Name: Taher Barwaniwala **email:** barwaniwalataher6@gmail.com **Mobile number**:+91 8238544856

Assignment: Cancer Cell Detection and Model Preparation

Introduction

In this assignment, the task is to develop a model for cancer cell detection using the provided dataset. The dataset consists of images of skin lesions associated with different types of skin cancer. The goal is to build a model that can accurately classify these images into their respective cancer types. The assignment requires using deep learning and machine learning concepts and tools such as numpy, Anaconda, Google Colab, and Python. The recommended approach is to use Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) or Generative Adversarial Networks (GAN).

Dataset

The dataset for this assignment can be obtained from the following source:

- [Skin Cancer MNIST: HAM10000](https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-ham10000)

The dataset contains images of skin lesions associated with seven different types of skin cancer. Each image is labeled with its corresponding cancer type.

Problem Statement

The task is to develop a model that can accurately classify skin lesion images into their respective cancer types. This involves building a deep learning model using concepts and tools such as CNN, RNN with LSTM, or GAN. The model should be trained on the provided dataset and evaluated using appropriate metrics. The ultimate goal is to achieve a high accuracy in predicting the cancer types based on the input images.

Approach

The recommended approach for solving this problem is to use a CNN-based model. The provided code includes a Python implementation using the Keras library. The model is built using the Xception architecture, which is a deep convolutional neural network that has shown good performance on image classification tasks. The code also includes data preprocessing steps such as data cleaning, exploratory data analysis (EDA), loading and resizing images, normalization, label encoding, and train-test splitting.

The steps involved in the approach are as follows:

1. Importing libraries: Import the necessary libraries for data manipulation, visualization, and model building.

2. Dictionary of images and labels: Create a dictionary mapping image IDs to their file paths and a dictionary mapping cancer types to human-friendly labels.

3. Data processing: Load the metadata of the dataset and create new columns for better readability. Perform data cleaning by filling null values with the mean age. Explore the dataset through EDA to gain insights into the distribution of cancer types, age, gender, and other variables.

4. Loading and resizing images: Load the images and resize them to a consistent size for model input. Visualize samples of images for each cancer type.

5. Train-test split: Split the dataset into training and testing sets for model training and evaluation.

Name: Taher Barwaniwala **email:** barwaniwalataher6@gmail.com **Mobile number**:+91 8238544856

6. Normalization: Normalize the pixel values of the images by subtracting the mean and dividing by the standard deviation.

7. Label encoding: Perform one-hot encoding on the labels to convert them into categorical form.

8. Training and validation split: Further split the training set into training and validation sets for model training and evaluation.

9. Model building: A Convolutional Neural Network (CNN) model was built using the Xception architecture. The base model was initialized with pre-trained weights from ImageNet and frozen to prevent weight updates during training. Additional layers were added to the model for classification.

10. Optimizer and Annealer: Define the optimizer for the model and set up a learning rate annealer to adjust the learning rate during training.

11. Data augmentation: Apply data augmentation techniques to the training data to prevent overfitting.

12. Model fitting: Train the model using the augmented training data and evaluate its performance on the validation set.

13. Model evaluation: Evaluate the model's performance on the testing set using metrics such as accuracy and loss. Plot the model's validation loss and accuracy during training

14.Confusion Matrix: The confusion matrix was plotted to visualize the performance of the model in classifying different cell types.

Results

The model achieved a validation accuracy of 78.9277% and a validation loss of 0.694829. On the test set, the model achieved an accuracy of 79.0315% and a loss of 0.729872.

Conclusion

In this project, a model for cancer cell detection was developed using the Skin Cancer MNIST HAM10000 dataset. The approach involved preprocessing the data, building a CNN model with the Xception architecture, training the model using data augmentation, and evaluating its performance. The model showed promising results in classifying different types of skin lesions. Further improvements can be made by fine-tuning the model, exploring other architectures, or incorporating advanced techniques such as transfer learning.