A New Efficient Method to Improve Handwritten Signature Recognition

AnusaraHirunyawanakul*, KedkarnChaiyakhan, RatipornChanklan, KittisakKerdprasop, NittayaKerdprasop

Data Engineering Research Unit, School of Computer Engineering, Institute of Engineering, Suranaree University of Technology, NakhonRatchasima 30000, Thailand.

*Corresponding author: Anusara.cpe8@gmail.com

Abstract

In this research we demonstrate the improvement for handwritten recognition using edge detection technique and our novel technique of adding intensive data. We collect totally 600 signatures from 30 people. Then we transform the hand written signatures images to be image file and resize to 144 x 38 pixels along the width and the height, respectively. Every pixel is encoded its intensity value from 0 to 255. The value 0 is the highest intensity (black) and 255 is white. Next, we use 4 different algorithms: Support Vector Machine (with linear, polynomial, radial basis, and sigmoid kernel functions), k-Nearest Neighbors, Perceptron, and Naïve Bayes (using Gaussian, multinomial, and Bernoulli density functions). From the experiment result, SVM with polynomial kernel function shows the highest accuracy (95.33%). Then we use 4 techniques of edge detection: Sobel, Prewitt, Robert, Canny and Thinning technique. With Sobel edge detection technique, we found that the accuracy is gained to 96% (higher than the highest of original data). We also observe that Sobel technique can improve the accuracy of k-NN with a significant level (from 78.67% to 91.33%). Moreover, we try to append the high intensity color data. And by this technique, we notice significant improvement of k-NN accuracy up to 96%. In SVM with linear function, after applying our technique the accuracy is improved to 98.00% which is the highest accuracy of this research.

Keywords: image recognition, handwritten signature, support vector machine, k-nearest neighbor, naïve Bayes

1. Introduction

The use of biometric in authentication or individual identification receives much attention in the current. It provides convenience of not having to carry identification documents, which reduces the problem of document falsification. The signature is external identity which is widely used for identifying individual. Signature of a person is distinct and it is hardly to be forged or counterfeited.

Vargas et al. reviewed the handwritten signatures focusing on the grey-scale measurement and co-occurrence matrix technique and local binary pattern base on MCYT-75 and GPDS-100 databases. The result was that the EER (Equal Error Rate) = 16.27%.

Guerbai et al. proposed the powerful use of OC-SVM for handwritten signature verification. The result from the experiment was 5-7% AER (Average Error Rate) in CEDAR dataset, whereas 15 - 17% AER in GPDS dataset.

Frias-Martinez et al. demonstrated the handwritten recognition based on the Support Vector Machine (SVM) and compared to a traditional classification technique like Multi-Layer Perceptron (MLP). The experimental results showed that SVM could provide up to 71% accuracy rate, which is better than the MLP technique.

Zheng et al. conducted edges and gradients detection, which was an innovative method for finding clearer edges. They used Least Squares Support Vector Machine (LS-SVM) with radial basis kernel function and Sobel and Canny edge detection. The outcome revealed that these techniques were even more effective than applying only a single machine learning technique.

Most researches on signature recognition often focused on a comparative study to find algorithms suitable for the signature recognition. However, we often encounter problems of a similar signature of different individuals or slightly different signatures of the same individual. Due to various environmental conditions, the accuracy of signature identification turns derogated. We have realized the...
importance of pre-processing. It is the importance step that can affect the accuracy rate. This paper proposes a technique to enhance the signature recognition by focusing the improvement of signature images. The signature images will be improved by edge detection technique and thinning edgetechnique. In addition, we propose a novel concept, which is never seen in any previous signature recognition researches; that is, to append the interesting area of imagedataset. And in this case is high intensity color data.

2. Related Theorems and Algorithms

2.1 Pattern Recognition

The pattern recognition(5) is the study about object classification with respect to “Feature” of each “Class.” The method can be applied to various fields, for example, the individual identification using biological data, e.g. fingerprint, face, iris, DNA, or even a signature, as well as the recognition of documents, e.g. pattern recognition of spam mail.

Issues related to pattern recognition and classification have been of great interest at present. As a result, technology and various advanced tools have been developed to be applied. The classification often requires knowledge of various branches, e.g. data mining, artificial neural network, machine learning, data improvement processes such as image-data improvement by edge detection.

