JURNAL INFOTEL
VOL. 17, NO. 3, AUGUST 2025, PP. 649–665. DOI:10.20895/INFOTEL.V17I3.1228

RESEARCH ARTICLE

Sack Object Counter on a Conveyor Belt Based on Segmentation Using the Thresholding Technique

Ali Rizal Chaidir[1,*], Gramandha Wega Intyanto[2], Dodi Setiabudi[2], and Dirgahayu Kusuma Wibowo[3]

[1,2,3]Electrical Engineering, Universitas Jember, Jember Regency 68121, Indonesia
[4]CV Sinergi Teknokarya, Jember Regency 68121, Indonesia

*Corresponding email: ali.rizal@unej.ac.id

Received: August 25, 2024; Revised: August 05, 2025; Accepted: August 26, 2025.

Abstract: Automation technology enables better outcomes in terms of time efficiency, material usage, and reduced error rates in a process. Sensors and visual sensing are components or methods frequently used in industrial automation systems. Visual sensing methods can replace simple tasks typically performed by an operator's vision in the industry, such as counting specific objects. Object counting algorithms are tailored to the type of object being counted; for example, counting fish objects and counting sacks on a conveyor belt require different algorithms. Operators who are tired or unfocused can cause errors in counting objects, such as sacks on a conveyor belt, leading to financial losses. The main component used in this automatic counting system is a webcam. Each image frame is captured and processed in a computer to obtain parameters used as the basis for counting sack objects. The counting results are displayed on a monitor to facilitate the operator's view of the output. The method used is segmentation with a thresholding technique, which allows the separation of sack objects from the conveyor. The application of the segmentation method yields accurate counts; a total of 21 sack objects on the conveyor belt were accurately counted without error using this method. The use of filters did not affect the counting results, while the area size did. An area size of 50×50 provided the most accurate counting results and the best FPS (Frames Per Second) compared to other area sizes. This technique ensures that the calculation process does not result in errors that lead to losses.

Keywords: Conveyor belt, Object counter, Image processing, Segmentation, Thresholding

 ISSN: 2085-3688; e-ISSN: 2460-0997

1 Introduction

Indonesia possesses abundant natural resources, including sugarcane, which can boost economic potential. However, in 2022/2023, Indonesia was not among the top 10 countries globally for sugar production. One of the factors contributing to Indonesia's suboptimal sugar production is the limited use of automated machinery in the sugar agriculture industry. In fact, there are many aspects of automation related to the sugar industry that could be implemented in Indonesia to increase sugar production, such as:

- Sugarcane Harvesting: The harvesting process can be modernized with the use of automated sugarcane harvesters, which help improve efficiency and productivity in gathering sugarcane from the fields.
- Sugarcane Transportation: The transportation system from the fields to the factory can also be automated using trucks equipped with automation technology, making the transportation process easier and faster.
- Sugar Processing: In sugar factories, various processing stages, including sugarcane milling, refining, and crystallization, can be automated using machines that efficiently control these processes.
- Production Management: Automation systems can be applied in production management, including the automatic monitoring and control of operational conditions to enhance the quality and consistency of sugar products.
- Energy Usage: Energy consumption can be optimized with automation, for example, using sensors to regulate energy usage efficiently during the sugar production process.

By adopting automation technology, the sugar industry can increase efficiency, reduce production costs, and improve overall product quality. Another fact is that in recent decades, automation technology has become increasingly commonplace, and humans interact with it almost always. Automation technology is not only found in the sugar industry but also in everyday life, such as using washing machines for laundry and controlling lights based on sensor readings. This technology is also widely used in various fields, such as healthcare [1], education, agriculture, mining, and building construction. An example of automation in healthcare is the use of robots to assist doctors in performing surgeries [2, 3] and in examining internal organs using a small camera. In education, automation is applied in administrative processes such as attendance tracking and the distribution of learning materials. In agriculture, automation is more commonly used in off-farm and on-farm activities. In off-farm operations, automation is used for packaging, producing derivative products, and counting agricultural yields. An example of on-farm automation is the use of drones for fertilizer application [4, 5], soil processing, and harvesting. Sensor networks for monitoring environmental conditions and mining workers are useful for gathering real-time data, enabling faster and more informed decision-making in the mining sector. In building construction, IoT-based sensor networks in automation are useful for monitoring building conditions [6, 7], such as tilt and humidity.

In the industrial world, automation is crucial for reducing error rates in processes [8], enhancing time and material efficiency, and maximizing financial returns [9]. Automation technology almost always involves the use of sensors [10, 11] or visual sensing. Electronic sensors can partially replace human senses of taste, smell, and hearing, while visual sensing technology can partially substitute for vision.

