

Portable Device for Ornamental Shrimp Counting Using Unsupervised Machine Learning

Chi-Tsai Yeh* and Ming-Sheng Ling

Changzhou College of Information Technology,
No. 22 Mingxin Middle Road, Science Education City, Changzhou City, Jiangsu Province 213164, China

(Received December 29, 2020; accepted May 26, 2021)

Keywords: image segmentation, unsupervised learning, overlapping, counting, portable device

With the rapid development of emerging technologies, intelligent agriculture is incorporating techniques such as the Internet of Things, big data, cloud computing, artificial intelligence, blockchains, and fifth-generation mobile communication to improve work efficiency, prevent various disasters, and change the sales mode of agricultural products. Ornamental fishery is a part of agriculture and accounts for a significant proportion of commercial trade. This paper introduces image processing technology to help ornamental fisheries calculate the number of shrimps quickly. To solve the problem of overlapping live shrimps when counting, K-means unsupervised machine learning is adopted to determine the area of one shrimp. In addition, the proposed method using unsupervised machine learning is able to count different types of shrimp with high accuracy, such as crystal red shrimps, fire red shrimps, and Takashi Amano shrimps. We also analyze two background subtraction techniques, hue/saturation/value (HSV) histogram-based detection and Sobel edge detection, to compare the accuracy and calculation time of this application.

1. Introduction

The size of the trade in global ornamental fish has steadily increased in recent decades. There were only 28 countries exporting ornamental fish in 1976, compared with more than 125 countries at present. Moreover, the global export market for ornamental fish almost doubled from US\$ 177.7 million in 2000 to 337.7 million in 2016.^(1,2)

The global ornamental fish industry is worth US\$ 15 billion, and more than 2 billion live ornamental aquatic animals are traded per year, making it necessary to consider how to deal with large numbers of live ornamental aquatic animals. Until now, the calculation of numbers of live fish or shrimps has mainly been carried out manually, which has many weaknesses including subjectivity, slowness, low accuracy, worker fatigue, and even the injury and death of animals during the calculation.⁽³⁾

The development of computer vision technology with a tiny camera for image capture for counting shrimps is a strong possibility. In this study, we propose a method of counting ornamental shrimps that solves the above shortcomings of manual counting. The proposed

*Corresponding author: e-mail: yehchitsai@czcit.edu.cn
<https://doi.org/10.18494/SAM.2021.3240>

method employs a portable smart device consisting of a Raspberry Pi computer attached to a camera sensor and has the expected advantages of avoiding the need to physically touch the shrimps, rapid calculation, low cost, and portability.

The remainder of this paper is organized as follows. Section 2 describes related work on aquaculture using computer vision techniques. Section 3 shows the hardware and software elements of the proposed system. Section 4 explains the process flow used with our proposed method. Section 5 reports experimental results obtained with the implemented system. Finally, the article is concluded in Sect. 6.

2. Related Work

Zion summarized the applications of computer vision technology in aquaculture into five categories. His review described the state of the art and focused on needs in all stages of the aquaculture process from hatcheries to harvesting. These applications included counting, size measurement and mass estimation, gender detection and quality inspection, species and stock identification, and the monitoring of welfare and behavior. Optical methods to count objects were the most common approach. The counted objects included eggs, larvae, fry, and fish at various stages of growth.⁽⁴⁾ Silvério categorized these computer vision systems in aquaculture. These systems have the following common steps:⁽⁵⁾

- Capturing images
- Extracting features
- Detecting blobs
- Counting based on the blobs detected.

Most computer vision applications that perform image recognition require the above four steps. The image processing techniques are mostly used in steps 2 and 3, i.e., extracting features such as boundaries or a given color range and detecting blobs to segment the image. In this paper, we present a state-of-the-art application of computer vision for counting in aquaculture. A comprehensive summary of related studies for object identification or counting in aquaculture from 1995 to 2018 is included in Table 1.

Table 1
Related literature on the application of computer vision for object identification or counting in aquaculture.

