

Internet of Things for Underwater Shrimp Image Detection Using Blob Detector

Arif Setiawan^{a,e,*}, Hadiyanto Hadiyanto^{b,d}, Catur Edi Widodo^{c,d}

^a Doctoral Program of Information System, School of Postgraduate Studies, Diponegoro University, Semarang, Indonesia

^b Center of Biomass and Renewable Energy (CBIORE), Department of Chemical Engineering, Diponegoro University, Semarang, Indonesia

^c Department of Physics, Faculty of Science and Mathematics, Diponegoro University, Semarang, Indonesia

^d School of Postgraduate Studies, Diponegoro University, Semarang, Indonesia

^e Department of Information System, Faculty of Engineering, Muria Kudus University, Kudus, Indonesia

Corresponding author: *arif.setiawan@umk.ac.id

Abstract— Measuring biomass content is an important stage in harvesting shrimp as it will determine the harvest time. Manual detection has caused shrimp stress and eventually caused shrimp death; therefore, a new shrimp biomass determination is required. This research aims to design an IoT technique-based biomass measurement, using underwater shrimp video with fog and cloud computing processes to easily detect shrimp underwater, irrespective of the complex noise. The method consists of several steps: image processing using grayscale, thresholding, contour edge detection, labeling, and blob detection. The results revealed that the highest SSIM value in the thresholding process was 0.18, while the lowest MSE was 91.35. In addition, in the contour edge detection process, the highest PSNR value was 3.6, and the lowest MSE was 2.06. The blob detection process produces a maximum key performance of 566, 411, and 387 in the Laplacian of Gaussian (LoG), Difference of Gaussian (DoG), and Determinants of Hessian (DoH) methods, respectively. The Quality of Service (QoS) obtained throughput, loss, and delay values of 832.25, 0%, and 7.25 ms, respectively, in the data acquisition and computation processes, with the three parameters at a very good level. In conclusion, the IoT model is very suitable for underwater shrimp detection because it is a non-invasive method, contains high key performance blob detection, and has a very good QoS level and high-speed computation process.

Keywords— Shrimp underwater; image detection; blob detector; key performance; IoT.

Manuscript received 22 Mar. 2022; revised 18 Aug. 2022; accepted 19 Sep. 2022. Date of publication 30 Jun. 2023.
IJASEIT is licensed under a Creative Commons Attribution-Share Alike 4.0 International License.



I. INTRODUCTION

Shrimp is one of Indonesia's most popular seafood and contributes to almost half of the total seafood products. A large number of statistical data from the Food and Agriculture Organization (FOA) official website shows that the total worldwide shrimp yield exceeds six million tons, of which approximately 60% are sold in the world market [1]. Shrimp cultivation is an important aquaculture product that boosts a country's economy [2].

Therefore, monitoring its condition underwater is required to determine the population of shrimp culture [3]. Due to the increasing demand for high-quality shrimp, a method is required to detect its quality adequately [4]. The image segmentation technique is a popular method used in object recognition models, and the accuracy of this technique greatly affects object recognition and identification [5].

Pixels are found in digital images and used as input in the segmentation algorithm to generate information [6]. One of the simplest approaches to separating objects from the background and foreground is thresholding. This segmentation technique can be implemented in image processing and computer vision research because it can easily overcome the basic problem of object detection [7]. The optimum threshold is selected by minimizing the variance of the gray values concerning the mean in each phase. Obscurity is expressed at the beginning of the threshold process by allowing one pixel to be part of two or more phases comprising different membership values [8].

Contour segmentation is an image-processing step used to determine its geometry [9]. Edge contour segmentation is widely used to extract the image edges, making them clearer. The advantages of this method are high accuracy, more popularity, excludes the need to modify objects, and

suitability for monitoring at any time during the processes [10].

In computer vision, a blob algorithm identifies specific areas in an image. This method is widely used in detecting images using real-time monitoring systems that run on software platforms with high processing levels. Therefore, the implemented system requires the right infrastructure [11].

IoT innovations have taken sensor development to a new stage. It uses various sensors to provide numerous information and data collected from the camera and directly distributed to the network. The data collected allows the device to work independently [12]. In IoT, local computing on a fog network can lighten the burden on the cloud network, and this method also provides quick information with higher network efficiency [13]. The sensor's Wi-Fi gateway is connected to the router and the fog server and used for data communication. The use of this system is very effective due to its small size and low production cost [14].

This research proposes an IoT technique for detecting shrimp underwater using a blob detector. This IoT infrastructure is used for data transmission and computing processes carried out on fog and cloud servers. This method was used to detect the presence of shrimp underwater in aquaculture locations.

Wang [11] used a Laplacian of Gaussian (LoG) blob detector to evaluate the attenuation of adjacent overlapping image areas while maintaining its quality. The data used in this research were fluorescence microscopy cell images and electron micrograph nanoparticles. The experimental results show that the proposed method outperforms the existing ones for overlapping blob objects.