Visual sensing, commonly referred to as machine vision, is a scientific field that utilizes cameras as its primary component. The camera, which can be a webcam, captures specific images that are then processed in a computer to extract certain parameters [12]. One advantage of using visual sensing in automation systems is that it reduces wiring complexity compared to electronic sensors [13]. As a result, system installation is relatively easier, and it is generally more cost-effective for complex systems. There are several examples of visual sensing applications, such as detecting objects [14], colors, and shapes, classifying objects, and performing object counting [15,16]. The type of object being counted influences the algorithm used [17]. Visual sensing in automation systems uses image processing techniques to perform its functions [18]. The basic image processing techniques commonly used include image conversion, filtering, pixel scanning, and adding specific annotations or characters to images. Additionally, segmentation techniques are crucial for creating effective images that can be further processed [19,20], such as those using thresholding [21,22].

Object counting systems are an example of automation systems found in various fields, including transportation management, biomedicine [22], agriculture (e.g., counting soybean seeds and palm oil seeds) [23], and mining. In transportation management, object counting algorithms are used to count road users and assess traffic density, utilizing algorithms such as Tensorflow Object Counting or MOG2 Background Subtractor for image processing. Another example is in mining, where conveyor belts transporting coal might carry unwanted objects, such as large non-coal items [24]. These objects need to be sorted out, and monitoring them can pose safety risks in the production process. Machine vision is used to address this issue, with algorithms like adaptive weighted multi-scale Retinex (MSR) image enhancement improving the quality of images captured by cameras over conveyor belts. The object recognition accuracy for large foreign objects exceeds 97% [25]. Visual sensing can also be used to detect damage in conveyor belt systems. Conveyor belts are critical equipment in the transport process, such as in coal mining [26]. Damage to conveyor belts can compromise system stability and disrupt mining efficiency.

Based on the explanations provided, the researcher implemented automation technology in the sugar agriculture industry to increase sugar production. The implementation involves using an object counting algorithm with machine vision and image processing techniques. Operators count sugar packed in sacks while the sacks are on the conveyor belt. Operator fatigue is a factor that contributes to errors in counting sacks. Counting errors lead to incorrect revenue calculations and potential financial losses. An algorithm is needed to count sack objects on the conveyor belt, with image processing-based counting aimed at reducing counting errors and providing an easy way to monitor the results digitally. This article presents a detailed solution for counting sack objects on a conveyor belt using image processing and a segmentation method.

2 Research Method

2.1 Sack Object Counting Algorithm

The algorithm used is generally represented in the form of a flowchart, as shown in Figure 1. There are four main steps involved in counting sack objects on the conveyor belt. The process starts by capturing the frames to be processed and initializing a variable to store temporary data. The next step involves converting the image frame. The initial image frame obtained when the system starts operating is in BGR format; this image frame

is then converted into a binary image frame consisting of pixel values 255 and 0 [27, 28]. The principle of this method is to process the image to be segmented by determining a threshold value [22]. When an object with matching pixels is found, those pixels are uniformly segmented, allowing the image object to be extracted according to the predefined threshold value. The segmentation steps using the Thresholding technique include analyzing the grayscale or BGR values of the image, comparing the grayscale or BGR pixel values with the threshold value, binarizing the image, and finally extracting the object within the specific image based on the binarization result [22]. This conversion uses Equation (1). The binary image is used as the final image to determine the parameters of the sack object. The binary image is chosen because it has two possible values for each pixel—0 and 255—making it more suitable for processing and easier to apply compared to other techniques. Color segmentation techniques are more appropriate for separating the color of an object from its background. In contrast, texture segmentation is more effective for images that exhibit significant variations in pattern or texture, even if the colors are similar.

$$G(x,y) = \begin{cases} 255 & \text{if } \mathrm{Tr}_{\max} \geq Y(x,y) \geq \mathrm{Tr}_{\min} \cap \\ & \quad \mathrm{Tg}_{\max} \geq Y(x,y) \geq \mathrm{Tg}_{\min} \cap \\ & \quad \mathrm{Tb}_{\max} \geq Y(x,y) \geq \mathrm{Tb}_{\min} \\ 0 & \text{if else} \end{cases} \tag{1}$$

where $G(x,y)$ represents the binary image, $Y(x,y)$ represents the BGR image, $\mathrm{Tr}_{\max}$ and $\mathrm{Tr}_{\min}$ indicate the maximum and minimum threshold values for the red component, $\mathrm{Tg}_{\max}$ and $\mathrm{Tg}_{\min}$ indicate the maximum and minimum threshold values for the green component, and $\mathrm{Tb}_{\max}$ and $\mathrm{Tb}_{\min}$ indicate the maximum and minimum threshold values for the blue component.

The next step is to prepare an area with a specific pixel size and position it within a certain frame to be used for detecting the presence of a sack object. The algorithm for identifying a sack object is based on the number of pixels within the predefined area. If the number of pixels is greater than 80% of the total possible pixels with a value of 255 within that area (variable x), then the variable a should be incremented by 1. Subsequently, if this condition is met and the number of pixels in that area falls below 40% of the total possible pixels with a value of 255 in that area (variable y), it indicates that the sack object has passed through the area. Figure 1 illustrates the flowchart used when a 50×50 pixel area is employed, with values of $x = 2000$ and $y = 1000$ being used to determine the presence of a sack object.

The flowchart in Figure 1 is converted into source code written in the Python programming language, with each step represented as follows:

1. Video image acquisition
   Image acquisition is the process of capturing visual data from a sensor used in image processing. In this research, the sensor used for image acquisition is a Logitech C920 webcam with a capture distance of 1.5 meters—the pseudocode for video data acquisition is shown in Pseudocode 1.
   The algorithm testing is performed on a previously recorded video, stored on Drive E with the filename and format '5.mp4'. This video displays a total of 21 sack objects. In the source code, the video is stored in a variable named 'cap'. This study employs OpenCV as the open-source library for image processing. OpenCV provides an infrastructure for developing computer vision-based applications, thereby

facilitating the rapid development of automation applications. The library includes several easily utilized functions, such as filtering, transformation, edge detection, and segmentation.

Pseudocode 1 Image acquisition

Import computer vision library
Initialize `video_capture` with video file located at `"E:/5.mp4"`

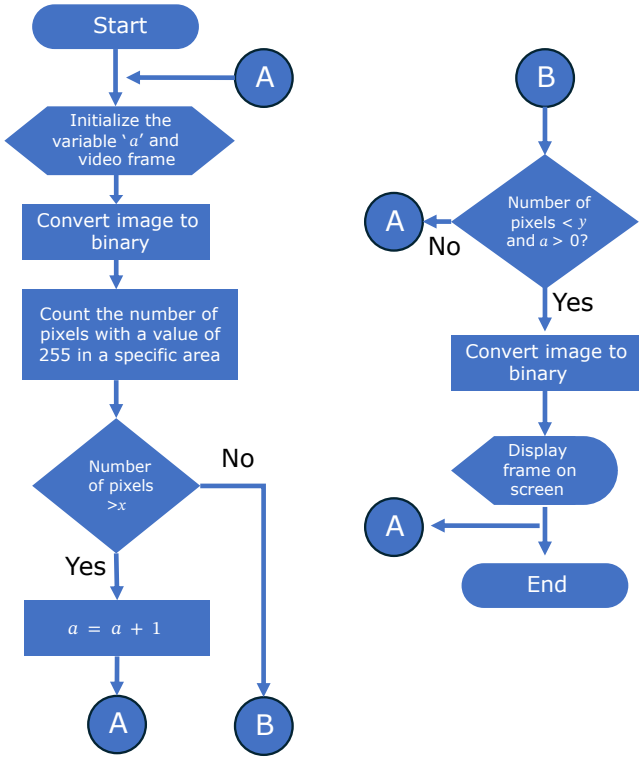


Figure 1: Flowchart of the sack object counting algorithm.

2. Conversion of BGR Image to Binary
   Converting the BGR image to binary is used to transform the original image, which utilizes the BGR color space from the webcam capture. The BGR image consists of three layers: Blue, Green, and Red, whereas the binary image has only one layer. The number of color space layers affects the data complexity and processing time; the more layers there are, the greater the data complexity and the longer the processing time. This is why the researchers chose to perform a direct conversion from the BGR color space to a binary image.
3. Counting Pixels with a Value of 255 in the Area Traversed by the Sack, with an Area Size of 50×50 Pixels
   The technique used at this stage involves counting pixels with a value of 255. This same technique is also employed in robot control and navigation [27,28], where pixels

Pseudocode 2 Color conversion to binary (thresholding)

```
Set R_threshold ← 30
Set G_threshold ← 30
Set B_threshold ← 30
Set base_R ← 210
Set base_G ← 210
Set base_B ← 210
Read frame from camera → frame
Apply thresholding to frame:
  frame_threshold ← inRange(frame,
    lower_bound = (base_B − B_threshold, base_G − G_threshold, base_R − R_threshold),
    upper_bound = (base_B + B_threshold, base_G + G_threshold, base_R + R_threshold)
  )
```

with a value of 255 are counted and used to determine the movement of the robot's actuators. In that research, the robot was capable of navigating and moving as desired. In this study, pixels within a specific area that have a value of 255 are counted, and the count is used to make decisions about a sack object. The binary image obtained serves as a reference for determining whether an object on the conveyor is a sack. The variable 'total_pixel' is used to store the count of white pixels in a predetermined area. In this part of the source code, the area is located at pixel coordinates (600, 350) with a size of 50×50 pixels.

