Designing the Yellow Head Virus Syndrome Recognition Application for Shrimp on an Embedded System

Chi Cuong Tran, Dinh Tu Nguyen, Hoang Dang Le, Trong Hieu Luu and Quoc Bao Truong*
College of Engineering, CanTho of University, Vietnam
*Correspondence: tqbao@ctu.edu.vn

Peer review: This article has been subject to a double-blind peer review process

Abstract

One of the most serious problems confronted by the shrimp farming industry is the disease caused by the yellow head virus (YHV). This research proposes an image processing algorithm to detect, identify and eliminate shrimp with the yellow head virus from the Litopenaeus vannamei gathering lines. Using a Raspberry Pi 3 module with the support of the OpenCV library which may be associated with Niblack's algorithm is primarily suitable for segmentation. First, the shrimp object was identified and separated from the background using the image segmentation technique and the boundary that surrounds the object. Then, identification of diseased shrimp was analysed based on colour threshold. In this study, the sample of shrimp disease group had the highest amount of ratio, with about 6% to 11%. Most of the samples without the disease had a ratio of 0%. The experimental results show that the system can identify and accurately determine the coordinates of shrimp with yellow head virus disease and send information to the shrimp classification system in the food industry.

Keywords: Litopenaeus vannamei, shrimp, yellow head virus syndrome, image processing

Introduction

At present, YHV syndrome occurs on Penaeus monodon, Litopenaeus vannamei, Metapenaeus ensis, Penaeus semisulcatus, and other shrimps in Vietnam, Thailand and India. The YHV is a shrimp virus that causes significant damage to farmed penaeid shrimp throughout Asia through mass pond fatalities, which results in heavy production losses and consequent severe economic damage. Susceptible species include Penaeus monodon (**Boonyaratpalin et al., 1993**), P. aztecus, P. duorarum, P. merguiensis, and P. setiferus **(Flegel et al., 1997; Flegel, 1997; Lightner et al., 1998**).

The traditional diagnostic methods include pathological observations or pathologic histology that do not allow for accurate detection of pathogens early. Many molecular methods like in situ, western blot, polymerase chain reaction (PCR), reverse transcription polymerase chain reaction (RT-PCR) **(Dhar et al., 2002**) which are developed to overcome these disadvantages. The PCR methods are currently being used widely and effectively in shrimp testing. However, one of the significant limitations of PCR are false-negative phenomenon which takes a lot of time and is a low-efficiency detection (**Amarakoon and Wijegoonawardane, 2017**).

Considering some practical applications in this field, the authors realize the shrimp identification and classification system during the process of gathering is very important. This paper proposes the method of image processing embedded with Raspberry Pi 3 module (**Tran et al., 2017**). It also considers the applicability of the system to observe, analyse and identify diseased shrimp of the shrimp classification process in the industry.

System Overview

The input images are recognized by the Pi Camera and connect with Raspberry Pi 3 via the Camera driver, the image is recorded in a raw form which will be processed using specific algorithms with the support of the OpenCV library (**Bradski and Kaehler, 2008**). This paper proposes the extraction method and sends a series of coordinates to the actuator block. At the same time, the features of the images will be shown on a PC/Laptop with the support of SSH/ Putty through IP address that is provided by Router. This model is shown in **Figure 1**.

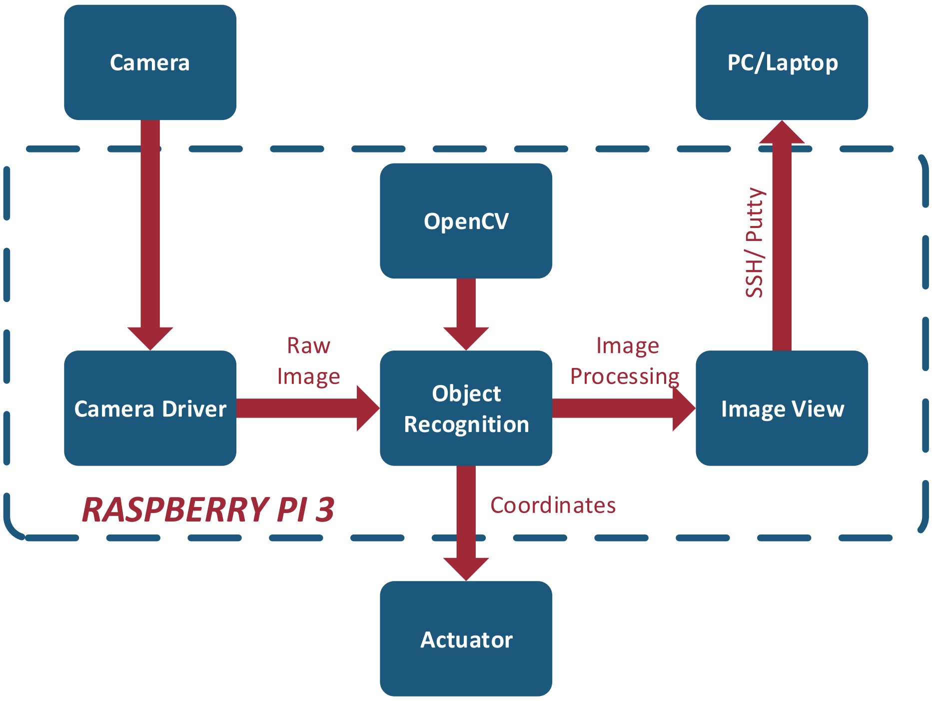


Figure 1 The diagram of an overall system

Proposed Method

Object Detection

The input image contains objects that will be reduced noise and removed background using the image segmentation algorithm. An adaptive threshold solution specifically is used specifically for this process. The detailed disposition steps are shown in **Figure 2**. This method is based on homogeneity – the same in term of grey level, colour, etc., which is a simple and common method in image segmentation because of its simplicity and fast performance.

