Deep Learning Handwriting Algorithm:
Translating Written Words to Text

By: Baron Davis, William Farris, Micheal Oyenekan, and Creighton Young, Team 15

UNIVERSITY OF ARKANSAS
College of Engineering

Computer Science & Computer Engineering

Introduction

The presence of hand-written documentation causes various issues when it comes to document processing due to the requirement that the text presented is machine-readable.

In order to convert hand-written documentation to machine-readable text, we can rely on an application of Deep Learning known as Handwriting Text Recognition (HTR).

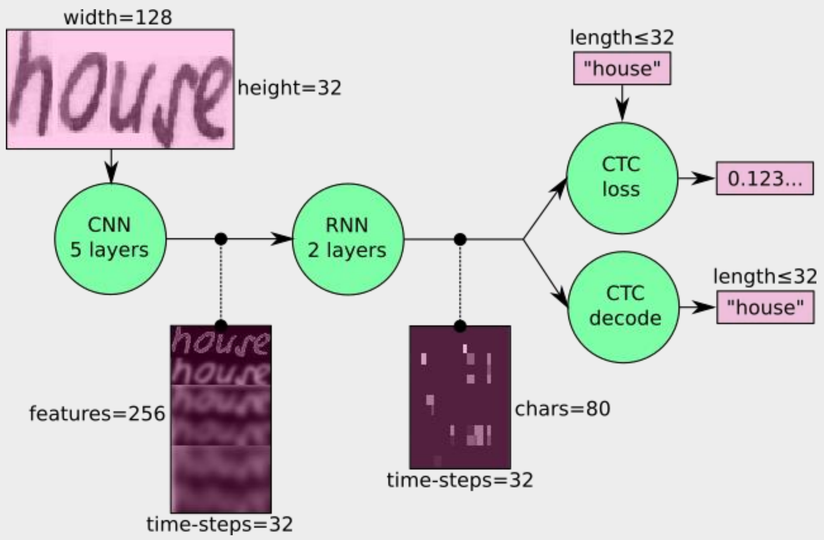
CSCE Capstone students experimented on an open-source HTR program to find ways to improve accuracy.

Model

SimpleHTR is an open-source Deep Learning Handwriting Recognition Model that takes in a handwritten input of either a full line or a single word. The model consists of Convolutional NN (CNN) Layers, Recurrent NN (RNN) Layers, and Connectionist Temporal Classification (CTC) layer.

The model processes the handwritten input, then attempts to give a machine processed equivalent, along with a probability that the output provided was correct.

Simple overview of the SimpleHTR Model



Acknowledgements

- CGI Sponsors: Nathaniel Zinda and Rishi Dhaka
- Open-Source Model: SimpleHTR by Harald Scheidl

  Link: https://github.com/githubharald/SimpleHTR
- Model Image credits to Harald Scheidl:
  https://towardsdatascience.com/build-a-handwritten-text-recognition-system-using-tensorflow-2326a3487cd5

Implementation

- IAM Dataset: The model was retrained using a database of handwriting samples and was able to provide an increase in accuracy of less than 1%.
- Augmented Dataset: An adjusted version of the IAM that either blurs, thickens, stretches, or add random noise to the image
- Optimizer: Between RMSProp and Adam Optimizers, the Adam Optimizer had the greater accuracy.
- Spellcheckers: Pyspellchecker and textblob to help correct the model's mistakes, and gingerit to check the grammar of the output.
- Deslantimg: An algorithm used in order to turn the slanted cursive handwriting into print for easier model correction
- Replace LSTM by 2D-LSTM
- Decoder: Uses word beam search decoding

Conclusion

Fortunately, for us at the last hours of project, we were able to increase the model accuracy from 66% to 81%