



The Impact of Morphological Processing and Feature Selection on Handwriting Recognition Accuracy

Joonas Löömps, Amnir Hadachi

► To cite this version:

Joonas Löömps, Amnir Hadachi. The Impact of Morphological Processing and Feature Selection on Handwriting Recognition Accuracy. The 4th international Conference on Multimedia, Scientific Information and Visualization for Information Systems and Metrics, 2017, Sevilla, Spain. hal-01508770

HAL Id: hal-01508770

<https://hal.science/hal-01508770>

Submitted on 10 Jul 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The Impact of Morphological Processing and Feature Selection on Handwriting Recognition Accuracy

Joonas Lõmps¹, and Amnir Hadachi¹ *

Institute of Computer Science, University of Tartu, Tartu, 51014 Estonia

Abstract. Handwriting or character recognition is one of the most challenging topics and the oldest application of pattern recognition. Till now, all the research progress and innovation has been made either regarding enhancing the preprocessing steps or via advancing the language modeling process. In this paper, we are presenting a combined method for Hiragana handwriting recognition through the exploration of morphological processing and features extraction. Our approach is tested in offline mode and uses two different classifiers for evaluation, the Support Vector Machine (SVM) and the K-Nearest Neighbour (KNN). The outcome of this investigation gave knowledge about the best combination and structure of the adequate method for performing a recognition accuracy of 90.3 percent on offline test data not used in the training.

Keywords: Morphological Processing, Support Vector Machine, K-Nearest Neighbor, Hiragana Handwriting Recognition, Features Selection.

1 Introduction

Handwriting recognition or optical character recognition is the art of identifying written characters with the help of a machine. In general, there are two ways of performing recognition: either in online mode or offline mode. The online mode refers to real-time character recognition of a time series of coordinates that represent the movements captured from a pen tip. However, the offline recognition concerns the identification based on already taken images of the text. Moreover, the online recognition yields good results due to the amount of data and relevant features that can be extracted, which is not the case in offline recognition [1]. The importance of handwriting recognition systems can be illustrated by the increasing need for them in many emerging fields of application, such as electronic libraries, personalized medicine, multimedia and all types of systems that require or use hand written inputs. Besides, automating the digitalization of handwritten characters offers lot of advantages when it comes to processing large amounts of papers, documents, and transforming or transferring data into computers.

* J. Lõmps and A. Hadachi are with the Institute of Computer Science, University of Tartu, Tartu, 51014 Estonia, e-mail: b12027@ut.ee / hadachi@ut.ee

Handwriting recognition approaches usually include the following steps: pre-processing, representation, segmentation or stroke, feature extraction and recognition. However, this is not a rule, some techniques do not include all these mentioned steps. In addition, all of them have the objective of cleaning, filtering and transforming the input before applying the recognition via the use of a classifier, such as K-Nearest Neighbour [3], Support Vector Machine [2], Hidden Markov Model [4], Euclidean Distance [5], etc.

Our research is conducted on Hiragana handwriting in offline mode. Most people believe that the Japanese language, especially the Hiragana, is based a complex writing system due to its morphology and photonic script [8]. Hence, the work presented in this paper focuses on investigating the impact of the morphological processing and feature selection on the accuracy recognition of our proposed method for recognizing the Hiragana characters.

2 Related Work

When it comes to handwriting recognition, a great deal of research has already been carried out. Since awhile now, most of the focus in research is on Japanese character recognition has been on applying statistical and probabilistic approach for classification like Neural Networks [9], genetic algorithms [10] or shape characteristics [11] have been used.

For example, in [12], the authors proposed a technique tested on Chinese characters and based on feature extractions and the Bayes classifier. The results yielded a recognition accuracy of 92,8%. Besides, there is also the case of using Hidden Markov Chain, as presented in [13], which gave interesting results with 92% accuracy for Kanji. Regarding Hiragana characters, the best results were 95.1% accuracy using a simple three-layer neural network [15] and 94% accuracy obtained using SVM [14].

