Automated detection of IDC in breast cancer whole slide image patches using Convoluted Neural Network

CKME 136 Capstone Project

Iwin Nayak
07-27-2020

Ryerson University

Automated detection of IDC in breast cancer whole slide image patches using Convoluted Neural Network

Introduction

Invasive ductal carcinoma (IDC) is one of the most common types of breast cancer (~55% incidence rate). The metastasis rate is extremely high with IDC cases. After biopsy, the cells have to be processed further by the histopathologist to determine the presence or absence of abnormality and then, determine if it is IDC or metastatic along with the grading the tumors. This is usually done by trained personnel for the multiple cross sections of the biopsy, which is extremely time consuming. The final result is also heavily dependent on the expertise of the histopathologist.

Deep learning can be of great help to automatically detect cancerous cells, classify them as metastatic or IDC and improve the speed of detection. This could further be extended to make a web-based platform where the original histology images could be uploaded and the results could be available instantly. This could be of great use in situations where a wet lab to perform histopathological analysis is limited in terms of trained personnel.

Computer Aided Diagnosis have been used by clinicians to support clinical findings as a second opinion confirmation since the 90s. In case of breast cancer detection, the observations made by histopathologists can sometimes be in conflict and also subjective. CAD systems help to remove doubts in cases of non-concordance between specialists.

The next sections will give an overview of the Literature review, Dataset and the Approach I will be using for the Capstone project.

Literature Review

This Literature review of past work has been divided into pre-CNN (1 to 5) and post-CNN (6 to 10) scenarios of research progress with varying accuracies based on the methods and the size of the datasets used.

1. Large-scale computations on histology images reveal grade-differentiating parameters for breast cancer (Petushi, Garcia, Haber, Katsinis, & Tozeren, 2006)

Aim of the paper was to develop an automated computational method to classify H&E stained slides
Method Used included MATLAB Image Processing Toolbox with grayscale conversion, segmentation with adaptive thresholding and morphological operations, blob labelling and classification and feature extraction and nucleic classification using supervised learning
For the Classification of images, they also used LNKnet software that integrates neural network, statistical, ML classification, clustering and feature selection algorithms.
They divided the image dataset of 1062 feature vectors into 536 feature vectors for training and 526 for testing. From the LNKnet software, Linear Discriminant Analysis and Forward/Backward Search methods were used for feature selection. Quadratic classifier was selected out of the Gaussian Linear Discriminant, Gaussian Quadratic, K-Nearest Neighbor, Binary Decision Tree, Parzen Window Histogram, Naive Bayes

and Support Vector Machine classifiers in LNKnet as it gave the least minimum mean classification errors when evaluated across the testing set.

2. Automated gland and nuclei segmentation for grading of prostate and breast cancer histopathology (Naik et al., 2008)

Aim of the paper was to present an automated nuclei and gland segmentation scheme for prostate and breast histopathology in conjunction with manual segmentation. Their dataset consisted of 54 images for breast cancer and 44 images for prostate cancers.
The method used was a Bayesian classifier to generate likelihood scenes of structures of interest in the image based on image intensity and textural information. Centroids were used to extract information from the images with graph algorithms including Voronoi, Delauney triangulation and minimum spanning tree. The algorithms were evaluated by comparing their accuracies.
For Prostate Cancer grading, the feature sets were reduced using a non-linear dimensionality method called graph embedding which were then classified using SVM.
For Breast Cancer grading, the feature sets were reduced using principal component analysis (PCA) and further classification was done using SVM.
The accuracies obtained compared favorably with the corresponding manual segmentation results.

3. Computer-aided diagnosis of breast cancer based on fine needle biopsy microscopic images (Kowal, Filipczuk, Obuchowicz, Korbicz, & Monczak, 2013)
Aim of the study was to test and compare four clustering algorithms, K-means, fuzzy C-means, competitive learning neural networks and Gaussian mixture models for the color space and adaptive thresholding for the gray-space in five hundred real case medical images of breast cancer from fifty patients.
The dataset contained five hundred images from fifty patients, of which twenty-five were benign and twenty-five were malignant. To improve the accuracy of the predictions, the problem of inadequate segmentation of nuclei, was dealt with a two-step approach.
Firstly, the background to foreground segmentation is carried out by taking a grayscale image as input and outputting a binary image that represents the segment, also known as adaptive thresholding. For each pixel a threshold is calculated, if it is below the threshold, it is the background and if it is above the threshold, it is the foreground. Secondly, four different clustering algorithms, were applied to identify the nuclei and the accuracy were compared.
Mean and Variances for twenty-one standardized features from the images were calculated and then classified using k-nearest neighbors, naïve Bayes with kernel density estimate and decision trees. The classification was performed with 50-fold cross validation, making sure that the images from the same patient were never in the same training and testing set at the same time.
From the accuracy results it was determined that out of the four clustering methods K-means was the best choice (with respect to the computational time and number of parameters required). In terms of classifier performance, it was observed that k-NN was more accurate than Naïve Bayes and Decision tree.