Authors	Applications	Targets	Results
Newbury <i>et al.</i> , 1995 ⁽⁶⁾	backpropagation network	artificial fish	94% accuracy, overlapping
Huang, 2002 ⁽⁷⁾	HSV background subtraction	fluorescent lamp fry	error ranging between 0.9 and 32%, non-overlapping
Friedland <i>et al.</i> , 2005 ⁽⁸⁾	erosion–dilation filter	American shad eggs	FN classification error of 1%, non-overlapping
Zion, 2012 ⁽⁴⁾	shape model	fish fry	98% accuracy, overlapping
Flores <i>et al.</i> , 2008 ⁽⁹⁾	adaptive threshold and fast Fourier transform (FFT)	Peruvian scallop larvae	accuracy of 95% in 2 min, overlapping
Khantuwan and Khiripet, 2012 ⁽¹⁰⁾	adaptive threshold and co-occurrence color of histogram	shrimp larvae	accuracy of 97% in 1 min, overlapping
Coronel <i>et al.</i> , 2018 ⁽¹¹⁾	local normalization filter and iterative selection threshold	fish fingerlings	precision, recall, F measure of 99.80, 97.90, 98.83%, respectively, in < 1 s, non-overlapping

Newbury *et al.* used artificial fish to simulate the process of fish calculation, especially considering the fact that fish often overlap each other. Then they applied a backpropagation algorithm and achieved 94% accuracy when estimating the number of fish.⁽⁶⁾ Huang adopted the value of intensity (V) in the hue/saturation/value (HSV) color space to filter the target image, then performed thresholding for image segmentation, and calculated the area of fluorescent lamp fry based on the obtained foreground. The depth of each fry in an aquarium is different, resulting in fish of the same size having different areas in an image. In addition, the overlapping of fry was not considered, so errors of between 0.9 and 32% were obtained.⁽⁷⁾ In 2005, Friedland *et al.* used the mathematical morphological dilation–erosion algorithm to eliminate touching eggs, and then performed seven measurements to identify American shad eggs and debris in the water, enabling them to count the number of American shad eggs. The false negative (FN) rate in the identification, i.e., eggs judged to be debris, was 1%.⁽⁸⁾ Zion *et al.* developed image processing algorithms using shape models of various species and detected the number of overlapping fish fry with 98% accuracy.⁽⁴⁾

Flores *et al.* applied image processing technology to calculate the number of Peruvian scallop seedlings. The processing flow was as follows. Step 1, place the samples on a Sedgewick Rafter cell slide under a microscope; step 2, capture an image; step 3, find the region of interest; step 4, filter the image using Laplacian and median filters to enhance the image and as preprocessing for the next step; step 5, use adaptive thresholding for binarization; step 6, divide the targets into A with one seedling and B with two or more overlapping seedlings; step 7, find the area of a scallop seedling through the FFT of the histogram; step 8, calculate the total number of scallop seedlings. The experimental results showed that the calculation can be completed in 2 min per image with 95% accuracy.⁽⁹⁾

Khantawan and Khiripet proposed an image processing method that automatically calculates the number of shrimp larvae using a co-occurrence color histogram template comparison technique. The whole algorithm can be divided into three parts: image enhancement, separation of juvenile shrimps from the background, and classification. The accuracy of automatic counting was experimentally shown to be 97%.⁽¹⁰⁾ Compared with manual counting, the error between automatic counting and manual counting was between -1.5 and 6% , and the average error was 3.25% . They used MATLAB 2010a run on an Intel Core™ i5 laptop, and it took an average of 1 min to analyze an image of 100 juvenile shrimps. Coronel *et al.* proposed a set of efficient image filtering and segmentation methods that used two image processing technologies, a local normalization filter, and an iterative selection threshold to perform background removal. In terms of counting accuracy, the combination of these two methods obtained an average precision, recall, and F measure of as high as 99.80, 97.90, and 98.83%, respectively.⁽¹¹⁾

Table 2 shows the related studies on object recognition and counting that apply computer vision technology. Some of the objects included are as small as microbes, colonies, blood cells, rotifers, and *Aedes aegypti* larvae, as well as field insects, pests, and mice. Barbedo and Fuad *et al.* found in their review that color space conversion is mainly used in the calculation of these targets.^(12,13) Alver *et al.*, Mohamed and Far, Barbedo, and Das *et al.* used the conversion between the color models of RGB, cyan/magenta/yellow/key, International Commission on Illumination lab, and HSV. Then, they applied filter technologies such as Laplacian and median

Table 2

Related literature on the application of computer vision for object identification or counting in other domains.

