Jurnal Ilmu Komputer dan Informasi (Journal of Computer Science and Information)
12/2 (2019), 67-74. DOI: http://dx:doi:org/10:21609/jiki:v12i2:677

CHARACTER IMAGE SEGMENTATION OF JAVANESE SCRIPT USING CONNECTED COMPONENT METHOD

Yuna Sugianela, Nanik Suciati

Informatics Department, Faculty of Information and Comunication Techonology, Sepuluh Nopember Institute of Technology, Sukolilo, Surabaya, 60111, Indonesia

Email: nelaneliyuna@gmail.com, naniksuciati@gmail.com

Abstract

Automation of Javanese script translation is needed to make it easier for people to understand the meaning of ancient Javanese script. By using Javanese script image as input, the translation system generally consists of character segmentation, character recognition, and combining the recognized characters as a meaningful word. The segmentation which obtains region of interest of each character, is an important process in the translation system. In the previous research, segmentation using projection profile method can separate each character well. The method can overcome characters overlapping, but it still produces truncated characters. In this study, we use the modified of the connected component method in the segmentation stage. The propose of this study is to reduce truncated characters and increase accuracy from the previous method. The first step of the proposed method is pre-processing that consists of converting input into binary image and cleaning noises. The next step is to determine the connected component labels, which further perform as candidate of characters. Some of the candidates are still represented by more than one labels, so that we need a process to merge the connected component labels that have centroid distance less than threshold. We evaluate the proposed method using Intersection over Union (IoU). The evaluation shows the best accuracy 93,26%.

Keywords: *Javanese script, image, character, segmentation, component*

Abstrak

Otomasi terjemahan skrip Jawa diperlukan untuk mempermudah orang memahami arti Aksara Jawa kuno. Dengan menggunakan citra dokumen Jawa sebagai input, sistem terjemahan umumnya terdiri dari segmentasi karakter, pengenalan karakter, dan menggabungkan karakter yang dikenali sebagai kata yang bermakna. Segmentasi untuk mendapatkan area karakter, merupakan proses penting dalam sistem penerjemahan. Pada penelitian sebelumnya, segmentasi menggunakan metode profil proyeksi dapat memisahkan setiap karakter dengan baik. Metode ini dapat mengatasi karakter yang tumpang tindih, tetapi masih menghasilkan karakter terpotong. Dalam penelitian ini, kami mengusulkan modifikasi metode *connected component* dalam tahap segmentasi. Penelitian ini bertujuan untuk mengurangi karakter terpotong dan menambah akurasi dari metode sebelumnya. Langkah pertama dari metode yang diusulkan adalah *pre-processing* yang terdiri dari mengubah masukan menjadi citra biner dan membersihkan *noise*. Langkah selanjutnya adalah menentukan label komponen yang terhubung, yang selanjutnya disebut sebagai kandidat karakter. Beberapa kandidat masih diwakili oleh lebih dari satu label, sehingga kita memerlukan proses untuk menggabungkan label komponen yang terhubung yang memiliki jarak *centroid* kurang dari ambang batas. Kami mengevaluasi metode yang diusulkan menggunakan *Intersection over Union* (IoU). Evaluasi menunjukkan akurasi terbaik 93,26%.

Kata Kunci: *Aksara Jawa, citra, karakter, segmentasi, komponen*

1. Introduction

Various intellectual properties about Javanese culture are available in ancient books written using Javanese script [1]. These ancient Javanese books have different content such as religions, linguistics, philosophies, myths, moral lessons, customary laws and norms, kingdoms, folklores, histories, etc. [2]. But not many people learn those books because they do not understand Javanese script [1]. To assist in the translation of Javanese documents, automation of the translation system was carried out. Javanese script document is converted into digital data to be stored as an image and processed according to need [3].

In general, the translation stage consists of segmentation to get characters from the image of Javanese script. Then each character is recognized

as Latin script. And the last is combining Latin writing that has been recognized to be a meaningful word. The Javanese manuscript uses the *scriptio continua* model [3], which means "written continuously" or script writing that does not use spaces or other punctuation.

The alphabet used in Javanese script consists of 20 main character which are syllabic. Javanese script also has *Murda* characters, *Swara* character, *Pasangan* character, *Sandhangan* character, punctuation marks, numbers, and some writing arrangement rules [4], [5].

Segmentation is one of the determinants of success in the automatic image recognition of Javanese script documents [6]. Several studies on the character segmentation of Javanese characters have been automatically carried out. Himamunanto [6] and Widiarti [7] apply the projection profile for line segmentation and character segmentation in Javanese script documents. The method was implemented on 87 pages of Hamong Tani document images which are handwritten documents of Javanese script. The study showed good results with an average percentage of truth of 84.255% [6]. In this study there are still errors in cut-ting the character of Javanese Script, the results of some segmented characters are overlap and some characters become incomplete. Improvement of segmentation results to 90.36% is obtained by applying binarization and removing noise to improve image conditions before segmentation. Besides that, the tilt of the character image on the document is also carried out when character segmentation process and combination with the connected component method to separate overlapping characters [7].

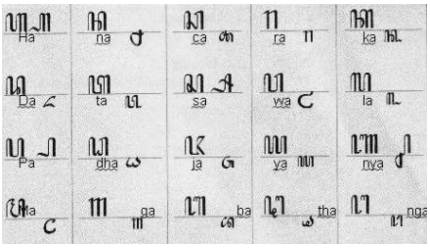
In our experiment, the implementation using projection profile based on [6], [7] has the result 71.8% of segmentation accuracy. There are many errors occurs in overlap line or character.

The connected component method utilizes the pixel neighbor information that is assumed in one letter to be a collection of interconnected pixels. This method can overcome overlapping characters, but there are characters that should be connected are separated by the segmentation stage. Vidyarthi [8] also performs text detection automatically using the Otsu binarization method and connected components. In the next stage, the characters separated by connected components will be combined with attention to distance information that allows one or more connected components labels to be one character.

In this study, segmentation of characters in Javanese script will be done with a combination of Otsu binarization method, connected component, and close labels merging. A character consists of one main syllable (*murdha, swara, angka, pasangan*, and *sandhangan*) or one main syllable with its punctuation (*pasangan, sandhangan*, etc). Close labels merging is useful for correcting characters that should be one but separate because of the connected component method.

2. Characteristic of Javanese Script

The alphabet used in Javanese script consists of 20 main character which are syllabic. This main character is often known as the “hanacaraka” script, which is taken from the five characters that started the sequence of letters, namely Ha-Na-Ca-Ra-Ka. Each principal letter has a pair of letters that function to connect consonant closed syllables with the next syllable. Figure 1.a shows the main character of Javanese script.



(a)

(b)

(c)

Figure 1. Character of Javanese script. (a) Main character and *pasangan* letter (b) *Sandhangan* (c) *Murdha* [4]

