

โปรแกรมประยุกต์การพิสูจน์เอกลักษณ์จระเข้โดยใช้การประมวลผลภาพ
CROCODILE IDENTIFICATION APPLICATION USING IMAGE PROCESSING



โดย
นางสาวรุ่งนภากรณ์ ลู่เสาวภาคย์
นางสาวนิชพร สุขใส
นางสาววรรณพร ประพินพงษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมประยุกต์การพิสูจน์เอกลักษณ์จระเข้โดยใช้การประมวลผลภาพ
CROCODILE IDENTIFICATION APPLICATION USING IMAGE PROCESSING

โดย

นางสาวรุ่งนภาภรณ์	สู่เสาวภาคย์	61010907
นางสาววนิชพร	สุขใส	61010924
นางสาววรรณพร	ประพินพงษ์	61010933

อาจารย์ที่ปรึกษา
รศ. ดร. พิสิฐ บุญศรีเมือง

ปฏิญานี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2564

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง โปรแกรมประยุกต์การพิสูจน์เอกลักษณ์ระยะเข้าโดยใช้การประมวลผลภาพ

CROCODILE IDENTIFICATION APPLICATION USING IMAGE PROCESSING

ผู้จัดทำ

- | | | |
|----------------------|--------------|----------|
| 1. นางสาวรุ่งนภาภรณ์ | ลู่เสาวภาคย์ | 61010907 |
| 2. นางสาวนิชพร | สุขใส | 61010924 |
| 3. นางสาววรรณพร | ประพินพงษ์ | 61010933 |

(รศ. ดร. พิสิฐ บุญศรีเมือง)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การดำเนินปริญญาานิพนธ์เรื่อง “โปรแกรมประยุกต์การพิสูจน์เอกลักษณ์จระเข้โดยใช้การประมวลผลภาพ” จะไม่สามารถสำเร็จลุล่วงไปได้ด้วยดี หากไม่ได้รับความความอนุเคราะห์และสนับสนุนอย่างดียิ่งจากอาจารย์ที่ปรึกษาปริญญาานิพนธ์คือ รศ.ดร.พิสิฐ บุญศรีเมือง รวมทั้งรุ่นพี่ทุกท่านในห้องปฏิบัติการทดลองที่กรุณาให้คำแนะนำ คำปรึกษา และแนวทางการแก้ไขปัญหาที่เป็นประโยชน์ต่อการศึกษาค้นคว้าวิจัยให้ปริญญาานิพนธ์นี้สำเร็จสมบูรณ์ยิ่งขึ้น รวมถึงสนับสนุนสถานที่เครื่องมือ และอุปกรณ์ต่าง ๆ ที่จำเป็นต้องใช้ในการจัดทำปริญญาานิพนธ์ ขอขอบพระคุณในความห่วงใย และความปรารถนาดีให้แก่คณะผู้จัดทำเป็นอย่างยิ่ง

ขอขอบคุณคณาจารย์และเจ้าหน้าที่ประจำภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้อบรมสั่งสอน ประสิทธิ์ประสาทวิชา ความรู้ และประสบการณ์ให้แก่ผู้จัดทำ

ขอกราบขอบพระคุณ บิดา มารดา และครอบครัว ที่ให้ความรัก ความห่วงใย และเป็นกำลังใจที่สำคัญเสมอมาในยามที่เกิดปัญหา จนกระทั่งปริญญาานิพนธ์นี้สำเร็จลุล่วงไปได้ด้วยดีและที่สำคัญคือสนับสนุนให้โอกาสทางด้านการศึกษาอันมีค่ายิ่งแก่ผู้จัดทำ

นางสาวรุ่งนภาภรณ์	คู่สาวภาคย์
นางสาวนิชพร	สุขใส
นางสาววรรณพร	ประพิณพงษ์
	ผู้จัดทำ

โปรแกรมประยุกต์การพิสูจน์เอกลักษณ์จระเข้โดยใช้การประมวลผลภาพ

CROCODILE IDENTIFICATION APPLICATION USING IMAGE PROCESSING

โดย	นางสาวรุ่งนภาภรณ์	คู่เสาวภาคย์	61010907
	นางสาวนิชพร	สุขใส	61010924
	นางสาววรรณพร	ประพินพงษ์	61010933

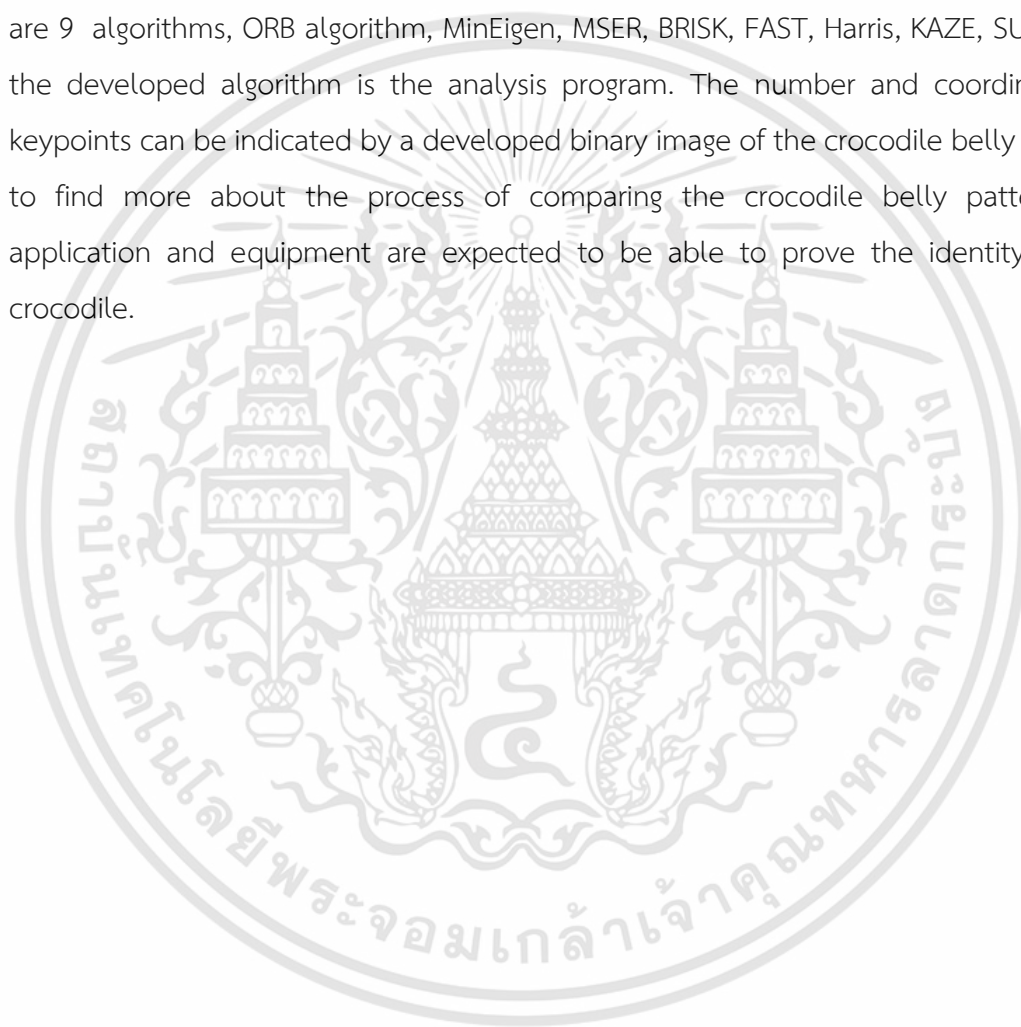
อาจารย์ที่ปรึกษา รศ.ดร.พิสิฐ บุญศรีเมือง

บทคัดย่อ

ปริญญานิพนธ์นี้นำเสนอโปรแกรมประยุกต์สำหรับพิสูจน์เอกลักษณ์จระเข้ โดยมีรายละเอียดตั้งแต่วิธีการวิเคราะห์อัลกอริทึม และการส่งข้อมูลภาพถ่ายรูปถ่ายหน้าท้องจระเข้ไปเก็บในฐานข้อมูล เพื่อรอการเปรียบเทียบชุดข้อมูลและนำไปสู่การยืนยันตัวตนต่อไป และในส่วนอัลกอริทึมพิสูจน์เอกลักษณ์จระเข้ที่ได้ศึกษามี 9 อัลกอริทึม คือ อัลกอริทึม ORB, MinEigen, MSER, BRISK, FAST, Harris, KAZE, SURF และอัลกอริทึมที่พัฒนาขึ้น คือ โปรแกรมวิเคราะห์รูปภาพแบบไบนารีของลายหน้าท้องจระเข้ที่พัฒนาขึ้นสามารถบอกจำนวนและพิกัดของจุดสำคัญ เพื่อหากระบวนการในการเปรียบเทียบลายหน้าท้องจระเข้ต่อไป ซึ่งคาดว่าโปรแกรมประยุกต์และอุปกรณ์นี้จะสามารถพิสูจน์เอกลักษณ์ของจระเข้ได้

ABSTRACT

This project presents a crocodile identification application. with details ranging from algorithm analysis methods and to sending photographic data of the crocodile belly pattern to the database to await data set comparison and further identity verification. The crocodile identification algorithm that has been studied. There are 9 algorithms, ORB algorithm, MinEigen, MSER, BRISK, FAST, Harris, KAZE, SURF, and the developed algorithm is the analysis program. The number and coordinates of keypoints can be indicated by a developed binary image of the crocodile belly pattern. to find more about the process of comparing the crocodile belly pattern this application and equipment are expected to be able to prove the identity of the crocodile.



สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	XIII
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของปริญญาานิพนธ์	1
บทที่ 2	
ทฤษฎีและหลักการที่เกี่ยวข้อง	2
2.1 จระเข้	2
2.2 MATHLAB	3
2.3 การประมวลผลภาพ (IMAGE PROCESSING)	4
2.4 Edge Detection Theory	30
2.5 Flutter	31
2.6 Dart	32
2.7 MongoOB	32
2.8 Node.js	33
2.9 JavaScript	34
2.10 JSON	34
2.11 Deep Learning	34
2.12 Convolution Neural Network (CNN)	35

สารบัญ (ต่อ)

	หน้า
บทที่ 3	
การออกแบบและการจัดทำปฏิญานินพนธ์	36
3.1 การออกแบบ	36
3.2 เครื่องมือที่ใช้ในการทดลอง	43
3.3 การจัดเก็บผลการทดลอง	43
บทที่ 4	
ผลการทดลอง	46
4.1 การทดสอบการแปลงรูปภาพจาก Physical ให้เหลือเพียงเส้นหน้า ท้องที่สนใจ	46
4.2 การทดสอบอัลกอริทึมในการวิเคราะห์รูปภาพ โดยพิจารณาจาก Gray Scale	49
4.3 การทดสอบอัลกอริทึมในการวิเคราะห์รูปภาพ โดยพิจารณาจาก Thin Image	65
4.4 การคำนวณหาเปอร์เซ็นต์ความถูกต้องการ Matching ของทุก อัลกอริทึม	81
4.5 การทดสอบอัลกอริทึม KAZE สำหรับใช้การประมวลผล	83
4.6 ทดสอบโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ (Mobile Application)	48
บทที่ 5	
สรุปผลและข้อเสนอแนะ	76
5.1 สรุปผล	76
5.2 ข้อเสนอแนะ	77
บรรณานุกรม	78
ภาคผนวก ก	
คำสั่งโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ (Mobile Application)	80
ภาคผนวก ข	
คำสั่ง MATLAB	94

สารบัญรูป

รูปที่	หน้า	
2.1	MATLAB Logo	3
2.2	ข้อมูลภาพแบบพิกเซล	5
2.3	ตัวอย่างการแสดงค่าพิกเซลเมตริกซ์	6
2.4	รูปภาพตัวอย่างก่อนทำ Histogram Equalization	7
2.5	Histogram ของรูปภาพตัวอย่างก่อนทำ Histogram Equalization	8
2.6	รูปภาพตัวอย่างหลังทำ Histogram Equalization	9
2.7	Histogram ของรูปภาพตัวอย่างหลังทำ Histogram Equalization	9
2.8	รูปตัวอย่างก่อนการทำ Gaussian Blur	10
2.9	รูปภาพตัวอย่างหลังการทำ Gaussian Blur ที่ให้ค่า $\sigma = 3$ และ 10 ตามลำดับ	11
2.10	Kernel สำหรับการประมาณค่าแบบไม่ต่อเนื่อง ที่ใช้กับตัวกรอง Laplacian ซึ่งกำหนด Laplacian โดยใช้ค่าลบสูงสุด (Negative Peak)	12
2.11	ฟังก์ชัน 2 มิติของ Laplacian ของ Gaussian (LoG) ซึ่งแกน x และ y คือ ค่าของส่วน เบี่ยงเบนมาตรฐาน	13
2.12	การประมาณแบบไม่ต่อเนื่อง (Discrete Approximation) กับฟังก์ชัน LoG ด้วย $\sigma = 1.4$	13
2.13	รูปภาพต้นฉบับ	15
2.14	รูปภาพที่ผ่านการ Global Thresholding	15
2.15	รูปภาพที่ผ่านการ Adaptive Thresholding	15
2.16	Ridge Ending และ Bifurcation	17
2.17	พิกเซลในวงกลมรอบ p	18
2.18	พีรามิดหลายชั้นที่สร้างโดย ORB	19
2.19	เวกเตอร์จากศูนย์กลางของมุม θ ถึงจุดเซ็นทรอยด์ -OC	20
2.20	เวกเตอร์คุณลักษณะซึ่งเป็นคอลัมน์	20
2.21	แพทช์คือสี่เหลี่ยมจัตุรัสของความกว้างและความสูงของพิกเซล	21
2.22	การตรวจจับมุมของ Moravec	23

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.23 จุดสนใจซึ่งไม่แปรผันกับการหมุนและการส่องสว่าง	24
2.24 หน้าต่างรอบๆ พิกเซลแต่ละพิกเซลในรูปภาพ	25
2.25 อนุพันธ์ย่อย Gaussian ใน xy	28
2.26 อนุพันธ์ย่อย Gaussian ใน y	28
2.27 พีรามิดรูปภาพที่ถูกปรับให้เรียบซ้ำ ๆ ด้วย Gaussian	29
2.28 Flutter	31
2.29 ภาษา Dart	32
2.30 MongoDB	32
2.31 Node.js	33
2.32 ข้อมูลภาพที่ซ้อนกันหลายชั้นโครงข่าย (Layer)	35
3.1 แผนผังการทำงานของโปรแกรมประยุกต์	36
3.2 ขั้นตอนในการแปลงภาพเป็นเส้น	38
3.3 แผนผังการทำงานของโปรแกรมวิเคราะห์รูปภาพ Gray Scale ของหลายหน้า ห้องของจระเข้	39
3.4 แผนผังการทำงานของโปรแกรมวิเคราะห์รูปภาพ Thin Image ของหลายหน้า ห้องของจระเข้	40
3.5 แผนผังการทำงานของโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ฝั่ง Register	41
3.6 แผนผังการทำงานของโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ฝั่ง Identify	42
3.7 ฐานข้อมูลที่ใช้เป็น MongoDB บันทึกค่า	43
4.1 รูปภาพหน้าห้องจระเข้ในชุด	45
4.2 รูปภาพ Grayscale	46
4.3 รูปภาพหน้าห้องจระเข้หลังการทำ Gaussian Blur	46
4.4 Canny Edge Detection	47
4.5 รูปที่ผ่านกระบวนการ Thresholding	47
4.6 รูปภาพที่ผ่านการ Morphological (Dilation)	48
4.7 รูปภาพที่ผ่านการ Morphological (Dilation)	48
4.8 การหา key point ภาพ Grayscale จระเข้E2/E3 ของอัลกอริทึม ORB	49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.9 การหา key point ภาพ Grayscale ละเอียดA2/A3 ของอัลกอริทึม ORB	49
4.10 การหา key point ภาพ Grayscale ละเอียดD1/D2 ของอัลกอริทึม ORB	49
4.11 การหา Matching ภาพ Grayscale ละเอียดE2/E3 ของอัลกอริทึม ORB	50
4.12 การหา Matching ภาพ Grayscale ละเอียดA2/A3 ของอัลกอริทึม ORB	50
4.13 การหา Matching ภาพ Grayscale ละเอียดD1/D2 ของอัลกอริทึม ORB	50
4.14 การหา key point ภาพ Grayscale ละเอียดE2/E3 ของอัลกอริทึม FAST	51
4.15 การหา key point ภาพ Grayscale ละเอียดA2/A3 ของอัลกอริทึม FAST	51
4.16 การหา key point ภาพ Grayscale ละเอียดD1/D2 ของอัลกอริทึม FAST	51
4.17 การหา Matching ภาพ Grayscale ละเอียดE2/E3 ของอัลกอริทึม FAST	52
4.18 การหา Matching ภาพ Grayscale ละเอียดA2/A3 ของอัลกอริทึม FAST	52
4.19 การหา Matching ภาพ Grayscale ละเอียดD1/D2 ของอัลกอริทึม FAST	52
4.20 การหา key point ภาพ Grayscale ละเอียดE2/E3 ของอัลกอริทึม BRISK	53
4.21 การหา key point ภาพ Grayscale ละเอียดA2/A3 ของอัลกอริทึม BRISK	53
4.22 การหา key point ภาพ Grayscale ละเอียดD1/D2 ของอัลกอริทึม BRISK	53
4.23 การหา Matching ภาพ Grayscale ละเอียดE2/E3 ของอัลกอริทึม BRISK	54
4.24 การหา Matching ภาพ Grayscale ละเอียดA2/A3 ของอัลกอริทึม BRISK	54
4.25 การหา Matching ภาพ Grayscale ละเอียดD1/D2 ของอัลกอริทึม BRISK	54
4.26 การหา key point ภาพ Grayscale ละเอียดE2/E3 ของอัลกอริทึม Harris	55
4.27 การหา key point ภาพ Grayscale ละเอียดA2/A3 ของอัลกอริทึม Harris	55
4.28 การหา key point ภาพ Grayscale ละเอียดD1/D2 ของอัลกอริทึม Harris	55
4.29 การหา Matching ภาพ Grayscale ละเอียดE2/E3 ของอัลกอริทึม Harris	56
4.30 การหา Matching ภาพ Grayscale ละเอียดA2/A3 ของอัลกอริทึม Harris	56
4.31 การหา Matching ภาพ Grayscale ละเอียดD1/D2 ของอัลกอริทึม Harris	56
4.32 การหา key point ภาพ Grayscale ละเอียดE2/E3 ของอัลกอริทึม KAZE	57
4.33 การหา key point ภาพ Grayscale ละเอียดA2/A3 ของอัลกอริทึม KAZE	57
4.34 การหา key point ภาพ Grayscale ละเอียดD1/D2 ของอัลกอริทึม KAZE	57
4.35 การหา Matching ภาพ Grayscale ละเอียดE2/E3 ของอัลกอริทึม KAZE	58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.36 การหา Matching ภาพ Grayscale จระเข้A2/A3 ของอัลกอริทึม KAZE	58
4.37 การหา Matching ภาพ Grayscale จระเข้D1/D2 ของอัลกอริทึม KAZE	58
4.38 การหา key point ภาพ Grayscale จระเข้E2/E3 ของอัลกอริทึม SURF	59
4.39 การหา key point ภาพ Grayscale จระเข้A2/A3 ของอัลกอริทึม SURF	59
4.40 การหา key point ภาพ Grayscale จระเข้D1/D2 ของอัลกอริทึม SURF	59
4.41 การหา Matching ภาพ Grayscale จระเข้E2/E3 ของอัลกอริทึม SURF	60
4.42 การหา Matching ภาพ Grayscale จระเข้A2/A3 ของอัลกอริทึม SURF	60
4.43 การหา Matching ภาพ Grayscale จระเข้D1/D2 ของอัลกอริทึม SURF	60
4.44 การหา key point ภาพ Grayscale จระเข้E2/E3 ของอัลกอริทึม MinEigen	61
4.45 การหา key point ภาพ Grayscale จระเข้A2/A3 ของอัลกอริทึม MinEigen	61
4.46 การหา key point ภาพ Grayscale จระเข้D1/D2 ของอัลกอริทึม MinEigen	61
4.47 การหา key point ภาพ Grayscale จระเข้A2/A3 ของอัลกอริทึม MinEigen	62
4.48 การหา Matching ภาพ Grayscale จระเข้A2/A3 ของอัลกอริทึม MinEigen	62
4.49 การหา Matching ภาพ Grayscale จระเข้D1/D2 ของอัลกอริทึม MinEigen	62
4.50 การหา key point ภาพ Grayscale จระเข้E2/E3 ของอัลกอริทึม MSER	63
4.51 การหา key point ภาพ Grayscale จระเข้A2/A3 ของอัลกอริทึม MSER	63
4.52 การหา key point ภาพ Grayscale จระเข้A2/A3 ของอัลกอริทึม MSER	63
4.53 การหา Matching ภาพ Grayscale จระเข้E2/E3 ของอัลกอริทึม MSER	64
4.54 การหา Matching ภาพ Grayscale จระเข้A2/A3 ของอัลกอริทึม MSER	64
4.55 การหา Matching ภาพ Grayscale จระเข้D1/D2 ของอัลกอริทึม MSER	64
4.56 การหา key point ภาพ Thin จระเข้E2/E3 ของอัลกอริทึม ORB	65
4.57 การหา key point ภาพ Thin จระเข้A2/A3 ของอัลกอริทึม ORB	65
4.58 การหา key point ภาพ Thin จระเข้D1/D2 ของอัลกอริทึม ORB	65
4.59 การหา Matching ภาพ Thin จระเข้E2/E3 ของอัลกอริทึม ORB	66
4.60 การหา Matching ภาพ Thin จระเข้A2/A3 ของอัลกอริทึม ORB	66
4.61 การหา Matching ภาพ Thin จระเข้D1/D2 ของอัลกอริทึม ORB	66
4.62 การหา key point ภาพ Thin จระเข้E2/E3 ของอัลกอริทึม FAST	67

