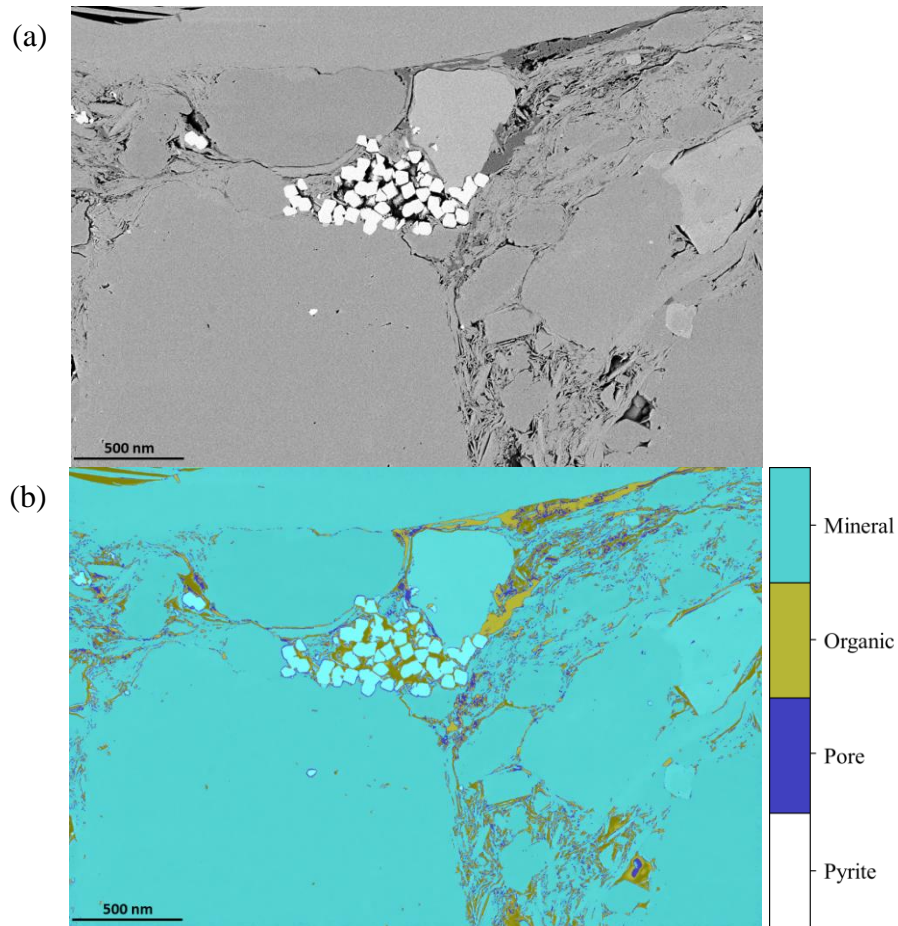
## 4.3 Image Segmentation

### 4.3.1 Method 1

Initial segmentation results proved promising. Unfortunately, due to class imbalance, the network focused on identifying the minerals, organics and pores classes and ignored pyrite. Examples of some results are shown in **Figure 4.6** and **Figure 4.7**. This initial work showed promise in the fact that machines could learn to segment effectively. Due to limitations with MATLAB at the time of testing, such as poor utilization of multi-GPUs, the next portion of the project was moved over to Python and TensorFlow.



**Figure 4.6 – Initial segmentation Results Woodford. (a) the SEM image segmented by the CNN U-Net on MATLAB. (b) The network was able to accurately segment mineral but had difficulty differentiating organics and pores. This example does not contain pyrite.**



**Figure 4.7 – Example CNN U-Net segmentation results using MATLAB. (a) Input SEM image. (b) Segmentation results shows the network was able to accurately able to segment minerals but had difficulty in differentiating between pore and organics in the image. The network was also not able to identify pyrite in the image.**

#### **4.3.2 Method 2**

This portion of the chapter focused on identifying mineral features in SEM image such calcite, clay, dolomite, pyrite, heavy minerals such as rutile etc., k-feldspar, organics, pores, plagioclase, and silica. Input images consisted of a 12-layered image made of the grayscale SEM image and 11 elemental maps. Model comparison showed mean intersection over union (mIoU) ranging from 64% to 87%, shown in **Table 4.4**. In determining pixel accuracies, intersection over union (IoU) was used as a metric comparison between class labels. A comprehensive comparison is shown in

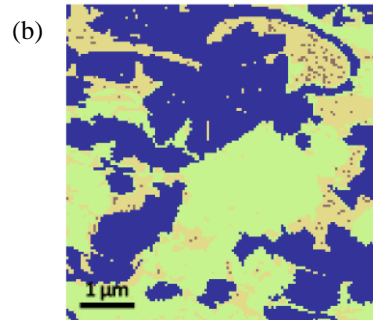
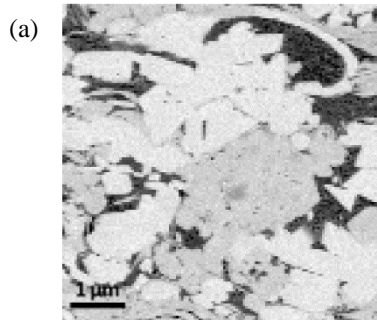
**Table 4.5** used to compare every label component in the dataset. When calculating the IoU metric, some images do not contain all the class labels. As a result, these values tend to be 0.

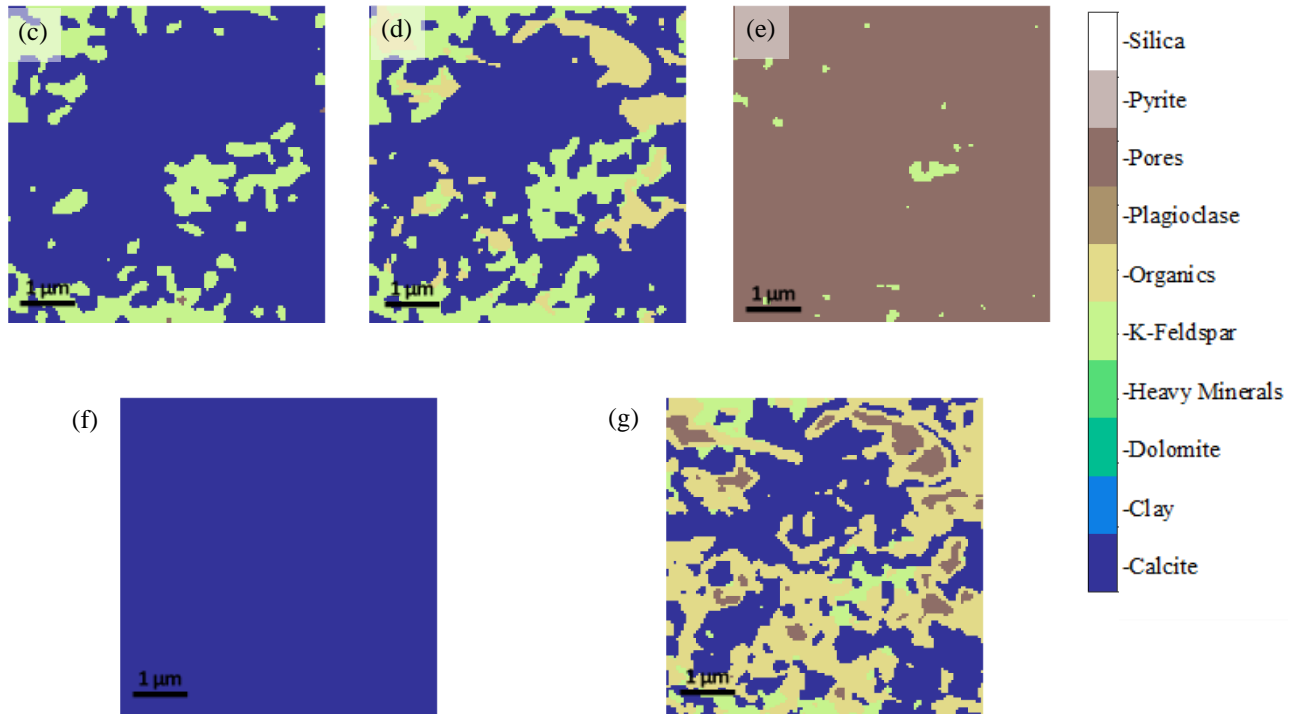
**Table 4.4 – Model mean intersection over union comparison between each model tested**

	mIoU, %
Model 1	0.6432
Model 2	0.6808
Model 3	0.8199
Model 4	0.7588
Model 5	0.8733

**Table 4.5 – Intersection over union per class for each model**

Class Label	IOU Score				
	Model 1	Model 2	Model 3	Model 4	Model 5
Calcite	0.4555	0.4782	0	0	0.5472
Clay	0	0	0	0	0
Dolomite	0	0	0	0	0
Heavy Minerals	0	0	0	0	0
K-Feldspar	0.2506	0.3163	0.0450	0.1802	0.1294
Organics	0	0.4226	0.1802	0	0.2296
Plagioclase	0	0	0	0	0
Pores	0	0	0	0	0.0625
Pyrite	0	0	0	0	0
Silica	0	0	0	0	0





**Figure 4.8 – Segmentation results produced from tested models. (a) shows the Eagle Ford SEM image with (b) the labeled image. Model 1’s predicted image is shown in (c), Model 2 is shown in (d), Model 3 is shown in (e), Model 4 is shown in (f) and Model 5 is shown in (g). Note that in (d) the that model was able to label pores to some degree whereas models 1-4 could not accurately identify not just pore but also k-feldspar. Due to the differences in network architecture, it is important to the note that the each model tested will have different predictions.**

Segmentation results are shown in **Figure 4.8**. For model comparison, we see that these models did great, but due to class imbalance, these models tended to focus on more available classes such as calcite, and k-feldspar and tended to ignore the smaller classes such as pores and heavy minerals. Based on the comparisons in **Fig. 4.9** and the accuracies table, we see that model 5 performed the best.

### 4.3.3 Method 3

As per results from Method 2, it was decided Model 5 was the best all around. By modifying this model and adding two additional layers, the accuracy was improved from 87% to



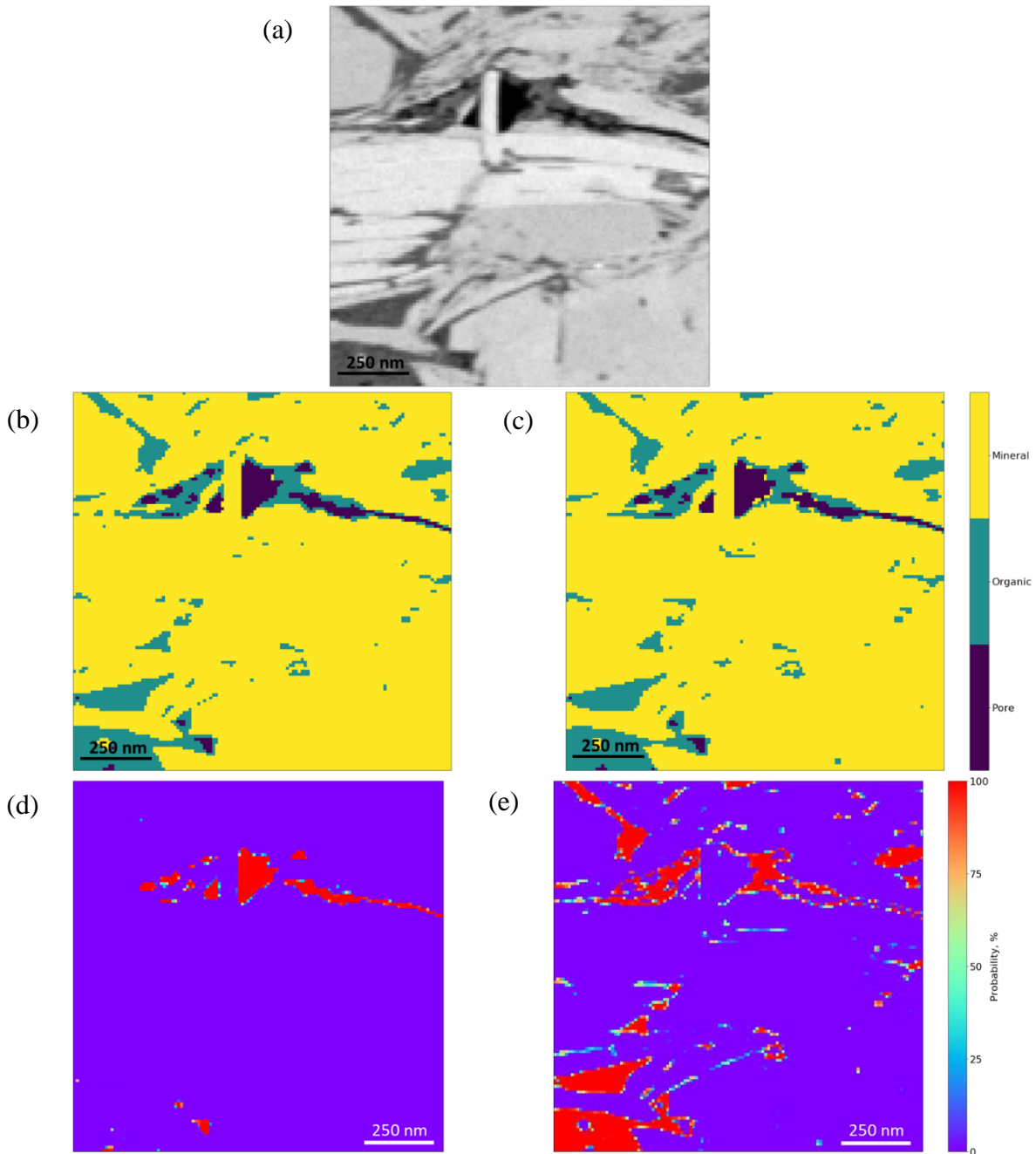
95%. The intersection over union (IoU) for this model was determined from **Table 4.6**, and the mean intersection over union was calculated as 94.7%.

