International Journal of Electrical and Computer Engineering (IJECE)
Vol. 9, No. 3, June 2019, pp. 1711~1719
ISSN: 2088-8708, DOI: 10.11591/ijece.v9i3.pp1711-1719

Moment invariant-based features for Jawi character recognition

Fitri Arnia, Khairun Saddami, Khairul Munadi
Departement of Electrical and Computer Engineering, Syiah Kuala University, Indonesia

Article Info

Article history:

Received Oct 20, 2018
Revised Nov 26, 2018
Accepted Dec 11, 2018

Keywords:

Feature extraction
Jawi character recognition
Moment invariant
Optical character recognition
Pattern classifier
Tree root (TR) algorithm

ABSTRACT

Ancient manuscripts written in Malay-Arabic characters, which are known as "Jawi" characters, are mostly found in Malay world. Nowadays, many of the manuscripts have been digitalized. Unlike Roman letters, there is no optical character recognition (OCR) software for Jawi characters. This article proposes a new algorithm for Jawi character recognition based on Hu's moment as an invariant feature that we call the tree root (TR) algorithm. The TR algorithm allows every Jawi character to have a unique combination of moment. Seven values of the Hu's moment are calculated from all Jawi characters, which consist of 36 isolated, 27 initial, 27 middle, and 35 end characters; this makes a total of 125 characters. The TR algorithm was then applied to recognize these characters. To assess the TR algorithm, five characters that had been rotated to 90o and 180o and scaled with factors of 0.5 and 2 were used. Overall, the recognition rate of the TR algorithm was 90.4%; 113 out of 125 characters have a unique combination of moment values, while testing on rotated and scaled characters achieved 82.14% recognition rate. The proposed method showed a superior performance compared with the Support Vector Machine and Euclidian Distance as classifier.

Corresponding Author:

Fitri Arnia,
Department of Electrical and Computer Engineering,
Faculty of Engineering, Syiah Kuala University,
Jl. Syech Abdurrauf No.7, Banda Aceh, Indonesia.
Email: f.arnia@unsyiah.ac.id

1. INTRODUCTION

Handwriting character recognition has become one of the most interesting research areas in the digital image processing field. The increased variety of characters to be recognized implies that the research into character recognition still gaining high attention and is still necessary. The mainstream research in character recognition includes Latin, Chinese, Arabic, Japanese, and Indian. In addition, the current research comprises the variation of Arabic characters, such as Urdu, Farsi, and Jawi.

Indonesia, especially Aceh, has many relic books that are written in Jawi characters. Jawi or Malay–Arabic terminology refers to the language written in Arabic letters but verbalized in Malay language. The Malay language has been used in Southeast Asia, including Indonesia, Malaysia, Brunei, Singapore, South Thailand, South Philippines, and South Myanmar [1]. There is a slight difference between Arabic and Jawi character, in which the latter has six additional letters adapted to the Malay language. The additional letters are letter ڤ for "p", letter ڽ for "ny", letter ڠfor "ng", letter چ for "Ca", letter ݢ for "Ga", and letter ۏ for "Va". Moreover, Jawi characters commonly do not use vowel marks (harakah) [2]. The list of Jawi characters can be seen in Table 1, in which the six additional characters are shown in bold. There are 36 isolated characters, 27 initial-form and middle-form characters, and 35 end-form characters. The total number of characters is 125.

Table 1. Jawi Characters

No	Character Name	Isolated	Beginning	Middle	End
1	Alif	ا			ـا
2	Ba	ب	بـ	ـبـ	ـب
3	Ta	ت	تـ	ـتـ	ـت
4	Tsa	ث	ثـ	ـثـ	ـث
5	Jim	ج	جـ	ـجـ	ـج
6	Ca	چ	چـ	ـچـ	ـچ
7	Ha	ح	حـ	ـحـ	ـح
8	Kha	خ	خـ	ـخـ	ـخ
9	Da	د			ـد
10	Dza	ذ			ـذ
11	Ra	ر			ـر
12	Za	ز			ـز
13	Sin	س	سـ	ـسـ	ـس
14	sya	ش	شـ	ـشـ	ـش
15	Sha	ص	صـ	ـصـ	ـص
16	Dha	ض	ضـ	ـضـ	ـض
17	Tha	ط	طـ	ـطـ	ـط
18	Zha	ظ	ظـ	ـظـ	ـظ
19	Ain	ع	عـ	ـعـ	ـع
20	Ghain	غ	غـ	ـغـ	ـغ
21	Nga	ڠ	ڠـ	ـڠـ	ـڠ
22	Fa	ف	فـ	ـفـ	ـف
23	Pa	ڤ	ڤـ	ـڤـ	ـڤ
24	Qa	ق	قـ	ـقـ	ـق
25	Kaf	ك	كـ	ـكـ	ـك
26	Ga	ݢ	ݢـ	ـݢـ	ـݢ
27	Lam	ل	لـ	ـلـ	ـل
28	Mim	م	مـ	ـمـ	ـم
29	Nun	ن	نـ	ـنـ	ـن
30	Nya	ڽ	ڽـ	ـڽـ	ـڽ
31	Waw	و			ـو
32	Va	ۏ			ـۏ
33	Haa	ه	هـ	ـهـ	ـه
34	Ya	ي	يـ	ـيـ	ـي
35	hamzah	ء			ء
36	Lamalif	لا			ـلا
	Total	36	27	27	35

