Pixel-Level Analysis of Handwriting: A CNN-Based Approach for Writer Verification

Val Rogel

December 16, 2024

1 Abstract

Given the prevalence of opportunities for the use of handwriting analysis in identity verification and criminal investigations, there are clear benefits to enhancing accuracy of handwriting identification, especially with regards to scanned or digitized text. In this study, I used image processing and convolutional neural network (CNN) techniques to explore whether pixel values extracted from images of lines could achieve binary predict of handwriting matches and non-matching. By leveraging the Institut für Informatik und angewandte Mathematik (IAM) handwriting database, thousands of individual lines were used to create a large sample of handwritten text. Although the precision and recall are modest, fine-tuning of the model and efforts to address imbalances of the training data are promising means through which these metrics may be improved.

2 Introduction

Forensic handwriting analysis, in combination with other methods of forensic analysis, and frequently strengthened by electronic device data and DNA samples, is considered a valid technique by the FBI Laboratory division. Several high-profile cases have used these methodologies, including the Lindbergh kidnapping case of the 1930s (Durina, 2021). Today the United States Postal Inspection Service uses handwriting analysis techniques in combination with analysis of ink, paper, and watermarks, to conduct investigations surrounding mail-related crimes (USPIS). However, this tool is considered to have questionable levels of accuracy, especially when experts disagree on the classification of a sample as to whether it comes from a certain individual.

In 2009, the National Research Council stated that there "may be a basis" for handwriting comparison in the "absence of intentional obfuscation or forgery" but that "analysis of inks and paper, being based on well-understood chemistry, presumably rests on a firmer scientific foundation." When evidence is presented in an investigative setting, it has the power to bias an investigation against or towards a certain individual, and it is important that forensic handwriting analysis has a strong scientific basis, ideally one which does not depend on the individual interpreting a sample.

A publication by the FBI Laboratory Division yielded that for 6,576 responses from 86 participants considered reasonably "expert" in the field of forensic handwriting analysis, 88.3% of trials on matched handwriting analysis sets and 77.4% of trials on non-matched sets were consistent with ground truth, with a subset of the inconsistent responses being an "inconclusive" response (Hicklin et al., 2022).

It appears as though handwriting identification analysis would strongly benefit from a data-driven approach which is both more decisive and more accurate, allowing even conservative estimates to be increasingly reliable compared to human experts. Handwritten identification of words and letters is an existing body of research, along with machine learning and neural network approaches to binary classification of handwriting samples. An existing study using a simple convolutional neural network (CNN) for n = 10 different writers. Its outcome of a 96% predictive accuracy proves promising for binary classification of handwriting when samples and number of possible writers are significantly larger (Abiodun et al., 2024).

3 Methods

3.1 Data Collection - Images

I imported preprocessed data from the IAM Off-line Database, which contains handwriting samples and associated metadata in English from 657 individuals (Marti and Bunke, 2002). Importantly, the sentences all originate from the Lancaster-Oslo/Bergen (LOB) corpus, which contains texts such as press releases, fiction writing, and letters. Individuals recruited by the original data collectors were asked to use rulers to make the image processing as easy as possible, and it was requested that these individuals write as naturally as possible. If the page was filled, they were asked to stop writing.

The images are publicly available, downloadable as PNG images with "256 gray levels." Available at the sentence, word, and line level, this study makes use of the 13,353 isolated and labeled text lines (Zimmerman and Bunke, 2000). As of 2018, there have been hundreds of publications using this dataset, which speaks to its quality as verified by other researchers.

3.2 Processing

Making use of the metadata, I matched the writing and file name with the file paths of images of individual lines. Next, pairs of file paths were generated, excluding matching lines written by the same individual. In order to limit runtimes for image preprocessing, I limited the scope of my dataset to n = 8135 pairs of samples which were almost entirely randomly selected, except for approximately 100 matching pairs designed to augment the matching samples dataset. 7964 pairs of writing samples came from two different individuals while 171 were written by the same individual.

For each pair of images in my 8135-pair sample size, I processed the image using the OpenCV Python library. Images were read in grayscale because they were scanned in the original dataset, eliminating the importance of ink color and paper type. This choice reasserts the goal to have a model that works well for digitized handwriting, rather than hard copies. Therefore, pixel values fell between 0 and 1, inclusive.

