

การนับจำนวนไข่หนอนไหมพันธุ์พื้นบ้านที่เหมาะสมในการฟักตัวด้วยการประมวลผลภาพ Automated Counting Indigenous Silkworm Eggs using Image Processing

สุพัตรา วยะยลุน*, วีระนันต์ วิบูลย์อรุณ, ทรงศักดิ์ มีสิทธิ์, ปฏิวดี อาสาเสน และ รัตติยา ธานี

Suphattra Wyalun*, Weeranun Vibulaut, Songsak Meesit,

Patiwat arsasen and Rattiya Thanee

สาขาวิชาเทคโนโลยีคอมพิวเตอร์ คณะเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยราชภัฏสุรินทร์

Computer Technology Department, Faculty of Industrial Technology, Surin Rajabhat University

*Email: suphatra@sru.ac.th

Received: March 19, 2024; Revised: April 20, 2024; Accepted: April 22, 2024

บทคัดย่อ

เส้นไหมที่มีคุณภาพของจังหวัดสุรินทร์มักเริ่มต้นด้วยการคัดเลือกไข่หนอนไหมเลี้ยงพันธุ์พื้นบ้านที่มีความแข็งแรง จำนวน 20,000 ฟอง วางกระจายในแนวนอนบนกระดาษกาวขนาด A4 โดยไม่ทับซ้อนกัน ซึ่งวิธีดังกล่าวจะส่งผลต่อการฟักตัว และเติบโตไปสู่หนอนไหมวัยอ่อนต่อไป ปัจจุบันศูนย์หม่อนไหมเฉลิมพระเกียรติฯ จังหวัดสุรินทร์ยังคงใช้วิธีการนับด้วยมือซึ่งมีความยากในการคัดเลือกไข่หนอนไหม เนื่องจากมีขนาดเล็กมาก ดังนั้นงานวิจัยนี้ได้นำเสนอแนวคิดในการใช้วิธีการนับจำนวนไข่ หนอนไหมพันธุ์พื้นบ้านที่เหมาะสมในการฟักตัวด้วยการประมวลผลภาพด้วย OpenCV และใช้ภาษาไพธอน และใช้รูปภาพในการทดสอบจำนวน ขนาดภาพกว้างตั้งแต่ 450 พิกเซล และความยาว 420 พิกเซล จำนวน 20 ภาพในการทดสอบ โดยมีการพัฒนา 6 ขั้นตอน ได้แก่ 1) อ่านภาพและแปลงจากปริภูมิสี RGB เป็น HSV 2) ประมวลผลในการรับช่วงสี 3) ปรับค่าสีโดยใช้ระดับสีเทา, Gaussian Blur และ Image Thresholding 4) ลบวัตถุขนาดเล็กโดยใช้การดำเนินการแปลงทางสัญญาณวิทยาการปิด 5) ค้นหาพื้นที่รูปร่าง พื้นที่เฉลี่ย และลบพื้นที่ที่มีขนาดเล็กเกินไป 6) ค้นหาผลลัพธ์โดยใช้การแปลงระยะทางและอัลกอริทึม Watershed หรือค้นหาวงกลมในแต่ละรูปร่างโดยใช้ HoughCircles ผลการศึกษาพบว่าโดยเฉลี่ยสามารถนับจำนวนไข่หนอนไหมได้ถูกต้อง ร้อยละ 71.32

คำสำคัญ : การนับจำนวนไข่หนอนไหม, โอเพ่นวีซีวิธีการตรวจจับขอบภาพ, ภาษาไพธอน

Abstract

Surin's quality silk usually begins with the selection of 20,000 healthy local silkworm eggs, distributed horizontally on an A4-sized adhesive paper without overlap. Now a day, In the Queen Sirikit Department of Sericulture, Surin, silkworm eggs are counted manually, that is difficult in the selection of the eggs, larva, because there is very small, so this research has presented an automated counting of the worm eggs silk ties folk in the pod with image processing with OpenCV and use Python and use the images in the test, with image width from 450 px and the length 420 px in a total of 20 images in the test with the development of the 6-step process: 1) Read image and convert from RGB to HSV color space 2) Get red color range 3) Gray Scale, Gaussian Blur and Image Thresholding 4) Remove small object using Morphological Transformation operation Closing 5) Find contour area, average area and remove too small areas and, 6)

Find result using Distance Transform and Watershed Algorithm or find circle in each contour using Hough Circles. The results showed that, on average, the number of silkworm eggs can be counted correctly with 71.32 percent accuracy.

