A Study of Comparison of Feature Extraction Methods for Handwriting Recognition

Fergyanto E. Gunawan[1], Intan A. Hapsari[2], Benfano Soewito[3], Sevenpri Candra[4]
[1-3]Binus Graduate Programs, Bina Nusantara University, Jakarta, Indonesia 11480
[4]School of Business & Management, Bina Nusantara University, Jakarta, Indonesia 11480
Emails: [1]fgunawan@binus.edu, [3]bsoewito@binus.edu, [4]scandra@binus.edu

***Abstract*—Automatic handwriting recognition system is very important for various areas of application such as banking and logistics sectors. The performance of such system strongly depends on its feature extraction methods. So far, many extraction methods have independently been studied and proposed, and the three widely adopted methods are Geometric Moment Invariant (GMI), United Moment Invariant (UMI), and Zernike Moment Invariant (ZMI). This study is performed to understand the relative performance of the three methods. For this purpose, the methods, in conjunction with Support Vector Machine classifier with RBF, and PuK kernels, are used to recognize characters taken from Char75K dataset. In addition, the combined features of GMI-UMI, GMI-ZMI, UMI-ZMI, and GMI-UMI-ZMI are also studied. The numerical results suggest the following. Among the three extraction methods, GMI, UMI, and ZMI methods, the latter two methods tend to provide better results by about 7–8% than the GMI method. Generally, when the features are combined, the results improve rather significant, about 7–8% improvement. Only the pair of GMI and UMI combination provide small or negligible improvement. Using the RBF kernel, GMI features alone result in 63% accuracy, UMI features alone 70% accuracy, and GMI-UMI combination result in 72% accuracy. Combination of UMI-ZMI features improve the accuracy significantly than each method along. We also find that the combination of the three methods, GMI-UMI-ZMI, tend to increase the accuracy significantly. The accuracy reaches the level of 96% for the RBF kernel and 89% for the PuK kernel.**

Keywords*—Geometric Moment Invariant (GMI), United Moment Invariant (UMI), Zernike Moment Invariant (ZMI), Support Vector Machine (SVM)

I. Introduction

Automatic handwriting recognition system is important for having many practical applications. In the banking industry for example, the system is required to read customer checks and receipt slips, which are traditionally read by bank tellers manually. The traditional process is slow, tedious, and prone to errors due to variation, size, and manner of handwriting. Similarity among letters in the alphabet and similarity between letters and numbers contribute to the difficulty of the automatic handwriting recognition. For example, we may easily find writing where the number ‘4’ looks very similar to the character ‘A’.

Similar to the cases of many classification problems, the feature extraction method is clearly an important aspect for the success of the classification besides the classification method. The three widely used feature extraction methods for handwriting recognition are Geometric Moment Invariant (GMI), United Moment Invariant (UMI) and Zernike Moment Invariant (ZMI) methods.

Many previous studies have been conducted using those extraction methods [1]–[14]. Reference [1] demonstrated that GMI features and Neural Network classifier are able to recognize Arabic letters at 97% accuracy. Reference [2] utilized UMI features, fast SOM classifier, and achieved 86% accuracy. Reference [3] utilized ZMI, hidden Markov models, and achieved 87–90% accuracy.

This study focuses on comparisons of the the three feature extraction methods. Support Vector Machine (VSM) classification would be used in conjunction with the three extraction methods. The data are taken from the Char74K dataset [15].

II. Research Methods

To study the accuracy of various feature extraction methods, we utilize images of characters in Char74K dataset [15]. The dataset has 64 classes (0–9, A–Z, a–z), 7705 characters of natural handwriting, 3410 characters of handwriting using table PC, and 62992 characters of computer generated. In total, the database has more than 74K character pictures.

Prior the feature extraction processes, all images are subjected to a number of preprocessing stages. Subsequently, those processes are resizing, color model transformation, binarization, and skeletonization.

