**DeepAthon: Model Documentation**

TASK GIVEN :

The task involves selecting a dataset from Kaggle, specifically a medical dataset, and building a model to predict the classification or other outcomes using machine learning techniques. The model can either be a custom-built model or use transfer learning techniques, which leverage pre-trained models to fine-tune and adapt to the current dataset.

Dataset Taken:  
**Acute Lymphoblastic Leukemia (ALL) image dataset**

Link: <https://www.kaggle.com/datasets/mehradaria/leukemia>

**About Dataset**

As a highly prevalent cancer, the definitive diagnosis of acute lymphoblastic leukemia (ALL) requires invasive, expensive, and time-consuming diagnostic tests. ALL diagnosis using peripheral blood smear (PBS) images plays a vital role in the initial cancer screening from non-cancer cases. The examination of these PBS images by laboratory users is riddled with problems such as diagnostic error because the non-specific nature of ALL signs and symptoms often leads to misdiagnosis.

The images of this dataset were prepared in the bone marrow laboratory of Taleqani Hospital (Tehran, Iran). This dataset consisted of 3256 PBS images from 89 suspected of ALL patients whose blood samples were prepared and stained by skill full laboratory staff. This dataset is divided into two classes benign and malignant. The former comprises hematogones; the latter is the ALL group with three subtypes of malignant lymphoblasts: Early Pre-B, Pre-B, and Pro-B ALL. All the images were taken using a Zeiss camera in a microscope with 100x magnification and saved as JPG files. A specialist using the flow cytometry tool made the definitive determination of the types and subtypes of these cells. After colour thresholding-based segmentation in the HSV colour space, we also provide segmented images.

Paper: [A Fast and Efficient CNN Model for B-ALL Diagnosis and its Subtypes Classification using Peripheral Blood Smear Images](https://doi.org/10.1002/int.22753)  
Source code: <https://github.com/MehradAria/ALL-Subtype-Classification>

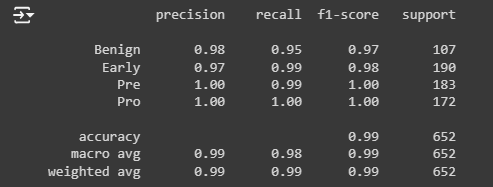
Model Used:  
 I have used the pre-trained VGG19 model for transfer learning. I created a Sequential model by copying all the layers from the original VGG19 model, excluding the final classification layer. I made all these layers non-trainable so that the pre-trained weights from VGG19 are utilized without updating them during training. This approach speeds up convergence and helps prevent overfitting, especially with smaller datasets.



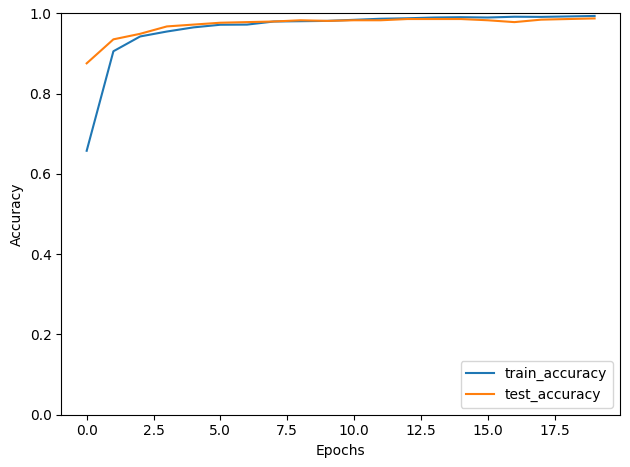
Finally, I added a Dense layer with 4 units and a softmax activation function. This setup allows the model to classify the input into four distinct categories, which matches the number of classes in my classification problem. The VGG19 model's powerful feature extraction combined with this custom classification layer provides an effective solution for multi-class classification.

Data Analysis:

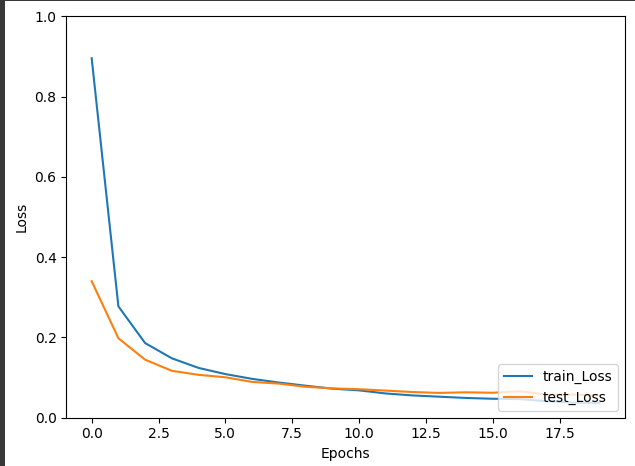
Classification report:



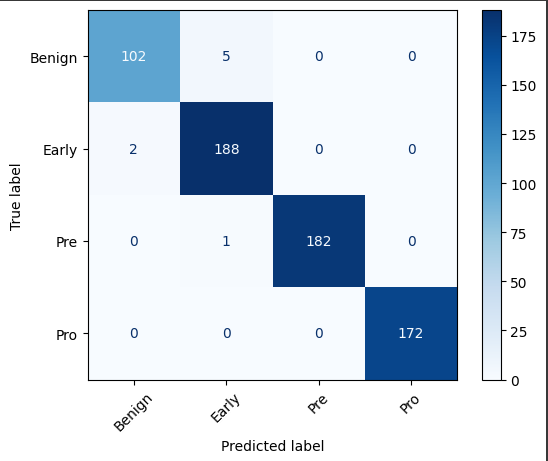
Accuracy Graph:



Loss Graph:



Confusion Matrix:



ROC Curve:

