Arabic Image to Text (OCR)

Harnessing the power of deep learning to bridge the gap between Arabic visual text and digital content, exploring advanced neural network architectures to accurately interpret and convert Arabic handwritten and printed text into editable formats using the CNN and LSTM networks.

Ismael Mousa ** Amjad Awad

Abstract

This study focuses on developing a robust Optical Character Recognition (OCR) system for Arabic handwritten and printed text images using deep learning architectures. We conducted multiple experiments on the Muharaf and Arabic-img2md datasets to evaluate the performance of different model architectures, including training from scratch and using pretrained ResNet backbones. Our approach combines convolutional layers and bidirectional LSTM networks with a CTC loss function to effectively recognize Arabic script from images. This work presents an Arabic OCR system leveraging deep learning, specifically employing ResNet-based architectures combined with sequence modeling using Bidirectional LSTMs and Connectionist Temporal Classification (CTC) loss. The system preprocesses the datasets, constructs an end-to-end trainable model, and evaluates performance using edit distance metrics. Results demonstrate the effectiveness of transfer learning in Arabic OCR tasks.

Introduction

Arabic handwritten OCR remains a challenging task due to the complex morphology of Arabic script, its cursive nature, and varying handwriting styles. Optical Character Recognition (OCR) for Arabic script presents unique challenges because of its cursive nature and diverse glyph shapes. This work explores deep learning techniques to improve recognition accuracy by experimenting with different network architectures and transfer learning strategies on Arabic text datasets. We leverage sequence learning with CTC loss applied to CNN-LSTM architectures, including ResNet variants as feature extractors. Recent advances in deep learning, especially convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have improved OCR accuracy significantly. This study investigates the use of ResNet architectures as feature extractors for Arabic OCR and evaluates their performance on publicly available datasets, using the CTC loss for sequence alignment without explicit segmentation. Arabic OCR is particularly challenging due to the cursive writing style with character connectivity, complex character shapes and diacritics, right-to-left writing direction, and diverse handwriting styles. We address these challenges through hybrid CNN-LSTM architectures, Connectionist Temporal Classification (CTC) loss, transfer learning with ImageNet-pretrained models, and custom preprocessing pipelines.

Datasets

The success of any Optical Character Recognition (OCR) system, especially for complex scripts like Arabic, hinges significantly on the quality and diversity of the datasets used for training and evaluation.

Muharaf

formally introduced in the paper titled *"Muharaf: Manuscripts of Handwritten Arabic Dataset for Cursive Text Recognition"*, is a meticulously curated collection aimed at advancing the field of Handwritten Text Recognition (HTR) for Arabic scripts. This dataset comprises over 1,600 high-resolution images of historical handwritten Arabic manuscripts. These documents encompass a wide array of genres, including personal letters, diaries, poems, church records, and legal correspondences, reflecting the rich tapestry of Arabic literary heritage. Each image in the Muharaf dataset is accompanied by detailed annotations, including spatial polygonal coordinates delineating text lines and other essential page elements. This granular level of annotation facilitates precise training and evaluation of OCR models, particularly in handling the cursive and context-dependent nature of Arabic script. The dataset is publicly accessible via Hugging Face , where it is available in Parquet format, ensuring efficient data handling and integration into machine learning pipelines. The dataset is partitioned into training, validation, and test sets, comprising 22,100, 1,070, and 1,330 samples respectively. Images are standardized to a size of 100x60 pixels with padding, and the text annotations cover a vocabulary of 74 unique characters, accommodating a maximum text length of 181 characters. Preprocessing steps include filtering non-Arabic characters, aspect ratio-preserving resizing, and normalization against a white background, ensuring consistency and quality across the dataset. You can access the Muharaf dataset: https://huggingface.co/datasets/aamijar/muharaf-public.

Arabic-Img2md

Available on Hugging Face , is a large-scale dataset designed to support the development and benchmarking of OCR systems for Arabic text. This dataset contains approximately 2.16 million samples, making it one of the most extensive publicly available resources for Arabic OCR research. Each sample in the dataset consists of an image-text pair, with images standardized to a size of 80x35 pixels. The textual content is limited to a maximum of 10 characters, covering a vocabulary of 37 unique Arabic characters. The dataset is provided in Parquet format, facilitating efficient data processing and integration into various machine learning frameworks. Preprocessing steps include dynamic padding to maintain aspect ratios, pixel value normalization, and sorting by text length to optimize batch training processes. The extensive size and standardized format of the Arabic-OCR-Dataset make it an invaluable resource for training deep learning models capable of handling the nuances of Arabic script. By leveraging the complementary strengths of the Muharaf and Arabic-OCR-Dataset, we aim to develop OCR models that are both accurate and generalizable across different types of Arabic text. The Muharaf dataset provides rich, contextually complex handwritten samples, while the Arabic-OCR-Dataset offers a vast array of short text instances, enabling our models to learn from a diverse set of examples and perform effectively in real-world scenarios. You can access the Arabic-Img2md dataset: https://huggingface.co/datasets/mssqpi/Arabic-OCR-Dataset.

