J. Dawn. Res. (2021)
www.dawningresearch.org

Journal of Dawning Research

New Computational Model For Mitosis To Save Lives From Cancer

Research Article

Michael Batavia[1]*

1 The Bronx High School of Science, Bronx, NY 10468

Abstract: In this paper, individualized pre-diagnoses to the many Americans affected with an abnormal form of mitosis in the breast (breast cancer) using a custom convolutional neural network (CNN) are proposed. It was predicted earlier this year that 281,550 cases of invasive breast cancer would be detected in women and 44,130 of them would be expected to die from the disease, making the disease the 2nd leading cause of cancer death in women. Although the effects of breast cancer can be limited with treatments such as chemotherapy, the cancer must first be diagnosed before being treated. As the five-year survival rate declines among untreated patients from 93% at stage 0 cancer to 15% at stage 4 cancer, it is critical that the cancer be diagnosed as early as possible before treatments become ineffective. Using a custom parallel distributed CNN with synchronous gradient descent, this research presents 83% validation accuracy and a 89% ROC-AUC score, 10% better than pathologist classification in both early and later stages of cancer, in order to classify benign and metastatic lymph node tumors in the breast for patients of all stages and determine where cancer cells have spread for effective treatment on those areas. The experimental data was acquired from a dataset in The Cancer Imaging Archive and from the *PatchCamelyon* dataset, which contains de-identified histopathological images of tumor tissue in 20x and 40x magnification. Consequently, this research proposes the best setup for analyzing cancerous lymph node tumors in order to maximize neural network accuracies in more generic CNNs. The impact of this study allows for the use of similarly created synchronous CNNs to be used to detect related cancers on the increasing number of patients diagnosed with these diseases yearwide.

Keywords: Parallelized Processing • Breast Cancer Pre-Diagnosis • Synchronized Gradient Descent • Tumor Magnification

1. Introduction

1.1. Background & Terminology

This project proposes a parallel distributed synchronous CNN that can detect an abundance of abnormal mitosis (cancer) in lymph nodes using the Breast Metastases to Auxiliary Lymph Nodes (*TCIA*) dataset available from The Cancer Imaging Archive [1] and from the *PatchCamelyon* dataset, accessible from the TensorFlow Datasets API, which can differentiate between tumor category 10% better than pathologist classification for early

* *E-mail: mbatavia27@gmail.com*

diagnosis and roughly 10% better than pathologist classification in later stages of cancer [2–4]. The importance in developing such a network is two-fold. First, many hospitals in the New York City area do not have the resources to effectively diagnose the increasing amount of patients with the disease before the cancer progresses. The longer doctors take to get back to their patient, the greater the progression of the breast cancer stage and the less effective their treatments on that specific stage [5]. This process can be generalized to the 30% of women affected with breast cancer in 2020 and the 80% of those who experienced invasive breast cancer from that group [6]. To encourage the detection of breast cancer in lymph nodes, researchers need to find ways to improve the early detection of cancer in patients more efficiently and make preliminary individualized treatment recommendations based on the guidance of neural networks.

In recent years, deep learning and neural networks have been able to fill the hole of binary and multi-class classification with their entrance into the data analysis, medical and literary fields. In the past century, classification accuracy in neural networks has improved due to new techniques such as data scaling and the advent of using GPUs to speed up calculations. In particular, CNNs have been the focus for image classification due to their ability to find patterns in image inputs and generalize to correctly classify them in the input dataset. For example, these networks have been used to improve detection of uterine cancer via a custom U-net infrastructure in order to analyze the encephalogram (graph of voltage fluctuations from the brain) of hospital patients to monitor the potential of a seizure with 4% accuracy over the current laboratory seizure classifying methods [7]. Unfortunately, deep learning suffers from intensive use of resources and the large quantity of input data, which take several days to process [8]. With the growing size of datasets that are processed for machine learning, it is important that scientists offer solutions to prevent out of memory errors in deep learning that halt medical advances and detection of cancers. This has led to the creation of optimizers to speed up the calculations in the network and to distribute the network across multiple processors.

Synchronous and asynchronous stochastic gradient descent (SGD) are two types of optimizers used to calculate the loss of the neural network. In asynchronous SGD, the parameter server [central server] fetches the newest weights [feature "importance" factors] instantaneously from the processors once they finish. Synchronous SGD, on the other hand, has a parameter server that waits for all weights to be calculated and aggregated in order to determine the best weights for the highest classification accuracy. Although synchronous SGD must wait for the slowest processor to finish before aggregating the weights, the limitation of asynchronous SGD is that the server fetches the weights too fast and does not obtain the absolute best ones. However, the problems of synchronous SGD can be amended by adding backup workers to calculate the newest weights and dropping the results of the slowest processors. Synchronous SGD was chosen for this study because the backup workers allow for the optimizer to converge at least as fast as asynchronous SGD with a small accuracy improvement [9]. Additionally, parallel distributed processing (PDP) — the data parallelism used in this study — allows for the computations on the data to be distributed across the multiple processors in a computer, exponentially decreasing the time taken to compute the accuracy. By using PDP and synchronous SGD effectively, a novel neural network could optimize past research in synchronous SGD for major impact on biomedical industries [8].

Due to this optimization of synchronous SGD, there are major impacts that the research could contribute to the scientific community and healthcare. First, many CNNs take several days to fully complete their training on large datasets, which is critically disadvantageous to cancer patients, whose treatment gradually becomes ineffective the more progressed the stage of cancer. Although chemotherapy treatment on breast cancer gives patients a 93% five year survival rate at stage 0 cancer, there is only a 15% five year survival rate at stage 4 cancer [10]. Recent advances in the field of deep learning though have led to the discovery of auto-generating neural networks and the generation of optimal hyperparameters to find the best accuracy for a CNN [11, 12]. Optimizing hyperparameters is one of the safest ways to maximize the validation accuracy while keeping the structure of the CNN constant. As a result, PDP, auto-hyperparameter tuning and auto neural network tuning can be used rapidly to give a quick and individual cancer pre-diagnosis for the doctor to confirm without waiting for the official diagnosis. Faster and definite diagnosis of invasive breast cancer could drastically reduce the amount of individuals who die each year from ineffective treatments at progressed stages of cancer.

Given the usage of synchronous SGD with a custom parallel distributed CNN and auto-hyperparameter tuning, the work outpaces standards in the field. Most research uses ADAM or RMSProp for optimization of their neural networks due to the fast convergence of the optimizers to obtain a decent accuracy and quality results [13]. However, a SGD optimizer was used here, as it has shown to be effective for most deep learning tasks when combined with Nesterov optimization and learning rate decay; producing more optimal solutions with greater validation accuracy than more adaptive methods such as ADAM. Furthermore, this approach also accounts for the difference in breast cancer data magnification in different datasets (not accounted for in many papers). This is shown from the 20x magnification of the lymph node data in the *TCIA* dataset and the 40x magnification in the *PatchCamelyon* dataset. In this paper, it is shown that images with a 40x magnification lead to a statistically better accuracy than those with 20x magnification.

