UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

DO WE REALLY NEED DEEP LEARNING? A STUDY ON PLAY INDENTIFICATION USING SEM IMAGES

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE

By

HANYAN ZHANG
Norman, Oklahoma
2021

DO WE REALLY NEED DEEP LEARNING? A STUDY ON PLAY INDENTIFICATION USING SEM IMAGES

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

BY THE COMMITTEE CONSISTING OF

Dr. Deepak Devegowda, Chair

Dr. Mark E. Curtis

Dr. Ali Ousseini Tinni

Dedicated to

My advisors, research partner, friends, and family

Table of Contents

List of Tables

List of Figures

Abstract

Deep learning has become an integral part of image classification and segmentation, especially with the use of convolutional neural networks (CNN) and their variants. Although computationally expensive and time-consuming, there are several promising applications to classify or segment SEM images, images of core and thin sections. But we have not really questioned the need for really deep networks in these applications? Can shallower networks be competitive in relation to deeper networks? Can a shallower network with a wider diversity of convolutional filters (breadth) do better than a deeper network? What image resolution and filter complexity do we need to achieve a high degree of accurate classification?

In this thesis, I assess image classification using over 8000 SEM images acquired from 22 different unconventional plays to answer the questions posed above and provide guidelines to select an optimal depth and breadth for image classification. I evaluate several different CNN architectures systematically by changing the breadth (the number of filters within each layer) or the depth of the network (the number of layers) to relate classification accuracy and the complexity of the CNN. I also test the performance of the different CNN's against different image resolutions to determine if there is a specific field-of-view that is necessary to obtain satisfactory play classification.

For all image resolutions considered, surprisingly, the simplest and shallowest one-layer model performs remarkably well with even 22 different classes (plays) to identify. Despite the simplicity of the network, I achieve over 80% accuracy in play identification (with correspondingly high recall and precision). A moderate increase in depth to 2 layers advances the accuracy to beyond 90%, even with a modest number of filters. Deeper networks that lack filter width perform poorly, indicating the significance of filter diversity in each of the convolutional layers of a CNN. The results from this study show that deeper networks are probably not necessary for image classification of SEM images/core or thin-section images.

The microstructural features within the samples probably necessitate a wider diversity of filters. Finally, although several studies have relied on transfer learning of ‘published’ or open-source CNNs for play identification and image segmentation, this study shows that the level of complexity required is far less, making training more efficient and reducing the likelihood of overfitting.

Chapter 1: Introduction and Literature Review

1.1 Motivation and Problem Statement

US-based shale plays are shown in **Fig. 1.1**, with the Haynesville, Barnett, Marcellus, Fayetteville, and Eagle Ford being a few of the most productive shale formations (Stephenson, 2015). As of 2019, 75% of the natural gas produced in the US was sourced from shales, and it is estimated that by 2050, the vast majority of natural gas will be produced from shales (EIA, 2021). Additionally, as of 2019, 63% of the crude oil produced in the US came from tight oil resources, including shale, sandstone, and carbonate rock formations (EIA, 2020).

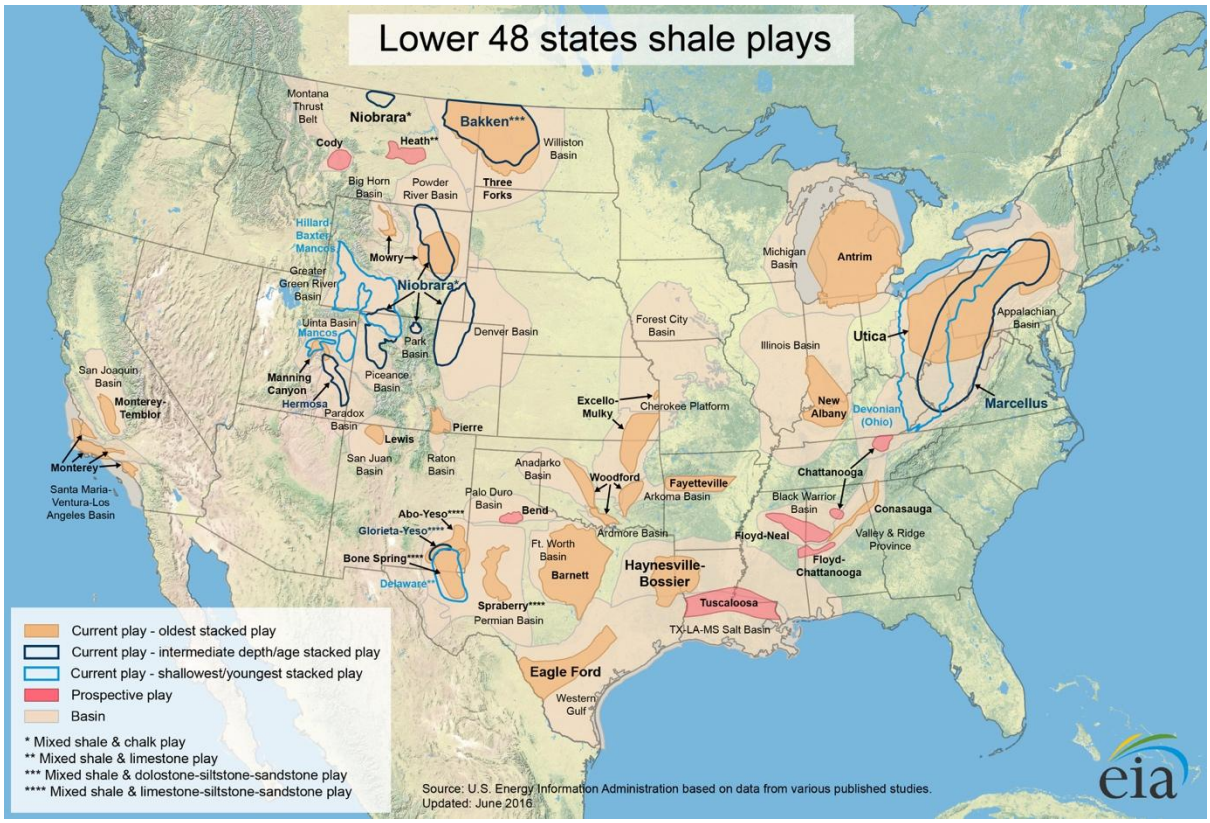


Fig. 1.1 - Shale plays in lower 48 states (EIA, 2016).

In organic-rich shales, hydrocarbons are located in the organic pores, which are thought to be hydrophobic (Curtis et al., 2010). The amount, distribution, and thermal maturity of the

organic matter significantly impact rock properties such as shale geomechanical properties and porosity (Bocangel et al., 2013; Curtis et al., 2014).

There consequently is a need to better understand the microstructure of shales including lithology, pore space, the interconnectivity of pores, grain size, and cementation (Prasad, 2001). Properties such as mineralogy, porosity, and pore size distribution are measured using laboratory-based experiments, including mercury injection capillary pressure (MICP) and nuclear magnetic resonance (NMR) (Prasad, 2001; Bocangel et al., 2013; Curtis et al., 2019; Dang et al., 2019), which can be time-consuming and destructive in the case of MICP (Misbahuddin, 2020).

Computational approaches to determine rock properties have steadily been gaining popularity and are typically referred to as digital rock physics (DRP) (Andrä, 2012). DRP involves several steps beginning with image acquisition using X-Ray CT (Computed tomography), Micro-CT, Focused Ion Beam, and Scanning Electron Microscopy (FIB-SEM) (Curtis et al., 2010; Misbahuddin, 2020). SEM and FIB-SEM are popular for visualizing the nanoscale features in shales (Curtis et al., 2010; Misbahuddin, 2020). In the case of FIB-SEM, as shown in **Fig. 1.2**, focused ion beam milling provides for better sample imaging compared to other sample preparation methods such as hand polishing, broad ion beam (BIB) or broad Ar+ ion beam (Curtis et al., 2010). In the SEM, the electron gun accelerates electrons through a high voltage of several hundred to 40 kV, which are then collated into electron beams using electromagnetic lenses and scanning coils that are rasterized to probe the sample. The resulting signals, which are secondary electrons (SE), backscattered electrons (BSE), or X-rays can be used to image the sample (Curtis et al., 2010).

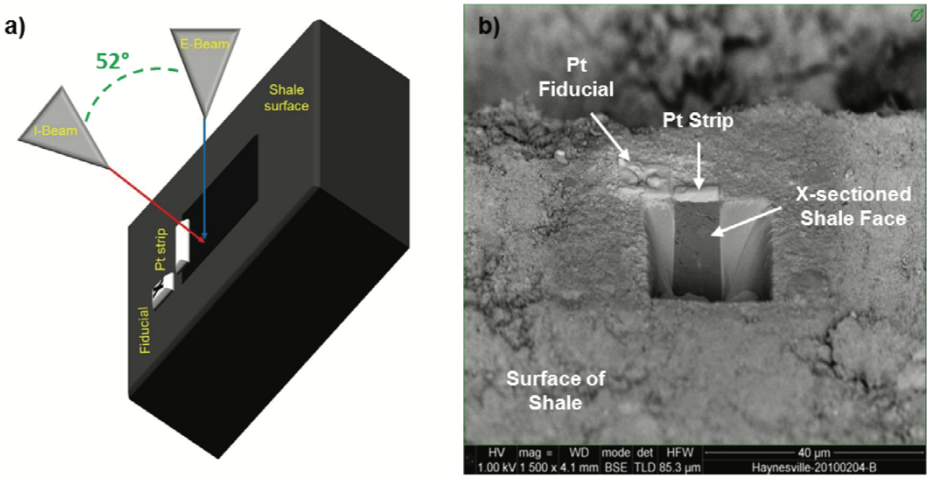


Fig. 1.2 - Imaging the shale microstructure using a dual-beam FIB/SEM system. (a) The ion beam (I-beam) is arranged 52 degrees to the electron beam (e-beam). I-beam can gently mill the surface to form a cross-section that can be used to image microstructure, (b) the BSE image of a cross-sectioned shale taken by the e-beam, which is arranged at a 38° angle to the normal of the I face (Curtis et al. 2010).

Both backscattered electron and secondary electron images are produced by assigning grayscale values to represent signal intensity measurements (Camp et al., 2013). Secondary electrons (SE) are low voltage electrons arising from the inelastic interactions between the primary electron beam and the sample (Nanakoudis, 2019). Secondary electron images are useful to study the topography of the sample surface, including rock fabric, texture, and mineral, pore, and microfossil morphology (Camp et al., 2013, Nanakoudis, 2019). These images also make it possible to evaluate near-surface negative aberrations such as pores and cracks from polished or ion-milled surfaces (Camp et al., 2013). The negative aberrations are represented as low intensity (dark) regions on the image (Camp et al., 2013).

Backscattered electrons (BSE) are electrons returned following scattering from the sample surface (Nanakoudis, 2019) and are sensitive to differences in atomic numbers. This sensitivity to atomic numbers is exploited for imaging. Generally, the higher the atomic number of a specific material, the lighter the color on the image (Curtis et al. 2010, Nanakoudis, 2019) and vice-versa.

X-ray signals are also generated from the interaction of the electron beam and the sample. These can be detected and measured using an energy-dispersive X-ray analyzer (EDX) system (Clelland and Fens, 1991). By detecting the X-ray spectrum, comparing it with a library of phases, and combining it with the BSE and SE signals, we can create false-colored digital phase and texture maps to characterize the mineralogy and lithology of the sample (Curtis et al. 2010, Lemmens et al., 2011, Camp et al., 2013).

It is also possible to construct 3D images of the rock by using an ion beam to gently mill away a 10-nm thick slice of the cross-sectional face. Then, an electron beam can be used to take a new SEM image. By repeating this procedure 300-600 times, the set of contiguous images are used to reconstruct a 3D model, as shown in **Fig. 1.3**.

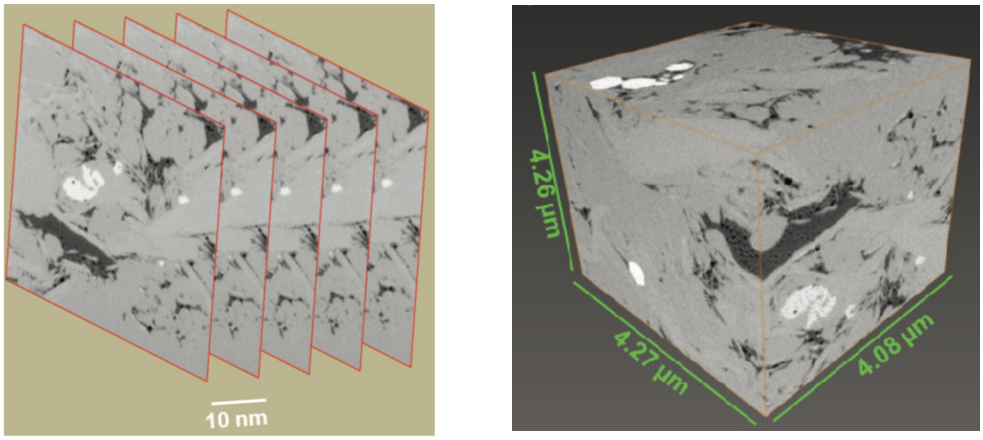


Fig. 1.3 - A set of continuous 2D SEM images showing the microstructure of the shale cross-section can be used to create a 3D model. The left figure is a set of continuous 2D SEM images, the right figure is an example of a 3D model reconstructed from 2D SEM images (Curtis et al., 2010).

The second step of DRP is image processing, including noise reduction, smoothing, and segmentation (Andrä, 2012). Segmentation allows images to be divided into continuous, disjoint, and uniform regions (Wu and Misra, 2019) to describe and locate the various microstructural constituents, as well as kerogen, organic matter, and pores in shale samples (Wu and Misra, 2019). In addition, images can also be used to calculate the porosity of the samples (Misbahuddin, 2020). Traditional manual segmentation by trained personnel has often

been the limiting factor for widespread adoption of DRP given that it is inordinately time-consuming, tedious, error-prone, and subjective, especially when segmenting a wide field-of-view and/or several independent images (Wu and Misra, 2019).

The resulting models can then be used for flow modeling (Deglint et al., 2019) or estimating geomechanical properties (Saad et al., 2018). While digital rock physics has been growing in significance, it remains computationally prohibitive and limited by the physical assumptions of the flow modeling. Nevertheless, huge strides in computational power and its democratization have enabled machine learning to significantly accelerate this process (Xu et al., 2019; Misbahuddin, 2020).

Image classification is another task that has also benefited tremendously from advances in machine learning, specifically in the area of deep learning using convolutional neural networks (CNN) (Le Cun et al., 1999). Image classification, when done in the context of SEM images or core images or thin sections, allows a trained algorithm to recognize the source of a specific image. Successful identification of the source formation indicates microstructural uniqueness. On the other hand, misclassification of one play for another indicates microstructural similarities, which probably also imply petrophysical similarities. These similarities can then be exploited in completion and well design by adopting successful practices from a previous play. My thesis focuses on image classification, but specifically, I address the need for appropriate levels of CNN complexity rather than assuming a one-size-fits-all approach that simply relies on a highly complex, deep network for the classification. Before I discuss my workflow, I provide the reader an introduction to some basic concepts in machine learning so that the results of my work in Chapters 2 and 3 are interpretable.

1.2 Machine Learning

Machine learning (ML) is a branch of artificial intelligence (AI) that enables computers to learn without explicit programming and specifically to analyze patterns in large amounts of data with less human intervention. The data can be very complex, derived from a diverse set of sources, and can take many forms, including but not limited to images, video frames, numbers, and words (Advani, 2020).

Machine learning can be divided into several categories based on the learning method. These are supervised learning, unsupervised learning, and reinforcement learning. I provide a brief overview of each of the methods in the next few sections.

1.3 Supervised Learning

Supervised learning methods are used to derive relationships between a given set of inputs and output and can be divided into two types: classification and regression. Regression deals with continuous output variables, while classification deals with discrete output labels (Nasteski, 2017) as shown in **Fig. 1.4**.

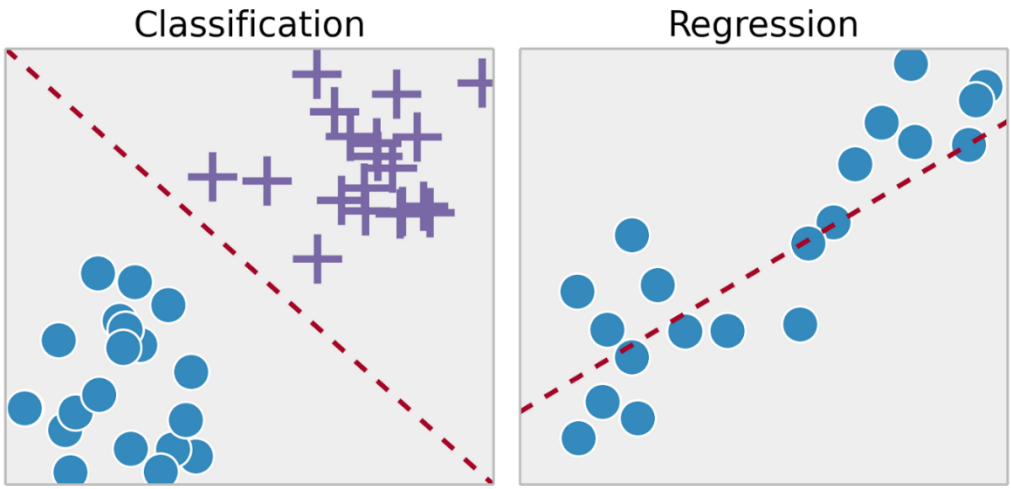


Fig. 1.4 - The difference between regression task and classification task in supervised learning. On the left, the classification version of supervised learning separates the inputs into given categories. On the right, the regression version of supervised learning assigns inputs to continuous numbers (Soni, 2018).

For regression tasks, commonly used algorithms include linear regression, non-parametric regression, support vector machines (SVM), nearest neighbors, Gaussian process regression, decision trees, random forests, and neural networks. For classification tasks, commonly used algorithms include K-nearest neighbor classification, support vector machines (SVM), nearest neighbors, Gaussian process classification, decision trees, random forest, neural network, and convolutional neural network (Scikit-Learn, n.d).

1.4 Unsupervised Learning

Unsupervised learning is another major category of machine learning that attempts to find some natural structure in a dataset (Soni, 2018) as shown in **Fig. 1.5**.

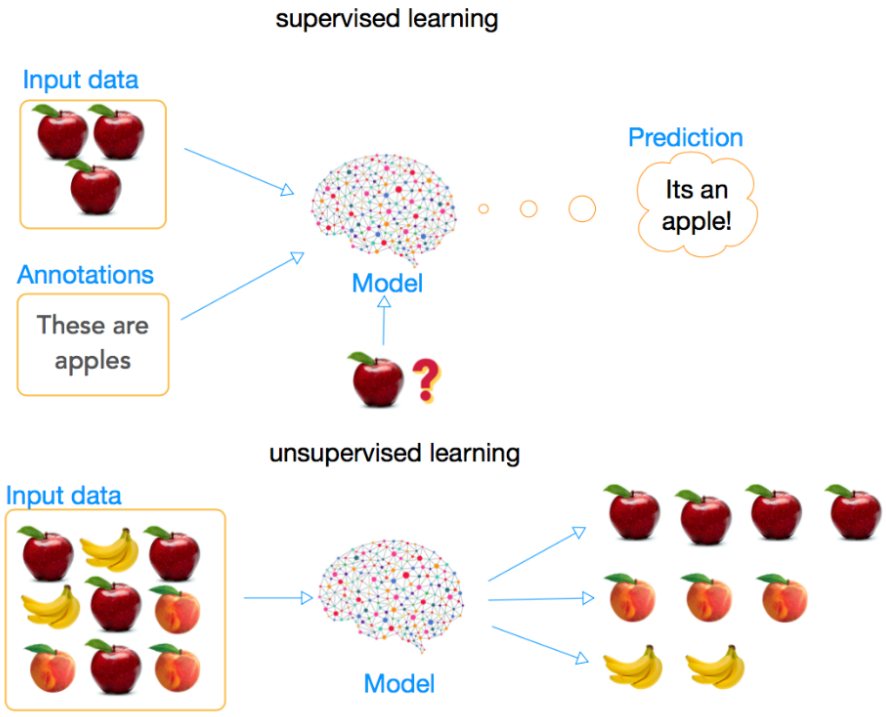


Fig. 1.5 - The difference between supervised learning and unsupervised learning. The figure above is a supervised learning process; it requires input data and output labels to train the model. The following figure is an unsupervised learning process, no output label is required to train the model (Ma, 2018).

The common tasks for unsupervised learning including clustering, representation learning, and density estimation (Soni, 2018). Clustering is often considered the most

significant task and is used for discovering the similarities between data points and grouping data points based on their similarities (**Fig. 1.6**). There is no prescribed grouping standard for clustering. Users can decide the number of clusters required and the degree of similarity between each data point to be grouped.

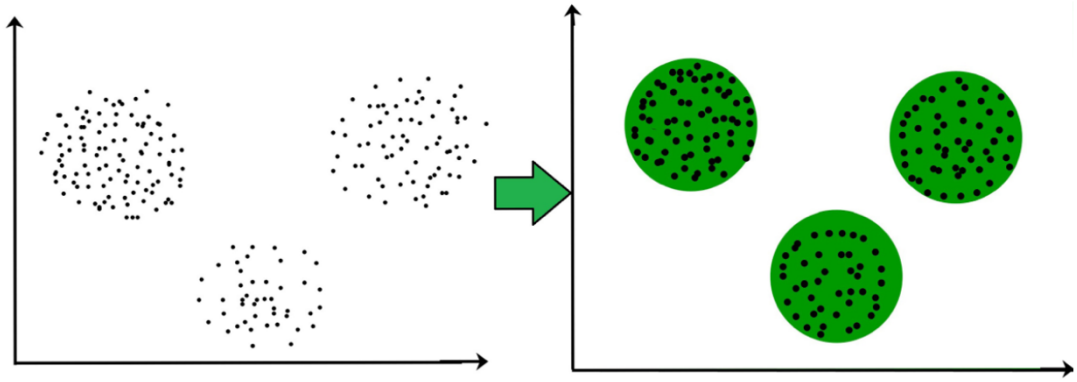


Fig. 1.6 - Clustering tasks group the data points to several group based on the similarities of the data shown in the right figure (Priy, 2020).

Popular cluster algorithms for unsupervised learning are DBScan, K-Means, hierarchical clustering, Gaussian mixture models, and others (Scikit-Learn, n.d). One of the more common uses of clustering is in reservoir characterization to identify rocktypes from core data (Gupta et al., 2018) or electrofacies from well log data (Torghabeh et al., 2014). Because the methods used in this thesis rely on convolutional neural networks, I begin with a brief introduction to neural nets. It is important to note that neural networks can be used in both unsupervised as well as in supervised mode, but for the rest of this thesis, I focus on supervised learning for image classification.

1.5 Neural Network

Neural networks are designed to imitate networks of neurons in the human brain. In general, the level of complexity does not approach that of the human brain with an excess of 100 billion neurons (Gurney et al., 1997).

Neurons in the brain communicate with each other using electrical signals. Every neuron has three parts: dendrites, cell body, and axon as shown in **Fig. 1.7**. Dendrites have a tree-like structure and transport multiple electric signals to the cell body. The cell body processes these signals by using a weighted sum and some form of thresholding, following which the result is transferred to the axon and subsequently passed on to dendrites belong to other cells. The process repeats for the downstream neurons, and the whole collection of neurons is termed a neural network (Hagan et al., 2014).

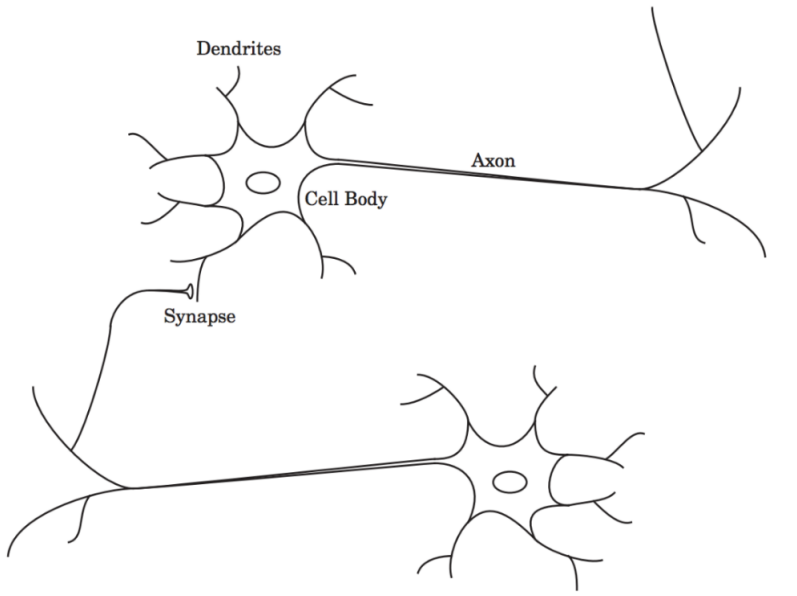


Fig. 1.7 - Structure of two connected neurons. Each neuron has three parts: dendrites, cell body, and axon. Electric signals are input from dendrites, processed in the cell body, output from axons, and send from synapse to the next neuron (Hagan et al., 2014).

The first artificial neuron was introduced in 1943 (McCulloch and Pitts, 1943) that computes the weighted sum of all input signals and compares the weighted sum to a given threshold. If the weighted sum is larger or equal to the given threshold, the output is 1, otherwise, the output is 0. Rosenblatt et al. (1958) introduce a computational version of the artificial neuron with four parts: the input layer, the weight terms for each of the inputs and bias, the summation function, and the activation function as shown in **Fig. 1.8**. During model training, the weights associated with each input are initialized randomly, and after each

iteration, the error between the network prediction and true measurement is computed. This provides a means to adjust the weights to minimize the error in predictions (Salhi, 2020). The perceptron can be considered as the basis of all neural networks.

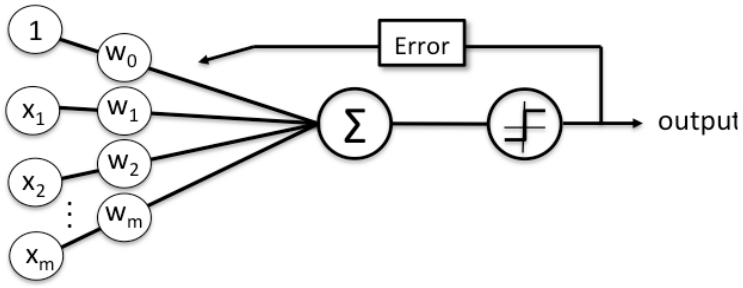


Fig. 1.8 - The architecture of the Perceptron Classifier. 1, X_1 to X_m represent values from the input layer. W_0 to W_m represent the weight carried by each input value. The weighted sum will be computed and passed to an activation function. An error will then be calculated, and the weight carried by each input will be improved to minimize the error. (Salhi, 2020).

With access to more computational power, artificial neural networks (ANNs) have grown to accommodate multiple layers with several hidden layers, each of which is composed of multiple neuron groups. Each neuron in this layer is connected to the neuron in the next layer to form a network as shown in **Fig. 1.9**. While the network is being trained on some data, the weights associated with each connection are tuned.

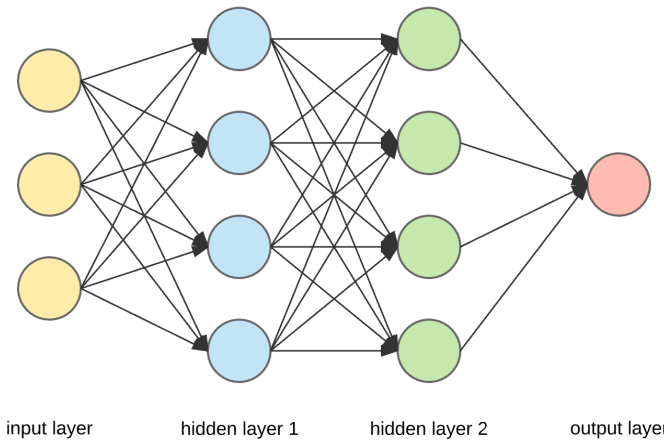


Fig. 1.9 - Modern artificial neural network (ANN) structure. Neurons are divided into several layers. Neurons in each layer are connected to neurons in the next layer (Sorokina, 2017).

1.6 Convolutional Neural Network for Image Classifications

Given that ANN architecture consists of several hidden layers potentially, the number of tunable parameters can be large, requiring access to vast amounts of computational power for model training. This is especially so for image classification, where each pixel serves as an input to the network. The high-resolution color images that are common these days can be associated with millions of input neurons. For example: a 2048 x 4096 image in RGB color will be associated with 2048 x 4096 x 3 channels = 25165824 input neurons. A classical neural network with over 25 million inputs would be computationally prohibitive and unwieldy.

Additionally, if the input were provided to an ANN pixel-by-pixel, the network will be sensitive to the location of a specific object within the image, rather than the larger scale attributes of the object (LeCun et al., 1999).

LeNet-5 is considered to be the first version of a Convolutional Neural Networks (CNN) specifically designed to overcome the problems with image classification using ANNs (LeCun et al., 1999). LeNet-5 can be seen as the beginning of the idea of LeNet-5 comes from the visual system of cats (Hubel and Weisel, 1962). Their study shows that the visual cortex is composed of a series of complex arrangements of cells that are sensitive to small sub-regions of the visual field, called the receptive field. These receptive fields are arranged to cover the entire visual field. The cells are of two types: general cells that respond to specific edge patterns in the receptive field, and complex cells that have larger receptive fields for identifying larger patterns. (Hubel and Weisel, 1962).

LeNet-5 divides an input image into several parts called receptive fields. Filters extract low-level features such as edges or curves in the receptive fields, and then transfer the captured low-level features to the next few layers to capture higher-level features (LeCun et al., 1999). I will discuss filter architecture in a later section.

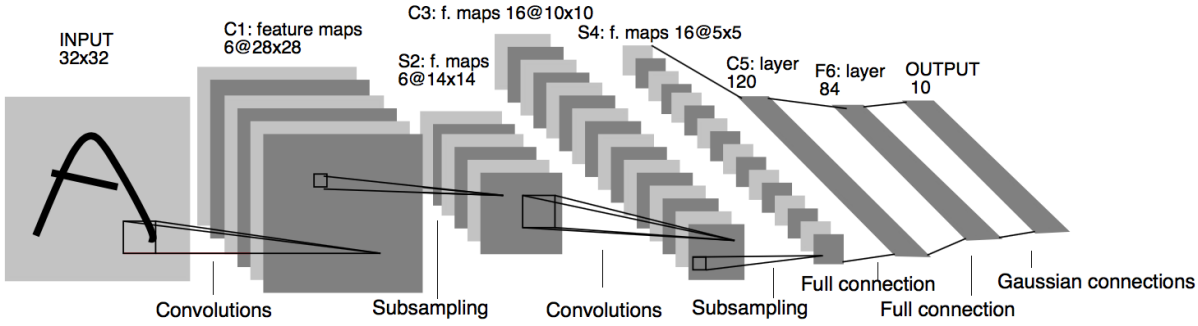


Fig. 1.10 - The architecture of the LeNet-5 proposed by LeCun. It has various types of layers, including convolution layer, subsampling (pooling layer), fully connected layer (LeCun et al., 1999).

Fig. 1.10 shows the architecture of LeNet-5 with an input layer, convolution layer, pooling layer, fully connected layer, and output layer. Each layer has a different purpose discussed in the following sections.

1.6.1 Convolution

An image is composed of a matrix of pixels associated with numbers as shown in **Fig 1.11**.

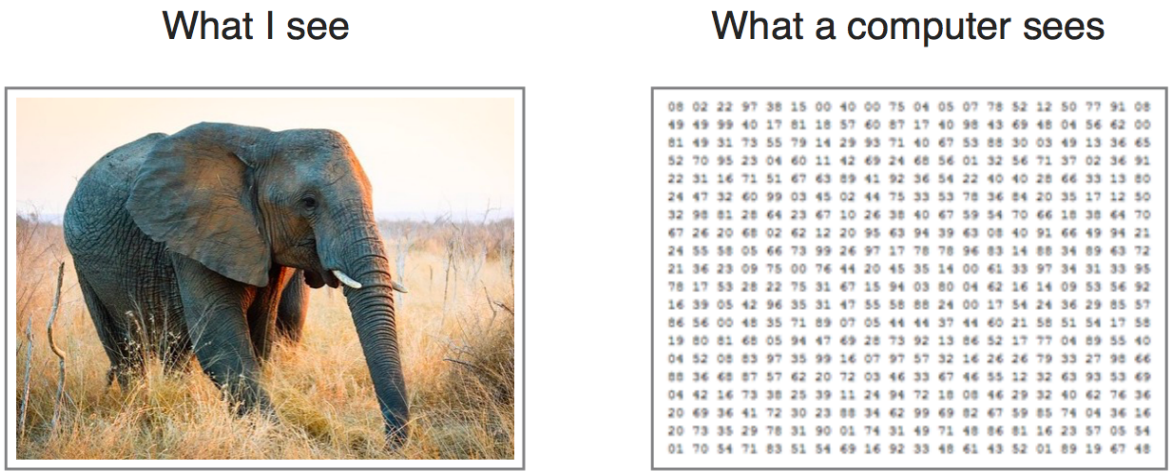


Fig 1.11 - An image composed of colors and objects is nothing but a matrix of pixels (Sorokina, 2017).

In order to recognize objects rather than individual pixels, CNNs use filters. The matrix shown in **Fig. 1.12** is an example of a filter designed to extract right-hand curves in a specific input image.

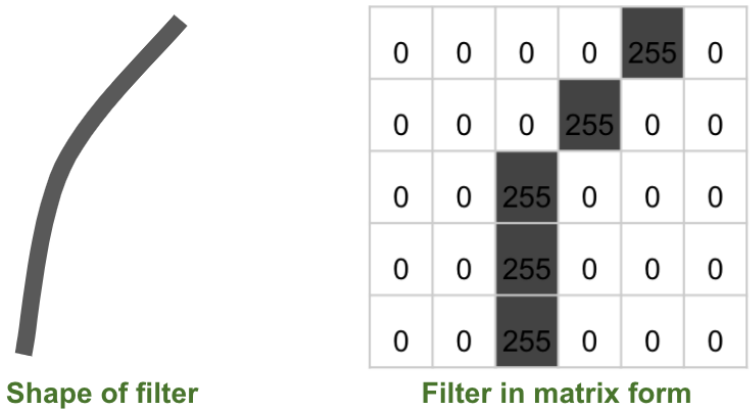


Fig 1.12 - The filter on the right is a matrix designed to extract right hand curves. (Stureborg, 2019).

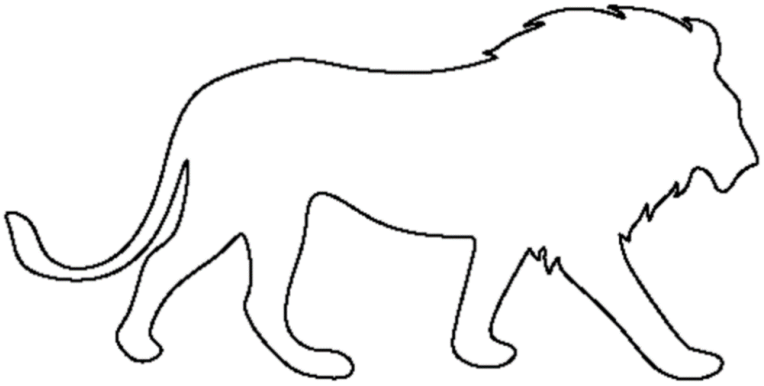


Fig. 1.13 - The input image example contains right-handed curves (Stureborg, 2019)

Let us consider the outline of a lion in **Fig. 1.13.** If the filter shown in **Fig. 1.12** is convolved with the image of the lion, the resulting set of numbers in a matrix form contains information about the location and existence of right-handed curves in the image.

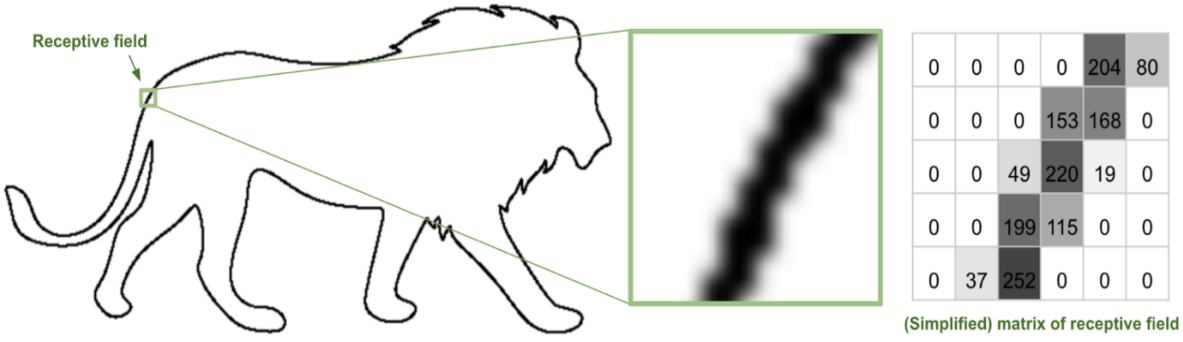


Fig. 1.14 – Digit matrix of the filter area (receptive field) (Stureborg, 2019).

Consider the receptive field in **Fig. 1.14** which is the portion of the image being 'viewed' by the filter. If we chose the filter shown in **Fig. 1.12** and perform an element-by-element multiplication of the filter and the receptive field followed by a summation of the result as shown in **Fig. 1.15**, we obtain a large number indicating the presence of a feature that the filter is designed to detect which in this case, is a right-hand curve. A low value, on the other hand, indicates the absence of a specific feature (Stureborg, 2019). In other words, the higher the resulting value, the filter more closely matches the receptive field.

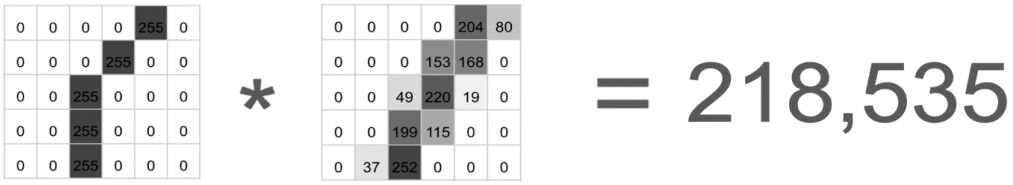


Fig. 1.15 - Element-by-element multiplication of the filter and receptive field results in a large sum, indicating the presence of a right-handed curve in the receptive field (Stureborg, 2019).

The filter is moved to a new location by a pre-specified number of pixels, and the process is repeated. The end result is a new matrix called a feature map, as shown in **Fig. 1.16**. It is referred to as a 'map' because it contains information locating specific features in the image (Stureborg, 2019). This process is more effective than artificial neural networks (LeCun, 1999) and provides an output feature map that is of smaller dimensions than the input image.

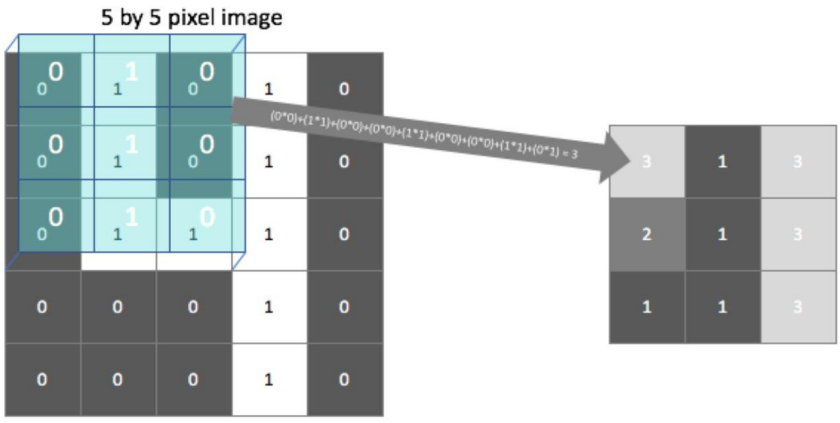


Fig. 1.16 - The left-side matrix represents the input image. The blue 3x3 matrix represents the filter. The right-side matrix represents the feature map (Stureborg, 2019).

The processing of the image through a filter is called convolution. Generally, in one convolution step, several randomly initialized filters are applied to the image to create multiple feature maps. The resulting feature maps provide information on the existence and location of specific features in the image.

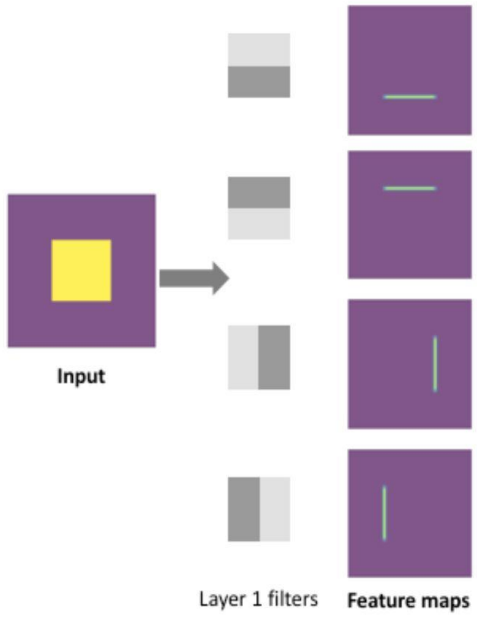


Fig. 1.17 - Visualization of a convolution step with four filters applied to the input image resulting in four feature maps. The unique features captured by each filter are displayed on each feature map (Jordan, 2017).

Another example of convolution is shown in **Fig. 1.17**. Four filters are applied to the original input image. Each filter has a different digit matrix seeking different features (Jordan, 2017), such as horizontal or vertical features of the original image, which is a yellow square. The convolution of each filter with the yellow square shows the location for each of the edges. Processing this information in a subsequent step will allow the CNN to detect the object as a square.

1.6.2 Pooling

The next step is pooling and essentially summarizes the patterns in each feature map (Choulwar, 2019). There are several types of pooling methods, of which the two most common ones are maximum and average pooling. **Fig. 1.18** is an example of maximum and average pooling.

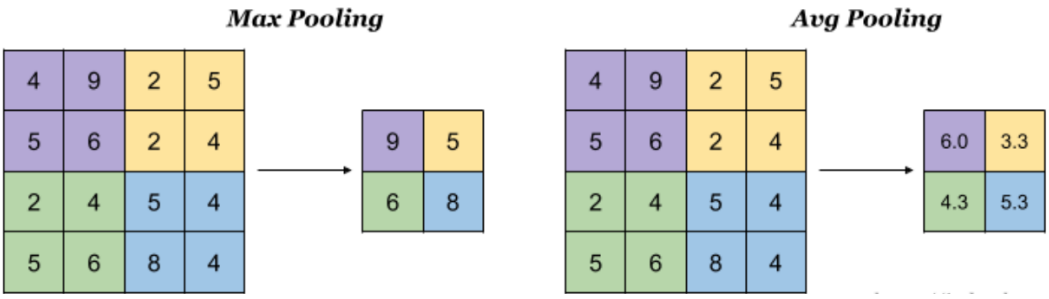


Fig. 1.18 - Max pooling and average pooling of an input feature map (Choulwar, 2019).

The max pooling method extracts the maximum value within a specified region, generally a square, of the feature map (Choulwar, 2019). Using max pooling, a 2x2 window applied to the matrix in **Fig. 1.18** results in a smaller matrix containing the numbers 9, 5, 6, and 8. Average pooling works on a similar principle but instead extracts the average value within a chosen n-by-n window (Choulwar, 2019). The matrix created by pooling methods are also treated like pixels and can have non-integer values. In the right panel of **Fig. 2.18**, the purple square corresponds to a 2x2 matrix of a feature map with an average value of (4+9+5+6)/4 = 6.0. The new feature map then only stores the value of 6.0.

Pooling reduces the dimensions of the feature maps while also retaining significant features and reducing noise (Choulwar, 2019). This has the advantage of reducing computational power requirements with fewer learning parameters. In modern CNN architecture, it is very common to have a pooling layer after each convolution layer. Among various pooling methods, the maximum pooling method is the most commonly used (Brownlee, 2019). The purpose is to amplify the most important features.

1.6.3 Depth

Depth refers to the number of layers of the CNN (Simonyan and Zisserman, 2014). The types of layers include convolutional layers, pooling layers, and fully connected layers. A deeper CNN is associated with many more layers. **Table. 1.1** is a summary of the well-known convolutional neural network architectures over the years showing rapid increases in the depth of CNNs used for image classification. (Chollet et al., 2017).

Architecture Name	Year	Total Number of Layers	Number of Convolutional Layers	Top-5 error on ImageNet
LeNet	1998	5	2	NA
AlexNet	2012	8	5	17.0 (Krizhevsky et al., 2012)
VGG-16	2014	16	13	9.3 (Simonyan and Zisserman, 2014; Fu and Rui, 2017)
GoogLeNet	2014	22	21	9.0 (Szegedy et al., 2014; Fu and Rui, 2017)
ResNet-50	2015	50	49	6.7 (He et al., 2015; Fu and Rui, 2017)
ResNet-152	2015	152	151	3.6 (He et al., 2015; Fu and Rui, 2017)
Xception	2017	126	32	5.5 (Chollet et al., 2017)

Table. 1.1 - Convolutional neural networks are getting deeper and deeper, stabilizing above 100 layers. The top-5 error is calculated by the proportion of the test images for which the correct label ranks top 5 of all labels (Fu and Rui, 2017)

As shown in **Table. 1.1**, LeNet-5 has 5 layers in total, 2 of which are convolutional layers, and the rest are pooling layers and fully connected layers (LeCun, 1999). In 2012, AlexNet (Krizhevsky et al., 2012) won the ImageNet image classification competition far surpassing second place and was the first implementation of a CNN architecture to a large-

scale image dataset. It has 8 layers in total, 5 of which are convolutional layers, and the rest are pooling layers, fully connected layers, and dropout layers. It uses multiple GPUs for model training, which greatly improves computational power.

The emergence of AlexNet underscored the effectiveness of CNNs for image classification. VGG-16 was launched in 2014, with a total of 16 layers, 13 of which are convolutional layers (Simonyan and Zisserman, 2014; Fu and Rui, 2017). Simultaneously, GoogLeNet was introduced with 22 layers, 21 of which are convolutional layers (Szegedy et al., 2014; Fu and Rui, 2017). It is clear to see that VGG-16 and GoogLeNet are deeper CNNs compared to AlexNet. Subsequently, we have seen two iterations of ResNet (He et al., 2015; Fu and Rui, 2017), one with 50 layers and the second version with 152 layers. Since then, the structure of the convolutional neural network now routinely exceeds 100 layers.

While it may be possible to achieve higher accuracies with deeper CNNs, this is not always the case. He et al. (2015) report that simply increasing the number of layers in ResNet can actually compromise test error because the network becomes more difficult to train, as shown in **Fig. 1.19**.

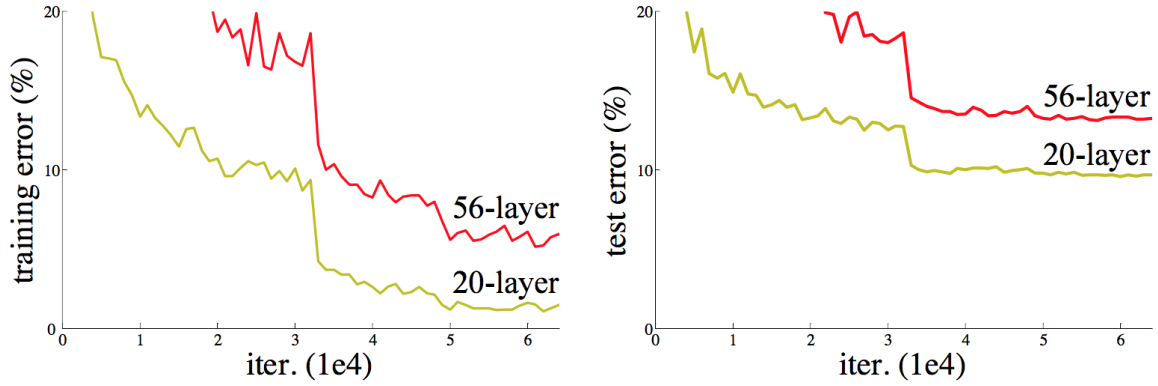


Fig. 1.19 - This figure shows the performance of two convolutional neural networks that have similar structures. The first network (green line) has 20 layers. The second network (red line) has 56 layers. Compared with the 56-layer convolutional neural network, the 20-layer network has lower training and testing errors, which shows that the accuracy will not necessarily improve as the model gets deeper (He et al., 2015)

As the depth of the CNN increases, training and the associated computational cost become more expensive. These requirements are also exacerbated by the need to provide more training data, and in general, the depth of the network should be dictated by the availability of training data (Brownlee, 2018).

