Handwritten Alphanumeric Character Recognition with Deep Neural Networks using the Extended MNIST Data Set

Matthew Smith
Department of Computer Science and Information Technology
Hood College

Introduction

This poster describes the work of creating a neural network to perform handwritten character recognition using the Extended MNIST (EMNIST) data set, which consists of uppercase alphabetic, lowercase alphabetic, and numeric characters.

Handwriting recognition using the original MNIST data set of handwritten numeric characters is a common first project in machine learning and computer vision. This research examines how successfully similar techniques extend to data containing all possible alphanumeric characters, excluding special characters, with applications in document digitization and automated data entry.

The EMNIST data set contains 814,255 128x128 images of handwritten characters, the majority of which are numeric. The spread of the data is shown in Figure 1. An encoding algorithm was used to resize them and assign each pixel a grayscale color value between 0 and 1.

Topics discussed include data exploration, pre-processing and normalization, model specification and training using TensorFlow, analysis of accuracy on testing data, and generalization of the model to new data.

Methods

Images were first run through an encoding algorithm in which they were cropped to the bounds of the character, padded, downsized to a lower resolution, then interpreted as a two-dimensional array of color values. An example of an encoded character is shown in Figure 2.

The data set consisted of 814,255 rows (one for each image) and 1,025 columns (1,024 for image data, and one for the label). The data set was shuffled and split such that 80% of the data was used for training and the remaining 20% was used for testing.

Implementation of the neural network was done with the Sequential model API from TensorFlow. The layers of the neural network consisted of an input layer, a hidden layer with 256 neurons, a hidden layer with 128 neurons, a hidden layer with 64 neurons, a dropout regularization layer, and an output layer with 62 neurons (one for each label). Training the model consisted of 40 epochs with a learning rate of 0.002 and a dropout rate of 0.35. A visualization of model training is shown in Figure 3.

After the model was trained on the EMNIST data, it was tested on handwritten characters not included in the original data set as a means of assessing the model's generalizability. This data was run through the same encoding algorithm as the EMNIST data.