2.2 Image Edge Detection

The image edge detection(6) is used to detect lines showing around the shape of an object by cutting away any other details, e.g. color or streaked. The image used to represent the shape of the object is represented as a “Binary Image.” Edge detection can be done in a variety of ways with similar principles; that is, to find the difference of color between the “Gray Scale” of one point and the other point. If the light intensity is very different, the edges will be clearer; however, if the color difference is less, the edges can be vague. The edge detection can be applied in computer vision, e.g. boundary separation between object and background or object recognition etc.

2.2.1 Sobel Edge Detection

Sobel edge detection(6) is the edge detection method by using the 2 filters with the size 3 x 3 called “Sx” and “Sy” to separate objects and background. The gradient values of each band will be computed and create the filters. Example of Sx and Sy filter are in figure 1.

\[
S_x = \begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1 \\
\end{bmatrix}
\]

\[
S_y = \begin{bmatrix}
1 & 2 & 1 \\
0 & 0 & 0 \\
1 & 2 & 1 \\
\end{bmatrix}
\]

Fig.1. Sobel filter.

The calculation of sobel gradient value is shown in equation 1.

\[
Sobel \text{ gradient} = \sqrt{(S_x^2 + S_y^2)} (1)
\]

2.2.2 Prewitt Edge Detection

Prewitt edge detection(6) is detection technique using the same concept as the Sobel edge detection. The differential is the value in the filters that shown in figure 2.

\[
h_x = \begin{bmatrix}
-1 & 0 & 1 \\
0 & 0 & 0 \\
1 & 0 & 1 \\
\end{bmatrix}
\]

\[
h_y = \begin{bmatrix}
-1 & 0 & 1 \\
1 & 0 & 1 \\
1 & 0 & 1 \\
\end{bmatrix}
\]

Fig.2. Prewitt filter.

Prewitt gradient can be calculated as shown in equation 2.

\[
Prewitt \text{ gradient} = \sqrt{(h_x^2 + h_y^2)} (2)
\]

2.2.3 Robert Edge Detection

Robert edge detection(6) is a technique using 2x2 size filters called “Gx” and “Gy”. The concept of this edge detection is to calculate the gradient of an image which is summarized from the differences between diagonally adjacent pixels. The filter of Robert edge detection is shown in figure 3.

\[
G_x = \begin{bmatrix}
1 & 0 \\
0 & -1 \\
\end{bmatrix}
\]

\[
G_y = \begin{bmatrix}
0 & 1 \\
-1 & 0 \\
\end{bmatrix}
\]

Fig.3. Robert filter.

Robert gradient can be calculated as shown in equation 3.
\[ G(F(i, j)) = |f(i + 1, j + 1) - f(i, j)| + \\
|f(i + 1, j) - f(i, j + 1)| \quad (3) \]

2.2.4 Canny Edge Detection

The first step of Canny edge detection\(^6\) is to eliminated noise. Noise can be removed by using Gaussian Filter to clear the speckles and smooth the edge of image. In the second step, a gradient operator will be applied to achieve the gradient’s intensity and direction. Then, The non-maximum suppression is used for thinning the images’ edge by determining if the pixel is a better candidate than its neighbors. The final step is using doublethresholding algorithm to specify contour pixels and make the edge continuous.

The calculation of Gaussian Filter which is applied in Canny edge detection can be explained by equation 4.

\[ S[i, j] = G[i, j, \sigma] * I[i, j] \quad (4) \]

where \( S[i, j] \) is image after filtered,
\( I[i, j] \) is image for finding edge detection,
\( G[i, j, \sigma] \) is Gaussian filter, and
\( \sigma \) is a variable to control the distribution of smoothing the image.

