Javanese Character Recognition Using Hidden Markov Model

Anastasia Rita Widiarti, and Phalita Nari Wastu

***Abstract*—**Hidden Markov Model (HMM) is a stochastic method which has been used in various signal processing and character recognition. This study proposes to use HMM to recognize Javanese characters from a number of different handwritings, whereby HMM is used to optimize the number of state and feature extraction. An 85.7 % accuracy is obtained as the best result in 16-stated vertical model using pure HMM. This initial result is satisfactory for prompting further research.

***Keywords*—**Character recognition, off-line handwriting recognition, Hidden Markov Model.

I. Introduction

THE richness of Javanese culture is stored in many ancient books written in Javanese handwritings. Unfortunately, only a few people could actually read those manuscripts. As these manuscripts have valuable contribution to knowledge, automatic off-line handwriting recognition systems are needed to provide bigger access to these manuscripts.

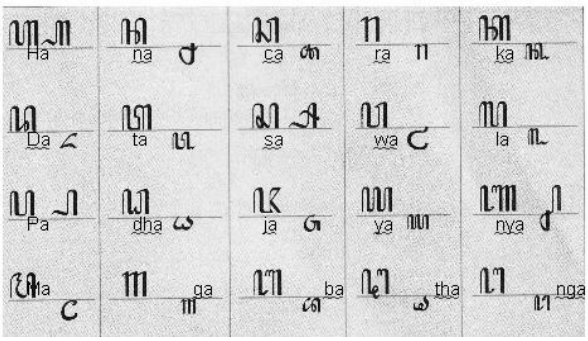

Fig. 1 The Legena of Javanese character [2]

Different methods have been applied in many handwriting recognition researches. One of the most successful and popular method is Hidden Markov Model, which works well not only for word or character recognition, but also for signal processing. In this study HMM is used to recognize Javanese handwritings. As mentioned by T.E. Behren [1], Javanese script has 103 characters, i.e. 20 main phonetic characters called *legena* or *dentawyanjaya*, 20 characters of *pasangan*, and 63 character of *sandhangan* which include all numbers and special characters used in poems and folksongs. The characters recognized in this study are the *legena*, as shown in Fig. 1.

The main goal of this study is to find the accuracy of HMM to recognize the shape of Javanese characters. The secondary goal is to figure out which feature extraction performs the best result among other feature extraction possibilities.

Anastasia Rita Widiarti is a faculty member at the department of Informatics Engineering, Sanata Dharma University, in Yogyakarta, Indonesia (corresponding author to provide phone: +6281328341628; fax: 01-0274-886529; e-mail:rita_widiarti@yahoo.com).

Phalita Nari Wastu is a student at the department of Informatics Engineering, Sanata Dharma University, in Yogyakarta, Indonesia (e-mail: sweetdevilnawa@gmaill.com).

II. Pre Processing

To perform the complete tasks of off-line handwriting recognition, selected Javanese documents are scanned and pre-processed to filter the characters from background noise. The scanned documents are 1549x2340 pixel RGB images, which is then transformed into binary images. Segmentation into separated characters follows, continued by resizing the image to make all characters into the same size 72x72 pixels image.

III. Feature Extraction

Feature is an instance used as a model in Hidden Markov Method. This instance is extracted during the training and testing. Extraction is done by taking the discrete pixel information from each vector. Because there is no previous work that provides information about the best Javanese handwriting feature extraction, the best form within possible extractions should be found. Vertical and horizontal feature extraction is then conducted, adopted from the articles by Nopsuwanchai and Povey [3], and by Theeramunkong at al [5]. A Thai feature extraction is adapted to use in this study based on the type similarity of Thai and Javanese characters. Four (4) feature extractions for the Javanese characters are:

1. Character divided into 1 horizontal vector (1H).
2. Character divided into 2 horizontal vectors (2H).
3. Character divided into 1 vertical vector (1V).
4. Character divided into 2 vertical vectors (2V).