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.63 การหา key point ภาพ Thin จระเข้A2/A3 ของอัลกอริทึม FAST	67
4.64 การหา key point ภาพ Thin จระเข้D1/D2 ของอัลกอริทึม FAST	67
4.65 การหา Matching ภาพ Thin จระเข้E2/E3 ของอัลกอริทึม FAST	68
4.66 การหา Matching ภาพ Thin จระเข้A2/A3 ของอัลกอริทึม FAST	68
4.67 การหา Matching ภาพ Thin จระเข้D1/D2 ของอัลกอริทึม FAST	68
4.68 การหา key point ภาพ Thin จระเข้E2/E3 ของอัลกอริทึม BRISK	69
4.69 การหา key point ภาพ Thin จระเข้A2/A3 ของอัลกอริทึม BRISK	69
4.70 การหา key point ภาพ Thin จระเข้D1/D2 ของอัลกอริทึม BRISK	69
4.71 การหา Matching ภาพ Thin จระเข้E2/E3 ของอัลกอริทึม BRISK	70
4.72 การหา Matching ภาพ Thin จระเข้A2/A3 ของอัลกอริทึม BRISK	70
4.73 การหา Matching ภาพ Thin จระเข้D1/D2 ของอัลกอริทึม BRISK	70
4.74 การหา key point ภาพ Thin จระเข้E2/E3 ของอัลกอริทึม Harris	71
4.75 การหา key point ภาพ Thin จระเข้A2/A3 ของอัลกอริทึม Harris	71
4.76 การหา key point ภาพ Thin จระเข้D1/D2 ของอัลกอริทึม Harris	71
4.77 การหา Matching ภาพ Thin จระเข้E2/E3 ของอัลกอริทึม Harris	72
4.78 การหา Matching ภาพ Thin จระเข้A2/A3 ของอัลกอริทึม Harris	72
4.79 การหา Matching ภาพ Thin จระเข้D1/D2 ของอัลกอริทึม Harris	72
4.80 การหา key point ภาพ Thin จระเข้E2/E3 ของอัลกอริทึม KAZE	73
4.81 การหา key point ภาพ Thin จระเข้A2/A3 ของอัลกอริทึม KAZE	73
4.82 การหา key point ภาพ Thin จระเข้D1/D2 ของอัลกอริทึม KAZE	73
4.83 การหา Matching ภาพ Thin จระเข้E2/E3 ของอัลกอริทึม KAZE	74
4.84 การหา Matching ภาพ Thin จระเข้A2/A3 ของอัลกอริทึม KAZE	74
4.85 การหา Matching ภาพ Thin จระเข้D1/D2 ของอัลกอริทึม KAZE	74
4.86 การหา key point ภาพ Thin จระเข้E2/E3 ของอัลกอริทึม SURF	75
4.87 การหา key point ภาพ Thin จระเข้A2/A3 ของอัลกอริทึม SURF	75
4.88 การหา key point ภาพ Thin จระเข้D1/D2 ของอัลกอริทึม SURF	75
4.89 การหา Matching ภาพ Thin จระเข้E2/E3 ของอัลกอริทึม SURF	76

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.90 การหา Matching ภาพ Thin จระเข้A2/A3 ของอัลกอริทึม SURF	76
4.91 การหา Matching ภาพ Thin จระเข้D1/D2 ของอัลกอริทึม SURF	76
4.92 การหา key point ภาพ Thin จระเข้E2/E3 ของอัลกอริทึม MinEigen	77
4.93 การหา key point ภาพ Thin จระเข้A2/A3 ของอัลกอริทึม MinEigen	77
4.94 การหา key point ภาพ Thin จระเข้D1/D2 ของอัลกอริทึม MinEigen	77
4.95 การหา Matching ภาพ Thin จระเข้E2/E3 ของอัลกอริทึม MinEigen	78
4.96 การหา Matching ภาพ Thin จระเข้A2/A3 ของอัลกอริทึม MinEigen	78
4.97 การหา Matching ภาพ Thin จระเข้D1/D2 ของอัลกอริทึม MinEigen	78
4.98 การหา key point ภาพ Thin จระเข้E2/E3 ของอัลกอริทึม MSER	79
4.99 การหา key point ภาพ Thin จระเข้A2/A3 ของอัลกอริทึม MSER	79
4.100 การหา key point ภาพ Thin จระเข้D1/D2 ของอัลกอริทึม MSER	79
4.101 การหา Matching ภาพ Thin จระเข้E2/E3 ของอัลกอริทึม MSER	80
4.102 การหา Matching ภาพ Thin จระเข้A2/A3 ของอัลกอริทึม MSER	80
4.103 การหา Matching ภาพ Thin จระเข้D1/D2 ของอัลกอริทึม MSER	80
4.104 รูปภาพหน้าห้องจระเข้ก่อนกระบวนการ Deep Learning	83
4.105 รูปภาพหน้าห้องจระเข้หลังผ่านกระบวนการ Deep Learning	83
4.106 key point ของจระเข้ตัวเดียวกันในช่วงวัย 3 เดือนและช่วงวัย 6 เดือน	84
4.107 การ Matching ของจระเข้ตัวเดียวกันในช่วงวัย 3 เดือนกับช่วงวัย 6 เดือน	84
4.108 key point ของจระเข้ตัวเดียวกันในช่วงวัย 3 เดือน	84
4.109 การ Matching ของจระเข้ตัวเดียวกันในช่วงวัย 3 เดือน	85
4.110 key point ของจระเข้ตัวเดียวกันในช่วงวัย 6 เดือน	85
4.111 การ Matching ของจระเข้ตัวเดียวกันในช่วงวัย 6 เดือน	85
4.112 key point ของจระเข้คนละตัวกันในช่วงวัย 3 เดือน	86
4.113 การ Matching ของจระเข้คนละตัวกันในช่วงวัย 3 เดือน	86
4.114 key point ของจระเข้คนละตัวกันในช่วงวัย 6 เดือนและช่วงวัย 6 เดือน	86
4.115 การ Matching ของจระเข้คนละตัวกันในช่วงวัย 6 เดือน	87

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.116	หน้าหลักโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่โดยใช้ชื่อว่า Crocodile	88
4.117	หน้าลงทะเบียนของโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่	89
4.118	รูปภาพหน้าหน้าจอที่จะใช้ที่ต้องการพิสูจน์ของโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่	90
4.119	หน้าแสดงผลรูปภาพหน้าหน้าจอที่จะใช้ที่ต้องการพิสูจน์	90



สารบัญตาราง

ตารางที่		หน้า
4.1	การหาค่า key point ของแต่ละอัลกอริทึมจากรูปภาพแบบ Grayscale	74
4.2	การหาค่าเปอร์เซ็นต์การ Matching ของแต่ละอัลกอริทึมจากรูปภาพแบบ Grayscale	74
4.3	การหาค่า key point ของแต่ละอัลกอริทึมจากรูปภาพแบบ Thin	75
4.4	การหาค่าเปอร์เซ็นต์การ Matching ของแต่ละอัลกอริทึมจากรูปภาพแบบ Thin	75
4.5	การหาค่า key point และเปอร์เซ็นต์การ Matching โดยใช้อัลกอริทึม KAZE	87

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากจระเข้เป็นสัตว์เศรษฐกิจที่บ่งชี้ถึงน้ำชนิดหนึ่งที่มีมูลค่าต่อตัวสูง เพราะอวัยวะสามารถนำมาจำหน่ายและใช้ประโยชน์ได้ทุกส่วน และเนื่องด้วยจระเข้ทั้งสามชนิดในประเทศไทย ได้แก่ จระเข้แม่น้ำจืดพันธุ์ไทย จระเข้แม่น้ำเค็ม และตะโขง เป็นสัตว์ที่อยู่ในรายชื่อของอนุสัญญาว่าด้วยการค้าระหว่างประเทศซึ่งชนิดสัตว์ป่าและพืชป่าที่ใกล้สูญพันธุ์ CITES Appendix I ซึ่งมีรายละเอียดคือ ห้ามไม่ให้ทำการค้าระหว่างประเทศโดยเด็ดขาดยกเว้นว่าจะสามารถเพาะพันธุ์ได้จนถึงรุ่นที่ 2 จะอนุญาตให้เกิดการค้าระหว่างประเทศได้ โดยมีการตรวจสอบว่าจระเข้ตัวนั้นมาจากการเพาะเลี้ยงหรือไม่ ด้วยการฝังไมโครชิพเข้าไปในตัวจระเข้ ซึ่งไมโครชิพนี้มีความยาวประมาณ 1 เซนติเมตร เส้นผ่านศูนย์กลางประมาณ 1.5 มิลลิเมตร ข้อมูลจากคนเลี้ยงบางส่วนระบุว่า การฝังไมโครชิพมีโอกาสทำให้จระเข้ตายหรือเป็นหมัน รวมถึงการสูญเสียเวลาและงบประมาณในกระบวนการต่างๆ ที่เกี่ยวกับไมโครชิพ ปริมาณงานที่นึกถึงคือโปรแกรมที่พิสูจน์เอกลักษณ์จระเข้ได้โดยใช้การประมวลผลจากภาพซึ่งจะทำให้สามารถระบุตัวจระเข้ได้โดยไม่มีความเสี่ยงที่จะมีผลข้างเคียงต่อตัวจระเข้

1.2 วัตถุประสงค์

- 1) ออกแบบและพัฒนาอัลกอริทึมสำหรับการพิสูจน์เอกลักษณ์จระเข้จากรูปลายหน้าท้องจระเข้
- 2) ออกแบบและสร้างฐานข้อมูลสำหรับพิสูจน์เอกลักษณ์จระเข้
- 3) ออกแบบและสร้างโปรแกรมประยุกต์เพื่อระบุเอกลักษณ์จระเข้จากรูปภาพ

1.3 ขอบเขตของปริมาณงาน

ออกแบบและสร้างอัลกอริทึมโดยโปรแกรม MATLAB สำหรับการพิสูจน์เอกลักษณ์จระเข้จากรูปลายหน้าท้องของจระเข้ช่วงวัย 3 เดือน และ 6 เดือน ที่ถูกถ่ายผ่านโปรแกรมประยุกต์ เปรียบเทียบกับรูปลายหน้าท้องจระเข้ในฐานข้อมูล และทำการสร้างโปรแกรมประยุกต์ที่สามารถระบุเอกลักษณ์จระเข้จากรูปภาพ เพื่อเพิ่มความแม่นยำในการระบุเอกลักษณ์ของจระเข้แต่ละตัว

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

ปริญญานิพนธ์เรื่อง “โปรแกรมประยุกต์การพิสูจน์เอกลักษณ์จระเข้โดยใช้การประมวลผลภาพ” ในบทนี้จะกล่าวถึงหลักการการทำงานของอุปกรณ์และโปรแกรมที่เกี่ยวข้องกับระบบการประมวลผลภาพที่ประกอบด้วยกระบวนการที่ใช้ในโปรแกรมระบุเอกลักษณ์จระเข้ และเทคนิค Minutiae Matching ที่ถูกนำมาประยุกต์ใช้ รวมทั้งข้อมูลของจระเข้ในประเทศไทย ซึ่งมีทฤษฎีและหลักการที่เกี่ยวข้องดังต่อไปนี้

2.1 จระเข้

จระเข้เป็นสัตว์เลื้อยคลานขนาดใหญ่ มีชื่อทางวิทยาศาสตร์ว่า Crocodylidae ซึ่งในปัจจุบันจัดเป็นสัตว์เศรษฐกิจที่มีมูลค่าสูง อันเนื่องมาจากอวัยวะทุกส่วนสามารถนำมาจำหน่ายได้ โดยเฉพาะเนื้อ หนัง และเลือดสำหรับจระเข้ในประเทศไทย มี 3 ชนิดด้วยกัน ได้แก่ จระเข้บ้านจัด สายพันธุ์ไทย จระเข้บ้านเค็ม และตะโขง ซึ่งมีความดุร้ายที่ต่างกันออกไป

จระเข้เป็นสัตว์เลื้อยคลานขนาดใหญ่ที่สุดในโลก บางชนิดเมื่อโตเต็มที่อาจมีความยาวถึง 9 เมตร มีอายุยืนถึง 70 ปี เป็นสัตว์เลือดเย็น มีหัวใจครบ 4 ห้องแต่ไม่สมบูรณ์ สมองเจริญดีกว่า สัตว์เลื้อยคลานชนิดอื่น บรรพบุรุษของจระเข้ถือกำเนิดในยุคไทรแอสสิก (Triassic) ตอนต้นประมาณ 190 ล้านปีมาแล้ว มีวิวัฒนาการมาเรื่อย ๆ

มีลักษณะโดยรวมคือ ส่วนปลายของหัวแผ่กว้างหรือเรียวยาวบางตัวปากเป็นรูปตัว v และ u ขากรรไกรยาวและกว้าง เมื่อหุบปากแล้วจะเห็นฟันซี่ที่ 4 ของขากรรไกรล่างเนื่องจากขอบปากบนตรงตำแหน่งนี้เป็นรอยหยักเว้า ส่วนปลายของขากรรไกรล่างข้างซ้ายและข้างขวาเชื่อมต่อกัน เป็นพื้นที่แคบ กระดูกเอนโทพเทอริกอยู่ชิดกับแถวของฟันที่กระดูกแมคซิลลา กระดูกพาลามีน มีก้านยื่นสั้นอยู่ทางด้านหน้าและไม่ถึงช่องในเบ้าตา พื้นผิวด้านบนของลิ้นไม่มีสารเคอราติน ต่อมา ขจัดเกล็ดบนลิ้นมีขนาดใหญ่ มีก้อนเนื้อที่ปลายปากนูนสูงที่ช่องเปิดรูจมูกเรียกว่า ขี้หมา ก้อนขี้หมา หรือหัวขี้หมา ซึ่งจะแตกต่างออกไปตามชนิดและเพศหรือขนาด โคนหางเป็นกล้ามเนื้อใหญ่และแข็งแรง เรียกว่าบ้องตันใช้ในการพาดเพื่อป้องกันตัว หางแบนยาวใช้โบกว่ายน้ำ

จระเข้ ถือเป็นสัตว์ที่อยู่บนสุดของห่วงโซ่อาหาร เนื่องจากเป็นสัตว์ผู้ล่ากินเนื้อขนาดใหญ่ที่ไม่มีศัตรูตามธรรมชาติ ยกเว้นจระเข้ในวัยอ่อน ที่ตกเป็นอาหารของสัตว์ขนาดใหญ่กว่าชนิดต่าง ๆ ได้ จระเข้ตัวโตเต็มวัยจะมีพฤติกรรมกินอาหารแบบหมุนตัว กล่าวคือเมื่อจับเหยื่อที่มีขนาดใหญ่ขณะอยู่ใต้น้ำและต้องการกินเหยื่อจะใช้ปากจับไว้และหมุนตัวเองเพื่อฉีกเนื้อเหยื่อออกเป็นชิ้น ส่วนเหยื่อที่มี

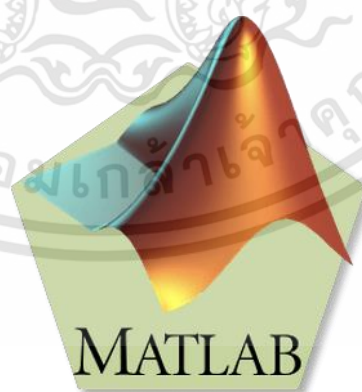
ขนาดเล็กถูกบดให้แหลกด้วยหินขนาดใหญ่โดยใช้หินต้นเหยื่ออัดแน่นกับเพดานของอุ้งปาก นอกจากนี้แล้วจะเขี่ยกลืนก้อนกรวดหรือก้อนหินเข้าไปในกระเพาะหรือลำไส้เพื่อช่วยในการบดอาหารด้วย