Thus, a custom CNN that elevates previous research with parallel distributed networks and effectively pre-diagnose uncontrolled mitosis in the breast in early stages was created in this paper to prevent such cancers from progressing. With the number of invasive cancer identifications rising in 2021 compared to noninvasive cancer, it is of critical importance to find ways to pre-diagnose the disease before common treatments like chemotherapy become ineffective. By using auto-hyperparameter tuning and Nesterov accelerated gradient descent, the end solution is optimal, robust, smooth and has the ability to be spread across many GPUs and various contexts. With a 83% validation accuracy and a 89% ROC-AUC score, 10% better than pathologist classification in early stages of cancer and roughly 10% better than pathologist classification in later stages of cancer, the network benefits from the use of synchronous SGD compared to the slow learning rate experienced by asynchronous SGD.

1.2. Literature Review

To improve the efficiency and accuracy of CNNs, several networks such as ResNet50, HOGWILD and FCNs have been tested in the ImageNet Large Scale Visual Recognition Challenge, which has asked neural networks to categorize 200,000 images into 200 categories since 2010 (see the review detailed in [14] for further description of the ImageNet data). The dataset behind the ImageNet challenge has become a model to test image classification algorithms with overlapping objects, large size and need for multi-class classification in the same image. Despite the success of past neural networks in sequentially solving the challenges, there were minor limitations that could be improved upon, which were completed with this new and derived CNN revised with a medical focus.

One of the first CNNs to achieve 1^{st} place in the ImageNet challenge in 2015 was ResNet50. This network capitalized on using residual functions to separate the data such that select pieces of image data can skip through neural network layers before all layers are concatenated in a final step for detection. This technique is effective because no extra parameters or computational complexity need to be provided to the model; only the residual connections only improve the model's accuracy [15]. Pre-trained networks like ResNet50 are used in this paper not only for their improvement in this model's accuracy but for the ability to deal with small datasets when custom built neural networks are unable to train well on them. The VGG16 network was used in this paper for this very reason but trained on several configurations, ResNet50 presents at least a 3.57% error rate on the test set from the ImageNet data.

However, one of the limitations of this pre-trained network is the large amount of parameters needed for storing the network. Although ResNet50 is one of the golden standards for pre-trained image classification networks, the innovating skip connections add more parameters to store and compute than a typically constructed CNN; consequently, it is not the most efficient model based on the amount of computations needed to find these parameters for asymptotically large data. Therefore, fellow researchers looked toward faster networks with less parameters to save in order to speed up future predictions.

One change to the model architecture to the ResNet50 architecture was using only convolutional layers compared to the common technique of alternating convolutional, pooling and fully connected (dense) layers in order to concatenate the final outputs. This was done in order to recognize more specific regions of images and it has been shown to be fast at converging due to the continuous decrease in stride and increase in feature detail of the model over time throughout the network [16]. Tested on the ImageNet dataset, the scientists recorded the pixel accuracy, mean IU (average of true positives recorded) and received a 20% increase in accuracy over the leading image segmentation algorithm; it was also able to generalize well on the SIFT Flow dataset, which measures multi-class classification with 33 picture categories in 3 background categories.

However, a limitation of the paper includes not realizing the bounds of IU on the dataset for measuring accuracy and not being able to detect the best learning rate during training, which they empirically chose through a range of 3 learning rate values.

To connect the improvements in CNNs with distributed parallel processing, the HOGWILD neural network was made. The key idea behind HOGWILD expands on past bottlenecks in synchronous processing, where terabytes of data are unable to be processed efficiently in time from slowdown due to low latency and high bandwidth [17]. Therefore, the researchers behind the network created an asynchronous algorithm without locking (the main problem described above) that allowed the processors to update weights in a parameter server as soon as they are calculated. From the use of their algorithm, they proposed that the weights can be rapidly updated in a rate from $\frac{1}{\sqrt{k}}$ to a rate of $\frac{1}{k}$, where k is the number of machines.

Despite this improvement in updating weights, the algorithm is specific to sparse data and only applies to asynchronous processing, which may not identify the optimal weights due to the constant updates in asynchronous processing.

To improve on the drawbacks of HOGWILD, the Rudra network was designed in order to study different types of parameter updates to a central custom parameterized server. In the system, the weights upload in a tree model to the server to eliminate the network traffic that occurs in asynchronous processing when a barrage of weights are uploaded to the network and in synchronous processing when lagging processors lead to a slower weight update. Using the CIFAR10 and ImageNet data, the scientists studied the efficiency of synchronous, asynchronous and a softsync [the server updates the weights only after backpropagation through the network several times] training protocol with the tree model to determine the best protocol and result in the best accuracy and least time taken. Using 30 learners and a mini-batch size of 4, they report a 18.09% error on the test set over 140 epochs and 1,573 seconds [18].

Although this is a considerable improvement to past processes, the researchers encountered limitations to the mini-batch size. They recommend that future algorithms must complete training in small mini-batch sizes, due to slower computation speeds at bigger batch sizes for large data. However, this conclusion has not been proved by other papers.

These improvements in CNNs have been extended to the medical field to aid surgical procedures and x-ray diagnosis. Uterine segmentation has often been used as a treatment for uterine fibroids - benign tumors in the female pelvis. Traditionally, this technique has often been done using image processing tasks such as using a support vector machine or by comparing the means between the predicted tumor areas and those diagnosed by professional radiologists [19]. However, a custom designed neural network, dubbed U-net, was used with patients with endometrial cancer, cervical cancer and leiomyoma to see if the tumors could be detected more accurately compared to those diagnosed by a professional radiologist [20]. With a dice similarity coefficient of 0.82 (compares similarity of two samples - the diagnosis of radiologists against the U-net), Kurata and his colleagues in [20] showed that segmentation of the uterus is faster and clinically feasible with a U-net compared to waiting for diagnoses from pathologists.

Despite these improvements, one limitation of their study includes not considering the anatomical context of the uterus, which is often difficult to take proper images of for precise pattern recognition. Furthermore, the U-net in the scientists' research was later eclipsed in its efficiency by the use of newer techniques such as graph cuts to improve the speed of pre-diagnosing the tumors.

Overall, a variety of studies have shown the promise of convolutional neural networks and distributed processes for faster processing and increased accuracy of classifying images. However, the limitations in the research prevent the created CNNs to be widely applicable to cancer detection tasks without a lengthy initial setup by radiologists such as the empirical choice of the learning rate in the non-dense networks above. In addition, inefficiencies in finding the global maximum in networks such as HOGWILD can be likely fixed by applying synchronous SGD. By extending these researchers' limitations with PDP, synchronous SGD and anatomically correct images of lymph nodes, this research overcomes these limitations with a more in-depth and efficient classification of malignant tumors in breast cancer lymph nodes with a 8% difference from leading pathologist lymph node classification and with a baseline neural network accuracy on the CIFAR10 dataset of 91.75%.

