

---

# Implementation of Different Machine Learning Techniques to Compute Handwriting Similarity

---

**Arun Krishnamurthy**  
Department of Computer Science  
University at Buffalo  
Buffalo, NY 14214  
arunkris@buffalo.edu

**Ashwin Vijayakumar**  
Department of Computer Science  
University at Buffalo  
Buffalo, NY 14214  
ashwinvi@buffalo.edu

**Srivatsa Hegde**  
Department of Computer Science  
University at Buffalo  
Buffalo, NY 14214  
shegde@buffalo.edu

## Abstract

In this work, we present a comparative review of applying different machine learning approaches to the problem of handwriting similarity. We build a probabilistic graphical model and deep learning models, and as well as perform simple machine learning using an existing tool [1]. The dataset used for this purpose is the AND handwriting dataset, described in [2], which consists of multiple samples of the word 'and' written by various authors.

## 1 Introduction

Handwriting examination is based on the premise that each person's handwriting has its own individual characteristics. Human experts are able to break down a handwriting sample and specify the unique characteristics of the handwriting style. This is done with the idea that given two handwriting samples, one must be able to say with some degree of certainty if they were written by the same person or not [2]. There have been several approaches to perform this task computationally. In this work, we will take a look at three different approaches, namely, probabilistic graphical models, simple machine learning models, and deep learning models.

## 2 Problem Description

In this project, we will be looking at three different approaches to building a model that can look at two samples of a handwritten word and say whether or not they were written by the same author. To this end, we will build two models: a probabilistic model, and a deep learning model. We then perform simple machine learning analysis using the CEDAR-FOX software tool.

## 3 Dataset

The dataset used in this work consists of multiple samples of the word 'and' written by various authors. There are two sections of this dataset. The first one is a dataset with features. This has 3524 images of ANDs and there are 9 human observed features. This dataset is used for the PGM approach. The second part of this dataset is just without features. It's just a directory with 15518

images of ANDs by roughly 1500 authors. This is used for the deep learning approach. It requires some pre-processing to be useful; this part is explained in the deep learning section.

## 4 Methodologies

### 4.1 Probabilistic Graphical Models

The PGM given below has been constructed using the BNST algorithm as explained in the NCJRS report “Statistical Examination of the Characteristics of Handwriting” [5]

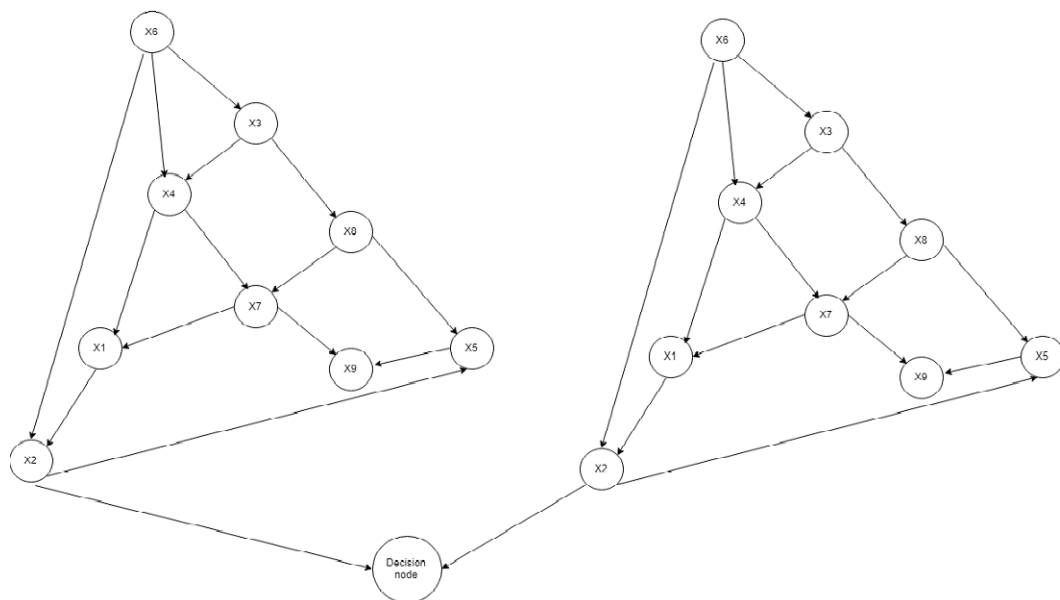


Figure 1: PGM

#### 4.1.1 Working

The 2 images are fed as input to the 2 branches of the PGM(with the same structure), which are connected by a decision node that would output a value between 0 and 1. A value from 0 indicates the 2 images are dissimilar and a 1 indicates the 2 images are exactly similar.

