International Journal of Electrical and Computer Engineering (IJECE)
Vol. 7, No. 5, October 2017, pp. 2863~2872
ISSN: 2088-8708, DOI: 10.11591/ijece.v7i5.pp2863-2872

Cursive Handwriting Segmentation Using Ideal Distance Approach

Fitrianingsih[1], Sarifuddin Madenda[2], Ernastuti[3], Suryarini Widodo[4], Rodiah[5]
[1, 3, 4, 5]Departement of Informatics, Gunadarma University
[2]Doctoral Program in Information Technology, Gunadarma University

Article Info

Article history:

Received Jan 11, 2017
Revised Apr 7, 2017
Accepted Jun 11, 2017

Keywords:

Cursive
Handwriting
Ideal distance
Offline
Segmentation

ABSTRACT

Offline cursive handwriting becomes a major challenge due to the huge amount of handwriting varieties such as slant handwriting, space between words, the size and direction of the letter, the style of writing the letter and handwriting with contour similarity on some letters. There are some steps for recursive handwriting recognition. The steps are preprocessing, morphology, segmentation, features of letter extraction and recognition. Segmentation is a crucial process in handwriting recognition since the success of segmentation step will determine the success level of recognition. This paper proposes a segmentation algorithm that segment recursive handwriting into letters. These letters will form words using a method that determine the intersection cutting point of image recursive handwriting with an ideal image distance. The ideal distance of recursive handwriting image is an ideal distance segmentation point in order to avoid the cutting of other letter's section. The width and height of images are used to determine the accurate segmentation point. There were 999 recursive handwriting input images taken from 25 researchers used for this study. The images used are the images obtained from preprocessing step. Those are the images with slope correction. This study used Support Vector Machine (SVM) to recognize recursive handwriting. The experiments show the proposed segmentation algorithm able to segment the image precisely and have 97% success recognizing the recursive handwriting.

Corresponding Author:

1. INTRODUCTION

Letter recognition based on handwriting (especially recursive handwriting) has long been identified as a problem that is difficult to solve by computer. The characteristics of offline handwriting are influenced by the nature of individual author, hardware and data acquisition [1-3]. In general, the procedures of recursive offline handwriting recognition start with image acquisition then preprocess the images by correcting the images that contain slanted letter and document. The next procedure is morphology step followed by segmentation process to determine the cutting point of feature extraction and recognition. In this study morphological operation and histogram projection were carried on to explore characteristic of offline handwriting. Morphological steps were performed to obtain (i) the required text and (ii) identification of object position that being represented as regions inside the image. The opening operation was performed to cut off the text objects that connect to one or two other vertex of the text. On the other hand, the operation closing was conducted to remove the small holes inside the segmented letters. There are two steps in segmentation procedure, (i) horizontal projection and (ii) vertical projection. In this study, finding the most critical vertex from characters is an important step before searching for segmentation compiler for characters.

The important vertexes in this study are the endpoint vertex and the vertex of the branch. Both of the vertexes will be used as the starting and ending vertex to start the searching step to find the nearest neighbor to determine the character's segmentation compiler. From each of the important vertex, this study explores and investigates all possible character segmentation compilers [4]. From the input data (handwriting images), we conduct preprocessing procedure using Otsu Method then searching for the possible segmentation zone that can act as the cutting vertex. To obtain the appropriate segmentation result, Genetic Algorithm plays important role performing non-linear segmentation [5].

This study exploring vertical letter separation after skeleton methods was performed to count the amount of pixels in letter that will be used as candidate for point separator. These candidates will be evaluated based on their distance to each other. The candidates that are in seven pixels distance will be merged as a separator point. The letters used for implementation were taken using scanner of digital camera. Right and left contour analysist were carried on since Latin or Roman letter has at least one left contour [6].

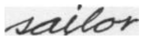
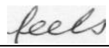
Another research were exploring maximum average letter similarity to find the letter's point for letter separation step. All letter's point candidate that will be used as a separator formed a graph that contain weight of letter similarity. These graphs are used to find the point separator by using average longest path algorithm. Whereas, the similarity of the letters are employed as graph's weight. These weights were classified using SVM. The features that being classified by SVM can be extracted using histograms of oriented gradients. The oriented graph will form a cycle if 0 weight path of point separator's candidate was added to the first candidate. This cycle reduce the difficulties of finding the average longest route to ratio maximum cycle [6].

2. RESEARCH METHOD

2.1. Data Sets

In this study, the input images were offline handwriting letters which were obtained from local dataset and IAM [7] with JPG and PNG format. The size of the images are varied with total images are 999 images. The overall images used in this study are the images with recursive handwriting characteristics. These characteristic contain; (i) normal upright recursive handwriting, (ii) italic letters in overall handwriting (slope) and (iii) tilting letter or text font (slanting). The examples of the images used in this study can be seen in Table 1.

Table 1. Experimented Image (Local and IAM Dataset)

	Image	
Slope		
Slant		

2.2. Preprocessing

Preprocessing step in this study were performed by corrected the images that have slant and slop characters. This processing are done using affine 2D transformation. The slope of the document often occur in image acquisition resulting the handwriting inside document written in tilt position. The characteristic of recursive handwriting are being influenced by many factors. Some of the factors are (i) the style of each individual, (ii) the orientation of the handwriting (to determine the status and position of the text, whether the text is horizontal or vertical) and (iii) whether the handwriting form a certain angle [5]. The image slant correction can be done by detecting the angle of slant of the original image (θ). This angle generally ranged between 0° until 180°. The image will be reconstructed till generate horizontal angle of 0. In recursive slant letter, shearing operation was performed. This operation were carried out by shifting initial image in the direction of x or y-axis by a certain scale. The shifting of the images were performed by defining the matrix below [8]:

$$\begin{bmatrix} 1 & sh_y & 0 \\ sh & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where :

sh_y =Shearing factor along y-axis

sh_x = Shearing factor along x-axis

The result of slanting correction in preprocessing step can be seen in Figure 1.

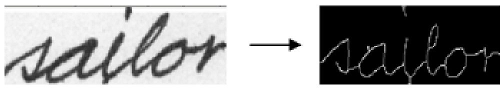


Figure 1. The result of slanting correction letter

2.3. Morfology

This study used morfologi step of Zhang Suen’s method to perform thinning process. This process is very crucial since it was the first step before performing the next steps [8]. Thinning will eliminate the outer most layer of handwriting image pattern leaving only the line that consist of one pixel only. This line is called skeleton of the image. Each iteration of this method consist of two sub successive iteration that performed on contour point of image region, assuming a pixel with value one (1) is pixel foreground and pixel with value zero (0) is the background. An object pixel have 8-neighbor rule. The first iteration will represent Cartesian coordinate for each element as small squares. The alteration of skeleton in this study were made in binary form that is performed in one image. If the pixel in the handwriting image have at least one of its four neighborhoods (p0, p2, p4, p6) with the value less than its each grey level value or the whole eight (8) of its neighbor is less than their grey value, then its binary value turned into 1 or 0. The second iteration made during bwmorph instruction. The origin of structuring elements positioned in each of the pixel in the input image, then the value of structuring elements correspond to the pixel value.

2.4. Histogram

This study used morfologi step of Zhang Suen’s method to perform thinning process. The next step after performing recursive handwriting thinning morphology process was to obtain the image intensity information using histogram method. Histogram of a digital image has intensity level ranged between [0, L-1] with discrete function [9].

$$H(r_k) = n_k \tag{2}$$

For recursive image with pixel level 0 – 225, the histogram will draw pixel frequency distribution in 0–225 interval. On the other hand, for the recursive binary image which own 0 and 1 level, the histogram will draw their frequency distribution on foreground with 1 value. The next step is to add this value vertically. The process can be seen in Figure 2.

