CSE463 Lab Assignment 4

Deadline: 18th April, 11:59PM

Submission instructions:

- Open a folder in Google Drive and name it as- ID_Name_Lab4.
- You do not have to submit a dataset for this assignment, but you must link the dataset on which you implemented the models.
- You must submit an .ipynb file with all the generated outputs such as training outputs, and final evaluation scores.
- You may also submit a pdf, with the results.
- Note that in **Question 2:** You have to classify the same dataset on at least 3 different pre-trained models.

QUESTION 1

Part 1:

Implement the VGG-19 architecture (explained below) in Python using a deep learning library such as TensorFlow or PyTorch to build a sequential model. [6]

Reference (VGG-19 Architecture):

VGG-19 consists of **16 convolutional layers** and **3 fully connected layers**, making a total of **19 layers**.

Define the Input Layer: [0.5]

- **Input size:** 224 × 224 × 3 (Height × Width × Channels).

Add Convolutional Layers: [2]

- Use **3×3 filters** for all convolutional layers.
- Apply **ReLU activation** after each convolutional operation.
- Group convolutional layers into blocks:
    - **Block 1:** 2 convolutional layers with **64 filters** each.
    - **Block 2:** 2 convolutional layers with **128 filters** each.
    - **Block 3:** 4 convolutional layers with **256 filters** each.
    - **Block 4:** 4 convolutional layers with **512 filters** each.
    - **Block 5:** 4 convolutional layers with **512 filters** each.

Add Max Pooling Layers: [1]

- After each block, add a **MaxPooling layer** with a **2×2 pool size** and a **stride of 2** to reduce spatial dimensions.

Flatten the Output: [0.5]

- At the end of the convolutional blocks, **flatten** the 3D feature maps into a 1D vector.

Add Fully Connected Layers: [2]

- Add **two dense (fully connected) layers** with **4096 neurons** each and **ReLU activation**.
- Add a **final dense layer** with **N neurons** (where N is the number of classes in your dataset) and use **softmax activation**.

Part 2: Plot the accuracy and save the model. [5]

Part 3: Discussion

- The role of each layer in the architecture **[3]**
- Why VGG-19 uses small filters (3×3) **[1]**

QUESTION 2

Choose a medical image dataset from Kaggle (e.g., lung cancer, skin cancer, etc.). Apply classification on the dataset using at least three different classification models based on Keras Applications. You can explore Keras Applications to implement any model of your choice.

In your solution, ensure the following:

1. **[**5 marks**] Dataset Selection**: Provide a brief description of the dataset you have chosen. Eg. Dataset size, Train-Test-Validation size, Number of classes, and any unique features.
2. **[**5 marks**] Data Loading and Preprocessing**: Include the steps you performed to prepare the dataset for classification (e.g., resizing, normalization, augmentation if required).
3. **[**5*3=15 marks**] Model Implementation**: Use at least three different pre-trained models from Keras Applications (e.g., VGG16, ResNet50, InceptionV3, etc.). For each model:
    - Load the pre-trained model with appropriate weights.
    - Modify the top layers to suit your classification task.
    - Compile and train the model.

4. **[**5 marks**] Model Evaluation**: Evaluate the performance of each model using appropriate metrics (e.g., accuracy, precision, recall, or F1 score).
5. **[**5 marks**] Comparison**: Compare the performance of the different models. Which one performed the best, and why do you think that is the case?