Keywords : Silkworm Egg Count, Open VC, Edge Detection Method, Python Language

1. บทนำ

การคัดเลือกไข่หนอนไหมพันธุ์พื้นบ้านเป็นสิ่งจำเป็นมากสำหรับศูนย์หม่อนไหมเฉลิมพระเกียรติฯ เนื่องจากเป็นหน่วยงานที่ต้องควบคุมคุณภาพของสายพันธุ์ไหมที่มีความเหมาะสมสำหรับการเลี้ยงดูของเกษตรกรผู้เลี้ยงไหม

“ไขไหม” เป็นหนึ่งในปัจจัยพื้นฐานที่สามารถบ่งบอกความสำเร็จในการเลี้ยงไหมของเกษตรกร ทุกขั้นตอนของการผลิตไขไหมจึงต้องให้ความสำคัญและเอาใจใส่ดูแลเพื่อให้ได้ไขไหมที่ตรงตามพันธุ์ มีคุณภาพ ได้มาตรฐาน และปราศจากโรค อันจะเป็นหลักประกันให้เกษตรกรได้รับผลผลิตจริงไหมที่มีปริมาณและคุณภาพสูง โดยเฉพาะในพื้นที่จังหวัดสุรินทร์ และจังหวัดบุรีรัมย์ที่มีความต้องการไขหนอนไหมพื้นบ้านเป็นจำนวนมาก ดังนั้นศูนย์หม่อนไหมเฉลิมพระเกียรติฯ จังหวัดสุรินทร์ จึงต้องดำเนินการคัดเลือกไขหนอนไหมพื้นบ้านที่มีสุขภาพแข็งแรง ปลอดเชื้อส่งต่อไปยังเกษตรกร ปัจจุบันทางศูนย์ใช้วิธีการนับจำนวนไขด้วยมือจึงเกิดาคาดเคลื่อนในการนับจำนวนไขไหมเพื่อโรยในแผ่นกาวที่จัดเตรียมไว้ โดยมาตรฐานทั่วไปกำหนดให้ 1 แผ่นบรรจุไขไหมได้ 20,000-25,000 ฟอง จึงทำให้เสียเวลาและใช้บุคลากรจำนวนมากในการดำเนินการในขั้นตอนนี้เพื่อให้เสร็จทันต่อความต้องการของเกษตรกร [1]

อย่างไรก็ตามเพื่อศึกษาแนวทางในการการนับจำนวนไขหนอนไหมพันธุ์พื้นบ้านที่เหมาะสมกับการเจริญเติบโตเป็นไหมวัยอ่อนด้วยการประมวลผลภาพ ด้วยการตรวจสอบเส้นขอบ (Edge) การตรวจจับขอบภาพ (Edge Detection) ซึ่งตรวจสอบจากเส้นขอบลากผ่านหรือใกล้เคียงกับจุดใดแล้ววัดระดับความเข้มเทาในตำแหน่งที่ใกล้เคียงกับจุดดังกล่าว หากแตกต่างกันมากแสดงว่านั่นคือเส้นขอบ[2] ซึ่งงานวิจัยนี้ได้ใช้อัลกอริทึม Watershed [9][17] และค้นหาผลลัพธ์การแปลงระยะทางในไลบรารี OpenCV บน Python [16] และ Numpy มาใช้ในการตรวจสอบรูปทรง รูปร่าง ขอบภาพและสี [3] เพื่อนำไปประยุกต์ใช้และต่อยอดเพื่อนำไปสู่เครื่องมือ

หรือแอปพลิเคชันเพื่อรองรับการทำงานในลำดับต่อไป

2. วิธีดำเนินงาน

การค้นหาไขหนอนไหมบนภาพเพื่อนับจำนวนโดยใช้วิธีการประมวลผลภาพ สามารถนับจำนวนวัตถุที่คล้ายกันในภาพด้วยความแม่นยำ และวิธีการนับจำนวนเมื่อวัตถุในภาพเกิดการซ้อนทับกันหรือถูกบดบัง งานวิจัยนี้ใช้ภาพทั้งสิ้นจำนวน 20 ภาพ ขนาดความกว้าง 420 px ความยาว 450 px ที่มีพื้นหลังแบบมีลวดลายและไม่มีลวดลาย ตามบล็อกไดอะแกรม 6 ขั้นตอน รูปที่ 1



