

A Performance of Combined Methods of VCG and 16BCD for Feature Extraction on HSV

Arief Bramanto Wicaksono Putra, Mulyanto, Bedi Suprpty, Achmad Fanany Onnilita Gaffar*

Department of Information Technology, Politeknik Negeri Samarinda, East Kalimantan, Indonesia

Email: ariefbram@gmail.com; yanto1294@gmail.com; bedirheody@gmail.com; onnygaffar212@gmail.com*

*Corresponding author

Received: 28 October 2020; Revised: 13 January 2021; Accepted: 28 February 2021; Published: 08 June 2021

Abstract: The digital signature image is a digital pattern with highly variable features. The pattern recognition of digital signature images aims to build a specific characteristic capable of representing a considerable pattern variation while maintaining the boundary conditions of authentication. The feature as an attribute that describes the characteristics of a pattern becomes a determinant factor of reliability of a method of recognizing digital signature image pattern for Handwritten Signature Verification (HSV). To construct HSV required two types of signature samples that are the original signature samples used as training samples and the guess signature samples (consist of valid and imposter signature) which are used as test samples. This study proposes two unique features of 16-Bits Binary Chain to Decimal (16BCD) and Virtual Center of Gravity (VCG). The 16BCD feature obtained from image segmentation with a 4x4 pixel region. All pixels in each region of the segmentation result rearranged into a 16-bit binary chain. The VCG feature is a virtual representation of the Original Signature Pattern (OSP) gravity center against Pattern Space and Background. The verification mechanism uses criteria: the percent of acceptable correlation coefficients for the acceptable feature of 16BCD feature, Mean Absolute Error (MAE) against 16BCD, and the percent deviation of acceptable distance to the VCG feature prototype. Verification test results obtained Acceptance Rate (AR) 80% (which states the percentage of HSV success based on a number of original signature samples) with an efficiency of 90% (which states the percentage of success of HSV in distinguishing valid or forgery signature based on a sample of guessing signatures).

Index Terms: Handwritten Signature Verification (HSV); Virtual Center of Gravity (VCG); 16BCD; Original Signature Pattern (OSP)

1. Introduction

The signature verification technique performed by visual inspection by comparing its visualization cannot guarantee its authenticity. It is because it involves human visual perception factors that are very susceptible to mistakes. Signatures have a reasonably low consistency pattern, compared to other biometric attributes (retina, face, fingerprints), so the verification is very complicated.

The modern approach to Handwriting Signature Recognition/Verification (HSR/HSV) consists of three main stages: pre-processing, feature extraction, and classification. The pre-processing stage aims to suppress noise, normalize, and compressing data in a certain amount of information to be stored. The feature extraction stage aims to extract features from image characters as their representations. Feature selection is related to the technique used to select the most relevant features in order to improve classification accuracy. Classification aims to recognize a class of features based on the nature of the prototype features produced in a certain way [1]. Features defined as attributes that describe the characteristics of a pattern. It determines the reliability of the method of recognizing digital signature patterns.

In principle, there are two types of features, namely statistical and structural features. Statistical features based on statistical parameters from raw data. Structural features based on raw data behaviour such as attributes, objects, geometric properties that can be a collection of strokes, the number of bifurcation points, the number of contour points, and so on. There are many methods and approaches for feature extraction of handwritten signature images. A zoning based feature extraction method for recognition of handwritten signature is the one of feature extraction method which introduced in [2]. The zoning method is a feature extraction method based on statistical techniques that calculates the number of black pixels in each zone. A zone is obtained from an image that is divided into 16 parts of the same size. The signature arrays of 16 lengths generated are converted into a 128-bit binary string. The Hough Transform was introduced in [3] to detect the signature line. The classification has performed by using Histogram of a Gradient (HoG) features and a Euclidean distance. The Algorithms of BRISK (Binary Robust Invariant Scalable Keypoints) and FAST

(Features from Accelerated Segments) have been used to detect and extract signature features in [4]. The pixel density and center of gravity distance have been used as features in [5] while the Principal Component Analysis (*PCA*) is used to analyse those features. Reference [6] proposed an algorithm to correct the alignment of the input signature which can be used at the pre-processing stage to achieve better results in the signature detection process. Several methods of extracting statistical features have also discussed in [7-12]. Reference [13] had used Gabor Filter to extract the surf feature and critical point matching, while the classification is based on Hidden Markov Model (*HMM*). Reference [14] had addressed feature types of feature extraction for off-line handwritten signature verification that is divided into three main categories: global features, local features, and geometric features. Several methods of extracting structural features have also discussed in [15-19]. A deep learning model for high-level representation of feature extraction of handwritten signature was discussed in [20-28].

Finally, several classification and signature verification methods have researched in [5, 7, 13, 15, 19, 27, 29-31].

Digital images of signatures consist of a variety of patterns which have very varied features. Various methods of pattern recognition of digital signature images aim to produce unique characteristics that can represent a variety of patterns while maintaining boundary authentication conditions. This study focuses on the extraction of handwritten signature features as part of the HSV model. This study proposes a combination of two unique features of VCG and 16BCD through the proposed HSV model which is expected to increase the percentage of acceptance rate and its efficiency. The performance of the two proposed features is tested through a handwritten signature verification process using the Fuzzy Rule-based verifier.

2. Related Works

Signature is one of the characteristics of human biology called biometrics, which can change due to age, mood, environmental conditions, and others. Its results in the inconsistency of several signatures from the same person, which does not match each other perfectly when compared. A unique feature of the signature that represents the signature characteristic is required to eliminate the inconsistency factor. The selection of good features and their extraction method is essential concerning the verification stage. Several studies that have focused on this have been conducted.

Freeman Chain Code (FCC) feature has been used in [32] to describe the signature image's object boundary by recording the direction in which the next pixel is located and its corresponding neighbor in the image. This feature was chosen because of its simplicity in representing data and fast computation time. In [33], the signature feature is extracted from the signature contour, which is the vertex and trough of the signature contour after rotation. Each vertex and trough point is obtained from each 8-neighborhood block operation observed over all parts of the signature contour image. The sum of the relative distances between successive vertices and troughs yields a total divided by the signature field. This feature extraction is called the Crest-Trough method. This method is classified as local image feature extraction—other local image feature extraction methods such as Local Weighting Pattern (LWP) [34]. LWP is built based on a pixel's local weight, which is the sum of the binary code. This binary code is generated from all threshold values between the 8-neighborhood pixel and the pixel center. Various statistical-based feature extraction methods can be applied to LWP texture images as visual features and produce unique features. The use of simple geometric features has also been implemented in achieving offline signature verification in [35]. The features referred to are Baseline Slant Angle (BSA), Aspect Ratio (AR), and Normalized Area (NA), and Center of Gravity. Signature prototypes were obtained for each subject due to combining the above geometric features from the subject's actual signature sample set. The use of a combination of several visual features has also been carried out in [36]. This study has proposed a combination of texture features of the Gray Level Co-occurrence Matrix (GLCM), Local Binary Pattern (LBP), and Discrete Wavelet Transform (DWT). All the texture features are extracted from the gray image.

There are two main tasks for signature recognition and verification: 1). identify the signature of the owner, 2). classify guess signature whether it is genuine or forged. There are various offline signature verification techniques, such as Support Vector Machine (SVM), Multi-Layer Perceptron's Neural Network, Gray level Distribution, Discrete Cosine Transform (DCT) and Hidden Markov Model (HMM), Two step Transitional Feature, and others [37]. Artificial Neural Network (ANN) and Deep Learning approaches have also been used for signature verification in [6, 7, 20-22, 26, 28, 33, 35, 38, 39].

