Early Diagnosis of Invasive Ductal Carcinoma Using Deep Learning

Yawar Ashraf - 1006293672
Kamran Ramji - 1002934284
Raihan Faruque - 1000743644

Introduction

The goal of the project is to detect Invasive Ductal Carcinoma (IDC) at an early stage. IDC is the most common form of breast cancer and a Deep learning model will be used to complete this task. A breast Histopathology dataset from Kaggle will be used to train and fine tune the model to its optimal performance. This dataset contains Mammography scan patches that have been divided into 50 by 50 pixels in size and organized with respect to patient number and whether or not they exhibit signs of IDC. The model architecture and development pipeline can be visualized in Figure 1 below.

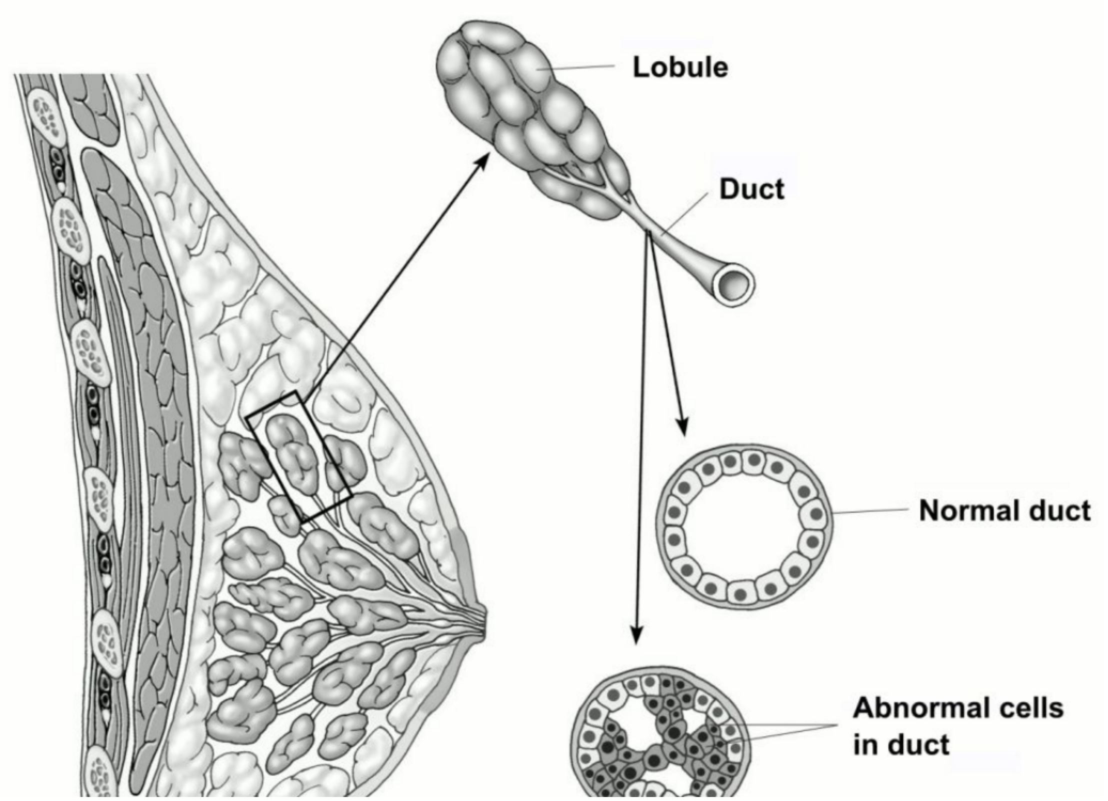
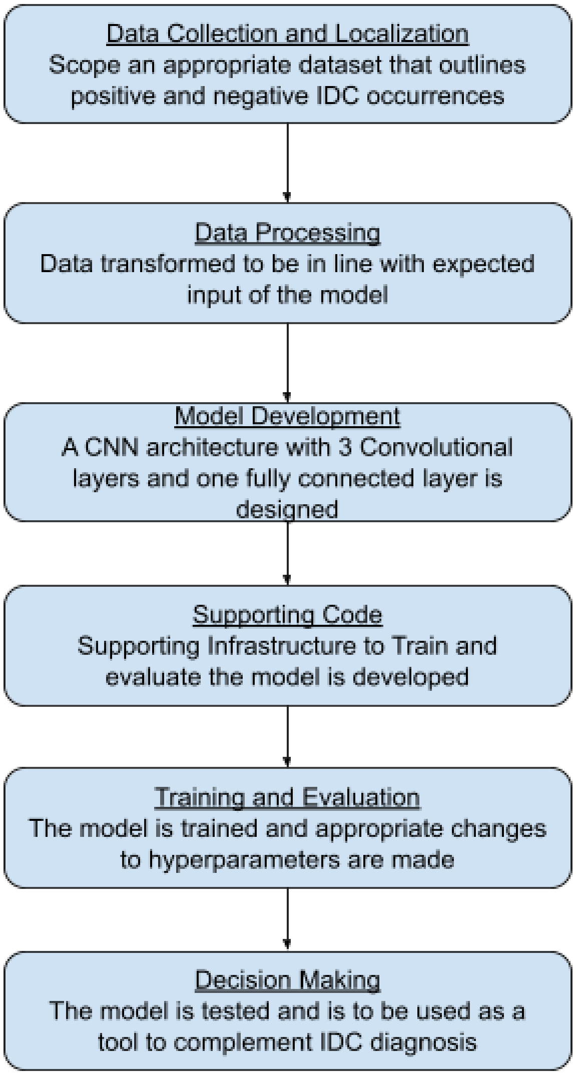