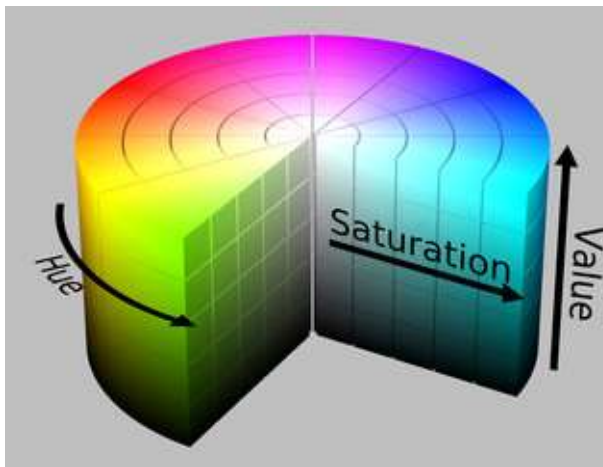
รูปที่ 1 ขั้นตอนการประมวลผลภาพ

2.1 แปลงภาพจากปริภูมิสี RGB เป็นระบบสีแบบ HSV

ผลการทดสอบการแปลงจากปริภูมิสี RGB [18] เป็นระบบสีแบบ HSV โดยเลือกใช้ภาพขนาด กว้าง 420 พิกเซล ยาว 450 พิกเซล ในการทดลอง ผลการทดสอบจะได้รูปไข่สี

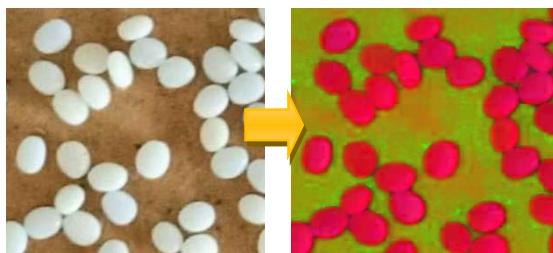
แดง ช่วงที่กำหนดสีแดงที่ต่ำสุด คือ [0,70,0] – [40,255,255] และค่าสีแดงที่สูงที่สุดระหว่าง [170,70,0] – [180,255,255] โดยช่วงที่กำหนดมีความหมายดังนี้

ตำแหน่งแรก คือ ค่าสีของสีหลัก (Hue) [15] หรือจะเรียกว่า เฉดสี, ตำแหน่งที่สอง คือ ความบริสุทธิ์ของสี (Saturation) และ ตำแหน่งที่สาม คือ ความสว่างของสี (Value) ซึ่งสามารถวัดได้ โดยค่าความเข้มของความสว่างของแต่ละสี ผลที่ได้จะเป็น ภาพไบนารี (Binary Image) หรือเรียกว่าหน้ากาก (Mask) โดยสีขาว คือ พิกเซล (Pixel) ที่มีค่าสีอยู่ในช่วงที่กำหนด ส่วน พิกเซล (Pixel) อื่น ๆ จะมีสีดำ [4] ดังรูปที่ 2



รูปที่ 2 HSV Color Model

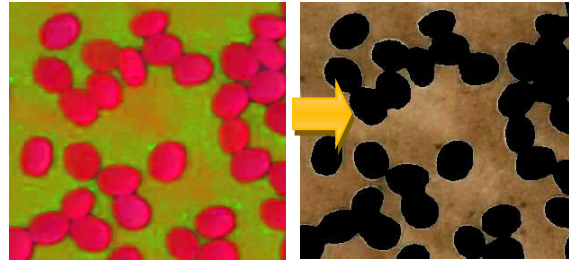
นำภาพต้นฉบับในรูปที่ 3 ก) มาแปลงปริภูมิสี RGB เป็น HSV ดังรูปที่ 3 ข)



รูปที่ 3 ก) ภาพต้นฉบับ รูปที่ 3 ข) แปลงปริภูมิสี RGB เป็น HSV

2.2 การรับค่าช่วงสีแดง (Get red color range)

ผลการทดสอบการรับค่าช่วงสีแดงที่แสดงบนปริภูมิสี HSV ภาพวัตถุจะปรากฏเป็นสีดำพื้นหลังจะเป็นสีน้ำตาล ดังรูปที่ 4



รูปที่ 4 รับช่วงค่าสีแดง

2.3 แปลงภาพเป็นสีเทา (Gray Scale) เป็นภาพเกาส์เซียนเบลอ (Gaussian Blur) และแปลงภาพสีเป็นขาว-ดำ (Image Thresholding)

