

Writer Identification using a Deep Neural Network

Jun Chu and Sargur Srihari

Department of Computer Science and Engineering
University at Buffalo, The State University of New York
Buffalo, NY 14269, USA
[{jchu6, srihari}@buffalo.edu](mailto:{jchu6,srihari}@buffalo.edu)

ABSTRACT

Most work on automatic writer identification relies on handwriting features defined by humans[6, 4]. These features correspond to basic units such as letters and words of text. Instead of relying on human-defined features, we consider here the determination of writing similarity using automatically determined word-level features learnt by a deep neural network. We generalize the problem of writer identification to the definition of a content-irrelevant handwriting similarity. Our method first takes whether two words were written by the same person as a discriminative label for word-level feature training. Then, based on word-level features, we define writing similarity between passages. This similarity not only shows the distinction between writing styles of different people, but also the development of style of the same person. Performance with several hidden layers in the neural network are evaluated. The method is applied to determine how a person's writing style changes with time considering a children's writing dataset. The children's handwriting data are annually collected. They were written by children of 2nd, 3rd or 4th grade. Results are given with a whole passage (50 words) of writing over one-year change. As a comparison, similar experiments on a small amount of data using conventional generative model are also given.

Keywords

Deep Neural Network, Writer Identification, Similarity Measure

1. INTRODUCTION

It has been believed that every person has its consistent writing individuality and is distinct from others. A lot of important issues are related to this topic, such as signature identification. Therefore, it is very important to work out some kind of invariance in one's writing style. A lot of previous work has achieved great success in this direction, such as [6]. Yet most of them are based on human-defined features. In this paper, we hope to make a step towards automatic Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

ICVGIP '14, December 14-18, 2014, Bangalore, India
Copyright 2014 ACM 978-1-4503-3061-9/14/12\$15.00
<http://dx.doi.org/10.1145/2683483.2683514>

handwriting feature extraction. The children's handwriting data are collected for the study of the development and evolution of individuality of handwriting. Samples are from a large number of students who are learning or have just learned writing. The data are collected annually. The collection is expected to continue for 11 years so as to follow the students until high school graduation. The students write the same paragraph each year, twice for printed ones and twice for cursive ones. Each student was asked to copy the following paragraph:

The brown fox went into the barn where he saw the black dog. After a second, the black dog saw the fox too. The brown fox was fast and quick. The black dog was not fast and he lost the fox. The fox hid in a hole and waited for the black dog to go home. After the black dog went 4 home, the fox was able to go to the hole he called home and saw all the other foxes. The other foxes were glad to see him and they all asked him to tell them about his day.

Cursive samples of the scanned handwritten paragraph are shown in Figure 2. The frequently occurring word "and" is used as a base for feature extraction. Features are collected by human examiners and input through a Truthing Tool. For each cursive sample "and", 12 features are collected as in Figure 1. The related entries of the features are used to describe children's handwriting style and the dataset is to be used by machine learning algorithms.

This kind of handcraft feature extraction has a lot of drawbacks. First, because the judgments of shapes and form are made by different human examiners. They are very subjective and unstable. In fact, we find numerous conflicts and inconsistency in the delivered feature data. The imputation of missing and inconsistent data become a big problem[1]. Second, it is costly to extract data in this way, and actually, only a little portion of the image data are extracted and labelled. Features solely from "and" are also quite inadequate. We need features about other letters and other words to portray a person's individuality. Third, these features are based on experience of teachers. However, they are not necessarily intrinsic. Sometimes we need more subtle and accurate features which could be hard to describe in a natural language.

One approach to solving this problem is to extract features from strokes and shapes automatically using human designed feature extractors. There are a lot of studies on how to extract useful features (such as using geometric properties of strokes) for writer verification. Yet, the purpose of

our method is different from some writer identification problems. Judicial officials may want very robust algorithm to tell the identity of the writer even if the writer twists and tries to deceive his or her true identity. The study of children's handwriting doesn't require the ability of withstanding this. Here we would like to study the writing habits and their way of changing. Thus, all elements related to the writing style of children are taken into consideration except only for the contents. Here we suppose the students were writing naturally without twisting.