Another research was conducted on the use of a blob detector, filtration, gradient location, and estimation to search for shape estimates. The LoG detector scale ranged within a certain scale in the blob detection. The proposed method was compared with several others on microscopy cell images, and the results showed higher area precision and more precise shape estimation [15].

Malinowski et al. [16] used a blob detector to detect the pupil and iris based on an ellipsis or vision of light entering the eye. In this study, the camera used was perpendicular to the eye. Iris and pupil images in shape detection segmentation to detect pupillary edges. Dave et al. [17] used a blob analysis for real-time vehicle detection and showed the morphology operation and binary logical operation, and Support Vector Machine (SVM) for vehicle detection.

Another research describes using blob to detect IoT-based vehicles using image processing, such as segmentation, region, edge feature extraction, image labeling, blob analysis, and object detection. Data on vehicles were collected from cameras installed on the highway and the side road to record traffic. The results showed that the proposed method could better detect vehicles for validation than traditional methods [18]. The blob detector is also used to detect fake images, conducted by copying one area and pasting it to another using keypoint detection, original keypoint image, and fake image compared to the area of detection [19]. This research aimed to

design an IOT-based biomass detection which was implemented in the shrimp harvesting technique.

II. MATERIAL AND METHOD

The image detection process comprises several stages. The data used were collected from a shrimp farm with a pond made of tarpaulin. The process was directly carried out on the fog network by monitoring the shrimp using cameras placed in the pool. IoT infrastructure was wirelessly connected to its gateway for data communication. Fig. 1 (a) is a graphical representation of IoT, which shows that the fog server stores video data collected using a camera on its fog storage.

The research was carried out to compare the speed of fog and cloud computing. The first trial was conducted on the fog server, while the second was conducted on the cloud server. The steps taken are to extract the underwater shrimp video file from a sample of 3 images using the image processing stages: gray scaling, thresholding, contour edge detection, labeling, and blob detection. The research flow is shown in Fig. 1 (b).

A. Image thresholding

Image segmentation is fundamental in many images, video, and computer vision applications. It is often used to partition an image into separate areas and analyze it. The gray level of pixels in foreground objects completely differs from the background. Thresholding is a simple but effective tool for separating foreground objects from the background. The pixels in the image are divided into two large groups according to their gray level. It functions as a detector to distinguish between background and foreground objects in the image [20].

Image segmentation is partitioning a digital image into several smaller segments. The purpose is to simplify and convert the image representation into a more readable format and easier to analyze. The segmentation process must efficiently separate the foreground from the background and other foreign objects in the original image [21]. The threshold technique is one of the important strategies in image segmentation. The grayscale pixel value is represented at the gray level L , and the total pixel value is denoted by N [22].

$$L = [1,2,3,\dots,L] \quad (1)$$

$$N = n_1 + n_2 + n_3 + \dots + N_l \quad (2)$$

The pixels are divided into background C_b and foreground C_f , by giving the threshold value t .

$$C_b = [1,2,3,\dots,t] \quad (3)$$

$$C_f = [t+1, t+2, t+3,\dots,L] \quad (4)$$

The formula to determine the background and foreground variance for the threshold value t is as follows:

Background C_b :

$$\text{Weight } W_b = \sum_{i=1}^t \frac{n_i}{N} \quad (5)$$

$$\text{Mean } \mu_b = \frac{\sum_{i=1}^t i \cdot n_i}{\sum_{i=1}^t n_i} \quad (6)$$

$$\text{Variance } \sigma_b^2 = \frac{\sum_{i=1}^t (i - \mu_b)^2 \cdot n_i}{\sum_{i=1}^t n_i} \quad (7)$$

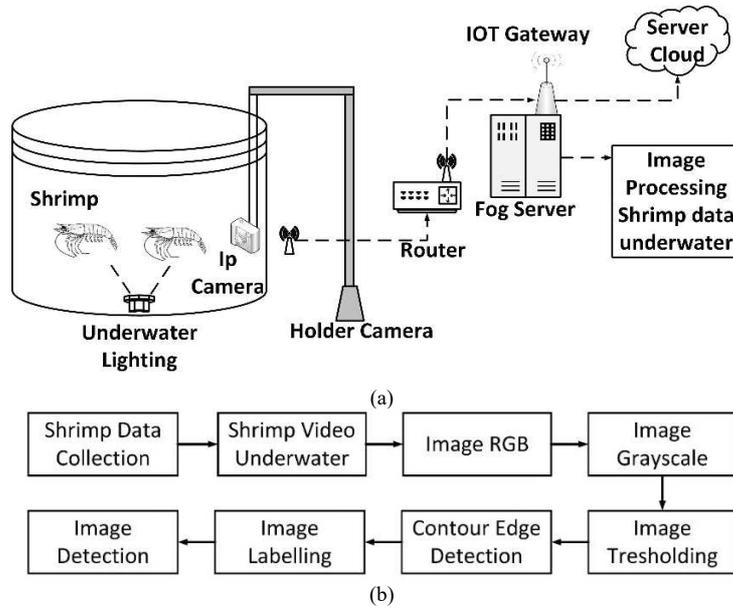


Fig. 1 (a) Research framework; (b) Research flowchart

Foreground Cf:

$$\text{Weight } w_f = \sum_{i=1+1}^L \frac{n_i}{N} \quad (8)$$

$$\text{Mean } \mu_b = \frac{\sum_{i=1}^L i \cdot n_i}{\sum_{i=1}^L n_i} \quad (9)$$

$$\text{Variance } \sigma_f^2 = \frac{\sum_{i=1+1}^L (i - \mu_f)^2 \cdot n_i}{\sum_{i=1+1}^L n_i} \quad (10)$$

The variance in class σ_w^2 is determined by summing the two variances multiplied by the weight.

$$\text{Variance in class } \sigma_w^2 = W_b \sigma_b^2 + W_f \sigma_f^2 \quad (11)$$

B. Image Contour

The initial approach to contour detection aims to measure the presence of object area boundaries in a particular image. Roberts, Sobel, and Prewitt used various operators to detect edges by measuring the image's grayscale using a locally derived filter. Marr and Hildreth used the zero crossing of the LoG operators and the Canny detector to model the edges using a threshold calculation. The digital image area detection algorithm tries to partition the image pixels into several components to ensure the resulting segmentation is neither too coarse nor too fine. The difference in the internal component $\text{Int}(R)$ is the largest weight, with each step comprising a combination of $R1$ and $R2$, which are connected by edges, where $(R) = k / |R|$ and k is a scale parameter that can be used to set the size of the edge component [23].

$$\text{Min}(\text{Int}(R1) + T(R1), \text{Int}(R2) + T(R2)) \quad (12)$$

Canny edge detection smoothens the image with a Gaussian filter used to calculate the value in the directional gradient and remove false edges [24] [25], and its formula method is presented as follows:

$$G(x + y) = \frac{1}{2x\delta^2} \exp\left(-\frac{x^2 + y^2}{2x\delta^2}\right) \quad (13)$$

Sobel is the most common edge detection with good performance results and typically used on images with detectable noise and increased margins [26]. The 3x3 convolution mask is usually used to detect gradients in the X

and Y directions [27]. Prewitt is easy to implement, using a 3x3 convolution mask. Its filter is a fast method for edge detection and is only suitable for soundless images with good contrast [28]

Roberts plays a role in eliminating noise due to its ability to calculate everything rapidly. Operator Robert contains a pair of 2x2 convolution masks. One of the masks is rotated 90 degrees. Laplacian edge detector provides a brief overview and mathematical description of the grayscale image with discontinuity between the two areas at varying gray levels. Edge detection method LoG contains a pair convolution 3x3. LoG is based on second-order derivatives, using the following formula [29]:

$$V^2 f = \frac{\delta^2 f}{\delta x^2} + \frac{\delta^2 f}{\delta y^2} \quad (14)$$

C. Image Labeling

This is a segmentation method used to measure and identify the object area in the image. Connected Component Labeling (CCL) produces a new image, which results from the process whose components are related. The labeling operation of the object area assigns a unique identity to all pixels with a value of 1 in the area. The results are the components of the extractable image object. Algorithm CCL can work on binary images using four or 8-connectivity methods [30].

D. Image Blob Detection

The blob detector is based on an area scale representation of the image. For a digital image, the smoothed scale is stacked into its representation. Different types of multiscale differential operators produce varying sizes of blobs in the domain of the scale space. One commonly used algorithm is the Mask Gaussian Multiscale, while the Operator Log generates a Gaussian kernel-based method. The other version is a method of Difference of Gaussians (DoG) and Determinant of Hessian (DoH) [15]. The conventional detection process included additional blob information such as color or texture. Descriptors are used to enhance the feature vector of the blobs to be extracted from the original image at all scales. Features used in blobs include color and texture,

with the original image feature vector extracted to ensure the blur does not affect the feature vector. The stability of this feature is indicated by the formula 15, where $f = [f1, f2, f3, \dots, fn]$ and $g = [g1, g2, g3, \dots, gn]$ correspond to the blobs and dist of the Euclidean distance [31].

$$S = \frac{1}{\text{dist}(f,g)} \quad (15)$$

LoG is given the input image $f(x,y)$, masked with Gaussian mask.

$$g(x,y,t) = \frac{1}{2\pi t} e^{-\frac{x^2+y^2}{2t}} \quad (16)$$

At a certain scale t is given a representation of the scale space $L(x,y;t) = g(x,y,t) * f(x,y)$, then the laplacian is applied to the operator.

$$\nabla^2 L = L_{xx} + L_{yy} \quad (17)$$

For Difference of Gaussians (DoG), from the L scale representation (x,y,t) :

$$\partial_t L = \frac{1}{2} \nabla^2 L \quad (18)$$

From the Laplacian formula $\nabla_{norm}^2 L(x,y;t)$ can also be calculated as Gaussian smoothing boundary difference between two images.