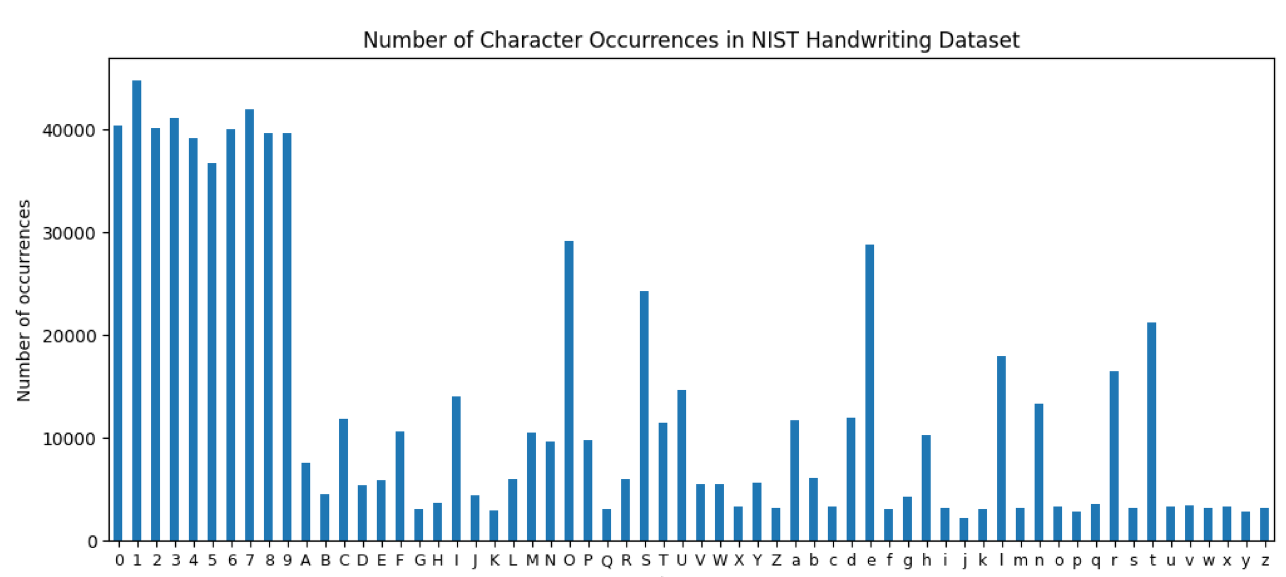

Figure 1: Counts for each character in the EMNIST data set

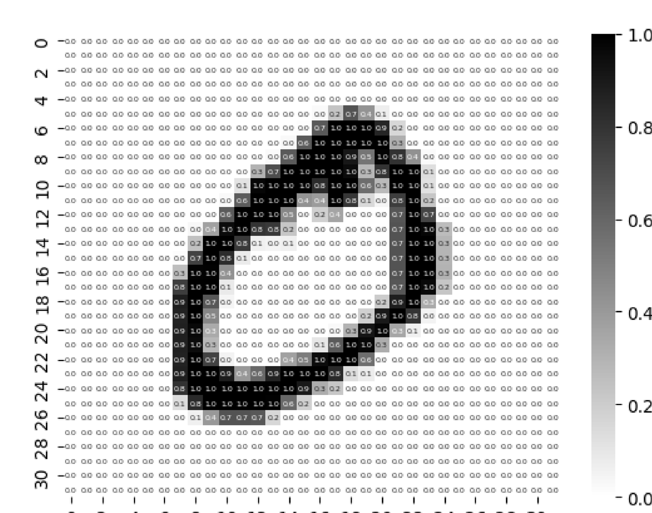

Figure 2: Example of the character "0" as a matrix of grayscale color values

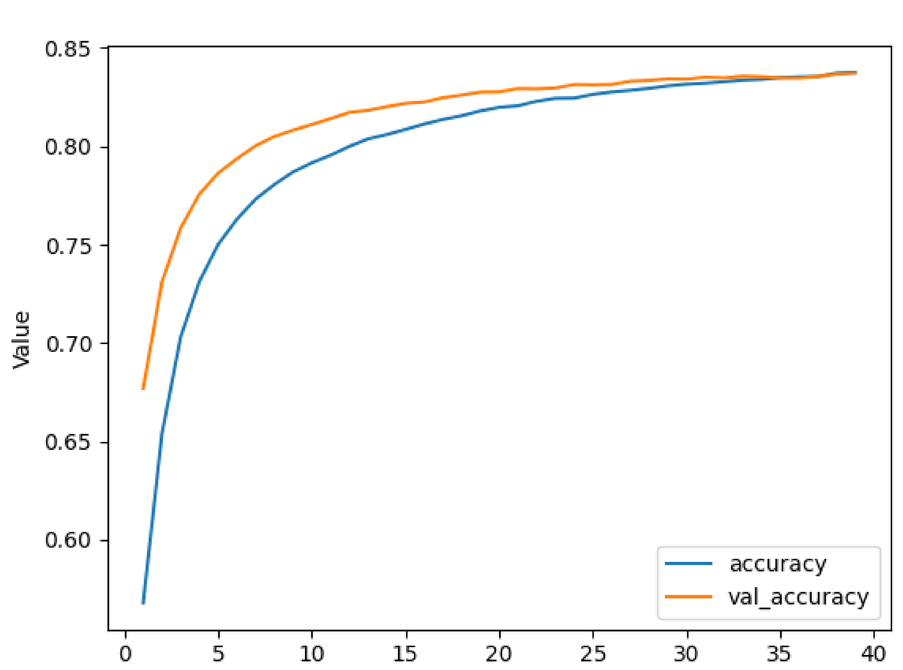