Now, in this paper, we try to find out a way to extract information about children's **cursive** handwriting directly without the need of human examiners and also to find the suitable features itself. In comparison, our method is about using discriminative way to find out features using deep neural network (DNN) automatically without the requirement of any prior knowledge. This method has several advantages. First, we can handle and extract features from numerous data automatically without efforts in feature extractor design. It needs preprocessing such like rule line removal, cleaning, or denoising but doesn't require very accurate ones. Second, our method is robust to noise. Because the writing ability of students is limited, usually there are numerous corrections and marks. These corrections and marks are hard to clean. Third, since the method is not content sensitive, it doesn't require very accurate word or letter segmentation, which is very challenging for cursive texts[3].

Initial stroke of "a"	staff right		staff left		staff center		
Formation of "a" staff	tented		retraced		looped		no staff
Number of "n" arches	one		two		retraced		combination
Shape of "n" arches	pointed	rounded					
Location of "n" mid	above base	below base	at base				
Formation of "d" staff	tented	retraced	looped				
Formation of "d" initial	overhand	underhand	straight across				
Formation of "d" terminal	curved up	straight		curved down	d no obvious end stroke		
Symbol	unusual		symbol				
a-n relationship	a taller	and	a equal	and	a smaller		
a-d relationship	a taller	and	a equal	and	a smaller		
n-d relationship	n taller	and	n equal	and	n smaller		

Figure 1: The 12 features extracted manually from cursively written “and”.

2. WORD EXTRACTION

Children write their text according to a form shown in Figure 2. Each page is of high resolution and contains a lot of information. But they are also very noisy, full of corrections and irrelevant marks and hard to segment. In order to compare the writing style between two students, we need to first reduce the data to an acceptable scale. Thus, we first extract words from each of the passage and compare writing styles on this level. Our algorithm is very simple but is fast and works on all kinds of text images even though some of them are really written like a mess.

2.1 Rule line removal

The spatial relationships between letters and rule lines are sometimes taken advantage of. They show one aspect of a person's writing habit and personality. However, in order to simplify the word segmentation and signify the importance

Student ID Number LUR 2021 E1 11/12 Third Grade 833 (2 times cursive)

The brown fox went into the barn where he saw the black dog. After a second, the black dog saw the fox too. The brown fox was fast and quick. The black dog was not fast and he lost the fox. The fox hid in a hole and waited for the black dog to go home. After the black dog went home, the fox was able to go to the hole he called home and saw all the other foxes. The other foxes were glad to see him and they all asked him to tell them about his day.

3

Student ID Number DIK 2021 H4 11/12 Third Grade 833 (2 times cursive)

The brown fox went into the barn where he saw the black dog. After a second, the black dog saw the fox too. The brown fox was fast and quick. The black dog was not fast and he lost the fox. The fox hid in a hole and waited for the black dog to go home. After the black dog went home, the fox was able to go to the hole he called home and saw all the other foxes. The other foxes were glad to see him and they all asked him to tell them about his day.

3

Figure 2: Examples of scanned paragraphs.

box	went	into	the	and	were
he	saw	the	black	dog.	After
saw	the	black	dog	saw	the
box	too.	The	box	were	not
and	went	the	dog	saw	not
box	and	not	the box	The box	had
in	the	and	were	the	the dog
to	the	box	the	black	by
not	the	were	the	to	to
the	he	were			

The	the	brown	where	saw	the
black	dog	After	saw	and	the
black	dog	saw	the	box	too
The	brown	box	was	fast	and
much	The	black	dog	was	mat
fast	and	he	lost	the	box.
The	box	had	in	hole	and
wanted	for	the	black	dog	go
home	After	the			