Another application of Neural Network is illustrated in [20], where the authors proposed an algorithm which utilises Size-Translation-Rotation-Invariant vectors. During the process of constructing the vectors a neural network approach was used and it was created by design, not by training. The testing of the algorithm showed an accuracy rate of 96.2%.

As we know, one of the challenges with handwriting is the absorption variation of the same character among different handwriting styles. Besides, there are graphical similarities of different characters in Japanese script. To overcome these issues, the authors in [18] proposed a technique based on template matching and enhanced this by using a couple of strategies focusing on cutting off a number of recognition classes and introducing a fine detail classification for the recognition.

Another approach was presented in [19], based on virtual example synthesis. The latter was done by synthesizing virtual examples from a number of real samples. The results obtained demonstrated that by choosing the appropriate eigenvectors and base samples they had good performances compared to pattern segmentation techniques and SVM.

In general, all mentioned approaches were not tested on datasets with high variability regarding the handwriting style. In addition, the situation in the literature regarding the recently attempted techniques is that researchers are focusing either on soft computing approaches for classification or shape/morphology characteristics for classification.

3 Methodology

3.1 Adopted Method

The method adopted in this article can be resumed as follows: preprocessing, morphology processing, feature extraction and classification (Fig. 1). During the preprocessing step, the image is processed using gray scale and binarization techniques. Next phase is morphological processing, where we investigate the effect of erosion and dilate methods. Then, it is the feature extraction. The features chosen for this work are the HOG feature, Horizontal and vertical projections, zoning and raw data. Finally, the classifiers tested in this architecture are SVM and KNN.

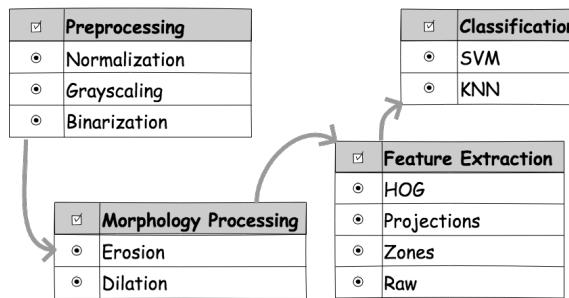


Fig. 1. The Adopted Method

3.2 Preprocessing

Before the character image is ready for morphological processing, it has to go through preprocessing for more consistent and accurate results. Our approach to preprocessing consists of three parts. For the data to be comparable, it has to have same dimensions: we chose them to be 50x50 pixels. Normalization takes an input image of a character, crops the image as much as possible to only leave the character, and resizes it to 50x50. Normalized image of a character is then turned into a grayscale image, meaning it only carries its intensity information and is represented exclusively by different shades of gray, varying from black at the weakest to white at the strongest level of intensity. This is followed by binarization of the grayscale image using the Otsu's method, as our test images were

all on white paper [6]. That leaves us with a digital image of a character where every background pixel is either a 0 for black or 255 for white for foreground pixels.

3.3 Morphology process

Mathematical morphology provides a set of different operators for processing digital images based on their shape. The operators are especially useful for the analysis of binary images. When correctly used, the mathematical morphology operators can be used for edge detection, noise removal, image enhancement, image segmentation etc, all while preserving their shape and eliminating irrelevancies. Mathematical morphology uses sets to represent shapes and the operators use methods defined in the set theory to transform the input image [7]. Our work focuses on the two most basic operations in mathematical morphology: erosion and dilation. Both operations take two inputs. One is the input image, that is to be modified, and the other is the structuring element. The structuring element determines the details regarding how the operator modifies the input image.

Erosion

Erosion is a morphological transformation that combines two sets using the vector subtraction of set elements (Fig. 2). If A and B are sets in the Euclidean space E and A is a binary image, then the erosion of the binary image A by the structuring element B is defined as:

$$A \ominus B = \{x \in E \mid x + b \in A, \forall b \in B\} \quad (1)$$

Example of an erosion operation:

$$\begin{aligned} A &= \{(0, 0), (0, 1), (1, 0), (1, 1), (1, 2), (1, 3), (2, 1), (2, 2), \\ &\quad (2, 3), (2, 4), (3, 2), (3, 3), (4, 5)\} \\ B &= \{(0, -1), (-1, 0), (0, 0), (0, 1), (1, 0)\} \\ A \ominus B &= \{(1, 1), (2, 2), (3, 2)\} \end{aligned}$$

This is the definition used for erosion operation by [7]. For simplicity, the structuring element B can be slid across the binary image A and where B is contained in A , its origin point $(0, 0)$ is present in $A \ominus B$.