1.6.4 Breadth

The number of filters in each convolutional layer within a CNN is called the breadth of the network, sometimes also referred to as the width (Krizhevsky et al., 2012; Garg et al., 2019). A diversity of filters enables the network to extract more features embedded in the image, hopefully aiding classification. AlexNet (Krizhevsky et al., 2012) is an example of exploiting filter breadth using several filters. Krizhevsky et al. (2012) implemented a total of 96 filters in the first convolutional layer. Cross-GPU parallelization speed up the training process by training half of the filters on the first GPU and training the other half on the second GPU (Khandelwal, 2020). **Fig. 1.20** shows 96 filters in the first convolutional layer of the AlexNet model trained on ImageNet. The various shadings associated with each of the filters correspond to some specific shape/feature to be identified by each of the filters.

Fig. 1.20 shows a few grayscale filters while a few are associated with colored patterns. However, there are a few filters with similar shapes and textures, potentially indicating their similarity in terms of feature extraction (Garg et al., 2019).

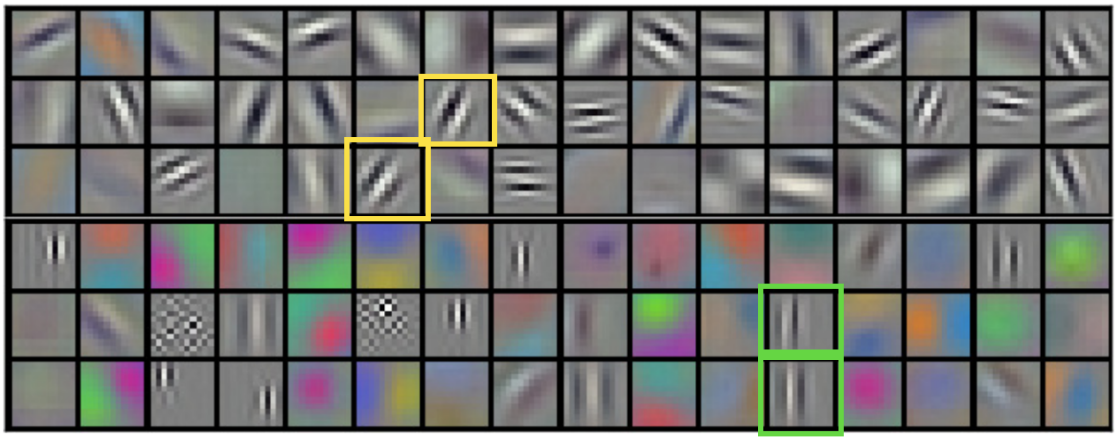


Fig. 1.20 - 96 filters learned by the first convolutional layer of AlexNet trained on the ImageNet dataset. Cross-GPUs parallelization was used for model training. The top 48 filters are learned by the first GPU, the bottom 48 filters are learned by the second GPU (Krizhevsky et al., 2012). The yellow and green boxes show filters that are extracting specific patterns. However, the boxes also show the occurrence of duplicate filters, indicating that the number of filters in the first convolutional layer can be reduced

Repetitive use of the same filter within a layer may increase the number of training parameters but may not contribute to increases in accuracy, necessitating optimization of the network width (Garg et al., 2019). As shown in **Table. 1.2**, although the depth of subsequent versions of CNNs continually increases, the first layer is associated with fewer filters.

Architecture Name	Year	Total Number of Layers	Number of Filters in the First Convolutional Layer
LeNet-5	1998	5	6
AlexNet	2012	8	96
VGG-16	2014	16	64
GoogLeNet	2014	22	64
ResNet-50	2015	50	64
ResNet-152	2015	152	64
Xception	2017	126	32

Table. 1.2 - Comparison of the recent architecture of convolutional neural networks (Fu and Rui, 2017)

In 2014, the VGG-16 network trained on the same dataset as AlexNet reduced the number of filters in the first convolutional layer from 96 to 64. GoogLeNet in 2014 also used 64 filters instead of 96 filters (Fu and Rui, 2017). A year later, ResNet-50 and ResNet-152 also only employed 64 filters on the first convolutional layer (He et al., 2015; Fu and Rui, 2017). Xception in 2017 further reduced the number of filters in the first convolutional layer to 32 (Chollet, 2017). There clearly is a relationship between depth and breadth of the network, and this thesis focuses on gaining some insights into this relationship when applied to SEM image classification.

1.6.5 Activation Functions

As mentioned in **Fig 1.8**, before the results are fed to an activation function, the artificial neuron calculates the weighted sum of the inputs and the bias. The simplified equation (Moawad, 2019) of the artificial neuron without the activation function is:

$$\text{Output} = \sum(\text{weight} * \text{input}) + \text{bias} \quad \ldots\ldots\ldots\ldots\ldots (1)$$

By using a linear function, the model will be capable of demarcating linearly separable classes or labels as seen in **Fig. 1.21**. However, linearly separable classes are the exception and not the norm, with examples such as in **Fig. 1.22** being more prevalent.

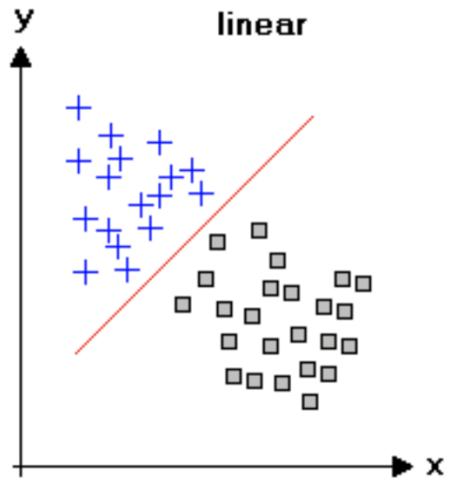


Fig. 1.21 - A dataset successfully separated by using a linear function y = ax+b. (Kanani, 2019)

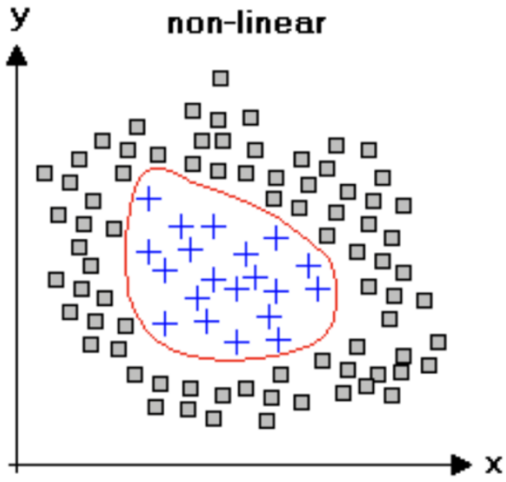


Fig. 1.22 - A non-linear function is required to separate the Non-linearly separable classes (Kanani, 2019)

A curvilinear decision boundary such as in **Fig. 1.22** can be approximated by successive linear boundaries, which would increase model complexity substantially. A more practical option is to use nonlinear functions (Kanani, 2019). Activation functions convert a linear function into a nonlinear function, and there are several variants of activation functions as shown in **Fig. 1.23**.

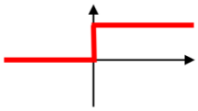
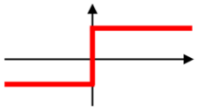
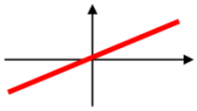
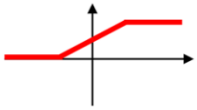
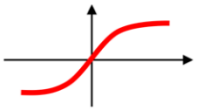
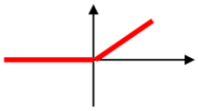
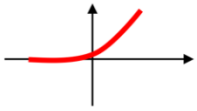
Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	Multi-layer Neural Networks	
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z) = max(0, z)$	Multi-layer Neural Networks	
Rectifier, softplus	$\phi(z) = \ln(1 + e^{z})$	Multi-layer Neural Networks	

Fig. 1.23 - Popular activation functions and associated graphs that can convert a linear function to non-linear (Moawad, 2019).

Popular activation functions include the sigmoid and ReLU (Rectified Linear Units) functions. While the sigmoid function is commonly used in regression tasks (Yang, 2020), the ReLU is more common for classification using CNNs (Kanani, 2019) and has been shown to be better than the sigmoid for classification tasks (Bhumbra, 2018). The equation of the ReLU function shown in **Fig 1.23** is:

$$f(z) = \max(0, z) \quad \ldots\ldots\ldots\ldots\ldots (2)$$

The ReLU function is much simpler and minimizes the effect of vanishing gradients while tuning weights in a network during training where vanishing gradients are a problem (Bhumbra, 2018).

1.6.6 Batch Size, Iteration, and Epoch

Batch size selection is an important aspect of CNN training (Radiuk, 2017). If the dataset is relatively small, the entire dataset can be used in one batch to train the CNN. However, in most cases, with large datasets, memory limitations constrain the number of images that the CNN can process at once. The dataset is therefore split into batches that are fed sequentially, with model parameters updated after each batch, otherwise known as an iteration. Each batch is created randomly from the entire dataset.

Although the input images in each batch are randomly selected to minimize bias, the size of the batch can still influence model performance. If the batch size is too small, each batch may not have a good representation of the entire dataset, which may lead to over-fitting. An example is shown in **Fig. 1.24**, where the training curve with a batch size of 16 fluctuates greatly with poor model performance while a batch size of 1024 results in higher accuracy (Radiuk, 2017). On the other hand, if the batch size is too large, training consumes substantial memory resources, and with a larger number of images to process, model training becomes slower. **Table. 1.3** shows the time taken to train a model with a batch size of 1024 compared to a smaller batch size (Radiuk, 2017). Overall, the batch size greatly affects training efficiency and accuracy and needs to be optimally selected.

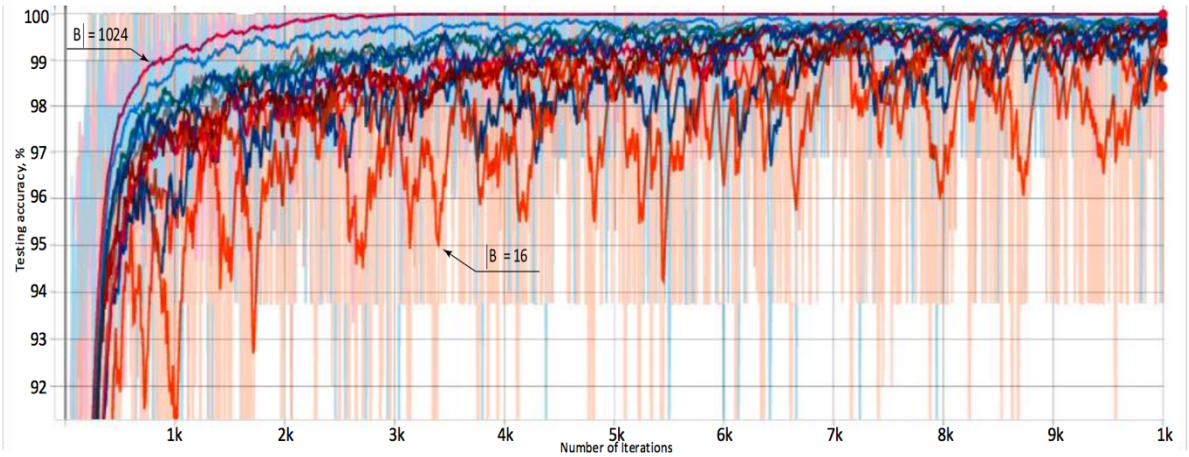


Fig. 1.24 - The x-axis is the number of iterations, and the y-axis is the accuracy. The red line represents the accuracy of model training with a batch size of 16. The plum red line represents the accuracy of model training with a batch size of 1024. The red line with a batch size of 16 shows that the training curve fluctuates, and the overall model performance is poor (Radiuk, 2017).

THE TRAINING TIME EFFICIENCY

$\|B\|$	Time efficiency, h		$\|B\|$	Time efficiency, h	
	MNIST	CIFAR-10		MNIST	CIFAR-10
16	0.28	3.52	150	1.82	7.29
32	0.45	2.48	200	2.25	9.57
50	0.65	3.18	250	3.31	11.80
64	0.93	4.00	256	2.88	12.68
100	1.13	6.35	512	9.35	17.82
128	1.63	6.50	1024	14.23	27.47

Table. 1.3 – Training time of models with different batch sizes. |B| represents the batch size, the MNIST, and the CIFAR-10 are two image data sets for model training. Training with a batch size of 1024 takes 14.23 hours and 27.47 hours, respectively on two datasets. In comparison with smaller batch sizes, this is a much longer training time (Radiuk, 2017).

One epoch is complete when the CNN views the entire dataset (Sharma, 2017). In the training process, the entire dataset is fed several times to the network over several epochs. The batches are re-selected randomly for each epoch. A rule of thumb is to terminate the training process after test accuracy plateaus.

1.6.7 Accuracy and Loss

Accuracy and loss function are two methods for evaluating the performance of convolutional neural networks. For image classification tasks, accuracy refers to the percentage of images correctly classified. The loss function is more complex and is of several types, such as binary cross-entropy, categorical cross-entropy, and sparse categorical cross-entropy (Keras, n.d). Binary cross-entropy is only used when classifying two categories. If the dataset has multiple categories, categorical cross-entropy is typically used, assuming that the labels are in a one-hot format. This work uses categorical cross-entropy for labeling multiple formations. The equation of categorical cross-entropy is:

$$\mathrm{CCE} = -\frac{1}{N}\sum_{i=1}^{N}\log\left(p_i[y_i]\right) \quad \text{..... (3) (Zurück, 2020)}$$

In the equation, CCE represents categorical cross-entropy. N represents the number of categories. y represents a binary indicator, p represents the predicted probability of observation in a category (Zurück, 2020).

By using this equation, the confidence level of the prediction made by the network can be calculated. A low loss means that the prediction is correct, and the network is confident in the prediction. High loss may mean that even if the prediction is correct, the network has low confidence in the prediction. The CNN calculates the loss after each epoch and adjusts its parameters to reduce the loss in the next epoch.

1.6.8 Confusion Matrix

Confusion matrices provide another metric to evaluate model performance for several categorical labels. **Fig. 1.25** is an example of a confusion matrix for a two-category dataset. The 'actual value' at the left of the matrix represents the true labels ("negative" and "positive"). The 'predicted value' on the top of the matrix represents the label predicted by the model. If the

model predicts a "positive" label as "positive", the prediction is correct and a true positive. If the model predicts that a "negative" label is "negative", the result is a true negative (Radecic, 2019).

On the contrary, if the "positive" label is predicted as "negative" by the model, we obtain a false negative. If the "negative" label is predicted as "positive" by the model, it is denoted as a false positive (Radecic, 2019).

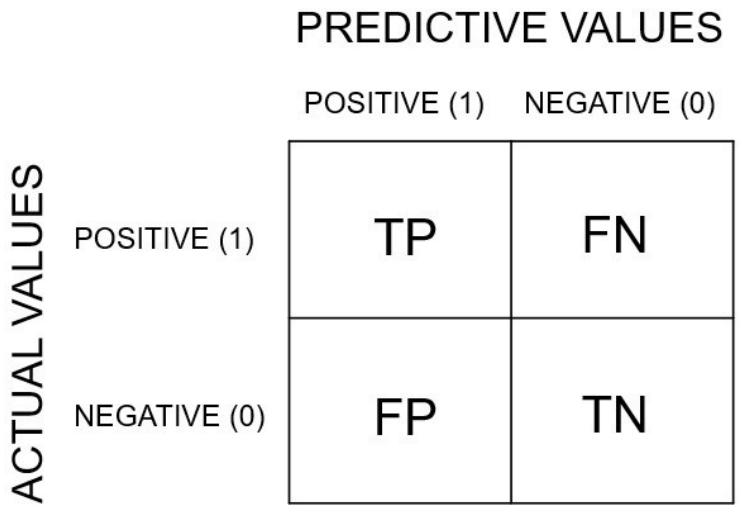


Fig. 1.25 - Confusion matrix of a model that has two outcomes 'negative' and 'positive'. True negative (TN) means the "negative" label is predicted as "negative". True positive (TP) means the "positive" label is predicted as "positive." False-positive (FP) means the "negative" label is predicted as "positive." False-negative (FN) means the "positive" label is predicted as "negative." (Radecic, 2019)

The accuracy can also be calculated by using the information on the confusion matrix. The formula is as follows:

$$\text{Accuracy} = \frac{\text{TP+TN}}{\text{TP+FP+FN+TN}} \ldots\ldots\ldots\ldots\ldots (4)$$

The formula shows that the accuracy is equal to the sum of true positive and true negative divided by the number of all predicted labels (the sum of true positive, true negative, false positive, and false negative) (Radecic, 2019).

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (5)$$

Recall refers to the percentage of successfully detected cases, which can be calculated using the number of the true positive cases divided by the sum of true positive and false negative cases shown in the above formula (Radecic, 2019).

The confusion matrix can also be used to evaluate the performance of models with multiple categories as shown in **Fig. 1.26**.

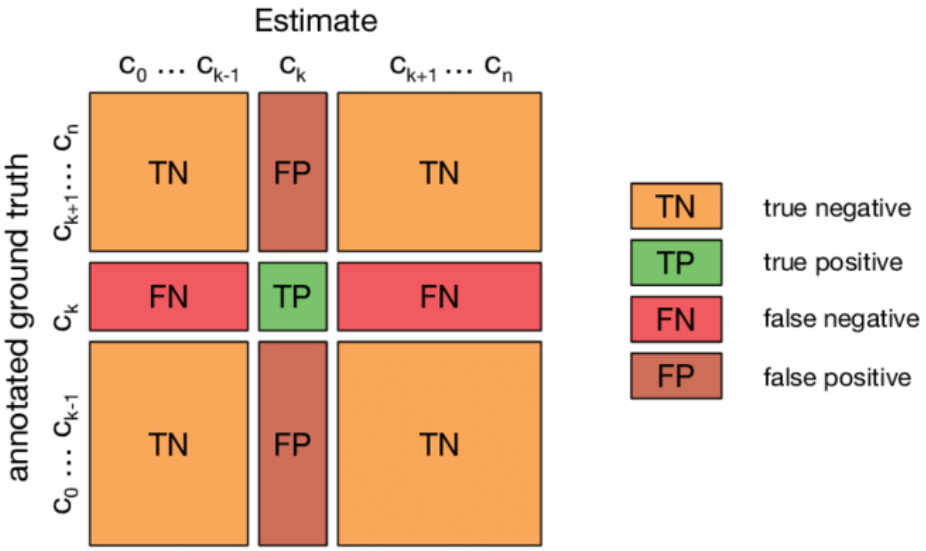


Fig. 1.26 - Confusion matrix with multiple classes (Krüger, 2016).

The x-axis corresponds to the predicted labels, and the y-axis corresponds to the true labels. If the model predicts category C_k as C_k, it is a true positive, and the off-diagonal elements refer to incorrectly predicted class labels.

1.7 Convolutional Neural Network Visualization

This section provides insights into the inner workings of a CNN in terms of visualization of convolution filters, feature maps, and heatmaps.

1.7.1 Convolutional Filter

I discussed convolution earlier, but in this section, I will be showing convolutional filters used for object detection. Filters can be visualized as shown in **Fig 1.27**, for each of the convolutional layers. (Yosinski et al., 2015).

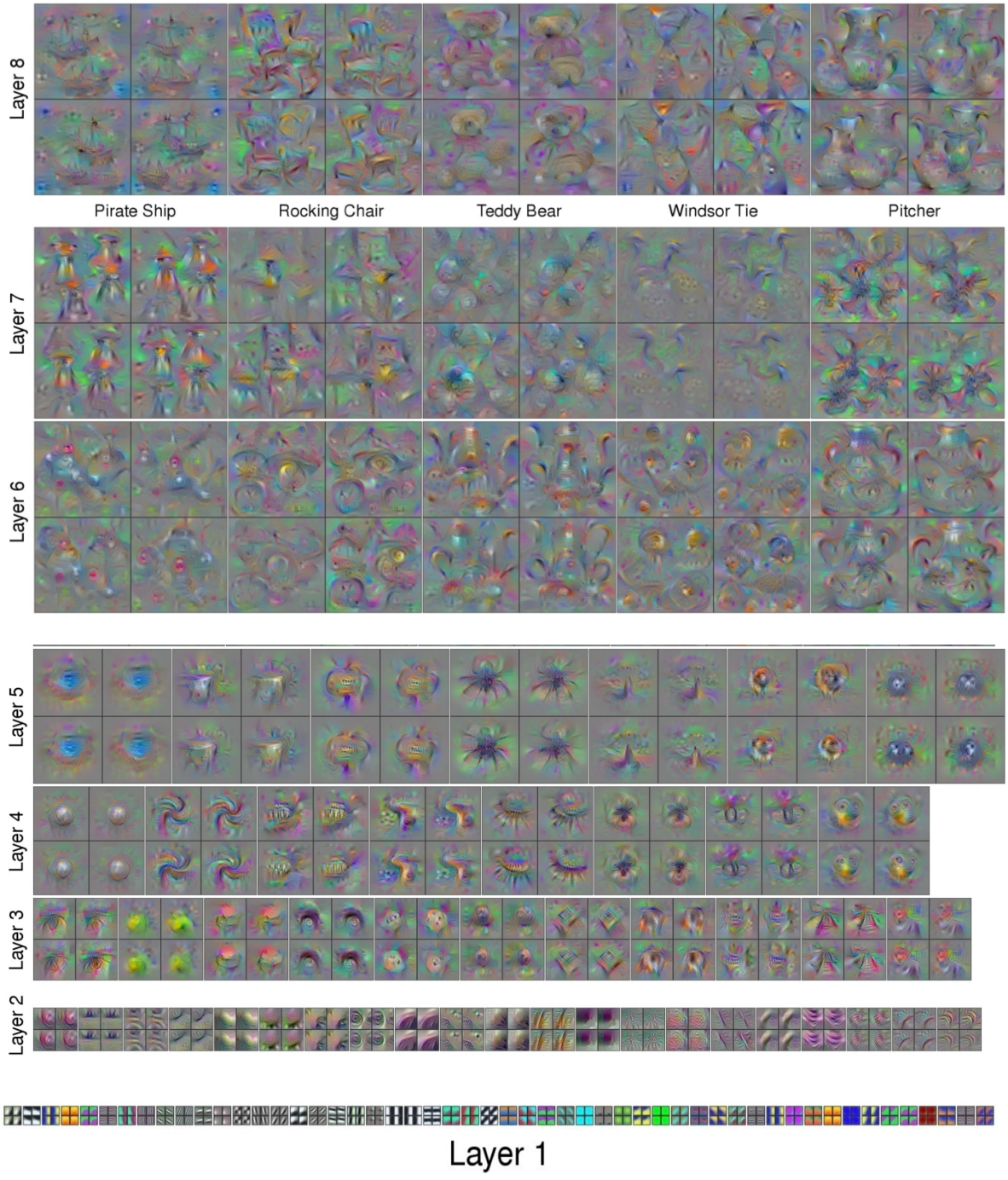


Fig. 1.27 - Example of convolutional filters in each convolutional layer of a trained 8-layer convolutional neural network. Shallow layers can capture small-scale features such as edges, while deeper layers can capture large-scale features (Yosinski, 2015).

It is quite apparent that in Layer 8, we can begin to see the outlines of the original image which will aid in image classification. In preceding filters in Layers 6 and 7, we can observe a few large-scale features, but these are not necessarily interpretable by humans. In even earlier layers, the filters are simply comprised of curves or edges, which are considered to be low-level features.

1.7.2 Feature Map

A feature map is an output obtained after convolution with a filter. Example feature maps extracted from different layers of a training network are shown in **Fig. 1.29**, **Fig. 1.30**, and **Fig. 1.31** when processed on the image in **Fig. 1.28**.

Fig. 1.28 - The image input into a trained 16-layer convolutional neural network (Brownlee, 2019).

Fig. 1.29 - Example feature maps output from the first convolutional layer of a trained 16-layer convolutional neural network (Brownlee, 2019).

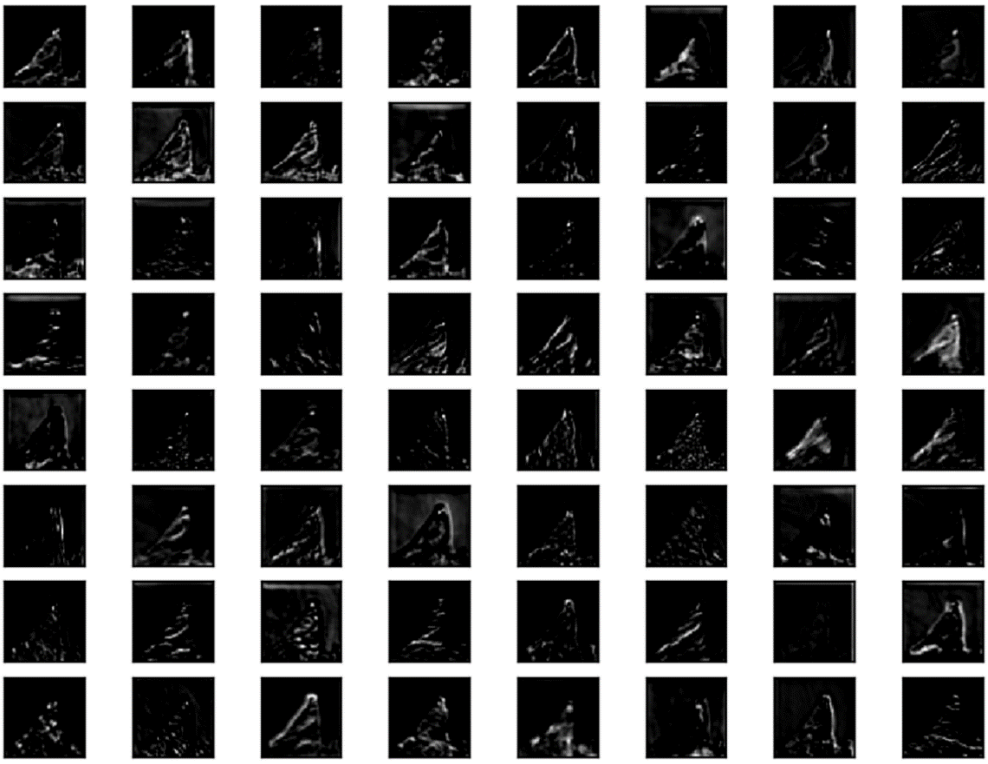


Fig. 1.30 - Example feature maps output from the third convolutional layer of a trained 16-layer convolutional neural network (Brownlee, 2019).

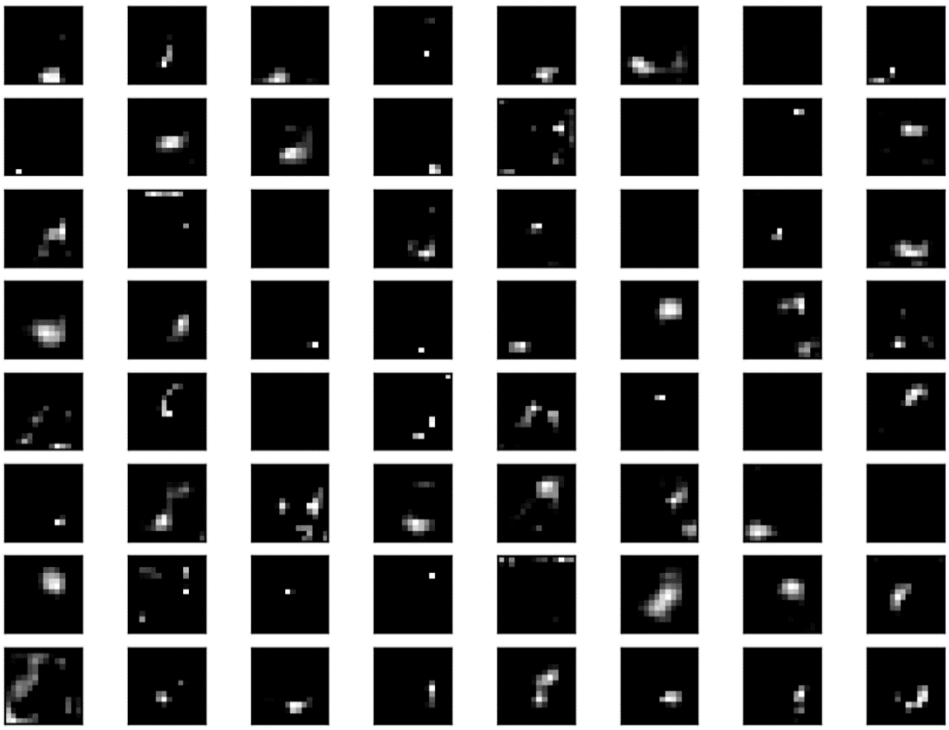


Fig. 1.31 - Example feature maps output from the fifth convolutional layer of a trained 16-layer convolutional neural network (Brownlee, 2019).

In shallower layers, such as in the first (**Fig 1.29**) and third (**Fig. 1.30**) layer, low-level features such as shades and edges are being extracted by filters in the corresponding layers. The feature maps extracted from Layer 5 as shown in **Fig. 1.31**, on the other hand, are more abstract and non-intuitive to interpret but have been shown to contain higher-level information such as the location of actual objects (Brownlee, 2019).

1.7.3 Heatmap

Heatmaps are generated during the processing of an image and are a composite rendering of the critical features chosen by the network to aid in classification. The method I use to generate heatmaps from my trained network is the Gradient-weighted Class Activation Mapping (Grad-CAM), which does not necessitate rebuilding the network architecture or re-training the network (Selvaraju et al., 2019). To output Grad-CAM heatmaps for classification

tasks, the gradient of a category is taken relative to the last convolutional layer to produce a rough weighted location map with essential features highlighted (Selvaraju et al., 2019). The workflow of output a Grad-CAM heatmap is shown in **Fig.1.32**

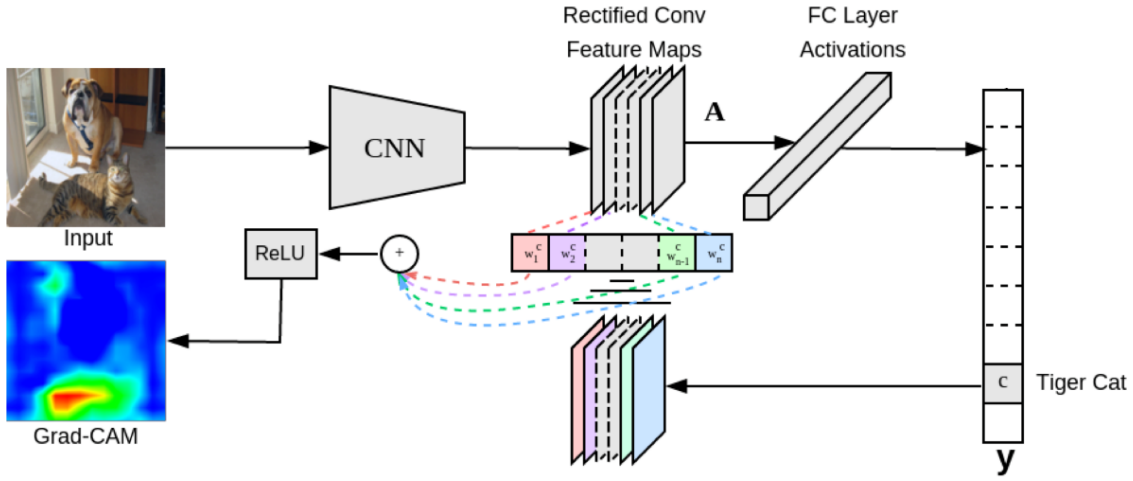


Fig. 1.32 - Workflow to output a Grad-CAM heatmap (Selvaraju et al, 2016).

As shown in **Fig. 1.32**, an image consisting of a cat and a dog input to a trained CNN gives a prediction as a tiger cat. The prediction and the original image will then be forward propagated through the trained network to get the raw prediction score before the softmax function. After that, all the signals will be backpropagated to get the feature maps in the target convolutional layer with the gradient of all classes set to 0 except the 'tiger cat' class set to 1 (Selvaraju et al., 2016). Each feature map will then be weighted and combined into a single location map with essential features highlighted, referred to as a 'heatmap' (Selvaraju et al., 2016). Two examples of the heatmap are shown in **Fig. 1.33.**

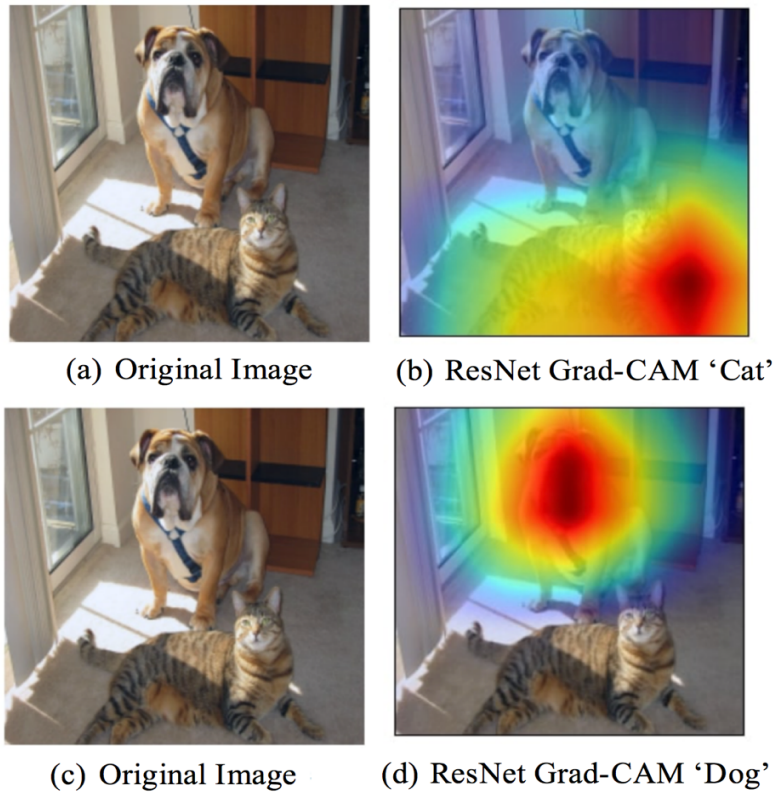


Fig. 1.33 - (a, c) The original image with a dog and a cat input to the pre-trained ResNet . (b, d) Grad-CAM heatmaps of the input image. (b) The pre-trained ResNet recognizes the cat, and the corresponding heat map highlights the cat. (d) The pre-trained ResNet recognizes the dog, and the corresponding heat map highlights the dog (Selvaraju, 2019).

As shown in **Fig. 1.33**, the heatmaps are colored from blue to red, corresponding to the importance of each feature. The hotter colors (red and yellow) reveal the most critical features of a given category, while the colder colors (blue) indicate less essential features (Selvaraju, 2019). An image shown in **Fig. 1.33(a)** and **Fig. 1.33(c)** is input to the trained network, and the network recognizes the cat and the important features necessary for the identification are shown in **Fig. 1.33(b)**, while the dog in the image is regarded as the less crucial feature and covered in blue. In contrast, while identifying the dog in **Fig. 1.33(d)**, features related to the dog are found to be more important.

This thesis focuses on image classification, so although CNNs are well-suited for image segmentation (Ronnenberger, 2015; Badrinarayanan, 2016), that specific application is outside

the scope of this work, and there is a limited discussion of image segmentation beyond the literature review in the next section.

1.8 Application of Convolutional Neural Network in Petrographic Data

Deep learning, especially convolutional neural networks (CNN) and its variants, have shown great promise for image classification, segmentation, identification, and evaluation tasks (Krizhevsky et al., 2012; Ronneberger et al., 2015). Their applicability has been extended to the study of scanning electron microscopy (SEM) images, core images, and thin sections over the past few years (Wang et al., 2014; Xu et al., 2019).

Knaup (2019) used CNNs to classify SEM images that belong to 18 unconventional formations using transfer learning (Pan and Yang, 2009) of an Alexnet CNN (Krizhevsky et al., 2012). Her dataset consists of over 28000 SEM images. A deeper 22-layer model, Inception-3 (Szegedy et al., 2015), was also tested but was seen to be susceptible to severe overfitting (Knaup, 2019). **Fig. 1.34** is an example of prediction using one of the SEM images in the dataset showing that the trained network correctly predicts the image source as the Eagle Ford oil window with high confidence. The figure also shows that there is a moderate likelihood of it belonging to the gas window of the Eagle Ford.

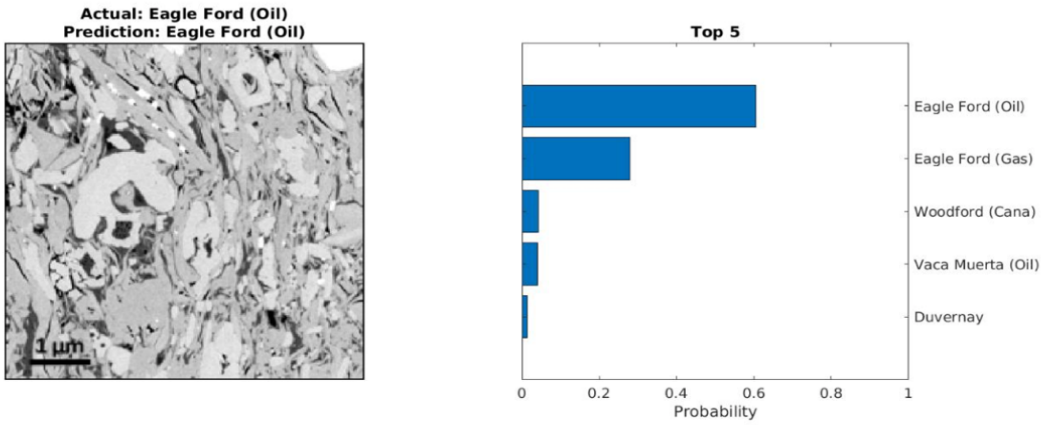


Fig. 1.34 - The model correctly predicts an Eagle Ford (Oil) SEM image as an Eagle Ford (Oil) sample with ~60% confidence (Knaup, 2019).

Knaup (2019) also segmented SEM images to identify pores, organic material, calcite, and K-feldspar. She tested 5 different CNN architectures shown in **Fig. 1.35**. Model 1, Model 2, and Model 3 are fully convolutional neural networks as shown in **Fig. 1.35(a), 1.35(b),** and **1.35(c)**, Model 4 and Model 5 have a U-Net architecture (Ronneberger et al., 2015) with local connections shown in **Fig. 1.35(d)**, and **Fig. 1.35(e)**.

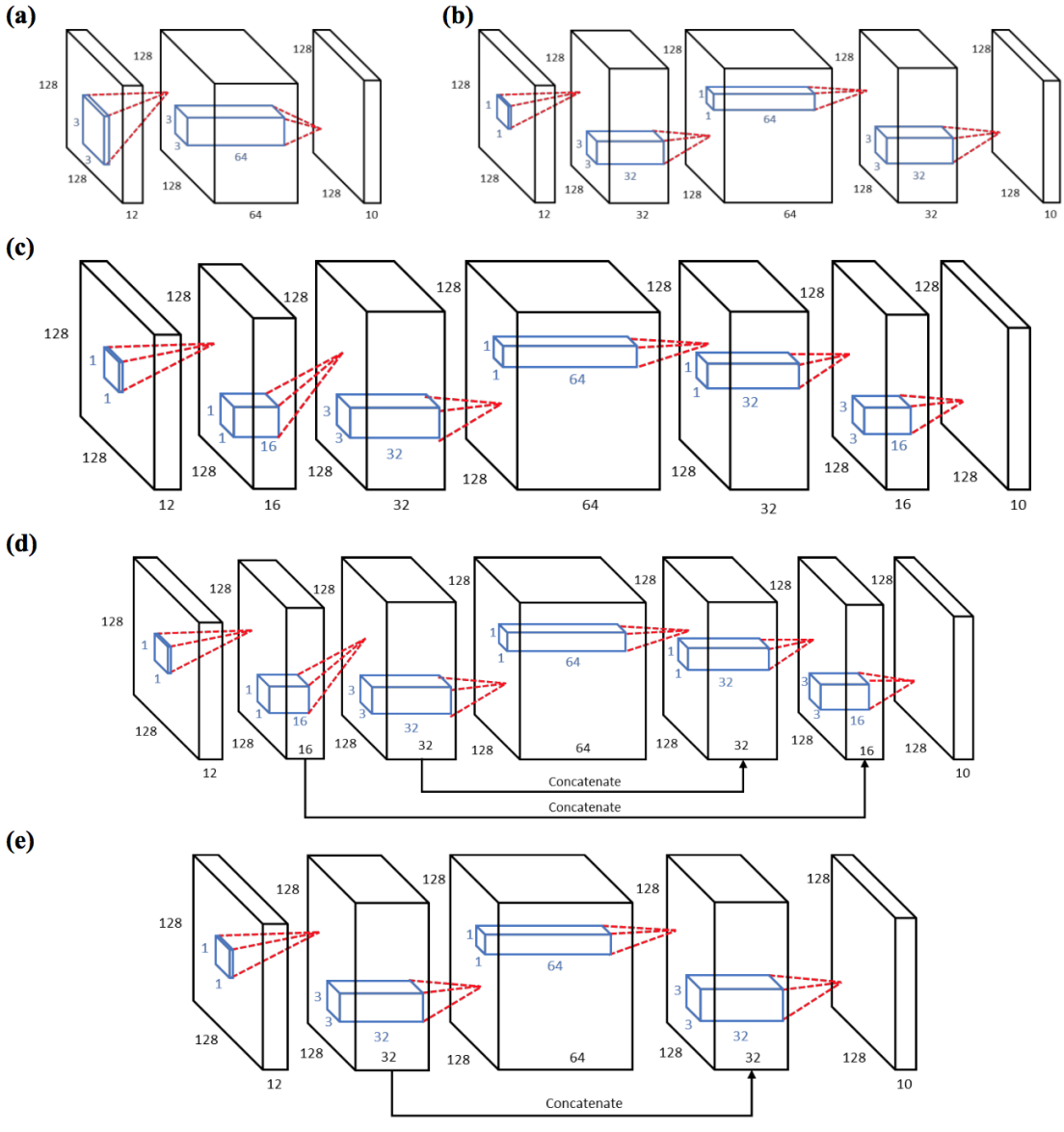


Fig. 1.35 - 5 different architectures tested for image segmentation. (a) Model 1, (b) Model 2, (c) Model 3, (d) Model 4, (e) Model 5. Models 1, 2, and 3 are fully convolutional neural networks, Models 4 and 5 have modified U-Net architectures (Knaup, 2019).

Knaup (2019) uses 208 multi-layered SEM images with a resolution of 10nm, each with Energy-Dispersive X-ray Spectroscopy (EDS) elemental data and hand labels. These images are augmented to over 5000 images and fed to the network. An example of the segmentation is shown in **Fig. 1.36**. The 5-layer improved U-Net CNN was seen to be superior to other CNN architectures, and the performance of the 5-layer U-Net CNN was shown to be significantly high as well, with a pixel classification accuracy of 87%.

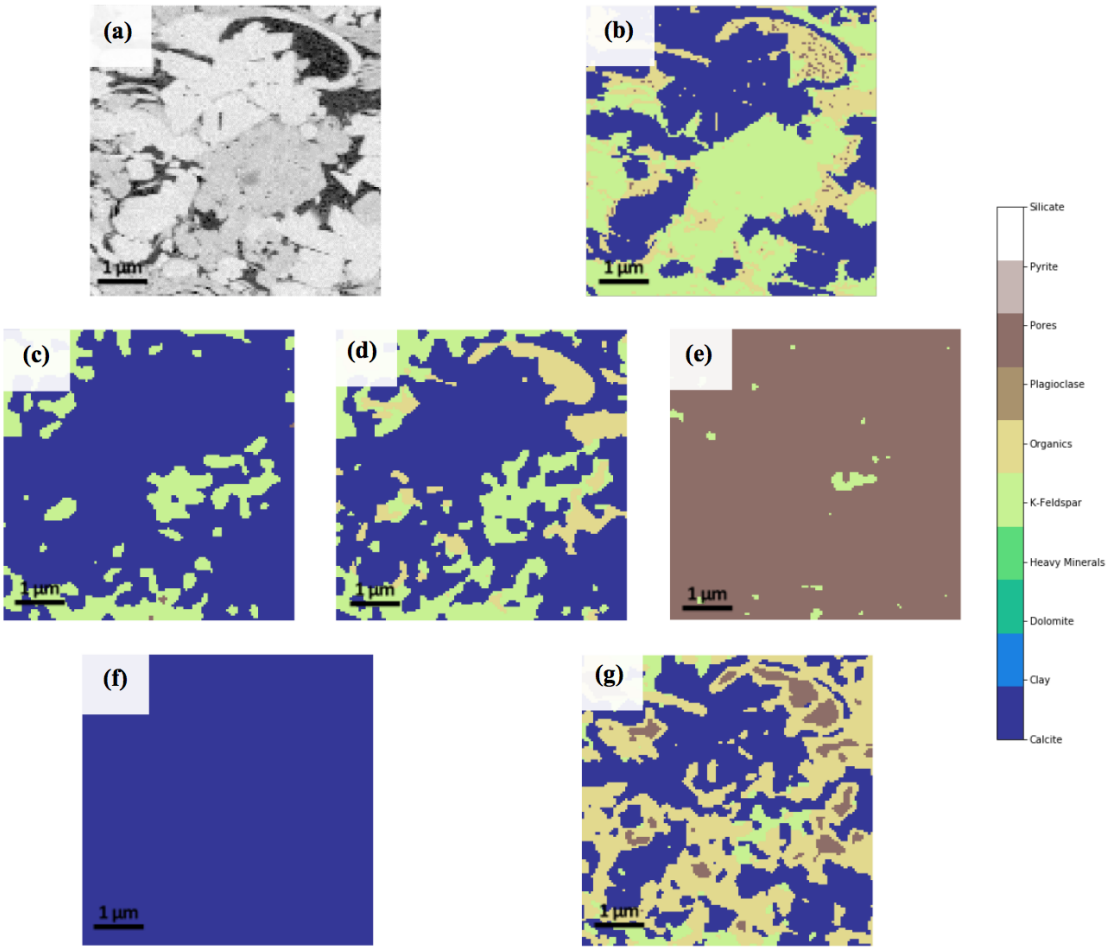


Fig. 1.36 - (a) Inputted SEM image, (b) hand-labeled image, (c) predicted image by using the 3-layer Model1 (d) predicted image by using the 5-layer Model2, (e) predicted image by using the 7-layer Model3, (f) predicted image by using the 7-layer U-Net Model4. (g) predicted image by using a 5-layer U-Net Model5 (Knaup, 2019).

Karimpouli and Tahmasebi (2019) also use gray-scale high-resolution micro-computed tomography (µCT) images of rock samples for segmentation. Because of the need for large

amounts of training data, they use a cross-correlation-based simulation for data augmentation (Karimpouli and Tahmasebi, 2019). They generate over 20000 images using the specified data augmentation method from just 20 original Berea sandstone images (Karimpouli and Tahmasebi, 2019). An 18-layer SegNet (Badrinarayanan et al., 2016) and a modified 38-layer SegNet is used for model training as shown in **Fig. 1.37**. The 38-layer SegNet was shown to be capable of better segmentation accuracy when identifying pore spaces, quartz, other minerals, K-feldspar, and zirconium, respectively (Karimpouli and Tahmasebi, 2019). Kazak et al. (2020) also propose a similar study for tight gas reservoirs in the Berezov formation using a U-Net architecture (Ronneberger et al., 2015).