Figure 3: The extracted words from the two paragraphs in Figure 2 respectively.

of letter shapes, we need to first remove rule lines. There are many complicated methods for underline removal[2]. We have hundreds of images to be processed, so we want to do it fast. Here, because most of the images scanned are clean, we apply a very quick and simple while efficient algorithm. For the broken lines, since they usually appear to be small connected components, they are easy to remove by setting a threshold of the size of a connected component. Now the solid lines have a lot of intersections with the letter parts, we hope that we can remove the solid lines while keeping the intersection parts because they contain so much critical information. Suppose all the rule lines are perfectly horizontal and uniform (in intensity), an easy way to remove solid lines is to first compute the mean value of each horizontal pixel line and deduct it from each pixel line in the original image. However, this will also remove the joint parts with letters. So we must take advantage of the peripheral information around the rule line to patch back the joint parts. As for implementation, before horizontal scanning, we build a mask by compressing the image in vertical direction to intensify the joint part where a stroke comes across the rule line. And then stretch it back to the original scale by enlarging pixels. The compression rate could be adjusted. We use this mask to patch back the intersection parts. In experiments, we can see, the bigger a stroke’s intersection angle with the rule line, the better the joint part is kept. Since all the images are scanned with little deviation angle in direction, rule lines are not perfectly horizontal. Fortunately, almost all the images scanned are in good position so that even some of the rule lines are not perfectly horizontal, they just have a very tiny slope. Therefore, an easy way to solve this problem is to separate the whole passage image verti-

cally into a few banding sub-images. And in each sub image, rule lines are approximately vertical. Thus, we simply do all the process each time in one sub-image.

2.2 Word segmentation

Word segmentation is a complex task for handwriting materials like cursive written texts[7, 3]. However, in this article, the contents of the words are irrelevant. Thus, a very accurate segmentation is unnecessary. Our method is simply based on the lengths of the word and their gaps. In the hope of reducing the effect of various lengths of word samples, we use a normal distribution to exclude outliers and only keep those words with reasonable lengths. A lot of mis-segmentation would be deleted in this process. However, there are still some mis-clustering in the data. But since all we care about is writing style instead of the content. We can just consider them as a normal word. Yet, the kept images still have different sizes and scales, so we add white margins to those short ones so as to unify their sizes. And finally we resize the image to 40×80 so that the resolution is low enough for the DNN to efficiently process.

3. DATA PREPARATION

We concatenate two word images together as a data point and input it to the lowest layer of the DNN. There are two kinds of labels: the one from the same writer and the one from distinct writers. We separate the students considered into two groups: the first group for training and the second for testing. The two groups have no intersections. For the training data, half of them are concatenations of words from the same writer and half from distinct ones. Notice that it is not necessary that the two words are the same. Actually, we randomly choose the combination of words from the dataset. For example, if we can extract 50 words from a passage written by a student, we have up to C_{50}^2 possible different combinations of words for the first half of the data for his part. See Figure 4 as an example. The test data have the same distribution but written by totally a different group of students.

box	The	the	and	fast	to
The	box	brown	hole	fast	work
he	to	the	box	puck	all
boxes	box	box	ind	black	The
the	miss	ter	the	the	p
the	black	boxes	the	saw	went
run	the	lost	was	The	clear
in	the	The	saw	and	the
boxes	After	The	black	the	red
to	the	brown	the	water	After
on	use	the	car	the	dog

Figure 4: A subset of the training data. Each of the data is a concatenation of two words either written by the same person or not.