4. Remote Computer-Aided Breast Cancer Detection and Diagnosis System Based on Cytological Images (George, Zayed, Roushdy, & Elbagoury, 2014)
The aim of this study was to develop an intelligent remote detection and diagnosis system for breast cancer based on cytological images.
The dataset includes ninety-two fine needle aspirated cell specimens out of which forty-five were of benign tumors and forty-seven were of malignant tumors
First, the cell nuclei were imagined to be circles instead of ellipses (to save computational time) and the circles were located with a feature extraction technique called as Hough Transform, in which the circles(nuclei) are detected with an algorithm inside a single pixel cell, known as accumulator space. The false-positives (including imperfect circles and other cells or debris) were eliminated using Otsu's thresholding method, which converts an image to a binary image and calculates the variation between the

black and white pixels and an unsupervised clustering technique called, fuzzy c-means clustering, that allows one datum to belong to two or more clusters, through the iterative optimization steps the procedure stops at a termination criterion between 0 and 1 and converges to a local minimum.
The segmentation of the nuclei boundaries was accomplished with the application of the marker-controlled watershed transform. In which, the image is treated like a topographical map and the brightness of each point is the height and it finds the lines that run along those ridges. When over segmentation occurs, the images are pre-processed or similar images are merged.
Twelve features are presented to four supervised algorithms (SVM, learning vector quantization (LVQ), probabilistic neural network and back-propagation algorithm) with 10-fold cross validation. The performance of the networks was compared based on resulted error rate, correct rate, sensitivity, and specificity. Results showed that PNN and SVM were better at prediction compared to LVQ and multilayer perceptron.

5. Breast Cancer Diagnosis From Biopsy Images Using Generic Features and SVMs (Brook et al., 2007)
The aim of the research was to propose a fully automated classification of healthy, benign tumor and malignant cancer cells using SVM on PCA reduced features, where the color component in the images was projected on the principal axis. As SVM is only applicable in binary classification, they first decomposed the multi-class problem into many binary problems and then combining the results. The Dataset used consisted of 361 samples (119: normal tissue, 102: benign carcinoma, 140: invasive ductal carcinoma) and the accuracy was 93.4% that could be increased to 96.4% if 20 percent of the images were rejected. To further increase their accuracy nearly half of the images will have to be rejected, which would be impractical.

6. Research frontier: Deep machine learning–a new frontier in artificial intelligence research (Arel, Rose, & Karnowski, 2010)
The article shows how the use of deep learning has affected the directions of research, mainly the use of Convoluted Neural Networks and Deep Belief Networks. CNNs are used mainly on two-dimensional data, like images and videos. They work primarily by reducing the learning parameters by leveraging spatial relationships and improving upon the gradient in weight space with respect to the loss function. In CNNs small parts of the images are used as inputs to the lowest layer of the hierarchy and the information moves through layers in the network that have filters. This is done to get main features of the data and provides a variance to the different aspects of the data such as shift, scale and rotation. This variance in data is used by the processing units to fundamental features such as if a part of the data has differently oriented edges or corners. After providing weights to the features if the weighting is small the image is blurry. Different weights cause the output to work similar to the AND or OR function. The outputs thus obtained are then further passed through another filter and the process is repeated up to a random number of times, known as epochs. The outputs are sometimes combined in feature pools, this pooling is not trained or learned by the network but predetermined by the executer. At the end of the process, the results of previous steps are fed to a feed forward neural network, in which there is no feedback from the outputs of the previous steps towards the inputs throughout the network.

7. ImageNet Classification with Deep Convolutional Neural Networks (Krizhevsky, Sutskever, & Hinton, 2017)
In this paper the authors have described how they used an eight layered deep convolutional neural network to classify a million images into thousand different classes. Their neural network has sixty million parameter and six hundred and fifty thousand neurons and consisted of five convolutional networks. To reduce the computational efforts, they used non-saturating neurons and GPU implementation and reduced overfitting by using the dropout regularization method, that artificially enlarges the dataset. The network was trained on the raw RGB values that were centered in the pixels. They obtained the top-1 error rate of 37.5% and top-5 error rate of 17.0%, that is the image to which a label is applied is not among the one or five most probable labels. The most efficient version of applying predictions of many different models to reduce the

test errors and avoid overfitting, was by using a method called as *dropout*, in which, the output of every hidden neuron with an equally like probability is set to zero. These neurons are "dropped out" and do not contribute to the final output.

8. Breast cancer histopathological image classification using Convolutional Neural Networks (Spanhol, Oliveira, Petitjean, & Heutte, 2016)

This was among the first researches done to classify breast cancer images using deep learning methods, particularly, the Convoluted Neural Networks. The research also used a large dataset of 7909 images divided into benign and malignant tumors. The researchers used a modified form of AlexNet (Krizhevsky et al., 2017) deep neural network architecture in the research.

Supervised training was done with extracting random patches of images and using the combination of the patches for recognition. To reduce over-fitting, they reduced the dimensionality of the original images and then extracted the patches by both overlapping the images by 50% and using patches that did not overlap. Classification was done by extracting the set of all non-overlapping patches, this was done to balance the classification performance and computational cost. After running the model, the results were combined by using the sum, product and max rule, which maximizes the sum of probabilities on all patches of the images. The salient advantage of using the deep learning technique, is that features do not have to be selected before-hand and lets the model learn them along the layers of computations.

The research was not accurate for high magnifications and only the edges of the nuclei were extracted.

9. Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks (Cruz-Roa et al., 2014)

The aim of the research was to train a CNN on whole-slide patches of breast cancer histology slides to detect invasive ductal carcinoma regions. The dataset consisted of 162 slides which was randomly split into training, validation and evaluation. Training and validation consisted of 82883 and 31352 patches respectively and testing consisted of 50963 patches. A Random Forest Classifier was used where the best attribute is chosen from individual decision trees that are made up of randomly selected attributes. The results obtained in this research show that CNN are more accurate at predicting IDC than the previously used methods of fuzzy color histogram (FCH) and RGB Histogram (RGBH). CNN yielded a F-measure and Balanced accuracy of 71.80% and 84.23% respectively, compared to FCH (67.53%, 78.74%) and RGBH (66.64%, 77.24%). They suggest that future work should be done with a larger cohort and including more layers and neurons in CNN architecture.