3. Performance Measurement

One of the simplest techniques to measure pattern similarity is to use the Euclidean Distance expressed by

$$\text{dist}(X, Y) = \|X - Y\| = \sqrt{(X_n - X)^2 + (Y_n - Y)^2} \quad (1)$$

where X_n and Y_n are the pixel coordinate positions adjacent to the reference pixels, (X, Y) . One technique to measure the pattern similarity that is considered powerful is the correlation coefficient between the feature prototype model resulted from a training process with the features of the pattern to be verified, expressed by

$$r_{xy} = \frac{\text{cov}(X, Y)}{\sigma_x \cdot \sigma_y} = \frac{E[(X - \mu_x) \cdot (Y - \mu_y)]}{\sqrt{E(X^2) - E^2(X)} \cdot \sqrt{E(Y^2) - E^2(Y)}} \quad (2)$$

where X and Y are two pixel values that have the same position among the feature prototype model with the features of the pattern to be verified, μ_x and μ_y are the expected values of the set X and Y , σ_x and σ_y are deviation standard of the set X and Y . Because r_{xy} within the interval $\{-1, 1\}$ then for $r_{xy} \rightarrow 0$ is said *Strongly Uncorrelated* [40].

To construct *HSV* required two types of signature samples that are the signature samples used as training samples and the guess signature samples which are used as test samples. On the pattern recognition of digital signature are known some kind of sample signature as follows:

1. *Original Signature (OS)* is the genuine signature as a training sample.
2. *Valid Signature (VS)* is the *OS* as a guess signature sample.
3. *Forgery signature (FS)* is the imposter signature as a guess signature sample.

The verification criteria used in this study are:

1. Percentage of acceptable correlation coefficient on 16BCD feature prototype.
2. Percentage of acceptable absolute error against the 16BCD feature prototype.
3. Percentage of acceptable deviation of distance against the VCG feature prototype.

In a biometric system, the reference threshold defined as a value that can determine a person's authenticity. Some statistical calculations used in this study to test the authenticity of the handwritten signatures are as follows [41-43]:

1. *FRR (False Rejection Rate)* which states the fraction of the number of rejected *VS* divided by the number of *VS* samples, mathematically expressed by:

$$FRR = \frac{N_{VS(rejected)}}{N_{VS(samp)}} \times 100\% \quad (3)$$

2. *FAR (False Acceptance Rate)* which states the fraction of the number of accepted *FS* as *VS* divided by the number of *FS* samples, mathematically expressed by:

$$FAR = \frac{N_{FS(accepted)}}{N_{FS(samp)}} \times 100\% \quad (4)$$

Because in this study using the same number of *VS* and *FS* samples with the number of training samples then it can be stated $N_{VS(samp)} = N_{FS(samp)} = N_{(samp)}$. The performance of feature extraction methods based on verification results is expressed by:

1. *AR (Acceptance Rate)* which states the percentage of *HSV* success based on a number of *OS* pattern (*OSP*) samples, mathematically expressed by:

$$AR = (100 - FRR) \% \quad (5)$$

2. *Efficiency (%)* which states the percentage of success of *HSV* in distinguishing *VS* or *FS* based on a sample of guessing signatures, mathematically expressed by:

$$efficiency(\%) = \frac{N_{(samp)} - (N_{FRR} + N_{FAR})}{N_{(samp)}} \times 100\% \quad (6)$$

4. The Proposed Method

A. The HSV Model

Signatures to be verified are captured with certain technologies to obtain Digital Image Signature (*DIS*) in the default JPG format. After going through the pre-processing and pattern segmentation stages, features of the *Guess Signature Pattern (GSP)* are obtained through the extraction process. The features obtained are then verified using the built-in Model Verifier. The verification result will be state: *accepted* or *not accepted*.

The proposed HSV model is shown in Fig. 1, whereas its construction stages are shown in Fig. 2.

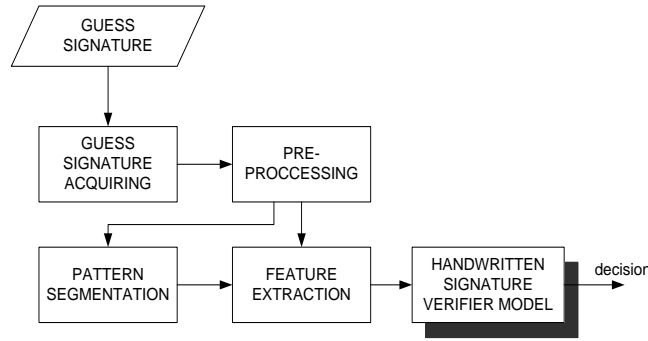


Fig. 1. The proposed HSV Model

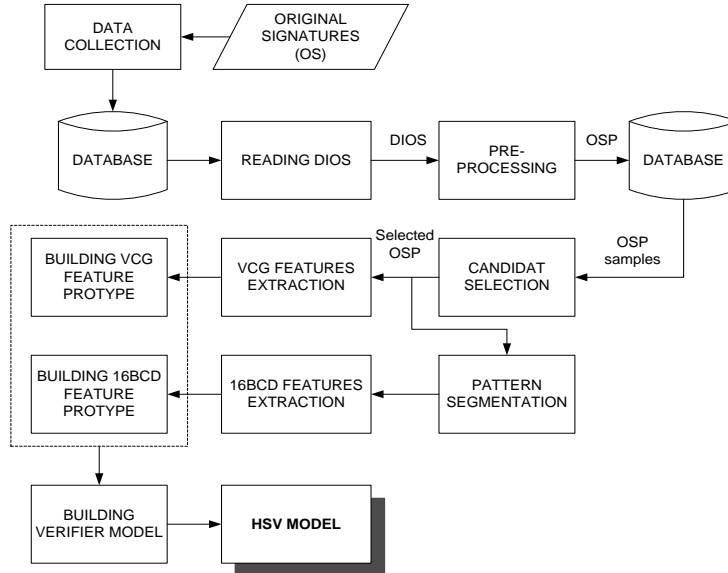


Fig. 2. The stages of the proposed HSV Model construction

B. Data Collection

The *OS* is captured using certain technologies to acquire Digital Image of *OS (DIOS)*, then stored into the database. *DIOS* is taken from the same person with at least four samples. *DIOS* stored in JPG format with the size of $M \times N$ pixels, where $M = N > 64$.

C. Reading DIOS

Reading DIOS: By using the *M-File* MATLAB program, the *DIOS* file reading result will be stored in an array declared with

$$DIOS = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \dots & \dots & \dots & \dots \\ a_{M1} & a_{M2} & \dots & a_{MN} \end{bmatrix} \quad (7)$$

D. Pre-processing

1) *RGB to Grayscale image conversion*: The default *DIOS* is true color (*RGB*). To reduce image complexity, then the image needs to be converted into a binary image after converted into a grayscale type. Grayscale is a measure of gray pixel levels that can be calculated in various ways. In MATLAB, the gray-value of each *RGB* pixel is represented by:

$$gray = 0.2989 * R + 0.587 * G + 0.114 * B \quad (8)$$

where *R*, *G*, and *B* are *RGB* components of an image.

2) *Noise Filtering*: Grayscale images need to be filtered to eliminate the possibility of noise, so the binary images obtained contain only elements of the signature pattern. In this study used Gauss Filter, which is stated by:

$$h_{i,j} = \frac{1}{2\pi\sigma^2} \cdot \exp\left(-\frac{i^2+j^2}{2\sigma^2}\right) \quad (9)$$

3) *Grayscale to binary image conversion*: The binary image derived from the process of normalizing grayscale images in intervals $\{0, 1\}$ with specific threshold values. The relationship between gray-level and binary level shown in Fig. 3.

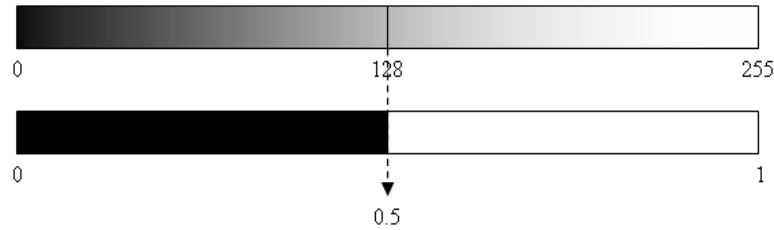


Fig. 3. Relation between Gray level and Binary level

If threshold = 0.5 then all pixels with gray values 0 - 128 will be normalized to 0 (black), whereas all pixels with gray values above 128 will be normalized to 1 (white).

4) *Cutting Edge/Cropping Image*: It is necessary to cut the edges of *X* and *Y* coordinates on the image to obtain the maximum signature pattern. The result is an image with a signature pattern that has a maximum *X* and *Y* value.