**Table 4.6 – Intersection over Union per class**

	IoU
Pore	92
Organic	92.7
Mineral	99.3
Total	94.7

Results taken from the testing set are shown in **Figure 4.9**. As you can see, the model predicted images (**Figure 4.9b**) did better than the ilastik image (**Figure 4.9c**). As a result, this caused the mIoU metric to be lower than it is in reality. Therefore, future work will be conducted using this model to generate a new training dataset which will then be used in another model.

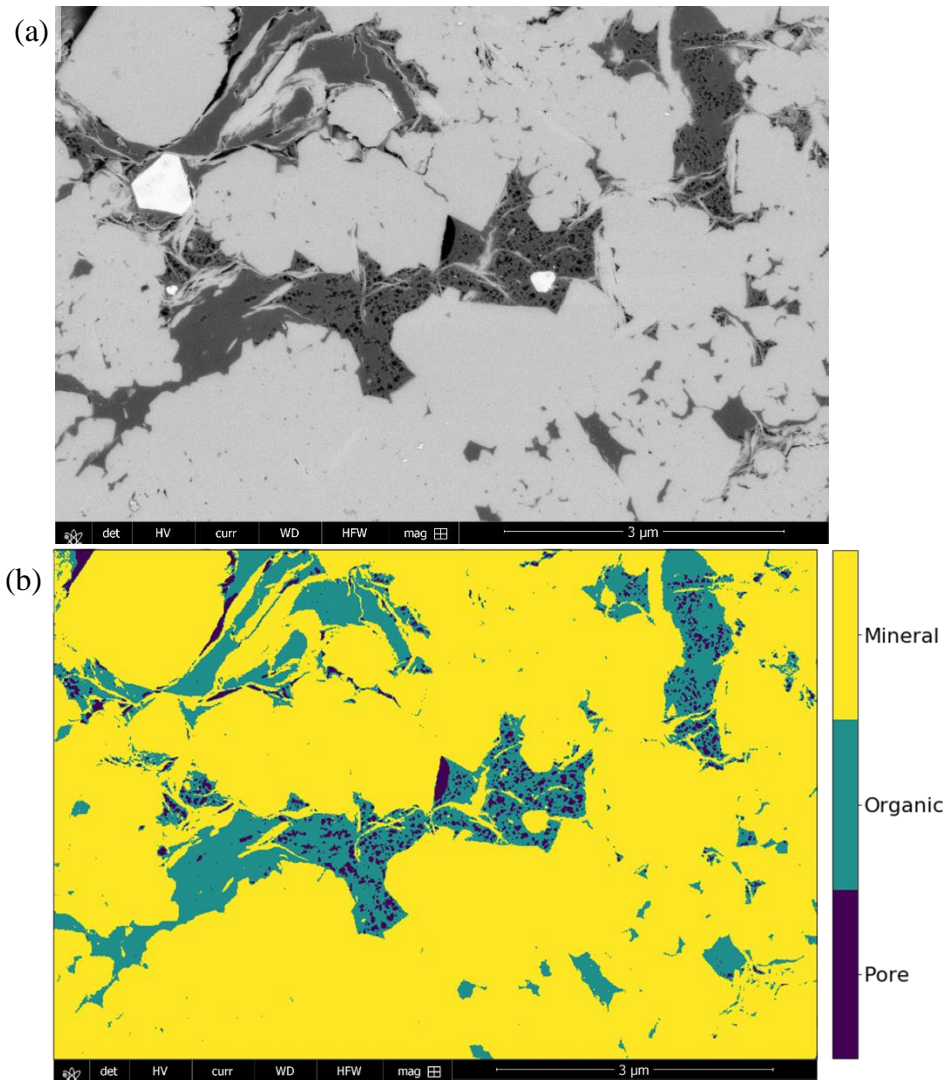
**Figs. Figure 4.9d** and **Figure 4.9e** show the representation of what the predicted results look like. The highest uncertainty between pore, organic and mineral lies along the boundary between classes.



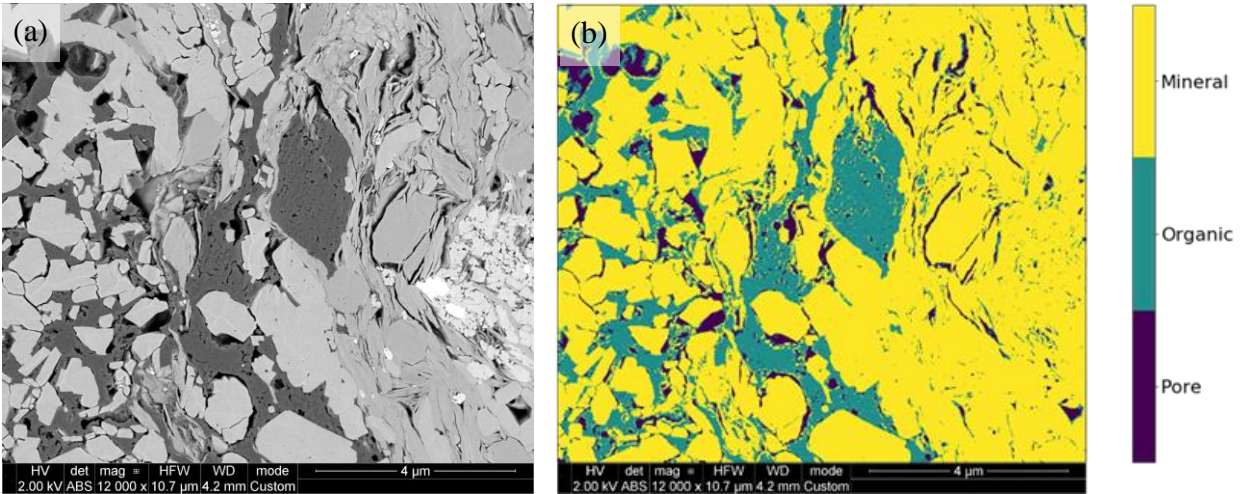
**Figure 4.9 – Unseen testing set used in model evaluation. (a) SEM image, (b) model predicted image, (c) Ilastik labeled image used for reference, (d) class representation for the pore class as a pixel probability map, and (e) the pixel probability map for organics with red being 100% probability being the given class and purple having the least probability of being the given class.**

To prove this network can work on images that were not in the original dataset, the network was run on another Woodford image, taken from another well several years prior to the

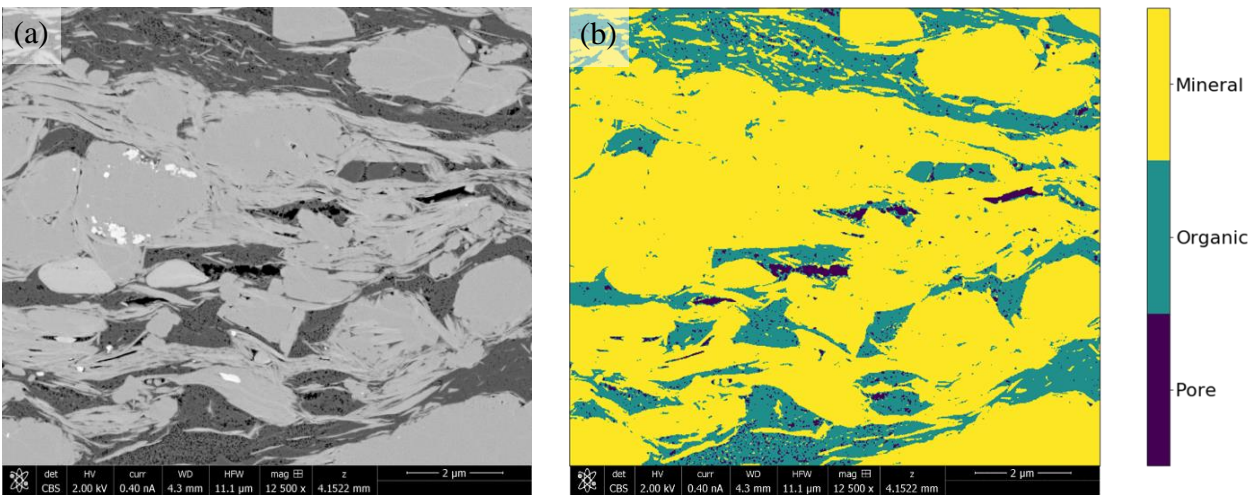
MAPS training images. The results are shown in **Figure 4.10**. Visually, it is apparent that this network does well with high contrast images. The model was run on additional unseen images such as the Eagle Ford (**Figure 4.12**), Marcellus (**Figure 4.13**), Vaca Muerta (**Figure 4.14**), and the Osage Formation (STACK) (**Figure 4.15**) which high success.



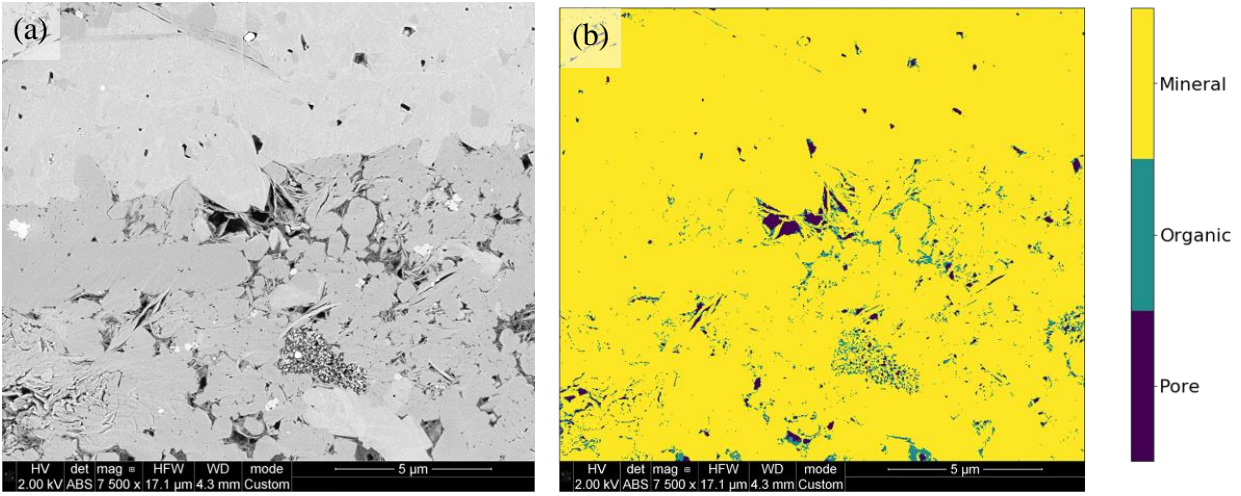
**Figure 4.10 – Unseen Woodford image taken several years prior to the Woodford MAPS training set and segmented with a convolutional neural network. The network was able to accurately segment pores and organics and also able to identify inorganic and mixed pores located in the upper left corner.**