Pseudocode 3 Set detection area size

```
Initialize total_pixel ← 0
for j = 0 to 49 do
  for i = 0 to 49 do
    if frame_threshold[j + 600, i + 350] = 255 then
      total_pixel ← total_pixel + 1
    else
      total_pixel ← total_pixel + 0
    end if
  end for
end for
```

4. Detecting and Counting Sack Objects
   The value stored in the variable 'total_pixel' is checked; if it is greater than 2000, the variable a is incremented by 1. If it is less than 1000 and the value of the variable a, then the variable 'object_count' is incremented by 1. The variable 'object_count' holds the count of sacks detected.

Pseudocode 4 Counting objects

```
if total_pixel > 2000 then
    a ← a + 1
else if total_pixel < 1000 then
    a ← a + 0
    if a > 0 then
        object_count ← object_count + 1
        a ← 0
    end if
end if
```

2.2 System Usage Configuration

A camera is positioned above the conveyor belt, carrying sack objects, to capture each frame of the image, with a distance of 1.5 meters between the camera and the conveyor. The camera used is a Logitech C920 webcam. Each image frame is captured and then transmitted to a computer for processing using the developed algorithm. The computer used in this study has the following specifications: 2.00 GB RAM and an Intel(R) Celeron(R) CPU N3350 @ 1.10 GHz. The operator uses the processed image results to determine the count of sack objects. Figure 2 illustrates the system usage configuration, which is also employed for fish counting and tracking systems [29]. In that study, the camera was placed above a fish container to count the fish using image processing algorithms. The average counting accuracy was 91%, with a detection accuracy of 94%. During the trials, there were occurrences of both overcounting and undercounting. Factors affecting the accuracy of counting and detection include foreign objects or shadows caused by lighting conditions. These factors are also experienced in this study. Solutions to address these issues include ensuring that external lighting does not interfere with the frame capture process through the webcam, such as providing covers on all sides of the area between the webcam and the sack objects. Another solution is to provide additional lighting, such as LED lights directed at the objects [19].

2.3 System Testing Scenarios

Testing is conducted to evaluate the performance of the developed algorithm. There are scenarios for testing each subsystem or sub-algorithm. The testing of BGR to binary image conversion is performed by examining the conversion results. The conversion is considered successful if the resulting image displays a certain number of white pixels or pixels with a value of 255 that form the sack object, as described by Eq. (1). The next test evaluates the impact of area size on the count of sack objects. This test is used to determine the optimal FPS (Frames Per Second) that can accurately count sack objects for specific area sizes. Four different area sizes are used in this test.

Another test assesses the effect of the distance between sack objects on the counting results. The outcomes of this test can serve as guidelines for the operator to place sack objects on the conveyor, ensuring accurate counting by the system. The impact of the blurring filter size on the counting is also tested. This test aims to understand the effect of applying a blurring filter on the sack object counting results produced by the algorithm. Three sizes

of blurring filters are used in this test: 5×5, 7×7, and 9×9. The blurring filter used is an averaging filter, which computes the average value of all pixels in the kernel area and replaces the central element. The processing is performed through convolution between the original image and the filter kernel, as described by Eq. (2).

$$h(x) = f(x, y) \cdot g(x, y) \tag{2}$$

The filter function is denoted as $g(x, y)$, and the convolution process is performed on the original image, denoted as $f(x, y)$.

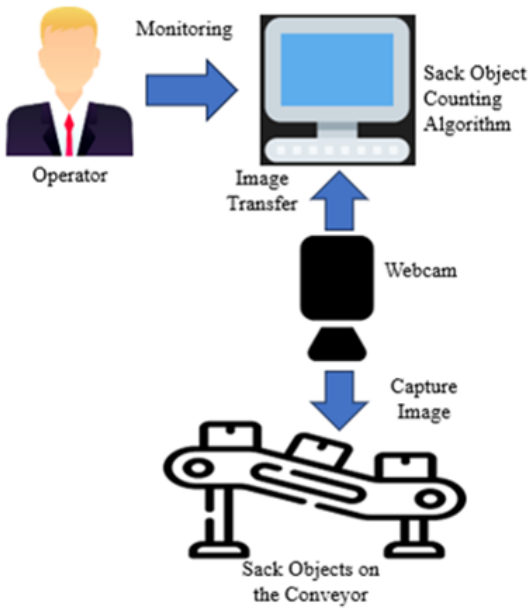


Figure 2: System configuration illustration.

3 Results and Discussion

Testing was conducted to evaluate the performance of the developed system. The results obtained from the testing are as follows.