4. DNN MODEL

One advantage of the classification problem is that we can produce numerous independent training data with labels to overcome the drawbacks of DNN. Since telling the writing style of a word needs very high level abstraction, the existing unsupervised learning[5] can hardly find variations with respect to writing style while ignoring the tremendous difference of shape of different words i.e. to figure out that two different words are written by the same person. Experiments show that supervised learning is the best choice, so we design a DNN model with 4 hidden layers. See Figure 5. The validation of this structure is given in section 6. The lowest input layer corresponds to the raw image input: an vectorized image with two extracted words concatenated together in it. The input layer is then connected to upper stacked hidden layers. The adjacent stacked layers l and $l + 1$ are densely connected by weights $W^{(l)}$, on top of which is the output layer with 2 nodes in it. Between the stacked layers, we use the sigmoid function $f(z) = \frac{1}{1+\exp(-z)}$ as the activation function so as to increase the nonlinearity of our model. Therefore, the feedforward operation could be described as

$$\begin{aligned} z^{(l+1)} &= W^{(l)} a^{(l)} + b^{(l)}, \\ a^{(l+1)} &= f(z^{(l+1)}). \end{aligned}$$

where b is the bias term.

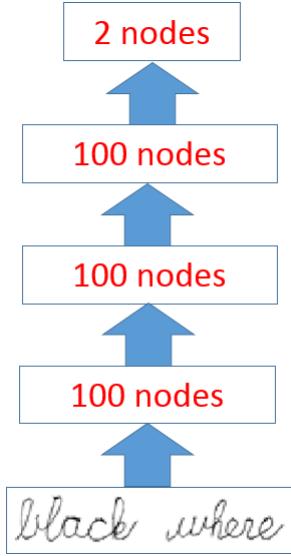


Figure 5: Structure of the DNN model

5. TRAINING

5.1 Word level feature training

We use standard backpropagation to implement mini-batch stochastic gradient descent[8]. Suppose the loss function of one data point of the model is $E(W, b; x, y)$ where W and b are the model parameters, x is the input and y is the label. We add up the loss functions for one data point together with the weight decay regularization term as the loss func-

tion of our model. Suppose there are N training data.

$$\begin{aligned} L(W, b) &= \frac{1}{N} \sum_{i=1}^N E(W, b; x^{(i)}, y^{(i)}) \\ &\quad + \frac{\lambda}{2} \sum_l \sum_i \sum_j (w_{ji}^{(l)})^2. \end{aligned}$$

Then we use stochastic gradient descent algorithm to evaluate the parameters. Using chain rule, we have the partial derivative of the loss function with respect to the value of each node $\delta_i^{(l)} = \frac{\partial E(W, b; x, y)}{\partial z_i^{(n_l)}}$, for $l < n_l$, where n_l is the number of layers in the DNN. We have

$$\delta_i^{(l)} = f'(z_i^{(l)}) \sum_j w_{ji}^{(l)} \delta_j^{(l+1)}.$$

Using these, we can computer the partial derivatives with respect to the parameters

$$\begin{aligned} \frac{\partial E(W, b; x, y)}{\partial w_{ij}^{(l)}} &= a_j^{(l)} \delta_i^{(l+1)}, \\ \frac{\partial E(W, b; x, y)}{\partial b_i^{(l)}} &= \delta_i^{(l+1)}. \end{aligned}$$

Here we use squared error as the loss function.

5.2 Measurement of Similarity

We can use the DNN described above to define and compute the similarity between two written paragraphs. Suppose we use $\phi(a, b)$ to denote whether word image a and word image b are written by the same person given by the DNN model we described. If so, $\phi(a, b) = 1$, otherwise $\phi(a, b) = 0$. Suppose we can extract m words from paragraph A and n words from paragraph B . We define their similarity as $s(A, B) = \frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n \phi(a, b)$. This kind of measurement could be very useful for the study of children's handwriting individuality development. See Figure 6 as an illustration.