2.2 MATLAB

MATLAB เป็นภาษาคอมพิวเตอร์ระดับสูงที่มาพร้อมด้วยสภาพแวดล้อมการทำงานเชิงโต้ตอบ ซึ่งสามารถคำนวณคณิตศาสตร์ที่ซับซ้อนได้อย่างรวดเร็วมากกว่าภาษาคอมพิวเตอร์สมัยก่อน เช่น ภาษาซี (C) ภาษา C++ หรือ ภาษา FORTRAN

MATLAB เป็นภาษาคอมพิวเตอร์ระดับสูงที่ใช้สำหรับคำนวณเชิงตัวเลข (Numerical Computing) แสดงผลกราฟฟิก และเขียนโปรแกรมประยุกต์ทำให้เราสามารถคำนวณผลลัพธ์ พัฒนา อัลกอริทึม สร้างแบบจำลอง และแอปพลิเคชันได้ง่ายและรวดเร็วมก ภายในตัว MATLAB ประกอบด้วยภาษาคอมพิวเตอร์ กลุ่มฟังก์ชัน สำเร็จรูป (Toolbox) ในแต่ละสาขาวิชา และฟังก์ชัน พื้นฐานจำนวนมาก ทำให้การวิเคราะห์ทำได้หลากหลายวิธี พร้อมกับคำตอบที่รวดเร็ว

สำหรับการทำงาน MATLAB สามารถทำงานได้ทั้งในลักษณะของการติดต่อโดยตรงคือการเขียนคำสั่งเข้าไปทีละคำสั่ง เพื่อให้ MATLAB ประมวลผลไปเรื่อยๆ หรือสามารถที่จะรวบรวมชุดคำสั่งเรานั้นเป็นโปรแกรมก็ได้ ข้อสำคัญอย่างหนึ่งของ MATLAB ก็คือข้อมูลทุกตัวจะถูกเก็บในลักษณะของแถวลำดับคือ ในแต่ละตัวแปรจะได้รับการแบ่งเป็นส่วนย่อยเล็ก ๆ ขึ้น ซึ่งการใช้ตัวแปรเป็นแถวลำดับ ใน MATLAB เราไม่จำเป็นที่จะต้องจองมิติเหมือนกับการเขียนโปรแกรมในภาษาขั้นต่ำทั่วไป ซึ่งทำให้เราสามารถที่จะแก้ปัญหาของตัวแปรที่อยู่ในลักษณะของเมทริกซ์และเวกเตอร์ได้โดยง่าย แสดงดังรูปที่ 2.1



รูปที่ 2.1 MATLAB Logo [1]

2.3 การประมวลผลภาพ (IMAGE PROCESSING)

2.3.1 ทฤษฎีการประมวลผลภาพ

เป็นการประยุกต์ใช้งานการประมวลผลสัญญาณบนสัญญาณ 2 มิติ เช่น ภาพนิ่ง (ภาพถ่าย) หรือภาพวิดีโอ (วิดีโอ) และยังรวมถึงสัญญาณ 2 มิติอื่น ๆ ที่ไม่ใช่ภาพด้วยแนวความคิด และเทคนิค ในการประมวลผลสัญญาณ สำหรับสัญญาณ 1 มิตินั้น สามารถปรับมาใช้กับภาพได้ไม่ยาก แต่นอกเหนือจากเทคนิคจากการประมวลผลสัญญาณแล้ว การประมวลผลภาพก็มีเทคนิคและแนวความคิดที่เฉพาะ เช่น (Connectivity และ Rotation Invariance) ซึ่งจะมีความหมายกับสัญญาณ 2 มิติเท่านั้น แต่อย่างไรก็ตามเทคนิคบางอย่างจากการประมวลผลสัญญาณใน 1 มิติ จะค่อนข้างซับซ้อนเมื่อนำมาใช้กับ 2 มิติ

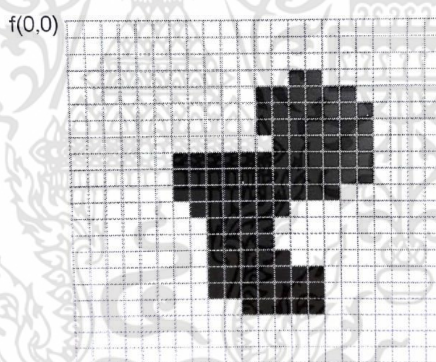
การประมวลผลภาพนั้น จะอยู่ในรูปของการประมวลผลสัญญาณแอนะล็อก (Analog) โดยใช้อุปกรณ์ปรับแต่งแสง (Optics) ซึ่งวิธีเหล่านั้นก็ไม่ได้หายสาบสูญ หรือเลิกใช้ไป ยังมีใช้เป็นส่วนสำคัญ สำหรับการประยุกต์ใช้งานบางอย่าง เช่น ฮอโลกราฟี (Holography) แต่ เนื่องจากอุปกรณ์คอมพิวเตอร์ในปัจจุบันราคาถูกลงและเร็วขึ้นมาก การประมวลผลภาพดิจิทัล (Digital Image Processing) จึงได้รับความนิยมมากกว่า เพราะการประมวลผลทำได้ซับซ้อนขึ้น แม่นยำ และง่ายในการลงมือปฏิบัติ การประมวลผลภาพดิจิทัล หมายถึง การเรียกใช้ขั้นตอนหรือกรรมวิธีใดมากระทำกับภาพ โดยมีวัตถุประสงค์เพื่อปรับปรุงคุณภาพของภาพ ให้ได้ภาพใหม่ที่มีคุณสมบัติตามต้องการ เช่น ความคมชัด หรือการประหยัดพื้นที่ในการเก็บข้อมูล หรือใช้สำหรับการประมวลผลระดับสูง เช่น การจดจำรูปร่างลักษณะให้ได้อย่างแม่นยำ การประมวลผลภาพแบ่งออกได้เป็น 2 ระดับ คือ

1. การประมวลผลภาพระดับต่ำ (Low Level Image Processing) เป็นการประมวลผลขั้นแรกสุดก่อนที่จะนำไปสู่การประมวลผลภาพระดับสูงต่อไป นั่นคือหลังจากที่ได้ภาพมา ซึ่งภาพที่ได้จะมีองค์ประกอบต่าง ๆ มากมาย รวมถึงสิ่งที่ไม่ต้องการด้วย ในที่นี้จะเรียกว่า "สัญญาณรบกวน" ซึ่งทำให้ภาพที่ได้มีคุณภาพไม่ดี ยังไม่สามารถนำไปใช้ในการประมวลผลได้ ดังนั้น การประมวลผลภาพระดับต่ำ จึงต้องมีการกำจัดสัญญาณรบกวนออก การทำภาพให้ชัด (High Pass Filter) การหาขอบภาพ (Edge Detection) การแปลงไบนารีอิเมจการแบ่งแยกรูปร่าง วัตถุ (Image Segmentation) เป็นต้น เพื่อหาค่าตัวแปรต่าง ๆ มาอธิบายข้อมูลภาพ และมีวัตถุประสงค์ที่จะนำตัวแปรเหล่านี้มาใช้ในการประมวลผลภาพระดับสูงต่อไป

2. การประมวลผลภาพระดับสูง (High Level Image Processing) เป็นการทำให้คอมพิวเตอร์รู้จักและเข้าใจภาพนั้นได้ เช่น การจดจำใบหน้าคน หรืออาจจะเป็นการจดจำตัวอักษร เป็นต้น ความแตกต่างของการประมวลผลภาพระดับต่ำและระดับสูงคือ ข้อมูลที่นำมาใช้ในการประมวลผล โดยการประมวลผลระดับต่ำจะใช้ค่าความสว่างหรือความเข้มแสงโดยตรง

ส่วนการประมวลผลภาพระดับสูงข้อมูลที่น่ามาใช้ในการประมวลผลจะถูกแสดงในรูปแบบของสัญลักษณ์ โดยสัญลักษณ์เหล่านี้จะแสดงถึงสิ่งต่าง ๆ ที่อยู่ในภาพและการใช้ตัวแปรที่ได้จากการประมวลผลภาพระดับต่ำมาอธิบายถึงสัญลักษณ์เหล่านี้ การประมวลผลภาพระดับสูงนั้น ส่วนใหญ่มักจะใช้ทฤษฎีต่าง ๆ เข้ามาใช้เป็นตัวช่วยในการทำงาน หรือเป็นหัวใจของโปรแกรม เช่น ตรรกศาสตร์คลุมเครือ (fuzzy logic) โครงข่ายประสาท (Neural Network) อย่างที่กล่าวไปแล้วว่าการประมวลผลภาพระดับสูง จำเป็นต้องใช้ข้อมูลที่ได้มาจากการประมวลผลระดับต่ำ ดังนั้นจะเห็นได้ว่าการประมวลผลภาพระดับต่ำมีความสำคัญมากสำหรับการทำให้คอมพิวเตอร์รู้จักและเข้าใจภาพได้

ในส่วนการแทนภาพด้วยข้อมูลแบบดิจิทัลจากที่ได้กล่าวไปแล้วในตอนต้น ข้อมูลภาพแบบดิจิทัลเป็นภาพที่ถูกตัดแปลงมาจากภาพแบบต่อเนื่อง หรือภาพแอนะล็อก ให้อยู่ในรูปตัวเลข หรือภาพดิจิทัลด้วยวิธีการดิจิไทเซชัน (Digitization) โดยภาพแอนะล็อกจะถูกแบ่งเป็นพื้นที่สี่เหลี่ยมเล็ก ๆ ที่เรียกว่า จุดภาพ (Pixels) โดยแต่ละพิกเซลจะใช้ (X,Y) ในการระบุตำแหน่ง การแสดงข้อมูลภาพดิจิทัลสามารถอธิบายได้ด้วยแมทริกซ์ $(M \times N)$ และให้จุดต่าง ๆ ที่อยู่ ในแมทริกซ์เป็นจุดที่พิกัด (X,Y) ใด ๆ เป็นส่วนประกอบของภาพ แสดงดังรูปที่ 2.2



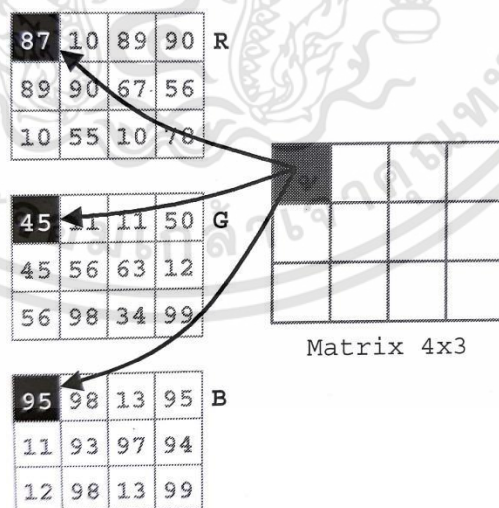
รูปที่ 2.2 ข้อมูลภาพแบบพิกเซล [2]

เมื่อเราเปรียบเทียบระหว่างภาพและพิกเซลแมทริกซ์ จะเห็นว่าจุดกำเนิดของภาพจะอยู่ที่มุมล่างซ้าย แต่จุดกำเนิดของพิกเซลจะอยู่ที่มุมบนซ้าย ซึ่งจะเป็นลักษณะการประมวลผลภาพในกราฟิกของคอมพิวเตอร์ที่ได้กล่าวไปแล้วก่อนหน้านี้

ค่าของพิกเซลหรือฟังก์ชัน (x,y) ณ จุดใด ๆ จะแสดงได้ด้วยคู่ของความเข้มข้นของแสง ซึ่งอาจแบ่งได้หลายระดับ ถ้ามี 2 ระดับก็จะเป็นแค่ 0 กับ 1 จากรูปที่ 5 จุดต่าง ๆ ที่แสดงอยู่ในพิกัดก็คือ พิกเซล ซึ่งคือความสว่าง (Luminance, L) ของภาพ ถ้าภาพนั้นเป็นภาพขาวดำขนาด 8 บิต จะมีค่า L เท่ากัน หรือเท่ากับ 256 คือ ตั้งแต่ระดับ 0 จนถึง 255 บางครั้งคู่ความสว่าง (L) อาจหมายถึงระดับความละเอียดของภาพ (Image resolution) ถ้าพิกเซลเป็นภาพขาวดำ จะอ่านค่าภาพดิจิทัลในรูปแบบแมทริกซ์ 2 มิติ ขนาด $(M \times N)$ ได้ดังสมการ (2.1)

$$f(x) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,n-1) \\ f(1,0) & f(1,1) & \dots & f(1,n-1) \\ \dots & \dots & \dots & \dots \\ f(m-1,0) & f(m-1,1) & \dots & f(m-1,n-1) \end{bmatrix}_{m \times n} \quad (2.1)$$

โดยค่าที่ $f(x,y)$ จะอยู่ในช่วง 0 ถึง 255 เช่น อ่านค่าพิกเซลจากภาพหนึ่งได้ $f(x,y)$ เท่ากับ 10 แสดงว่าจุดพิกเซลนั้นมีความสว่างน้อยมากหรือค่อนข้างจะดำ ถ้าอ่านได้เป็น 255 แสดงว่าจุดพิกเซลนั้นมีความสว่างมากหรือเป็นสีขาว จะช่วยให้เข้าใจการแสดงค่าพิกเซลในแมทริกซ์มากขึ้น เริ่มต้นด้วย พิกัด $f(x,y) = f(0,0)$ ค่าของพิกเซลที่ได้ก็จะเป็นการผสมสีระหว่างค่าของแม่สีทั้งสาม ซึ่งได้แก่แดง (R) เขียว (G) และน้ำเงิน (B) ซึ่งในที่นี้ก็จะมามีค่าเท่ากับ (87, 45, 95) แสดงดังรูปที่ 2.3



รูปที่ 2.3 ตัวอย่างการแสดงค่าพิกเซลแมทริกซ์ [3]

โดยทั่วไปแล้วภาพที่นำมาใช้ในการประมวลผลภาพดิจิทัลนั้น แบ่งออกได้เป็นสองประเภทหลักๆ คือ

1. ภาพเคลื่อนไหว โดยทั่วไปแล้วภาพเคลื่อนไหวคือ ภาพนิ่งที่นำมาแสดงต่อ ๆ กันแบบต่อเนื่องจะต้องใช้รูปภาพอย่างน้อย 24 รูปต่อหนึ่งวินาที เนื่องจากสายตาของคนเราเมื่อนำภาพนิ่งมาฉายติดต่อกันมากกว่า 24 รูปต่อหนึ่งวินาทีแล้ว ก็จะมองภาพนั้นว่าเป็นภาพเคลื่อนไหว เพราะสายตาเราแยกไม่ออกเนื่องจากมีความเร็วมากเกินไป แต่หากนำภาพนิ่งมาฉายติดต่อกันน้อยกว่า 24 รูปต่อหนึ่งวินาทีแล้ว เราจะมองเห็นภาพนั้นไม่ต่อเนื่อง

2. ภาพสีหรือภาพนิ่ง ภาพที่นำเข้ามาประมวลผลในคอมพิวเตอร์นั้น ถ้าในระบบ RGB ก็จะใช้ความเข้มแสงสีแดง เขียว และน้ำเงินความหมายของภาพนิ่งก็คือ มีอยู่เพียงภาพเดียว (ถ้าเป็นภาพเคลื่อนไหวก็จะประกอบไปด้วยภาพนิ่งหลายๆ ภาพ) ภาพนิ่งที่นำมาใช้ก็มีอยู่หลายรูปแบบ ไม่ว่าจะเป็น .bmp หรือ .jpg เป็นต้นส่วนการเลือกใช้นั้นก็แล้วแต่ความเหมาะสม ส่วนใหญ่จะเป็น .bmp เพราะไม่ต้องถอดรหัสก่อน เนื่องจากภาพที่เป็น .jpg มีการบีบอัดภาพให้มีขนาดเล็ก ดังนั้นหากจะนำมาใช้ต้องคลายข้อมูลออกมาก่อนที่จะนำภาพไปประมวลผลต่อไป

2.3.2 ตัวอย่างกระบวนการที่ใช้ในการประมวลผลภาพ

2.3.2.1 Histogram Equalization

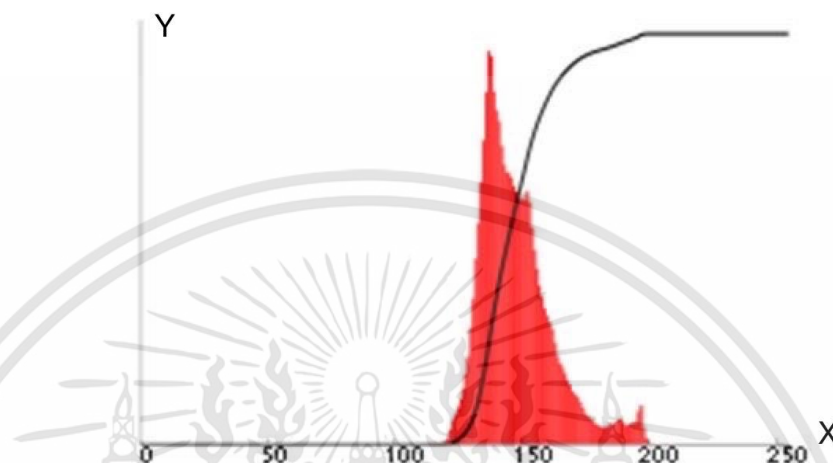
ฮิสโตแกรมเป็นการแสดงการกระจายความเข้มสีของภาพแบบกราฟิก หรือกล่าวคือจำนวนของพิกเซลสำหรับแต่ละค่าความเข้มที่พิจารณา แสดงดังรูปที่ 2.4 รูปภาพก่อนทำ Histogram Equalization



รูปที่ 2.4 รูปภาพตัวอย่างก่อนทำ Histogram Equalization [4]

แกน X หมายถึงระดับโทนสี (0 คือ สีดำ- 255 คือ สีขาว) แกน Y หมายถึงจำนวนพิกเซลในภาพ กราฟสีดำคือ Cumulative Histogram และกราฟสีแดงคือ Corresponding Histogram ดังนั้นฮิสโตแกรมดังกล่าวนี้จะแสดงจำนวนพิกเซลสำหรับแต่ละระดับของค่าคอนทราสต์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(สีดำไปจนถึงสีขาวในแนวแกน x) และเมื่อมีพิกเซลสูงสุดมีจำนวนมาก ค่าคอนทราสต์ ณ ระดับนั้นก็ จะสูงขึ้นด้วย แสดงดังรูปที่ 2.5



