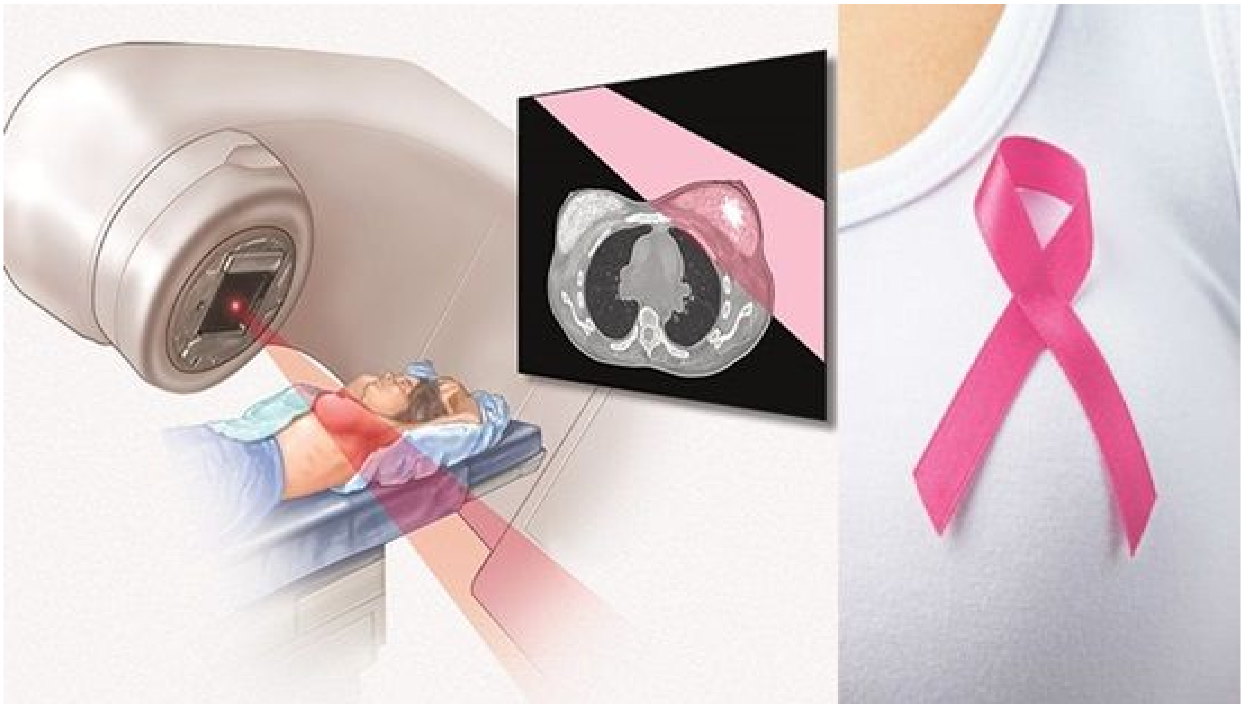


Breast Cancer (IDC) Detection

Using Digital Pathology Images.

25 November 2018

Animesh Seemendra
Machine Learning Engineer Advance Nanodegree
Capstone Project Proposal

Domain Background

Breast Cancer is the most common type of cancer in woman worldwide accounting for 20% of all cases. In 2012 it resulted in 1.68 million new cases and 522,000 deaths. One of the major problems is that women often neglect the symptoms, which could cause more adverse effects on them thus lowering the survival chances. In developed countries, the survival rate is although high, but it is an area of concern in the developing countries where the 5-year survival rates are poor. In India, there are about one million cases every year and the five-year survival of stage IV breast cancer is about 10%. Therefore it is very important to detect the signs as early as possible.

