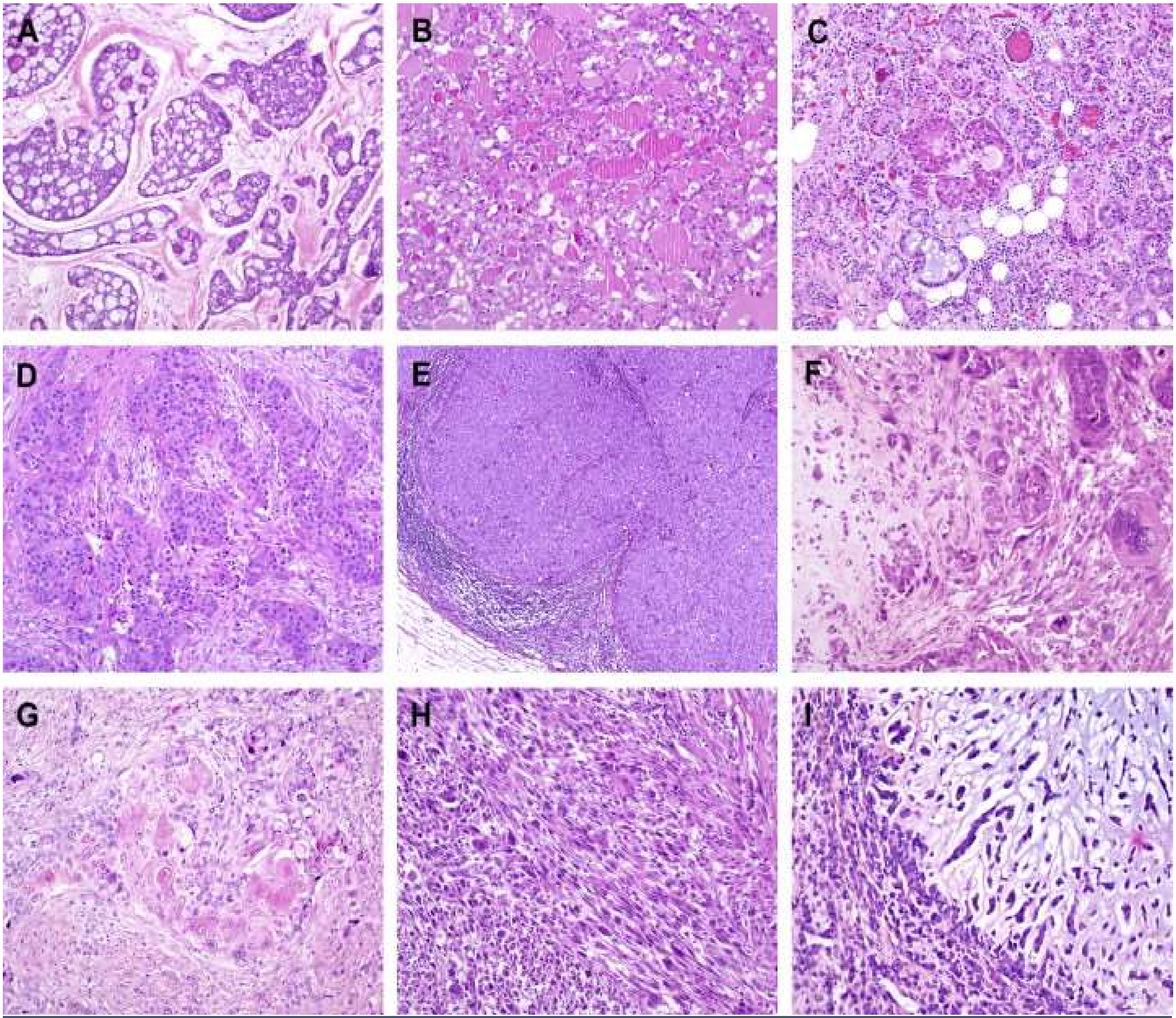

Breast Cancer Classification

Classification of Tumorous Microscopic Images of Breast Cancer Using Deep Learning

Name [Date] [Course title]

Breast Cancer Images Classification

Problem Introduction:

Breast cancer is a disease seen mainly in women and is seen as a major cause of death among women. In 2018, the total deaths due to breast cancer in women was seen to be **627,000** out of **2.1 million** cases which were diagnosed. **Invasive Ductual Carcinoma (IDC)** in diagnosing breast cancer, since its subsequent digitalization is more feasible due to advancements in slide scanning technology, as well as the reduction of storage load cost in recent years. The digitalized approaches in deep learning has aided a lot in diagnosing and controlling breast cancer, with power to pre-identify the disease via deep learning methods.

