README for CIFAR-10 Image Classification Project

Project Overview

This project demonstrates the use of a Convolutional Neural Network (CNN) to classify images from the CIFAR-10 dataset. It includes:

1. A trained model (`cifar_cnn50.h5`) for evaluation.
2. Scripts for loading the test dataset, making predictions, and generating evaluation metrics.
3. Visualizations such as confusion matrices and classification reports.

Setup Instructions

1. **Install Dependencies**
    - Ensure Python is installed.

Install the required libraries using the following command:
Copy code
```
pip install -r requirements.txt
```

    -
2. **Files Included**
    - `main.py`: Main script for evaluating the model.
    - `Cifar_cnn50.h5`: Download from git link. Pre-trained CNN model.
    - `README.txt`: Instructions for running the project.
    - `cifar10_notebook.ipynb`: Jupyter notebook for detailed experimentation (optional).

How to Run the Project

1. **Execute the Evaluation Script**

Run the following command to evaluate the model on the CIFAR-10 test set:
css
Copy code
```
python main.py
```

    -
2. **What the Script Does**
    - Loads the CIFAR-10 test dataset.
    - Loads the pre-trained CNN model (`cifar_cnn50.h5`).
    - Makes predictions and computes evaluation metrics:
        - Accuracy
        - Confusion matrix
        - Classification report (precision, recall, F1-score)
3. **Outputs**
    - Test accuracy printed in the console.

- Confusion matrix plotted to visualize misclassifications.
- Detailed classification report printed in the console.

Key Components

1. **Model File**:
   - `cifar_cnn50.h5`: A pre-trained CNN model created for CIFAR-10 image classification.
2. **Scripts**:
   - **`main.py`**:
     - Contains the `Cifar10` class, which manages model loading, prediction, and evaluation.
     - Key methods:
       - `load_testset()`: Prepares the CIFAR-10 test set.
       - `load_model()`: Loads the trained CNN model.
       - `predict()`: Makes predictions on the test set and calculates accuracy.
       - `evaluate(predictions)`: Displays a confusion matrix.
       - `report(predictions)`: Prints a classification report.
3. **Notebook**:
   - `cifar10_notebook.ipynb`: Optional for users who wish to explore or modify the model training process.

Evaluation Results

Upon running the script, the following results are provided:

- Test accuracy percentage.
- A confusion matrix plot.
- A detailed classification report with precision, recall, and F1-score for each class.

Troubleshooting

- **Dependencies Not Installed**: Ensure the `requirements.txt` file is used to install necessary libraries.
- **File Not Found**: Ensure all files (`main.py`, `cifar_cnn50.h5`) are in the same directory as the script.
- **Python Version Compatibility**: Use Python 3.7 or higher.

Feel free to expand or modify the notebook or scripts for further experimentation!