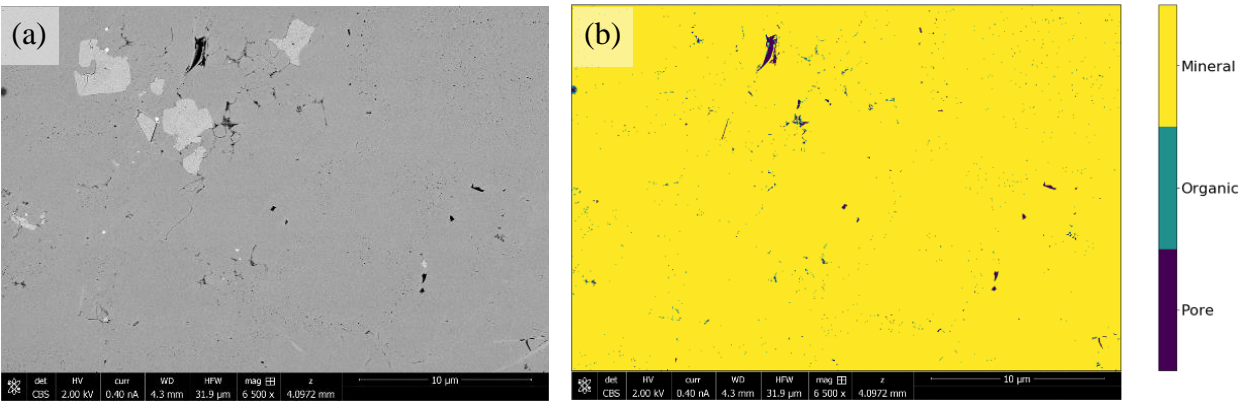
**Figure 4.11 – Eagle Ford SEM image segmented using a convolutional neural network. The Network did fairly well in this image. It was able to accurately segment the pores and pore backs in the top right corner of the image.**



**Figure 4.12 – Marcellus formation SEM image segmented using a convolutional neural network. The network was able to accurately label pores and organics in the image. The network was also able to identify, and label clay minerals located within the organics.**



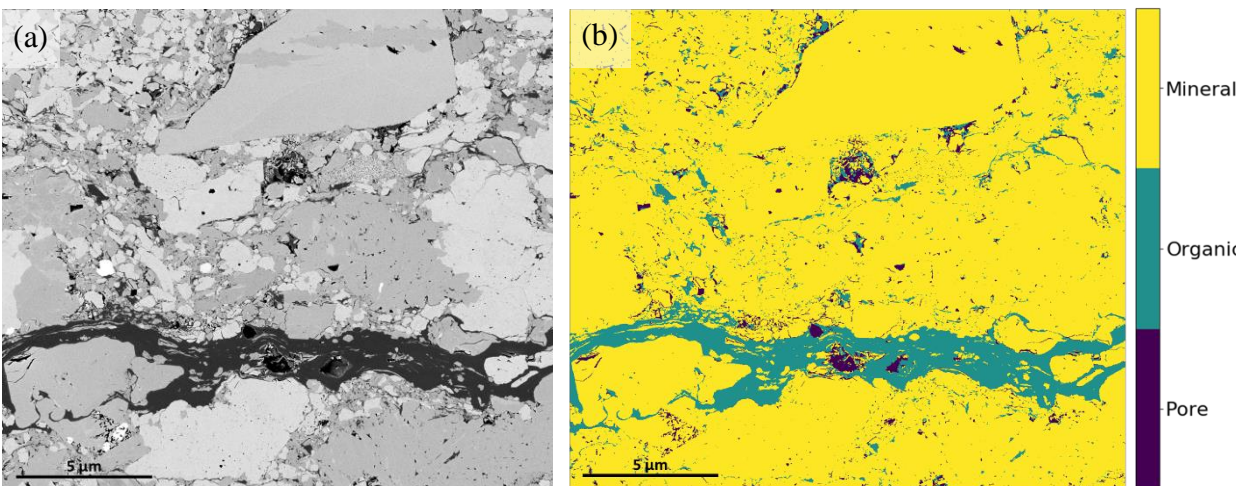
**Figure 4.13 – Vaca Muerta SEM image segmented using a convolutional neural network. The network was able to accurately differentiate between the organics and the minerals within the clay cluster in the middle of the image.**



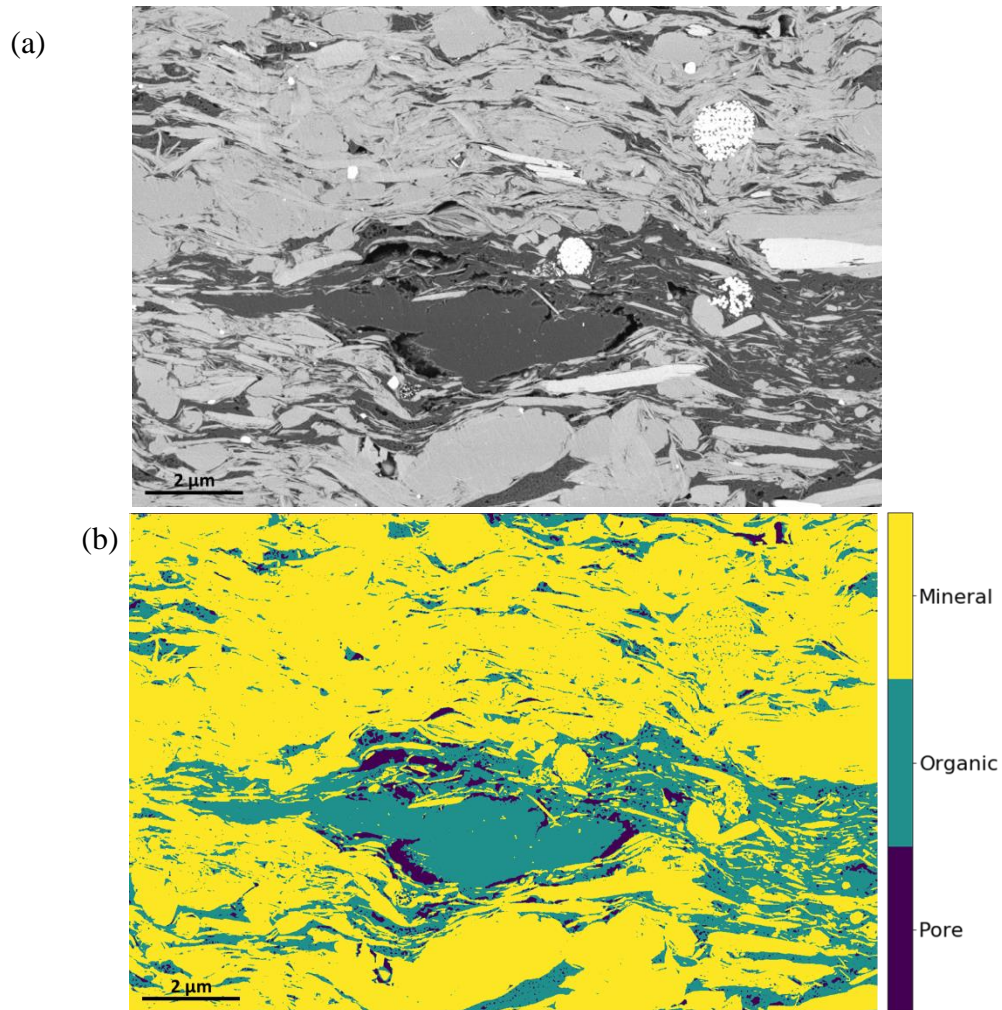
**Figure 4.14 – Osage Formation (STACK) SEM image segmented using a convolutional neural network. The network was able to segment the nano-pores in the images but incorrectly labeled them as organic. More training data would resolve this issue in the network.**

Due to the nature of SEM images, a representative area becomes relevant. Tran (2017) presented his work on developing a Representative Elementary Area (REA) in large-area/high-resolution MAPS data to determine the optimal size for investigating microstructural heterogeneity in organic rich shales. He concluded that a minimum area of 0.8 mm<sup>2</sup> for porosity and 1.2 mm<sup>2</sup> for organic matter for the Wolfcamp formation was required for statistical representation. Using these

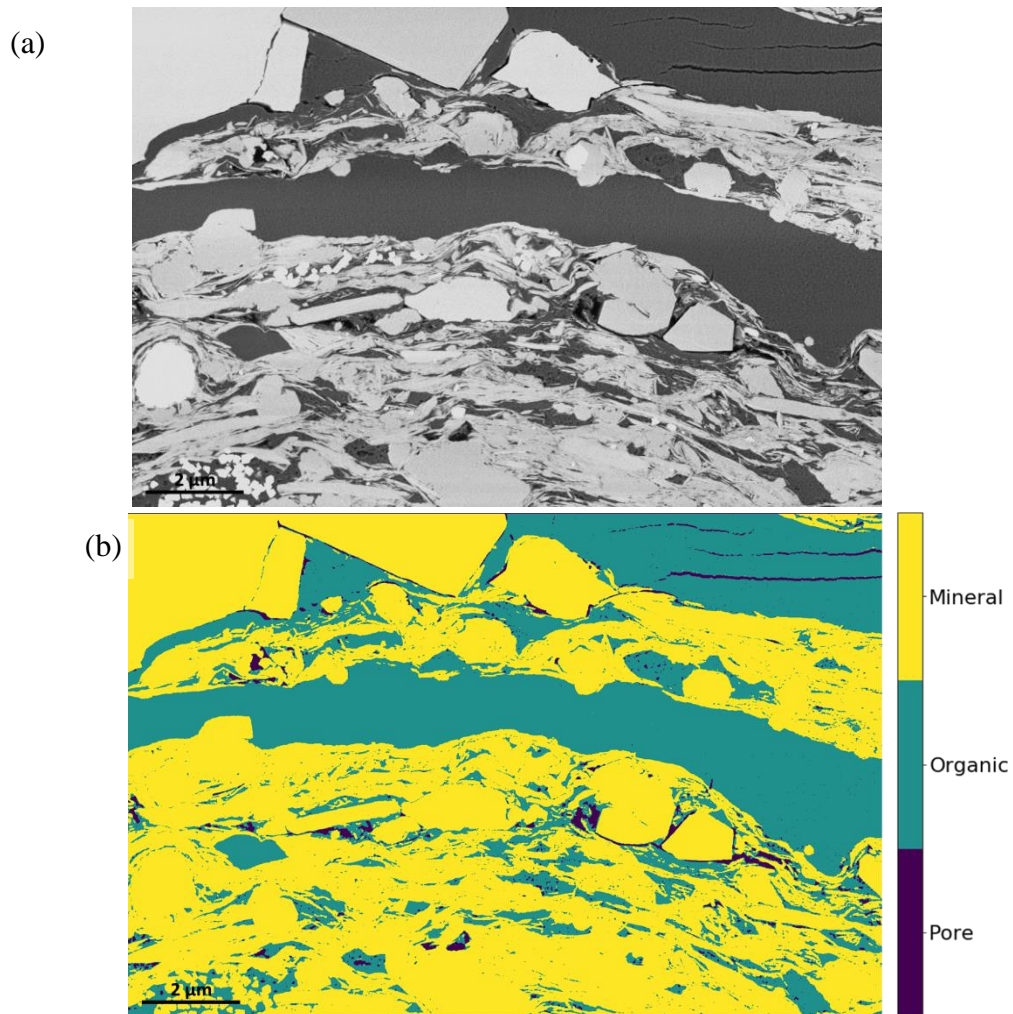
recommendations in determining porosity and total organic matter (TOM), the segmentation network was run on MAPS to determine areal porosity and areal TOM for a Vaca Muerta dataset and three Marcellus datasets. An example of a single tile segmentation of the Vaca Muerta MAPS dataset is shown in **Figure 4.15**. An example of a single tile from each of the three Marcellus MAPS datasets is shown in **Figure 4.16**, **Figure 4.17**, and **Figure 4.18**, respectively. Their respective calculations for porosity and TOM with comparative lab measurements are shown in **Table 4.7**.