$$\nabla_{norm}^2 L(x,y;t) \approx \frac{t}{\Delta t} (L(x,y;t + \Delta t) - L(x,y;t)) \quad (19)$$

For Determinant of Hessian (DoH)

$$\det H_{norm} L = t^2 (L_{xx} L_{yy} - L_{xy}^2) \quad (20)$$

Where HL is the Hessian matrix

$$(x,y;t) = \underset{(x,y,t)}{\text{argmax local}} ((\det H_{norm} L)(x,y;t)) \quad (21)$$

The blob point (x,y) and the t -scale defined from the geometric differential operations lead to a blob descriptor that is covariant with rotation and capable of scaling in the image domain [32]. Feature matching is a process to determine points in the image area from the blob detector in each method used. Each point is encoded as a binary descriptor $D(i)$ [19].

$$Hd = \sum_{k=1}^z \text{XOR}(\beta D_k(i), \beta D_k(j)) \quad (22)$$

E. Internet of Things

Internet of things (IoT) is a technology widely used for data retrieval and processing. Its use for aquaculture makes hardware and software systems more sophisticated, thereby increasing crop yields. Device IoT can be used for monitoring weather data, livestock, site temperatures, early detection of disease in animals, and disturbances that cause animal death [33].

The applied intelligent system is a technology that plays an essential role in the field of aquaculture. Its architecture is used to acquire and process big data used for computing processes. The IoT architecture used in intelligent systems comprises several layers for processing. This method is proven to be able to effectively process big data with intelligent equipment and provide the information needed by companies or organizations in more detail [34].

Quality of Service (QoS) measures the services the network providers offer to users. The measurement parameters used include fault tolerance, energy efficiency, bandwidth, packet loss, delay, and throughput. QoS is used in IoT to measure the quality of data transmission and process

speed. This is because the QoS quality of wireless networks is very volatile due to the influence of the equipment [35].

III. RESULT AND DISCUSSION

This research was conducted using python 3, implemented on 3 digital images of underwater shrimp. The experiment started by converting the original image into a grayscale image using the thresholding process, as shown in Fig. 2 (a) [36]. The Structural Similarity Index Measure (SSIM) and Mean Squared Error (MSE) values for 3 images were generated from the results of the computational process [37]. The process results with the highest SSIM and the lowest MSE of the computational process at the threshold with a value of 0.01, as shown in Fig. 2 (b). This means the otsu algorithm's segmentation with a threshold value of 0.01 produces better structural similarities between grayscale and binary images, with high-quality pixel values. The error value calculated using MSE shows that the threshold value of 0.01 produces a low error. This means that the average value variance estimators of the grayscale and biner images produced are low.

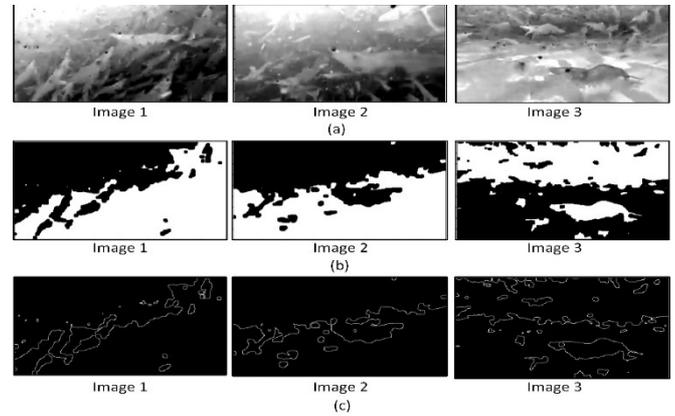


Fig. 2 (a) Grayscale image; (b) Thresholding image; (c) Edge detection of shrimp image data underwater

The next process uses image data with a threshold value of 0.01. The experiment was continued for the edge detection process on 3 contours of digital shrimp images underwater. The edge detection process uses 5 methods: Canny, Sobel, Prewitt, Roberts, and LoG. From these results, the value of Peak Signal to Noise Ratio (PSNR) and Mean Squared Error (MSE) for 3 images are determined [37]. Edge detection process indicates that the highest PSNR and the lowest MSE are found in the canny method with 3.6 and 2.06, respectively. This means the canny edge detection algorithm produces a high image quality signal ratio than the detected noise, as shown in Fig. 2 (c). Furthermore, the average error value between the binary image and the edge detection results produces the lowest close to zero.

The data was used to determine the result of the Canny edge detection process. The experiment was continued for the image contour process on 3 digital shrimp images, using the Region of Interest (ROI). Fig. 3 (a) shows the results of the image contour process consisting of several forms of contours in the form of a shrimp image.