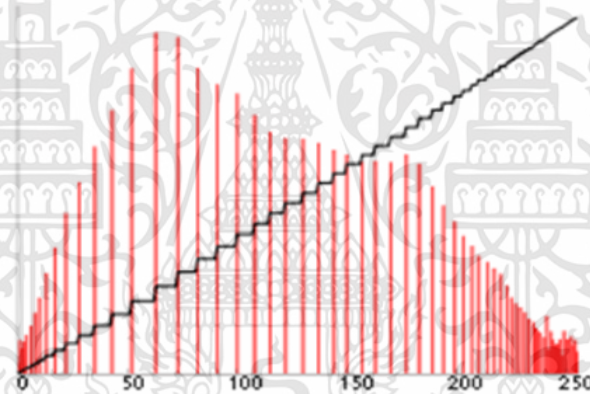
รูปที่ 2.5 Histogram ของรูปภาพตัวอย่างก่อนทำ Histogram Equalization [4]

Histogram Equalization เป็นเทคนิคการประมวลผลภาพของคอมพิวเตอร์ที่ใช้ในการปรับปรุงค่าคอนทราสต์ (Contrast) ในภาพ สามารถทำได้โดยการกระจายค่าความเข้มที่พบบ่อยที่สุดอย่างมีนัยสำคัญ เช่น ยืดช่วงความเข้มของภาพ ซึ่งวิธีนี้จะเพิ่มคอนทราสต์โดยรวมของรูปภาพ และข้อมูลที่ใช้งานได้จะถูกแสดงด้วยค่าคอนทราสต์ที่ใกล้เคียงกัน หรือก็คือการทำให้พื้นที่ที่มีค่าคอนทราสต์ ณ พื้นที่นั้นต่ำ ได้ค่าคอนทราสต์ที่สูงขึ้นมา หรือก็คือมีความคมชัดเพิ่มขึ้นนั่นเอง

ฮิสโตแกรมสีของรูปภาพจะแสดงจำนวนของพิกเซลในแต่ละองค์ประกอบสี แต่ Histogram Equalization ไม่สามารถใช้กับรูปภาพที่เป็นระบบสีของ RGB เนื่องจากจะทำให้เกิดการเปลี่ยนแปลงอย่างมากในสมมูลสีของภาพ อย่างไรก็ตามหากภาพถูกแปลงเป็นระบบสีอื่น เช่น HSL / HSV อัลกอริทึมนี้จะสามารถนำไปใช้กับความสว่างหรือค่าในพิกเซลโดยไม่ส่งผลให้เกิดการเปลี่ยนแปลงของสีและความอิ่มตัวของภาพ แสดงดังรูปภาพที่ 2.6 และรูปที่ 2.7



รูปที่ 2.6 รูปภาพตัวอย่างหลังทำ Histogram Equalization [5]



รูปที่ 2.7 ฮิสโตแกรมของรูปภาพตัวอย่างหลังทำ Histogram Equalization [5]

2.3.2.2 Gaussian Blur

ในการประมวลผลภาพเกาส์เซียนเบลอ (Gaussian Blur) เป็นผลมาจากการทำให้ภาพเบลอโดยฟังก์ชันเกาส์เซียน เป็นเอฟเฟกต์ชนิดหนึ่งที่ใช้กันอย่างแพร่หลายในซอฟต์แวร์กราฟิก ซึ่งการทำงานโดยทั่วไปจะใช้เพื่อลดสัญญาณรบกวนของภาพ ด้วยเทคนิคการจัดการภาพแบบนี้จะทำให้ภาพเป็นภาพเบลอที่คล้ายกับการดูภาพผ่านหน้าจอบริเวณแสง ซึ่งแตกต่างอย่างชัดเจนจากเอฟเฟกต์โบเก้ (Bogeh) ที่เกิดจากการที่เลนส์ที่อยู่นอกโฟกัสหรือเงาของวัตถุที่อยู่ในระดับความสว่าง ตามปกติการปรับภาพแบบเกาส์เซียนนิยมใช้มากในขั้นตอนของการ Pre-Processing ในอัลกอริทึมการวิเคราะห์ภาพในคอมพิวเตอร์เพื่อปรับปรุงโครงสร้างภาพในระดับต่างๆ รวมไปถึงเพื่อดูการแสดงพื้นที่สเกลและการนำพื้นที่สเกลดังกล่าวไปใช้

การใช้เกาส์เซียนเบลอมีผลในการลดองค์ประกอบความถี่สูงของภาพ ดังนั้นเกาส์เซียนเบลอ จึงเป็นวงจรกรองความถี่ต่ำผ่าน (low pass filter) ชนิดหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกาส์เซียนเบลอเป็นตัวกรองการเบลอภาพที่ใช้ฟังก์ชันแบบเกาส์ ซึ่งแสดงถึงการแจกแจงปกติ (Normal Distribution) ในสถิติศาสตร์ สำหรับการคำนวณเพื่อนำไปใช้กับแต่ละพิกเซลในภาพ จะมีสูตรคำนวณของฟังก์ชันเกาส์เซียน (Gaussian Function) ในมิติเดียว ดังสมการที่ (2.2)

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (2.2)$$

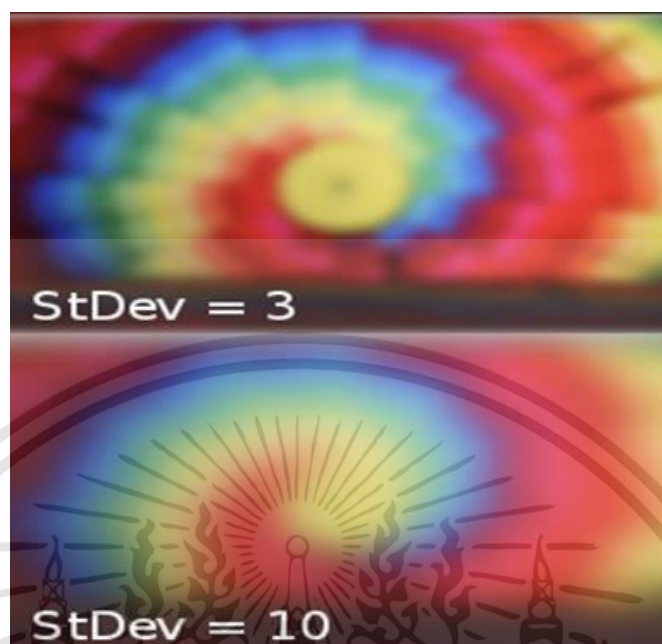
ในสองมิติ ฟังก์ชันเกาส์เซียนเป็นผลลัพธ์ของฟังก์ชันเกาส์เซียนทั้งสองในแต่ละมิติ มีสูตรคำนวณดัง สมการที่ (2.3)

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}} \quad (2.3)$$

โดยที่ x คือระยะห่างจากจุดกำเนิดในแกนนอน y คือระยะทางจากจุดกำเนิดในแกนตั้ง และ σ คือ ค่าส่วนเบี่ยงเบนมาตรฐานของการแจกแจงแบบเกาส์เซียน แสดงดังรูปภาพที่ 2.8 รูปตัวอย่างก่อน ทำ Gaussian Blur และรูปภาพที่ 2.9 แสดงรูปภาพหลังการปรับค่า σ



รูปที่ 2.8 รูปตัวอย่างก่อนการทำ Gaussian Blur [6]



รูปที่ 2.9 รูปภาพตัวอย่างหลังการทำ Gaussian Blur ที่ให้ค่า $\sigma = 3$ และ 10 ตามลำดับ [6]

2.3.2.3 Laplacian / Laplacian of Gaussian

ลาปลาเซียนเป็นการวัดแบบไอโซโทรปิก (Isotropic) 2 มิติของอนุพันธ์เชิงพื้นที่อันดับที่ 2 ของภาพ ซึ่งลาปลาเซียนของภาพจะเน้นพิจารณาบริเวณที่มีการเปลี่ยนแปลงความเข้มอย่างรวดเร็ว และมักใช้สำหรับการตรวจจับขอบ (Edge Detection)

ตัวกรองแบบลาปลาเซียน (Laplacian) มักจะถูกนำไปใช้กับภาพที่ได้รับการปรับให้เรียบด้วยบางสิ่งบางอย่างที่ใกล้เคียงกับฟิลเตอร์การเบลอแบบเกาส์เซียน เพื่อลดความไวต่อสัญญาณรบกวน ซึ่งโดยปกติแล้วโอเปอเรเตอร์ (Operator) จะใช้ภาพสีเทาระดับเดียวเป็นอินพุต และสร้างภาพสีเทาระดับอื่นๆออกมาเป็นเอาต์พุต

ลาปลาเซียนนิยามได้เป็น $L(x, y)$ ของภาพที่มีค่าความเข้มสีของพิกเซลที่ $I(x, y)$ ซึ่งสามารถคำนวณได้จากสมการที่ (2.4)

$$L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \quad (2.4)$$

เนื่องจากภาพอินพุตถูกแสดงเป็นชุดของพิกเซลที่ไม่ต่อเนื่อง (Discrete) เราจึงต้องค้นหา Discrete Convolution Kernel ซึ่งสามารถประมาณอนุพันธ์อันดับสองในนิยามของ Laplacian ได้ โดย Kernel ขนาดเล็กที่ใช้โดยทั่วไปจะแสดงดังรูปที่ 2.10

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

รูปที่ 2.10 Kernel สำหรับการประมาณค่าแบบไม่ต่อเนื่อง ที่ใช้กับตัวกรอง Laplacian ซึ่งกำหนด Laplacian โดยใช้ค่าลบสูงสุด (Negative Peak) [7]

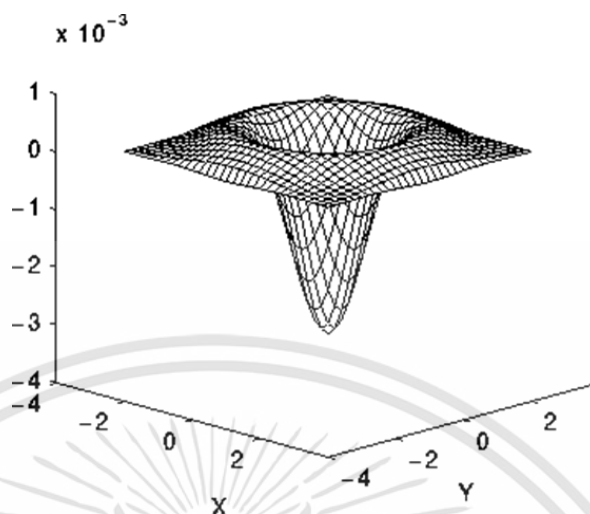
การใช้หนึ่งใน Kernel เหล่านี้จะทำให้สามารถคำนวณลาปลาเซียน ได้โดยใช้วิธีการคอนโวลูชันแบบมาตรฐาน (Standard Convolution Method) เนื่องจาก Kernel เหล่านี้มีความไวต่อสัญญาณรบกวนมาก ในการแก้ปัญหานี้มักจะใช้วิธีการที่จะนำรูปภาพไปผ่านฟิลเตอร์การเบลอแบบเกาส์เซียนก่อนที่จะใช้ฟิลเตอร์ Laplacian การทำแบบนี้จะลดองค์ประกอบสัญญาณรบกวนความถี่สูงก่อนที่จะถึงขั้นตอนการแยกความแตกต่าง (Differentiation)

ในความเป็นจริงเนื่องจากการดำเนินการคอนโวลูชัน (Convolution) นั้นเชื่อมโยงกันทำให้เราสามารถคอนวอลฟต์ (Convolve) ฟิลเตอร์การเบลอแบบเกาส์เซียนกับตัวกรองแบบลาปลาเซียน (Laplacian) จากนั้นจึงคอนวอลฟต์ (Convolve) ตัวกรองแบบไฮบริดนี้กับภาพเพื่อให้ได้ผลลัพธ์ตามที่ต้องการ การใช้วิธีนี้มีข้อดีสองประการ ได้แก่

1. เนื่องจากทั้ง Kernel ของ Gaussian และ Laplacian มักจะเล็กกว่าภาพมาก วิธีนี้จึงใช้การดำเนินการทางคณิตศาสตร์ที่ไม่ยุ่งยากมากนัก
2. เคอร์เนลของ LoG (Laplacian of Gaussian) สามารถทำการคำนวณล่วงหน้าได้ ดังนั้นจึงจำเป็นต้องทำการ convolution เพียงแค่ครั้งเดียวในขณะที่ใช้งานรูปภาพ

ฟังก์ชัน LoG (Laplacian of Gaussian) 2-D มีศูนย์กลางอยู่ที่จุดศูนย์กลางและมีค่า เบี่ยงเบนมาตรฐานแบบเกาส์ มีรูปแบบดังสมการที่ (2.5) และแสดงดังรูปที่ 2.11

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}} \quad (2.5)$$



รูปที่ 2.11 ฟังก์ชัน 2 มิติของ Laplacian ของ Gaussian (LoG) ซึ่งแกน x และ y คือค่าของส่วน
เบี่ยงเบนมาตรฐาน [8]

โดยเคอร์เนลชนิดไม่ต่อเนื่อง (Discrete Kernel) ที่ประมาณฟังก์ชัน LoG นี้ (สำหรับค่า
 $\sigma = 1.4$) แสดงดังรูปที่ 2.12

0	1	1	2	2	2	1	1	0
1	2	4	5	5	5	4	2	1
1	4	5	3	0	3	5	4	1
2	5	3	-12	-24	-12	3	5	2
2	5	0	-24	-40	-24	0	5	2
2	5	3	-12	-24	-12	3	5	2
1	4	5	3	0	3	5	4	1
1	2	4	5	5	5	4	2	1
0	1	1	2	2	2	1	1	0

รูปที่ 2.12 การประมาณแบบไม่ต่อเนื่อง (Discrete Approximation)
กับฟังก์ชัน LoG ด้วย $\sigma = 1.4$ [9]

เป็นเรื่องต้องทราบว่าในขณะที่ Gaussian ถูกทำให้แคบลงเรื่อย ๆ เคอร์เนล LoG จะกลายเป็นเหมือนกับเคอร์เนล Laplacian ธรรมดาที่แสดงในรูปที่ 2.14 เป็นเพราะว่าการปรับภาพให้เรียบด้วย Gaussian ที่แคบมาก ($\sigma < 0.5$ พิกเซล) บนตารางที่ไม่ต่อเนื่อง (Discrete Grid) จะไม่มีผลใด ๆ เกิดขึ้น ดังนั้นบนตารางที่ไม่ต่อเนื่อง ลاپลาเซียนแบบง่ายจึงถูกมองว่าเป็นกรณีจำกัดของ LoG สำหรับ Gaussians ที่แคบมากๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.4 Adaptive Threshold

การทำ Thresholding ใช้เพื่อแบ่งส่วนภาพโดยใช้หลักการตั้งค่าพิคเซลทั้งหมดที่มีค่าความเข้มสูงกว่าค่า Threshold เป็นค่าโดดเด่น (Foreground) และพิคเซลที่เหลือทั้งหมดเป็นค่าพื้นหลัง (Background)

ในขณะที่การดำเนินการแบบ Conventional Threshold ใช้ค่า Global Threshold สำหรับการพิจารณาทุกพิคเซล แต่ Adaptive Threshold จะเปลี่ยนค่า Threshold ของภาพได้แบบไดนามิก (Dynamic) Thresholding ในยุคหลังๆ ที่ซับซ้อนมากขึ้นนี้ สามารถรองรับสภาพแสงที่เปลี่ยนแปลงในภาพ เช่น สิ่งรบกวนที่เกิดขึ้นจากการไล่ระดับสีของแสงหรือเงาที่มีสีเข้ม

Adaptive Threshold โดยทั่วไปจะใช้ภาพ Grayscale หรือภาพสีเป็นอินพุตและในการใช้งานที่ง่ายที่สุดจะแสดงผลภาพไบนารีที่แสดงถึงการแบ่งส่วนแต่ละพิคเซลในภาพจำเป็นจะต้องคำนวณค่า Threshold ซึ่งหากค่าของพิคเซลต่ำกว่าค่า Threshold มันจะถูกตั้งค่าเป็นค่าพื้น ถ้าสูงกว่าก็จะเป็นค่าโดดเด่น

มีสองวิธีหลักในการคำนวณหา Threshold ได้แก่ วิธีคำนวณหาแบบ Chow และ Kaneko และ (ii) แบบ Local Thresholding ซึ่งทฤษฎีเบื้องหลังทั้งสองวิธีนี้คือ บริเวณที่เล็กกว่าของภาพมีแนวโน้มที่จะประมาณการณได้ว่าการส่องสว่างที่สม่ำเสมอจึงเหมาะแก่การทำ Threshold ดังนั้น Chow และ Kaneko จึงแบ่งภาพออกเป็น Array ของ Array ที่ทับซ้อนกัน (แบ่งภาพออกเป็นภาพย่อย) แล้วหาเกณฑ์ที่เหมาะสมสำหรับแต่ละชุดย่อยโดยการตรวจสอบจากฮิสโตแกรม เกณฑ์สำหรับแต่ละพิคเซลจะพบได้โดยการแก้ไขผลลัพธ์ของภาพย่อย