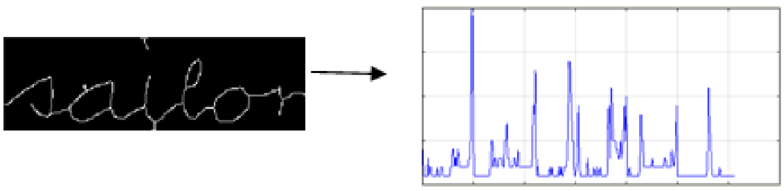


Figure 2. Histogram of vertical image pixels quantity

2.5. Segmentation using Ideal Distance

Segmentation is the process of recursive handwriting partition into individual letter. Before the partition step performed, it is important to find the segmentation point inside the segmented image. This segmentation point was taken from the sum of vertical image pixel. The sum was carried out downward until the bottom of image pixel that has passed through preprocessing step. The image that has passed through preprocessing step will be used to decide the segmentation point. The foreground that is part of this handwriting image (the image that has passed through the processing step) has one (1) value, therefore when we sum up the pixel vertically, the result is a value that will be used as the segmentation point. Before examine the segmentation point, it is important to compute the ideal distance of the points in an image. The ideal recursive handwriting image distance is the ideal replacement of segmentation point in order to avoid cutting off the section of the letter in an image. As long as the spacing distance is ideal, the segmentation point cannot be done. On the other hand, after passing through the ideal spacing distance, the segmentation can be done by examining the amount of vertical pixel. The ideal spacing distance is calculated from the image width divided by the devisor that follows the width of the image in order to adjust the number of letters that contained in an image. For example, the width of the image after preprocessing step in Figure 3 is 310 pixels with the ideal spacing distance 38.75. The value 38.75 indicating subsequent to placement segmentation point, within 38.75 distance pixel is impossible to re-segment the pixel point since the distance spacing of the letter is already ideal.

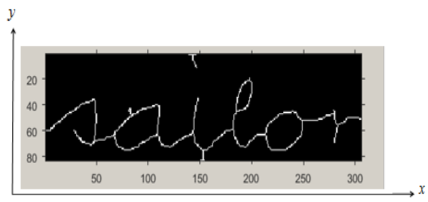


Figure 3. Image with 310 x 85 pixels

Figure 4 shows the algorithm proposed in this study to determine tha devisor value dan segmentation point based on ideal distance between letters.

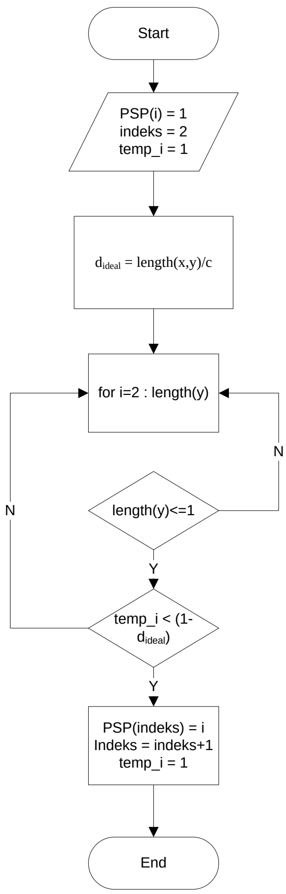


Figure 4. Flowchart to determine dividers value

There are some proposed steps to determine the segmentation point of a candidate image in recursive offline handwriting recognition:
1. Determine the image width(x) of handwriting $f(x, y)$
2. Create a Possible Segmentation Point (PSP) array to store segmentation point with the first index is the first horizontal pixel position.
3. Create variable temp_i as pointer checker.
4. Determine the ideal distance using the formula (3) :

$$d_{ideal} = \frac{length(y)}{c} \tag{3}$$

where :

$$\begin{cases} x \geq 500_{px} & ; \quad c = \frac{f(x,y}{12} \\ x \geq 300_{px} \leq x \leq 499 & ; \quad c = \frac{f(x,y)}{8} \\ x \leq 299_{px} & ; \quad c = \frac{f(x,y)}{6} \end{cases}$$

d_{ideal} = Ideal distance
$f(x, y)$ = Recursive handwriting image
x = The Image width
y = The Image height

5. Create iteration starting from the 2nd pixel untill the width of image pixles, then add each iteration in new pixel.
6. If the sum of vertical image pixel ≤ 1, then we redo the checking. If variabel temp_i < from the pixel checking possition is reduced with d_{ideal} then:
   a. The content of PSP array index is the newest position of current iteration
   b. Pointer indeks PSP array increase one
   c. Pointer temp_i is worth the current iteration position.

2.6. Words Feature Extraction Recognition

The training set used in this study are SVM [9]. After training the classifier using the result of feature letter by letter extraction from training set, the next step is performing feature image extraction from the test set. Test set is a set of images that contain the letters to be recognized. The result of this test set is the letter's weight and label that will be used for recognition step.

The steps of feature extraction and recursive offline handwriting recognition can be seen below:
1. Load 'trainedClassifier_Final_20.mat' which are the results from the training set.
2. Load the segmented image in img variable then performing feature extraction based on HOG to img. The result of feature extraction will be stored in testFeatures variable.
3. Performing score prediction to compare the result form feature extraction testFeatures variable with the score of classifier.
4. The obtained score prediction will be compared with score label of classifier and then will be loaded in label variable.

