NITTE EDUCATION TRUST | NMAM INSTITUTE OF TECHNOLOGY

(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC

MEDICAL IMAGE ANALYSIS FOR CLASSIFICATION OF TUMORS AND SUBTYPES

Project Report submitted by

Kaushik Agrawal	4NM17CS082
Leston John Alva	4NM17CS092
Marlon Fernandes	4NM17CS102

UNDER THE GUIDANCE OF

Mrs. Shwetha G K
Assistant Professor Gd. II

In partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering in
Computer Science and Engineering

from

Visvesvaraya Technological University

Department of Computer Science and Engineering, NMAM Institute of Technology,
Nitte - 574110
(An Autonomous Institution affiliated to
VTU Belgaum)

June 2021

CERTIFICATE

Certified that the project work entitled

"Medical Image Analysis for Classification of Tumors and Subtypes"

is a bonafide work carried out by

Kaushik Agrawal	***Leston John Alva***	***Marlon Fernandes***
(4NM17CS082)	***(4NM17CS092)***	***(4NM17CS102)***

in partial fulfilment of the requirements for the award of

Bachelor of Engineering Degree *in* ***Computer Science*** *and* ***Engineering***

prescribed by ***Visvesvaraya Technological University, Belgaum***

during the year 2020-2021.

It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library.

The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the Bachelor of Engineering Degree.

Signature of Lecturer **Signature of HOD** **Signature of Principal**

ACKNOWLEDGEMENT

The satisfaction that accompanies the completion of any task would be incomplete without the mention of all the people, without whom this endeavour would have been a difficult one to achieve. Their constant blessings, encouragement, guidance and suggestions have been a constant source of inspiration

First and foremost, my gratitude to my project guide **Mrs. Shwetha G K**, Assistant Professor Gd. II, Dept. of CSE, NMAMIT, Nitte for the constant guidance throughout the course of this Project and for the valuable suggestions.

I also take this opportunity to express a deep sense of gratitude to the project coordinators for their valuable guidance and support.

My thanks to our beloved HOD, **Dr. Jyothi Shetty,** for extending support in carrying out this project in the department and providing us with all necessary facilities.

My sincere thanks to our beloved principal, **Dr. Niranjan N Chiplunkar** for permitting us to carry out this project at our college and providing us with all needed facilities.

Finally, thanks to staff members of the Department of Computer Science and Engineering and our friends for their honest opinions and suggestions throughout the course of our Project.

Kaushik Agrawal (4NM17CS082)
Leston John Alva (4NM17CS092)
Marlon Fernandes (4NM17CS102)

ABSTRACT

A tumor is a mass or lump of tissue that may resemble swelling. Not all tumors are cancerous but it is a good idea to consult a doctor if one appears. We mainly deal here with the concept of breast cancer. The method initially is preprocessing images using color deconvolution to highlight stained objects.

Tumors when cancerous cause threatening organ tumors or cancer. Tumors can vary in size from a tiny nodule to a large mass, depending on the type, and they can appear almost anywhere on the body. Not all tumors resemble the cancerous properties but it is always a good option to consult a doctor if any swelling or tumor appears in a person's body.

In this project we will mainly concentrate on the types of breast cancer. The most common types of breast cancer are: Ductal Carcinoma In Situ (DCIS), Invasive Ductal Carcinoma (IDC), Inflammatory Breast Cancer (IBC), Metastatic Breast Cancer (MBC). Here we will mainly concentrate on Invasive Ductal Carcinoma (IDC) for our project.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

A tumor is a lump of tissue that may resemble swelling. A tumor develops when cells reproduce too quickly. Tumors can vary in size from a tiny nodule to a large mass, depending on the type, and they can appear almost anywhere on the body. Not all tumors resemble the cancerous properties but it is always a good option to consult a doctor if any swelling or tumor appears in a person's body. There are mainly three types of tumors. They are Benign, Premalignant and Malignant**.** In this project we will mainly concentrate on the types of breast cancer. The most common types of breast cancer are:

- Ductal Carcinoma In Situ(DCIS)
- Invasive Ductal Carcinoma(IDC)
- Inflammatory Breast Cancer(IBC)
- Metastatic Breast Cancer(MBC)

Here we will mainly concentrate on Invasive Ductal Carcinoma(IDC) for our project.

1.1 TYPES

1) **Ductal Carcinoma In Situ(DCIS):**

   Ductal carcinoma in situ (DCIS) is the presence of abnormal cells inside a milk duct in the breast. DCIS is considered the earliest form of breast cancer. DCIS is noninvasive, meaning it hasn't spread out of the milk duct and has a low risk of becoming invasive. DCIS is usually found during a mammogram done as part of breast cancer screening or to investigate a breast lump. While DCIS isn't an emergency, it does require an evaluation and a consideration of treatment options. Treatment may include breast-conserving surgery combined with radiation or surgery to remove all of the breast tissue. A clinical trial studying active monitoring as an alternative to surgery may be another option.

2) **Inflammatory Breast Cancer:**

Inflammatory breast cancer is a rare type of breast cancer that develops rapidly, making the affected breast red, swollen and tender. Inflammatory breast cancer occurs when cancer cells block the lymphatic vessels in skin covering the breast, causing the characteristic red, swollen appearance of the breast. Inflammatory breast cancer is considered a locally advanced cancer meaning it has spread from its point of origin to nearby tissue and possibly to nearby lymph nodes. Inflammatory breast cancer can easily be confused with a breast infection, which is a much more common cause of breast redness and swelling. Seek medical attention promptly if you notice skin changes on your breast.