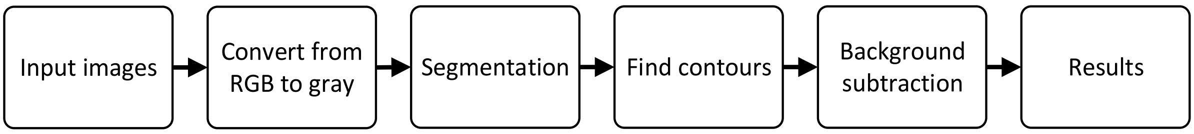


Figure 2 The object detection

Otsu's Algorithm

Otsu algorithm was launched by Nobuyuki Otsu (**Huang et al., 2012**). The objective of this algorithm is to determine a threshold T automatically based on greyscale values of the pixels in order to replace the use of fixed segment in the problem binary image based on threshold greyscale. The basic content of the method is described as follows: First, convert the input images to greyscale images and statistics on the number of grey levels, assuming there is $L(0 \leq L \leq 255)$ greyscale in the image. We dichotomize the pixels into two classes and C_2 (background and objects, or vice versa) by a threshold at level T; C_1 denotes pixels with levels $[1,T]$, and C_1 denotes pixels with levels $[T+1,L]$.Total number of pixels is called *N*, $h[i]$ number of pixels in the grey level $i(0 \le i \le 255)$ and probability of appearance greyscale level i is: $p_i = \frac{h[i]}{N}$. From that, we calculate the optimal threshold *T** by the formula (1):

$$T^* = \underset{0 \le T < L}{Arg\,Max}\left\{\sigma_B^2(T)\right\} \tag{1}$$

$\sigma_B^2(T)$ is identified by formula from (2) to (6).

Variance:

$$\sigma_B^2(T) = \omega_1(T)\omega_2(T)\left(\mu_2(T) - \mu_2(T)\right)^2 \tag{2}$$

Probability appear of C_1:

$$\omega_1(T) = P_1 = \sum_{i=0}^{T-1} p_i \tag{3}$$

Probability appear of C_2:

$$\omega_2(T) = P_2 = \sum_{i=T}^{L-1} p_i = 1 - P_1 \tag{4}$$

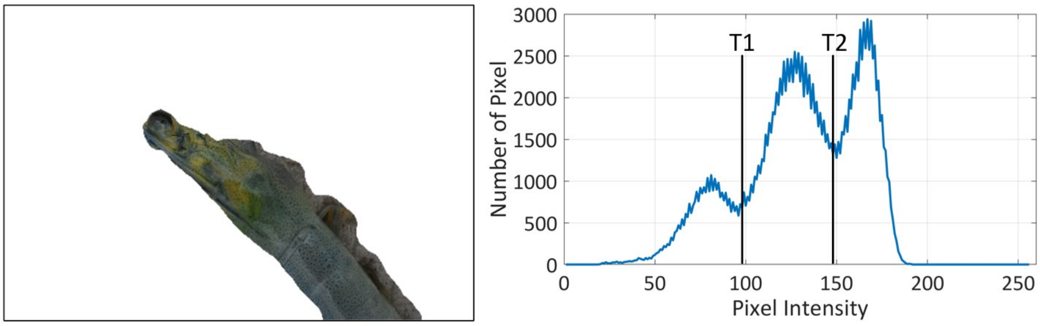
Average grey level of C_1:

$$\mu_1(T) = \sum_{i=0}^{T-1} iP(i / C_1) = \sum_{i=0}^{T-1} ip_i / \omega_1(T) \tag{5}$$

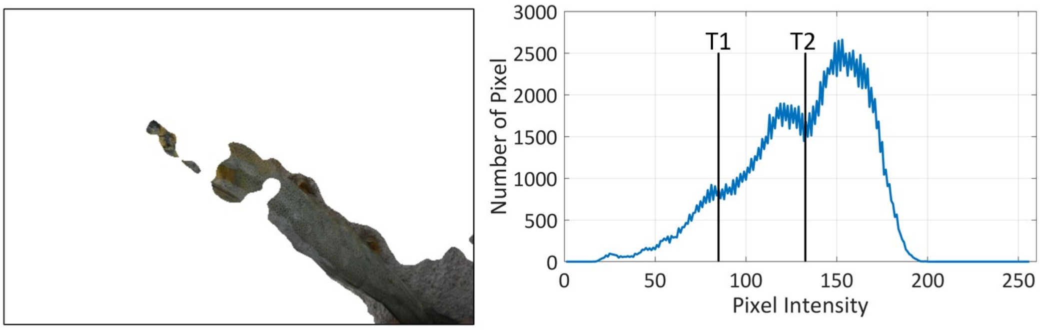
Average grey level of C_2:

$$\mu_2(T) = \sum_{i=T}^{L-1} iP(i / C_2) = \sum_{i=T}^{L-1} ip_i / \omega_2(T) \tag{6}$$