The general chart of feature extraction and recursive offline handwriting recognition can be seen in Figure 5 below.

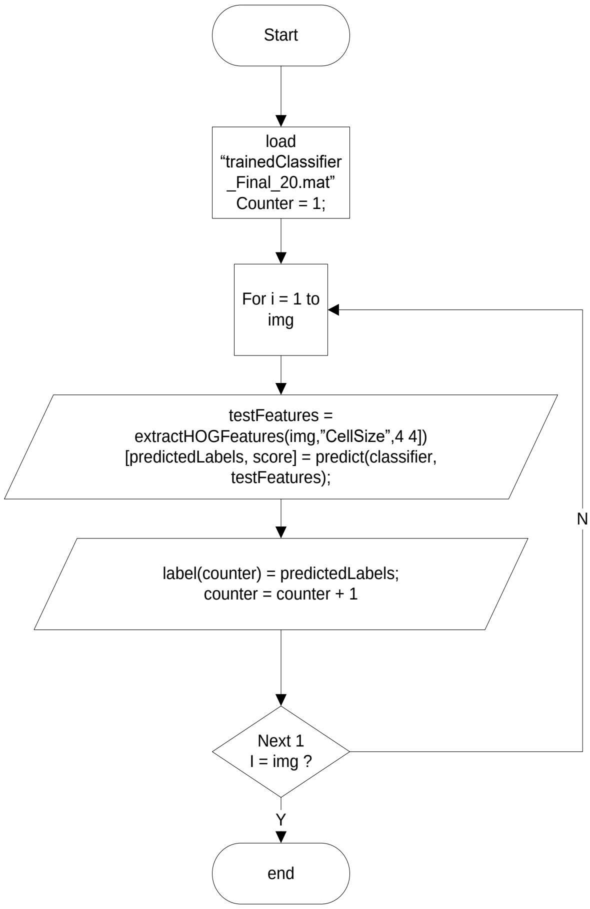
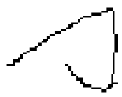
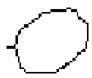


Figure 5. Feature extraction and recursive offline handwritting recognition

Table 2 shows the result of feature image extraction with their weights and labels

Table 2. Image, Weight and Label from Feature Extraction

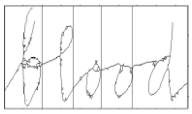
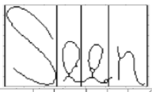
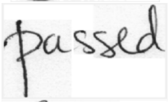
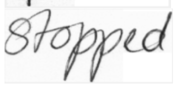
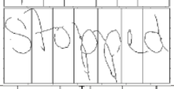
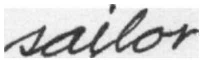
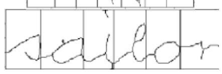
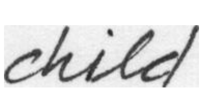
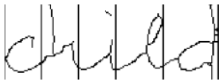
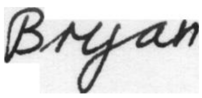
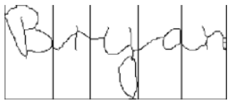
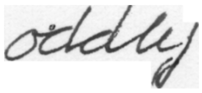
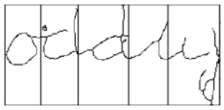
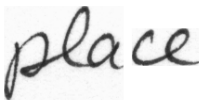
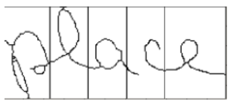
Citra	Bobot	Label
	0.9901	s
	0.9908	a
	0.9925	i
	0.9973	l
	0.9901	o
	0.9906	r

3. RESULTS AND ANALYSIS

3.1. Experimental Image

Segmentation process is based on segment point that being applied inside the program. This segmentation point was obtained from image column that contain vertical pixel less than or equal to one and cannot be found in ideal distance. For example, Table 3 shows 10 images out of 999 images that were successfully segmented.

Table 3. Segmentation Based On Ideal Distance

No	Original Image	Segmented Image
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		

The results from segmentation point placement were not always precise and in correct position. These were due to some characteristic of recursive handwriting that differs with the printed handwriting. The differences are:

1. There is no empty space between letter in recursive handwriting, therefore the quantity of vertical pixels in segmentation point is more than or equal to one.
2. There are some letters that have characteristic as segmentation candidates. These letters have letter lines in which their vertical pixels quantity equal to one. For example U, V, W, N, M, and H.