นำภาพไปทำการเบลอ (Gaussian Blur)[19] ทำให้ภาพมีลักษณะที่นุ่มนวล แล้วเปลี่ยนภาพไปเป็นภาพขาว - ดำ เพื่อให้หาได้การปรับภาพให้เรียบคือค่าสีของแต่ละภาพ พิกเซลของรูปภาพถูกปรับโดยใช้ตัวกรอง Gaussian ซึ่งสร้างแมทริกซ์ย่อย (sub-matrix mask) ให้สอดคล้องกับ มาส์ก (mask) สำหรับการกรองโดยใช้สมการเกาส์เซียน [5] ในสมการ (1) ดังนี้

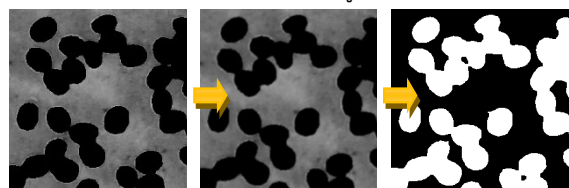
$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

โดยที่ (x, y) คือตำแหน่งของสมาชิกใน mask และ σ เป็น พารามิเตอร์และคำนวณขนาดและทิศทางการเปลี่ยนสี [6] ดังนี้

$$GM(i, j) = \sqrt{g_x^2(i, j) + g_y^2(i, j)} \quad (2)$$

$$\text{and } GD(i, j) = \arctan \left[\frac{g_y(i, j)}{g_x(i, j)} \right] \quad (3)$$

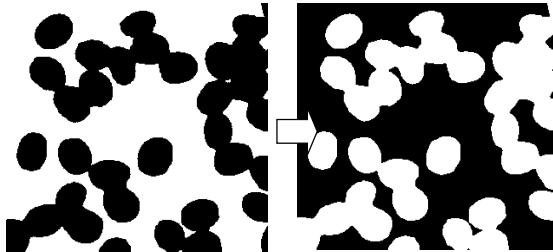
โดยที่ (i, j) คือ ตำแหน่งของพิกเซล g_x และ g_y คือ ขนาดของการเปลี่ยนแปลงค่าสีใน x และ y ตามลำดับ ค่าของ g_x และ g_y สามารถเป็นได้หลายวิธี เช่น โดยการใช้โรเบิร์ต โอเปอเรเตอร์ ใช้มาส์กของตัวปฏิบัติการแบบพิววิตหรือหน้ากากของตัวปฏิบัติการแบบ Sobel ใน อัลกอริทึม การแบ่งภาพด้วย Watershed จะนำผลจาก สมการ (2) และ (3) โดยมีผลลัพธ์ดังรูปที่ 5



รูปที่ 5 ก) ภาพสีเทา ข) ภาพเกาส์เซียนเบลอ ค) ภาพขาว-ดำ

2.4 ทดสอบการลบวัตถุขนาดเล็กโดยใช้การดำเนินการแปลงทางสัณฐานวิทยาการปิด (Morphological Transformation operation Closing)

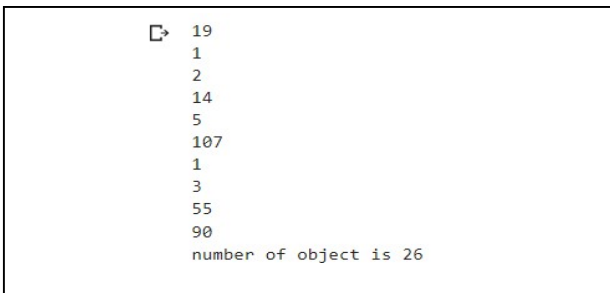
การเปลี่ยนแปลงทางสัณฐานวิทยาการปิดจะทำให้เกิดการลบภาพวัตถุขนาดเล็กออกจากภาพ ดังรูปที่ 6



รูปที่ 6 ลบภาพวัตถุขนาดเล็กออกจากภาพด้วยวิธีแปลงทางสัณฐานวิทยาการปิด

2.5 ค้นหารูปร่างในพื้นที่และคำนวณหาช่วงเฉลี่ยพื้นที่

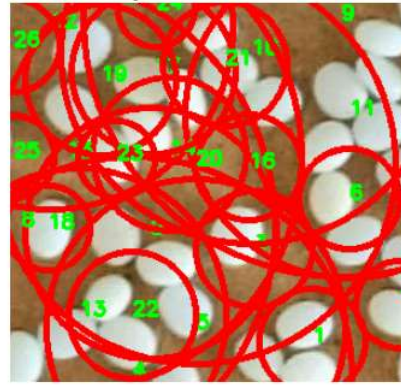
ผลทดสอบการค้นหารูปร่างภาพจะทำการค้นหาพื้นที่รูปร่างพื้นที่เฉลี่ย และลบพื้นที่ที่เล็กเกินไปแล้วนับจำนวนรูปร่างโดยกำหนดให้ภาพเล็กเป็นภาพใหญ่ที่สุดแล้วหาค่าเฉลี่ยของพื้นที่และค่าเฉลี่ยของรัศมี ผลลัพธ์ของวัตถุที่พบและผลลัพธ์คือ 26 ดังรูปที่ 7



