Assignment: The "Springfield" Identity

Overview

In this assignment, you will build a Convolutional Neural Network (CNN) from scratch to solve a multi-class image classification problem.

Your goal is to train an AI model capable of recognizing characters from the TV show *The Simpsons*. You will handle the entire pipeline: data loading, architecture design, training, and inference.

The Data

You can download the dataset via the Simpsons.zip (One Drive URL)

Directory Structure: The training data is organized hierarchically by class. You must parse this structure to create your dataset:

Plaintext

```
characters_train/
├── homer_simpson/
│   ├── pic_001.jpg
│   ├── pic_002.jpg
│   └── ...
├── marge_simpson/
│   ├── pic_001.jpg
│   └── ...
└── ...
```

- **Twist:** I have withheld **20% of the data** as a private **Test Set**. You have never seen these images. Your model will be evaluated based on how well it performs on this unseen data.

Constraints

- **No Pre-trained Models:** You are **NOT** allowed to use transfer learning or import pre-trained weights (e.g., no `ResNet`, `VGG`, `EfficientNet` loaded with ImageNet weights). You must design and train your CNN architecture entirely from scratch.
- **Libraries:** You may use standard deep learning libraries (PyTorch/TensorFlow/Keras).

Deliverables

You must submit exactly **two Jupyter Notebooks**.

Notebook 1: `train.ipynb`

This notebook contains your entire training pipeline.

- **Reproducibility is Mandatory:** You **must** set random seeds at the start of the notebook (for Python, NumPy, and your DL framework) to ensure results are reproducible.
- **"Run All" Requirement:** I should be able to open this notebook, click **"Run All Cells"**, and it must:
    1. Load the dataset from the structure defined above.
    2. Train your model from scratch.
    3. Save the final trained model artifact (e.g., `model.pth` or `model.keras`) to the local disk.
- **Penalty:** If the notebook crashes, requires manual intervention, or fails to save the model, points will be deducted.

Notebook 2: `inference.ipynb`

This notebook is for evaluation only. It must load the saved model from Notebook 1 and generate predictions on a new dataset.

- **Function Requirement:** You must implement a function with the following signature (or equivalent):

  Python

  ```
  def infer(data_dir, model_path):
      # Logic to load the model
      # Logic to iterate over images in 'data_dir'
      # Logic to predict class for each image
      # Logic to save results.json
      pass
  ```

- **Output:** When executed, this notebook must produce a file named **`results.json`** in the current directory.
- **JSON Format:** The file must map the filename to the predicted class name:

  JSON

  ```
  {
      "pic_999.jpg": "homer_simpson",
      "pic_1000.jpg": "marge_simpson",
      ...
  }
  ```

Evaluation Metric

Your model will be evaluated against the hidden Test Set using the **Macro F1 Score**.

- A high Macro F1 score requires your model to perform well across *all* characters, not just the most frequent ones.

Due Date: 8 days from assignment date