3) **Metastatic Breast Cancer(MBC):**

Metastatic breast cancer refers to cancer that originates in the breast but has spread to one or more other locations in the body, such as the lungs, liver, or bones. The majority of patients with MBC have experienced a recurrence following a prior diagnosis of earlier-stage disease; however, patients can also be diagnosed initially with this advanced form of breast cancer. Although the majority of patients diagnosed with MBC have not historically been considered to have curable disease, more-advanced precision treatment options that are tailored to an individual cancer's characteristics continue to be developed and are improving long-term survival rates and quality of life. Importantly, extensive research is currently underway, with the promise of even greater individualized treatment options in the future for patients with MBC.

1.2 INVASIVE DUCTAL CARCINOMA

Invasive Ductal Carcinoma (IDC) is the most common subtype of all breast cancers. To assign an aggressiveness grade to a whole mount sample, pathologists typically focus on the regions which contain the IDC. As a result, one of the common pre-processing steps for automatic aggressiveness grading is to delineate the exact regions of IDC inside of a whole mount slide.

1.2.1 SYMPTOMS

Invasive Ductal Carcinoma symptoms include:

- A lump in your breast
- Thickened breast skin
- Rash or redness on your breast
- Swelling in your breast
- New pain in your breast
- Dimpling on your breast or the skin of your nipple
- Nipple pain
- Inverted nipple
- Nipple discharge
- Lumps under your arm
- Changes to your breast or nipple that are different from the ones you have with your period

1.2.2 DIAGNOSIS

IDC is usually found as the result of an abnormal mammogram. To diagnose cancer, you'll get a biopsy to collect cells for analysis. The doctor will remove a bit of tissue to look at under a microscope. They can make a diagnosis from the biopsy results. If the biopsy confirms you have cancer, you'll likely have more tests to see how large the tumor is and if it has spread:

- **CT scan:** It's a powerful X-ray that makes detailed pictures inside your body.
- **MRI:** It uses strong magnets and radio waves to make pictures of the breast and other structures inside your body.
- **Bone scan:** The doctor injects a tracer into your arm. They take pictures to find out if cancer has traveled to your bones.
- **Chest X-ray:** It uses low doses of radiation to make pictures of the inside of your chest.

- **Fine Needle Aspiration (FNA) Biopsy:** A Fine Needle Aspiration (FNA) Biopsy is a simple procedure that involves passing a thin needle through the skin to sample fluid or tissue from a cyst or solid mass. The sample of cellular material taken during an FNA is then sent to a pathology laboratory for analysis.

1.2.3 TREATMENT

Most women with IDC have surgery to remove the cancer. The treatment options are usually:

- **Lumpectomy:** The surgeon only removes the tumor and a bit of the tissue around it to help make sure all the cancer cells have been removed. You might hear it called breast-conserving surgery.
- **Mastectomy:** The surgeon removes an entire breast.

Which one you get depends on the size of your tumor and how much it has spread throughout your breast and surrounding lymph nodes.

In addition to surgery, other treatments may include:

- **Radiation:** This usually follows your surgery.
- **Hormone therapy:** You'll get it if your cancer is hormone receptor-positive (meaning estrogen helps it grow). These drugs block or lower the amount of estrogen in your body.
- **Chemotherapy:** These medications target cancer cells throughout your body. Doctors may also use It before surgery to shrink tumors and after to kill any cancer cells left behind.
- **Targeted therapy:** These medications block cancer cell growth. You might get them along with chemotherapy.

CHAPTER 2

LITERATURE SURVEY

The performance of automatic tumor tissue classification using traditional machine learning methods greatly depends on the choice of meaningful descriptive features derived from the tissue images. Identification of such features requires domain-specific expert knowledge and is not a straightforward process. This paper studies the feasibility of using deep learning for automatic tumor tissue classification by presenting a deep convolutional neural network (CNN) that consists of multiple hidden layers with convolutional, max-pooling and fully connected layers. Compared to traditional machine learning methods, the proposed CNN is able to automatically learn the layers of features to ameliorate the difficulties of prescribing features for the image classification problems. Several practical ways to tune the parameters for the CNN are also discussed, which will be helpful for users who are keen to adopt a similar approach to their own applications. The performance of our CNN is compared to two well-known machine learning methods – the support vector machine (SVM) and the extreme learning machine (ELM). The comparison shows that our approach achieves significant improvement in terms of classification accuracy. More importantly, the proposed approach is general and can be applied to other biomedical or biological datasets.

2.1 EXISTING SYSTEMS

In the case of IDC analysis and prediction, there are numerous ways to solve the problem. Each method has its own advantages and disadvantages.

Some of the methods or the models used for the prediction and analysis are Convolution Neural Network(CNN), Support Vector Machine(SVM), Extreme Learning Machine(ELM), ResNet50 etc.

1) **Convolution Neural Network(CNN)**

   Currently, CNNs are the most researched machine learning algorithms in

medical image analysis. The reason for this is that CNNs preserve spatial relationships when filtering input images. As mentioned, spatial relationships are of crucial importance in radiology, for example, in how the edge of a bone joins with muscle, or where normal lung tissue interfaces with cancerous tissue. A CNN takes an input image of raw pixels, and transforms it via Convolutional Layers, Rectified Linear Unit (ReLU) Layers and Pooling Layers. This feeds into a final Fully Connected Layer which assigns class scores or probabilities, thus classifying the input into the class with the highest probability.

2) RestNet50

ResNet makes it possible to train up to hundreds or even thousands of layers and still achieves compelling performance. Taking advantage of its powerful representational ability, the performance of many computer vision applications other than image classification have been boosted, such as object detection and face recognition. Since ResNet50 blew people's minds in 2015, many in the research community have dived into the secrets of its success, and many refinements have been made in the architecture.