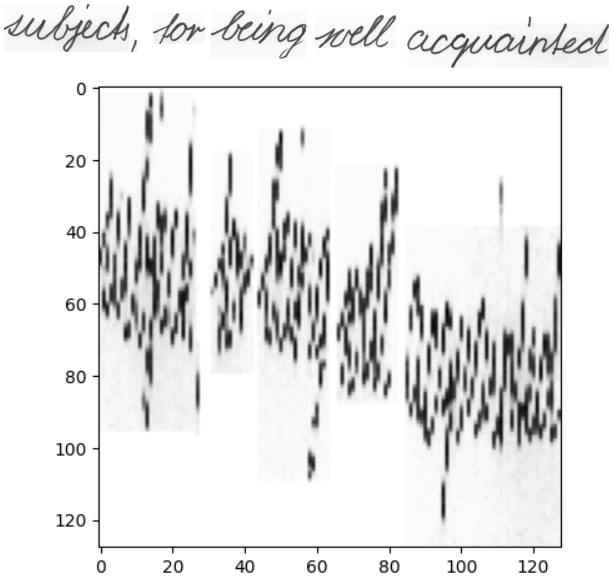

Figure 1: *Example of a handwritten line before (top) and after (bottom) additional processing. Original text reads: "subjects, for being well acquainted"*

3.3 Building a Convolutional Neural Network

The convolutional neural network was designed to take in 128x128 grayscale images. It contains two convolutional layers with a ReLU activation function and two max pooling layers. The output involves the use of a sigmoid activation function for the sake of binary classification. Finally, the model contains an Adam optimizer, which was selected for its memory efficiency and its combination of the advantages of both AdaGrad and RMSProp (Kingma and Ba, 2014). The convolutional neural network contains n = 7,391,873 total parameters.

The train-test split was 80-20, respectively, and the model was fitted with ten epochs and a batch size of 32. Aside from being selected without replacement, the train-test data was otherwise randomly sorted into the train and test categories. Finally, training loss, validation loss, true positive, false positive, true negative, and false negative counts were calculated, along with corresponding rates.

3.4 Code availability

Code link: https://github.com/valcsuremm/GU-5423

4 Results

4.1 Dataset Features

The images used in this research had a mean of 7.548 words per line (standard deviation = 2.125).

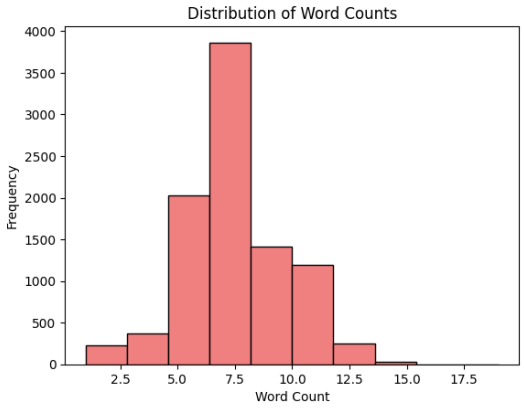

Figure 2: *Histogram displaying the distribution of the word count for the n = 9386 images that were used in the study (duplicate file paths removed)*

The number of characters per line had a mean of 42.729 and a standard deviation of 10.4339. The associated histograms have a rather sizeable spread. This suggests that the model will also gain information about the number of strokes (a larger number of pixel values close or equal to 1 will indicate more strokes in a line).

4.2 Model performance

Over the course of the ten epochs, the loss for training data fell while the validation loss rose slightly, from 0.1296 to 0.1490. Though there is some discrepancy in the loss values between training and validation data, the accuracy for both training and validation sets remains consistently above .97 (1592 of 1627 total predictions were correct with a threshold value of .5). The number of predictions that fell at a value greater than the threshold of .05 = 19 and the count of predictions $\leq .5 = 1608$

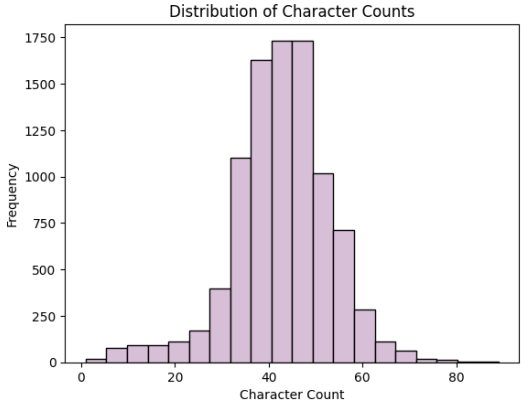

Figure 3: *Histogram displaying the distribution of the character counts for the n = 9386 images that were used in the study (duplicate file paths removed)*

The calculated true positive rate, also known as recall, is 0.409 while the true negative rate (specificity) is.994. The F1-Score is 0.5070, the area under the curve (AUC) value is 0.674 (Pedregosa et al., 2011). Despite its relatively low value, the f-score is likely one of the most important scores in determining the algorithm performance; this is especially true due to the risks of misleadingly high AUC value when datasets are imbalanced (Zou et al., 2016).