Solutions segment images by Otsu algorithm are a simple algorithm for calculating the threshold T. It is available for global thresholding. So, threshold T is an important and factor that determines the success of the algorithm. **Figure 3(a)** shows, two threshold T_1 and T_2 is determined through the average value calculated on the histogram. The area between T_1 and T_2 is the area of the object, the rest is the background. For the case of shrimp sample 1, we detect the object easy because of the clearly partition of the peaks on the histogram. In contrast, the incorrect segmental result is shown in **Figure 3(b)** because of the difficulty in determining threshold T_1 and T_2 based on the histogram.



(a) Shrimp sample 1



(b) Shrimp sample 2

Figure 3 The results of images segmentation by Otsu algorithm

Niblack's Algorithm

A threshold $T(x,y)$ is a value such that

$$b(x,y)=\begin{cases}0 & I(x,y)\leq T(x,y)\\ 1 & otherwise\end{cases} \quad (7)$$

where $b(x,y)$ is the binarized image and $I(x,y)\in[0,1]$ be intensity of a pixel at location (x,y) of the image I. In the local adaptive technique, a threshold is calculated for each pixel, based on some local statistics such as range, variance, or surface-fitting parameters of the neighboring pixels. It can be approached in different ways, such as background subtraction (**Lu et al., 2010**), water flow model (**Oh et al., 2005**), mean and standard derivation of pixel values (**Sauvola and Pietikäinen, 2000**) and local image contrast (**Su et al., 2010**). Some drawbacks of the local thresholding techniques are region size dependant, individual image characteristics, and time-consuming. Therefore, some researchers use a hybrid approach that applies both global and local thresholding methods (**Gatos et al., 2006**) and some use morphological operators (**Le et al., 2011**). Niblack (**Niblack, 1985**), Sauvola and Pietikäinen (**Sauvola and Pietikäinen, 2000**) use the local variance technique while Bernse (**Bernse, 1986**) uses midrange value within the local block.

Niblack (Niblack, 1985) proposed an algorithm that calculates a pixel-wise threshold by shifting a rectangular window across the image. This method varies the threshold over the image, based on the local mean and local standard deviation. Let the local area be $b \times b$. Also, the threshold $T_{ni}(x,y)$ at pixel $f(x,y)$ is determined by the following equation (8):

$$T_{ni}(x,y) = \mu_{ni}(x,y) + k_{ni}\sigma_{ni}^2(x,y) \tag{8}$$

$$\mu_{ni}(x,y) = \frac{1}{b^2}\left[\sum_{j=y-\frac{b}{2}}^{y+\frac{b}{2}}\left(\sum_{i=x-\frac{b}{2}}^{x+\frac{b}{2}} f(i,j)\right)\right] \tag{9}$$

$$\sigma_{ni}^2(x,y) = \frac{1}{b^2}\left[\sum_{j=y-\frac{b}{2}}^{y+\frac{b}{2}}\left(\sum_{i=x-\frac{b}{2}}^{x+\frac{b}{2}} \left(\mu_{ni}(x,y) - f(i,j)\right)^2\right)\right] \tag{10}$$

As in the definition of equation(9), (10), $\mu_{ni}(x,y)$ are the local mean $\sigma_{ni}^2(x,y)$ are standard deviation values of a local area. The size of the local window, b, should be small enough to accurately reflect the local illumination level and adequately large to include both objects and the background. Trier and Jain (**Trier and Jain, 1995**) recommend taking a 15×15 neighborhood and $k_{ni} = -0.2$. Thus, Niblack's algorithm cannot be applied to the varying resolution input images.

(a) Shrimp sample 1

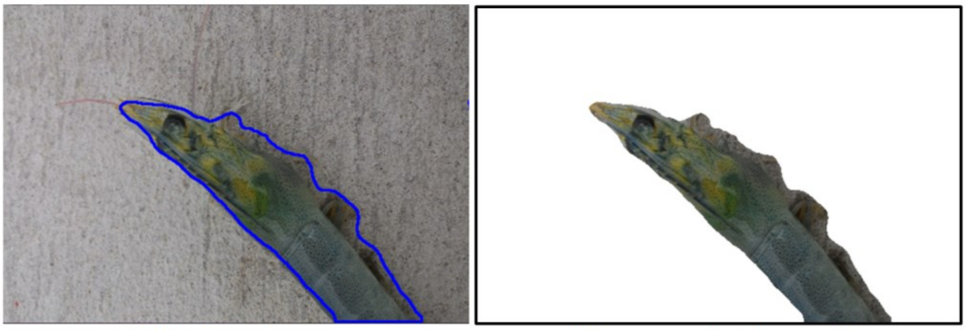
(b) Shrimp sample 2

Figure 4 The results of image segmentation by Niblack's algorithm

Image segmentation techniques using Otsu algorithm is not suitable for many cases such as approximately uniform illumination or object is similar to the background. In this case, the image segment using the adaptive threshold method is quite appropriate the effectiveness of image segmentation using the Niblack's algorithm is presented in **Figure 4**.