Authors	Applications	Targets	Results
Alver <i>et al.</i> , 2005 ⁽¹⁴⁾	thresholding	rotifers	accuracy could be adjusted according to requirements
Pistori <i>et al.</i> , 2010 ⁽¹⁸⁾	auto-adjustable observation	mice and <i>Aedes aegypti</i> larvae	85% mean classification rate
Lu and Qin, 2010 ⁽²⁰⁾	improve K-means via hidden Markov model	insects in stored grain	recognition accuracy of 98% with normal pattern and 87% with lateral pattern
Mohamed and Far, 2012 ⁽¹⁵⁾	Otsu thresholding	white blood cell nuclei	80.6% accuracy
Barbedo, 2012 ⁽¹²⁾	median filter and thresholding	micro-organisms and colonies	correct identification of more than 90% of cases and overall deviation of 8%
Qing <i>et al.</i> , 2014 ⁽²¹⁾	AdaBoost classifier, support vector machine, thresholding	white-backed planthoppers	85.2% correct detection and 9.6% false detection
Barbedo, 2014 ⁽¹⁶⁾	color models transformation, thresholding	adult whiteflies	counting result within deviation of 1% from 748 images
Das <i>et al.</i> , 2014 ⁽¹⁷⁾	edge detection	blood cells	85% identification accuracy
Martin <i>et al.</i> , 2015 ⁽²³⁾	extended region growing detection	pests	identification and counting with 90% accuracy
Fuad <i>et al.</i> , 2017 ⁽¹³⁾	color models transformation, thresholding	<i>Aedes aegypti</i> larvae	review of techniques

filters to enhance the image, and they removed the background using an appropriate thresholding technology.^(14–17) Other researchers such as Pistori *et al.*, Lu and Qin, and Qing *et al.* adopted emerging methods to count and identify target images. The technologies include artificial neural networks, machine learning, and sample comparison techniques.^(18–22)

3. Materials and Methods

To achieve our goals of cost-effectiveness and portability, the entire proposed system is based on Raspberry Pi with a Raspberry Pi camera for calculation and image acquisition.⁽²⁴⁾ Raspberry Pi is a tiny computer developed by Raspberry Pi Foundation. The Raspberry Pi Zero W model is mainly dependent on the Broadcom BCM2835 SoC with a core architecture of a 32-bit ARM11 processor and a basic size of $65 \times 30 \text{ mm}^2$. It has an operating frequency of 1 GHz, a memory of 512 MB SDRAM, two USB-2.0 ports, Wi-Fi and Bluetooth support, a mini high-definition multimedia interface port, a camera serial interface connector, and a mobile industry processor with a micro secure digital memory card slot. The Raspberry Pi camera module has an OmniVision OV5647 five-megapixel sensor. The camera module can be used to take high-definition video as well as still photographs. There are numerous third-party libraries built for it, including the Picamera Python library.

Figure 1 shows the entire hardware device. The lower left is the Raspberry Pi camera, and the upper right is the Raspberry Pi Zero W. The two devices are installed above and below the cover,

