

Sum of Squared Difference (SSD) Template Matching Testing on Writing Learning Application

Widya Dharma Sidi^{a1}, I Gede Arta Wibawa^{a2}

^aInformatics Department, Faculty of Science and Mathematic, Udayana University
Kampus Bukit Jimbaran Street, Jimbaran, Badung, Bali

¹widya.sidi@cs.unud.ac.id

²artawibawa@cs.unud.ac.id

Abstract

This research was conducted to determine the accuracy of the Sum of Squared Difference (SSD) Template Matching method in the Application of Learning Numbers Writing Games. This game application is an application created to help early childhood in recognizing Arabic numbers, namely numbers from 0 to 9. In the SSD Template Matching method there are several processes including Preprocessing, thinning, feature extraction, and classification (SSD template matching). In testing the game application involves 10 respondents who were asked to write numbers correctly as requested by the application. For each number writing test, it is tested by 3 times. From the tests conducted, obtained an accuracy of 94.67%.

Keyword: *Template Matching, Sum of Squared Difference (SSD), Education Game, Optical Character Recognition, Mobile Learning*

1. Introduction

Technological developments bring various positive and negative impacts. One impact caused by technological development is the use of gadgets in early childhood. This certainly has positive and negative impacts. One of the negative impacts generated is that children become dependent on gadgets. One of the causes of this is game applications that are currently very much on gadgets, especially smartphones. Therefore we need a game application that is not only entertaining, but able to provide knowledge to children. One important education in early childhood is the introduction of numbers. Therefore, by utilizing the development of existing technology, the authors are interested in developing applications for learning to write numbers called "Ayo Menulis". This learning method is known as mobile learning. Mobile learning is an innovation in learning methods by utilizing mobile devices such as smartphones, laptops, etc. [7]. With this game application is expected to help reduce the negative impact of the use of gadgets especially smartphone on children, by providing education on number recognition.

This game application will display Arabic numbers from 0 to 9, which later the user is asked to write down the number displayed on the canvas provided. In identifying whether or not the user's input with the numbers displayed, template matching method is used. The template matching method used in this study is the Sum of Squared Difference (SSD) template matching. The principle of this method is to squeeze the results of the reduction of each pixel between two images (reference template and image) [4]. Before the SSD matching template process is performed, namely pre-processing, thinning, and feature extraction. In the pre-processing process resizing and thresholding are carried out.

Previous research related to this research is research conducted by Made Sulatri Dewi et al. in 2014 concerning the development of an application for the introduction of Balinese script using the template matching method. In that study tested the accuracy of the applications developed. There

are two test subjects in the study, namely admin and user, and obtained accuracy results on the admin with an average of 95.45% while the user obtained an average of 86.25% [3].

The second research related to this research is the research conducted by Suryo Hartanto et al. in 2014 which included optical character recognition using the template matching correlation method. Whereas in this study the Sum of Squared Difference (SSD) template matching method will be used. In that study tested the accuracy of the template matching correlation method in the optical character recognition program by testing 5 font types. Testing in that study was carried out three times, with the results of the first test 92.24%, 93.54% for the second test, and 92.90% for the third test. With an average success rate of 92.90% [6].

2. Research Methods

Character identification can be done by various methods, the method used in this study is template matching. In general, the process of template matching is to operate between image input and template image. The flow of the template matching method in this study, can be seen in the flowchart (figure 1).

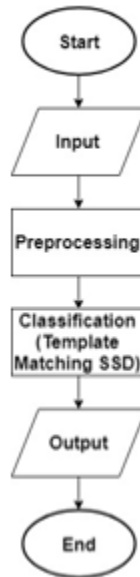


Figure 1. Template Matching Flowchart

2.1. Preprocessing

The preprocessing process is an important step in the template matching method. Because at this stage there is a resizing process that is adjusting the size of the input image with the template. This is important because the template matching method can be done if the pixel sizes of the two images are matched the same [5]. In addition to resizing there are also other processes such as thresholding, thinning, edge removal.

2.1.1. Resize

The input image obtained from the user's handwriting cannot directly process with the template matching method. This is due to the difference in pixel size between the two images. In handwritten images, the pixel size is 192 x 192 while the template has a size of 64 x 64 pixels. The image in handwriting is bigger because it makes it easier for users to write numbers. Because the template method compares each pixel between two images of the same size, it is necessary to adjust the image size in the handwritten image.