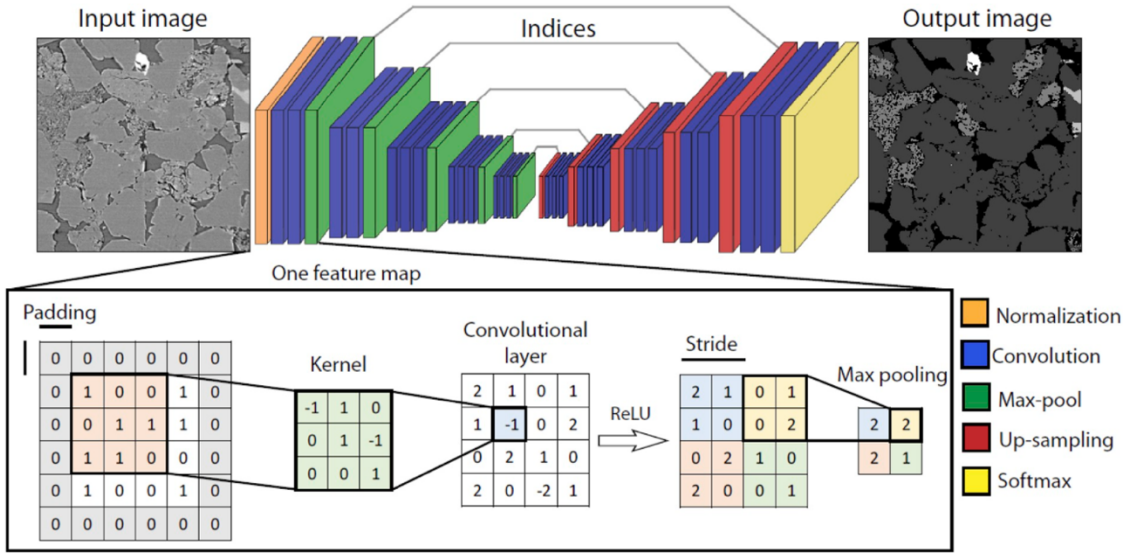


Fig. 1.37 - Workflow of digital rock image segmentation uses the general architecture of SegNet. The original SEM image is input into the improved SegNet architecture, composed of 4 encoders consist of several convolutional layers and maximum pooling layers. And 4 decoders consist of convolutional layers and up-sampling layers. The output result will be a segmented image composed of 5 phases (Karimpouli et al., 2019).

Rushood et al. (2020) use an automatic segmentation method for pores and the matrix from micro-computed tomography (CT) sandstone images. They test three datasets: the first dataset is limited to 600 images; the second data set is a complete dataset with 2200 images; and the third dataset is applied with data augmentation methods for a total of 17600 images.

They use a model with U-Net (Ronneberger et al., 2015) architecture to train the dataset, as shown in **Fig. 1.38**.

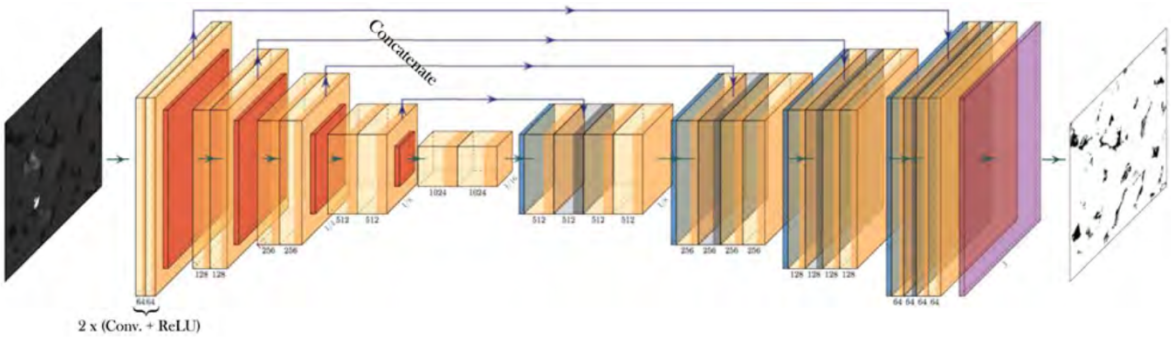


Fig. 1.38 - The original micro-CT image is input into a model with U-Net architecture. It has dual convolution layers followed by ReLU function in each encoder, and up-sampling layers followed by convolution layers in each decoder. The output result will be a segmented image. (Rushood et al., 2020).

Thin section analysis continues to be a time-consuming, laborious process that requires trained experts. Machine learning has enabled a speed-up of thin-section image interpretation. For example, Wei (2019) classifies 8 different types of thin section images obtained using single-polarization light and three different CNN architectures: a pre-trained 16-layer VGG16 model, an untrained 16-layer VGG16 model, and an untrained 42-layer Inception-v3 model.

Later, Su et al. (2020) use a concatenated convolutional neural network (Con-CNN) to successfully classify thin section images. Their dataset includes 13 classes of 92 rock samples and 196 petrographic thin sections for a total of 63504 image patches for training and validation of the 5-layer Con-CNN (Su et al., 2020). A deep network architecture named ResNet-50 (He et al., 2015) was also tested without a significant increase in accuracy.

Jiang et al. (2021) successively trained CNNs to identify vuggy facies using borehole-resistivity images from a well in the Arbuckle Group in Kansas. For model training, two datasets were used: a complete dataset with 4285 images; and a cleaned dataset with 4129 images. They test several models: the shallowest and simplest model has two convolutional layers with 32 and 64 filters, respectively, while the most complex model has four

convolutional layers with 64, 100, 128, 150 filters. The deeper network outperformed the simpler network on both the complete and cleaned datasets.

CNNs can also aid in core image classification. Alzubaidi et al. (2020) use a CNN to automatically classify the lithology of core images into several classes such as sandstone, limestone, shale, and non-core sections, with 93.12% accuracy.

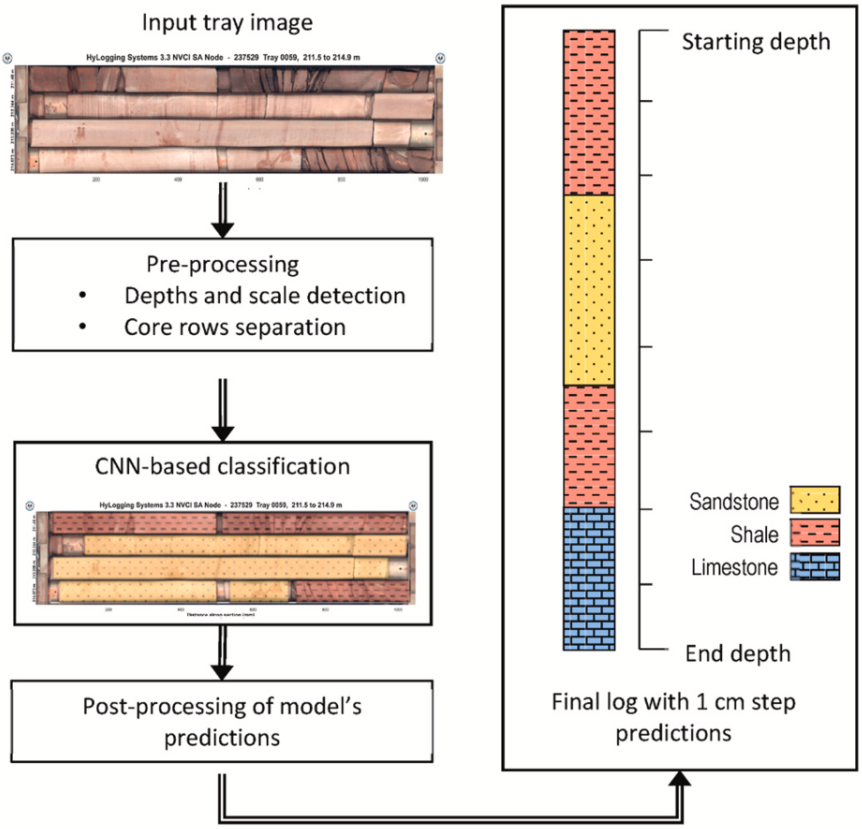


Fig. 1.39 - Workflow for automatically classifying the lithology of core images. For pre-processing, the depth and scale of the tray images are first detected, then the core rows were separated. After that, the images feed into the CNN for classification, and the results are then processed to create the final log with a 1cm step. (Alzubaidi et al., 2020).

The Alzubaidi et al. (2020) workflow is shown in **Fig. 1.39**. The raw data includes 406 sandstone trays, 291 shale trays, and 161 limestone trays from 28 boreholes in South Australia. 54000 images were used to train the network, 13500 were used for validation and 9000 images used for test (Alzubaidi et al., 2020). They test several pre-trained networks such as ResNet (He et al., 2015), Inception-v3 (Szegedy et al., 2015), and ResNeXt-50 (Xie et al., 2016) with the corresponding confusion matrices shown in **Fig. 1.40**.

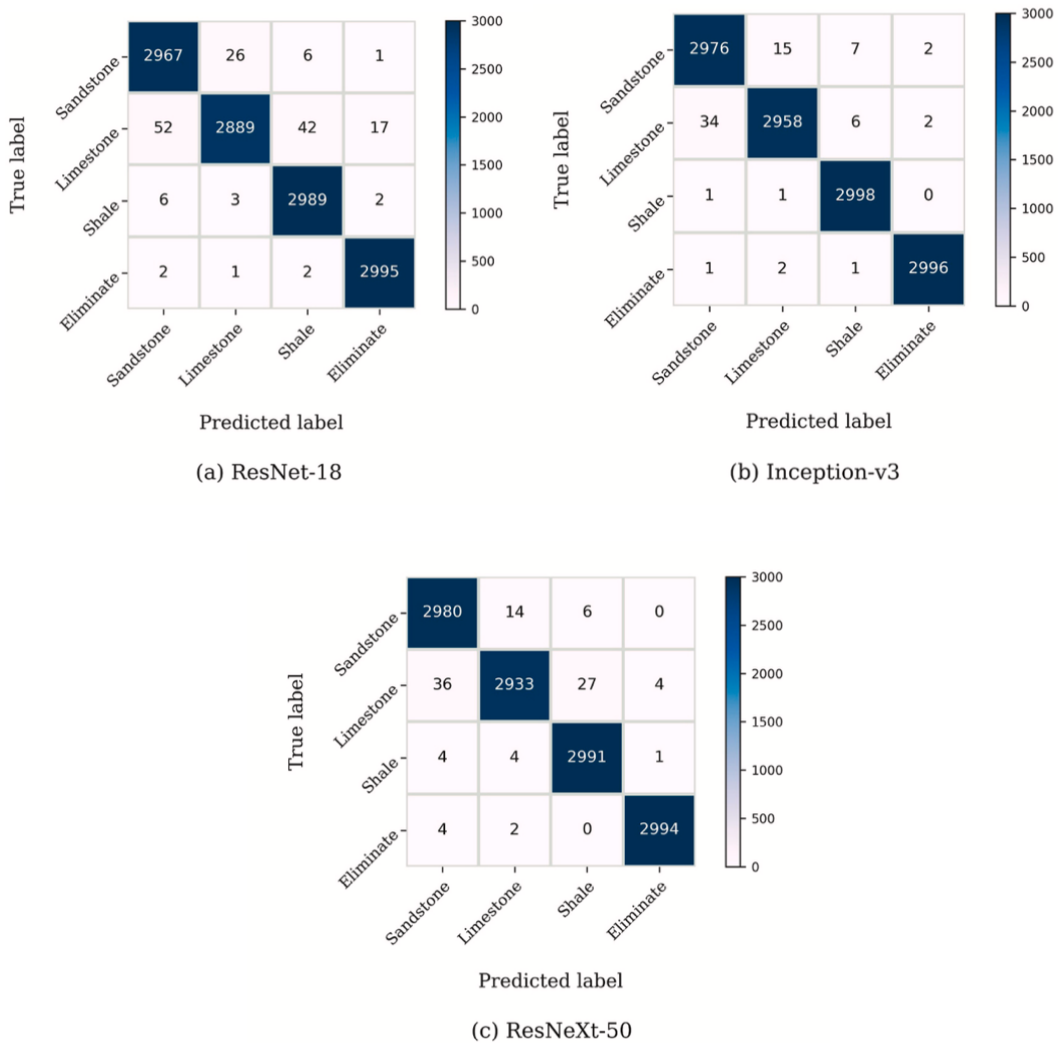


Fig. 1.40 - Confusion matrix for model performance on test images. ResNeXt-50 shows the best performance (Alzubaidi et al., 2020).

In each of the above applications, there are several CNN architectures used for both classification and segmentation. Knaup (2019) indicates that, in her experience, deeper models suffer from overfitting problems and demonstrates that a 5-layer model outperforms the 7-layer model. For thin section classification tasks, Su et al. (2020) also report that increasing model depth from 5 to 50 layers does not result in higher accuracies.

Given the access to abundant computational power, we have not yet matched the problem to be solved to an appropriate level of CNN complexity; rather, in the case of petrographic data, the trend has been to seek deeper and more complex networks. In this thesis,

I attempt to answer whether we really need deeper networks, and whether shallower networks can instead be more competitive? I answer these questions in the context of image classification specifically. I also address whether a shallower network with a wider diversity of convolutional filters (breadth) can outperform a deeper network? What image resolution and filter complexity do we need to achieve a high degree of accurate classification? In the end, I provide guidelines to select the appropriate level of depth and breadth for formation identification using SEM images.

1.9 Thesis Organization

This thesis is organized into four chapters and is structured as follows:

- Chapter 1: This chapter reviews the literature on machine learning, convolutional neural networks (CNN), CNN visualization, and its use in the oil and gas field. This chapter provides background for shallow versus deep network analysis on play identification.
- Chapter 2: This chapter addresses filter depth and breadth for play identification from SEM images belong to 8 formations at 25nm/pixel and 10nm/pixel resolution.
- Chapter 3: I follow Chapter 2 by extending the dataset to 22 different plays and consider two other resolutions at 50nm/ pixel and 25nm/pixel. This chapter also analyzes the sensitivity of filter complexity to the number of classes to be labeled.
- Chapter 4: In this chapter, I provide a summary of my findings and draw conclusions that will aid practitioners of machine learning in attempting to extend their work to image classification.

Chapter 2: SEM Image Classification for a Dataset Comprising 8 Formations

In this chapter, I investigate the effect of varying filter depth and breadth when classifying images acquired from eight formations at two different resolutions. I hypothesize that filter depth and breadth are likely to be a function of the number of classes/labels to be identified, and in this chapter, a modest number of classes allows me to provide a more in-depth look at filter performance. By modifying the resolutions of the images, I can also assess whether the classification accuracy is sensitive to image field-of-view.

In a subsequent chapter, I expand the number of classes to 22 and assess the sensitivity of network complexity to the number of classes to be identified.

2.1 Description of the Systematic Approach

I create a systematic workflow to test the depth and breadth sensitivity on the image database with eight formations, as shown in **Fig. 2.1**.

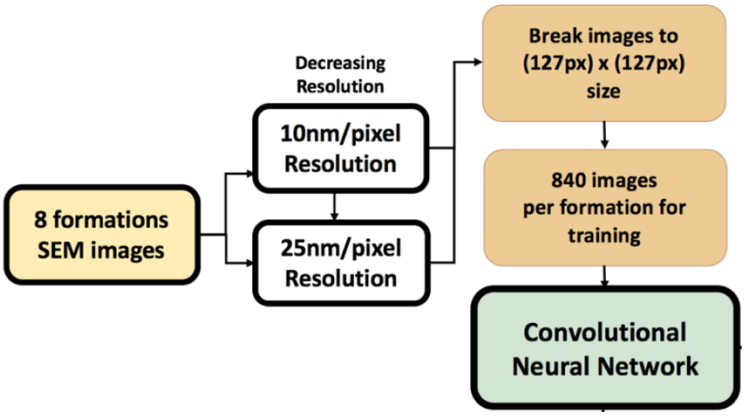


Fig. 2.1 - Systematic approach of testing depth and breadth sensitivity using datasets with 8 plays.

In this chapter, I use a dataset of grayscale SEM images from 8 formations: Green River, La Luna, Horn River Evie, Point Pleasant, Alum, Wolfcamp, Osage, Duverna. The details of the dataset are shown in **Table. 2.1**.

Play	Resolution (nm)	Bit Depth	# of images
Wolfcamp	10	8	799
Alum	10	16	900
Duvernay	10	16	900
Osage	10	16	900
La Luna	10	16	275
Point Pleasant	10	16	900
Green River	10	16	400
Horn River Evie	10	16	400

Table. 2.1 - SEM image information including resolution, bit depth, and the number of images in each play. The bit-depth is the number of bits used to symbolize the color of a single-pixel.

The grayscale SEM images are standardized to 8-bit depth (8-bits are used to represent the grayscale levels). I construct two datasets with varying resolutions: 25nm/pixel and 10nm/pixel. Then the images are sliced to 127 x 127 pixels without overlap. At 25nm/pixel, the field-of-view for a 127x127 pixel image is 3x3 µm, and at 10nm/pixel, the field-of-view is 1x1 µm. Henceforth, when referring to image resolution, I will use the units of nm/px and not nm/pixel. A 25nm/px and a 10nm/px image from the exact same location are shown in **Fig. 2.2**. My hypothesis is that at the 10nm/px resolution, we may miss some of the larger-scale features that might aid identification.

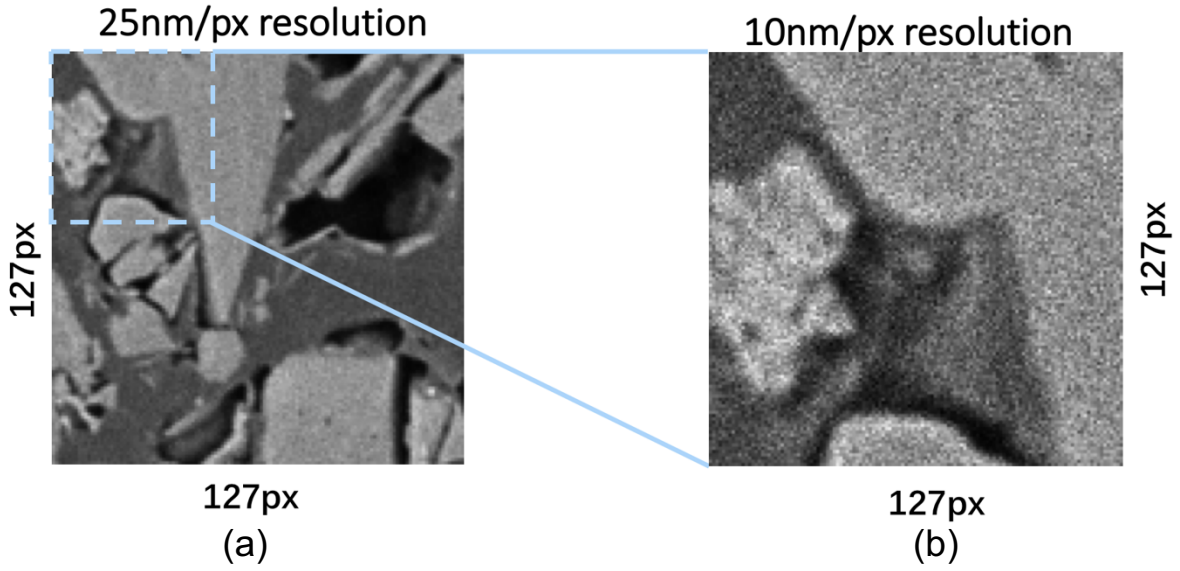


Fig. 2.2 - (a) An example of the 25nm/px resolution 127x127 pixel size (3x3 µm field-of-view) image, (b) an example of the 10nm/px resolution 127x127 pixel size (1x1 µm field-of-view) image.

My dataset includes 840 images per formation with a total of 6720 images for training, 180 images per formation and a total of 1440 images for validation, and 180 images per formation and a total of 1440 images for testing.

2.2 Models Considered in this Chapter

I use Google Colaboratory (Colab) (Bisong, 2019) that provides two Tesla V100-SXM2-16GB GPUs for parallel computing. I use the open-source TensorFlow library version 2.2 (Abadi et al., 2015) as a machine learning platform. Keras (Chollet, F. 2015) is subsequently used as an interface to the TensorFlow library to provide a python interface for the construction of CNNs.

For all the models I test, the number of layers refers to the number of convolutional layers. The pooling layer and the flattening layer do not contain trainable parameters (Jiang, 2021). I follow the 5-convolutional layer AlexNet (Krizhevsky et al., 2012) architecture to build the network inspired by a similar play identification task by Knaup (2019). Nonetheless, I remove one of the three fully connected layers for simplicity. This thesis will mainly focus on the convolutional layers of each network. Each model has been trained and tested over three times, and the best test accuracy is recorded.

# of Layers	1	2	3	5
# of Filters	1	--	--	--
# of Filters	2	2, 2	2, 4, 4	2, 4, 4, 8, 8
# of Filters	4	4, 4	4, 8, 8	4, 8, 8, 16, 16
# of Filters	8	8, 8	8, 16, 16	8, 16, 16, 32, 32
# of Filters	16	16, 16	16, 32, 32	16, 32, 32, 48, 48
# of Filters	32	32, 32	32, 64, 64	32, 64, 64, 96, 96

Table. 2.2 - CNN architecture tested using SEM images from the 8 formations at 25nm/px and 10nm/px resolutions.

To test the sensitivity of convolutional neural networks to depth and width, I start with the simplest 1-layer, 1-filter CNN architecture shown in **Fig. 2.3** and evaluate the trained model

performance on the test datasets. I then increase the breadth of the network by increasing the number of filters in the 1-layer model, as shown in **Table. 2.2**. Successively, I then increase the depth (add a layer to the CNN) and increase breadth (add filters to each new layer) and assess the sensitivity of the network accuracy to both depth and breadth. The number of filters in each layer of the deep networks successively increases, following VGG-16 architecture (Simonyan et al., 2014). For each CNN architecture listed in **Table. 2.2**, I run multiple tests and chose the highest accuracy for each model considered.

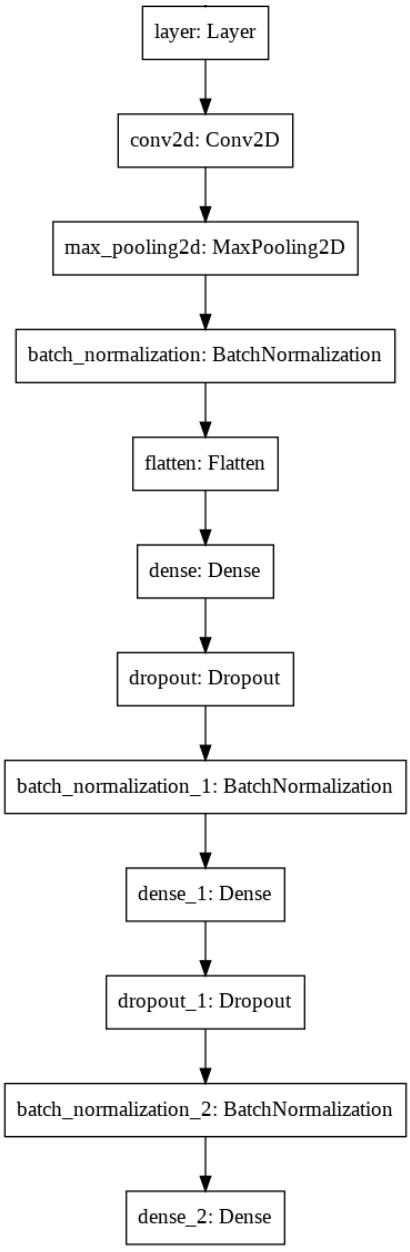


Fig. 2.3 - The simplest 1- layer 1-filter network architecture.

2.3 25nm/px Resolution SEM Images

In this section, I use a dataset of gray-scale SEM images with 25nm/px resolution from 8 formations: Green River, La Luna, Horn River Evie, Point Pleasant, Alum, Wolfcamp, Osage, Duvernay. The images are reshaped to 127x127 pixels without overlap, equivalent to a 3x3 µm field-of-view as shown in **Fig. 2.4**.

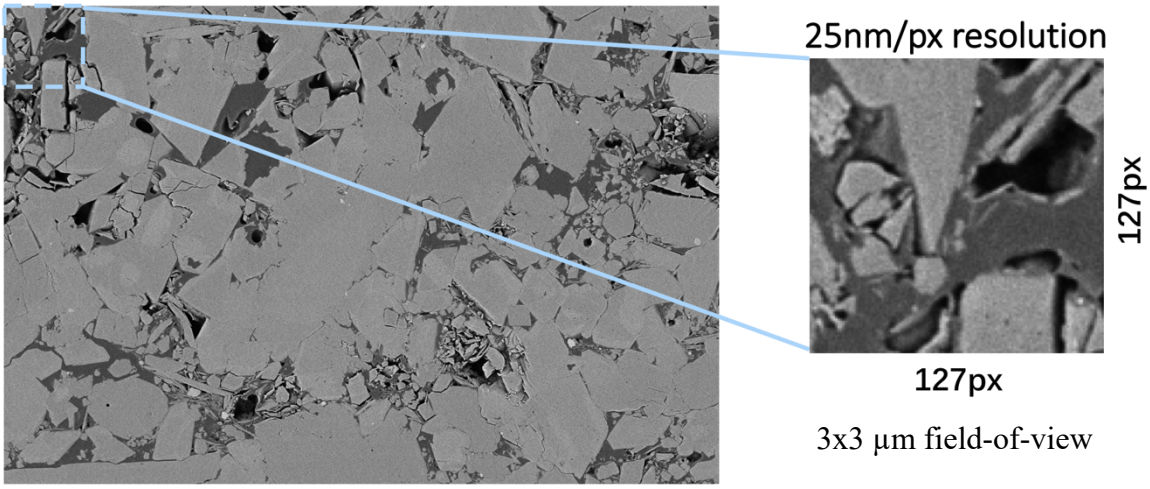


Fig. 2.4 - Example of a raw SEM image from Green River at 10nm/px resolution. It is rescaled to 25nm/px resolution and sliced to 127x127 pixels size (3x3 µm field-of-view) to fit into the model. The left figure is the raw image; the right figure is an example of the rescaled and sliced images for CNN model training.

Table. 2.3 shows the detailed information of the input images from each play for the 25nm/px resolution dataset. The input images from 8 plays are split into a training set (840 images per formation and a total of 6720 images), a validation set (180 images per formation and a total of 1440 images), and a test set (180 images per formation and a total of 1440 images) Twenty example images from the 25nm/px resolution dataset are shown in **Fig. 2.5**.

Play	Resolution (nm)	Bit Depth	# of images for training	# of images for validation	# of images for testing
Wolfcamp	25	8	840	180	180
Alum	25	8	840	180	180
Duvernay	25	8	840	180	180
Osage	25	8	840	180	180
La Luna	25	8	840	180	180
Point Pleasant	25	8	840	180	180
Green River	25	8	840	180	180
Horn River Evie	25	8	840	180	180

Table. 2.3 - 25nm/px resolution (3x3 μm field-of-view) SEM image information of 8 plays.

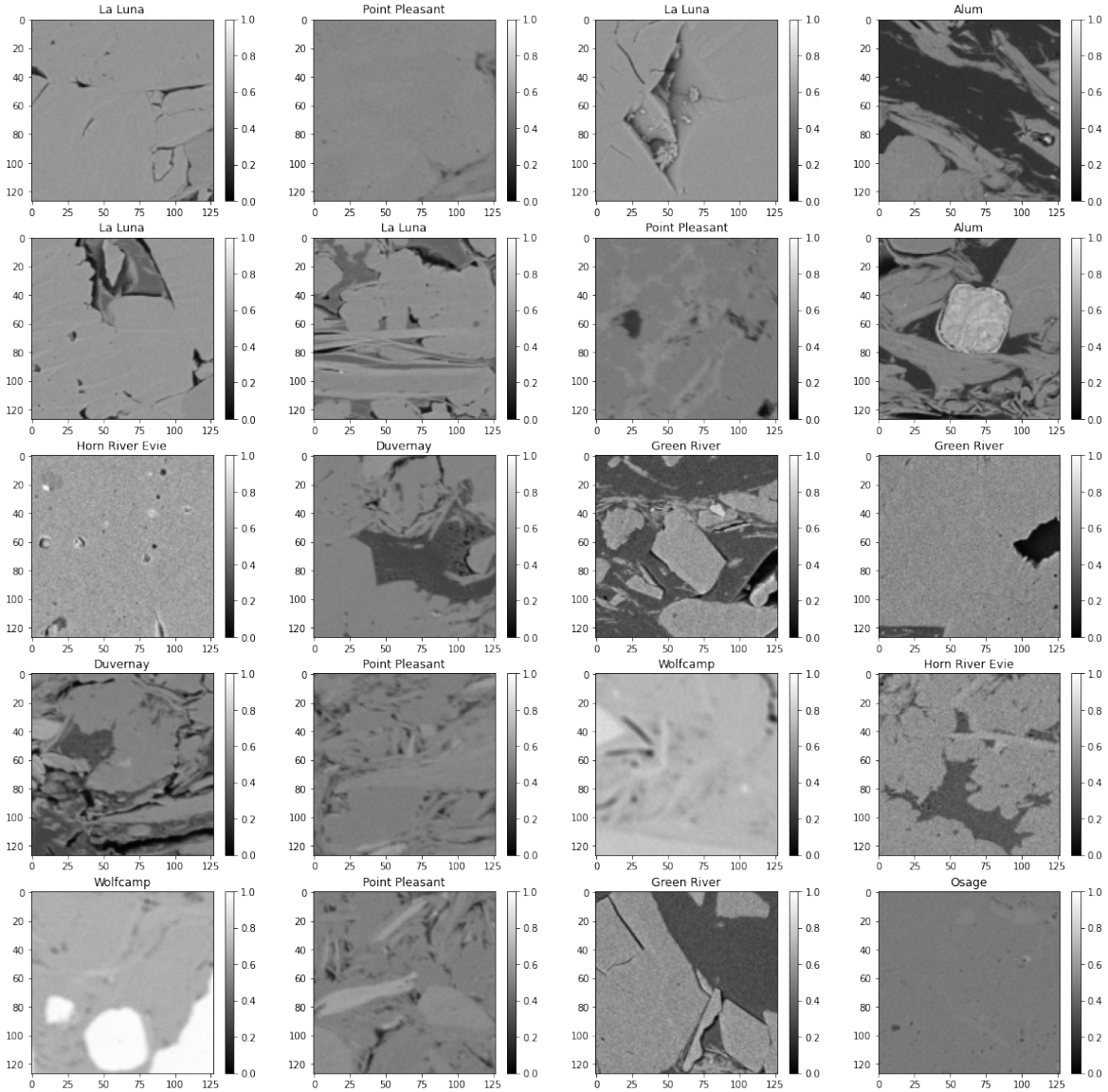


Fig. 2.5 - Twenty grayscale SEM images from 8 plays for play identification at 25nm/px resolution 127x127 pixel size (3x3 μm field-of-view).

2.4 25nm/px Resolution: Shallow Network Results

The simple 1-layer 1-filter CNN only achieves 65% accuracy on the 25 nm/px resolution (3x3 μm field-of-view) dataset. Increasing the breadth in the 1-layer network from 1 filter to 4 filters, the network accuracy goes up to 80%. The model architecture and the corresponding test results are shown in **Table. 2.4**. A subsequent increase to 8, 16, and 32 filters in the 1-layer model does not substantially enhance the accuracy showing that increases in filter width/diversity do not provide any measurable benefit beyond a certain limit. Moreover, with extremely limited depth (1-layer), even the 1-layer 32-filters model is limited in terms of accuracy, underscoring the need for an increased depth of the network. However, I do want to point out that even the simple 1-layer, 4-filter does provide ~80% accuracy, which is a substantial increase over the potential accuracy of 100/8 = 12.5% obtainable just by pure chance.

# of Layers	1	1	1	1	1
# of Filters	1	4	8	16	32
Accuracy %	65	79	83	87	81

Table. 2.4 - Accuracy of the 1-layer networks test on the SEM images at 25nm/px resolution.

The corresponding confusion matrices of the 1-layer 1-filter networks are shown in **Fig. 2.6.** The extremely simple 1-layer 1-filter network achieves a total accuracy of 65%, including higher recall obtained from Wolfcamp and Osage samples. The recall is calculated by the true positives of a given category divided by the true positives and false negatives. On the other hand, the network misclassified a few of the Point Pleasant samples as Duvernay and a few of the Duvernay samples as Point Pleasant samples, indicating that the trained 1-layer 1-filter network detects some similarities between these formation pairs. There is also an appreciable number of false positives for the Horn River Evie when the input image is actually from the Green River.

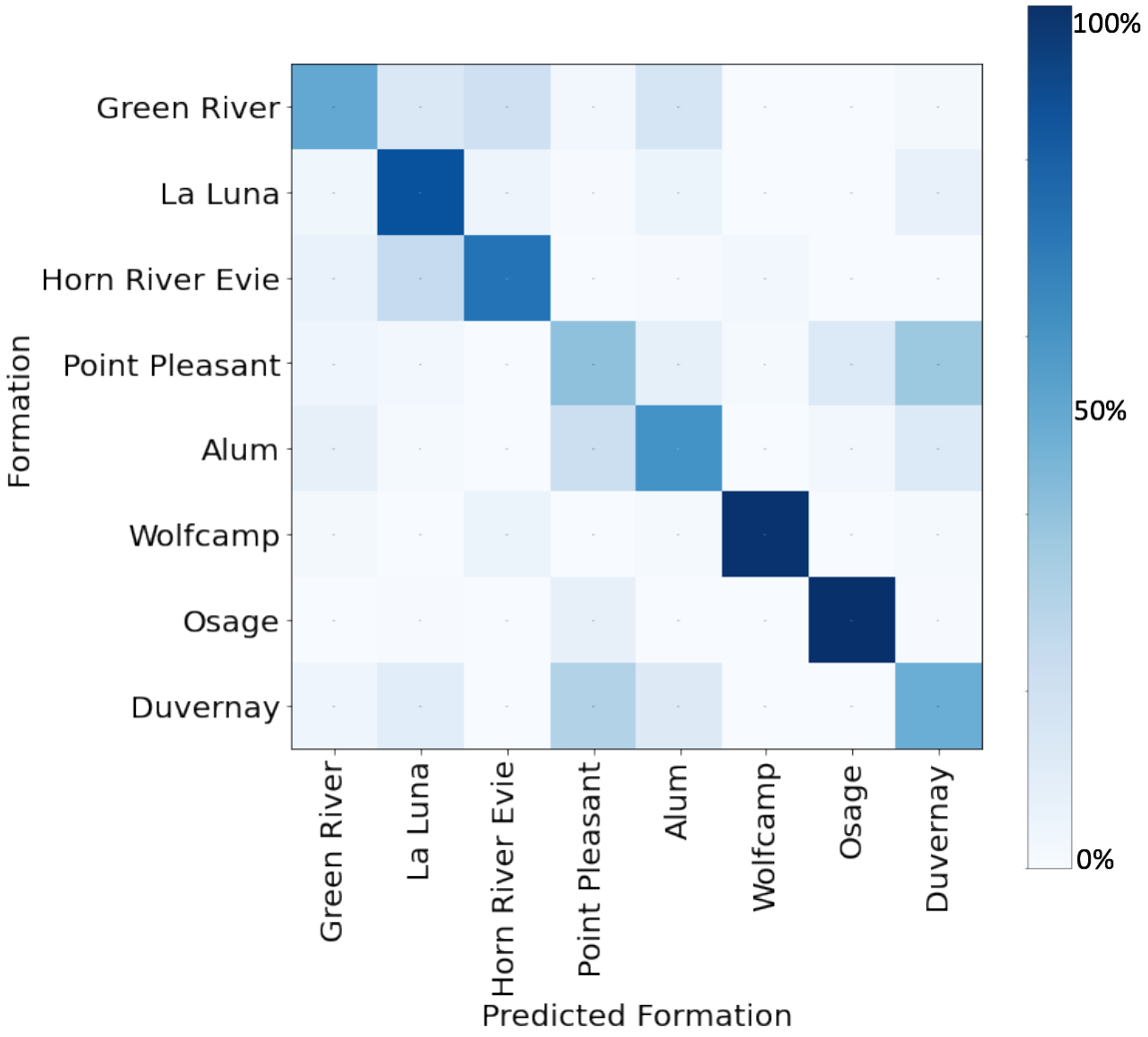


Fig. 2.6 - Confusion matrix of the 1-layer 1-filter network trained on 25nm/px resolution dataset achieves a total accuracy of 65%.

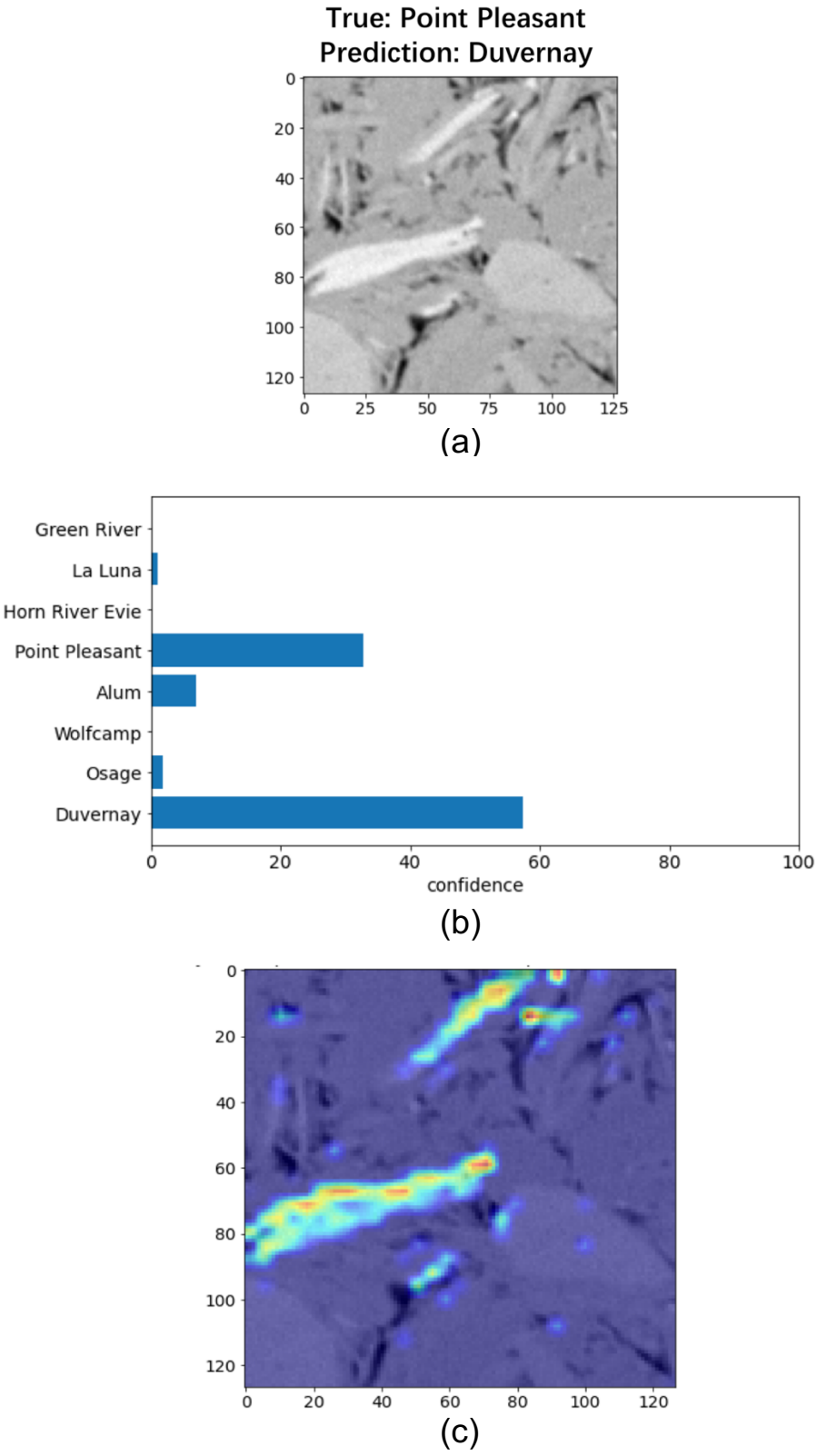


Fig. 2.7 - (a) The Point Pleasant image input to the 1-layer 1-filter network is misclassified as a Duvernay sample, (b) the model predicts the image with over a 60% probability being a Duvernay sample, (c) heatmap output from the convolutional layer.

I show an incorrectly predicted image sample from the formation pair with the highest misclassification rate obtained from the 1-layer 1-filter trained CNN. In this case, a sample from the Point Pleasant is fed to the network shown in **Fig. 2.7(a)**. The network predicts this Point Pleasant image as a Duvernay sample with ~60% probability, with under 30% probability of it being a Point Pleasant sample as shown in **Fig. 2.7(b).** I also show the heatmap obtained using the same image, which highlights important features that led to the CNN classification decisions. The hotter color (red and yellow) reveals the most important features for identifying the play, and the colder color (blue) indicates less critical features. As shown in **Fig. 2.7(c)**, in the only convolutional layer, the trained CNN considers the lighter chlorite clay platelets feature to be important. It is important to mention that the image classification could be enhanced if, for instance, mineralogical information was available.

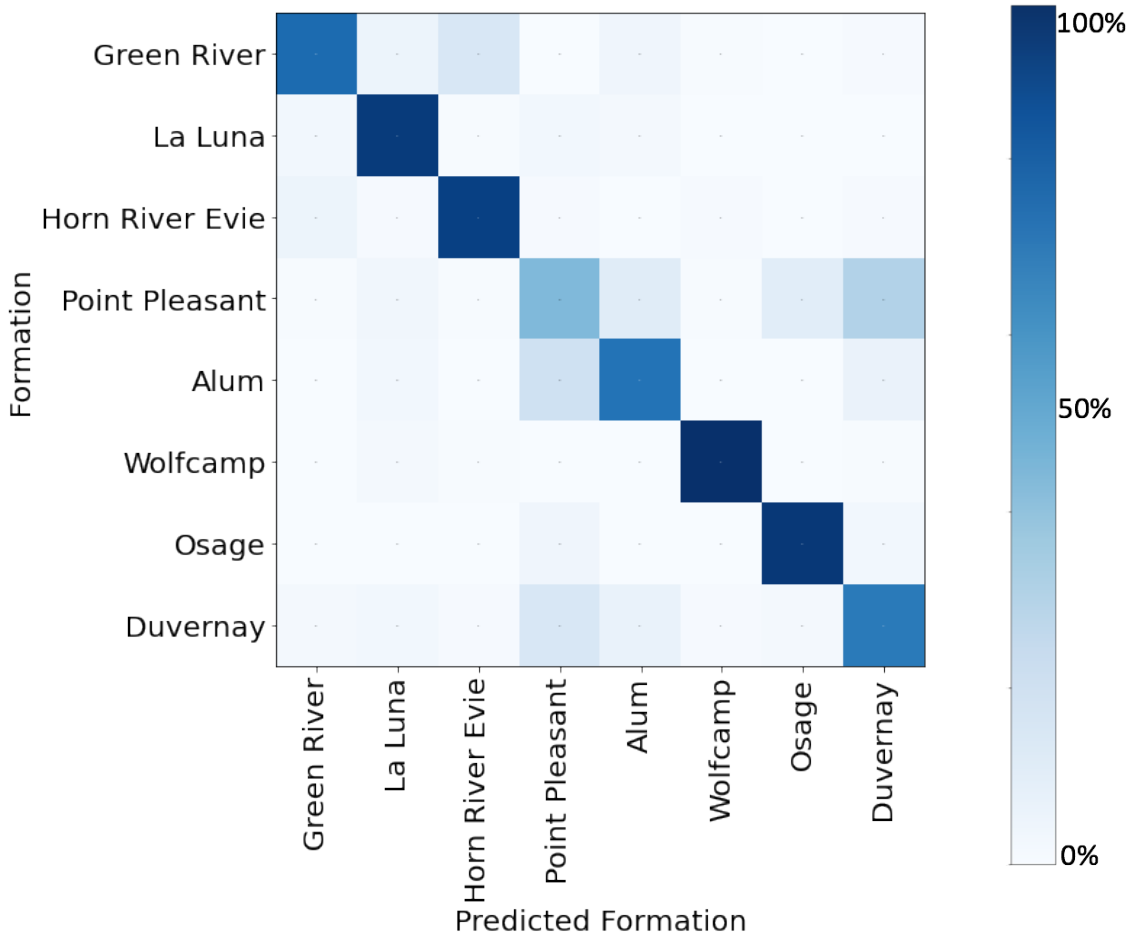


Fig. 2.8 - Confusion matrix of the 1-layer 4-filters network trained on 25nm/px resolution dataset achieves a total accuracy of 79%.

Conversely, as shown in **Fig. 2.8**, if the number of filters in the 1-layer network increases from 1 to 4 filters, there are fewer off-diagonal elements. On the other hand, the network appears to continue to misclassify an appreciable number of Point Pleasant images as Duvernay samples, and Green River samples as Horn River Evie samples, indicating that a few of the SEM images from these pairs probably display similar microstructural features, confounding the shallow 1-layer network.

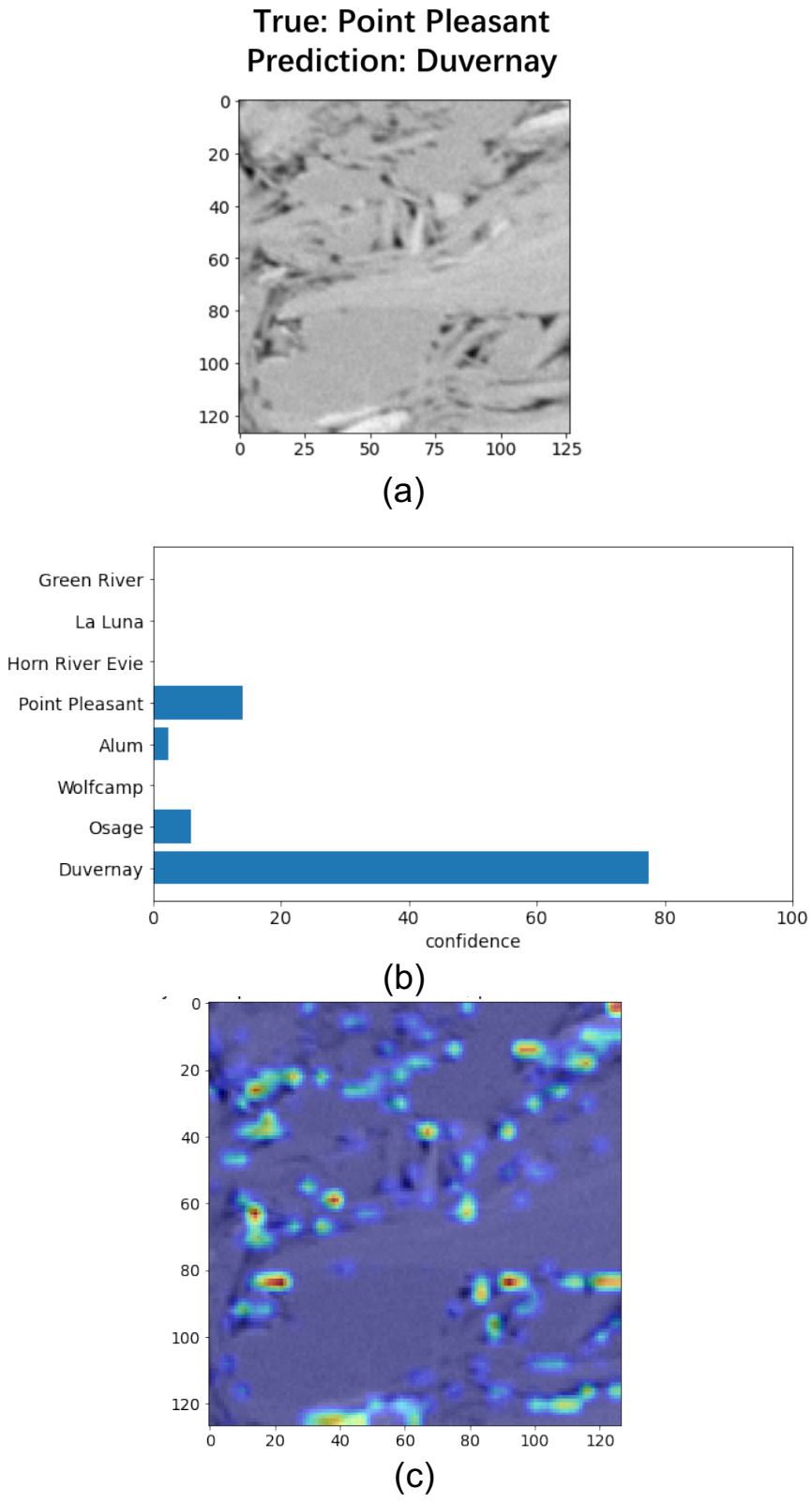


Fig. 2.9 - (a) The Point Pleasant image input to the 1-layer 4-filters networks is misclassified as a Duvernay sample, (b) the model predicts the image with over about an 80% probability being a Duvernay sample, (c) heatmap output from the convolutional layer.