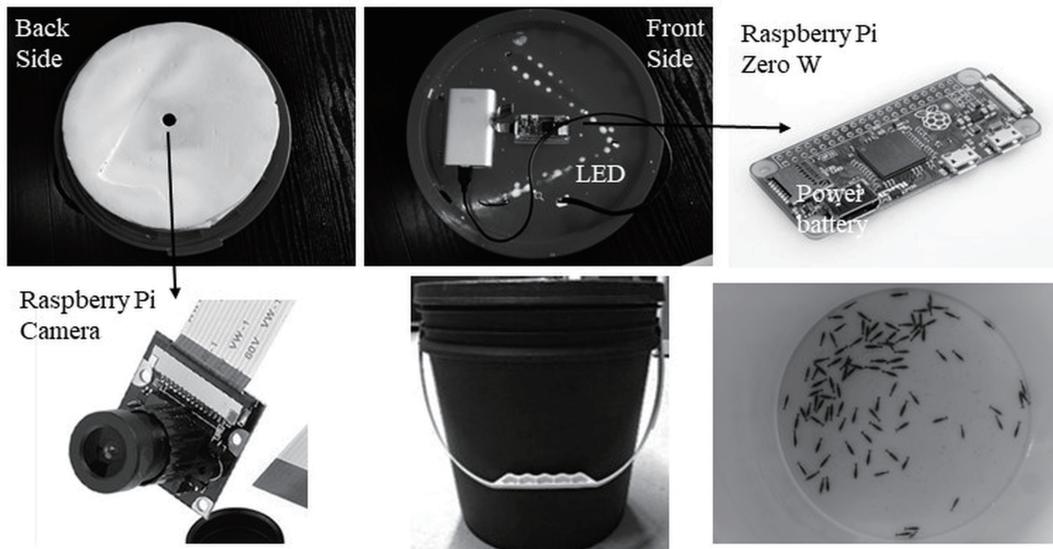


Fig. 1. Portable automated shrimp-counting device.

respectively. As shown in the upper left of Fig. 1, the back of the lid is used to capture the picture, and an LED is used to provide the light source. As shown in the upper middle of Fig. 1, a battery is installed on the front of the lid to power the operation of the whole system. The lower right of Fig. 1 shows the inside of the plastic bucket containing shrimps, and the lower middle of Fig. 1 shows the appearance of the bucket.

Table 3 lists all the software and hardware elements. The application installed in the mobile devices provides the user interface and controls the Raspberry Pi to perform the process of shrimp calculation. Therefore, the ionic framework is used to deploy the application in various mobile phones, including Android and iPhone. Raspberry Pi Zero W is the main computing unit, which calls the OpenCV library for image processing and the scikit-learn library for machine learning calculation. The Apache HTTP server allows users to interact with the Raspberry Pi.

Table 3
Hardware and software elements of portable automated shrimp-counting device.

Hardware	Software	Description
Mobile device	Hypertext Markup Language, cascading style sheets, JavaScript	web technologies provide user interface (UI)
	Ionic framework	open-source UI toolkit for building high-performance, high-quality mobile and desktop apps
	Android/iOS	mobile operating system
Raspberry Pi Zero W	OpenCV	free library aimed at computer vision
	scikit-learn	free library aimed at machine learning
	Python	programming language
	Apache Hypertext Transfer Protocol (HTTP) Server	free and open-source cross-platform web server providing access to images
Raspberry Pi Camera	Picamera Python library	captures images or video

4. Proposed Method

Figure 2 shows the flowchart of image processing for the portable automated shrimp-counting system. Each image is subjected to processing techniques such as resizing, grayscale conversion, feature extraction, and thresholding. The program in the Raspberry Pi computes the areas of the extracted contours and stores them in an array. Then, the program determines whether each contour contains noise or shrimps depending on its area. Considering the overlapping of shrimps, it divides whole contours into two clusters. One cluster is for a single shrimp, and the other cluster is for more than one shrimp.

Considering practical use, we use two different methods to extract features. Sobel edge detection uses a pair of 3×3 convolution masks, one to estimate the gradient in the x -direction and the other to estimate the gradient in the y -direction. The Sobel algorithm effectively highlights the shrimps via edge detection.⁽²⁵⁾ The other method, HSV filtering, can extract a specific range as a feature.⁽²⁶⁾ Figure 3 shows the results of using Sobel edge detection and HSV filtering after thresholding. In terms of noise and area, the HSV filtering method is clearly better than the Sobel method. However, HSV filtering is impossible to implement for shrimps with multiple colors, such as crystal red shrimps.

The K-means clustering algorithm is an unsupervised machine learning algorithm. The aim of this algorithm is to divide S points into K ($\leq S$) clusters as shown in Eq. (1). The within-cluster sum of squares, $\|x - \mu_i\|^2$, which is the Euclidean distance, is minimized. The cluster center μ_i is the mean value of all points x in S_i . The K-means clustering method simply divides all the blob

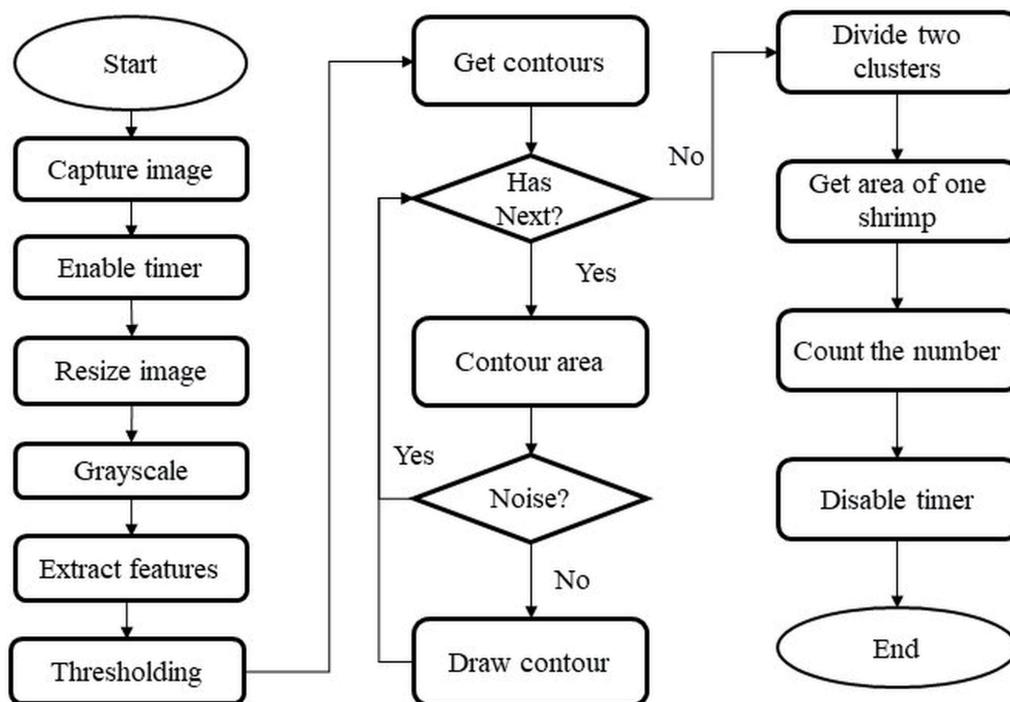


Fig. 2. Image process flow for portable automated shrimp-counting system.