5) *Resize image to 128 × 128 pixels*: The obtained signature patterns called *OS Pattern (OSP)* then converted to a default size of 128 × 128 pixels. The obtained *OSP* sample expressed by Eq. (7) with $M = N = 128$. All the element values of the *OSP* array are 0 or 1.

E. Candidate Selection

If there are n training samples, the correlation test is performed on four samples. Three received samples are maintained, while the next data sample added to the next correlation test. The process runs continuously until all the training data samples tested. The correlation test between images is done based on the row and column vectors of the image. The row vector is the sum of all pixel values in each image column. The column vector is the sum of all the pixel values in each image row. The row vector of the four samples to be tested is expressed by:

$$\begin{aligned}
 X1_j &= \sum_{i=1}^{128} OSP1_{i,j} & X2_j &= \sum_{i=1}^{128} OSP2_{i,j} \\
 X3_j &= \sum_{i=1}^{128} OSP3_{i,j} & X4_j &= \sum_{i=1}^{128} OSP4_{i,j} \\
 j &= 1 \dots 128
 \end{aligned} \tag{10}$$

The column vector of the four samples to be tested is expressed by

$$\begin{aligned}
 Y1_i &= \sum_{j=1}^{128} OSP1_{i,j} & Y2_i &= \sum_{j=1}^{128} OSP2_{i,j} \\
 Y3_i &= \sum_{j=1}^{128} OSP3_{i,j} & Y4_i &= \sum_{j=1}^{128} OSP4_{i,j} \\
 i &= 1 \dots 128
 \end{aligned} \tag{11}$$

Referring to Eq. (2) then the correlation test against the row vector of the four samples can be done by using the following formula:

$$\begin{aligned}
 a &= (X1_j - \text{mean}(X1_j)) \\
 b &= (X2_j - \text{mean}(X2_j)) \\
 r_{(X1, X2)} &= \frac{\sum_{j=1}^{128} a.b}{\sqrt{\sum_{j=1}^{128} a^2 \cdot \sum_{j=1}^{128} b^2}}
 \end{aligned} \tag{12}$$

The correlation test performed on a row vector against the other row vectors of the four samples. Equation (12) also used to perform the correlation test on a column vector against the other column vectors of the four samples. The results of the correlation test for each vector of the four samples (row and column) will be a 4×4 size matrix. If the matrices C and D , respectively, are the results of the correlation test on the row and column vectors of the four samples of OSP then to calculate the deviation toward the perfect correlation (1) is used the following formula:

$$\begin{aligned}
 c &= \sum_{j=1}^4 C_{i,j} & d &= \sum_{j=1}^4 D_{i,j} \\
 cd &= (c + d) / 2 & dev &= |cd - 1| \\
 i &= 1 \dots 4
 \end{aligned} \tag{13}$$

Three OSP samples that had the smallest deviation selected as candidates.

F. VCG feature

The VCG feature is a virtual representation of the center of gravity of OSP toward Pattern Space and Background. The VCG representation shown in Fig. 4. That figure shows that variable d_{yW} is Euclidean Distance between OSP and W , whereas variable d_{yB} is Euclidean Distance between OSP and B .

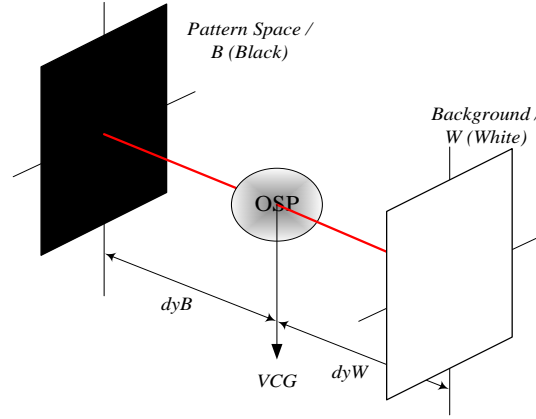


Fig. 4. VCG representation

Generally, the *VCG* feature expressed by:

$$\begin{aligned}
 d_{yW(M,M)} &= \sqrt{\sum_{i=1}^M \sum_{j=1}^N (y_{i,j} - W_{j,i})^2} \\
 d_{yW} &= \frac{1}{M} \sum_{i=1}^M \sum_{j=1}^N d_{yW(i=j)} \\
 d_{yB(N,N)} &= \sqrt{\sum_{j=1}^N \sum_{i=1}^M (B_{j,i} - y_{i,j})^2} \\
 d_{yB} &= \frac{1}{N} \sum_{j=1}^N \sum_{i=1}^M d_{yB(i=j)}
 \end{aligned} \tag{14}$$

where:

$W_{(N,M)}$: the identity array (all elements are 1) of size $N \times M$ with a column vector M as input.

$B_{(N,M)}$: a zero array (all elements are 0) of size $N \times M$ with a column vector M as input.

$y_{(M,N)}$: *OSP* array with size $M \times N$

$d_{yW(M,M)}$: Euclidean Distance between y and W , as an array of size $M \times M$

$d_{yB(N,N)}$: Euclidean Distance between y and B , as an array of size $N \times N$

d_{yW} : mean of diagonal sum of $d_{yW(S,Q)}$

d_{yB} : mean of diagonal sum of $d_{yB(S,Q)}$

The *VCG* feature mathematically is expressed by:

$$VCG_i = (d_{yW_i}, d_{yB_i}) \tag{15}$$

where i is the selected *OSP* number. The *VCG* feature will be a spatial coordinate pair (d_{yW}, d_{yB}) in virtual coordinates as shown in Fig. 5.

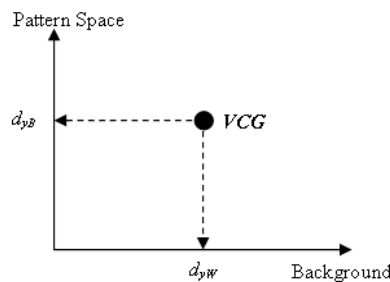


Fig. 5. VCG feature

G. 16BCD feature

The 16-bit Binary Chain to Decimal (16BCD) feature obtained by region-based segmentation. It is the division of signature patterns performed by regions of a specific size. For example, with the size of the region 4×4 pixels, then for the pattern size 16×16 pixels will be obtained 16 segments, as shown in Fig. 6.

For OSP 128×128 pixels in size, with segmentation based region of 4×4 pixels size will be obtained 32 segments. The 16-bit Binary Chain obtained by arranging the pixel OSP value of each segment up to a 16-bit binary chain. Mathematically expressed by:

$$\begin{aligned} {}^{16}BC_{iseg,jseg} &= \sum_{i=k}^{(k+sizereg-1)} \sum_{j=m}^{(m+sizereg-1)} OSP_{i,j} \\ NOSP_{iseg,jseg} &= dec({}^{16}BC_{iseg,jseg}) \end{aligned} \quad (16)$$

where:

${}^{16}BC_{iseg,jseg}$: 16-Bit Binary Chain on $(iseg,jseg)$ coordinate.

k : $iseg * sizereg + 1 - sizereg$

m : $jseg * sizereg + 1 - sizereg$

$sizereg$: size of region

$OSP_{i,j}$: string of OSP pixel size on (i,j) coordinate

seg : the number of segment.

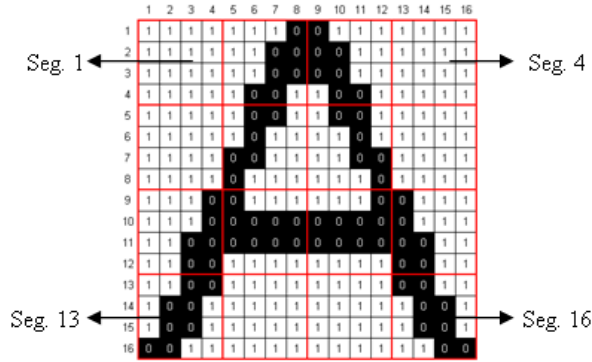


Fig. 6. Example of segmentation based region 4×4 pixel's size

The 16BC settings for each segment then converted to decimal numbers. The 16BCD feature obtained is then used to convert OSP into a new pattern called $NOSP$. The 16BCD feature expressed by:

$$\begin{aligned} y &= f(x) = NOSP_{i,j} \\ x &= i * j + (seg - j) * (i - 1) \end{aligned} \quad (17)$$

where x is the segment number of OSP , y is the value of each OSP pixel, seg is the number of segmentation = size of column/row of $NOSP$. The curve function of Eq. (17) fitted by using 3rd polynomial stated by:

$$p_i(x) = p_{i(1)} \cdot x^3 + p_{i(2)} \cdot x^2 + p_{i(3)} \cdot x + p_{i(4)} \quad (18)$$

where $p_{i(1)} \dots p_{i(4)}$ are the polynomial coefficients such that $p_i(x) \rightarrow y_i$, and the index of i is the selected OSP number.