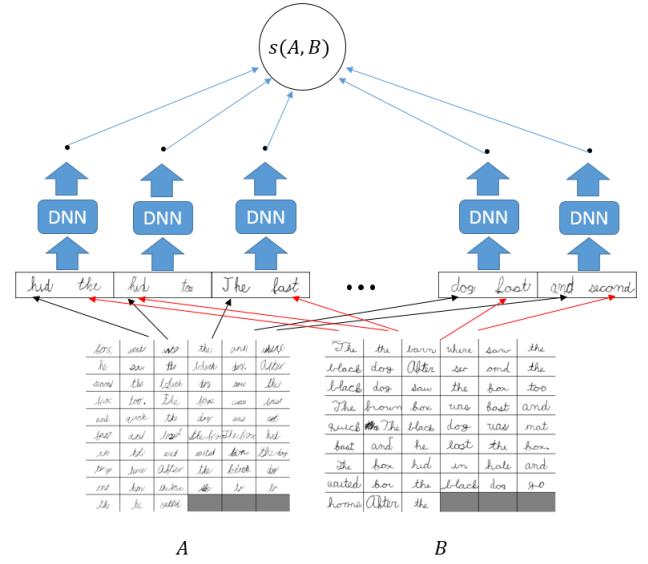


Figure 6: Illustration of how similarity is computed based on word level features.

Scale of training data	Average Error Rate
50 pairs of words from each student	38%
100 pairs of words from each student	33%
300 pairs of words from each student	30%

Table 1: The relationship between the scale of the training data and performance of the algorithm. The error rate is accurate to unit digit.

6. EXPERIMENTS

6.1 Word level feature training

We collect 300 students' first page of their writing for training and another 30 students' for testing. We can extract about 50 words on average from each of the passage. For the training set, for each person, we randomly sample 300 pairs of words labelled as '01'; then we randomly sample a pair of students 300×300 times, each time sampling a pair of words written by them respectively, labelled as '10'. Therefore, we have the same amount of data labelled as '01' or '10'. For the testing set, similarly, for each person, we randomly sample 200 pairs of words labelled as '01'; then we randomly sample a pair of students 30×200 times, each time sampling a pair of words written by them respectively, labelled as '10'. We find a 40×80 -100-100-100-2 can reduce the error rate to less than 30%. This is not a bad performance since all the words tested have never been seen by the model and we did not give our model any human designed features. We can conclude that the model has the ability of capturing writing style between different people while ignoring the very content of the text. The visualization of the weights in the first layer is shown in Figure 7. The amount of data fed into the DNN is crucial to the performance. The more combinations we input, the less overfitting we will see as shown in Table 1. During the training the learning rate α is set to 0.01. The sparse penalty λ is set to 0.001. We use a mini-batch stochastic gradient descent backpropagation to train the neural network.

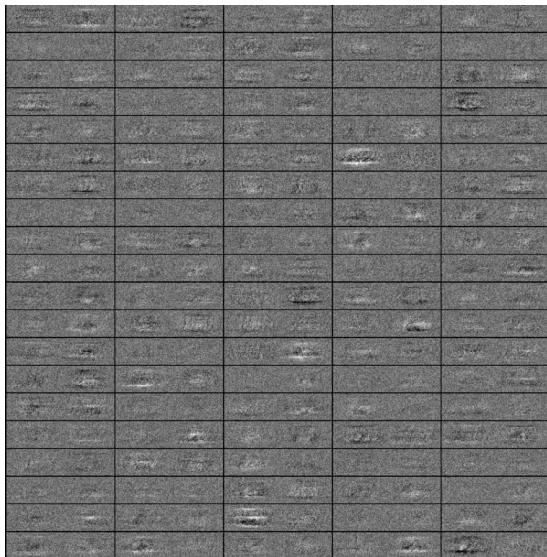


Figure 7: Visualization of the weights of the first hidden layer.

6.2 Relationship between the structure of the network and performance

A lot of experiments show that 100-node layers usually give good performance. Now we evaluate why we choose the network with this depth. We compare the performance of the different structures with the concatenated image as the input layer and 2-node layers as the output layer with several 100-node hidden layers between them. See figure 8 showing the relationship between depth and performance. Each structure is evaluated 5 times. This is why we choose 40×80 -100-100-100-2 structure as our model.