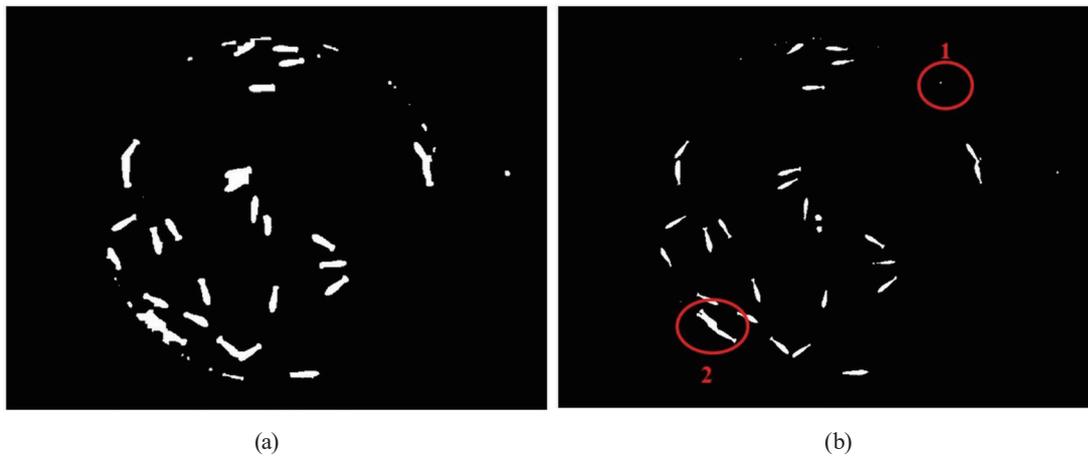


Fig. 3. (Color online) Results of (a) Sobel edge detection and (b) HSV filter.

contours into two clusters. The cluster with the larger number, oneShrimp, is identified as a shrimp, and the other cluster, moreShrimps, is identified as multiple shrimps. For the oneShrimp cluster, the area of a shrimp is determined by the median value in this cluster. The program calculates the rounding integer operation for each element of the moreShrimps cluster.

$$\arg_S \min \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2 \quad (1)$$

5. Experimental Results and Discussion

We apply the HSV filter to calculate the number of fire red shrimps, and perform Sobel edge detection to calculate the numbers of fire red shrimps, crystal shrimps, and Takashi Amano shrimps. The experimental results in Table 4 show that the system based on Raspberry Pi can count shrimps accurately in 1 s. The average accuracy rate of 100 runs was maintained at above 96%, except for fire red shrimps counted by the Sobel method, and the standard deviation was low. According to the results of analyzing the counting accuracy, there are two conditions of the calculation results: shrimps were undercounted or overcounted.

Table 4
Results of using different feature extraction methods.

Actual shrimp # Features	Average shrimp #	Standard deviation	Accuracy rate (%)	Calculation time (ms)
Fire red (25) HSV filter	25.8	3.14	96.80 (+3.20)	600.7
Fire red (25) Sobel	31.35	1.47	74.60 (+25.40)	681.3
Crystal red (25) Sobel	25.96	1.36	96.16 (+3.84)	681.1
Takashi Amano (25) Sobel	25.45	1.68	98.20 (+1.80)	654.9

The undercounted conditions are as follows:

- Situation 1 - Direct neglect: (1) The feature of the target shrimps is not obvious enough to be extracted. For example, the color of the shrimp is not within the specified color space, or the edge is unable to be detected clearly. (2) The filtered area is too small to be considered as noise, as shown in red circle 1 of Fig. 3(b).
- Situation 2 - Indirect neglect: There is too much overlap, as shown in red circle 2 of Fig. 3(b). Actually, there are three shrimps in that blob, but two shrimps completely overlap except for their tails. On the basis of the area calculation, we divide the blob area by the area of a shrimp. In the case of rounding, the result shows two shrimps instead of three shrimps.

The overcounted condition: Too many shrimps are so close together that the area of the blob contour includes the space between the shrimps.

The results of using different feature extraction methods are shown in Table 4. This system exhibits satisfactory speed (< 700 ms) and high accuracy. The average accuracy rate of 100 runs was maintained at above 96% in our experiment. The number in parentheses in the first column of Table 4 indicates the number of the shrimps. The number in parentheses in the fourth column of Table 4 indicates the direction of accuracy rate. A positive value indicates overcounting, and a negative value indicates undercounting. The low standard deviation indicates the high quality of the programming in terms of performing stable calculations. Furthermore, the different extraction methods have different accuracies. The HSV filter is suitable for fire red shrimps, whereas Sobel edge detection has poor accuracy as shown in Table 4.

6. Conclusions

In this study, we applied image segmentation, edge detection, and unsupervised machine learning methods to the Raspberry Pi Zero W. The proposed cost-effective and portable device increases the practicality of a portable automated shrimp-counting system. This system exhibits satisfactory speed to complete the calculation in one second and high accuracy at 96%. The experimental results show that different extraction methods have different accuracies on different species of shrimps. Therefore, we suggested that different extraction methods should be used for different species of shrimps to keep the accuracy high.

