THE UNIVERSITY OF SOUTH ALABAMA

COLLEGE OF ARTS AND SCIENCES

# Multi-Script Handwriting Identification using Stroke Decomposition

BY

Joshua Jude Thomas

A Thesis Proposal

Submitted to the Graduate Faculty of the

University of South Alabama

in partial fulfillment of the

requirements for the degree of

Master of Science

in

Electrical and Computer Engineering

April 2023

Approved: Date:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Chair of Thesis Committee: Dr. First Name, Middle Initial, Last Name

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Committee Member: Dr. First Name, Middle Initial, Last Name

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Committee Member: Dr. First Name, Middle Initial, Last Name

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Chair of Department: Dr. First Name, Middle Initial, Last Name

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Director of Graduate Studies: Dr. First Name, Middle Initial, Last Name

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Dean of the Graduate School: Dr. J. Harold Pardue

# Multi-Script Handwriting Identification using Stroke Decomposition

A Thesis

Submitted to the Graduate Faculty of the

University of South Alabama

in partial fulfillment of the

requirements for the degree of

Master of Science

in

Electrical and Computer Engineering

by

Joshua Jude Thomas

B.S., University of South Alabama, 2021

A.S., Coastal Alabama Community College, 2017

April 2023

**LIST OF ABBREVIATIONS**

SVM Support Vector Machine

LLR Log Likelihood Ratio

CNN Convolutional Neural Network

HIM Handwriting Identification Model

HVM Handwriting Verification Model

SD Stroke Decomposition

# ABSTRACT

Joshua Jude Thomas, M. S., University of South Alabama, April 2023. Multi-Script Handwriting Identification using Handwritten Stroke Decomposition Chair of Committee: Ryan Benton, Ph.D.

This study will implement and test the effectiveness of adding a preprocessing step, named **Stroke Decomposition**, to a typical Handwriting Identification model. Stroke Decomposition works by approximating the different writing strokes that make up the characters in a handwritten document; The idea of this study is that the characters of different writing scripts (English, Arabic, Mandarin, etc.) have common strokes which can be extracted and used in the handwriting identification process. This study tests this idea in the **Multi-Script Writer Identification** problem domain, which tries to identify the writer of a questioned document in a setting where a writer can possibly write in multiple writing scripts. The Stroke Decomposition step will be added to two different handwriting identification models (using the **Handwriting Identification Model** (HIM) and the **Handwriting Verification Model** (HVM)) and the performance will be measured on various multi-script and single-script handwriting datasets.

# CHAPTER I

**Introduction**

## What is Handwriting Identification?

**Handwriting Identification** is the process of classifying the writer of a handwritten Questioned Document based on handwriting habits contained in that document. Handwriting Identification predates computational processing by a fair bit. Evolving from “recognition witnesses” which was simply a person familiar with an accused persons handwriting, Handwriting Identification has been used in court for centuries [1]. According Harralson and Miller in *Huber and Headrick’s Handwriting Identification: Facts and Fundamentals* forensics experts compare twenty one “discriminating elements of handwriting” that deal with properties such as word size, word placement, margin sizes, abbreviation choices, etc. [1].

Computationally assisted Handwriting Identification is typically done by extracting visual features from a set of known documents and questioned documents. Those features are usually aggregated into a **feature vector** and are then used to compare the known documents to the questioned documents, and the writer is predicted by a similarity score or a likelihood [2].

### Verification and Identification Model

There are two main architectures used in computationally assisted Handwriting Identification: the **Handwriting Identification Model** and the **Handwriting Verification Model** [2].

**The Handwriting Verification Model** estimates the likelihood that two documents were written by the same person (two-class classification) [2]. The extracted feature vectors of a document are passed to a decision function that can either directly classify the writer such as with a Support Vector Machine (SVM) [3], or output a similarity/difference score such as the Log Likelihood Ratio (LLR) [4].