อีกทางเลือกหนึ่ง คือการคำนวณหาค่า Local Threshold คือการตรวจสอบค่าความเข้มของพื้นที่ใกล้เคียงในแต่ละพิคเซล ค่าสถิติตรงนี้จะเหมาะสมที่สุดหรือไม่จะขึ้นอยู่กับภาพอินพุตเป็นส่วนใหญ่ ฟังก์ชันที่ง่ายและกระชับประกอบด้วยค่าเฉลี่ยของการกระจายความเข้มในพื้นที่ (Local Intensity Distribution) ดังสมการที่ (2.6)

$$T = \text{mean} \quad (2.6)$$

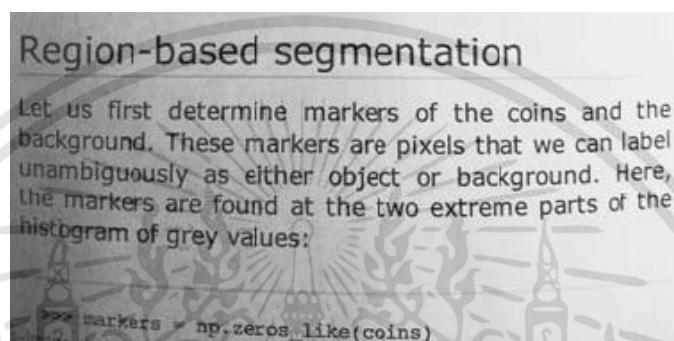
และค่ามัธยฐาน ดังสมการที่ (2.7)

$$T = \text{median} \quad (2.7)$$

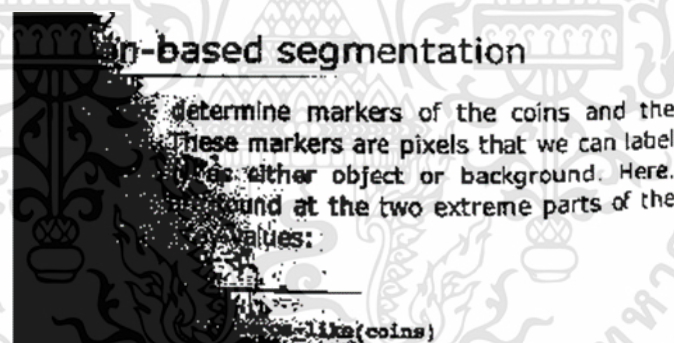
หรือค่าเฉลี่ยของค่าต่ำสุดและค่าสูงสุด ดังสมการที่ (2.8)

$$T = \frac{\text{min} + \text{max}}{2} \quad (2.8)$$

การกำหนดขนาดของพื้นที่ใกล้เคียงจะต้องมีขนาดใหญ่พอที่จะครอบคลุมค่าของพิกเซลโดดเด่นและพิกเซลพื้นหลังได้อย่างเพียงพอมีฉะนั้นจะมีการเลือกค่า Threshold ที่ไม่มีประสิทธิภาพในทางกลับกัน ถ้าเลือกกำหนดพื้นที่ให้มีขนาดใหญ่เกินไปอาจละเมิดทฤษฎีของการ ส่องสว่างโดยประมาณ โดยวิธีนี้มีความซับซ้อนของการคำนวณน้อยกว่าวิธี Chow และ Kaneko และให้ผลลัพธ์ที่ดีสำหรับบางการใช้งาน ดังรูปที่ 2.13 รูปที่ 2.14 และรูปที่ 2.15 แสดงการปรับ Thresholding



รูปที่ 2.13 รูปภาพต้นฉบับ [10]



รูปที่ 2.14 รูปภาพที่ผ่านการ Global Thresholding [10]

Region-based segmentation

Let us first determine markers of the coins and the background. These markers are pixels that we can label unambiguously as either object or background. Here, the markers are found at the two extreme parts of the histogram of grey values:

```
>>> markers = np.zeros_like(coins)
```

รูปที่ 2.15 รูปภาพที่ผ่านการ Adaptive Thresholding [10]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2.5 Morphological (Opening)

Opening และ Closing เป็นสองตัวดำเนินการที่สำคัญจากสัญฐานวิทยาทางคณิตศาสตร์ (Mathematical Morphological) ทั้งสองการดำเนินการนั้นได้มาจากการดำเนินงานขั้นพื้นฐานของ Erosion และ Dilation เช่นเดียวกับการดำเนินการเหล่านั้นซึ่งโดยปกติแล้วจะนำไปใช้กับภาพไบนารี เอฟเฟกต์พื้นฐานของ Opening ค่อนข้างคล้ายกับ Erosion คือ มักจะลบพิกเซลโดดเด่น (สว่าง) บางส่วนออกจากขอบของพื้นที่ของพิกเซลโดดเด่น อย่างไรก็ตามการทำลาย พิกเซลนั้นจะน้อยกว่า Erosion โดยทั่วไป

เช่นเดียวกับตัวดำเนินการทางสัญฐานวิทยาอื่น ๆ การดำเนินการที่แน่นอนจะถูก กำหนดโดยองค์ประกอบโครงสร้าง ผลลัพธ์ที่คาดหวังของตัวดำเนินการนี้ คือการรักษาพื้นที่โดดเด่น ที่มีรูปร่างคล้ายกับองค์ประกอบของโครงสร้างนี้ หรือทำให้ภาพม็อดประกอบโครงสร้างอย่างสมบูรณ์ในขณะที่กำจัดพื้นที่อื่น ๆ ที่ไม่ใช่พื้นที่โดดเด่นทั้งหมด

กล่าวคือ Opening เป็นการทำ Erosion ก่อน แล้วนำผลลัพธ์ที่ได้ไปทำ Dilation โดยปกติใช้ในการแยก region ใหญ่ ๆ ให้เป็น region ย่อย ๆ อธิบายได้ดังสมการที่ (2.9) ซึ่งกำหนดให้ A เป็น Binary Image B เป็น Structure Element และ Dilation ของ A และ B เขียนแทนได้ด้วยสัญลักษณ์ $A \oplus B$

$$A \cdot B = (A \oplus B) \oplus B \quad (2.9)$$

2.3.2.6 Morphological (Dilation)

Dilation เป็นหนึ่งในสองตัวดำเนินการพื้นฐานในด้านสัญฐานวิทยาทางคณิตศาสตร์ โดยทั่วไปจะใช้กับภาพไบนารี แต่ก็ยังมีรุ่นที่ทำงานกับภาพ Grayscale ได้อยู่ ซึ่งผลกระทบพื้นฐานของการดำเนินการในภาพไบนารีคือการค่อยๆขยายขอบเขตของพื้นที่ของพิกเซลโดดเด่น (โดยทั่วไปคือพิกเซลสีขาว) ดังนั้นพื้นที่ของพิกเซลโดดเด่นจะมีขนาดโตขึ้นในขณะที่หลุมในพื้นที่เหล่านั้นมีขนาดเล็กลง

กล่าวคือ Dilation ใช้ในการเพิ่มขอบให้หนาขึ้น อธิบายได้ดังสมการที่ (2.10) ซึ่ง กำหนดให้ A เป็น Binary Image B เป็น Structure Element และ Dilation ของ A และ B เขียนแทนได้ด้วยสัญลักษณ์

$$A \oplus B = \{B_z\} \cap A \neq \emptyset \quad (2.10)$$

2.3.2.6 Minutiae Matching

ในโปรแกรมการระบุเอกลักษณ์ระยะแรกของงานวิจัยนี้เลือกที่จะปรับใช้หลักการ และวิธีคิดของหลักการ Minutiae Matching ซึ่งแต่เดิมเป็นอัลกอริทึมเพื่อจับคู่ลายนิ้วมือสำหรับเครื่องสแกนลายนิ้วมือ ดังนั้นจึงขออธิบายหลักการโดยเปรียบเสมือนว่าลายจะเข้คือลายนิ้วมือ

ซึ่งวิธีนี้ใช้หลักการที่กล่าวว่า ลายนิ้วมือคนเราแต่ละคนนั้น ประกอบขึ้นจากเส้น (ridge) ช่องว่าง (valley) Bifurcation เป็นจุดที่แยกเส้นออกเป็นหลายเส้น และ Endings คือตำแหน่งที่เส้นนั้นสิ้นสุดลง ในขั้นตอนการบันทึกนิ้วเพื่อการสมัครลงทะเบียน จุดมินูเซ่จะถูกสร้างขึ้นในรูปแบบของตำแหน่งที่เกี่ยวข้องซึ่งกันและกัน ทั้งตำแหน่งและทิศทาง ซึ่งจะได้รับการบันทึกเป็นค่าอ้างอิงและแปลงค่าเป็นข้อมูลดิจิทัลเก็บไว้ เรียกว่า Template ข้อมูลนี้จะนำไปเก็บไว้เป็นค่าของลายนิ้วมือนั้นแบบ เพื่อนำไปใช้ตรวจสอบบุคคลเมื่อมีการเปรียบเทียบจากการใช้งานที่การจับคู่แบบมินูเซ่จะเกิดขึ้นเมื่อมีการใช้งาน finger scan ของผู้ใช้ ภาพลายนิ้วมือใหม่ของผู้ใช้ที่ได้รับเข้ามาก็จะผ่านกระบวนการตรวจสอบแบบมินูเซ่ เช่นเดียวกับภาพลายนิ้วมือนั้นแบบจุดอ้างอิงที่ได้รับการแยกออกมาเป็นจุดมินูเซ่จะถูกแปลงเป็นข้อมูลดิจิทัลเช่นเดียวกันและจะได้รับการเปรียบเทียบระดับความเหมือนกับภาพต้นแบบที่เคยบันทึกไว้ในตอนลงทะเบียนโดยพยายามที่เปรียบเทียบจุดเหมือนให้มากที่สุดเท่าที่จะเป็นไปได้ในค่า Verification Threshold ที่กำหนดแน่นอน ซึ่งค่าดังกล่าวจะเป็นตัวกำหนดว่าผลการเปรียบเทียบที่ได้นั้นต้องใกล้เคียงกันขนาดไหนจึงจะเป็นการเข้าคู่เหมือนของมินูเซ่แสดงดังรูปที่ 2.16



รูปที่ 2.16 Ridge Ending และ Bifurcation [11]

ซึ่งจากหลักการ Minutiae จะประกอบด้วยสองส่วนสำคัญคือ Extractor (วิเคราะห์ Input เพื่อหาจุดสำคัญและ Matching (อาศัยข้อมูลจากจุดสำคัญเพื่อ Matching)

แต่ในส่วนตัวระบบที่ทางผู้จัดทำได้นำหลักการดังกล่าวมาออกแบบจึงได้ใช้หลักการวิเคราะห์ในการหาจุด key point ของลายนิ้วมือมาใช้สร้างเป็นแพทเทิร์นของฟิสิกส์ในการหาจุดสำคัญทั้งหมด 8 แบบสำหรับลายจะเข้แล้วทำการตรวจสอบทุกฟิสิกส์ของลายหน้าท้องหาตำแหน่งและจำนวนของจุดสำคัญที่มีแพทเทิร์นแบบนั้นๆ ทั้งหมดมาเก็บไว้รอการประมวลผล เพื่อเทียบเคียงอัตลักษณ์ต่อไป

2.3.2.7 Oriented FAST and Rotated BRIEF (ORB)

(ORB) ได้รับการพัฒนาที่ห้องปฏิบัติการ OpenCV โดย Ethan Rublee, Vincent Rabaud, Kurt Konolige และ Gary R. Bradski ในปี 2011 เป็นทางเลือกที่มีประสิทธิภาพและเป็นไปได้ของ SIFT และ SURF ORB เกิดขึ้นเนื่องจาก SIFT และ SURF เป็นอัลกอริทึมที่ได้รับการจดสิทธิบัตร อย่างไรก็ตาม ORB นั้นสามารถใช้งานได้ฟรี

ORB ทำงานเช่นเดียวกับ SIFT ในการทำงานของการตรวจจับคุณสมบัติ (และดีกว่า SURF) ในขณะที่เกือบสองคำสั่งของขนาดเร็วขึ้น ORB สร้างบนเครื่องตรวจจับ FAST Key point ที่รู้จักกันดีและตัวบ่งชี้ BRIEF เทคนิคทั้งสองนี้มีความน่าสนใจเนื่องจากมีประสิทธิภาพที่ดีและราคาต่ำ การสนับสนุนหลักของ ORB มีดังนี้

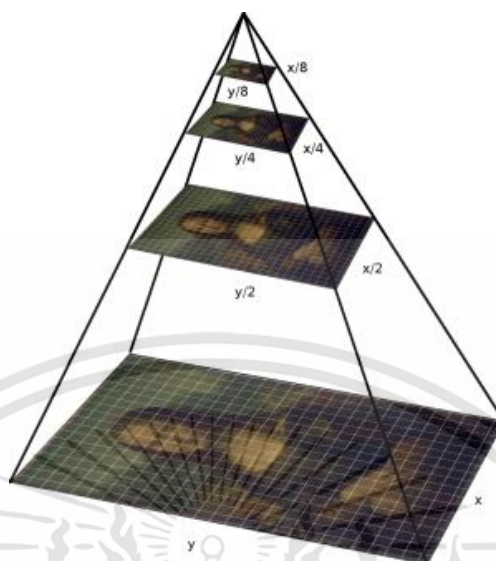
1. Fast (Features from Accelerated and Segments Test) กำหนดพิกเซล p ในอาร์เรย์อย่างรวดเร็วเปรียบเทียบความสว่างของ p กับรอบ 16 พิกเซลที่อยู่ในวงกลมเล็ก ๆ รอบ p แสดงดังรูปที่ 2.17 พิกเซลในวงกลมนั้นจะถูกจัดเรียงเป็นสามคลาสคือ เบากว่า p เข้มกว่า p และคล้ายกับ p หากมากกว่า 8 พิกเซลมืดหรือสว่างกว่า p กว่าจะถูกเลือกเป็นจุดสำคัญ ดังนั้นจุดสำคัญที่ค้นพบโดยเร็วจะให้ข้อมูลเกี่ยวกับตำแหน่งของการกำหนดขอบในภาพ สำหรับข้อมูลเพิ่มเติมอ้างอิงถึงอัลกอริทึมเบื้องต้นของ FAST (คุณลักษณะจากการทดสอบเซกเมนต์แบบเร่งรัด)



รูปที่ 2.17 พิกเซลในวงกลมรอบ p [12]

อย่างไรก็ตามคุณลักษณะของ FAST ไม่มีองค์ประกอบการวางแนวและคุณสมบัติหลายอย่าง อัลกอริทึมของ ORB ใช้พีระมิดภาพหลายชั้นแสดงดังรูปที่ 2.18 พีระมิดภาพคือการแสดงภาพหลายภาพในภาพเดียวซึ่งประกอบด้วยลำดับของภาพทั้งหมดซึ่งเป็นเวอร์ชันของภาพที่ความละเอียดต่างกัน แต่ละระดับในพีระมิดมีรูปภาพที่สุ่มตัวอย่างต่ำกว่าระดับก่อนหน้า เมื่อ ORB สร้างพีระมิดมันจะใช้อัลกอริทึมที่รวดเร็วในการตรวจจับจุดสำคัญในภาพ ด้วยการตรวจจับจุดสำคัญในแต่ละระดับ ORB จะค้นหาจุดสำคัญได้อย่างมีประสิทธิภาพในระดับที่แตกต่างกัน ด้วยวิธีนี้ ORB จึงเป็นค่าคงที่เฉพาะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.18 พีระมิดหลายชั้นที่สร้างโดย ORB [12]

หลังจากค้นหาจุดสำคัญ ORB จะกำหนดทิศทางให้แต่ละจุดสำคัญเช่นซ้ายหรือขวา ขึ้นอยู่กับระดับความเข้มเปลี่ยนไปรอบ ๆ จุดสำคัญนั้นอย่างไร สำหรับการตรวจจับการเปลี่ยนแปลง ความเข้มของ ORB ใช้ความเข้มของจุดเซ็นทรอยด์จุดเซ็นทรอยด์เข้มถือว่าความเข้มของมุมถูกขดเขย จากจุดศูนย์กลางและเวกเตอร์นี้อาจใช้เพื่อกำหนดทิศทางก่อนช่วงเวลาของแพทช์ถูกกำหนดเป็นดัง สมการที่ (2.11)

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \quad (2.11)$$

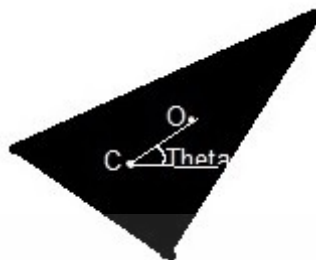
สามารถหาจุดเซ็นทรอยด์ของศูนย์กลางมวลของแพทช์สมการที่ (2.12)

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \quad (2.12)$$

และสามารถสร้างเวกเตอร์จากศูนย์กลางของมุม O ถึงจุดเซ็นทรอยด์ $-OC$ แสดงดังรูปที่ 2.19 และการวางแนวของแพทช์สามารถหาได้จากสมการที่ (2.11)

$$\theta = \text{atan} 2(m_{01}, m_{10}) \quad (2.13)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

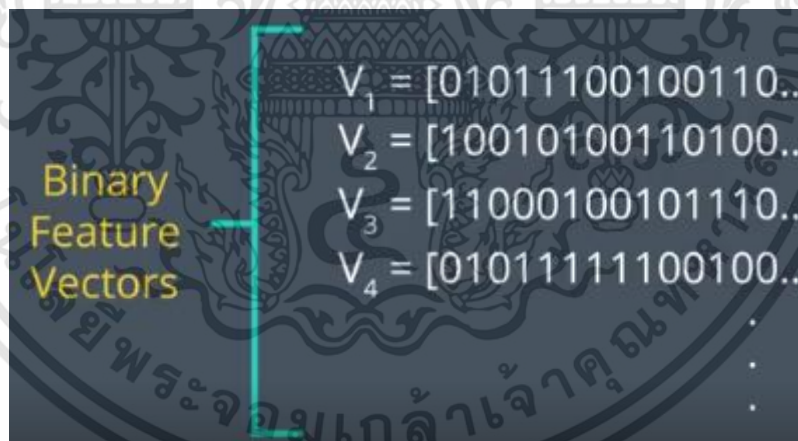


รูปที่ 2.19 เวกเตอร์จากศูนย์กลางของมุม O ถึงจุดเซ็นทรอยด์ $-OC$ [12]

เมื่อคำนวณการวางแนวของแพทช์แล้วจะสามารถหมุนแพทช์ ซึ่งเป็นการหมุนแบบ บัญญัติและจากนั้นคำนวณตัวอธิบายซึ่งจะได้ค่าความแปรปรวนหมุน

2. Brief (Binary robust independent elementary feature)

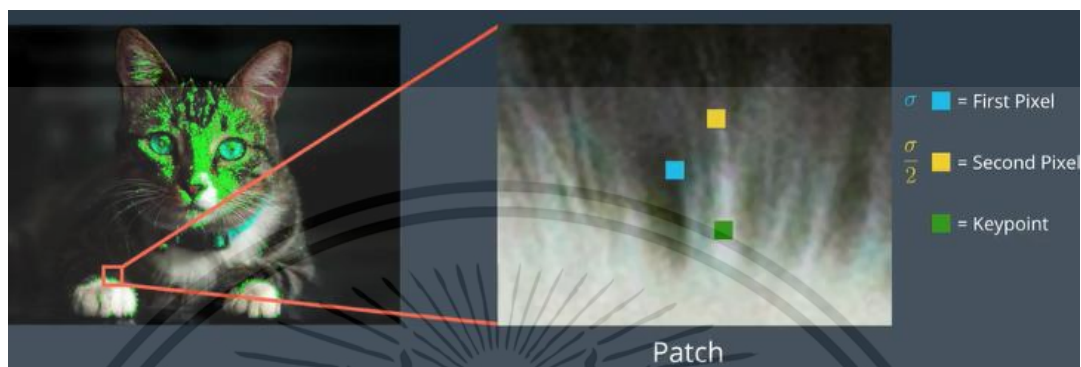
โดยสรุปแล้วจุดสำคัญทั้งหมดที่พบโดยอัลกอริทึมแบบเร็วและแปลงเป็นเวกเตอร์ คุณลักษณะไบนารีเพื่อให้จุดสำคัญสามารถเป็นตัวแทนวัตถุได้ เวกเตอร์คุณลักษณะไบนารียังรู้จักใน ฐานะตัวอธิบายคุณลักษณะไบนารีเป็นเวกเตอร์คุณลักษณะที่มีเพียง 1 และ 0 โดยย่อ แต่ละจุดสำคัญ ถูกอธิบายโดยเวกเตอร์คุณลักษณะซึ่งเป็นคอลัมน์ 128–512 บิต แสดงดังรูปที่ 2.20