Acknowledgments

Special thanks go to Creative Way Global Corporation for providing us with different species of shrimps and for financial support. This work is supported by the Industrial Big Data and Intelligent Engineering Technology R&D Center, the Provincial Engineering Research Center of Jiangsu, the Innovation and Application of Jiangsu Small Business Industrial Internet Engineering Research Center, the General Program of Natural Science Research in Jiangsu University, Research on Key Technologies of Disaster Tolerant Mobile Data Collection for Underwater Acoustic Sensor Network (19KJB520023), and the Smart Education Engineering Center (CXPT201701G).

References

- 1 S.-B. Satam, N.-H. Sawant, M.-M. Ghughuskar, V.-D. Sahastrabuddhe, V.-V. Naik, A.-U. Pagarkar, and A.-N. Sawant: *Adv. Agric. Res. Technol. J.* **2** (2018) 193. http://isat.org/vol-ii-issue-ii-july-2018/II_AARJ_II_2_Satam%20et%20al_193-197.pdf
- 2 K. Raja, P. Aanand, S. Padmavathy, and J.-S. Sampathkumar: *Int. J. Fisheries Aquatic Stud.* **7** (2019) 06. <https://www.fisheriesjournal.com/archives/2019/vol7issue2/PartA/7-1-48-550.pdf>
- 3 M. Solahudin, W. Slamet, and A.-S. Dwi: *Earth Environ. Sci.* **147** (2018) 012014. <http://doi.org/10.1088/1755-1315/147/1/012014>
- 4 B. Zion: *Comput. Electron. Agric.* **88** (2012) 125. <https://doi.org/10.1016/j.compag.2012.07.010>
- 5 Automatic Fish Counting in Aquariums: https://fenix.tecnico.ulisboa.pt/downloadFile/1689244997255462/resumo_65359.pdf (accessed April 2020).
- 6 P.-F. Newbury, P.-F. Culverhouse, and D.-A. Pilgrim: *Aquaculture* **133** (1995) 45. [https://doi.org/10.1016/0044-8486\(95\)00003-K](https://doi.org/10.1016/0044-8486(95)00003-K)
- 7 C.-H. Huang: A Simple Design of Automatic Counting System for Fish Larvae (National Sun Yat-sen University, Kaohsiung, 2002) Chap. 2. <https://hdl.handle.net/11296/6erp85>
- 8 K.-D. Friedland, D. Ama-Abasi, M. Manning, L. Clarke, G. Kligys, and R.-C. Chambers: *J. Sea Res.* **54** (2005) 307. <https://doi.org/10.1016/j.seares.2005.06.002>
- 9 A. Flores, P. Crisostomo, and J. Lopez: *Proc. 2008 IEEE 7th Int. Caribbean Conf. Devices, Circuits and Systems (IEEE, 2008)* 1–4. <https://doi.org/10.1109/ICDCS.2008.4542660>
- 10 W. Khantuwan, and N. Khiripet: *Proc. 2012 IEEE 9th Int. Conf. Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (IEEE, 2012)* 1–4. <https://doi.org/10.1109/ECTICon.2012.6254280>
- 11 L. Coronel, W. Badoy, and C. Namoco: *Int. Arab. J. Inf. Tech.* **15** (2018) 708. <https://iajit.org/PDF/July%202018,%20No.%204/11042.pdf>
- 12 J.-G.-A. Barbedo: *Proc. 2012 IEEE 12th Computational Science and Its Applications (IEEE, 2012)* 83–87. <http://dx.doi.org/10.1109/ICCSA.2012.23>
- 13 M.-A.-M. Fuad, M.-R. Ab Ghani, R. Ghazali, M.-F. Sulaima, M.-H. Jali, T. Sutikno, and Z. Jano: *Telkomnika* **15** (2017) 1199. <https://doi.org/10.12928/telkomnika.v15i3.6422>
- 14 M.-O. Alver, T. Tennøy, J.-A. Alfredsen, and G. Øie: *Aquacult. Eng.* **36** (2007) 115. <https://doi.org/10.1016/j.aquaeng.2006.09.002>