3) Support Vector Machine(SVM)

Support Vector Machine (SVM) was first heard in 1992, introduced by Boser, Guyon, and Vapnik in COLT-92. Support vector machines (SVMs) are a set of related supervised learning methods used for classification and regression. They belong to a family of generalized linear classifiers. In other terms, Support Vector Machine (SVM) is a classification and regression prediction tool that uses machine learning theory to maximize predictive accuracy while automatically avoiding over-fit to the data. Support Vector machines can be defined as systems which use the hypothesis space of a linear function in a high dimensional feature space, trained with a learning algorithm from optimization theory that implements a learning bias derived from statistical learning theory.

4) Extreme Learning Machine(ELM)

ELM (Extreme Learning Machines) are feedforward neural networks. Extreme learning machine (ELM), as a new learning framework, draws increasing attention in the areas of large-scale computing, high-speed signal processing, artificial intelligence, and so on. ELM aims to break the barriers between the conventional artificial learning techniques and biological learning mechanism and represents a suite of machine learning techniques in which hidden neurons need not to be tuned. ELM theories and algorithms argue that “random hidden neurons” capture the essence of some brain learning mechanisms as well as the intuitive sense that the efficiency of brain learning need not rely on computing power of neurons.

Comparing the Existing Models from the survey papers we found the below accuracies:
CNN (around 93-95%), RestNet(around 92%), SVM(84-85%) and ELM(85-87%). Henceforth, we go with the CNN method and propose a model for CNN.

2.2 PROPOSED SYSTEM

We have used convolution neural networks using layers, models etc., with Keras interface to solve the problem of Medical Image Analysis for Classification of Tumors and Subtypes.

Our data set consists of over 90000 images of breast histopathology of different shapes and sizes in a different environment. We import the images and remove any imperfect images that might affect the model. We equalise the dimensions and maintain a constant colour space among the images. Later the images will be standardised in terms of lighting by equalising their histograms.

The training data will be passed to our CNN model having multiple convolutions and max-pooling layers and it will be run over multiple epochs and the constraints are modified till we get consistent and satisfactory results.

We aim to create a model with over 94% accuracy and a prototype light enough to run on systems with low computing power.

CHAPTER 3

PROBLEM DEFINITION

3.1 PROBLEM STATEMENT

To develop a model, that would detect IDC in a fast and lightweight system that is capable of real time classification of IDC tumors

3.2 PROBLEM DEFINITION

According to the National Cancer Registry Programme of the India Council of Medical Research (ICMR), more than 1300 Indians die every day due to cancer. Between 2012 and 2014, the mortality rate due to cancer increased by approximately 6%. In 2012, there were 478,180 deaths out of 2,934,314 cases reported.

This situation can be improved if cancer is diagnosed in the early stage. The main aim of the project is to find an optimal way through which we can diagnose tumors using machine learning algorithms.

CHAPTER 4

SYSTEM REQUIREMENTS SPECIFICATION

System relies on the training data that is provided to the machine learning models and can be the subject of interest for multiple samples provided in testing the model.

4.1 HARDWARE REQUIREMENTS

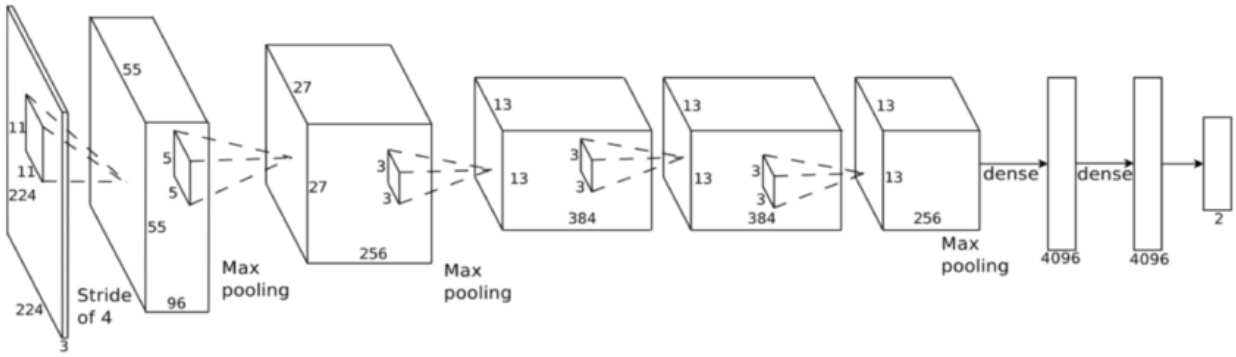
- Kaggle Notebook with GPU acceleration and Network enabled.
- PC or Laptop with
    - Intel i5 or Ryzen 5 1600 or higher.
    - Nvidia GTX or RTX or Radeon series graphics card.
    - 5GB of storage space.
    - 8GB or more RAM

4.2 SOFTWARE REQUIREMENTS

- Language: Python 3.6
- Platform: Anaconda or Kaggle
- Notebook: Jupyter
- Libraries:
    - Keras
    - TensorFlow
    - NumPy
    - Pandas
    - Sklearn

CHAPTER 5

SYSTEM DESIGN



CNN is a machine learning algorithm that can take an input image, assign importance to various aspects or objects in the image, and be able to differentiate one from another. CNN works by extracting the features from the images.