รูปที่ 2.20 เวกเตอร์คุณลักษณะซึ่งเป็นคอลัมน์ [12]

กล่าวโดยสรุปการปรับภาพให้เรียบโดยใช้เคอร์เนล Gaussian เพื่อป้องกันไม่ให้ตัวบ่งชี้ไว ต่อสิ่งรบกวนความถี่สูง ดังนั้นการเลือกคู่ของพิกเซลแบบสุ่มในบริเวณที่กำหนดไว้รอบ ๆ จุดสำคัญนั้น บริเวณที่กำหนดไว้รอบ ๆ พิกเซลเรียกว่าแพทช์ซึ่งเป็นสี่เหลี่ยมจัตุรัสของความกว้าง และความสูงของ พิกเซลแสดงดังรูปที่ 2.21 โดยพิกเซลแรกในคู่สุ่มนั้นมาจากการแจกแจงแบบเกาส์ ซึ่งมีศูนย์กลางอยู่ที่ จุดสำคัญพร้อมกับการเบี่ยงเบนแบบคว้นหรือการแพร่กระจายของซิกมา พิกเซลที่สองในการจับคู่แบบ สุ่มนั้นมาจากการแจกแจงแบบเกาส์ซึ่งมีศูนย์กลางอยู่ที่พิกเซลแรกโดยมีค่า เบี่ยงเบนมาตรฐานหรือ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การแพร่กระจายของซิกมาสองพิกเซล ตอนนี้อ้าพิกเซลแรกสว่างกว่าพิกเซล ที่สองมันจะกำหนดค่า 1 ให้กับบิตอื่นที่เกี่ยวข้อง นอกจากนั้นกำหนดให้เป็น 0



รูปที่ 2.21 แพทช์คือสี่เหลี่ยมจัตุรัสของความกว้างและความสูงของพิกเซล [12]

การเลือกคู่สุ่มและกำหนดค่าให้กับจุดสำคัญสำหรับเวกเตอร์ 128 บิต ให้ทำซ้ำขั้นตอนนี้ 128 ครั้งสำหรับจุดสำคัญ จากนั้นสร้างบทสรุปแบบเวกเตอร์เช่นนี้สำหรับแต่ละประเด็นในภาพ อย่างไรก็ตาม BRIEF ยังไม่แปรผันกับการหมุนดังนั้น ORB จึงใช้ rBRIEF หรือ BRIEF ที่ตระหนักถึงการหมุน ORB พยายามเพิ่มฟังก์ชันนี้โดยไม่สูญเสียความกว้างและความเร็วของ BRIEF พิจารณาโปรแกรมแก้ไขภาพให้เรียบ (p) การทดสอบไบนารี $\tau(p; x, y)$ ถูกกำหนด ดังสมการที่ (2.14)

$$\tau(p; x, y) = \begin{cases} 1 & : p(x) < p(y) \\ 0 & : p(x) \geq p(y) \end{cases} \quad (2.14)$$

โดยที่ $p(x)$ คือค่าความเข้มของ p ที่จุด x คุณลักษณะถูกนิยามเป็นเวกเตอร์ของการทดสอบไบนารี n ดังสมการที่ (2.15)

$$f(n) = \sum_{1 < i < n} 2^{i-1} \tau(p; x_i y_i) \quad (2.15)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประสิทธิภาพการจับคู่ของ BRIEF ลดลงอย่างรวดเร็วสำหรับการหมุนในระนาบที่มากกว่า 2 ถึง 3 องศา ORB เสนอวิธีการตัดท้ายสั้น ๆ ตามทิศทางของจุดสำคัญ สำหรับชุดคุณลักษณะใด ๆ ของการทดสอบไบนารี n ที่ตำแหน่ง (i, j) จึงต้องการเมทริกซ์ $2 \times n$ ดังสมการที่ (2.16)

$$S = \begin{pmatrix} x_1, \dots, x_n \\ y_1, \dots, y_n \end{pmatrix} \quad (2.16)$$

โดยใช้การวางแนวแพทช์ θ และเมทริกซ์การหมุนที่สอดคล้องกัน R_θ และสร้างเวกเตอร์ S ของ S_θ ที่นำหน้า ดังสมการที่ (2.17)

$$S_\theta = R_\theta S \quad (2.17)$$

ขณะนี้ตัวดำเนินการ BRIEF ที่นำหน้าจะไปดังสมการที่ (2.18)

$$g_n(p, \theta) = f_n(p) | (x_i, y_i) \in S_\theta \quad (2.18)$$

จนกว่าจะลดมุมลงเหลือ 12 องศา และสร้างตารางการค้นหาของรูปแบบ BRIEF ที่คำนวณล่วงหน้าแล้ว トラバิดที่การวางแนวของจุดสำคัญ θ สอดคล้องกับมุมมองชุดจุดที่ถูกต้อง S_θ จะถูกใช้เพื่อคำนวณตัวบ่งชี้

ORB ระบุอัลกอริทึม rBRIEF ดังนี้

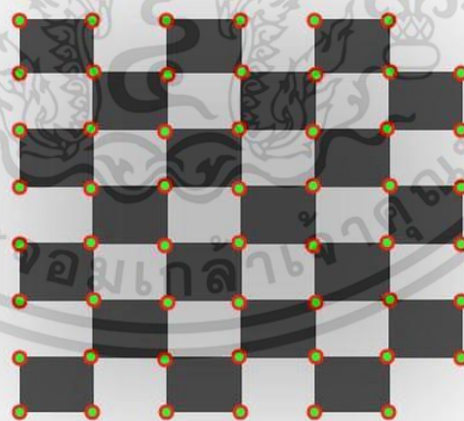
- 1) ทำการทดสอบแต่ละครั้งและเทรนแพทช์ทั้งหมด
- 2) ส่งการทดสอบตามระยะห่างจากค่าเฉลี่ย 0.5 สร้างเวกเตอร์ T
- 3) การสืบค้นอื่น ๆ

- ใส่การทดสอบครั้งแรกในเวกเตอร์ผลลัพธ์ R และลบออกจาก T
- ทำแบบทดสอบครั้งต่อไปจาก T และเปรียบเทียบกับการทดสอบทั้งหมดใน R ถ้าความสัมพันธ์แบบสัมบูรณ์มีค่ามากกว่าค่าเริ่มต้นให้กำจัดทิ้งไป ถ้าไม่ให้ เพิ่มไปยัง R
- ทำซ้ำขั้นตอนก่อนหน้านี้จนกว่าจะมี 256 การทดสอบใน R หากมีน้อยกว่า 256 การทดสอบให้เพิ่มเกณฑ์และทำอีกครั้ง

rBRIEF แสดงให้เห็นถึงการปรับปรุงที่สำคัญในความแปรปรวนและความสัมพันธ์ที่ดีกว่า BRIEF

2.3.2.8 Harris Corner Detector

เป็นตัวดำเนินการตรวจจับมุมที่นิยมใช้ในอัลกอริธึมการมองเห็นของคอมพิวเตอร์เพื่อแยกมุมและสรุปคุณสมบัติของภาพ Chris Harris และ Mike Stephens เปิดตัวครั้งแรกในปี 1988 จากการปรับปรุงเครื่องตรวจจับมุมของ Moravec เมื่อเทียบกับรุ่นก่อนหน้าเครื่องตรวจจับมุมของ แฮร์ริส (Harris) จะคำนึงถึงความแตกต่างของคะแนนมุมโดยอ้างอิงกับทิศทางโดยตรงแทนที่จะใช้แพทช์การเปลี่ยนเกียร์สำหรับทุกมุม 45 องศาและได้รับการพิสูจน์แล้วว่ามีความแม่นยำในการแยกแยะระหว่างขอบและมุม ตั้งแต่นั้นมาได้มีการปรับปรุงและนำมาใช้ในหลายอัลกอริทึมเพื่อประมวลผลภาพล่วงหน้าสำหรับแอปพลิเคชันที่ตามมา แสดงดังรูปที่ 2.22 [13]

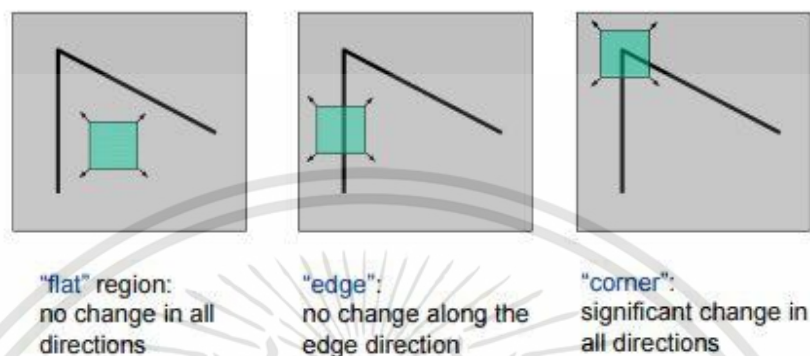


รูปที่ 2.22 การตรวจจับมุมของ Moravec [13]

มุมคือจุดที่มีพื้นที่ใกล้เคียงอยู่ในสองทิศทางที่โดดเด่นและแตกต่างกัน กล่าวอีกนัยหนึ่งมุมสามารถตีความได้ว่าเป็นจุดเชื่อมของสองขอบโดยที่ขอบคือการเปลี่ยนแปลงความสว่างของภาพอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

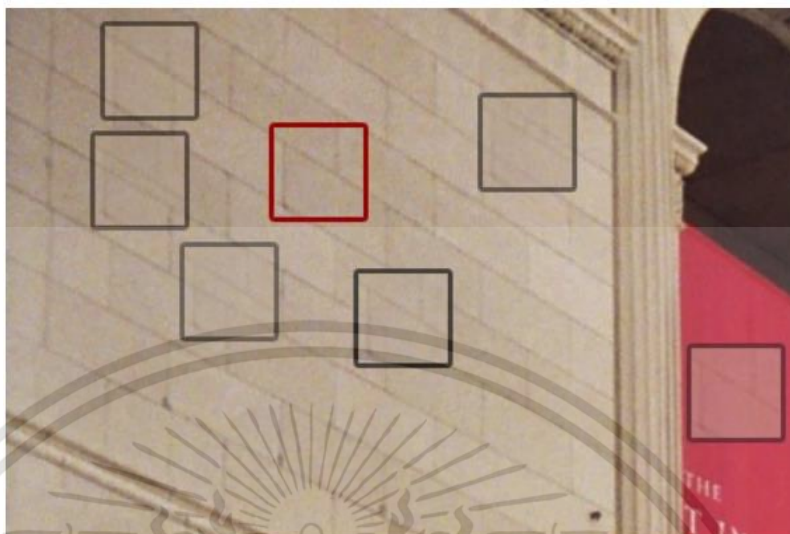
กะทันหัน มุมเป็นคุณสมบัติที่สำคัญในภาพและโดยทั่วไปเรียกว่าจุดสนใจซึ่งไม่แปรผันกับการหมุนและการส่องสว่าง แสดงดังรูปที่ 2.23



รูปที่ 2.23 จุดสนใจซึ่งไม่แปรผันกับการหมุนและการส่องสว่าง [13]

เนื่องจากมุมจึงถือเป็นคุณสมบัติที่ดีกว่าหรือเหมาะสำหรับการทำแผนที่แพตช์ ถ้าหาพื้นที่ราบแล้วจะไม่พบการเปลี่ยนแปลงการไล่ระดับสีในทิศทางใด ๆ ในทำนองเดียวกันในบริเวณขอบจะไม่มีการเปลี่ยนแปลงการไล่ระดับสีตามทิศทางขอบ ดังนั้นทั้งพื้นที่ราบและพื้นที่ขอบจึงไม่ดีสำหรับการจับคู่แพตช์เนื่องจากไม่ได้มีความโดดเด่นมากนัก ซึ่งมีแพตช์ที่คล้ายกันมากมายตามขอบในขอบเขตขอบ ในขณะที่อยู่ในพื้นที่มุมเราสังเกตเห็นการเปลี่ยนแปลงของการไล่ระดับสีอย่างมีนัยสำคัญในทุกทิศทาง เนื่องจากมุมนี้ถือว่าดีสำหรับการจับคู่แพตช์ ซึ่งการเลื่อนหน้าต่างไปในทิศทางใด ๆ จะทำให้รูปลักษณ์เปลี่ยนไปมาก และโดยทั่วไปจะมีความเสถียรมากกว่าเมื่อเปลี่ยนมุมมอง

การตรวจจับมุม คือการพิจารณาหน้าต่างเล็ก ๆ รอบ ๆ พิกเซลแต่ละพิกเซลในรูปภาพ ต้องการระบุหน้าต่างต่างพิกเซลทั้งหมดที่ไม่ซ้ำกัน ความเป็นเอกลักษณ์สามารถวัดได้โดยการเลื่อนแต่ละหน้าต่างเป็นจำนวนเล็กน้อยในทิศทางที่กำหนดและวัดปริมาณการเปลี่ยนแปลงที่เกิดขึ้นในค่าพิกเซล แสดงดังรูปที่ 2.24



รูปภาพที่ 2.24 หน้าต่างรอบๆ พิกเซลแต่ละพิกเซลในรูปภาพ [13]

การใช้ผลรวมกำลังสองผลต่าง (SSD) ของค่าพิกเซลก่อนและหลังการเปลี่ยนแปลงและระบุนหน้าต่างพิกเซลที่ SSD มีขนาดใหญ่สำหรับการเลื่อนทั้ง 8 ทิศทาง ให้กำหนดฟังก์ชันการเปลี่ยนแปลง $E(u, v)$ เป็นผลรวมของผลต่างผลรวมกำลังสองทั้งหมดโดยที่ u, v คือพิกัด x, y ของทุกพิกเซลในหน้าต่าง 3×3 และ I คือค่าความเข้มของพิกเซล คุณสมบัติในภาพคือพิกเซลทั้งหมดที่มีค่า $E(u, v)$ มากตามที่กำหนดโดยเกณฑ์บางประการ ดังสมการที่ (2.19)

$$E(u, v) = \sum_{x,y} w(x, y) [I(x + u, y + v) - I(x, y)]^2 \quad (2.19)$$

เพิ่มฟังก์ชัน $E(u, v)$ ให้มากที่สุดสำหรับการตรวจจับมุม หมายความว่าต้องทำการขยายเทอมสองให้ได้มากที่สุด โดยใช้ Taylor Expansion กับสมการข้างต้นและใช้ขั้นตอนทางคณิตศาสตร์บางอย่างจะได้ ดังสมการที่ (2.20)

$$E(u, v) \approx [u \ v] \left(\sum \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix} \quad (2.20)$$

ตอนนี้ทำการเปลี่ยนชื่อเมทริกซ์และกำหนดให้เป็น M จะได้ดังสมการที่ (2.21)

$$M = \sum w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (2.21)$$

ดังนั้นจะได้ดังสมการที่ (2.22)

$$E(u, v) \approx [u \ v] M \begin{bmatrix} u \\ v \end{bmatrix} \quad (2.22)$$

2.3.2.9 คุณสมบัติ KAZE

แทนที่จะใช้พื้นที่มาตราส่วนเกาส์เซียน มันเหมาะสมแล้วที่จะทำให้การเบลอบแบบปรับได้เฉพาะที่ ซึ่งจะทำให้สัญญาณรบกวนเบลอ ในขณะที่รายละเอียดและขอบจะยังคงอยู่ ด้วยเหตุนี้จึงได้มีการเสนอสเกลสเกลที่ไม่เป็นเชิงเส้น วิธีการแพร่แบบไม่เชิงเส้นทำได้ดีกว่าวิธีเชิงเส้นมาก ความสามารถในการทำซ้ำและความโดดเด่นเพิ่มขึ้นอย่างมากจากการใช้การแพร่กระจายแบบไม่เชิงเส้น

อย่างไรก็ตาม ก่อนหน้าคุณลักษณะของ KAZE วิธีการที่ไม่เป็นเชิงเส้นส่วนใหญ่นั้นไม่มีประสิทธิภาพ ทำให้ไม่มีประโยชน์สำหรับการตรวจจับคุณสมบัติและการสร้าง 3D ขึ้นใหม่ วิธีการเหล่านี้โดยปกติประกอบด้วยการแยกฟังก์ชันโดยใช้รูปแบบออยเลอร์ไปข้างหน้า รูปแบบออยเลอร์ต้องใช้ขนาดขั้นที่เล็กมากสำหรับการบรรจบกัน และด้วยเหตุนี้การวนซ้ำหลายครั้งเพื่อให้ได้ระดับมาตราส่วนที่ต้องการและต้นทุนในการคำนวณที่สูง Weickert และคณะ แนะนำรูปแบบที่มีประสิทธิภาพสำหรับการกรองการแพร่กระจายแบบไม่เชิงเส้นใน

ดังนั้นเพื่อสร้างสเกลสเกลที่ไม่เป็นเชิงเส้น Alcantarilla et al เสนอการใช้การแพร่กระจายสื่อไฟฟ้าแบบแปรผันและเทคนิคการแยกตัวดำเนินการเสริมโดยใช้อัลกอริทึมที่เรียกว่า Thomas Algorithm ซึ่งเป็นตัวแปรพิเศษของอัลกอริทึมการกำจัดแบบเกาส์เซียน ดังสมการที่ (2.23)

$$\frac{\partial L}{\partial t} = \text{div}(c(x, y, t) \cdot \nabla L) \quad (2.23)$$

คุณสมบัติของ KAZE มีความสามารถในการทำซ้ำได้สูงกว่า SIFT และ SURF นอกจากนี้ KAZE ยังทำงานได้ดีกว่าบนพื้นผิวที่มีความแม่นยำ การเรียกคืน และการเปลี่ยนรูป ซึ่งเป็นข้อพิสูจน์ว่า พื้นที่มาตราส่วนไม่เชิงเส้นนั้นดีกว่าพื้นที่มาตราส่วนเชิงเส้นมากเพียงใด

