THE DETECTION AND COUNTING OF OBJECT BOTTLES IN THE BOXS BASED ON IMAGE PROCESSING USING WATERSHED ALGORITHM

[1] CHAERUR ROZIKIN, [2] MOHAMAD IQBAL SURIANSYAH, [3]ARIES SUHARSO, [4] KUSNADI, [5] MUHAMMAD FAJAR ESTU NUGROHO, [6]NURLANA SANJAYA

[1, 3, 4, 5, 6]Universitas Singaperbangsa Karawang, Faculty of Computer Science, West Java, Indonesia
[2]Pakuan University, Faculty of Natural Science and Mathematics, West Java, Indonesia
E-mail: [1]chaerur.rozikin@staff.unsika.ac.id, [2]mohamad.iqbal@unpak.ac.id

ABSTRACT

Quality control using image processing and machine vision offers faster performance and high accuracy, with minimum human contact. A significant application refers to bottle counting in a box-based pattern to deliver appropriate quantity, according to customers' demand. The precise arrangement with very close proximity to each other poses a significant challenge. As a result, missed detection tends to occur during segmentation due to interconnected boundaries and contacts. The watershed algorithm provides a possible solution with the capacity to segregate touching and overlap. A particular approach to bottle detection and counting used in this research. The results showed the average detection accuracy for 9-12 bottles at 99.5726%. However, with mean values for memory usage on every detection of 0.6 MiB and time interval on every 3 boxes detection within 0.21 secs, the use of the watershed algorithm was strongly believed as a suitable alternative.

Keywords: *Watershed Algorithm, Object Counting, Image Segmentation*

1. INTRODUCTION

Workforce performance appears as a critical factor in measuring the continuous production of industrial goods concerning the targeted output. A decline beneath or above this threshold possibly instigates a corresponding decrease in productivity[1]. However, human inspection tends also to impose several obstacles, including the high dependency on assessment results from the examiner's perspective, based on the ability to comprehend the object and personal knowledge. This process occasionally becomes very dull, and therefore, the test quality is assumed to diminish [2].

Under these conditions, inspection and quality control processes become very significant, specifically during production, to ensure optimum results according to consumers' demand [3]. In addition, various strategies to achieve these objectives are thoughtfully considered to reduce operational costs [4].

In informatics, machine vision is described as a science known to offer numerous methods in implementing quality and process control measures as well as mobility guidance for digital image-based robots. This concept is applied mainly in spare part inspections, product performance, and completion [5]. Figure 1 represents the quality control process with machine vision and a camera placed on a conveyor. Machine vision provides a less-contact application and functions effectively to deliver fast output without errors [6]. In this research, quality control covers box-based bottle counting to ensure smooth delivery of goods to consumers based on required
quantity.

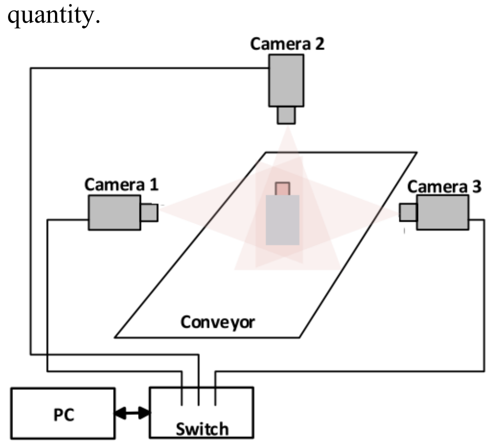


Figure 1: General Architecture of Machine Vision For Quality Control

Digital image segmentation is essential in machine vision systems to determine the objects to count. In [7], the inclusion of thresholding confirmed the background separation of the egg object, as the computer was able to determine the location with a different label in a digital image. However, the major challenge errored during object detection in similar and close proximity or connected boundaries [8]. Furthermore, the thresholding method does not guarantee the absence of holes and wrong pixel additions in the resulting segmentation [9]. The bottles in this research appear closely and directly in contact with each other, and therefore, the application of the thresholding approach in detection and counting possibly generates errors.

The watershed process is employed in anticipating object contact and intersection, as the algorithm represents the image as topographic surface condition, where each element value describes the structural height [10]. Also, the image is visualized as not flat but saturates the bottom plain or catchment basin with water. This tends to subsequently fill up the threshold, and the peak areas separating each catchment basin after the process are referred to as watersheds [11]. The watershed algorithm shows the capacity to segment the image in order to determine associated objects [12]. In addition, an easier separation between adjacent leucocytes and erythrocytes in the microscopic appearance of blood is observed. Another instance is the segmentation and detection of ripe tomatoes [13]. In this research, the use of the watershed algorithm succeeded in segmenting the tomato fruit image on the tree with proximity to each other and further causing an overlap, with appropriate detection accuracy up to 81.66%.

This paper aims to propose a new method to do quantity control in the production line using a watershed algorithm and show the accuracy watershed algorithm can do bottle in the boxes detection and counting. The expected result of this paper would be how a good watershed algorithm can perform bottle detection and counting.