Any CNN consists of the following:

1. The input layer is an image.

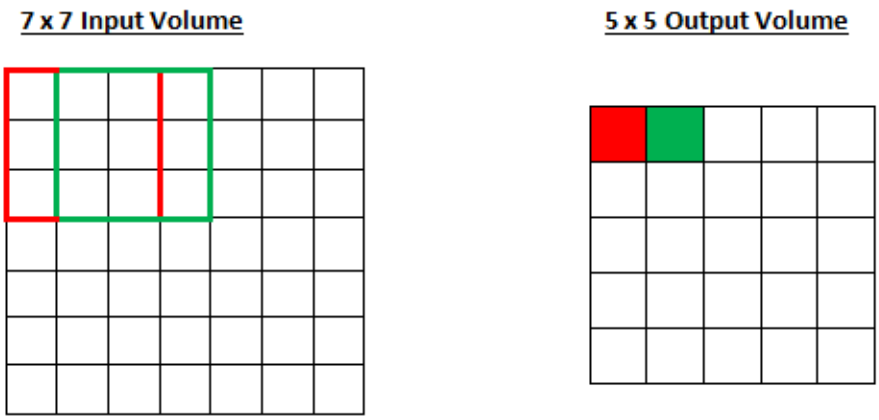
2. The output layer is a binary or multi-class label.

3. Hidden layers consist of convolution layers, ReLU (Rectified Linear Unit) layers, the pooling layer, and a fully connected Neural Network.

Convolution Layer: Our model uses 2D Convolution Layer, this layer creates a convolution kernel that is wind with layers input which helps produce a tensor of outputs.Some of the parameters of Conv2D used -

- **filters:** Integer, the dimensionality of the output space (i.e. the number of output filters in the convolution).
- **kernel_size:** An integer or tuple/list of 2 integers, specifying the height and width of the 2D convolution window. Can be a single integer to specify the same value for all spatial dimensions.

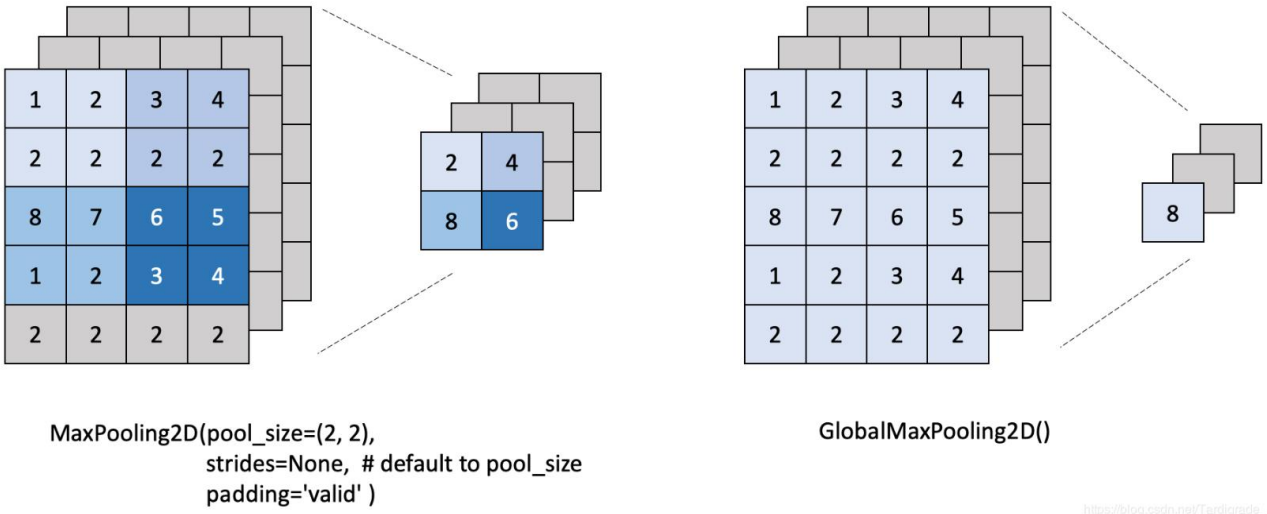
- **Kernel Initializer:** Kernel Initializer are used to statistically initialise the weights in the model. This generates the weights and distributes them as starting weights. The kernel_initializer used in our model is ‘he_uniform’, this draws samples from a normal distribution.
- **Padding:** Padding is simply a process of adding layers of zeros to our input images so as to avoid the problems mentioned above. This prevents shrinking as, if p = number of layers of zeros added to the border of the image, then our (n x n) image becomes (n + 2p) x (n + 2p) image after padding. "valid" means no padding. "same" results in padding with zeros evenly to the left/right or up/down of the input such that output has the same height/width dimension as the input.
- **Stride:** Stride denotes how many steps we are moving in each step in convolution.By default it is one. An integer or tuple/list of 2 integers, specifying the strides of the convolution along the height and width. Can be a single integer to specify the same value for all spatial dimensions.



- **Activation function:** Activation functions are computational functions for neuron computation and interaction. They are functions that engage each neuron cell in active learning of patterns between input data and its corresponding target data.

Max Pooling Layer: Pooling layer is used to reduce the number of neurons necessary in subsequent layers. There are two most common functions used in

Pooling operation (a) Average Pooling and (b) Maximum Pooling (or Max Pooling), in our model we used Max pooling layer. The max pooling layer picks up the maximum intensity value of the image from a part of the Convolution layer.



MaxPooling2D(pool_size=(2, 2),
strides=None, # default to pool_size
padding='valid')

GlobalMaxPooling2D()

Batch Normalisation: Batch Normalization is a technique that is designed to automatically standardize the inputs to a layer in a deep learning neural network. It applies a transformation that maintains the mean output close to 0 and the output standard deviation close to 1.

Dense Layer: The dense layer is a neural network layer that is connected deeply, which means each neuron in the dense layer receives input from all neurons of its previous layer. It is found to be the most commonly used layer in the models.The output generated by the dense layer is an ‘m’ dimensional vector. Thus, a dense layer is basically used for changing the dimensions of the vector. Dense layers also apply operations like rotation, scaling, translation on the vector.