There were also letters that are not close perfectly, For example the writing of cursive ‘a’ that was written with imperfect closed writing resulted the loop in letter ‘a’ determined as a cutting point. As a result, these letters were not detected as segmentation candidate. Figure 6 shows unsuccessful segmentation candidate.

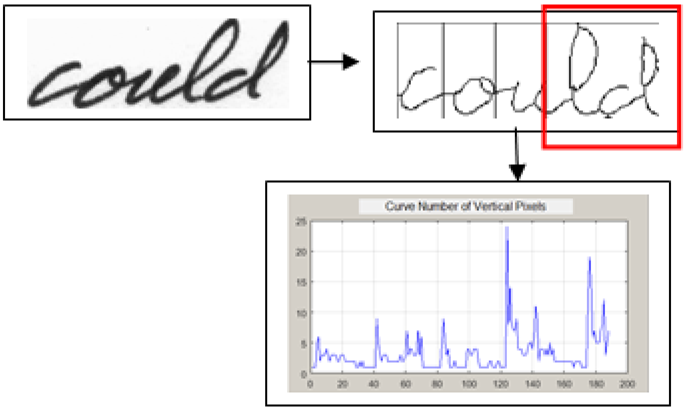


Figure 6. Letters with unsuccessful segmentation

Figure 6 shows the unsuccessful image segmentation with 3o slant of letter. Letter L and D were not correctly segmented because of the amount of vertical pixels between letter L and D ranged on y-axis between 130 until 150 pixels. The resulted pixels did not meet the conditions less than or equal one. As consequences, the cutting process cannot be done correctly at this point. The example of image recognition can be found in Figure 7.

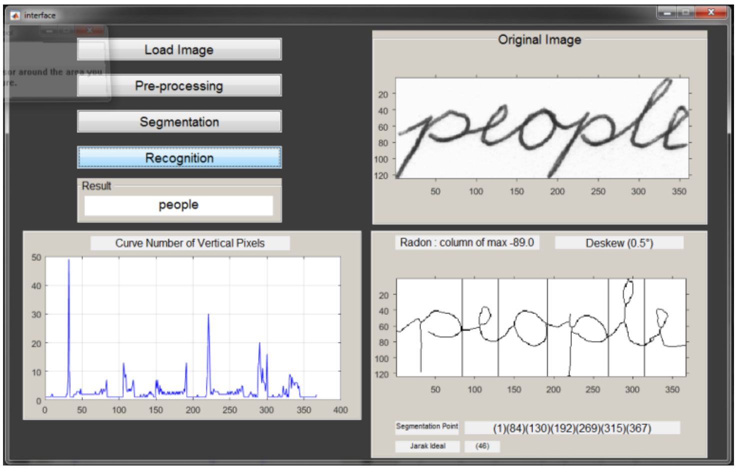


Figure 7. The result of letter recognition

Previous research (Salvi et al, 2012) performed segmentation method on finding the separation point of the letters using the average of maximum letter's similarity. All the candidates of separation point will formed a graph with letter similarity as their weight. The graph employed in their research try to find the separation point using the longest average route algorithm. This method produced errors in segmentation process. The errors can be seen in Figure 8. The created graph have the edge from candidate no 2 and all the route the same as the image cutting based on candidate no 2. The final step of second validation is searching the longest greedy route from the created graph, resulted over segmentation on handwriting image.

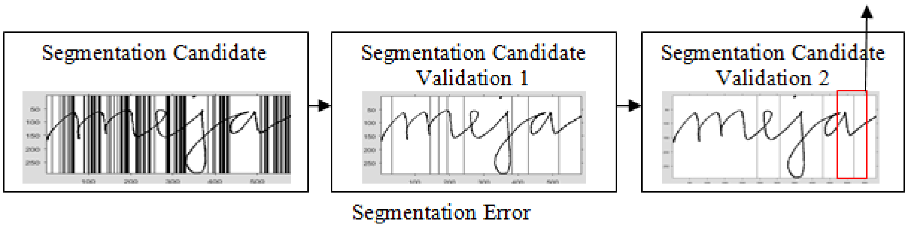


Figure 8. Greedy algorithm segmentation result

3.2. The Result of System Perfomance

The performance measurement of offline recursive handwriting in this study was computed using ratio of successfulness and failure. The computation can be done using the formulas below:

$$The\_Success\_Ratio = \frac{\Sigma SuccessfulSegmentation}{\Sigma TotalImages} x100\% \quad (4)$$

$$The\_Failure\_Ratio = \frac{\Sigma UnsuccessfulSegmentation}{\Sigma TotalImages} x100\% \quad (5)$$

The result of successful segmentation ratio and failure based on active image contour in this study can be seen in Table 4 and Figure 9.