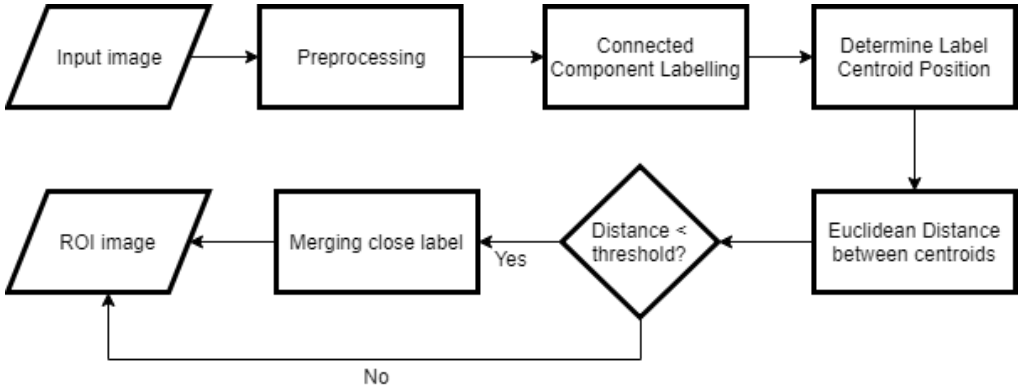


Figure 2.Diagram of proposed method

Figure 1.a also consists of *Pasangan* letter of main characters. Javanese script also has *Murda* characters which are used to initiate the writing of title names, self, geography, government institutions, legal entities, and others. There is also the *Swara* script which is the front vowel. In addition, there are also *Sandhangan*, punctuation marks, numbers, and some writing arrangement rules [4] [5].

3. Proposed Method

In this paper we propose the method to segment characters in Javanese script. Input of this study is image of Javanese script. The input should be converted into binary image and cleaned from noises, this stage is called pre-processing. The next step is to get the connected component label. The last step is merging the close connected component label. Figure 2 is diagram of proposed method.

Preprocessing

Input image is converted to a grayscale image. Then the grayscale image will be binarized. Binarization results are black images (pixel value 0) for background and white (pixel value 1) for foreground or Javanese script writing. The method for binarization process is Otsu Thresholding [9].

This otsu method is used to determine the threshold that minimizes the intraclass variance of the thresholded black and white pixels. The global threshold can be used to convert a grayscale image to a binary image. The stages and formulas used in the otsu method are explained in [9].

The next step is cleaning of binarized image. Cleaning is a process to remove isolated pixels (individual 1s that are surrounded by 0s), such as the center pixel in this pattern [10].

Figure 3 is the example of pre-processing stage. Figure 3.d shows the result of pre-processing stages the cleaning process seems can remove some noises (they are shown by yellow circles in Figure 3.c) from Otsu Thresholding process.

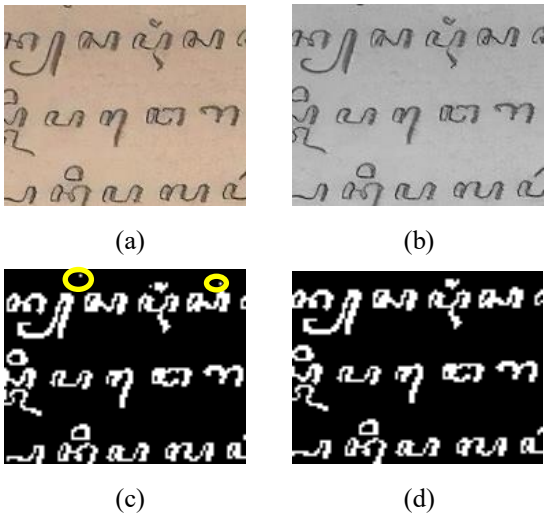


Figure 3. Example of pre-processing (a) input image (b) grayscale image (c) Otsu thresholding (d) cleaning

Get Connected Component Label

In the preprocessed image, we found the 8-neighbors connected component. It is assumed that one letter is a collection of connected pixels.

Connected components are the object points that are connected to each other. Connected components in the pixel matrix are assigned a unique label. These labels indicate the number of object points from the given image [8].

For two-dimensional images, there are two main types of algorithms connected component labeling: algorithms based on label-propagation and algorithms based on label-equivalence-resolving. In this study we use two scan algortihm. Two-scan algorithm [11] is one of the label-equivalent-resolving algorithms. This algorithm consists of two scans.

During the first scan, they assign a temporary label to the object's pixel, and record the label

equivalence. Label equations are completed during or after the first scan. During the second scan, all equivalent labels are replaced by the representative label.

In binary image *N x N*, *b(x, y)* is pixel values on *(x, y)* in the image and V_0 as the value of the object pixel and V_B as the background pixel value. Using the mask shown in Figure 3, all conventional pixel-based raster-scan algorithms scan images in the direction of the raster-scan once to process pixels one at a time.