As demonstrated in the previous example, I show a false-negative image sample from the formation that has the lowest recall when tested on the 1-layer 4-filters trained CNN. Again, it is a sample from the Point Pleasant shown in **Fig. 2.9(a)**. The network predicts this Point Pleasant image as a Duvernay sample with over 80% probability as shown in **Fig. 2.9(b).** The heatmaps obtained from the 1-layer 4-filters trained CNN is shown in **Fig. 2.9**. It is quite clear that the more important features picked by this network are different from a simple grayscale contrast picked by the simpler 1-layer, 1-filter model as shown in **Fig. 2.9(c).**

The 1-layer 1-filter and 1-layer 4-filters CNN feature map (filter output) are shown in **Fig. 2.10**, showing the important features captured by each filter that led to the CNN classification decisions. An SEM image from the Alum formation as shown in **Fig. 2.10(a)**, is fed to the 1-layer 1-filter trained CNN, following which I obtain the filter as shown in **Fig. 2.10(b)**. The feature map appears to be capturing grayscale contrasts in the original image but ignores other microstructural features, such as pores and organics.

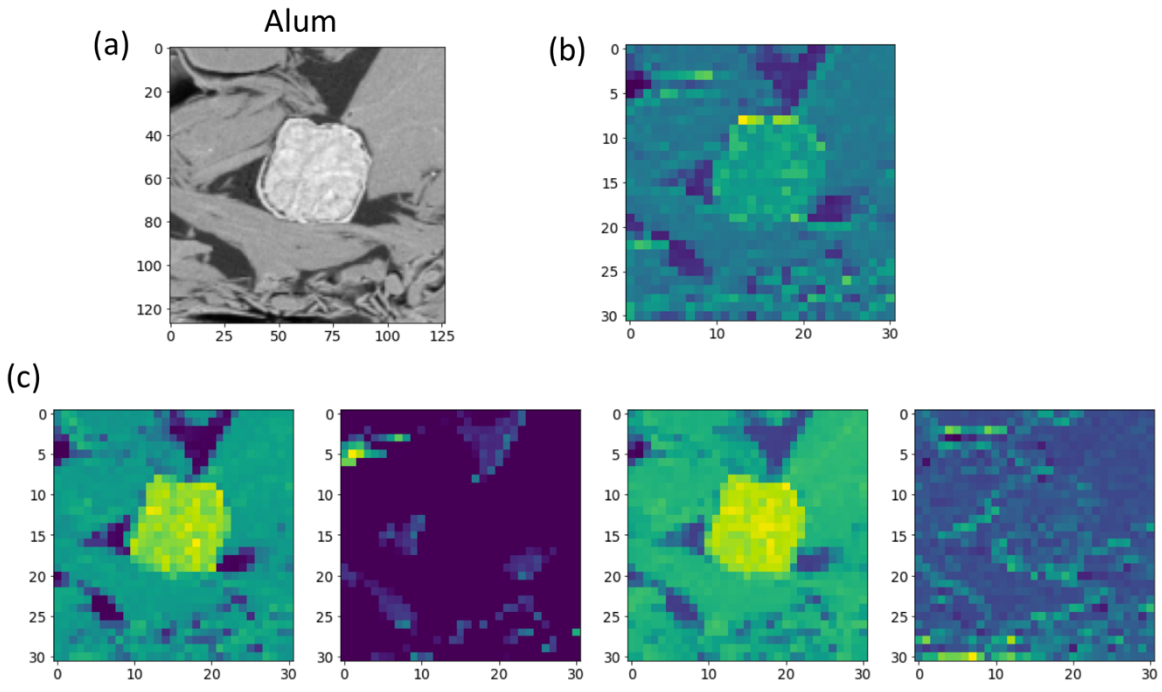


Fig. 2.10 - Feature maps output from the two shallow networks. (a) An SEM image fed into the two networks, (b) a feature map output from the 1-layer 1-filter network, (c) four feature maps output from the 1-layer 4-filters network.

In contrast, the feature maps from the 1-layer 4-filters CNN from the analysis of the same image are different as shown in **Fig. 2.10(c)**. In the first and third panels, the filters appear to be sensitive to grayscale contrasts, while the second panel is picking out darker organic/pores. The fourth panel appears to be detecting edges. Although, in this example, the feature selection is apparent, there are several cases where the feature maps make non-intuitive choices.

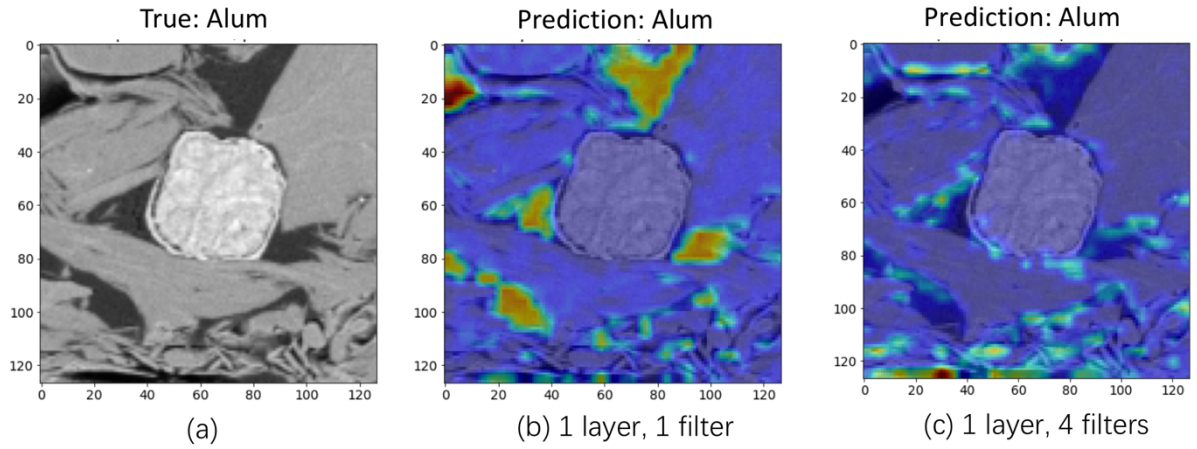


Fig. 2.11 - (a) An SEM image fed into the two shallow networks, (b) heatmap output from the 1-layer 1-filter network. (c) heatmap output from the 1-layer 4-filters network.

The heatmap from the 1-layer 1-filter network (**Fig. 2.11(b)**) shows the organic matter on the original grayscale image to be important. Increasing the breadth to 4 filters, the heatmap (**Fig. 2.11(c)**) shows that the 1-layer 4-filters network no longer picks the darker organic matters as the critical feature, but instead focuses on a few select microstructural elements. Analysis of the picks in **Fig. 2.11(c)** are non-intuitive choices. Unfortunately, this is a recurring theme in this thesis. Important features extracted from an SEM image by a trained CNN often do not correspond to features a trained operator would use to aid identification.

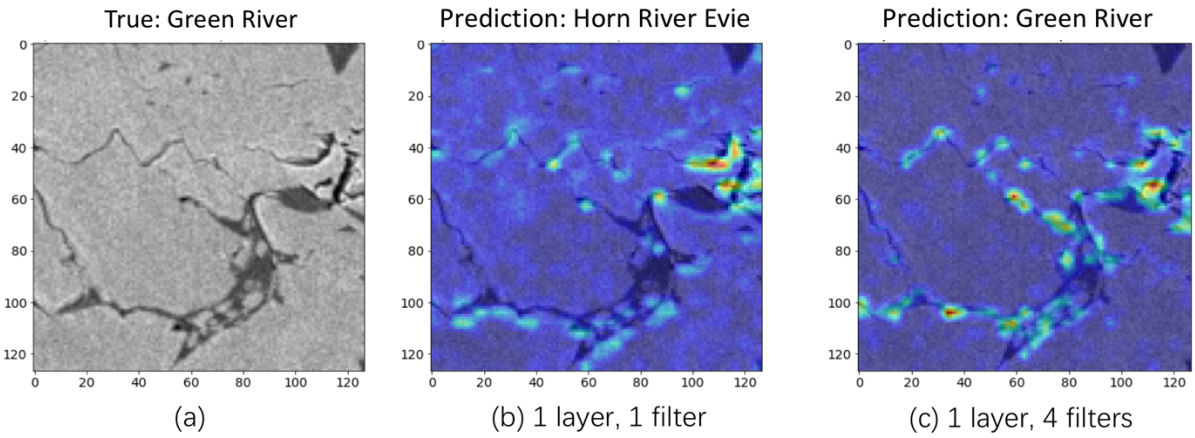


Fig. 2.12 - (a) An SEM image fed into the two shallow networks, (b) heatmap output from the 1-layer 1-filter network which misclassified this Green River image as a Horn River Evie sample, (c) heatmap output from the 1-layer 4-filters network which correctly classified the image as a Green River sample.

There are several examples of misclassification by the 1-layer 1-filter network when focusing on the grayscale contrast. **Fig. 2.12** is one such example where the 1-layer 1-filter network misclassifies a Green River image as a Horn River Evie sample by considering solely the grayscale contrast and picking the darkest pore feature to make the prediction. The 1-layer 4-filter network however extracts a few other additional features with a resulting correct classification. As mentioned earlier, these picks are difficult to interpret. So, while we are able to visualize important elements for classification, it is essential to note that the significant features that aid manual classification may not be the same as those picked by CNN.

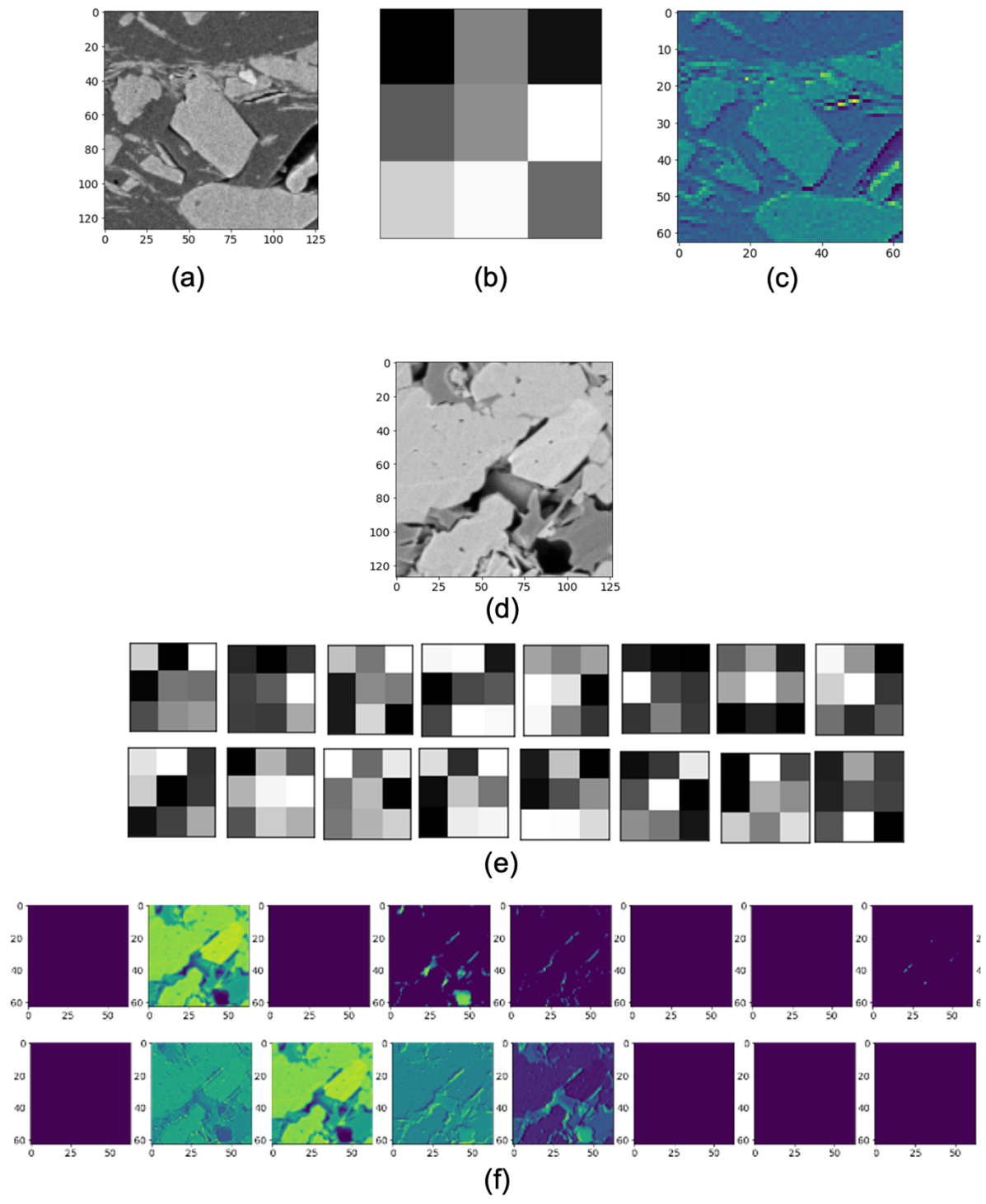


Fig. 2.13 - (a) Green River sample input to the 1-layer 1-filter network (b) filter in the 1-layer 1-filter network (c) feature map output from the 1-layer 1-filter network (d) La Luna sample input to the 1-layer 16-filters network (e) filter in the 1-layer 16-filters network (f) feature map output from the 1-layer 16-filters network

I also show the pixel value of each filter and corresponding filter output (feature maps) of a specific image in the 1-layer 1-filter network and 1-layer 16-filters network. A Green River

sample is input to the network, as shown in **Fig. 2.13(a)**. The pixel value of the filter in the 1-layer 1-filter network is shown in **Fig. 2.13(b)**. This filter is shown to extract some grayscale information and edges as shown in **Fig. 2.13(c)**. The 1-layer 16-filters network can potentially extract several more features with the filters shown in **Fig. 2.13(e)**. The feature maps processed by each filter are shown in **Fig. 2.13(f)**. The 2^{nd} and 11^{th} filters capture the grayscale of the input sample. The 4^{th}, 5^{th}, and 8^{th} filters extract darker diagonal features, including pores, and the 10^{th}, 12^{th}, and 13^{th} filters capture the edges of the darker features of the input sample.

As mentioned earlier, a subsequent increase to 8, 16, and 32 filters in the 1-layer model does not substantially enhance the accuracy, underscoring the need for an increased depth of the network, which I will discuss in the next section.

2.5 25nm/px Resolution: Modest and Deep Network Results

While 1-layer networks can achieve over 80% accuracy with an increase in filter breadth, I also test the sensitivity of the classification to changes in filter depth. In this section, I consider CNNs with 2, 3, and 5 convolutional layers. **Table. 2.5** shows the performance of the shallow (1-layer), modest (2-, 3- layer), and deep (5-layer) networks, and more detailed results are shown in **Appendix Fig. A1**.

As shown in **Table. 2.5(a),** I increase the number of layers but retain 2 filters in Layer1. The shallow 1-layer 2-filters model shows similar accuracy of slightly over 70% as the 2-layer 2, 2-filters model. In contrast, the 3-layer and 5-layer networks outperform the shallow networks with over 80% accuracy.

I then double the number of filters (breadth) in each layer for all the models as shown in **Table. 2.5(b)**. The 1-layer 4-filters network still shows about 80% accuracy. However, the 2-layer, the 3-layer, and the 5-layer networks achieve a comparable accuracy of ~ 90%. A

comparison with **Table. 2.5(a)** shows over a 5% increase in accuracy with a doubling of the number of filters in each layer.

If I double the number of filters (breadth) to 8 and 16 filters in the first layer, as shown in **Table. 2.5(c)** and **Table. 2.5(d)**, the accuracy of the shallow 1-layer network remains below 90%, while the 2-, 3-, and 5- layers networks achieved comparable accuracy of over 90%. A further increase to 32 filters in Layer 1 does not lead to an increase in accuracy as shown in **Table. 2.5(e)**. Again, the 2-, 3-, and 5- layer networks are capable of exceed 90% accuracy.

The results in **Table. 2.5** show that beyond a certain depth or beyond a certain layer width, there are no appreciable improvements in performance. However, below these thresholds, filter performance is compromised as with the 2-layer 2, 2-filters model.

# of Layers	1	2	3	5
# of Filters	2	2, 2	2, 4, 4	2, 4, 4, 8, 8
Accuracy %	73	76	86	85

(a)

# of Layers	1	2	3	5
# of Filters	4	4, 4	4, 8, 8	4, 8, 8, 16, 16
Accuracy %	79	88	87	93

(b)

# of Layers	1	2	3	5
# of Filters	8	8, 8	8, 16, 16	8, 16, 16, 32, 32
Accuracy %	83	91	94	94

(c)

# of Layers	1	2	3	5
# of Filters	16	16, 16	16, 32, 32	16, 32, 32, 48, 48
Accuracy %	87	94	96	96

(d)

# of Layers	1	2	3	5
# of Filters	32	32, 32	32, 64, 64	32, 64, 64, 96, 96
Accuracy %	81	93	96	95

(e)

Table. 2.5 - Shallow vs. deep network performance on 25nm/px resolution dataset.

The confusion matrix corresponding to the modest 2-layer 8, 8-filters model with a classification accuracy of 91% is shown in **Fig. 2.14.** Fewer off-diagonal elements can be observed compared with the 1-layer networks with higher recall for the La Luna, Horn River Evie, Alum, Wolfcamp, and Osage formations. The network however continues to misclassify a few Point Pleasant samples as Duvernay samples, a few Duvernay samples as Point Pleasant samples, and a few Green River samples as Horn River Evie samples, which was observed in the shallower networks as well.

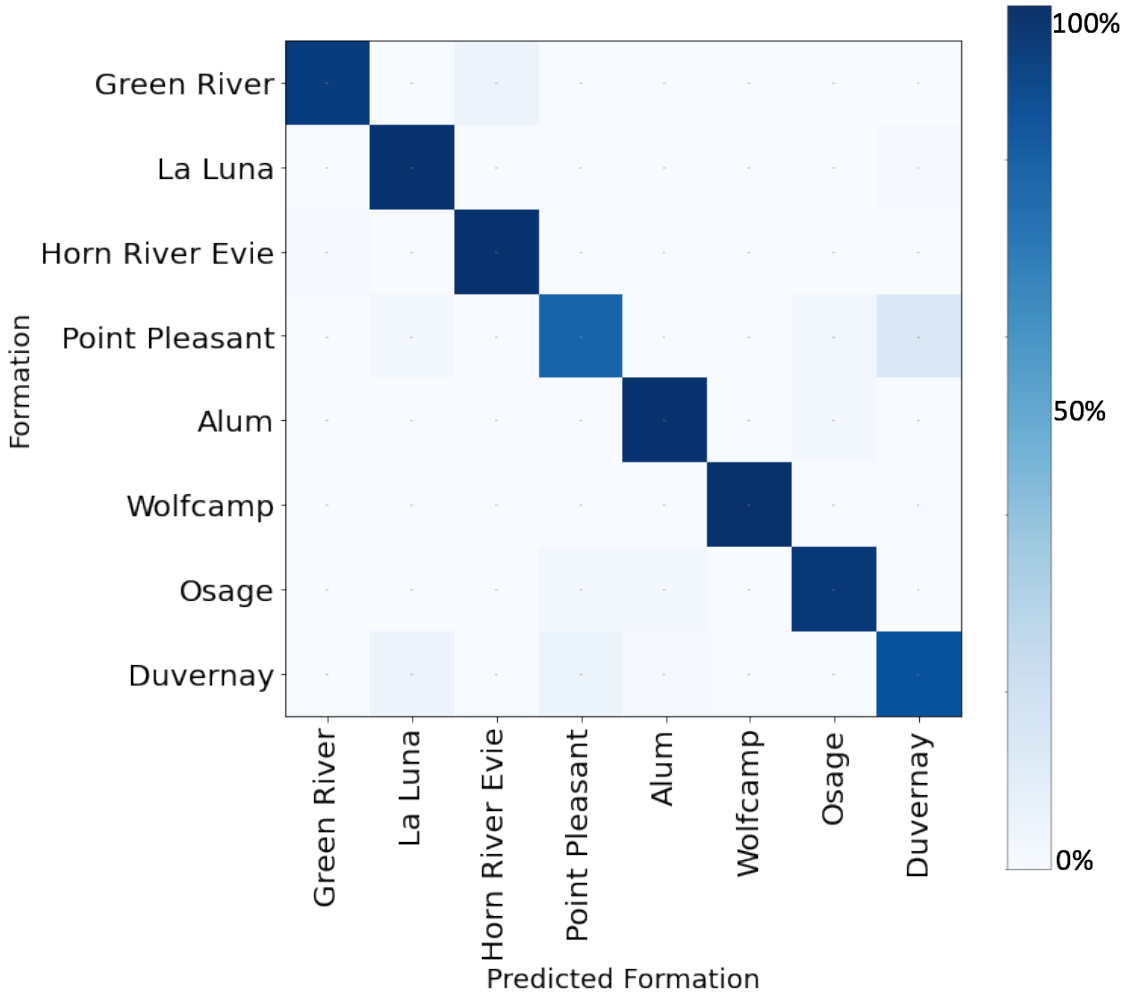


Fig. 2.14 - Confusion matrix of the 2-layer 8, 8-filters network trained on 25nm/px resolution dataset achieves a total accuracy of 91%

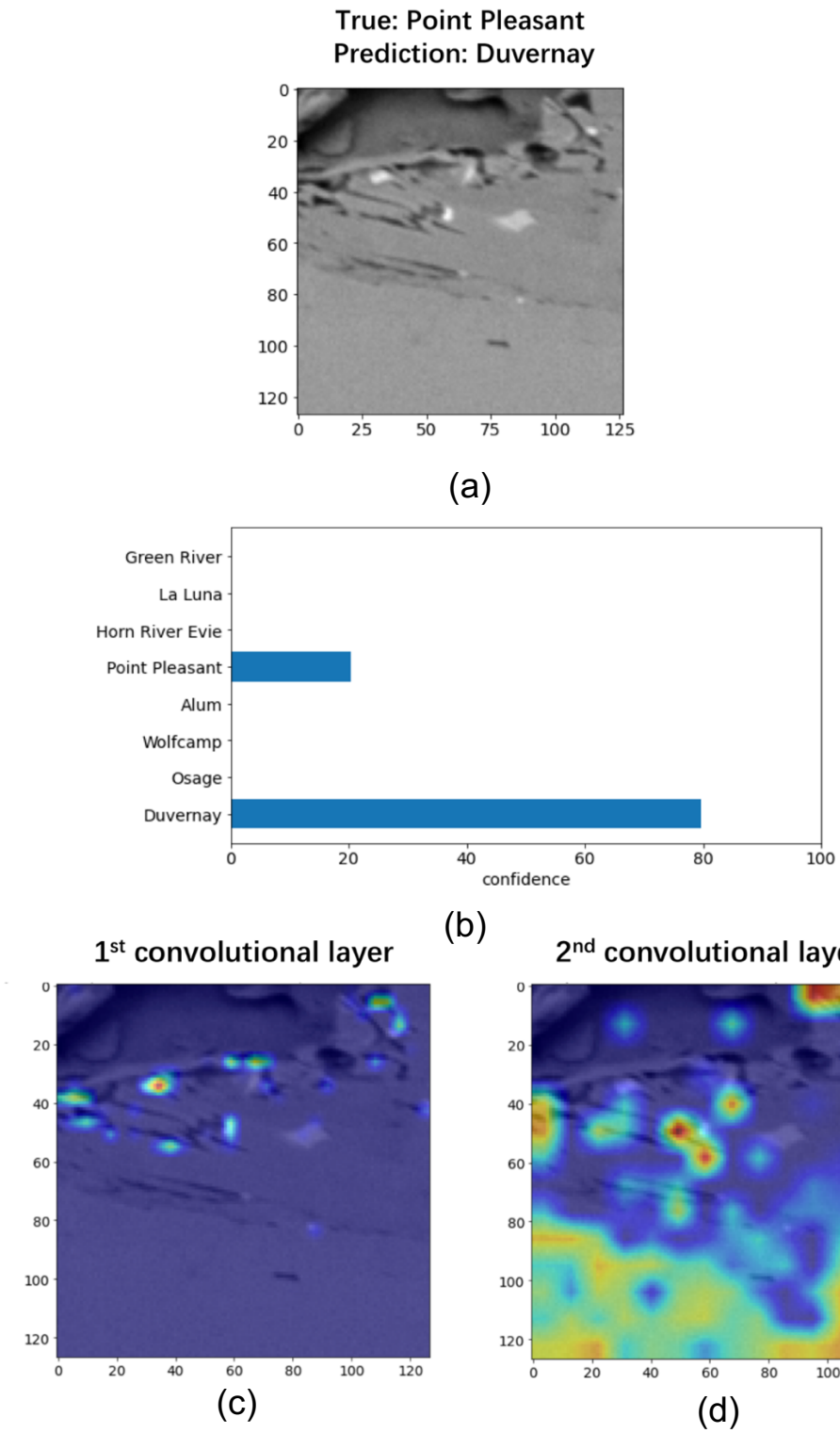


Fig. 2.15 - (a) The Point Pleasant image input to the 2-layer 8, 8-filters networks is misclassified as a Duvernay sample, (b) the model predicts the image with ~80% probability being a Duvernay sample, (c) heatmap output from the 1st convolutional layer, (d) heatmap output from the 2nd convolutional layer.

I show a misclassified Point Pleasant sample and the corresponding heatmaps obtained from the 2-layer 8, 8-filters trained CNN in **Fig. 2.15.** The network predicts this Point Pleasant

image as a Duvernay sample with over 80% probability as shown in **Fig. 2.15(b).** The heatmaps obtained using the same Point Pleasant image are shown in **Fig. 2.15(c).** The heatmap from the 1st convolutional layer shows grayscale contrast being picked as features which are then fed to the 2nd convolutional layer. The choices are less intuitive now as shown in **Fig. 2.15(d)**. Nevertheless, a few Point Pleasant samples being misclassified as Duvernay samples indicates some similarity in the microstructure.

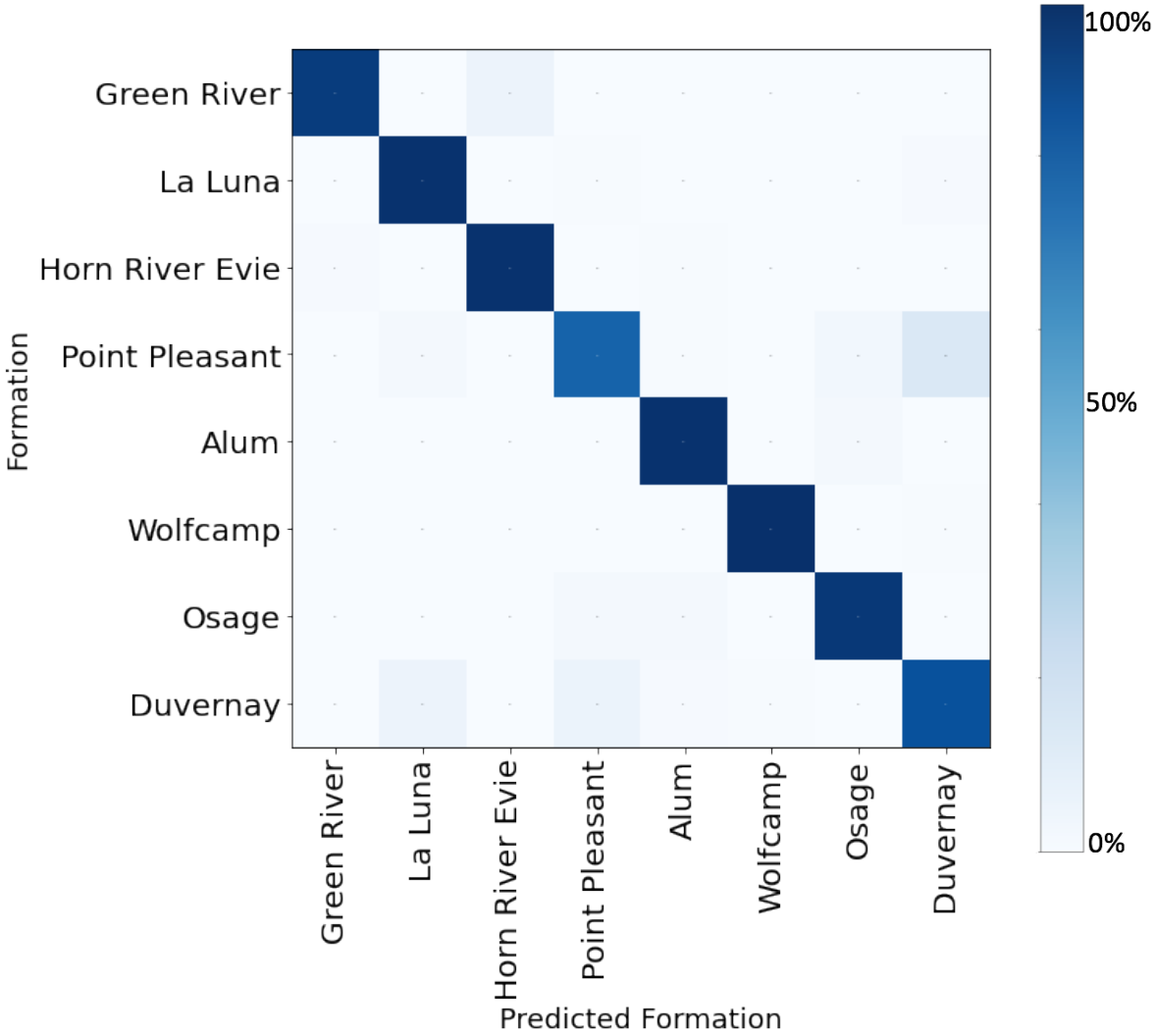


Fig. 2.16 - Confusion matrix of the 5-layer 32, 64, 64, 96, 96-filters network trained on 25nm/px resolution dataset achieves a total accuracy of 95%.

In contrast, the deepest and broadest 5-layer 32, 64, 64, 96, 96-filters model achieves 95% total accuracy, with in excess of 90% recall for each formation except for a small

percentage of the Point Pleasant images, which again are misclassified as Duvernay samples, resulting in 80% recall as shown in **Fig. 2.16**. The modest and deep networks both misclassify several Point Pleasant samples as Duvernay, indicating these two formations may have similar microstructure.

As discussed previously, I show a misclassified Point Pleasant sample and its heatmaps obtained from the 5-layer 32, 64, 64, 96, 96-filters trained CNN in **Fig. 2.17.** The network predicts this Point Pleasant sample as Duvernay with 80% probability as shown in **Fig. 2.17(b).** I also show the heatmaps obtained using the same Duvernay image. In the shallow 1st convolutional layer, the trained CNN picks the lightest shaded features in the original grayscale image as shown in **Fig. 2.17(c)**. These features from the 1st layer combine into the next few convolutional layers to capture larger-scale features. In the 3rd layer, the organic matter appears to be an important feature aiding classification as shown in **Fig. 2.17(d)**. All the features then combine into the last convolutional layer (the 5th layer) to pick large-scale features including organic matters to make the prediction as shown in **Fig. 2.17(e)**.

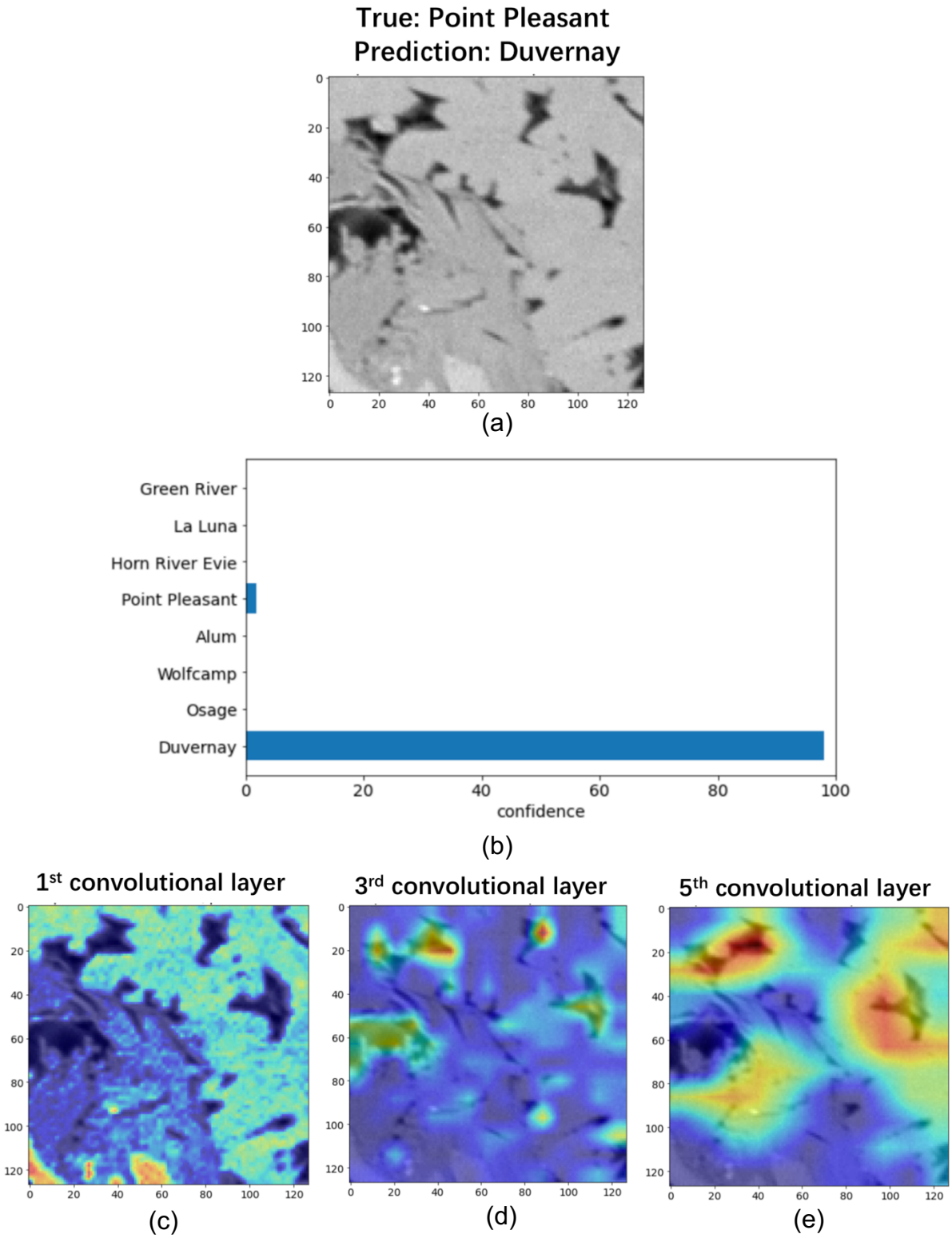


Fig. 2.17 - (a) The Point Pleasant image input to the 5-layer 32, 64, 64, 96, 96-filters network is misclassified as a Duvernay sample, (b) the model predicts the image with over 95% probability being a Duvernay sample, (c) heatmaps output from the 1st convolutional layer, (d) heatmap output from the 3rd convolutional layer, (e) heatmap output from the 5th convolutional layer.

2.6 25nm/px Resolution: Shallow vs. Deep Network Performance

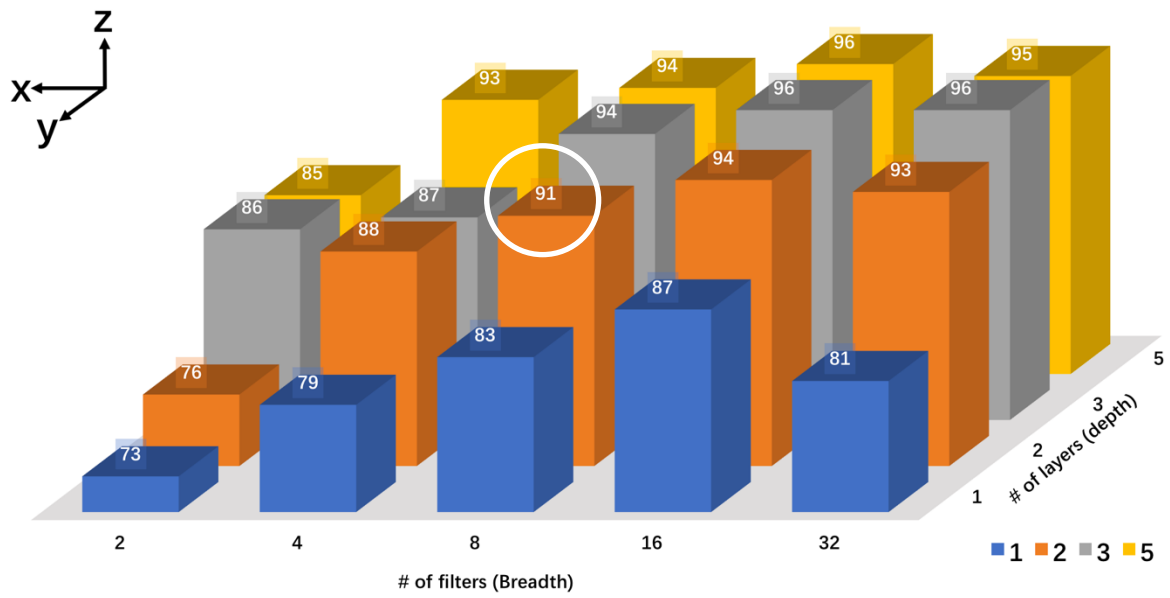


Fig. 2.18 - 3D bar chart showing the accuracy of the CNNs in varying depth and breadth. The 2-layer 8, 8-filters CNN in the white circle is sufficient for the dataset at 25nm/px resolution, which provides over 90% accuracy.

The corresponding 3D bar chart of the testing results for 25nm/px resolution (3x3µm field-of-view) is shown in **Fig. 2.18**. The x-axis refers to the number of filters (breadth), the y-axis refers to the number of layers (depth), and the z-axis refers to the accuracy of each network. We can observe that a 2-layer 8, 8-filters CNN provides a very satisfactory accuracy of over 90%for the dataset at 25nm/px resolution when classifying eight formations.

Besides, we can observe that a simple 1-layer 4-filter is capable of achieving ~80% accuracy; however, subsequent increases to 8, 16, and 32 filters in the 1-layer model do not substantially enhance the accuracy. It indicates that increases in filter width/diversity do not provide measurable benefits beyond a specific limit. In contrast, an increase in depth from 3 layers to 5 layers does not enhance the accuracy either also demonstrating that there are limits to filter performance when going to deeper networks. I did not observe any symptoms of overfitting in this exercise, likely because the dataset I have available is large and diverse.

Moreover, when the number of filters is limited, a deeper network such as the 5-layer 2-filter network fails to achieve (an arbitrary but reasonable benchmark) of 90% accuracy. In the next chapter, I expand the number of classes to 22 to test whether increasing the number of classes requires additional breadth (number of filters) and depth (number of layers).

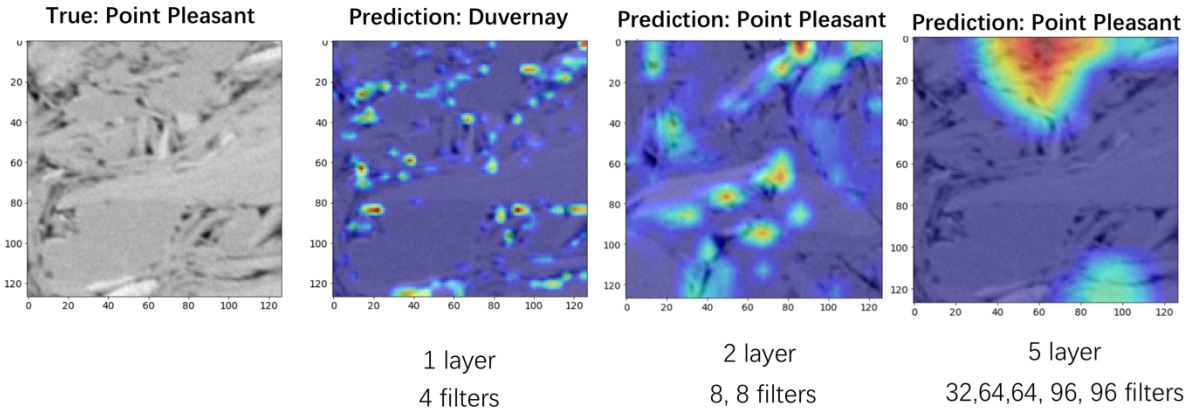


Fig. 2.19 - Heatmaps output from the 1-layer, 2-layer, and 5-layer CNNs.

There are a few cases where the shallower networks misclassify an image while the deeper networks correctly identify the source of the image. **Fig. 2.19** is such an example where the 1-layer 4-filters networks incorrectly misclassify a Point Pleasant sample as a Duvernay sample. The 2-layer 8, 8 filters network and the 5-layer 32, 64, 64, 96, 96 filters network, on the other hand, both provide the correct answer.

2.7 Comparison between the 25nm/px and the 10nm/px with 8 Formations

In this section, I test another dataset of grayscale SEM image at 10nm/px resolution from the same 8 formations: Green River, La Luna, Horn River Evie, Point Pleasant, Alum, Wolfcamp, Osage, Duvernay. The images are reshaped to 127x127 pixels without overlap, equivalent to a 1x1 μm field-of-view as shown in **Fig. 2.20**. My rationale for this experiment is to test the sensitivity of CNN complexity to the field-of-view. With higher resolution, the field-of-view is restricted when using the same number of pixels. It is possible that larger-scale features may be missing, thereby compromising classification.

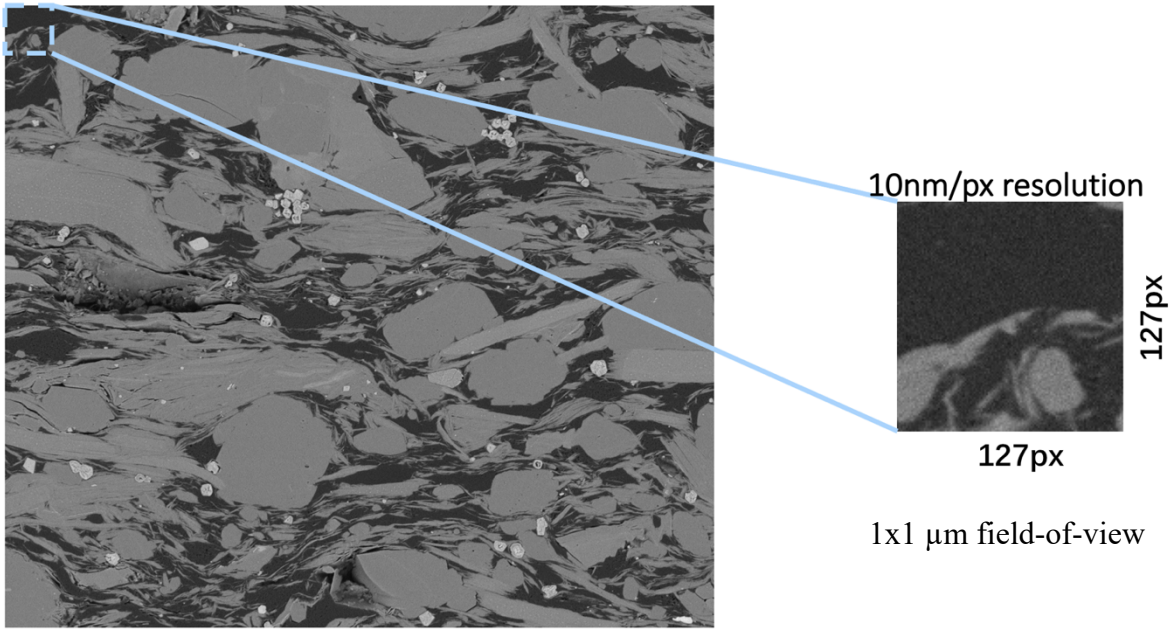


Fig. 2.20 - Example of a raw SEM image from Alum with 10nm/px resolution. It is sliced to 127x127 pixels size (1x1 µm field-of-view) to fit into the model. The left figure is the raw image; the right figure is an example of the rescaled and sliced images for CNN model training.

Table. 2.6 shows the detailed information of the input images from each play with 10nm/px resolution. I use the same number of images as with the previous study at 22nm/px for training, validation, and testing: 840 images per formation and a total of 6720 images for training; 180 images per formation and a total of 1440 images for validation; and 180 images per formation and a total of 1440 images for testing. Twenty example images from the dataset are shown in **Fig. 2.21**.

Play	Resolution (nm)	Bit Depth	# of images for training	# of images for validation	# of images for testing
Wolfcamp	10	8	840	180	180
Alum	10	8	840	180	180
Duvernay	10	8	840	180	180
Osage	10	8	840	180	180
La Luna	10	8	840	180	180
Point Pleasant	10	8	840	180	180
Green River	10	8	840	180	180
Horn River Evie	10	8	840	180	180

Table. 2.6 - 10nm/px resolution (1x1 µm field-of-view) SEM image information of 8 plays.