Flatten: A flatten operation on a tensor reshapes the tensor to have the shape that is equal to the number of elements contained in tensor non including the batch dimension.

5.1 DESIGN FLOW

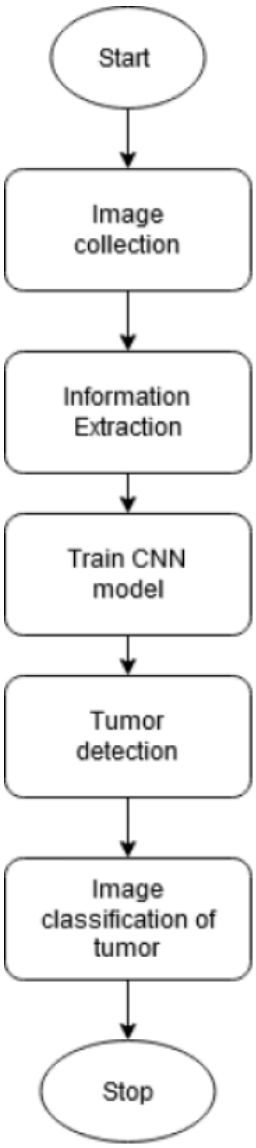
Image Collection: This is the first step that is composed of hardware, and its main function is to collect the images.

Information Extraction: The information from observing the image is extracted. This information is used by the medical experts to treat the tumor of the patient.

Train CNN Model: The CNN model that is being used has to be trained, this is done by preparing the training and testing data, CNN layers are built using Tensorflow library, Optimizer is selected, then network is trained and checkpoints are saved and finally the model is tested.

Tumor Detection: After collection of images, training CNN Model, and classification of the images the last and most important part of the process is to detect the tumor and it's type.

Image Classification of tumor: Image classification is used in Convolutional neural networks. The main task of image classification is acceptance of the input image and the following definition of its class.



CHAPTER 6

IMPLEMENTATION

6.1 Importing the required libraries:

```
import pandas as pd
import numpy as np
import keras
import matplotlib.pyplot as plt
import os

from glob import glob
from keras.preprocessing.image import load_img, img_to_array
```

6.2 Data Extraction:

Data is extracted from the histopathology dataset which is a Kaggle dataset.

```
files = glob('/kaggle/input/breast-histopathology-images/*/*/*')
```

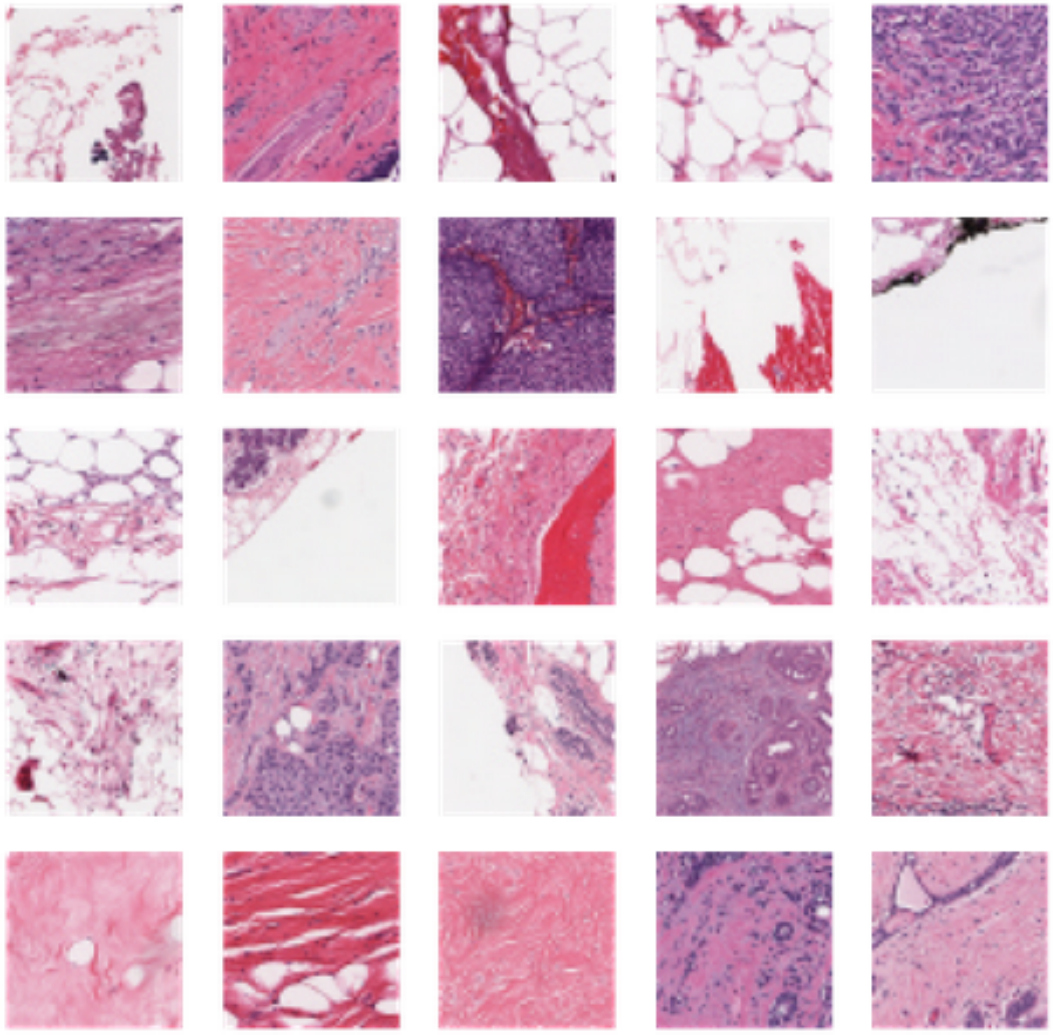
6.3 Sample Image Function:

```
def sample_img(files):
    plt.figure(figsize= (10,10))
    ind = np.random.randint(0, len(files), 25)
    i=0
    for loc in ind:
        plt.subplot(5,5,i+1)
        sample = load_img(files[loc], target_size=(150,150))
        sample = img_to_array(sample)
        plt.axis("off")
        plt.imshow(sample.astype("uint8"))
        i+=1
```

Here we will be loading the images which are present in our dataset.