There is nothing to do if the current pixel *b(x, y)* is the background pixel. If there are no pixel objects inside the mask other than the pixels currently, it is labeled new. It will be labeled as a minimum in the mask, and all different labels in the mask are recorded as equivalent labels.

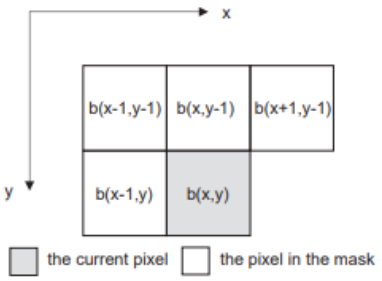


Figure 4. Mask of 8-connected [12]

With the first scan above, the connected component illustrated in Figure 4.a., for example, is temporarily labeled as shown in Figure 4.b, where label 1 and label 4, label 2 and label 5, label 2 and label 3, and label 3 and label 4 are recorded as equivalent labels.

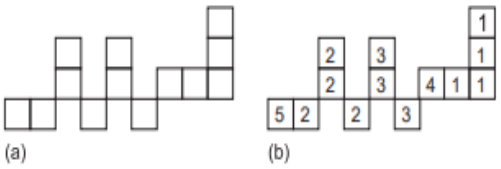


Figure. 5. Illustration of temporary labeling by the first scan. (a) Connection component example (b) Temporary label specified after first scan. [12]

There are several methods for recording and completing label equivalence. The first is to use *L x L* two-dimensional array tables [13]. Another method is an application called a union-find algorithm [14]. The next method is equivalent-label-set [15], [16]. After the label equivalence is completed, the second scan is done by replacing all labels that are equivalent to the representative label [13].

Merging Close Label

The next step is determining the ROI (Region of Interest) of each character according to the label detected. But there are some characters that have more than one label connected component. This causes the possibility of characters being cut when the ROI is determined. To avoid this, a combination of labels is close together.

TABLE 1
DISTANCE OF CENTROID

Result of labelling	Distance
	7.288
	17.82
	21.23

Figure 6 is an example of a Javanese script character that should be connected but has a different connected component label.

Figure 6.Centroid of connected component labelling

The connected component labels that are close together are calculated based on the Euclidean distance [17] between centroids. Centroid is the midpoint of the location of each detected component label. Equation (1) is the formula of Euclidean distance. Point (x, y) and (a, b) are centroid of close labels. Table 1 is the example of distance between two centroids.

$$dist\big((x,y),(a,b)\big) = \sqrt{(x-a)^2 + (y-b)^2} \quad (1)$$

In this study a distance threshold will be determined for combining two or more labels. If the distance between two labels is less than or equal to the threshold, the two labels are labelled as one. Otherwise, if the distance is more than the threshold, then the two labels will not be merged. The next step is determining ROI according to the new label. Figure 7 is an example of the result of merging several connected component labels.

Figure 7. Result of merge close label

4. Experiment Result

In the experiment we will compare the usage of the threshold of pixel disctance between pixels in combining the connected component label at the character segmentation stage. We use 18, 20, 22, 24, 26, and 28 pixels for threshold. To measure the performance of the segmentation method, the Intersection over Union (IoU) method is used.
IoU [18] is an evaluation metric. Any algorithm ROI or uses a boundary box as output can be evaluated using IoU. In applying IoU, it takes ground truth ROI from the dataset image and predicted ROI from the model created. IoU calculations can be determined in Figure 8.

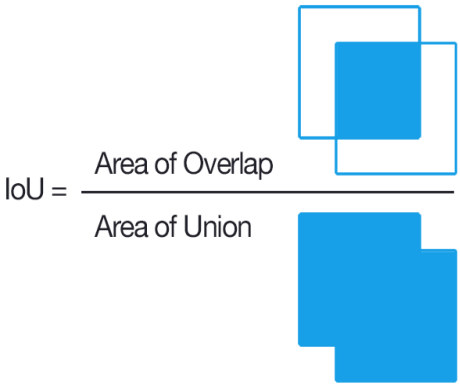


Figure 8. Formula to get IoU [17]

Dataset

To evaluate the proposed method, we use a scanned Javanese Script. The dataset is from Javanese script with the title "*BLOEMLEZING UIT JAVAANSCHE WERKEN (PROZA)*" which was published in 1942. The distance between lines is neat and consistent. But for each character looks less consistent and there are overlaps. Figure 9 is Javanese script used.