Fig. 2 shows an example of four feature extraction on *Ha* character.

IV. Hidden Markov Models (HMM)

HMM is a stochastic system which is assumed from the Markov chain with unknown parameters, and the challenge is to find the hidden parameter(s) from observed parameters [4]. Continuous models are performed during the signal

processing, otherwise discrete HMM is done at the image processing as this study does.

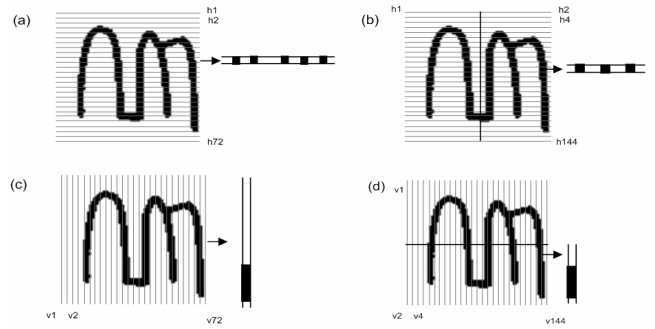

Fig. 2 Feature extractions on Javanese character recognition. (a) 1H; (b) 2H; (c) 1V; (d) 2V;

After features are extracted, a Markov model is able to be generated. From the discrete data, HMM generates a chain that consists of a number of states with transition probabilities that make these states connect to others as shown in Fig. 3.

As explained by L. R. Rabiner [6], Hidden Markov Model can be generated from parameters:

$$\lambda = (A, B, \pi) \tag{1}$$

Where:

1. N is the number of state, denoted as S = { S_1, S_2, S_3, ..., S_n } and state at the time t is q_t.
2. M is the number of observed symbol at any state. Individual symbols denoted as V = { V_1, V_2, V_3, ..., V_m }.
3. Transition probability matrices A = {a_{ij}},
   where $a_{ij} = P (q_{t+1} = S_j \mid q_t = S_i)$, $1 \le i, j \le N$ (2)
   $a_{ij} \ge 0$ and $\sum_{j=1}^{N} a_{ij} = 1$
4. The observation symbol probability distribution in state j. B = { bj(k) } where
   $b_i(k) = P (v_k \text{ in } t \mid q_t = S_j)$, (3)
   $1 \le j \le N$, $1 \le k \le M$
5. The initial state distribution π = { π_i }, where
   $\pi_i = P (q_i = S_j)$, $1 \le i \le N$ (4)

At the appropriate value of N, M, A, B and π, HMM can be used as a generator to give an observation sequence:

$$O = O_1 O_2 O_3 \ldots O_t \tag{5}$$

where each observation O_t is one of the symbols from V, and T is the number of observations in the sequence.

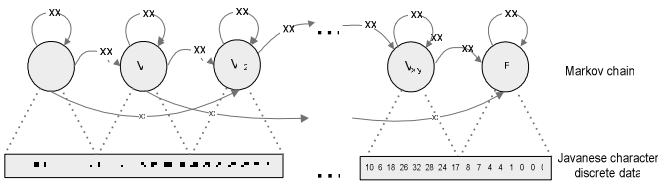

Fig. 3 Figure of Markov chain generated from the "ha" character

A. Data Trains

Every *legena* character is modelled by the forward-backward algorithm or Baum-Welch algorithm. This algorithm is used to find means and variance as maximum likelihood to model all 20 *legena* characters, calculated from the observation sequence O and HMM $\lambda = (A, B, \pi)$.

