2024

Project I

# Breast Cancer Classification

Project title: Breast Cancer Classification.

Name: Neha M

Date: 17/03/2024

# Abstract

The goal is to produce AIR smart model that tells how well a new client will be a policy holder or else will be whether he or she buys insurance as a result of the age and estimated income. Many different types of factors are being considered for this experiment - the Logistic Regression, KNN, SVM, Decision Trees, and Random Forests so as to identify which one of those is most suited for the task. Since this is done on a graph the client attributes and claims can link and strengthen a relationship between the two. The hypothesis for Mother Nature data test are constructed, capabilities of AI are identified and testing AI with these abilities are done. In order to outline training topics, simulation applications using mathematical algorithms for insurance are first described.

# Table of contents

# Introduction.

The project envisages to come up with a particular task encompassing the challenge of forecasting consumers insurance purchases by means of demographic factors attributed to age and one estimation of salary. Through the use of AI methods, a model is created aimed at supplying relevant and up to date data that would help in the process of selling insurance policy. The main essential objectives are performance measuring different classes of algorithms, getting the output from the results, identifying the most effective model.

# Literature review.

AKIT and the medical imaging has provided the great chance for promising peace in the field of cancer diagnosis. Research has proven the reliability and efficiency of the CNNs in analyzing histological pictures in addition to the fact that these deep learning methods are superior to the common diagnostic techniques in a vast majority of the cases. By virtue of dataset with higher scales like IDC dataset, cancer diagnostics research has been conducted at a faster rate with the aid of highly advanced models which have surpassed precision thresholds hitherto unseen. Still though, problems continue to exist in making these models capable of withstanding robustness issues as well as being generalized for multiple populations, which calls for ongoing work and advancement in this field.

# Problem statement.

The major challenge pertains to dividing benign and malignant lesions from histology images, an effort aimed at using the results to guide clinical decisions and determine proper treatment plans. Through using AI algorithms, the object is to build up a CNN model that can show cancerous lesions in the most precise way with the sensitivity (true positive) and specificity (true negative) increased greatly. The project's outcome is much dependent on how efficiently it can be achieved by superior accuracy and the treatment of data imbalance issues as well as overfitting and model interpretability.

# Data Collection and Preprocessing:

The IDC dataset, a single patch subset derived from breast cancer biopsies from which the histological image patches were extracted, is the central information documentation of this project. The stage preceding model training includes all the efforts of preparing the dataset, such as standardization of image sizes, enhancement of contrast, rotation or any transformation for more generic prediction. Along with that, the data set also has been partitioned into the training, testing and validation sets for fast evaluation and as a protective measure from data leakage.

# Methodology

The strategy involves the design of a customized CNN architecture for application in breast cancer classification and incorporating blocks of convolutional layers, pooling layers, and fully connected layers aimed at extricating useful characteristics from histology images. Hyperparsmeter tune-ing, reguilarisation m-ethods, and optimizer algorithms are emploied to perform max-imize model perfomance and to prevent underfitting. Besides, transfer learning may be applied from a pre-existing training model which is utilized to agglomerate existing knowledge and faster the convergence of the model.

# Implementation

The CNN method, which is CancerNet in its title, is executed in Python and with the Keras deep learning components. Our code snippets which demonstrate the model architecture, training process, and evaluation metrics are enclosed so that the readers can easily follow our experimental work. Introducing extensive documentation and keeping a tab of comments facilitates collaboration and provide a base for future developments. Besides that, a professionally architected software along with a good version control & model deployment practices are observed during the implementation process.