1.3. Research Problem

Based on the limitations of these experiments, a research question was constructed to address the effect of parallel distributed networks, slide magnification and synchronous SGD on sample breast cancer datasets. The two experiments addressed in this report were: "How can an effective parallel distributed CNN make individualized predictions in less time than pathologist manual classifcation for detecting cancer tumors?" and "How does lymph node slide magnification affect the performance of neural networks?" To test this, benchmarked parallelized processing on the sample CIFAR10 dataset was completed to test how the custom CNN compares to leading machine learning classifiers and later to manual pathologist classification in the real-world datasets. With breast cancer being continuously marked as the most commonly diagnosed cancer among American women year after year, using data parallelism could make informed diagnosis and treatments for breast cancer in the *TCIA* and *PatchCamelyon* datasets faster and more accurately [6]. This was done using the Keras framework in Python code, as will be described in the methodology section below.

1.4. Significance

This study has both an academic significance for researchers and a technical significance for the general audience. The academic significance is the creation of a CNN that can make medical diagnosis via PDP in multiple GPUs, a technique not currently used by radiologists and data scientists. By fine-tuning and hypertuning CNNs with different hyperparameters, hospitals can find the best version of the network for the accurate pre-diagnosis in a variety of cancers. Furthermore, this research reports the best slide magnification for breast cancer datasets so that this magnification can become the common standard or extended for radiologists to scan their data with for maximal use in CNN classification. The difference between this verified magnification to lesser magnifications was compared as well to quantify the difference in accuracy for improperly magnified histopathological images [21]. Finally, the usage of synchronous gradient descent and PDP in this classification allows for this CNN to remain a competitor in efficiently analyzing datasets approaching the scale of petabytes in the future.

The technological significance of this study is that the custom CNN can detect the increasing amount of individualized breast cancer diagnoses made in the United States. Earlier this year, it was predicted that 12% of women will develop breast cancer in their lifetime and that 261,550 cases of it will be invasive in women and 2,650 in men [6]. Without quick and informed diagnosis from doctors via mammography images, invasive breast cancer can spread to other parts of the body and permanently damage immunity by destroying the lymph cells around the breast area. It is of critical importance to diagnose breast cancer by going to a hospital when the first signs are detected. Currently available treatments for cancer such as chemotherapy and radiology have the best effectiveness during early stages of cancer; as the stages of cancer progress, the success of these treatments drastically decrease as the cancer cells become resistant to the treatments. This is particularly important to the groups more vulnerable to breast cancer such as black women and Ashkenazi Jewish women, whose higher rate of BRCA mutations put them at higher risk to the cancer [6]. As a result, pre-diagnosis immensely increases the chance of early diagnosis to prevent the cancer from progressing. According to the research done by Kirilov and Borovska, the five-year survival rate for breast cancer declines from 93% at stage 0 cancer to just 15% at stage 4 cancer [10]. This custom CNN is able to pre-diagnose lymph node tumors by roughly 10% as the stage of cancer progresses, likely being able to increase the percentage of patients who survived after being diagnosed for the cancer after a period of 5 years to 32% in hospitals. By testing the CNN on the histopathological images outside of the datasets used, this CNN will be able to be extended for use in detecting cancer tumors in hospitals nationwide.

1.5. Research Hypothesis

The research hypothesis studied in this paper is that if asynchronous CNNs with PDP over multiple GPUs are applied to detect malignant lymph node tumors in the breast and are controlled for magnification, then the CNNs will receive an accuracy of greater than 80% detecting the tumors. In order to test that the these types of CNNs are sufficient to handle breast cancer images, they will first be tested on the CIFAR10 dataset, which has a larger training set available of 32 x 32 x 3 RGB images that present the same task as breast cancer diagnosis on a larger scale but can be benchmarked to asses baseline performance.

2. Methodology

The key innovations in this study was the creation of a parallelized synchronous stochastic gradient descent algorithm in a CNN to differentiate between benign and malignant tumors in breast cancer lymph nodes and the statistically significant improvement in accuracy by using greater magnifications while scanning lymph node images. The coding was completed through the use of Google Colaboratory (`https://colab.research.google.com`), a browser based coding interface suitable for machine learning, Python 3.6 with the TensorFlow Keras 2 library for the model, the `matplotlib` library for visualization and the `numpy` library for image normalization.

The deployment of the model on 2 GPUs was done using a paid PaperSpace multi-GPU virtual machine on the cloud; all of the necessary libraries were installed on a virtual environment with the `pip` package manager before deployment. To benchmark the performance of the CNN on everyday datasets, the CIFAR10 dataset was used to test performance, which contains 60,000 32 x 32 x 3 RGB images in 10 categories; both CIFAR10 and breast cancer dataset results are recorded here [22].

Before the data was pre-processed, it was converted into file formats easily parsable by machine learning algorithms. As mentioned above, the CIFAR10 data is composed of 60,000 32 x 3 x 3 RGB images in 10 categories; the data was easily imported through a HTML request in Keras and trained in one batch. The situation was similar in the *PatchCamelyon* dataset where the data was imported from the TensorFlow Datasets library (each image was a 96 x 96 x 3 RGB image) but only 5% of the data was used to load it in one batch for consistency. All images from the *PatchCamelyon* dataset were loaded in 40x magnification. For the *TCIA*, each of the 130 images were converted from the SVS file format to PNG format via use of the `pyvips` library and resized to 96 x 96 x 3. All images loaded from this dataset were loaded at 20x magnification. The first experiment in this study about the robustness of the custom CNN was completed with the 5% of the *PatchCamelyon* data whereas the second experiment was conducted by loading 130 images from both the *PatchCamelyon* and the *TCIA* datasets to contrast the difference in classifying lymph node tumors of different magnifications.

After the initial preprocessing of downloading and converting the images, the custom CNN was constructed based on the architecture of the VGGNet16 and Inception CNN models. The input shape of the each group of images corresponded to the experiment being run. All images for the CIFAR10 benchmark were 32 x 32 x 3 while all other images were 96 x 96 x 3. The first convolutional layer of the custom CNN was configured with a 3 x 3 kernel and 128 filters. The weights were initialized with He initialization with “same” padding. He initialization initializes the weights to values drawn from a distribution of numbers with a mean of 0 and a standard deviation given by Equation 1.

$$\sigma(x) = \sqrt{\frac{2}{x}} \tag{1}$$

In this case, the input parameter of x would be 32 or 96 depending on the image input shape. These initial weight values speed up processing in the CPUs and GPUs and can be easily computed as the network continues to learn compared to random weight initialization.

