HANDWRITTEN ARABIC TEXT RECOGNITION

MERNA MUSTAFA EL-REFAIE – NEVEEN SAMIR NAGY

Faculty of Engineering, Alexandria University

ABSTRACT

Arabic language is one of the most challenging languages because of its characters representation. Arabic characters can be represented in more than one shape, not like English characters, and that the main reason that makes Arabic language more challenging. In this project, we build a model that recognizes the handwritten Arabic texts and returns an encoded text as a result.

CONTACT

<Merna El-Refaie>
Email: mernamustafa97@gmail.com
<Neveen S.Nagy>
Email:
neveensamirnagy@gmail.com

INTRODUCTION

Handwriting text recognition is defined as the task of transforming a language represented in its spatial form of graphical marks into its symbolic Representation. It is a challenging problem due to having various kinds of handwritten characters such as digit, numeral, cursive script and symbols. Among cursive scripts, Arabic text recognition is considered as a more challenging problem, compared to handwritten Latin text recognition, due to joined writing, same character variations, a large number of words and ligatures, variations in font style, etc..

DATASET

KHATT Dataset contains 2000 paragraphs written by 1000 writers collected from 46 sources and gathered across 18 countries. The paragraphs are segmented into a total number of 6744 unique text lines divided into three disjoint sets, training (70%), validation (15%) and testing (15%).

Set	Unique Text Lines
Train	4837
Validation	939
Test	960

Table 1. KHATT Dataset.

DATA PREPROCESSING

As we see in Figure 1. the image in the dataset contains white space, the line may be skewed, and the images have different size, so the following preprocessing operations are applied on the dataset:

- Prune extra white space regions.
- De-skew lines using skew detection and correction techniques.
- Resize image.

Figure 2. represents result image after preprocessing.

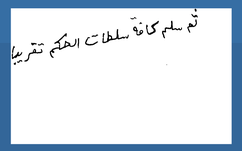

Figure 1. Image from dataset.

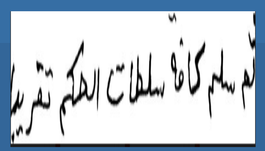

Figure 2. Result after preprocessing.

MODEL ARCHITECTURE

Model Architecture:

- **The input Layer which is the image.**
- **2 CNN Layers with Max-Pooling.**
- **Dense Layer with Dropout.**
- **2 GRU Layers, one with go backward and one without.**
- **Add Layer.**
- **2 GRU Layers, one with go backward and one without.**
- **Concatenation Layer.**
- **Dense layer with number of units = number of classes which are the Arabic characters + Special Symbols.**
- **CTC Layer to decode output from model and compute loss.**

Figure 3. shows model architecture.

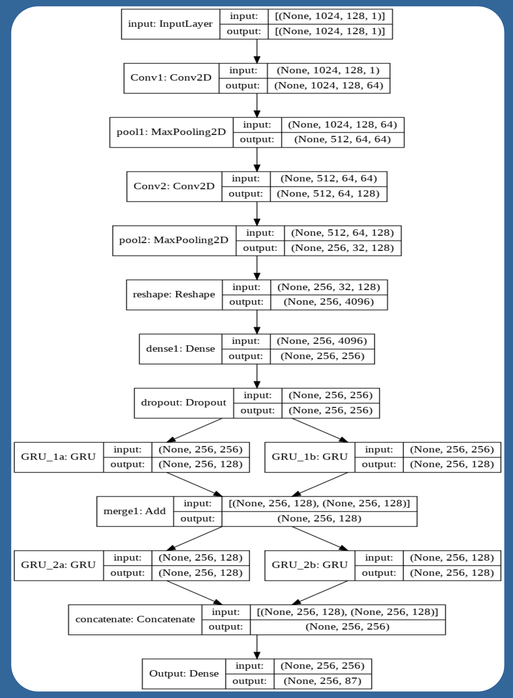

Figure 3. Final Model Architecture.

DISCUSSION

- The final model is trained on KHATT dataset (unique text lines) with number of epochs = 100 and starting with learning rate = 0.001 to be reduced if there is no improvement after 5 epochs and stop after 10 epochs when there is no improvement.
- CNN layers are used to extract feature maps from input image and then use RNN layers (GRU) to extract the predicted output text.
- CTC layer is used to decode output and compute loss by comparing ground truth with output text from model.
- Add attention mechanism on final model (Model 1).
- Appling hyperparameters tuning on final output by increasing the number of filters in CNN layers and units in GRU layers (Model 2), decreasing number of filters in CNN layers and units in GRU layers (Model 3), and increase batch size (Model 4).
- Change in final model architecture by adding CNN layers (Model 5).
- Add CNN layers and attention mechanism on final model (Model 6).

Table 2. represents the result of models. Figure 4. represents plotting of loss of final mode. Chart 1. shows results of models compared by character error rate (CER), word error rate (WER), and sequence error rate (SER).

RESULTS

Model	CER	WER	SER
1	20.96	64.74	99.89
2	96.07	99.91	100
3	21.93	66.96	100
4	19.04	61.14	100
5	19.4	61.09	100
6	19.71	61.33	99.89
Final	**18.93**	**59.81**	**100**

Table 2. Results.

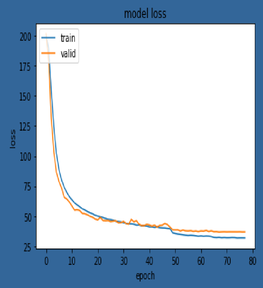

Figure 4. Loss of Final Model.

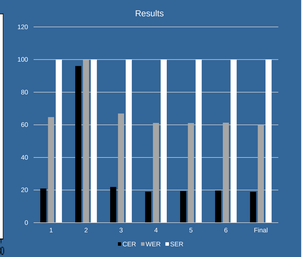

Chart 1. Results of Models.

CONCLUSION

- Data Augmentation increases result. (75.8% → 80.02%).
- Tune hyperparameters may lead to underfitting or get more better result.
- Less deep model may lead to better result then deeper.
- Figure 4. represents plotting for loss.
- Our final result:
  CER → 18.93%
  Recognition Rate → 81.07%

FUTURE WORK

- Increase dataset by mixing found datasets.
- Tune more hyperparameters.
- Try to change input to model to be a paragraph not a line.

REFERENCES

1. [KHATT: A Deep Learning Benchmark on Arabic Script]
2. [A Deep Learning based Arabic Script Recognition System: Benchmark on KHAT]
3. [HTR-Flor: A Deep Learning System for Offline Handwritten Text Recognition]
4. [Image OCR], https://github.com/...