รูปที่ 7 ผลลัพธ์การนับจำนวนวัตถุ

6. ค้นหาผลลัพธ์โดยใช้การแปลงระยะทาง (Distance Transform) และ อัลกอริทึม Watershed หรือค้นหารูปร่างในแต่ละรูปร่างโดยใช้ HoughCircles

ผลการทดสอบนี้ได้แสดงถึงการค้นหารูปร่างในภาพแต่ละพื้นที่เพราะใช้ขั้นตอนใหม่มีลักษณะเป็นวงกลม ดังรูปที่ 8



รูปที่ 8 ผลลัพธ์การค้นหารูปร่างกลมด้วย HoughCircles

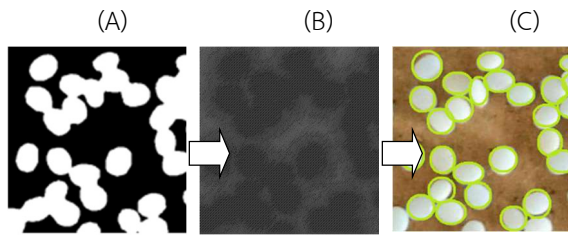
โดยมีการกำหนดรหัสเทียม (Pseudo Code) สำหรับแสดงคำสั่งที่ใช้เขียนลำดับขั้นตอนด้วย Watershed Algorithm เพื่อค้นหารูปร่างกลม HoughCircles ดังรูปที่ 9

```

1 res1 = img.copy()
2 count = 0 #result
3 for con in contours:
4     area = cv2.contourArea(con)
5     radian = int(math.sqrt(area)/math.sqrt(3.14))
6     #print(radian)
7     minRad = int(radian * 0.3)
8     maxRad = int(radian * 2)
9     radius12 = int(1.2 * radian)
10    mask_temp = np.zeros(mask.shape[:2], dtype=np.uint8)
11    gray_temp = np.zeros(gray.shape[:2], dtype=np.uint8)
12    cv2.drawContours(gray_temp, [con], 0, (255), -1)
13    #circles = cv2.HoughCircles(mask_temp, cv2.HOUGH_GRADIENT, 1, 1.2 * radian, param1=100, param2=10, minRadius=minRad, maxRadius=maxRad)
14    #circles = cv2.HoughCircles(gray_temp, cv2.HOUGH_GRADIENT, 1,
15    #                             60 * param1*0.5, param2=0.5, minRadius=minRad, maxRadius=maxRad)
16    if circles is not None:
17        circles = np.uint16(np.around(circles))
18        for i in circles[0, -1]:
19            radius = i[2]
20            if radius > radius_avg:
21                count += 1
22                center = (i[0], i[1]) # circle center
23                cv2.putText(res1, str(count), center, cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2) # Put text at center
24                cv2.circle(res1, center, radius, (0, 0, 255), 3)
25
26 print('number of object is', count)
27 cv2.imshow('res1')
    
```

รูปที่ 9 รหัสเทียม (Pseudo Code) เพื่อแสดงการนับจำนวนไข่นอนใหม่

เมื่อใช้อัลกอริทึม Watershed [9][10] ร่วมกับการแปลงระยะทาง (Distance Transform) แล้ว จะทำให้เกิดการเปลี่ยนแปลงภาพที่มีการลบภาพวัตถุขนาดเล็กออกจากภาพด้วยวิธีแปลงทางสัณฐานวิทยาการปิด รูป (A) ซึ่งในขั้นตอนนี้ จะทำการแปลงระยะทาง (Distance Transform) [13] [21] โดยแสดงผลเป็นระยะทางจากจุดภาพ (Pixel) ไปยังบริเวณที่สนใจ (Region of Interest, ROI) ที่ใกล้ที่สุด จะแสดงผลตามรูป (B) และเมื่อใช้อัลกอริทึม Watershed จะทำการสกัดภาพเบื้องหน้าและเบื้องหลังออกจากกัน ซึ่งขั้นตอนนี้จะทำให้เกิดการแบ่งแยกภาพเป็นกลุ่ม ๆ ซึ่งมีวัตถุที่สัมผัสกันและแสดงเส้นขอบสีเขียว ดังรูป (C) ดังปรากฏในรูปที่ 10 จาก Pseudo Code ผู้วิจัยได้ใช้คำสั่ง HoughCircles เพื่อค้นหารูปร่างกลม [11] [12] แล้วทำการ Threshold ทำให้สามารถนับจำนวนไข่นอนใหม่ได้