10. Classification of breast cancer histology images using Convolutional Neural Networks (Araújo et al., 2017)

The aim of the research was to use CNN to classify whole slide images into four classes viz., normal tissue, benign tumors, in situ carcinoma and malignant carcinoma and also in further overall classes of either cancerous or non-cancerous. The dataset consists of 269 images, out of which, 249 images were used for training and 20 images were used for testing. The images were selected based on the criteria that the four classes would be balanced to avoid overfitting and also so that the pathology classification could be clearly determined from the images. The classification was based on the nuclei features as well as tissue structure. The images were classified using either majority voting where the most common patch label decides the image label, or maximum probability of the class label, or sum of probability of the class labels, determined by adding the probabilities and assigning the class with the highest value. Both CNN and CNN+SVM were used to classify the image patches. It was observed that the proposed CNN architecture had better accuracy in a patch wise classification, 66.7%, compared to a CNN+SVM based method of 65.0%. The proposed CNN architecture also has better sensitivity, 80.6% compared to a CNN+SVM classifier, 74.5%, in detecting carcinomas from the patches.

To begin the classification of images it is very important to identify and segment histological structures. The separation of nucleus has been done by fuzzy c-means clustering and adaptive thresholding (Petushi

et al., 2006) but this often causes improper detection when the slides are over stained. Some of the techniques (Naik et al., 2008) involve using complicated pre-processing steps to create handcrafted and complex features to characterize the relatively small dataset of histopathology images. The final accuracy of the model is directly dependent on the accuracy of these pre-processing steps.

(Kowal et al., 2013) used fine needle aspirated subsets of cells to perform their research, however this method, does not identify and segment cells that get covered by other cells while staining. Their method also has low accuracy for high resolution images. To overcome this drawback, (George et al., 2014), used further transformation methods to better detect the nuclei in the images. But these improved methods also do not deal with real life scenarios of entire tissue section, instead relying on detection of individual cells.

When computing power and the sizes of databases increased, CNNs could be effectively used to classify images. Instead of devising ways to correctly extract features, CNNs learn the features that are useful, directly from the training images or image patches by optimizing the classification loss function or the cost of incorrectly predicting the class.

Krizhevsky, Sutskever, & Hinton developed a type of convoluted neural network in 2012, for the detection of images in the ImageNet dataset consisting of 15 million images categorized into 22000 categories. They used the computational power of graphical processing units and obtained competition winning error rates and described the methods in their 2017 paper (Krizhevsky et al., 2017) on the topic. This CNN architecture was further modified (Spanhol et al., 2016) to be used in detection of carcinoma in an extensive dataset of 7909 images obtained from 82 patients. This was the first time where a large dataset was used for automated detection of breast cancer and was made possible with the advances of CNN. The shortcomings in this research, was that the modified CNN architecture could not accurately detect identifying features in high magnifications and the patches have to compressed to improve accuracy.

Further progress was made with innovative techniques such as sliding the model through an image to obtain the probabilities and then thresholding was applied to obtain the final results (Cruz-Roa et al., 2014). This led to F1-scores of 0.780, that were better than even the cutting-edge techniques available at that point in time.

Using the knowledge acquired through the work done by past research, it can be noted that it is important to use a large dataset. I will be using a dataset that consists of 277,524 images and using a CNN architecture that consists of a 3X3 convolutional filter. The labels I am interested in, is to detect if a particular image is either IDC positive or negative and will be training the model to accurately identify these labels in the image patches that have been divided into benign and malignant. The Dataset and Approach I will be using are elaborated in further sections.

Dataset

The original dataset consisted of 162 whole mount slide images of Breast Cancer (BCa) specimens scanned at 40x. From that, 277,524 patches of size 50 x 50 were extracted (198,738 IDC negative and 78,786 IDC positive).

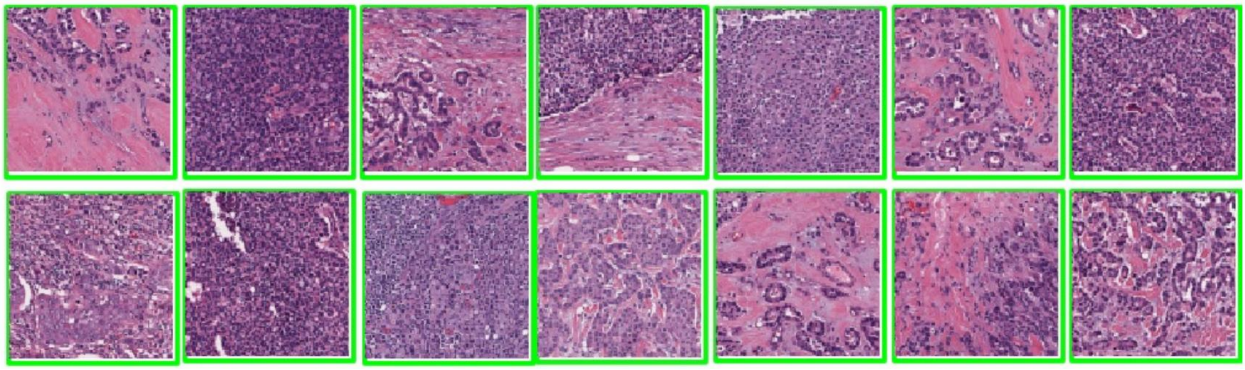
Each patch's file name is of the format:

