Project Card

Biologically Inspired Artificial Inteligence

Project topic: Handwriting Recognition: Architectural Impact Analysis

Project assumptions: Optimizing CNN-RNN Models with CTC Loss for Handwritten Text Recognition

Authors: Kamil Sobociński, Jakub Boduch

Overview

- What will happen:

The project focuses on developing and optimizing a hybrid CNN-RNN architecture for handwritten text recognition. The research involves systematic experimentation with different architectural components to understand their individual and combined impact on model performance.

- Why:

Handwritten text recognition represents one of the most challenging problems in computer vision due to inherent variability in human handwriting styles, character formations, spacing, and writing angles. Traditional OCR systems struggle with handwritten documents, making this an important area for improvement. The project aims to identify optimal architectural configurations through data-driven analysis rather than arbitrary parameter selection.

Method of Implementation

- Technology Stack:

  - Deep Learning Framework: TensorFlow/Keras
  - Image Processing: OpenCV
  - Numerical Computing: NumPy
  - Data Management: Scikit-learn

- Methodology:

  - Baseline Establishment: Implementation of fundamental CNN-RNN architecture as performance benchmark
  - Individual Component Analysis: Systematic modification of single parameters while keeping others constant to isolate impact
  - Strategic Combination: Integration of best-performing individual modifications
  - Performance Evaluation: Comprehensive accuracy analysis

- Architecture Components:

  - CNN Feature Extraction: Hierarchical feature learning from raw pixels
  - RNN Sequence Modeling: Bidirectional LSTM for temporal dependencies
  - CTC Loss Function: Handles variable-length sequences without explicit alignment

Sources

- Data Sources:

  - Primary Dataset: Kaggle - IAM Handwriting Word Database
  - Dataset Size: 38,305 samples (27,579 training / 3,065 validation / 7,661 test)

- Libraries and Frameworks:

  - TensorFlow/Keras: CNN, RNN, Dense layers and CTC loss implementation
  - OpenCV: Image preprocessing and manipulation
  - NumPy: Numerical computations and array operations
  - Scikit-learn: Data splitting (train_test_split)

- Reference Sources:

  - Python TensorFlow/Keras Documentation: https://www.tensorflow.org/api_docs/python/tf/keras/layers
  - Machine Learning Mastery Tutorial: https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/
  - YouTube Playlist - Python Lessons → Machine Learning Training Utilities: https://www.youtube.com/playlist?list=PLbMO9c_jUD46i756J63JJ1_XahCU9Mdaj