Graphical user interface

Description automatically generated with medium confidence

Figure 1. The Handwriting Verification Model

The **Handwriting Identification Model**, on the other hand, matches a Questioned Document to a database of known writers (n-class classification). Here the extracted features of a document are typically a multi-class classifier such as Nearest Neighbor [5], [6]. The Writer Identification Model seems to be the more popular model overall.

Diagram

Description automatically generated with medium confidence

Figure 2. The Handwriting Identification Model

## Multi-Script Writer Identification

The Multi-Script Writer Identification problem is a similar to normal Handwriting Identification, except that a writer is not limited to a single writing script, but can write in multiple scripts such as English, Arabic, Mandarin, and so on [7]. The ICFHR 2018 Competition on Multi-Script Writer Identification [7] is a popular reference in papers on Multi-Script Writer Identification [6]. The competition paper uses the CERUG [8], LAMIS-MSHD [9], and the WDAD [7] datasets: which contain Chinese/English, Arabic/French, and Farsi (Persian)/English Multi-Script datasets, respectively. There are six tasks introduced by the competition, each of which uses one of the databases (above). Each task has the Writer Identification system train on one language in the dataset, and test on the other. The goal of Multi-Script Writer Identification, according to the competition paper [7], is to find “… writing patterns that are common across different scripts [and] may be exploited to identify the writer”. This problem is based on the assumption that there are ingrained patterns in a person’s handwriting that are stable across different writing scripts [7].

## Stroke Decomposition Algorithm

The purpose of the **Stroke Decomposition** (SD) Preprocessing algorithm, which will be developed in this study, is to improve the performance of Multi-Script Writer Identification. The idea of this study is that the SD preprocessing step will transform the, often radically different, handwriting characters into simpler sub-strokes that will be more comparable. Figure 3 demonstrates the importance of this idea; The characters of the four different writing scripts can look very different from one another and may confuse some of the feature extraction algorithms due to the overall visual differences in the characters.

A picture containing text, document

Description automatically generated

Figure 3. (From left to right) Chinese, Bengali, Tamil, and English. Writing Scripts of three different origins that look very different on a document-level scale.

The SD algorithm is based on the concepts of Morphological Processing and Graph Theory. With Morphological Processing a binary (can be grayscale with grayscale morphology) image is processed via spatial filtering with a Structuring Element SE to identify shapes an patterns in the image, to perform preprocessing and post processing on the image, to find the boundary or convex hull of an image, and so on [10]. Morphological Processing is used to remove noise and unnecessary components of document images as well as to preprocess the document images for graph theory processing. Graph Theory is the study of graph data structures, which can be described as a set of observations (vertices) and their relation to one another (edges). Here a processed document image, binarized and thinned by morphology, is converted into a graph representation and

# CHAPTER II

**Review of Literature in Multi-Script Handwriting Identification**

The ICFHR 2018 Multi-Script Handwriting Identification competition is a popular reference point for more recent literature. However, the competition paper itself reports on the successes of four different systems submitted to the competition. These systems are the LIMPAF-I, LIMPAF-II, Tokyo System, and the Nuremberg System [7]. The LIMPAF-I and LIMPAF-II were submitted by the same group; LIMPAF-I uses Uniform Complete Local Binary Patterns (U-LBP) [11] for its feature extraction while the LIMPAF-II uses Oriented Basic Image Features (oBIF) [12]. For classification, both systems used a multi-class SVM. The Tokyo system used two CNNs to extract features from randomly selected sub-images of a writing sample. Features extracted from writing samples were passed into a “Transfer Neural Net” to transform the extracted features, of writing samples of possibly different writing scripts, into a more uniform representation. These transformed features were then finally classified by K-Nearest Neighbors [13]. The Nuremberg system was actually based on another work ([14]) which extracted features by a pre-trained CNN. The extracted features were then “PCA-Whitened” and encoded in a visual bag of words algorithm called VLAD. The feature vectors of the Nuremberg systems were classified by measuring the Cosine-Distance between each sample.

