RESEARCH

Introduction of Profile Areas of Data Science :Project 3

Suresh Kumar Choudhary, Aman Jain and Frenny Macwan

Full list of author information is available at the end of the article
*Equal contributor

Abstract

Goal of the Project: To build a Deep Learing algorithm to classify the breast tumour tissue as Benign (Non - Cancer) or Malignant (Cancer) through breast cancer histopathology images.

Main Result of the Project: A Deep Learning Based Classifier (CNN) was built to classify the tumour into cancerous or non cancerous.

Personal Key Leanings: We created a deep learning classifier for the first time, and understood the vital aspects of the algorithm.

Estimate working hours: 8

Project Evaluation: 1

Number of Words: 1000

Keywords: Convolutional Neural Network CNN; DenseNet201; Deep Learning; breastcancer histopathology images

Scientific Background

Mortality Rate of Breast Cancer is very high as compared to the other types of Cancers. Diagnosis of Breast Cancer can be achieved by X-Rays, Sonography etc however biopsy is the only way to diagnose with confidence if cancer is really present. Among biopsy techniques, the most common are fine needle aspiration, core needle biopsy, vacuum-assisted, and surgical (open) biopsy. and later placed under the microscope. Breast Cancer histopathology image analysis is done, and in particular is being used for the automated classification[1] of benign or malignant images.

Goal

The goal of this project is to develop or train a Deep Learning based classifier (CNN) that works directly on the images and classify the breast tumour tissue as Malignant (cancer) or Benign (non cancer).

Data

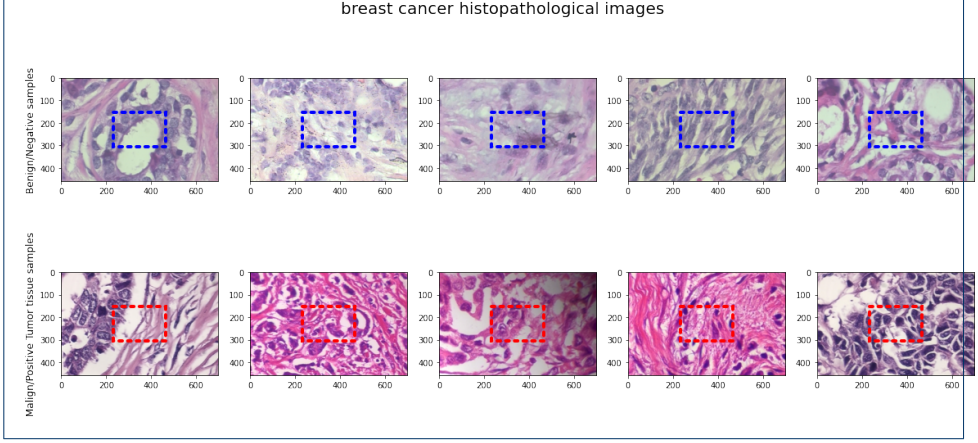
Dataset[2] is composed of 7,909 microscopic images of breast tumor tissue collected from 82 patients using different magnifying factors (40X, 100X, 200X, and 400X) from Breast Cancer Histopathological Image Classification (BreakHis). To date, it contains 2,480 benign and 5,429 malignant samples (700X460 pixels, 3-channel RGB, 8-bit depth in each channel, PNG format). For model building, we have restricted ourselves with 400X images. The details of dataset with 400x images can be found in figure 1

The individual class random samples of dataset can be found in figure 2

Figure 1 Dataset Details(400X Images)

Description	
Format	PNG
Size	700x460
Channels	3
Bits per channel	8
Data type	Unsigned char
Total images	1820
Malign	1232
Benign	588

Figure 2 Dataset Samples (Malign and Benign)



Proposed Method

A pretrained DenseNet201 with added layer and a customized Convilutional Neural Network(CNN) have been used to get results on above stated dataset. The DenseNet has different versions, like DenseNet-121, DenseNet-160, DenseNet-201, etc. The numbers denote the number of layers in the neural network. The DenseNet 201 has the following layers :

- 5 - Convolution and Pooling Layers
- 3 - Transition Layers (6, 12, 48)
- 1 - Classification Layer (32)
- 2 - Dense Block (1*1 and 3*3)