Find Contours and Remove Background

After splitting the objects, binary images obtained at the image segment stage are filtered using a 5×5 Median filter to eliminate noise. We will delimit the object's field and draw the boundary using the contour method in order to find the boundary around objects that were binary. Then, this method saves the boundary into the vector that points lying on this boundary. The results of the search of the object are shown in **Figure 5(a).**



(a) Find contours (b) Remove background

Figure 5 The results of find contours and remove background

The result of the subtraction process between finding contours and original images are shown in **Figure 5(b)**. This process is very important in image processing. The purpose of the subtraction should be to eliminate noise from the background and increase the accuracy of the next steps.

YHV Recognition

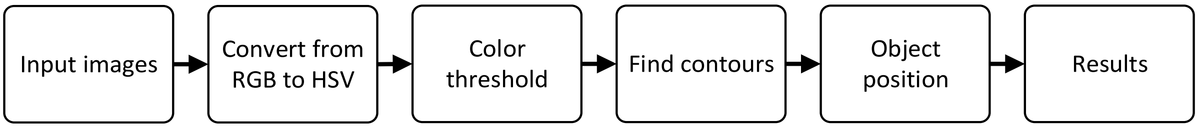


Figure 6 The identification of YHV syndrome

The detailed process to identify shrimp with YHV syndrome is shown in **Figure 6**.

HSV Colour Space

The RGB image is converted into the HSV space after object detection, because the HSV space and the HSL space are the same. Besides that, it is much used in image processing and a part of the computer vision technology. HSV is not the same as RGB, which is image intensities separation of colour information. This separation can be very helpful if we want to focus on the intensity component. Actually, sometimes we want to separate the different colour components from intensity for many various reasons such as increasing the saturation of colour or shadow removal.

The main reason is that it separates the colour information from saturation or intensity. Because the values are separated, you can construct a chart or a threshold rule by saturation and hue. In theory, this will work regardless of the change of saturation on the value channel. In fact, it is just a nice improvement. Even by indicating the colour you still have a very significant representation of the base colour that would probably work far better than RGB. The final result is a stronger colour threshold than the simpler ones.

Colour Threshold and Find Contours

Colour threshold is a simple and common method in image segmentation because of its simplicity and fast performance (**Kulkarni, 2012**). The next steps of the algorithm will be performed and show for Hue component of HSV colour representation. In this step, the image is filtered in a range which is set by the user. The setting of the filter was prepared to detect YHV syndrome. White pixels are pixels that satisfy the filter's condition. Many small white pixels that can be seen will be eliminated in further steps of the algorithm. As it can be seen the red coloured object is easily extracted by thresholding HSV values. However, the drawback is that function findContours finds any white coloured contour that was found in the thresholded image. The morphological operations (such as erosion and dilution) are able to eliminate noise effects.

Object Position

After separating the object from the background and delimiting the object by the contours, we find the centre of the object. To determine the coordinates of the object we have to find the moment of the image that is shown in theoretical (**Belkasim et al., 1991**).

The mathematical equation of the moment is shown in (11):

$$\mu_n = \int_{-\infty}^{+\infty} (x-c)^n f(x)dx \tag{11}$$

where n^{th} is the moment around point c. When you apply on the 2D space, we have two independent variables to present (11). That is presented again in (12):

$$\mu_{m,n} = \int\int (x-c_x)^m (x-c_y)^n f(x,y)dydx \tag{12}$$

where, $f(x,y)$ is the continuous function. So, we have to digitize each pixel that is shown in (13):

$$\mu_{m,n} = \sum_{x=0}^{\infty}\sum_{y=0}^{\infty} (x-c_x)^m (y-c_y)^n f(x,y) \tag{13}$$

After we calculate the region of the binary image, we have to calculate the 0^{th} moment.

$$\mu_{0,0} = \sum_{x=0}^{w} \sum_{y=0}^{h} x^0 y^0 f(x,y) \tag{14}$$

The formula is written again after x_0 and y_0 that is shown in (15).

$$\mu_{0,0} = \sum_{x=0}^{w} \sum_{y=0}^{h} f(x,y) \tag{15}$$

To determine the centre of the object, we have to compute on two axes.

$$centroid = \left(\frac{\mu_{1,0}}{\mu_{0,0}}, \frac{\mu_{0,1}}{\mu_{0,0}} \right) \tag{16}$$

The total number of pixels will be aggregated and expressed as follows:

$$\begin{aligned} sum_x &= \sum \sum x f(x,y) \\ sum_y &= \sum \sum y f(x,y) \end{aligned} \tag{17}$$

After that, we get the average by dividing the total of number pixel. Its formula is shown in (18).

$$\begin{aligned} \mu_{1,0} &= \frac{sum_x}{\mu_{0,0}} \\ \mu_{0,1} &= \frac{sum_y}{\mu_{0,0}} \end{aligned} \tag{18}$$