The recognition of Jawi characters is important because Jawi characters were widely used in the Southeast Asian countries for 600 years [3]. The Jawi characters were used for writing Malay language, and can be found on many tombstones, currency, royal decrees, books, and so on. Furthermore, almost 15,000 manuscripts are written in Jawi with various contents, namely, tales, theology, and poems that can be found in museums all over the world in, for instance, England, the Netherlands, Malaysia, Indonesia, and many other countries. The ancient manuscripts not only hold a record of the history and culture of Southeast Asian countries, but also contain medical prescriptions, military strategies, navigation techniques, and other useful knowledge [4].

Research in the digital image and computer vision field has yielded optical character recognition (OCR) for Chinese [5], Arabic [6], [7], Indian [8], and Latin characters [9], but no OCR system has yet been presented for Jawi characters. An OCR application framework requires four steps, namely: (1) preprocessing [10], [11], [12]; (2) segmentation; (3) feature extraction [13], [14]; and (4) pattern classiffiication [15]. At present, few algorithms for Jawi character recognition exist and these recognition algorithms have only been tested on a limited number of Jawi characters. For instance, they have been tested only on the isolated characters [16]. In recent years, character recognition of Arabic variants, such as Farsi and Urdu, has gained attention. However, less attention has been paid to Jawi characters compared with Urdu and Farsi. Thus, the development of a Jawi character recognition algorithm with a complete dataset is indeed required so the ancient Jawi manuscripts can be utilized in easier ways.

Table 2 shows the details of the accuracy levels of character recognition methods using pattern classifiers. The method proposed by Razak that used Hamming classification achieved the highest accuracy levels, with 97% recognition rate, but it was not robust to scaling operations [17]. The methods in Table 2 need a training process. In common, the number of training characters is proportional to the recognition accuracy; the greater the number of training characters, the higher the recognition accuracy is. Moreover, another problem of a pattern classifier established using local optima solutions, such as neural networks, is that it may decrease the accuracy level [18]. On the other hand, methods such as Support Vector Machine

(SVM) and Hidden Markov Models (HMM) that are based on global optimum solutions, require longer time and more memory to execute the program. The Adaboost algorithm fails due to noise. In Adaboost, there are two points of weakness: the learning process and classification process [19]. After all, the result of each pattern classifier is influenced by the feature selection [20]. The method proposed by Heryanto et al. was able to reach 95.67% recognition rate, but the testing process reached an accuracy of 73.59% [21].

Table 2. Performance of Pattern Classifier in Jawi Character Recognition

No	Research	Methods	Scope application	Recognition Accuracy
1	Nasrudin [14]	Invariant Future, Trace transform, MLP	Isolated printed Jawi	Not details
2	Nasrudin [16]	Triple Feature, SVM, MLP, Bayesian, etc	Handwritten Jawi	72.02%
3	Razak [17]	Discrete Wavelet Transform (DWT), Hamming classification	Jawi	97%
4	Heryanto [21]	Feature Vector, Hybrid ANN, Dynamic programming	Jawi word isolated	Training = 95.67% Testing = 73.59%
5	Redika [22]	Feature Vector, HMM	Jawi words	84%
6	Nasrudin [23]	Triple Feature TT, EC, CC	Isolated Printed and handwritten Jawi	Printed = 80.56% Handwritten = 69,44 %

Machine learning algorithms such as Artificial Neural Network (ANN) and SVM have been used to classify a pattern, and widely applied in character recognition [24]. It is used to perform the classification or recognition in a phase known as the training phase. The training method categorizes the extracted features into several classes of similar patterns. The ANN has been used since the 1980s, and at the end of 1990s SVM was introduced as new and better classification method [25]. The drawback of machine learning is the method requires experimentation method to model the training phase [26]. SVM as part of machine learning method shows more superior performance in character recognition system than other methods such as ANN and K-nearest Neighbor (KNN) [27]. However, the SVM processes information as a holistic pattern and fails in training local pattern. Furthermore, SVM consumes more storage and requires more complex computation than other learning algorithms such as ANN and HMM. Another disadvantage of SVM is the requirement to determine the kernel methods.