**Figure 4.15 – Vaca Muerta (oil window) MAPS image used for petrophysical comparison to lab measurements. This is an example of 1 out 144 tiles from a MAPS image on what is considered a representative elementary area.**

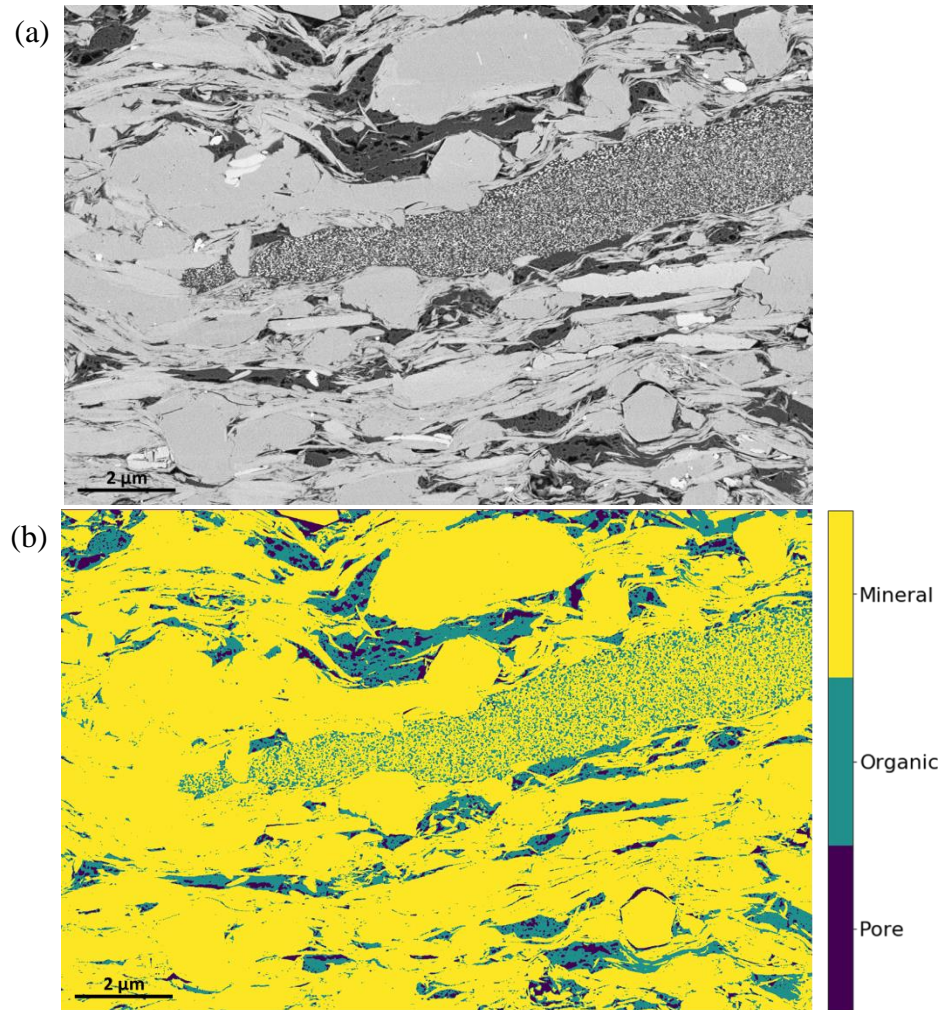


**Figure 4.16 – (a) Marcellus 6XXX.10' example tile from MAPS tile set used in estimating petrophysical properties using Representative Elementary Area. (b) This image was segmented using a convolutional neural network and able to accurately identify clay minerals within the organics and identify organics within the pyrite framboid.**



**Figure 4.17 – (a) Marcellus 6XXX.35' tile from MAPS tile set used in estimating petrophysical properties using Representative Elementary Area. (b) This image was segmented using a convolutional neural network and able to accurately identify the cracks within the organics located in the top right corner of the image.**



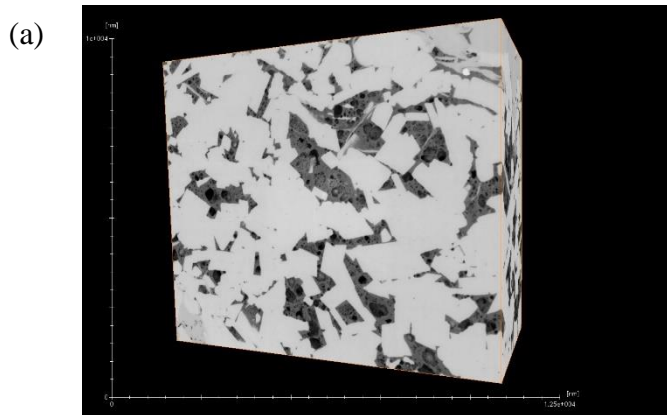


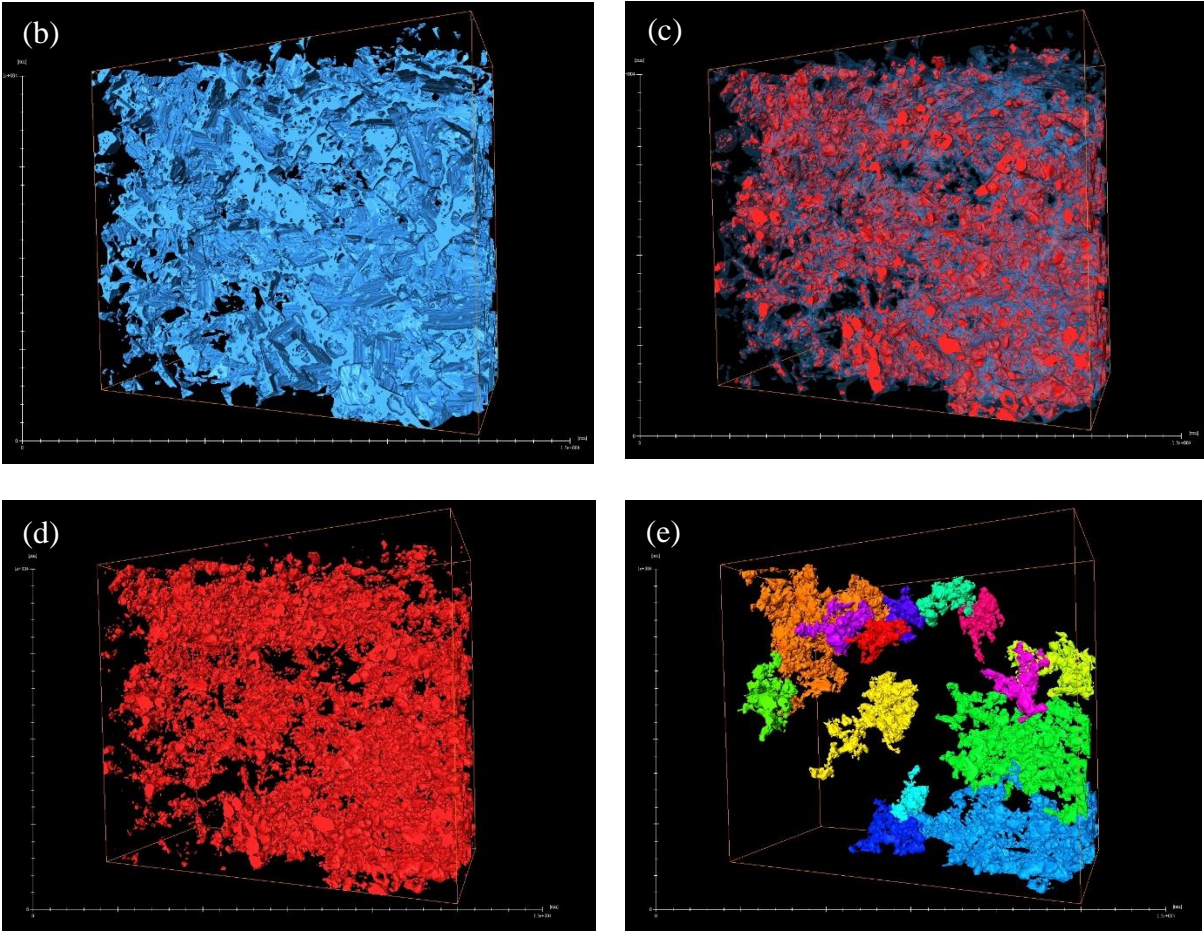
**Figure 4.18 – (a) Marcellus 6XXX.62' tile from MAPS tile set used in estimating petrophysical properties using Representative Elementary Area. (b) This image was segmented using a convolutional neural network and able to accurately differentiate organics and clay in the middle of the image.**

**Table 4.7 – Model predicted porosity and organic matter compared to lab measurements**

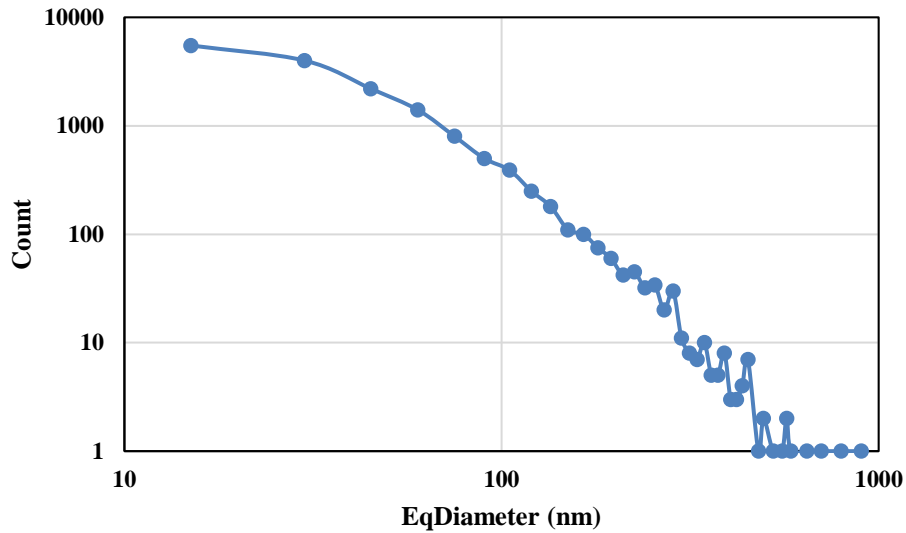
	Model Predicted		Lab Measurements	
	Porosity (Areal %)	OM (Areal %)	Porosity (Vol. %)	TOC (wt. %)
<b>Vaca Muerta</b>	2.2	5.2	2.6	4.2
<b>Marcellus 6XXX.10'</b>	2.8	18.6	10.2	9.7
<b>Marcellus 6XXX.35'</b>	7.9	18.7	9.1	9.8
<b>Marcellus 6XXX.62'</b>	2.5	10.5	9.3	6.4

This model was also applied to 3-D FIB-SEM volumes. **Figure 4.19** shows 3-D examples of the organic, porosity and connected porosity generated using Aviso 9 after segmented using a CNN. In **Figure 4.19b**, the network was able to accurately label organics, **Figure 4.19d** the pore structure, and **Figure 4.19c** displays the results of both organics and pores. When compared to traditional methods of segmentation, this method proved to be much faster in generating segmented results. **Figure 4.20** displays the generated pore size distribution using Aviso 9.





