Standard -OCR Project Documentation

Abstract

This documentation elaborates on the development and deployment of a character recognition system that utilizes computer vision techniques and deep learning models. It encompasses the dataset structure, problem statement, solution framework, notebook structure, and comprehensive setup instructions to facilitate the project's execution.

In the present digital age, companies relying on paper-based systems are striving to enhance their operational efficiency by transitioning towards digital information management. The advantages of a digital system, including secure and facile information sharing and retrieval, stand in stark contrast to the constraints inherent in paper-based systems. Accumulating piles of paper pose a significant challenge, diminishing productivity as the time wasted searching for pertinent information rises. Optical Character Recognition (OCR) emerges as the solution to this challenge. OCR facilitates the conversion of text from paper or unsearchable PDFs into a searchable and editable format. Leveraging this technology enables companies to swiftly transition their systems into digital formats, significantly reducing the need for re-entering data from scratch.

Introduction

This project centers around the development of a precise model for character recognition within images. Leveraging modern computer vision and deep learning methodologies, the primary goal is to accurately detect and classify characters depicted in input images.

Character recognition has been a subject of significant interest over the past decades, primarily due to its crucial role in image processing, enabling machines to interpret textual content. The journey traces back to 1957 when Frank Rosenblatt, Charles Wightman, and their contemporaries successfully designed the first neurocomputer. This neurocomputer employed a neural network and a "high-resolution camera" to detect characters. Their neural network was limited to a single-layer training capacity, using "The Perceptron Learning Rule." However, advancements in training methods, such as the "Back Propagation" algorithm, enabled the training of multi-layer networks. In this project, instead of directly feeding text into the neural network via a camera or optical scanner, a PNG image derived from an unsearchable PDF is employed. This approach mitigates noise issues arising from poor lighting or other distortions. The implemented OCR system specializes in recognizing monospaced characters, where all characters possess fixed width and height. This design choice significantly enhances efficiency, particularly when using monospaced fonts like "OCR A Extended," which serve as the training foundation for this OCR system.

About the Dataset

The dataset comprises two sections: Data and Data2, each containing Training and Testing directories with 36 subdirectories corresponding to different character classes. The training set consists of 573 images per class, while the testing set includes approximately 88 images per class. Understanding this structured arrangement is crucial for effective data organization and analysis.

Problem Statement

The primary objective is to build a highly accurate and efficient model capable of detecting characters within input images. This task is a computer vision challenge that demands advanced image processing, analysis, and deep learning techniques.

Solution Framework

- **Computer Vision Focus**: This task emphasizes computer vision over traditional OCR techniques due to the dataset's diverse character classes.
- **Model Selection**: Utilizing transfer learning with architectures like ResNet, Xception, Inception, and MobileNet to process and analyze data.
- **Model Optimization**: Focusing on creating an optimized architecture leveraging the strengths of the selected backbone architecture.

Notebook Structure

Set Up

This section involves importing necessary modules, setting hyperparameters, and constants used throughout the notebook.

Data Loading & Processing

Loading the dataset into memory, converting raw data, performing data augmentation, normalization, and resizing images.

Data Visualization

Inspecting the dataset for insights and identifying potential issues like class imbalance or data corruption.

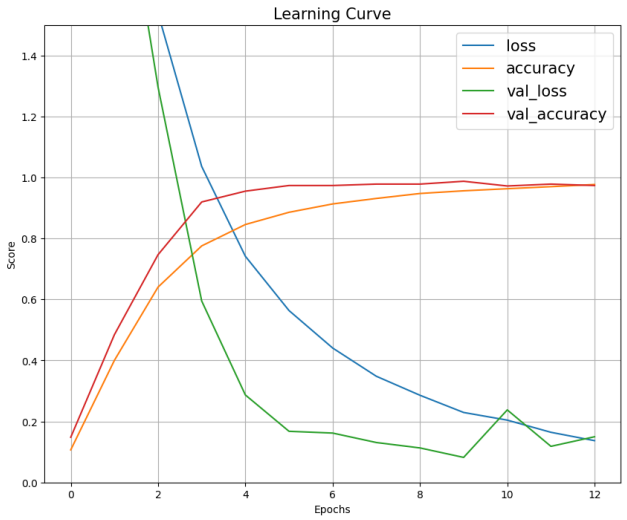
Backbone Comparison

Comparing performance among pre-trained backbones (ResNet, Xception, Inception, MobileNet) to determine the best performer.

All Screenshot Found in GitHub Rep

Model Building & Predictions

Building the model architecture based on the best backbone, training the model, evaluating predictions, and identifying areas for improvement.



Usage Guide

Setup and Installation

- **Prerequisites**: Python 3.11, required libraries (specified in requirements.txt).
- **Installation**: Clone the repository and install dependencies using **pip install -r requirements.txt**.

Running the Project

- **Flask Application**: Execute **python app.py**.
- **Accessing the Application**: Use a web browser to access the application, upload an image, and receive predictions for characters present in the image.

Here is Samples form Website: