Arabic letter recognition based on image processing.

Ainatul Radhiah[1], Carmadi Machbub[2], Egi M.I. Hidayat[3], Ary S. Prihatmanto[4],
Sekolah Teknik Elektro dan Informatika, Institusi Teknologi Bandung
Jl Ganesha 10, Bandung, Indonsia
[1]ainawind27@gmail.com, [2]carmadi@stei.itb.ac.id, [3]egi@lskk.ee.itb.ac.id, [4]asetijadi@lskk.ee.itb.ac.id,

Abstract **- In this research will be designed a system that can recognize isolated Arabic letters and Arabic letters in a sentence. System has five stage: pre-processing, thinning, segmentation, feature extraction and classification. In the pre-processing stage is done by binarization, the image is converted to a binary that have value 0 and 1. In the thinning stage is done with a Stentiford algorithm that has 4 templates, end point and number of conetivities to check whether an image can be deleted or not. In segmentation stage letter segmentation is done by Zidouri algorithm. In the feature extraction is done by 3 features that extracted, the first is normalized chain code, the second is number of dots, and the third is the position of dots. In the classification stage is done by Neural Network and Hidden Markov Model. The results show that the recognition of isolated Arabic letters with the Neural Network classification method reached 100% accuracy and the recognition result of the Arabic letters in the sentence reached 69% accuracy. While the recognition results of the isolated Arabic letters with Hidden Markov Model classification method reached 71% accuracy and the recognition result of the Arabic letters in the sentence reached 50% accuracy.**

Keywords: Arabic Letter Recognition , Stentiford Algorithm, Chain code, Neural Network, Hiden Markov Model

I. INTRODUCTION

Arabic language is used by more than 1 billion people in the world [1]. Arabic has 28 base letters and 3 additional letters that written right to left and and written cursively both printed or handwriting. Therefore the recognition of Arabic letters in sentences requires a segmentation process. Some Arabic letters have a similar shape and can be distinguished from the number of dots and the position of the dots. Each Arabic letter has a different shape, depending on it's position in the sentence, that is isolated, at the beginning, in the middle and at the end.

Previous research on the recognition of Arabic letters has grown. Nimas [2] and colleagues (2017), conducted research on the recognition of isolated Arabic letters using Neural Network with Backpropagation learning method and Learning Vector Quantisation (LVQ). The results showed that the recognition with Backpropagation achieved 98.81% accuracy and the recognition results with LVQ reached 51.19% accuracy. M. Albakor [3] and colleagues (2009) have conducted a reasearch on the recognition of the Arabic letters, this research involves a segmentation process, which produces an accuracy of 98.7%. Albadr [4] (2013) develop an Arabic letter recognition system in a sentence by extracting 24 features, including chain code, and classification using the decision tree generated by C4.5 algorithm, recognition results achieved 48% accuracy. Izakian [5] (2008), develop an isolated Farsi / Arab letters recognition system using the Support Vector Machine, the results achieved 97.4% accuracy.

Based on previous research, research on the recognition of the Arabic letters contained in the sentence has not been widely known. Therefore in this research will be developed an Arabic letter recognition system in isolated form and in the sentence.

The purpose of the development of Arabic letter recognition system is to help the process of learning Arabic letters either in isolated form or in sentence. As for the limitations of the problem in this research is, the letters used are printed Arabic letters in isolated positions and in sentences.

II. METHODOLGY

This research was conducted in 5 stages, the following is an explanation of each stage that is implemented.

2.1 Binarization

Binarization of the image is the process of converting the image into binary that have values 0 and 1. Grayscale image will be changed to black and white. A binarization process is required to perform the next steps on the recognition of Arabic letters and sentences. The way it does is by doing threshold on each color channel. The threshold used is 150. If the color channel is less than 150 it will be converted to black, and if the color more than 150 will be changed to white.

2.2 Thinning

One of the uses of thinning is in the pattern recognition application. The image used is the thinning that has been done binarization so that the image becomes binary image. This process erodes the pixels as much as possible without affecting the general shape. After thinnning process the pattern should still be recognized. The resulting image of the thinning algorithm is called the skeleton.

There are several popular thinning algorithms, including Zhang Suen, Stentiford and Hilditch. In this study the Stentiford algorithm was chosen as the best thinning algorithm. After a comparison between Zhang Suen, Stentiford and Hilditch. In the case of thinning Arabic letters, the algorithm of Zhang Suen and Hilditch has a deficiency in thinnning results. Figure 1 show comparison of the thinning results of the letters "ث" with Zhang Suen, Stentiford and Hilditch algorithms.

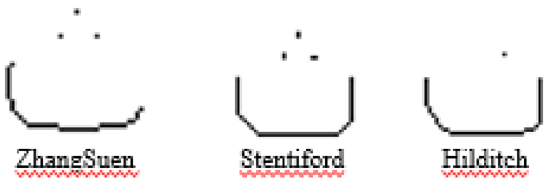


Figure 1. comparison of thinning algorithms

In Figure 1 can be seen that the result of thinning with Zhang Suen algorithm removes the right part of the letter, which should not be deleted, as in the thinning result with the Stentiford algorithm. While the results of thinning with Hilditch algorithm remove 2 dots of ث letters, so the ث letter has only 1 dot, which should have 3 dots. Thinning results with Stentiford algorithm look perfect without any mistake.

Stentiford algorithm uses a set of four templates to scan the image, that is T1, T2, T3, and T4 as shown in figure 1.

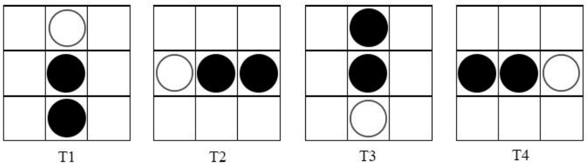


Figure 2. Templates of Stentiford Algorithm

Here are the steps to get the skeleton of an image with the Stentiford algorithm [6] :

1. Initially locate the pixel (i, j) that matches the T1 template. Matching this template moves from left to right and from top to bottom.
2. If the middle pixel is not an endpoint and has a number of connectivity = 1, then mark pixels for later deletion.
   The endpoint is a pixel which is the end limit and is only connected 1 pixel only. That is, if the black pixel has only one black neighbor of the eight possible neighbors.
   The number of connectivity is a measure of how many objects are connected to a certain pixel. Here is the formula to calculate the number of connectivity.

   $$C_n = \sum_{k \in S} N_k - (N_k \cdot N_{k+1} \cdot N_{k+2})$$

   Where:
   N_k is the value of the 8 neighbors around the pixels to be analyzed, and the value S = {1,3,5,7}
   N_0 is the value of the middle pixel.
   N_1 is the value of the pixel on the right of the central pixel and the rest are numbered sequentially in the opposite direction of the clock
3. Repeat steps 1 and 2 for all pixels that match the T1 template.
4. Similarly follow the above mentioned steps 1-3 for the templates: T2, T3, and T4.
5. Template T2, T3, and T4 matches the left, bottom, and right side of the image.
6. Pixels marked for deletion are set to white

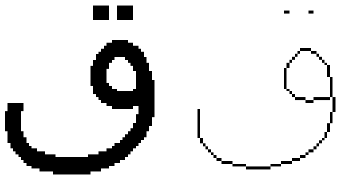


Figure 3. Result of Stentiford Thinnning algorithm

2.3 Segmentation

The letter segmentation was done using the Zidouri algorithm [7]. Here is the stage of zidouri segmentation.

Consider the following notation :

θ = Width of single dot in the document
Ls = Width of smallest character
Ls' = Width of two smallest character if appear together
Lm = Maximum Width of charcter in isolated form
B(x,y) = Location of Baseline
I = Image of sub-word

I’ = Image of sub-word without dots

E = Empty image of size I

Remove dots from character.

Steps in character segmentation

1. Skeletonize I'
2. Scan from right to left in row-wise fashion, to find a band of horizontal pixels having length >= Ls
3. Take vertical projection on the scanned band found in step 2. If no pixel is encountered, draw a vertical guide band on E.
4. Use special mark for the guide bands, which are drawn due to the scanned band (found in step 2) below the baseline B (x, y).
5. Repeat the procedure, for all the rows.

After performing above-mentioned steps, an image E with several guide bands is obtained. In order to select, correct guide band for sub-word dissection, we extract several features from each guide band:

Feature	Description
F1	Width of the guide band
F2	Distance from 1st predecessor from right, zero in case of 1st guide band
F3	Distance from 2nd predecessor from right, zero in case of 1 and 2nd guide band.
F4	1 if guide band drawn due to scanned band is above baseline 0 if guide band drawn due to scanned band is below baseline
F5	Midpoint of guide band

The judicious selection of guide band is driven through several rules. The feature sets {F1...F5} of each guide band are tested for each rule. If it satisfies rules then it is selected otherwise it is rejected.

Rule 1 : Choose guide band having highest relative width (F1) and F4 = 1

Rule 2 : Choose guide band if F2 > Ls and F4 = 1

Rule 3 : Choose guide band if F2 <= Ls and F3 > Ls' and guide band is not the last one.

Rule 4 : Choose guide band if F1 >= Lm and F4 = 1

For the 1 st guide band in the sets, even if it fails to qualify Rule 1 – 4 and the guide band next to it satisfies Rule 2 then it shoukd be selected.

If all guide bands fail to satisfy any rule, then apply less constrained rule base i.e., removing F4 condition except Rule 4.

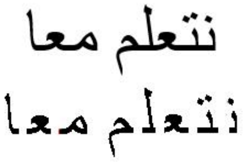


Figure 4. Results of Zidouri letter segmentation

2.4. Feature Extraction

In this research there are 3 stage feature extraction, that is number of dots, position of dots and chain code. Here is an explanation of each feature:

2.4.1 Chain code

In pattern recognition, chaincode is a technique to describe a structure of an object. Chain code is obtained by tracing the pixels of the object boundary based on predetermined directions. The result of the chain code is the numbers that indicate the direction that represents the boundary of the object. Chain code can only be done on binary image.

Here is how to extract the chain code of an object in an image:

1. Find a black pixel that has only 1 neighbor by tracing the pixels in the image starting from the top left corner until it finds a black pixel that has 1 neighbor, if not found a black pixel that has only 1 neighbor then grab the first black pixels encountered.
2. Do iteration on the image:
   1. Change the current pixel to 0
   2. Follow the priority of directions 1 to 8
   3. Move the pixel position
   4. Append direction to the chain code

The length of the chain code of an object changes according to the shape of an object. In this research will be classified with Neural Network and Hidden Markov Model. The input of the neural network must be fixed, not changeable. Therefore in this research will be normalized chain code. That is, making the length of the chain code of a fixed image in number and not changing.

2.4.1.1 Normalized Chain Code

In this research the chain code of the object will be normalized to 10 for each object of the letter image. steps 1 and 2 follow the steps developed by Izakian [5], and steps 3 and 4 were developed in this research. The following are the steps of chain chain normalization:

1. Chain code is converted into 2 dimensional matrix. The first line is the value of the chain code. The second line is the frequency of occurrence of each number in the chain code.Like the following chain code: 77773111222225833533333, After the first stage of chain chain normalization will be 2 x 9 matrix:
   7 3 1 2 5 8 3 5 3
   4 1 3 5 1 1 2 1 4

2. Eliminate all values that have only 1 frequency.
   7 3 1 2 5 8 3 5 3 ➡ 7 1 2 3
   4 1 3 5 1 1 2 1 4 4 3 5 6
3. Show chain code according to frequency of occurrence:
   777711122222333333
4. Perform chain code mapping to 10 chain code, the formula is :

   N3 : 777711122222333333

   N4[i] = n3[round(i/9 x N3length-1)

   Normalized chaincode is :

   7711222333

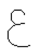
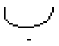
	1 8 6 5 4 8 6 6 4 4
	6 6 6 6 6 6 6 6 6 6
	6 6 6 6 5 4 1 1 8 8
	5 5 5 6 6 7 8 8 8 8

Figure 4 Chain codes for some example Arabic characters

2.4.2 Number of Dots

The feature of the number of dots is an important feature in Arabic letters, since some Arabic letters have the same shape but are only differentiated by the number of dots. Such as character ث ,ب, and ت.

The number of dots obtained by iterating on the image of the letter from the top left corner to the right, then to the bottom, if found the first black point calculate the chain code. Letters that have dots will have more than 1 chaincode. Then it will be checked, if the length of chaincode that found less than 7 then it will be calculated as chaincode of dots, and do summation of the number of dots. If the chaincode has a length greater than 7 it will be counted as the chaincode of the letter body.

2.4.3 Position of Dots

Position of dots is an important feature of Arabic letters. Some Arabic letters have the same shape and number of dots, but are distinguished by the position of dots.

Dots position is obtained by calculating dot position of letter and height of letter. The image will be divided into 5 parts. If dot position is in the position of less than 2/5 the image height then the position of dots is above which is represented with the number 0. If dot position is in position less than 3/5 the height of the image then the position of dots is in the middle which is represented with the number 1. If dot position is on position of more than 3/5 image height then position of dots is below which represented with number 2

Following the computation of the chain code, dot count and dot position, the arrangement of features will be as follow:

F = [*DotCount*, *DotPos*, *Chain_Code*]

2.5 Classification

The classification stage is performed with Neural Network and Hidden Markov Model (HMM).

2.5.1 Classification Using Neural Network

Artificial Neural Networks are a computational system whose network structure mimics the human nervous system in order to produce responses and behaviors such as biological Neural Networks. Information processing systems on Artificial Neural Networks have similar features of bilogical nervous tissue.
The following is how simple the Neural Network works compared to the biological neural network:

1. Processing of signals or information occurs in neurons.
2. Signals are sent between the neurons through a link, the dendrites and axons.
3. Liaison between neurons has a weight that will strengthen or weaken the signal.
4. Each neuron has an activation function that serves to determine the output of a neuron, whether the signal will be forwarded to another neuron or not.

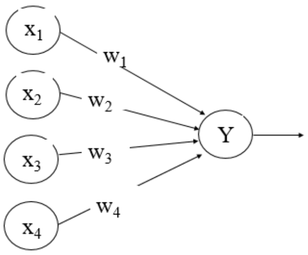


Figure 5. Neuron Models

Neurons are the principal information processing units of artificial neural networks that act on the impulses they receive and are transmitted to other neurons.
Neurons consist of 3 main elements, namely:

1. Group of of units connected by path
2. The summing unit that sums the input signal already multiplied by its weight.

3. The activation function used to determine the output of a neuron, that is determining whether the signal from the neuron input will be forwarded to another neuron or not.

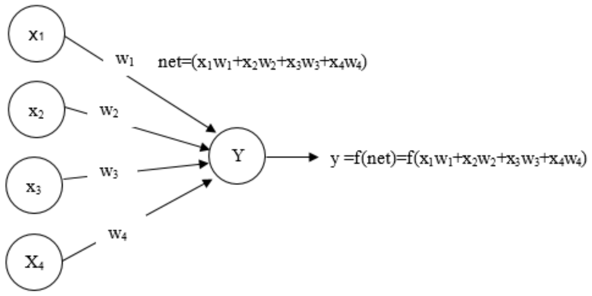


Figure 6. Summation of neuron impuls (net)

Information :

1. y receives input from neurons x1, x2, x3 and x4.
2. Neurons will transform the information received through the outgoing connection to other neurons. In artificial neural networks these relationships are called weights. Each neuron has a weight characterized by w1, w2, w3 and w4. The information is stored at a certain value on the weight.
3. Input information will be sent to neurons with certain weight values. The input will be processed by a function that sums up the values of all the weights that come. In Figure 6 the neuron impulses are summed:
   net = (x1w1 + x2w2 + x3w3) (1)
4. The sum result will be compared with a threshold value through the activation function of each neuron. The amount of impulse received by y follows the activation function y = f (net)
5. If the value of the activation function is strong enough (passed a certain threshold value), the signal will be forwarded to another neuron

In this research type of network architecture of neural network that used is network plural layer with 1 hidden layer. The activation function used for the hidden layer is the sigmoid activation function, and the activation function used for the output layer is the softmax activation function. Input of neurons for each sample is 12, the first neuron is the number of dots, the second neuron is the position of the dots and the third neuron is the chain code that has been normalized.

The following is the formulation of sigmoid activation function.

$$f(x) = \frac{1}{(1+e^{-x})} \qquad (2)$$

The following is the formulation of softmax activation function.

$$f(x) = = \frac{e^{xi}}{\sum_{j=0}^{k} +e^{xj}} \qquad (3)$$

2.5.2 Classification using Hidden Markov Model

HMM is a stochastic process in which one process can not be observed (hidden). This unobservable process can only be observed through a process that can be observed [8].

Basically HMM consists of three things[8]:

1) Evaluation

Evaluation is the process of calculating the probability of the observation sequence on the HMM model. Evaluation using forward and backward algorithm.

- Forward Algorithm

If the forward variable α_t (i), at time t and state i, then the equation

$$\alpha_t(i) = P(O_1,\ O_2,\ \ldots,\ O_T,\ q_t = i|\lambda \qquad (4)$$

With O = the matrix index is observed Resolution with n circumstances and observations until Q is iterated

Initialization :

$$\alpha_t(i) = \ \pi_i b_i(O_1),\ 1 \le i \le n \qquad (5)$$

Where π = initial state matrix and b_i (O_1) = the first observed matrix.

Induction :

$$\alpha_{t+1}(j) = \ \left[\sum_{i=1}^{n_i} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}) \ldots \qquad (6)$$

Where n_i = the number of states and a_ij = transition matrix.

Termination :

$$P(O|\lambda) = \sum_{i=1}^{n_i} \alpha_t(i) \qquad (7)$$

- Backward Algorithm

The state flows backward from the last observation at t. The backward probability equation β_t (i) is as follows:

$$\beta_t(i) = P(O_{t+1},\ O_{t+2},\ \ldots,\ O_T,\ q_t = i|\lambda) \qquad (8)$$

And analogous to forward procedure β_t (i) with step:

Initialization

$$\beta_t(i) = 1 \qquad 1 \le i \le n \qquad (9)$$

Induction

$$\alpha_{t+1}(j) = \ \sum a_{ij}\, b_j(O_{t+1})\beta_{t+1}(j),$$

$t = T-1, T-2, \dots, 1$,

$1 \leq i \leq n$ (10)

2) Decoding
Decoding is done to find the best state of observation sequence in HMM model with viterbi algorithm. The steps are as follows.
Initialization:
$\delta_t(i) = \pi_i b_i(O_1), 1 \leq i \leq n$ (11)

Rekursi :

$\delta_t(j) = \frac{max}{1 \leq i \leq n}\left[\delta_{t-1}(i)a_{ij}\right]b_j(O_t, 1 \leq j \leq n$ (12)

Terminasi :

$P = \frac{max}{1 \leq i \leq n}[\delta_t(i)]$ (13)

Where P = Probability

3) Parameter Estimation (Learning)
Baum - Welch algorithm performs learning to obtain parameters on the HMM model.
Parameter A
$A = \{a_{ij}\}$, for $1 \leq i, j \leq n$
$a_{ij} = P\left[q_{t+1} = X_j \,|q_t = X_i\right]$, Odds when state j at time t + 1 if at time t is in state I.
a_{ij} = Probability of transition from state I to state j
P = Probability
q_{t+1} = condition after q_t
q_t = condition when t
X_j = state j
X_i = state i
n = The number of hidden states in the model
Parameter B

$B = \{b_j(k)\}, for\ 1 \leq j \leq n, 1 \leq k \leq m$

$b_j(k) = P[y_k \text{ to } t\,|q_t = X_j]$
$0 \leq b_j(k)$ dan $\sum_{k=1}^{m} b_j(k) = 1$
$b_j(k)$ = Probability of observation matrix distribution
y_k = Probability with index k cluster in state j*m* = The number of different observation symbols in each state
In continuos density HMM is often characterized by density function or mixed density function in each state [13,14]. Assuming the use of Gaussian Mixture, the state emission density *j* is defined as:
$b_j(o_t) = \sum_{k=1}^{K} w_{jk}\mathcal{N}(O; \mu_{jk}; \Sigma_{jk}), j = 1,2, \dots, N$ (14)

Where K is the number of mixture and w_jk is the mixing coefficient for the Gaussian k state of state j with the stochastic constraint:

$\sum_{k=1}^{K} w_{jk} = 1j = 1,2, \dots, N$ (15)

where $\mathcal{N}$ Shows Gaussian density function with mean $\mu_{jk} \epsilon\ R^d$ A covarian matrix $\sum_{jk} \epsilon\ R^{dxd}$ for k^{th} *mixture.*

Initial matrix at state i

$$\pi = \{\pi_i\}$$

$$\pi_i = P[q_1\, X_i],\ 0 \leq \pi_i \text{ dan} \sum_{i=1}^{n} \pi_i = 1$$

So HMM can be denoted by

$\lambda = (A, B, \pi)$.

In this research the number of observe sequences is 12, consists of number of dots, position of dots and normalized chain code.

III. Experiment and Discussion

Java is used to develop code. For training data 3 fonts used, those are Arial Unicode Ms, Tahoma and Times New Roman. For testing data 3 fonts used, those are Arial Unicode Ms, Tahoma, Times new Roman. 31 isolated Arabic letters with 3 different fonts used as an isolated Arabic recognition test data, and 10 Arabic sentences used for the recognition of Arabic letters in sentences. Table I shows the experimental results with Neural Network classification. The recognition of isolated Arabic letters yields recognition accuracy with an average of 100%, and for the recognition of Arabic letters in sentences yields recognition accuracy wit ang average of 69%.

Table 1. Performance of Arabic Recognition with Neural Network Classification

Font	Accuracy of Isolated Arabic Character	Accuracy of Arabic Character In Sentence
Arial Unicode Ms	100%	66%
Tahoma	100%	66%
Times New Roman	100%	75%

Table II shows the experimental results with Hidden Markov Model classification. The recognition of isolated Arabic letters yields recognition accuracy with an average of 71%, and for the recognition of Arabic letters in sentences yields recognition accuracy wit ang average of 50%.

Table 2. Performance of Arabic Recognition with Hidden Markov Model Classification

Font	Accuracy of Isolated Arabic Character	Accuracy of Arabic Character In Sentence
Arial Unicode Ms	74%	49%
Tahoma	61%	50%
Times New Roman	77%	51%

IV. CONCLUSION

The chain-code-based approach for Arabic letter recognition has been a key feature in this research, to improve the recognition accuracy the number of dots and position of dots feature has been added, these three features have been able to provide different features for each letter so that the results obtained are quite good. The results showed that classification with Neural Network get better result compared to Hidden Markov Model.

REFERENCE

[1] Ismail, B., Fahd, B., and Yassine, S. (2013): 0Arabic reading machine for visually impaired people using TTS and OCR, *4th International Conference on Intelligent Systems, Modelling and Simulation*, 1.

[2] Nimas, A. M., Victor, A., and Nashrul H. (2016): Comparative analysis of the accuracy of backpropagation and learning vector quantisation for pattern recognition of hijaiyah letters, *6th International Conference on Information and Communication Technology for The Muslim World*, 4.

[3] M. Albakor, K. Saeed, and F. Sukkar. (2009): Intelligent system for Arabic character recognition, *World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, 1.

[4] Iping, S., and Albadr, N. (2013): Arabic character recognition system development, *The 4th International Conference on Electrical Engineering and Informatics (ICEEI 2013)*, 1.

[5] H. Izakian, S. A. Monadjemi, B. Tork, L., K. Zamanifar. (2008): Multi-font farsi/arabic isolated character recognition using chain codes, *World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol:2, No:7*, 1, 3.

[6] P. Anil, K. (2014): Edge detection based on otsu method and stentiford algorithm, *International Journal of Engineering Research & Technology (IJERT)*, 3.

[7] Zidouri, A. (2010): On multiple typeface arabic script recognition, *Research Journal of Applied Sciences Engineering and Technology*, 3.

[8] Devi, H., Hanif, F., Egi, M. I. H., and Carmadi M. (2016): Comparison of Indonesian speaker recognition using vector quantization and hidden markov model for unclear pronunciation problem, *IEEE 6th International Conference on System Engineering and Technology (ICSET)*, 3.