2. LITERATUR REVIEW

Quality control is a method used in the manufacturing process, from the beginning to the end. Quality control ensures that the goods manufactured are guaranteed and thoroughly excellent in response to the requirements established and desired by customers[14].

An analog image is a description of the object. Because it is analog, an analog image cannot be processed by a computer, so that an analog image converted into an image[15]. Images are represented by a $M * n$ sized matrix where the intersection of the matrix lines and columns is called a pixel. Each pixel has parameters in the form of coordinates and color intensity. So that images can be written in the form of a matrix:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,M-1) \\ f(1,0) & \dots & \dots & f(1,M-1) \\ \dots & \dots & \dots & \dots \\ f(N-1,0) & f(N-1,1) & \dots & f(N-1,M-1) \end{bmatrix} \quad (1)$$

From the above equation, x dan y is a collection of rows and columns as a coordinate position of pixels and $f(x,y)$ is a function that represents the intensity of color in color in those pixels[15]. Digital image processing is a digital image processing process using a digital computer using a specific algorithm[16].

Machine visions are the methods and approaches used to provide image-based examinations used in quality control, process control, and vision in robots[5]. Object counting is one application of machine vision used in the quality control process. The calculation of objects is a problem that often occurs in the application process in machine vision[6]. Feature extraction is a process commonly used in machine learning and digital image processing. Many varied data are selected to be more informative and avoid redundancy to be processed and used further. Feature extraction can also be done to make it easier for humans to interpret data[17]. Examples of feature extraction in digital image processing are edge detection, image segmentation, and thresholding. Segmentation in digital images divides the image into several areas to have the same visual characteristics. The goal is to simplify and narrow down the scope to be analyzed.

Image segmentation is the pre-processing stage of digital image processing commonly used in image pattern classification, image visualization, and image compression[10]. ROI in digital image processing is the area in the image selected for further processing. ROI has many functions, including simplifying image data processing to compressing the image into smaller pieces. ROI generated from the image segmentation process[18]. Thresholding is the simplest segmentation method. A grayscale image is separated into two parts, black and white, or called a binary image, where the white part is considered 0 and the black part is deemed to be 1. Each pixel is positioned according to the input value of the thresholding[19]. There are several thresholding

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

methods, and each method gives different results in segmenting the image into a binary image.

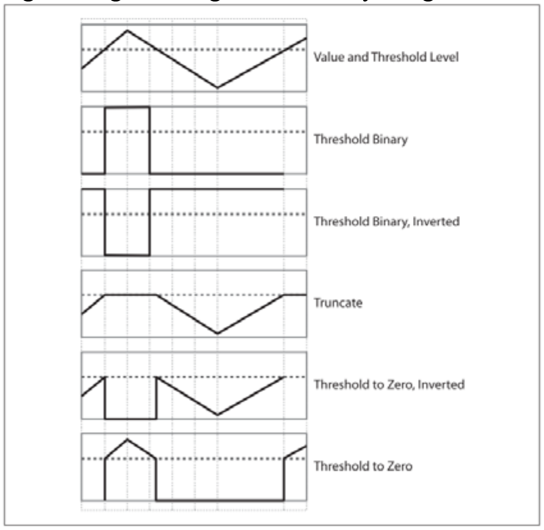


Figure 2: Thresholding Results on Different Method in OpenCV Libraries[20]

The above shows several methods available in the OpenCV library for thresholding. Each method treats pixels that are below the threshold as either 0 or 1. Typically, the usual threshold method places pixels whose degrees of gray are below the threshold placed at 0 and those above the threshold assigned to the maximum value. Confusion matrix or error matrix is a machine learning and digital image processing analysis tool[13] that contains information about the system's actual classification and classification results. The confusion matrix has two dimensions, one dimension includes the existing classification index, and the other dimension contains the index classification results by the system[21].

		Predicted		
		A_1	... A_j ...	A_n
Actual	A_1	N_{11}	N_{1j}	N_{1n}
	⋮ A_i ⋮	N_{i1}	⋮ ... N_{ij} ... ⋮	N_{in}
	A_n	N_{n1}	N_{nj}	N_{nn}

Figure 3: Concept of Confusion Matrix[21]

The performance of classification or detection can be determined using a confusion matrix. Some of the metrics used in the confusion matrix as the basis for calculating the algorithm performance measurement are as follows:

- True Positive
  There is a total number of detections that accurately detect positive detection.
  $$tp_i = N_{ii} \quad (2)$$
- True Negative
  The number of detections for which the algorithm correctly detects a negative class as a negative class.
  $$fp_i = \sum_{l=1}^{n} N_{li} - tp_i \quad (3)$$
- False Positive
  Represents the number of detections where the algorithm incorrectly predicts a negative detection as positive.
  $$fn_i = \sum_{l=1}^{n} N_{il} - tp_i \quad (4)$$
- False Negative
  Represents the number of detections where the algorithm incorrectly predicts a positive detection as a negative detection.
  $$tn_i = \sum_{l=1}^{n} \sum_{k=1}^{n} N_{lk} - tp_i - fn_i - fp_i \quad (5)$$

Watershed was one of the main areas of study in the field of topography. Apart from a lot of topography, the watershed algorithm has proven beneficial and effective in various other applications, such as image segmentation[11].