After the initialization as shown in Figure 1, each convolutional layer was followed up with batch normalization and a LeakyReLU activation layer to proceed to the next layer based on its weights. The kernel size also alternated during layers; this was done to reduce parameter size as convolutional layers with a 1 x 3 kernel and a 3 x 1 kernel have the same effectiveness as a layer with a 3 x 3 kernel. Each layer set doubled the amount of filters than the first; two measures were taken to reduce overfitting the data. As seen in Figure 1, a dropout layer was applied roughly after every two convolutional layers and a ℓ_2 regularization parameter was applied to each layer in the model to regularize the weights from vanishing. The values for the penalty term and dropout ratio were determined through hyperparameter tuning. Compared to the standard VGGNet ending of three fully connected layers, this network ended with an average pooling layer to collect the outputs, followed by softmax activation with the number of categories to receive the output of the model. For the CIFAR10 data, the softmax was set to 10 outputs while for the TCIA dataset, the softmax was set to 2 outputs.

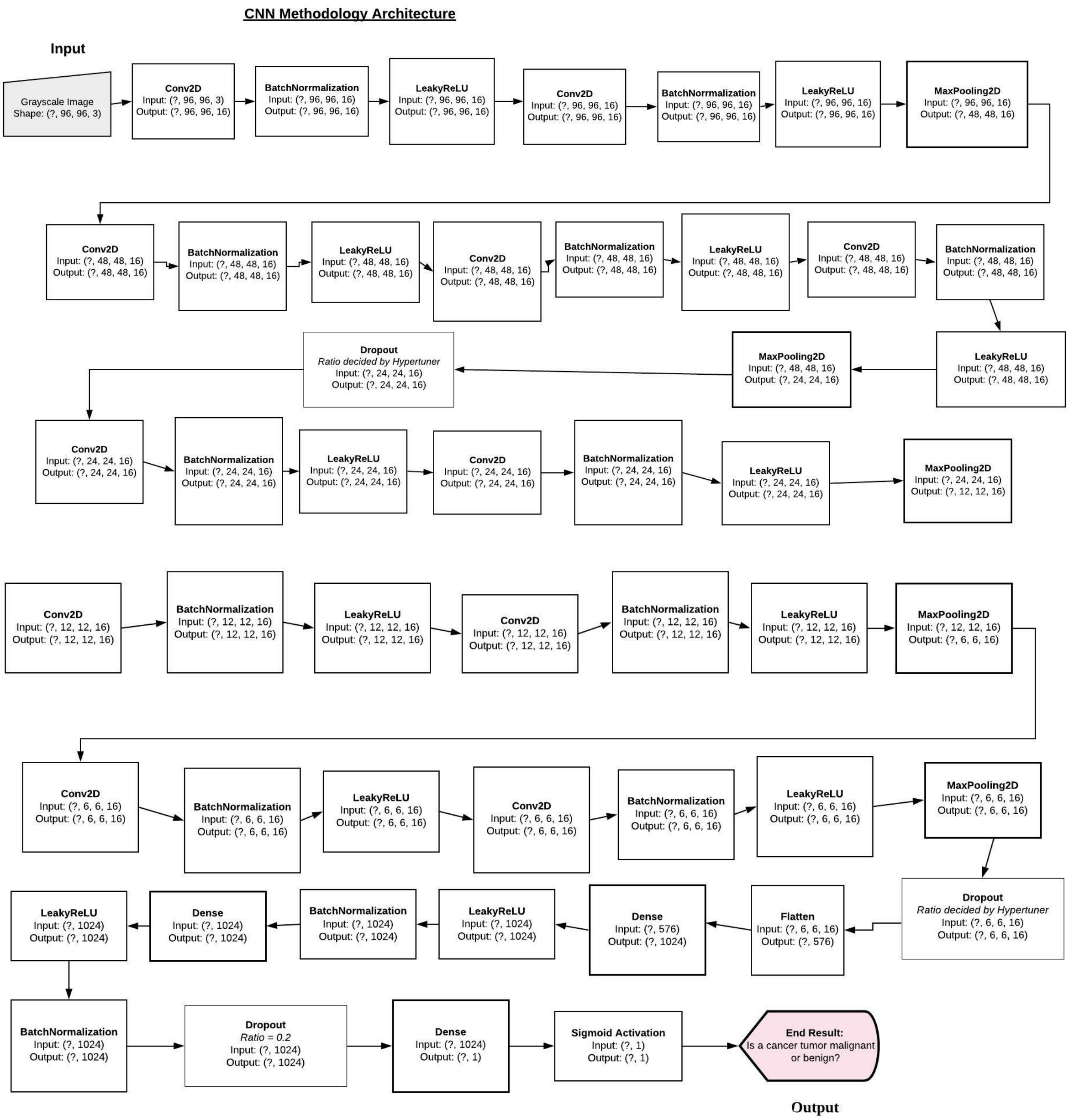

CNN Methodology Architecture
Input
Grayscale Image Shape: (?, 96, 96, 3)
Conv2D Input: (?, 96, 96, 3) Output: (?, 96, 96, 16)
BatchNorrmalization Input: (?, 96, 96, 16) Output: (?, 96, 96, 16)
LeakyReLU Input: (?, 96, 96, 16) Output: (?, 96, 96, 16)
Conv2D Input: (?, 96, 96, 16) Output: (?, 96, 96, 16)
BatchNorrmalization Input: (?, 96, 96, 16) Output: (?, 96, 96, 16)
LeakyReLU Input: (?, 96, 96, 16) Output: (?, 96, 96, 16)
MaxPooling2D Input: (?, 96, 96, 16) Output: (?, 48, 48, 16)
Conv2D Input: (?, 48, 48, 16) Output: (?, 48, 48, 16)
BatchNormalization Input: (?, 48, 48, 16) Output: (?, 48, 48, 16)
LeakyReLU Input: (?, 48, 48, 16) Output: (?, 48, 48, 16)
Conv2D Input: (?, 48, 48, 16) Output: (?, 48, 48, 16)
BatchNormalization Input: (?, 48, 48, 16) Output: (?, 48, 48, 16)
LeakyReLU Input: (?, 48, 48, 16) Output: (?, 48, 48, 16)
Conv2D Input: (?, 48, 48, 16) Output: (?, 48, 48, 16)
BatchNormalization Input: (?, 48, 48, 16) Output: (?, 48, 48, 16)
LeakyReLU Input: (?, 48, 48, 16) Output: (?, 48, 48, 16)
MaxPooling2D Input: (?, 48, 48, 16) Output: (?, 24, 24, 16)
Dropout Ratio decided by Hypertuner Input: (?, 24, 24, 16) Output: (?, 24, 24, 16)
Conv2D Input: (?, 24, 24, 16) Output: (?, 24, 24, 16)
BatchNormalization Input: (?, 24, 24, 16) Output: (?, 24, 24, 16)
LeakyReLU Input: (?, 24, 24, 16) Output: (?, 24, 24, 16)
Conv2D Input: (?, 24, 24, 16) Output: (?, 24, 24, 16)
BatchNormalization Input: (?, 24, 24, 16) Output: (?, 24, 24, 16)
LeakyReLU Input: (?, 24, 24, 16) Output: (?, 24, 24, 16)
MaxPooling2D Input: (?, 24, 24, 16) Output: (?, 12, 12, 16)
Conv2D Input: (?, 12, 12, 16) Output: (?, 12, 12, 16)
BatchNormalization Input: (?, 12, 12, 16) Output: (?, 12, 12, 16)
LeakyReLU Input: (?, 12, 12, 16) Output: (?, 12, 12, 16)
Conv2D Input: (?, 12, 12, 16) Output: (?, 12, 12, 16)
BatchNormalization Input: (?, 12, 12, 16) Output: (?, 12, 12, 16)
LeakyReLU Input: (?, 12, 12, 16) Output: (?, 12, 12, 16)
MaxPooling2D Input: (?, 12, 12, 16) Output: (?, 6, 6, 16)
Conv2D Input: (?, 6, 6, 16) Output: (?, 6, 6, 16)
BatchNormalization Input: (?, 6, 6, 16) Output: (?, 6, 6, 16)
LeakyReLU Input: (?, 6, 6, 16) Output: (?, 6, 6, 16)
Conv2D Input: (?, 6, 6, 16) Output: (?, 6, 6, 16)
BatchNormalization Input: (?, 6, 6, 16) Output: (?, 6, 6, 16)
LeakyReLU Input: (?, 6, 6, 16) Output: (?, 6, 6, 16)
MaxPooling2D Input: (?, 6, 6, 16) Output: (?, 6, 6, 16)
Dropout Ratio decided by Hypertuner Input: (?, 6, 6, 16) Output: (?, 6, 6, 16)
Flatten Input: (?, 6, 6, 16) Output: (?, 576)
Dense Input: (?, 576) Output: (?, 1024)
LeakyReLU Input: (?, 1024) Output: (?, 1024)
BatchNormalization Input: (?, 1024) Output: (?, 1024)
Dense Input: (?, 1024) Output: (?, 1024)
LeakyReLU Input: (?, 1024) Output: (?, 1024)
BatchNormalization Input: (?, 1024) Output: (?, 1024)
Dropout Ratio = 0.2 Input: (?, 1024) Output: (?, 1024)
Dense Input: (?, 1024) Output: (?, 1)
Sigmoid Activation Input: (?, 1) Output: (?, 1)
End Result: Is a cancer tumor malignant or benign?
Output