2.1.2. Thresholding

The basic principle of the thresholding method is the separation of each pixel into different classes depending on the gray level of each pixel. Ideally before doing Thresholding, a grayscale process is needed, which is the process of changing the color of the image to grayish. But because in this research the image used is a black and white image, so the grayscale process is not carried out. So that the handwritten image and black template (foreground) will be initialized at pixel value 1, while the white color (background) will be initialized with pixel value 0.

2.1.3. Thinning

Thinning is the process of converting objects in an image into lines, which can also be called skeleton. The goal is to reduce the components in the image to become information that is fundamental. In this study the thinning process uses the Stentiford method. The reason for using this method is because it produces better thinning results compared to other methods, namely Zhang-Suen. This was discussed in a study conducted by Eva Fasdiana in 2017, which concluded that the Stentiford method produces better thinning results but requires a longer execution time than the Zhang-Suen method. In the thinning process, there are two methods that can be used, namely Stentiford and Zhang-Suen [2]. The length of the process in the Stentiford method is due to this method matching every 3x3 pixels in the image processed with four 3x3 pixel templates (figure 2).

The Stentiford method is iterative which is useful for eroding the outermost pixel layer so that no more layers can be removed. The principle of this method is to take a 3x3 pixel in the image, then match it to four templates (figure 2). If the pixels that are matched do not match then the pixel will be marked, then it will shift to the next pixel until all the pixels in the image have been matched. Pixels that were previously marked at the end of the iteration will be deleted. The steps of the Stentiford method are as follows.

1. Look for the pixel in the coordinates (i, j) that matches the T1 template (figure 2). With this template all the upper pixels of the image are removed.
2. If the midpoint pixel is not an endpoint and has a connectivity number = 1, mark the pixel to be deleted at the end of the iteration.
 - Endpoint pixel : pixel is the final boundary and is only connected to 1 pixel. Example: a black pixel has only one neighbor which is also black from eight possible neighbors.
 - Connectivity number : is a measure of how many objects are connected to a particular pixel (calculated according to the formula below).

$$Cn = \sum_{k \in S} Nk - (Nk \cdot Nk + 1 \cdot Nk + 2)$$

Where:

- Nk : value of 8 neighbors around the pixel to be analyzed (central pixel) and value $S = \{1,3,5,7\}$
 - $N0$: value of the central pixel
 - $N1$: the value of the pixels to the right of the central pixel and the rest are numbered sequentially by counterclockwise
3. Return to step 1 if the pixel location matches the T1 template
 4. Then repeat steps 1 - 3 for the rest of the templates that have not been matched (T2, T3, T4)
 5. If there are pixels that have been marked before for the erasure process, then delete them by changing them to white

6. If there are still pixels that can be deleted in step 5, then repeat all processes starting from step 1, if not the process ends

All four templates have different functions. The T1 template functions to look for pixels that can be removed at the top edge of the object, moving from left to right up and down. T2 template functions to match the left edge of the object, which moves from bottom to top, from left to right. The T3 template will look for pixels at the bottom edge, moving from right to left, bottom to top. T4 will match the pixels on the right side of the object from top to bottom, right to left [2].

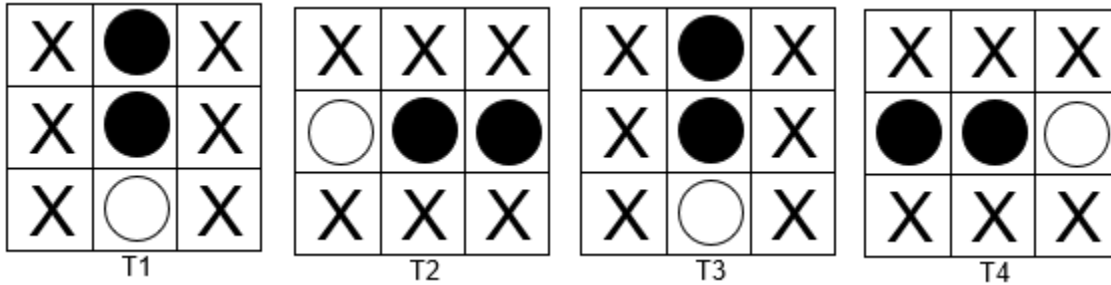


Figure 2. Zhang-Suen Method Template

2.1.4. Edge Removal

Deleting the edges of the image is necessary, to optimize the matching process between the image and the template. Because if the image and template have edges, it will reduce the number of matching pixels because the edge pixels are also matched. Therefore in this study, the template no longer contains edges. But in the picture it is necessary to remove these edges. In this study the removal of edges in the image using the Connected Component Labeling (CLC) method. Connected component labeling is the process of giving different labels to each character, so that the characters can be separated from one another based on the label they have. The advantage of the connected component labeling method is that it is not affected by the slope of the object, so that it can still separate objects well even though the position of the object in the image is tilted (during the threshold process successfully separating objects clearly). The essence of the algorithm in CCL is by checking the value of the first black pixel with the scanline from the top right bottom left. If the next black value is found, it will check the black pixel on the left and the top. If another black pixel is found then it is still considered to be in one region, if not then check again horizontally from left to right [1].