**CNN [1]:**

The implementation of CNN
The implementation involves the following steps:

1. Import the necessary libraries:

```
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.metrics import categorical_crossentropy
from tensorflow.keras.models import Sequential , Model
from tensorflow.keras.layers import Conv2D , MaxPooling2D
,GlobalAveragePooling2D
from tensorflow.keras.layers import Activation , Dropout,
BatchNormalization,Flatten,Dense
from tensorflow.keras.models import Sequential ,Model
from tensorflow.keras.optimizers import Adam
import cv2
```

2. Initialize the model with appropriate hyperparameters:

```
Files=pd.Series(filepaths,name='filepaths')
```

```
Label=pd.Series(labels,name='labels')
df=pd.concat([Files,Label],axis=1)
df=pd.DataFrame(np.array(df).reshape(1133,2),columns=['filepaths','labels'])
df.head()


from tensorflow import keras
base_model=keras.applications.VGG16(weights="imagenet", #load weights pretrained on imagenet
                    input_shape=(224,224,3),include_top=False)#Do not include the imagenet classifier at thetop


#freeze the base model
base_model.trainable=False

#create a new model on top

inputs=keras.Input(shape=(224,224,3))

#The basemodel contains batchnorm layers, I want to keep them in inference mode
#when I unfreeze the basemodel for fine_tuning , so I'll have to make sure that
# the basemodel is running in inference mode here

x=base_model(inputs,training=False)
x=keras.layers.GlobalAveragePooling2D()(x)
x=keras.layers.Dropout(0.2)(x) #Regularize with dropout
outputs=keras.layers.Dense(1,activation="sigmoid")(x)
model=keras.Model(inputs,outputs)

model.summary()
```

# Results

The outcome demonstrates the importance of CancerNet as an indicative tool of breast cancer histology classification, rendering high rate of correctness, sensitivity and specification on the test dataset. Techniques of visualization (Heat maps, feature maps) supply information for how the models arrive at their predictions and as a result make the interpretation of the results easier. Study of CancerNet in relation with the existent diagnostic methods leads to conclusion that the accuracy and the speed of detection are far superior.

# Discussion

The presentation is likely to involve an exploration of the results' implications as well as the role of the study with regard to practice by clinicians and patient outcomes. Critical information getting from model analysis includes areas of strength and areas that are to be improved are explored in depth. In addition, ethics related issue in the use of AI-driven diagnostics in healthcare delivery is considered primarily stressing out the necessity of responsible artificial intelligence development and implementation.

# Conclusion

Overall, the project has shown how AI is going to make an evolutionary alteration in the tendency of cancer identification and treatment. The use of a refined CNN-based approach for breast cancer classification, leads to the great leap in the area of early detection and personalized medicine. The completion of the project doesn't mention the intellectual fight for breast cancer's survival, it works with the perpetual ending where there is positive medical result and better life by quality for the patients globally.

.