Metric	Non-match, 0	Match, 1	Overall
Precision	0.98	0.67	.667
Recall	.99	.41	.409
f1-score	.99	0.51	.5070
Support	1583	44	1627

5 Conclusion

At face value, the results of the convolutional neural network are promising due to the high level of accuracy. However, the considerably low F1-score and recall both indicate concerns of overfitting, which may have resulted from the disproportionately large number of non-matches versus matches in the pairs dataset. If input data were to include a significantly higher proportion of matched pairs, there might be an excess of false negatives, thereby decreasing the reliability of the model.

When compared to an existing study with a smaller sample size, the overall precision and recall scores in this study show drastic under-performance; one particular study had an overall F1-score of .90 compared to this study's .51 (Abiodun et al., 2024). Ultimately, it appears as though the convolutional neural network was inclined to predict non-matches, which was typically in accordance with the labels, since the vast majority of the dataset had non-matching individuals write the samples.

It would be remiss to omit the fact that, while the original dataset was designed to represent unconstrained handwriting, there were specific guidelines set in place which were designed to improve the ease of processing (as explained in Section 3.1). An improved model might eliminate constraints such as the use of a ruler or guidelines, and might ask the participants to free-write, rather than copying sentences of text.

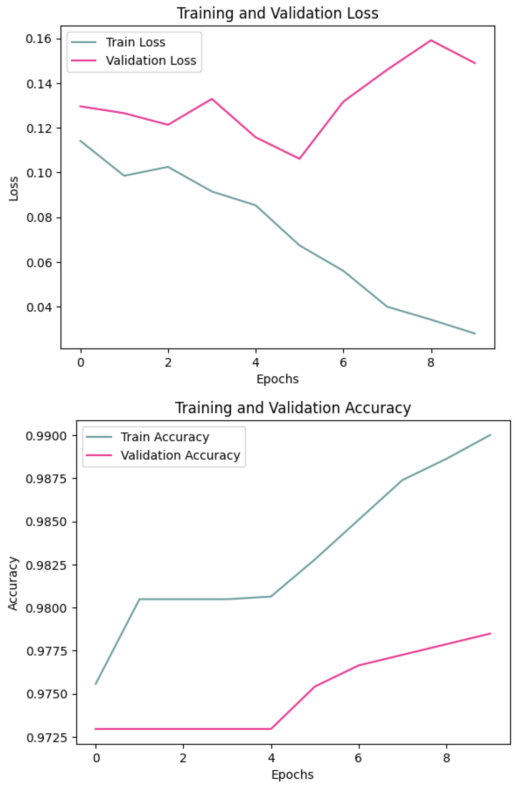

Figure 4: *Loss (top) and accuracy (bottom) over ten epochs. The model performs a bit better on its training data than the validation data as anticipated.*

In comparison to the FBI Laboratory Division, this model might be seen to underperform current experts (performance described in Section 2), and improvements would be necessary if a model such as this were used to guide investigations.

In the field of handwriting analysis, especially in an investigatory setting, future samples' metadata could be improved by including distinguishable features, namely handwriting complexity, which includes features such as the number of pen lifts per unit line length and the presence of feathering of the line, which indicates changes in pressure while writing (Bird and Jones, 2024). Moreover, an ethical tool used for investigation or trial evidence should have an extremely low false positive rate with the possibility of an "inconclusive" outcome, rather than the binary offered in this particular paper.

Another major application of this type of research is handwriting detection for security purposes. Signatures or documents of concern in fraud cases are potential uses of a model such as this one after improvement. A model for this purpose might be trained on signatures and names specifically, rather than sentences from a corpus as in this study. In addition, recognition of a specific writer might

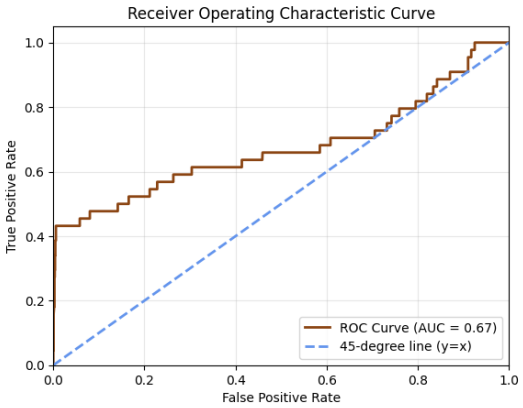

Figure 5: *Receiver operating curve (x-axis = False Positive Rate and y-axis = True Positive Rate)*

prove useful to the development of accessibility tools for individuals with lifelong visual disabilities or age-related impairments. In the context of document processing and historical artifacts, writer detection holds promise with identifying the writer of unsigned documents and letters for the sake of museum and academic collections.

