

ANN Based Handwritten Signature Recognition System

Myat Mon Kyaw¹, San San Nwe¹, Myint Myint Yee²

¹Faculty of Information Science, University of Computer Studies, Maubin, Myanmar

²Information Technology Supporting and Maintaining Department, University of Computer Studies, Maubin, Myanmar

How to cite this paper: Myat Mon Kyaw | San San Nwe | Myint Myint Yee "ANN Based Handwritten Signature Recognition System" Published in International Journal of Trend in Scientific Research and Development (ijtsrd), ISSN: 2456-6470, Volume-3 | Issue-5, August 2019, pp.2094-2097, <https://doi.org/10.31142/ijtsrd27866>



IJTSRD27866

ABSTRACT

Handwritten Signature Verification (HSV) is an automated method of verifying a signature by capturing features about a signature's shape (i.e., static features) and the characteristics of how the person signs his/her name in real-time (i.e., dynamic features). This system provides a method of handwritten signature recognition and verification using the shapes of the signatures, artificial neural network and neural network simulation tool. The shapes of signatures are used to find the features points for features extraction. Then the extracted features are trained by using artificial neural network. A comparison of extracted features is done between the original signature and other relative signatures by using neural simulation toolbox. If the features are matched, the system shows that the signature is verified and the person is authorized and unauthorized.

KEYWORDS: HSV, artificial neural network

INTRODUCTION

Biometrics based verification systems are improved systems in terms of security than traditional verification techniques such as passwords etc. It is due to the fact that biometric characteristics of every person are unique and cannot be lost, stolen or broken.

There are two types of biometrics:

1. Behavioral
2. Physiological

Copyright © 2019 by author(s) and International Journal of Trend in Scientific Research and Development Journal. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (CC BY 4.0) (<http://creativecommons.org/licenses/by/4.0>)



Signature recognition is a behavioral biometric. Signature verification is one of the most important features for checking the authenticity of a person because it is quite accurate and cost efficient too. On the other hand one doesn't have to remember the authentication key like pin code or password. The signature of a genuine signer stays almost constant.

A. Theory Background

Handwritten signatures are considered as the most natural method of authenticating a person's identity. The basic goal of the handwritten signatures is to provide an accurate method in order to verify a person's identity based on the way in which he/she signs his/her name. Hence for this reason, the handwritten signatures are widely accepted, socially and legally throughout the world. The hand-written signature verification uses the features conveyed by every signatory such that the features considered have a unique understanding and the way of signing presents the behavioral biostatistics. Some researchers considered common issues with the extraction of identification data from different biometric types, and protection of such data against conceivable attacks. Handwritten signatures are very much dependent on the user's psychology and has great difference in different surroundings and time.

B. Color Space

The representation of colors in an image is achieved using a combination of one or more color channels that are combined to form the color used in the image. The representation we use to store the colors, specifying the number and nature of the color channels, is generally known as the color space.

C. Features

The process of the image in such a way that the image, or properties of it, can be adequately represented and extracted in a compact form amenable to subsequent recognition and classification. Representations are typically of two basic kinds: internal (which relates to the pixels comprising a region) or external (which relates to the boundary of a region).

Extracted attributes will have numerical values and will be usually stored in arrays. With these values, neural network can be trained and we can get a good end results. The problem of well-defined datasets lies also in carefully chosen algorithm attributes. Attributes are important and can have a crucial impact on end results. The most important attributes for handwriting algorithms are:

- The horizontal position, counting pixels from the left edge of the image, of the center of the smallest rectangular box that can be drawn with all "on" pixels inside the box.
- The vertical position, counting pixels from the bottom, of the above box.
- The width, in pixels, of the box.
- The height, in pixels, of the box.
- The mean horizontal position of all "on" pixels relative to the center of the box and divided by the width of the box. This feature has a negative value if the image is "leftheavy" as would be the case for the letter L.
- The mean vertical position of all "on" pixels relative to the center of the box and divided by the height of the box.

- The mean squared value of the horizontal pixel distances as measured in 6 above. This attribute will have a higher value for images whose pixels are more widely separated in the horizontal direction as would be the case for the letters W or M.
- The mean squared value of the vertical pixel distances as measured in 7 above.
- The mean value of the squared horizontal distance times the vertical distance for each "on" pixel. This measures the correlation of the horizontal variance with the vertical position.
- The mean value of the squared vertical distance times the horizontal distance for each "on" pixel. This measures the correlation of the vertical variance with the horizontal position.
- The mean number of edges (an "on" pixel immediately to the right of either an "off pixel or the image boundary) encountered when making systematic scans from left to right at all vertical positions within the box. This measure distinguishes between letters like "W" or "M" and letters like "I" or "L."
- The sum of the vertical positions of edges encountered as measured in 13 above. This feature will give a higher value if there are more edges at the top of the box, as in the letter "Y."
- The mean number of edges (an "on" pixel immediately above either an "off pixel or the image boundary) encountered when making systematic scans of the image from bottom to top over all horizontal positions within the box.
- The sum of horizontal positions of edges encountered as measured in 15 above.