The ROI contour data was used to determine the area of the shrimp image object [38]. This process identifies the pixel

value of the image as 1. The Intersection over Union (IOU) value was obtained for 3 images, the value of matching object detection in digital images [39]. The digital image labeling is described from the experimental results as shown in Fig. 3 (b). The computation result shows that the tenth digital shrimp image produces the highest IOU with a value of 0.61. Therefore, the prediction area with the ground truth image has high proximity with significant intersection values.

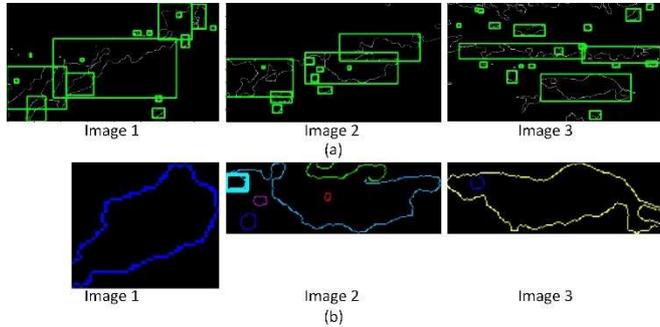


Fig. 3 (a) ROI Image contour; (b) Image labelling of image shrimp underwater

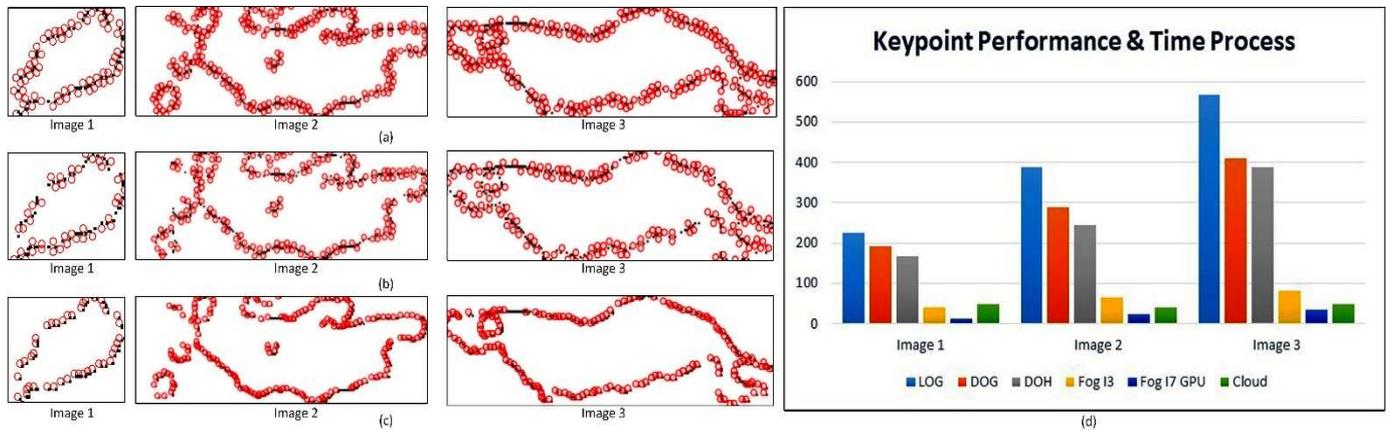


Fig. 4 (a) Blob detection using LoG; (b) Blob detection using DoG; (c) Blob detection using DoH; (d) Key performance blob detection and time process on server fog and server cloud

The three QoS parameters, namely throughput and packet loss for transmitting shrimp digital image data between underwater camera and fog server, are very good. Therefore, data communication on IoT infrastructure using underwater cameras and fog servers runs very well. The second QoS generated in this experiment obtained throughput, packet loss, and latency delay values of 832.25 kbps, 0%, and 7.25 ms, respectively. This running process at an internet speed of 10 Mbps and the statistics QoS were shown in Table 2.

The table 2 shows that the three QoS parameters, for transmitting shrimp digital image data between fog server and cloud server are at a very good level. This means that data communication on IoT infrastructure using fog server and cloud server runs adequately. The computing process uses IoT infrastructure, namely fog servers with I3 and I7 PC specifications equipped with GPUs and cloud servers [42]. The computing process on a fog server with I3 specification and without GPU takes longer than on a cloud server. Meanwhile, the computing process on a fog server with an I7 specification with GPU is quicker, as shown in Fig. 4 (d). Therefore, computing on fog and cloud servers with I7 GPU

The experiment further conducted in detecting blobs on digital shrimp images was to determine its texture area. The methods used are LoG as shown in Fig. 4 (a), DoG as shown in Fig. 4 (b), and DoH as shown in Fig. 4 (c) [15]. Its computational process result generates key point performance values for 3 images [40]. The graph illustrates that the LoG algorithm produces the highest key point performance in detecting the shrimp area with a value of 566. Therefore, the pixels in the shrimp area bordering the background are detected with a high value in the 2-dimensional image.