**Figure 4.19 – 3D FIB-SEM segmentation of the Eagle Ford. (a) shows the gray scale 3D representation, (b) the Total Organic Matter, (c) combined Organic Matter and Porosity, (d) total porosity, (e) pockets of connected porosity shown within the 3D scan.**



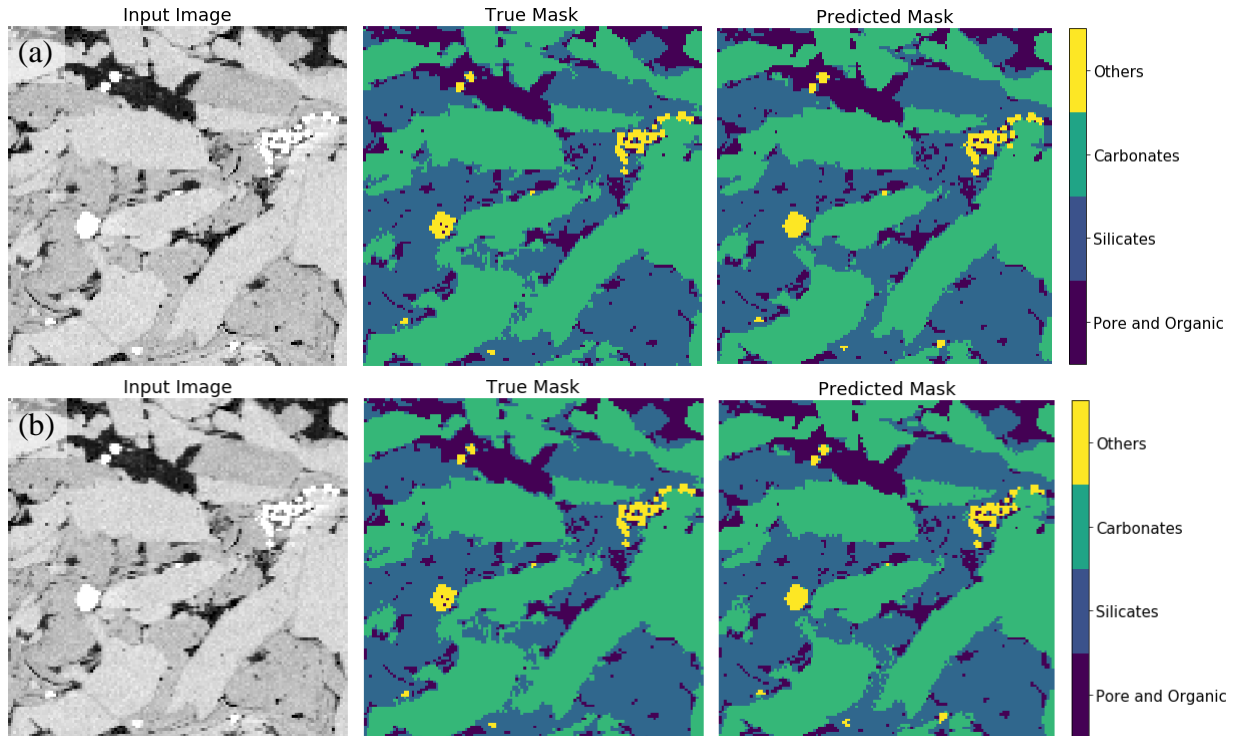
**Figure 4.20 – Pore size distribution derived from 3D segmentation of FIB-SEM scan.**

#### **4.3.4 Method 4**

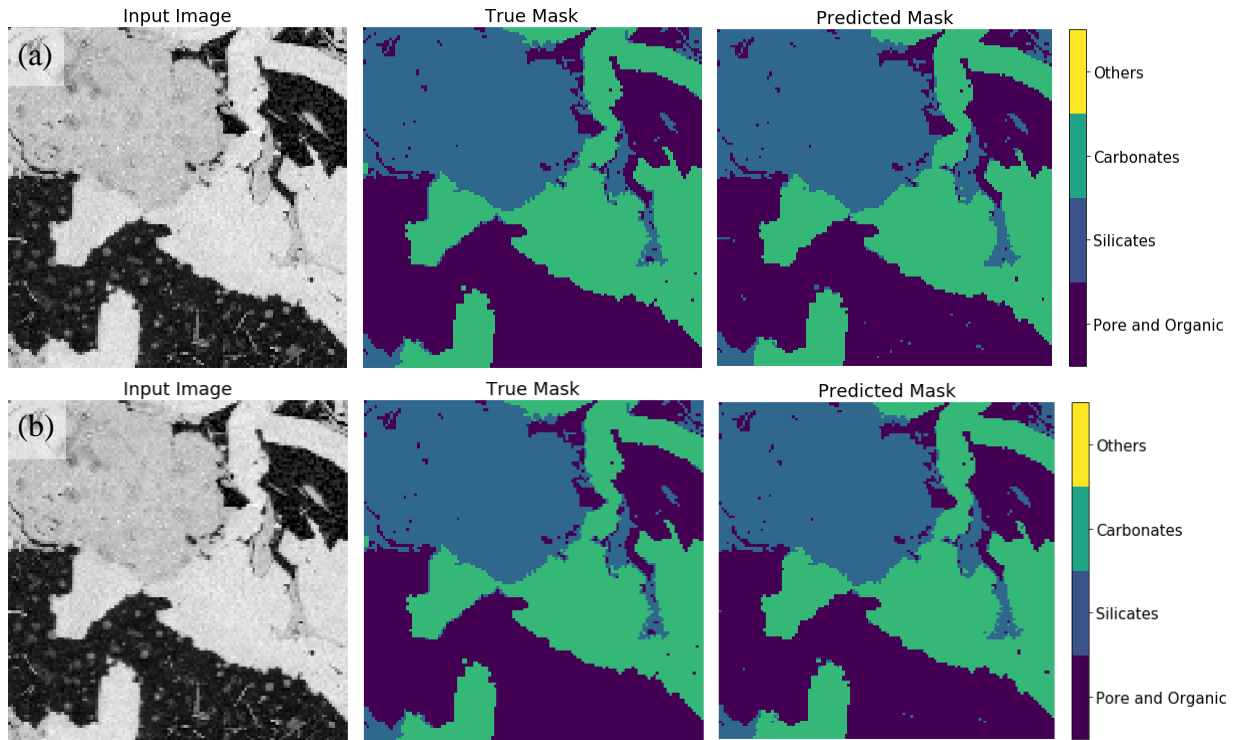
With great success with Method 3, we wanted to fine tune the model by testing filter size on applications to mineralogy. Results from this method are shown in **Figure 4.22**, **Figure 4.23** and **Figure 4.24**. From these images, we see that Model 3 with a 7x7 filter does a better job at smoothing and differentiating between grain boundaries. In **Figure 4.22**, overall Model 1 appears to have better labeling and smoothing of the grain boundaries. In addition, it appears that the network did better in correcting silicate labels from the true label as compared to model 2. In **Figure 4.22**, we see similar results to **Figure 4.22**, where Model 1 appears to perform well and even do better than human labeling of the image. Model 1 also appears to begin to label some of the minerals within the organics whereas the hand labeled images incorrectly labeled it as organic. This shows promise in accurately identifying mineral groups in SEM images. However, in **Figure 4.21**, we see that both models are struggling to identify the mineral structure as carbonate. This is especially the case along the grain/organic boundaries.

**Table 4.8 – Intersection over Union metric values for model comparison in mineral segmentation**

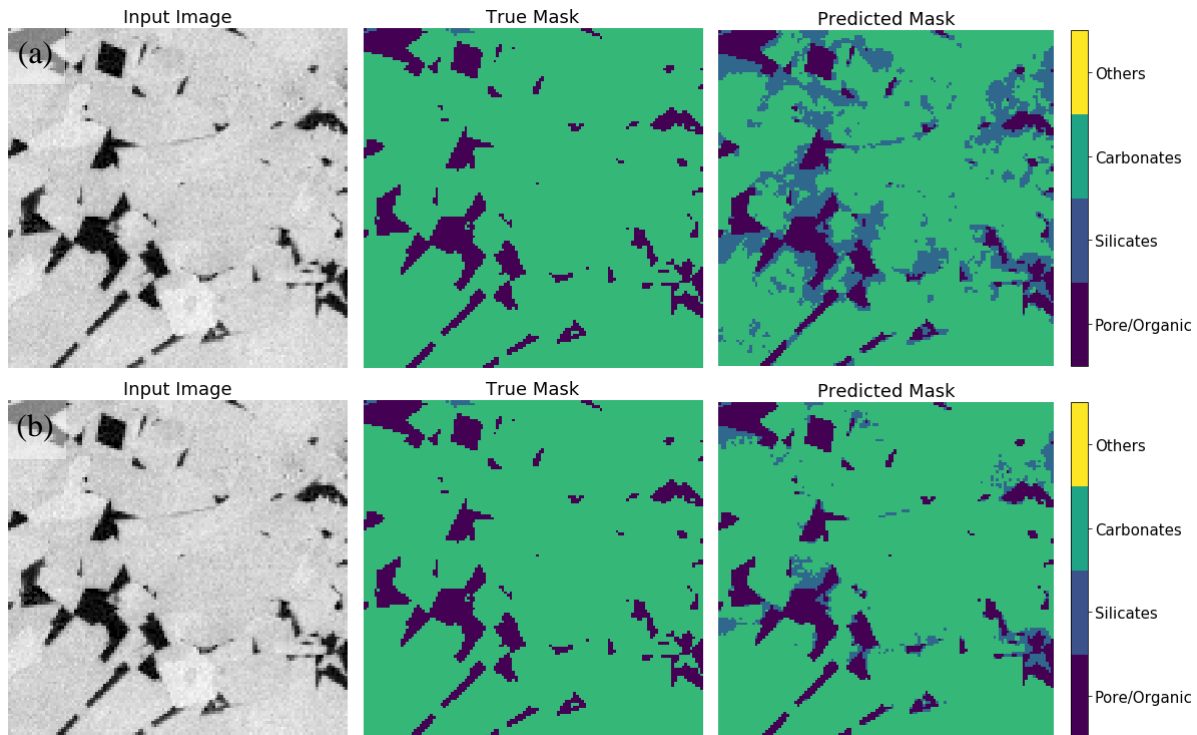
	Model 1	Model 2
Pore/Organic	97.5	97.7
Silicates	89.1	92.4
Carbonates	95.7	97
Others	90.1	90.5
Mean IoU	93.1	94.4



**Figure 4.22 – Prediction results from unseen test set for models 1 and 2. (a) Displays prediction results from model 1 and (b) prediction results from model 2 with a filter size 7x7. From both of these models, we actually see the predicted results are actually better than their hand labeled images, especially along the boundary.**



**Figure 4.23 – Prediction results testing models 1 and 2. (a) Displays prediction results from model 1 with a filter size of 1x1 and (b) displays prediction results from model 2 with a filter size 7x7. From both of these models, we actually see the predicted results are actually better than their hand labeled images by providing smoother boundaries between mineral grains.**



**Figure 4.24 – Example of what the convolutional neural network is confusing. . (a) Displays prediction results from model 1 with a filter size of 1x1 and (b) displays prediction results from model 2 with a filter size 7x7. This result shows that Model 2 is the best when identifying large areas of carbonates. More training data and network modification will help in correcting incorrect labeling such as these models.**

These results show promise in being able to accurately identify mineral groups in SEM images. Significant work needs to be done in order to separate minerals like what was done in Method 2. Future work in this area should include identifying the wettability of pores, such as identifying mixed, organic pores and/or inorganic pores.

## Chapter 5: Conclusions

### 5.1 Summary

- Evaluation of image quality proved to have an accuracy of 95%. This model proved to be effective in determining whether an image is ‘good’ or ‘bad’ and is suitable for further processing.

- Formation identification is defined as the identification of identifying similarities between microstructures of different formations. Higher accuracy in predicting formation indicates a uniqueness in microstructure. A lower accuracy in predicting formation may indicate similarity in microstructure between plays.
- Successfully developed a machine learning method as an automated and robust feature extraction tool for shale SEM images with a reasonably high degree of accuracy in identifying porosity and organic matter.
- Pore and organic size distributions can be accurately derived from MAPS SEM images using a representative elementary area determined by Tran (2017). The current model can segment a SEM MAPS dataset with an overall accuracy (mIoU) of 95% for three classes (pore, organic, mineral) without user input.
- The three-class segmentation model can process a 2048x3072 SEM image in <6 seconds without user input allowing for faster analysis of single SEM images as SEM MAPS data.
- A large number of images can be quickly batch-processed with reasonable accuracy and can also be applied to 3-D FIB-SEM image stacks. This allows for faster, easier, and more robust analysis in deriving, porosity, pore size distributions and organic matter content from 3-D image volumes.
- Basic mineralogy groups such as silicate, carbonate, and heavy minerals can be accurately identified in SEM images in combination with a pore/organic network.