รูปที่ 10 ผลลัพธ์การใช้ Distance Transform and Watershed Algorithm

3. ผลการศึกษา

ข้อมูลที่น่ามาทดสอบและใช้ในการทดลองคือ ข้อมูลรูปภาพที่มีนามสกุลเป็น jpg และ jpeg โดยมีไฟล์ข้อมูลรูปภาพไข่นอนใหม่ในลักษณะต่าง ๆ กัน ที่นำเข้ามาทดสอบในระบบมีจำนวนทั้งสิ้น 20 ภาพ โดยมีลักษณะพื้นหลังแตกต่างกัน จากการทดลอง เพื่อทดสอบความถูกต้องและความสมบูรณ์ของภาพ ดังตารางที่ 1

ตารางที่ 1 ผลการทดสอบ

| ภาพที่ | ภาพ | จำนวนจริง | การประมวลผลภาพ | ความถูกต้อง (%) |
|--------|-----|-----------|----------------|-----------------|
| 1 | | 245 | 198 | 80.82 |
| 2 | | 88 | 86 | 97.73 |
| 3 | | 418 | 316 | 75.60 |
| 4 | | 6 | 1 | 16.67 |
| 5 | | 425 | 338 | 79.53 |
| 6 | | 465 | 220 | 47.31 |
| 7 | | 25 | 18 | 72.00 |

| ภาพที่ | ภาพ | จำนวนจริง | การประมวลผลภาพ | ความถูกต้อง (%) |
|--------|-----|-----------|----------------|-----------------|
| 8 | | 47 | 33 | 70.21 |
| 9 | | 428 | 402 | 93.93 |
| 10 | | 232 | 191 | 82.33 |
| 11 | | 43 | 38 | 88.37 |
| 12 | | 498 | 384 | 77.11 |
| 13 | | 30 | 14 | 46.67 |
| 14 | | 318 | 242 | 76.10 |
| 15 | | 203 | 164 | 80.79 |
| 16 | | 213 | 177 | 83.10 |
| 17 | | 259 | 177 | 68.34 |
| 18 | | 432 | 341 | 78.94 |
| 19 | | 880 | 598 | 67.95 |
| 20 | | 658 | 283 | 43.01 |

ค่าเฉลี่ยความถูกต้องในการนับจำนวน 71.32

ผลการทดสอบภาพไข่นอนใหม่ โดยการใช้การแบ่งส่วนรูปภาพโดยใช้อัลกอริทึม Watershed [14] ใน OpenCV Python ประยุกต์มาจากแนวคิดสันปันน้ำ และผสมผสานเข้ากับความสามารถของไลบรารี OpenCV [15] ทำให้สามารถระบุขอบเขตและแยกวัตถุในรูปภาพได้อย่างแม่นยำ

ซึ่งความแม่นยำในการนับจำนวนไข่หนอนไหมนี้ ขึ้นอยู่กับความละเอียดของภาพ ความชัดเจนของภาพที่นำมาใช้ในการทดสอบ จากภาพที่ใช้ในการทดสอบ จำนวน 20 ภาพ มีค่าความถูกต้องมากกว่า 50 เปอร์เซ็นต์ จำนวนทั้งสิ้น 16 ภาพ โดยมีค่าความถูกต้องระหว่าง 97.73 – 67.95 เปอร์เซ็นต์ และพบว่า มีจำนวน 4 ภาพ ที่มีค่าน้อยกว่า 50 เปอร์เซ็นต์ มีค่าความถูกต้องระหว่าง 16.67 – 43.01 เปอร์เซ็นต์ ทั้งนี้เนื่องจากภาพในการทดสอบไม่ชัดเจน เบลอมากเกินไป ทำให้ไม่สามารถประมวลผลได้ จึงส่งผลต่อการค้นหารูปวงกลมในแต่ละรูปร่างและไม่สามารถนับจำนวนไข่หนอนไหมได้