For the function in OpenCV, the coordinate of the centre is computed as follow:

$$\begin{aligned} c_x &= int\left(M[\text{‘}m10\text{’}] / M[\text{‘}m00\text{’}]\right) \\ c_y &= int\left(M[\text{‘}m01\text{’}] / M[\text{‘}m00\text{’}]\right) \end{aligned} \tag{19}$$

Experimental Results

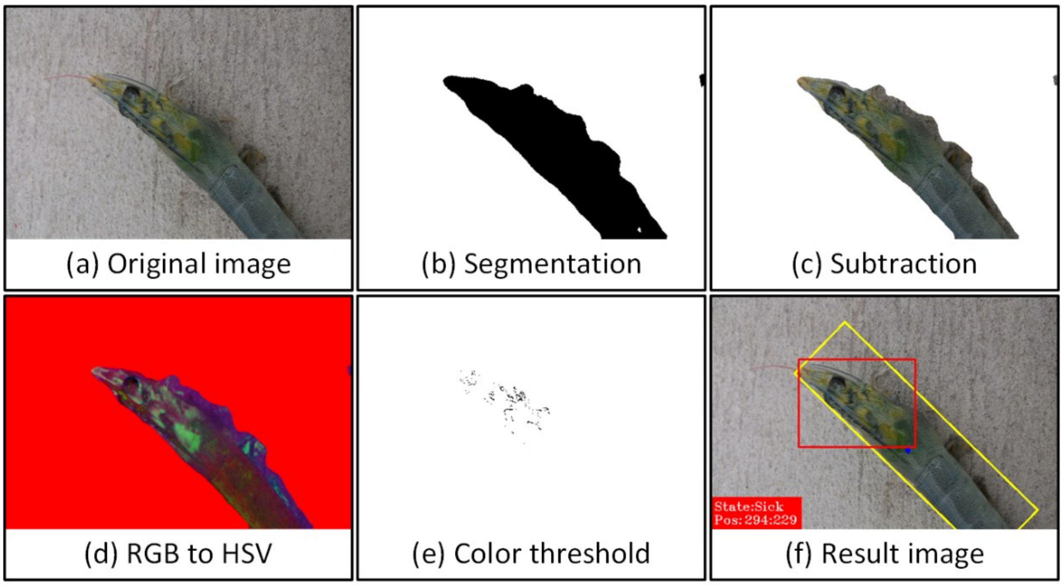


Figure 7 The process of identifying the disease location of diseased shrimp

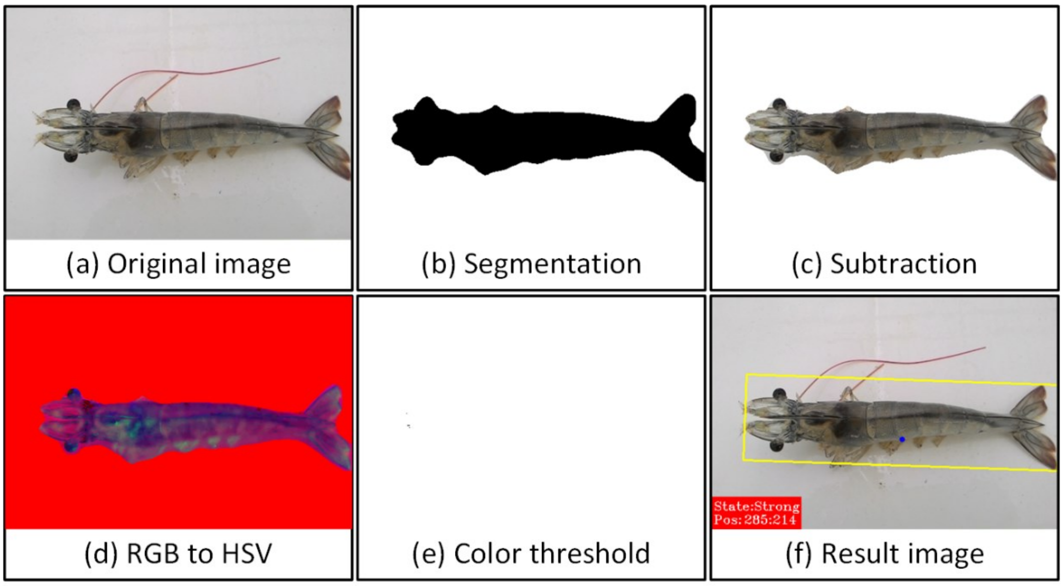


Figure 8 The process of identifying healthy shrimp

In this part, YHV inflection of shrimp will be illustrated by some typical identification results. These results were analysed and experimented on in the module Raspberry 3 using OpenCV library and taken from some sample images on the Raspberry Pi's camera. The process of identifying the shrimp region and identifying the disease location of shrimp is shown in **Figure 7**. The process of identifying healthy shrimp is shown in **Figure 8**.