u_xX_yY_classC.png –> example 10253_idx5_x1351_y1101_class0.png

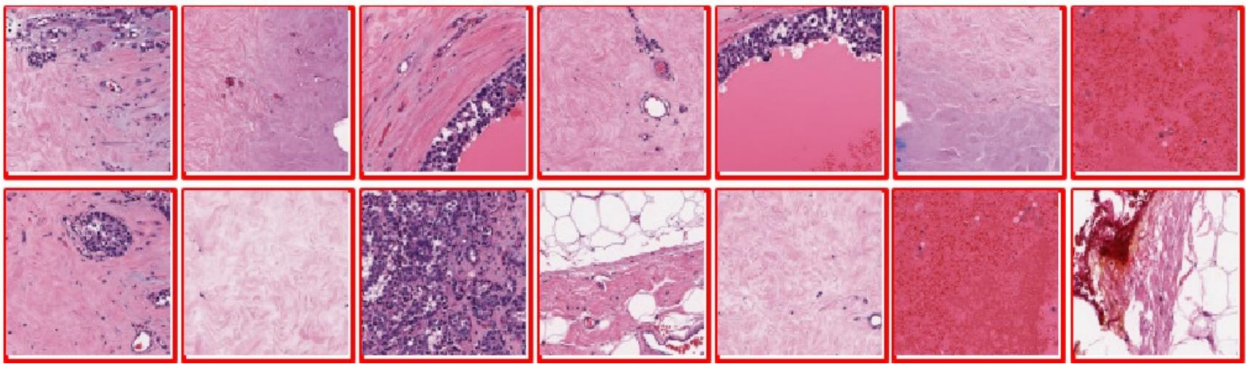
Where u is the patient ID (10253_idx5), X is the x-coordinate of where this patch was cropped from, Y is the y-coordinate of where this patch was cropped from, and C indicates the class where 0 is non-IDC and 1 is IDC.

Data Source: http://andrewjanowczyk.com/wp-static/IDC_regular_ps50_idx5.zip

Positive IDC cases (Class 1)



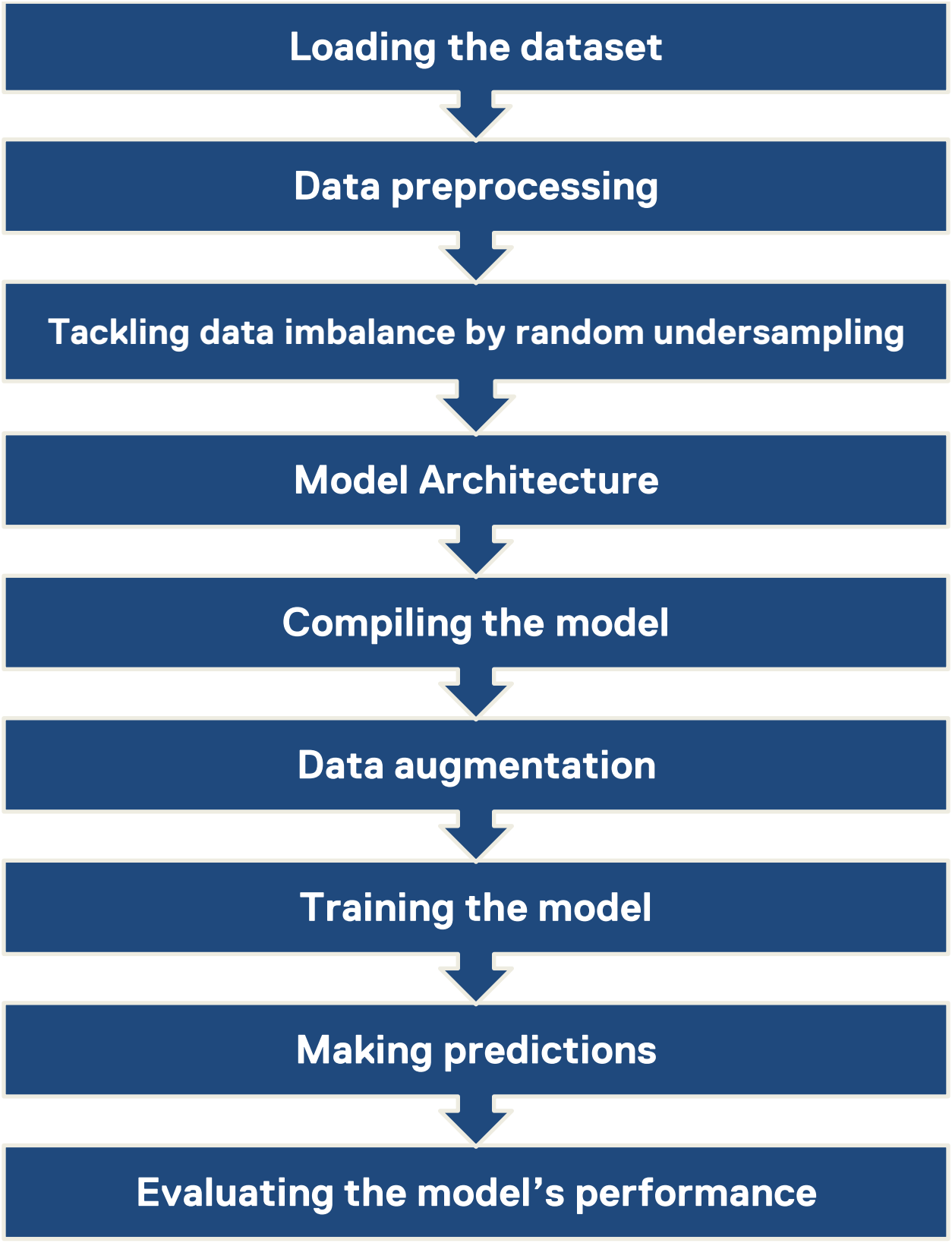
Healthy Tissue /Negative IDC cases (Class 0)



Pic.1) Some examples of the Class 0 and Class 1 patches are shown in the above images (Cruz-Roa et al., 2014) from the dataset

Approach

The Approach steps are based on Bikram Baruah's GitHub source code (Baruah, 2019). In the detailed description of the steps, some brief code snippets are shown from this source code



Step 1: Loading the dataset

The dataset contains 279 folders and subfolders of 0 and 1 inside each of the folders will be saved to class_one and class_zero. The class_one and class_zero will save the image locations of the all class 1 and 0 respectively.