## References

- Ambrose, R.J., Hartman, R.C., Diaz-Campos, M., Akkutlu, I.Y. and Sondergeld, C.H. 2010. New Pore-scale Considerations for Shale Gas in Place Calculations. SPE Unconventional Gas Conference, 23-25 February, Pittsburgh, Pennsylvania. SPE-131772-MS. <https://doi.org/10.2118/131772-MS>.
- Andrä, H., Combaret, N., Dvorkin, J., Glatt, E., Han, J., Kabel, M., Keeham, Y., Krzikalla, F., Lee, N., Madonna, C., Marsh, M., Mukerji, T., Saenger, E.H., Sain, R., Saxena, N. Ricker, S., Wiegmann, A., and Zhan, X. 2013. Digital rock physics benchmarks – Part I: Imaging and segmentation. *Computers & Geosciences* **50**: 25-32. <https://doi.org/10.1016/j.cageo.2012.09.005>
- Ayodele, T.O. 2010. Types of Machine Learning Algorithms. In *New Advances in Machine Learning*, first edition, Y. Zhang, Chap. 1, 19-48. Kosovo: IntechOpen. <https://doi.org/10.5772/9385>
- Blaus, B. 2014. Medical gallery of Blausen Medical 2014. *WikiJournal of Medicine*. <https://doi.org/10.15347/wjm/2014.010>
- Boston. 2020. What is GPU Computing? Boston, <https://www.boston.co.uk/info/nvidia-kepler/what-is-gpu-computing.aspx> (accessed 06 April 2020).
- Brace, W.F. 1980. Permeability of crystalline and argillaceous rocks. *International Journal of Rock Mechanics and Mining Sciences & Geomechanics Abstracts* **17**(5): 241-251. [https://doi.org/10.1016/0148-9062\(80\)90807-4](https://doi.org/10.1016/0148-9062(80)90807-4)
- Brownlee, J. 2017. What is the Difference Between Test and Validation Datasets? *Machine Learning Mastery*. <https://machinelearningmastery.com/difference-test-validation-datasets/> (accessed 08 April 2020)
- Brownlee, J. 2019 A Gentle Introduction to k-Fold Cross-Validation. *Machine Learning Mastery*. <https://machinelearningmastery.com/k-fold-cross-validation/> (accessed 09 April 2020).
- Bustin, R.M., Bustin, A.M.M., Cui, A., Ross, D., and Pathi, V.M. 2008. Impact of Shale Properties on Pore Structure and Storage Characteristics. SPE Shale Gas Production Conference, 16-18 November, Fort Worth, Texas, USA. SPE-119892-MS. <https://doi.org/10.2118/119892-MS>
- Cameron, D. 2014. Big Data in Exploration and Production: Silicon Snake-Oil, Magic Bullet, or Useful Tool. Presented at SPE Intelligent Energy Conference and Exhibition, 1-3 April, Utrecht, The Netherlands. SPE-167837-MS. <https://doi.org/10.2118/167837-MS>
- Chen, Z., Liu, X., Yang, J., Little, E., Zhou, Y. 2020. Deep learning-based method for SEM image segmentation in mineral characterization, an example from Duvernay Shale samples in Western Canada Sedimentary Basin. *Computers & Geosciences* **138**. <https://doi.org/10.1016/j.cageo.2020.104450>
- Cover, T.M. and Hart, P.E. 1967. Nearest neighbor pattern classification. *IEEE Trans. Inform. Theory* **13**:21-27.

- Clarkson, C.R., Solano, N., Bustin, R.M., Chalmers, G.R.L., He, L. Melnichenko, Y.B., Radliński 2013. Pore Structure characterization of North American shale gas reservoirs using USANS/SANS, gas adsorption, and mercury intrusion. *Fuel* **103**: 606-616. <https://doi.org/10.1016/j.fuel.2012.06.119>
- Curtis, M.E. Ambrose, R.J., and Sondergeld, C.H. 2010. Structural Characterization of Gas Shales on the Micro- and Nano-Scales. Paper presented at the Canadian Unconventional Resources and International Petroleum Conference, 19-21 October, Calgary, Alberta, Canada. SPE-137693-MS. <https://doi.org/10.2118/137693-MS>.
- Curtis, M.E., Goergen, E.T., Jernigen J.D., Sondergeld, C.H., and Rai, C.S., 2014. High-Resolution Mapping of the Distribution and Connectivity of Organic Matter in Shales. Paper presented at the SPE Annual Technical Conference and Exhibition, Amsterdam, The Netherlands, 27-29 October. SPE-170787-MS. <https://doi.org/10.2118/170787-MS>
- Curtis, M.E. 2019. Lecture 7 (Geological Sample Preparation). Presented at the University of Oklahoma.
- Drakos, G. 2018. How to select the Right Evaluation Metric for Machine Learning Models: Part 3 Classification Metrics. *Medium*. <https://medium.com/@george.drakos62/how-to-select-the-right-evaluation-metric-for-machine-learning-models-part-3-classification-3eac420ec991> (accessed 08 April 2020).
- EIA. 2011. Maps: Oil and Gas Exploration, Resources, and Production: North American Shale Plays. [https://www.eia.gov/oil\\_gas/rpd/northamer\\_gas.pdf](https://www.eia.gov/oil_gas/rpd/northamer_gas.pdf)
- EIA. 2016. Maps: Oil and Gas Exploration, Resources, and Production: Lower 48 states shale plays. [https://www.eia.gov/maps/images/shale\\_gas\\_lower48.pdf](https://www.eia.gov/maps/images/shale_gas_lower48.pdf)
- EIA. 2020. Assumptions to the Annual Energy Outlook 2020: Oil and Gas Supply Module: United States. <https://www.eia.gov/outlooks/aeo/assumptions/pdf/oilgas.pdf> (Accessed 1 March 2020).
- Foote, K.D. 2019. A Brief History of Machine Learning. Dataversity, <https://www.dataversity.net/a-brief-history-of-machine-learning/#> (accessed 20 November 2019).
- Fredrich, J.T., Menéndez, B., and Wong, T.F. 1995. Imaging the Pore Structure of Geomaterials. *Science* **268**(5208): 276-279. <https://doi.org/10.1126/science.268.5208.276>
- Géron, A. 2019. *Hands-on Machine Learning with Scikit-Learn, Keras & TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, second edition. Sebastopol, California: O'Reilly Media, Inc.
- Goergen, E.T., Curtis, M.E., Jernigen, J., Sondergeld, C., and Rai, C.. 2014. Integrated Petrophysical Properties and Multi-Scaled SEM Microstructural Characterization. Paper presented at URTeC, 25-27 August, Denver, Colorado, USA. URTeC-1922739. <https://doi.org/10.15530/urtec-2014-1922739>
- Goldstein, G.I, Newbury, D.E., Echlin, P., Joy, D.C., Fiori, C., Lifshin, E. 1981. Scanning electron microscopy and x-ray microanalysis; Plenum Press: New York, New York.

- Ghahramani, Z. 2004. Unsupervised Learning. In *Advanced Lectures on Machine Learning LNAI 3176*, Bousquet, O., Raetsch G., and von Luxburg, U., Chap. 5, 73-113. Berlin, Germany: Springer-Verlag Berlin Heidelberg.
- Gupta, P. 2017. *Decision Trees in Machine Learning*. Medium: Towards Data Science. <https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052> (accessed 03 April 2020).
- Herman, G.T. 2009. *Fundamentals of Computerized Tomography: Image reconstruction from Projections: Second Edition*.
- Higham, D. 2018. How we use image semantic segmentation. The Medium: DigitalBridge, 21 February 2018, <https://medium.com/digitalbridge/how-we-use-image-semantic-segmentation-e85fac734caf> (accessed 25 April 2019)
- Iassibiv, P., Gebrenegus, T., and Tuller, M. 2009. Segmentation of X-ray computer tomography images of porous materials: A crucial step for characterization and quantitative analysis of pore structures. *Water Resources Research* **45**(9):
- Ishwarappa, and Anuradha, J. 2015. A Brief Introduction on Big Data 5Vs Characteristics and Hadoop Technology. *Procedia Computer Science* **48**: 319-324. <https://doi.org/10.1016/j.procs.2015.04.188>
- Jordan, J. 2018, Evaluating Image Segmentation Models. *Jeremy Jordan*. <https://www.jeremyjordan.me/evaluating-image-segmentation-models/> (accessed 08 April 2020)
- Kalam, M.Z. 2012. Digital Rock Physics for Fast and Accurate Special Core Analysis in Carbonates. In *New Technologies in Oil and Gas Industry*, ed. J.S. Gomes, Chap. 9
- Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, and A.Y. 2004. A local search approximation algorithm for k-means clustering. *Computational geometry*, 28: 89-112. <https://doi.org/10.1016/j.comgeo.2004.03.003>
- Knackstedt, M.A., Arns, C.H., Limaye, A., Skellariou, A., Senden, T.J., Sheppard, A.P., Pinczewski, W.V., and Bunn, G.F.. 2004. Digital Core Laboratory: Properties of reservoir core derived from 3D images. Presented at SPE Asia Pacific Conference on Integrated Modelling for Asset Management, 29-30 March, Kuala Lumpur, Malaysia. <https://doi.org/10.2118/87009-MS>
- Knackstedt, M., Carnerup, A., Golab, A., Sok, R., Young, B. and Ripe, L. 2013. Petrophysical Characterization of Unconventional Reservoir Core At Multiple Scales. *Petrophysics* **54**(3): 216-223.
- Knaup, A.S., Jernigen, J.D., Curtis, M.E., Sholeen, J., Borer IV, J.J., Sondergled, C., and Rai, C. 2019. Unconventional Reservoir Microstructural Analysis Using SEM and Machine Learning. Presented at URTeC, 22-24 July, Denver, Colorado, USA. URTeC-638. <https://doi.org/10.1530/urtec-2019-638>

- Krizhevsky, A., Sutskever, I., Hinton, G.E. 2017. ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*, **60**(6): 84-90. <http://doi.acm.org/10.1145/3065386>
- Kulkarni, M. 2018. Decision trees for Classification: A Machine Learning Algorithm. Xorient Blog, <https://www.xoriant.com/blog/product-engineering/decision-trees-machine-learning-algorithm.html> (accessed 04 April 2020)
- Loucks, R.C., Reed, R.M., Ruppel, S.C., and Jarvie, D.M. 2009. Morphology, Genesis, and Distribution of Nanometer-Scale Pores in Siliceous Mudstones of the Mississippian Barnett Shale. *Journal of Sedimentary Research* **79**: 848-861. <https://doi.org/10.2110/jsr.2009.092>.
- LeCun, Y., Boser, B., Denker, J.S., Henderson, D. 1989. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, **1**(4): 541-551. <https://doi.org/10.1162/neco.1989.1.4.541>
- McCullough, W.S. and Pitts, W. 1943. A Logical Calculus of the Ideas Immanent in Nervous Activity. *Bulletin of Mathematical Biophysics* **5**:115-133.
- Mahendru, K. 2019. What are Loss Functions in Machine Learning and How Do They Work?. *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2019/08/detailed-guide-7-loss-functions-machine-learning-python-code/> (accessed 08 April 2020)
- MissingLink.ai. 2020. 7 Types of Activation Functions in Neural Networks: How to Choose? Missing Link.ai. <https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/> (accessed 06 April 2020).
- Mohammadpoor, M. and Torabi F. 2018. *In press*. Big Data analytics in oil and gas industry: An emerging trend. *Petroleum*. <https://doi.org/10.1016/j.petlm.2018.11.001>
- Monterio, S.N. and Paciornik, S. 2017. From Historical Backgrounds to Recent Advances in 3D Characterization of Materials: An Overview. *JOM* **69**: 84-92. <https://doi.org/10.1007/s11837-016-2203-8>.
- Nielsen, M. 2015. *Neural Networks and Deep Learning*, Determination Press.
- NewTechDojo. 2018. 13 List of Machine Learning Algorithms with Details [2018 Updated]. NewTechDojo, <https://www.newtechdojo.com/list-machine-learning-algorithms/> (03 April 2020)
- OmniSci. 2020. CPU vs GPU: Definition and FAQs. OmniSci. <https://www.omnisci.com/technical-glossary/cpu-vs-gpu> (Accessed 06 April 2020)
- Osisanwo, F.Y., Akinsola, J.E.T., Awedele, O., Hinmikaiye, J.O., Olakanmi, O., and Akinjobi, J. 2017. Supervised Machine Learning Algorithms: Classification and Comparison. *International Journal of Computer Trends and Technology*, **48**(3): 128-138. <https://doi.org/10.14445/22312803/IJCTT-V48P126>
- Prasad, M. 2001. Mapping impedance microstructures in rocks with acoustic microscopy. *The Leading Edge* **20**(2): 113-224. <https://doi.org/10.1190/1.1438902>