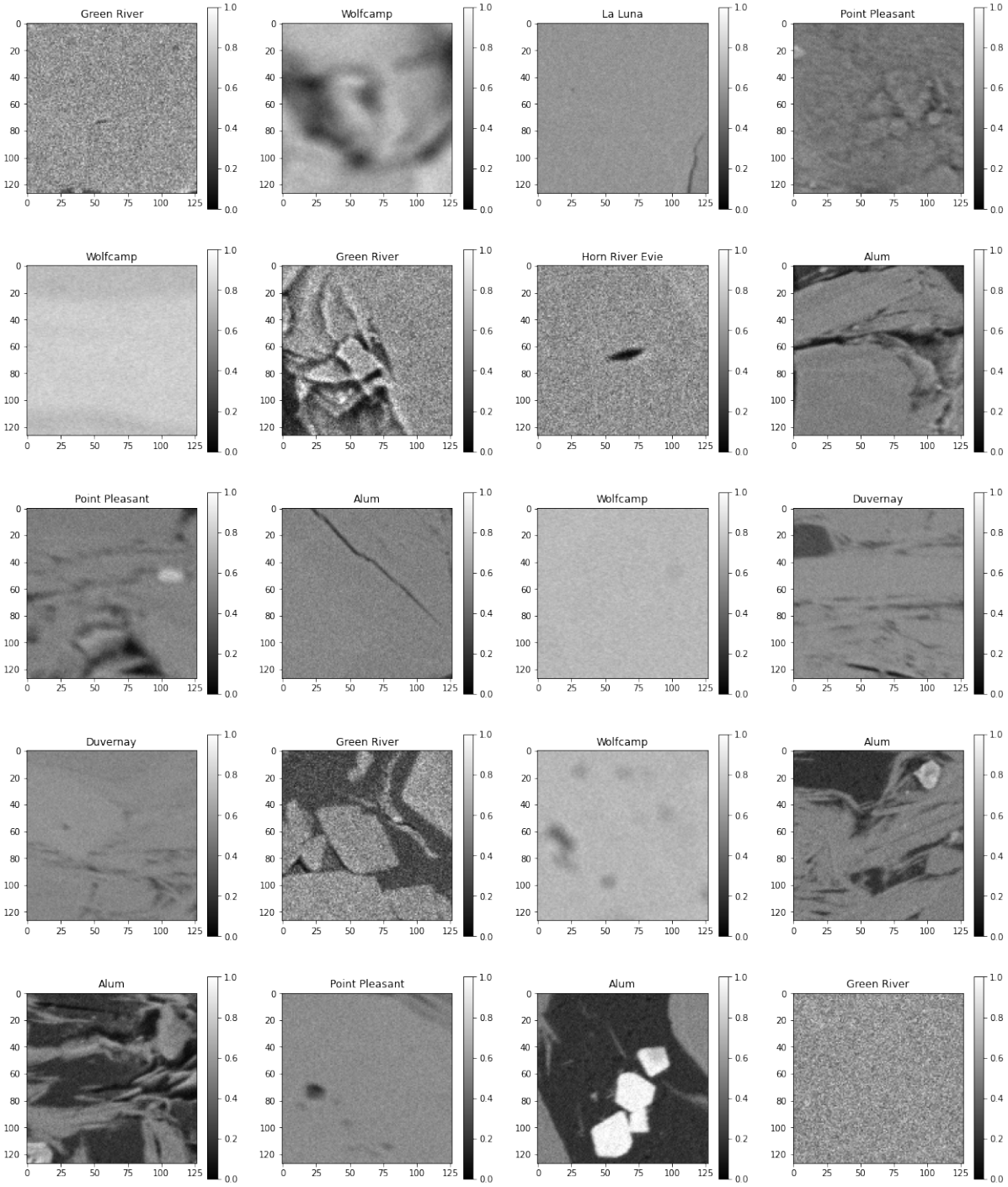
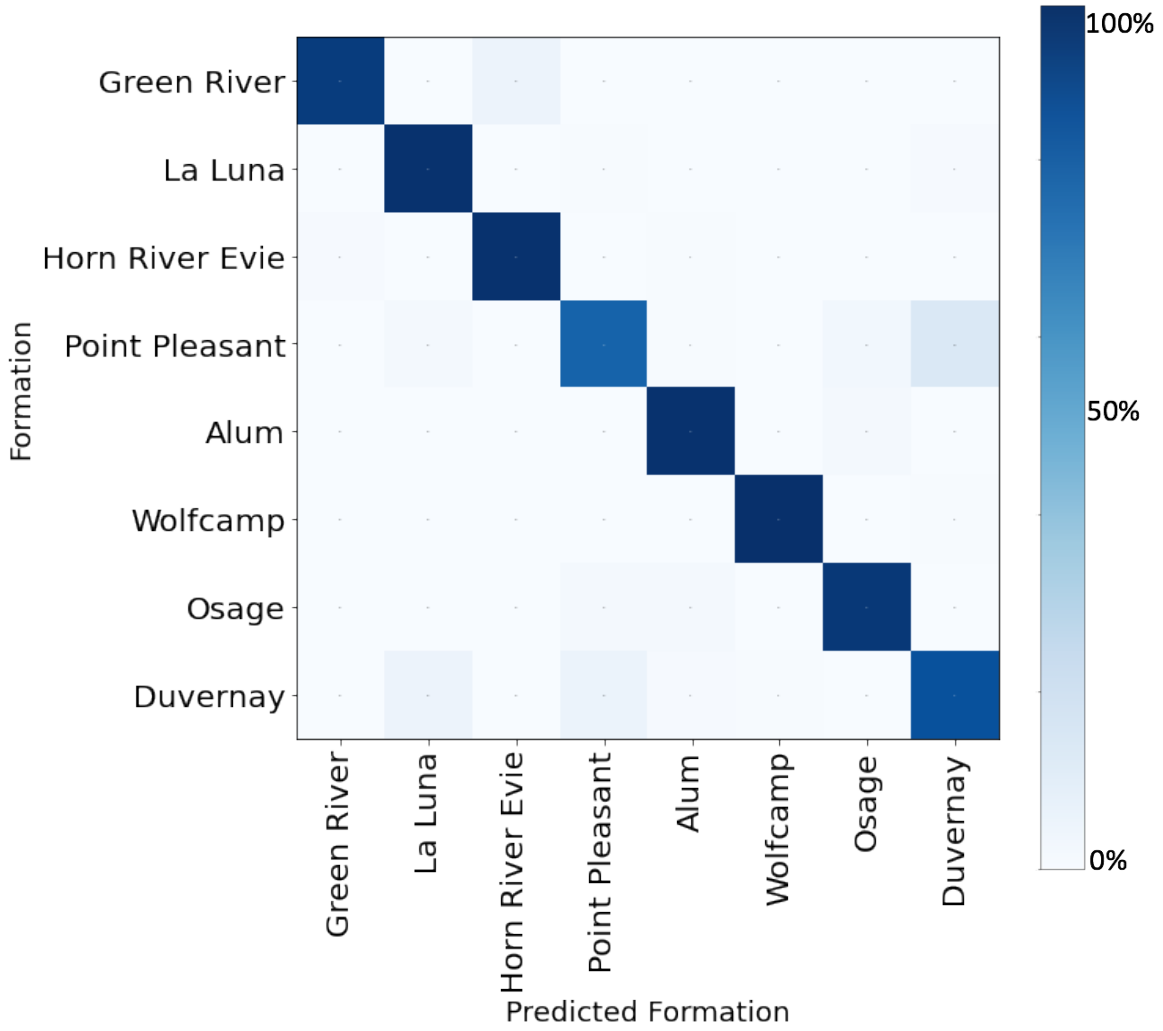


Fig. 2.21 - Twenty grayscale SEM images from 8 plays for play identification at 10nm/px resolution 127x127 pixel size (1x1 μm field-of-view).

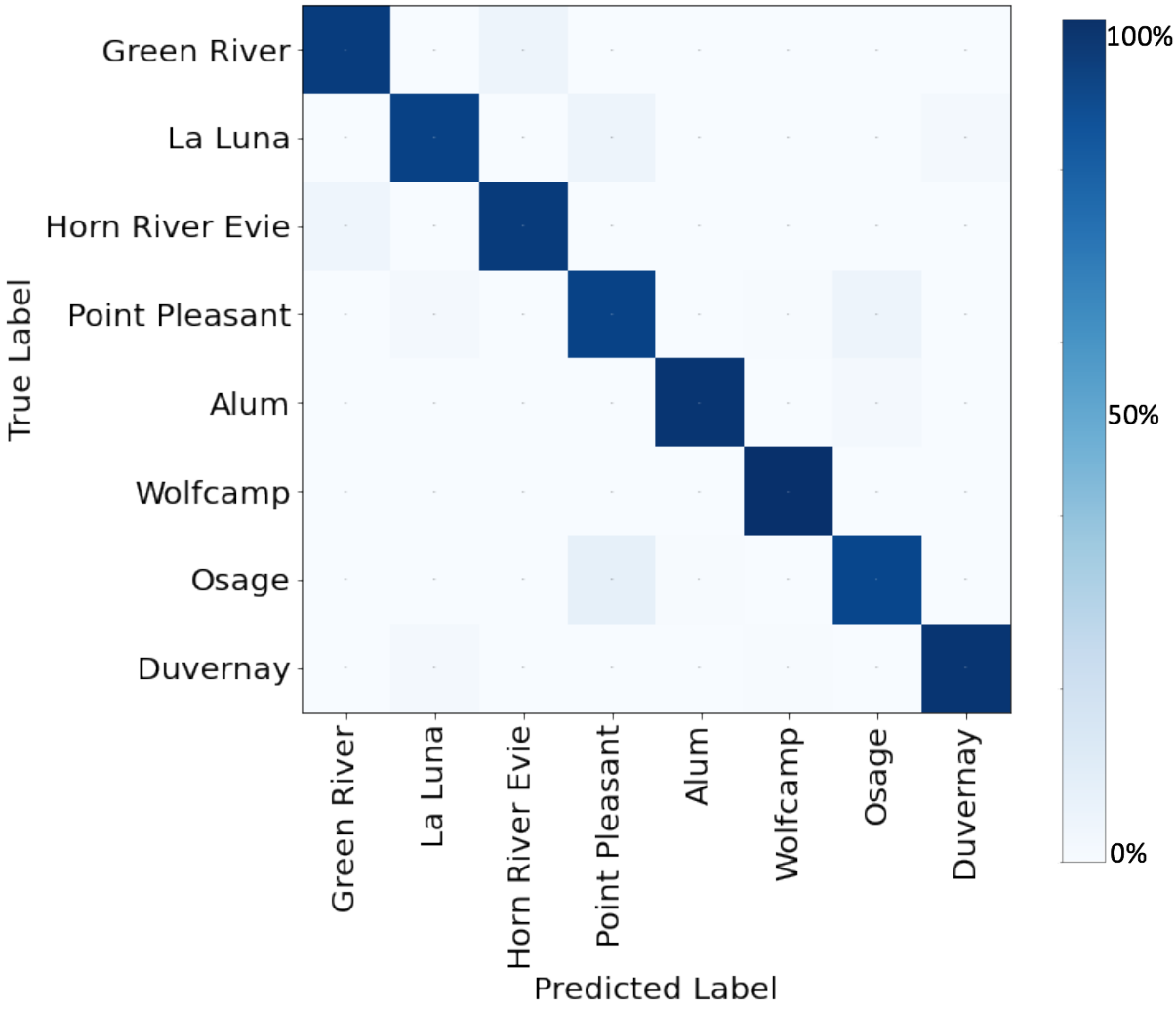
In the interests of brevity, I focus on a few interesting results. I compare the confusion matrices of two models that achieve high accuracy on both 25nm/px and 10nm/px resolution datasets. **Fig. 2.22** shows the corresponding confusion matrices of the modest 2-layer 8, 8-

filters models trained on 25nm/px and 10nm/px resolution dataset separately for total accuracy of 91% and 95%, respectively. Unexpectedly, the rate of misclassification of Point Pleasant samples as Duvernay samples, which was quite common at 25nm/px resolution, is significantly reduced at 10nm/px resolution. The only possible explanation for this is that the significant features in these formations are close to 10nm resolution that get blurred or merge with the background at a lower resolution of 25nm/px.

In contrast, the network trained on the 10nm/px resolution dataset shows a higher misclassification rate between Osage samples and Point Pleasant samples. However, the network trained on the 25nm/px resolution dataset does not misclassify these formations, perhaps indicating that there are larger features in the two formations that are not captured with a 1x1 μm field-of-view. The network, however, misclassifies a few of the Green River samples as Horn River Evie samples at both 25nm/px and 10nm/px resolutions, indicating the potential similarity of the microstructures in this formation pair.



(a) 25nm/px resolution

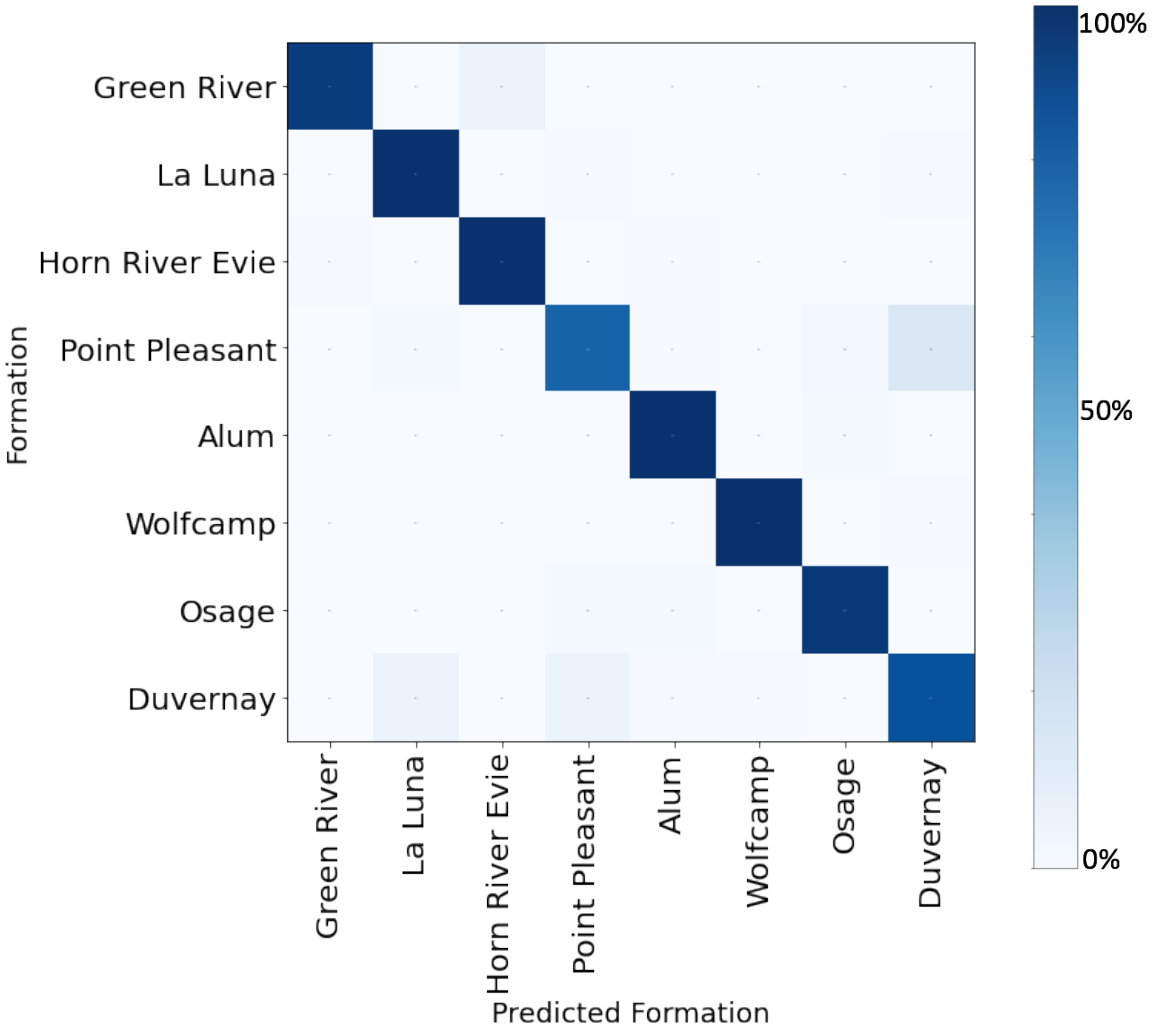


(b) 10nm/px resolution

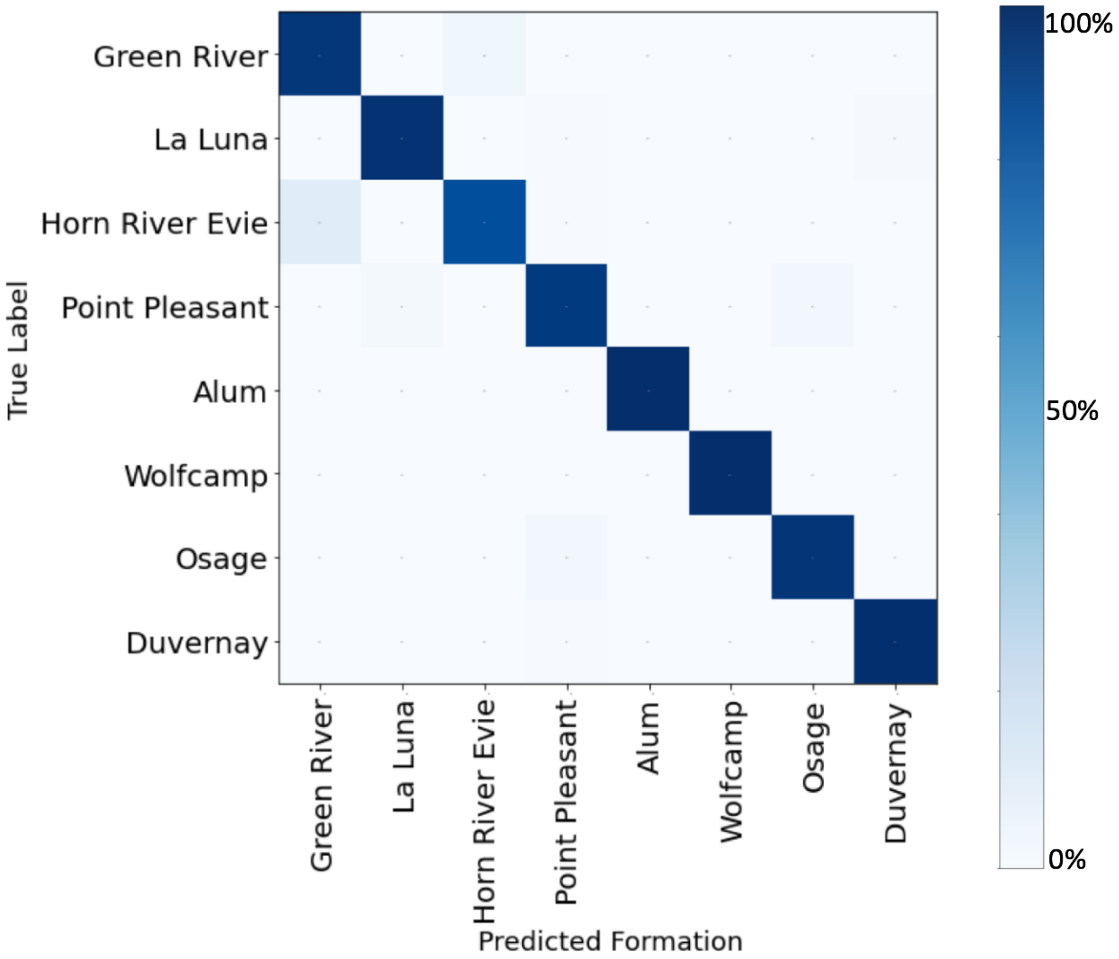
Fig. 2.22 - Comparison of confusion matrix corresponding to the 2-layer 8, 8-filters CNN trained on (a) 25nm/px resolution dataset achieves a total accuracy of 91%, (b) 10nm/px dataset achieves a total accuracy of 95%.

I also compare the corresponding confusion matrices of the deep 5-layer 32, 64, 64, 969, 96-filters models trained on 25nm/px and 10nm/px resolution datasets separately as shown in **Fig. 2.23** with a total accuracy of 95% and 97%, respectively. Both networks achieve high recall on La Luna, Alum, and Wolfcamp samples underscoring their unique microstructure at the higher and lower resolutions. In contrast, both networks misclassify a few Green River samples as Horn River Evie samples, and at the 10nm/px resolution, the network also

misclassifies a few Horn River Evie samples as Green River samples, indicating that the important features that aid classification are larger than the 1x1 μm field-of-view. On the other hand, as observed in modest network, the 25nm/px network misclassifies several Point Pleasant samples as Duvernay, while the 10nm/px does not suffer from the same rate of misclassification.



(a) 25nm/px resolution



(b) 10nm/px resolution

Fig. 2.23 - Comparison of confusion matrix corresponding to the 5-layer 32, 64, 64, 96, 96-filters CNN trained on (a) 25nm/px resolution dataset achieves a total accuracy of 95%, (b) 10nm/px resolution dataset achieves a total accuracy of 97%.

The examples above show a high rate of misclassification between the Green River and Horn River Evie samples. I show how a 5-layer 32, 64, 64, 96, 96-filter network performs on a few select images from these plays. **Fig. 2.24(a)** is a Green River sample correctly predicted by the network. **Fig. 2.24(b)** is another Green River sample incorrectly predicted with high confidence as a Horn River Evie sample. **Fig. 2.24(c)** is a Horn River sample misclassified by the network as a Green River sample, with lower confidence. It is important to note that all three of these images are characterized by white noise, which could be a feature that the CNN is capturing as a significant feature of the image which results in misclassification. There are

potentially two methods to address this: One is to add white noise to all images to allow the CNN to focus on unique microstructural features to add classification or alternatively, de-noise the images. De-noising however can remove some important features if they are at the scale at which the noise is present. I however did not test these effects in this work.

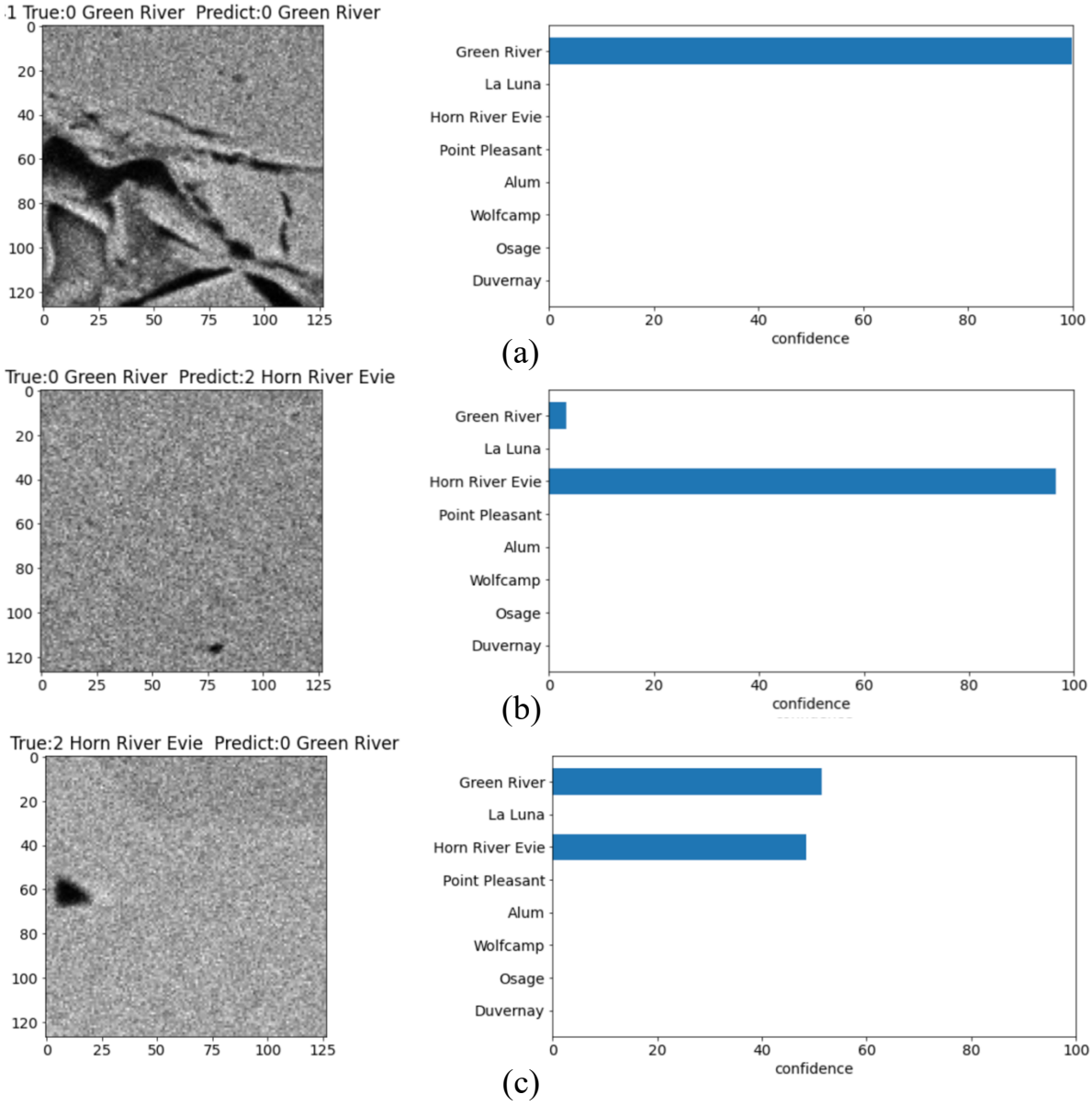


Fig. 2.24 - Green River and Horn River Evie samples test on 5-layer 32, 64, 64, 96, 96-filters. (a) Green River sample correctly classified by the network. (b) Green River sample misclassified by the network. (c) Horn River Evie sample misclassified by the network.

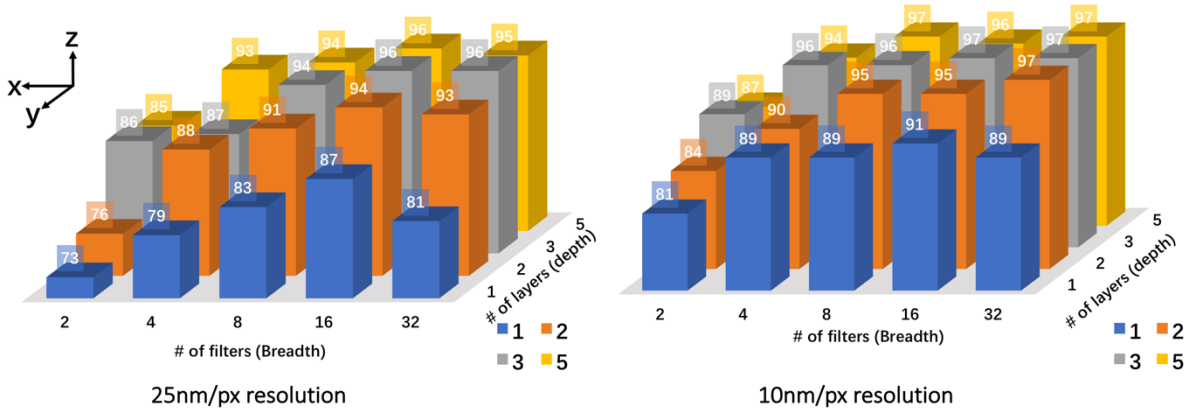


Fig. 2.25 - Comparison of the accuracy for each model trained on 25nm/px resolution dataset and 10nm/px dataset.

More detailed test results for each model at 10nm/px resolution dataset with 8 formations are shown in **Appendix Fig. A2** and **Fig. A3**.

I compare the test results for the 25nm/px resolution (3x3 µm field-of-view) dataset and the 10nm/px resolution (1x1 µm field-of-view) dataset for all models as shown in **Fig. 2.25**. As mentioned earlier, the x-axis refers to the number of filters (breadth), the y-axis refers to the number of layers (depth), and the z-axis refers to the accuracy of each network. As shown in **Fig. 2.25**, the models trained on 10nm/px resolution (1x1 µm field-of-view) dataset appear to be superior to the model trained on 25nm/px resolution (3x3 µm field-of-view) dataset, indicating that the significant features may get blurry/merged at the 25nm/px resolution, but are resolvable at a smaller field-of-view (10nm/px resolution). For both datasets with eight plays, a 2-layer 8,8 filters network is sufficient, which provides a very satisfactory accuracy of over 90%.

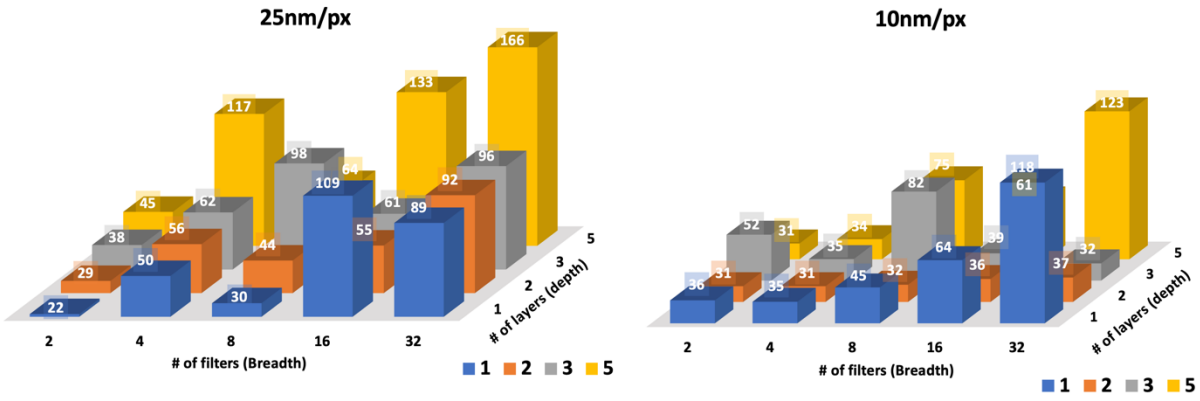


Fig. 2.26 - Comparison of the training time in seconds for each model trained on 25nm/px dataset and 10nm/px resolution dataset.

I also show the training time for the 25nm/px resolution (3x3 µm field-of-view) dataset and the 10nm/px resolution (1x1 µm field-of-view) dataset for all models as shown in **Fig. 2.26**. The x-axis refers to the number of filters (breadth), the y-axis refers to the number of layers (depth), and the z-axis refers to the training time of each network in seconds. For both resolutions, the deepest and broadest network takes the longest time to train, even for the modestly sized dataset used in this study. These differences will grow rapidly as the size of the image dataset increases. For example, for the ImageNet database with ~14 million images (Yang et al., 2019), a deeper network will require greater training time with the need for several more batches to be fed to the network at each epoch during training.

From the results on 25nm/px resolution (3x3µm field-of-view) and 10nm/px resolution (1x1µm field-of-view), I can conclude that:

- For both datasets, a network with 2-layers and 8 filters on each of the two layers provides acceptable accuracies as shown in **Fig. 2.27.**
- This experiment shows that both depth and breadth are essential. Increases in one without increasing the other do not lead to improved accuracy of classification.
- At both 10nm/px and 25nm/ px resolution, a diversity of filters is essential. The additional filters are shown to extract additional features to make a prediction.

- For the images tested, the 10nm/px resolution images outperform the 25nm/px resolution images, which means important features can still be captured at 10nm/px resolution.
- Significant features of Green River samples and Horn River Evie samples are unresolvable at both 10nm/px resolution and 25nm/px resolution, confounding both the modest and deep CNNs.

With this background, in the next chapter, I perform the same tasks but with images from 22 plays.

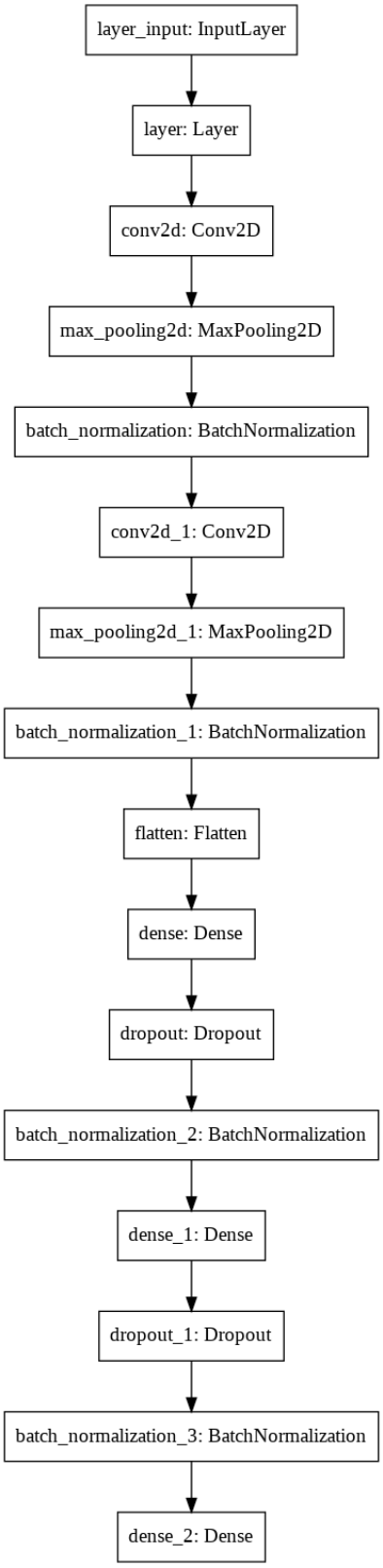


Fig. 2.27 - 2-layer 8, 8-filters network architecture.

Chapter 3: SEM Image Classification for a Dataset Comprising 22 Formations

In this chapter, I adopt a similar approach as described in Chapter 2 and investigate the sensitivity of filter depth and breadth in labeling SEM images. In order to see if model complexity depends on the number of labels, this chapter focuses on image classification for 22 plays.

3.1 Description of the Systematic Approach

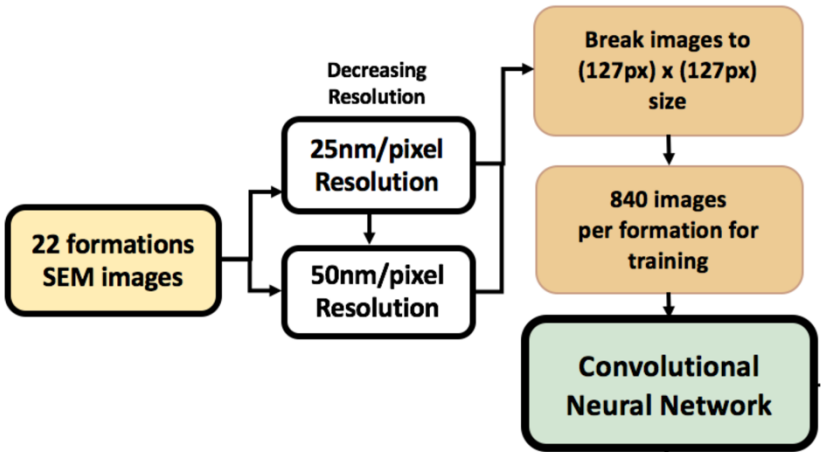


Fig. 3.1 - Systematic approach of testing depth and breadth sensitivity using datasets with 22 plays.

I create a systematic workflow to test the depth and breadth sensitivity on the complete image database, as shown in **Fig. 3.1**. The image database includes over 8000 grayscale SEM images from 22 different unconventional plays worldwide, as shown in **Table 3.1**.

Play	Resolution (nm)	Bit Depth	# of images
Haynesville	25	8	144
Wolfcamp	10	8	799
Alum	10	16	900
Montney	25	8	144
Eagle Ford Oil	25	8	144
Eagle Ford Gas	25	8	144
Avalon/Leonard	25	8	144
Vaca Muerta Oil	25	8	144
Vaca Muerta Gas	25	8	144
Duvernay	10	16	900
Osage	10	16	900
La Luna	10	16	275
Kimmeridge	25	8	144
Point Pleasant	10	16	900
Green River	10	16	400
Horn River Evie	10	16	400
Collingwood	25	8	57
Marcellus	20	8	1480
Woodford	20	8	100
Utica	25	8	144
Niobrara	25	8	144
Barnett	25	8	144

Table. 3.1 - SEM image information including resolution, bit depth, and the number of images in each play for entire database. As mentioned earlier, the bit-depth is the number of bits used to symbolize the color of a single-pixel.

The images are normalized to 8-bit depth and are rescaled to two datasets with varying resolutions: 25nm/px and 50nm/px. In the previous chapter, I considered a 10nm/px resolution. However, at that resolution, I had a sufficient number of images for only 8 plays. The images are then sliced to 127 x 127 pixels without overlap to ensure that the image can be rescaled to various resolutions. At 25nm/px, the field-of-view for an image with 127x127 pixels is 3x3 μm, and at 50nm/px, the field-of-view is 6x6 μm, so I will be investigating the effect of field-of-view as well. A 25nm/px and a 50nm/px image from the exact location are shown in **Fig. 3.2**.

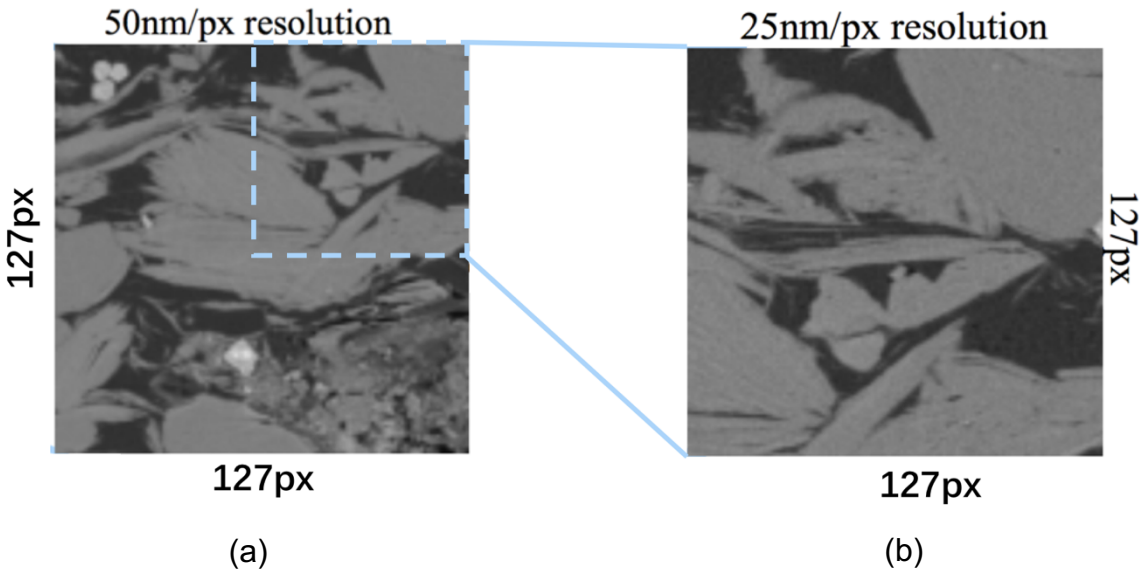


Fig. 3.2 - (a) An example of the 50nm/px resolution 127x127 pixel size (6x6 μm field-of-view) image, (b) an example of the 25nm/px resolution 127x127 pixel size (3x3 μm field-of-view) image.

I use the same number of images per formation as in the previous chapter for training and validation: 840 images per play and a total of 18480 images for training; 180 images per play and a total of 3960 images for validation. The remaining images for each play were used for testing.

3.2 Models Considered in this Chapter

# of Layers	1	2	3	5
# of Filters	1	--	--	--
# of Filters	2	2, 2	2, 4, 4	2, 4, 4, 8, 8
# of Filters	4	4, 4	4, 8, 8	4, 8, 8, 16, 16
# of Filters	8	8, 8	8, 16, 16	8, 16, 16, 32, 32
# of Filters	16	16, 16	16, 32, 32	16, 32, 32, 48, 48
# of Filters	32	32, 32	32, 64, 64	32, 64, 64, 96, 96

Table. 3.2 - CNN architecture tested using SEM images from the 22 formations at 50nm/px and 25nm/px resolutions.

I use the same machine learning environment as in the previous chapter with TensorFlow (Abadi et al., 2015) as the machine learning platform running on Google Colaboratory

environment (Bisong, 2019). I follow a modified AlexNet architecture but use fewer layers and filters as well as smaller sizes of filters (3x3). Additionally, I remove one of the three fully connected identification layers for simplicity, which results in 4096 neurons in the first dense layer with 10% of the neurons dropped, and 1024 neurons in the second dense layer with 10% of the neurons dropped.

To test the sensitivity of convolutional neural networks to depth and width, I use the same model architectures as described in the previous chapter. The list of models considered in this chapter are shown in **Table. 3.2**.

I start with the simplest 1-layer, 1-filter CNN architecture and evaluate the trained model performance on the test datasets. It is important to note that the number of layers refers to the number of convolutional layers. I then increase the breadth of the network by increasing the number of filters in the 1-layer model, as shown in **Table. 3.2**. I then progressively increase the depth (add a layer to the CNN) and increase breadth (add filters to each new layer) and assess the sensitivity of the network accuracy to both depth and breadth. In addition, I relate image resolution to the CNN architecture. For each CNN architecture listed in **Table. 3.2**, I run multiple tests and chose the best performing accuracy for each model considered.

3.3 50nm/px Resolution SEM Images

In this section, the raw cross-sectioned grayscale SEM images from 22 plays in various pixel sizes were rescaled to 50nm/px resolution and then sliced to 127x127 pixels without overlap, which is equivalent to a 6x6 μm field-of-view as shown in **Fig. 3.3**.

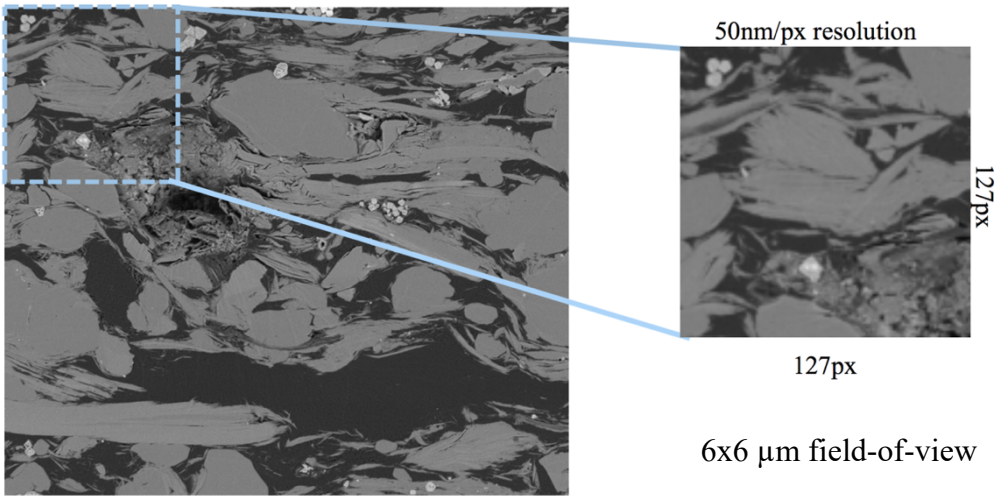


Fig. 3.3 - Example of a raw SEM image from Alum with 10nm/px resolution. It is rescaled to 50nm/px resolution and sliced to 127x127 pixels size (6x6 µm field-of-view) to fit into the model. The left figure is the raw image; the right figure is an example of the rescaled and sliced images for CNN model training.

Table. 3.3 describes the dataset used in this study. As mentioned earlier, I used 840 images per formation for testing and 180 images per formation and a total of 3960 images for validation. Over 100000 images were used for testing. The unprecedented size and diversity of this dataset allows me to investigate CNN model complexity in great detail. Twenty example images from the dataset are shown in **Fig. 3.4**.

Play	Resolution (nm)	Bit Depth	# of images for training	# of images for validation	# of images for testing
Haynesville	50	8	840	180	5892
Wolfcamp	50	8	840	180	8568
Alum	50	8	840	180	4380
Montney	50	8	840	180	5892
Eagle Ford Oil	50	8	840	180	5892
Eagle Ford Gas	50	8	840	180	5892
Avalon/Leonard	50	8	840	180	5892
Vaca Muerta Oil	50	8	840	180	5892
Vaca Muerta Gas	50	8	840	180	5892
Duvernay	50	8	840	180	4380
Osage	50	8	840	180	4380
La Luna	50	8	840	180	630
Kimmeridge	50	8	840	180	5892
Point Pleasant	50	8	840	180	4380
Green River	50	8	840	180	3780
Horn River Evie	50	8	840	180	3780
Collingwood	50	8	840	180	1716
Marcellus	50	8	840	180	16740
Woodford	50	8	840	180	180
Utica	50	8	840	180	5892
Niobrara	50	8	840	180	5892
Barnett	50	8	840	180	5892

Table. 3.3 - 50nm/px resolution (6x6 μm field-of-view) SEM image information of 22 plays.

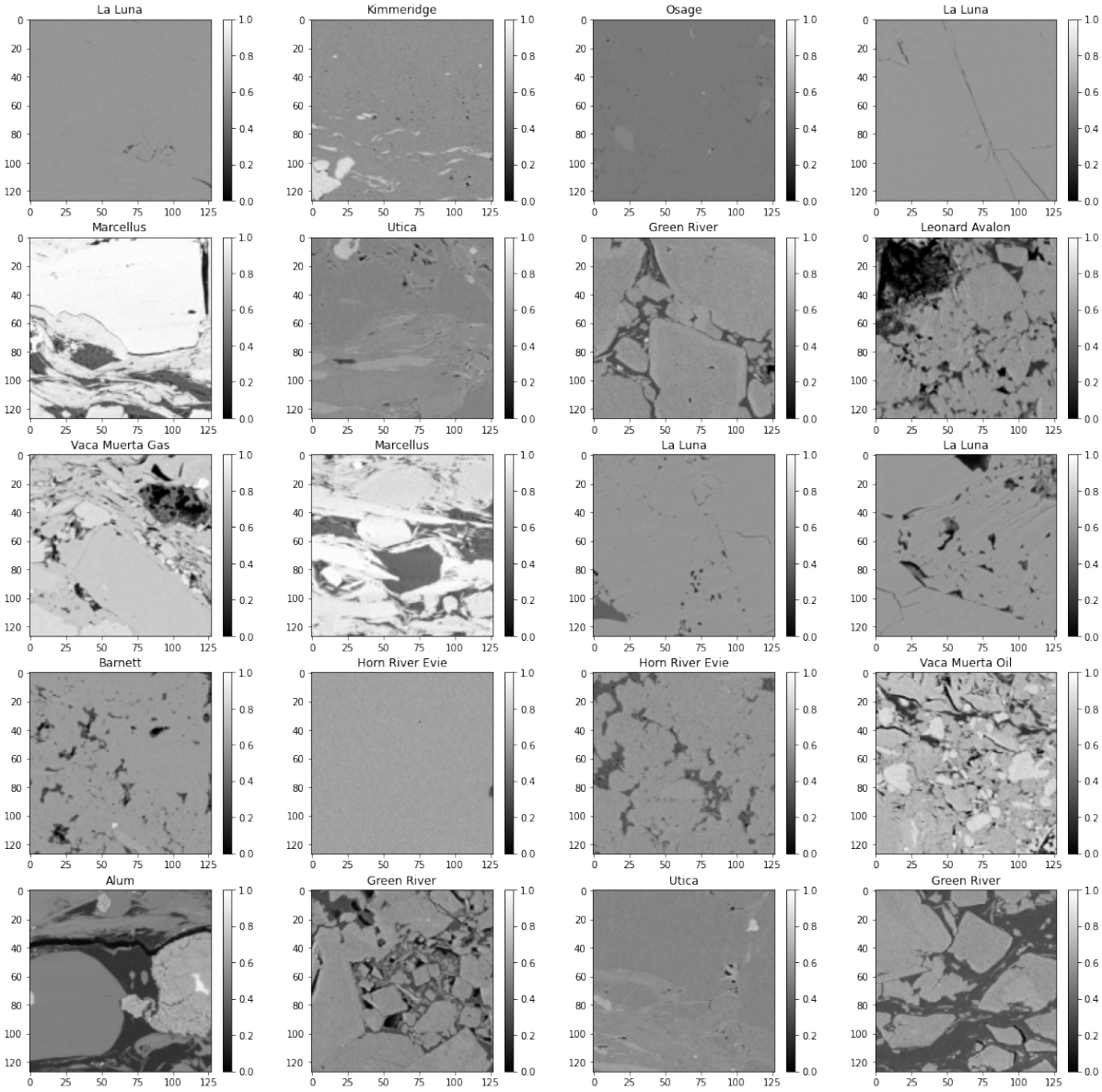


Fig. 3.4 - Twenty grayscale SEM images from 22 plays for play identification at 50nm/px resolution (6x6 μm field-of-view).

3.4 50nm/px Resolution: Shallow Network Results

The model architecture and corresponding test results are shown in **Table. 3.4**. The simple 1-layer 1-filter CNN only achieves 54% accuracy on the 50nm/px resolution dataset. Increasing the breadth in the 1-layer network from 1 filter to 4 filters, the network accuracy goes up to 80%. However, as shown in the previous chapter (**Table. 2.4**), a subsequent increase of breadth

to 8, 16, and 32 filters in the 1-layer model does not substantially enhance the accuracy. It indicates that increases in filter width/diversity does not provide any measurable benefit beyond a certain limit. Moreover, with extremely limited depth (1-layer), even the 1-layer 32-filters model is limited in terms of accuracy, underscoring the need for an increased depth of the network. However, I do want to point out that even the simple 1-layer, 4-filters trained on the 50nm/px resolution with even 22 formations does provide an accuracy of ~80%. This is remarkable because, by pure chance alone, we can obtain a 100/22 = 4.5% chance of being correct. A simple model providing over 80% accuracy demonstrates that deep networks are perhaps not necessary. However, reaching 90% accuracy levels (an arbitrarily chosen benchmark) will necessitate some depth to capture a few higher-level features.

# of Layers	1	1	1	1	1
# of Filters	1	4	8	16	32
Accuracy %	54	77	73	79	81

Table. 3.4 - Accuracy of the 1-layer networks test on the SEM images at 50nm/px resolution.

The confusion matrix corresponding to the 1-layer 1-filter model with total accuracy of 54% is shown in **Fig. 3.5**. The images from the Haynesville and Osage formations are classified with over 90% recall, indicating these two formations have unique small-scale microstructures that can be resolved even with a simple 1-layer 1-filter network. However, in this case, there are also several off-diagonal elements that are more dominant than the diagonal entries. For example, the network has a greater propensity of misclassifying the Horn River Evie samples as Collingwood samples rather than Horn River Evie.