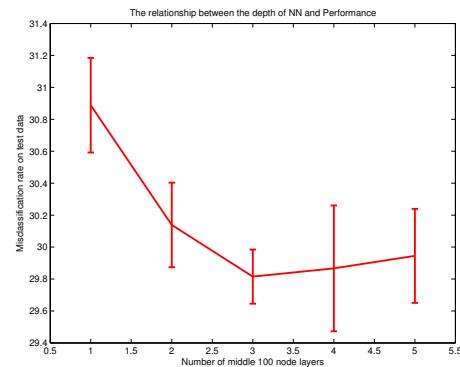


Figure 8: Relationship between the depth of the DNN and performance

6.3 Applying the similarity on paragraph level

Using the similarity defined in section 6, we can see what we get on the paragraph level test data. First, we compare the similarity outcome over data from the same year, then we compare the outcome over one year change. We take another 30 passages written by different students from the first year data as the test data. We give two histograms showing the similarity applied to passage itself and between passages written by different students. For over one year change comparison, we pick out 25 passages from the first year data and 25 passages from the second year data and compare the similarity between them. See figure 9. Even with one year gap, passages written by the same students still show higher similarity.

7. CEDAR-FOX RESULTS

CEDAR-FOX is a software system for handwriting comparison developed by the Center of Excellence for Document Analysis and Recognition at University at Buffalo. The system has interfaces to scan handwritten document images. When two handwritten document images, one known and another unknown, are presented to the software, it extracts a set of macro, micro and style features which are designed by human beings[9]. After that it computes a quantity known as Log Likelihood Ratio (LLR) with these features to describe the similarity between the two document images. When computing, it uses a generative model where the Likelihood Ratio is approximated using distributions of distances. A positive LLR value indicates that the system believes that the two documents were written by the same writer. While a negative LLR value indicates that the sys-

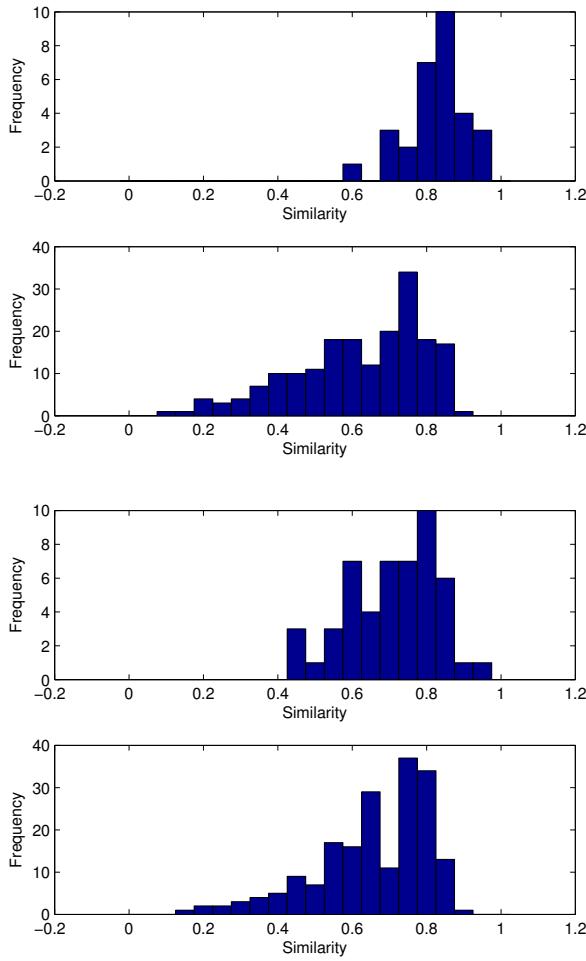


Figure 9: Upper two: similarity distribution from first year data. The first one corresponds to comparison of data from the same passage. The second one corresponds to comparison of data from different students. Lower two: similarity distribution over one year change. The first one corresponds to comparison of data from same students but with one year gap. The second one corresponds to comparison of data from different students and with one year gap.

