

PHÂN CỠ TÔM BẰNG KỸ THUẬT XỬ LÝ ẢNH TRÊN MÁY TÍNH RASPBERRY PI

IMPLEMENTATION OF SHRIMPS GRADING USING IMAGE PROCESSING ON RASPBERRY PI COMPUTERS

Trương Chí Cường¹, Huỳnh Thanh Lộc¹, Võ Hoàng Tâm², Lương Vinh Quốc Danh¹

¹*Trường Đại học Cần Thơ; lvqdanh@ctu.edu.vn*

²*Trường Đại học Sư phạm Kỹ thuật Vĩnh Long*

Tóm tắt - Trong bài viết này, tác giả trình bày ứng dụng kỹ thuật xử lý ảnh để xác định trọng lượng tôm. Phương pháp xử lý ảnh để xác định trọng lượng tôm bao gồm các công đoạn sau: thu nhận ảnh tôm từ camera, chuyển đổi không gian màu, tách đối tượng bằng kỹ thuật phân ngưỡng ảnh, xác định diện tích của đối tượng và xây dựng hàm toán học biểu diễn mối liên hệ giữa diện tích và trọng lượng của đối tượng từ các mẫu ngẫu nhiên. Giải thuật xử lý ảnh được viết bằng ngôn ngữ Python kết hợp thư viện OpenCV và cài đặt chương trình trên máy tính Raspberry Pi 3. Thử nghiệm kiểm chứng trên số lượng 50 mẫu tôm thẻ chân trắng được chọn ngẫu nhiên cho thấy phương pháp đề xuất có thể xác định trọng lượng tôm với độ sai số trung bình là 0,34g. Kết quả nghiên cứu có thể được ứng dụng trong việc chế tạo thiết bị phân cỡ tôm tự động cho các nhà máy chế biến thủy sản.

Từ khóa - Phân cỡ tôm; Python; Raspberry Pi; Tôm thẻ chân trắng; Xử lý ảnh

1. Đặt vấn đề

Trong những năm gần đây, mặt hàng tôm trở thành một trong những sản phẩm thủy sản xuất khẩu chủ lực của Việt Nam. Theo Hiệp hội Chế biến và Xuất khẩu Thủy sản Việt Nam, sản lượng tôm xuất khẩu của Việt Nam chiếm 45% sản lượng tôm toàn cầu [1]. Tại các nhà máy chế biến thủy sản, phân cỡ tôm là một trong những khâu quan trọng trong quy trình chế biến tôm xuất khẩu. Trong thực tế, việc phân cỡ tôm tại một số cơ sở chế biến thủy sản được thực hiện thủ công với số lượng nhân công lớn để có thể đảm bảo năng suất yêu cầu. Tuy nhiên, tình trạng thiếu hụt lao động và giá thành nhân công ngày càng tăng đã gây không ít khó khăn cho các doanh nghiệp chế biến thủy sản. Trong nỗ lực nhằm ổn định hoạt động sản xuất và tăng năng suất công đoạn phân cỡ tôm, nhiều doanh nghiệp đã đầu tư các thiết bị phân cỡ tôm tự động. Phần lớn các thiết bị phân cỡ tôm tự động hiện có trên thị trường sử dụng nguyên lý cơ - điện tử, theo đó, trọng lượng tôm được xác định bằng các cảm biến trọng lượng [2], [3]. Phương pháp xác định trọng lượng bằng cảm biến trọng lượng có tốc độ xử lý cao, tuy nhiên, đòi hỏi cảm biến có độ chính xác cao, chi phí đầu tư và bảo trì cao.

Kỹ thuật xử lý ảnh đã được ứng dụng trong nhiều lĩnh vực khác nhau từ nhận dạng khuôn mặt đến phân loại trái cây [4]-[7]. Kỹ thuật xử lý ảnh cũng được sử dụng ở khâu phân loại sản phẩm trong nuôi trồng và chế biến thủy sản [8]-[10]. Trong bài viết này, tác giả trình bày ứng dụng kỹ thuật xử lý ảnh để xác định trọng lượng tôm. Phương pháp xử lý ảnh để xác định trọng lượng tôm bao gồm các công đoạn sau: thu nhận ảnh tôm từ camera, chuyển đổi không gian màu, tách đối tượng bằng kỹ thuật phân ngưỡng ảnh, xác định diện tích của đối tượng và thiết lập hàm toán học biểu diễn mối liên hệ giữa diện tích và trọng lượng của đối

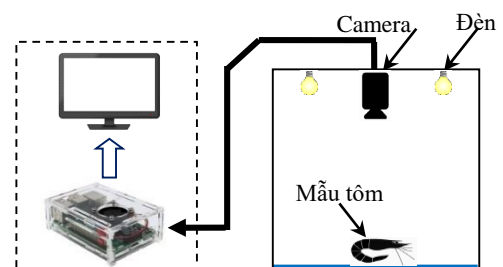
Abstract - In this paper, the authors present the application of image processing techniques to determine the weight of shrimps. The image processing method to determine shrimp weight consists of the following stages: collecting image from camera, converting color space, thresholding images, extracting feature and establishing mathematical functions describing the relationship between area and weight of the objects from random samples. Image processing algorithms are written in Python language with the OpenCV library and installed on the Raspberry Pi 3 computers. Experimental validation with 50 white-leg shrimps randomly selected shows that the proposed approach can be used to determine the shrimp weight with an average error of 0.34g. The study results can be applied for the implementation of automatic shrimp grading machines for seafood processing plants.

Key words - Shrimp grading; Python; Raspberry Pi; White-leg shrimp; Image processing

tượng từ các mẫu ngẫu nhiên [9]. Trong nghiên cứu này, tác giả đã xây dựng được phương trình biểu diễn sự liên hệ giữa diện tích và trọng lượng của tôm thẻ chân trắng. Bên cạnh đó, việc xây dựng giải thuật xử lý ảnh bằng ngôn ngữ Python [11] kết hợp thư viện OpenCV [12] và cài đặt chương trình trên máy tính Raspberry Pi 3 [13] cũng được xem là một đóng góp của tác giả. Kỹ thuật này có thể mang đến một giải pháp khả thi trong việc chế tạo thiết bị phân cỡ tôm tự động sử dụng trong các nhà máy chế biến thủy sản.

2. Phương pháp thực hiện

2.1. Hệ thống thu nhận hình ảnh



Kit Raspberrry Pi 3 +

Hình 1. Hệ thống thu nhận hình ảnh tôm

Sơ đồ khối của hệ thống thu nhận hình ảnh mẫu tôm được trình bày ở Hình 1. Phần cứng hệ thống bao gồm 2 phần chính: bộ kit Raspberry Pi 3 và hộp chụp ảnh. Hộp chụp ảnh được sử dụng để thu nhận hình ảnh mẫu tôm và truyền về bộ kit Raspberry Pi 3 để xử lý. Đây hộp chụp ảnh là nơi đặt mẫu tôm, được phủ bằng một lớp giấy xốp màu xanh dương nhằm giảm sự phản xạ ánh sáng. Một camera Raspberry Pi NoIR V2, độ phân giải tối đa 8Mpixels, được

lắp đặt ở chính giữa nắp hộp để thu hình ảnh mẫu tôm. Khoảng cách từ mẫu tôm đến camera là khoảng 30 cm. Hai bóng đèn LED ánh sáng vàng, công suất 4W/bóng, được bố trí bên trong hộp chụp ảnh để đảm bảo điều kiện chiếu sáng mẫu tôm ổn định và sự đồng nhất về chất lượng của các hình ảnh thu được.

2.2. Xác định trọng lượng mẫu tôm

Tôm thẻ chân trắng (tên khoa học *Litopenaeus vannamei*) sử dụng trong nghiên cứu là loại còn nguyên vỏ và đầu được mua tại chợ ở Tp. Cần Thơ. Các mẫu tôm sau khi mua về được cân để xác định trọng lượng bằng cân điện tử do hãng OHAUS [14] sản xuất có độ sai số là $\pm 0,01g$ (Hình 2). Các mẫu tôm được làm khô nước bằng giấy thấm nước trước khi cân.

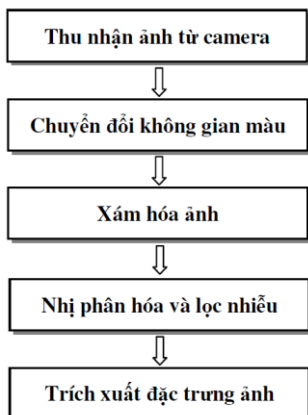


Hình 2. Thiết bị cân khối lượng tôm.

Trong nghiên cứu này, tổng cộng có 100 mẫu tôm được chọn ngẫu nhiên và được mua thành 2 đợt, mỗi đợt 50 con. Theo đó, 50 mẫu tôm đầu tiên được sử dụng để xây dựng mô hình tính toán trọng lượng tôm có trọng lượng cụ thể như sau: 13 con tôm có trọng lượng trong khoảng 4,85g ~ 8,44g, 10 con trong khoảng 9,05g ~ 9,78g, 11 con trong khoảng 10,35g ~ 12,33g, 11 con trong khoảng 14,62g ~ 20,83g, 5 con trong khoảng 22,79g ~ 29,67g. Nhóm 50 mẫu tôm thứ 2 được sử dụng để kiểm chứng mô hình đã xây dựng, có trọng lượng phân bố như sau: 10 con trong khoảng 5,58g ~ 9,52g, 15 con trong khoảng 10,51g ~ 14,38g, 12 con trong khoảng 15,44g ~ 19,82g, 8 con trong khoảng 20,25g ~ 24,58g, 5 con trong khoảng 25,73g ~ 29,67g.

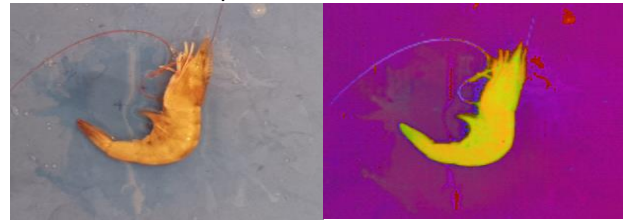
2.3. Các bước xử lý hình ảnh

Quá trình thu nhận và xử lý hình ảnh mẫu tôm bao gồm 5 bước và được thực hiện bởi chương trình xử lý viết bằng ngôn ngữ Python và thư viện OpenCV cài đặt trên kit Raspberry Pi 3 (Hình 3).



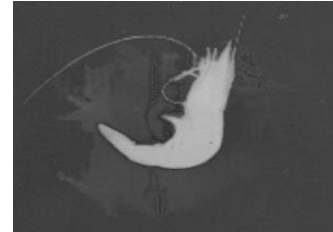
Hình 3. Lưu đồ các bước xử lý hình ảnh mẫu tôm

Bước 1: Thu nhận ảnh từ camera



(a)

(b)



(c)



(d)

(e)

Hình 4. Hình ảnh mẫu tôm qua các bước xử lý (a) thu nhận ảnh từ camera, (b) chuyển đổi không gian màu, (c) xám hóa, (d) nhị phân hóa và (e) lọc nhiễu

Hình ảnh của mẫu tôm sau khi được chụp bởi camera sẽ được truyền về kit Raspberry Pi 3 để xử lý (Hình 4a). Hàm `cv2.imread()` trong thư viện OpenCV được sử dụng để đọc hình ảnh. Để dung hòa giữa chất lượng hình ảnh và tốc độ xử lý, các hình ảnh chụp từ camera được thiết lập độ phân giải là 480×480 pixels.