The image \( S[i, j] \) is used in equation 5 and 6 to find the \( x, y \) partial derivatives \( P[i, j] \) and \( Q[i, j] \). The size and direction gradient are explained in equations 7 and 8, respectively.

\[ P[i, j] \approx (S[i, j + 1] - S[i, j]) + S[i + 1, j + 1] - S[i + 1, j])/2 \quad (5) \]

\[ Q[i, j] \approx (S[i, j] - S[i + 1, j]) + S[i, j + 1] - S[i + 1, j + 1])/2 \quad (6) \]

\[ M[i, j] = \sqrt{P[i, j]^2 + Q[i, j]^2} \quad (7) \]

\[ \theta[i, j] = \arctan Q[i, j], P[i, j] \quad (8) \]

2.3 Thinning Edge

The edge thinning\(^6\) is an important preparation process (pre-processing) that is widely used to slenderize image with thick edges, which is produced from the edge detection. This is to remove the excessively thick edge pixels. Usually, excessively thick edge slenderizing is used in character recognition and signature recognition to eventually generate thinnest edge lines with only one pixel.

The thinning edge operation can be done by using \( P1 \) and \( P2 \) filters. The first step in doing that is to use the \( P1 \) filter; using the 3x3 template to scan the image data and then decide whether or not the pixels around the edges can be deleted. If the pixels can be deleted, mark them but do not delete them yet. After scanning throughout the image, delete the marked pixels. In the final step, \( P2 \) filter is used as when using the \( P1 \) filter. After deleting the marked pixels, repeat these steps until no more image data can be deleted.

2.4 Perceptron

Perceptron\(^7,8\) is one of the most popular algorithms used in classification. This algorithm is based on basic linear function model to classify the data with centroid as representative population. The linear function model mechanism is to create a line connecting the centroid of two groups and then create a perpendicular line to break the groups apart. Perceptron employs this method of linear function model as an initiative separation line. After that, the algorithm will check for the misclassified data point. If a fault exists, the counterbalance to weight of that data point to achieve the accurate classification, or to achieve classification with least erroneous data.

Another advantage of the Perceptron is that we can tune the learning rate to determine for the algorithm’s accuracy. If learning rate is too small value, the weighting in erroneous data will be also small and we will see gradual changes of the separation line. On the other hand, if the learning rate is too high in value, it will result in too aggressive changes that will affect other data and that data must be classified many times. This can explain figure 4, where \( \eta \) is the learning rate values.
2.5 Support Vector Machine

Support Vector Machine (SVM) is an algorithm based on a linear function model, which is developed from the Perceptron algorithm. It is a way to increase flexibility of classification to acquire large margin as much as possible. The concept of this algorithm is to place the data onto feature space and draw lines connecting the edges of each group. And then, the algorithm uses these data points on the edge to represent groups. The nearest data points of each group are called “support vector”. Then data separation lines of both groups are created to classify the data with the largest margins as possible, that shown in figure 5. In some cases, this algorithm can allow for misclassification to achieve the larger margin by using slack variable.

Another advantage of the Support Vector Machine is that the processing time is less than Perceptron algorithm because Support Vector Machine does not require all the data points to be calculated. In addition, the Support Vector Machine provides various kinds of functions, called kernel, to fit a specific type of data distribution. These kernel functions include linear, polynomial, radial basis, and sigmoid.

2.6 Naive Bayes

Naive Bayes is an algorithm that uses the Bayes theorem to assist in classification. It is based on the assumption that the attributes of the sample are independent. The algorithm is suitable for the set of large sample. The modeling is in the form of conditional probability. The advantage of this method of learning is that we can use the data and “Prior knowledge” to help in learning. This algorithm gives good performance when compare with the other algorithms. In terms of the calculation, the principles of probability will be used and will be based on the theory of Bayes.

2.7 k-Nearest Neighbors

k-Nearest Neighbors (k-NN) is a popular classification algorithm in the field of pattern recognition. The concept of this algorithm is classifying the new data base on the k closest training examples. And the class of new data will be assigned by the majority class label of the k closest training data.
The example of k-NN algorithm when k=1, 2 and 3.

Figure 6 shows the classification of k-Nearest Neighbors with the different k values. Results will vary depending on the number of the closest k, for example when k=1 for the incoming data (represented by x) will be classified as (-). When k=2, the incoming data can be classified as either (-) or (+). When k=3, the incoming data will be classified as (+).

The objectives of this research are:

1) To study and compare the effectiveness of hand-written signature recognition models from 4 learning algorithms: Perceptron, SVM, Naïve Bayes, and k-NN.
2) To improve the accuracy of hand-written signature recognition model by using image improvement and by addition of high intensity data.