A. Preprocessing Processes

Each image initially has the size of 1200 × 900 pixels. This dimension is excessive for our computational resources; thus, the image size is reduced prior feature extractions. The image size is reduced to 32 × 24 pixels using Fast Image Resizer [16]. Some images of the lowercase ‘a’ after the process are shown in Fig. 1.

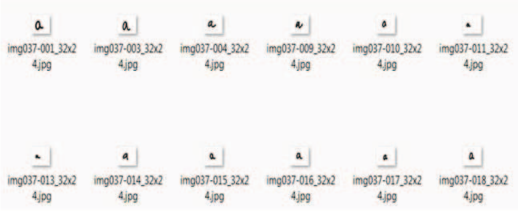

Fig. 1. Examples of images of ‘a’ character after size reduction process where the initial size of 1200 × 900 pixels is reduced to the size of 32 × 24 pixels. The original images are taken from Char74K dataset [15].

978-1-5090-1709-6/16/$31.00 ©2016 IEEE

The second preprocessing process is converting the images from the RGB color model to the grayscale model using the equation:

$$I_S = \frac{I_R + I_G + I_B}{3}, \tag{1}$$

where I_S is the gray-scale intensity, and I_R, I_G, and I_B are respectively the intensities of the red, green, and blue channels.

The third is the binarization process where the image in the gray-scale model is converted into the binary image. The conversion is performed following:

$$I_B(x, y) = \begin{cases} 1, & \text{if} \quad I_S(x, y) > T \\ 0, & \text{otherwise} \end{cases}, \tag{2}$$

where $I_B(x, y) \in [0, 1]$ is the binary image intensity and T is a threshold.

The last preprocessing process is skeletonization that focuses on geometrical and topological properties of the shape. In the current work, the skeletonization utilizes Zhang-Suen algorithm [17]. A typical result of the skeletonization process is shown Fig. 2.

Fig. 2. An example result of the skeletonization process.

B. Feature Extraction Methods

1) Geometric Moment Invariant: The Geometric Moment Invariant (GMI) features were introduced by Ref. [4]. GMI characteristics have been discussed in a great length in Ref. [5]. The following procedures of computing the seven GMI features are on the basis of Ref. [6]. Firstly, we scan the binary image from the left to the right, and from the top to the bottom. We define the moments m_{pq} as: $m_{pq} = \sum_x \sum_y x^p y^p f(x, y)$, where p and q are integers with values of $0, 1, 2, \ldots$, and $f(x, y)$ is the intensity value at the location x and y. The coordinates of the object centroid are $\bar{x} = m_{10}/m_{00}$ and $\bar{y} = m_{01}/m_{00}$. Moreover, we define the central moments μ_{pq} as: $\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y)$. We define the normalized central moments η_{pq} as: $\eta_{pq} = \mu_{pq}/\mu_{00}^{\gamma}$, where $\gamma = (p + q + 2)/2$ and $p, q \in 2, 3, \ldots$. Finally, we define the seven GMI features, $\varphi_1, \varphi_2, \ldots, \varphi_7$, as the following.

$$\varphi_1 = \eta_{20} + \eta_{02} \tag{3}$$

$$\varphi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{4}$$

$$\varphi_3 = (\eta_{30} - 3\eta_{12})^2 + (\eta_{21} - \eta_{03})^2 \tag{5}$$

$$\varphi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{6}$$

$$\begin{aligned} \varphi_5 = {} & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ & + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \end{aligned} \tag{7}$$

$$\begin{aligned} \varphi_6 = {} & (\eta_{20} - \eta_{02}) \left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] + \\ & + 4\eta_{11} (\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \end{aligned} \tag{8}$$

$$\begin{aligned} \varphi_7 = {} & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) \left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ & - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03}) \left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \end{aligned} \tag{9}$$

2) United Moment Invariant: The United Moment Invariant (UMI) was proposed by Ref. [7] and has been discussed in a great length in Ref. [5]. The following UMI feature formulations, $\theta_1, \theta_2, \ldots, \theta_8$, are on the basis of Ref. [8].