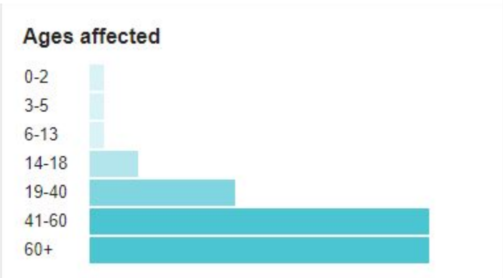


Fig 1

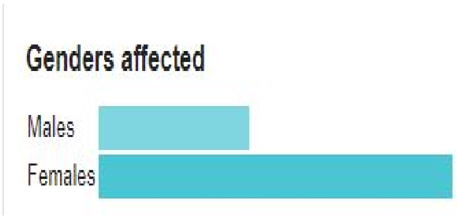


Fig 2

The data is on Breast cancer in India taken from various hospitals (Apollo and others)

Invasive ductal carcinoma (IDC) is the most common form of breast cancer. About 80% of all breast cancers are invasive ductal carcinomas. Doctors often do the biopsy or a scan if they detect signs of IDC. The cost of testing for breast cancer sets one back with $5000, which is a very big amount for poor families and also manual identification of presence and extent of breast cancer by a pathologist is critical. Therefore automation of detection of breast cancer using Histopathology images could reduce cost and time as well as improve the accuracy of the test. This is an active research field lot of research papers and articles are present online one that I like is -https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5453426/ as they used deep learning approach to study on histology images and achieved the sensitivity of 95% which is greater than many pathologists (~90%). This shows the power of automation and how it could help in the detection of breast cancer. The motivation to work on this domain came from the lesson I studied in the nanodegree about skin cancer and how AI could help medical field and saves multiple lives, therefore, it inspired me to create my capstone project on topic related to cancer which is the most deadly form of all diseases and especially breast cancer which often get neglected.

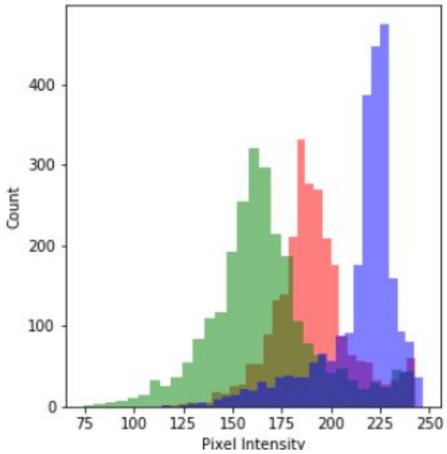
Problem Statement

The idea is to use pathology test images and classify them as IDC(+) and IDC(-). Accurately identifying and categorizing breast cancer subtypes is an important clinical task, and automated methods can be used to save time and reduce error. The pathological tests include images of the tissues, the task is to train a computer to use these images and respond on whether the person is IDC(+) or IDC(-). Since it is a medical field problem it is important that sensitivity of the output should be high.

Datasets and Inputs

The problem statement involves using images that are obtained during pathology tests to detect whether the patient is IDC positive or negative. Histopathology images are the images of tissues that are obtained during pathology tests, therefore, these images will act as inputs for the problem.

The dataset that contains histopathology images for breast cancer is present on kaggle at : https://www.kaggle.com/paultimothymooney/breast-histopathology-images . The dataset contains 198,738 of negative IDC and 78,786 of positive IDC, therefore, it is a good dataset with enough data for our task. The original dataset consisted of 162 whole mount slide images of Breast Cancer (BCa) specimens scanned at 40x. However, the data that I have selected contains images that are cropped from the original dataset i.e it contains patches of regions where the IDC occurs, making it more specific to our problem. All images present in the data are of different dimensions, most of them are (50 X 50) while others have one or both dimension less than 50 which means that images are small. The RGB pixel intensity of the images is :



This is for IDC(-) but it lot similar to IDC(+), we can also see that pixels have a higher count at the end of the intensity range.

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.

The data needed to be split between training, validation and testing set. But classes are not even so we need to take care of this too. Therefore I would prefer to split the data into 70-30 % between training and testing and 30% from training set as a validation set. Sampling is required in this type of data to maintain even balance between classes. We are working with medical images so I would prefer undersampling over oversampling. The data might get small therefore image argumentation may be required.

Solution Statement

Our data involves images with the classes written on data file name, therefore, we would need to extract the class name from it and create a column to store them. We also need to split the dataset into the training set, validation set and testing set. Testing set for checking how good the model works on completely unseen data and validation set to check and avoid underfit or overfit, the will also help to select the best model. One hot encoding will be done in classes column so that it could work better with our model. Image processing step is also required to reduce the pixel range from 0-250 to 0-1. After it CNN model is to be used to predict the class, CNN creates an effective architecture the 2D structure of the image, therefore, it would be the best to use, considering that we are working with the images.