A function will be then created that will take the starting and end index of the images, read them using OpenCV's cv2.imread() and resize the images which are not in the 50X50X3 format, to this format.

The result of the function is two arrays

X: array of the resized images

Y: array of the respective labels

```
def process_images(lowerIndex,upperIndex):
    height = 50
    width = 50
    channels = 3
    x = []
    y = []
    for img in imagePatches[lowerIndex:upperIndex]:
        full_size_image = cv2.imread(img)
        image = (cv2.resize(full_size_image, (width,height), interpolation
=cv2.INTER_CUBIC))
        x.append(image)
        if img in classZero:
            y.append(0)
        elif img in classOne:
            y.append(1)
        else:
            return
    return x,y
```

Step 2: Data preprocessing

The list X will then be converted to a numpy array and also converted to a float32 data type, for space saving. The images will be normalized to values between 0 and 1 by dividing them by 255.

The dataset will be split into an 85:15 training: testing set.

```
X, Y = process_images(0,60000)
X = np.array(X)
X = X.astype(np.float32)
X /= 255.
```

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,Y,test_size=0.15)
```

Step 3: Tackling data imbalance by random under-sampling

By randomly removing the samples from the majority class, I will deal with the problem of data imbalance, by making sure the sample size of both the classes are equal. The images which were not in the original dimensions of 50X50X3 will then be converted to this size.

```
X_trainShape = X_train.shape[1]*X_train.shape[2]*X_train.shape[3]
X_testShape = X_test.shape[1]*X_test.shape[2]*X_test.shape[3]
X_trainFlat = X_train.reshape(X_train.shape[0], X_trainShape)
X_testFlat = X_test.reshape(X_test.shape[0], X_testShape)
```

```
from imblearn.under_sampling import RandomUnderSampler
random_under_sampler = RandomUnderSampler(ratio='majority')
```

```
X_trainRus, Y_trainRus = random_under_sampler.fit_sample(X_trainFlat, y_tr
ain)
X_testRus, Y_testRus = random_under_sampler.fit_sample(X_testFlat, y_test)
```

Step 4: Model Architecture

I will create a model architecture in a layer by layer manner as described in (Khan & Yong, 2017).
The architecture will have convolutional layers, max pooling layers, dropout layers and fully connected layers.
The first layer is a convolutional layer with 32 filters each of size 3 x 3. In which the input shape in the first layer will be specified to 50 x 50 x 3.
I will be using the Rectified linear unit (ReLU) activation function for all the layers except the final output layer.
The second layer is a pooling layer which is used for dimensionality reduction. Max Pooling with a 2x2 window only considers the maximum value in a 2x2 window.
The third layer is again a convolutional layer of 64 filters each of size 3 x 3 followed by another max pooling layer of 2x2 window. The Complexity of learning increases in direct proportion to the number of filters, starting with simple features to the more complex features of the images.
The next two layers are again convolutional layers with the same filter size but increasing number of filters; for example, 128 and 256.
I will then flatten the layers from 3D feature map to 1D feature vectors and then add the fully connected layers.
I will then add a dropout layer with 0.5 rate (Hinton et al., 2012), which would randomly switch off 50% of the neurons and avoid overfitting. This will help to generalize the model and reduce the training time.
The next layer of 128 neurons is followed by a 0.5 dropout layer, then another 128-neuron dense layer and then a final layer or the output layer with the neurons equal to the class size. The activation layer in the output layer is set to sigmoid as I am interested in binary classification of the images, wither class one or class zero.

```
model.add(Conv2D(32, kernel_size=(3,3),
                 activation='relu',
                 input_shape=(50,50,3)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, (3,3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(128, (3, 3), activation='relu'))
model.add(Conv2D(256, (3, 3), activation='relu'))
model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(128, activation='relu'))
model.add(Dense(num_classes, activation='sigmoid'))
```

Step 5: Compiling the model

I will be using the ADAM (adaptive moment estimation) optimizer as the binary cross entropy loss function (Brownlee, 2017). This is used to make the network "learn from its mistakes" in an iterative manner, to get better accuracy (metric used while compiling), as it progresses through the layers of architecture.
The learning rate in ADAM for each parameter is adaptive, as opposed to stochastic gradient descent, and the learning rate changes as the network weight for the loss function changes.

```
model.compile(loss=keras.losses.binary_crossentropy,
              optimizer=keras.optimizers.Adam(lr=0.00001),
```

```
              metrics=['accuracy'])
```

Step 6: Data Augmentation

I will be using the Keras ImageDataGenerator to perform data augmentation. This will generate small batches of the images in real time during the training phase and perform transformations, like, flipping or rotating the images, along the axes. This will help in classifying the images even if they are originally rotated at an angle and flipped horizontally or vertically.

```
datagen = ImageDataGenerator(
    featurewise_center=True,
    featurewise_std_normalization=True,
    rotation_range=180,
    horizontal_flip=True,vertical_flip = True)
```

Step 7: Training the model

I will be using the method described in (Bhatia, 2018) to train the model using NVIDIA GPU.

I will use an epoch number of around 90 and avoid overfitting by using Early Stopping regularization. I will also use ModelCheckpoint from Keras to save the model, when the validation loss is minimum. The best model generated thus, will be used in later steps for predictions and evaluating the performance.

I will use the model.fit_generator() to train the model, with the variable set to training to later plot the training loss and validation loss to give me an idea of the variance.

To validate, I will use the results obtained after under-sampling the test sets.

```
training = model.fit_generator(datagen.flow(X_trainRusReshaped,Y_trainRusH
ot,batch_size=batch_size),steps_per_epoch=len(X_trainRusReshaped) / batch_
size, epochs=epochs,validation_data=(X_testRusReshaped, Y_testRusHot), ver
bose=1, callbacks=[early_stopping_monitor, model_checkpoint])
```

Step 8: Making predictions

I will use the results obtained after the ModelCheckpoint step and use the predict function for class predictions.