Forward variable is defined as partial observation from sequence state probability, denoted as O_1 , O_2 ,...,O_t (until time t) and state S_i at the time t, with λ and α as *t(i)* as shown in Fig. 4. Backward variable is defined as partial observation from sequence state probability t+1 to the current state, where state S_i at the time t, with λ and α as *t(i)* as shown in Fig. 5. This sequence state probability is denoted as:

$$P(O \mid \lambda) = \sum_{i=1}^{N} \alpha_t(i)\beta_t(i) = \sum_{i=1}^{N} \alpha_t(i) \tag{6}$$

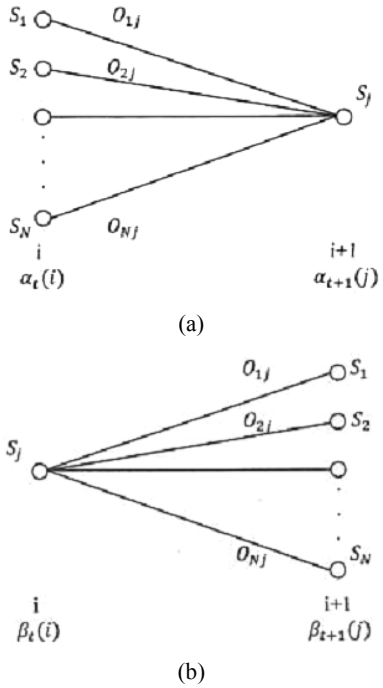

Fig. 5 Illustration of Baum-Welch operations. (a) Forward variable α_{t+1} (b) Backward variable β_{t+1} [6]

Probability at the state *Si* at the time t, with observation sequence O, and HMM λ, can be denoted as:

$$\gamma_t = \frac{\alpha_t(i)\beta_t(i)}{P(O \mid \lambda)} \tag{7}$$

A Vector Quantifier (*VQ*) is needed to map each observation vector to an indexed codebook [6]. Baum-Welch definitions to estimate the new model λ= (*A*,*B*, Π) are:

$$\overline{\pi} = \gamma_1(i) \tag{8}$$

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \zeta_t(i,j)}{\sum_{t=1}^{T} \gamma(i)} \quad (9)$$

$$\bar{b}_j(V_k) = \frac{\sum_{t=1 s.t. O_t = v_k}^{T} \gamma_t(i)}{\sum_{t=1}^{T} \gamma_t(i)} \quad (10)$$

with $$\zeta_t(i,j) = \frac{\alpha_t(i)\alpha_{ij}b_j(O_{t+1})\beta_{t+1}(i)}{P(O \mid \lambda)} \quad (11)$$

Where:

1. $\bar{\pi}_i$ is the expected frequency in state S_i at time $t=1$,
2. $\bar{a}_{ij}$ is the expected number of transitions from state S_i to state S_j divided by the expected number of transitions from state S_i,
3. $\bar{b}_j(k)$ is the expected number of times in state j and observing symbol υ_k divided by the expected number of times in state j.

B. The Use of Viterbi Algorithm in Testing Phase

Best path can be found by using the Viterbi algorithm. Fig. 5 shows illustration of search best path, and path 12, 22 is the best path probability. The closest probability state sequence $Q = \{q_1 q_2 \ldots q_T\}$ is calculated to the observed sequence $O = \{O_1 O_2 \ldots O_T\}$ that can be defined as:

$$\delta_T(i) = \max_{q_1, q_2, \ldots, q_t} = P[q_1 q_2 \ldots q_T = i, O_1 O_2 \ldots O_T \mid \lambda] \quad (12)$$

$\delta_T(i)$ is the best probability at time T, which is calculated at first observation T and ends in state S_i.

.

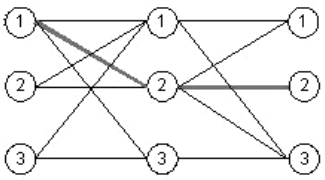

Fig. 5 Maximum path found by Viterbi algorithm. Bold line is the best path probability

V. Implementation

The system is used to recognize *legena* characters from the books written by Raden Ngabehi Yasadipura [7] and [8]. Data input is scanned images from these books, and the process can be seen in Fig. 6. Images that are inputted into the system are the pre-processed images, which are extracted from the training and testing as seen in Fig. 7.