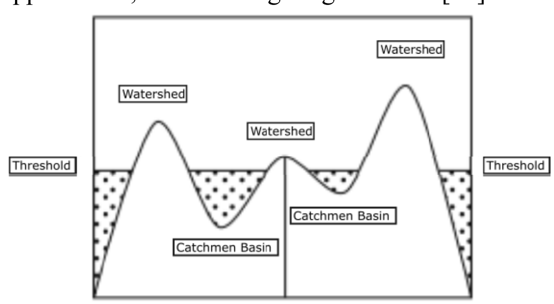


Figure 4: Representation of Watershed Algorithm

The Watershed Transform principle based on seeing a picture in three dimensions: two spatial coordinates vs. gray levels in a "topographic" interpretation[22]. This method performs better on the segmentation of touching and overlapping objects[8].

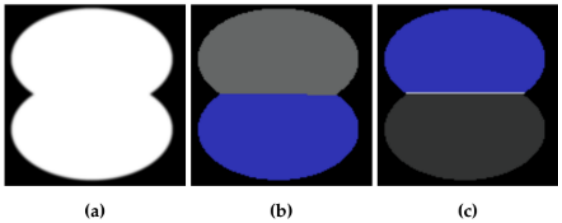


Figure 5: Different Segmentation Results[10]

Figure 5 shows different segmentation results for touching objects between segmentation without watershed(b) and using watershed(c). It shows how the watershed algorithm clearly shows the boundary between two things.

3. RELATED WORKS

In 2018, [12] applied a watershed algorithm to separate leukocyte and erythrocyte in the microscopic image of blood. The watershed algorithm gave 0.32 seconds in the segmentation process, and segmentation accuracy for leukocyte and erythrocyte is 97.2% and 94.8%. We know that components inside microscopic images are too complex to be segmented. Cells are often touching each other or even overlapping, giving the segmentation process a challenge. By using a watershed algorithm, segmentation between those components inside blood microscopic images is easily segmented.

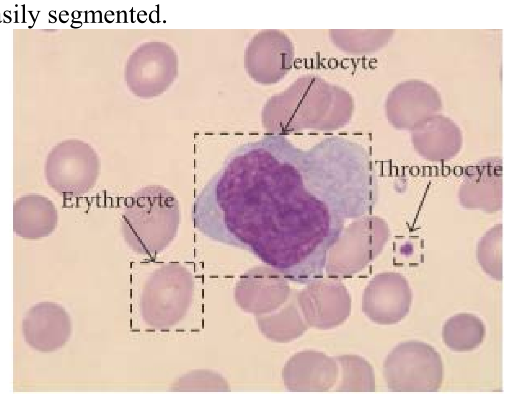


Figure 6: Sample Blood Image[12]

In [13], a watershed algorithm was used to segmentation ripe tomato fruit and locate tomatoes in a tree using an image. [13] combined watershed algorithm and color HSV analysis to separate between each fruit and separate fruit with leaves. The result was measured using a confusion matrix and reaches 81.66% of accuracy.

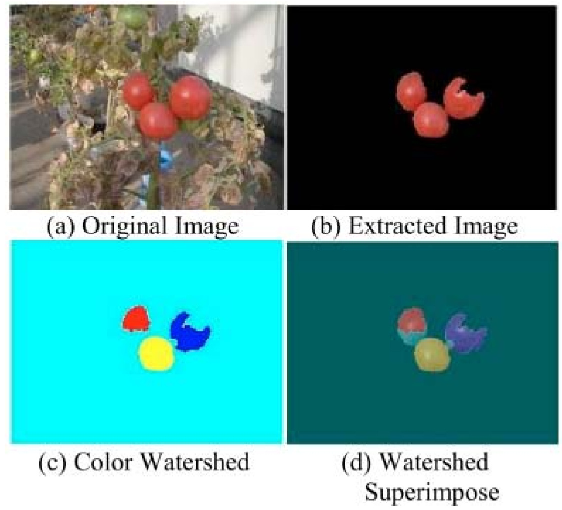


Figure 7: Watershed in Detecting Mature Tomatoes

Using a watershed algorithm, [23] created an android application to count the number of bacteria inside a colony of bacteria. The challenges were to count touching and overlapping bacteria on the image that can not be easily be solved using thresholding-based segmentation. Using a watershed algorithm, the researcher managed to get 100% accuracy on counting numbers of bacteria. The problem was quite similar to [23], [24] segmented rice grains by image using a watershed algorithm. The image used also consists of touching and overlapping rice grains. The average accuracy using proposed method reached 94.36%.

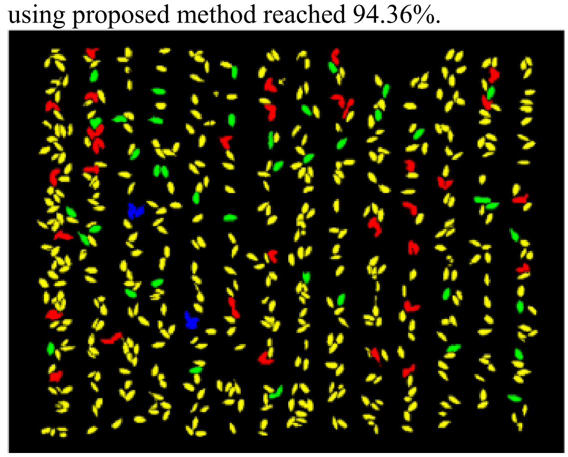


Figure 8: Segmentation Result on Rice Grains[24]

In the bottles counting problem, [6] has explained their proposed method using canny edge and hough circle transform method. Since the bottle cap has the shape of a circle, hough circle transform quickly detects bottlecaps. Unfortunately, hough transforms primary step is edge detection means that using hough transform on very noisy images can increase the complexity of algorithm and resource usage.

Journal of Theoretical and Applied Information Technology
15th June 2021. Vol.99. No 11

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

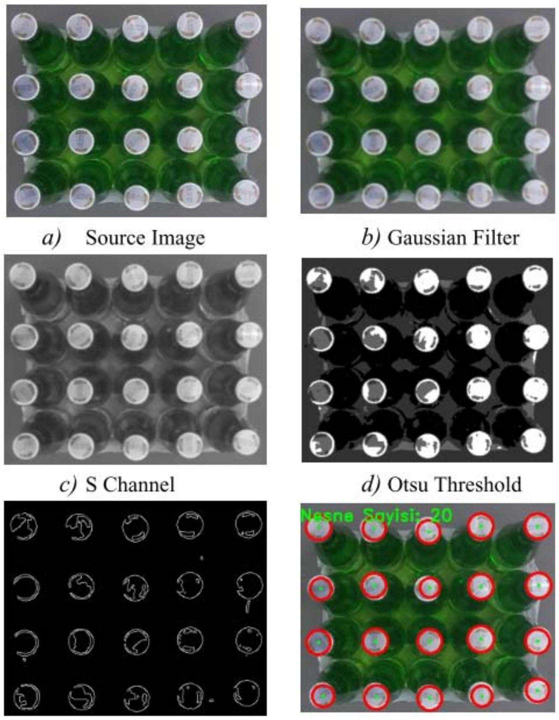


Figure 9: The Use of Hough Circle Transform in Quantity Control

The above shows are different of research on object detection and counting. They differ on each other from research objectives to the method used. Researchers studied counting leukocyte and erythrocyte, bacteria colony, mature tomato fruits, and rice grains using watershed algorithms to prove that watershed can be used in real-time object detection and depending. In contrast, the use of computer vision to quantity control of bottles was done by only using hough transform.

4. RESEARCH OBJECTS AND METHODOLOGY

The object in this research referred to the ability of a watershed algorithm to detect and calculate the bottle quantity. However, the box was positioned on the platform without necessary facilities, as represented in Figure 2. This portion was covered with black cloth to indicate the conveyor. There are 3 boxes in the video frame with a distance of + - 5 - 10 cm, and the number of each bottle ranged between 9 – 12, by random evaluation. Furthermore, the bottle cap was shaped in circular form, with a light blue appearance.

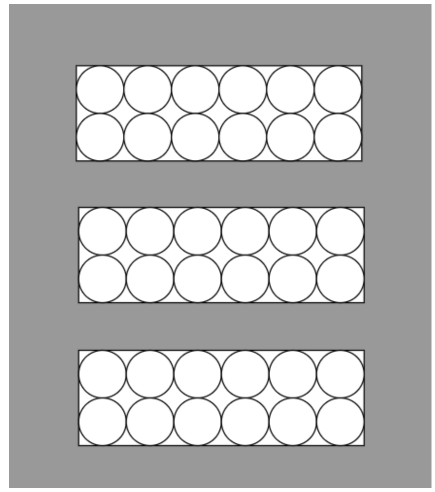


Figure 10: The Elements in Each Frame

The camera is used for recording 720p resolution video in a static position with a distance of 170cm from the platform. This video frame comprises three primary elements, including the background in cardboard bases, boxes, and bottles. Figure 12 provides a comprehensive description of the process.

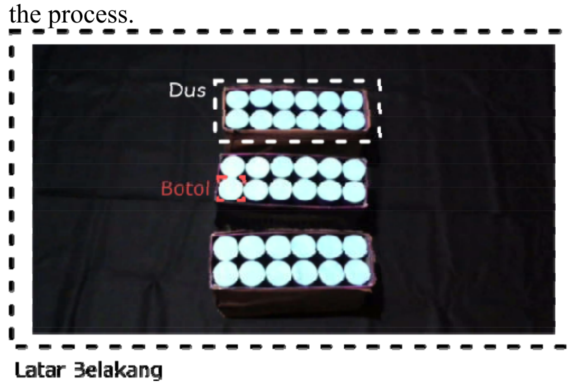


Figure 11: Position of Boxes on The Platform

Figure 11 shows the division of each frame into two stages, termed find contour and watershed. Find contour determines the location and area of each box in the video frame. The result is describing as a region of interest, ROI, and the image selected for the next phase. Furthermore, ROI is known to perform several functions, including simplifying image data processing to more minor compressions. Also, ROI generated from image segmentation [18], where the bottle quantity was calculated, mainly using Python programming language [25], [26] Open CV [19], [27], and Scikit-image[28].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