H. Feature prototype

The Feature prototype is an identity feature (Feature ID) obtained from a mix of selected OSP samples as candidates. Feature ID is a unique feature of $DIOS$ of a person.

The combined VCG features of the three OSP candidates' relationships obtained by using the following formula:

$$VCG = \frac{1}{3} \sum_{i=1}^3 VCG_i = (d_{yW}, d_{yB}) \quad (19)$$

Euclidean Distance between combined *VCG* features with each *VCG* feature is expressed by

$$d_i = \sqrt{(d_{yW} - d_{yW_i})^2 + (d_{yB} - d_{yB_i})^2} \quad (20)$$

From Eq. (20) then the *VCG* feature prototype can be stated by:

$$dist_{range} = [0 \ d_{max}] \quad dist_{space} = \pi \cdot (d_{max})^2 \quad (21)$$

Because there are three selected samples as candidates, by using Eq. (17) and (18) there will be three 16BCD features. The 16BCD feature prototype built by arranging all 16BCD features into one array, expressed by:

$$Y = [y_1 \ y_2 \ y_3] \quad Y_P = [y_{min} \ y_{mid} \ y_{max}] \quad (22)$$

where *Y* is array of 128×3 size.

Refer to Eq. (2), the correlation coefficient between 16BCD features in its prototype calculated by using the following formula:

$$\begin{aligned} r(i)_{min} &= r_{ymin,y(i)} & r(i)_{mid} &= r_{ymid,y(i)} \\ r(i)_{max} &= r_{ymax,y(i)} & i &= 1...3 \end{aligned} \quad (23)$$

The correlation coefficient range is obtained by finding the value of the maximum correlation coefficient as follows:

$$\begin{aligned} r(i) &= \max([r(i)_{min} \ r(i)_{mid} \ r(i)_{max}]) \\ r_{range} &= sort(r)_{ascending} \end{aligned} \quad (24)$$

Since the perfect value of the correlation coefficient is 1, then

$$CC_{range} = [\min(r_{range}) \ 1] \quad (25)$$

MAE (Mean Absolute Error) between 16BCD features in its prototype expressed by:

$$\begin{aligned} e(i)_{min} &= f_i - Y_{min} & e(i)_{max} &= f_i - Y_{max} \\ e(i)_{mid} &= f_i - Y_{mid} \\ AE_i &= [e(i)_{min} \ e(i)_{mid} \ e(i)_{max}] \\ MAE &= \frac{1}{3} (e(i)_{min} + e(i)_{mid} + e(i)_{max}) \\ i &= 1...3 \end{aligned} \quad (26)$$

Since a smaller *MAE* is better (near zero), the *MAE* range can be expressed by:

$$MAE_{range} = [0 \ \max(sort(MAE)_{ascending})] \quad (27)$$

5. Fuzzy Rule based Verifier Model

Fuzzy Logic is a method for mapping input space to output space using a membership function. A collection of several membership functions that represent certain linguistic variables is called the Fuzzy Set. The fundamental difference between crisp and fuzzy sets is that crisp sets always have a unique Membership Function (*MF*). In contrast, each fuzzy set has a membership value that is limited by the representative function of membership. This is shown in Fig. 7. The Fuzzy set is characterized by a *MF* that maps each element into the universe of discourse (*X*) into intervals $\{0,1\}$, expressed by

$$\mu_X : X \rightarrow \{0,1\} \quad (28)$$

where μ_X is the degree of membership of *X*.

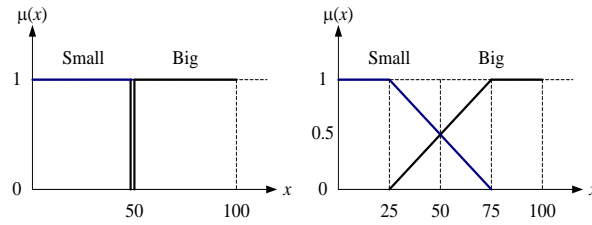


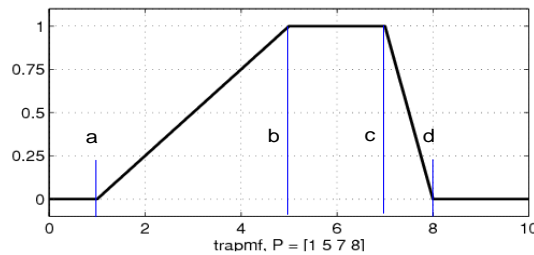
Fig. 7. The difference between fuzzy and crisp sets

Some commonly used MF as shown in Fig. 8, which mathematically expressed by:

$$\text{trapmf}(x, [a, b, c, d]) = \begin{cases} \frac{x-a}{b-a}, & a \leq x \leq b \\ 1, & b \leq x \leq c \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & x \leq a, x \geq d \end{cases} \quad (29)$$

$$\text{trimf}(x, [a, b, c]) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a \leq x \leq c \\ c, & c \leq x \end{cases}$$

Fuzzy Inference System (*FIS*) also called fuzzy inference engine is a system that can do the reasoning with a similar principle as humans doing reasoning with their instinct. There are several known *FIS* types like *MAMDANI*, *SUGENO*, and *Tsukamoto*. The easiest *FIS* is *MAMDANI*. The *FIS* works on linguistic rules and has a fuzzy algorithm that provides a mathematical approximation. The *FIS* is shown in Fig. 9 [44].



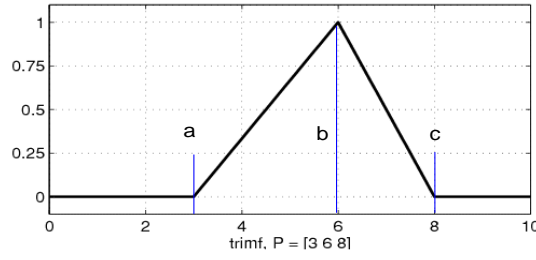


Fig. 8. Trapezoidal and Triangular MF

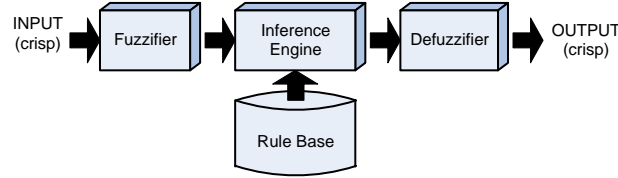


Fig. 9. Fuzzy Inference System

The Verifier model builds on the built-in *VCG* and *16BCD* feature prototypes. The Fuzzy Rule Based (*FRB*) Verifier Model is shown in Fig. 10.

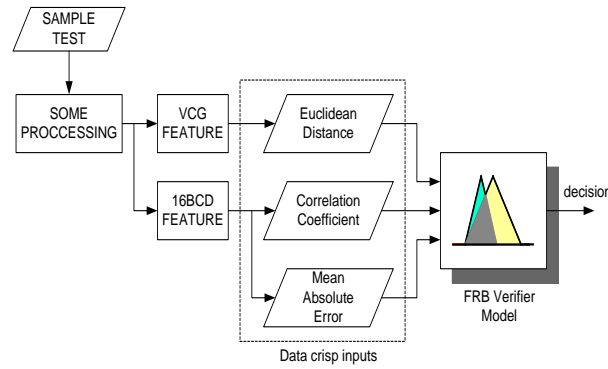


Fig. 10. Fuzzy Rule-Based Verifier Model

There are three crisp data that are used as input of *FRB* Verifier Model as follows:

1. Euclidean Distance between *VCG* features of sample test and feature prototype.
2. Correlation Coefficient between *16BCD* features of sample test and feature prototype.
3. *MAE* between *16BCD* features of sample test and feature prototype.