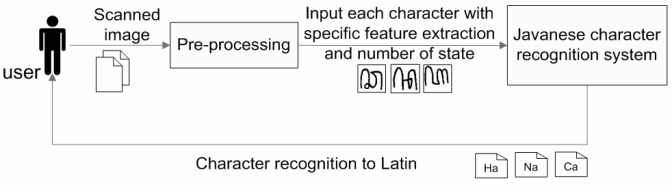

Fig. 6 Illustration of general flow diagram for the recognition system

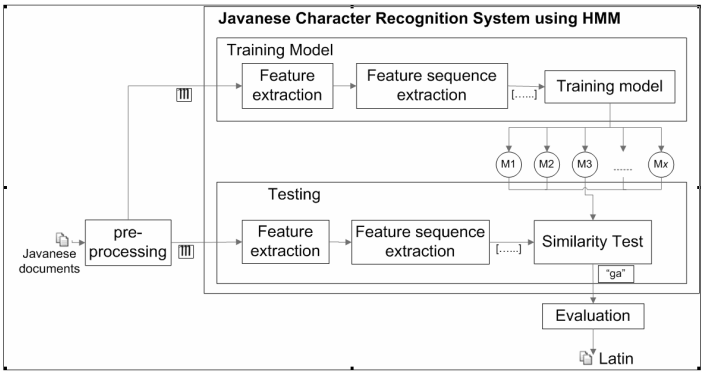

Fig. 7 Block diagram of the recognition system

VI. Experimental Result

This study uses 1000 Javanese characters as input consisting of 20 *legena* characters with 50 samples for each character. A 5-fold cross validation method is performed, which tests 200 sets of *legena*, and trains the remaining 800.

1. Experiment I

   Components:

   a. 20 character of *legena*, which are *ha, na, ca, ra, ka, da, ta, sa, wa, la, ma, ga, bha, tha, nga, pa, dha, ja, ya, nya*.
   b. Total character is 20x50 = 1000.
   c. Characters are resized to be 72 x 72 binary images (0 and 1 pixel information).
   d. Characters are divided into 72 horizontal vectors (1H).
   e. The data inputted are characters which have been thinned and also character which have not been thinned.

As result of experiment I look in the Table II, and the result of 5-fold cross validation see Table I.

TABLE I
CONFUSION MATRIX FOR THE 18-STATED 1H FEATURE

	ha	*na*	*ca*	*ra*	*ka*	*da*	*sa*	*wa*	*la*	*pa*	*dha*	*Ja*	*ya*	*nya*	*ma*	*ga*	*ba*	*tha*	*nga*
ha	26	0	0	0	0	0	0	1	0	3	0	0	9	0	0	0	0	0	0
na	0	46	0	0	2	0	0	0	0	0	0	2	0	0	0	0	0	0	0
na	1	1	46	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
ra	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ka	0	2	0	0	46	0	0	0	0	0	0	0	0	2	0	0	0	0	0
da	0	0	0	0	0	44	0	0	0	0	4	0	0	0	0	0	0	2	0
ta	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0
sa	0	0	8	0	0	0	42	0	0	0	0	0	0	0	0	0	0	0	0
wa	0	0	0	0	0	2	0	35	1	0	8	0	0	0	0	1	0	2	1
la	2	0	4	0	0	0	0	12	25	0	0	0	0	0	0	0	0	0	0
pa	0	0	0	0	0	0	0	11	4	35	0	0	0	0	0	0	0	0	0
dha	0	0	0	0	0	4	2	0	0	0	41	0	0	0	0	0	0	1	2

ja	0	0	0	0	0	0	0	0	0	0	0	48	0	0	2	0	0	0	0
ya	7	0	0	0	0	0	0	0	0	0	0	0	43	0	0	0	0	0	0
nya	0	0	1	0	2	0	0	0	0	0	0	0	0	39	0	0	8	0	0
ma	0	0	0	0	0	0	0	0	0	0	0	0	0	0	47	2	1	0	0
ga	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0
ba	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0
tha	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	47	3
nga	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	48

TABLE II
ACCURACY AT 1H FEATURES

Number of State	Accuracy (%)
15	81
16	81
17	84,4
18	85,4
19	83,4
20	83,2
21	83,8
22	81,7

2. Experiment II, III, and IV

The remaining three experiments can be seen in Table III. The Table III shows that the highest accuracy occurs at 21-stated 2V feature, but it has a higher number of state compared to the 16-stated 1V feature. A higher number of state means more time needed to process the recognition.

