**Deep Learning-Based Classification of Lung and Colon Cancer**

Team Number 18 - Shivani Shekhawat & Shalini Shree

**Introduction**

**Background:**

Lung and colon cancer are among the most common types of cancer worldwide. Histopathological analysis, which involves the microscopic examination of biological tissues to observe the appearance of diseased cells and tissues in very fine detail, is a crucial tool in diagnosing these cancers. However, the interpretation of these images can be challenging and time-consuming for pathologists. With the advent of machine learning and deep learning, there has been a significant improvement in medical imaging, including the classification of histopathological images.

**Motivation:**

The motivation behind this project is to leverage the power of deep learning to assist in the diagnosis process. Accurate classification of histopathological images can help in identifying the type and stage of cancer, which is crucial for developing effective treatment plans. The use of deep learning can potentially increase the efficiency and accuracy of this process, reducing the workload on healthcare professionals and increasing the chances of early detection and treatment.

**Goal:**

The goal of this project is to develop a deep learning-based model that can accurately classify histopathological images of lung and colon cancer. The model will be trained using the Lung and Colon Cancer Histopathological Images dataset available on Kaggle, which contains 25,000 images with 5 classes, including lung and colon cancer and healthy samples.

**Methodology**

The methodology for this project will involve several steps, including data preprocessing, model selection, training, and evaluation.

**ML Algorithms:**

* In this project, we plan to use PyTorch, an open-source machine learning library, for developing and training neural network-based deep learning models, PyTorch uses dynamic computation, which allows greater flexibility in building complex architectures compared to other popular deep learning frameworks like TensorFlow, which use static computation graphs
* For data preprocessing, the images from the dataset will be resized and normalized to ensure they can be effectively processed by the deep learning model. Data augmentation techniques, such as rotation, scaling, and flipping, may also be used to increase the diversity of the training data and improve the model's ability to generalize
* The model will be based on Convolutional Neural Networks (CNNs), which have shown excellent performance in image classification tasks. Specifically, pre-trained models like VGG16, VGG19, ShuffleNet V2, GoogLeNet, and ResNet18 will be explored for transfer learning. Transfer learning allows us to leverage the knowledge gained from training on large datasets and apply it to our specific task, which can lead to improved performance and reduced training time
* In addition to the CNNs, other machine learning techniques such as Support Vector Machines (SVMs) may be used for classification. These methods can provide a different perspective on the data and may improve the model's performance
* The model will be trained using the training set from the dataset, and its performance will be evaluated using the test set. The evaluation metrics will include accuracy, precision, recall, and F1 score, which provide a comprehensive view of the model's performance

**Parallel Methods:**

* To incorporate parallel machine learning algorithms, we will explore techniques such as data parallelism, model parallelism, and GPU-based parallelism. Data parallelism involves training the model on different subsets of the data simultaneously, while model parallelism involves splitting the model across multiple devices and training each part of the model in parallel, Graphics Processing Units (GPUs) can be used to parallelize machine learning tasks and improve performance. The CUDA framework is a popular tool for GPU-based parallelism
* In this project, we also plan to use Dask, a parallel computing library, we will use Dask to distribute image processing tasks across multiple machines or GPUs and explore techniques such as data parallelism and model parallelism to speed up the training process and improve the performance of the deep learning model. We will also explore the use of Dask for parallel feature engineering and prediction/scoring of large datasets
* Hyperparameter tuning is an important step in the machine learning pipeline, we will use techniques such as manual search, grid search, and randomized search to find the optimal set of hyperparameters for the deep learning model, this will also be parallelized by spinning up multiple tasks in which each task handles training on a single set of hyperparameters
* These techniques can help to speed up the training process and improve the overall performance of the model

**Description of Dataset**

The Lung and Colon Cancer Histopathological Images dataset, also known as the LC25000 dataset, contains **25,000** color images with 5 classes of 5,000 images each sizing up to **2GB**. The images are **768 x 768 pixels** in size and are in JPEG file format. The five classes include colon adenocarcinomas, benign colonic tissues, lung adenocarcinomas, lung squamous cell carcinomas, and benign lung tissues. The dataset was collected for the purpose of detecting lung and colon cancer malignancies using hybrid systems with fused features. The dataset has been used in various studies for lung and colon cancer diagnosis using deep-learning models.

This dataset provides a diverse set of images for training and testing the deep learning model. The dataset is suitable for exploring techniques such as data parallelism and model parallelism to speed up the training process and improve the performance of the deep learning model.

**Data Source**

The data source for this project is the Lung and Colon Cancer Histopathological Images dataset available on Kaggle. The dataset can be accessed at the following link:

<https://www.kaggle.com/datasets/andrewmvd/lung-and-colon-cancer-histopathological-images>