Computer Vision Assessment

1 Assessment Description

This assessment focuses on image classification using deep learning models for leukemia type detection. You will use different deep learning architectures to train models and evaluate their performance. Based on the results, you should suggest the best model for the dataset.

2 Dataset Description

The dataset contains images of blood cells categorized into six classes:

1. MM (Multiple Myeloma)
2. Healthy
3. CML (Chronic Myeloid Leukemia)
4. CLL (Chronic Lymphocytic Leukemia)
5. AML (Acute Myeloid Leukemia)
6. ALL (Acute Lymphoblastic Leukemia)

3 Usage

Split the dataset into training (60%), validation (20%), and test (20%) sets.

4 Step-by-step guide

1. Download the dataset and upload it to your preferred development environment (e.g., Google Colab, local machine).
2. Preprocess the image data.
3. Split the dataset into training (60%), validation (20%), and test (20%) sets.
4. Build and train the following deep learning models:
   i) Convolutional Neural Network (CNN) from scratch
   ii) Transfer Learning with pre-trained models
   iii) Any other architecture of your choice
5. Implement techniques to prevent overfitting (e.g., dropout, early stopping, data augmentation).
6. Perform hyperparameter tuning to optimize each model's performance.
7. Use the validation set to fine-tune the models and prevent overfitting.

8. Evaluate the models using the test set.

9. Use appropriate evaluation metrics such as accuracy, precision, recall, F1-score, and confusion matrix.

10. Visualize the evaluation results using appropriate plots (e.g., accuracy curves, confusion matrices, sample predictions).

11. Compare the performance of different models and suggest the BEST architecture based on accuracy and other relevant metrics.

5 Deliverables

1. Code implementation of the entire pipeline.

2. Evaluation results and visualizations for each model.

3. A final recommendation on the best performing model with justification.

6 Note

Ensure that you comment your code thoroughly and follow best practices for clean and efficient implementation. Consider the trade-offs between model complexity, training time, and performance in your final recommendation.