This research used IoT for data retrieval and transmission on wireless networks using QoS [41], which was measured in two trials. The first was data transmission underwater shrimp digital video from the Wi-Fi underwater camera to the fog server. Meanwhile, the second was data transmission underwater shrimp digital images from the fog to the cloud server. The first experiment led to a throughput value of 30.97 kbps, a packet loss of 0.01%, and a latency delay of 22.7 ms, as shown in Table 1.

specifications is able to reduce the density of data transmission processes for shrimp digital images.

TABLE I
VALUE QUALITY OF SERVICE SHRIMP DATA TRANSMISSION FROM UNDERWATER CAMERA TO FOG SERVER

QoS Parameters	Value	Index	Category
Throughput	30,97 kbps	4	Very good
Packet Loss	0.01%	4	Very good
Delay	22.7 ms	4	Very good

TABLE II
VALUE QUALITY OF SERVICE SHRIMP DATA TRANSMISSION FROM FOG SERVER TO CLOUD SERVER

QoS Parameters	Value	Index	Category
Throughput	832,25 kbps	4	Very good
Packet Loss	0%	4	Very good
Delay	7.25 ms	4	Very good

IV. CONCLUSION

This study investigated the use of IoT technology for underwater shrimp detection using blob detector with non-invasive methods. Among DOH, DOG and LOG algorithms

technique, the LOG algorithm produced the highest keypoint performance (566) compared to the DOH and DOG algorithm. The data communication on IoT infrastructure using fog and cloud server was running effectively, with throughput, packet loss and latency delay value of 30.97 kbps, 0.01% and 22.7 ms for underwater to fog server, and 832.25 kbps, 0%, and 7.25 ms for fog server to cloud server. This concludes that the LOG algorithm has higher pixels in the shrimp area bordering the background in the 2-dimensional image and get high key performance blob detection, the IoT infrastructure model using I7 specs, get very good level for three QoS parameters and quicker for speed computation process. Future research is recommended to increase the accuracy using more research data and the capturing the size of shrimp object. Using feature extraction and machine learning algorithms can also be another alternative for detecting the object and compare to the current result.

ACKNOWLEDGMENT

The authors are grateful to the Information Systems Doctoral Program at Diponegoro University for facilitating this research and Muria Kudus University's funding.