3.1 Results of BGR to Binary Image Conversion

The results of this stage of testing are shown in Figure 3. It is observed that the conversion from BGR to a binary image performs well, with the binary image produced from BGR showing more precision compared to the binary image generated from grayscale images. This is because the grayscale pixel values are obtained from the average values of the corresponding BGR pixels. The sack objects in the binary image appear consistent with those in the BGR image. The quality of the conversion or separation of the background from the sack objects is influenced by the threshold values set in the algorithm. The technique

used to optimize the conversion results involves determining the threshold value by directly sampling pixels from the object. The threshold value is used to separate pixels into two groups, unlike other segmentation techniques, such as contour-based segmentation, which cannot produce derivative images that clearly separate foreground and background. Contour-based techniques focus on detecting edges or contours within the image [30, 31]. Edges are identified as significant changes in pixel color, requiring operators like Sobel, Laplacian of Gaussian (LoG), and Canny to detect edges in the image. If such segmentation techniques were applied, the derivative image of the sack object in the BGR image (Figure 3) would show a binary image with several white pixels on the edges of the sack object, resulting in the algorithm never concluding that the object is a sack.

Another segmentation technique, color and texture segmentation [32], uses a model that produces an image visually showing pixel grouping with consistent texture or color from homogeneous pixels of the original image, or it can be viewed as reconstructing the original image composed of regions with homogeneous properties. This model unifies pixels with similar colors, textures, or correlations. This segmentation technique relies on a degradation parameter, identical to the thresholding technique used in this study, which also has a parameter affecting segmentation results, i.e., the threshold value. Color and texture segmentation could be applied in this case, where sack objects can be segmented into white pixels, allowing the algorithm to conclude that the object is a sack. This segmentation technique can handle more than two groups of similar textures or colors, unlike the thresholding technique, which separates foreground and background. Therefore, the proportional technique to address the sack object counting problem in this study is the thresholding technique.

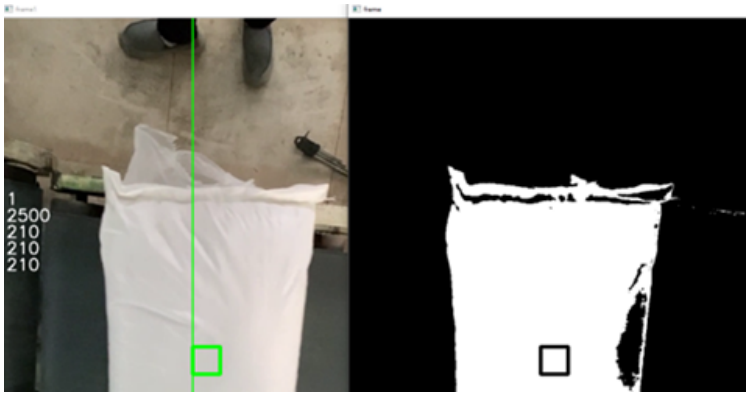


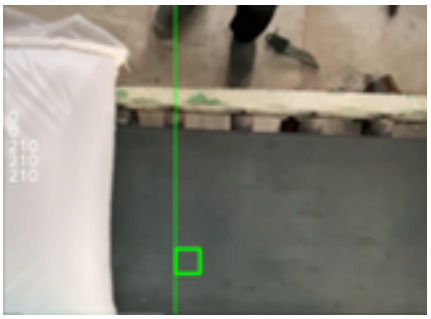
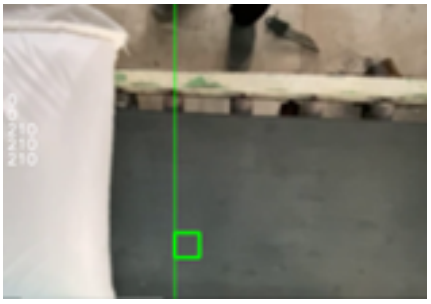
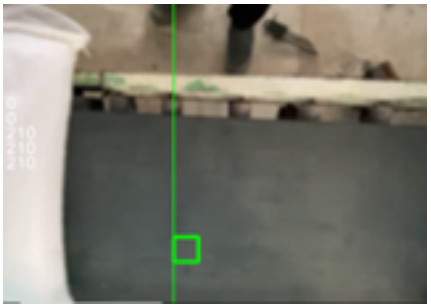
Figure 3: BGR image and binary image.

3.2 Results of Testing the Effect of Blurring Filter Size on Counting Accuracy and FPS

Blurring filters are among the filters that can reduce noise or enhance image quality in certain cases [33] and are widely used to reduce noise [34]. The effects of applying a blurring filter are shown in Table 1. This table illustrates the impact of the blurring filter on images containing sack objects. With a 9x9 filter size, the sack objects appear more blurred

compared to a 5×5 filter size. The testing indicates that the blurring filter can influence the image conversion process from BGR to binary. Table 2 shows that the presence or absence of a blurring filter does not affect the sack object counting results, but it does impact the FPS (Frames Per Second) value. The FPS without the filter is higher compared to when the filter is applied to the BGR image before conversion to binary. This result is attributed to the fact that blurring filters require higher computational resources [34].