6 References

Abadi, M., & TensorFlow, A. A. B. P. (2016, November). Large-scale machine learning on heterogeneous distributed systems. In Proceedings of the 12th USENIX symposium on operating systems design and implementation (OSDI'16)(Savannah, GA, USA (pp. 265-283).

Abiodun, A. O., Awoniran, O. M., Ukhurebor, K. E., Emuoyibofarhe, O. N., Adeyemo, A., & Oaihimire, I. E. (2024). Original Research Article A deep learning approach for forensic handwriting analysis. Journal of Autonomous Intelligence, 7(5).

Bird, C., & Jones, K. (2024). Complexity, Features, and Comparisons in Forensic Handwriting Examination. Wiley Interdisciplinary Reviews: Forensic Science, e1537.

Bradski, G. (2000). The OpenCV library. Dr. Dobb's Journal of Software Tools.

Durina, M. (2021). 90 Years Later: Revisting the Lindbergh Kidnapping Case. Journal of the American Society of Questioned Document Examiners, 24(2).

Hicklin, R. A., Eisenhart, L., Richetelli, N., Miller, M. D., Belcastro, P., Burkes, T. M., ... & Eckenrode, B. A. (2022). Accuracy and reliability of forensic handwriting comparisons. Proceedings of the National Academy of Sciences, 119(32), e2119 944119.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. Computing in science & engineering, 9(03), 90-95.

Kingma, D. P. and Ba, J (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

U. Marti and H. Bunke. The IAM-database: An English Sentence Database for Off-line Handwriting Recognition. Int. Journal on Document Analysis and Recognition, Volume 5, pages 39 - 46, 2002.

National Research Council, Strengthening Forensic Science in the United States: A Path Forward (The National Academies Press, 2009).

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825–2830.

United States Postal Inspection Service, United States Government. (n.d.-b). Handwriting Analysis. United States Postal Inspection Service.

Xml.etree.ElementTree - The Elementtree XML API. Python documentation. (n.d.). https://docs.python.org/3/library/xml.etree.elementtree.html

Zou, Q., Xie, S., Lin, Z., Wu, M., & Ju, Y. (2016). Finding the best classification threshold in imbalanced classification. Big Data Research, 5, 2-8.

Table 1 The MI-CLAIM checklist

From: Minimum information about clinical artificial intelligence modeling: the MI-CLAIM checklist

Before paper submission			
Study design (Part 1)	**Completed: page number**		**Notes if not completed**
The clinical problem in which the model will be employed is clearly detailed in the paper.	☑	1	
The research question is clearly stated.	☑	1	
The characteristics of the cohorts (training and test sets) are detailed in the text.	☑	1-2	
The cohorts (training and test sets) are shown to be representative of real-world clinical settings.	☐		not clinical; would depend on # of comparisons in an investigation
The state-of-the-art solution used as a baseline for comparison has been identified and detailed.	☑	2	
Data and optimization (Parts 2, 3)	**Completed: page number**		**Notes if not completed**
The origin of the data is described and the original format is detailed in the paper.	☑	1	
Transformations of the data before it is applied to the proposed model are described.	☑	1-2	
The independence between training and test sets has been proven in the paper.	☑	2	
Details on the models that were evaluated and the code developed to select the best model are provided.	☐		only 1 model
Is the input data type structured or unstructured?	☑ Structured ☐ Unstructured		
Model performance (Part 4)	**Completed: page number**		**Notes if not completed**
The primary metric selected to evaluate algorithm performance (e.g., AUC, F-score, etc.), including the justification for selection, has been clearly stated.	☑	3	
The primary metric selected to evaluate the clinical utility of the model (e.g., PPV, NNT, etc.), including the justification for selection, has been clearly stated.	☑	3	
The performance comparison between baseline and proposed model is presented with the appropriate statistical significance.	☐		did not use a baseline model
Model examination (Part 5)	**Completed: page number**		**Notes if not completed**
Examination technique 1[a]	☐		?
Examination technique 2[a]	☐		?
A discussion of the relevance of the examination results with respect to model/algorithm performance is presented.	☑	2-3	
A discussion of the feasibility and significance of model interpretability at the case level if examination methods are uninterpretable is presented.	☑	3	
A discussion of the reliability and robustness of the model as the underlying data distribution shifts is included.	☑	3	
Reproducibility (Part 6): choose appropriate tier of transparency			**Notes**
Tier 1: complete sharing of the code		☑	Colab
Tier 2: allow a third party to evaluate the code for accuracy/fairness; share the results of this evaluation		☐	
Tier 3: release of a virtual machine (binary) for running the code on new data without sharing its details		☐	
Tier 4: no sharing		☐	