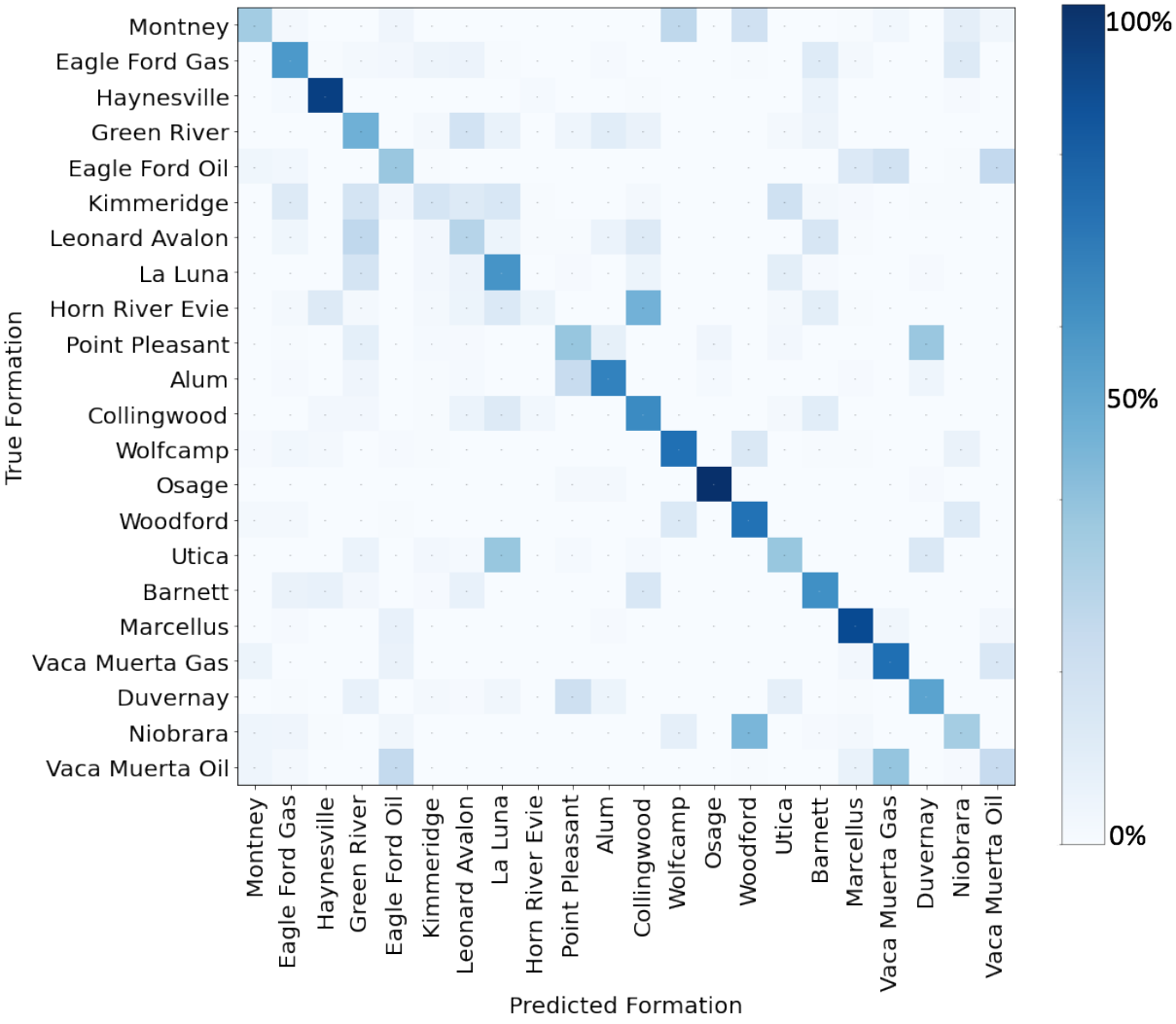


Fig. 3.5 - Confusion matrix of the 1-layer 1-filter network trained on 50nm/px resolution dataset, achieves a total accuracy of 54%.

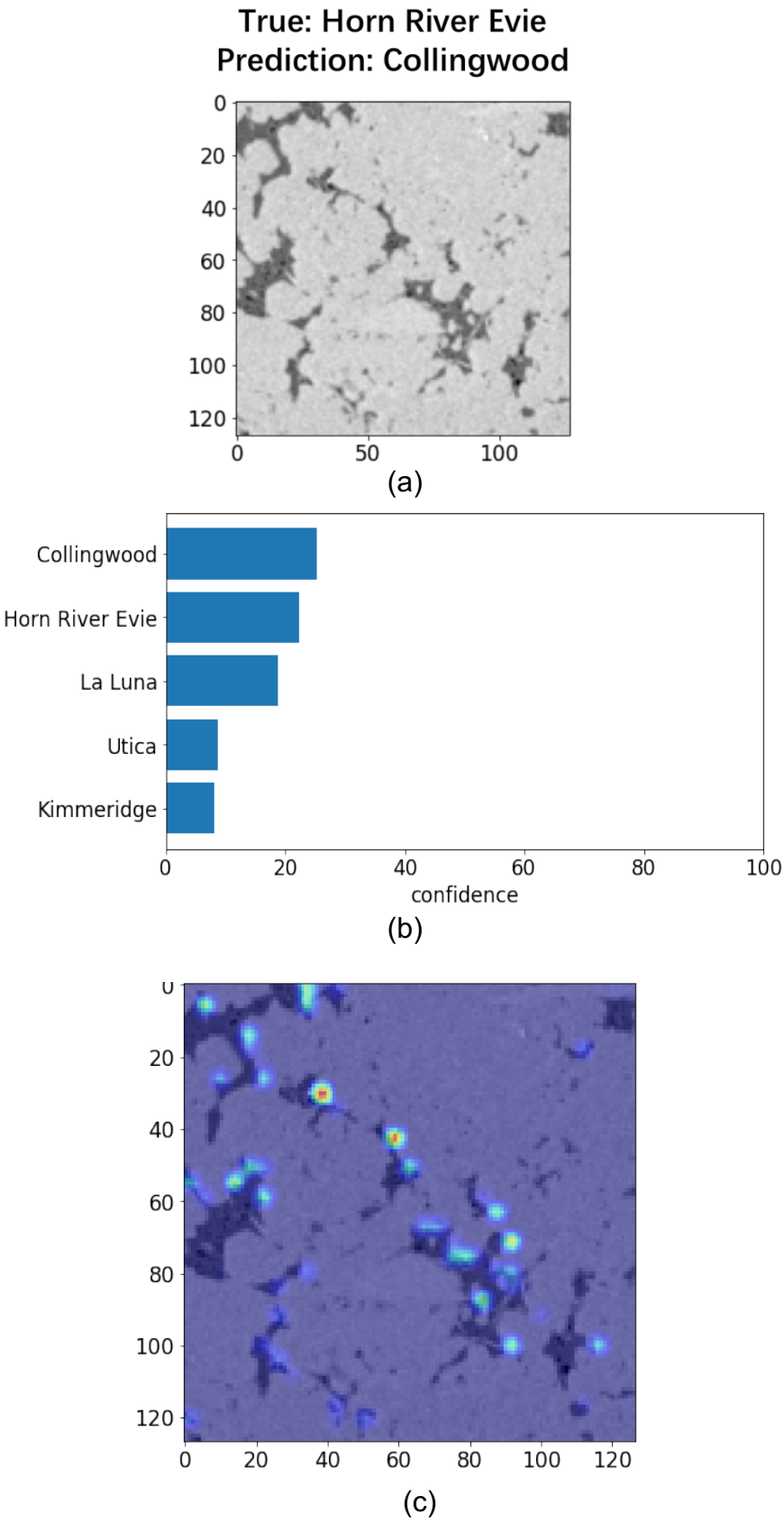


Fig. 3.6 - (a) The Horn River Evie image input to the 1-layer 1-filter network is misclassified as a Collingwood sample, (b) shows the top-5 probability, the model predicts the image with 30% probability being a Collingwood sample (c) heatmap output from the convolutional layer.

I show an incorrectly predicted image sample from the formation pair with the highest misclassification rate with the 1-layer 1-filter trained CNN. This is a sample from the Horn River Evie formation, as shown in **Fig. 3.6(a)**. I show the top-5 probability for the predicted classes on this image. The network predicts this Horn River Evie image as being more likely a Collingwood sample, with a lower probability of it being a Horn River Evie or La Luna sample as shown in **Fig. 3.6(b).** Unfortunately, the probability of it being a Collingwood sample is predicted to be marginally higher and the image is misclassified.

I also show the heatmap obtained using the same Horn River Evie image fed to the 1-layer 1-filter trained CNN showing important features that led to the CNN classification decisions. As mentioned in the last chapter, the hotter color (red and yellow) reveals the most important features for identifying the play, and the colder color (blue) indicates less critical features. As shown in **Fig. 3.6(c)**, in the only convolutional layer, the trained CNN picks around the edges of the organic matter and considers these non-intuitive choices as important features for classification.

The confusion matrix corresponding to the 1-layer 4-filters model is shown in **Fig. 3.7**. The resulting confusion matrix is more diagonally dominant than the 1-layer 1-filter model, indicating that prediction recall for each formation has improved with an increase in breadth. There is a preponderance of Eagle Ford oil window samples being misclassified as Vaca Muerta oil window samples, and a few Point Pleasant samples being misclassified as Alum or Duvernay samples. Again, this is an astonishing result indicating that a very simple, easy to interpret network such as a 1-layer, 4-filters CNN is capable of identifying these SEM images at 50nm/px resolution.

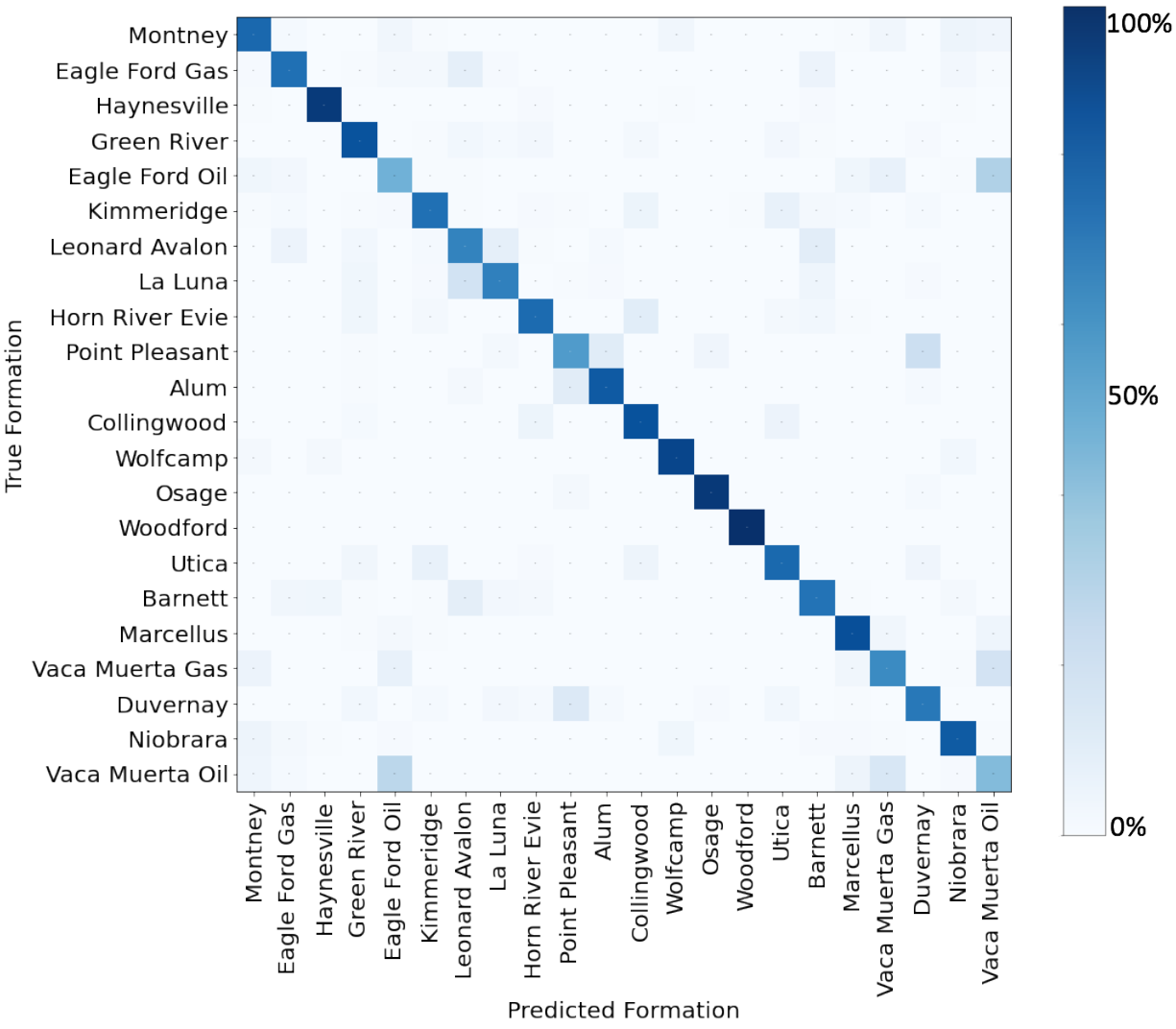


Fig. 3.7 - Confusion matrix of the 1-layer 4-filters network trained on 50nm/px resolution dataset achieves a total accuracy of 77%.

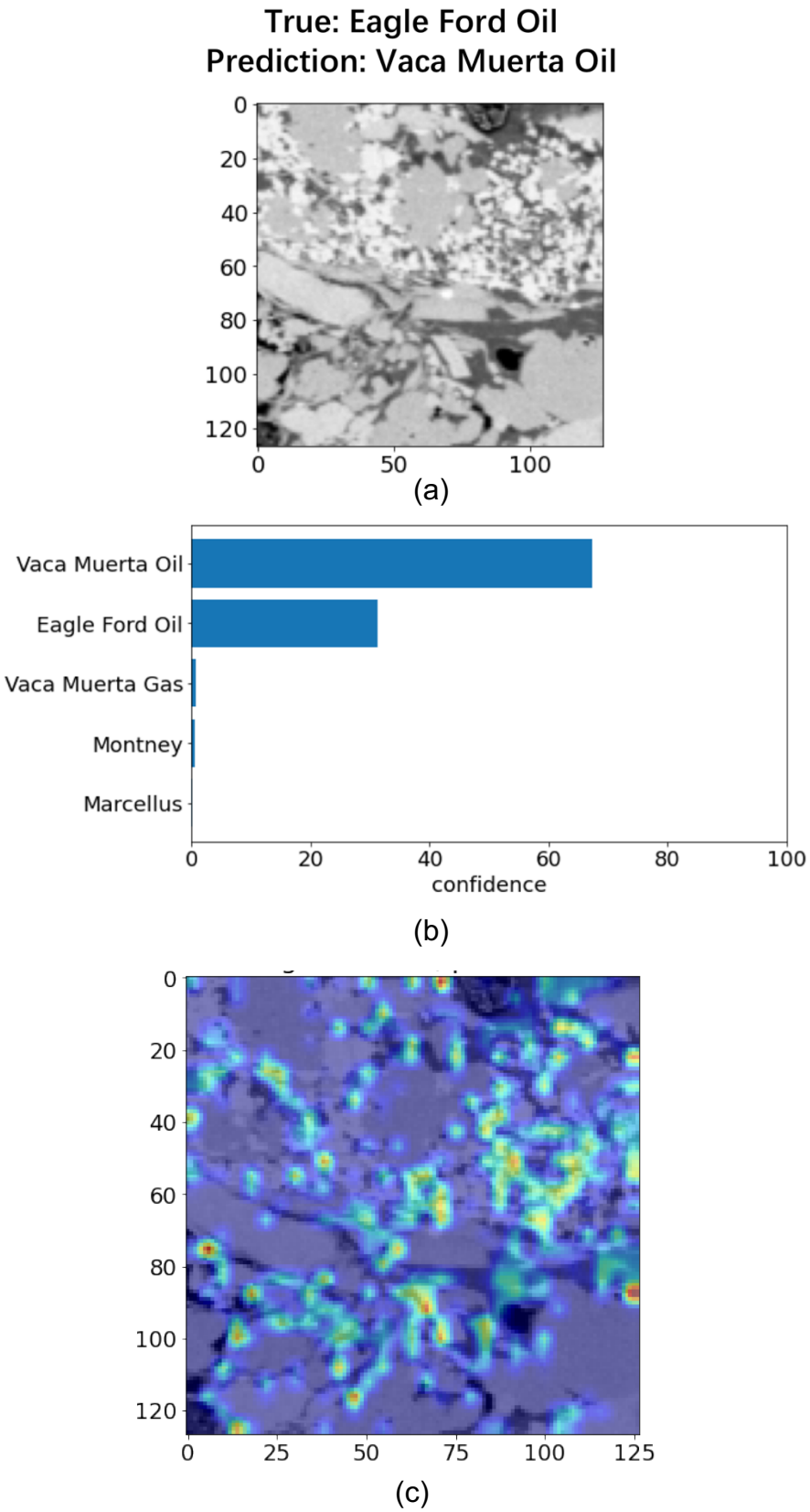


Fig. 3.8 - (a) The Eagle Ford oil image input to the 1-layer 4-filters network is misclassified as a Vaca Muerta oil window sample (b) shows the top-5 probability, the model predicts the image with over 70% probability being Vaca Muerta oil window sample, (c) heatmap output from the convolutional layer.

As I did previously, I show a misclassified image sample from the formation pair with the highest misclassification rate when tested on the 1-layer 4-filters trained CNN. The Eagle Ford oil window sample is shown in **Fig. 3.8(a)**. The network predicts this image with over a 60% probability of being a Vaca Muerta oil window sample with a relatively lower probability of it being an Eagle Ford oil window sample as shown in **Fig. 3.8(b).** This is important because for this specific case of misclassification, the Eagle Ford oil window samples are very similar to the Vaca Muerta samples in terms of mineralogy, porosity, and the distribution and amount of organics (EIA, 2017).

The heatmap obtained using the same Eagle Ford oil window image from the 1-layer 4-filters trained CNN is shown in **Fig. 3.8(c).** With increasing breadth, the network captures additional significant features, which are again non-intuitive choices.

The feature map or filter output obtained from an image fed to a 1-layer 1-filter trained CNN is shown in **Fig. 3.9(b)**. The feature map for the simplest 1-layer 1-filter network appears to extract some grayscale information and edges. However, increasing the width to four filters captures several other features as shown in **Fig. 3.9(c)**. Specifically, in the third panel, all of the inorganic material is considered important, which showcases the organic material very well.

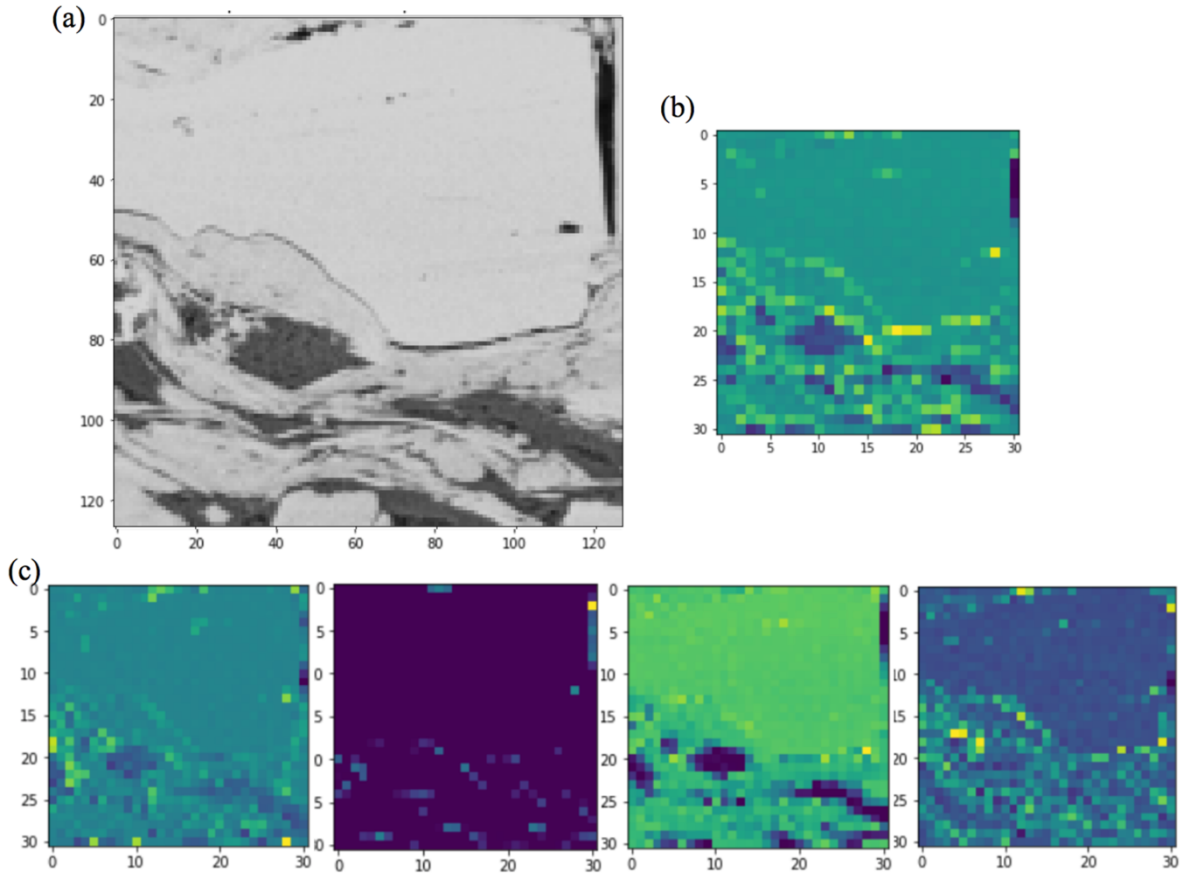


Fig. 3.9 - Feature maps output from the two shallow networks. (a) An SEM image fed into the two shallow networks, (b) a feature map output from the 1-layer 1-filter network, (c) four feature maps output from the 1-layer 4-filters network.

The 1-layer 1-filter and 1-layer 4-filters CNN heatmaps corresponding to the image in **Fig 3.9(a)** are shown in **Fig. 3.10**, showing important features that led to the CNN classification decisions. The heatmap, as mentioned earlier, is a composite of the feature maps. The heatmap from the 1-layer 1-filter network (**Fig. 3.10(b)**) shows the network focusing on the edges of the organic material as well as the interface between the clay and the carbonate. There appear to be a few similarities in the heatmap when increasing the breadth to 4 filters (**Fig. 3.10(c)**).

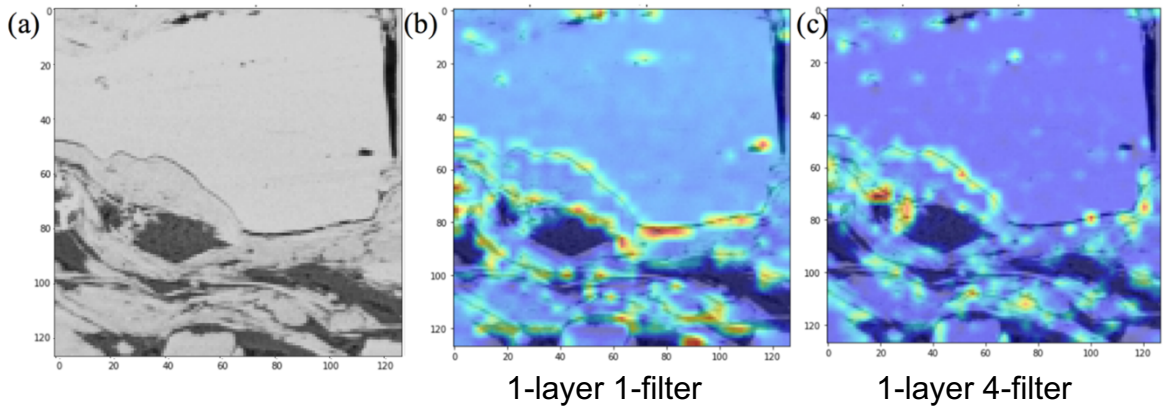


Fig. 3.10 - (a) A SEM image fed into the two shallow networks, (b) heatmap output from the 1-layer 1-filter network, (c) heatmap output from the 1-layer 4-filters network.

There are several cases where the 1-layer 1-filter network makes an incorrect prediction while the 1-layer 4-filters network is accurate. **Fig. 3.11** is such an example where the 1-layer 1-filter network misclassifies the Woodford images as Niobrara. On the other hand, the 1-layer 4-filters network picks a few additional features to aid classification. In both cases, the significant features are non-intuitive picks. **Appendix Fig. A4** to **Fig. A9** shows more examples of heatmaps from different plays.

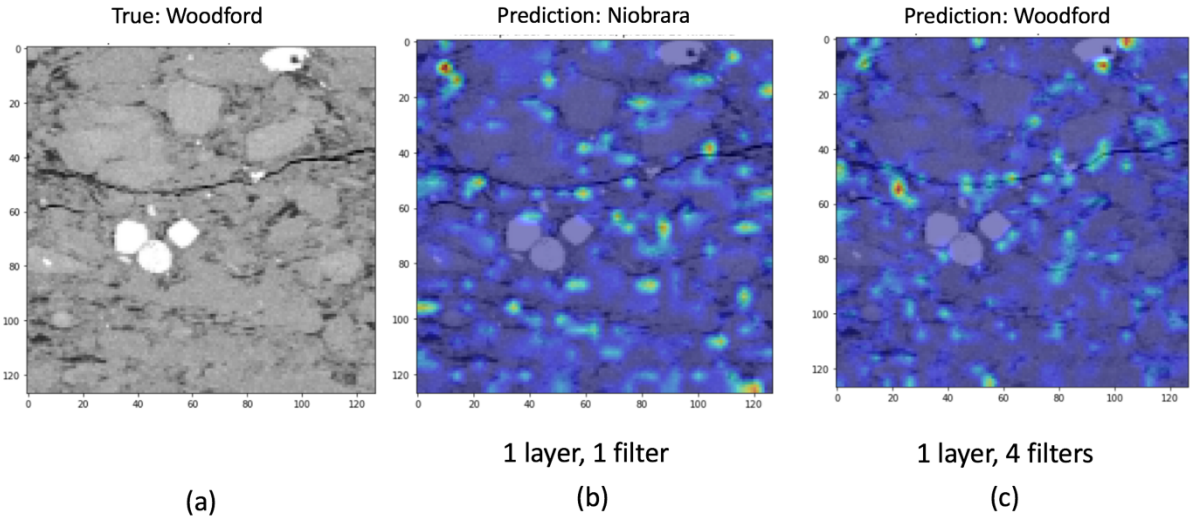


Fig. 3.11 - (a) An SEM image from Woodford fed into the two shallow networks, (b) heatmap output from the 1-layer 1-filter network which misclassified the Woodford image as a Niobrara sample, (c) heatmap output from the 1-layer 4-filters network which correctly classified the image as a Woodford sample.

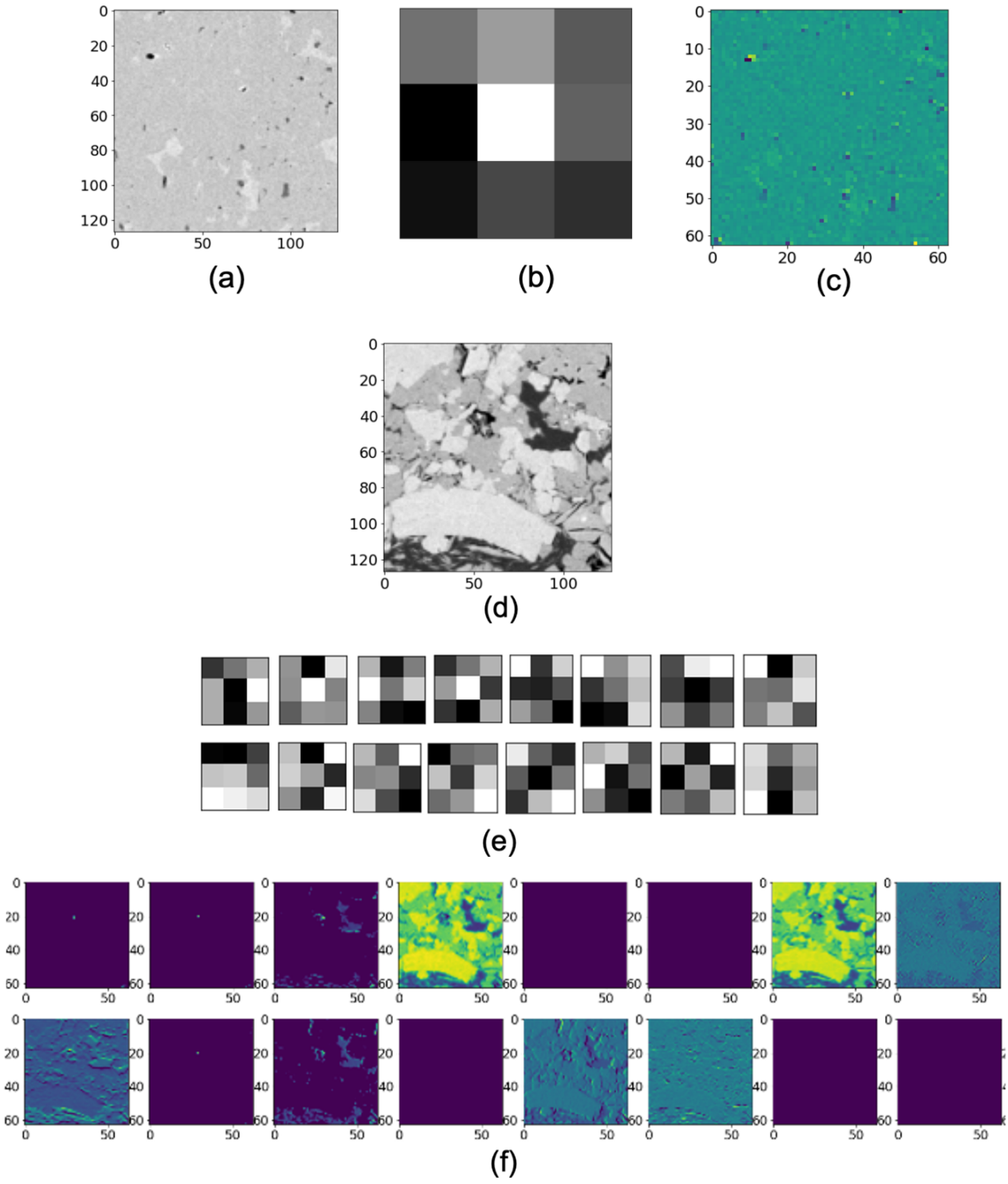


Fig. 3.12 - (a) Horn River Evie sample input to the 1-layer 1-filter network (b) filter in the 1-layer 1-filter network (c) feature map output from the 1-layer 1-filter network (d) Vaca Muerta oil window sample input to the 1-layer 16-filters network (e) filters in the 1-layer 16-filters network (f) feature maps output from the 1-layer 16-filters network

I show the pixel value of each filter and filter output (feature maps) of a specific image in the 1-layer 1-filter network and 1-layer 16-filters network. A Horn River Evie sample is input to the network, as shown in **Fig. 3.12(a)**. The pixel value of the only filter in the 1-layer 1-filter network is shown in **Fig. 3.12(b)**. This filter is shown to captures the grayscale of the

input image as shown in **Fig. 3.12(c)**. The filters in the 1-layer 16-filters network can potentially extract several more features with the filters shown in **Fig. 3.12(e)**. A Vaca Muerta oil window sample as shown in **Fig. 3.12(d)**, is input to the network. The feature maps processed by each filter are shown in **Fig. 3.12(f)**. The 3rd and 11th filters capture the darker feature, including pores and organic materials of the input sample. The 4th and 7th filters capture grayscale. The 8th filter captures the edges of the lighter features. The 9th and 14th filters capture the horizontal features, and the 13th filter captures the diagonal features.

As mentioned earlier, the 1-layer models achieve ~80% accuracy for the 50nm/px dataset with 22 plays, indicating the need for an increase in depth of the network, which I will discuss in the next section.

3.5 50nm/px Resolution: Modest and Deep Network Results

In this section, I investigate the effect of increasing network depth (number of layers) by changing the number of layers to 2, 3, and 5 layers with various filter configurations. The model architecture and the corresponding results are shown in **Table. 3.5,** and more detailed results are shown in **Appendix Fig. A10.**

# of Layers	1	2	3	5
# of Filters	2	2, 2	2, 4, 4	2, 4, 4, 8, 8
Accuracy %	69	70	82	85

(a)

# of Layers	1	2	3	5
# of Filters	4	4, 4	4, 8, 8	4, 8, 8, 16, 16
Accuracy %	77	81	92	92

(b)

# of Layers	1	2	3	5
# of Filters	8	8, 8	8, 16, 16	8, 16, 16, 32, 32
Accuracy %	73	88	93	95

(c)

# of Layers	1	2	3	5
# of Filters	16	16, 16	16, 32, 32	16, 32, 32, 48, 48
Accuracy %	79	91	94	95

(d)

# of Layers	1	2	3	5
# of Filters	32	32, 32	32, 64, 64	32, 64, 64, 96, 96
Accuracy %	80	93	96	96

(e)

Table. 3.5 - Shallow vs. deep network performance on 50nm/px resolution 22 plays dataset.

As shown in **Table. 3.5(a)**, the extremely shallow 1-layer 2-filters model has a comparable accuracy of ~70% with the 2-layer 2, 2-filters model. In contrast, the 3-layer and 5-layer networks outperform the shallow networks with over 80% accuracy.

When the number of filters (breadth) I doubled in each layer for all the models as shown in **Table. 3.5(b),** the 1-layer 4-filters network and the 2-layer 4, 4 filters networks continue to have 80% accuracy**.** However, the 3-layer and the 5-layer networks achieve a comparable

accuracy of over 90%. A comparison with **Table 3.5(a)** shows a 10% increase in accuracy when doubling the number of filters in each layer.

If I further increase the number of filters (breadth) to 8 and 16 filters in the first layer, as shown in **Table. 3.5(c)** and **Table. 3.5(d)**, the accuracy of the shallow 1-layer network remains around 80%, while the 2-, 3-, and 5- layers networks achieve comparable accuracies of greater than 90%. A further increase to 32 filters in Layer 1 does not lead to any appreciable increase in accuracy, as shown in **Table. 3.5(e)**.

The results in **Table. 3.5** show that beyond a certain depth or beyond a certain layer width, there are no appreciable improvements in performance. However, below these thresholds, filter performance is compromised as with the 2-layer 2, 2-filters model.

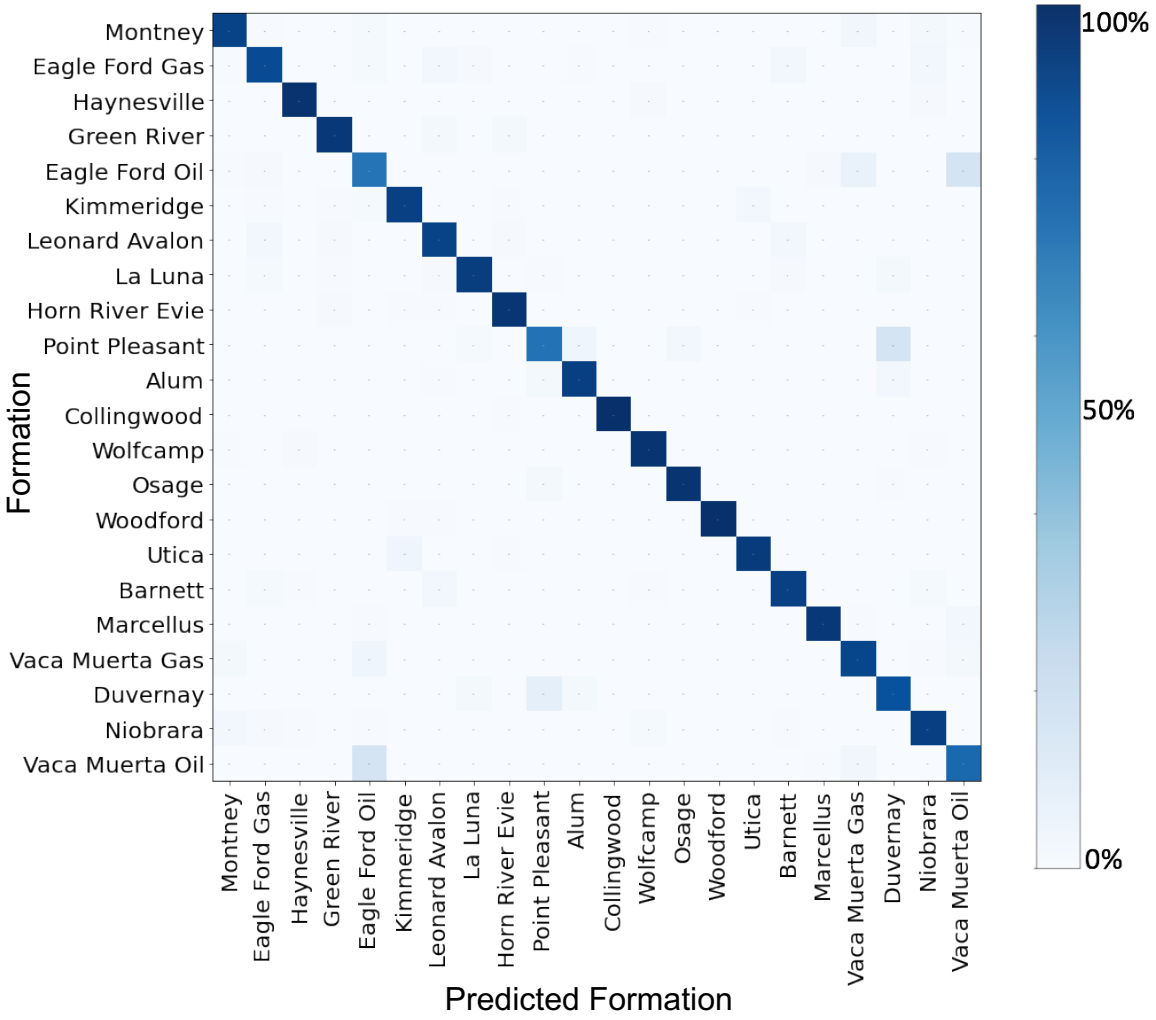


Fig. 3.13 - Confusion matrix of the 2-layer 16, 16 filters CNN trained on 50nm/px resolution dataset achieves a total accuracy of 91%.

The confusion matrix corresponding to the modest 2-layer 16, 16-filters model is shown in **Fig. 3.13.** It achieves a total accuracy of 91%, with higher prediction recall for the Haynesville, Green River, Alum, Collingwood, Wolfcamp, Osage, and Woodford formations. The network however incorrectly classifies a great percentage of the Eagle Ford oil window sample as a Vaca Muerta oil window sample, the Vaca Muerta oil window sample as an Eagle Ford oil window sample, the Point Pleasant sample as a Duvernay sample, and the Duvernay sample as a Point Pleasant sample, indicating that the trained 2-layer 16, 16-filters network detects some similarities between these formation pairs.

I show a misclassified Vaca Muerta oil window sample and its heatmaps obtained from the 2-layer 16, 16-filters trained CNN shown in **Fig. 3.14**. As shown in **Fig. 3.14(b)**, the network predicts this Vaca Muerta oil window image with over 50% probability of being an Eagle Ford oil window sample, and a slightly lower probability of it being a Vaca Muerta oil window sample. I also show the heatmaps obtained using the same Vaca Muerta oil window image, as shown in **Fig. 3.14(c).** Within the 1st convolutional layer of the 2-layer network, the trained CNN picks feature along the boundaries of the organic and inorganic material. The heatmap from the 2nd convolutional layer shows the organic matter to be a significant feature for the network as shown in **Fig. 3.14(d)**.

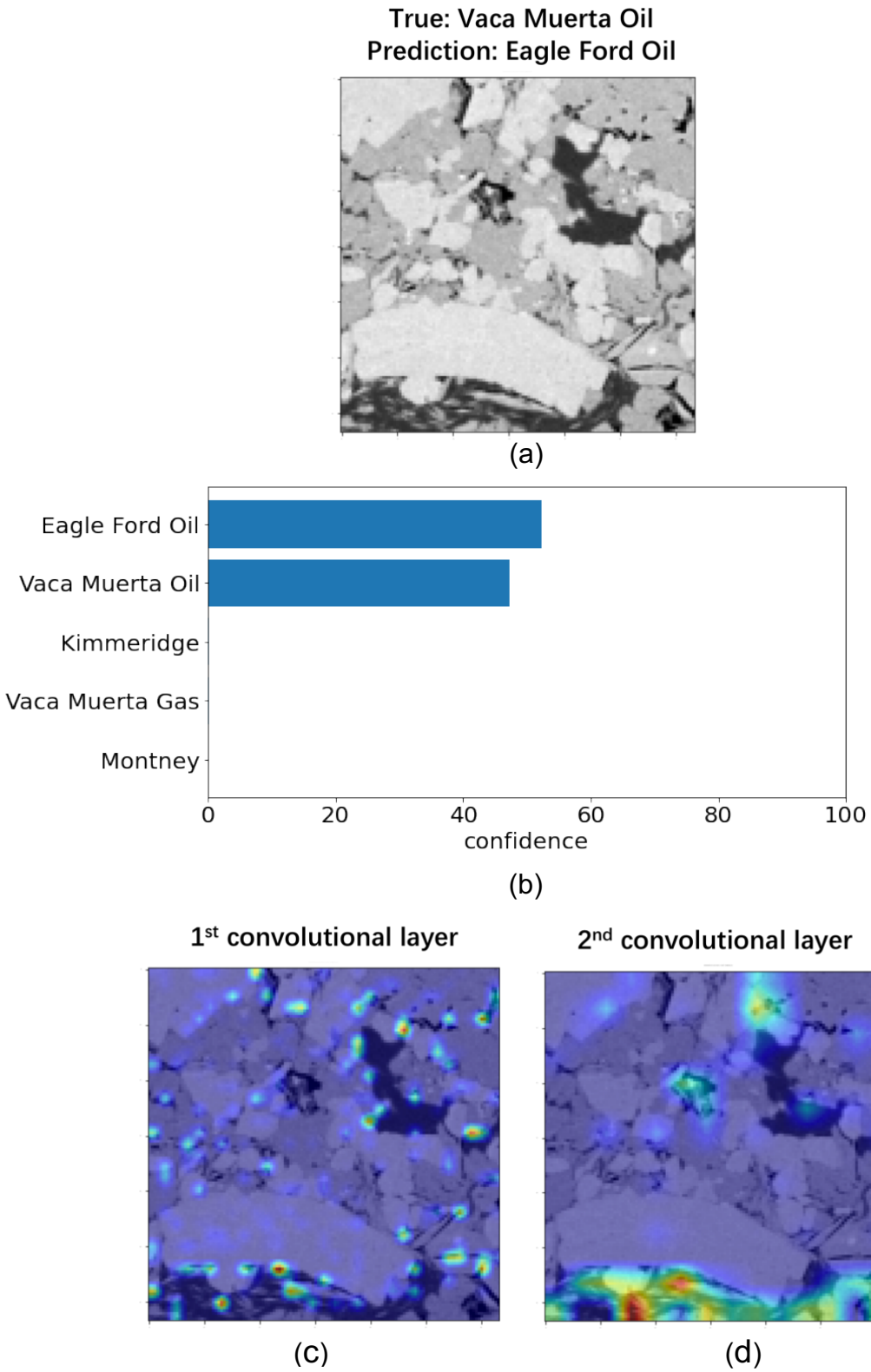


Fig. 3.14 - (a) The Vaca Muerta oil window image input to the 2-layer 16, 16-filters network is misclassified as an Eagle Ford oil window sample, (b) the model predicts the image with over a 50% probability being an Eagle Ford oil window sample, (c) heatmap output from the 1st convolutional layer, (d) heatmap output from the 2nd convolutional layer.

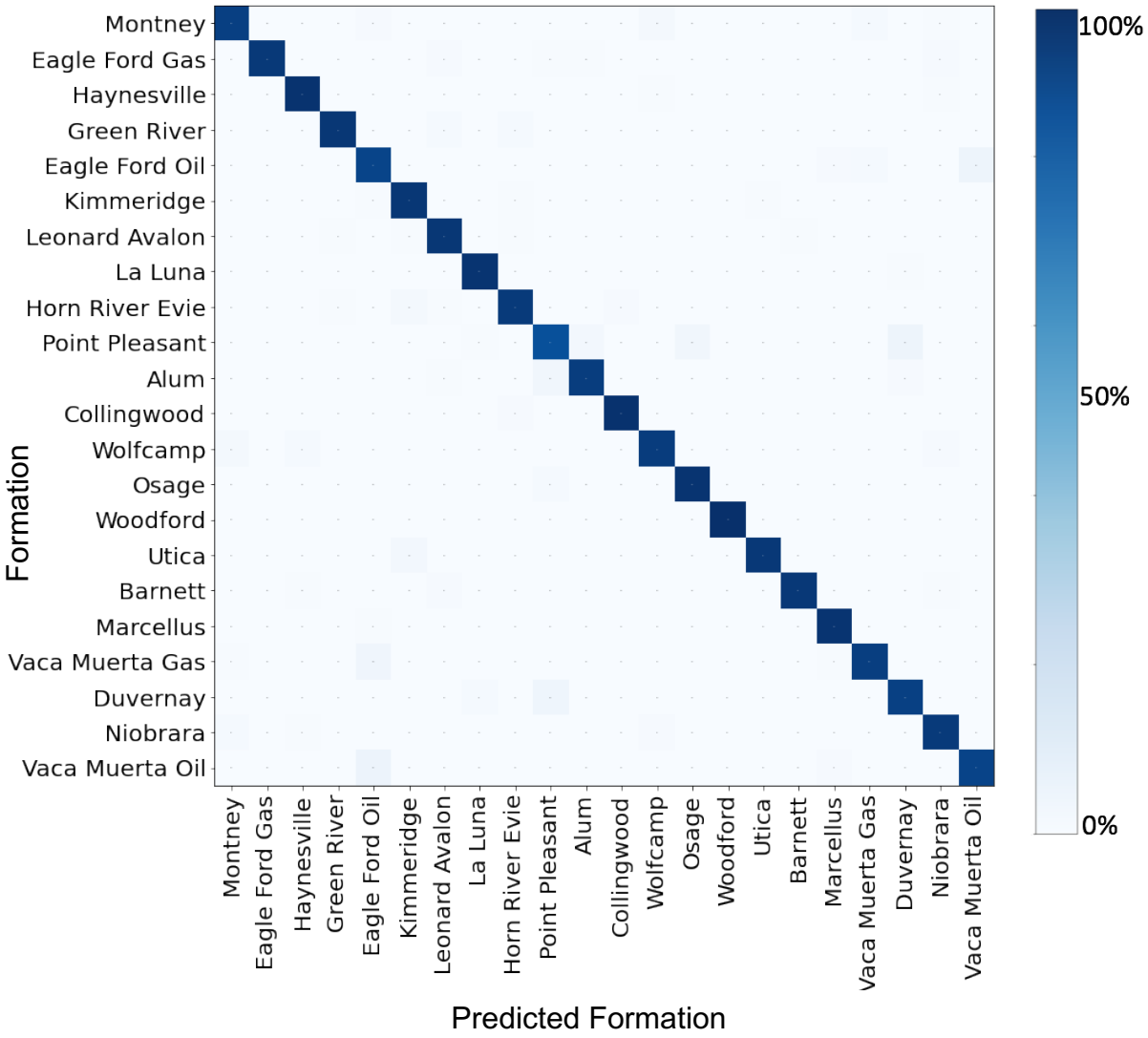


Fig. 3.15 - Confusion matrix of the 5-layer 32, 64, 64, 96, 96-filters CNN trained on 50nm/px resolution dataset achieves a total accuracy of 96%.

In contrast, the deepest and broadest 5-layer 32, 64, 64, 96, 96-filters model achieves 96% total accuracy, and fewer off-diagonal elements can be observed as shown in **Fig. 3.15** compared to a shallower 2-layer 16, 16 filters network. It achieves in excess of 90% recall for each formation except a few Eagle Ford oil window and Point Pleasant samples which are misclassified as the Vaca Muerta oil window and Duvernay samples, respectively. The misclassified pairs are common to all the shallow, modest, and deep networks, indicating similar microstructures. As Knaup (2019) points out, this is important information because

microstructural similarities can potentially be exploited to apply learnings from one formation/play to the other.