#### 4.1.2 Multinet

The image features were used to create a multinet structure that was later used to train the Bayesian Model and subsequently test the model to determine its correctness. The data for the multinet has 18 features ( 9 each from 2 images) and 1 decision node that would take a binary value, 0 if images are from different authors and 1 if the images are from the same author.

A sample of 10000 entries from a million possible entries were used to train the Bayesian model. The Bayesian model was tested on 870 images.

#### 4.1.3 Testing and Accuracy Considerations

Inference is done using the map-query function. 9 features of the test data set(essentially an image) is used along with the similarity feature to predict the values of the 9 features of a matching image. This is compared against the actual values of the matching images in the test data set.

To predict accuracy we have devised our own method. We create a dataframe of true false values based on this idea. If the predicted value matches the actual value of the test data, we assign a true to

the position, if not we assign a false value. If more than half of the values in a row are true we predict that the 2 images are similar else they are different.

The CPDs obtained.

## 4.2 Probabilistic Graphical Models

Here are the CPDs obtained:

CPD table for node f2	CPD table for node f1	CPD table for node f6
f2	f1	f6
phi(f2)	phi(f1)	phi(f6)
f2_0	f1_0	f6_0
0.0315	0.0715	0.0001
f2_1	f1_1	f6_1
0.8959	0.2599	0.1000
f2_2	f1_2	f6_2
0.0205	0.4659	0.8698
f2_3	f1_3	f6_3
0.0425	0.2008	0.0300
f2_4	f1_4	f6_4
0.0096	0.0019	0.0001
CPD table for node f7	CPD table for node f9	CPD table for node f3
f7	f9	f3
phi(f7)	phi(f9)	phi(f3)
f7_0	f9_0	f3_0
0.6774	0.0078	0.0001
f7_1	f9_1	f3_1
0.3161	0.0265	0.9998
f7_2	f9_2	f3_2
0.0065	0.9575	0.0001
CPD table for node f8	CPD table for node f5	CPD table for node f4
f8	f5	f4
phi(f8)	phi(f5)	phi(f4)
f8_0	f5_0	f4_0
0.2200	0.0152	0.4600
f8_1	f5_1	f4_1
0.2300	0.0152	0.1500
f8_2	f5_2	f4_2
0.2800	0.9543	0.0100
f8_3	f5_3	f4_3
0.1900	0.0152	0.3800
f8_4		f4_4
0.0800		0.0001
f8_5		
0.0001		

Figure 2: CPDs

## 4.3 CEDAR-FOX

CEDAR-FOX is a software tool used for forensic comparison of handwriting. It was developed at CEDAR, the Center of Excellence for Document Analysis and Recognition at the University at Buffalo. CEDAR-FOX has capabilities for interaction with the questioned document examiner to go through processing steps such as extracting regions of interest from a scanned document, determining lines and words of text, recognize textual elements. The final goal is to compare two samples of writing to determine the log-likelihood ratio under the prosecution and defense hypotheses. It can also be used to compare signature samples [1] [3].

We used the batch verification part of this tool. This accepts a batch file of pairs of images and gives out a result. A positive final score indicates same writer while a negative score means different writers. [4]

Based on a rudimentary exploration of this tool, we realized that batch verification does not work beyond 3 pairs in a batch. In absence of proper documentation or application support, we decided to automate whatever parts of this verification process that we could in order to process as many files as we could and minimize manual effort. So, we wrote a python function to create multiple batch files of 3 pairs each. We then took these batch files and ran it through the software tool. We could not automate this part of the process because there was no API to communicate with the software. Once the output files were produced through the tool, we had another custom python function that found and read all the output files, and computed an accuracy percent.

#### 4.4 Deep Learning

#### 4.5 Dataset Preprocessing

The training data set we use for the deep learning task consists of a set of images that are named according to the author's of each individual image.