Table 1: Effect of blurring filter size on image appearance

Blurring Filter Size	Image Appearance
5×5	
7×7	
9×9	

3.3 Results of Testing the Effect of Area Size on Counting Accuracy and FPS

The size of the area is a crucial component of the sack object counting system. Table 3 illustrates the differences in the area sizes displayed in the images. The image sizes are

Table 2: Effect of blurring filter size on sack counting and fps (frames per second)

Blurring Filter Size	Sack Counting Results	FPS
Without filter	21	20
5×5	21	15
7×7	21	15
9×9	21	14

marked in green, with the smallest size being 20×20 pixels and the largest being 110×110 pixels. The results of this stage of the test are shown in Table 4. The sizes of the 50×50 and 80×80 pixels accurately count the number of sack objects, totaling 21 objects (which is the same as the number of sack objects in the test video). In contrast, the other sizes yield incorrect counting results. The size of the 50×50 area provides a higher FPS (Frames Per Second) compared to the size of 80×80. This is because the subalgorithm scans and counts white pixels more efficiently with a smaller area size. The 80×80 area size requires the algorithm to scan a maximum of 6400 pixels, whereas the 50×50 area size requires scanning 2500 pixels.

3.4 Results of Testing Sack Counting with Close and Sparse Spacing

The distance between individual sack objects affects the accuracy of the counting results. Table 5 illustrates examples of distances where sacks are counted correctly and distances where sacks are not counted accurately. When sack objects are in proximity or in contact with each other, the algorithm counts them as a single sack object. It occurs because the area is filled with white pixels, amounting to more than 80% of the total pixels in the area, or has not yet reached below 40% of the total pixels in the area. Sack objects with a minimal pixel distance greater than 40% can be counted accurately, as the area can transition from being at least 80% filled with white pixels to a maximum of 40% filled with white pixels. The test video can be viewed at the following link: https://unej.id/pengujianKarungKonv. The counting results are accurate, and the algorithm performs as expected. Similarly accurate counting results are also demonstrated in other algorithms for different objects using segmentation techniques [19]. The use of YOLO object detection can also serve as a first step in creating a counting system, but it requires hardware with higher specifications compared to the hardware used for the proposed algorithm [35]. Other research utilizes a proximity sensor as the primary component for calculating objects [36]; however, it requires several additional electronic components to support it, and the data acquisition process is more complex compared to the proposed technique. The weakness of this technique is that it can only be applied to count objects that have a single type of color and are distinct from the conveyor color, and it relies on consistent light intensity. The use of a light intensity control system using certain types of control can help ensure that the light around the calculated object is consistent.

Table 3: Effect of area size on image appearance

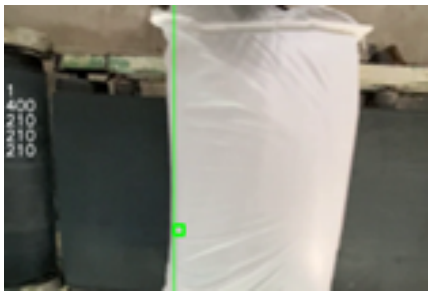
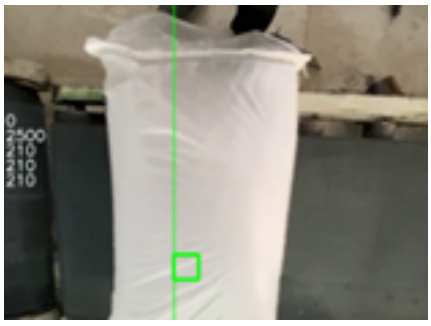
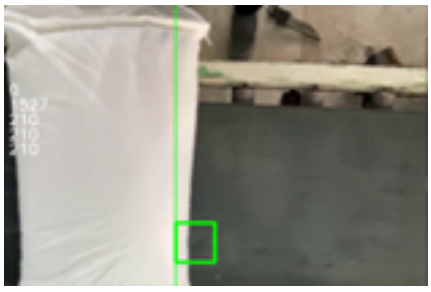
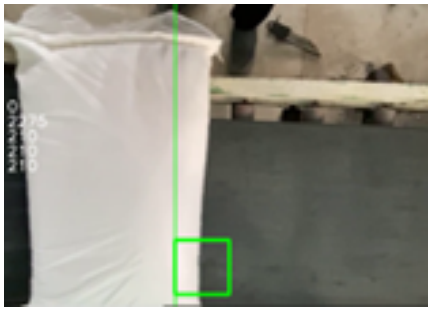
Area Size	Image Appearance
20 × 20	
50 × 50	
80 × 80	
110 × 110	

Table 4: Effect of area size on sack counting and fps (frames per second)

Area Size	x	y	**Sack Counting Results**	**FPS**
20×20	320	160	34	26
50×50	2000	1000	21	20
80×80	5120	2560	21	12
110×110	9680	4840	20	8