Dilation

Dilation is a pseudo-inverse of the erosion. Instead of combining two sets using vector subtraction of set elements, it uses their addition (Fig. 3). If A and B are

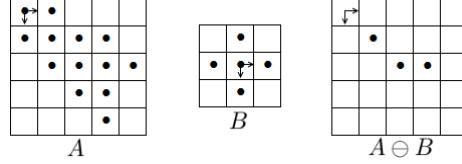


Fig. 2. Illustration of the erosion process

sets in Euclidean space and A is a binary image, then the dilation of the binary image A by the structuring element B is defined as:

$$A \oplus B = \{c \in E \mid c = a + b, \exists a \in A, \exists b \in B\} \quad (2)$$

Example of dilate operation:

$$\begin{aligned} A &= \{(1, 1), (1, 2), (2, 2), (2, 3), (3, 3)\} \\ B &= \{(1, -1), (0, 0), (0, 1)\} \\ A \oplus B &= \{(2, 0), (1, 1), (2, 1), (3, 1), (1, 2), (2, 2), (3, 2), \\ &\quad (4, 2), (1, 3), (2, 3), (3, 4), (2, 4), (3, 4)\} \end{aligned}$$

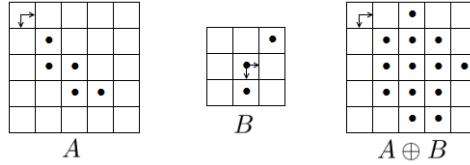


Fig. 3. Illustration of the dilation process

This is the definition used for the dilation operation by [7]. For simplicity, the structuring element B may be slid across the binary image A . Wherever the origin point of B matches a point on A , the structuring element B can be stamped onto image A , creating $A \oplus B$.

3.4 Feature Extraction

Image representation plays a crucial role in a recognition system. Simple binary image can be fed to a recognizer, but it might not achieve the desired results. A more compact and characteristic representation of an image is required for most of the recognition systems to avoid unnecessary complexity and to increase the recognition accuracy rates [16]. Extracting a set of features for each class helps to distinguish it from other classes while remaining invariant to characteristic differences within the class [17]. In the following sections, we describe the extracted features used in our approach.

Raw data

The simplest feature is the output of preprocessing without any feature extraction. With our approach, it either gives us an eroded or dilated binary image of a character. Due to using 50x50 images of character it gives us a vector of 2500 data points per character.

Projection Histogram

Using a Projection Histogram (PH) is a way to represent a two-dimensional (2-D) image signal into 1-D signal. In short, it counts the number of foreground pixels in each column and row in an image and turns the values into a vector with a size of $n + m$ where n and m represent the dimensions of the image [21]. Example of it can be seen on figure 4.

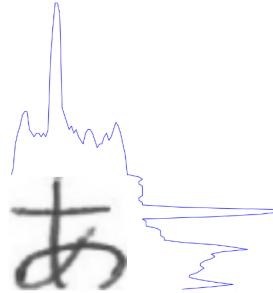


Fig. 4. Example of a Projection Histogram

Zoning

Zoning is a feature that divides the input image into a number of predefined sizes and add up the intensities of foreground pixels in the given zone [22]. We tested two different variations of a number of zones and their sizes: 25 zones of 10x10 and 100 zones of 5x5.

To find the zoning feature vector of an image, all pixel intensities in a given zone need to be summed. The values are then transformed into a vector with a size of l , where l is the number of zones.

If $f(i, j)$ is a digital image and $I(i, j)$ is the pixel intensity of that image, then the pixel intensities $I(i, j)$ of an image in a zone k are calculated by

$$V_k = \sum_{i=1}^n \sum_{j=1}^m I_k(i, j) \quad (3)$$

where $1 \leq k \leq l$, where l is the number of zones and n and m are the dimensions of the zone.

Equation (3) describes the summation of intensity values of k^{th} zone in the image. The final vector is a list of intensities of every zone.

$$V = [V_k], 1 \leq k \leq l \quad (4)$$

In our approach, it simplifies the counting of the foreground pixels due to the use of binary images as input for extracting the features with results that can be seen in figure 5.

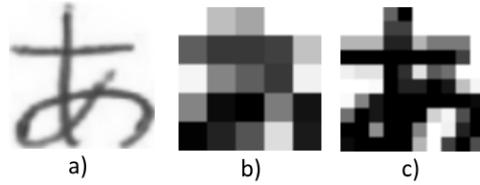


Fig. 5. An example result of zoning. a) base image, b) zoning $k = 25$, c) zoning $k = 100$

Histogram of Oriented Gradients

Objects' appearance and shape can be characterized by the distribution of local intensity gradients or edge directions. It begins by dividing the image into smaller regions called cells, and then for every cell a local 1-D histogram of gradient direction over the pixels of the cell is compiled. Concatenation of these histograms makes up the descriptor. For improved accuracy, the cell values can be normalized by combining cells into a larger region of the image, called a block, and using its intensity for normalization [23].

3.5 Classification

Classification is the problem of identifying the category to which a new observation belongs based on a training set of observations whose category is already known. Classifier is an algorithm that implements classification and usually consists of training and testing. Training a classifier introduces examples of observations that belong into a certain category, while testing uses the algorithm to predict to which category the given observation belongs to.

For our investigation, k -nearest neighbour and support vector machine classifiers are used, as they are two of the most well known and commonly used classifiers.

K-Nearest Neighbour

The k -nearest neighbours rule [24] is one of the simplest and most well known methods for classification. It classifies examples based on the majority of the

k -nearest neighbours found in a training set.

Given a set of n pairs $(x_1, \theta_2), \dots, (x_n, \theta_n)$, where x_i takes values in a metric space X defined by a metric d and θ takes values in the set of possible class indexes $\{1, 2, \dots, M\}$.

For any new pair (x, θ) , only the value of x is known, and it is necessary to estimate its θ by using the information provided in the set of trained classified points. Then $x_n \in \{x_1, x_2, \dots, x_n\}$ is the nearest neighbour to x if

$$\min d(x_i, x) = d(x_n, x) \quad i = 1, 2, \dots, n \quad (5)$$

The nearest neighbour rule decides that x belongs to the category θ_n of its nearest neighbour x_n . In our approach $k = 3$.

Support Vector Machine

Support vector machine is a discriminative classifier defined by a separating hyperplane. Given labelled training data, the algorithm outputs an optimal hyperplane which categorizes new examples. The optimal separating hyperplane maximizes the margin (M) of the training data. Let $|\beta_0 + \beta^T x| = 1$ represent a hyperplane, where x denotes the training example closest to the hyperplane (also called support vector). Then the distance between a point x and a hyperplane (β, β_0) is

$$distance = \frac{|\beta_0 + \beta^T x|}{\|\beta\|} \quad (6)$$

As the numerator is equal to one and the distance to the support vector is

$$distance_{support\ vectors} = \frac{|\beta_0 + \beta^T x|}{\|\beta\|} = \frac{1}{\|\beta\|} \quad (7)$$

The margin is then equal to twice the distance of the closest example $M = \frac{2}{\|\beta\|}$. Maximizing M is equivalent to the problem of minimizing a function $L(\beta)$ subject to some constraints. The constraints model the requirement for the hyperplane to classify all the training example x_i . Formally

$$\min_{\beta, \beta_0} L(\beta) = \frac{1}{2} \|\beta\|^2 \quad subject\ to \quad y_i(\beta^T x_i + \beta_0) \geq 1 \quad \forall i, \quad (8)$$

where y_i represents each of the labels of the training examples. This problem can be solved using Lagrange multipliers to obtain the weight vector β and the bias β_0 of the optimal hyperplane.

4 Experimental Results and Discussion

The handwritten Hiragana characters used for testing were collected by the authors with the help of University's Department of Languages of Asian Region. This database consists of 12251 characters, approximately 173 images of every

71 different Hiragana characters, and includes 28 different handwriting styles from Japanese language students with varying experience with the language. As consequence, our dataset has a significant variability about an average of 49%. Some of the characters have a considerable variability due to the different handwriting styles. For example, figure 6 gives an idea about the variability of each character in our dataset. Based on our experiments, the mathematical mor-

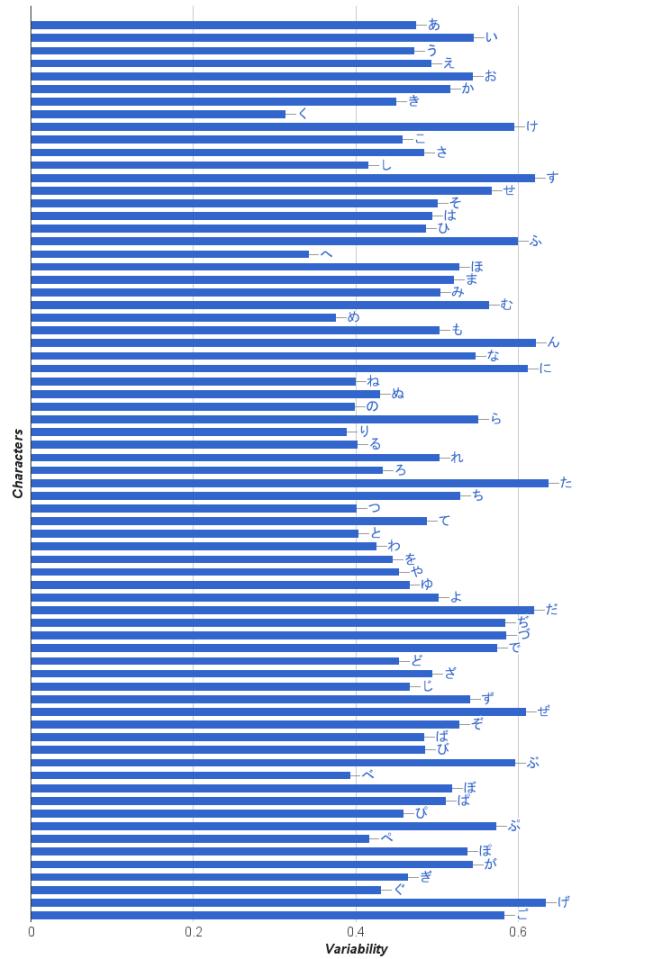


Fig. 6. Handwriting dataset variability

phology operations, erosion and dilation, affect the average recognition accuracy rates no matter which technique is used for feature selection or classification. We noticed that the changes in accuracy depend much more on the classifier than the features. Support vector machines seem to be less affected by the mor-

phological operators than k -nearest neighbours, as it can be seen in tables 1 and 2.

In all of our tested combinations of features and classifiers, while using morphological erosion before extracting the feature, resulted in a drop in the recognition accuracy rate. The most affected combination was KNN along with raw data going from 78.84% to 56.95%, resulting in a 21.89% drop in recognition accuracy. On the other hand, using dilation resulted in an increase of recognition accuracy rate. There were two combinations where dilation reduced the accuracy; SVM with PH and SVM with HOG. The combination of classifiers and the feature that benefited the most from using morphological dilation was KNN with raw data at 11.24%.

	SVM - erosion	SVM	SVM - dilation
PH	67.38%	70.56%	69.16%
Zone25	77.40%	81.33%	83.33%
Zone100	86.04%	87.26%	89.13%
Raw	85.22%	88.92%	90.36%
HOG	79.48%	84.04%	83.04%
HP+Zone25	83.84%	85.65%	86.62%

Table 1. Accuracy rates of SVM classifier with different features and morphological operators

	KNN - erosion	KNN	KNN - dilation
PH	63.94%	72.73%	78.29%
Zone25	84.06%	87.67%	88.46%
Zone100	82.81%	87.10%	90.22%
Raw	56.95%	78.84%	90.08%
HOG	64.99%	74.20%	81.77%
HP+Zone25	84.31%	88.25%	89.98%

Table 2. Accuracy rates of KNN classifier with different features and morphological operators

All that we discussed above can be clearly seen in our recognition rate matrix (Figure 7). The matrix gives an idea about recognition rates per technique and character. The latter allows us also to notice that applying morphology increases in most cases, the recognition rate while using any method or combination of methods. In addition, it is still hard to raise the recognition rate of characters that have a lot of shape similarities, due to their resemblance with other characters.