REFERENCES

- [1] X. Dong and V. Raghavan, "Trends in Food Science & Technology Recent advances of selected novel processing techniques on shrimp allergenicity: A review," *Trends Food Sci. Technol.*, vol. 124, no. April, pp. 334–344, 2022, doi: 10.1016/j.tifs.2022.04.024.
- [2] S. Ray *et al.*, "Role of shrimp farming in socio-economic elevation and professional satisfaction in coastal communities of Southern Bangladesh," *Aquac. Reports*, vol. 20, no. June 2020, p. 100708, 2021, doi: 10.1016/j.aqrep.2021.100708.
- [3] C. N. Udanor *et al.*, "An internet of things labelled dataset for aquaponics fish pond water quality monitoring system," *Data Br.*, vol. 43, p. 108400, 2022, doi: 10.1016/j.dib.2022.108400.
- [4] Z. Liu, X. Jia, and X. Xu, "Study of shrimp recognition methods using smart networks," *Comput. Electron. Agric.*, vol. 165, Oct. 2019, doi: 10.1016/j.compag.2019.104926.
- [5] J. C. Ovalle, C. Vilas, and L. T. Antelo, "On the use of deep learning for fish species recognition and quantification on board fishing vessels," *Mar. Policy*, vol. 139, no. August 2021, p. 105015, 2022, doi: 10.1016/j.marpol.2022.105015.
- [6] W. Xu *et al.*, "Shadow detection and removal in apple image segmentation under natural light conditions using an ultrametric contour map," *Biosyst. Eng.*, vol. 184, pp. 142–154, 2019, doi: 10.1016/j.biosystemseng.2019.06.016.
- [7] X. Wu, L. Bi, M. Fulham, D. Dagan, L. Zhou, and J. Kim, "Neurocomputing Unsupervised brain tumor segmentation using a symmetric-driven adversarial network," *Neurocomputing*, vol. 455, pp. 242–254, 2021, doi: 10.1016/j.neucom.2021.05.073.
- [8] M. Calkovský *et al.*, "Materials Characterization Comparison of segmentation algorithms for FIB-SEM tomography of porous polymers: Importance of image contrast for machine learning segmentation," vol. 171, no. June 2020, 2021, doi: 10.1016/j.matchar.2020.110806.
- [9] M. Garrett *et al.*, "Deep learning model for automatic contouring of cardiovascular substructures on radiotherapy planning CT images: Dosimetric validation and reader study based clinical acceptability testing," *Radiother. Oncol.*, vol. 165, pp. 52–59, 2021, doi: 10.1016/j.radonc.2021.10.008.
- [10] Q. Hu *et al.*, "A method for measuring ice thickness of wind turbine blades based on edge detection," *Cold Reg. Sci. Technol.*, vol. 192, no. March, p. 103398, 2021, doi: 10.1016/j.coldregions.2021.103398.
- [11] G. Wang, C. Lopez-Molina, and B. De Baets, "Automated blob detection using iterative Laplacian of Gaussian filtering and unilateral second-order Gaussian kernels," *Digit. Signal Process. A Rev. J.*, vol. 96, p. 102592, 2020, doi: 10.1016/j.dsp.2019.102592.
- [12] M. M. Rashid, A. A. Nayan, M. O. Rahman, S. A. Simi, J. Saha, and M. G. Kibria, "IoT based Smart Water Quality Prediction for Biofloc Aquaculture," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 6, pp. 56–62, 2021, doi: 10.14569/IJACSA.2021.0120608.
- [13] J. Singh, P. Singh, and S. Singh, "Journal of Parallel and Distributed Computing Fog computing : A taxonomy , systematic review , current trends and research challenges," *J. Parallel Distrib. Comput.*, vol. 157, pp. 56–85, 2021, doi: 10.1016/j.jpdc.2021.06.005.
- [14] S. Suryono, A. Khuriati, and T. Mantoro, "A fuzzy rule-based fog–cloud computing for solar panel disturbance investigation," *Cogent Eng.*, vol. 6, no. 1, pp. 1–19, 2019, doi: 10.1080/23311916.2019.1624287.
- [15] O. Li and P. lang Shui, "Subpixel blob localization and shape estimation by gradient search in parameter space of anisotropic Gaussian kernels," *Signal Processing*, vol. 171, 2020, doi: 10.1016/j.sigpro.2020.107495.
- [16] K. Malinowski and K. Saeed, "An iris segmentation using harmony search algorithm and fast circle fitting with blob detection," *Biocybern. Biomed. Eng.*, vol. 42, no. 1, pp. 391–403, 2022, doi: 10.1016/j.bbe.2022.02.010.
- [17] J. D. Trivedi, S. D. Mandalapu, and D. H. Dave, "Vision-based real-time vehicle detection and vehicle speed measurement using morphology and binary logical operation," *J. Ind. Inf. Integr.*, vol. 27, no. August 2021, p. 100280, 2022, doi: 10.1016/j.jii.2021.100280.
- [18] D. N. Triwibowo, E. Utami, and S. Sukoco, "Analisis BLOB Detection Pada Pendeteksian dan Perhitungan Kendaraan di Jalan Tol," *Inspir. J. Teknol. Inf. dan Komun.*, vol. 10, no. 1, p. 1, 2020, doi: 10.35585/inspir.v10i1.2532.
- [19] P. Niyishaka and C. Bhagvati, "Copy-move forgery detection using image blobs and BRISK feature," *Multimed. Tools Appl.*, no. September, 2020, doi: 10.1007/s11042-020-09225-6.
- [20] C. Anitha and R. M. S. Kumar, "GEVE : A generative adversarial network for extremely dark image / video enhancement," *Pattern Recognit. Lett.*, vol. 155, pp. 159–164, 2022, doi: 10.1016/j.patrec.2021.10.030.
- [21] A. F. A. Fernandes *et al.*, "Deep Learning image segmentation for extraction of fish body measurements and prediction of body weight and carcass traits in Nile tilapia," *Comput. Electron. Agric.*, vol. 170, no. November 2019, p. 105274, 2020, doi: 10.1016/j.compag.2020.105274.
- [22] Y. Song, S. Ren, Y. Lu, X. Fu, and K. K. L. Wong, "Computer Methods and Programs in Biomedicine Deep learning-based automatic segmentation of images in cardiac radiography: A promising challenge," vol. 220, 2022, doi: 10.1016/j.cmpb.2022.106821.
- [23] G. Li, S. Ma, H. Li, F. Liu, and L. Lin, "Terrace compression method " and its application in heterogeneity contour detection of transmission images," vol. 514, no. December 2021, pp. 1–8, 2022, doi: 10.1016/j.optcom.2022.128114.
- [24] X. Hu and Y. Wang, "Catena Monitoring coastline variations in the Pearl River Estuary from 1978 to 2018 by integrating Canny edge detection and Otsu methods using long time series Landsat dataset," vol. 209, no. November 2021, 2022.
- [25] R. Priyadharsini and T. S. Sharmila, "Object Detection in Underwater Acoustic Images Using Edge Based Segmentation Method," *Procedia Comput. Sci.*, vol. 165, pp. 759–765, 2019, doi: 10.1016/j.procs.2020.01.015.
- [26] P. Gao, Y. Song, M. Song, P. Qian, and Y. Su, "Scripta Materialia Extract nanoporous gold ligaments from SEM images by combining fully convolutional network and Sobel operator edge detection algorithm," *Scr. Mater.*, vol. 213, no. November 2021, p. 114627, 2022, doi: 10.1016/j.scriptamat.2022.114627.
- [27] R. A. A. S and S. Gopalan, "ScienceDirect Comparative Comparative Analysis Analysis of Eight Eight Direction Direction Sobel Sobel Edge Edge Detection Detection Algorithm for Brain MRI Images Algorithm for Brain Tumor MRI Images," *Procedia Comput. Sci.*, vol. 201, pp. 487–494, 2022, doi: 10.1016/j.procs.2022.03.063.
- [28] M. Huang, Y. Liu, and Y. Yang, "Edge detection of ore and rock on the surface of explosion pile based on improved Canny operator," *Alexandria Eng. J.*, vol. 61, no. 12, pp. 10769–10777, 2022, doi: 10.1016/j.aej.2022.04.019.
- [29] Y. Xu, B. Sun, X. Yan, J. Hu, and M. Chen, "Multi-focus image fusion using learning based matting with sum of the Gaussian-based modified Laplacian," *Digit. Signal Process.*, vol. 106, p. 102821, 2020, doi: 10.1016/j.dsp.2020.102821.
- [30] D. Zhang, H. Ma, and L. Pan, "A gamma-signal-regulated connected components labeling algorithm," *Pattern Recognit.*, vol. 91, pp. 281–290, 2019, doi: 10.1016/j.patcog.2019.02.022.
- [31] C. Ricotta and S. Pavoine, "A new parametric measure of functional dissimilarity : Bridging the gap between the Bray-Curtis dissimilarity