$$\theta_1 = \frac{\sqrt{\varphi_2}}{\varphi_1}, \tag{10}$$

$$\theta_2 = \frac{\varphi_6}{\varphi_1 \varphi_4}, \tag{11}$$

$$\theta_3 = \frac{\sqrt{\varphi_5}}{\varphi_4}, \tag{12}$$

$$\theta_4 = \frac{\varphi_5}{\varphi_3 \varphi_4}, \tag{13}$$

$$\theta_5 = \frac{\varphi_1 \varphi_6}{\varphi_2 \varphi_3}, \tag{14}$$

$$\theta_6 = \frac{\left(\varphi_1 + \sqrt{\varphi_2}\right) \varphi_3}{\varphi_6}, \tag{15}$$

$$\theta_7 = \frac{\varphi_1 \varphi_5}{\varphi_3 \varphi_6}, \quad \text{and} \tag{16}$$

$$\theta_8 = \frac{(\varphi_3 + \varphi_4)}{\sqrt{\varphi_5}}. \tag{17}$$

3) Zernike Moment Invariant: Zernike Moment Invariant (ZMI) was introduced by Teague in 1980 [9] and it has been widely used for pattern recognition [10], [11] and especially for character recognition [12]. Reference [13] concluded that ZMI is better than Hu's moments for object recognition. The following ZMI feature formulations, $\mathrm{ZM}_1, \mathrm{ZM}_2, \ldots, \mathrm{ZM}_6$, are on the basis of Ref. [14].

$$\mathrm{ZM}_1 = \frac{3}{\pi} \left[2(\eta_{20} + \eta_{02} - 1)\right] \tag{18}$$

$$\mathrm{ZM}_2 = \frac{9}{\pi^2} \left[(\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2\right] \tag{19}$$

$$\mathrm{ZM}_3 = \frac{16}{\pi^2} \left[(\eta_{03} - 3\eta_{21})^2 + (\eta_{30} - 3\eta_{12}^2)\right] \tag{20}$$

$$\mathrm{ZM}_4 = \frac{144}{\pi^2} \left[(\eta_{03} - 3\eta_{21})^2 + (\eta_{30} + \eta_{12})^2\right] \tag{21}$$

$$\begin{aligned} \mathrm{ZM}_5 = {} & \frac{13824}{\pi^4} \left[(\eta_{03} - 3\eta_{21})(\eta_{03} + \eta_{21})\right. \\ & \left\{(\eta_{03} + \eta_{21})^2 - 3(\eta_{30} + \eta_{12})^2\right\} \\ & \left. -(\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{21}) \left\{(\eta_{30} + \eta_{12})^2 - 3(\eta_{03} + \eta_{21})^2\right\}\right] \end{aligned} \tag{22}$$

$$\begin{aligned} \mathrm{ZM}_6 = {} & \frac{864}{\pi^3} \left[(\eta_{02} - \eta_{20}) \left\{(\eta_{30} + \eta_{12})^2 - (\eta_{03} + \eta_{21})^2\right\}\right. \\ & \left. +4\eta_{11}(\eta_{03} + \eta_{21})(\eta_{30} + \eta_{12})\right] \end{aligned} \tag{23}$$

C. Support Vector Machine

In the present study, we only use the Support Vector Machine (SVM) for linearly separable data. The SVM is a numerical method to compute an hyperplane for separating a two-class dataset. It can easily be extended to multiple-class problem. The SVM establishes the hyperplane, governed by $(\mathbf{w}, b)$, by using the support vectors, which are the data points that are closest to the hyperplane. The following SVM formulation is derived from Refs. [18], [19]; readers are advised to the two sources for detail exposition.