- 15 M. Mohamed and B. Far: *Proc. 2012 IEEE 14th e-Health Networking, Applications and Services (IEEE, 2012)* 202–207. <https://doi.org/10.1109/HealthCom.2012.6379408>
- 16 J.-G.-A. Barbedo: *J. Asia-Pac. Entomol.* **17** (2014) 685. <https://doi.org/10.1016/j.aspen.2014.06.014>
- 17 B.-K. Das, K.-K. Jha, and H.-S. Dutta: *Proc. 2014 IEEE Intelligent Computing Applications (IEEE, 2014)* 208–212. <https://doi.org/10.1109/ICICA.2014.52>
- 18 H. Pistori, V.-V.-V.-A. Odakura, J.-B.-O. Monteiro, W.-N. Gonçalves, A.-R. Roel, J. de Andrade Silva, and B.-B. Machado: *Pattern Recognit. Lett.* **31** (2010) 337. <https://doi.org/10.1016/j.patrec.2009.05.015>
- 19 P.-L.-F. Albuquerque, V. Garcia, A.-D.-S.-O. Junior, T. Lewandowski, C. Detweiler, A.-B. Gonçalves, and H. Pistori: *Comput. Electron. Agric.* **167** (2019) 105015. <https://doi.org/10.1016/j.compag.2019.105015>
- 20 Y. Lu and S. Qin: *Proc. 2010 IEEE Electrical and Control Engineering (IEEE, 2010)* 1997–2000. <https://doi.org/10.1109/iCECE.2010.491>
- 21 Y. Qing, D.-X. Xian, Q.-J. Liu, B.-J. Yang, G.-Q. Diao, and J. Tang: *J. Integr. Agric.* **13** (2014) 1736. [https://doi.org/10.1016/S2095-3119\(14\)60799-1](https://doi.org/10.1016/S2095-3119(14)60799-1)
- 22 Y. Qing, G.-T. Chen, Z. Wang, C. Zhang, B.-J. Yang, and J. Tang: *J. Integr. Agric.* **16** (2017) 1547. [https://doi.org/10.1016/S2095-3119\(16\)61497-1](https://doi.org/10.1016/S2095-3119(16)61497-1)
- 23 A. Martin, D. Sathish, C. Balachander, T. Hariprasath and G. Krishnamoorthi: *Proc. 2015 IEEE 2nd Int. Conf. Electronics and Communication Systems (IEEE, 2015)* 1229–1234. <https://doi.org/10.1109/ECS.2015.7124779>
- 24 L. Ujjainiya, and M.-K. Chakravarthi: *ARPN J. Eng. Appl. Sci.* **10** (2015) 3001. http://www.arpnjournals.com/jeas/research_papers/rp_2015/jeas_0415_1869.pdf
- 25 O.-R. Vincent and O. Folorunso: *Proc. 2009 Informing Science & IT Education Conf. (Informing Science Institute Publishing, 2009)* 97–107. <https://doi.org/10.28945/3351>
- 26 C.-T. Yeh and M.-C. Chen: *Microsyst. Technol.* (2019) 1. <https://doi.org/10.1007/s00542-019-04570-5>

About the Authors

Chi-Tsai Yeh received his B.S. degree from National Cheng Kung University in 1995, his M.S. degree from National Sun Yat-Sen University in 1997, and his Ph.D. degree from National Kaohsiung University of Science and Technology in 2019. From 2011 to 2019, he was an assistant professor at Shih Chien University. Since 2019, he has been an associate professor at Changzhou College of Information Technology. His research domains include computing vision, cloud computing, and sensors. (yehchitsai@czcit.edu.cn)



Ming-Sheng Ling received his B.S. degree from Nanjing University of Aeronautics and Astronautics, China, in 2007. He is a senior engineer in the School of Software and Big Data, Changzhou College of Information Technology. His research domains include software engineering, computer vision, and sensors. (lingmingsheng@ccit.js.cn)