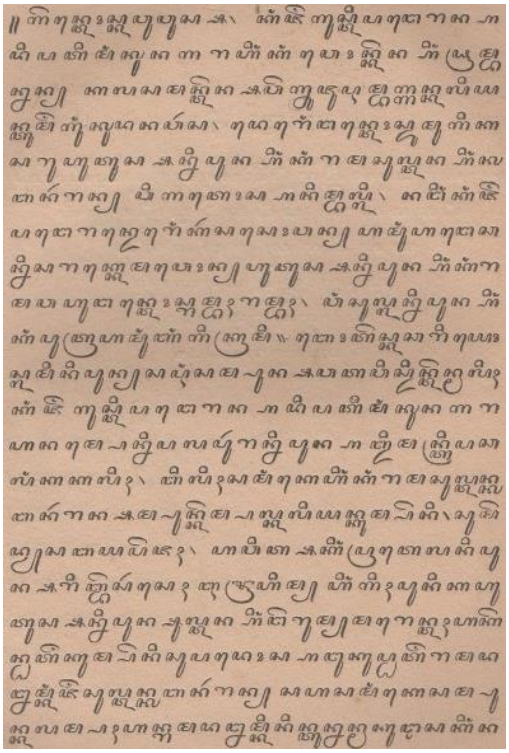


Figure 9. Image of Javanese script document

Result and Analysis

We create the ground truth data from dataset image In a page of dataset image, we get 413 components that each components consist of one or more Javanese character (only main characte, or main with *sandhangan*, or main with *pasangan*, etc). The component consists of one character or more because we avoid the error in segmentation because there are many various size in Javanese characters and distance between them.

Figure 10 is the example of ground truth ROI. Figure 10.a only consists of one character. The example of ROI that consists of one character,

a. Main character (ha-na-ca-ra-ka) without pasangan or sandhangan
b. Pasangan, sandhangan, murdha, or Swara that whose position is in a line with the main letter.

Then in Figure 10.b there are two characters in the ROI. The example of ROI that consists of two characters,

a. Main character with pasangan or sandhangan that whose position is below or above main character
b. Pasangan, that whose position is in a line with the main letter, with sandhangan that whose position is below or above main character.
c. Murdha, that whose position is in a line with the main letter, with sandhangan or pasangan that

whose position is below or above main character.

Figure 10.c consists of three characters. The example of ROI that consists of three characters,

a. Main character, pasangan, murdha with two sandhangan above its position or above and below its position.
b. Main character or murdha with pasangan and sandhangan.

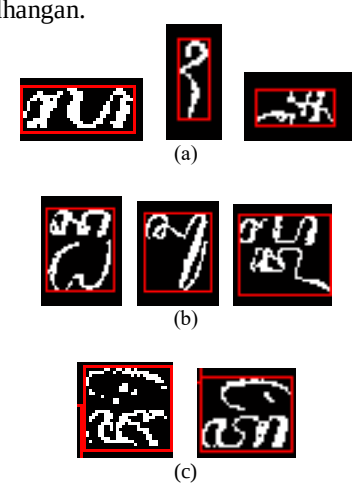


Figure 10. Example of ground truth ROI (a) one character (b) two characters (c) three characters

Figure 11 is the example of IoU calculation of our experiment. We calculate IoU of all ROIs from each threshold experiment compared to 431 ROIs from ground truth. The IoU of each ROI from experiment result will be computed with ROI from ground truth that has the closest position in the script. If ROI of result has full overlap area with ROI of ground truth, the value of IoU is 1 or perfect IoU area. Then we compute the average of accuracy of IoU of all ROIs in a script.