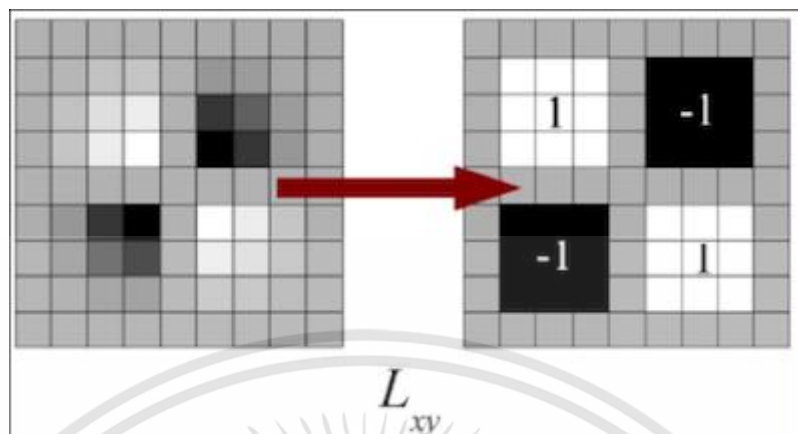
2.3.2.10 วิธีการ SURF (Speeded Up Robust Features)

เป็นอัลกอริธึมที่รวดเร็วและมีประสิทธิภาพสำหรับการแทนค่าและการเปรียบเทียบรูปภาพแบบคงที่ ความสนใจหลักของแนวทาง SURF อยู่ที่การคำนวณตัวดำเนินการอย่างรวดเร็วโดยใช้ตัวกรองกล่องจึงทำให้แอปพลิเคชันแบบเรียลไทม์เช่นการติดตามและการจดจำวัตถุ

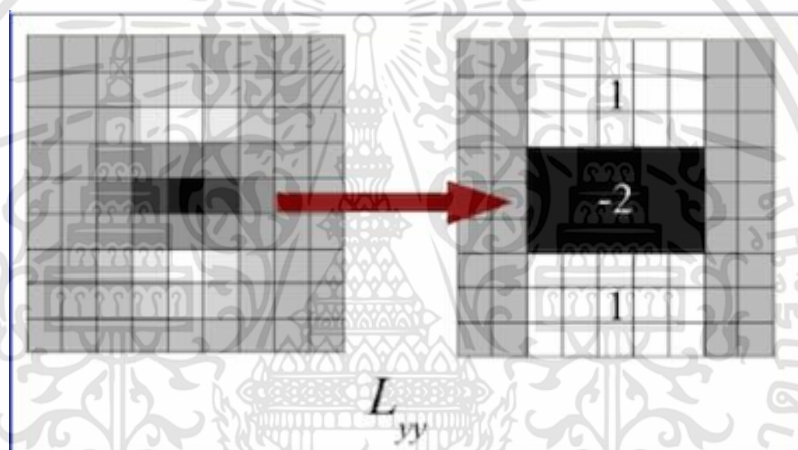
SURF ใช้เมทริกซ์ Hessian เนื่องจากประสิทธิภาพที่ดีในการคำนวณเวลาและความแม่นยำ แทนที่จะใช้การวัดที่แตกต่างกันในการเลือกตำแหน่งและมาตราส่วน (เครื่องตรวจจับ Hessian-Laplace) การโต้คลื่นอาศัยดีเทอร์มิแนนต์ของเมทริกซ์ Hessian สำหรับทั้งสองอย่าง เมื่อพิจารณาถึงพิกเซล Hessian ของพิกเซลนี้ ดังสมการที่ (2.24)

$$H(f(x,y)) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2} & \frac{\partial^2 f}{\partial x \partial y} \\ \frac{\partial^2 f}{\partial x \partial y} & \frac{\partial^2 f}{\partial y^2} \end{bmatrix} \quad (2.24)$$

ในการคำนวณดีเทอร์มิแนนต์ของเมทริกซ์ Hessian อันดับแรกเราต้องใช้คอนโวลูชันกับเคอร์เนล Gaussian จากนั้นจึงอนุพันธ์อันดับสอง หลังจาก Lowe ประสบความสำเร็จกับการประมาณ LoG (SIFT) SURF ก็ผลักดันการประมาณทั้ง Convolution และอนุพันธ์ลำดับที่สองให้ดียิ่งขึ้นด้วยตัวกรองกล่อง อนุพันธ์ Gaussian ลำดับที่สองโดยประมาณเหล่านี้และสามารถประเมินได้ด้วยต้นทุนการคำนวณที่ต่ำมากโดยใช้ภาพอินทริกัลและไม่ขึ้นกับขนาดและนี่เป็นส่วนหนึ่งของสาเหตุที่ SURF เร็ว แสดงดังรูปที่ 2.25 และรูปที่ 2.26



รูปที่ 2.25 อนุพันธ์ย่อย Gaussian ใน xy [14]



รูปที่ 2.26 อนุพันธ์ย่อย Gaussian ใน y [14]

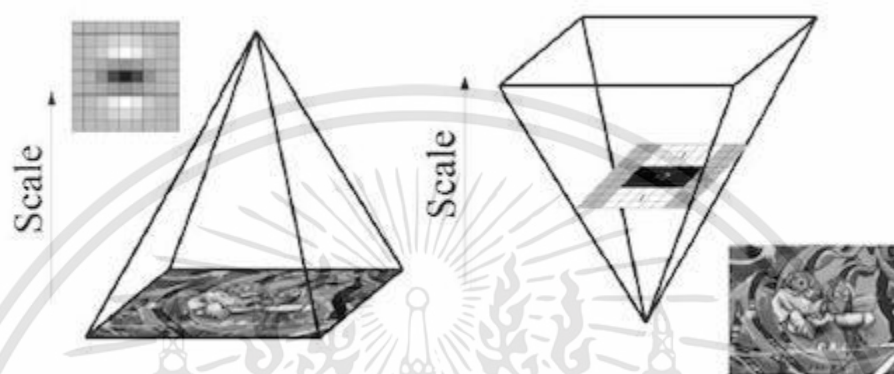
ฟิลเตอร์กล่อง 9×9 เป็นค่าประมาณสำหรับอนุพันธ์ลำดับที่สองของเกาส์เซียนที่มี $\sigma = 1.2$ แสดงการประมาณเหล่านี้ด้วย D_{xx} , D_{yy} และ D_{xy} สามารถแทนดีเทอร์มิแนนต์ของเฮสเซียน (โดยประมาณ) ดังสมการที่ (2.25)

$$\det(\mathcal{H}_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2 \quad (2.25)$$

โดยปกติช่องว่างขนาดจะใช้เป็นปิรามิดรูปภาพ ภาพจะถูกปรับให้เรียบซ้ำ ๆ ด้วย Gaussian จากนั้นจึงทำการสุ่มตัวอย่างย่อยเพื่อให้ได้พีระมิดในระดับที่สูงขึ้น เนื่องจากการใช้ฟิลเตอร์กล่องและภาพอินทิกรัลจึงไม่จำเป็นต้องใช้ฟิลเตอร์เดิมซ้ำกับผลลัพธ์ของเลเยอร์ที่กรองก่อนหน้านี้ แต่สามารถใช้ฟิลเตอร์ขนาดใดก็ได้ที่ความเร็วเดียวกันกับรูปภาพต้นฉบับโดยตรงและ แม้จะขนาดกัน ดังนั้นจึงวิเคราะห์พื้นที่สเกลโดยการอัปเดตขนาดฟิลเตอร์ เช่น $9 \times 9 \rightarrow 15 \times 15 \rightarrow 21 \times 21 \rightarrow 27 \times 27$ เป็นต้น แทนที่จะลดขนาดภาพซ้ำ ๆ ดังนั้นสำหรับแต่ละอ็อกเทฟใหม่การเพิ่มขนาดตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กรองจะเพิ่มขึ้นเป็นสองเท่าพร้อมกันช่วงเวลาการสุ่มตัวอย่างสำหรับการแยกจุดสนใจ (σ) สามารถเพิ่มเป็นสองเท่าได้เช่นกันซึ่งจะช่วยให้สามารถปรับขนาดตัวกรองได้โดยมีต้นทุนคงที่ ในการกำหนดจุดสนใจในรูปภาพและบนสเกลจะใช้การปราบปรามที่ไม่สูงสุดในย่าน $3 \times 3 \times 3$ แสดงดังรูปที่ 2.27



รูปที่ 2.27 พีรามิดรูปภาพที่ถูกปรับให้เรียบซ้ำ ๆ ด้วย Gaussian [14]

การสร้างตัวอธิบาย SURF เกิดขึ้นในสองขั้นตอน ขั้นตอนแรกประกอบด้วย การแก้ไขการวางแนวที่ทำซ้ำได้โดยอาศัยข้อมูลจากพื้นที่วงกลมรอบจุดสำคัญ จากนั้นเราสร้างพื้นที่สี่เหลี่ยมจัตุรัสที่สอดคล้องกับการวางแนวที่เลือกและดึงตัวบอก SURF ออกมา

2.3.2.11 MinEigen มีข้อมูลเกี่ยวกับจุดคุณลักษณะที่ตรวจพบในภาพ อินพุตระดับสีเทา 2 มิติ ฟังก์ชัน detectMinEigenFeatures ใช้อัลกอริธึมค่าลักษณะเฉพาะขั้นต่ำที่พัฒนาโดย Shi และ Tomasi เพื่อค้นหาจุดคุณสมบัตินั้น

2.3.2.12 MSER (ขอบเขตสูงสุดที่เสถียรสูงสุด) บนผลลัพธ์ของตัวดำเนินการ Sobel เพื่อค้นหาพื้นที่ที่เสถียรของภาพ โดยกำหนดค่า

2.3.2.13 BRISK (จุดสำคัญที่ปรับขนาดได้คงที่แบบไบนารี)

เป็นวิธีการใหม่สำหรับการตรวจจับจุดสำคัญรายละเอียดและการจับคู่ การประเมินที่ครอบคลุมเกี่ยวกับชุดข้อมูลมาตรฐานแสดงให้เห็นถึง ประสิทธิภาพที่ปรับได้และคุณภาพสูงของ BRISK เช่นเดียวกับอัลกอริธึมที่ล้ำสมัยแม้ว่าจะมีค่าใช้จ่ายในการคำนวณที่ต่ำกว่ามาก อนุญาตสู่ความเร็วในการประยุกต์ใช้เครื่องตรวจจับแบบ FAST บนสเกลพื้นที่ที่รวมกับการประกอบของตัวอธิบายบิตสตรีมจากการเปรียบเทียบความเข้มที่ดึงมาโดยการสุ่มตัวอย่างเฉพาะของแต่ละจุดสำคัญของลวดลายนั้น

2.4 Edge Detection Theory

ขอบ (Edge) เกิดจากการเปลี่ยนแปลงของค่าสีของพิกเซลระหว่างสองบริเวณในรูปภาพ ซึ่งเป็นลักษณะพื้นฐานที่สำคัญของรูปภาพ การตรวจจับขอบ (Edge detection) จึงเป็นพื้นฐานของการประมวลผลภาพในหลายๆ ด้าน เช่น การแบ่งส่วนภาพ (image segmentation) การกำหนดตำแหน่งภาพ (image registration) และการรู้จำแบบรูป (pattern recognition) วิธีการตรวจจับขอบโดยทั่วไปจะใช้การดำเนินการพื้นฐานที่ประกอบด้วยการดำเนินการผลต่างอันดับหนึ่ง เช่น การดำเนินการโรเบิร์ต (Robert operator) การดำเนินการพรีวิทท์ (Prewitt operator) และการดำเนินการโซเบล (Sobel operator) และการดำเนินการผลต่างอันดับสอง เช่น การดำเนินการลาปลาซ (Laplace operator) วิธีการเหล่านี้สามารถใช้งานได้ง่าย แต่คุณภาพของภาพขอบที่ได้จะขึ้นอยู่กับค่าเกณฑ์ที่กำหนด (threshold) หากค่าเกณฑ์มีค่าสูง จะทำให้ขอบบางส่วนไม่ถูกตรวจจับ ในทางกลับกันหากค่าเกณฑ์มีค่าต่ำ จะทำให้เกิดความผิดพลาดในการตรวจจับขอบโดยเราสามารถหาความชันได้โดยการใช้ derivative ดังสมการที่ (2.26)

$$f'(x) = \frac{df}{dx}(x) \quad (2.26)$$

แต่เราต้องหาความชันในแบบ discrete เนื่องจากค่า intensity ไม่เป็นแบบ continuous ดังสมการที่ (2.27)

$$\frac{df}{du}(u) \approx \frac{f(u+1) - f(u-1)}{(u+1) - (u-1)} = \frac{f(u+1) - f(u-1)}{2} \quad (2.27)$$

2.4.1 CANNY

Canny (1986) ได้นำเสนอวิธีการตรวจจับขอบที่เรียกว่า วิธีแคนนี่ (Canny method) ซึ่งประกอบด้วยการดำเนินการหลัก 4 ขั้นตอน คือ การปรับค่าสีโดยใช้ตัวกรองเกาส์เซียนเพื่อกำจัดสัญญาณรบกวน การหาขนาดและทิศทางของการเปลี่ยนแปลงค่าสีของพิกเซล การพิจารณาขอบที่เป็นไปได้โดยการเปรียบเทียบขนาดของการเปลี่ยนแปลงค่าสีของพิกเซลนั้นกับพิกเซลข้างเคียงที่อยู่ในทิศทางของการเปลี่ยนแปลงค่าสี และการใช้ค่าเกณฑ์สองระดับ (ค่าเกณฑ์สูง และค่าเกณฑ์ต่ำ) เพื่อกำหนดพิกเซลที่เป็นขอบเข้มและขอบไม่เข้ม วิธีแคนนี่สามารถตรวจจับขอบได้ดี อย่างไรก็ตามวิธีนี้มีความซับซ้อนในการดำเนินการ และมีความไม่สะดวกในการใช้งาน เนื่องจากมีความจำเป็นต้องกำหนดค่าเกณฑ์ทั้งสองจากผู้ใช้งานเพื่อให้ได้ภาพขอบที่มีประสิทธิภาพมากยิ่งขึ้น จึงมีการพัฒนาวิธี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แคณินในขั้นตอนต่างๆ เช่น ปรับเปลี่ยนขั้นตอนการพิจารณาขอที่เป็นไปได้ (Ding, Goshtasby, 2001) ปรับเปลี่ยนวิธีการหาขนาดและทิศทางของการเปลี่ยนแปลงค่าสีของพิกเซล (Xun Wang, Jian-Qiu Jin, 2007) และนำเสนอการหาค่าเกณฑ์ แบบอัตโนมัติ ด้วยวิธีการของ Otsu (Li et al., 2009) งานวิจัยนี้นำเสนอการปรับเปลี่ยนตัวกรองที่ใช้ในการปรับค่าสีวิธีการหาขนาดและทิศทางของ การเปลี่ยนแปลงค่าสีที่ง่ายขึ้น และวิธีการหาค่าเกณฑ์ทั้งสองแบบอัตโนมัติเพื่อความสะดวกในการใช้งาน

2.5 Flutter

Flutter คือ Cross-Platform Framework ที่ใช้ในการพัฒนา Native Mobile Application (Android/iOS) พัฒนาโดยบริษัท Google Inc. โดยใช้ภาษา Dart ในการพัฒนาที่มีความคล้ายกับภาษา C# และ Java อีกหนึ่งจุดเด่นของ Flutter คือ การปรับแต่ง UI (User Interface) ที่มีความยืดหยุ่น แยกการออกแบบเพื่อเน้นไปที่ประสบการณ์ของผู้ใช้งาน UX (User Experience) โดย UI จะใกล้เคียงกับ Native และตรงตาม Design Guideline ที่ถูกต้องและมีความสามารถในการทำ Hot Reload ที่ทำให้การแก้ไขโค้ดสามารถแสดงผลได้ทันทีในระหว่างที่รันแอปพลิเคชัน และยังรวมไปถึงมี Widget ที่พร้อมให้เลือกใช้มากมาย ทำให้พัฒนาแอปพลิเคชันได้ไวเหมาะสำหรับองค์กรที่ต้องการแอปที่สวยงามและมีประสิทธิภาพ ในหลักสูตรมีการสอน State Management โดยใช้ BLoC (Business Logic Component) ที่นิยมในกลุ่มนักพัฒนา Flutter ในการจัดการ Local/Global State เพื่อรองรับระบบที่มีขนาดใหญ่และซับซ้อนดูเป็นมืออาชีพ รวมถึงการเขียนโค้ดที่ทำงานร่วมกับ Native API โดยใช้ภาษาสมัยใหม่อย่าง Kotlin และ Swift เพื่อให้ผู้เข้าอบรมสามารถรับมือกับ Requirement ที่ต้องเชื่อมต่อกับ Native Android และ iOS แสดงดังรูปที่ 2.28



รูปที่ 2.28 Android Studio [15]

2.6 Dart

Dart นั้นเป็นภาษาโปรแกรมที่เอาไว้สำหรับสร้างแอปพลิเคชันบนแพลตฟอร์มที่หลากหลายโดยได้ทั้ง mobile, desktop, server และทั้ง web สิ่งที่เป็นที่นิยมที่สุดที่ทำให้คนสนใจมา

เรียนภาษา Dart กันก็คือเพื่อที่จะเอาไปใช้ร่วมกับ Flutter ที่เป็นเครื่องมือช่วยสร้าง UI ของ Google ซึ่งใช้ได้ทั้งกับ Android และ iOS หรือจะเป็นใน Desktop กับ Web ก็ยังได้

ภาษา Dart นี้ถูกสร้างโดย Google และปล่อยให้ใช้งานแบบ open source ทำให้ทุกคนสามารถนำไปใช้งานได้ฟรีๆ และการที่ Dart ถูกออกแบบมาให้ใช้งานได้ง่ายและมีประสิทธิภาพแบบภาษาเชิงวัตถุอื่นๆอย่าง Java C# C++ จึงเป็นตัวเลือกภาษาที่น่าสนใจในการศึกษาเป็นภาษาแรกอีกด้วย แสดงดังรูปที่ 2.29



รูปที่ 2.29 ภาษา Dart [16]

2.7 MongoOB

MongoDB เป็น open-source document database โดยเป็นฐานข้อมูลแบบ NoSQL คือไม่มี relation (ความสัมพันธ์) ของตารางแบบ SQL ทั่วไป แต่จะเก็บข้อมูลเป็นแบบ JSON (JavaScript Object Notation) แทนการบันทึกข้อมูลทุกๆ record ใน MongoDB ซึ่งเราจะเรียกมันว่า Document และการเก็บข้อมูล document ใน MongoDB จะถูกเก็บไว้ใน Collections ซึ่งเปรียบเทียบได้กับ Table ใน Relational Database ทั่ว ๆ ไป แต่แตกต่างกันที่ Collection ไม่จำเป็นต้องกำหนดโครงสร้างใดๆให้มันเหมือน SQL ปกติทั่วไป แสดงดังรูปที่ 2.30



รูปที่ 2.30 MongoDB [17]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 Node.js