Bước 2: Chuyển đổi không gian màu

Trong bước này, hình ảnh sẽ được chuyển từ không gian màu RGB (đỏ, lục, lam) sang HSV (màu sắc, độ bão hòa, cường độ sáng). Kết quả của bước này cho hình ảnh có sự phân biệt rõ giữa vùng ảnh mẫu tôm và màu nền như ở Hình 4b.

Bước 3: Xám hóa ảnh

Để giảm thời gian xử lý hình ảnh, các ảnh màu sẽ được chuyển đổi sang ảnh xám trước khi thực hiện các bước tiếp theo bằng cách sử dụng hàm `cv2.cvtColor()` của thư viện OpenCV (Hình 4c).

Bước 4: Nhị phân hóa và lọc nhiễu

Trước tiên, ảnh xám sẽ được chuyển đổi sang ảnh nhị phân bằng cách sử dụng hàm `cv2.threshold()` của thư viện OpenCV (Hình 4d). Thuật toán phân ngưỡng ảnh tự động Otsu được áp dụng ở bước này [15].

Sau khi được nhị phân hóa, ảnh có thể chứa các lỗi như điểm trắng trên nền ảnh hoặc chấm đen trong vùng diện tích ảnh tôm. Ngoài ra, sự không trơn tru của đường viền vùng diện tích tôm cũng có thể dẫn đến sai số trong việc tính toán ở bước 5. Do đó, một phép toán lọc ảnh sẽ được áp dụng tiếp theo để giải quyết các vấn đề trên (Hình 4e).

Bước 5: Trích xuất đặc trưng ảnh

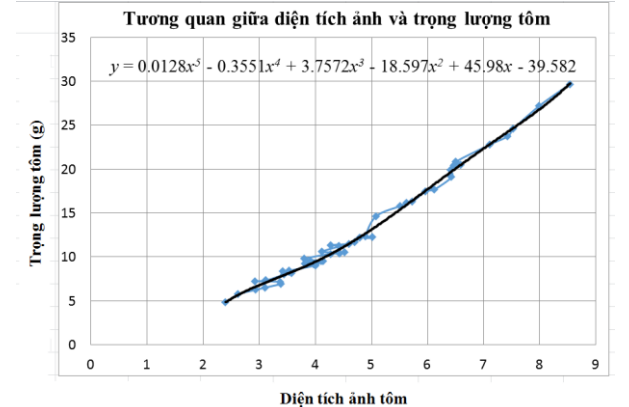
Chức năng tìm đường viền (edge detection) trong thư viện OpenCV được sử dụng để tính toán vùng diện tích ảnh tóm. Các hàm `cv2.findContours()` và `cv2.contourArea()` được sử dụng để xác định vùng diện tích ảnh tóm.

2.4. Sự tương quan giữa diện tích ảnh và trọng lượng của tôm

Dữ liệu vùng diện tích ảnh tóm sau khi trích xuất được kết hợp với dữ liệu trọng lượng tôm tương ứng được ghi nhận trước đó để xác định hàm số biểu diễn sự tương quan giữa diện tích ảnh tóm và trọng lượng tôm. Các dữ liệu này được xử lý bằng phần mềm Excel [16] và cho kết quả là hàm số

$$y = 0,0128x^5 - 0,3551x^4 + 3,7572x^3 - 18,597x^2 + 45,98x - 39,582$$

với bình phương của hệ số tương quan $R^2 = 0,982$, trong đó x là diện tích ảnh của tôm và y là trọng lượng tôm (Hình 5).