Mathematically, the fuzzy rule is expressed by:

$$\begin{aligned}
 R_i(x, y) &= R_i(x_1, x_2, \dots, x_n, y) \\
 &= (A_{j(i)}(x) \wedge B_{j(i)}(x) \wedge C_{j(i)}(x) \wedge D_{j(i)}(y)) \\
 &= \min_{j=1}^{m=3} (A_{j(i)}(x), B_{j(i)}(x), C_{j(i)}(x), D_{j(i)}(y))
 \end{aligned} \tag{30}$$

where n is the number of input and m is the number of linguistic variables. With $n=3$, $m=3$, then the number of rules are $3^3 = 27$, and the fuzzy output is expressed by

$$\begin{aligned}
 \tilde{D}(y) &= (R \circ (A(x), B(x), C(x))) \\
 &= \bigvee_{i=1}^{n=27} (A_{j(i)}(x) \wedge B_{j(i)}(x) \wedge C_{j(i)}(x) \wedge D_{j(i)}(y)) \\
 &= \max_{i=1}^{m=27} \left(\min_{j=1}^{m=3} (A_{j(i)}(x), B_{j(i)}(x), C_{j(i)}(x), D_{j(i)}(y)) \right)
 \end{aligned} \tag{31}$$

The crisp output is defuzzification of the fuzzy output by using the following formula:

$$x_c = \frac{\sum_{i=1}^n x_i \cdot \mu(x_i)}{\sum_{i=1}^n \mu(x_i)} \quad (32)$$

where x_c is the centroid value, x_i is x value of i th point correspond to $\mu(x_i)$ from the aggregation result, $\mu(x_i)$ is the degree of membership of x_i , n is the number of $(x_i, \mu(x_i))$ pairs in the curve.

A guess signature sample is *verified/accepted* if:

1. Euclidean Distance is in the representation space of *VCG* feature prototype.
2. The correlation coefficient is not less than the minimum range of correlation coefficient of *16BCD* feature prototype.

MAE is not less than the minimum range of the *MAE* of *16BCD* feature prototype

6. Experiments

This study has used the same number of samples, both *OS* and *FS*, as many as ten samples. All data were collected from several people in the Computer Vision laboratory, Department of Information Technology, Politeknik Negeri Samarinda, East Kalimantan, Indonesia. All of the *OS* samples have also used as a guess signature samples. All the *OS* samples were processed according to the stages, as follows.

A. Pre-processing

The pre-processing stage until binary image conversion shown in Fig. 11. An example of OSP results shown in Fig. 12.

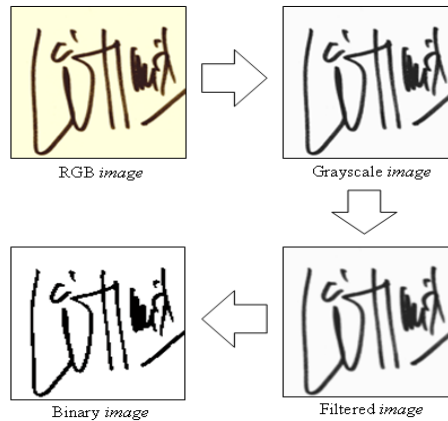


Fig. 11. An example of the pre-processing stage until binary image conversion

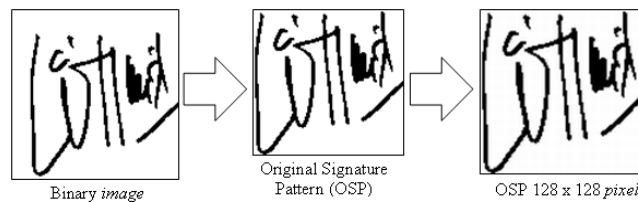


Fig. 12. An example of the OSP (128 × 128 pixels size)

B. Candidate selection

The *OSP* selection stage as a candidate was to perform the correlation test by using Eq. (10), (11), (12), & (13). For example, the correlation test results on the row and column vectors of the four *OSP* samples yield the matrices *C* and *D*, respectively, as follows:

$$C = \begin{bmatrix} 0 & 0.2721 & 0.4577 & 0.2611 \\ 0.2721 & 0 & 0.2779 & 0.1730 \\ 0.4577 & 0.2779 & 0 & 0.4096 \\ 0.2611 & 0.1730 & 0.4096 & 0 \end{bmatrix}$$

$$D = \begin{bmatrix} 0 & 0.7771 & 0.8534 & 0.7581 \\ 0.7771 & 0 & 0.6570 & 0.6965 \\ 0.8534 & 0.6570 & 0 & 0.7085 \\ 0.7581 & 0.6965 & 0.7085 & 0 \end{bmatrix}$$

By using Eq. (13) then obtained as follows:

$$dev = [0.6898 \quad 0.4268 \quad 0.6822 \quad 0.5034]$$

Since the maximum deviation toward perfect correlation (1) is 0.6898 was in column 1, then *OSP1* was rejected. The *OSP* samples selected as candidates were *OSP2*, *OSP3*, and *OSP4*.

C. VCG feature extraction

The *VCG* feature for each selected *OSP* has extracted by using Eq. (14) & (15) then obtained:

$$VCG_1 = (d_{yW_1}, d_{yB_1}) = (5.0353, 9.9728)$$

$$VCG_2 = (d_{yW_2}, d_{yB_2}) = (4.6506, 10.1910)$$

$$VCG_3 = (d_{yW_3}, d_{yB_3}) = (4.1591, 10.3933)$$

D. 16BCD feature extraction

Each *OSP* candidate was segmented using a region of 4×4 pixels. For example, the obtained *OSP* segmentation have shown in Fig. 13.

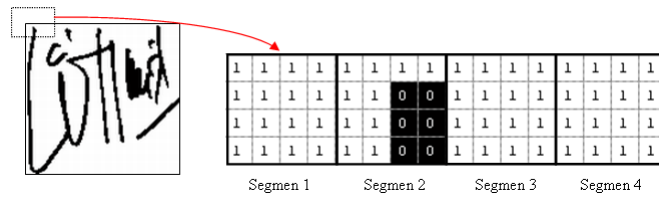


Fig. 13. 16BC formation

The 16BC can be stated as follows:

Binary Chain Segment 1: 1111111111111111
 Binary Chain Segment 2: 1111110011001100
 Binary Chain Segment 3: 1111111111111111
 Binary Chain Segment 4: 1111111111111111

The result of arranging 16BC for each segment then converted into decimal number as follows:

segment1:

$$\text{round}\left(\text{dec}[1111111111111111] \times \frac{255}{65535}\right)$$

$$= \text{round}\left(65535 \times \frac{255}{65535}\right) = 255$$

segment 2:

$$\text{round}\left(\text{dec}[1111110011001100] \times \frac{255}{65535}\right)$$

$$= \text{round}\left(64716 \times \frac{255}{65535}\right) = 252$$

segment 3:

$$\text{round}\left(\text{dec}[1111111111111111] \times \frac{255}{65535}\right)$$

$$= \text{round}\left(65535 \times \frac{255}{65535}\right) = 255$$

segment 4:

$$\text{round}\left(\text{dec}[1111111111111111] \times \frac{255}{65535}\right)$$

$$= \text{round}\left(65535 \times \frac{255}{65535}\right) = 255$$

In the same way, a 32×32 pixels New *OSP* (*NOSP*) have generated where each pixel contains a decimal value of 16BCD created. An example of *NOSP* shown in Fig.14.

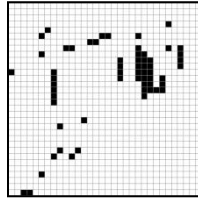


Fig. 14. An example of *NOSP*

The 16BCD feature have obtained by using Eq. (17) with the fitting curve of the function by using Eq. (18). An example of the results of fitting curve function of 16BCD feature have shown in Fig.15.