In this article, we propose a novel technique for Jawi character recognition system, and we refer it to as the Tree Root (TR) Algorithm. The algorithm does not use complex mathematical computation, but performs simple classification process and requires less storage memory. It does not require training process; it saves feature template for recognizing process. The method was tested with a dataset of Jawi handwriting with the complete character set: 36 isolated, 27 initial-form, 27 end-form, and 35 middle-form characters. The following section discusses the proposed method. In section three, we discuss the experimental process of the proposed algorithm. Section four presents and discusses the results, and finally, the conclusion is presented in section five.

2. PROPOSED METHOD

In digital image processing and computer vision, the moment invariant is mostly used for describing object characteristics. There are two types of feature extractions based on moment invariants: contour-based or shape-based moment invariants [28] and region-based moment invariants [29]. Moment invariants are very useful for extracting object features with unique characteristics regardless of location, size, and orientation. Hu proposed seven invariant moments, known as Geometric Moment Invariants (GMI) that are invariant toward translation, rotation, scaling, and mirroring as follows [30].

$$
\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} \\
\phi_2 &= (\eta_{20} + \eta_{02})^2 + 4\eta_{11}{}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\
\phi_6 &= (\eta_{20} - \eta_{02})\left[(\eta_{30} + 3\eta_{12})^2 - 3(\eta_{21} - \eta_{03})^2\right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\
&\quad + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]
\end{aligned}
\tag{1}
$$

2.1. The tree root (TR) algorithm

Based on moment values (1), we developed the tree root (TR) algorithm. In the algorithm, M_n denotes the nth moment, with $n = \{1,2,3,4,5,6,7\}$ and G_{nr} refers to a group of characters with the nth moment and moment value r. For example, the character group of first moment with $r = 1$ is denoted as G_{11}. The TR algorithm is explained as follows.

1. Calculate seven moment values of all characters.
2. Round the moment value to the smallest r integer number.
3. Set $n = 1$.
4. Group all characters based on their moment values namely G_{nr}.
5. Count the members of all G_{nr}, if r is unique, then save the moment value r and n and the character is unique and recognized. Otherwise, $n = n + 1$.
6. Check n value, if $n = 7$, the character in this group G_{nr} is unrecognizable. Otherwise, return to step 4.

3. RESEARCH METHOD

In this section, we present the research method for testing the TR algorithm. The experiment was conducted in three stages: (1) data collection and template production; (2) feature extraction process; and (3) testing.

3.1. Data collection and template production

The Jawi characters were collected according to the following steps: (1) write the characters on the paper; (2) scan the paper and save the character image; (3) crop the character part in the image, by positioning the cropped part in the middle of the image; and (4) normalize the images into 160×140 pixels. The Jawi character template was prepared in two forms: binary and thin. Examples of binary template Jawi characters are shown in in Figure 1, while examples of thin template Jawi characters are shown in Figure 2.

Figure 1. Examples of binary template Jawi characters

Figure 2. Examples of thin template Jawi characters

3.2. Feature extraction process

In the feature extraction process, (1) was used to calculate all moment values of the character images. The moment values were stored and used as feature in recognition process. Based on these values, the TR algorithm recognizes the characters. The moment values of the character were classified using the proposed method.

3.3. Testing

The TR algorithm was tested on sets of rotated and scaled characters, to find out whether the algorithm can recognize the rotated and scaled Jawi characters. The characters were rotated 90o and 180o to the right and scaled by scale factors of 2 and 0.5. Fourteen characters were chosen, and each character was chosen to represent each shape group of the Jawi alphabet. We used seven characters from the binary template: isolated jim, end-form ha, middle-form ya, initial ga, initial ng, isolated pa, and end-form pa, and seven characters from thin template, consisting of isolated alif, isolated kaf, middle-form haa, middle-form ya, end-form ra, middle-form ha, and end-form va.

Furthermore, The SVM and Euclidean distance (ED) as classifier were applied to compare the performance of the proposed method. We applied the SVM classifier and ED to the moment values

calculated by (1), and then compared the results with that of the proposed method. In data training for SVM and ED, we used features of 125 characters for each binary and thin template. In testing stages, we used both rotated and scaled characters. In SVM training, we used one against all strategy for classifying each class of character. After training, we tested the classified class using the testing characters. The ED value was computed between the features, i.e., the moment values of binary or thin template of training and testing character. The ED value was sorted, and the smallest ED value indicated the recognized character.

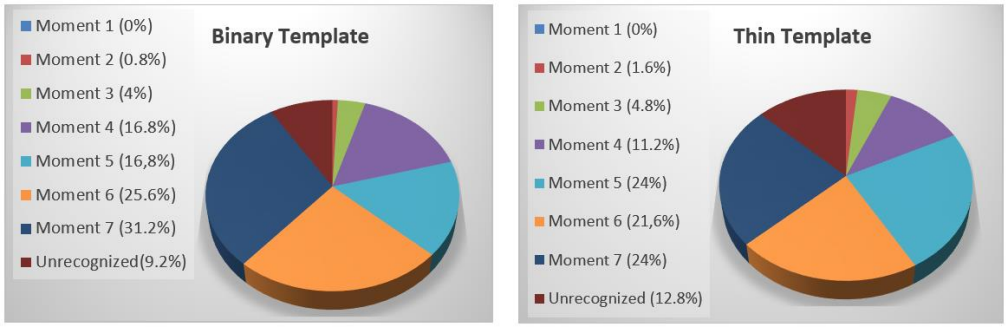
4. RESULTS AND ANALYSIS

The TR algorithm classified the Jawi characters into seven groups. The first group consisted of one, i.e., the first moment value. In this case, the character is represented by one, i.e., the first moment values. The second group consisted of two, i.e., the first and the second moment values. In this case, the character is represented by two, i.e., the first and the second moments. Similar rules are applied to the remaining groups. Within the same group, each character is differentiated by a unique value of its highest moment.

4.1. Recognition results of binary template characters

Figure 3a shows the percentage recognition rate in each "moment group" of binary template characters. Based on the results, there was one character that can be represented by two moment, i.e., 0.8%. Five characters were represented by three moments, or 4%. The characters that can be recognized by four moments were 15 characters or 12%. There were 21 characters recognizable by five moments, or 16.8%. Furthermore, recognizable characters by six and seven moments were 32 characters (25.6%) and 39 characters (31.2%), respectively. From a total of 125 characters, there were twelve characters that cannot be recognized as unique characters. The percentage of unrecognizable characters using Hu's moment invariant features is 9.6%.

Figure 3b shows the recognition rate of thin template characters. There were two characters that can be represented by two moments or 1.6%. The percentage of recognizable characters based on three moments was 4.8% or 6 characters from the total of 125 characters. Up to 14 characters or 11.2% and 30 characters or 24% can be recognized by four and five moments, respectively. Furthermore, the recognized character with six moments was 27 characters or 21.6%, as the recognition rate with seven moments was 30 characters (24%). Sixteen characters were not recognizable as unique characters, which is 12.8% of the total of 125 characters.



(a) (b)

Figure 3. Percentage of recognition in each "moment group" of:
(a) binary template characters, (b) thin template characters

4.2. Testing with rotated and scaled characters

Tables 3 and Table 4 show the test results with binary and thin characters, including rotated and scaled characters. Seven characters were chosen as the testing character, and shown in 'input character' column. Each character was rotated by 90° and 180° and was scaled by two scaling factor: 0.5 and 2. The unrecognized characters of binary template as shown in Table 3 are Jim initial-form (rotated 180°), Ha end-form (rotated 180°), P end-form (rotated 90°), G initial-form (0.5 factor scaled), and Ng initial-form (0.5 factor scaled). Table 3 shows that the proposed method can recognize eleven rotated and twelve scaled characters from total 28 characters.

As shown in Table 3, character Jim in initial form has different moment value for each scaled factor; the character ➔ with scaled factor of 0.5 has the moments 1, 4, 12, 10, 21, . . ., while the character ➔ with scaled factor of 2 has the moments 1, 4, 12, 10, 22, ... However, because the moment value of each scaled

character matches the saved feature, that was the first four moments, 1, 4, 12, 10, the character was recognized.

In addition, Table 4 shows that Kaf isolated-form, Haa middle-form, R end-form, Ya middle-form, Ha end-form, V end-form, and alif isolated-form are the unrecognized characters of thin template. Table 4 presents nine rotated and eight scaled characters form 28 characters succeed to be recognized. These results suggest that the TR algorithm is robust to rotation and scaling operations, as will be discussed in the following.