```
plt.plot(training.history['loss'])
plt.plot(training.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

Step 9: Evaluating the model's performance

I will be using the confusion matrix to evaluate the performance of the model.

```
confusion_matrix = metrics.confusion_matrix(y_true=y_true_labels, y_pred=y
_pred_labels)
print(confusion_matrix)
```

The above 9 steps will be then fine-tuned with respect to the hyperparameters (learning rate, batch sizes, changing the filters of the CNN layers, adding more layers) to improve the initially obtained accuracy rate.

Results

The CNN architecture was run using increasing batch sizes of the images starting from 60,000 then 80,000 and finally 100,000.

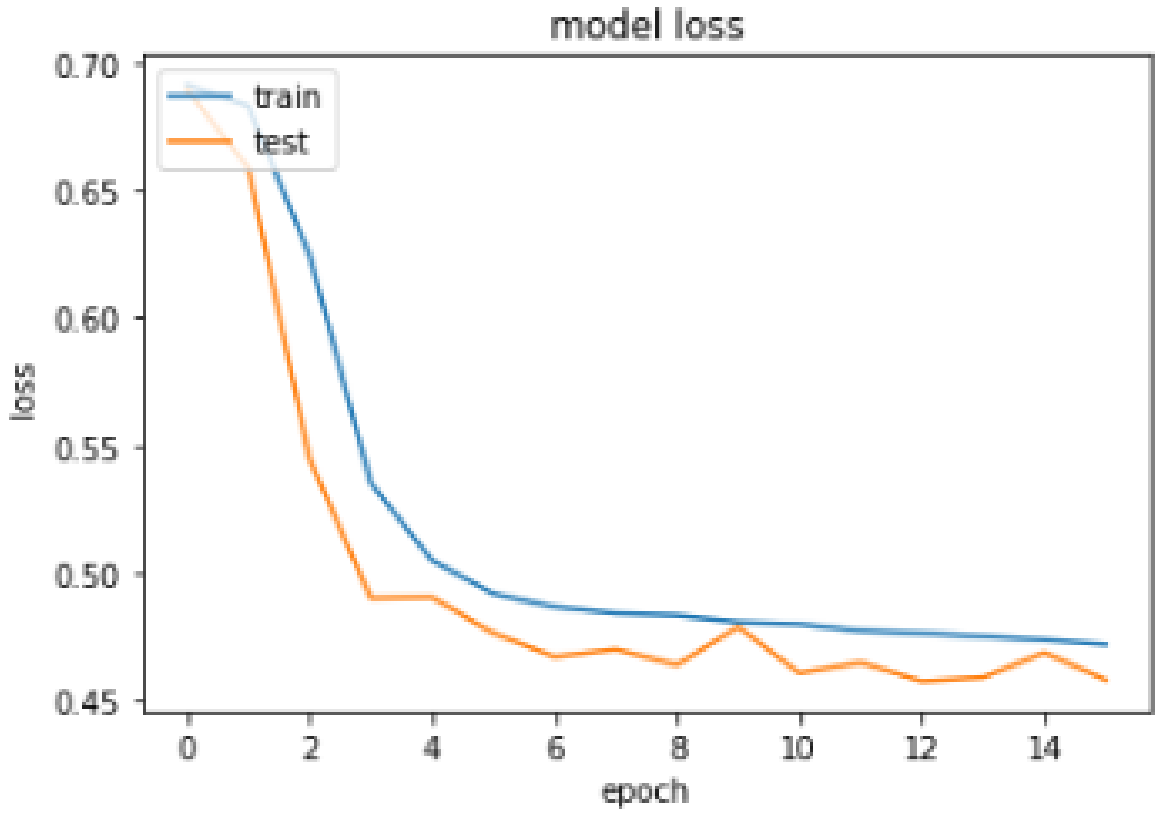
The confusion matrix can be simplified as follows

Predicted malignant and actually malignant *(True Positive)*	Predicted malignant but actually benign *(False Positive)*
Predicted benign but actually malignant *(False Negative)*	Predicted benign and actually benign *(True Negative)*

The below images show the model loss graphs and the confusion matrix according to the image sizes used to design the model.

All the model loss graphs, show that the proposed CNN model produces very low variance. The plots confirm that the model is not over-fitting, irrespective of the batch sizes of the images.