We consider the point sets $\mathbf{x}_i \in \Re^d$, as the support vectors, with the categories $y_i \in [-1, +1]$. The hyperplane that separates $y_i = -1$ from those of $y_i = +1$ should satisfy

$$<\mathbf{w}, \mathbf{x}> +b = 0, \qquad (24)$$

where $\mathbf{w} \in \Re^d$, $<\mathbf{w}, \mathbf{x}>$ denotes the inner dot product of $\mathbf{w}$ and $\mathbf{x}$, and b is a scalar constant. The hyperplane is obtained by solving:

$$\min_{\mathbf{w},b} L_p = \frac{1}{2} <\mathbf{w}, \mathbf{w}> - \sum_i \alpha_i \left[y_i \left(<\mathbf{w}, \mathbf{x}_i> +b \right) - 1 \right], \qquad (25)$$

where $\alpha_i \geq 0$. For the case where the data are linearly not separable, the feature vector $\mathbf{x}_i$ would be transformed with a kernel function. Three types of the kernel functions would be evaluated. The first kernel is polynomial type where $K(\mathbf{x}, \mathbf{y}) = (1+ <\mathbf{x}, \mathbf{y}>)^d$. The second type is Radial Basis Function (RBF) where $K(\mathbf{x}, \mathbf{y}) = \exp(- <(\mathbf{x}-\mathbf{y}), (\mathbf{x}-\mathbf{y})> /(2\sigma^2))$. The third type is Pearson VII universal kernek (PuK) where $f(\mathbf{x}) = H/[1+(2(\mathbf{x}-\mathbf{x}_0)\sqrt{2^{1/\omega}-1}/\sigma)^2]^\omega$. The parameter d is an integer, and would be evaluated for $d = 1, 2, \text{and}, 3$, and σ and ω are parameters with positive values. The parameter H is the peak height at $\mathbf{x}_0$.

D. Performance Evaluation

The performance indicators are the confusion matrix and the classification accuracy. The definition of the confusion matrix is given in Table I. The accuracy definition is given in Eq. (26).

TABLE I. CONFUSION MATRIX TABLE

		Prediction	
		Negative	Positive
Actual	Negative	TN	FN
	Positive	FP	TP

$$\text{Accuracy} = \frac{\text{TN} + \text{TP}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \qquad (26)$$

In the equation, TN denotes true negative, FN denotes false negative, FP denotes false positive, and TP denotes true positive.

III. RESULTS

A. The Examples of the Extracted Features

The number of features obtained from GMI method is 7 features, UMI 8 features, and ZMI 6 features. In the Appendix section, some examples of those features for a few characters and numbers are presented in Table II for GMI method, Table III for UMI method, and in Table IV for ZMI method.

B. Comparison of Geometric Moment Invariant, United Moment Invariant, and Zernike Moment Invariant Features

The results are summarized in Fig. 3 in forms of bar diagram and boxplot. A few notable results of the current work are of the following.

Among the three extraction methods, GMI, UMI, and ZMI methods, the latter two methods tend to provide better results by about 7–8% than the GMI method.

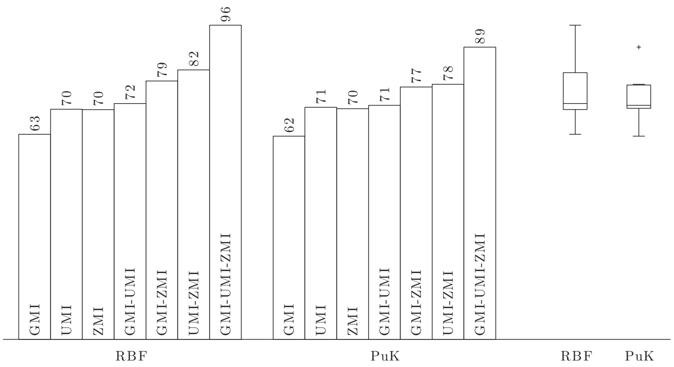

Fig. 3. The level of accuracy of the handwriting recognition using GMI, UMI, ZMI, GMI-UMI, GMI-ZMI, GMI-UMI-ZMI feature extraction methods and support vector machine classifier with kernel functions: RBF and PuK. On the right side, the accuracy distributions are graphically shown in form of boxplots.