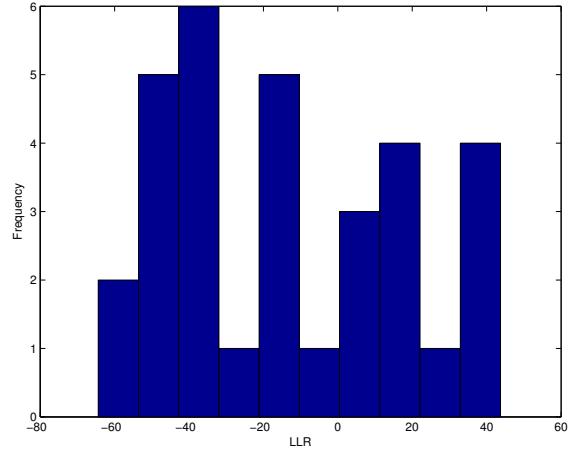


Figure 10: CEDAR-FOX results on small amount of data over one year change

tem believes that the two documents were written by different writers. We pick up a few student samples from the entire dataset and compute their LLRs over one year change respectively. We get the results shown in Figure 10. The system gives good results for students showing positive LLR values but poor results for students showing negative LLR. It seems the hand-crafted features fail to perform ideally for this problem. CEDAR-FOX is a practical tool for handwriting comparison whose effectiveness is justified by many practical problems. This shows the necessity of developing more sophisticated deep learning methods to crack the problem of children’s handwriting. A future generation of handwriting software should be able to generate more useful features automatically.

8. CONCLUSIONS

For our method of training DNN, the more training examples we generate, the better result we get. The major challenge of our method is overfitting. Due to the abstractness of writing style, a lot of attributes are more significant than writing style, such as the contents, the length of the word and so on. So we need big data to enhance the ability of generalization. With the definition of similarity, through simple combination we can produce numerous independent training data to enhance the performance.

Why unsupervised methods don’t work? Since telling the writing style of a word needs very high level abstraction, the existing unsupervised learning can hardly find variations with respect to writing style while ignoring the tremendous difference of the shapes of different words i.e. to figure out whether two different words are written by the same person. Experiments show that supervised learning is the best choice.

The use of the measurement: The measurement proposed here has various applications. For example, it can help us understand the development of writing individuality or study the effectiveness of teachers’ intervention.

9. REFERENCES

- [1] Z. Xu and S. N. Srihari. Missing Value Imputation: With Application to Handwriting Data. 2014.
- [2] W. AbdAlmageed, J. Kumar, and D Doermann. Page rule-line removal using linear subspaces in monochromatic handwritten arabic documents. *Document Analysis and Recognition*, pages 768–772, 2009.
- [3] G. Louloudis, B. Gatos, I. Pratikakis, C. Halatsis. Line and word segmentation of handwritten documents. *1st International Conference on Frontiers in Handwriting Recognition*, pages 247-252, 2008.
- [4] M. Bulacu and L. Schomaker. Text-independent writer identification and verification using textural and allographic features. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, 29(4):701-717, 2007.
- [5] G. E. Hinton, and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science* 313(5786):504-507, 2006.
- [6] S. N. Srihari., S. H. Cha, H. Arora and S. Lee. Individuality of handwriting. *Journal of Forensic Sciences*, 47(4):856-872, 2002
- [7] J. Park, V. Govindaraju, S. N. Srihari. Efficient word segmentation driven by unconstrained handwritten phrase recognition. *Document Analysis and Recognition*, 1999.
- [8] R. Hecht-Nielsen. Theory of the backpropagation neural network. *Neural Networks*, 1989.
- [9] S. N. Srihari, C. Huang, H. Srinivasan. On the Discriminability of the Handwriting of Twins. *Journal of Forensic Sciences*, 53(2):430-446, 2008