

Image Classification using CIFAR-10 Dataset by S BHUMIKA & NIDHI SINHA

This project focuses on implementing a convolutional neural network (CNN) to classify images from the CIFAR-10 dataset. CIFAR-10 consists of 60,000 32x32 color images in 10 classes, with 6,000 images per class. The objective is to correctly classify these images into their respective categories.

Table of Contents

- ✓ Introduction
- ✓ Setup
- ✓ Usage
- ✓ Results
- ✓ Contributing
- ✓ License

Introduction

This repository contains code for training and evaluating a CNN model using TensorFlow/Keras to perform image classification on the CIFAR-10 dataset. The model architecture, training process, and evaluation metrics are included.

- I. Setup
- II. Prerequisites
- III. Python 3.x
- IV. TensorFlow
- V. Keras
- VI. NumPy
- VII. Matplotlib

Project Structure

1. ``data/``: Directory for storing the CIFAR-10 dataset.
2. ``models/``: Contains trained model files or checkpoints.
3. ``src/``: Source code directory:
 - ``train.py``: Script for training the CNN model.
 - ``evaluate.py``: Script to evaluate the trained model's performance.
 - ``model.py``: Definition of the CNN architecture.
 - ``utils.py``: Utility functions for data preprocessing, visualization, etc.
4. ``requirements.txt``: List of dependencies and libraries required to run the project.

Installation

1. Clone this repository:

```
https://github.com/Nidhi-Sinha01/AIML-mini-project_BoomNi/tree/main
```

2. Navigate to the project directory:

```
bash
cd image-classification-cifar10
```

3. Install the necessary dependencies:

```
```bash
pip install -r requirements.txt
```
```

Usage

1. Download the CIFAR-10 dataset and place it in the `data/` directory.

2. Training the model:

Run the following command to train the CNN model:

```
```bash
python src/train.py
```
```

3. Evaluating the model:

Once the model is trained, assess its performance using:

```
```bash
python src/evaluate.py
```
```

This script will display metrics such as accuracy, loss, and any other relevant evaluation results.

Results

After running the evaluation script, the model's performance metrics will be displayed in the console. Additionally, visualizations of training/validation accuracy and loss will be available in the `results/` directory.

License

This project is licensed under the MIT License. For more information, refer to the LICENSE file.