Generally, when the features are combined, the results improve rather significant, about 7–8% improvement. Only the pair of GMI and UMI combination provide small or negligible improvement. Using the RBF kernel, GMI features alone result in 63% accuracy, UMI features alone 70% accuracy, and GMI-UMI combination result in 72% accuracy. Combination of UMI-ZMI features improve the accuracy significantly than each method along.

We also find that the combination of the three methods, GMI-UMI-ZMI, tend to increase the accuracy significantly. The accuracy reaches the level of 96% for the RBF kernel and 89% for the PuK kernel.

IV. CONCLUSIONS

One major issue in all classification problems is determining relevant features, which can be used to accurately and precisely differentiate objects in the area of interest. Many feature extraction methods have been proposed for the case of handwriting recognition problem; however, differences in the classification methods and datasets make difficult to obtain insights regarding the relative strengths of the existing methods. Three widely used feature extraction methods for handwriting recognition are Geometric Moment Invariant (GMI), United Moment Invariant (UMI), and Zernike Moment Invariant (ZMI) methods. The purpose of the study is to compare the performance of the three methods using the same classifier and dataset. The effects of the combined features of the three methods are also studied. The dataset is Char74K and the classifier is Support Vector Machine (SVM) method. The results suggest that the combined GMI-UMI-ZMI features provides the recognition up to 96% of the level of accuracy using the SVM-RBF kernel and 89% using PuK kernel. On their own, UMI and ZMI features tend to be slightly better than GMI features. Generally, the combined features, GMI-UMI, GMI-ZMI, UMI-ZMI, result in the higher level of accuracy in comparison GMI, UMI, or ZMI features alone.

REFERENCES

[1] M. Rashad, K. Amin, M. Hadhoud, and W. Elkilani, "Arabic character recognition using statistical and geometric moment features," in *Electronics, Communications and Computers (JEC-ECC), 2012 Japan-Egypt Conference on*. IEEE, 2012, pp. 68–72.

[2] Y. Wang, A. Peyls, Y. Pan, L. Claesen, and X. Yan, “A fast self-organizing map algorithm for handwritten digit recognition,” in *Multimedia and Ubiquitous Engineering*. Springer, 2013, pp. 177–183.

[3] I. El-Feghi, F. Elmahjoub, B. Alswady, and A. Baiou, “Offline handwritten arabic words recognition using zernike moments and hidden markov models,” in *Computer Applications and Industrial Electronics (ICCAIE), 2010 International Conference on*. IEEE, 2010, pp. 165–168.

[4] M.-K. Hu, “Visual pattern recognition by moment invariants,” *information Theory, IRE Transactions on*, vol. 8, no. 2, pp. 179–187, 1962.

[5] J. Flusser, T. Suk, and B. Zitova, *Moments and moment invariants in pattern recognition*. John Wiley and Sons, 2009.

[6] Z. Huang and J. Leng, “Analysis of hu’s moment invariants on image scaling and rotation,” in *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on*, vol. 7. IEEE, 2010, pp. V7–476.

[7] S. Yinan, L. Weijun, and W. Yuechao, “United moment invariants for shape discrimination,” in *Robotics, Intelligent Systems and Signal Processing, 2003. Proceedings. 2003 IEEE International Conference on*, vol. 1. IEEE, 2003, pp. 88–93.

[8] M. W. Nasrudin, S. N. Yaakob, R. R. Othman, I. Ismail, M. I. Jais, and A. S. A. Nasir, “Analysis of geometric, zernike and united moment invariants techniques based on intra-class evaluation,” in *Intelligent Systems, Modelling and Simulation (ISMS), 2014 5th International Conference on*. IEEE, 2014, pp. 7–11.

[9] M. R. Teague, “Image analysis via the general theory of moments*,” *JOSA*, vol. 70, no. 8, pp. 920–930, 1980.

[10] A. Khotanzad and Y. H. Hong, “Invariant image recognition by zernike moments,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 12, no. 5, pp. 489–497, 1990.