Table 5: Display of sack object distances affecting counting accuracy

Illustration of Sack Object Distances that Can Be Accurately Counted	**llustration of Sack Object Distances that Cannot Be Accurately Counted**
	
	

4 Conclusion

Each object-counting system using image processing employs different algorithms, which are determined based on the object to be counted. In the sack object counting system, operating on a single conveyor belt has several factors affecting counting accuracy and FPS (Frames Per Second). The distance between sack objects and the area size impact the counting results provided by the algorithm. Objects in contact with each other are not counted accurately by the algorithm. Area sizes of 50×50 and 80×80 successfully count the number of sack objects correctly, especially when the distance between sack objects is at least equal to the size of the area used by the algorithm, which is 50 pixels. The FPS for an area size of 50×50 is 20 FPS, which is better compared to 12 FPS for an area size of 80×80. The FPS value is also affected by the use of filters. FPS without a filter provides a better value of 20 FPS compared to 15 FPS with a filter. The use of a filter does not affect the accuracy of sack object counting. This technique can be used to ensure that the sack counting process does not result in an error, which would cause one party, either the buyer or seller, to experience a loss.

References

[1] S. M. Mohammad, "Implementation of automation in applications of healthcare, private, and public sectors in it," *International Journal of Innovations In Engineering Research And Technology (IJIERT)*, vol. 7, no. 5, pp. 364–372, 2020.

[2] M. Shahinpoor and S. Gheshmi, *Surgery Robotic*. Boca Raton: Pan Stanford Publishing, 2015.

[3] F. Graur, E. Radu, N. A. Hajjar, C. Vaida, and D. Pisla, "Surgical robotics—past, present and future," in *Mechanisms And Machine Science*, pp. 159–171, Springer International Publishing, 2018.

[4] G. P. Borikar, C. Gharat, and S. R. Deshmukh, "Application of drone systems for spraying pesticides in advanced agriculture: A review," in *International Conference On Advances In Mechanical Engineering*, 2022.

[5] S. Souvanhnakhoomman, "Review on application of drone in spraying pesticides and fertilizers," *International Journal Of Engineering Research & Technology (IJERT)*, vol. 10, no. 11, pp. 94–98, 2021.

[6] A. Katiyar and P. Kumar, "A review of internet of things (iot) in construction industry: Building a better future," *International Journal Of Advanced Computing Science And Engineering*, vol. 3, no. 2, pp. 65–72, 2021.

[7] M. Giovanardi, M. Trane, and R. Pollo, "Iot in building process: A literature review," *Journal Of Civil Engineering And Architecture*, vol. 15, no. 9, pp. 475–487, 2021.

[8] R. Hinai, A. Farh, and S. A. Hasani, "Image processing based automatic color object sorting using plc system," *International Journal Of Electrical And Electronics Research*, vol. 7, no. 2, pp. 50–62, 2019.

[9] S. Singh and S. Namekar, "A review on automation of industries," *International Journal Of Engineering Applied Sciences And Technology*, vol. 4, no. 12, pp. 298–300, 2020.

[10] C. Pardhasaradhi and A. D. Rani, "Industrial process automation and monitoring using iot," *International Journal Of Advance Research In Science And Engineering*, vol. 9, no. 10, pp. 13–22, 2020.

[11] G. S, K. U, S. S, O. R, and B. Y., "Colour based object sorting machine," *International Journal Of Advanced Research In Science, Communication And Technology (IJARSCT)*, vol. 2, no. 2, pp. 445–449, 2022.

[12] T. Dhaval, T. Kadam, S. Gupta, and P. V. Salunkhe, "Review on object counting system," *International Research Journal Of Engineering And Technology (IRJET)*, vol. 9, no. 3, pp. 967–970, 2022.

[13] T. J. Nuva, M. I. Ahmed, and S. S. Mahmud, "Design & fabrication of automatic color & weight based sorting system on conveyor belt," *Journal Of Integrated And Advanced Engineering (JIAE)*, vol. 2, no. 2, pp. 147–157, 2022.

[14] D. M. H. Ali, "Real-time objects detection, tracking, and counting using image processing techniques," *Al-Nahrain Journal For Engineering Sciences (NJES)*, vol. 26, no. 1, pp. 24–30, 2023.

[15] M. Parlak, "Camera based product counting of belt conveyors," *International Journal Of Electronics, Mechanical And Mechatronics Engineering*, vol. 4, no. 2, pp. 771–785, 2015.

[16] J. Moon, S. Lim, H. Lee, S. Yu, and K.-B. Lee, "Smart count system based on object detection using deep learning," *Remote Sensing*, vol. 14, no. 3761, pp. 1–17, 2022.

[17] V. B. Inchur, P. L. S, and P. Shankpal, "Implementation of blood cell counting algorithm using digital image processing techniques," in *International Conference On Recent Trends On Electronics, Information, Communication & Technology (RTEICT-2020)*, (Bangalore), 2020.