Table 4. Segmentation Performance

No	Image Source	Success	Fail
1	Local Dataset	495	5
2	IAM Dataset	487	12
Total		982	17

$$The\_Success\_Ratio\_of\_Segmentation = \frac{982}{999} x100\% = 98\%$$

$$The\_Failure\_Ratio\_of\_Segmentation = \frac{17}{999} x100\% = 2\%$$

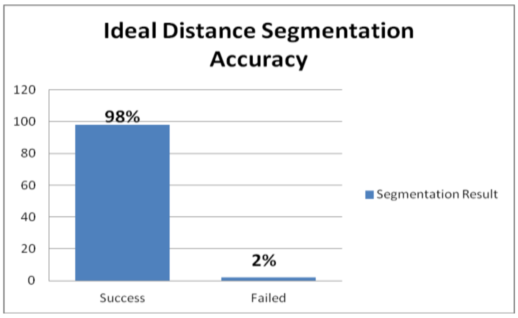


Figure 9. Segmentation accuracy

4. CONCLUSION

Based on the overall description and experiment's result, the proposed algorithm able to segment the offline recursive handwriting images precisely (around 98% accuracy) based on its cutting point, by calculating the ideal distance from offline recursive handwriting. There were 999 recursive handwriting data. Unsuccessful segmentation caused from handwriting images that experienced distortion on their y-axis (ranged between 130 until 150 pixels). These resulted pixels do not meet the conditions (≤ 1), as consequences the cutting point on those pixels cannot be perform. The accurate point to cut on segmentation step will achieve the accuracy of handwriting recognition till 98% of success.

ACKNOWLEDGEMENTS

We are indebted to the experts who have contributed towards development of the template.The authors would like to acknowledge to Gunadarma University.

REFERENCES

[1] Papandreou and Gatos, “Slant Estimation and Core-region Detection for Handwritten Latin Words.” Pattern Recognition Letters 35, pp 16-22. Elsevier. 2014

[2] Plamondon, R dan Djioua, Moussa. 2005. “Handwriting Stroke Trajectory Variability in the Context of the Kinematic Theory.” Advances in Graphonomics: Proceeding of IGS.

[3] Agrawal, Vashishtha, Kumar, “Slant Angle Estimation in Handwritten Documents,” Int. Journal of Computer Science and Management Studies. Vol. 14, Issue 5, pp. 1-5, 2014.

[4] Santos, Gabriela. Clemente, Tsang Ing Ren and Calvalcanti. “Text Line Segmentation Based on Morphology and Histogram Projection,” 10th International Conference on Document Analysis and Recognition, 978-0-7695-3725-2/09, IEEE. 2009

[5] Saba, Su Long, and Rehman. “Non-Linear Segmentation of Touched Roman Characters Based on Genetic Algorithm.” International Journal on Computer Science and Engineering (IJCSE). Vol. 02, No. 06, pp 2167-2172. 2010

[6] Choudhary, Rishi, and Ahlawat. “A New Character Segmentation Approach for Off-Line Cursive Handwritten Words. Information Technology and Quantitative Management (ITQM2013).” Procedia Computer Science 17, pp 88-95. Elsevier. 2013

[7] IAM Handwriting Database, sumber: http://www.fki.inf.unibe.ch/databases/iam-handwriting-database, Tanggal akses : 18 November 2016

[8] Fitrianingsih, Sarifuddin Madenda, Suryarini Widodo, Rodiah, “Slant Correction and Detection for Offline Cursive Handwriting using 2D Affine Transform,” International Journal of Engineering Research & Technology, ISSN. 2278-0181, Vol. 5 Issue 8, pp. 568-572, August 2016

[9] Catalin and Ruxandra, “Support Vector Machines and Evolutionary Algorithms for Classification.” ISBN. 3319069403. Pub. Springer Verlag Gmbh. pp. 78-114. 2014

[10] Salvi, Zhou, Waggoner and Wang. “Handwritten Text Segmentation using Average Longest Path Algorithm.” IEEE, pp. 506-512. 2013.