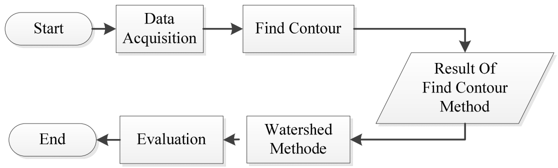


Figure 12: Overall Program Flowchart

Confusion matrix (CM), also known as error matrix, is described as a machine learning analysis and digital image processing tool [13], comprising two dimensions, including actual classification index and system results [21], assumed as the accuracy benchmark for a watershed algorithm.

4.1. FIND CONTOUR

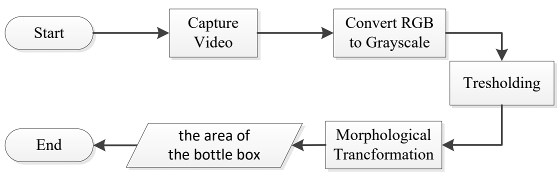


Figure 13: Box Detection Flowchart

There is a need to determine each box location in the video frame during counting. Figure 5 shows this research employed an approach with the find contour algorithm to evaluate the box position and area. Find contour used to select the image outline from the binary image [29], obtained from thresholding, known as the simplest segmentation method, where the grayscale image was separated into two portions, black and white, or called a binary image as the white and black segments were considered 0 and 1, respectively. Moreover, each pixel is positioned according to the input value[19].

Thresholding demands a specific boundary estimate to define the box section. Furthermore, a histogram analysis is required to observe regions with more frame appearance. Figure 6 reveals extensive pixels with a value of 44 and below, although with sufficient black background. It is possible that the pixels with a value of 44 and down served as the background, and above 44 were the box segments.

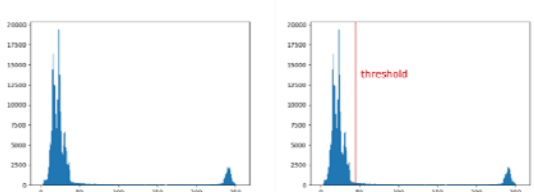


Figure 14: Histogram Analysis of Video Frames

Furthermore, thresholding was performed, using the 44 as the limit to generate a frame with a binary image. The results often indicated imperfections in the pixels due to noise, image texture, and inaccuracies (Figure 15).

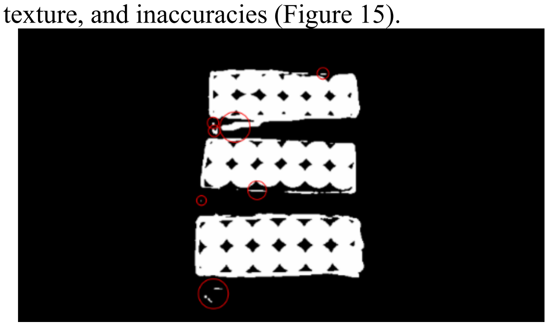


Figure 15: Noise on Thresholding Results

Morphological transformation is known to minimize the thresholding process imperfections and apply to a binary or grayscale image[30]. Figure 16 represents the improvement in previous results.


Figure 16: The Process Of Improving Images With Morphological Transformation

However, by using cv.findContours() function from OpenCV, the contour location and box sections inside the video frame become possibly determined.

4.2. WATERSHED

The result of the find contour showed the box location and area. Figure 18 depicts the watershed algorithm application to individual box samples.

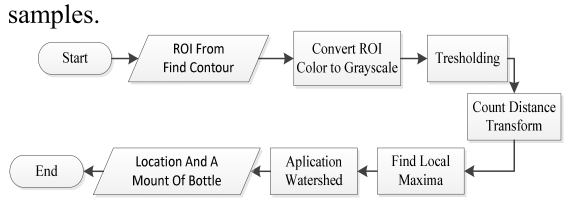


Figure 17: Flowchart of Bottle Detection with Watershed

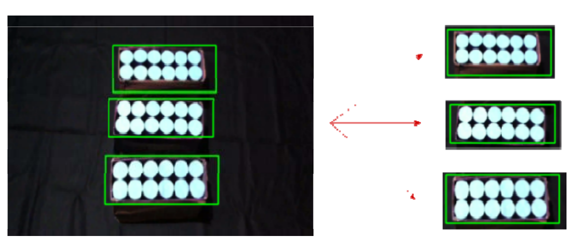
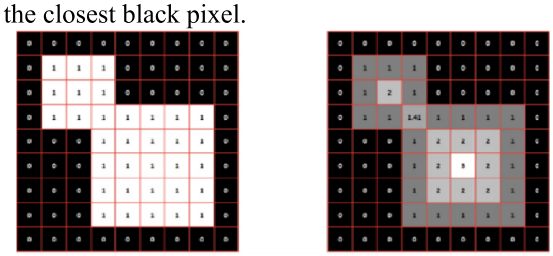