Table 3. TR Algorithm Testing on Binary Characters

			Rotated character		
No	Input character	Angle (degree)	Hu moment	Saved feature	Output
1.	ﺝ (Jim initial-form)	90	1,4,12,10,22,12,25	1,4,12,10	ﺝ (Jim initial-form)
		180	1,5,12,10,22,13,25	1,4,12,10	unrecognized
2.	ـﺤ (Ha end-form)	90	1,4,11,9,20,12,21	1,4,11,9,20,12	ـﺤ (Ha end-form)
		180	1,6,11,9,20,12,20	1,4,11,9,20,12	unrecognized
3.	ـﻳ (Ya middle-form)	90	1,5,16,13,28,17,29	1,5,16	ـﻳ (Ya middle-form)
		180	1,5,16,13,28,17,29	1,5,16	ـﻳ (Ya middle-form)
4.	ﮒ (G initial-form)	90	1,5,14,12,25,15,25	1,5,14,12,25	ﮒ (G initial-form)
		180	1,5,14,12,25,15,27	1,5,14,12,25	ﮒ(G initial-form)
5.	ﻎ (Ng initial-form)	90	1,5,14,9,23,13,21	1,5,14,9,23	ﻎ (Ng initial-form)
		180	1,5,14,9,23,13,22	1,5,14,9,23	ﻎ (Ng initial-form)
6.	ﭫ (P isolated-form)	90	1,5,14,9,22,13,22	1,5,14,9,22,13	ﭫ (P isolated-form)
		180	1,5,14,9,22,13,23	1,5,14,9,22,13	ﭫ (P isolated-form)
7.	ـﭫ (P end-form)	90	1,6,14,9,23,13,22	1,5,14,9,23	unrecognized
		180	1,5,14,9,23,13,22	1,5,14,9,23	ﭫ (P end-form)
			Scaled character		
No	Input character	Scaling factor	Hu moment	Saved feature	Output
1.	ﺝ (Jim initial-form)	0.5	1,4,12,10,21,12,23	1,4,12,10	ﺝ (Jim initial-form)
		2	1,4,12,10,22,12,23	1,4,12,10	ﺝ (Jim initial-form)
2.	ـﺤ (Ha end-form)	0.5	1,4,11,9,20,12,20	1,4,11,9,20,12	ـﺤ (Ha end-form)
		2	1,4,11,9,20,12,20	1,4,11,9,20,12	ـﺤ (Ha end-form)
3.	ـﻳ (Ya middle-form)	0.5	1,5,16,13,28,17,29	1,5,16	ـﻳ (Ya middle-form)
		2	1,5,16,13,28,17,29	1,5,16	ـﻳ (Ya middle-form)
4.	ﮒ (G initial-form)	0.5	1,5,14,11,25,15,26	1,5,14,12,25	unrecognized
		2	1,5,14,12,25,15,27	1,5,14,12	ﮒ (G initial-form)
5.	ﻎ (Ng initial-form)	0.5	1,5,14,9,22,13,22	1,5,14,9,23	Unrecognized
		2	1,5,14,9,23,13,22	1,5,14,9,23	ﻎ (Ng initial-form)
6.	ﭫ (P isolated-form)	0.5	1,5,14,9,22,13,23	1,5,14,9,22,13	ﭫ (P isolated-form)
		2	1,5,14,9,22,13,23	1,5,14,9,22,13	ﭫ (P isolated-form)
7.	ـﭫ (P end-form)	0.5	1,5,14,10,23,15,22	1,5,14,10,23,15	ـﭫ (P end-form)
		2	1,5,14,10,23,15,23	1,5,14,9,23	ـﭫ (P end-form)