Figure 1. **The architecture of the custom CNN with the *PatchCamelyon* data. Each box corresponds to either a convolutional, normalization, pooling or leaky rectifier layer. The numbers in each box correspond to the shape of the downsampled image when passed to the next due to the pooling operations in the network. The '?' refers to the yet unknown batch size of the images when passed to the classifier. Since this CNN was used for the first experiment, the batch size of images passed in was 128.**

Prior to hypertuning, a constant term was applied to the beginning of the network in order to control for the imbalance in the *TCIA* dataset and partial imbalance in the *PatchCamelyon* data. This constant term is given by Equation 2, where $\mathbf{x_i}$ and $\mathbf{x_f}$ represent the training and testing pixel value matrices and $\mathbf{y_i}$ and $\mathbf{y_f}$ represent the class indices (metastatic or benign). This constant term is attached to the last fully connected layer in all neural networks in the paper.

$$b(\mathbf{x_i}, \mathbf{x_f}, \mathbf{y_i}, \mathbf{y_f}) = \ln(\frac{\|\mathbf{x_i}\|_0 + \|\mathbf{x_f}\|_0}{\|\mathbf{y_i}\|_0 + \|\mathbf{y_f}\|_0}) \tag{2}$$

Following the conversion of the images into pixel value matrices, the images were split into training and testing sets in order for the model to learn effectively. Each image set was fed through an image generator that augmented the data by flipping the normalized images vertically, horizontally and shifting their positions by a tenth of each of their width and height. The modified images were then combined with the original images to form a larger training and testing set. Finally, all training and testing images were normalized by finding the standard score for each pixel value matrix.

With the data augmented and model built, the cancer class was then detected using the custom CNN. The `kerastuner` library was used in order to hypertune the starting learning rate, the amount of ℓ_2 regularization and the number of filters in the CNN to their optimal values under 750 epochs. A learning rate step plateau scheduler was also created to automatically decrease the learning rate of the model over the training period. The starting learning rate (η_0) was set to an initial rate of 0.05 and was multiplied by a factor x according to Equation 3 every 400 epochs. Through previous research in the field, the factor was set to 0.01 for this project.

$$\eta(\eta_0, x) = \eta_0 x \tag{3}$$

Furthermore, a SGD optimizer was employed with a Nesterov momentum of 0.9 to train the data. On PaperSpace, the multi GPU model on Keras was used on the built model with 2 GPUs to employ data parallelism and create a parameter server for PDP. The model was compiled with the SGD algorithm above with categorical cross entropy loss and with accuracy and ROC-AUC score metrics.

The second experiment followed the same procedure with hypertuning and configuration of images as did the first experiment. The only difference was the use of a VGG16 pre-trained network with ImageNet weights to analyze the 130 images from the *TCIA* and *PatchCamelyon* datasets; only the last three convolutional layers being trainable to weights for added performance. A pre-trained network was chosen for this task due to their ability to accumulate features from previous datasets and recall them in new datasets; this feature helps to prevent custom-built neural networks from overfitting with small data. The network architecture for the VGG16 model is shown in Figure 2.

There were two main metrics reported from the neural networks for binary classification: accuracy and the ROC-AUC score. Since the CIFAR10 dataset was multiclass, only the accuracy was reported for the neural network benchmark. The model was fit on the augmented and original data with a mini-batch size of 128 for 200 epochs using the test data and the learning rate scheduler to validate the metrics and decrease the learning rate. Both experiments and the benchmark were monitored via TensorBoard, a real-time application that can graph the decrease in loss and increase in accuracy as the model trains over time [23]. After the loss and accuracy were graphed in Paperspace, the final graphs in via `matplotlib` and TensorBoard were obtained and saved to display in Figures 3-6. With the model processed, results could then be analyzed. The Python source code for this procedure can be found at this link: https://rb.gy/8wugs6.