Dataset Information:

The breast cancer histopathological image dataset (BreakHis)
contains **9109** microscopic images of breast tumor tissues collected from **82** patients, regarded as malignant or benign. The tissues are magnified at different scaling factors (**40X**, **100X**, **200X**, **400X**). In this dataset, it contains **2480** malignant and **5429** benign tumors. The images of tissues are taken to be **700 X 460 pixels**, **3-channel RGB**, **8**-bit depth in each format, and in **PNG**. It is believed that this dataset can become a benchmark for future classifications of breast cancer classification. We have made a model such that given an image, the model classifies the image in one of the two cancerous tumors (malignant or benign). The benign tumor is less dangerous, and it can be controlled by taking safety measures. In other case, the malignant tumor is a symbol of danger, it has a greater value of being affected the breast and caused cancer.

Summarizing all things, we have a dataset of almost **4GB** in size, having following characteristics:

- Images of breast tissues taken from microscope
- A total of **9109** images of tissues, regarded as malignant or benign
- All tissues are magnified at different scales: **40X**, **100X**, **200X**, and **400X**
- Each image having a resolution of **700** by **460**
- All images are colored (RGB – 3 channels)
- Since each image is magnified at different magnification scales, so we have a greater number of images. The total images after multiplying each image with its magnifications are **39,545**

Data Splits:

We have split the original data into 3 different sets, train, validation and test. For training purpose, we have used **25, 236** images, for validation are **6309** (**20%** of original data), and **8000** images are kept for testing purpose.

Background Study:

Machine Learning or simply ML is a field of study that aims to make machines learn through data without being programmed explicitly, and to make decisions on unseen data in an intelligent manner. It is understood as a subset of **Artificial Intelligence (AI)**, which aims to make machines behave intelligently like humans. The area of ML is divided into 3 parts, as given next:

- **Supervised Learning**
  - Type of machine learning that works on labelled data, and makes a function that maps the input to the given output
- **Unsupervised Learning**
  - Type of machine learning that works on unlabeled data to draw useful patterns from it and generate decisions
- **Reinforcement Learning**
  - Type of machine learning that aims to train an agent in an interactive environment to achieve its goal

Deep Learning:

Deep Learning is another class of machine learning that aims to solve tasks by using those algorithms that are inspired by the working of human brain. Deep Learning comprises of deep neural networks that work like the functionality of biological neurons in our brains. As in machine learning, deep learning also contains many algorithms, referred to as deep neural networks, used for specific tasks, some them are listed:

- Convolutional Neural Network (CNN)
  - Used for problems concerned with image-related tasks, like image classification, etc.

- Recurrent Neural Network (RNN)
  - Used for language-related tasks like language modelling

- Deep Brief Networks (DBNs), and so on

For our problem, since our problem is concerned with image-related task (signal classification), we will be using a deep convolutional neural network to model our problem.

Implementation Techniques:

For this problem, we have used **Python Programming Language** as a tool. Python provides good libraries for **machine** and **deep learning** that will be used for our dataset, which are listed below:

- Scikit-learn (for machine learning)
- Keras (for deep learning)
- Pandas (for data preprocessing)
- NumPy (for array computing)

Each of these libraries have a lot of built-in functions and methods that can be used for computing with images, tensors and arrays.

Algorithms Used:

Since the problem is concerned with image classification, so deep learning models are used for classification purpose. We have used a pre-trained deep convolutional neural network model from **Keras** library of python, named as **EfficientNetV2** for our problem. **EfficientNetV2** is a deep neural network that is used for image classification, comprising of many deep layers and is pre-trained on **ImageNet Dataset Benchmark**.

Model Evaluation Metrics:

For making a **deep learning model**, we have defined its evaluation and compilation metrics to calculate its results. This process involves some steps, as listed:

- In the first step, the model is compiled using an optimizer, loss function and accuracy metric

- In the second step, the model is fit to the model using the training data, and is validated on the testing data by using a deep learning algorithm

- While fitting the model, the number epochs (iterations) are also defined, which are nothing but the number of steps the model takes to go through the entire training dataset

In our case, we have used following metrics for model compilation:

- **Adam** (Adaptive Moment Estimation) algorithm for model optimization
- **Binary Crossentropy Loss** function as our loss metric, since we are dealing with binary classification problem
- **Keras Accuracy Metric** as our metric for calculating model accuracy on training and validation datasets

For model fitting, following methodology has been used:

- Fitted the data on **25, 236** samples**,** validated on **6309** samples
- Ran model for 10 epochs
- Best accuracy was achieved after running 10 iterations, each iteration took 10-15 minutes on average

Model Results:

After defining the model and fitting the deep learning model to our training data, we got following results. The first image shows a jpg file that describes the model logs at each training step. It consists of training loss, training accuracy, validation loss and validation accuracy:

```
_________________________________________________________________
model built!
Epoch 1/10
579/579 [==============================] - 599s 1s/step - loss: 0.2515 - accuracy: 0.8947 - Precision: 0.8948 - Recall: 0.8959 - val
_loss: 0.0632 - val_accuracy: 0.9799 - val_Precision: 0.9463 - val_Recall: 0.9797
Epoch 2/10
579/579 [==============================] - 574s 991ms/step - loss: 0.0773 - accuracy: 0.9715 - Precision: 0.9722 - Recall: 0.9706 -
val_loss: 0.1264 - val_accuracy: 0.9740 - val_Precision: 0.9157 - val_Recall: 0.9934
Epoch 3/10
579/579 [==============================] - 575s 994ms/step - loss: 0.0603 - accuracy: 0.9788 - Precision: 0.9764 - Recall: 0.9811 -
val_loss: 0.0314 - val_accuracy: 0.9899 - val_Precision: 0.9686 - val_Recall: 0.9940
Epoch 4/10
579/579 [==============================] - 581s 1s/step - loss: 0.0451 - accuracy: 0.9845 - Precision: 0.9861 - Recall: 0.9829 - val
_loss: 0.0773 - val_accuracy: 0.9751 - val_Precision: 0.9170 - val_Recall: 0.9964
Epoch 5/10
579/579 [==============================] - 577s 996ms/step - loss: 0.0414 - accuracy: 0.9865 - Precision: 0.9857 - Recall: 0.9873 -
val_loss: 0.0374 - val_accuracy: 0.9873 - val_Precision: 0.9872 - val_Recall: 0.9648
Epoch 6/10
579/579 [==============================] - 593s 1s/step - loss: 0.0392 - accuracy: 0.9871 - Precision: 0.9879 - Recall: 0.9866 - val
_loss: 0.0189 - val_accuracy: 0.9921 - val_Precision: 0.9903 - val_Recall: 0.9797
Epoch 7/10
579/579 [==============================] - 577s 996ms/step - loss: 0.0362 - accuracy: 0.9882 - Precision: 0.9877 - Recall: 0.9886 -
val_loss: 0.0307 - val_accuracy: 0.9903 - val_Precision: 0.9856 - val_Recall: 0.9779
Epoch 8/10
579/579 [==============================] - 575s 993ms/step - loss: 0.0347 - accuracy: 0.9885 - Precision: 0.9885 - Recall: 0.9884 -
val_loss: 0.0112 - val_accuracy: 0.9960 - val_Precision: 0.9905 - val_Recall: 0.9946
Epoch 9/10
579/579 [==============================] - 570s 984ms/step - loss: 0.0282 - accuracy: 0.9906 - Precision: 0.9908 - Recall: 0.9905 -
val_loss: 0.0401 - val_accuracy: 0.9868 - val_Precision: 0.9871 - val_Recall: 0.9630
Epoch 10/10
579/579 [==============================] - 569s 982ms/step - loss: 0.0331 - accuracy: 0.9886 - Precision: 0.9895 - Recall: 0.9876 -
val_loss: 0.0290 - val_accuracy: 0.9910 - val_Precision: 0.9781 - val_Recall: 0.9881
```

Model Evaluation Graphs:

We have also calculated the graphs for loss per iteration and accuracy per iteration (epoch) on training and validation data. The graphs are obtained as a result of complete model training for 10 iterations, and are shown below.

Firstly, the graph of accuracy training and validation:

Secondly, the graph of loss (binary crossentopy) for training and validation:

We have obtained an accuracy of **98%** on training dataset and the **98%** accuracy is achieved on validation data. Moreover, the loss for training and validation data is calculated as below **0.03**

Predictions on Test Dataset:

Below is an image that shows the performance on test dataset, containing the table of actual and predicted labels:

[196...

	actual	prediction
0	benign	malignant
1	malignant	malignant
2	malignant	malignant
3	benign	malignant
4	malignant	malignant
5	malignant	malignant
6	malignant	malignant
7	benign	malignant
8	benign	malignant
9	benign	malignant
10	malignant	malignant
11	malignant	malignant
12	malignant	malignant
13	malignant	malignant
14	benign	malignant