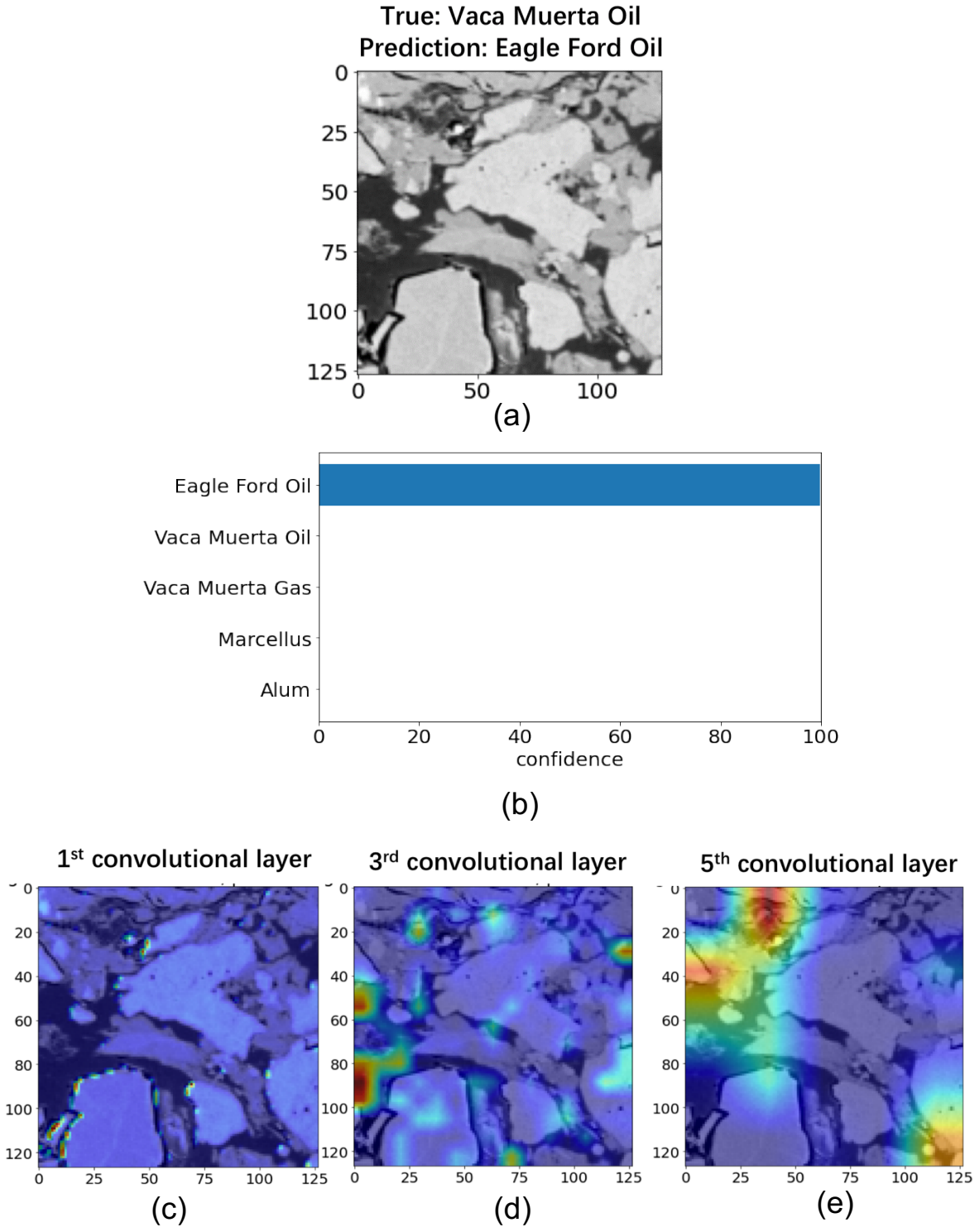


Fig. 3.16 - (a) The Vaca Muerta oil window image input to the 5-layer 32, 64, 64, 96, 96-filters network is misclassified as an Eagle Ford oil window sample, (b) the model predicts the image with a 100% probability being an Eagle Ford oil window sample, (c) heatmap output from the 1st convolutional layer, (d) heatmap output from the 3rd convolutional layer, (e) heatmap output from the 5th convolutional layer.

I show a misclassified image sample from the formation (Vaca Muerta oil window) with the highest misclassification rate when tested on the 5-layer 32, 64, 64, 96, 96-filters in **Fig. 3.16(a)**. The trained network is 100% confident this is an Eagle Ford oil window sample as shown in **Fig. 3.16(b).** I show the corresponding heatmaps obtained with the same Vaca Muerta oil window image fed into the 5-layer 32, 64, 64, 96, 96-filters trained CNN. As shown in **Fig. 3.16(c)**, in the shallow 1st convolutional layer, the trained CNN picks a few of the boundaries of the inorganic matrix. The organic matter is then seen to become important in the 3rd layer as shown in **Fig. 3.16(d)**. All the features then combine into the last convolutional layer (the 5th layer) as shown in **Fig. 3.16(e)**, where non-intuitive large-scale features are considered significant. Again, this misclassification underscores the petrophysical similarities between the Vaca Muerta oil window and the Eagle Ford oil window.

3.6 50nm/px Resolution: Shallow vs. Deep Network Performance

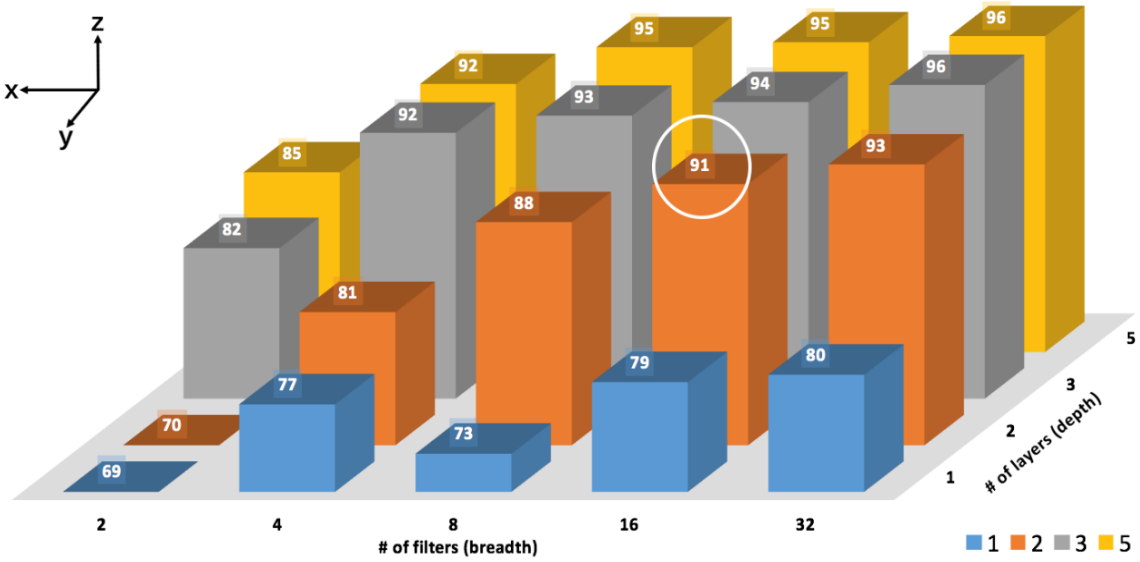


Fig. 3.17 - 3D bar chart showing the accuracy of the CNNs in varying depth and breadth. The 2-layer 16, 16-filters CNN in the white circle is sufficient for the dataset at 50nm/px resolution, which provides over 90% accuracy.

The composite test results are shown in a 3D bar chart in **Fig. 3.17**. The x-axis refers to the number of filters (breadth), the y-axis refers to the number of layers (depth), and the z-axis

refers to the accuracy of each network. We can observe that a 2-layer 16, 16-filters CNN at 50nm/px provides a very satisfactory accuracy of over 90%.

Looking at the 1-layer network models, it is clear that simply increasing the model breadth does not lead to high prediction accuracies, although they are acceptable. On the other hand, simply increasing the depth also limits filter performance.

A judicious choice of filter width and depth is necessary, and **Fig. 3.17** shows that a moderate depth of 2 layers with a wide filter diversity provides over 90% accuracy.

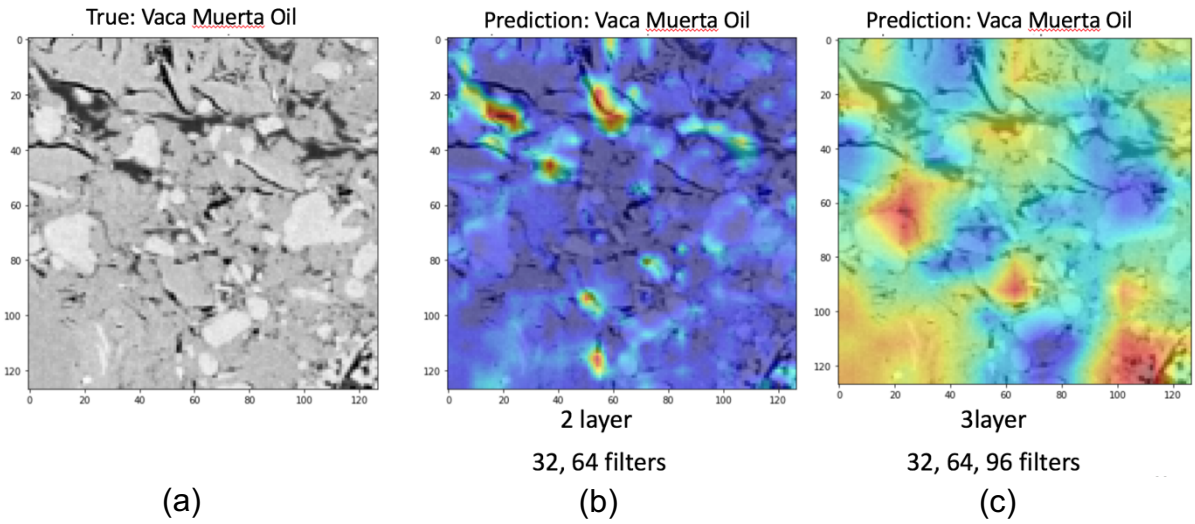


Fig. 3.18 - (a) An SEM image from Vaca Muerta oil window, (b) heatmap output from the 2-layer 32, 64-filters network, (c) heatmap output from the 3-layer 32, 64, 96-filters network.

There are a few cases where the shallow and modest networks both correctly identify the source of the image. **Fig. 3.18** is such an example, but the significant features selected by the network are quite different. The 2-layer 32, 64-filters network picks a few small-scale features including micro-fractures and organic matters, meanwhile, the 3-layer 32, 64, 96 filters network non-intuitively selects a large areal proportion of the image.

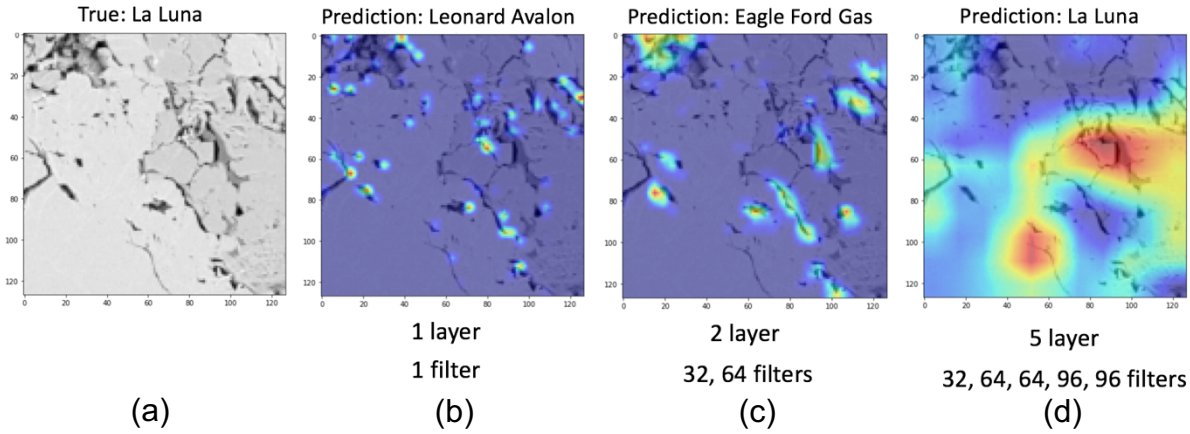


Fig. 3.19 - (a) An SEM image from La Luna, (b) heatmap output from the 1-layer 1-filter network, (c) heatmap output from the 2-layer 32, 64-filters network, (d) heatmap output from the 5-layer 32, 64, 64, 96, 96-filters network.

There are a few cases where the shallow and modest networks misclassify an image while a deeper network makes the right decision. **Fig. 3.19** is such an example where the 1-layer 1-filter network misclassifies a La Luna image as a Leonard Avalon sample by picking small-scale features including micro-fractures as shown in **Fig. 3.19(b)**. In addition, the 2-layer 32, 64-filters network misclassify the same La Luna image as an Eagle Ford gas window sample as shown in **Fig. 3.19(c)**. The deeper 5-layer network, on the other hand, makes the right decision as shown in **Fig. 3.19(c)**.

3.7 Comparison between the 50nm/px and the 25nm/px with 22 Formations

In the second part of this chapter, I test the same model architectures on a 25nm/px resolution (3x3µm field-of-view) dataset sourced from the same twenty-two formations mentioned earlier. In the previous chapter, although my hypothesis was that a smaller field-of-view would confound SEM image classification, I report that the models trained on the 10nm/px images marginally outperformed those trained on the 25nm/px images. The study was restricted to 8 plays and needs more exploratory analyses; however, these preliminary results appear to indicate that the key features for identification are more commonly in the range of 10nm/px resolution compared to 25 nm/px resolution.

Because the dataset is restricted at 10nm/px resolution, the highest resolution I can adopt is 25 nm/px. The raw grayscale SEM images from 22 plays in various pixel sizes shown in **Table 3.1** are rescaled to 25nm/px resolution and then sliced to 127x127 pixels without overlap, equivalent to a 3x3 μm field-of-view as shown in **Fig. 3.20**.

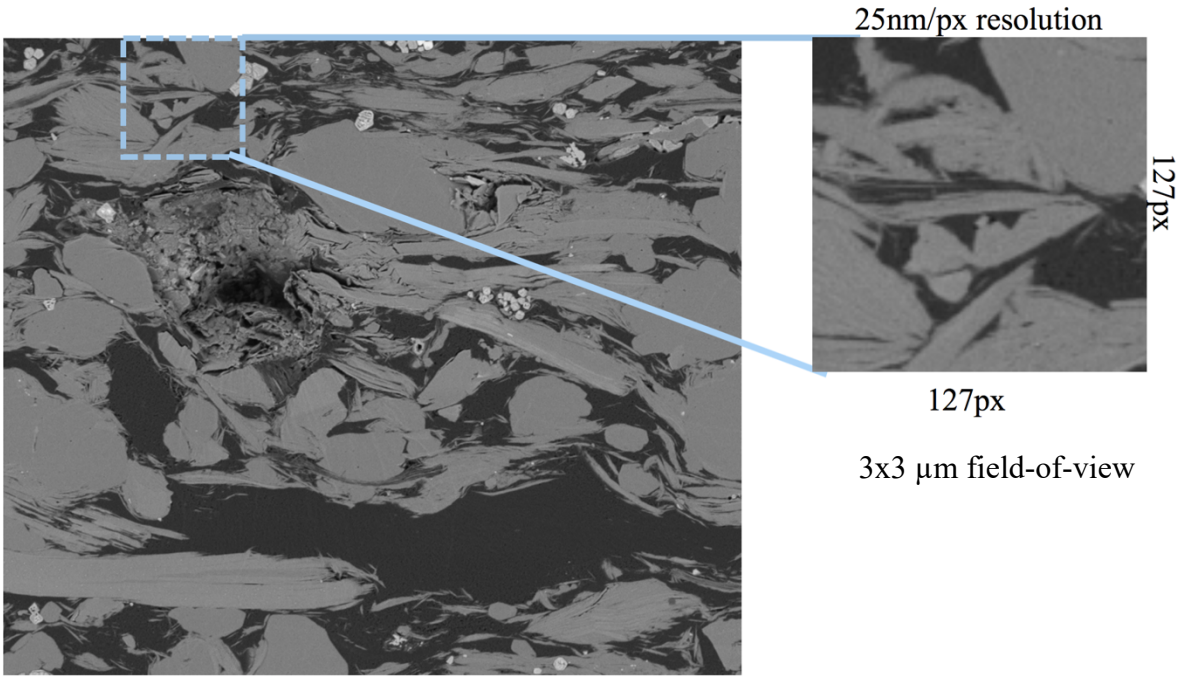


Fig. 3.20 - Example of a raw SEM image from Alum with 10nm/px resolution. It is rescaled to 25nm/px resolution and sliced to 127x127 pixels size to fit into the model. The left figure is the raw image; the right figure is one of the rescaled and sliced images for CNN model training.

Table 3.6 shows the detailed information of input images from each play. The input images from 22 plays are split into a training set, a validation set, and a testing set. As mentioned earlier, there are 840 images per formation and a total of 18480 images to create a balanced training set for training; 180 images per formation and a total of 3960 images for validation; and a total of over 400000 images for testing to assess model performance. Twenty example images in the 25nm/px resolution dataset are shown in **Fig. 3.21**.

Play	Resolution (nm)	Bit Depth	# of images for training	# of images for validation	# of images for testing
Haynesville	25	8	840	180	24732
Wolfcamp	25	8	840	180	14279
Alum	25	8	840	180	21780
Montney	25	8	840	180	24732
Eagle Ford Oil	25	8	840	180	24732
Eagle Ford Gas	25	8	840	180	24732
Avalon/Leonard	25	8	840	180	24732
Vaca Muerta Oil	25	8	840	180	24732
Vaca Muerta Gas	25	8	840	180	24732
Duvernay	25	8	840	180	21780
Osage	25	8	840	180	21780
La Luna	25	8	840	180	3030
Kimmeridge	25	8	840	180	24732
Point Pleasant	25	8	840	180	21780
Green River	25	8	840	180	16380
Horn River Evie	25	8	840	180	16380
Collingwood	25	8	840	180	6636
Marcellus	25	8	840	180	74700
Woodford	25	8	840	180	180
Utica	25	8	840	180	24732
Niobrara	25	8	840	180	24732
Barnett	25	8	840	180	24732

Table. 3.6 - 25nm/px resolution (3x3 μm field-of-view) SEM image information of 22 plays.

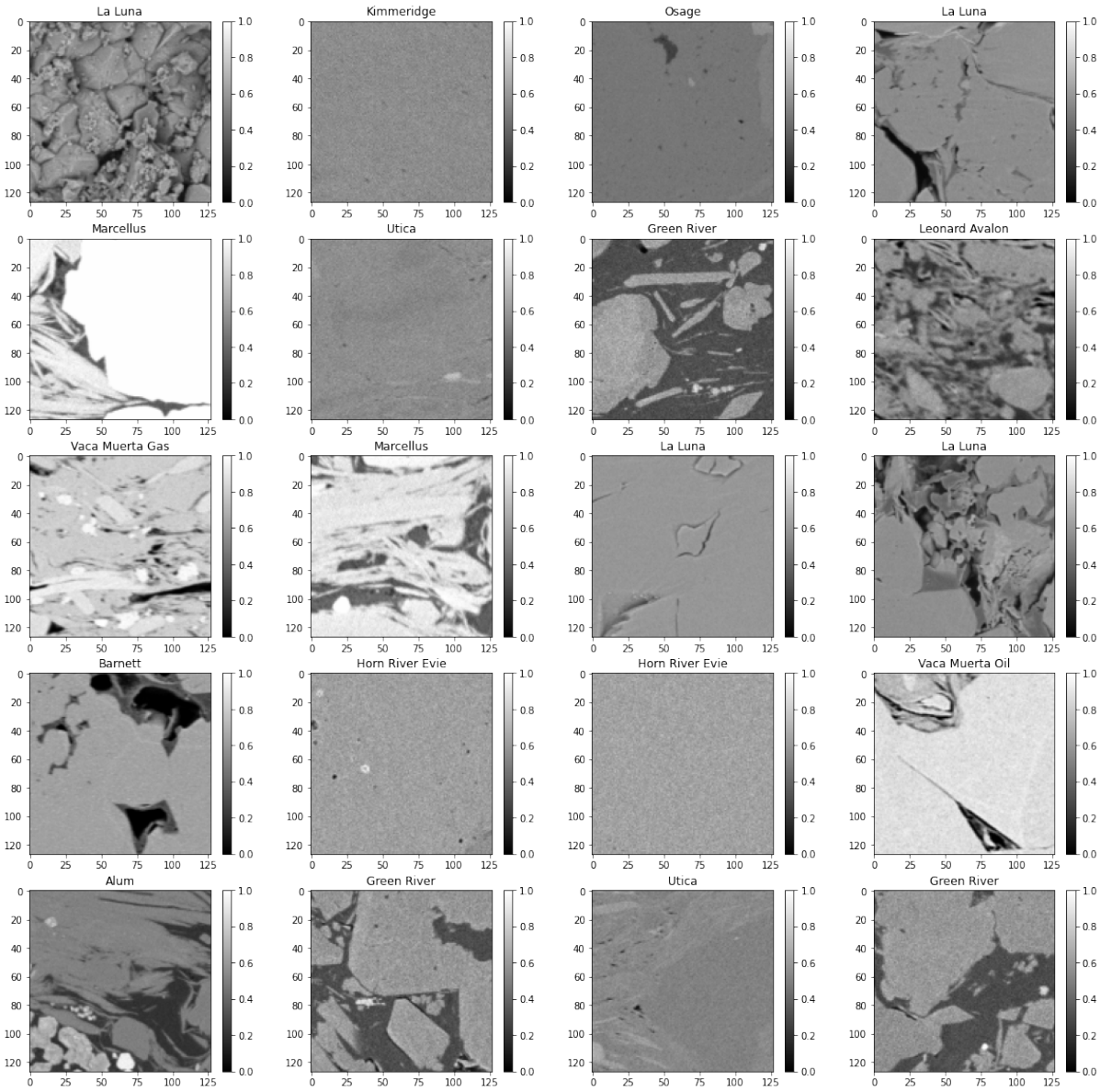
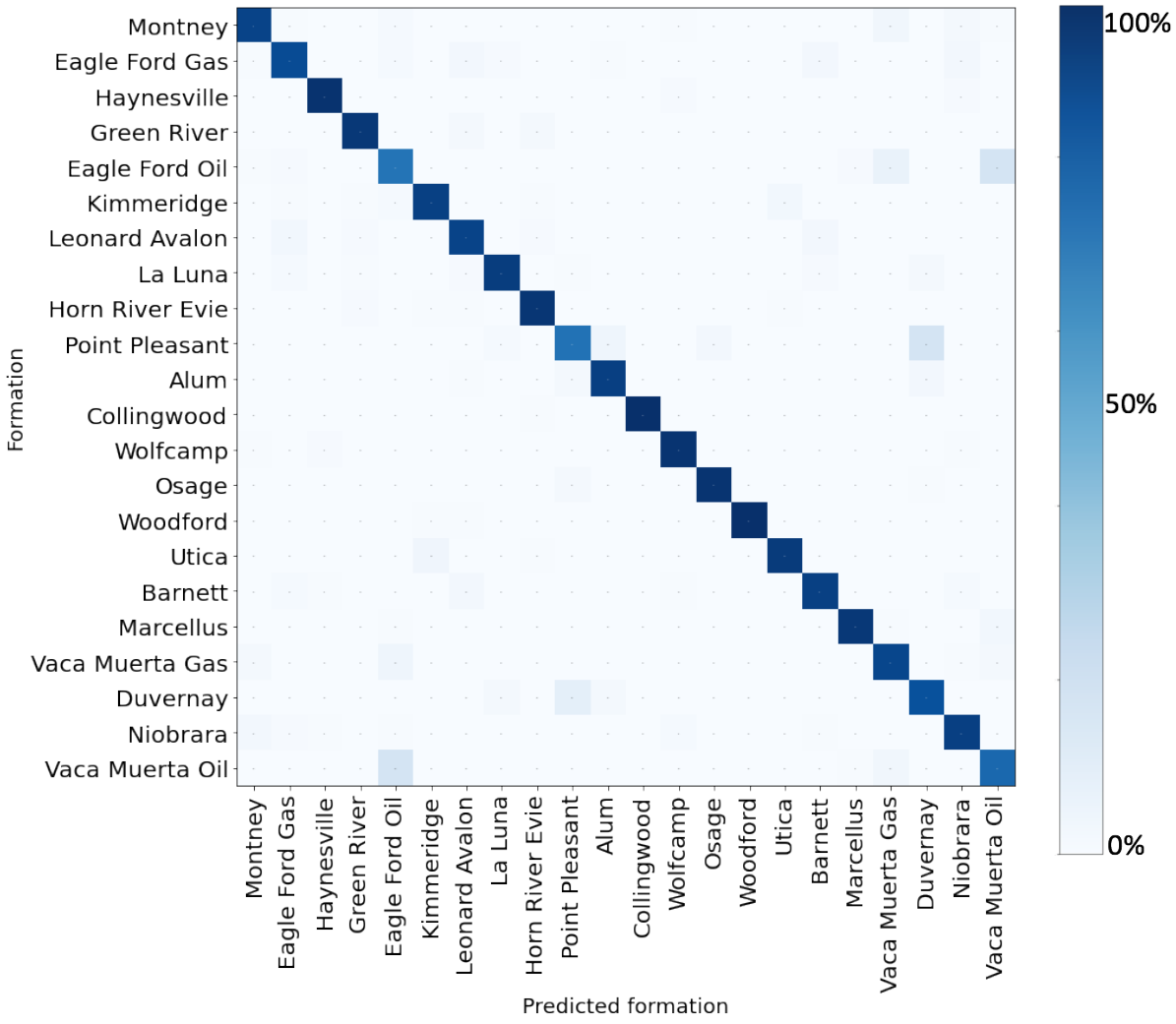


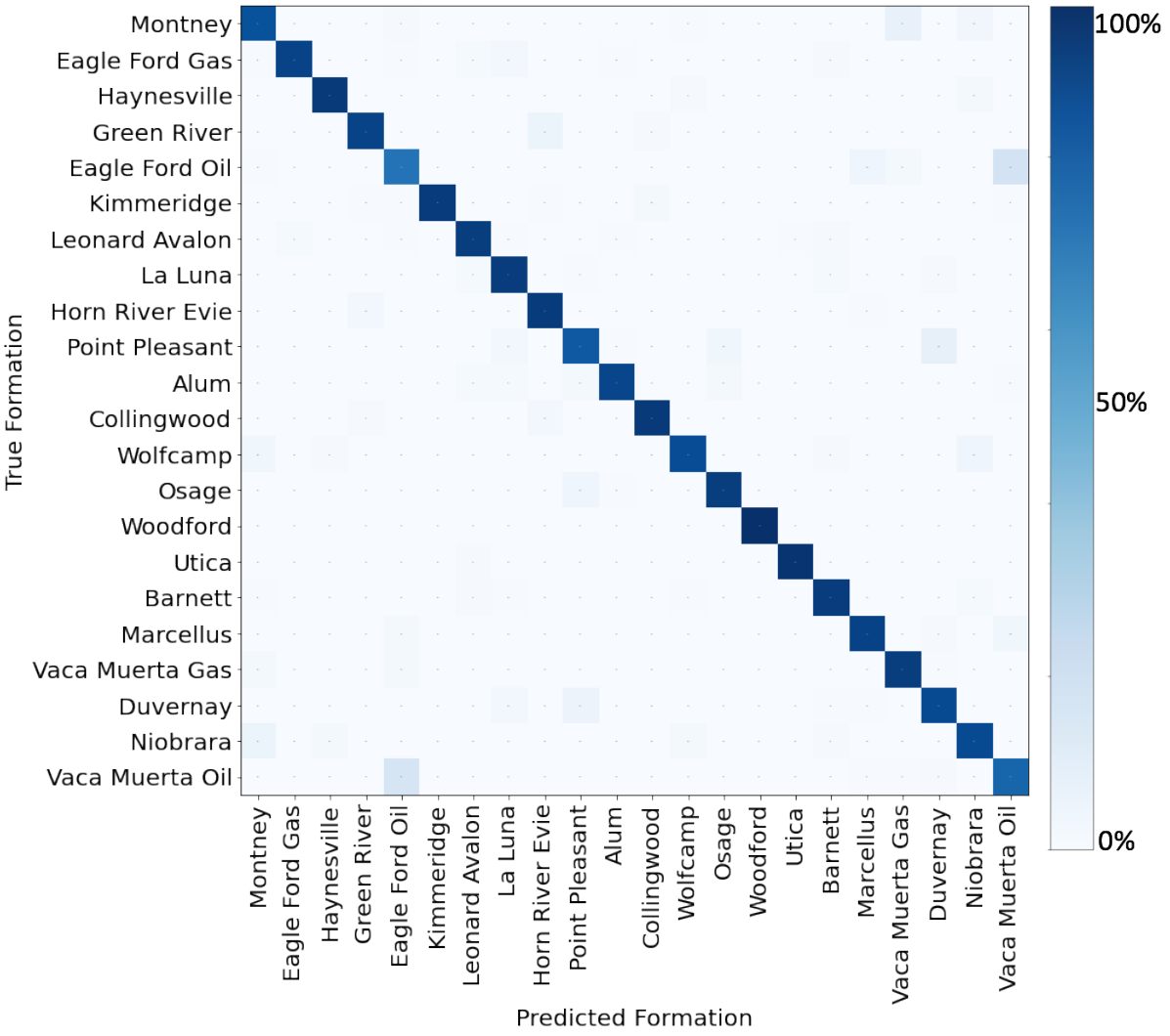
Fig. 3.21 - Twenty grayscale SEM images from 22 plays for play identification with 25nm/px resolution (3x3 μm field-of-view).

I present the confusion matrices corresponding to two high-accuracy models (2-layer 16, 16-filters) trained separately on 50nm/px and 25nm/px resolution datasets in **Fig. 3.22.** The models achieve ~ 91% accuracy when tested with comparable recall values achieved for each formation. As expected, both networks incorrectly classify a few of the Eagle Ford oil window samples as the Vaca Muerta oil window samples, the Vaca Muerta gas window samples as the Eagle Ford oil window samples, the Duvernay samples as the Point Pleasant samples, and the Point Pleasant samples as the Duvernay samples. Interestingly, the network trained on the

25nm/px resolution dataset incorrectly classified few Montney samples as Vaca Muerta gas window samples and a few of the Green River samples as Horn River Evie samples. The 50nm/px made the right decision with these plays indicating that a smaller field-of-view (increasing resolution) may limits the availability of larger-scale features for identification of the play.



(a) 50nm/px resolution

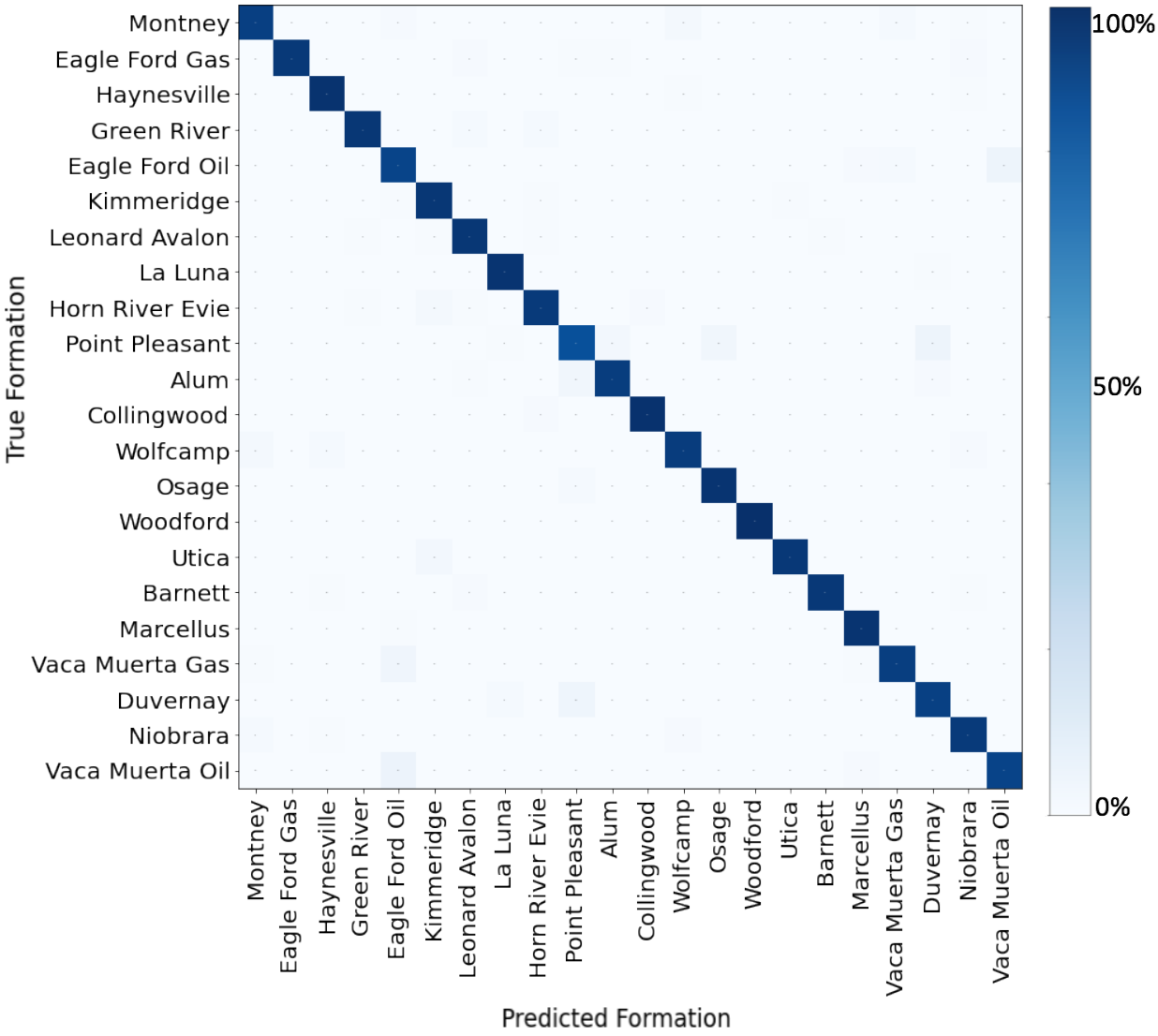


(a) 25nm/px resolution

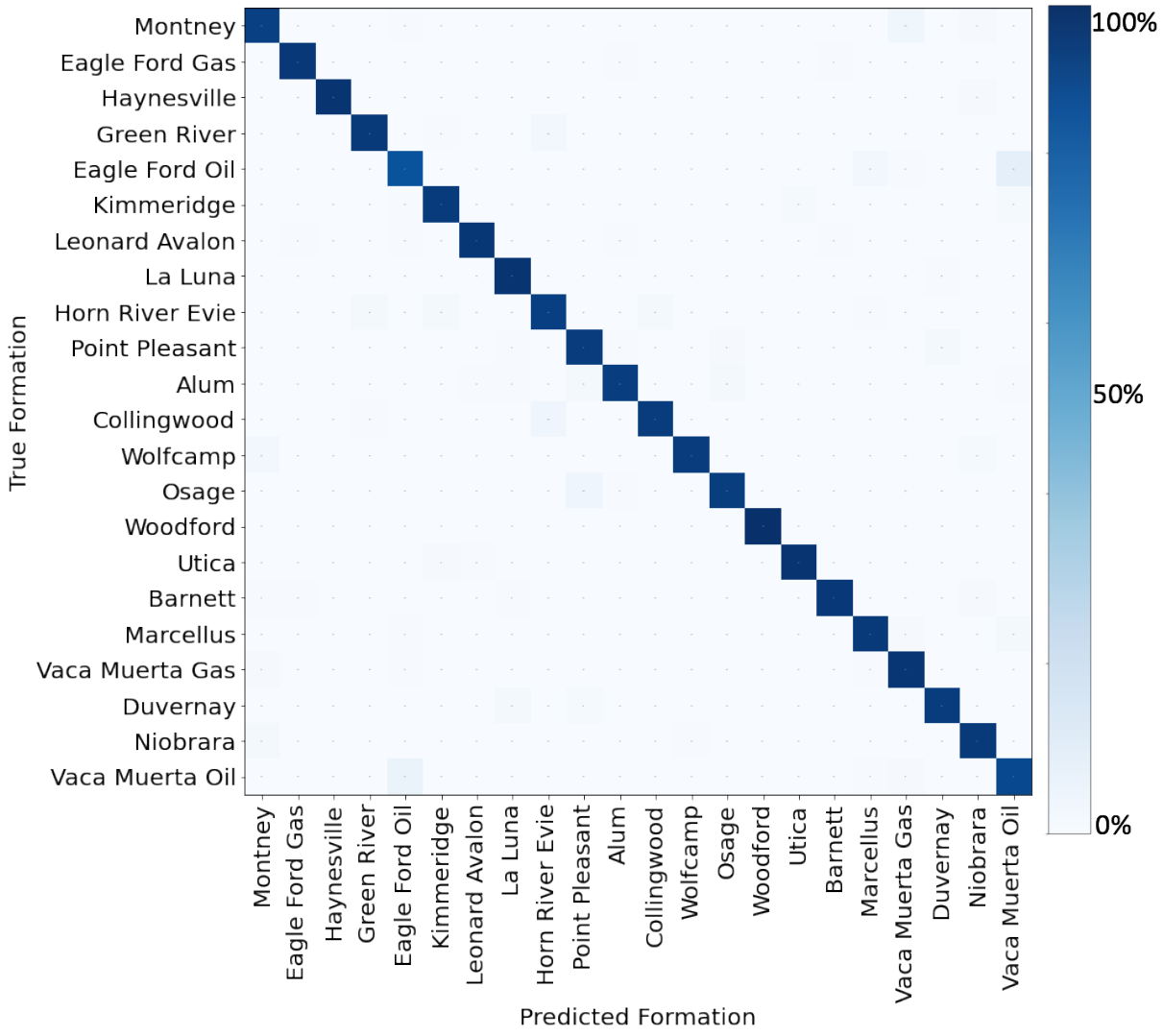
Fig. 3.22 - Comparison of confusion matrix corresponding to the 2-layer 16,16 filters CNN trained on (a) 50nm/px resolution dataset with a total accuracy of 91%, (b) 25nm/px resolution dataset with a total accuracy of 91%.

The deepest and broadest 5-layer 32, 64, 64, 96, 96-filters model achieves a comparable 96% and 95% total accuracy on 50nm/px resolution and 25nm/px resolution dataset respectively. Both networks still misclassify a few of the Eagle Ford oil window samples as the Vaca Muerta oil window samples, a few of the Vaca Muerta gas window samples as the Eagle Ford oil window samples, a few of the Duvernay samples as the Point Pleasant samples, and a few of the Point Pleasant samples as the Duvernay samples as shown in **Fig. 3.23**. It indicates these formation pairs may have similar microstructures that cannot be resolved by

both modest and deep CNNs. Moreover, even with 5-layers, the network trained on 25nm/px dataset still misclassified few Montney samples as Vaca Muerta gas window samples, and few Green River samples as Horn River Evie samples, indicating a smaller field-of-view (increasing resolution) impairs the classification of these formation pairs.



(b) 50/nm resolution



(b) 25/nm resolution

Fig. 3.23 - Comparison of confusion matrix corresponding to the 5-layer 32, 64, 64, 96, 96-filters CNN trained on (a) 50nm/px resolution dataset with a total accuracy of 96%, (b) 25nm/px resolution dataset with a total accuracy of 95%.

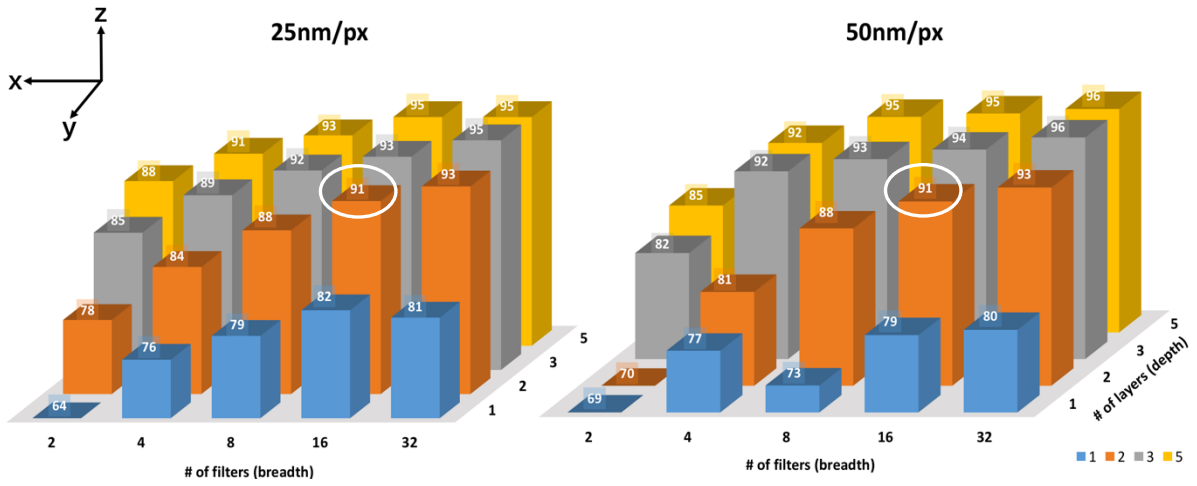


Fig. 3.24 - Comparison of the accuracy for each model trained on 25nm/px resolution dataset and 50nm/px resolution dataset with 22 formations.

I then compare the performance of each model trained on 50nm/px resolution (6x6 µm field-of-view) dataset and 25nm/px resolution (3x3 µm field-of-view) dataset using the 3D bar chart as shown in **Fig. 3.24**. More detailed test results for the 25nm/px resolution dataset with 22 formations are shown in **Appendix Fig. A11** and **Fig. A12**. The bar charts show comparable performances overall.

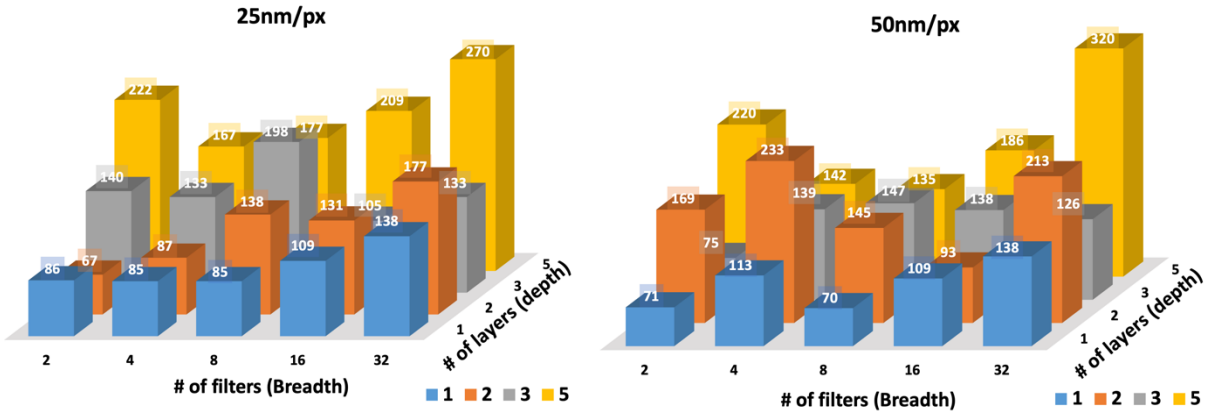


Fig. 3.25 - Comparison of the training time in seconds for each model in trained on 25nm/px resolution dataset and 50nm/px resolution dataset with 22 formations.

I then show the training time of each model trained on 50nm/px resolution (6x6 µm field-of-view) dataset and 25nm/px resolution (3x3 µm field-of-view) dataset using the bar chart as shown in **Fig. 3.25**. As mentioned previously, the x-axis refers to the number of filters

(breadth), the y-axis refers to the number of layers (depth), and the z-axis refers to the training time of each network in seconds. The deepest and broadest network still takes the longest time to train at both resolutions with a similar accuracy as a simpler model, which indicates that deeper networks are likely not necessary for the classification task outlined here.

From the CNN results trained on 25nm/px resolution (3x3μm field-of-view) and 50nm/px resolution (6x6μm field-of-view) dataset, I can conclude that:

- At both 25nm/px and 50nm/px resolution, a diversity of filters is essential because deep networks with limited breadth show poor performance.
- 1-layer networks with wide breadth show poor performance.
- For both datasets, a far simpler 2-layer 16, 16-filters network is sufficient for formation identification. The architecture of the network is shown in **Fig. 3.26**.
- For the images tested, we obtain similar accuracies for 25nm/px and 50nm/px resolution images. Overall, 25nm/px resolution captures several of the important features except for a few formation pairs as mentioned earlier.
- Point Pleasant and Duverney samples, Eagle Ford oil window and Vaca Muerta oil window samples confound all CNNs irrespective of network complexity at both 25nm/px resolution and 50nm/px resolution.
- At 25nm/px resolution, the microstructure of Montney and Vaca Muerta gas window samples, and the Green River and Horn River Evie samples may be similar.

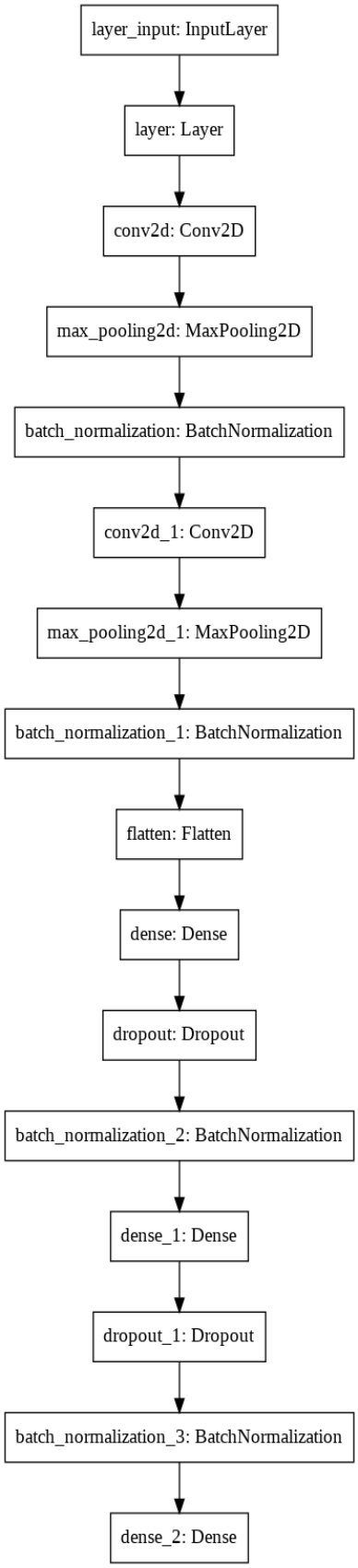


Fig. 3.26 - 2-layer 16, 16 filters network architecture.

3.8 Comparison between the 25nm/px 8 Formations and the 22 Formations

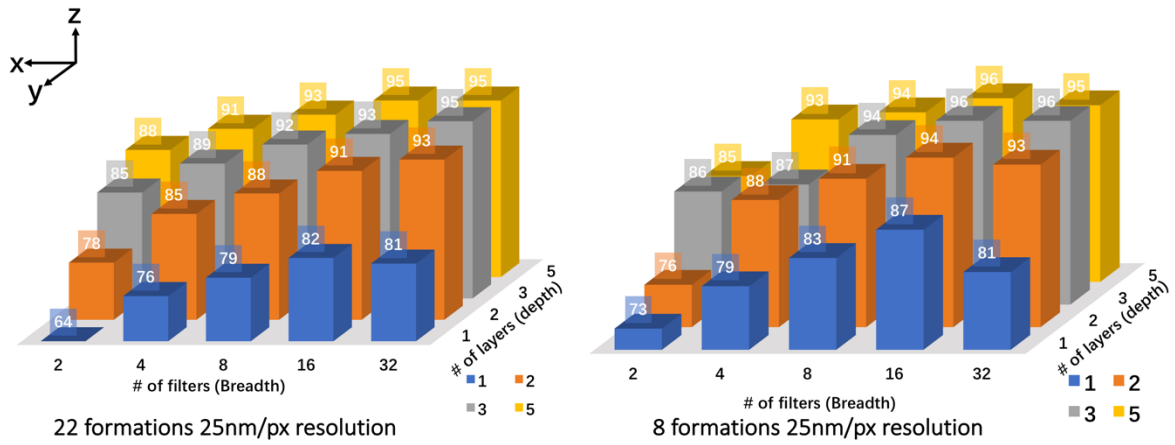


Fig. 3.27 - Comparison of the accuracy for each model trained on 25nm/px resolution dataset with 22 formations, and with 8 formations.