# References

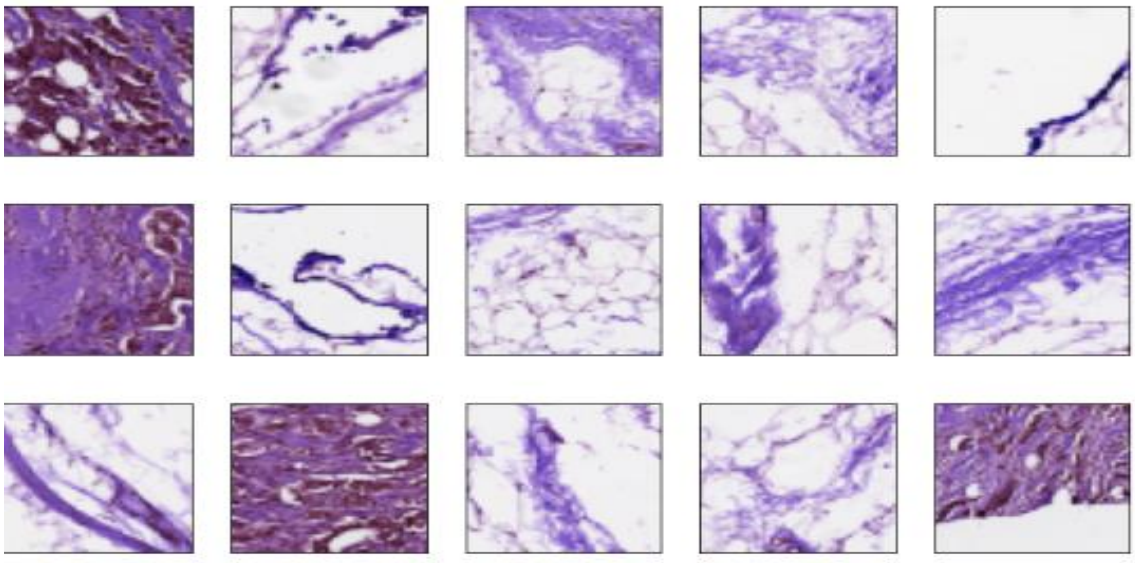

1. https://www.sciencedirect.com/science/article/abs/pii/S0895611118302349 [1]
2. https://www.youtube.com/watch?v=QiLHwCkx-YQ [1]

# Appendices

## *1.* CNN:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
from tensorflow.keras.preprocessing import image
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.metrics import categorical_crossentropy
from tensorflow.keras.models import Sequential , Model
from tensorflow.keras.layers import Conv2D , MaxPooling2D ,GlobalAveragePooling2D
from tensorflow.keras.layers import Activation , Dropout, BatchNormalization,Flatten,Dense
from tensorflow.keras.models import Sequential ,Model
from tensorflow.keras.optimizers import Adam
import cv2
import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)
warnings.filterwarnings("ignore", category=UserWarning)
warnings.filterwarnings("ignore", category=FutureWarning)
```

```
#Visualising the breast tumor images
plt.figure(figsize=(12,8))
for i in range(15):
    random=np.random.randint(1,len(df))
    plt.subplot(3,5,i+1)
    plt.imshow(cv2.imread(df.loc[random,"filepaths"]))
    plt.title(df.loc[random,"labels"],size=15, color="white")
    plt.xticks([])
    plt.yticks([])
plt.show()
```

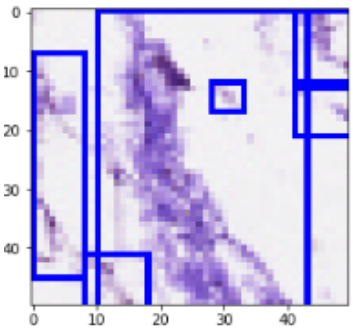


```
from sklearn.model_selection import train_test_split
train,test=train_test_split(df,train_size=0.95,random_state=0)
train_new,valid=train_test_split(train,train_size=0.90,random_state=0)

print(f"train set shape:{train_new.shape}")
print(f"test set shape:{test.shape}")
print(f"validation set shape:{valid.shape}")
```

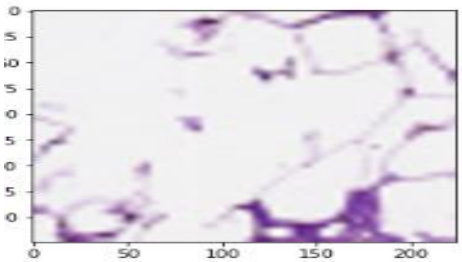
```
train set shape:(968, 2)
test set shape:(57, 2)
validation set shape:(108, 2)
```

```
for cnt in contours:
    x, y, w, h = cv2.boundingRect(cnt)
    # bounding the images
    if y < 50:
        table_image = cv2.rectangle(table_image, (x, y), (x + w, y + h), (0, 0, 255), 1)
```

```
#This is our last step. Here we use the method namedWindow to render our table with the extracted
#content and contours embedded on it. Below is the code snippet:

plt.imshow(table_image)
plt.show()
```




```
# Print image
import matplotlib.pyplot as plt
plt.imshow(input_data[0])
plt.show()

pred=loaded_model.predict(input_data)
if pred>=0.5:
    print("Malignant")
else:
    print("Benign")
```



Github link for the project:
https://github.com/Neha-M333/Breast-Cancer-Classification