The object detection system of balinese script on traditional Balinese manuscript with findcontours method

Ida Bagus Putra Manuaba[1]*, Komang Ayu Triana Indah [2]

[1,2] *Politeknik Negeri Bali, Indonesia*

Corresponding Author: *manuabaputra@pnb.ac.id*

Abstract: *Lontar* is a traditional Balinese manuscript with a Balinese script in it. Balinese traditional manuscripts can be more than 100 years old. The age factor of the Balinese manuscript has an impact on the Balinese script in it. Balinese script that has been written more than 10 years tends to be darker, this makes Balinese script not visible well, and this affects the image quality of the manuscript. This thing becomes the main issue in this research, Balinese script detection on Balinese manuscript images. the first of all is image processing using edge detection, canny and Sobel becomes the main algorithm of this process. After image processing, the Balinese manuscript will be processed with the findcontour method to detect an object that contains in it. The final process of this detection system is to separate detected objects into three main groups namely noise object, Balinese script object, and hole object. Application (Balinese script object detection system) is more accurate in detecting Balinese script objects in Balinese script under 1 year (new script), it tends to be more likely to find noise/dirt, this is because the writing of the *lontar* using a pencil first before using the knife media, this adds to the noise or dirt detected by the application The findcontour method can detect Balinese script objects with a detection result of 30% - 70% Balinese script objects.

Keywords: object detection, Balinese manuscript, edge detection, findcontour

Introduction

Lontar is a traditional Balinese manuscript with a Balinese script in it. a few years ago, a Balinese manuscript has been used to record the historical knowledge. Currently, *lontar* is a Balinese cultural product that has been recognized as a world cultural heritage [1], and Balinese scriptwriting has been cultural preservation. Digitalization of Balinese script from the manuscript to image is one of the cultural preservation. Digitalization of Balinese script is not easy to do, the first of all is Collecting Balinese manuscript and making it into an image is one of the digitalization of Balinese manuscript.

The object detection system of Balinese script on traditional Balinese manuscript is one of the digitalization of Balinese script. To detect, the Balinese script object on the traditional Balinese manuscript becomes the main issue of this research. The detection system for Balinese script objects in traditional Balinese script is one of the efforts to digitize Balinese script. Detecting Balinese script objects in traditional Balinese script is the main problem in this research. There are several processes to detect Balinese script object on Balinese manuscripts, like preprocessing, object detection with findcontour, and object filtering process.

The process of detecting of Balinese script on Balinese manuscript image is carried out in several stages such as preprocessing, contour segmentation, and object filtering processes. is an effort to improve the image, the reason or basis for this stage is that the input image is not good, such as the amount of noise that appears in the image preprocessing input, this is caused by the age and condition of the Balinese manuscript. Several edge detection algorithms are applied to perform preprocessing such as Sobel Operator and Canny.

The next process of this research is contour segmentation, in this process, the input image will be mapped into segments. The input image will be grouped into two parts, namely the segment which is the object on the palm leaf, and not the segment, namely the palm leaf media itself. After this process, the objects that have been segmented will be processed again in the object filtering process. The object being segmented can be in the form of a Balinese script or noise detected in the input image, so it is necessary to carry out an object filtering process. The process of filtering or filtering objects uses several parameters, such as size, shape, and distance. Objects that are filtered using the above parameters are Balinese script objects, while objects that are not filtered or do not match the above parameters are categorized as noise objects.

Methodology

System Overview

In general, the process in research is grouped into 3 three main processes. The first step of the group is image enhancement using edge detection with Sobel operator and prewitt. The purpose of this process is to improve the image, and clearly show the difference between the object and the background. The process takes input as an image and then apply efficient algorithms, and the results may be image, data, or features associated with that image [2].

The second step of the object detection system is to detect the object in the image. Object detection is a basic research direction in the fields of computer vision, deep learning, artificial intelligence, etc [3]. In this research, object detection will detect all of the objects in the Balinese manuscript image. In the process, objects are divided into three main groups, the first is noise objects, Balinese script objects, and holes objects. A Balinese script object is a single object that will be stored by the system. While noise and hole objects will be considered as non-literal objects.
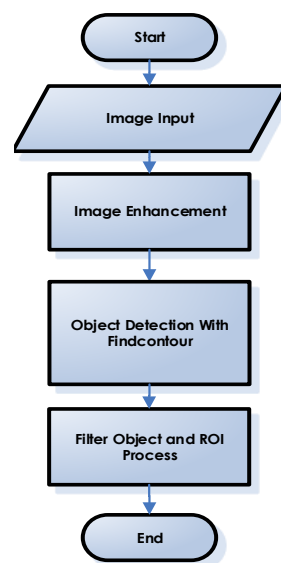