Figure 1: Invasive ductal Carcinoma in Situ [1]

Breast cancer is among the leading causes of cancer related death for women around the globe as it accounts for 13% of all cancer related deaths in women [2]. Mammography screening allows physicians to identify the disease at an early stage, however, this methodology is prone to high risk of False Positive and False Negatives. Therefore, using Artificial Intelligence to detect for this problem is a viable option. Machine Learning is suitable for this task since the machine learning models have been shown to outperform radiologists in early stage detection of different types of cancer [3].

Illustration



Data Collection and Localization
Scope an appropriate dataset that outlines positive and negative IDC occurrences

Data Processing
Data transformed to be in line with expected input of the model

Model Development
A CNN architecture with 3 Convolutional layers and one fully connected layer is designed

Supporting Code
Supporting Infrastructure to Train and evaluate the model is developed

Training and Evaluation
The model is trained and appropriate changes to hyperparameters are made

Decision Making
The model is tested and is to be used as a tool to complement IDC diagnosis

An example data image that is passed into the model

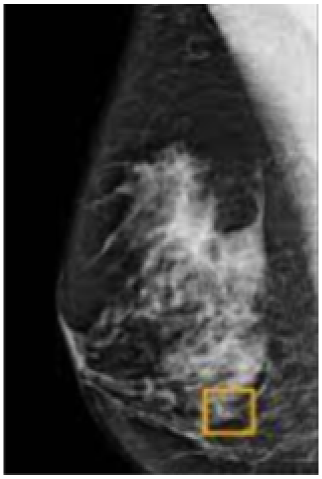


Figure 2: Project Development Pipeline

Background

The use of Artificial Intelligence (AI) is very promising in the domain of medical imaging and pattern detection. With the growing ease of collecting data it is becoming more feasible to train deep learning applications that require huge amounts of data to provide tangible results. This is reflected in the exponential growth of research done in recent years to employ this data for the healthcare industry as shown in Figure 1 [4]:

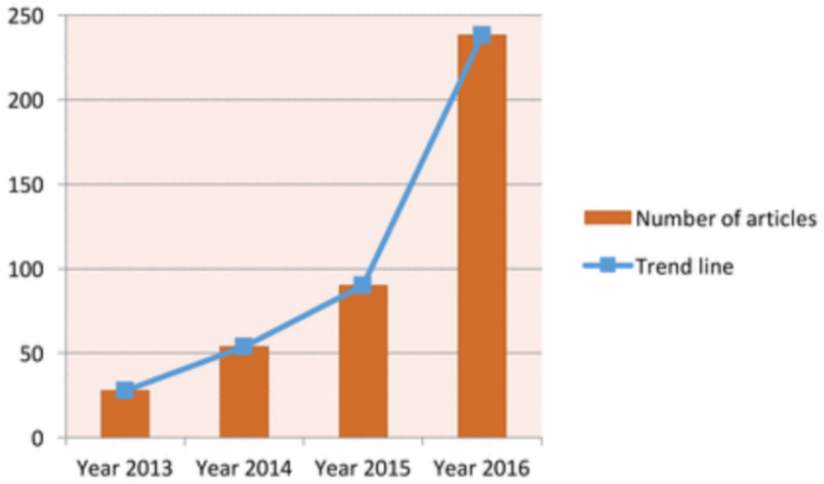


Figure 3: Current trend for deep learning. The data is generated by searching the deep learning in healthcare and disease category on PubMed.

The medical community is experiencing a transformative phase where AI has the potential to significantly impact the responsibilities of medical professionals which include both aiding in diagnostic and treatment tasks as well as planning out the treatment for the patient. AI has already begun its integration into the healthcare system, a recent Machine Learning tool developed at the University of Virginia School of Medicine can rapidly analyze children's biopsies in the early stages of disease onset and tell apart celiac disease and environmental enteropathy with a higher accuracy and reliability than doctors when they are more often confused. However, with the future integration of AI into the healthcare system, it is integral for health professionals and governments to oversee its development and performance in real world settings to guide its progress in the optimal direction [5].

Data Processing

Initially the data was sourced kaggle.com and the Breast histopathology Images dataset provided by Paul Mooney was used [6] . The dataset is divided with respect to mammography images of each patient and then each patient file is subdivided into a *0* and *1* directory. The naming convention used in this dataset was *uxXyYclassC.png* where *u* describes the patient ID, *X* is the x-coordinate of the original image from which the 50 by 50 pixel patch was extracted and *Y* is the y-coordinate of the original image from which the 50 by 50 pixel image was sourced.

For the data processing purposes of this project the data was organized in the manner as is illustrated in Figure 5. Instead of each patient subdirectory having divisions of 0 and 1, all zero and one files were distributed across the training, validation and testing dataset with a ratio of 70 to 15 to 15 respectively. There were a total of 397,477 and 157,560 images in the 0 and 1 class respectively; therefore, the ratio of division was applied to each class and not across the whole dataset in order to avoid randomized skewness of the training data.

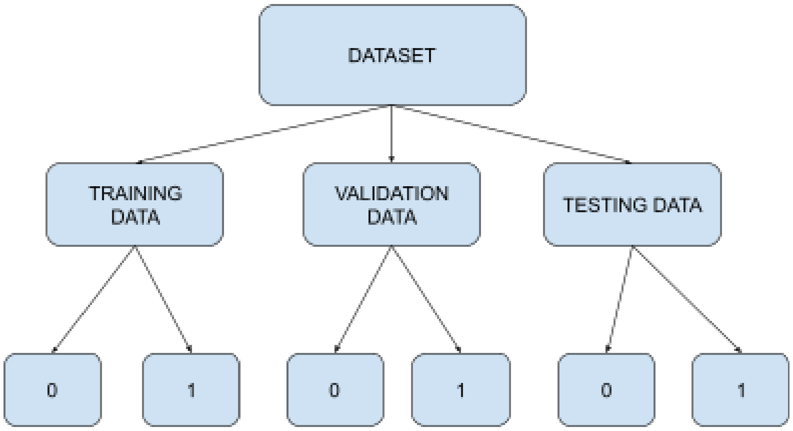


Figure 3: Preprocessed data’s directory architecture

In order to achieve this result, the *os* python library was used to walk through the complete dataset directory and rename the files to a new path as well. Only the class and patient number of the image were preserved, the class had to be preserved since it would be integral to train the model and the patient number was preserved to keep the possibility of future development and/or testing with respect to a single patient a viable option. Furthermore, a count was added to each file to make sure that all the class 0 files or the class 1 files of each patient do not collapse to a single file due to the naming convention. As we can see below in Figure 6, the files were moved to a new master dataset directory, this particular file was situated in the training subdirectory of the dataset and further found in the 1 (positive for IDC) directory of the training data. *12947* is the preserved patient number of the image and 1 in the file name also outlines the class of the image, and lastly the number *407575* is the file counter number used to prevent similarly named files to collapse to one entity. The dataset was initially preprocessed to set out 15 percent of the data for testing purposes which is unseen by the model. This data will be used once the team is satisfied with the hyperparameter tuning model and its evaluation will be provided in the final project submission.