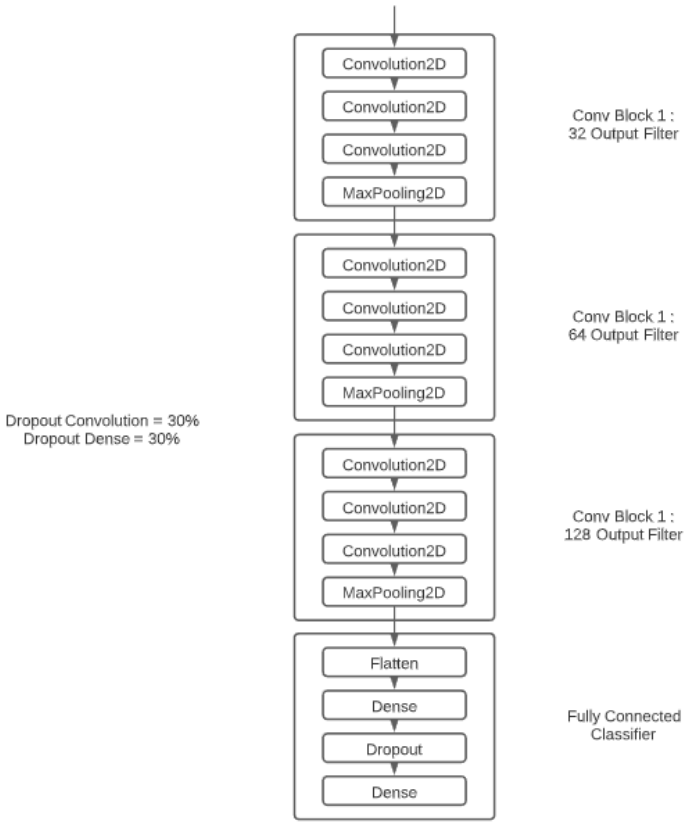
Hence summing up to 201 Layers:

$$[5 + (6 + 12 + 48 + 32) * 2] = 201$$

A proposed convolutional neural network (CNN) architecture consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN

typically consist of a series of convolutional layers that convolve with a multiplication or other dot product. The activation function is commonly a ReLU layer, and is subsequently followed by additional convolutions such as pooling layers, fully connected layers and normalization layers, referred to as hidden layers because their inputs and outputs are masked by the activation function and final convolution. We tried a couple of models with various architecture and parameters of CNN. However, the most effective we found was using 4 layers, and to preserve the feed-forward nature, each layer obtains additional inputs from all preceding layers and passes on its own feature-maps to all subsequent layers. The 1st layer, 2nd layer and 3rd layer are using 32, 64 and 128 filters respectively. Each layer we added 3 convolution nodes, 1 maxPooling Node and add a dropout layer after each convolutional layer and set the drop out rate to 0.3 (30%). The proposed convolutional neural network architecture can be visualized in figure 3.