Figure 1. System overview

Data

The data that will be used in this study is in the form of Balinese script images. The size of the documented Balinese *lontar* script is 30 cm (centimeter). There are 2 groups of Balinese script image data used in the research
1. New manuscript, a new manuscript is a manuscript that has just been written and is less than one year old (figure 2 a. New Balinese manuscript).
2. Old manuscripts, Old manuscripts are *lontar* manuscripts that have been previously written and are more than 1 year old (figure 2 b. Old Balinese manuscript).
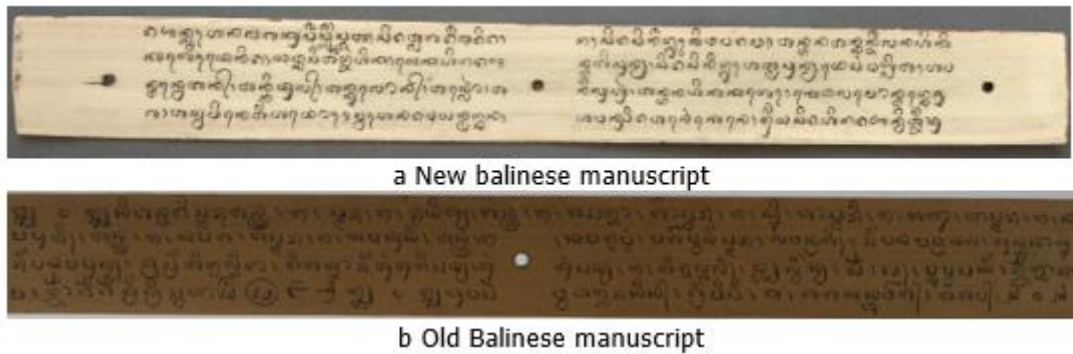
Figure 2. New Balinese manuscript and old Balinese manuscript

Pre-processing

Pre-processing is a part of image processing, it has different Computer Vision on basic techniques, and some authors use both terms interchangeably [4], [5]. Pre-processing is a process that is carried out before the main process is executed, namely the process of contour detection and object filtering. This process has an important role for the next process, in this process image improvement is carried out. The image improvement process is an effort made to improve the input image, one form of image improvement is edge detection. Edge detection is becoming an important part of problems based on computer graphics and image processing. After understanding the steps for preparing an image, the first structure you will find is the edges, or outlines, of each object in an image [6].
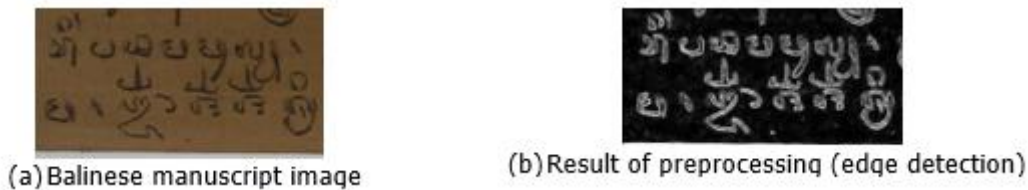

Figure 3. Balinese manuscript image and result of pre-processing (edge detection)

Edge detection is the primary step in identifying an image object, it is very essential to know the advantages and disadvantages of each edge detection filter. Figure 3 a is an example of an image source used for pre-processing, and figure 3 b result of pre-processing was the result of pre-processing used canny edge detection. Canny's edge detection algorithm is costlier in comparison to Sobel, Prewitt, and Robert's operator. Even though, Canny's edge detection algorithm has a better performance [7]. The Canny edge detection algorithm is found to be better in noise and blur conditions but the results are highly dependent on the adjustable parameters [8]. The canny edge detector gives better outcomes related to others with some optimistic points. The recognition is less sensitive to noise, adaptive in nature, and recognizes sharper edges when contrasted with others [9].

Object Detection with Findcontour

Findcontour is a method in opencv/emgu cv library. The application of the opencv library in research [10] has worked with a success rate of 80 to 90%. The working system of object detection is to separate the object from the background, then segment the object using the area approach. Segmentation is a process to partition an image into several parts (regions) or a collection of pixels. The segmentation process divides the image into areas that do not overlap or overlap. In general, image segmentation that is often used uses the intensity method using color and shape approaches.