Hình 5. Hàm số biểu diễn sự tương quan giữa diện tích ảnh tôm và trọng lượng tôm

3. Kết quả và thảo luận

3.1. Kết quả thực nghiệm và thời gian xử lý ảnh

Để kiểm chứng hàm số biểu diễn sự tương quan giữa diện tích và trọng lượng tôm đã thiết lập ở phần trên, 50 mẫu tôm mua ở đợt 2 được cân để xác định trọng lượng và chụp ảnh để tính diện tích vùng ảnh tóm. Dữ liệu diện tích ảnh tóm được đưa vào hàm số tương quan và kết quả tính toán bởi chương trình được so sánh với số liệu trọng lượng tôm như ở Bảng 1. Trong Bảng 1, độ sai lệch được xác định theo công thức sau: *độ sai lệch = trọng lượng ban đầu của tôm – trọng lượng tính toán bởi chương trình*. Kết quả thực nghiệm cho thấy, đa số các trường hợp có số liệu tính toán rất gần với số liệu đo đặc với sai số trung bình là 0,34g. Sai số này có thể đáp ứng được các yêu cầu thực tế về phân cỡ tôm tại các nhà máy chế biến thủy sản.

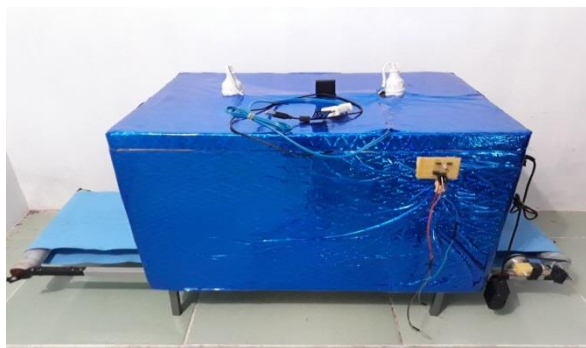
Bảng 1. Bảng số liệu kết quả thực nghiệm

STT	Trọng lượng thực tế (g)	Trọng lượng tính toán (g)	Độ sai lệch (g)
1	22,04	22,20	0,16
2	22,18	22,68	0,50
3	15,85	15,28	0,57
4	17,40	17,03	0,37
5	14,16	14,53	0,37

6	14,07	14,09	0,02
7	10,51	10,17	0,34
8	11,54	11,12	0,42
9	12,41	11,82	0,59
10	19,68	19,28	0,40
11	11,09	10,84	0,25
12	18,11	18,15	0,04
13	20,25	20,46	0,21
14	13,74	13,23	0,51
15	14,38	13,65	0,73
16	20,62	21,10	0,48
17	19,82	19,76	0,06
18	7,33	7,78	0,45
19	8,39	8,03	0,36
20	9,52	9,14	0,38
21	6,89	6,93	0,04
22	5,58	6,03	0,45
23	10,62	10,71	0,09
24	8,32	8,40	0,08
25	5,75	5,83	0,08
26	9,26	8,95	0,31
27	8,31	7,97	0,34
28	15,44	14,59	0,85
29	11,38	11,12	0,26
30	12,26	11,94	0,32
31	13,24	12,60	0,64
32	15,68	15,46	0,22
33	7,87	7,64	0,23
34	12,56	12,45	0,11
35	11,78	11,56	0,22
36	15,88	15,19	0,69
37	11,20	10,97	0,23
38	15,60	15,61	0,01
39	15,61	15,23	0,38
40	16,74	16,76	0,02
41	15,91	15,46	0,46
42	23,09	23,35	0,26
43	26,88	26,04	0,84
44	25,73	25,68	0,05
45	29,67	29,84	0,17
46	27,56	27,92	0,36
47	24,58	23,51	1,07
48	23,51	22,98	0,53
49	21,10	21,61	0,51
50	26,62	26,43	0,20

Một mô hình băng chuyền đơn giản với tốc độ chuyển động có thể thay đổi đã được xây dựng để đánh giá tốc độ xử lý hình ảnh của hệ thống (Hình 6). Kết quả thực nghiệm cho thấy, tổng thời gian xử lý trung bình một hình ảnh là 0,8s, trong đó 0,15s là thời gian xử lý ảnh và 0,65s là thời gian truyền dữ liệu ảnh từ camera về kit Raspberry Pi 3. Để tăng tốc độ xử lý hình ảnh, một webcam Logitech C170 [17] có độ phân giải tối đa 5 Mpixels đã được sử dụng thay thế camera Raspberry Pi NoIR V2 để thu nhận ảnh. Kết

quả thực nghiệm cho thấy, thời gian xử lý hình ảnh trung bình giảm còn 0,22s nhờ sự cải thiện về tốc độ truyền dữ liệu từ camera về kit Raspberry Pi 3. Theo đó, một thiết bị camera xử lý cho phép thực hiện phân cỡ tôm với tốc độ trên 250 mẫu tôm/phút.