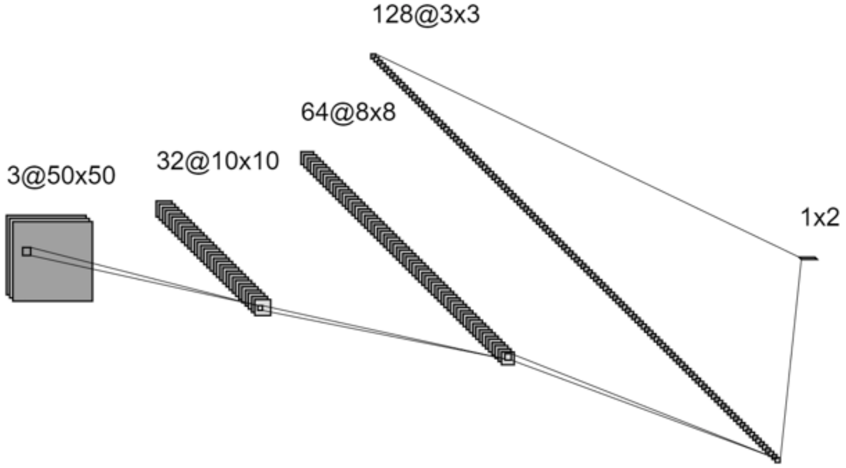
/breast-histopathology-imagesNEW/train/1/12947.1.407575.png

Figure 4: Directory and naming convention of a processed image [6]

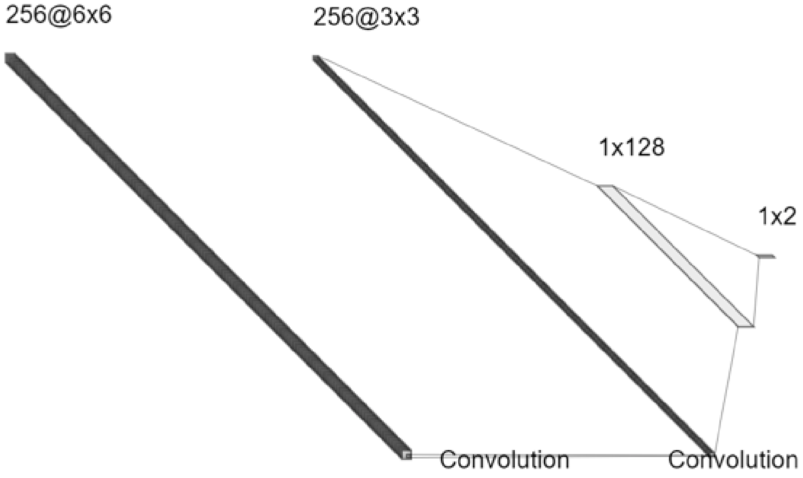
Architecture

We trained two different architectures, one of which was an end-to-end CNN that took raw 50 by 50 RGB images as input, and the second which used pretrained AlexNet features. The first network had three convolutional layers, with 32, 64 and 128 feature maps in each layer, as well as a final fully connected layer. The purpose of the increasing number of feature maps is to allow the model to learn increasingly complex representations deeper in the network. The network uses larger strides instead of max pooling layers to condense information, as we found that this worked better in practice. The pretrained architecture had two convolutional layers, as well as a single fully connected layer. The input to this architecture are the features extracted by a pretrained AlexNet (hence the “pretrained” appellation for this architecture). The input images are 50 by 50, but AlexNet expects images to be 224 by 224, and so a bilinear interpolation transformation was applied to the images before they were passed into AlexNet. The AlexNet weights were frozen during training, only the “custom” layers of the network (i.e. the two convolution, and single fully-connected layers) were updated during training.