Contours are used to detect and analyze shapes, to perform various types of recognition, and so on [11]. labeling in the segmentation process is important to limit the object being segmented [12]. The Findcontour method was developed by OpenCV (EmguCv) to detect looking for

objects or two-dimensional images. OpenCV is Open Computer Vision Library, it's launched in 1999 by Intel. In simple terms, contours can be interpreted as a curve that connects points continuously as a barrier to objects that have the same intensity or color. A simple technique is adopted to detect the contours of words in a line of text, the contour is traced in a line of text [13].

The contouring process in this study uses the findcontour method to detect all objects that appear in the Balinese manuscript. The concept of findcontour is to find boundaries (boundary) and map it in the form of coordinates (x,y). It only takes two coordinate points to describe a line as a marker of object boundaries. this method only takes four points as the boundary of the object ar-ea. The boundary illustration is shown in Figure 4.
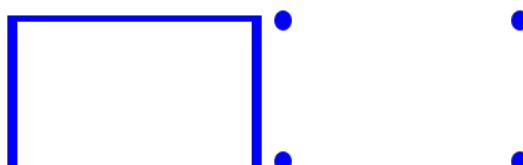
Figure 4. Boundaries object with chain approx none dan chain approx simple

Findcontour provides options on the object area to be constrained, such as cv.CHAIN_AP-PROX_NONE this method will mark the entire object area outside, the entire object area boundary will be saved. While the second picture shows points with cv.CHAIN_APPROX_SIMPLE to find out an object, the object boundaries that are taken are only 4 points as a marker of object boundaries. It only takes two coordinate points to describe a line as an object marker. After the detection process image will be desperate with ROI (region of interest) method. ROI detection has been applied in many fields, for example, medical imaging, security surveillance, databases, and re-mote sensing imaging [14]. ROI on the image is used to mark a part of the input image. ROI placement also plays a significant role in obtaining an effective ROI image retrieval [15].

Results and Discussions

Balinese manuscript image in the Balinese script detection system on Balinese manuscript image has been several image improvements, this process aims to obtain optimal image quality. One of the image improvement techniques to use for this image improvement process is edge detection with Sobel Operator and Canny.

(a) Edge detection with sobel   (b) Edge detection with Canny

Figure 5. Edge detection with sobel and edge detection with canny

The image improvement process by applying parameters is carried out continuously and automatically on the system until optimal detection results are found. Sobel and Canny are edge detection operators that are applied to the image improvement process for the study of Balinese script contour segmentation in Balinese *lontar* manuscripts. For the applied pre-processing method Canny edge detection show the better result (figure 5 b edge detection with canny) than Sobel edge detection (figure 5 an edge detection with sobel).

The contour detection process is a process to map the entire object that is written and appears on the Balinese papyrus script media. This study divides the objects that appear in the Balinese lontar script into three types, namely circle or hole objects, Balinese script objects, and non-Balinese script objects or dirt objects. Circle objects are common objects that appear in Ba-linese lontar manuscripts. A circle object is a hole object that is used to combine the writing on the Balinese lontar script. The whole object in this study is used as a parameter for marking the begin-ning of writing and marking the end of Balinese script writing. Meanwhile, Balinese script

objects and Balinese script objects are the main objects in this segmentation process The test was carried out with two types of data, the first test using *lontar* manuscript data written in less than one month.

The results of the first test image test showed that the number of objects detected was 5807 objects with the number of Balinese script objects being 63 and non-Balinese objects being 5744 objects. Figure 6 is an image capture of the application of the first image tested process. That figure shows some information like the capture of image detection objects from resource images, image capture of ROI process, and some information of image detection process.
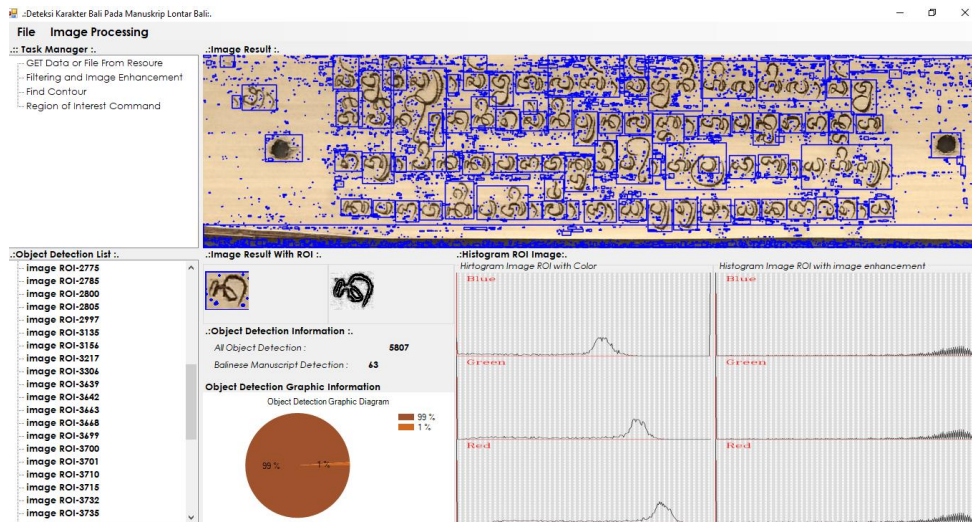