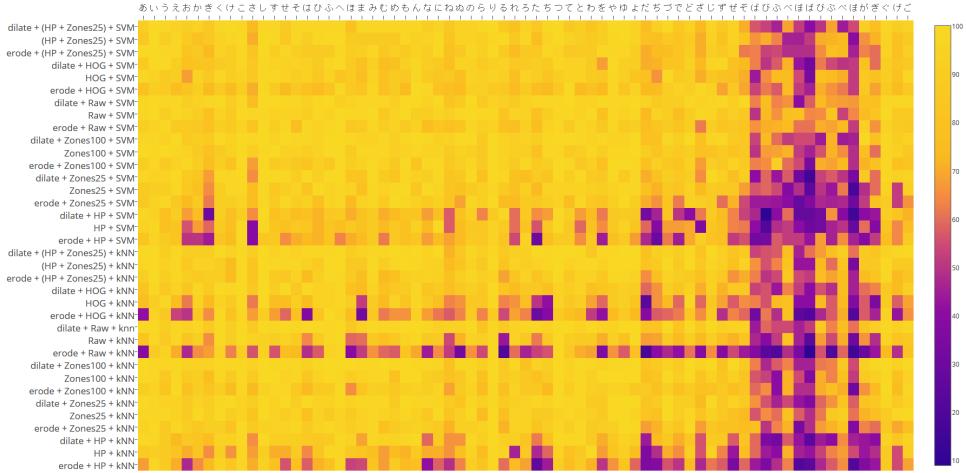


Fig. 7. Recognition rate per character and technique

For example, in figure 8 the characters are very similar. By adding the handwriting style variation, it becomes hard to differentiate between the characters using the algorithms.

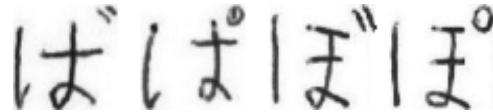


Fig. 8. Example of almost identical characters

5 Conclusion

Throughout our experiment, we collected different handwriting styles from the field, which makes our dataset with significant variability. After investigating different feature selections and applying morphological processing, we conclude that both steps have an important impact on the handwriting recognition accuracy. Since in most cases, we noticed an increase in recognition rate after applying the morphology processing and feature selection by almost 11% when using KNN and by almost 2% when using SVM.

Moreover, during this investigation we proposed a technique that gives a good recognition accuracy rate under constraints of significant variability of the dataset. The work presented is an encouraging evaluation for future investigations in increasing the handwriting recognition rate. In addition, introducing new ap-

proaches and methods in the chain process of handwriting recognition can be very beneficial for offline mode and real-time application.

Acknowledgment

The authors would like to thank the Chair of Distributed Systems of the Institute of Computer Science at the University of Tartu and also a special thanks to Ms Eri Miyano and her students from the Department of Languages of Asian Region at the University of Tartu for helping in the data collection campaign of the Hiragana dataset used in this research work.