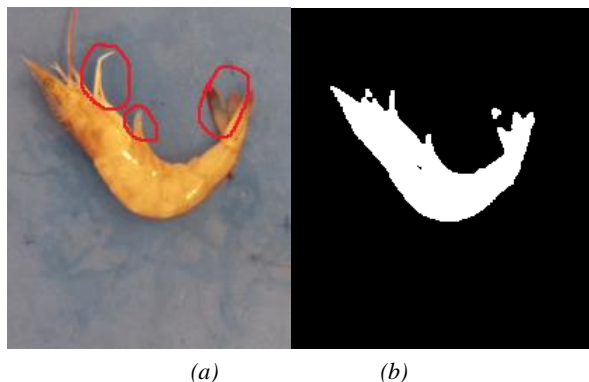


Hình 6. Mô hình băng chuyền đơn giản

3.2. Một số nguyên nhân gây sai lệch kết quả tính toán

3.2.1. Sai lệch do hình dạng tôm

Trọng lượng tôm được xác định dựa trên kết quả tính toán vùng diện tích tôm. Do vậy, vị trí của chân đầu, chân bụng và đuôi tôm là một trong các nguyên nhân dẫn đến sai lệch kết quả tính toán. Cụ thể, khi các chân và đuôi tôm đưa ra hoặc khấp lại sẽ làm sai lệch diện tích tính toán. Ở Hình 7, những vùng được khoanh tròn màu đỏ là nguyên nhân làm cho diện tích tính toán tăng lên.



Hình 7. Sai lệch trong tính toán diện tích tôm: (a) hình ảnh các vị trí gây sai lệch, (b) ảnh nhị phân của ảnh (a)

3.2.2. Sai lệch do camera chụp ảnh

Các hình ảnh được chụp bởi camera Raspberry Pi cho thấy khi đối tượng nằm càng xa trung tâm của hình ảnh thì sẽ có diện tích càng tăng. Đây là hiện tượng méo ảnh Pincushion (méo rìa cạnh) và được quyết định bởi chất lượng của camera. Sai lệch này có thể được giảm thiểu bằng cách sử dụng các loại camera với thấu kính có chất lượng cao [18]. Ngoài ra, trường hợp camera bị đặt nghiêng, không song song với mặt phẳng chụp ảnh cũng là một trong các nguyên nhân dẫn đến sai lệch kết quả tính toán.

4. Kết Luận

Nội dung bài viết đã trình bày ứng dụng việc ứng dụng kỹ thuật xử lý ảnh trên nền tảng máy tính Raspberry Pi 3, ngôn ngữ Python và thư viện OpenCV để xác định trọng lượng tôm thẻ chân trắng. Các kỹ thuật được sử dụng trong quá

trình xác định trọng lượng tôm bao gồm: thu nhận ảnh tôm từ camera, chuyển đổi không gian màu, tách đối tượng bằng kỹ thuật phân ngưỡng ảnh, xác định diện tích ảnh của tôm và xây dựng hàm toán học biểu diễn mối liên hệ giữa diện tích và trọng lượng tôm dựa trên 50 mẫu tôm được chọn ngẫu nhiên. Thực nghiệm kiểm chứng trên số lượng 50 mẫu tôm thẻ chân trắng được chọn ngẫu nhiên cho thấy phương pháp đề xuất có thể xác định trọng lượng tôm với độ sai số trung bình là 0,34g. Kết quả nghiên cứu cho thấy kỹ thuật này có thể mang đến một giải pháp khả thi trong việc chế tạo thiết bị phân cỡ tôm tự động sử dụng trong các nhà máy chế biến thủy sản. Trong thời gian tới, nhóm tác giả sẽ tiếp tục nghiên cứu giải pháp nhằm tăng tốc độ xử lý phân loại và cải thiện độ chính xác của giải thuật xác định trọng lượng tôm.