Methodology

Our approach began with implementing a custom data loader to preprocess images and texts by converting Arabic text into integer sequences via a character lookup. The preprocessing pipeline involved resizing images with padding and converting Arabic text to numerical sequences using a custom vocabulary and TensorFlow's StringLookup layer. We constructed a custom TensorFlow dataset pipeline for efficient batching and prefetching to improve training speed. Images were resized with padding to maintain aspect ratio and converted to RGB with a white background. Text labels were converted into sequences of character indices using TensorFlow's StringLookup. Data was normalized using preprocess_input from Keras for compatibility with ResNet pretrained weights. TensorFlow datasets were prepared with padded batches for efficient training.

The model architecture consists of an input followed by either a ResNet50 or custom CNN for feature extraction, then four bidirectional LSTM layers, a dense layer with softmax activation, and finally CTC loss for training. We use the Adam optimizer with a learning rate of 1e-4, batch sizes ranging from 4 to 16, and train for 10 epochs with early stopping. Training is performed on GPU-accelerated hardware. Feature extraction utilizes either pretrained ResNet50 or a custom CNN with batch normalization and max pooling. Sequence modeling is done with bidirectional LSTM layers having hidden sizes decreasing from 128 to 16 and dropout applied for regularization. CTC implementation includes a custom Keras layer, beam search decoding with width 100, and edit distance evaluation. The extracted features were then reshaped and fed into a stack of bidirectional LSTM layers. The final dense layer outputs character probabilities and is optimized using a Connectionist Temporal Classification (CTC) loss wrapped inside a custom Keras layer. A custom CTCLayer was implemented to compute the CTC loss during training. Training employed early stopping and batch loss logging callbacks, with the Adam optimizer set at a learning rate of 0.0001. Training was performed for up to 10 epochs with early stopping based on validation loss.

Results

When training from scratch on the Muharaf dataset for 10 epochs, the model showed a progressive reduction in loss on the training set, while the validation loss fluctuated but showed signs of convergence. The final model contained approximately 7.19 million parameters. Evaluation included edit distance calculations on the test set, measuring the average discrepancy between predicted and ground truth sequences. The pretrained ResNet50 backbone model was also evaluated. The dataset was split into 50,000 training samples, 1,000 validation samples, and 1,000 test samples. Images were preprocessed by resizing with padding to a fixed size of 80x35 pixels while preserving the aspect ratio, and pixel values were normalized using the standard ResNet preprocessing pipeline. The character vocabulary contained 37 unique Arabic characters, with a maximum text length of 10 characters. A TensorFlow StringLookup layer was used to convert characters to numerical indices, and a custom Connectionist Temporal Classification (CTC) layer was used for loss computation. The model architecture consisted of the ResNet50 base model (without the top classification layers), followed by resizing and reshaping layers, a dense layer with ReLU activation and dropout, and four stacked bidirectional LSTM layers with decreasing hidden sizes. The final output layer applied softmax activation to predict character probabilities for each timestep. The model contained over 31 million trainable parameters and was optimized using the Adam optimizer with a learning rate of 0.0001. Training was performed for up to 10 epochs with early stopping based on validation loss, using a batch size of 4 for efficient GPU memory usage. Training loss steadily decreased from 78.55 at epoch 1 to 2.68 by epoch 10, showing significant convergence. Validation loss followed a similar trend, decreasing from 56.19 at epoch 1 to approximately 23.62 at epoch 10, with some fluctuations after the 6th epoch.

Evaluation used an edit distance metric comparing predicted text sequences to ground truth labels on the test set. Predictions were decoded using CTC beam search decoding. This experiment demonstrated that the pretrained ResNet50 backbone effectively extracted visual features from Arabic text images, enabling the recurrent layers to learn accurate sequence modeling despite the complexity of Arabic script and limited maximum sequence length. The from-scratch model showed a training loss decrease from 349.96 to 35.30 but had fluctuating validation loss, with 7.19 million parameters. The ResNet50 pretrained model reduced training loss from 78.55 to 2.68 and validation loss from 56.19 to 23.62, using 31 million parameters. Pretrained models converge approximately three times faster and achieve the best validation edit distance around 23.62. The system effectively handles character variations, diacritic marks, and connected letters.

Model	Training Loss	Validation Loss	Parameters	Convergence Speed	Edit Distance
Scratch	349.96 to 35.30	Fluctuating	7.19 million	Slow	Higher
ResNet50	78.55 to 2.68	56.19 to 23.62	31 million	Fast	Lower

Conclusion

The baseline models trained from scratch demonstrate the feasibility of Arabic handwritten OCR using convolutional-recurrent architectures combined with CTC loss. The preprocessing pipelines preserved image quality and text integrity, and the models showed promising convergence within 10 epochs. Future experiments with pretrained ResNet backbones aim to explore improvements through transfer learning and deeper architectures. The integration of a pretrained ResNet50 backbone with recurrent sequence modeling and CTC loss provides a viable framework for Arabic OCR. While training loss showed steady improvement, validation loss indicated challenges likely due to dataset complexity or model capacity. Future work will explore deeper ResNet variants and additional datasets to improve generalization and accuracy.