2.2. Classification using *Sum of Squared Difference (SSD) Template Matching*

Template matching method is one method that can be used in classifying data. The workings of this method are to match the input image with a template image that has the same pixel size, then the appropriate number of pixels will be calculated. One type of template matching method is the Sum of Squared Difference (SSD) template matching. In the SSD template matching method, the measurement of compatibility is based on the degree of difference in pixel intensity between the two images (template and input image). The principle of this method is to squeeze the results of the reduction of each pixel between the input image and the template. In general, SSD directly uses the formulation of the number of quadratic errors, with the following formula [4].

$$R(x, y) = \sum_{x', y'} (T(x', y') - I(x + x', y + y'))^2$$

Where:

R : Correlation value between 2 matrices, that is image with reference template

Q : Matrix of template images
I : Image matrix or test data
x : Rows of the image matrix or test data
x' : Line from the template matrix
y : Column of the image matrix or test data
y' : Column of the template matrix

Images will be classified correctly if the value of the image correlation value is greater or equal to the template correlation value.

3. Result and Discussion

3.1. Preprocessing

3.1.1. Resize

At the stage of resizing the application, it runs well. The original image size is 192 x 192 pixels, successfully reduced to 64 x 64 pixels. This can be seen in figure 3.



Figure 3. Resizing Result

3.1.2. Thinning

Before thinning is done, the steps that need to be done are Thresholding. Thresholding is needed, to facilitate the execution of the Stentiford method. The results of thinning using the Stentiford method can be seen in the figure 4.

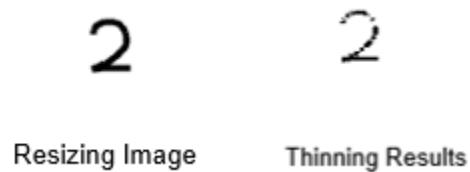


Figure 4. Thinning Result

3.1.3. Edge Removal

After the thinning step, the step that needs to be done is to erase the edges of the image. Deleting the edge of the image in this application has been going well. The results of edge removal using the Connected Component Labeling method can be seen in the figure 5.

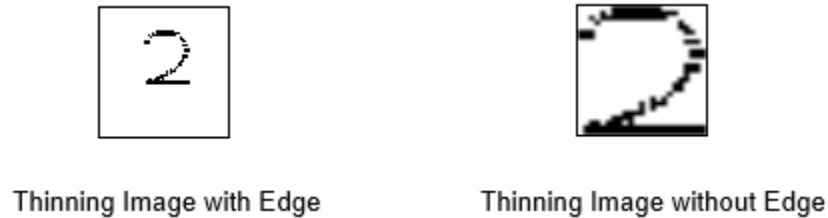


Figure 5. Edge Removal Result

3.2. Classification using *Sum of Squared Difference (SSD) Template Matching*

At this stage, the results of the image that has gone through the preprocessing process are matched with the template that has been set. In this study to test the classification of data, it is done by writing numbers as in figure (). If observed both images have similar shapes. Therefore, the handwritten drawing should be correct. In the template get a correlation value of 269 while in the image get a correlation value of 375. As explained in point 2.3 that the image is said to be true if the image correlation value is greater or equal to the template correlation value. This proves that the Sum of Squared Difference (SSD) Template Matching method is running properly.



Figure 6. Image and Template

3.3. Accuracy Testing

The test was carried out to determine the accuracy of the Sum of Squared Difference Template Matching method in the "Ayo Menulis" Game Application. The scheme in this test is that the user is asked to write 10 Arabic numbers, namely the numbers 0 through 9 correctly. It aims to get the accuracy of the SSD template matching method in this game application. The number of respondents involved in this test is 10 people, consisting of 1 admin and 9 users randomly. Each number writing is done three times, which aims to obtain more accurate results. To display the application interface can be seen in figures 7 and 8. Then for the results of testing this game application, can be seen in table 1.