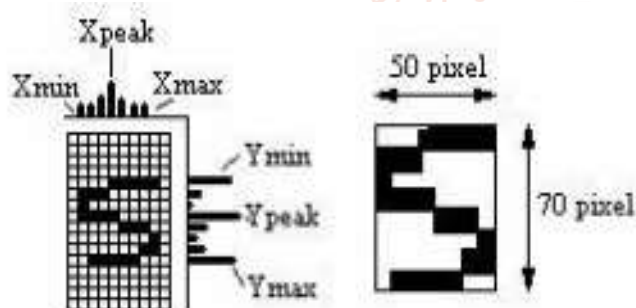


Figure1: Example of Image Defined Attributes For Feature Extraction

D. Image Segmentation

Segmentation is the name given to the generic process by which an image is subdivided into its constituent regions or objects. In general, completely autonomous segmentation is one of the most difficult tasks in the design of computer vision systems and remains an active field of image processing and machine vision research. Segmentation occupies a very important role in image processing because it is so often the vital first step which must be successfully taken before subsequent tasks such as feature extraction, classification, description, etc. can be sensibly attempted. The basic goal of segmentation, then, is to partition the image into mutually exclusive regions to which we can subsequently attach meaningful labels. The segmented objects are often termed the foreground and the rest of the image is the background. Note that, for any given image, we cannot generally speak of a single, 'correct' segmentation. Rather, the correct segmentation of the image depends strongly on the types of object or region we are interested in

identifying. What relationship must a given pixel have with respect to its neighbors and other pixels in the image in order that it be assigned to one region or another?

E. Artificial Neural Network

Artificial neural networks are inspired from their biological counterparts, the biological brain and the nervous system. The most important distinctive feature of a biological brain is its ability to "learn" and "adapt", while a conventional computer does not have such abilities. Conventional computers accomplish specific tasks based upon the instructions loaded to them, the so-called "programs" or "software".

Basic building block of neural networks is a "neuron". A neuron can be perceived as a processing unit. In a neural network, neurons are connected with each other through "synaptic weight"s, or "weight"s in short. Each neuron in a network receives "weighted" information via these synaptic connections from the neurons that it is connected to and produces an output by passing the weighted sum of those input signals (either external inputs from the environment or the outputs of other neurons) through an "activation function".

There are two main categories of network architectures depending on the type of the connections between the neurons, "feed-forward neural networks" and "recurrent neural networks". Feed-forward neural networks are the mostly encountered and used in many diverse applications, therefore they are chosen to exemplify the artificial neural networks. Back-propagation algorithm is the mostly used algorithm in the training of feed-forward neural networks, but it is also used, along with the modified versions of the algorithm, in the training of other types of neural networks. If there is no "feedback" from the outputs of the neurons towards the inputs throughout the network, then the network is referred as a "feed-forward neural network". Otherwise, if there exists such a feedback, i.e. a synaptic connection from the outputs towards the inputs (either their own inputs or the inputs of other neurons), then the network is called a "recurrent neural network". Usually, neural networks are arranged in the form of layers. Feed-forward neural networks fall into two categories depending on the number of the layers, either "single layer" or "multi-layer".

F. Neural Network Toolbox

Neural Network Toolbox provides algorithms, pre-trained models, and apps to create, train, visualize, and simulate both shallow and deep neural networks. You can perform classification, regression, clustering, dimensionality reduction, time-series forecasting, and dynamic system modeling and control. Deep learning networks include convolutional neural networks (ConvNets, CNNs), directed acyclic graph (DAG) network topologies, and auto-encoders for image classification, regression, and feature learning. For time-series classification and regression, the toolbox provides long short-term memory (LSTM) deep learning networks. You can visualize intermediate layers and activations, modify network architecture, and monitor training progress.

G. Overview of the system

In the signature training process, firstly, the program takes the input signature image. And then, the image is converted from RGB to Gray as preprocessing step. Second, the resulted

signature image from preprocessing step is applied features extraction method using the shape of the signatures. Then, find features points for features extraction. And then, the value of points resulted are trained using Artificial Neural Network. Finally, the trained feature vectors are saved into the feature dataset using cell.