Figure 3 Proposed Convolutional Neural Network Architecture

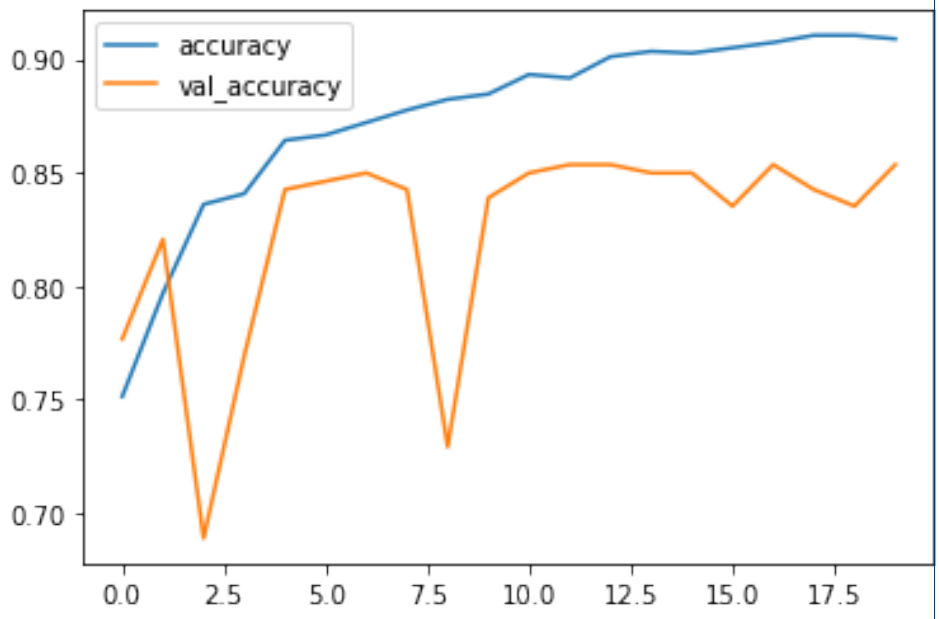


Result

As part of model evaluation, the dataset has been divided into three parts as training set(70%), validation set(15%) and test set(15%). We trained the model on training set, validated on validation set and tested on test set. The test set has been used to calculate the accuracy 4, loss 5, area under curve(auc), precision/recall 7, ROC analysis for both densenet201 with additional layers and convolutional neural

network(CNN) with proposed architecture. We found the best results with convolutional neural network with proposed architecture with accuracy . We also generated the ROC Curve 6 for these two and calculated their respective AUCs and found out proposed CNN the best model with AUC as 0.87.

Figure 4 Proposed CNN Model Accuracy with Epochs



Since Malignant samples are cancerous, and we require more precision because of it's sensitive nature, proposed one gave a precision of 0.84 whereas the densenet201 gave a precision of 0.66.

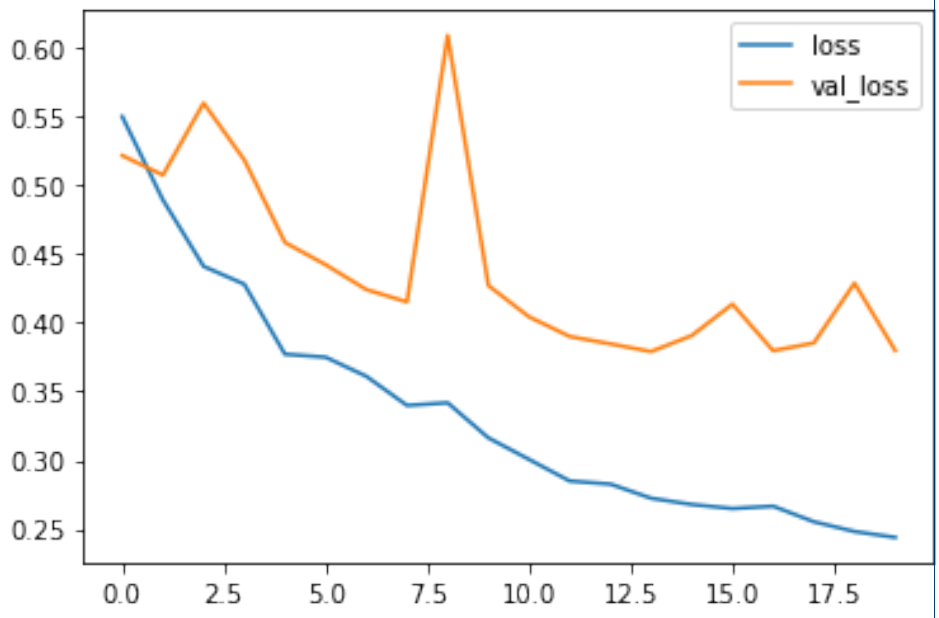
Discussion-1

As part of result, we have plotted the accuracy and loss for training set and validation set to evaluate the proposed model. We found that if we increases the number of epochs , the accuracy also increases for training set but for validation set it becomes stable after some epochs. However, model loss decreases if we increase the epochs. Hence, we can conclude still if we increase the epochs then it might possible that we may get better results.

Discussion-2

Breast cancer is the most common cancer amongst women in the world. It accounts for 25 percent of all cancer cases, and affected over 2.1 Million people in 2015 alone.Early diagnosis significantly increases the chances of survival. The key challenges against it's detection is how to classify tumors into malignant (cancerous) or benign (non cancerous). Recently, deep learning models have made remarkable progress in computer vision, specifically in biomedical image processing, due to their abilities to automatically learn complicated and advanced features from images, which inspired various researchers to leverage these models in the classification of breast cancer histopathology images. So we can consider this as a data science project.