TABLE III
ACCURACY AT 2H, 1V AND 2V FEATURES

Number of State	2H Feature (%)	1V Feature (%)	2V Feature (%)
15	68,8	85,0	74,1
16	69,1	85,7	76,2
17	71,2	83,2	74,3
18	71,2	85,1	75,6
19	71,8	83	79,9
20	70,8	83,3	81,5
21	72,6	79,6	86,4
22	71,5	77,5	81,7

VII. CONCLUSION AND FUTURE WORK

This study experiments off-line recognition to 20 types of *legena*, with 50 samples for each character. Characters are extracted by the 1V, 2V, 1H and 2H feature. The best feature extraction is found in the 85.7% accuracy of 1V feature, with the consideration being the least number of states to suppress the time complexity. Best accuracy for each feature is shown in Table IV.

TABLE IV
BEST ACCURACY AT FEATURE 1H, 2H, 1V AND 2V

Type of feature	Number of State	Accuracy (%)
1H	18	85.4
2H	21	72.6
1V	16	85.7
2V	21	86.4

Hidden Markov Models yields a good accuracy in Javanese-script character recognition, and it has good raw data handle and flexibility to input some parameters. This research should be continued for character recognition at the level of words, sentences, and even documents. Meanwhile, prototype has not yet been implemented for the rest of 103 Javanese characters.

ACKNOWLEDGMENT

This research was supported by Indonesian Director General of Higher Education (DIKTI) and Sanata Dharma University of Yogyakarta Indonesia.

REFERENCES

[1] T.E. Behren, *A. Sĕrat Jatiswara: Struktur dan Perubahan di dalam Puisi Jawa 1600-1930*. Jakarta: Indonesian-Netherlands Cooperation in Islamic Studies (INIS), 1995.

[2] Sam Muharto, and W. Nataatmaja, *Trampil Basa Jawa 5: Jilid 5 kangge Kelas V SD/ MI.* Solo: PT. Tiga Serangkai Pustaka Mandiri, 2008.

[3] Roongroj Nopsuwanchai, and Dan Povey, "Discriminative Training for HMM-Based Offline Handwritten Character Recognition". *IEEE in the Seventh International Conference on Document Analysis and Recognition (ICDAR 2003)*.

[4] Teresa M. Przytycka, *Encyclopedia of The Human Genome: Hidden Markov Models*. USA: Nature Publishing Group, 2007.

[5] T. Theeramunkong, C. Wongtapan, and S. Sinthupinyo, "Off-line Isolated Handwritten Thai OCR Using Islandbased Projection with N-gram Models and Hidden Markov Models," IEEE. 2001.

[6] L. R. Rabiner, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," *Proceeding of the IEEE*, vol 77, pp. 257-286, 1989.

[7] R. Ngabehi Yasadipura I, *Menak China II*. Bantanisentrem: Bale Pustaka, 1934.

[8] R. Ngabehi Yasadipura I, *Menak Sorangan*. Batavia: Bale Pustaka, 1936.

Anastasia Rita Widiarti was born in Gunungkidul, Daerah Istimewa Yogyakarta, Indonesia, on April 24, 1969. She received the master degree in computer science from Gadjah Mada University, Yogyakarta, Indonesia, in 2006. Since 2000 she has been teaching at department informatics engineering faculty of science and technology at Sanata Dharma University. Her current research interest is Javanese document image analysis, and pattern recognition.