4. บทสรุป

การนับจำนวนไข่หนอนไหมพันธุ์พื้นบ้านเป็นสิ่งจำเป็นมาก เนื่องจากอัตราหนอนไหมฝักตัวที่เหมาะสมก่อนส่งต่อไปยังเกษตรกรมีอัตราส่วนที่ 1: 20000 ฟอง เป็นมาตรฐานที่ทางศูนย์หม่อนไหมต้องทำหน้าที่ควบคุมก่อนส่งต่อไปยังเกษตรกรผู้เลี้ยง เนื่องจากมีผลต่อการเจริญเติบโต และฝักตัวเป็นหนอนไหมที่มีคุณภาพ ดังนั้นการนับจำนวนไข่ไหมด้วยการประมวลผลภาพโดยใช้การแบ่งส่วนรูปภาพโดยใช้อัลกอริทึม Watershed ใน OpenCV Python ซึ่งเป็นเทคนิคที่ใช้กันแพร่หลายในการแบ่งส่วนภาพซึ่งเป็นไลบรารีโอเพ่นซอร์ส สอดคล้องกับ A.S. Kornilov และ Safonov [7] กล่าวว่า Watershed อัลกอริทึมเป็นเทคนิคที่ใช้กันอย่างแพร่หลายในการแบ่งส่วนภาพแบบโอเพ่นซอร์สที่เหมาะสมสำหรับการแบ่งส่วนภาพ 2 มิติและ 3 มิติ ซึ่งใช้หน่วยความจำมากในการประมวลผลโดยเฉพาะภาพที่มีขนาดใหญ่และเวลาในการประมวลผลมาก ในงานวิจัยนี้ได้ผสมผสานวิธี Distance Transform [21] และ วิธี Thresholding [22]ซึ่งเหมาะสำหรับภาพที่มีการซ้อนทับกัน และมีรูปทรงที่แตกต่างกันไปสอดคล้องกับ Chuanpit Mungkala [8] ที่ได้ใช้อัลกอริทึม Watershed เพื่อลดข้อผิดพลาดในการแบ่งส่วนภาพถ่ายทางการแพทย์ด้วยการเปลี่ยนแปลงของกลุ่มน้ำ ด้วยวิธี Thresholding [20] วิธีการแบ่งบรรทัด และวิธีการสร้างรูปร่างแบบแอคทีฟ

ผลจากการวิจัยนี้สามารถช่วยให้ผู้ปฏิบัติหน้าที่ประหยัดเวลาและมีความถูกต้อง ในการนับจำนวน ซึ่งจาก

การศึกษาดังกล่าวเป็นองค์ความรู้ที่มีประโยชน์ และสามารถนำไปพัฒนาเป็นแอปพลิเคชันเพื่ออำนวยความสะดวกในการนับจำนวนไข่ไหม หรือพัฒนาเป็นนวัตกรรมเครื่องนับไข่ไหมพร้อมกับแอปพลิเคชันควบคุมการทำงานในการแสดงผลได้ต่อไป โดยการศึกษาที่พบว่ามีปัญหาคือรูปมีความซับซ้อนมาก หรือภาพมีมิติที่ลึกทำให้การแปลงภาพไม่ได้ผลดีเท่าที่ควร ผลการวิจัยพบว่าค่าเฉลี่ยความถูกต้องในการนับจำนวนไข่หนอนไหมมีค่าเฉลี่ย 71.32 จึงควรมีการศึกษาเพิ่มเติมเพื่อให้ค่าความเชื่อมั่นมีค่าเฉลี่ยสูงขึ้นและเกิดความแม่นยำในการนับจำนวนมากยิ่งขึ้น

5. เอกสารอ้างอิง

- [1] K. Kiratiratanapruk and W. Sinthupinyo, "Silkworm Egg Image Analysis using Different Color Information for Improving Quality Inspection," *IEEE*, pp. 16, 2016.
- [2] S. Nakharacruangsak and S. Choonprawat, "An Applied Local Binary Pattern by Hyperbolic Secant for Edge Detection of Images and applied to Images Retrieval," *Journal of Energy and Environment Technology*, vol. 5, no. 1, pp. 2. Jan. 2018.
- [3] P. Sittijuk, W. Sanchana and W. Niyomdutsadee, "The recommendation using colors and pictures in package design by integrated Ontology with K-mean Algorithm," *Information Technology Journal*, Vol.17, no. 1, pp. 78, Jan. 2021.
- [4] N. Sanujit and B. Dasri, "Fore Detection System by Using Finding Countour and HSV model," *Science Technology and Innovation (STIJ)*, Vol. 1, no.2 , pp. 12, Jan. 2021.
- [5] A. H. Ibrahim, P. Kumam and W. Kumam, "A Family of Derivative-Free Conjugate Gradient Methods for Constrained Nonlinear Equations and Image Restoration," in *IEEE Access*, Vol. 8, pp. 162714–162729, 2020.