Abbas *et al* combines both LBP and oBIF, creating a histogram out of the LBP and oBIF descriptors over the whole range of the document and then classifying the writer with an multi-class SVM [15]. Semma, Hannad, Siddiqui, Lazrak, and Kettani extract features from sub-images obtained from a Harris Corner Detector. The sub-images are then fed into a CNN for feature extraction, and then transformed with VLAD. The VLAD vectors are then used to classify the writer with Nearest Neighbor [16].

# CHAPTER III

**Methodology of Study**

## Sub-Stroke Decomposition

In much of the literature, the input into the feature extraction algorithms are random sub-images of the documents (like the Tokyo system in the competition paper) [7], using key point detectors such as the Harris Corner Detector [16] as guides. One study aggregate local features generated from image patches into global features through processes like max-pooling and average-pooling [13].

This study plans to test the effect of separating the characters of a handwritten document into the individual sub-strokes that make them up; With the idea that breaking the characters of a document into sub-strokes will reduce the statistical noise, produced from the document-level differences in characters of different writing scripts, in the document and make writer classification between different writing scripts easier. Since there is no information about how the writer wrote the characters during the making of a questioned document, the sub-strokes making up a character must be approximated. There is already some literature in sub-stroke decomposition. Kim, Kim, Choi, and Kim decompose Chinese character into their constituent sub-strokes using morphological processing [17]. However, to the best of our knowledge, there is no readily available software package that performs Sub-Stroke Decomposition. So, most of the time of this study will be spent developing the Sub-Stroke Decomposition algorithm.

The implementation details of the Sub-Stroke Decomposition algorithm are not yet known, but some assumptions can be made. Given a handwritten document, the character contained in that document can be decomposed into one or more sub-strokes. These sub-strokes can then be isolated for analysis and common strokes between languages can be found via a clustering algorithm (figure 4). It’s worth noting that since we do not actually know how exactly a writer produces a stroke (a handwriting stroke can be produced in many ways) we must approximate the sub-strokes. Furthermore, some of the sub-strokes may have to be broken up such as in cases like cursive writing, where an entire word can be produced with one stroke.

Once the sub-strokes making up a handwritten image have been extracted the visual features of the sub-strokes can be extracted exactly like any methods above. In fact, the extracted sub-strokes could be considered sub-images of the handwritten document themselves except that the other sub-strokes, either connected or nearby, are filtered out.

Diagram

Description automatically generated

Figure 4. General Idea of Stroke Decomposition and Clustering. (from top to bottom) Mandarin (Chinese handwriting), English, and Bengali. Note that the Bengali character was produced entirely with one stroke, and thus may have to be broken up in the actual stroke decomposition algorithm.

### Methods of Sub-Stroke Comparison

This study will test two main ways of comparing extracted sub-strokes. The first way would be based on a sub-stroke matching process where similar sub-strokes across handwritten documents are found and compared. The other way is to train a clustering algorithm on a large dataset of handwritten documents written in multiple writing scripts and languages; The N most common sub-strokes can be found and subsequently extracted from each document. The result of this method would be that each document is reduced to common between multiple writing scripts, thus extracting the features from only those sub-strokes.

## Stroke-Decomposition in View of the Overall Model

This study will test the effectiveness of Stroke Decomposition by using the Handwriting Identification Model, as well as using the prepared dataset by the ICHFR competition testing methods (Chapter V). Stroke Decomposition will be used as a preprocessing step to prepare the handwritten documents prior to the actual feature extraction and classification, and common sub-stroke clusters will be found prior to training.

### General Process

a handwritten document, converted into a digital image either through scanning or by taking a picture, will be first be binarized to produce an image of either background (false) or foreground elements (the handwriting itself, true). Following that will be various preprocessing steps to reduce any noise present in the image such as small patches of foreground (likely being unintended strokes of the writing tool) and large blobs of foreground (likely being drawing or some texturing unimportant to the handwriting identification process). Once the document is sufficiently prepared then the Stroke Decomposition can be performed, reducing the document into a series of sub-images containing its individual sub-strokes. After the sub-strokes are extracted then the Handwriting Identification Model can perform as usual. The sub-strokes go through any remaining pre-processing and are finally processed by the chosen feature extraction algorithm.