Figure 6. Image result object detection of Balinese script

The first image sample in figure 6 was tested to show that the image has good texture, all object in the *lontar* manuscript was detected as good. As seen in figure 6 at the image result feature, almost all the Balinese object has been detected. For the enhancement process, the application was used edge detection with canny. The blue boundary is the marking of object detection, not only Balinese letter or *aksara*, but the application was detected all object of image resources like noise and hole object. Table 1 shows the data from the Balinese script object detection process.

Table 1. Result of object detection (new manuscript)

No	Description	Total	Percentage
1	All object detected	5807	99%
2	Balinese script object	63	1%
4	Total Balinise script	89	
5	detection result	63/89	70%

The results show that the first stage of testing can detect Balinese script objects up to 70%. The percentage of result detection of the real object, i.e., Balinese letter or *aksara* depends on the object *aksara* has detected. The percentage of result object detection is measured by dividing the real objects with object detection as shown in equation (1).

$$Accuration\ of\ object\ detection = \frac{Balinese\ Scrit\ was\ Detected}{Real\ Balinese\ Script}\ x\ 100 \tag{1}$$

The application has the feature called ROI to extract Balinese object that has been detected and separated from the main image. Object detection list is relust of ROI process, and if data in this feature is chosen, the application will show the object with a detailed histogram report about that object.

The next level of application tested was used with the old manuscript. For this test, the application could detect 897 objects. Figure 7 is the captured image of the old manuscript testing process.
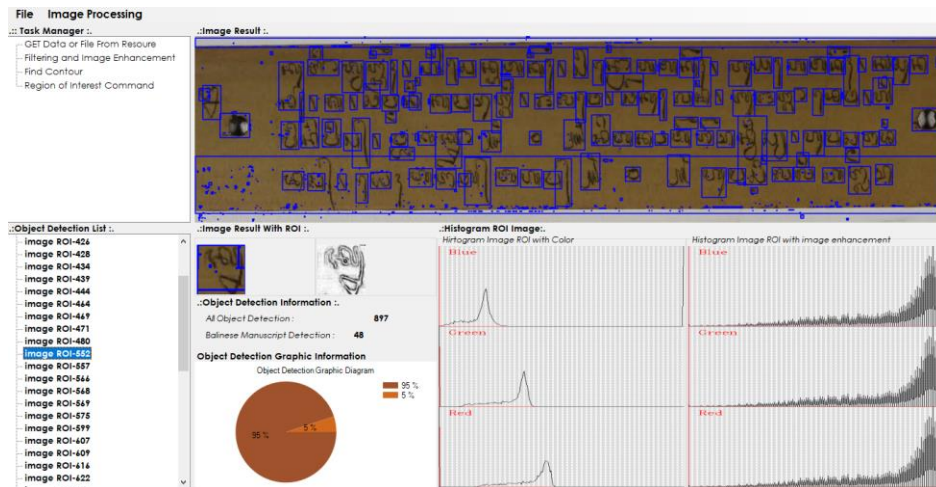

Figure 7. Graphic diagram of the old manuscript test

For this test, the application detects 48 Balinese script objects and 849 non-Balinese script objects. The percentage of detection results for this level reached 30 % for Balinese script objects. The result of this process is shown in Table 2.