Figure 18: ROI from the FindContour

Figure 17 indicates the initial watershed algorithm as a thresholding process in ascertaining background and foreground areas. A distance transform matrix was generated based on these conditions, in the form of pixels with intensities of 0 and 1 for dark and bright segments, respectively. Therefore, distance transform serves as an operator representing the distance of one pixel from another, with an intensity of 0 [31]. The formula for evaluating this parameter is as follows:

$$d(p_1, p_2) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (6)$$

Unlike the threshold result matrix, distance transform was not specified as 1 and 0 but corresponded to the distance between the pixel and the closest black pixel.



a b

Figure 19: Threshold Matrix and Distance Transform Matrix Differences

The segmentation areas were the peaks of the topographic structure, based on a watershed algorithm. Moreover, the distance transform result showed the pixels were further apart from the closest black counterpart, indicating higher values. These points were the peaks, and the determination referred to as local maxima (Figure 20) [31]. Furthermore, with the peak_local_max function from the Scikit-Image library, each vertex coordinate was observed in an array.

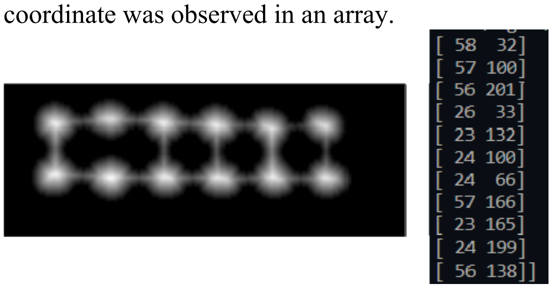


Figure 20: The Local Maxima Determination of The Distance Transform Matrix

Subsequently, each vertex was labeled and determined as the center point of the contour for the detected bottle cap (Figure 21). The results were further assessed for accuracy.

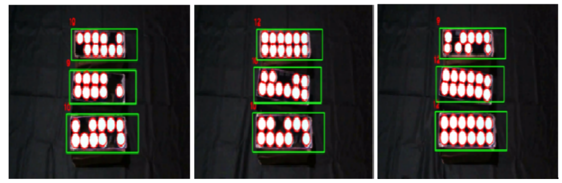


Figure 21: Segmentation Results of Watershed

4.3. EVALUATION

The confusion matrix evaluation model with multiple classes was applied. Before this process, a test case prepared, using 39 video data and 117 boxes, where each contained bottles numbering between 9-12. Table 1 represents the random arrangements determined by the RANDBETWEEN function in Microsoft Excel software.

Table 1: Test Case Example

No	Actual Number Of Boxes To -			Watershed Detection Results		
	1	2	3	1	2	3
1	12	11	9			
2	9	10	11			
3	10	9	10			
4	12	12	9			
5	12	12	12			
6	9	9	10			

The program was executed, and the detection results were recorded in the previously generated table. These data are converted into a confusion matrix table (Table 2).

Table 2: Confusion Matrix Test Result

		Detection			
		9	10	11	12
Actual Total	9	28	0	0	0
	10	0	28	1	0
	11	0	0	31	0
	12	0	0	0	29

Based on the confusion matrix, the detection accuracy, recall, F-1 Score, and precision of each class. Table 3 shows the impact of the accuracy, precision, recall, and F-1 Score. Below are the formula to get each accuracy, recall, F-1 Score, and precision of each class.

$$Accuracy = \sum_{i=1}^{n} N_{ii} / \sum_{i=1}^{n} \sum_{j=1}^{n} N_{ij} \quad (7)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$Precision_i = N_{ii} / \sum_{k=1}^{n} N_{ki} \quad (8)$$

$$Recall_i = N_{ii} \,/\, \sum_{k=1}^{n} N_{ik} \quad (9)$$

$$F-score = \frac{2 \times Precision_i \times Recall_i}{Precision_i + Recall_i} \quad (10)$$

Table 3: Results of Accuracy, Precision, Recall, And F-1 Score.

Number of Bottles					
9	TP	28		Precision	1
	TN	89		*Recall*	1
	FP	0		*F1-Score*	1
	FN	0		Accuracy	1
10	TP	28		Precision	1
	TN	88		*Recall*	0,965517
	FP	0		*F1-Score*	0,982456
	FN	1		Accuracy	0,991453
11	TP	31		Precision	0,96875
	TN	85		*Recall*	1
	FP	1		*F1-Score*	0,984127
	FN	0		Accuracy	0,991453
12	TP	29		Precision	1
	TN	88		*Recall*	1
	FP	0		*F1-Score*	1
	FN	0		Accuracy	1

Table 3 shows the accuracy, precision, recall, and F-1 Score of each number of bottles. First, we count each true positive, true negative, false positive, and false negative using a formula used in a theoretical framework and using those data. We can estimate accuracy, precision, recall, and F-1 Score.

Table 4 represents the conclusion data for the calculation results.