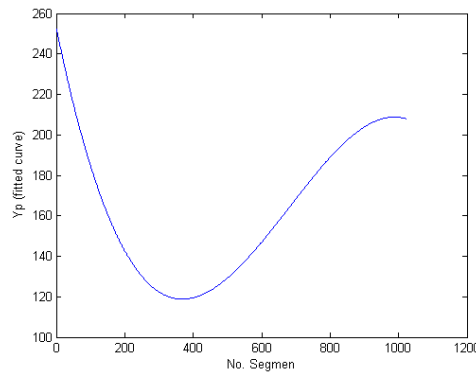


Fig. 15. An example of the results of fitting curve function of 16BCD feature

E. Building feature prototype

1) *VCG feature prototype*: The combined *VCG* features was obtained by using Eq. (19) as follows:

$$VCG = (4.6150, 10.1857) = (d_{yW}, d_{yB})$$

Euclidean Distance between combined VCG features with VCG_1 , VCG_2 , and VCG_3 were obtained by using Eq. (20) as follows:

$$d_1 = 0.3912 \quad d_2 = 0.0524 \quad d_3 = 0.3406$$

It appears that the furthest distance of VCG feature of all OSP candidates toward the combined VCG feature was $d_{max} = 0.3912$. The VCG feature prototype was obtained by using Eq. (21) as follows:

$$dist_{range} = [0 \ 0.3912] \quad dist_{space} = \pi \cdot (0.3912)^2$$

Graphically, the VCG feature prototype shown in Fig. 16.

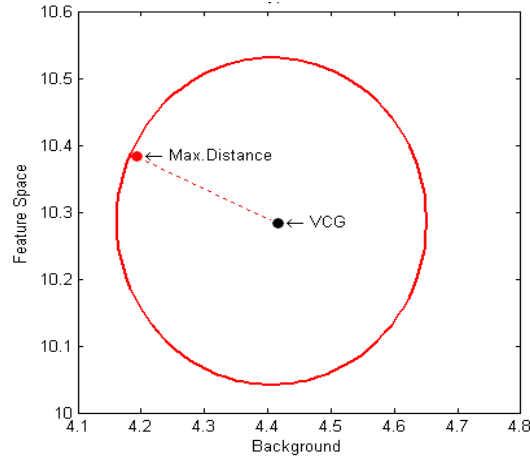


Fig. 16. The VCG feature prototype

2) *16BCD feature prototype*: By using *16BCD* feature then obtained three NOSP as shown in Fig. 17.

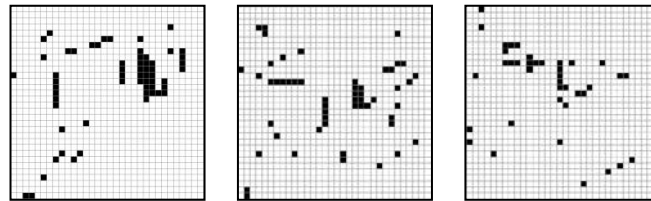


Fig. 17. All the NOSP

By using Eq. (22) then obtained the 16BCD feature prototype as shown in Fig. 18.

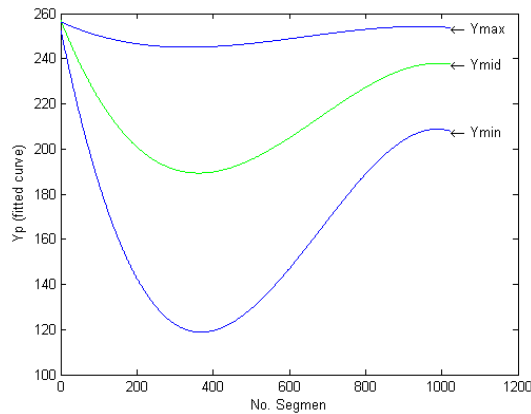


Fig. 18. The 16BCD feature prototype

The correlation coefficient range of the 16BCD feature prototype was calculated by using Eq. (23), (24), & (25) with the results as follows:

$$CC_{range} = [0.9696 \quad 1]$$

The MAE range of the 16BCD feature prototype was calculated by using Eq. (26), & (27) with the results as follows:

$$MAE_{range} = [0 \quad 6.6883]$$

F. Building FRB Verifier Model

For all crisp data inputs use linguistic value: *SMALL*, *MEDIUM*, *LARGE*. Fuzzy sets built as inputs were: Distance Test (A), Correlation Coefficient Test (B), and MAE Test (C). The Fuzzy Acceptance Level set (D) was used as the output verification model with its linguistic values: *LOW* (D_1), *MEDIUM* (D_2), and *HIGH* (D_3). All the fuzzy sets used were shown in Fig. 19, 20, 21, & 22.

From the feature analysis results obtained that the perfect resemblance occurs when:

1. Distance test \rightarrow *SMALL* (A_1) (close to zero)
2. Correlation Coefficient Test \rightarrow *LARGE* (B_3) (close to one)
3. MAE Test \rightarrow *SMALL* (C_1) (close to zero)

Fuzzy rules built with the following combination:

1. The acceptance level is in the linguistic value of *HIGH* (D_3) if at least two inputs are qualifying the perfect resemblance $\rightarrow [A_1, B_3, \dots], [A_1, \dots, C_1], [\dots, B_3, C_1]$.
2. The acceptance level is in the linguistic value of *MEDIUM* (D_2) if at least two inputs are in the linguistic value of *MEDIUM* $\rightarrow [A_2, B_2, \dots], [A_2, \dots, C_2], [\dots, B_2, C_2]$, or all three inputs are in linguistic values *SMALL*, *MEDIUM*, *LARGE* alternately $\rightarrow [A_1, B_2, C_3], [A_3, B_1, C_2], [A_2, B_3, C_1]$.
3. The acceptance level is in the linguistic value of *LOW* (D_1) if at least two inputs are on a linguistic value contrary to the condition of perfect resemblance $\rightarrow [A_3, B_1, \dots], [A_3, \dots, C_3], [\dots, B_1, C_3]$.

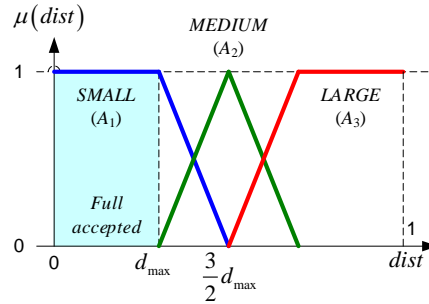


Fig. 19. Fuzzy sets for Distance Test

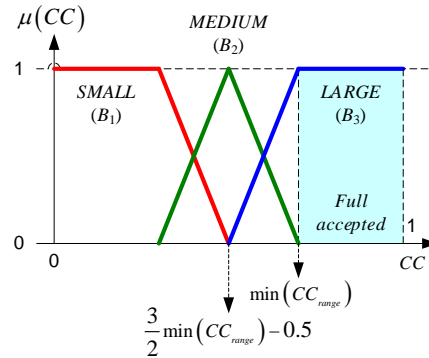


Fig. 20. Fuzzy sets for Correlation Coefficient Test

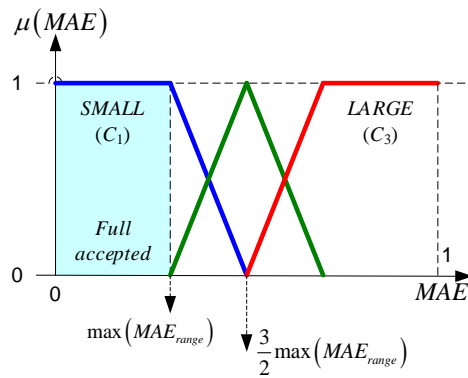


Fig. 21. Fuzzy sets for MAE Test

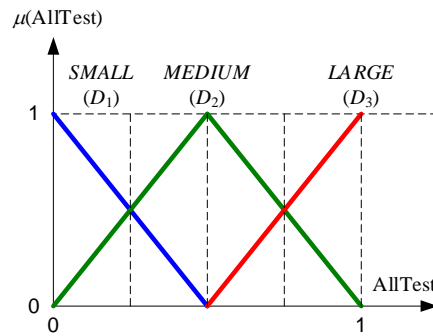


Fig. 22. Fuzzy sets for The Output of All Test

The *FIS MAMDANI* method performed using Eq. (30) to obtain all the rules implication, and then aggregated by using Eq. (31) to obtain the fuzzy output. The crisp output obtained by using Eq. (32).