- Puget, J.F. 2016. IT Best Kept Secret is Optimization: What is Machine Learning? IBM Community, [https://www.ibm.com/developerworks/community/blogs/jfp/entry/What\\_Is\\_Machine\\_Learning?lang=en](https://www.ibm.com/developerworks/community/blogs/jfp/entry/What_Is_Machine_Learning?lang=en) (accessed 3 September 2019).
- Reimer, L. 1998. *Scanning electron microscopy: physics of image formation and microanalysis*, second edition. Springer.
- Ren, Q., Li, M., and Han, S. 2019. Tectonic discrimination of olivine in basalt using data mining techniques based on major elements: a comparative study from multiple perspectives. Big Earth Data, <https://doi.org/10.1080/20964471.2019.1572452>
- Rodriguez, R., Crandall, D., Song, X., Verba, C, and Soeder, D. 2014. *Imaging Techniques for Analyzing Shale Pores and Minerals*. NETL-TRS-6-2014. NETL Technical Report Series; U.S. Department of Energy, National Energy Technology Laboratory: Morgantown, WV.
- Ronneberger, O. Fischer, P., and Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, eds Navab, N., Hornegger, J., Wells, W., Frangi, A. Springer, Cham. [https://doi.org/10.1007/978-3-319-24574-4\\_28](https://doi.org/10.1007/978-3-319-24574-4_28)
- Rosebrock, A. 2016. Intersection over Union (IoU) for object detection. *Pyimagesearch: you can master Computer Vision, Deep Learning, and OpenCV*. <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/> (accessed 07 April 2020).
- Rosenblatt, F. 1958. The Perceptron: A probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review* 65(6): 386-408.
- SAS. 2019. Machine Learning: What it is and why it matters. SAS, [https://www.sas.com/en\\_us/insights/analytics/machine-learning.html](https://www.sas.com/en_us/insights/analytics/machine-learning.html) (accessed 28 August 2019).
- Saif, T., Lin, Q., Bijeljic, B., and Blunt, M.J. 2017. Microstructural imaging and characterization of oil shale before and after pyrolysis. *Fuel* 197: 562-574. <http://doi.org/10.1016/j.fuel.2017.02.030>
- Samuel, A.L. 1959. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development* 3(3): 221-229.
- Slatt, R.M., and O'Brien, N.R. Pore types in the Barnett and Woodford gas shales: Contribution to understanding gas storage and migration pathways in fine-grained rocks. *AAPG Bulletin* 95 (12): 2017-2030. <https://doi.org/10.1306/03301110145>
- Sok, R.M., Varslot, T., Ghous, A., Latham, S., Sheppard, A.P., and Knackstedt, M.A. 2010. Pore Scale Characterization of Carbonates at Multiple Scales: Integration of Micro-CT, BSEM, and FIBSEM. *Petrophysics* 51(6): 379-387.
- Sondergeld, C.H., Ambrose, R.J., Rai, C.S., and Moncrieff, J. 2010. Micro-Structural Studies of Gas Shales. SPE Unconventional Gas Conference, 23-25 February, Pittsburgh, Pennsylvania, USA. SPE-131771-MS. <https://doi.org/10.2118/131771-MS>

- Sondergeld, C.H., Rai, C.S., and Curtis, M.E. 2013. Relationship Between Organic Shale Microstructure and Hydrocarbon Generation. SPE Unconventional Resources Conference-USA, 10-12 April. SPE-164540-MS. <https://doi.org/10.2118/164540-MS>.
- Tiu, E. 2019. Metrics to Evaluate Your Semantic Segmentation Model. *Medium: Towards Data Science*. <https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2> (accessed 07 April 2020)
- Tran, H. 2017. *Investigating Microstructural Heterogeneity in Organic Shale via Large-Scale, High resolution SEM Imaging*. MS thesis, University of Oklahoma, Norman, Oklahoma (May 2017).
- Turing, A.M. 1950. Computing Machinery and Intelligence. *Mind* 49: 433-460. <https://doi.org/10.1093/mind/LIX.236.433>
- Wiederhold, G. McCarthy, J., and Feigenbaum, E. 1990. Memorial Resolution: Arthur L. Samuel. Stanford University Historical Society.
- Zeiler, M.D. and Fergus, R. 2014. Visualizing and Understanding Convolutional Networks. European Conference on Computer Vision, Zurich, Switzerland, 5-12 September.

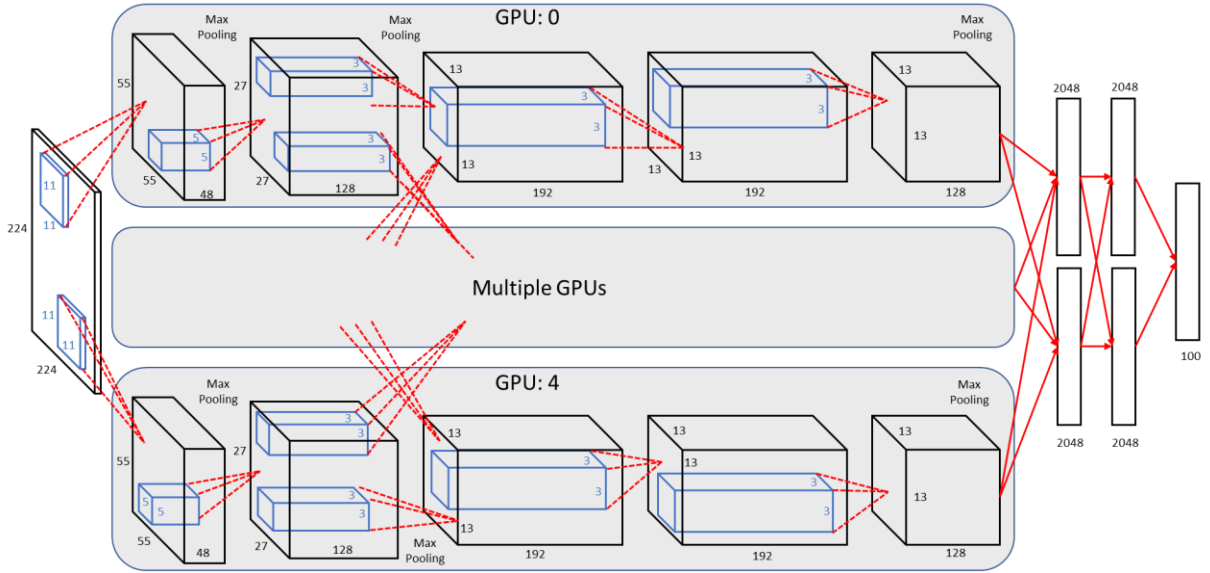
## Appendix A

**Table A.5.1 – Model training parameters for Image Quality**

	Parameters
Optimizer	ADAM
Initial Learning Rate	0.001
Learn Rate Drop Factor	0.05
Learn Rate Drop Period	114
L2-Regularization	0.0001
Mini-batch Size (images)	128
Gradient Decay Factor (%)	0.9
Squared Gradient Decay Factor (%)	0.999

**Table A.5.2 – Initial training parameters used in Formation Classification with seven formations**

	Parameters
Optimizer	ADAM
Initial Learning Rate	0.001
Learn Rate Drop Factor	0.05
Learn Rate Drop Period	114
L2-Regularization	0.0001
Mini-Batch Size (images)	256
Gradient Decay Factor (%)	0.9
Squared Gradient Decay Factor (%)	0.999



**Figure A.5.1 – AlexNet architecture showing delineation of responsibilities between multiple GPUs during the training process. Each GPU communicates with the other GPUs on specific layers. This is shown in between the second and third layer and between the fully connected layers at the end of the network (modified from Krizhevsky et al. 2017; Knaup et al. 2019)**

**Table A.5.3 – Parameters used in Model training**

	Parameters
Optimizer	ADAM
Initial Learning Rate	0.001
Beta 1	0.9
Beta 2	0.999
Epsilon	1e-7
Mini-batch Size (images)	64
Gradient Decay Rate	4e-5
Loss	Categorical Cross-entropy



		Confusion Matrix																				
Output Class	Alum	187	0	0	1	0	0	0	1	0	0	0	3	0	0	0	0	0	0	0	0	97.4%
		5.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.6%
	Barnett	0	146	0	0	0	1	0	0	0	1	2	0	6	0	0	7	2	0	0	0	88.5%
		0.0%	4.3%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.2%	0.0%	0.0%	0.2%	0.1%	0.0%	0.1%	0.0%	11.5%
	Collingwood	1	0	152	9	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0	3	90.5%
		0.0%	0.0%	4.4%	0.3%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	9.5%
	Duvernay	0	1	7	164	2	2	0	0	1	0	1	0	2	1	0	9	0	3	4	3	82.0%
		0.0%	0.0%	0.2%	4.8%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.1%	0.0%	0.3%	0.0%	0.1%	0.1%	0.1%	0.1%	18.0%
	Eagle Ford (Gas)	0	0	0	1	148	3	1	0	0	0	0	2	1	4	0	0	0	0	1	1	91.4%
		0.0%	0.0%	0.0%	0.0%	4.3%	0.1%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	8.6%
	Eagle Ford (Oil)	0	0	0	1	3	138	0	0	0	0	0	4	1	2	0	1	0	3	5	0	87.3%
		0.0%	0.0%	0.0%	0.0%	0.1%	4.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.1%	0.0%	0.1%	0.0%	0.1%	0.1%	0.1%	0.0%	12.7%
	Evie (Horn River)	0	0	0	0	0	1	151	0	1	0	0	0	0	0	4	0	0	1	3	0	93.8%
		0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.4%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.1%	0.0%	0.1%	0.0%	0.1%	6.2%
	Green River	0	0	0	0	0	1	0	152	0	0	0	1	0	0	0	0	0	0	1	0	98.1%
		0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.4%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	1.9%
	Haynesville	0	0	0	0	1	0	0	0	141	1	0	0	0	1	4	3	1	5	0	0	89.8%
		0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	4.1%	0.0%	0.0%	0.0%	0.0%	0.1%	0.1%	0.0%	0.1%	0.0%	0.0%	0.0%	10.2%
	Kimmeridge	0	0	0	0	0	1	0	0	157	1	0	1	0	1	0	1	2	1	0	0	95.7%
		0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.6%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	4.3%
La Luna	0	0	2	0	1	0	0	0	1	0	189	1	0	1	2	0	0	1	3	2	93.1%	
	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	5.5%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.1%	0.1%	0.1%	6.9%	
Leonard	0	1	0	0	0	0	0	0	2	0	0	1	142	3	0	0	2	0	0	0	91.0%	
	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	4.1%	0.1%	0.0%	0.0%	0.0%	0.1%	0.0%	0.1%	0.0%	9.0%	
Marcellus	6	0	0	1	0	2	0	0	0	0	0	1	172	0	0	3	0	1	0	4	90.5%	
	0.2%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	5.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.1%	9.5%	
Niobrara	0	2	0	0	1	3	0	0	0	1	0	0	140	0	0	0	4	3	0	0	90.9%	
	0.0%	0.1%	0.0%	0.0%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.1%	0.0%	0.0%	0.0%	0.1%	0.1%	0.0%	0.0%	9.1%	
Osage	0	0	0	0	0	3	0	1	1	1	0	0	0	186	2	0	1	0	0	0	95.4%	
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	5.4%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	4.6%	
Point Pleasant	0	1	1	12	1	3	0	0	3	0	0	1	2	0	0	165	1	5	1	6	81.7%	
	0.0%	0.0%	0.0%	0.3%	0.0%	0.1%	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%	0.0%	0.0%	4.8%	0.0%	0.1%	0.0%	0.2%	0.1%	18.3%	
Utica (Michigan)	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	162	0	0	0	0	98.8%	
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	4.7%	0.0%	0.0%	0.0%	0.0%	1.2%	
Vaca Muerta (Gas)	0	8	0	1	0	0	0	0	2	0	0	0	1	1	0	7	0	127	6	0	83.0%	
	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.0%	0.0%	3.7%	0.2%	0.0%	17.0%	
Vaca Muerta (Oil)	0	3	1	1	4	3	1	1	0	2	4	2	0	0	0	0	2	0	4	135	1	82.3%
	0.0%	0.1%	0.0%	0.0%	0.1%	0.1%	0.0%	0.0%	0.0%	0.1%	0.1%	0.1%	0.0%	0.0%	0.0%	0.1%	0.0%	0.1%	3.9%	0.0%	17.7%	
Woodford (Cana)	0	0	2	7	2	4	0	0	0	0	0	0	2	0	0	5	0	0	1	115	86.3%	
	0.0%	0.0%	0.1%	0.2%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	4.2%	13.7%	
	96.4%	90.1%	92.1%	82.4%	89.7%	83.6%	96.8%	97.4%	93.4%	97.5%	95.0%	91.0%	91.5%	89.7%	93.5%	82.9%	96.2%	77.0%	81.8%	87.9%	90.3%	
	3.6%	9.9%	7.9%	17.6%	10.3%	16.4%	3.2%	2.6%	6.6%	2.5%	5.0%	9.0%	8.5%	10.3%	6.5%	17.1%	1.8%	23.0%	18.2%	12.1%	9.7%	
	Alum	Barnett	Collingwood	Duvernay	Eagle Ford (Gas)	Eagle Ford (Oil)	Evie (Horn River)	Green River	Haynesville	Kimmeridge	La Luna	Leonard	Marcellus	Niobrara	Osage	Point Pleasant	Utica (Michigan)	Vaca Muerta (Gas)	Vaca Muerta (Oil)	Woodford (Cana)		
	Target Class																					