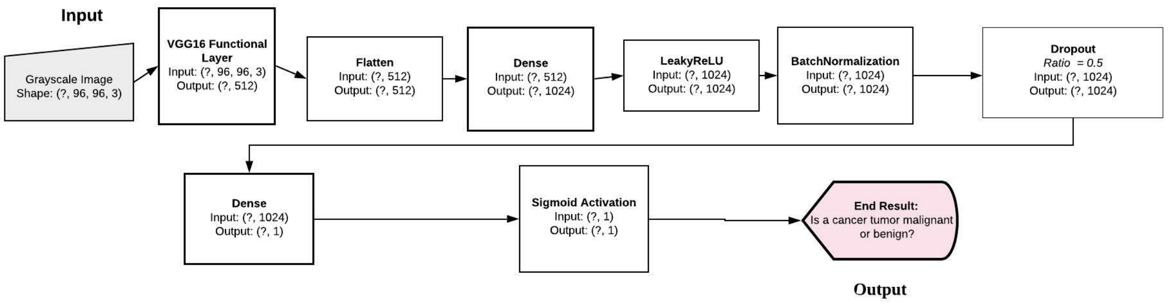


Figure 2. **The architecture of the VGG16 network used for the 130 samples of data found in *The Cancer Imaging Archive* and *PatchCamelyon* datasets. Each box corresponds to either a convolutional, normalization, pooling or leaky rectifier layer. The first box in the figure corresponds to the VGG16 pretrained base of the network. The numbers in each box correspond to the shape of the downsampled image when passed to the next due to the pooling operations in the network. The ? refers to the yet unknown batch size of the images when passed to the classifier. Since this CNN was used for the second experiment, the batch size of images passed in was 8.**

3. Results

3.1. Interpretation of Results

The demonstration and use of this custom CNN are shown in Figures 3-4 with Figure 3 indicating the benchmark performance of the custom CNN on the CIFAR10 dataset and its predicted effect on Figure 4 — the performance of the CNN on the *PatchCamelyon* dataset.

As depicted from the figures, the benchmark validation accuracy of the CIFAR10 dataset at 91.75% leads to the impressive results with the accuracy of the CNN on the *PatchCamelyon* dataset at 83% and the model's ability to distinguish between benign and malignant tumors (ROC-AUC score) at 88.5%. Likewise, the effect of the magnifications of the lymph node images in the VGG16 model are shown in Figures 5 and 6 where Figure 5 depicts the performance of the model with the *TCIA* slides at 20x magnification and Figure 6 depicts the performance of the model with the *PatchCamelyon* at 40x magnification. The best accuracy of the model in Figure 5 was 72% with a ROC-AUC score of 60% and the best accuracy of the model in Figure 6 was 80% with a ROC-AUC score of 85%. Thus, the VGG16 model predicts the correct tumor type when breast cancer slides are viewed at 40x magnification better than in 20x magnification. To test if this relationship is statistically significant, Figure 7 graphs the mean accuracy of both classifiers with their margin of error. Because the error bars do not overlap, the difference in slide magnification has a statistically significant effect on the accuracy of CNNs in tumor classification.

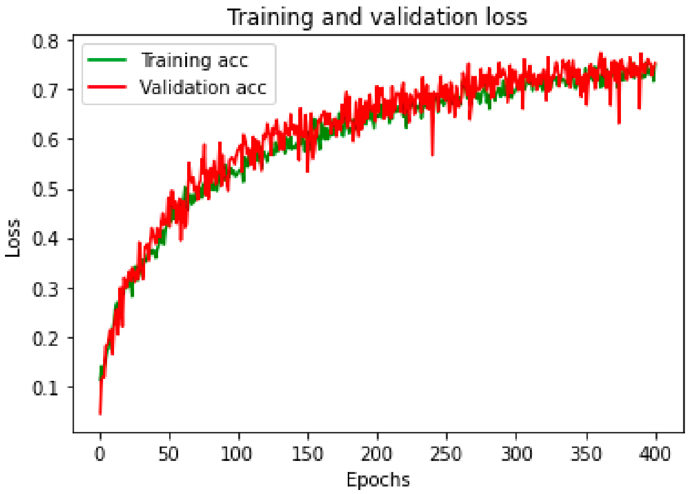


Figure 3. **The accuracy and validation accuracy of the custom CNN on the CIFAR10 data with the architecture shown in Figure 1. The CNN received the highest validation accuracy at 91.75% at 500 epochs after dropping the learning rate by a factor of 10 five times according to Equation 3 from the hypertuner selected learning rate of 0.05 to 0.00005.**

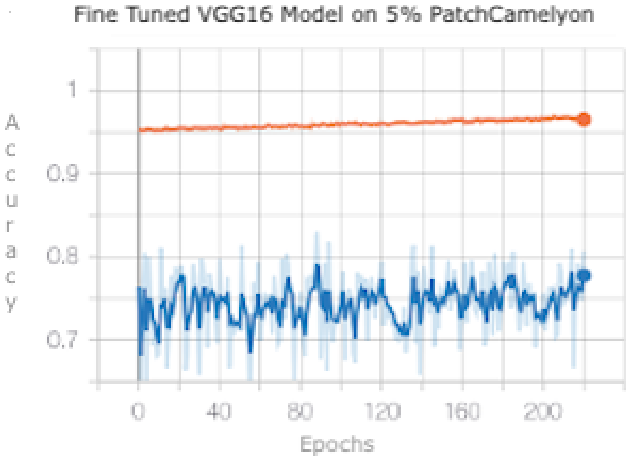


Figure 4. **The accuracy and testing accuracy of the custom CNN (taken on TensorBoard) on 5% of the PatchCamelyon data with the architecture shown in Figure 1. The red line of the graph signifies the training accuracy over the 500 epochs whereas the blue line signifies the testing accuracy over the epochs. The CNN received the highest validation accuracy at 83% at 500 epochs (past training efforts not shown) after dropping the learning rate by a factor of 10 five times according to Equation 3 from the hypertuner selected learning rate of 0.05 to 0.00005.**

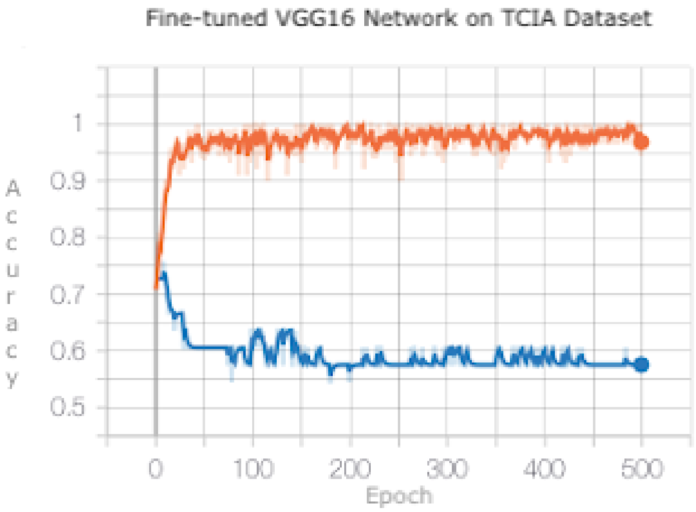


Figure 5. **The accuracy and testing accuracy of the pretrained and fine-tuned VGG16 network (taken on TensorBoard) on the TCIA dataset with the architecture shown in Figure 2. The red line of the graph signifies the training accuracy over the 500 epochs whereas the blue line signifies the testing accuracy over the epochs. The CNN received the highest validation accuracy at 72% at 500 epochs (past training efforts not shown) after dropping the learning rate by a factor of 10 five times according to Equation 3 from the hypertuner selected learning rate of 0.05 to 0.00005.**