G. Test results

All the *OS* and *FS* samples were used as guess signature samples. A guess signature sample was rejected if it has a test value below 50%, otherwise was accepted. The performance of the overall acceptance test result was calculated by using Eq. (3), (4), (5), & (6) as shown in Table 1.

Table 1. Test Results

Data	Acceptance Level		
	OS	GUESS	
		VS	FS
1	30.83	45.23	41.81
2	90.74	44.19	19.42
3	90.74	33.79	19.42
4	90.74	90.74	19.42
5	90.74	90.74	19.42
6	90.74	86.35	19.42
7	90.74	46.64	19.42
8	90.74	90.74	19.42
9	51.72	90.74	19.42
10	38.09	90.74	19.42
N_{FRR}	2	2	
N_{EAR}			0
FRR	20.00%	20.00%	
FAR			0.00%
AR	80.00%		
Efficiency		90.00%	

7. Conclusions

After conducting this study, the following conclusions have obtained:

1. The VCG feature indicates a trend towards a pattern space represented in the spatial position of the VCG coordinates. With this feature, the consistency of pattern features will be maintained although the pattern is dilated.
2. In principle, the 16BCD feature is a compression pattern with a 4×4 region compression window. The resulting binary chain will be a 16-bit binary number, which converted into a gray-level interval $\{0, 255\}$. With this feature, the value of a pixel equal to 8 neighbors (8-neighborhood) will remain indicated differently after the conversion process.
3. FRB has used as a Model Verifier to describe the uncertainty of a sample's pre-processing results into a certain degree of certainty. With FRB, the authentication of a signature pattern can be expressed directly in the degree of acceptance level.
4. This study has used the same number of samples, both OS and FS, as many as ten samples. All of the OS samples have also used as a guess signature samples. Verification test results obtained Acceptance Rate (AR) 80% with an efficiency of 90%.

Acknowledgment

The authors wish to acknowledge the financial support from the Research and Community Service Unit (Unit Penelitian & Pengabdian Masyarakat) Politeknik Negeri Samarinda 2020 and would like to express their heartfelt thanks to The Modern Computing Research Center, Department of Information Technology, Politeknik Negeri Samarinda

References

- [1] S. Theodoridis and K. Koutroumbas, *Pattern Recognition - Second Edition*. America: Academic Press - An imprint of Elsevier, 2003.
- [2] E. Hussain, A. Hannan, and K. Kashyap, "A Zoning based Feature Extraction method for Recognition of Handwritten Assamese Characters " *International Journal of Computer Science and Technology*, vol. 6, 2015.
- [3] O. de Lima, S. Janakiraman, E. Saber, D. C. Day, P. Bauer, M. Shaw, R. Twede, and P. Lea, "Signature line detection in scanned documents," pp. 3254-3258, 2016.
- [4] F. S. Mohamad, F. M. Alsuhiat, Mohamad Afendee Mohamed, M. Mohamad, and A. A. Jamal, "Detection and Feature Extraction for Images Signatures," *International Journal of Engineering & Technology*, vol. 7, 2018.
- [5] A. J. S, P. C. R. T. V, S. K. K. B. Raja, D. Anvekar, V. K. R, S. S. Iyengar, L. M. Patnaik, A. C, and P. K. S, "OSPCV: Off-line Signature Verification using Principal Component Variances " *IOSR Journal of Computer Engineering (IOSR-JCE)*, vol. 7, 2015.
- [6] S. Nam, H. Park, C. Seo, and D. Choi, "Forged Signature Distinction Using Convolutional Neural Network for Feature Extraction," *Applied Sciences*, vol. 8, p. 153, 2018.
- [7] R. Kumar, S. Kaur, and B. Gupta, "Signature Recognition using Statistical-ANN Hybrid Technique," *International Journal of Scientific Research and Engineering Development*, vol. 2, 2019.
- [8] T. Kutzner, C. Pazmiño-Zapatier, M. Gebhard, I. Bönninger, W.-D. Plath, and C. Travieso, "Writer Identification Using Handwritten Cursive Texts and Single Character Words," *Electronics*, vol. 8, p. 391, 2019.
- [9] B. B. Rad and M. Masrom, "Metamorphic Virus Detection in Portable Executables Using Opcodes Statistical Feature," *International Journal of Advanced Science Engineering and Information technology (IJASEIT)*, vol. 1, 2017.
- [10] M. Sharif, M. A. Khan, M. Faisal, M. Yasmin, and S. L. Fernandes, "A framework for offline signature verification system: Best features selection approach," *Pattern Recognition Letters*, 2018.
- [11] N. Umniati, A. B. Mutiara, T. M. Kusuma, and S. Widodo, "Component-connected Feature for Signature Identification," *International Journal on Advanced Science Engineering Information Technology (IJASEIT)*, vol. 8, 2018.
- [12] C. S. V, P. Mukherjee, G. DS, and V. Pulabagari, "Online Signature Verification Based on Writer Specific Feature Selection and Fuzzy Similarity Measure," *ResearchGate - publication/333259994*, 2019.
- [13] M. R. Kaur and M. P. Choudhary, "Handwritten Signature Verification Based on Surf Features Using HMM," *International Journal of Computer Science Trends and Technology (IJCTST)*, vol. 3, 2015.
- [14] M. A. Mohamad, D. Nasien, H. Hassan, and H. Haron, "A Review on Feature Extraction and Feature Selection for Handwritten Character Recognition," *(IJACSA) International Journal of Advanced Computer Science and Applications*, vol. 6, 2015.
- [15] T. Jadhav, "Handwritten Signature Verification using Local Binary Pattern Features and KNN," *International Research Journal of Engineering and Technology (IRJET)*, vol. 6, 2019.
- [16] L. LI, B. ZHANG, L. GAO, and Y. WU, "Spectral Feature Extraction Based on Orthogonal Polynomial Function Fitting," *2014 8th IAPR Workshop on Pattern Recognition in Remote Sensing, Stockholm, Sweden, IEEE*, 2014.
- [17] M. Mehri, P. Herouet, J. Lerouget, P. Gomez-Kramer, and R. Mullot, "A Structural Signature Based on Texture for Digitized Historical Book Page Categorization," *2015 13th International Conference on Document Analysis and Recognition (ICDAR), Tunis, Tunisia, IEEE*, 2015.
- [18] Z. Shen, X. Ma, and Y. Li, "A Hybrid 3D Descriptor With Global Structural Frames and Local Signatures of Histograms," *IEEE Access*, vol. 6, pp. 39261-39272, 2018.
- [19] P. Wei, H. Li, and P. Hu, "Inverse Discriminative Networks for Handwritten Signature Verification," *Computer Vision and Pattern Recognition (CVPR), IEEE Xplore*, 2019.