Table 4: Conclusion of Calculation Results

Total	*Precision*	*Recall*	*F1-Score*	*Accuracy*
9	1	1	1	1
10	1	0,965517	0,982456	0,991453
11	0,9688	1	0,984127	0,991453
12	1	1	1	1
AVG	0,9922	0,991379	0,991646	0,995726

5. DISCUSSION

Inbox bottle counting plays an essential role in the quality control process, particularly in the bottled beverage industry, and ensures adequate supply to consumers in the appropriate quantity. Close arrangement in bottle boxes with adjacent edges, simple image segmentation, including thresholding, does not correctly separate these connected objects. Therefore, a watershed algorithm was employed to provide a suitable alternative due to its high separation ability. This study tends to apply a watershed algorithm to inbox bottle counting to determine the system accuracy, including in terms of detection.

5.1. Accuracy Evaluation

The data comprised 39 videos, each containing three bottle boxes. In addition, the program developed using the watershed algorithm to count the number of bottles in the individual compartment. Accuracy was measured using the confusion matrix and gave 99.5726% accuracy.

5.2. Resource Usage

The resource used to perform detection was 6Gb of DDR4 memory, and AMD Ryzen 3 CPU gave an average detection time of 0.2112 secs. Using memory profiler Python library, the average memory used to run the program was 0.6 Mb. These results reflected an accuracy extending to 99.5726% and were assumed to be very fast in detection and calculation. Moreover, the use of minimum computer resources confirmed the watershed algorithm was applied correctly.

Table 5: Average Time Spent on Each Detection

Frame	Time(s)
1	0,640998
2	0,127997
3	0,102999
4	0,137997
5	0,128998

6	0,114001
7	0,111995
8	0,100004
9	0,104997
10	0,135996
Average	0,170598

5.3. Disadvantages of Current Method

Watershed is one of many segmentation algorithms to segment images. Watershed sometimes gives over-segmentation for some pictures [22], including this problem written in this paper. To tackle the over-segmentation provided by the watershed algorithm, first, we selected the region of boxes. Then we decided on parts that have area, height, and width in the specified range. Selecting a region of interest means adding more resources to be used since we use the thresholding method to choose the area of interest.

5.4. Future Research

This paper has shown how watersheds perform object detection and object counting of the bottles in the boxes. Since the form only used a black platform as background and blue color of bottlecaps means only one environment used during research, future researches expected to examine the obtained watershed algorithm accuracy within a natural environment, using a conveyor. Testings conducted with a background showing a contrast level slightly different from bottles and boxes. Also, there is a need for comparison with other segmentation algorithms to measure the accuracy, time, and resources.

REFERENCES:

[1] A. Saptari, J. X. Leau, P. K. NG, dan E. Mohammad, "Human Error and Production Rate Correlation in Assembly Process of Electronics Goods," 2014, doi: 10.1299/jsmedsd.2014.24._2309-1_.

[2] Y. J. Cha, K. You, dan W. Choi, "Vision-based detection of loosened bolts using the Hough transform and support vector machines," *Autom. Constr.*, vol. 71, no. Part 2, hal. 181–188, 2016, doi: 10.1016/j.autcon.2016.06.008.

[3] M. B. Ramzan dan C. W. Kang, "Minimization of inspection cost by determining the optimal number of quality inspectors in the garment industry," *Indian J. Fibre Text. Res.*, vol. 41, hal. 346–350, 2016.

[4] C. W. Kang, M. B. Ramzan, dan B. Sarkar, "Effect of inspection performance in smart manufacturing system based on human quality control system," *Int J Adv Manuf Technol*, vol. 94, hal. 4351–4364, 2017, doi: 10.1007/s00170-017-1069-4.

[5] S.Sathiyamoorthy, "Industrial application of machine vision," *Int. J. Res. Eng. Technol.*, hal. 678–682, 2014.

[6] M. Baygin, M. Karakose, A. Sarimaden, dan E. Akin, "An image processing based object counting approach for machine vision application," *Int. Conf. Adv. Innov. Eng.*, 2018.

[7] A. Yudhana, Sunardi, dan S. Saifullah, "PERBANDINGAN SEGMENTASI PADA CITRA ASLI DAN CITRA KOMPRESI WAVELET UNTUK IDENTIFIKASI TELUR," *Ilk. J. Ilm.*, vol. 8, no. Desember, hal. 190–196, 2016.

[8] A. Rosebrock, "Waterhsed OpenCV," 2015. https://www.pyimagesearch.com/2015/11/02/watershed-opencv/ (diakses Jan 10, 2021).

[9] H. P. Narkhede, "Review of Image Segmentation Techniques," *Int. J. Sci. Mod. Eng.*, vol. 8, no. 18, hal. 2319–6386, 2013.

[10] A. S. Kornilov dan I. V Safonov, "An Overview of Watershed Algorithm Implementations in Open Source Libraries," *J. Imaging*, vol. 4, no. 10, hal. 123, 2018, doi: 10.3390/jimaging4100123.