3. Research Framework

This research consists of 6 stages (and diagrammatically shown in figure 7) as follows:

1) Collecting 600 signatures from 30 university students who use the hand-written signature in daily life and turn to images file by scanning device. We adjust the image’s color to black and white. Then, equalize their size. The raw data of all hand-written signature image files are stored at the main author’s website: https://sites.google.com/site/nhinganusaracpesut/signature/datasets

2) Using of edge detection technique and thinning edges to sharpen signature images.

3) Converting the data into a numeric table in accordance with color intensity, and then converting the numeric table to the array data.

4) Input the array data to the 4 learning algorithms to recognize correct owner of each signature. Studying and comparing the performances and drawing the initial conclusions.

5) Adding each image data with high intensity area appending to the original image and choosing edge detection technique that gives highest accuracy in the image improvement process (the experimental results show that the Sobel edge detection technique gives highest precision).

6) Comparing the learning accuracy of models that are created from a training data set annotated with high intensity area. Then observe results and draw the conclusions.
4. Experimental Results

With all 600 signatures from 30 individuals, we use Python 2.7 Language on Editor Spyder to predict the results of signature recognition.

The image improvement at the second step of our proposed framework yields the results as shown in figure 8.

From table 1, the four algorithms used in this study include Perceptron algorithm, Support Vector Machine algorithm (Linear Function, Polynomial Function, Radial Basis Function, and Sigmoid Function), Naive Bayes algorithm (Gaussian Function, Multinomial Function, and Bernoulli Function), and k-Nearest Neighbors algorithm. These algorithms are used in comparative test. The results obtained indicate that, in using the original image files, the SVM-Polynomial Function provides the highest accuracy of 95.33% and we see 94.67% by SVM-Linear Function. The Naive Bayes algorithm with Multinomial Function gives 82.67% of accuracy, and the k-Nearest Neighbors algorithm gives 78.67% of accuracy.

<table>
<thead>
<tr>
<th>Original</th>
<th>Sobel Edge Detection</th>
<th>Prewitt Edge Detection</th>
<th>Robert Edge Detection</th>
<th>Canny Edge Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fig. 8. Example images after applying edge detection and thinning techniques to signature images of three persons</td>
<td><img src="image1.jpg" alt="Example Images" /></td>
<td><img src="image2.jpg" alt="Example Images" /></td>
<td><img src="image3.jpg" alt="Example Images" /></td>
<td><img src="image4.jpg" alt="Example Images" /></td>
</tr>
</tbody>
</table>

![Table 1. Experimental results of signature recognition with image improvement techniques](image5.jpg)

Table 1. Experimental results of signature recognition with image improvement techniques

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Function</th>
<th>Original</th>
<th>Sobel</th>
<th>Prewitt</th>
<th>Robert</th>
<th>Canny</th>
<th>Thinning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Perceptron</td>
<td>linear</td>
<td>68.67%</td>
<td>86.67%</td>
<td>65.33%</td>
<td>72.00%</td>
<td>64.00%</td>
<td>45.33%</td>
</tr>
<tr>
<td>SVM</td>
<td>linear</td>
<td>95.33%</td>
<td>96.00%</td>
<td>82.00%</td>
<td>89.33%</td>
<td>83.33%</td>
<td>86.00%</td>
</tr>
<tr>
<td></td>
<td>poly</td>
<td>94.00%</td>
<td>2.00%</td>
<td>14.67%</td>
<td>2.00%</td>
<td>88.67%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>rbf</td>
<td>0.67%</td>
<td>0.67%</td>
<td>0.67%</td>
<td>0.67%</td>
<td>0.67%</td>
<td>0.67%</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>sigmoid</td>
<td>0.67%</td>
<td>0.67%</td>
<td>0.67%</td>
<td>0.67%</td>
<td>0.67%</td>
<td>0.67%</td>
</tr>
<tr>
<td></td>
<td>gaussian</td>
<td>56.67%</td>
<td>20.67%</td>
<td>65.33%</td>
<td>72.00%</td>
<td>65.33%</td>
<td>71.33%</td>
</tr>
<tr>
<td></td>
<td>multinomial</td>
<td>82.67%</td>
<td>89.33%</td>
<td>87.33%</td>
<td>86.67%</td>
<td>85.33%</td>
<td>75.33%</td>
</tr>
<tr>
<td></td>
<td>bernoulli</td>
<td>0.67%</td>
<td>17.33%</td>
<td>68.67%</td>
<td>72.00%</td>
<td>67.33%</td>
<td>0.67%</td>
</tr>
<tr>
<td>kNN</td>
<td>k = 3</td>
<td>78.67%</td>
<td>91.33%</td>
<td>8.00%</td>
<td>28.00%</td>
<td>6.00%</td>
<td>9.33%</td>
</tr>
</tbody>
</table>
Table 2. Experimental results when using additional intensive data technique