We first categorize the images based on author names into separate folders. We have a total of around 1500 authors out of which about 900 were selected for training and 300 for validation and testing each.

#### 4.6 Loss function

Unlike a classification task where the network is able to classify its input into the various classes, the Siamese twin architecture is not built to learn features used for image classification [6]. Rather, the aim of the network is to identify pairs of images with high similarity and dissimilarity with a single training example of each pair, also known as one shot learning. To measure the similarity (and dissimilarity), we use the contrastive loss function [7] that takes the form -

$$(1 - Y) \frac{1}{2} (D_W)^2 + (Y) \frac{1}{2} \{ \max(0, m - D_W) \}^2$$

Figure 3: Loss

Where,  $D_W$  is the Euclidean distance given by -

$$\sqrt{\{G_W(X_1) - G_W(X_2)\}^2}$$

Figure 4: Distance

#### 4.7 Architecture

Here is the model architecture that was used.

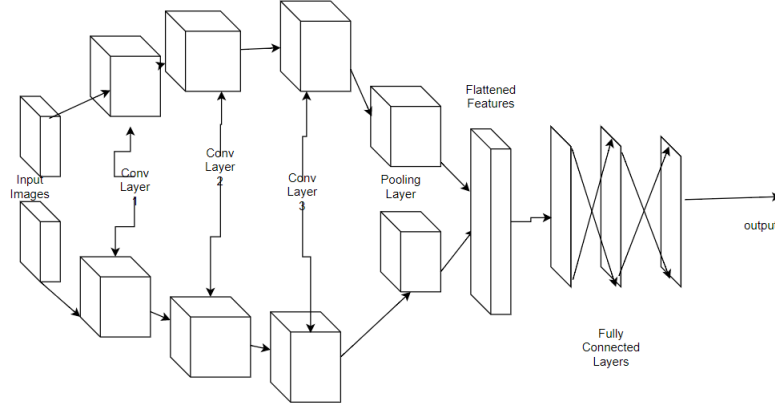


Figure 5: Model Architecture

#### 4.8 Accuracy and Dissimilarity measure

The dissimilarity measures produced by our approach are not normalized. We normalize them to fit within  $[\min/\max, 1]$  by dividing by max value with the higher values representing higher dissimilarity. Then we discretize the values by assigning 0 to the values less than the median and 1 to the values greater than the median. We then compare these values to the ground truth extracted from the data set and calculate the accuracy by the dividing the former with the latter.

#### 4.9 Hyperparameter tuning

We tuned the model using the following few hyperparameters and obtained the optimum value as -

- a) Number and size of Convolutional layers: 3 Convolutional Layers (1,4,filter=3), (4,8,filter=3), (8,8,filter=3)
- b) Number of fully connected layers: 3 Fully connected layers
- c) Learning rate: 0.001
- d) Dropout: 0.2
- e) Batch size:
- f) Number of epochs: 50

### 5 Results

Accuracy of Deep Learning Approach: 77.195

Accuracy of CEDAR FOX Approach: 81.12

Accuracy of PGM Approach: 25.14

### References

- [1] S. N. Srihari, B. Zhang, C. Tomai, S. Lee, Z. Shi, and Y. C. Shin, "A system for handwriting matching and recognition," in Proceedings of the Symposium on Document Image Understanding Technology (SDIUT 03), Greenbelt, MD, pp. 67–75, 2003.
- [2] M. Puri, S. N. Srihari, and Y. Tang, "Bayesian network structure learning and inference methods for handwriting," in Document Analysis and Recognition (ICDAR), 2013 12th International Conference on. IEEE, 2013, pp. 1320–1324.
- [3] CEDAR FOX: <https://en.wikipedia.org/wiki/CEDAR-FOX>

- [4] CEDAR-FOX: User Manual
- [5] NCJRS: <https://www.ncjrs.gov/App/Publications/abstract.aspx?ID=263834>
- [6] Koch, G., Zemel, R. S., & Salakhutdinov, R. (2015). Siamese neural networks for one-shot image recognition. In ICML Deep Learning Workshop.
- [7] Hadsell, R., Chopra, S., and LeCun, Y. (2006). Dimensionality reduction by learning an invariant mapping. In Proc. Computer Vision and Pattern Recognition Conference (CVPR'06). IEEE Press