Figure 3: Visualization of model's accuracy across 40 epochs of training

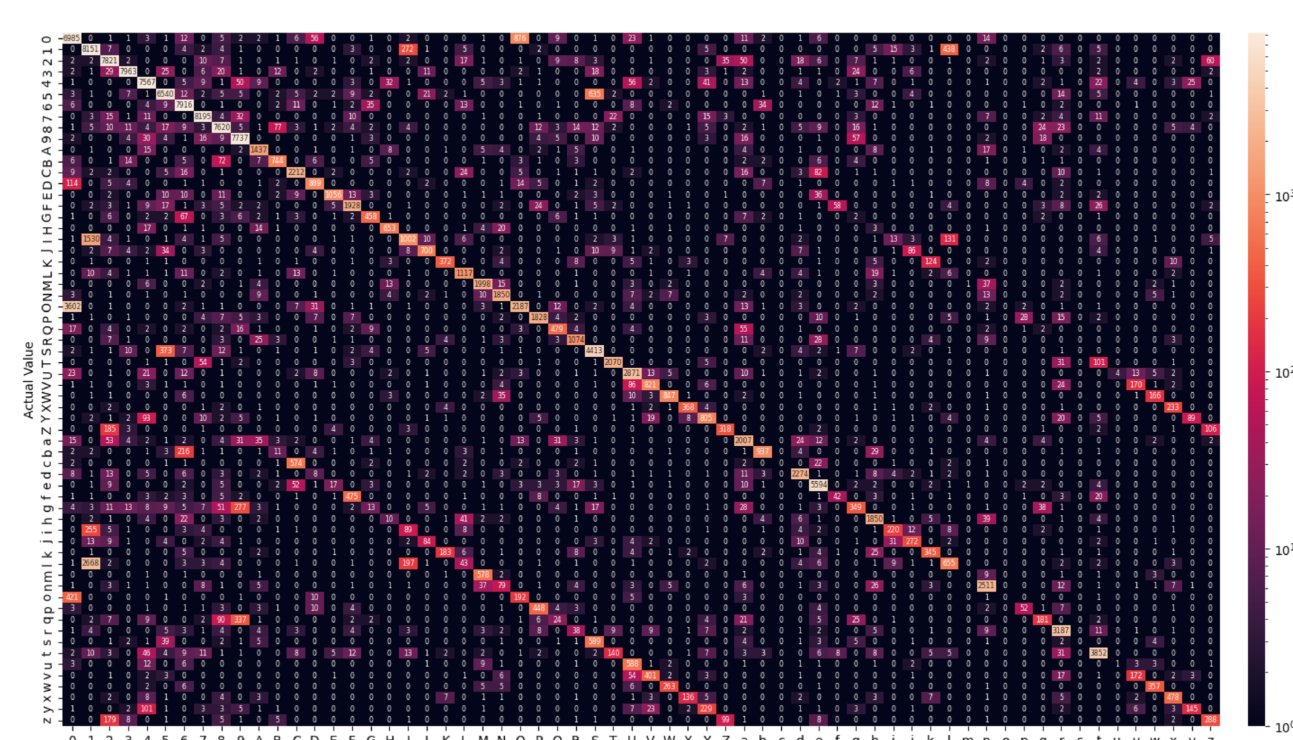

Figure 4: Confusion matrix of actual characters in the testing set versus the predicted character by the model. A brighter color means a greater number of predictions.