Benchmark Model

The breast cancer detection using pathology images is a critical task even for pathologists. Considering our data contains images and in order to retain their 2D structure, CNN would be the best choice. Therefore I would consider a Vanilla CNN model to be my benchmark model. And will try to beat it with different transfer learning and image argumentation techniques.

Evaluation Metrics

The performance of the model will be evaluated using ROC curve and confusion matrix. A receiver operating characteristic curve, i.e., ROC curve, is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied. The ROC curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The true-positive rate is also known as sensitivity, recall or probability of detection in machine learning. The false-positive rate is also known as the fall-out or probability of false alarm and can be calculated as (1 − specificity). It uses the concept of true positive, true negative, false positive and false negative.

Sensitivity = $\frac{\text{True Positive}}{\text{True Positive + False Negative}}$.	Recall = $\frac{\text{True Positive}}{\text{True Positive + False Negative}}$.
Specificity = $\frac{\text{True Negative}}{\text{True Negative + False Positive}}$.	Precision = $\frac{\text{True Positive}}{\text{True Positive + False Positive}}$.

The perfect classification has the area under the ROC curve equal to 1. Therefore closer the area of our ROC curve to 1 better would be our model. The third is a confusion matrix, it is a two by two table that contains four outcomes produced by a binary classifier. Various measures, such as error-rate, accuracy, specificity, sensitivity, and precision, are derived from the confusion matrix. Sensitivity can be calculated from the confusion matrix, which is important to know when we work in the medical domain i.e how many of the patients were told about having breast cancer our of how many were actually having it.

The ROC curve and confusion matrix would be a good evaluation matrix because they both are used for binary classification and our data is also based on binary classification. These metrics could help in evaluating the model through sensitivity, specificity, recall and precision which all are important and are always considered while working in this domain with it would provide us with the visualization of the correctness of the model.

Project Design

The workflow of solving the problem would be in the following order :

- Import Modules
- Explore and Visualizing Data
- Extraction of classes from the file path
- Preprocessing the Data
    - Converting categorical data into one hot encoding.
    - Image processing
        - Image resize to 50X50
        - Image pixel scaling to (0-1) from (0-250)
- Sampling and Training, validation and testing data split.
- Model using CNN from scratch
    - Model Building
    - Model evaluation
- Model using Transfer Learning
    - Model Building

    - Model Evaluation
- Model using transfer learning and Image argumentation
    - Model building
    - Model evaluation
- Comparison of models
- Evaluation of best model using confusion matrix and ROC curve
- An algorithm to direct supply the image to the model.
- Final Conclusion

Model Details:

Our data are images of tissues and we need to capture the complex structure of them and also the size of the images are small so therefore I am considering that the architecture should have 2 layers of CNN with increasing number of filters to capture the complex part along with it there should be max-pooling layers to reduce the size , also certain dropout layers need to be added in order to avoid overfitting. The model would somewhat look like :

```
model = Sequential()
model.add(Conv2D(32,activation='relu', input_shape=input_shape))
model.add(Dropout(0.35))
model.add(Conv2D(64, activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.4))
model.add(Flatten())
model.add(Dense(256, activation='relu'))
model.add(Dropout(0.45))
model.add(Dense(num_classes, activation='softmax'))
```

And since it is a binary classification problem the compile part would be like:

```
model.compile(optimizer='rmsprop',loss='binary_crossentropy',metrics=['accuracy'
])
```

Although since the model is not tested yet, this is just the architecture that I have in my mind, it may go through lot changes while working on the project. I have good experience with some transfer learning models such as Inception_V3 , VGG-16, therefore, I tend to go with them initially. Finally, if the things do not work out with the vanilla model, I tend to increase the number of filters and layers to capture the more complex structure of the images and add image argumentation techniques to improve performance and even try different optimizers. And while working with learning models I would try VGG-19 if others fail to impress in terms of performance.

Refrences

- https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5453426/
- https://www.breastcancer.org/research-news/study-on-accuracy-of-biopsy-results
- https://www.cbsnews.com/news/breast-biopsies-often-get-it-wrong/
- https://en.wikipedia.org/wiki/Breast_cancer
- http://www.breastcancerindia.net/statistics/trends.html
- https://classeval.wordpress.com/introduction/basic-evaluation-measures/
- https://en.wikipedia.org/wiki/Receiver_operating_characteristic