Project Description: CNN-Based Image Classification Using Kaggle Dataset

Introduction

The objective of this project is to compare the performance of several deep learning classifiers on a challenging image classification dataset containing a large number of features and instances. The project aims to determine which classifier performs best based on evaluation metrics such as accuracy, precision, recall, and F1-score. The selected dataset presents a real-world problem with potential industry impact, such as medical diagnosis, autonomous driving, or object recognition.

Three different deep learning classifiers will be implemented and compared:

1. **Deep Neural Network (DNN)** – A baseline fully connected neural network for image classification.
2. **Customized Convolutional Neural Network (CNN)** – A manually designed CNN architecture optimized for the dataset.
3. **Fine-Tuned CNN – A pre-trained CNN model** (e.g., VGG16, ResNet, or EfficientNet) fine-tuned on the dataset for improved performance.

The project will use TensorFlow/Keras for model implementation and evaluation. The best-performing trained model will be saved for deployment and further discussion.

Dataset

The selected dataset is sourced from Kaggle and consists of a large collection of labeled images representing a real-world classification problem. Possible datasets include:

1- **Brain Tumor MRI**
   a. MRI scans classified into tumor/no tumor.
   b. Good for binary medical classification.

2- **Satellite Image Classification**
   a. Images of rivers, forests, urban areas, etc.
   b. Good for land-use classification.

3- *Garbage Classification (12 Classes)*
   a) **Use Case**: Multi-class waste sorting for recycling.
   b) **Best For**: CNN classification models (ResNet, EfficientNet, etc.)

Other Kaggle Datasets is acceptable (Intermediate or advance)
https://www.kaggle.com/search?q=cnn+image+classification+in%3Adatasets.

Methodology

1. Data Preprocessing
   - Image resizing, normalization, and augmentation.
   - Train-test split and label encoding.
2. Model Implementation
   - DNN: A fully connected neural network with multiple hidden layers.
   - Custom CNN: A manually designed architecture with convolutional, pooling, and dense layers.

    - Fine-Tuned CNN: Transfer learning using a pre-trained model (e.g., VGG16) with fine-tuning on the dataset.
3. Evaluation Metrics
    - Accuracy, Precision, Recall, F1-Score
    - Confusion Matrix and ROC Curve (if applicable)
4. Model Saving
    - The best-performing model will be saved in .h5 or .keras format for future inference.

Expected Outcomes

- A comparative analysis of the three classifiers, highlighting strengths and weaknesses.
- Identification of the best-performing model for the given dataset.
- Insights into how deep learning techniques can be optimized for image classification tasks.

Formal Report:

The project requires the submission of a formal report that discusses the results and the problem that being solved . The report should include a description of the dataset and the selected classifiers, the implementation of the classifiers, the evaluation metrics used, and a comparison of the performance of the classifiers. The report also should also discuss the challenges faced during the project and the insights gained from the results. The report should include the following sections:

1- Introduction and problem statement
2- Data description and preprocessing steps if needed
3- Methodology (including the description of the classifiers used,)
4- Results and performance evaluation
5- Discussion and interpretation of results
6- References used in the projects

The report contains a title and list of names, IDs , and emails for all participants in the project.

Deliverable:

1- Your Trained models (provide us with your jupyter notebook code)
2- the Formal report (must be uploaded on Moodle before the discussion due).

Team members (4 to 5 Members max)

Submission Deadline: Monday, [19/5/2025], 11:59 PM (strict cutoff).

Discussion will be Tuesday 20/5/2025 in lab time.