TÀI LIỆU THAM KHẢO

- [1] VASEP, “Tôm Việt Nam xuất khẩu chiếm 45% sản lượng tôm toàn cầu”. URL: <https://vietnambiz.vn/tom-viet-nam-xuat-khau-chiem-45-san-luong-tom-toan-cau-29334.html>, truy cập ngày 04/9/2018.
- [2] QCM Technologies, “Máy phân cỡ tôm điện tử”. URL: <http://qcm.com.vn/>, truy cập ngày 04/9/2018.
- [3] AME Professional, “Hệ thống phân cỡ tôm theo trọng lượng”. URL: <https://ame.vn/san-pham/thiet-bi-bao-dam-chat-luong/thong-can-phan-co/>, truy cập ngày 04/9/2018.
- [4] Rahimeh, R., Mehran A., Behzad I., “A review on feature extraction techniques in face recognition”. *Signal & Image Processing: An International Journal (SIPIJ)* Vol.3, No.6, pp. 1-14, 2012.
- [5] Blasco, J., Alexios, N., Gómez, J., Moltó, E., “Citrus sorting by identification of the most common defects using multispectral computer vision”. *Journal of Food Engineering*, Vol. 83, pp. 384-393, 2007.
- [6] Jarimopas, B., Jaisin, N., “An experimental machine vision system for sorting sweet tamarind”. *Journal of Food Engineering*, Vol. 89, pp.291-297, 2008.
- [7] Mendoza, F., Aguilera, J., “Application of Image Analysis for Classification of Ripening Bananas”. *Journal of Food Science*, Vol.69, E471-E477, 2004.
- [8] Balaban, M. O., Güllün, F. S., Soriano, M.G., Ruiz, E.G., “Using Image Analysis to Predict the Weight of Alaskan Salmon of Different Species”. *Journal of Food Science*, Vol.75, E157-E162, 2010.
- [9] Pan, P.M., Li, J.P., Lv, G.L., Yang, H., Zhu, S.M., Lou, J.Z., “Prediction of shelled shrimp weight by machine vision”. *Journal of Zhejiang University. Science B* 10(8), pp.589-594, 2009.
- [10] Trương Quốc Bảo, Nguyễn Chánh Nghiệm, Nguyễn Minh Kha, Huỳnh Hoàng Giang, Võ Minh Trí, “Phát triển thuật toán xử lý ảnh để phát hiện và đếm tôm giống”. Kỷ yếu Hội nghị toàn quốc lần thứ 3 về Điều khiển và Tự động hoá – VCCA 2015, trang 323-329, 2015.
- [11] Python, Python. URL: <https://www.python.org/>, truy cập ngày 04/9/2018.
- [12] OpenCV Tutorials. URL: <https://bit.ly/2oldVxA>, truy cập ngày 04/9/2018.
- [13] Raspberry Pi 3, Raspberry Pi 3 Model B. URL: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>, truy cập ngày 04/9/2018.
- [14] OHAUS, OHAUS lab balance. URL: <https://us.ohaus.com/en-US/>, truy cập ngày 04/9/2018.
- [15] Image Thresholding, Otsu's Binarization. URL: <https://bit.ly/2MSIAI3>, truy cập ngày 04/9/2018.
- [16] Excel trendline types, equations and formulas. URL: <https://www.ablebits.com/office-addins-blog/2019/01/16/excel-trendline-types-equations-formulas/#trendline-equations-formulas>, truy cập ngày 04/9/2018.
- [17] Logitech. URL: <https://bit.ly/2YNRSWj>, truy cập ngày 04/12/2018.
- [18] Camera lens corrections. URL: <https://www.cambridgeincolour.com/tutorials/lens-corrections.htm>, truy cập ngày 04/9/2018.