Table 4. TR Algorithm Testing on Thin Characters

			Rotated character		
No	Input character	Angle (degree)	Hu moment	Saved feature	Output
1.	ﻙ (Kaf isolated-form)	90	1,5,15,20,37,23,39	1,4	unrecognized
		180	1,5,15,20,37,23,36	1,4	unrecognized
2.	ـﻫ (Haa middle-form)	90	1,5,18,16,36,22,34	1,5,17,9	unrecognized
		180	1,5,18,16,36,22,34	1,5,17,9	unrecognized
3.	ـﻳ (Ya middle-form)	90	1,5,21,20,41,23,42	1,5,21,20	ـﻳ (Ya middle-form)
		180	1,5,21,20,41,23,41	1,5,21,20	ـﻳ (Ya middle-form)
4.	ﺭ (R end-form)	90	1,5,17,18,37,21,37	2,5,17,18	unrecognized
		180	2,5,17,18,37,21,37	2,5,17,18	ﺭ (R end-form)
5.	ـﺤ (Ha end-form)	90	1,5,23,16,37,19,37	1,5,23	ـﺤ (Ha end-form)
6.		180	1,5,23,16,37,19,36	1,5,23	ـﺤ (Ha end-form)
7.	ﻒ(V end-form)	90	1,6,16,16,33,21,33	1,6,16	ﻒ(V end-form)
		180	1,6,16,16,33,21,32	1,6,16	ﻒ(V end-form)
	ﺍ (Alif isolated-form)	90	1,5,21,21,43,24,45	1,5,21,21	ﺍ (Alif isolated-form)
		180	1,5,21,21,43,24,44	1,5,21,21	ﺍ (Alif isolated-form)
			Scaled character		
No	Input character	Scaling factor	Hu moment	Saved feature	Output
1.	ﻙ (Kaf isolated-form)	0.5	1,4,16,14,31,19,30	1,4	ﻙ (Kaf isolated-form)
		2	1,4,16,14,31,19,30	1,4	ﻙ (Kaf isolated-form)
	ـﻫ (Haa middle-form)	0.5	1,5,17,19,39,23,37	1,5,17,9	ـﻫ (Haa middle-form)
2.		2	1,5,17,19,37,22,39	1,5,17,9	ـﻫ (Haa middle-form)
	ـﻳ (Ya middle-form)	0.5	1,5,20,18,39,22,37	1,5,21,20	unrecognized
3.		2	1,5,21,20,41,23,42	1,5,21,20	ـﻳ (Ya middle-form)
	ﺭ (Ra end-form)	0.5	2,5,16,18,36,21,37	2,5,17,18	unrecognized
4.		2	2,5,17,18,37,21,37	2,5,17,18	ﺭ (R end-form)
	ـﺢ (Ha end-form)	0.5	1,5,22,16,36,19,35	1,5,23	unrecognized
5.		2	1,5,21,16,35,19,35	1,5,23	unrecognized
6.	ﻒ(V end-form)	0.5	1,6,15,15,30,19,30	1,6,16	unrecognized
7.		2	1,6,16,15,32,22,32	1,6,16	ﻒ(V end-form)
	ﺍ (Alif isolated-form)	0.5	1,5,20,20,43,25,40	1,5,21,21	Unrecognized
		2	1,5,21,21,43,24,43	1,5,21,21	ﺍ (Alif isolated-form)

Another example in Table 4, character ﻫ (Ya middle-form) was rotated by 90◦, and has the moment value 1, 5, 21, 20, 23, 42, and character ﻫ was rotated by 180◦ has the moment value 1, 5, 21, 20, 23, 41, thus both characters are recognized as ﻫ, because the feature are 1, 5, 21, 20. The different moment value at the fifth to seventh moment is ignored, because it was not saved. The proposed method resulted in better recognition rate on binary template than thin template. Most characters in binary template become unique at the sixth and seventh moments while most characters in thin template become unique in the fifth, the sixth, and the seventh moments. Table 3 and 4 demonstrates that the TR algorithm can recognize the rotated and scaled characters, although these characters have the difference moment values at the latest moments. The features contain not all moment value combinations, but only the unique ones.

4.3. Comparison with other classification methods

In this subsection, we present comparison result of the proposed method with other benchmarking classification methods. Some of the benchmarking pattern classifiers are support vector machine (SVM) and Euclidean Distance (ED). Table 5 shows the results of recognition using SVM classifier and ED. The SVM classifier used 125 characters in data training phase. By using binary template, SVM failed to recognize the Jawi character. None of testing character was recognized, thus, the recognition accuracy of binary template was 0%. By using thin template, SVM recognized three characters, thus, the recognition accuracy was 10.71%. The SVM had better performance on thin characters than binary characters. On the other hand, the proposed method had better recognition rate on binary characters than thin characters.

ED achieved better result than SVM classifier but less than the proposed method. By using binary template, ED recognized eleven characters, thus, the recognition accuracy was 39.29%. On thin template, ED had less accurate than binary template by recognizing 4 characters or 14.29%. Training phase is not required by the proposed and ED method, instead both methods directly recognize the characters by matching the features, i.e., the moment values of each character.