[11] Y. Bin and P. Jia-Xiong, “Improvement and invariance analysis of zernike moments using as a region-based shape descriptor,” in *null*. IEEE, 2002, p. 120.

[12] I. El-Feghi, F. Elmahjoub, B. Alswady, and A. Baiou, “Offline handwritten arabic words recognition using zernike moments and hidden markov models,” in *Computer Applications and Industrial Electronics (ICCAIE), 2010 International Conference on*. IEEE, 2010, pp. 165–168.

[13] R. K. Sabhara, C.-P. Lee, and K.-M. Lim, “Comparative study of hu moments and zernike moments in object recognition,” *SmartCR*, vol. 3, no. 3, pp. 166–173, 2013.

[14] N. A. Bakar and S. M. Shamsuddin, “United zernike invariants for character images,” in *Visual Informatics: Bridging Research and Practice*. Springer, 2009, pp. 498–509.

[15] T. D. Campos. (2012) The chars 74k dataset. [Online]. Available: http://www.ee.surrey.ac.uk/CVSSP/demos/chars74k/

[16] AdionSoft. (2015) Fast image resizer. Download on February 2015. [Online]. Available: http://adionsoft.net/fastimageresize/

[17] W. Chen, L. Sui, Z. Xu, and Y. Lang, “Improved zhang-suen thinning algorithm in binary line drawing applications,” in *Systems and Informatics (ICSAI), 2012 International Conference on*. IEEE, 2012, pp. 1947–1950.

[18] N. Christianni and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge, UK: Cambridge University Press, 2000.

[19] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York: Springer, 2008.

Appendix

TABLE II. The examples of characters and theirs feature values of the GMI method.

Char.	Feature Values						
a	6.01×10^{-1}	2.28×10^{-3}	2.70	1.31×10^{-1}	7.70×10^{-2}	5.12×10^{-3}	6.80×10^{-2}
b	6.36×10^{-1}	7.94×10^{-2}	3.66	1.04	1.13	1.98×10^{-1}	2.62×10^{-2}
c	4.92×10^{-1}	2.85×10^{-2}	9.05×10^{-2}	4.52×10^{-2}	1.10×10^{-3}	5.21×10^{-3}	1.61×10^{-3}
A	6.09×10^{-1}	1.55×10^{-2}	4.33	1.49×10^{-1}	1.19×10^{-1}	1.85×10^{-2}	1.81×10^{-2}
A	7.94×10^{-1}	4.44×10^{-1}	1.20	6.66×10^{-1}	5.47×10^{-1}	3.37×10^{-1}	1.13×10^{-1}
B	4.94×10^{-1}	2.91×10^{-2}	1.09×10^{-1}	1.41×10^{-2}	5.12×10^{-4}	2.20×10^{-3}	2.20×10^{-4}
C	6.02×10^{-1}	1.08×10^{-2}	2.12	9.87×10^{-2}	3.40×10^{-2}	1.02×10^{-2}	3.50×10^{-2}
C	6.27×10^{-1}	5.17×10^{-2}	9.11×10^{-2}	2.67×10^{-3}	7.92×10^{-6}	5.64×10^{-4}	2.31×10^{-5}
0	6.38×10^{-1}	5.29×10^{-2}	2.95×10^{-1}	8.92×10^{-2}	1.42×10^{-2}	2.05×10^{-2}	1.24×10^{-2}
1	6.69×10^{-1}	1.71×10^{-2}	4.65	5.66×10^{-1}	8.46×10^{-1}	5.89×10^{-2}	8.62×10^{-1}
1	5.59×10^{-1}	5.13×10^{-2}	4.40×10^{-1}	4.30×10^{-1}	2.90×10^{-2}	1.12×10^{-2}	2.17×10^{-1}
2	7.85×10^{-1}	1.25×10^{-1}	8.72×10^{-1}	3.52×10^{-1}	1.17×10^{-1}	2.49×10^{-2}	2.49×10^{-2}

TABLE III. The examples of characters and theirs feature values of the UMI method.