End-to-end CNN Architecture Diagram:



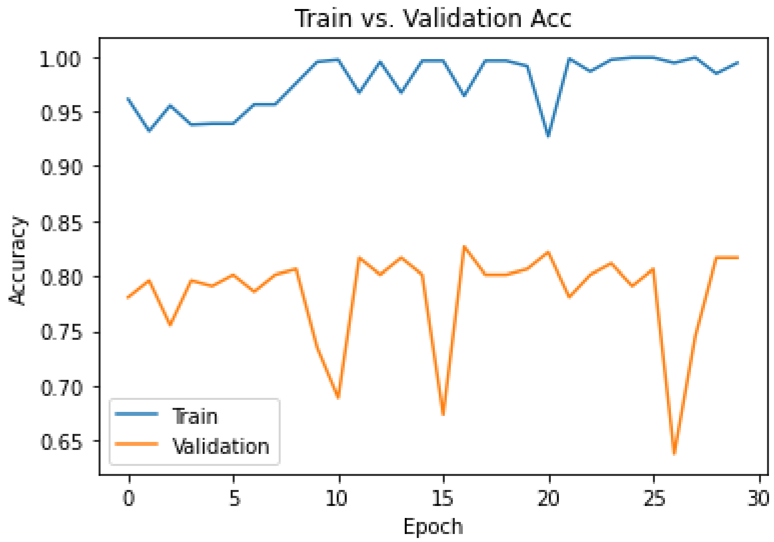
Pretrained AlexNet Architecture Diagram:



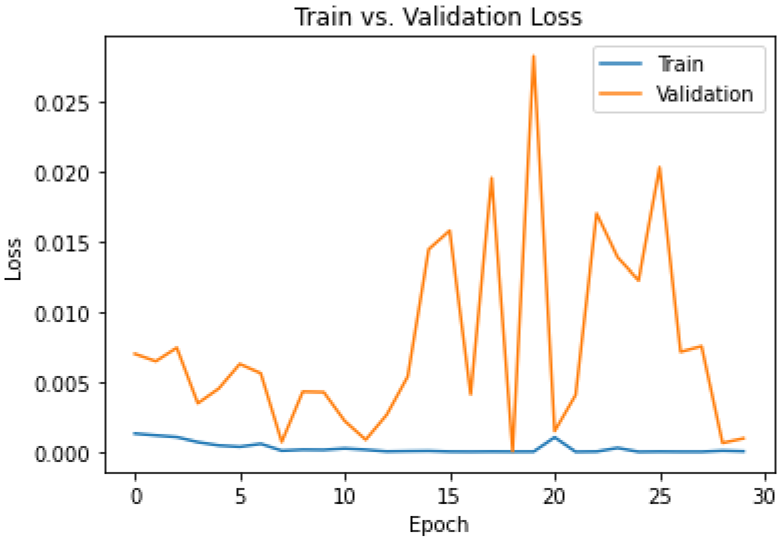
Baseline Model

Our baseline model is a simple neural net with only a single hidden linear layer of 2500 neurons. This choice of baseline model is reasonable because this architecture is capable of performing many image classification tasks such as handwritten digit recognition. The training curves provided for the baseline model demonstrate that while the neural net does achieve a reasonable level of accuracy on the validation set, it does not perform as well as the non-baseline architectures.

Baseline Training vs. Validation Accuracy:



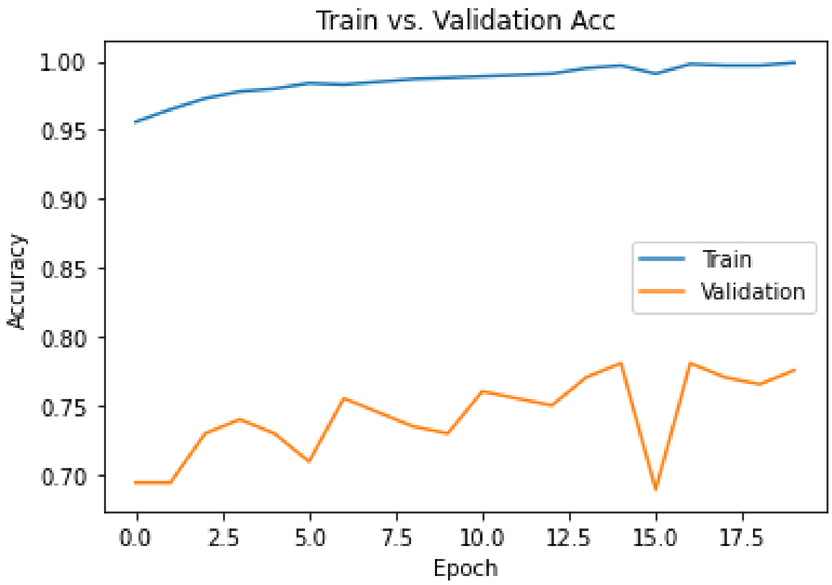
Baseline Training vs. Validation Loss:



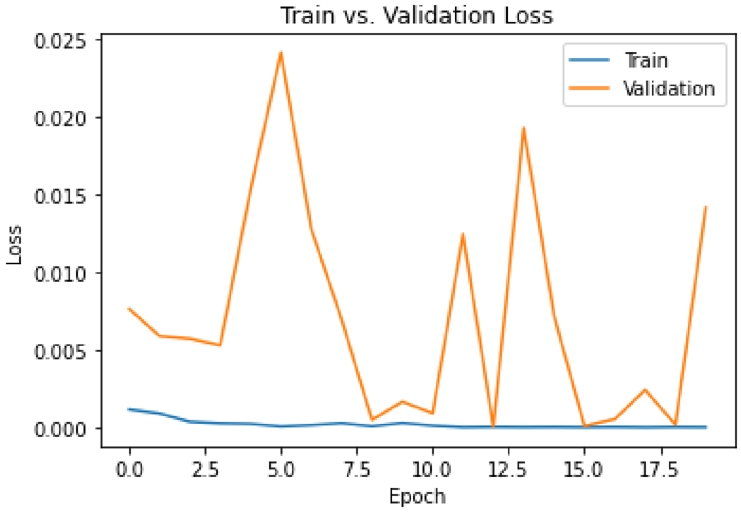
Quantitative Results

The training curve for the pretrained model demonstrates the inability of this model to generalize to new data. While the training accuracy increases to above 95%, the validation accuracy remains low, not exceeding 80%. This discrepancy is likely due to the fact that the AlexNet features are trained on ImageNet, and therefore are not specialized to detecting the specific kinds of features that are present in our dataset, related specifically to IDC. The end-to-end CNN architecture generalized much better, with a slight divergence between the validation and training accuracy after the first two epochs. Overall, our peak validation accuracy was 86.2%.

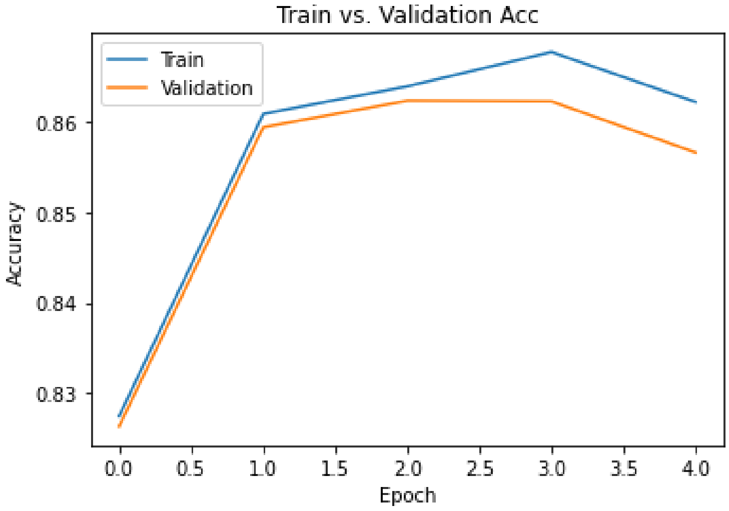
Training vs. Validation Accuracy for Pre-Trained Architecture:



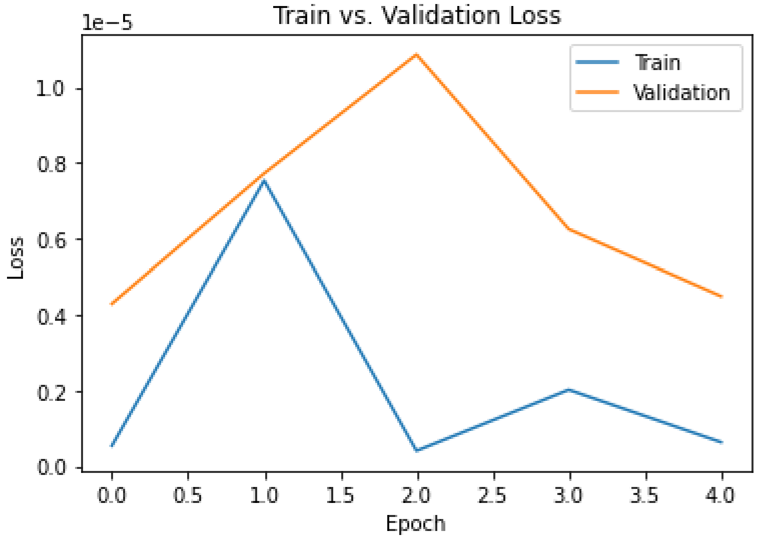
Training vs. Validation Loss for Pre-Trained Architecture:



Training vs. Validation Accuracy for End-to-End CNN Architecture:

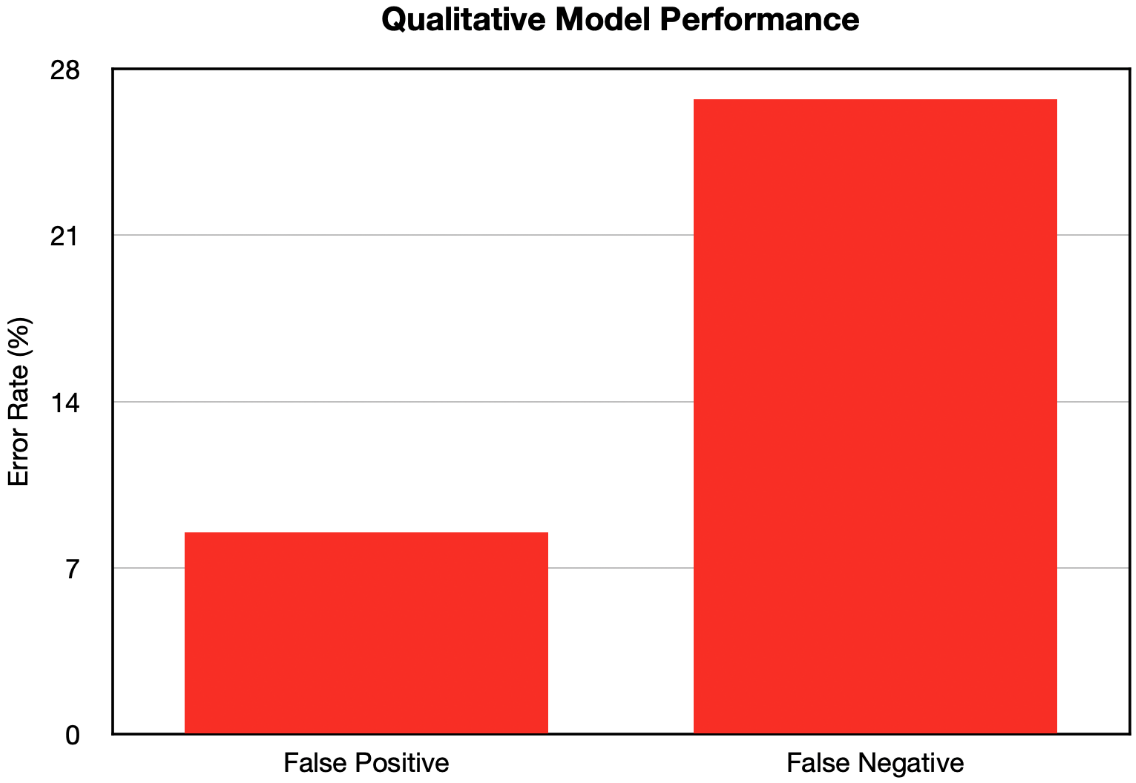


Training vs. Validation Loss for End-to-End CNN Architecture:



Qualitative Results

The overall test accuracy of our model is 86.35%. However, if this is broken down into false positives, and false negatives, there is a false positive rate of 8.5% and a false negative rate of 26.7%. The false negative rate is an especially critical statistic, since a false negative in this context is fatal for the patient. Furthermore, since the vast majority of patients (in the real world, outside of our relatively balanced training set) do not have cancer, a high overall accuracy is meaningless. It is more important to actually detect cases of cancer.



Model Evaluation on Unseen Data

Our model had a test accuracy of 86.35%, with a false positive rate of 8.5% and a false negative rate of 26.7%. The test data was held out at the data processing stage, and was not used until the very end of the project. The two different model architectures, as well as the baseline model, were tuned, compared and selected based on their validation accuracy. The test data was not used to adjust any hyperparameters or modify the architecture in any way. Therefore, this is an accurate measure of the performance of the model.

Discussion

To evaluate the model, we can not simply use the validation accuracy, we have to look at the false negative rate as well. This is because even if we have a high validation accuracy, having a high false negative will mean someone with breast cancer will not seek treatment if the model predicts a negative result. To offset this, our model needs to have a very high threshold to predict a negative result so that if there is any doubt about the result, the tool should predict a positive result, so a medical professional can take a look to confirm the presence of cancer. Our model has a false negative rate of 26%, so this tool is best suited to be used as a tool to help doctors speed up diagnosis. Our validation accuracy is 86.2% which is less than the state of art accuracy of 98% [7].

Furthermore, a major hurdle in the development and fine tuning of the model was the lack of computational power readily available. Additionally, the inherent lack of time due to the team academic commitments elsewhere hindered the teams' ability to train the model for multiple hours. Since the project was developed and trained on local machines, the lack of computational power limited the number of iterations that could reasonably be done to fine tune all hyper parameters. Given more computational resources and time, a more aggressive approach to explore hyperparameter tuning would have resulted in an architecture better suited to tackle the problem at hand and generate more reliable and better results.

Ethical Considerations

The analysis of data can be particularly challenging due to issues with slide preparation, variations in scanning and staining across sites and vendor platforms, as well as the variance in the intensity of the disease itself due to biological variations [8]. Keeping these factors in mind, a model-building paradigm can be implemented to tackle these issues by modifying the patch selection technique that aims to identify a suitable training set containing information-rich exemplars to build a more robust model in an attempt to reduce false negatives and false positives [8]. The training data used is exceptionally large with 70% of the 277,000 image patches used for training which mitigates the disadvantage of having a few noisy data points.

The model is known to have a high negative positivity rate which can prove fatal to a patient if they are diagnosed incorrectly and do not take early measures to tackle the disease. Therefore, this tool is only meant to be used for a doctor's diagnosis of IDC and not the primary medium of assessment. Moreover, a threshold can be set to better filter out uncertain results generated by the model and consequently lower both the false positive and false negative rates.

References

1. https://www.cancer.org/cancer/breast-cancer/understanding-a-breast-cancer-diagnosis/types-of-breast-cancer/invasive-breast-cancer.html
2. https://www.cancer.ca/en/cancer-information/cancer-type/breast/statistics/?region=on
3. https://www.nature.com/articles/d41586-020-00847-2#:~:text=In%20people%20with%20only%20one,as%20well%20as%20the%20computer.
4. https://svn.bmj.com/content/2/4/230
5. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6199205/
6. https://www.kaggle.com/paultimothymooney/breast-histopathology-images
7. Shen, L., Margolies, L.R., Rothstein, J.H. *et al.* Deep Learning to Improve Breast Cancer Detection on Screening Mammography. *Sci Rep* 9, 12495 (2019).
8. https://www.jpathinformatics.org/article.asp?issn=2153-3539;year=2016;volume=7;issue=1;spage=29;epage=29;aulast=Janowczyk