In this section, I investigate the sensitivity of classification accuracy to the number of classes. In the previous chapter, I investigated the performance of various CNNs trained on a 25 nm/px dataset from 8 plays. In this chapter, the dataset is extended to 22 plays. The accuracy of the various models corresponding to both datasets is shown in a 3D bar chart in **Fig. 3.27**.

1-layer models tend to do just marginally better with fewer classes to identify. However, moving to 2 layers and beyond, the accuracies are quite comparable. In other words, the necessary filter complexity is largely independent of the number of classes/labels to be identified.

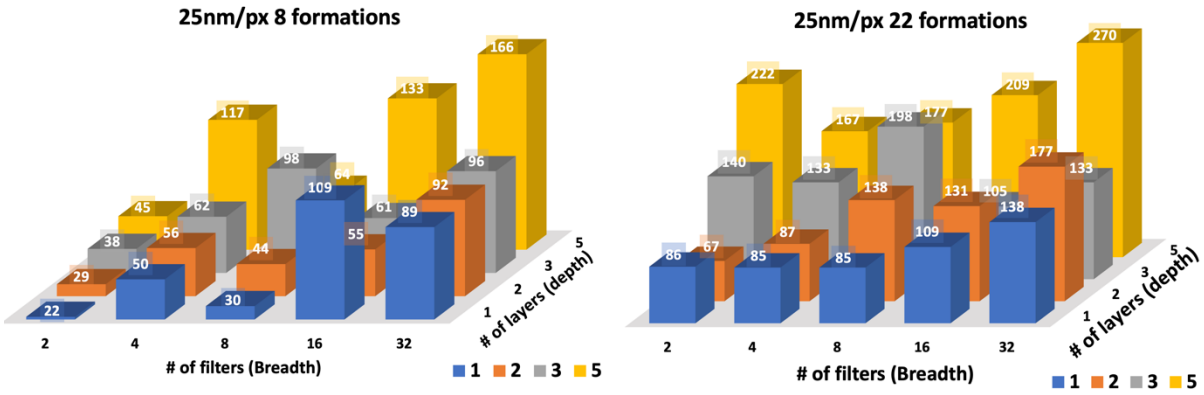
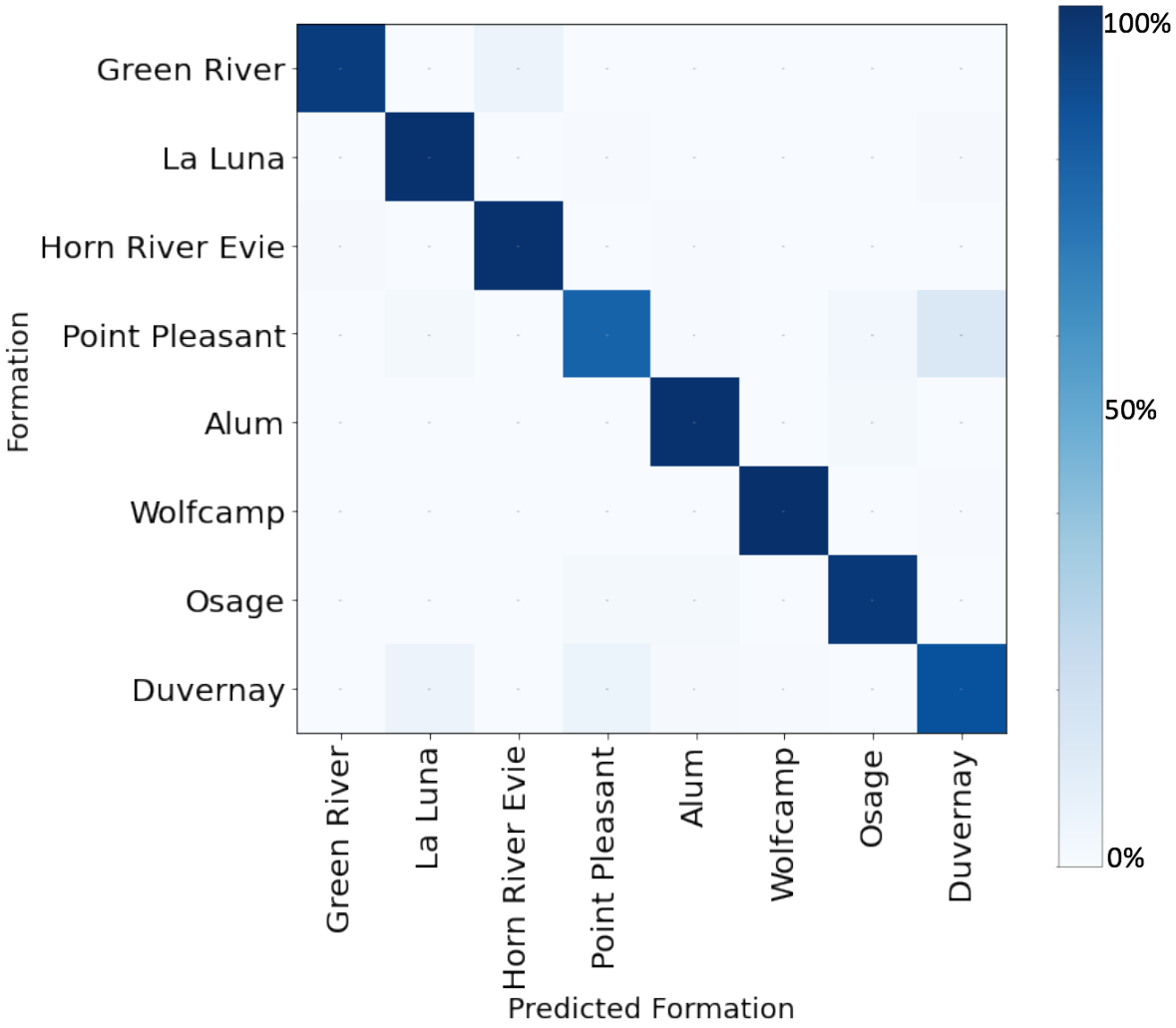


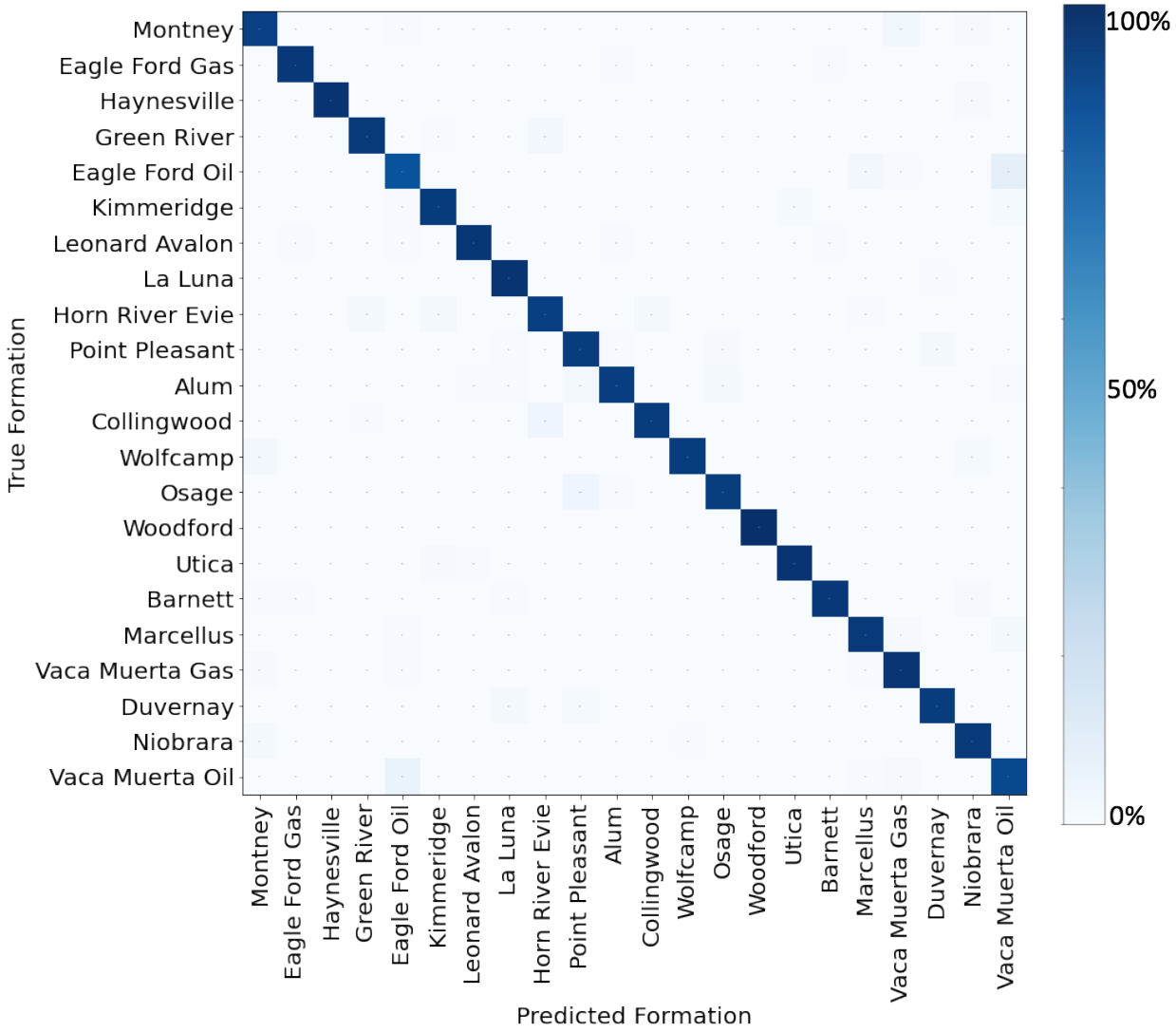
Fig. 3.28 - Comparison of the training time in seconds for each model trained on 8 formations dataset and 22 formations dataset at 25nm/px resolution.

I then compare the training time of each model trained on 8 formations and 22 formations at 25nm/px resolution (3x3 μm field-of-view) using the 3D bar chart as shown in **Fig. 3.28**. The dataset with images from 8 plays has a total number of 6720 training images, and the dataset with images from 22 formations has a total number of 18480 training images. **Fig. 3.28** clearly shows that as the number of images increases, the training time significantly increases. As mentioned previously, I expect a linear increase in training time with increases in training data.

In terms of the confusion matrices, I compare the 5-layer 32, 64, 64, 96, 96-filters networks trained on 25nm/px resolution images from eight formations and from twenty-two formations in **Fig. 3.29**. Both networks misclassify a few of the Point Pleasant samples as Duvernay samples and a few of the Green river samples as Horn River Evie samples, regardless of the number of classes to be identified.



(a) 25nm/px resolution with 8 formations



(b) 25nm/px resolution with 22 formations

Fig. 3.29 - Confusion matrix of the 5-layer 32, 64, 64, 96, 96-filters CNN trained on (a) 25nm/px resolution 8 formations dataset with a total accuracy of 96%, (b) 25nm/px resolution 22 formations dataset with a total accuracy of 95%.

Chapter 4: Conclusions

In this study, I conclude that:

- For the 10nm/px, 25nm/px, and 50nm/px resolution datasets, both the diversity of filters and depth are essential irrespective of the number of categories presented.
- For 10nm/px, 25nm/px, and 50nm/px resolution datasets, either increases in filter width/diversity or increases in depth do not provide measurable benefits beyond a specified limit. For all datasets considered, no more than a 2-layer network is essential.
- The accuracy of classification is largely independent of the number of classes. I obtained comparable accuracies from models trained on the 25nm/px 8 plays dataset and the 25nm/px 22 plays dataset.
- The optimal resolution is 50nm for each play, except Point Pleasant and Duvernay samples with an optimal resolution of 10nm.
- In most models considered, the CNN confounded Point Pleasant and Duverney samples and Eagle Ford oil window and Vaca Muerta oil window samples indicating that at 25nm/px and 50nm/px resolution, there are similarities in these plays. EDX images, which include mineralogy-related information, can potentially aid in reducing the rate of misclassification. Point Pleasant samples are known to have chlorite, while the Duvernay is largely chlorite free.
- Significant features between Green River samples and Horn River Evie samples are resolvable at 50nm/px resolution, however, they are unresolvable at 25nm/px resolution and 10nm/px resolution. Several images from this formation pair at 10nm/px resolution are characterized by white noise and it appears that the CNN is recognizing this white noise as common to both plays, which exacerbates the misclassification problem. There are potentially two methods to address this: Adding

white noise to all images to allow the CNN to focus on unique microstructural features to add classification, or de-noise the images. De-noising can remove some important features if they are at the scale at which the noise is present. I did not test these effects in this work.

- The training time significantly increases as the number of images increases and as the complexity of the CNN increases. I obtain a longer training time for the 25nm/px 22 plays dataset with a total number of 18480 training images, compare with the 25nm/px 8 plays dataset with a total number of 6720 training images.

References

Abadi, M., Agarwal,A., Barham,P., Brevdo,E., Chen, Z., Citro, C., Corrado,S.G., Davis, A., Dean, J., Devin,M., Ghemawat, S., Goodfellow, I., Harp, A., Irving,G., Isard, M., Jozefowicz, R., Jia, Y., Kaiser, L., Kudlur,M., Levenberg, J., Mané, D., Schuster, M., Monga, R., Moore, S., Murray, D., Olah, C., Shlens,J., Steiner, B., Sutskever, I., Talwar, K., Tucker,P.,Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals,O., Warden, P., Wattenberg, M., Wicke,M., Yu, Y., and Zheng, X..2015.TensorFlow: Large-scale machine learning on heterogeneous systems. tensorflow.org (accessed 13 April 2021)

Advani, V. 2021. What is Machine Learning? How Machine Learning Works and Future of It?. Great Learning, https://www.mygreatlearning.com/blog/what-is-machine-learning/ (accessed 30 January 2021)

Agarwal, R. 2019. Demystifying Object Detection and Instance Segmentation for Data Scientists. Towards Data Science, https://towardsdatascience.com/a-hitchhikers-guide-to-object-detection-and-instance-segmentation-ac0146fe8e11 (accessed 1 March 2021)

Alzubaidi, F., Mostaghimi, P., Swietojanski, P., Clark, S., Armstrong, R. 2020. Automated Lithology Classification from Drill Core Images Using Convolutional Neural Networks. *Journal of Petroleum Science and Engineering* **197**. https://doi.org/10.1016/j.petrol.2020.107933.

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

Badrinarayanan, V., Kendall, A., andCipolla, R. 2016. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **39**(12): 2481-2495. http://doi.org/10.1109/TPAMI.2016.2644615.

Bhumbra, G. 2018. Deep Learning Improved by Biological Activation Functions.

Bisong E. 2019, Google Colaboratory. In: Building Machine Learning and Deep Learning Models on Google Cloud Platform. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4842-4470-8_7.

Bocangel, W., Sondergeld, C., and Rai, C., 2013. Acoustic Mapping and Characterization of Organic Matter in Shales. SPE-166331, paper presented at the SPE Annual Technical Conference and Exhibition, New Orleans, LA, September 30-October 2.

Brownlee, J. 2019. How to Visualize Filters and Feature Maps in Convolutional Neural Networks. Machine Learning Mastery. https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/ (accessed 15 February 2021)

Brownlee, J. 2019. How to Avoid Overfitting in Deep Learning Neural Networks. https://machinelearningmastery.com/introduction-to-regularization-to-reduce-overfitting-and-improve-generalization-error/ (accessed 11 March 2021)

Brownlee, J. 2019. A Gentle Introduction to Pooling Layers for Convolutional Neural Networks. https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/ (accessed 11 March 2021)

Camp, W.K. 2013. Enhancing SEM Grayscale Images through Pseudocolor Conversion: Examples from Eagle Ford, Haynesville, and Marcellus Shales. Paper presented at the SPE/AAPG/SEG Unconventional Resources Technology Conference, Denver, Colorado, USA, August 2013. doi: https://doi-org.ezproxy.lib.ou.edu/10.1190/urtec2013-240

Chollet, F. 2017. Xception: Deep Learning with Depth Wise Separable Convolutions. 1800-1807. https://doi.org/10.1109/CVPR.2017.195.

Clelland, W.D., and Fens, T.W. 1991. Automated Rock Characterization With SEM/Image-Analysis Techniques. *SPE Form Eval* **6**: 437- 443. doi: https://doi-org.ezproxy.lib.ou.edu/10.2118/20920-PA

Choulwar, A. 2019. The Art of Convolutional Neural Network. Medium, https://medium.com/@achoulwar901/the-art-of-convolutional-neural-network-abda56dba55c (accessed 3 February 2021)

Chollet, F. 2015. Keras. https://github.com/fchollet/keras

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. Mapping of Organic Matter Distribution on the Centimeter Scale with Nanometer Resolution. Paper presented at the SPE/AAPG/SEG Unconventional Resources Technology Conference, Denver, Colorado, USA, August. https://doi-org.ezproxy.lib.ou.edu/10.15530/URTEC-2014-1922757

Curtis, M.E., Sondergeld, C., and Chandra R.. 2019. Visualization of Pore Connectivity Using Mercury Injection Capillary Pressure Measurements, Micro X-ray Computed Tomography, and Cryo-Scanning Electron Microscopy. Paper presented at the SPE/AAPG/SEG Unconventional Resources Technology Conference, Denver, Colorado, USA, July. https://doi-org.ezproxy.lib.ou.edu/10.15530/urtec-2019-637

Dang, S. T., Sondergeld, C. H., and Rai, C. S..2019. Interpretation of Nuclear-Magnetic-Resonance Response to Hydrocarbons: Application to Miscible Enhanced-Oil-Recovery Experiments in Shales. *SPE Res Eval & Eng* **22**: 302–309. https://doi-org.ezproxy.lib.ou.edu/10.2118/191144-PA

Deglint, H.J., Clarkson, C.R., Ghanizadeh, A., DeBuhr, C., Wood, J.M. 2019. Comparison of micro- and macro-wettability measurements and evaluation of micro-scale imbibition rates for unconventional reservoirs: Implications for modeling multi-phase flow at the micro-scale. *Journal of Natural Gas Science and Engineering* **62**: 38-67, https://doi.org/10.1016/j.jngse.2018.11.026.

EIA. 2016. Lower 48 states Shale Plays. https://www.eia.gov/maps/images/shale_gas_lower48.jpg (accessed 10 March 2021)

EIA. 2017. Argentina seeking increased natural gas production from shale resources to reduce imports. https://www.eia.gov/todayinenergy/detail.php?id=29912. (accessed 16 April 2021)

EIA. 2020. How much shale (tight) oil is produced in the United States?. https://www.eia.gov/tools/faqs/faq.php?id=847&t=6 (accessed 10 March 2021)

EIA. 2021. Natural Gas Explained, Where Our Natural Gas Comes From. https://www.eia.gov/energyexplained/natural-gas/where-our-natural-gas-comes-from.php (accessed 10 March 2021)

Fu, J., Rui, Y. 2017. Advances in Deep Learning Approaches for Image Tagging. *APSIPA Transactions on Signal and Information Processing* (6). https://doi.org/10.1017/ATSIP.2017.12.

Gurney, K. 1997. An Introduction to Neural Network. London: UCL Press.

Gupta, I., Rai, C., Sondergeld, C., and Devegowda, D..2018. Rock Typing in Eagle Ford, Barnett, and Woodford Formations. *SPE Res Eval & Eng* **21**: 654–670. https://doi.org/10.2118/189968-PA.

Garg, I., Panda, P., Roy, K. 2019. A Low Effort Approach to Structured CNN Design Using PCA. *IEEE Access*, 1-1. https://doi.org/10.1109/ACCESS.2019.2961960.

Hagan, M. T., Demuth, H. B., Beale, M. H., Jesus, O. D. 2014. Neural Network Design, second edition.

Hubel, D. H., Wiesel, T. N. 1962. Receptive fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex **160**(1): 106–154. https://doi.org/10.1113/jphysiol.1962.sp006837.

He, K., Zhang, X., Ren, S. and Sun, J. 2016. Deep Residual Learning for Image Recognition, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 770-778. https://doi.org/10.1109/CVPR.2016.90.

Immunologia. http://www.ehu.eus/immunologia/stats/regression_y-x.php (accessed 7 February 2021)

Jiang, J, Xu, R, James, S. C., and Xu, C..2021. Deep-Learning-Based Vuggy Facies Identification from Borehole Images. *SPE Res Eval & Eng* **24** (2021): 250–261. https://doi-org.ezproxy.lib.ou.edu/10.2118/204216-PA

Jordan, J. 2017. Convolutional Neural Networks. Jeremy Jordan, https://www.jeremyjordan.me/convolutional-neural-networks/ (accessed 2 February 2021)

Kanani, B. 2019. Activation Functions in Neural Network. Study Machine Learning, https://studymachinelearning.com/activation-functions-in-neural-network/ (accessed 7 February 2021)

Karimpouli, S., Tahmasebi, P.. 2019. Segmentation of Digital Rock Images using Deep Convolutional Autoencoder Networks. *Computers and Geosciences* **126**: 142-150.

Keras. n.d. https://keras.io/api/losses/. (accessed 3 March 2021)

Kazak, A., Simonov, K., and Victor K.. 2020. Machine-Learning-Assisted Segmentation of FIB-SEM Images with Artifacts for Improved of Pore Space Characterization of Tight Reservoir Rocks. Paper presented at the SPE/AAPG/SEG Unconventional Resources Technology Conference, Virtual, July. https://doi-org.ezproxy.lib.ou.edu/10.15530/urtec-2020-2846

Knaup, A. S., Jernigen, J. D., Curtis, M. E., Sholeen, J. W., Borer, J. J., Sondergeld, C. H., and Rai, C.S. 2019. Unconventional Reservoir Microstructural Analysis Using SEM and Machine Learning. Paper presented at the SPE/AAPG/SEG

Unconventional Resources Technology Conference, July, Denver, Colorado, USA. https://doi.org/10.15530/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

Krüger, F. 2016. Activity, Context, and Plan Recognition with Computational Causal Behaviour Models.

LeCun Y., Haffner P., Bottou L., Bengio Y. 1999. Object Recognition with Gradient-Based Learning. In: Shape, Contour and Grouping in Computer Vision. Lecture Notes in Computer Science (1681). https://doi.org/10.1007/3-540-46805-6_19

Lemmens, H.J., Butcher, A.R., and Botha, P.W.S.K.. 2011. FIB/SEM and SEM/EDX: a New Dawn for the SEM in the Core Lab?. *Petrophysics* **52**: 452–456.

Loukas, S. 2020. Multi-class Classification: Extracting Performance Metrics from The Confusion Matrix. Towards Data Science, 19 January 2020, https://towardsdatascience.com/multi-class-classification-extracting-performance-metrics-from-the-confusion-matrix-b379b427a872 (accessed 12 February 2021)

Ma, Y., Liu, K., Guan, Z., Xu, X., Qian, X., and Bao, H. 2018. Background Augmentation Generative Adversarial Networks (BAGANs): Effective Data Generation Based on GAN-Augmented 3D Synthesizing. *Symmetry* (10):734. https://doi.org/10.3390/sym10120734.

McCulloch, W.S., Pitts, W. 1943. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics* **5:** 115–133. https://doi.org/10.1007/BF02478259

Misbahuddin, M. 2020. Estimating Petrophysical Properties of Shale Rock Using Conventional Neural Networks CNN. Paper presented at the SPE Annual Technical Conference and Exhibition, Virtual, October. https://doi-org.ezproxy.lib.ou.edu/10.2118/204272-STU

Moawad, A. 2019. Dense Layers Explained in A Simple Way. Towards Data Science, https://medium.com/datathings/dense-layers-explained-in-a-simple-way-62fe1db0ed75 (accessed 7 February 2021)

Nanakoudis, A., 2019. SEM: Types of Electrons and the Information They Provide. ThermoFisher Scentific. https://www.thermofisher.com/blog/microscopy/sem-signal-types-electrons-and-the-information-they-provide/ (accessed 5 May 2021)

Nasteski, V. 2017. An Overview of the Supervised Machine Learning Methods. *HORIZONS.B* **4**:51-62. https://doi.org/10.20544/HORIZONS.B.04.1.17.P05.

Nagyfi, R. 2018. The Differences Between Artificial and Biological Neural Networks. Towards Data Science, https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d (accessed 30 January 2021)

Paliwal, A. 2018. Understanding Your Convolution Network with Visualizations. Towards Data Science, https://towardsdatascience.com/understanding-your-convolution-network-with-visualizations-a4883441533b (accessed 10 February 2021)

Pan, S. J., and Yang, Q. 2010. A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, **22**(10):1345-1359, https://doi.org/10.1109/TKDE.2009.191.

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.

Priy, S. 2020. Clustering in Machine Learning. Geeks for Geeks. https://www.geeksforgeeks.org/clustering-in-machine-learning/ (accessed 30 January 2021)

Ray, S. 2018. History of AI. Towards Data Science, 11 August 2018, https://towardsdatascience.com/history-of-ai-484a86fc16ef (accessed 30 January 2021)

Radiuk, P. 2017. Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets. *Information Technology and Management Science* **20**:20-24. https://doi.org/10.1515/itms-2017-0003.

Radecic, D. 2019. A Non-Confusing Guide to Confusion Matrix. Towards Data Science, https://towardsdatascience.com/a-non-confusing-guide-to-confusion-matrix-7071d2c2204f (accessed 12 February 2021)

Ren, W., Zhang, M., Zhang, S., Qiao, J., Huang, J.. 2019. Identifying Rock Thin Section Based on Convolutional Neural Networks. Proceedings of 2019 the 9th International Workshop on Computer Science and Engineering, 345-351. https://doi.org/10.18178/wcse.2019.06.052.

Rosenblatt, F. 1958. The Perceptron: A probabilistic Model for Information Storage and Organization in the Brain. *Psychological Review* **65**(6): 386-408.

Rushood, I.A., Alqahtani, N., Wang, Y. D., Shabaninejad, M., Armstrong, R., and Peyman M.. 2020. Segmentation of X-Ray Images of Rocks Using Deep Learning. Paper presented at the SPE Annual Technical Conference and Exhibition, Virtual, October. doi: https://doi-org.ezproxy.lib.ou.edu/10.2118/201282-MS

Saad, B., Negara, A., and Shujath S.A. 2018. Digital Rock Physics Combined with Machine Learning for Rock Mechanical Properties Characterization. Paper presented at the Abu Dhabi International Petroleum Exhibition & Conference, Abu Dhabi, UAE. November. https://doi.org/10.2118/193269-MS

Salhi, R. 2020. Introduction to Neural Networks — Part 1. Towards Data Science, https://towardsdatascience.com/history-of-ai-484a86fc16ef (accessed 30 January 2021)

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh D., and Batra, D.. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *2017 IEEE International Conference on Computer Vision (ICCV)* 618-626. https://doi.org/10.1109/ICCV.2017.74.

Scikit-Learn. n.d. https://scikit-learn.org/stable/supervised_learning.html. (accessed 3 March 2021)

Shao, L., Zhu, F., and Li, X. 2015. Transfer Learning for Visual Categorization: A Survey. *IEEE Transactions on Neural Networks and Learning Systems* **26**(5):1019-1034. https://doi.org/10.1109/TNNLS.2014.2330900.

Sharma, S. 2017. Epoch vs Batch Size vs Iterations. Towards Data Science, 23 September 2017, https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9 (accessed 9 February 2021)

Shridhar, K. 2017. How Close Are Chatbots To Passing The Turing Test?. Chatbots magazine, 2 May 2017, https://chatbotsmagazine.com/how-close-are-chatbots-to-pass-turing-test-33f27b18305e (accessed 30 January 2021)

Soni, D. 2018. Supervised vs. Unsupervised Learning. Towards Data Science, https://towardsdatascience.com/supervised-vs-unsupervised-learning-14f68e32ea8d (accessed 30 January 2021)

Sorokina, K. 2017. Image Classification with Convolutional Neural Networks. Medium. https://medium.com/@ksusorokina/image-classification-with-convolutional-neural-networks-496815db12a8 (accessed 31 January 2021)

Stureborg, R. 2019. Conv Nets for dummies. Towards Data Science, https://towardsdatascience.com/conv-nets-for-dummies-a-bottom-up-approach-c1b754fb14d6 (accessed 31 January 2021)

Sharma, P. 2019. Decoding the Confusion Matrix. Towards Data Science, https://towardsdatascience.com/decoding-the-confusion-matrix-bb4801decbb (accessed 20 March 2021)

Stephenson. M. H. 2015. Shale gas in North America and Europe.

Su, C., Xu, S., Zhu, K., Zhang, X. 2020. Rock Classification in Petrographic Thin Section Images Based on Concatenated Convolutional Neural Networks. *Earth Science Informatics* **13**. https://doi.org/10.1007/s12145-020-00505-1.

Simonyan, K., Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z. B. 2015. Rethinking the Inception Architecture for Computer Vision.

Turing, A. M. 1950. Computing Machinery and Intelligence. *Mind* **LIX**(236): 433–460. https://doi.org/10.1093/mind/LIX.236.433.

Torghabeh, A.K., Rezaee, R., Moussavi-Harami, R. 2014. Electrofacies in gas shale from well log data via cluster analysis: A case study of the Perth Basin, Western Australia. *cent.eur.j.geo.* **6**: 393–402. https://doi.org/10.2478/s13533-012-0177-9

Wang, G., Ju, Y., Carr, T. R., Li, C., and Cheng, G. (2014). Application of artificial intelligence on black shale lithofacies prediction in Marcellus Shale, Appalachian Basin. Unconventional Resources Technology Conference. http://doi.org/10.15530/URTEC-2014-1935021

Wu, Y., and Misra, S. 2020. Intelligent Image Segmentation for Organic-Rich Shales Using Random Forest, Wavelet Transform, and Hessian Matrix. *IEEE Geoscience and Remote Sensing Letters*, **17**(7): 1144-1147, July, http://doi.org/10.1109/LGRS.2019.2943849.

Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K. 2016. Aggregated Residual Transformations for Deep Neural Networks

Xu, C., Misra, S., Srinivasan, P., and Ma, S..2019. When Petrophysics Meets Big Data: What can Machine Do?. Paper presented at the SPE Middle East Oil and Gas Show and Conference, Manama, Bahrain. doi: https://doi.org.ezproxy.lib.ou.edu/10.2118/195068-MS.

Yang, L. 2020.Why Sigmoid: A Probabilistic Perspective, https://towardsdatascience.com/why-sigmoid-a-probabilistic-perspective-42751d82686 (accessed 10 March 2021)

Yang, K., Qinami, K., Li Fei-Fei, Deng, J., Russakovsky, O.. 2019. Towards Fairer Datasets: Filtering and Balancing the Distribution of the People Subtree in the ImageNet Hierarchy. https://image-net.org/update-sep-17-2019 (accessed 6 May 2021)

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., and Lipson, J. 2015. Understanding Neural Networks Through Deep Visualization.

Zurück, 2020. SimLoss: Class Similarities in Cross Entropy. https://www.informatik.uni-wuerzburg.de/datascience/news/single/news/simloss-class-similarities-in-cross-entropy/. (accessed 7 April 2021)

Appendix

# of Layers	1	2	3	5
# of Filters	2	2, 2	2, 4, 4	2, 4, 4, 8, 8
Accuracy %	73	76	86	85
Loss	0.73	0.64	0.34	0.39
Training time	22s	29s	38s	45s

# of Layers	1	2	3	5
# of Filters	4	4, 4	4, 8, 8	4, 8, 8, 16, 16
Accuracy %	79	87	87	93
Loss	0.61	0.46	0.34	0.20
Training time	50s	56s	62s	117s

# of Layers	1	2	3	5
# of Filters	8	8, 8	8, 16, 16	8, 16, 16, 32, 32
Accuracy %	83	91	94	94
Loss	0.46	0.27	0.18	0.18
Training time	30s	44s	98s	64s

# of Layers	1	2	3	5
# of Filters	16	16, 16	16, 32, 32	16, 32, 32, 48, 48
Accuracy %	87	94	96	96
Loss	0.39	0.17	0.13	0.10
Training time	109s	55s	61s	133s

# of Layers	1	2	3	5
# of Filters	32	32, 32	32, 64, 64	32, 64, 64, 96, 96
Accuracy %	81	93	96	95
Loss	0.51	0.18	0.12	0.15
Training time	89s	92s	96s	166s

Fig. A1 - Shallow vs. deep network performance on 25nm/px resolution (3x3 μm field-of-view) dataset with 8 plays.

# of Layers	1	1	1	1	1	1
# of Filters	1	2	4	8	16	32
Accuracy %	67	81	89	89	91	89
Loss	0.81	0.50	0.30	0.30	0.24	0.29
Training time	26s	36s	35s	45s	64s	118s

Fig. A2 - Shallow 1-layerp network performance on 10nm/px resolution (1x1 μm field-of-view) dataset with 8 plays.

# of Layers	1	2	3	5
# of Filters	2	2, 2	2, 4, 4	2, 4, 4, 8, 8
Accuracy %	81	84	89	87
Loss	0.50	0.40	0.29	0.33
Training time	36s	31s	52s	31s

# of Layers	1	2	3	5
# of Filters	4	4, 4	4, 8, 8	4, 8, 8, 16, 16
Accuracy %	89	90	96	94
Loss	0.30	0.27	0.12	0.17
Training time	35s	31s	35s	34s

# of Layers	1	2	3	5
# of Filters	8	8, 8	8, 16, 16	8, 16, 16, 32, 32
Accuracy %	89	95	96	97
Loss	0.30	0.14	0.10	0.10
Training time	45s	32s	82s	75s

# of Layers	1	2	3	5
# of Filters	16	16, 16	16, 32, 32	16, 32, 32, 48, 48
Accuracy %	91	95	97	96
Loss	0.24	0.13	0.10	0.12
Training time	64s	36s	39s	61s

# of Layers	1	2	3	5
# of Filters	32	32, 32	32, 64, 64	32, 64, 64, 96, 96
Accuracy %	89	97	97	97
Loss	0.29	0.10	0.09	0.10
Training time	118s	37s	32s	123s

Fig. A3 - Shallow vs. deep network performance on 10nm/px resolution (1x1 μm field-of-view) dataset with 8 plays.

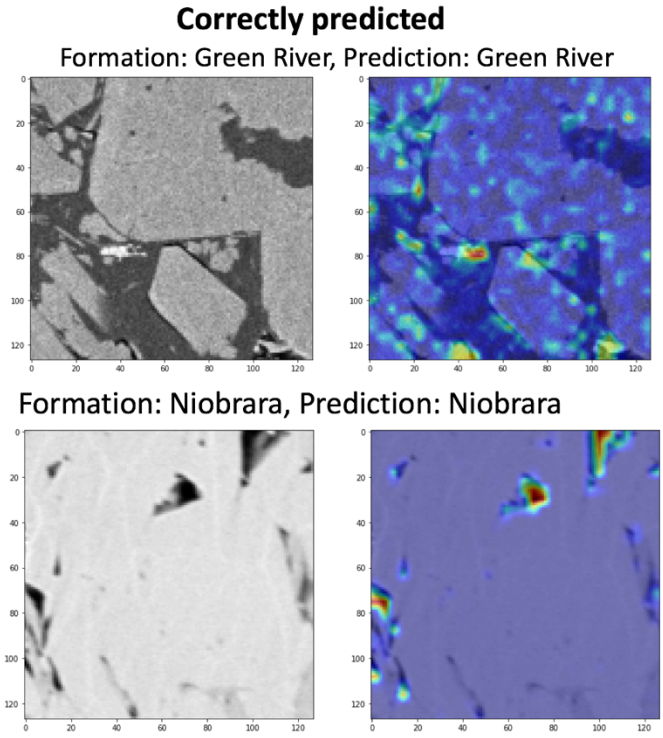


Fig. A4 - Left figures are original 50nm/px resolution (6x6 µm field-of-view) SEM images, right figures are heatmaps output from the 1 layer, 4 filter model with 22 plays.

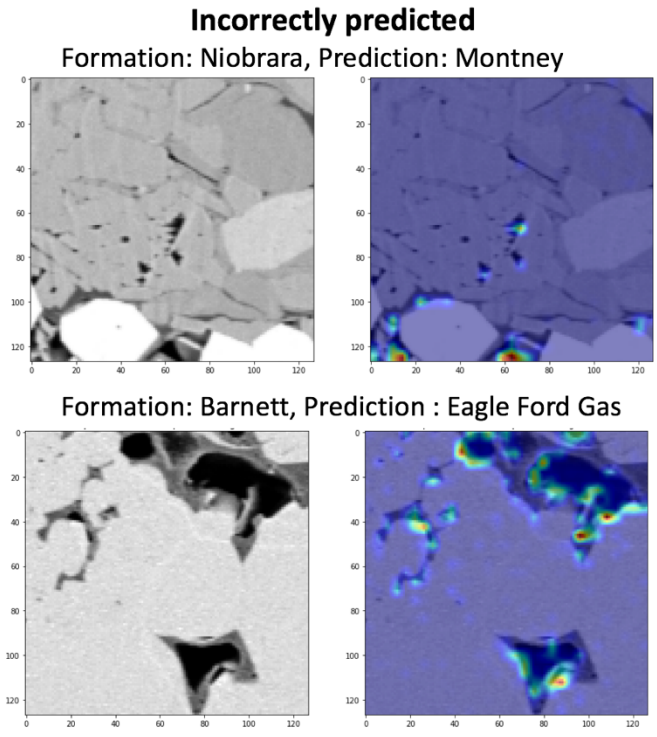


Fig. A5 - Left figures are original 50nm/px resolution (6x6 µm field-of-view) SEM images, right figures are heatmaps output from the 1 layer, 4 filter model with 22 plays.

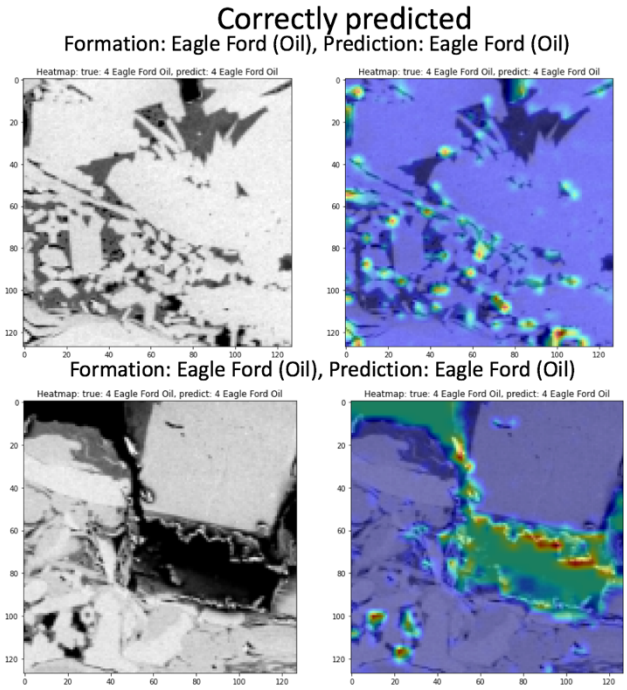


Fig. A6 - Left figures are original 50nm/px resolution (6x6 µm field-of-view) SEM images, right figures are heatmaps output from the 1 layer, 4 filter model with 22 plays.

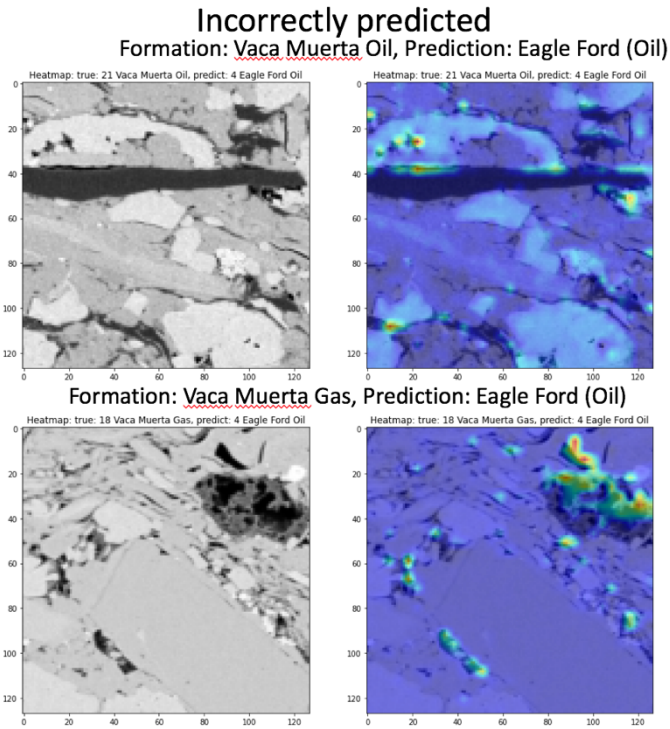


Fig. A7 - Left figures are original 50nm/px resolution (6x6 µm field-of-view) SEM images, right figures are heatmaps output from the 1 layer, 4 filter model with 22 plays.

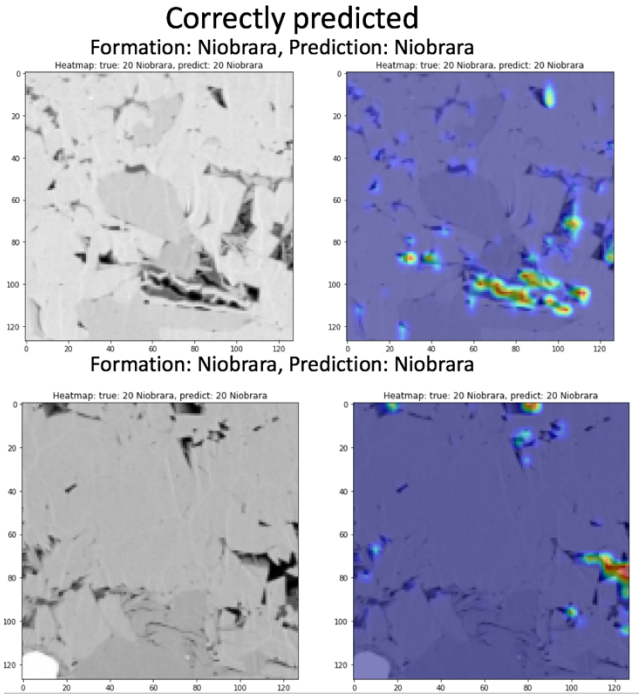


Fig. A8 - Left figures are original 50nm/px resolution (6x6 μm field-of-view) SEM images, right figures are heatmaps output from the 1 layer, 4 filter model with 22 plays.

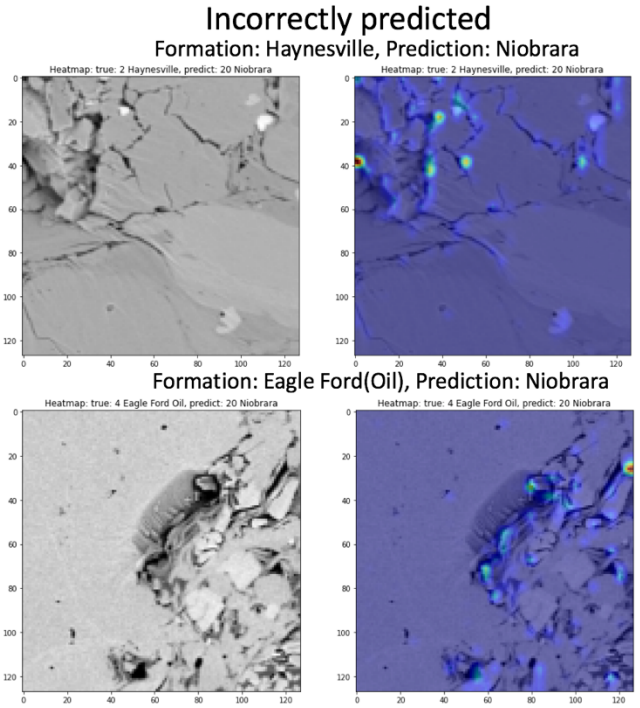


Fig. A9 - Left figures are original 50nm/px resolution (6x6 μm field-of-view) SEM images, right figures are heatmaps output from the 1 layer, 4 filter model with 22 plays.

# of Layers	1	2	2	3	3	5
# of Filters	2	2, 2	2, 4	2, 4, 4	2, 4, 8	2, 4, 4, 8, 8
Accuracy %	69	70	79	82	84	85
Loss	0.90	1.0	0.58	0.50	0.45	0.56
Training time	71s	169s	162s	75s	100s	220s

# of Layers	1	2	2	3	3	5
# of Filters	4	4, 4	4, 8	4, 8, 8	4, 8, 16	4, 8, 8, 16, 16
Accuracy %	77	81	89	92	90	92
Loss	0.73	0.55	0.34	0.26	0.28	0.26
Training time	113s	233s	158s	139s	144s	142s

# of Layers	1	2	2	3	3	5
# of Filters	8	8, 8	8, 16	8, 16, 16	8, 16, 32	8, 16, 16, 32, 32
Accuracy %	73	88	91	93	94	95
Loss	0.81	0.36	0.28	0.22	0.20	0.16
Training time	70s	145s	151s	147s	129s	135s

# of Layers	1	2	2	3	3	5
# of Filters	16	16, 16	16, 32	16, 32, 32	16, 32, 48	16, 32, 32, 48, 48
Accuracy %	79	91	92	94	95	95
Loss	0.75	0.28	0.25	0.18	0.16	0.15
Training time	109s	93s	400s	138s	240s	186s

# of Layers	1	2	2	3	3	5
# of Filters	32	32, 32	32, 64	32, 64, 64	32, 64, 96	32, 64, 64, 96, 96
Accuracy %	80	93	94	96	97	96
Loss	0.80	0.25	0.21	0.14	0.11	0.14
Training time	138s	213s	277s	126s	204s	320s

Fig. A10 - Shallow vs. deep network performance on 50nm/px resolution (6x6 µm field-of-view) dataset with 22 plays.

# of Layers	1	1	1	1	1
# of Filters	1	4	8	16	32
Accuracy %	51	76	79	82	81
Loss	1.38	0.69	0.70	0.56	0.57
Training Time	61s	85s	85s	109s	138s

Fig. A11 - Shallow network performance on 25nm/px resolution (3x3 µm field-of-view) dataset with 22 plays.

# of Layers	1	2	2	3	3	5
# of Filters	2	2, 2	2, 4	2, 4, 4	2, 4, 8	2, 4, 4, 8, 8
Accuracy %	64	78	83	85	87	88
Loss	1.24	0.62	0.50	0.43	0.39	0.49
Training time	86s	67s	115s	140s	220s	222s

# of Layers	1	2	2	3	3	5
# of Filters	4	4, 4	4, 8	4, 8, 8	4, 8, 16	4, 8, 8, 16, 16
Accuracy %	76	85	85	89	90	91
Loss	0.69	0.44	0.45	0.32	0.29	0.27
Training time	85s	87s	168s	133s	121s	167s

# of Layers	1	2	2	3	3	5
# of Filters	8	8, 8	8, 16	8, 16, 16	8, 16, 32	8, 16, 16, 32, 32
Accuracy %	79	88	90	92	94	93
Loss	0.70	0.38	0.34	0.26	0.19	0.65
Training time	85s	138s	196s	198s	109s	177s

# of Layers	1	2	2	3	3	5
# of Filters	16	16, 16	16, 32	16, 32, 32	16, 32, 48	16, 32, 32, 48, 48
Accuracy %	82	91	92	93	94	95
Loss	0.56	0.33	0.40	0.21	0.18	0.15
Training time	109s	131s	100s	105s	439s	209s

# of Layers	1	2	2	3	3	5
# of Filters	32	32, 32	32, 64	32, 64, 64	32, 64, 96	32, 64, 64, 96, 96
Accuracy %	81	93	92	95	96	95
Loss	0.57	0.23	0.42	0.15	0.15	0.15
Training time	138s	177s	250s	133s	338s	270s

Fig. A12 - Shallow vs deep network performance on 25nm/px resolution (3x3 µm field-of-view) dataset with 22 plays.