- and the Euclidean distance,” *Ecol. Modell.*, vol. 466, no. December 2021, p. 109880, 2022, doi: 10.1016/j.ecolmodel.2022.109880.
- [32] S. Wang, T. B. Eldred, J. G. Smith, and W. Gao, “Ultramicroscopy AutoDisk : Automated diffraction processing and strain mapping in,” *Ultramicroscopy*, vol. 236, no. March, p. 113513, 2022, doi: 10.1016/j.ultramic.2022.113513.
- [33] M. Zhang, X. Wang, H. Feng, Q. Huang, X. Xiao, and X. Zhang, “Wearable Internet of Things enabled precision livestock farming in smart farms: A review of technical solutions for precise perception, biocompatibility, and sustainability monitoring,” *J. Clean. Prod.*, vol. 312, no. May, p. 127712, 2021, doi: 10.1016/j.jclepro.2021.127712.
- [34] J. Wan, J. Li, Q. Hua, A. Celesti, and Z. Wang, “Intelligent equipment design assisted by Cognitive Internet of Things and industrial big data,” *Neural Comput. Appl.*, vol. 32, no. 9, pp. 4463–4472, May 2020, doi: 10.1007/s00521-018-3725-5.
- [35] M. I. Ghozali, W. H. Sugiharto, H. Susanto, M. A. Budihardjo, and S. Suryono, “Measurement performance quality of services (QoS) to optimizing on wireless sensor network topology for water pollution monitoring system,” *J. Phys. Conf. Ser.*, vol. 1943, no. 1, 2021, doi: 10.1088/1742-6596/1943/1/012019.
- [36] Y. Han, T. Song, J. Feng, and Y. Xie, “Grayscale-inversion and rotation invariant image description with sorted LBP features,” *Signal Process. Image Commun.*, vol. 99, no. June, p. 116491, 2021, doi: 10.1016/j.image.2021.116491.
- [37] U. Sara, M. Akter, and M. S. Uddin, “Image Quality Assessment through FSIM , SSIM , MSE and PSNR — A Comparative Study,” pp. 8–18, 2019, doi: 10.4236/jcc.2019.73002.
- [38] N. A. Gard, C. Bunker, and A. Yilmaz, “A spacetime model for one-shot active contour extraction scheme for human detection in image sequences,” *Comput. Vis. Image Underst.*, vol. 202, no. September 2020, p. 103113, 2021, doi: 10.1016/j.cviu.2020.103113.
- [39] S. Wu, J. Yang, X. Wang, and X. Li, “IoU-Balanced loss functions for single-stage object detection,” *Pattern Recognit. Lett.*, vol. 156, pp. 96–103, 2022, doi: 10.1016/j.patrec.2022.01.021.
- [40] C. Zheng *et al.*, “A mango picking vision algorithm on instance segmentation and key point detection from RGB images in an open orchard,” *Biosyst. Eng.*, vol. 206, pp. 32–54, 2021, doi: 10.1016/j.biosystemseng.2021.03.012.
- [41] S. Pallewatta, V. Kostakos, and R. Buyya, “QoS-aware placement of microservices-based IoT applications in Fog computing environments,” *Futur. Gener. Comput. Syst.*, vol. 131, pp. 121–136, 2022, doi: 10.1016/j.future.2022.01.012.
- [42] N. Ben Salah and N. Bellamine Ben Saoud, “Adaptive data placement in the Fog infrastructure of IoT applications with dynamic changes,” *Simul. Model. Pract. Theory*, vol. 119, no. April, p. 102557, 2022, doi: 10.1016/j.simpat.2022.102557.