[18] P. K. Thotapalli, C. R. V. Kumar, and B. Reddy, "Feature extraction of moving object over a belt conveyor using background subtraction technique," in *International Conference On Precision, Meso, Micro And Nano Engineering (COPEN 10)*, (Chennai), 2017.

[19] Z. Nie, M.-H. Hung, and J. Huang, "A novel algorithm of rebar counting on conveyor belt based on machine vision," *Journal Of Information Hiding And Multimedia Signal Processing*, vol. 7, no. 2, pp. 425–437, 2016.

[20] H. E. Khoukhi, Y. Filali, A. Yahyaouy, M. A. Sabri, and A. Aarab, "A hardware implementation of otsu thresholding method for skin cancer image segmentation," in *International Conference On Wireless Technologies, Embedded And Intelligent Systems (WITS)*, (Fez), 2019.

[21] C. Patgiri and A. Ganguly, "Comparative study on different local adaptive thresholding techniques for detection of sickle cell anaemia from microscopic blood images," in *IEEE 16th India Council International Conference (INDICON)*, (Rajkot), 2019.

[22] Y. Cheng and B. Li, "Image segmentation technology and its application in digital image processing," in *International Conference On Advance In Ambient Computing And Intelligence (ICAACI)*, (Ottawa), 2020.

[23] R. Vanijjirattikhan, S. Nithi-Uthai, K. Ekkachai, P. Tittinutchanon, and P. Tohdam, "High accuracy conveyor-based object counting algorithm," in *IECON 2019 - 45th Annual Conference Of The IEEE Industrial Electronics Society*, (Lisbon), 2019.

[24] B. Luo, Z. Kou, C. Han, and J. Wu, "A "hardware-friendly" foreign object identification method for belt conveyors based on improved yolov8," *Applied Sciences*, vol. 13, no. 11464, pp. 1–24, 2023.

[25] L. Dai, X. Zhang, P. Gardoni, H. Lu, X. Liu, G. Królczyk, and Z. Li, "A new machine vision detection method for identifying and screening out various large foreign objects on coal belt conveyor lines," *Complex & Intelligent Systems*, vol. 9, pp. 5221–5234, 2023.

[26] X. Guo, X. Liu, P. Gardoni, A. Glowacz, G. Królczyk, A. Incecik, and Z. Li, "Machine vision based damage detection for conveyor belt safety using fusion knowledge distillation," *Alexandria Engineering Journal*, vol. 71, pp. 161–172, 2023.

[27] A. R. Chaidir, K. Anam, and G. A. Rahardi, "Lane tracking pada robot beroda holonomic menggunakan pengolahan citra," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 8, no. 1, pp. 69–79, 2020.

[28] A. R. Chaidir, D. Wahyuherdiyanto, and G. D. Kalandro, "Pengendalian mobile robot non-holonomic berdasarkan gestur jari menggunakan template matching," *Jurnal Nasional Teknik Elektro*, vol. 9, no. 1, pp. 21–27, 2020.

[29] R. Lumauag and M. Nava, "Fish tracking and counting using image processing," in *IEEE 10th International Conference On Humanoid, Nanotechnology, Information Technology, Communication And Control, Environment And Management (HNICEM)*, (Baguio City), 2018.

[30] B. Karthicsonia and M. Vanitha, "Edge based segmentation in medical images," *International Journal Of Engineering And Advanced Technology (IJEAT)*, vol. 9, no. 1, pp. 449–451, 2019.

[31] B. Padmapriya, T. Kesavamurthi, and H. W. Ferose, "Edge based image segmentation technique for detection and estimation of the bladder wall thickness," in *International Conference On Communication Technology And System Design*, 2012.

[32] S. Panda and P. K. Nanda, "Color and texture segmentation using an unified mrf model," *Journal Of Computer And Communications*, vol. 10, no. 6, pp. 139–164, 2022.

[33] A. Gupta and S. Shantaiya, "Reduction of image blurring with digital filters," *International Journal Of Engineering Research And Applications*, vol. 4, no. 1, pp. 139–143, 2014.

[34] N. M. Ibrahim, A. A. Elfarag, and R. Kadry, "Gaussian blur through parallel computing," in *International Conference On Image Processing And Vision Engineering*, 2021.

[35] J.-D. Wu, B.-Y. Chen, W.-J. Shyr, and F.-Y. Shih, "Vehicle classification and counting system using yolo object detection technology," *International Information and Engineering Technology Association (IIETA)*, vol. 38, no. 4, pp. 1087–1093, 2021.

[36] V. Kelkar, S. Redkar, P. Volvoikar, C. Dhargalkar, and D. Shirodkkar, "Automated counting system for industrial conveyors," *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, no. 7, pp. 700–703, 2020.