Figure 5 Proposed CNN Model loss with Epochs



Appendix

Code -

- Aman - Data Gathering, Augmentation
- Suresh - Training and creating model, Result Evaluation(ROC, Accuracy, Precision, Recall etc)
- Frenny - Image Preprocessing

Report -

- Aman - Abstract, Scientific background
- Suresh - Results, Discussion 1
- Frenny - Goal, Data, Discussion 2

Author details

References

1. Spanhol, F.A., Oliveira, L.S., Petitjean, C., Heutte, L.: A dataset for breast cancer histopathological image classification. IEEE Transactions on Biomedical Engineering **63**(7), 1455–1462 (2015)
2. Dataset. https://web.inf.ufpr.br/vri/databases/breast-cancer-histopathological-database- breakhis/

Figure 6 ROC Curve for Densenet201 and Proposed Architecture

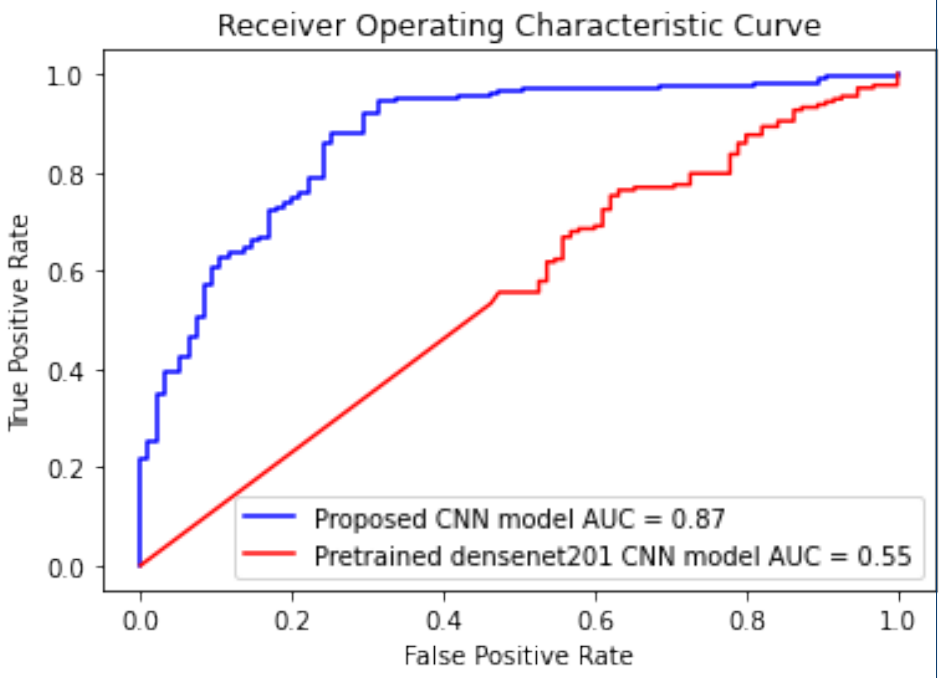


Figure 7 Proposed CNN Model classification Report

```
from sklearn.metrics import classification_report

# Generate a classification report

# For this to work we need y_pred as binary labels not as probabilities
y_pred_binary = predictions.argmax(axis=1)

report = classification_report(y_true, y_pred_binary, target_names=cm_plot_labels)

print(report)
```

```
              precision    recall  f1-score   support

      Benign       0.86      0.67      0.76        95
      Malign       0.84      0.94      0.89       178

    accuracy                           0.85       273
   macro avg       0.85      0.81      0.82       273
weighted avg       0.85      0.85      0.84       273
```

Figure 8 Densenet201 with additional layer Model classification Report

```
from sklearn.metrics import classification_report

# Generate a classification report

# For this to work we need y_pred as binary labels not as probabilities
y_pred_binary = predictions.argmax(axis=1)

report = classification_report(y_true, y_pred_binary, target_names=cm_plot_labels)

print(report)
```

```
              precision    recall  f1-score   support

      Benign       0.38      0.28      0.33        95
      Malign       0.66      0.75      0.71       178

    accuracy                           0.59       273
   macro avg       0.52      0.52      0.52       273
weighted avg       0.56      0.59      0.57       273
```