**Table A.1 – Confusion matrix for the original dataset before image augmentation. The right most column and the bottom row correspond to the percent of images that were correctly classified for each formation. The rows correspond to the predicted class whereas the columns correspond to the target labels.**

		Confusion Matrix																				
Output Class	Alum	757	1	0	3	2	2	0	1	0	0	2	123	0	0	1	0	0	5	84.4%		
		5.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.9%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	15.6%		
	Barnett	0	516	0	0	0	0	0	0	0	0	13	0	4	0	0	0	7	95.6%			
		0.0%	3.8%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	4.4%		
	Collingwood	12	13	656	46	4	8	0	0	3	0	2	1	8	2	0	17	0	9	48	38	75.7%
		0.1%	0.1%	4.8%	0.3%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%	0.0%	0.1%	0.3%	0.3%	24.3%	
	Duvernay	7	13	2	726	23	3	0	0	1	0	0	10	6	0	0	219	2	25	12	24	67.7%
		0.1%	0.1%	0.0%	5.3%	0.2%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	1.6%	0.0%	0.2%	0.1%	0.2%	32.3%	
	Eagle Ford (Gas)	0	0	0	0	556	8	0	0	0	0	0	0	10	8	0	3	0	0	2	0	94.7%
		0.0%	0.0%	0.0%	0.0%	4.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	5.3%
	Eagle Ford (Oil)	0	0	1	1	36	608	0	0	3	0	2	12	3	0	0	27	0	7	13	11	84.0%
		0.0%	0.0%	0.0%	0.0%	0.3%	4.4%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.2%	0.0%	0.1%	0.1%	0.1%	0.1%	16.0%
	Evie (Horn River)	0	34	0	0	0	2	614	1	1	0	1	0	3	14	5	0	12	30	0	85.4%	
		0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	4.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%	0.2%	0.0%	14.6%	
	Green River	1	0	0	0	0	0	2	621	0	0	0	2	0	0	0	0	0	1	0	99.0%	
		0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	1.0%	
	Haynesville	0	0	0	2	1	1	0	1	571	1	1	0	0	5	2	14	4	14	7	2	91.2%
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.2%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.1%	0.1%	0.0%	0.0%	8.8%	
Kimmeridge	0	0	0	0	0	1	0	0	2	618	0	0	1	0	1	1	1	0	0	0	98.9%	
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	1.1%	
La Luna	0	8	0	2	7	4	0	0	1	1	783	2	0	10	3	8	1	17	43	0	88.0%	
	0.0%	0.1%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	5.7%	0.0%	0.0%	0.1%	0.0%	0.1%	0.0%	0.1%	0.3%	0.0%	12.0%	
Leonard	0	0	0	0	0	1	0	0	2	618	0	0	1	0	1	1	1	0	0	0	99.3%	
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.7%	
Marcellus	0	1	0	0	0	0	0	0	0	1	0	0	564	0	0	2	0	0	0	1	99.1%	
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.9%	
Niobrara	0	17	0	1	10	10	0	0	3	0	0	3	0	0	590	0	11	0	29	5	2	86.6%
	0.0%	0.1%	0.0%	0.0%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.3%	0.0%	0.1%	0.0%	0.2%	0.0%	0.0%	13.4%	
Osage	0	2	1	0	0	2	8	0	7	19	3	0	4	1	771	8	6	7	18	0	90.0%	
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	5.6%	0.1%	0.0%	0.1%	0.1%	0.0%	10.0%	
Point Pleasant	0	2	0	0	0	0	0	0	3	2	0	0	7	0	0	438	1	2	0	1	95.6%	
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	3.2%	0.0%	0.0%	0.0%	0.0%	0.0%	4.4%	
Utica (Michigan)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	646	0	0	0	0	99.5%	
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.7%	0.0%	0.0%	0.0%	0.0%	0.5%	
Vaca Muerta (Gas)	0	9	0	0	1	2	0	0	8	1	0	1	0	2	3	17	0	488	5	3	90.4%	
	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	3.5%	0.0%	0.0%	9.6%	
Vaca Muerta (Oil)	0	28	1	4	22	11	0	0	2	0	5	13	4	2	1	9	2	42	480	2	76.4%	
	0.0%	0.2%	0.0%	0.0%	0.2%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%	0.0%	0.3%	3.5%	0.0%	0.0%	23.6%	
Woodford (Cana)	0	2	2	9	1	0	0	0	0	0	0	1	19	0	0	12	0	4	0	572	92.0%	
	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	4.2%	8.0%	
	97.4%	79.9%	98.9%	91.2%	83.9%	91.7%	98.4%	99.5%	94.4%	96.0%	98.4%	90.3%	75.2%	94.1%	96.9%	55.0%	97.4%	73.6%	72.4%	86.3%	88.3%	
	2.6%	20.1%	1.1%	8.8%	16.1%	8.3%	1.6%	0.5%	5.6%	4.0%	1.6%	9.7%	24.8%	5.9%	3.1%	45.0%	2.6%	26.4%	27.6%	13.7%	11.7%	
	Alum	Barnett	Collingwood	Duvernay	Eagle Ford (Gas)	Eagle Ford (Oil)	Evie (Horn River)	Green River	Haynesville	Kimmeridge	La Luna	Leonard	Marcellus	Niobrara	Osage	Point Pleasant	Utica (Michigan)	Vaca Muerta (Gas)	Vaca Muerta (Oil)	Woodford (Cana)		

**Table A.2 – Confusion matrix for model 2. The right most column and the bottom row correspond to the percent of images that were correctly classified for each formation. The rows correspond to the predicted class whereas the columns correspond to the target labels.**

		Confusion Matrix																				
Output Class	Alum	768	3	0	6	5	0	0	0	0	0	0	0	23	0	0	1	1	0	0	0	95.2%
		5.6%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.8%
	Barnett	0	577	0	1	1	1	0	0	0	0	0	26	0	5	0	0	0	15	1	0	92.0%
		0.0%	4.2%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	8.0%
	Collingwood	0	2	618	2	2	0	0	0	0	0	0	0	1	1	0	2	0	1	7	2	96.9%
		0.0%	0.0%	4.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	3.1%
	Duverney	7	11	24	771	15	5	0	0	4	0	0	6	16	1	0	136	2	30	6	9	73.9%
		0.1%	0.1%	0.2%	5.6%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	1.0%	0.0%	0.2%	0.0%	0.1%	26.1%
	Eagle Ford (Gas)	0	0	0	0	565	3	0	0	1	0	0	3	10	0	6	0	0	0	0	1	95.9%
		0.0%	0.0%	0.0%	0.0%	4.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.1%
	Eagle Ford (Oil)	1	1	1	0	44	640	0	0	0	0	10	3	0	0	8	0	5	5	3	0	88.8%
		0.0%	0.0%	0.0%	0.0%	0.3%	4.7%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	11.2%
	Evie (Horn River)	0	30	0	0	3	615	0	4	0	0	4	4	1	7	1	0	7	23	0	0	88.0%
		0.0%	0.2%	0.0%	0.0%	0.0%	4.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.1%	0.0%	0.2%	0.0%	0.0%	12.0%
	Green River	0	0	0	0	0	0	0	624	0	0	0	0	0	0	0	1	1	0	0	0	99.7%
		0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.5%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.3%
	Haynesville	0	0	1	0	0	0	0	554	0	0	0	1	0	4	0	1	0	0	0	0	98.8%
		0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	1.2%
	Kimmeridge	1	0	0	0	0	0	0	1	632	1	1	0	0	3	4	2	1	1	0	0	97.7%
		0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.6%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.3%
La Luna	0	6	8	2	4	0	0	8	2	783	0	0	2	0	2	0	12	27	1	0	91.4%	
	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	5.7%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.2%	0.0%	0.0%	8.6%	
Leonard	0	0	0	0	1	1	0	0	1	0	564	10	0	0	0	0	1	0	0	0	97.6%	
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.1%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.4%	
Marcellus	0	0	0	0	0	0	0	0	0	673	0	0	1	0	0	0	0	0	0	0	99.9%	
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.9%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	
Niobrara	0	0	0	0	0	0	0	0	0	595	0	1	0	5	1	0	0	0	0	0	98.0%	
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.3%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.0%	
Osage	0	3	1	0	0	4	9	0	15	8	9	0	2	1	785	7	2	15	16	1	89.4%	
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	5.7%	0.1%	0.0%	0.1%	0.1%	0.0%	10.6%	
Point Pleasant	0	1	1	1	2	0	0	0	6	0	0	0	6	1	0	599	0	3	0	5	95.8%	
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.4%	0.0%	0.0%	0.0%	0.0%	4.2%	
Utica (Michigan)	0	0	0	1	0	0	0	1	0	0	0	0	0	0	3	655	0	0	0	0	99.2%	
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.8%	0.0%	0.0%	0.0%	0.0%	0.8%	
Vaca Muerta (Gas)	0	5	1	1	0	2	0	3	0	0	0	0	5	0	5	0	528	2	1	0	95.5%	
	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	3.8%	0.0%	0.0%	0.0%	4.5%	
Vaca Muerta (Oil)	0	6	1	1	19	4	0	0	5	0	3	15	2	4	1	11	0	39	573	0	83.8%	
	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.1%	0.0%	0.3%	4.2%	0.0%	0.0%	16.2%	
Woodford (Cana)	9	1	7	9	2	0	0	3	0	0	1	7	0	0	5	0	0	0	640	0	94.9%	
	0.0%	0.0%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	4.7%	0.0%	5.2%	
	98.8%	89.3%	93.2%	96.9%	85.2%	96.5%	98.6%	100%	91.6%	98.1%	98.4%	90.0%	89.7%	94.9%	96.6%	75.3%	96.8%	79.6%	86.4%	96.5%	92.8%	
	1.2%	10.7%	6.8%	3.1%	14.8%	3.5%	1.4%	0.0%	8.4%	1.9%	1.6%	10.0%	10.3%	5.1%	1.4%	24.7%	1.2%	20.4%	13.6%	3.5%	7.2%	
	Alum	Barnett	Collingwood	Duverney	Eagle Ford (Gas)	Eagle Ford (Oil)	Evie (Horn River)	Green River	Haynesville	Kimmeridge	La Luna	Leonard	Marcellus	Niobrara	Osage	Point Pleasant	Utica (Michigan)	Vaca Muerta (Gas)	Vaca Muerta (Oil)	Woodford (Cana)		
	Target Class																					

**Table A.3 – Confusion matrix for model 3. The right most column and the bottom row correspond to the percent of images that were correctly classified for each formation. The rows correspond to the predicted class whereas the columns correspond to the target labels.**