6.4 Sample Image Generated:

```
sample_img(files)
```



The images generated here are of the above type. These images are mainly the breast histopathology images which later need to undergo the required processing to give the best accuracy and performance factors.

6.5 Train Data:

The below function loads data for adding train and test data.

```
def load_balanced_data(files, size, start_index):
    half_size = int(size/2)
    count=0
    res = []
    y = []
    for file in files[start_index:]:
        if (count!=half_size):
            if file[-5] == '1' and file.endswith(".png"):
                img = load_img(file, target_size = (50,50))
                pixels = img_to_array(img)
                pixels /= 255
                res.append(pixels)
                y.append(1)
                count += 1

    for file in files[start_index:]:
        if(count!=0):
            if(file[-5] == '0'):
                img = load_img(file, target_size = (50,50))
                pixels = img_to_array(img)
                pixels /= 255
                res.append(pixels)
                y.append(0)
                count -= 1
    return np.stack(res), y
```

Here we will be creating the training data which is later processed with the test data.

```
x_train, y_train = load_balanced_data(files, 90000,0)
```

6.6 Test Data:

```
x_test, y_test =load_balanced_data(files, 35000,90000)
```

Here we will generate the test dataset which undergoes processing with the training data.

6.7 Importing Keras Model:

```
from keras.models import Model
from keras.layers import Input, Dense, Dropout, Conv2D, Activation, MaxPooling2D, Flatten, BatchNormalization
```

Here the keras models and layers are being imported as standard libraries. The layers include such as in[ut, dense, flatten, activation, maxpooling etc.

6.8 Model Definition:

```
def model_def():

    model = Sequential([
        Conv2D(32, (3,3), kernel_initializer='he_uniform',input_shape = (50,50,3), padding="same", activation='relu'),
        MaxPooling2D(pool_size= (2,2)),
        BatchNormalization(),
        Conv2D(32, (3,3), kernel_initializer='he_uniform', padding="same", activation='relu'),
        MaxPooling2D(pool_size= (2,2)),
        BatchNormalization(),
        Conv2D(64, (3,3), kernel_initializer='he_uniform', padding="same", activation='relu'),
        MaxPooling2D(pool_size= (2,2)),
        BatchNormalization(),
        Conv2D(128, (3,3), kernel_initializer='he_uniform', padding="same", activation='relu'),
        MaxPooling2D(pool_size= (2,2)),
        Flatten(),
        Dense(128, activation = "relu"),
        #Dense(32, activation = "relu"),
        Dense(1, activation = "sigmoid")
    ])
    model.compile(optimizer = keras.optimizers.SGD(0.01, momentum=0.9), loss="binary_crossentropy", metrics = ['acc'])
    return model
```

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 50, 50, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 25, 25, 32)        0
_________________________________________________________________
batch_normalization (BatchNo (None, 25, 25, 32)        128
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 25, 25, 32)        9248
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 12, 12, 32)        0
_________________________________________________________________
batch_normalization_1 (Batch (None, 12, 12, 32)        128
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 12, 12, 64)        18496
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 6, 6, 64)          0
_________________________________________________________________
batch_normalization_2 (Batch (None, 6, 6, 64)          256
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 6, 6, 128)         73856
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 3, 3, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 1152)              0
_________________________________________________________________
dense (Dense)                (None, 128)               147584
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 129
=================================================================
Total params: 250,721
Trainable params: 250,465
Non-trainable params: 256
```

The model is trained using 4 convolution layers with the use of ReLU activation function along with the flatten and 2 dense layers with the activation function ReLU. The loss function used here is binary cross entropy. We have used the above model since it gives the highest accuracy and more smooth variation and loss function curve compared to other models.

6.9 Training the Model:

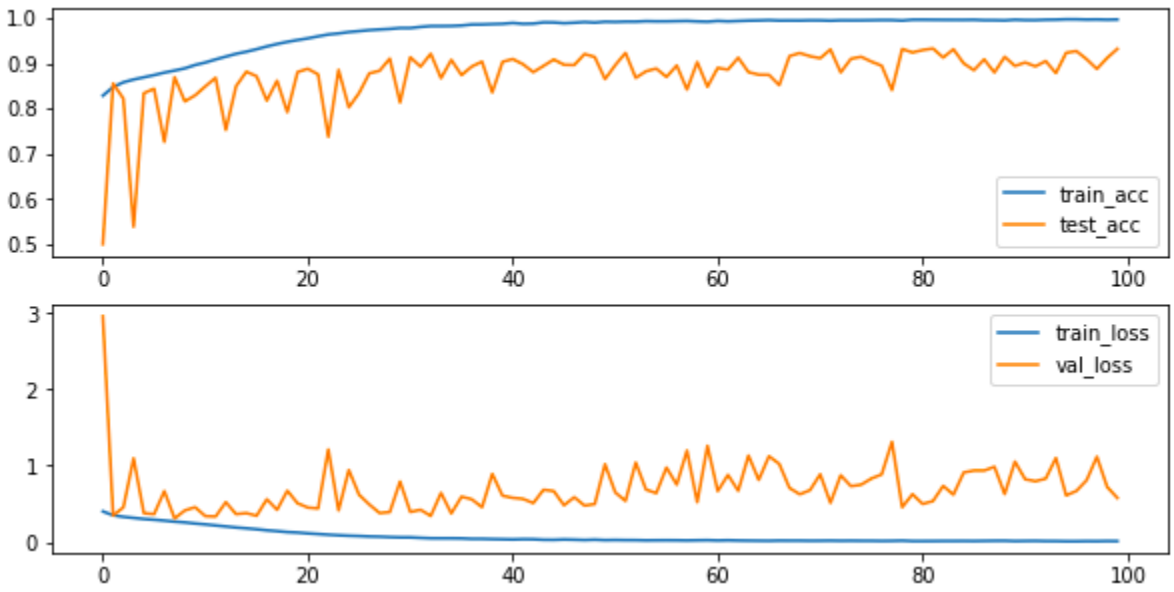
```
history = model.fit(x_train, y_train, validation_data=(x_test, y_test), epochs = 100, batch_size=32)
```