Figure 7. Main Page Interface



Figure 8. Writing Page Interface

Table 1 Accuracy test results

Respondents	Number of Correct Answers	Number of Wrong Answers	Accuracy
1	29	1	96.67 %
2	27	3	90 %
3	29	1	96.67 %
4	29	1	96.67 %
5	28	2	93.33 %
6	29	1	96.67 %
7	28	2	93.33 %
8	29	1	96.67 %
9	29	1	96.67 %
10	27	3	90 %
Average			94.67%

From table 1 it can be seen that the smallest percentage of accuracy obtained from application testing is 90%, with the total number correct is 27 of 30 tests. And the biggest percentage of accuracy is 96.67%, with the total number correct is 29 of 30 tests. The average accuracy obtained from all respondents is 94.67%. When testing, respondents average error only when entering a certain number. To find out more details can be seen in table 2.

Table 2 Character input error

Respondents	Incorrect Number Testing
1	Number 2
2	Number 1,2, and 4
3	Number 4
4	Number 2
5	Number 2 and 9
6	Number 9
7	Number 2 and 4
8	Number 4
9	Number 2
10	Number 2,4, and 9

Can be seen in table 2, the average test error is found in numbers 2, 4 and 9. This is due to differences in the way the numbers are written. For example, in writing number 2 which has a difference in the lower left corner. Some do circular writing at an angle, there are also those who write it without circular. Whereas on the template that has been set, the correct writing should be without circular at that angle. So it can be ascertained, if the method of writing in a circle at the

bottom left corner produces the wrong test results. To clarify the different ways of writing, can be seen in Figure 6.

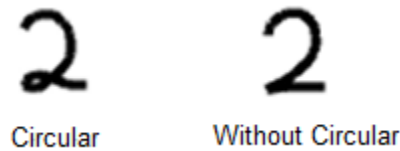


Figure 9

Based on tests conducted, it was found that the template matching process that occurred in this game application was running according to its function. Then based on tests conducted to test the accuracy of the SSD template matching in game applications, an accuracy percentage of 94.67% is obtained. This percentage is obtained from the average percentage of the results of the ten respondents involved in testing game applications. According to the research conducted by Suryo Hartanto, the percentage obtained in this study can be categorized very well [6]. From these results, the SSD template matching method is feasible if implemented in this game application. Even so to get better results on the SSD matching template method, more templates are needed to match the image (test data). Because in this game application, the template used is only one template. This is what causes the lack of accuracy of this method.

4. Conclusion

From the research that has been done, the following conclusions are obtained:

1. In the research conducted, the Sum of Squared Difference (SSD) template matching method has been successfully implemented in the "Ayo Menulis" game application. The processes in this template matching method are also running well. The process is Preprocessing, Thining, Feature Extraction, and Classification (template matching). This can be seen in the test results of each process that has also been carried out in this study.
2. In this study, testing has been done on game applications involving 10 respondents. The test results obtained the smallest application accuracy that is equal to 90% and the largest is 96.67%. With the average percentage of accuracy being achieved at 94.67%. This shows that the method of Sum of Squared Difference (SSD) template matching is feasible to be implemented in this game application.
3. To obtain better accuracy in this game application, you can replace existing methods or optimize the Sum of Squared Difference (SSD) method matching template by adding a template that will be matched with the image (test data). Because this research only uses one template, which causes a decrease in accuracy in this method.

References

- [1] Kukuh Yudhistiro. Menghitung Obyek 2d Menggunakan Connected Component Labeling. Seminar Nasional Sistem Informasi UNMER. p. 499 – 510. 2017
- [2] Leornadus Beni Aji Prabangkoro. Algoritma Thinning Stentiford Untuk Mendapatkan Bentuk Dasar Huruf. Sanata Dharma University. 2018.

- [3] Made. Sulatri. Dewi. and Made. Windu Antara. Kesiman. *Aplikasi Pembelajaran Pengenal Aksara Bali Menggunakan Metode Template Matching. Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI). vol. 3, Nomor 1, p. 42 - 50 . 2014*
- [4] M. B. Hisham. *Template Matching Using Sum of Squared Difference and Normalized Cross Correlation. 2016*
- [5] Raden Sofian Bahri. *Perbandingan Algoritma Template Matching Dan Feature Extraction Pada Optical Character Recognition. Jurnal Komputer dan Informatika (KOMPUTA). Edisi. I Volume. 1, p. 29 – 35. 2012*
- [6] Suryo Hartanto. *Optical Character Recognition Menggunakan Algoritma Template Matching Correlation. Jurnal Masyarakat Informatika, vol.5, nomor 9. p.1 – 11. 2014*
- [7] Yuyu Laila Sulastrri and Luki Luqmanul Hakim. *Pembelajaran Berbasis Mobile. Jurnal Pengajaran MIPA. Volume 19, Nomor 2. p. 173 – 178. 2014*