Handwritten Character Recognition

CODEALPHA ML Internship Project 3

Name: SAYAB GULFARAZ

ID: CA/S1/1492

**1. Objective:**

* The goal is to recognize handwritten characters from scanned images of names and convert them into digital text using deep learning models.
* This project focuses on the challenging task of recognizing handwritten characters, which can vary significantly due to different writing styles.

**2. Dataset:**

* The dataset consists of **over 400,000 handwritten names** collected from charity projects.
* The dataset is split into:
  + **Training set**: 331,059 images.
  + **Testing set**: 41,382 images.
  + **Validation set**: 41,382 images.
* Each image file corresponds to a name, and the goal is to correctly identify the characters in each name.

**3. Data Preprocessing:**

* **Sampling and splitting**: The training and validation data were randomly sampled to ensure balanced sets.
* **Character mapping**: Two dictionaries were created:
  + char\_to\_label: Converts characters into numerical labels.
  + label\_to\_char: Converts numerical labels back into characters.
* **Data Generator**: A DataGenerator class was built to load and preprocess images (grayscale, resized to 256x64 pixels, normalized) and corresponding labels in batches for model training.

**4. Model Architecture:**

* The model uses **Convolutional Neural Networks (CNN)** followed by **Recurrent Neural Networks (RNN)** with Long Short-Term Memory (LSTM) layers:
  + **CNN layers**: Extract spatial features from the images.
  + **MaxPooling** and **Dropout** layers: Reduce the size of the feature maps and prevent overfitting.
  + **Bidirectional LSTM layers**: Capture sequential dependencies between the characters in the name.
  + **CTC Layer**: A **Connectionist Temporal Classification (CTC)** layer is used to handle sequence prediction, allowing the model to predict variable-length sequences of characters.
* **Optimizer**: Stochastic Gradient Descent (SGD) with momentum is used to train the model.

**5. Model Training:**

* The model is trained for 6 epochs using the training data, and an early stopping callback is added to prevent overfitting.
* The loss function used is **CTC loss**, which is well-suited for sequence prediction tasks like handwritten character recognition.

**6. Model Evaluation:**

* After training, the model was evaluated on the validation set, and the predictions were compared with the ground truth.
* A utility function (decode\_batch\_predictions) was used to decode the output of the network and convert the predicted labels back into readable text.
* The results showed a strong match between predicted text and ground truth for many names, though some discrepancies were observed, likely due to the variability in handwriting styles.

**7. Model Performance:**

* Ground truth and predicted names were compared, with many accurate predictions such as "BENOIT", "ANGELINE", and "VERDELET".
* Some errors in predictions occurred, likely due to noisy data or overlapping characters in the handwriting.

**8. Model Testing:**

* The model was tested on unseen data (new handwritten names), where it successfully predicted names like "JULES".
* The model demonstrated its capability to generalize and predict new handwritten names that it had not seen during training.