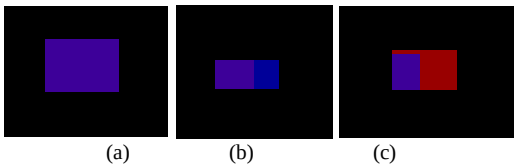


Figure 11. IoU Calculation (blue is ground truth area, red is result area, and purple is intersection of ground truth and result area) (a) excellent intersect area (b) ground truth area is wider than the result area (c) result area is wider than the ground truth area

Table 2 is the average of IoU accuracy of our experiment. Accuracy is the result of IoU computation that is multiplied by 100%. We compare the usage of 6 different thresholds. From the table we know that 26 is the best threshold for merge close label of connected component.

TABLE 2
RESULT OF EXPERIMENT

Threshold (pixels)	Accuration of IoU (%)	Total ROI Result
18	77.2862	501
20	84.2401	473
22	87.6887	457
24	90.8936	441
26	93.2566	426
28	90.3666	498

Figure 12 is the example of segmentation result. From the figure we know that in threshold 18 pixels there are many error cutting characters because some labels should be one character (not corresponding with the ground truth) are separated into two, three, or more areas, it is shown by yellow circle in Figure 12. While for the threshold that is too high, like 28 pixels, there are labels from other characters that join the wrong character, it is shown by yellow circle in Figure 12.g. The best result for an image script is 26 pixels threshold.

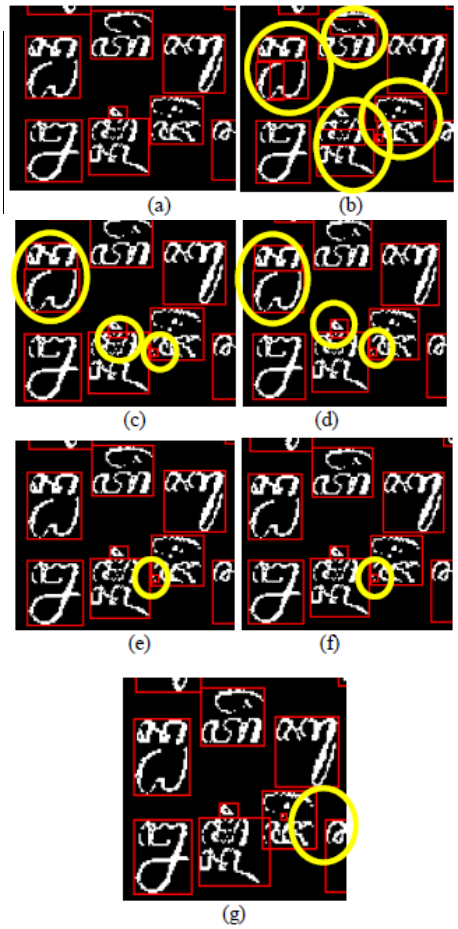


Figure 12. ROI from character segmentation (a) Ground Truth (b) Result of threshold 18 pixels (c) Result of threshold 20 pixels (d) Result of threshold 22 pixels (e) Result of threshold 24 pixels (f) Result of threshold 26 pixels (g) Result of threshold 28 pixels

Mistake that often occurs in this study is there are sandhangan whose position on the right side until below the main letter. The quality of the original image and the not good quality of pre-processing results cause the sandhangan does not have the same label connected component, so that when merging near the centroid (the sandhangan itself and the main letters around it) there will be an error. Figure 13 is the example of the often occurs mistake in experiment.

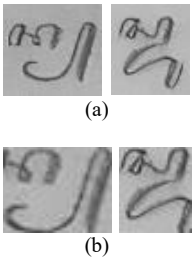


Figure 13. Example of mistake ROI (a) Ground truth (b) Experiment Result

5. Conclusion

In this study, we proposed a new segmentation of characters in Javanese script document image. The input is image of Javanese script document. The input should be converted into binary image and cleaned from noises, this step is called pre-processing. The next step is to get the connected component label that perform the candidate of a character. The last step is merging the close connected component label. Close labels merging is useful for correcting characters that should be one but separate because of the connected component method.