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Function</th>
<th>Original</th>
<th>Sobel Edge</th>
<th>Sobel Edge + Additional Intensive Data Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Accuracy</td>
<td>Accuracy</td>
<td>Accuracy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Train</td>
<td>Test</td>
<td>Train</td>
</tr>
<tr>
<td>Perceptron</td>
<td>linear</td>
<td>100%</td>
<td>68.67%</td>
<td>100%</td>
</tr>
<tr>
<td>SVM</td>
<td>poly</td>
<td>100%</td>
<td>95.33%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>rbf</td>
<td>100%</td>
<td>0.67%</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td>sigmoid</td>
<td>4.22%</td>
<td>0.67%</td>
<td>4.22%</td>
</tr>
<tr>
<td>Naive Bayes</td>
<td>gaussian</td>
<td>100.0%</td>
<td>56.67%</td>
<td>99.56%</td>
</tr>
<tr>
<td></td>
<td>multinomial</td>
<td>97.33%</td>
<td>82.67%</td>
<td>94.00%</td>
</tr>
<tr>
<td></td>
<td>bernoulli</td>
<td>4.22%</td>
<td>0.67%</td>
<td>49.78%</td>
</tr>
<tr>
<td>kNN</td>
<td>k = 3</td>
<td>92.67%</td>
<td>78.67%</td>
<td>98.22%</td>
</tr>
</tbody>
</table>

Fig. 9. Accuracy comparisons of original signature image data recognition, recognition from Sobel edge detected data, and recognition from both Sobel and additional intensive data techniques

The accuracy improvement of learning algorithms after using image processing with edge detection technique and thinning reveals that k-Nearest Neighbors algorithm’s accuracy is increased by 12.66%; that is, from 78.67% to 91.33% with the use of Sobel edge detection technique. This increment is very significant. The accuracy of SVM-Linear Function is increased by applying the Sobel edge detection technique as well and the improved accuracy is 96.00%. Sobel edge detection is the best image processing technique applied prior to the signature recognition with learning algorithms.

Table 2 shows the results of using the additional intensive data technique with Sobel edge detection: SVM-Linear Function provides the accuracy of 98.00%, which is the highest accuracy in this research. Moreover, we also found that the k-Nearest Neighbors algorithm provides higher accuracy by using the additional intensive data; that is the accuracy increases from 91.33% to 96.00% (which is higher than the maximum value of the original data).

The accuracy comparisons of signature image recognition without any other techniques, recognition with edge detection technique, and recognition with both edge detection and our additional intensive data techniques are shown in figure 9. It can be noticed that the combination of Sobel edge detection technique and our novel proposed additional intensive data technique yields the highest recognition rate at 98%.
5. Conclusions

We study the problem of handwritten signature recognition with the main objective of devising techniques to improve recognition accuracy rate. According to the signatures collected from hand-written users for this research, the SVM-Linear Function is the most suitable learning algorithm for modeling the signature recognition with the edge detection technique applied for image improvement and the additional intensive data technique newly proposed for accuracy improvement. This combination of edge detection and additional intensive data techniques provides the accuracy rate of up to 98.00%. For the technique of image improvement, the researchers note that we have possibility to achieve higher accuracy if we study more advanced techniques of image processing.

Acknowledgment

This research has been supported by grant from the National Research Council of Thailand (NRCT) and Suranaree University of Technology through the funding of research unit.

References