In the signature testing process, the input testing signature is processed using the same methods in the training process to get features points. And then, the features vector are matched using Neural Network Simulation Toolbox. If the features are matched, the system shows that the signatures are verified and the person is authorized and unauthorized. And then, this system will also show both the comparison of training and testing signature image and other information. If not, the system shows that signature is unknown.

H. System Implementation



Figure2: Welcome Window

When the user clicks the 'Signature Registration' button, the training process page will appear as shown in figure 3. User must input the name of the folder that contains the signatures of one organization. If the user clicks the 'Load Images' button, the input signature images will be trained.

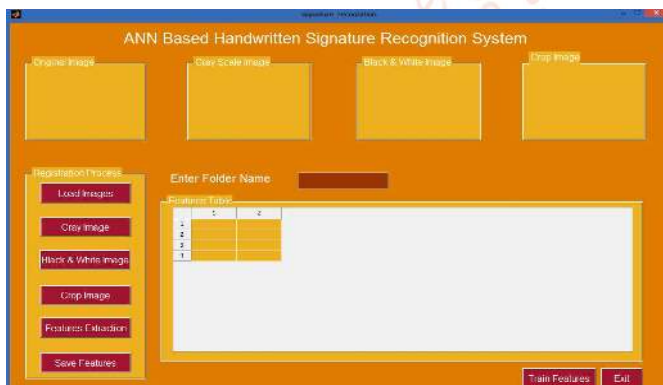


Figure3: Process of 'Signature Registration' button

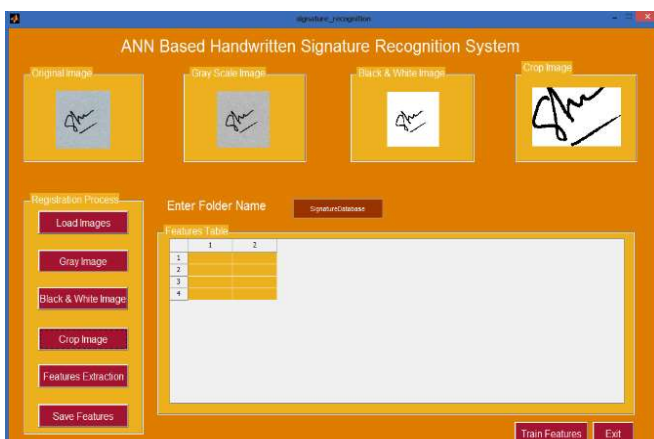


Figure4: After cropping the image

For preprocessing step, user can click the "Gray Image" and "Black & White Image" button. When user clicks "Gray Image" button the gray image converted from RGB image will appear at the right side of the original Signature RGB image. And, if user clicks the "Black & White Image" button the binary image converted from Gray image will be displayed at the right side of the Gray image. Then, user can click the 'Crop Image' button. The cropped signature image will appear as shown in figure 4.



Figure5: Feature extraction from the image

To find the features, user can click the 'Features Extraction' button as shown in figure 5. The feature point values will also appear in the features table.

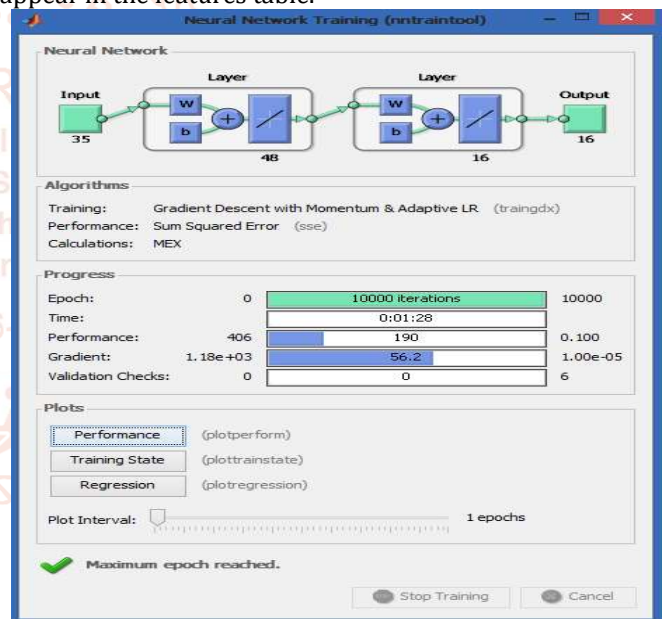


Figure6: Feature training page

To train the features with artificial neural network, user can click the 'train features' button as shown in figure 6. In testing process, when the user clicks the 'Signature Verification' button to verify the signatures, the testing process page will appear.

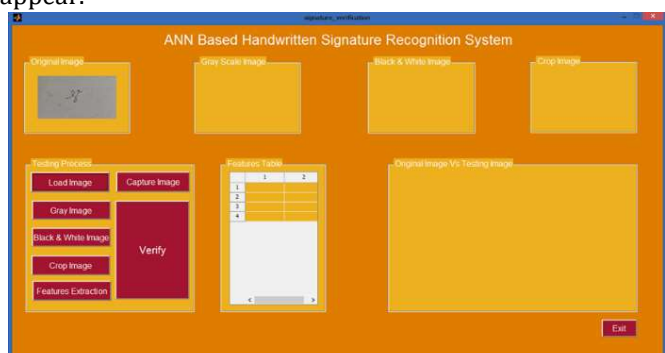


Figure 7: Signature Verification

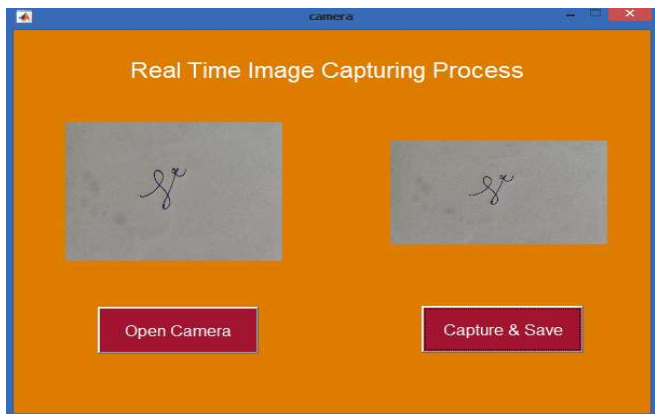


Figure8: Capture and Save real time image

In this page, user must input the test signature image for testing. When the user clicks the 'Capture Image' button to obtain the testing signature image, the real time process will appear. When the user clicks the 'Open Camera' button, the camera is opened and the image is captured and saved. The user can upload testing image by clicking 'Load Image' button to make testing.

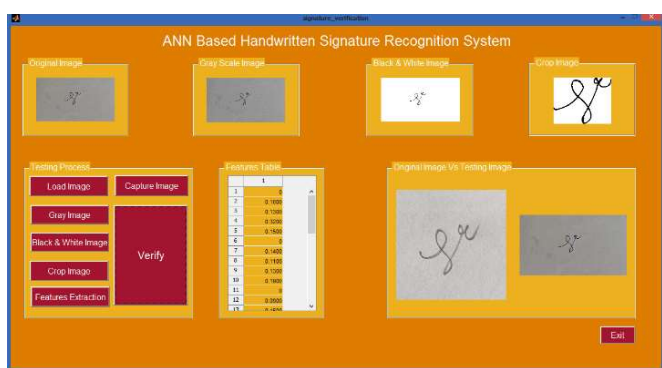


Figure9: Verification for testing process

To verify the input testing signature, user can click the 'Verify' button after features have been extracted. If the input testing signature is located in the feature dataset, the testing signatures image and the corresponding information message box about this signature will be displayed. If the features are matched, "verify" message box will be displayed and not matched, "not verify" message box will be displayed. Finally, if the user click "Exit" button, the testing process will finish.

Conclusion

Signatures continue to be an important biometric trait because it remains widely used primarily for authenticating the identity of human beings. It is designed to recognize the signatures of people and to use in every organization. Generally the failure to recognize/verify a signature was due to poor image quality and high similarity between 2 signatures. This system can be used in many areas such as finance, insurance, health, education and telecommunication. This system can do real-time verification. It helps to reduce the cases of forgeries by taking the fake signatures. This system can be used in many areas such as finance, insurance, health, education and telecommunication. This system can do real-time verification. It helps to reduce the cases of forgeries by taking the fake signatures. The overall performance and accuracy rate is about 75% according to the training dataset and 25% of error rate.

References

- [1] A Maccabe, "A review of Dynamic Handwritten Signature Verification," James Cook University, P.7, 1998.
- [2] A T Wilson, "Off-line Handwritten Recognition using Artificial Neural Network", University of Minnesota, 2000.
- [3] David J. Janson, "Training product unit neural networks with genetic algorithm", 1993.
- [4] Dorothy A. Pender, "Neural Networks and handwritten signature verification, Stanford University, 1991.
- [5] E A Breimer, "A Learning Approach for Designing Dynamic Programming Algorithms", Rensselaer Polytechnic Institute, 2000.
- [6] R Adhaem, "Automatic Computer Techniques for Personal Signature Verification", Baghdad University, 2001.