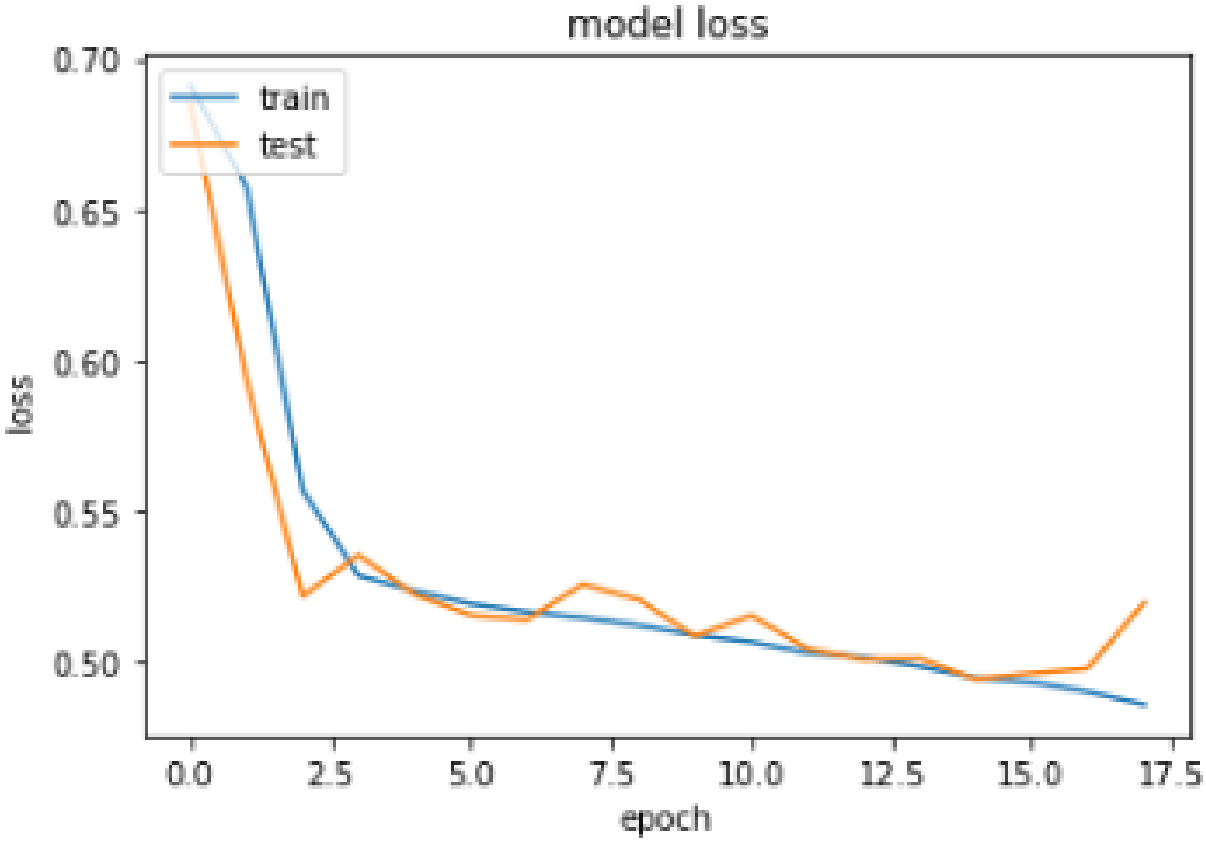
1) For 60,000 images



Confusion Matrix

		Actual Class	
		Positive	Negative
Predicted Class	Positive	1873	434
	Negative	479	1828

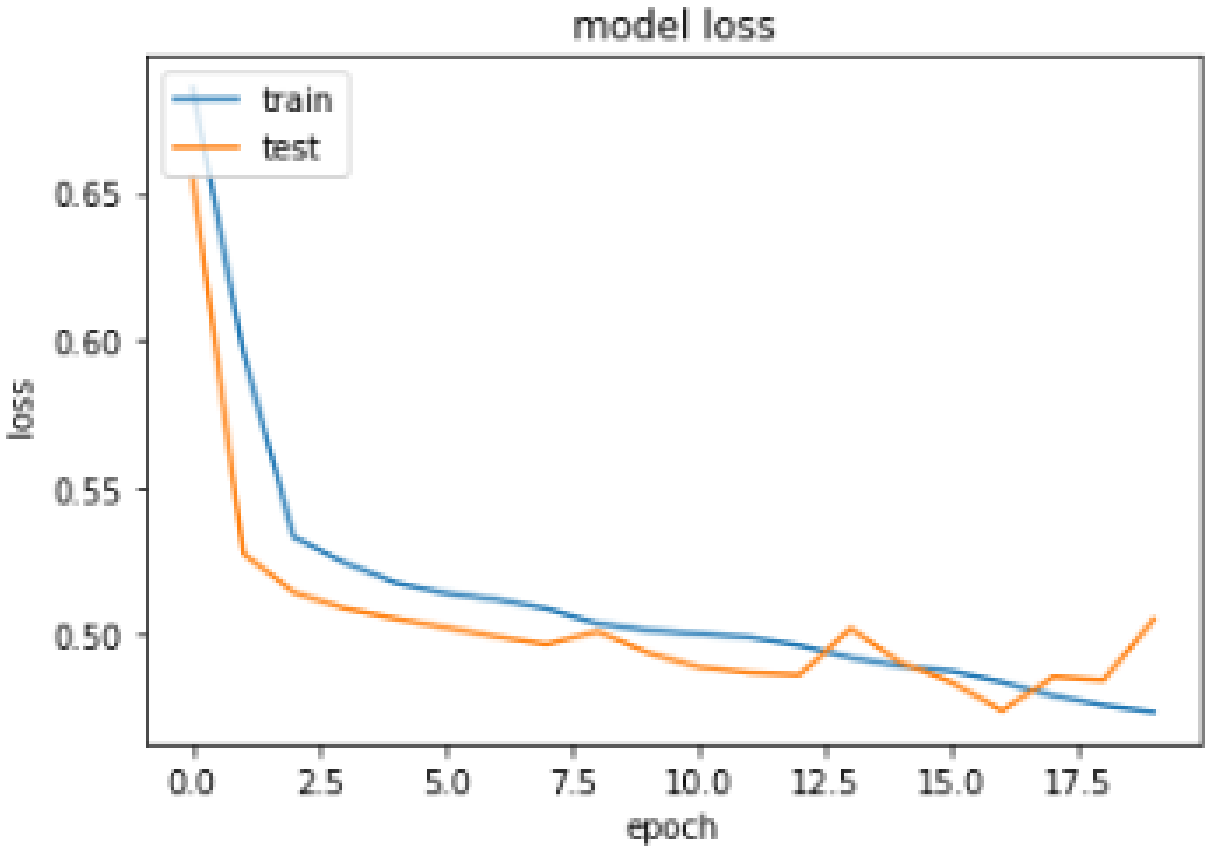
2) For 80,000 images



Confusion Matrix

		Actual Class	
		Positive	Negative
Predicted Class	Positive	2478	813
	Negative	674	2617