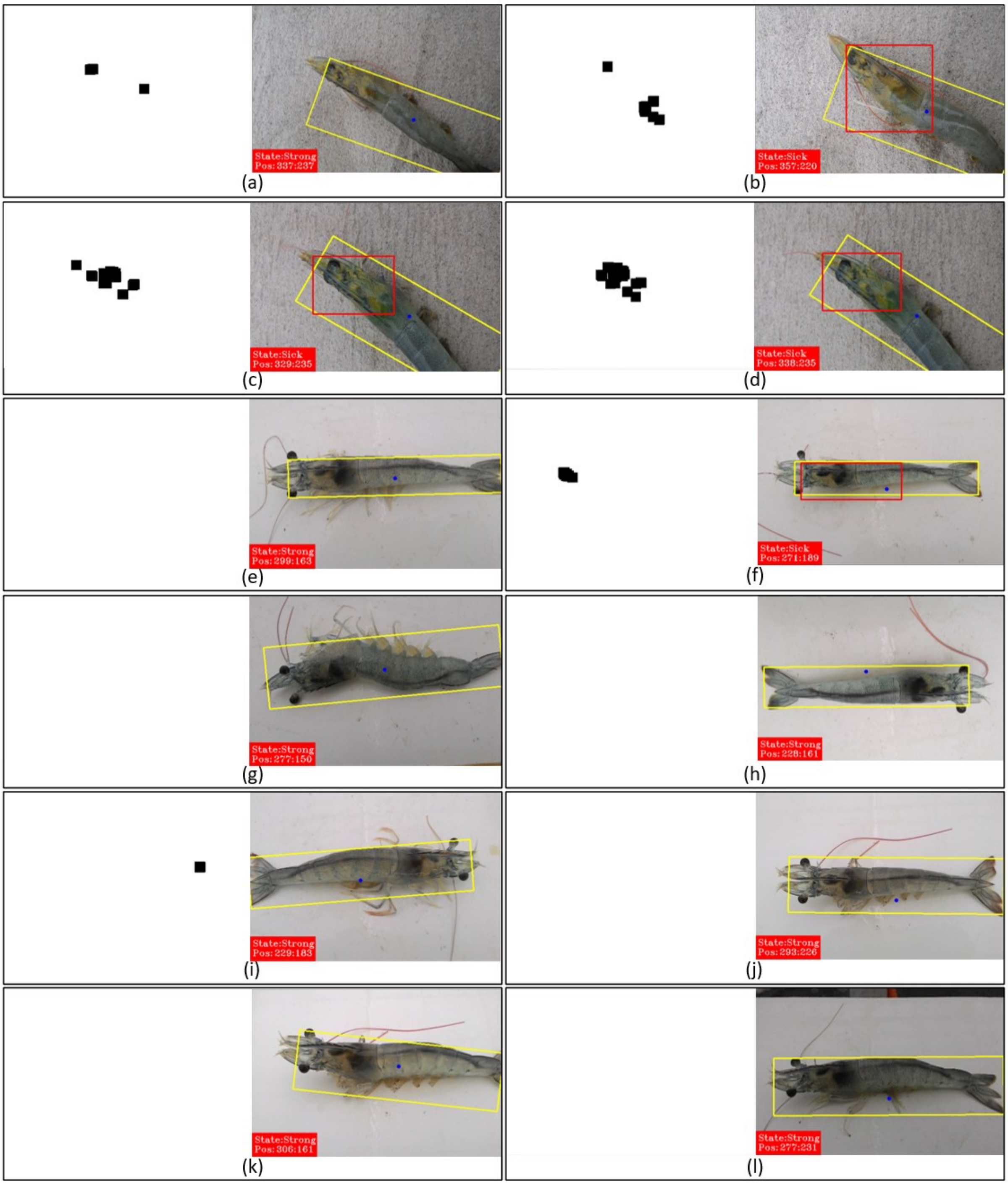


Figure 9 The result of determining the state for many samples using Otsu's algorithm