Table 5. Comparison of Proposed Method with SVM and ED

Classifier	Training character	Testing character	Recognized	Accuracy
Binary template				
SVM	125	28	0	0%
ED	-	28	11	39.29%
Proposed Method	-	28	23	82.14%
Thin template				
SVM	125	28	3	10.71%
ED	-	28	4	14.29%
Proposed Method	-	28	17	60.71%

SVM classifier has lower recognition accuracy. Comparing to the proposed method and ED that did not require more dataset to perform training process, SVM failed in recognizing Jawi character with only one dataset. This lower performance was caused by the lack of data training for SVM classifier. In machine learning method, the used of less training data would decrease the method's performance. On the other hand, the proposed method was not affected by the number of training data because it did not need training stages, and use the features directly in recognition process.

5. CONCLUSION

In this article, we have presented a novel feature extraction and recognition technique for Jawi characters based on moment invariant features, which is called the Tree Root (TR) algorithm. This algorithm assigned each character a unique combination of Hu's moment values. The TR algorithm was applied to two character templates, binary and thin, each consisting of 125 Jawi characters. In the binary template, the TR can recognize 90.4% of the characters, while in the thin template it can recognize 87.2% of the characters. The results showed that the TR algorithm successfully recognized rotated and scaled characters. The proposed method achieved higher recognition rate, compared with SVM classifier and ED using the same testing character.

ACKNOWLEDGEMENTS

This research is funded by Syiah Kuala University.

REFERENCES

[1] Moain AJ. "Language design; History of Javanese character (in Bahasa)," *Dewan Bahasa dan Pustaka*; 1996.
[2] K. Saddami, *et al*., "A Database of Printed Jawi Character Image," *2015 IEEE Conference on Image Information Processing (ICIIP)*, pp. 56-59, 2015.
[3] M.F. Nasrudin, *et al*., "Handwritten cursive Jawi character recognition: A survey," *IEEE Conference on Computer Graphics, Imaging and Visualisation (CGIV)*, pp. 247-256, 2008.
[4] F. Arnia and K. Munadi, "Binarization of Ancient Document Images based on Multipeak Histogram Assumption," *Telecommunication Computing Electronics and Control (TELKOMNIKA)*. vol. 15(3), pp. 1317-1327, 2017.
[5] T.N. Yang and S.D. Wang. "A rotation invariant printed Chinese character recognition system," *Pattern recognition letters*, vol. 22(2), pp. 85-95, 2001.
[6] N.N. Kharma and R.K. Ward, "A novel invariant mapping applied to hand-written Arabic character recognition," *Pattern Recognition*, vol. 34(11), pp. 2115-2120, 2001.
[7] A.H. Hassin, "Printed Arabic character recognition using HMM," *Journal of Computer Science and Technology*. Vol. 19(4), pp. 538-43, 2004.
[8] R.S. Kunte and R.D. Samuel, "A simple and efficient optical character recognition system for basic symbols in printed Kannada text," *Sadhana*, vol. 32(5), 2007.
[9] Iqbal A, ABM M, Tahsin A, Sattar MA, Islam MM, Murase K. "A Novel Algorithm for Translation, Rotation and Scale Invariant Character Recognition," Japan Society for Fuzzy Theory and Intelligent Informatics. InSCIS & ISIS SCIS & ISIS 2008 (pp. 1367-1372), 2008.
[10] F. Arnia, *et al*., "Improvement of binarization performance by applying dct as pre-processing procedure "*6th International Symposium Communications, Control and Signal Processing (ISCCSP)*, 2014 on 2014 May 21 (pp. 128-132), 2014.
[11] S. Muchallil, *et al.*, "Performance Comparison Of Denoising Methods For Historical Documents," *Jurnal Teknologi*, vol.77, no. 22, pp. 143-137, 2015.
[12] Fardian, *et al.*," Identification of Most Suitable Binarisation Method For Acehnese Ancient Manuscripts Restoration Software User Guide," *Jurnal Teknologi*, vol. 77, no. 22, pp. 95-102, 2015.
[13] M.F. Nasrudin, *et al.*, "Invariant features from the trace transform for jawi character recognition*," International Work-Conference on Artificial Neural Networks*, pp. 256-263, 2009.
[14] N.S. Rani, *et al.*, "A Zone Based Approach for Classification and Recognition of Telugu Handwritten Characters," *International Journal of Electrical and Computer Engineering*, Vol. 6(4), pp. 1647, 2016.
[15] M. Kef, et al., "A novel fuzzy approach for handwritten Arabic character recognition," *Pattern Analysis and Applications*, vol. 19(4), pp. 1041-1056, 2016.
[16] M.F. Nasrudin and M. Petrou, "Offline handwritten Jawi recognition using the trace transform," *International Conference on Pattern Analysis and Intelligent Robotics (ICPAIR)*, pp. 87-91, 2011.
[17] Z. Razak, *et al.*, "Off-line jawi handwriting recognition using hamming classification," *Information Technology Journal*, vol. 8, pp. 971-81, 2009.
[18] S. Haykin, "Neural networks: a comprehensive foundation," 2nd ed. New Jersey: *Prentice Hall PTR*; 1999.
[19] S. Kutin and P. Niyogi, "The interaction of stability and weakness in AdaBoost," *Technical Report*, TR-2001–30, 2001.
[20] M.F. Nasrudin, *et al.*, "Object signature features selection for handwritten Jawi recognition," *Distributed Computing and Artificial Intelligence*, pp. 689-98, 2010.
[21] A. Heryanto, et al., "Offline Jawi handwritten recognizer using hybrid artificial neural networks and dynamic programming," *International Symposium in Information Technology*, vol. 2, pp. 1-6, 2008.
[22] R. Redika, *et al.*, "Handwritten Jawi words recognition using hidden Markov models," *International Symposium in Information Technology*, vol. 2, pp. 1-5, 2008.
[23] M.F. Nasrudin, *et al.*, Jawi character recognition using the trace transform," *7th International Conference in Computer Graphics, Imaging and Visualization (CGIV)*, pp. 151-156, 2010.
[24] M. Cheriet, *et al.*, "Character recognition systems: a guide for students and practitioners," New Jersey: *John Wiley & Sons*, 2007.
[25] V. Vapnik, "The nature of statistical learning theory," 2nd ed. New York: *Springer science & business media*; 2013.
[26] M. Ghiassi, *et al.*, "Automated text classification using a dynamic artificial neural network model," *Expert Systems with Applications*, vol. 39(12), pp. 10967-10976, 2012.
[27] S. Arora, *et al.*, "Performance comparison of SVM and ANN for handwritten devnagari character recognition," *arXiv preprint arXiv*:1006.5902, 2010.
[28] Z. Ju and C. Su, "One-pass Moment Algorithm for Graphical Primitives," *International Journal of Electrical Engineering and Computer Science*, vol. 12(5), pp. 3818-3824, 2014.
[29] R. Muralidharan and C. Chandrasekar, "Object recognition using SVM-KNN based on geometric moment invariant," *International Journal of Computer Trends and Technology*, vol. 1(1), pp. 215-220, 2011.
[30] M.K. Hu, "Visual pattern recognition by moment invariants," *IRE transactions on information theory*, vol. 8(2), pp. 179-87, 1962.