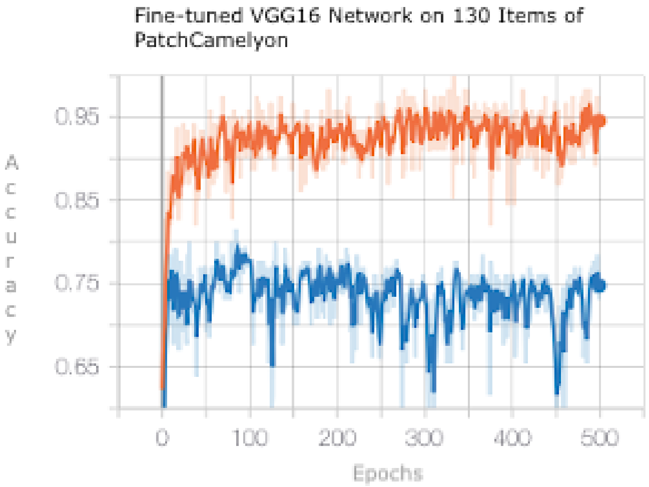
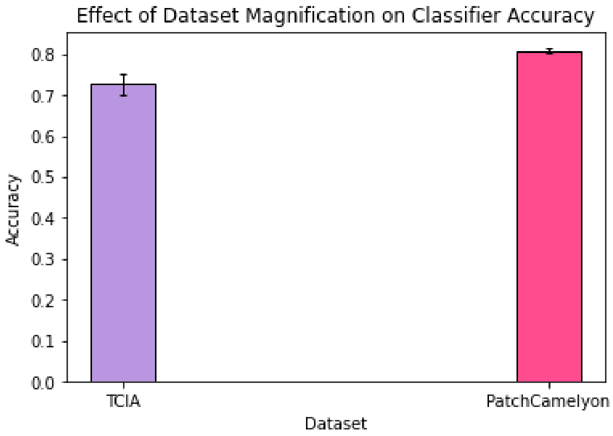


Figure 6. **The accuracy and testing accuracy of the pretrained and fine-tuned VGG16 network (taken on TensorBoard) on 130 items of the PatchCamelyon dataset with the architecture shown in Figure 2. The red line of the graph signifies the training accuracy over the 500 epochs whereas the blue line signifies the testing accuracy over the epochs. The CNN received the highest validation accuracy at 80% at 500 epochs (past training efforts not shown) after dropping the learning rate by a factor of 10 five times according to Equation 3 from the hypertuner selected learning rate of 0.05 to 0.00005.**



Note: TCIA refers to the dataset obtained from the Cancer Imaging Archive by Campanella et al.

Figure 7. **This bar graph indicates the statistically significant difference between the 20x magnification and the 40x magnification of the breast cancer lymph node slides found in *The Cancer Imaging Archive* and *PatchCamelyon* datasets respectively with 130 samples. The mean accuracy of the pretrained network on the *TCIA* dataset was 0.72715 ± 0.025. The mean accuracy of the pretrained network on the *PatchCamelyon* dataset was 0.80769 ± 0.006. Since the error bars in the bars do not intersect with each other, this shows that the difference between the detection accuracies of both slide inputs into the VGG16 pretrained network is statistically significant.**

4. Discussion

4.1. Discovery

Based on the results, a parallel distributed convolutional neural network was discovered that can be used with synchronous SGD to obtain greater than a 80% accuracy and ROC-AUC score, accepting the research hypothesis. Furthermore, it was shown that CNNs can classify cancerous lymph node data with 40x magnification statistically better than data at 20x magnification. These results extend into the biochemistry field because they indicate that this custom CNN is 10% better than pathologists at pre-diagnosing cancer (an abundance in abnormal mitosis) in early stages and roughly 10% better than pathologist classification in later stages of cancer [2]. Therefore, the model has promise to be adopted in hospitals for pre-diagnosis for immediate treatment of the cancer before the cancer stage progresses and chemotherapy treatments become ineffective. This is especially critical due to the decline in the five-year survival rate of patients undergoing chemotherapy as the stage of cancer progresses ranging from 93% at stage 0 cancer to 15% at stage 4 cancer [10]. Since pathologist errors only delay the ability to diagnose the cancer in lymph nodes, this custom CNN can be used by doctors to preliminarily diagnose cancerous tumors for chemotherapy treatment before the stage of the cancer progresses.

The study also indicates that the more detail present in the histopathological slide, the better biomedical CNNs are at analyzing the data and finding relevant patterns in future histology data. This confirms the thesis of CNNs and their applicability to detailed medical datasets [24]. Due to the benchmark performance of this custom CNN on the CIFAR10 dataset, the model also shows the future ability of it to extend to histology datasets of other cancers at 40x magnification. Since CNNs learn to differentiate between benign and malignant tumors by learning simple patterns like horizontal and vertical lines before increasing in complexity, the simple patterns that the custom CNN learned from differentiating breast cancer tumors can be easily applied to the shapes of tumors of other cancers and increase the speed of pre-diagnoses of these related cancers.

Finally, the PDP was shown to be a major improvement in speeding up the execution of the custom CNN model and can be used with cloud based GPU systems to multitask the classification of histopathological data for faster pre-diagnoses. In the increasingly computerized world of hospitals with new patients being added to the hospital database when they arrive, adding a GPU cloud-based detection algorithm would only be a step up to aid the large quantity of yearly individuals affected by 2$^{\text{nd}}$ most common cancer among women in the U.S [6].

4.2. Limitations and Future Work

Although there were many advantages of the work to advancing the biotechnical fields, the limitations to this research were mainly due to the limited amount of data for breast cancer on The Cancer Imaging Archive. The *TCIA* dataset used in this paper had only 130 items, which needed to be accounted for in Figure 5-7 since small datasets are harder to learn from in deep learning. In order to confirm the results of magnification on classifier accuracy, large medical datasets like those the size of *PatchCamelyon* should be compared to one another to see if the results found in this paper is representative of larger populations.

For future experiments, experimentation with programs like AutoAugment will be done to select the best data augmentation routines on images for the CNN to reduce overfitting [25]. More importantly, it would be interesting to study how the skip connections presented in ResNet could affect the performance of the model, since it improved upon the VGGNet models and decreased the training loss by 0.042 and the validation loss by 0.026 in select biomedical segmentation tasks [26]. Despite these limitations and next steps, the 10% neural network improvement at detecting cancerous tumors than pathologists at both earlier and later stages will have wonders on the biochemical fields if implemented and show that thousands of individuals can be pre-diagnosed at early stages of cancer to prevent breast cancer treatments from failing at later stages. This is truly a new computational framework to prevent the dangers of abnormal mitosis in the breast from reaching dangerous conditions.