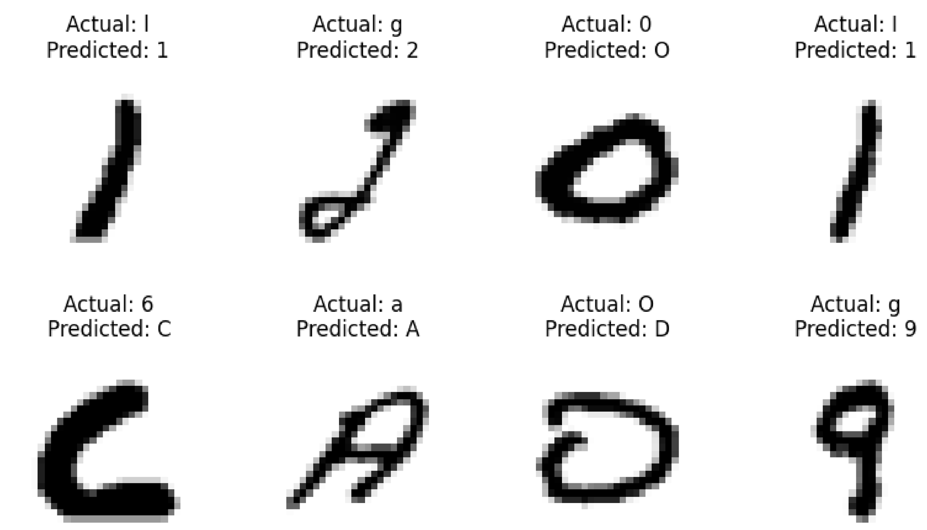

Figure 5: Examples of incorrect classifications made by the model

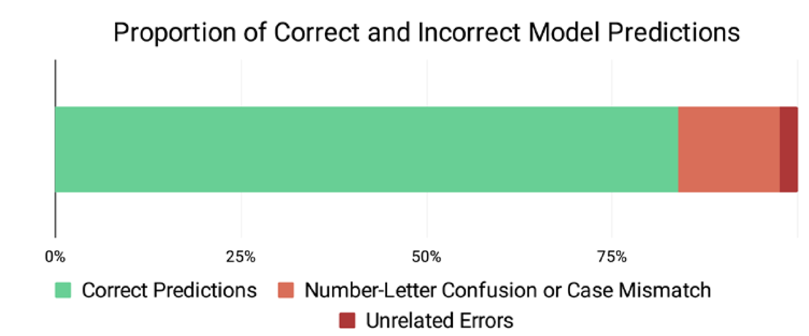

Figure 6: Proportions of common errors made by the model

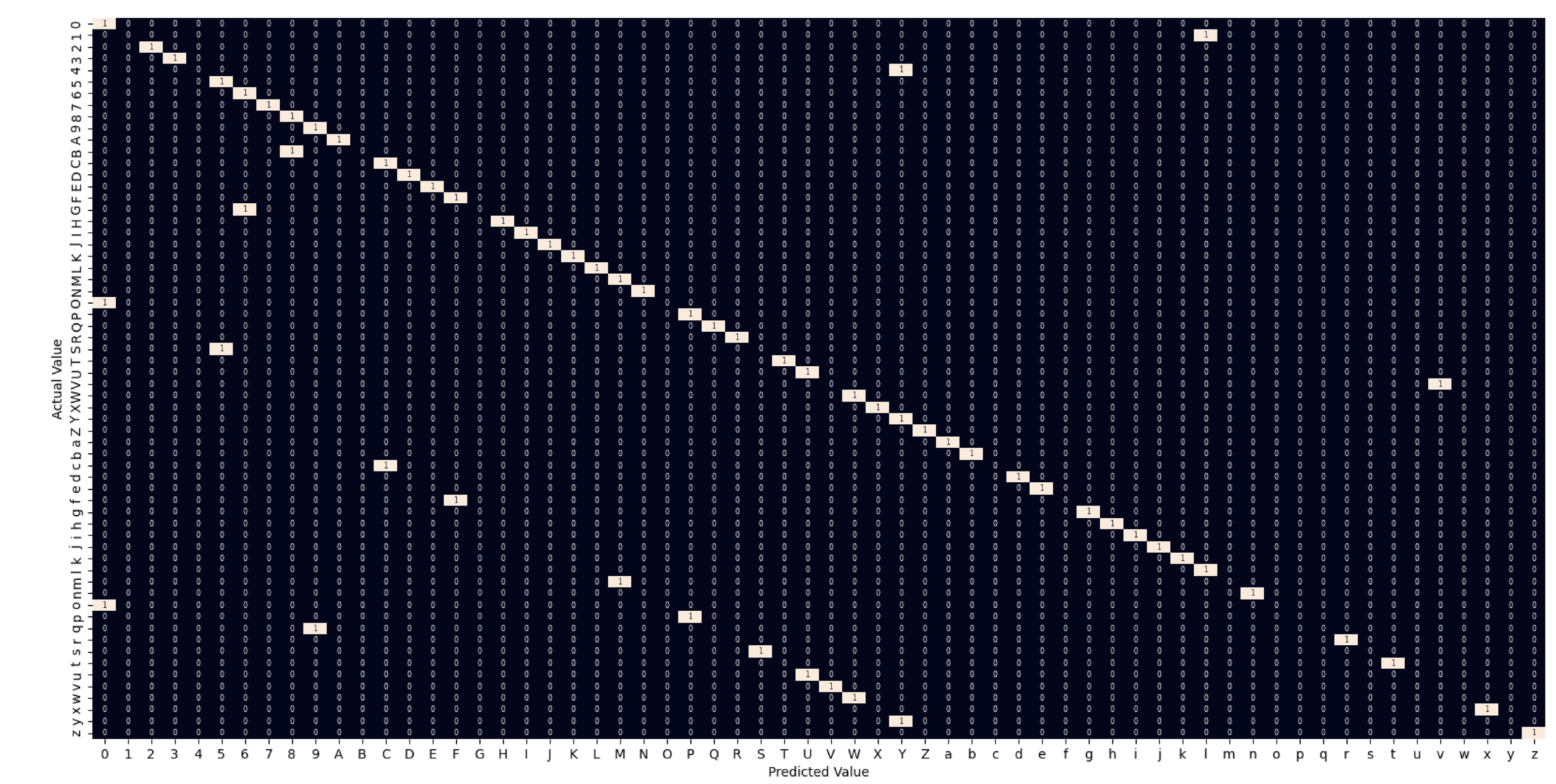

Figure 7: Confusion matrix of actual characters in the external data set versus the predicted character by the model. A white cell indicates a prediction, while a black cell indicates no prediction.

Results

The confusion matrix in Figure 4 shows the predictions made by the model on the testing set. The model achieved an accuracy of 84% on the testing data.

Examples of incorrect classifications made by the model are shown in Figure 5. Some commonly confused characters were 0, O, and o, and 1, I, and l. These are examples of the two most common sources of errors: number-letter confusion and case mismatch.

As shown in Figure 6, number-letter confusion and case mismatch made up about 85% of the incorrect predictions made by the model. Other than the two aforementioned sources of error, another potential source of error is reduced detail from low image resolution due to the encoding process.

The confusion matrix in Figure 7 shows the predictions made by the model on new handwritten characters. The model's accuracy was only 70% on this data. Many of the same errors are visible, specifically number-letter confusion and case mismatch.

Possible sources of error for the new data include messiness of the handwriting, low sample size, different data distribution compared to the EMNIST data set, and a different scanning and normalization process than what the NIST used before releasing their data set.

Future Work

To increase accuracy, future work may include the use of convolutional neural networks or different encoding techniques that mitigate loss of detail in the images.

In applications, two simple changes can be made to significantly raise accuracy: using separate models for numeric and alphabetic data, and discarding information about case sensitivity. These changes would remove 85% of errors, bringing this model up to a hypothetical maximum accuracy of about 97% (assuming the model would correctly predict all cases it previously mistook due to number-letter confusion or case mismatch).

Contact

Matthew Smith
matthewsmith2003@yahoo.com
(240) 818-9022

Faculty Advisor

Jiang Li, Ph.D.
lij@hood.edu
(301) 696-3564

References

1. https://www.nist.gov/srd/nist-special-database-19