Table 2. Result of object detection (old manuscript)

No	Descriptions	Total	Percentage
1	All object detected	897	95%
2	Balinese script object	48	5%
4	Total Balinise script	115	
5	Detection result	48/115	30%

The difference between the two sample kinds of the image has been realized as a result of object detection. The first test has shown 5807 of all object detection and only 63 objects of the Balinese script object. And the second level of testing used the new script. The result of this test has shown 897 objects detected and 115 Balinese script objects. It means the new script has a lot of noise compared to the old script. This is due to the process of lontar making, before using a carving knife to write Balinese script in lontar, lontar was written based on some sketches using pencils.

The new script has a lot of stretch marks compared to the Balinese old script. The Strech mark from the old script is blurry or doesn't have the stretch mark, a lot of noise object was detected in the old script is from the stain or age of the script.

Conclusion

Based on the results of the discussion, it is concluded that the best algorithm in pre-processing is the canny algorithm which is intended to perform image improvements in the form of edge detection on the input image of Balinese script images. The object detection process using findcontour will detect all objects that appear above the Balinese script, whether it is a circle/hole object, a Balinese script object, and an object that is not a Balinese script or a hole object. So to produce Balinese script object segmentation, parameters are needed that are used to separate these objects. With this technique, all Balinese script objects with various shapes and sizes can be detected properly.

Application (Balinese script object detection system) is more accurate in detecting Balinese script objects in Balinese script under 1 year (new script), it tends to be more likely to find

noise/dirt, this is because the writing of the *lontar* using a pencil first before using the knife media, this adds to the noise or dirt detected by the application The findcontour method can detect Balinese script objects with a detection result of 30% - 70% Balinese script objects. Several things that cause the object detection result is not optimal is the grouping of objects that are not optimal is a simple object classification (objects are divided into Balinese script objects, object noise, and hole objects). If there is a Balinese script object that does not have characteristics according to the system, the object is not included as a Balinese script object or noise object. Two Balinese characters can be detected as a single object; this is because the distance between the characters is very close.

Acknowledgments

References

[1]   I. B. R. Putra I.B, "Lontar; Manuskrip perekam peradaban dari Bali," *Jumantara*, vol. 3, no. 1, pp. 148-166, 2012.

[2]   J. Canny, "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679-698, 1989.

[3]   M. Young, *The Technical Writer's Handbook*. USA: University Science Books, 2002.

[4]   M.C. Shin, D. Goldgof, and K. W. Bowyer, "Comparison of edge detector performance through use in an object recognition task, computer vision and image understanding," *Computer Vision and Image Understanding*, vol. 84, no. 1, pp. 160-178, 2001.

[5]   M. Heath, S. Sarkar, T. Sanocki, and K. W. Bowyer, "Comparison of edge detectors: A methodology and initial study, computer vision and image understanding," *Computer Vision and Image Understanding*, vol. 69, no. 1, pp. 38-54, 1998.

[6]   N. Senthilkumaran, R. Rajesh, "Edge detection techniques for image segmentation, international journal of recent trends in engineering," *International Journal of Recent Trends in Engineering*, vol. 1, no. 2, pp 250-253, 2009.

[7]   A. Gharipour and A. W. Liew. "Segmentation of cell nuclei in fluorescence microscopy images: An integrated framework using level set segmentation and touching-cell splitting," *International Journal Of Artificial Intelegence Research*, vol. 2, no. 1, pp. 28-36, 2018.

[8]   S. Chaiprapat and S. Rujikietgumjorn, "Modeling of positional variability of a fixture workpiece due to locating errors," *International Journal of Advanced Manufacturing Technology*, vol. 36, pp. 724–731, 2008.

[9]   A. K. Jain & F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Pattern Recognition*, vol. 24, pp. 1167–1186, 1991.

[10]   D. Dunn, W. E. Higgins, and J. Wakeley, "Texture segmentation using 2-D gabor elementary functions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 2, pp. 130–149, 1994.

[11]   A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filter," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 55-73, 1990.

[12]   S. Himanshu, *Practical Machine Learning and Image Processing*, New York: Apress, 2019.

[13]   S. R. Yhann and T. Y. Young, "Boundary localization in texture segmentation," *IEEE Transactions on Image Processing*, vol. 4, no. 6, pp. 849-856, 1995.

[14]   F. Kurniawan, A. R. Khan, and D. Mohamad, "Contour vs non-contour based word segmentation handwriten text lines: An experimental analysis," *International Jurnal of Digital Content Technology and its Applications*, vol. 3, no. pp. 127-131, 2009.

[15] Open Source Computer Vision Library, Contours in OpenCV. Retrieved from OpenCV website https://docs.opencv.org/, 2020.