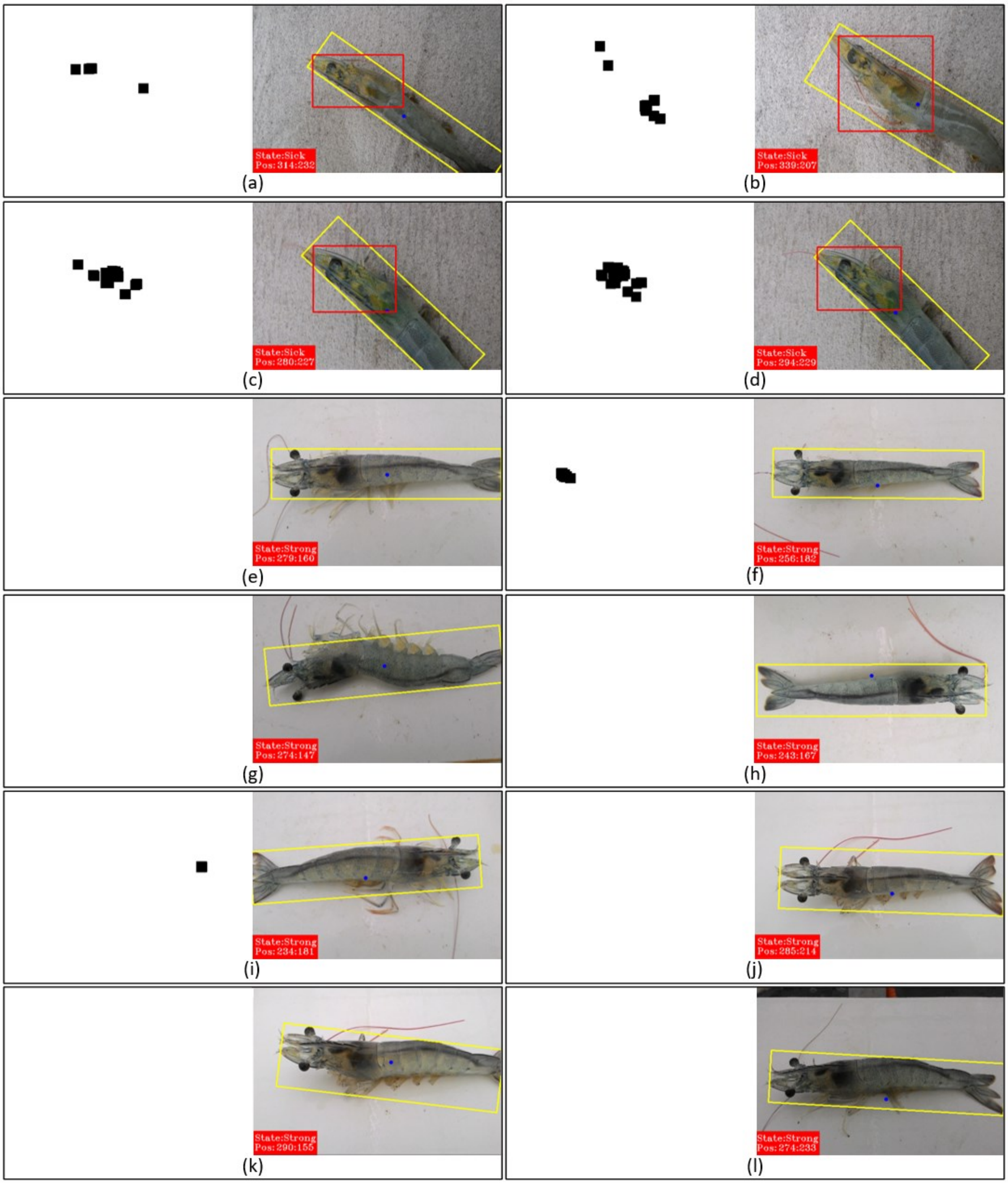


Figure 10 The result of determining the state for many samples using Niblack's algorithm

To examine the performance of the proposed method for identifying the YHV disease. These images are taken from many shrimps which different lighting condition and background. The results are not accurate in some cases when using the Otsu algorithm. In the above description, the incorrect segmental result is shown in **Figure 9(a)** and **Figure 9(f)** because of the difficulty in threshold determination and that based on the histogram. That is the reason leading to the calculation of the ratio between shrimp areas and disease areas incorrect (**Table 1**). The results identified in shrimp YHV disease with 100% accuracy for Niback's algorithm. However, in **Figure 10(f)** and **Figure 10(i)** appears interference from the process of taking the colour threshold. This is a common occurrence, because the shrimp head colour is similar to diseased shrimp but a lighter colour. It also caused noise due to light conditions when collecting images. This study uses a simple algorithm to solve this problem based on the noise levels of the process of obtaining a colour threshold. In this process, if the ratio between the identification area and the shrimp area is low, it is considered as interference.

Table 1 Experimental results of the disturbance effect on disease identification in shrimp (S/D are the ratios of shrimp areas and disease areas. P is positive for YHV and N is negative for YHV)

Samples	Ratio of S/D using Otsu (%)	Ratio of S/D using Niblack (%)	YHV (manual)	YHV (Otsu)	YHV (Niblack)
1	3.67	6.25	**P**	**N**	**P**
2	7.23	9.80	P	P	P
3	5.25	7.03	P	P	P
4	8.13	10.52	P	P	P
5	0	0	N	N	N
6	6.54	4.11	**N**	**P**	**N**
7	0	0	N	N	N
8	0	0	N	N	N
9	1.68	1.49	N	N	N
10	0	0	N	N	N
11	0	0	N	N	N
12	0	0	N	N	N

In **Table 1**, Samples of shrimp diseases have high disease rate areas from 6% to 11% by Niblack's algorithm. Most of the samples without the disease had a ratio of 0%. Except for some cases, shrimp is not sick, but rates below 5% and the disease can be excluded based on experience. This value is the ratio of the noise area after the dilating and shrimp area identified. As described in **Figure 3** and **Figure 4**, Image segmentation techniques by Niblack's algorithm more accurate than Otsu algorithm. Therefore, the ratios of shrimp areas and disease areas of Niblack's algorithm area gives more accurate results.

Conclusions

In this study, we presented an image processing method for identifying the YHV location of shrimp. The experimental results showed that the adaptive threshold is more effective than the Otsu algorithm in identifying with less noise caused by the background. The proposed method for identification based on the characteristics of the YHV syndrome in shrimp is colour. The results of this study may be contribute to the development of an automated shrimp classification system with various diseases in the food industry.