- [6] N. Tepkasetkul and J. Wetweerapong, "Modifications of Canny Method for Image Edge Detection", in Proceeding of NGRC, Khonkan University Thailand, Mar. 15, 2019, pp. 301 (in Thai).
- [7] A. S. Komilov and I. V. Safonov, "An Overview of Watershed Algorithm Implementations in Open Source Libraries." *Journal of Imaging*, Vol. 4, no. 10, pp. 123, 2018.
- [8] C. Mungkala, "Enhance Watershed Segmentation for Primary Medical Imaging", *Progress in Applied Science and Technology*, Vol. 13, no.1, pp. 7, 2023.
- [9] P. Neubert and P. Protzel, "Compact Watershed and Preemptive SLIC: On Improving Trade-offs of Superpixel Segmentation Algorithms," 22nd International Conference on Pattern Recognition, Stockholm, Sweden, pp. 996, 2014.
- [10] T. Srikanth et al., "Color Image Segmentation using Watershed Algorithm", *International Journal of Computer Science and Information Technologies*. Vol. 2, no. 5, pp. 2333, 2011.
- [11] D. Koc-Sa, S. Selim, N. Aslan & B. T. San., "Automatic citrus tree extraction from UAV images and digital surface models using circular Hough transform", *Computers and electronics in agriculture*, Vol. 150, pp. 289, 2018.
- [12] C. Chen, S. Yi, J. Mao, F. Wang, B. Zhang, F. Du, "A Novel Segmentation Recognition Algorithm of *Agaricus bisporus* Based on Morphology and Iterative Marker Controlled Watershed Transform." *Agronomy*, Vol.13, no. 2, pp. 347. 2023.
- [13] S. R. Machireddy, "Automatic Extraction of Watershed Model by using Python Programming." *Environmental Science and Engineering*. Vol 2, no. 2, pp. 47. 2023.
- [14] G. Aryotejo Adj, P. W., Ernawan, F. and Mufadhol, M., "Detecting and counting coin using opencv and watershed algorithm." In *AIP Conference Proceedings* Vol. 2738, no. 1, pp. 202, 2023.
- [15] M. Sayeed, A. Ayesha, N and M. A. Sayeed, "Detecting Crows on Sowed Crop Fields using Simplistic Image processing Techniques by Open CV in comparison with TensorFlow Image Detection API", *International Journal for Research in Applied Science and Engineering Technology*, Vol. 8, no. 3., pp. 64, 2020.
- [16] L. Wang, S. Wang, and Y. Deng, "Under water animals detecting robot based on watershed algorithm", In *2nd International Conference on Mechatronics Engineering and Information Technology*, pp. 296, Atlantis Press., 2017.
- [17] P. Sanyal and S. Barui, "The watershed transform in pathological image analysis: application in recticulocyte count from supravital stained smears" *International Journal of Research in Medical Sciences*, Vol. 7, no. 3, pp. 872, 2019.
- [18] A. A. Fauzi, "Implementation of ColorSpace, GrabCut, and Watershed Methods on Digital Image Segmentation of Coral and Fish Objects." *Journal of Computer Networks, Architecture and High Performance Computing*, Vol. 5, no. 1, pp. 88, 2003.
- [19] J. Pecchia and D. Beyer, "Pest management on US commercial mushroom farms." *Outlooks on Pest Management*, Vol. 24, no. 1, 28-29. 2013.
- [20] S. Jardim, J. António, Mora C. Graphical Image Region Extraction with K-Means Clustering and Watershed. *Journal of Imaging*, Vol. 8, no. 6, pp. 163, 2022.
- [21] R. Büchner, M. Vörös, H. Allaga, A. Varga, Bartal, A. Szekeres, A. Varga, S. Bajzát, J. Bakos-Barczi, N. Misz, A. et al, "Selection and Characterization of a *Bacillus* Strain for Potential Application in Industrial Production of White Button Mushroom." *Agronomy* 2022, Vol. 12, pp. 467, 2022.
- [22] G. Li, Y. Zhang, B. Xu and Li, X., "Image analysis and processing of skin cell injury based on opencv." In *Journal of Physics: Conference Series*, IOP Publishing, Vol. 1237, No. 3, pp. 8, 2019.