```
Epoch 1/100
2813/2813 [==============================] - 33s 10ms/step - loss: 0.4417 - acc: 0.8111 - val_loss: 2.9593 - val_acc: 0.5003
Epoch 2/100
2813/2813 [==============================] - 27s 10ms/step - loss: 0.3564 - acc: 0.8437 - val_loss: 0.3467 - val_acc: 0.8553
Epoch 3/100
2813/2813 [==============================] - 27s 10ms/step - loss: 0.3315 - acc: 0.8557 - val_loss: 0.4523 - val_acc: 0.8221
Epoch 4/100
2813/2813 [==============================] - 27s 10ms/step - loss: 0.3178 - acc: 0.8624 - val_loss: 1.0967 - val_acc: 0.5388
Epoch 5/100
2813/2813 [==============================] - 27s 10ms/step - loss: 0.2986 - acc: 0.8707 - val_loss: 0.3772 - val_acc: 0.8339
Epoch 6/100
2813/2813 [==============================] - 27s 10ms/step - loss: 0.2880 - acc: 0.8742 - val_loss: 0.3641 - val_acc: 0.8438
Epoch 7/100
2813/2813 [==============================] - 27s 10ms/step - loss: 0.2788 - acc: 0.8793 - val_loss: 0.6661 - val_acc: 0.7263
```

```
Epoch 94/100
2813/2813 [==============================] - 27s 10ms/step - loss: 0.0101 - acc: 0.9969 - val_loss: 1.1003 - val_acc: 0.8785
Epoch 95/100
2813/2813 [==============================] - 27s 10ms/step - loss: 0.0082 - acc: 0.9974 - val_loss: 0.6093 - val_acc: 0.9232
Epoch 96/100
2813/2813 [==============================] - 27s 10ms/step - loss: 0.0071 - acc: 0.9975 - val_loss: 0.6683 - val_acc: 0.9269
Epoch 97/100
2813/2813 [==============================] - 27s 10ms/step - loss: 0.0093 - acc: 0.9967 - val_loss: 0.8098 - val_acc: 0.9090
Epoch 98/100
2813/2813 [==============================] - 27s 10ms/step - loss: 0.0095 - acc: 0.9968 - val_loss: 1.1176 - val_acc: 0.8876
Epoch 99/100
2813/2813 [==============================] - 27s 10ms/step - loss: 0.0101 - acc: 0.9966 - val_loss: 0.7204 - val_acc: 0.9118
Epoch 100/100
2813/2813 [==============================] - 27s 10ms/step - loss: 0.0099 - acc: 0.9970 - val_loss: 0.5755 - val_acc: 0.9319
```

6.10 Plotting Variation and Loss Graph:

```
plt.figure(figsize = (12,6))
plt.subplot(2,1,1)
plt.plot(history.history['acc'], label="train_acc")
plt.plot(history.history['val_acc'], label = "test_acc")
plt.legend()
plt.subplot(2,1,2)
plt.plot(history.history['loss'], label = "train_loss")
plt.plot(history.history['val_loss'], label = "val_loss")
plt.legend()
```



The above graph shows the validation and loss curve for the above processed model.

6.11 Classification Report:

```
print(classification_report(y_test, res))
```

```
              precision    recall  f1-score   support

           0       0.99      0.88      0.93     17500
           1       0.89      0.99      0.94     17500

    accuracy                           0.93     35000
   macro avg       0.94      0.93      0.93     35000
weighted avg       0.94      0.93      0.93     35000
```

6.12 Confusion Matrix:

```
print(confusion_matrix(y_test, res))
```

```
[[15328  2172]
 [  211 17289]]
```

6.13 Saving the Model:

```
model.save('model.h5')  # creates a HDF5 file 'my_model.h5'
```

CHAPTER 7

RESULTS AND CONCLUSIONS

7.1 RESULTS

1. The CNN model was created, trained and further refined to achieve the theoretical accuracy as per the intended objective.
2. All the objectives set for the model have been satisfied within the stipulated time with minimal errors.
3. The practical use of the model yielded desirable results usable in real life scenarios with minor modifications.
4. Given ideal conditions, the practical application of the model provided optimal outputs, and has been recorded to be highly accurate.
5. The model was successfully trained for future use.

7.1.1 COMPARISON WITH OTHER MODELS

The dataset used to train the models contains images of size 50x50

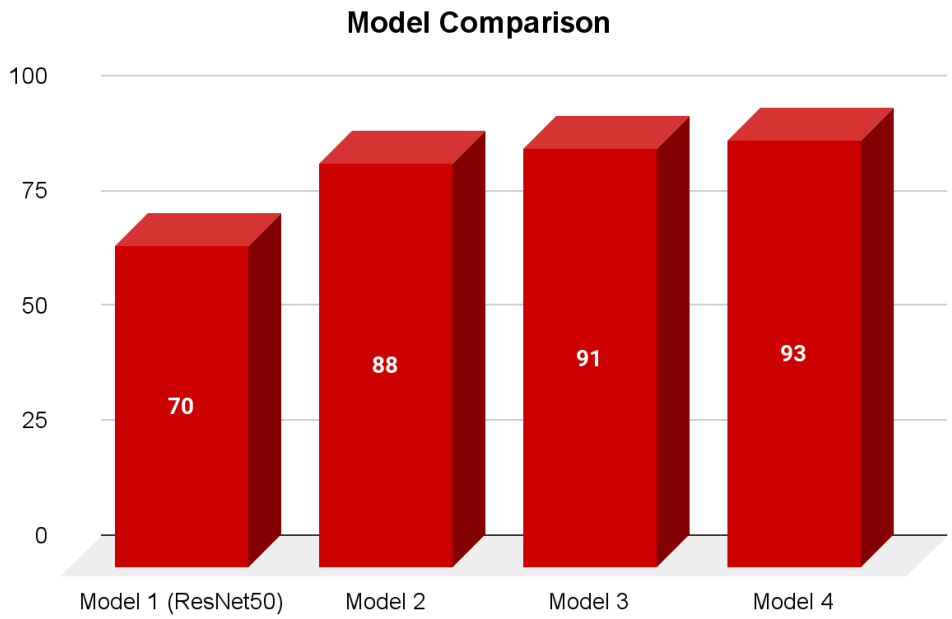
Model 1 is the ResNet50 which is a pre-trained model we have used. It also has a Flatten layer and a dense layer with **sigmoid** as activation function. The optimizer used for this model is **RMSprop** with learning rate of the learning rate of 0.01 and a loss function **binary cross entropy**. The accuracy obtained by the model is **70%**.

Model 2 is a CNN model with 5 convolution layers with ReLU(Rectified Linear Unit) as the activation function. It has a flatten layer and 2 dense layers with **ReLU** and one dense layer with **sigmoid** as activation function. The optimizer used for this model is **Adam** (Adaptive Moment Estimation) with a learning rate of 0.01 and loss function **binary cross entropy**. The accuracy obtained for this model is **88%**.

Model 3 is a CNN model with 4 convolution layers in which the activation function used for the layer is '**ReLU'**(Rectified Linear Unit). It also consists of one flatten layer along with the 3 dense layers in which the first two dense layers use '**ReLU'** activation function and the third layer uses an activation function called **sigmoid**.

The optimizer used for the model is **SGD**(Stochastic Gradient Descent). The loss function used for the model is **binary cross entropy**. This model gave an accuracy of **91%**.

Model 4 is the model which has shown the most accuracy compared to the previous models, it has four convolution layers arranged in sequential manner. The activation function used in all the layers is **ReLU**. The model uses two dense layers; the first dense layer uses **'ReLU'** activation function whereas the second dense layer uses 'sigmoid' activation function. The optimizer used for the model is **SGD**(Stochastic Gradient Descent). This model gave an accuracy of **93%**.



7.2 CONCLUSION

The idea of application of convolution neural networks to solve the problem pertaining to IDC classification has proven to provide a feasible solution, and the results obtained on deploying this model have been satisfactory. The model can be improved further by adding more images to the dataset or using a different dataset for IDC.

Over time, this technique can be refined and can be used in the real time scenario. The model designed with necessary improvements with new technology can be used in the field of Medicine. The system can be optimized to classify tumors based on the new dataset which can perform classification of different types of breast cancer tumors.

CHAPTER 8

FUTURE WORK

- A GUI interface can be created to detect IDC from the input image. Camera functionality to scan images can be used as input for predicting IDC in the model. We can create an application or a website where the users can scan images and get the predicted output. More Functionalities can be added like cloud capabilities to make it accessible anytime and anywhere.
- The model can always be improved further by training the model with a new dataset for IDC from another source.
- We can always find more ways to improve the accuracy using a different model architecture. Also the model could be tested by adding more convolution layers and dense layers.

REFERENCES

1. **“Deep Learning Applications in Medical Image Analysis” , 2017.**
   Justin Ker, Lipo Wang, Jai Rao, And Tchoyoson Lim,
   Available:https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8241753
2. **“A deep learning approach for tumor tissue image classification”**
   X. Yanget al., inProc. Int. Conf. Biomed. Eng., Calgary, AB, Canada, 2016.[Online].
   Available: http://dx.doi.org/10.2316/P.2016.832-025
3. **“Cell Nuclei Segmentation in Cytological Images Using Convolutional Neural Network and Seeded Watershed Algorithm”**
   Marek Kowal, Michał Żejmo, Marcin Skobel, Józef Korbicz, Roman Monczak
   Available:https://link.springer.com/content/pdf/10.1007/s10278-019-00200-8.pdf
4. **Convolutional neural network-based models for diagnosis of breast cancer**
   Mehedi Masud, Amr E. Eldin Rashed, M. Shamim Hossain
   Available:https://link.springer.com/content/pdf/10.1007/s00521-020-05394-5.pdf
5. **Medical image analysis based on deep learning approach**
   Muralikrishna Puttagunta & S. Ravi,
   Available:https://link.springer.com/content/pdf/10.1007/s11042-021-10707-4.pdf
6. **Classification of Breast Cancer Histology Images Using Multi-Size and Discriminative Patches Based on Deep Learning**
   Yuqian Li, Junmin Wu, and Qisong Wu, Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8636921
7. **Deep Learning Applied for Histological Diagnosis of Breast Cancer**
   Yasin Yari , Thuy V. Nguyen, And Hieu T. Nguyen
   Available: https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9186080