BIOGRAPHIES OF AUTHORS

Fitri Arnia received B. Eng degree from Universitas Sumatera Utara (USU), Medan in 1997. She finished her master and doctoral degree from Universsity of New South Wales (UNSW), Sydney, Australia and Tokyo Metropolitan University, Japan in 2004 and 2008 respectively. She has been with the Department of Electrical Engineering, Faculty of Engineering, Syiah Kuala University since 1999. Dr. Arnia was a visiting scholar in Tokyo Metropolitan University (TMU), Tokyo, Japan in 2013 and Suleyman Demirel University (SDU), Isparta, Turkey in 2017. She is a member of IEEE and APSIPA. Her research interests are signal, image and multimedia information processing.

Khairun Saddami received the B.Eng. degree in electrical engineering from Syiah Kuala University, Banda Aceh, Indonesia, in 2015. He is currently a PhD candidate in Postgraduate program in Engineering at Syiah Kuala University. He is also research assistant in Multimedia and Signal Processing research group (Musig), Electrical and Computer Engineering Department, Syiah Kuala University. His research interests include computer vision and image processing.

Khairul Munadi received the B.E. degree from Sepuluh Nopember Institute of Technology, Surabaya, Indonesia, in 1996, and the M.Eng. and Ph.D. degrees from Tokyo Metropolitan University (TMU), Japan, in 2004 and 2007 respectively, all in electrical engineering. From 1996 to 1999, he was with Alcatel Indonesia as a system engineer. Since 1999, he joined Syiah Kuala University, Banda Aceh, Indonesia, as a lecturer at the Electrical Engineering Department. He was a visiting researcher at the Information and Communication Systems Engineering, Faculty of System Design, TMU, Japan, from March 2007 to March 2008. His research interests include multimedia signal processing and communications.