The experiment is evaluated using Intersection over Union (IoU) accuracy. The best threshold from experiment result is 26 pixels, it performs 93,26% accuracy of IoU.

This method successfully resolves the wrong character cutting problem from the previous method that has the result 71.8% of accuracy. This segmentation method can be developed for feature extraction and automatic recognition using classification. For further research it can be corrected to combine the connected component label automatically using the clustering method. Because the threshold distance used still causes errors in some characters.

References

[1] A. R. Widiarti and P. N. Wastu, "Javanese Character Recognition Using Hidden Markov Model," *World Acad. Sci. Eng. Technol.*, vol. 3, no. 9, pp. 2201–2204, 2009.

[2] M. Soleh, "Handwritten Javanese Character Recognition using Descriminative Deep Learning Technique," *Int. Conf. Inf. Technol. Inf. Syst. Electr. Eng.*, pp. 325–330, 2017.

[3] A. R. Widiarti, "The Model and Implementation of Javanese Script Image Transliteration," *Int. Conf. Soft Comput. Intell. Syst. Inf. Technol.*, 2017.

[4] H. Hardjawijana, "Pedoman Penulisan Aksara Jawa." Yayasan Pustaka Utama, Yogyakarta, 2002.

[5] A. M. Sulaiman, "HANACARAKA: Aksara Jawa yang Mulai Ditinggalkan," no. August, 2011.

[6] A. R. Himamunanto and A. R. Widiarti, "Javanese Character Image Segmentation of Document Image of Hamong Tani," *Digit. Herit. Int. Congr.*, pp. 641–644, 2013.

[7] A. R. Widiarti, A. Harjoko, and S. Hartati, "Preprocessing Model of Manuscripts in Javanese Characters," *J. Signal Inf. Process.*, no. November, pp. 112–122, 2014.

[8] A. Vidyarthi, N. Mittal, and A. Kansal, "Text and Non-Text Region Identification Using Texture and Connected Components," *Int. Conf. Signal Propag. Comput. Technol.*, pp. 604–609, 2014.

[9] P. Smith, D. B. Reid, C. Environment, L. Palo, P. Alto, and P. L. Smith, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. Syst. Man. Cybern.*, vol. 20, no. 1, pp. 62–66, 1979.

[10] MathWorks, "Morphological operations on binary images." [Online]. Available: Morphological operations on binary images. [Accessed: 01-May-2018].

[11] R. Haralick and L. Shapiro, *Computer and Robot Vision Volume 1*. 1992.

[12] L. He, Y. Chao, K. Suzuki, and K. Wu, "Fast connected-component labeling," *Pattern Recognit.*, vol. 42, pp. 1977–1987, 2009.

[13] A. Rosendfeld and J. . Pfalts, *Sequential Operations in Digital Picture Processing . Journal of the*, vol. 13. 1966.

[14] K. Wu, E. Otoo, A. Shoshani, and L. Berkeley, "Optimizing Connected Component Labeling Algorithms," *Pattern Anal.*, 2008.

[15] L. He, Y. Chao, and K. Suzuki, "A Linear-Time Two-Scan Labeling Algorithm," *IEEE Int. Conf. Image Process.*, pp. 241–244, 2007.

[16] F. Chang, C. Chen, and C. Lu, "A Linear-

Time Component-Labeling Algorithm Using Contour Tracing Technique," *Comput. Vis. Image Underst.*, vol. 93, no. 2, pp. 206–220, 2003.

[17] Advanced Project R&D, *Euclidean Distance*. 2005.

[18] A. Rosebrock, "Intersection over Union (IoU) for object detection - PyImageSearch," 2016. [Online]. Available: https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/. [Accessed: 18-Oct-2018].