Deep Learning Assignment

Task 1: Artificial Neural Network (ANN)

Task Description

Design and implement a **basic Artificial Neural Network (ANN)** using a **tabular dataset**.

Explanation

In this task, students are expected to understand how a simple neural network works with numerical or categorical data. Students should:

- Select a suitable tabular dataset (e.g., student performance, medical data, house prices).
- Perform data preprocessing such as:
  - Handling missing values
  - Feature scaling (normalization or standardization)
  - Encoding categorical variables
- Design an ANN with:
  - Input layer
  - At least one hidden layer
  - Output layer
- Train the model and evaluate it using appropriate metrics such as **accuracy** or **mean squared error (MSE)**.
- Explain:
  - The ANN architecture
  - Activation functions used
  - Loss function and optimizer selection

Task 2: Convolutional Neural Network (CNN)

Description

Build a **Convolutional Neural Network (CNN)** for **image classification** using a publicly available dataset.

Explanation

This task introduces students to image-based deep learning. Students should:

- Use any public image dataset (e.g., animals, objects).
- Apply image preprocessing such as resizing and normalization.
- Design a CNN architecture including:
  - Convolution layers
  - Pooling layers
  - Fully connected layers
- Train and test the model.
- Evaluate performance using:
  - Accuracy
  - Precision
  - Recall
  - F1-score
- Briefly explain what each evaluation metric indicates about model performance.

Task 3: CNN-Based Image Recognition System using Flask

Task Description

Develop an **end-to-end image recognition system** using **CNN and Flask**, based on a **self-collected dataset**.

Explanation

This is a real-world practical task where students will apply CNNs to solve a realistic problem.

Dataset Collection

- Collect an **equally balanced image dataset** of **at least 3 different people**.
- Each person must have **200–500 images**.
- Images should include:
  - Different angles
  - Different lighting conditions
  - Slight facial expressions

Dataset should be organized as:

```
dataset/
  train/
    person1/
    person2/
    person3/
  test/
    person1/
    person2/
    person3/
```

Model Development

- Build a CNN.
- Train the model on the collected dataset.
- Evaluate the model performance.

Flask Deployment

- Create a **Flask web application** that:
    - Allows image upload or camera capture or live video.
    - Loads the trained CNN model
    - Predicts and displays the name of the recognized person
- Display:
    - Uploaded image
    - Predicted label