- [20] P. Cao, W. Xia, M. Ye, J. Zhang, and J. Zhou, "Radar-ID: human identification based on radar micro-Doppler signatures using deep convolutional neural networks," *IET Radar, Sonar & Navigation*, vol. 12, pp. 729-734, 2018.
- [21] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Writer-independent Feature Learning for Offline Signature Verification using Deep Convolutional Neural Networks," *International Joint Conference on Neural Networks (IJCNN)*, Vancouver, BC, Canada, IEEE, 2016.
- [22] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Learning features for offline handwritten signature verification using deep convolutional neural networks," *Pattern Recognition*, vol. 70, pp. 163-176, 2017.
- [23] X. He, Y. Wei, H. Zhang, T. Zhang, F. Yuan, Z. Huang, F. Han, and B. Song, "Grading of Clear Cell Renal Cell Carcinomas by Using Machine Learning Based on Artificial Neural Networks and Radiomic Signatures Extracted From Multidetector Computed Tomography Images," *Acad Radiol*, May 27 2019.
- [24] V. Kleber Santos Leite Melo and B. Byron Leite Dantas, "A Fully Convolutional Network for Signature Segmentation from Document Images," *6th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Niagara Falls, NY, USA, IEEE, pp. 540-545, 2018.
- [25] P. Maergner, V. Pondenkandath, M. Alberti, M. Liwicki, K. Riesen, R. Ingold, and A. Fischer, "Combining graph edit distance and triplet networks for offline signature verification," *Pattern Recognition Letters*, vol. 125, pp. 527-533, 2019.
- [26] S. T. Panchal and V. V. Yerigeri, "Offline signature verification based on geometric feature extraction using artificial neural network," *IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)*, vol. 13, 2018.
- [27] V. Ruiz, I. Linares, A. Sanchez, and J. F. Velez, "Off-line handwritten signature verification using compositional synthetic generation of signatures and Siamese Neural Networks," *Neurocomputing*, 2019.
- [28] V. L. F. Souza, A. L. I. Oliveira, and R. Sabourin, "A writer-independent approach for offline signature verification using deep convolutional neural networks features," *ResearchGate - publication/326697052*, 2018.
- [29] K. B. Johnston and A. M. Peter, "Variable Star Signature Classification using Slotted Symbolic Markov Modeling," *New Astronomy*, vol. 50, pp. 1-11, 2017.
- [30] J. J. d. M. Sá Junior, L. C. Ribas, and O. M. Bruno, "Randomized neural network based signature for dynamic texture classification," *Expert Systems with Applications*, vol. 135, pp. 194-200, 2019.
- [31] Y. Serdouk, H. Nemmour, and Y. Chibani, "Handwritten signature verification using the quad-tree histogram of templates and a Support Vector-based artificial immune classification," *Image and Vision Computing*, vol. 66, pp. 26-35, 2017.
- [32] A. N. Azmi, D. Nasien, and F. S. Omar, "Biometric signature verification system based on freeman chain code and k-nearest neighbor," *Multimedia Tools and Applications*, vol. 76, pp. 15341-15355, 2016.
- [33] J. Poddar, V. Parikh, and S. K. Bharti, "Offline Signature Recognition and Forgery Detection using Deep Learning," *Procedia Computer Science 170, Elsevier*, 2020.
- [34] A. F. O. Gaffar, A. B. W. Putra, A. Wajiansyah, and Supriadi, "Texture Feature Extraction based on Local Weighting Pattern (LWP) using Fuzzy Logic Approach " *International Journal of Computer Applications (IJCA)*, vol. 179, pp. 1-8, 2018.
- [35] N. M. Tahir, A. N. Ausat, U. I. Bature, K. A. Abubakar, and I. Gambo, "Off-line Handwritten Signature Verification System: Artificial Neural Network Approach," *International Journal of Intelligent Systems and Applications*, vol. 13, pp. 45-57, 2021.
- [36] H. A. Al-Jubouri, "Integration Colour and Texture Features for Content-based Image Retrieval," *International Journal of Modern Education and Computer Science*, vol. 12, pp. 10-18, 2020.
- [37] A. Salam Shah, M. N. A. Khan, and A. Shah, "An Appraisal of Off-line Signature Verification Techniques," *International Journal of Modern Education and Computer Science*, vol. 7, pp. 67-75, 2015.
- [38] M. M. Yapıcı, A. Tekerek, and N. Topaloglu, "Deep learning-based data augmentation method and signature verification system for offline handwritten signature," *Pattern Analysis and Applications*, vol. 24, pp. 165-179, 2020.
- [39] R. K. Mohapatra, K. Shaswat, and S. Kedia, "Offline Handwritten Signature Verification using CNN inspired by Inception V1 Architecture," *2019 Fifth International Conference on Image Information Processing (ICIIP)*, IEEE, Shimla, India, 2019.
- [40] M. J. Evans and J. r. S. Rosenthal, *Probability and Statistics - The Science of Uncertainty, Second Edition*: W. H. Freeman, 2009.
- [41] V. Conti, "Biometric Authentication Overview: a Fingerprint Recognition Sensor Description," *International Journal of Biosensors & Bioelectronics*, vol. 2, 2017.
- [42] R. Damasevicius, R. Maskeliunas, E. Kazanavicius, and M. Wozniak, "Combining Cryptography with EEG Biometrics," *Comput Intell Neurosci*, vol. 2018, p. 1867548, 2018.
- [43] J. Malik and D. Girdhar, "Reference Threshold Calculation for Biometric," *International Journal of Image, Ghaphic and Signal Processing*, vol. 2, pp. 46-53, 2014.
- [44] A. B. W. Putra, A. F. O. Gaffar, A. Wajiansyah, and I. H. Qasim, "Feature-Based Video Frame Compression Using Adaptive Fuzzy Inference System " in *2018 International Symposium on Advanced Intelligent Informatics (SAIN)*, IEEE, Yogyakarta, Indonesia, pp. 49-55, 2018.

Authors' Profiles



Arief Bramanto Wicaksono Putra. Born in Balikpapan, January 20, 1983. Completed undergraduate (D4) majoring in Information Technology at Electronic Engineering Polytechnic Institute of Surabaya in 2006. Completed postgraduate study of Electrical Engineering Department at Brawijaya University Malang in 2014. Beginning in 2008 working as a lecturer in the Department of Information Technology, State Polytechnic of Samarinda until now. His representative published articles two years ago list as follow : In 2019 with IEEE conference published as Prediction of The Topographic Shape of The Ground Surface Using IDW Method through The Rectangular-Neighborhood Approach, Feature-Based Video Frame Compression Using Adaptive

Fuzzy Inference System, Steganography for Data Hiding in Digital Audio Data using Combined Least Significant Bit and 4-Wrap Length Method, and Measurement of Electrical Power Usage Performance using Density Based Clustering Approach. In 2020 with IEEE conference published as A Deep Auto Encoder Semi Convolution Neural Network for Yearly Rainfall Prediction, A multi-frame blocking for signal segmentation in voice command recognition. And Journal article list : Implementation of the Naive Bayes Classifier Method for Potential Network Port Selection (MECS publisher with SCOPUS Indexing). Image transformation using fuzzy-based filtering for the texture class's measurement based on the distance of the feature and Magic cube puzzle approach for image encryption (SCOPUS indexing), Areas of interest: Computer Vision, Computer Networks, Robotics & Artificial Intelligent



Bedi Suprpty. Born in Samarinda, December 10, 1978. Completed undergraduate (S1) majoring in Information Technology at STMIK Widya Cipta Daharma Samarinda in 2001. Completed postgraduate study of Information System Department at Diponegoro University Semarang in 2013. Beginning in 2002 working as a lecturer in the Department of Information Technology, State Polytechnic of Samarinda until now. Her representative published articles list as follow : Design of Information System for Acceptance Selection of Prospective Employees Online Using Tahani Fuzzy Logic Method and Simple Additive Weighting (2016), Rainfall prediction using fuzzy inference system for preliminary micro-hydro power plant planning (2018, SCOPUS Indexing). Areas of interest: Human Computer Interaction, Intelligent Computing, Artificial

Intelligent (AI).



Mulyanto, received the bachelor's degree of computer science from University of Indonesia 1999 and then received the master of computer science from Gadjah Mada University at 2016.

He is a lecture at Department of Information Technology, State Polytechnic of Samarinda. His current research interests include modelling and simulation, artificial intelligent and intelligent computing. (<https://www.scopus.com/authid/detail.uri?authorId=57202191130>)



Achmad Fanany Onnilita Gaffar, Born in Surabaya, October 23, 1969. Completed undergraduate (S1) majoring in Electrical Engineering at Sepuluh Nopember Institute of Technology Surabaya in 1995. Completed postgraduate study of Electrical Engineering Department at Brawijaya University Malang in 2011. Beginning in 1998 working as a lecturer in the Department of Electrical Engineering, as a lecturer at the Department of Information Technology, State Polytechnic of Samarinda from 2012 until now. In the last 4 years, Scopus indexed articles have been produced 20 articles in the form of journals and IEEE proceedings (<https://www.scopus.com/authid/detail.uri?authorId=57193740086>), Areas of interest: Robotics & Artificial Intelligent, Intelligent Control. Machine Learning, Computer Vision, Data Science

How to cite this paper: Arief Bramanto Wicaksono Putra, Mulyanto, Bedi Suprpty, Achmad Fanany Onnilita Gaffar, " A Performance of Combined Methods of VCG and 16BCD for Feature Extraction on HSV", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.13, No.3, pp. 13-32, 2021.DOI: 10.5815/ijigsp.2021.03.02