Char.	Feature Values							
a	5.49×10^{-2}	4.76×10^{-2}	1.28	3.07×10^{-1}	2.94	1.19×10^{2}	6.44	4.97
b	2.81×10^{-1}	8.93×10^{-2}	1.73	2.70×10^{-1}	1.02×10^{-1}	1.60×10^{2}	3.02	7.01
c	2.24×10^{-1}	1.62×10^{-1}	1.16	5.20×10^{-1}	1.24	1.97×10^{1}	3.21	3.10
A	3.97×10^{-1}	1.68×10^{-2}	9.77×10^{-1}	9.89×10^{-1}	1.10×10^{-1}	8.05×10^{1}	5.90×10^{1}	2.01
A	1.00	1.00	1.00	1.00	1.00	2.00	1.00	2.00
B	1.00	1.00	1.00	1.00	1.00	2.00	1.00	2.00
C	5.02×10^{-1}	4.98×10^{-1}	1.37	4.87×10^{-1}	5.15×10^{-1}	1.16×10^{1}	9.78×10^{-1}	3.54
C	1.82×10^{-1}	1.30×10^{-1}	1.79	1.34×10^{-1}	1.64×10^{-1}	2.17×10^{2}	1.02	1.39×10^{1}
0	4.97×10^{-1}	3.97×10^{-1}	2.46	3.98×10^{-2}	1.05×10^{-2}	5.75×10^{2}	1.00×10^{-1}	6.23×10^{1}
1	2.98×10^{-1}	5.92×10^{-2}	2.03	7.61×10^{-2}	1.23×10^{-2}	1.19×10^{3}	1.28	2.72×10^{1}
1	6.13×10^{-1}	1.66×10^{-1}	3.96×10^{-1}	2.19×10^{-2}	6.15×10^{-2}	6.98×10^{1}	1.32×10^{-1}	2.06×10^{1}
2	5.47×10^{-1}	2.07×10^{-1}	1.19	1.82×10^{-1}	8.91×10^{-2}	5.80×10^{1}	8.78×10^{-1}	7.38

TABLE IV. The examples of characters and theirs feature values of the ZMI method.

Character	Feature Values					
a	1.39	3.76×10^{-3}	8.12×10^{-3}	6.50×10^{-2}	3.68×10^{-6}	4.70×10^{-5}
b	1.32	1.14×10^{-2}	2.67×10^{-2}	2.02×10^{-1}	1.72×10^{-3}	7.77×10^{-3}
c	9.67×10^{-1}	1.79×10^{-2}	6.25×10^{-2}	4.23×10^{-1}	1.51×10^{-2}	1.50×10^{-3}
A	1.22	1.37×10^{-2}	4.02×10^{-4}	1.72×10^{-3}	5.47×10^{-7}	1.39×10^{-4}
A	1.10	1.46×10^{-1}	0.00	0.00	0.00	0.00
B	9.22×10^{-1}	2.25×10^{-1}	0.00	0.00	0.00	0.00
C	1.20	2.34×10^{-2}	1.18×10^{-2}	1.06×10^{-1}	1.05×10^{-4}	1.61×10^{-3}
C	1.40	2.58×10^{-3}	1.53×10^{-2}	1.34×10^{-1}	2.16×10^{-4}	8.84×10^{-4}
0	1.11	1.92×10^{-2}	6.64×10^{-2}	5.92×10^{-1}	2.33×10^{-5}	2.97×10^{-4}
1	1.40	6.13×10^{-3}	2.16×10^{-3}	4.98×10^{-3}	5.37×10^{-6}	3.44×10^{-4}
1	9.54×10^{-1}	6.27×10^{-2}	3.45×10^{-2}	2.94×10^{-2}	1.27×10^{-2}	4.74×10^{-2}
2	4.37×10^{-1}	1.57×10^{-1}	1.82×10^{-1}	1.26	1.54×10^{-1}	3.23×10^{-2}