Acknowledgments

Foremost, I would like to sincerely thank Dr. Shapovalov of The Bronx High School of Science for his numerous peer reviews of my drafts and Dr. Yuefan Deng of Stony Brook University for his guidance and support of this project.

Conflict of Interest

The author of this article declares that he has no conflict of interest.

Human Studies/Informed Consent

No human studies were carried out by the author for this article.

Animal Studies

No animal studies were carried out by the author for this article.

References

[1] Campanella, G., Hanna, M. G., Brogi, E., Fuchs, T. J. (2019). Breast Metastases to Axillary Lymph Nodes [Data set]. *The Cancer Imaging Archive*. doi: 10.7937/tcia.2019.3xbn2jcc

[2] Bejnordi B.E., Veta M., van Diest P.J., van Ginneken B., Karssemeijer N., Litjens G., ... & CAMELYON16 Consortium. (2017). Diagnostic Assessment of Deep Learning Algorithms for Detection of Lymph Node Metastases in Women With Breast Cancer. *JAMA*: 318(22), 2199-2210. doi: 10.1001/jama.2017.14585

[3] Veeling B.S., Linmans J., Winkens J., Cohen T., & Welling M. (2018). Rotation Equivariant CNNs for Digital Pathology. *ArXiv*, preprinted at https://arxiv.org/abs/1806.03962.

[4] *TensorFlow Datasets. A collection of ready-to-use datasets.* (2021). Retrieved November 8, 2021, from https://www.tensorflow.org/datasets.

[5] American Cancer Society. (2021). *Cancer Facts & Figures 2021.* Cancer.org. https://www.cancer.org/content/dam/cancer-org/research/cancer-facts-and-statistics/annual-cancer-facts-and-figures/2021/cancer-facts-and-figures-2021.pdf.

[6] *U.S Breast Cancer Statistics.* (2021). BreastCancer.org. Retrieved November 8, 2021, from https://www.breastcancer.org/symptoms/understand_bc/statistics.

[7] Tian, X., Deng, Z., Ying, W., Choi, K., Wu, D., Qin, B., Wang, J., Shen, H., and Wang, S. (2019). Deep multi-view feature learning for EEG-based epileptic seizure detection. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*: 1-1. doi: 10.1109/TNSRE.2019.2940485.

[8] Zinkevich, AM., Weimer, M., Smola, A. and Li, L. (2010). Parallelized Stochastic Gradient Descent. *NIPS'10 Proceedings of the 23rd International Conference on Neural Information Processing Systems*, 2: 2595-2603.

[9] Chen, J., Monga, R., Bengio, S., and Josefowicz, R. (2016). Revisiting Distributed Synchronous SGD. *ArXiv*, preprinted at https://arxiv.org/abs/1604.00981.

[10] Kirilov, K. and Borovska, P. (2019). Conceptual model of integrated approach for in silico knowledge data discovery for breast cancer diagnostics and precision therapy. *AIP Conference Proceedings*, 2172(1), 02003. doi: https://doi.org/10.1063/1.5133485.

[11] Mendoza, H., Klein, A., Feurer M., Springenberg JT., and Hutter F. (2016). Towards Automatically-Tuned Neural Networks. *JMLR: Workshop and Conference Proceedings*, 1: 1-8.

[12] Mantovani, RG., Horváth, T., Cerri, R., Vanschoren, J and de Carvalho, A. (2016). Hyper-parameter Tuning of a Decision Tree Induction Algorithm. *2016 5th Brazilian Conference on Intelligent Systems (BRACIS)*: 37-42, doi: 10.1109/BRACIS.2016.018.

[13] Ruder, S. (2016). An overview of gradient descent optimization algorithms. *ArXiv*, prepreinted at https://arxiv.org/abs/1411.4038.

[14] Al-Saffar, AAM., Tao, H and Talab, A. (2017). Review of deep convolution neural network in image classification. *2017 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, 26-31. doi: 10.1109/ICRAMET.2017.8253139.

[15] He, K., Zhang, X., Ren, S and Sun, J. (2015). Deep Residual Learning for Image Recognition. *ArXiv*, preprinted at https://arxiv.org/abs/1512.03385.

[16] Long, J., Shelhamer E. and Darrel T. (2014). Fully Convolutional Networks for Semantic Segmentation. *ArXiv*, preprinted at https://arxiv.org/abs/1411.4038.

[17] Niu, F., Recht, B., Re, C., and Wright, SJ. (2011). HOGWILD!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent. *ArXiv*, preprinted at https://arxiv.org/abs/1106.5730.

[18] Gupta, S., Zhang, W. and Wang, F. (2016). Model Accuracy and Runtime Tradeoff in Distributed Deep Learning: A Systematic Study. *ArXiv*, preprinted at `https://arxiv.org/abs/1509.04210`.

[19] Fallahi, A., Pooyan M., Ghanaati, H. and Oghabian M. (2011). Uterine Segmentation and Volume Measurement in Uterine Fibroid Patients' MRI Using Fuzzy C-Mean Algorithm and Morphological Operations. *Iranian Journal of Radiology*: 8(3), 150-156. doi: 10.5812/kmp.iranjradiol.17351065.3142.

[20] Kurata, Y., Nishio, M., Kido, A., Fujimoto, K., Yakami, M., Isoda, H., and Togashi, K. (2019). Automatic segmentation of the uterus on MRI using a convolutional neural network. *Comput Biol Med*, 114. doi: 10.1016/j.compbiomed.2019.103438.

[21] Gupta, V., and Bhavsar, A. (2017). Breast Cancer Histopathological Image Classification: Is Magnification Important? *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*: 769-776. doi: 10.1109/CVPRW.2017.107.

[22] Krizhevsky, A. (2009). Learning Multiple Layers of Features From Tiny Images.

[23] Wongsuphasawat, K., Smilkov, D., Wexler J. and Wilson, J. (2017). Visualizing Dataflow Graphs of Deep Learning Models in TensorFlow. *IEEE Transactions on Visualization and Computer Graphics*, 24(2): 1-12. doi: 10.1109/TVCG.2017.2744878.

[24] Krizhevsky, A., Sutskever I., and Hinton, G. (2017). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*: 60(6), 84-90, doi: 10.1145/3065386.

[25] Cubuk, ED., Zoph, B., Mane, D., Vasudevan, V., and Le, QV. (2019). AutoAugment: Learning Augmentation Policies from Data. *ArXiv*, preprinted at `https://arxiv.org/abs/1805.09501`.

[26] Drozdal, M., Vorontsov, E., Chartrand, G., Kadoury, S., and Pal, C. (2016). The Importance of Skip Connections in Biomedical Image Segmentation. *ArXiv*, preprinted at `https://arxiv.org/abs/1608.04117`.