The feature extraction algorithm chosen for this study is a headless CNN. CNN is a popular choice of feature extraction algorithm in Handwriting Identification. This study will experiment with both CNN models made from scratch, and with pre-trained models. Feature Vectors will be extracted from the CNN and then the writer(s) will be classified via K-Nearest Neighbors.

|  |
| --- |
| Handwriting Identification Process (Overview) |
| Step 1: Binarization  Step 2: Artefact Removal (noise, small textures in the document, and ink-blobs)  Step 3: Stroke Decomposition and Clustering  Step 4: Feature Extraction of processed Sub-Strokes via CNN  Step 5: Classification via K-Nearest Neighbors |

Table 1. Simplified View of this Studies Handwriting Identification Process

# CHAPTER IV

**Results**

## Hypothesis of the Study

The main hypothesis of this study is that there exist common writing strokes in all writing scripts that can be isolated to improve the performance of handwriting identification when dealing with multiple writing scripts or an unknown writing script. The main idea behind this is that machine learning methods such as CNN, which are predominantly used today for the task of Handwriting Identification, can become confused when new incoming data looks different from what the training data looks like. Decomposing a handwritten character into a set of sub-strokes would, in theory, remove the large visual difference from handwriting characters in different writing scripts. The result is that many unique handwriting characters should be able to be broken down into multiple, more comparable, handwriting strokes.

## Potential Issues

Since the goal of Stroke Decomposition is to remove the large visual difference from handwriting character by separating the individual sub-strokes, there may be some loss of information. In practice, nearby handwriting strokes may have an influence on each other. The presence of a handwriting stroke may cause the author of the document to screw, or compress, any subsequent strokes. Some handwriting strokes may also be super imposed on one another. Since the sub-strokes making up a handwritten character may have an affect on one another, decomposing and analyzing them separately may result in the loss of useful information and negatively impact the classification performance of the overall model.

# CHAPTER V

**Anticipated Method of Analysis**

The ICFHR 2018 Multi-Script Handwriting Identification competition [7] will be used as a reference for testing the performance difference when adding the Stroke Decomposition step. There are six main classification tasks in the competition. For every two tasks, a Multi-Script dataset is chosen and a Handwriting identification model trains on one writing script, and then trains on the other writing script. For task 1 and 2, the CERUG [8] dataset is used. Task 1 has the model train on Chinese and test on English. Task 2 has the model train on English and test on Chinese. For task 3 and 4, the LAMIS-MSHD [9] dataset is used. Task 3 has the model train on Arabic and test on French. Task 4 has the model train on French and test on Arabic. For task 5 and 6, the WDAD [7] is used. Task 5 has the model train on Farsi and test in English. Task 6 has the models train in English and test on Farsi. The resulting performance metrics from each of these tasks are presented separately.

# CHAPTER VI

**Conclusion**

Multi-Script Handwriting Identification attempts to classify the writer of a handwritten document in a setting where there can be multiple writing scripts or languages in use, and with the possibility that a writer can create documents in more than one writing script. As such, Multi-Script Handwriting Analysis seeks common features between the different writing scripts that are both effective and consistent. Stroke Decomposition may provide a more robust comparison between characters by isolating common sub-strokes found in all, or many, writing scripts. While the use of Stroke Decomposition may result in a small performance penalty due to the loss of information about how the sub-strokes affect each other, it may provide more comparable handwriting samples between questioned documents written in different writing scripts or languages. This will hopefully result in a net positive impact on the performance of Multi-Script Handwriting Identification.

# REFERENCES

[1] H. Harralson and L. Miller, *Huber and Headrick’s Handwriting Identification: Facts and Fundamentals*, 2nd ed. CRC PRess, 2017.

[2] S. N. Srihari, “Individuality of Handwriting,” *Journal of Forensic Science*, vol. 47, no. 4, p. 17, Jul. 2002.

[3] A. Foroozandeh, A. Askari Hemmat, and H. Rabbani, “Offline Handwritten Signature Verification and Recognition Based on Deep Transfer Learning,” in *2020 International Conference on Machine Vision and Image Processing (MVIP)*, Feb. 2020, pp. 1–7. doi: 10.1109/MVIP49855.2020.9187481.