[11] R. Romero-Zaliz dan J. F. Reinoso-Gordo, "An updated review on watershed algorithms," *Stud. Fuzziness Soft Comput.*, vol. 358, hal. 235–258, 2018, doi: 10.1007/978-3-319-62359-7_12.

[12] H. Miao dan C. Xiao, "Simultaneous Segmentation of Leukocyte and Erythrocyte in Microscopic Images Using a Marker-Controlled Watershed Algorithm," *Comput. Math. Methods Med.*, vol. 2018, 2018.

[13] M. H. Malik, T. Zhang, H. Li, M. Zhang, S. Shabbir, dan A. Saeed, "Mature Tomato Fruit

Detection Algorithm Based on improved HSV and Watershed Algorithm," *IFAC-PapersOnLine*, vol. 51, no. 17, hal. 431–436, 2018, doi: 10.1016/j.ifacol.2018.08.183.

[14] S. M. Wiranti, "Analisis Pengendalian Kualitas Kemasan Botol Plastik Dengan Metode Statistical Process Control (Spc) di PT. Sinar Sosro KPB Pandeglang," *J. Intent*, vol. 2, no. 1, hal. 94–102, 2019.

[15] P. N. Andono, T. Sutojo, dan Muljono, *Pengolahan Citra Digital*. Penerbit Andi, 2017.

[16] P. Chakravorty, "What Is a Signal? [Lecture Notes]," *IEEE Signal Process. Mag.*, vol. 35, no. 5, hal. 175–177, 2018, doi: 10.1109/MSP.2018.2832195.

[17] S. Sarangi, M. Sahidullah, dan G. Saha, "Optimization of data-driven filterbank for automatic speaker verification," *Digit. Signal Process. A Rev. J.*, vol. 104, hal. 102795, 2020, doi: 10.1016/j.dsp.2020.102795.

[18] A. H. Pratomo, W. Kaswidjanti, dan S. Mu'arifah, "Implementasi Algoritma Region of Interest (ROI) Untuk Meningkatkan Performa Algoritma Deteksi Dan Klasifikasi Kendaraan," *J. Teknol. Inf. dan Ilmu Komput.*, vol. 7, no. 1, hal. 155–162, 2020, doi: 10.25126/jtiik.202071718.

[19] A. S. Herlambang, O. D. Nurhayati, dan K. T. Martono, "Sistem Pendeteksi Kualitas Daging Dengan Ekualisasi Histogram Dan Thresholding Berbasis Android," *J. Teknol. dan Sist. Komput.*, vol. 4, no. 2, hal. 404–413, 2016, doi: 10.14710/jtsiskom.4.2.2016.404-413.

[20] A. Kaehler dan B. Gary, *Learning OpenCV---Computer Vision with the OpenCV Library*, vol. 16, no. 3. O'Reilly Media, Inc., 2008.

[21] X. Deng, Q. Liu, Y. Deng, dan S. Mahadevan, "An improved method to construct basic probability assignment based on the confusion matrix for classification problem," *Inf. Sci. (Ny).*, 2016, doi: 10.1016/j.ins.2016.01.033.

[22] N. Amoda dan R. K. Kulkarni, "Efficient Image Segmentation Using Watershed Transform," *Int. J. Comput. Sci. Technol.*, vol. 4, no. 2, hal. 214–218, 2013.

[23] C.-F. Wong, J. Y. Yeo, dan S. K.-E. Gan, "APD Colony Counter App: Using Watershed Algorithm for improved colony counting," *Sci. Phone apps Mob. Device*, vol. 5, no. 5, hal. 10–13, 2019, doi: 10.30943/2019/c23122019.

[24] S. Tan, Z. Mai, L. Qi, dan Y. Wang, "Segmentation and counting algorithm for touching hybrid rice grains," vol. 162, no. April, hal. 493–504, 2019, doi: 10.1016/j.compag.2019.04.030.

[25] J. Enterprise, *Otodidak Pemrograman Python*. Elex Media Komputindo, 2017.

[26] D. Kuhlman, *A Python Book*. 2013.

[27] R. V, N. K, dan I. R. J, "Foreground algorithms for detection and extraction of an object in multimedia," *Int. J. Electr. Comput. Eng.*, vol. 10, no. 2, hal. 1849–1858, 2020, doi: 10.11591/ijece.v10i2.pp1849-1858.

[28] V. Der Walt, J. L. Sch, dan J. Nunez-iglesias, "scikit-image : image processing in Python," *PeerJ*, hal. 1–18, 2014, doi: 10.7717/peerj.453.

[29] S. Suzuki dan K. A. be, "Topological structural analysis of digitized binary images by border following," *Comput. Vision, Graph. Image Process.*, vol. 30, no. 1, hal. 32–46, 1985, doi: 10.1016/0734-189X(85)90016-7.

[30] N. Efford, *Digital Image Processing : A Practical Introduction Using Java*. Pearson Education, 2000.

[31] Z. Strahinja, "#007 OpenCV projects – Image segmentation with Watershed algorithm," 2020. http://datahacker.rs/007-opencv-projects-image-segmentation-with-watershed-algorithm/ (diakses Jan 16, 2021).