GR5074: Projects in Advanced Machine Learning

Project 2 – CNN & Transfer Learning for Medical Image Classification

Your second project for this course involves building a deep learning pipeline to classify medical images, drawing from real-world data and deploying robust models. The task will assess your ability to preprocess data, implement and compare multiple models (including transfer learning), and interpret model performance. You are expected to clearly document your methods and results in a reproducible and structured manner.

Expected Deliverables

Your submission must include:

- A **Jupyter Notebook** (`.ipynb`) that serves as your final report, with **all output cells visible**.
- A **PDF version** of the notebook that includes both **code and outputs**.
- A **GitHub repository** containing the notebook and the PDF. You must include the **link to your GitHub repo** at the top of your notebook.
- You may build on the **starter `.ipynb` file** provided (which includes basic data loading and preprocessing), or choose to organize your notebook in your own structure. Either way, the submission should be clean, complete, and easy to follow.

Project Requirements

1. **Dataset and Exploratory Data Analysis**
    - Start by describing the dataset. Include basic statistics and image samples to show the types of images available (e.g., COVID-positive and negative chest x-rays).
    - **Check if the dataset is balanced across classes.** If it's imbalanced:

        - Discuss potential strategies such as class weighting, oversampling, undersampling, or augmentation.

        - Indicate **which method you chose**, and discuss **how model performance changed** as a result.

    - Reflect on the practical value of this classification task. Who might benefit from your model in a real-world setting?

2. **Baseline CNN Model**

    - Build and train a **basic Convolutional Neural Network (CNN)** to serve as a **baseline**.

    - Clearly describe the architecture, loss function, optimizer, evaluation metrics, and training configuration.

    - Report the model's **training, validation, and test performance**.

3. **Transfer Learning with ResNet**

    - Implement **ResNet** using **transfer learning**.

    - Fine-tune the model and compare its performance with the baseline CNN.

    - Discuss how using pre-trained features influences your model's training and generalization.

4. **Additional Architectures**

    - Implement **three additional models** of your choice.

    - Use consistent data splits and preprocessing across all models to ensure fair comparison.

5. **Performance Comparison**

    - Evaluate all models on the same **test set**.

    - **Highlight the model that achieved the best test performance**.

    - Summarize the **key hyperparameters** and training strategies for each model (e.g., learning rate, batch size, number of epochs, optimizer).

- Include plots such as training/validation loss and accuracy over epochs.

6. **Augmentation**
    - For at least one model, re-train it using **data augmentation techniques**.
    - Describe the types of augmentations used (e.g., flipping, cropping, rotation) and how they affected performance.
7. **Interpretability & Insights**
    - Reflect on which model performed best and why.
    - Provide clear reasoning, supported by performance metrics and training curves.
    - Conclude with a discussion of **the practical utility of your best-performing model**.
        - Who would benefit from using this model?
        - In what types of real-world scenarios would your solution be useful?

Dataset Source

Citation:
M.E.H. Chowdhury, T. Rahman, A. Khandakar, R. Mazhar, M.A. Kadir, Z.B. Mahbub, K.R. Islam, M.S. Khan, A. Iqbal, N. Al-Emadi, M.B.I. Reaz,
"Can AI help in screening Viral and COVID-19 pneumonia?" arXiv preprint, 29 March 2020.
URL: https://arxiv.org/abs/2003.13145