Acknowledgement

In this paper, the dataset pictures were taken by Mr. Le Be Hai who works for Viet Uc Seafood Groups. We would like to express our thank you to Mr. Be Hai, former mechatronics students at Can Tho University, who helped us with our initial searches on applying computer vision to the shrimp industry.

Quoc Bao Truong has been working with CanTho University, Vietnam since 1998. He received his B.S. degree in Computer Science from Can Tho University, Can Tho, Vietnam in 1998 and his Masters degree in Computer Science from Natural Science University, National University of Ho Chi Minh, Vietnam in 2003. He obtained his Ph.D. in Mechatronics and Intelligent Control at Ulsan University, Ulsan, Korea 2011. Since then, he has been a senior lecturer in the Automative Engineering Department, College of Technology, Can Tho University. His research interests are computer vision, deepth learning, pattern recognition, and intelligent control for mobile robot.

References

Amarakoon, A. G. U. & Wijegoonawardane, P. 2017. A comparative analysis of Yellow Head Virus (YHD) diagnostic methods adopted in Sri Lanka to investigate the accuracy and specificity of the virus. *World Scientific News*, 66), pp 181-192.

Belkasim, S. O., Shridhar, M. & Ahmadi, M. 1991. Pattern recognition with moment invariants: a comparative study and new results. *Pattern recognition*, 24(12), pp 1117-1138.

Bernse, J. 1986. Dynamic thresholding of grey-level images. Proc. 8th Int. Conf. on Pattern Recognition, 1986.

Boonyaratpalin, S., Supamattaya, K., Kasornchandra, J., Direkbusaracom, S., Aekpanithanpong, U. & Chantanachooklin, C. 1993. Non-occluded baculo-like virus, the causative agent of yellow head disease in the black tiger shrimp (Penaeus monodon). *Fish Pathology*, 28(3), pp 103-109.

Bradski, G. & Kaehler, A. 2008. *Learning OpenCV: Computer vision with the OpenCV library*: " O'Reilly Media, Inc.".

Dhar, A. K., Roux, M. M. & Klimpel, K. R. 2002. Quantitative assay for measuring the Taura syndrome virus and yellow head virus load in shrimp by real-time RT-PCR using SYBR Green chemistry. *Journal of Virological Methods*, 104(1), pp 69-82.

Flegel, T., Boonyaratpalin, S. & Withyachumnarnkul, B. 1997. Progress in research on yellow-head virus and white-spot virus in Thailand. *Diseases in Asian Aquaculture III*, 285-295.

Flegel, T. W. 1997. Major viral diseases of the black tiger prawn (Penaeus monodon) in Thailand. *World Journal of Microbiology and Biotechnology*, 13(4), pp 433-442.

Gatos, B., Pratikakis, I. & Perantonis, S. J. 2006. Adaptive degraded document image binarization. *Pattern Recognition*, 39(3), pp 317-327.

Huang, M., Yu, W. & Zhu, D. An improved image segmentation algorithm based on the Otsu method. 2012 *13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD)*. IEEE, 135-139.

Kulkarni, N. 2012. *Color Thresholding Method for Image Segmentation of Natural Images*.

Le, T. H. N., Bui, T. D. & Suen, C. Y. 2011. Ternary entropy-based binarization of degraded document images using morphological operators. *2011 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE.

Lightner, D., Hasson, K., White, B. & Redman, R. 1998. Experimental infection of western hemisphere penaeid shrimp with Asian white spot syndrome virus and Asian yellow head virus. *Journal of Aquatic Animal Health*, 10(3), pp 271-281.

Lu, S., Su, B. & Tan, C. L. 2010. Document image binarization using background estimation and stroke edges. *International Journal on Document Analysis and Recognition*, 13(4), pp 303-314.

Niblack, W. 1985. *An Introduction to Digital Image Processing*. Strandberg Publishing Company.

Oh, H.-H., Lim, K.-T. & Chien, S.-I. 2005. An improved binarization algorithm based on a water flow model for document image with inhomogeneous backgrounds. *Pattern Recognition*, 38(12), pp 2612-2625.

Sauvola, J. & Pietikäinen, M. 2000. Adaptive document image binarization. *Pattern Recognition*, 33(2), pp 225-236.

Su, B., Lu, S. & Tan, C. L. 2010. Binarization of historical document images using the local maximum and minimum. *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*. ACM.

Tran, C. C., Nguyen, D. T., Le, H. D., Truong, Q. B. & Nguyen, C. N. 2017. An application of movement direction control for the three wheeled moblie robots using visual information. *Seventh International Conference on Information Science and Technology*, 121-128.

Trier, O. D. & Jain, A. K. 1995. Goal-directed evaluation of binarization methods. *IEEE transactions on Pattern analysis and Machine Intelligence,* 17(12)**,** pp 1191-1201.

To cite this article:

Tran, C.C., Nguyen, D.T., Le, H.D., Luu, T.H., and Truong, Q.B. (2019) Designing the Yellow Head Virus Syndrome Recognition Application for Shrimp on an Embedded System. *Exchanges: The Interdisciplinary Research Journal*, 6(2), 48-63. Retrieved from: https://doi.org/10.31273/eirj.v6i2.309