References

1. R. Plamondon, S. N. Srihari, "On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey", Journal IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 22 issue 1, 2000.
2. D. Singh; M. Aamir Khan; A. Bansal; N. Bansal, "An application of SVM in character recognition with chain code", IEEE Conference on Communication, Control and Intelligent Systems (CCIS) , 2015.
3. K. B. Baiju, K. Sabeerath, "Online recognition of Malayalam handwritten scripts — A comparison using KNN, MLP and SVM", IEEE International Conference on Advances in Computing, Communication and Informatics (ICACCI), 2016.
4. H. Choudhury; S. Mandal; S. Devnath; S. R. Mahadeva Prasanna; S. Sundaram, "Combining HMM and SVM based stroke classifiers for online Assamese handwritten character recognition" Annual IEEE India Conference (INDICON), 2015.
5. D. K. Patel; T. Som; M. K. Singh, "Multiresolution technique to handwritten English character recognition using learning rule and Euclidean distance metric", IEEE International Conference on Signal Processing and Communication (ICSC), 2013.
6. Otsu N. "A Threshold Selection Method from Gray-Level Histograms", IEEE Transactions on Systems, Man and Cybernetics, 1979.
7. Haralick, Robert M., Stanley R. Sternberg, and Xinhua Zhuang. "Image analysis using mathematical morphology." IEEE transactions on pattern analysis and machine intelligence 4, 1987.
8. S. Das; S. Banerjee, "A Pattern Recognition Approach for Handwritten Hiragana Character Identification", IEEE International Conference on Computational Intelligence and Computing Research, 2012.
9. Y. Shimozawa, "Neural Networks which consist of Simple Recurrent Networks for Character Recognition by Template Matching", Journal of the Information Processing Society 49(10), 3703-3714, 2008.
10. Y Shimozawa; S. Okoma,"The post processing of Character Recognition by Genetic Algorithms", Journal of the Information Processing Society, 40(3) 1106-1116, 1999.
11. F. Mohammad; S. A. Husain, "Character Recognition Using Mathematical Morphology", International Conference on Electrical Engineering, 2007.
12. B. Zhu, X.-D. Zhou, C.-L. Liu, and M. Nakagawa. "A robust model for on-line handwritten Japanese text recognition". IJ- DAR, 13(2):121-131, 2010.
13. J. Tokuno, N. Inami, S. Matsuda, M. Nakai, H. Shimodaira, S. Sagayama, "Context-dependent substroke model for HMM-based on-line handwriting recognition", Eighth International Workshop on Frontiers in Handwriting Recognition. Proceedings, 2002.

14. N. Tsuchiya, S. Ozawa, and S. Abe. Training three-layer neural network classifiers by solving inequalities. In Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on, volume 3, pages 555-560 vol.3, 2000.
15. T. Agui, H. Takahashi, M. Nakajima, and H. Nagahashi, "Recognition of handwritten katakana in a frame using moment invariants based on neural network", IEEE International Joint Conference on Neural Networks, 1991.
16. N. Arica, F.T. Yarman-Vural, "An overview of character recognition focused on off-line handwriting", IEEE Transactions on Systems, Man, and Cybernetics, 2001.
17. I. S. Oh, J. S. Lee, and C. Y. Suen, "Analysis of class separation and combination of class-dependent features for handwriting recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, 1999.
18. H. Takahashi and T.D. Griffin, "Recognition enhancement by linear tournament verification", Proc. 2nd ICDAR, Tsukuba, Japan, pp.585-588, 1993.
19. Maruyama, K.-I.; Maruyama, M.; Miyao, H.; Nakano, Y.; "Handprinted Hiragana recognition using support vector machines"; Frontiers in Handwriting Recognition, Proceedings. Eighth International Workshop, Pages: 55-60, 2002.
20. Barnes, D. Manic, M. Univ. of Idaho, Moscow, ID, USA "STRICR-FB, a Novel Size-Translation- Rotation-Invariant Character Recognition Method", 3rd Conference on Human System Interactions (HSI), pp.163-168, 2010.
21. K. Wang, Y. Y. Tang, and C. Y. Suen, "Multi-layer projections for the classification of similar Chinese characters," in Proc. 9th Int. Conf. Pattern Recognition, pp.842-844, 1988.
22. P. N. Sastry; T.R. V. Lakshmi; N.V. K. Rao; T.V. Rajinikanth; A. Wahab, "Telugu Handwritten Character Recognition using Zoning Features", International Conference on IT Convergence and Security (ICITCS), 2014.
23. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, pages 886–893, 2005.
24. T. Cover and P. Hart. Nearest neighbor pattern classification. In IEEE Transactions in Information Theory, IT-13, pages 21–27, 1967.