[4] M. A. Shaikh, M. Chauhan, J. Chu, and S. Srihari, “Hybrid Feature Learning for Handwriting Verification,” Aug. 2018, pp. 187–192. doi: 10.1109/ICFHR-2018.2018.00041.

[5] B. V. Dhandra and M. B. Vijayalaxmi, “A Novel Approach to Text Dependent Writer Identification of Kannada Handwriting,” *Procedia Computer Science*, vol. 49, pp. 33–41, Jan. 2015, doi: 10.1016/j.procs.2015.04.224.

[6] X. Wu, Y. Tang, and W. Bu, “Offline Text-Independent Writer Identification Based on Scale Invariant Feature Transform,” *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 3, pp. 526–536, Mar. 2014, doi: 10.1109/TIFS.2014.2301274.

[7] C. Djeddi, S. Al-Maadeed, I. Siddiqi, G. Abdeljalil, S. He, and Y. Akbari, “ICFHR 2018 Competition on Multi-Script Writer Identification,” in *2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Aug. 2018, pp. 506–510. doi: 10.1109/ICFHR-2018.2018.00094.

[8] S. He, M. Wiering, and L. Schomaker, “Junction detection in handwritten documents and its application to writer identification,” *Pattern Recognition*, vol. 48, no. 12, pp. 4036–4048, Dec. 2015, doi: 10.1016/j.patcog.2015.05.022.

[9] C. Djeddi, A. Gattal, L. Souici-Meslati, I. Siddiqi, Y. Chibani, and H. El Abed, “LAMIS-MSHD: A Multi-script Offline Handwriting Database,” in *2014 14th International Conference on Frontiers in Handwriting Recognition*, Sep. 2014, pp. 93–97. doi: 10.1109/ICFHR.2014.23.

[10] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd edition. Upper Saddle River, NJ: Pearson, 2007.

[11] Z. Guo, L. Zhang, and D. Zhang, “A Completed Modeling of Local Binary Pattern Operator for Texture Classification,” *IEEE Transactions on Image Processing*, vol. 19, no. 6, pp. 1657–1663, Jun. 2010, doi: 10.1109/TIP.2010.2044957.

[12] A. J. Newell and L. D. Griffin, “Writer identification using oriented Basic Image Features and the Delta encoding,” *Pattern Recognition*, vol. 47, no. 6, pp. 2255–2265, Jun. 2014, doi: 10.1016/j.patcog.2013.11.029.

[13] H. T. Nguyen, C. T. Nguyen, T. Ino, B. Indurkhya, and M. Nakagawa, “Text-independent writer identification using convolutional neural network,” *Pattern Recognition Letters*, vol. 121, pp. 104–112, Apr. 2019, doi: 10.1016/j.patrec.2018.07.022.

[14] V. Christlein and A. Maier, “Encoding CNN Activations for Writer Recognition,” in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, Apr. 2018, pp. 169–174. doi: 10.1109/DAS.2018.9.

[15] F. Abbas, A. Gattal, C. Djeddi, I. Siddiqi, A. Bensefia, and K. Saoudi, “Texture feature column scheme for single- and multi-script writer identification,” *IET Biometrics*, vol. 10, no. 2, pp. 179–193, 2021, doi: 10.1049/bme2.12010.

[16] A. Semma, Y. Hannad, I. Siddiqi, S. Lazrak, and M. E. Y. E. Kettani, “Feature learning and encoding for multi-script writer identification,” *IJDAR*, vol. 25, no. 2, pp. 79–93, Jun. 2022, doi: 10.1007/s10032-022-00394-8.

[17] J. W. Kim, K. I. Kim, B. J. Choi, and H. J. Kim, “Decomposition of Chinese character into strokes using mathematical morphology,” *Pattern Recognition Letters*, vol. 20, no. 3, pp. 285–292, Mar. 1999, doi: 10.1016/S0167-8655(98)00147-0.