3) For 100,000 images



Confusion Matrix

		Actual Class	
		Positive	Negative
Predicted Class	Positive	3347	974
	Negative	858	3463

All the important comparative metrics are shown in the table below

	60,000 images	80,000 images	100,000 images
Sensitivity	79.63%	78.62%	79.60%
Specificity	80.81%	76.30%	78.05%
Precision	81.19%	75.30%	77.46%
Negative Predictive Value	79.24%	79.52%	80.14%
False Positive Rate	19.19%	23.70%	21.95%
False Discovery Rate	18.81%	24.70%	22.54%
False Negative Rate	20.37%	21.38%	20.40%
Accuracy	80.21%	77.41%	78.80%
F1 Score	80.40%	76.92%	78.51%
Matthews Correlation Coefficient	60.44%	54.87%	57.62%

Conclusions

As observed in the above results, the proposed CNN architecture performs the best even when 60,000 images are used as input. All of the metrics, in the case of 60,000 images, are better compared to when the batch size is 80,000 and 100,000.

The time taken to run the epochs is less in case of using 60,000 images (˜4 minutes) compared to 80,000 images (˜ 6 minutes) and 100,000 (˜8 minutes). The time taken is important when this model would be deployed in real world case scenarios as the test results would be obtained much quicker, leading to faster diagnosis and treatments.

The Matthews Correlation Coefficient, which is the measure of the quality of binary classifications (benign vs malignant), is also better. This metric is the most informative single score to establish the quality of a binary classifier prediction in a confusion matrix context (Chicco, 2017)

The False Negative rate is also better compared to the other batch sizes, which would mean that it is less likely that a malignant tumor, or IDC, is incorrectly classified as a benign tumor. In cases such as classification of tumors, having a lower false negative is better than having a lower false positive.

References

Araújo, T., Aresta, G., Castro, E., Rouco, J., Aguiar, P., Eloy, C., . . . Campilho, A. (2017). Classification of breast cancer histology images using Convolutional Neural Networks. *PLOS ONE, 12*(6), e0177544. Retrieved from https://doi.org/10.1371/journal.pone.0177544. doi:10.1371/journal.pone.0177544

Arel, I., Rose, D. C., & Karnowski, T. P. (2010). Deep Machine Learning - A New Frontier in Artificial Intelligence Research [Research Frontier]. *IEEE Computational Intelligence Magazine, 5*(4), 13-18. doi:10.1109/MCI.2010.938364

Brook, A., El-Yaniv, R., Isler, E., Kimmel, R., Meir, R., & Peleg, D. (2007). Breast Cancer Diagnosis From Biopsy Images Using Generic Features and SVMs.

George, Y. M., Zayed, H. H., Roushdy, M. I., & Elbagoury, B. M. (2014). Remote Computer-Aided Breast Cancer Detection and Diagnosis System Based on Cytological Images. *IEEE Systems Journal, 8*(3), 949-964. doi:10.1109/JSYST.2013.2279415

Kowal, M., Filipczuk, P., Obuchowicz, A., Korbicz, J., & Monczak, R. (2013). Computer-aided diagnosis of breast cancer based on fine needle biopsy microscopic images. *Computers in Biology and Medicine, 43*(10), 1563-1572. Retrieved from http://www.sciencedirect.com/science/article/pii/S0010482513002175. doi:https://doi.org/10.1016/j.compbiomed.2013.08.003

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Commun. ACM, 60*(6), 84–90. Retrieved from https://doi.org/10.1145/3065386. doi:10.1145/3065386

Naik, S., Doyle, S., Agner, S., Madabhushi, A., Feldman, M., & Tomaszewski, J. (2008, 14-17 May 2008). *Automated gland and nuclei segmentation for grading of prostate and breast cancer histopathology.* Paper presented at the 2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro.

Petushi, S., Garcia, F. U., Haber, M. M., Katsinis, C., & Tozeren, A. (2006). Large-scale computations on histology images reveal grade-differentiating parameters for breast cancer. *BMC Med Imaging, 6*, 14. Retrieved from https://www.ncbi.nlm.nih.gov/pubmed/17069651. doi:10.1186/1471-2342-6-14

Spanhol, F. A., Oliveira, L. S., Petitjean, C., & Heutte, L. (2016, 24-29 July 2016). *Breast cancer histopathological image classification using Convolutional Neural Networks.* Paper presented at the 2016 International Joint Conference on Neural Networks (IJCNN).

Baruah, B. (2019, January 3). *Bikramb98/invasive-ductal-carcinoma-cnn*. GitHub. https://github.com/bikramb98/invasive-ductal-carcinoma-cnn

Bhatia, A. (2018, January 25). *Set up GPU Accelerated Tensorflow & Keras on Windows 10 with Anaconda*. https://medium.com/@ab9.bhatia/set-up-gpu-accelerated-tensorflow-keras-on-windows-10-with-anaconda-e71bfa9506d1

Brownlee, J. (2017, July 2). Gentle Introduction to the Adam Optimization Algorithm for Deep Learning. *Machine Learning Mastery*. https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/

Chicco, D. (2017). Ten quick tips for machine learning in computational biology. *BioData Mining*, *10*. https://doi.org/10.1186/s13040-017-0155-3

Cruz-Roa, A., Basavanhally, A., González, F., Gilmore, H., Feldman, M., Ganesan, S., Shih, N., Tomaszewski, J., & Madabhushi, A. (2014). *Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks* (M. N. Gurcan & A. Madabhushi, Eds.; p. 904103). https://doi.org/10.1117/12.2043872

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *CoRR, abs/1207.0580*. http://arxiv.org/abs/1207.0580

Khan, S., & Yong, S.-P. (2017). A deep learning architecture for classifying medical images of anatomy object. *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 1661–1668. https://doi.org/10.1109/APSIPA.2017.8282299