Node.js คือ Cross Platform Runtime Environment สำหรับฝั่ง Server และเป็น Open Source ซึ่งเขียนด้วยภาษา JavaScript หรือก็คือ Platform หรือ JavaScript Framework ตัวหนึ่ง ที่เขียนด้วย JavaScript ที่ทำหน้าที่อยู่ฝั่ง Backend และมีโมดูลที่ช่วยสำหรับการทำตัวเป็น Web Server แสดงดังภาพที่ 2.31



รูปที่ 2.31 Node.js [18]

2.7.1 Express.js

Express.js หรือ Express นั้นเป็นเว็บเฟรมเวิร์คจาก NPM ที่ใช้สำหรับพัฒนาเว็บแอปพลิเคชันหรือเว็บไซต์บน Node.js ที่ทำงานที่ฝั่งของ Backend ตัวของเฟรมเวิร์คนี้ถูกพัฒนาจากโมดูล http ซึ่งเป็นโมดูลของ Node.js เอง นั้นหมายความว่าใช้เพียงแค่ Node.js อย่างเดียวก็สามารถทำทุกอย่างเหมือนกับ Express ได้ แต่ใช้มันเพื่อทำให้การพัฒนาเว็บแอปพลิเคชันบน Node.js ทำได้ง่ายขึ้น และ Express.js มีคุณสมบัติที่โดดเด่นคือ

- การจัดการ Routing ที่ง่าย
- ฟังก์ชันที่ช่วยสำหรับ HTTP
- สนับสนุน Template engines สำหรับสร้าง View
- ทำงานได้รวดเร็วและมีประสิทธิภาพ
- สนับสนุน Middleware

Express.js เป็นแพ็คเกจที่ได้รับความนิยมเป็นอย่างมากในการพัฒนาเว็บแอปพลิเคชันบน Node.js และมันถูกเผยแพร่อยู่บน NPM ในการใช้งาน Express สามารถช่วยให้เราทำงานได้สะดวกและรวดเร็วขึ้น เนื่องจาก API ส่วนมากของมันถูกออกแบบมาเพื่อการพัฒนาเว็บไซต์โดยเฉพาะและง่ายต่อการใช้งาน แทนที่จะใช้ Low-level APIs ของ Node.js โดยตรง

2.9 JavaScript

JavaScript คือ ภาษาคอมพิวเตอร์สำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต ที่กำลังได้รับความนิยมอย่างสูง JavaScript เป็น ภาษาสคริปต์เชิงวัตถุ (ที่เรียกกันว่า "สคริปต์" (script) ซึ่งในการสร้างและพัฒนาเว็บไซต์ (ใช้ร่วมกับ HTML) เพื่อให้เว็บไซต์ของเราดูมีการเคลื่อนไหว สามารถตอบสนองผู้ใช้งานได้มากขึ้น ซึ่งมีวิธีการทำงานในลักษณะ "แปลความและดำเนินงานไปที่ละคำสั่ง" (interpret) หรือเรียกว่า อ็อบเจ็กโอเรียนเตด (Object Oriented Programming) ที่มีเป้าหมายในการออกแบบและพัฒนาโปรแกรมในระบบอินเทอร์เน็ตสำหรับผู้เขียนด้วยภาษา HTML สามารถทำงานข้ามแพลตฟอร์มได้ โดยทำงานร่วมกับ ภาษา HTML และภาษา Java ได้ทั้งทางฝั่งไคลเอนต์ (Client) และ ทางฝั่งเซิร์ฟเวอร์ (Server)

2.10 JSON

เจสัน (JSON) เป็นรูปแบบสายอักขระ (String) ชนิดหนึ่งที่ถูกจัดเก็บในรูปแบบที่สามารถอ่านและเข้าใจได้ง่าย ถ้าจะให้เข้าใจง่าย ๆ ก็คือเจสัน (JSON) เปรียบเสมือนรูปแบบของอาเรย์ (Array) ชนิดหนึ่งที่ใช้รับส่งข้อมูลผ่านอแจ็กซ์เพราะซึ่งปกติแล้วถ้าเราต้องการรับ-ส่งข้อมูลผ่านอแจ็กซ์ต่าง ๆ นั้นจะต้องรับ-ส่งมาในรูปแบบของสายอักขระทั้งก่อน และเมื่อฝั่งอแจ็กซ์ทำการรับค่าที่ทำการส่งค่ากลับมาจากเซิร์ฟเวอร์ก็จะต้องนำสายอักขระเหล่านั้นมาตัดตำแหน่งที่ต้องการเพื่อเอาสายอักขระตัวที่ต้องการมาใช้ แต่สำหรับเจสัน (JSON) แล้วสามารถรับส่งชุดค่าตัวแปรได้ทั้งฝั่งไคลเอนต์ (Client) และฝั่งเซิร์ฟเวอร์ (Server) โดยทั้ง 2 ฝั่งสามารถทำการเข้ารหัสและถอดรหัสโดยใช้เจสันเอนโค้ด (JSON Encode) และ เจสันดีโค้ด (JSON Decode) เพื่ออ่านค่าตัวแปรเหล่านั้น และจะเรียกใช้งานมันได้อย่างไร ซึ่งปกติแล้วจะอยู่ในรูปแบบของอาเรย์และสำหรับตัวแปรเจสัน (JSON) นั้นไม่จำกัดแค่รับส่งข้อมูลผ่านเว็บเบราว์เซอร์เท่านั้น แต่ยังสามารถนำเจสัน (JSON) ไปประยุกต์กับการรับส่งข้อมูลในรูปแบบอื่น ๆ ได้ เช่นการจับเก็บข้อมูลในรูปแบบของ สายอักขระในข้อความหรือการรับส่งผ่านตัวให้บริการเว็บไซต์ (Web Service) ก็สามารถทำได้เช่นเดียวกัน

2.11 Deep Learning

Deep Learning คือวิธีการเรียนรู้แบบอัตโนมัติด้วยการเลียนแบบการทำงานของโครงข่ายประสาทของมนุษย์ (Neurons) โดยนำระบบโครงข่ายประสาท (Neural Network) มาซ้อนกันหลายชั้น (Layer) และทำการเรียนรู้ข้อมูลตัวอย่าง ซึ่งข้อมูลดังกล่าวถูกนำไปใช้ในการตรวจจับรูปแบบ (Pattern) หรือจัดหมวดหมู่ข้อมูล (Classify the Data)

โดยทั่วไประบบโครงข่ายประสาทจะเรียนรู้ได้เพียงไม่กี่ชั้น เนื่องจากยังไม่มีข้อมูลสอน (Training Data) หรือความสามารถด้านคอมพิวเตอร์ยังไม่สูงพอ อย่างไรก็ตามในช่วงหลายปีมานี้เทคโนโลยีได้มีการพัฒนามากขึ้น จึงทำให้มีข้อมูลชั้นของโครงข่ายได้ง่ายขึ้นและมากขึ้น ยังมีซ้อนกันหลายชั้น โครงข่ายยิ่งมีความซับซ้อนและลึกขึ้น จึงเป็นที่มาของคำว่า Deep Learning ตามรูปแบบของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Machine Learning โดยทั่วไปเมื่อมีข้อมูลดิบเข้ามาจะไม่มีกระบวนการประมวลโดยอัตโนมัติ แต่จะต้องอาศัยความรู้เฉพาะทาง (Domain Knowledge) สำหรับคุณลักษณะในการจัดหมวดหมู่บางประเภท (Hand Craft Features) แต่ถ้าเป็น Deep Learning จะรับข้อมูลดิบเข้ามาทันทีและทำการประมวลอัตโนมัติเพื่อหาข้อมูลตัวอย่างที่จำเป็นในการตรวจจ็กรูปแบบหรือจัดหมวดหมู่ข้อมูล ความสามารถในการเรียนรู้คุณลักษณะอัตโนมัติทำให้ Deep Learning เป็นประโยชน์อย่างยิ่งสำหรับการใช้งานในสถานการณ์ต่าง ๆ สิ่งท้าทายที่ยังต้องเผชิญ คือการหาโครงข่ายระบบประสาทที่เหมาะสมและการค้นหาตัวแปรที่มีผลต่อสมรรถนะในการสอน (Training Performance) ของโครงข่าย ยังคงเป็นเรื่องยากที่จะได้รู้ว่า Deep Learning สามารถเรียนรู้ลักษณะใดบ้าง นอกจากนี้ Deep Learning ยังมีลักษณะไม่ต่างจาก Machine Learning นั่นคือ ยังไม่สามารถจัดการข้อมูลรับเข้าที่มีความละเอียดเฉพาะทาง (Carefully Crafted Input) จึงอาจทำให้โมเดลเกิดการอนุมานผิดพลาด (Wrong Inferences) แสดงดังรูปที่ 2.32



รูปที่ 2.32 ข้อมูลภาพที่ซ้อนกันหลายชั้นโครงข่าย (Layer) [19]

2.12 Convolution Neural Network (CNN)

Convolutional Neural Network (CNN) หรือโครงข่ายประสาทแบบคอนโวลูชันเป็นโครงข่ายประสาทเทียมหนึ่งในกลุ่ม bio-inspired โดยที่ CNN จะจำลองการมองเห็นของมนุษย์ที่มองเห็นพื้นที่เป็นที่ย่อยๆ และนำกลุ่มของพื้นที่ย่อยๆ มาผสานกัน เพื่อดูว่าสิ่งที่เห็นอยู่เป็นอะไร

การมองเห็นพื้นที่ย่อยของมนุษย์จะมีการแยกคุณลักษณะ (feature) ของพื้นที่ย่อยนั้น เช่น ลายเส้นและการตัดกันของสี ซึ่งการที่มนุษย์รู้ว่าพื้นที่ตรงนี้เป็นเส้นตรงหรือสี่เหลี่ยมเพราะมนุษย์ดูทั้งจุดที่สนใจและบริเวณรอบ ๆ ประกอบกัน

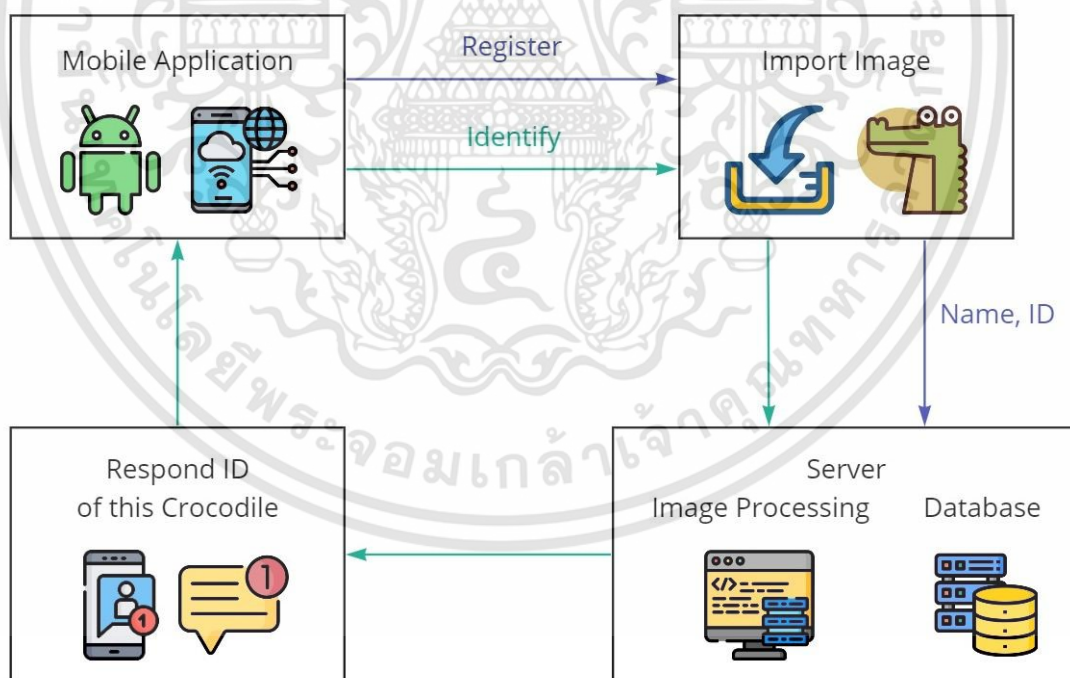
บทที่ 3

การออกแบบและการจัดทำปฏิญานิพนธ์

3.1 การออกแบบ

3.1.1 การออกแบบการทำงานของระบบ

โปรแกรมประยุกต์การพิสูจน์เอกลักษณ์จระเข้โดยใช้การประมวลผลภาพ แบ่งการทำงานเป็น 2 ส่วน ได้แก่ การลงทะเบียนจระเข้ และการพิสูจน์เอกลักษณ์จระเข้ โดยการลงทะเบียนจระเข้มีวิธีการทำงานเริ่มต้นจากใช้โปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ (Mobile Application) นำเข้ารูปภาพหลายหน้าห้องจระเข้ พร้อมทั้งระบุชื่อและหมายเลขประจำตัวจระเข้ จากนั้นผ่านกระบวนการประมวลผลภาพแล้วเก็บค่าที่ได้ไปยังฐานข้อมูล และการพิสูจน์เอกลักษณ์จระเข้มีวิธีการทำงานเริ่มต้นจากใช้โปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ (Mobile Application) นำเข้ารูปภาพหลายหน้าห้องจระเข้ จากนั้นผ่านกระบวนการประมวลผลภาพแล้วนำค่าที่ได้มาพิสูจน์เอกลักษณ์ว่าตรงกับเอกลักษณ์ของจระเข้ใดในฐานข้อมูล และส่งผลลัพธ์ที่ได้กลับมาแสดงที่โปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ (Mobile Application) โดรนแผนภาพการทำงานแสดงดังรูปที่ 3.1



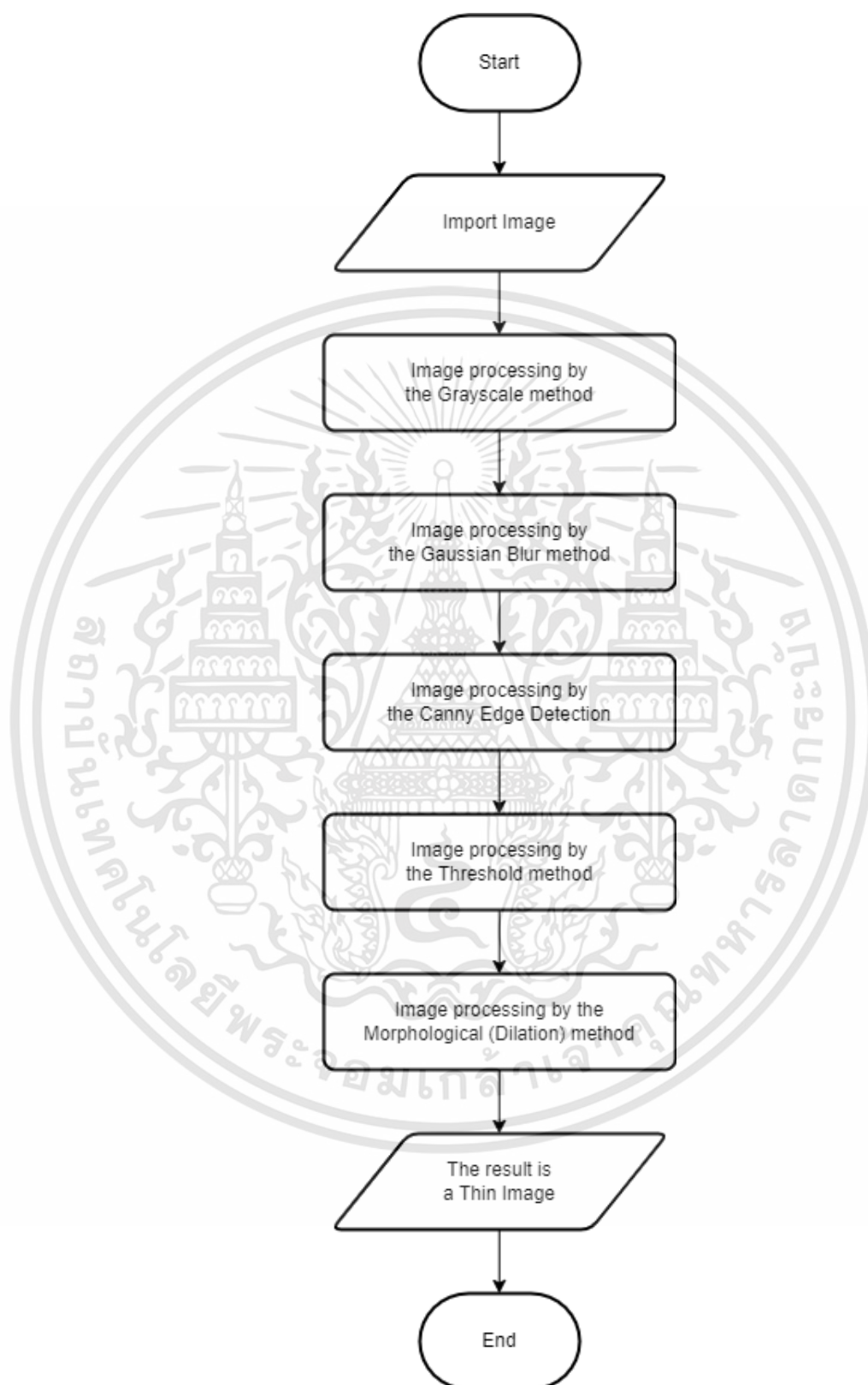
รูปที่ 3.1 แผนภาพการทำงานของโปรแกรมประยุกต์

3.1.2 การออกแบบอัลกอริทึมในการวิเคราะห์รูปภาพ

ในส่วนของอัลกอริทึมนี้จะนำรูปภาพจากข้อมูลที่เก็บ มาวิเคราะห์และสังเคราะห์ภาพให้ได้เป็นค่าของตำแหน่งและจำนวนของจุดสนใจในที่ขอเรียกแทนจุดสนใจนี้ว่า “จุด key point” ซึ่งขั้นตอนทั้งหมดจะไล่เรียงตั้งแต่เป็นรูปภาพ Raw Image ที่เป็นไฟล์ชนิดนามสกุล .png และนำไปผ่านกระบวนการวิธีการประมวลผลภาพและกระบวนการทำให้เส้นหน้าท้องที่ได้มีความกว้าง 1 pixel จนได้รูปภาพที่พร้อมสำหรับการใช้งานอัลกอริทึม แล้วจึงนำเข้าใช้งานด้วยอัลกอริทึมกวาดหาจุด key point โดยมีเงื่อนไขในการหาดังนี้

3.1.2.1 ขั้นตอนการแปลงรูปภาพจาก Physical ให้เหลือเพียงเส้นหน้าท้อง จระเข้ที่สนใจ แสดงแผนผังการทำงานของอัลกอริทึมดังรูปที่ 3.2 โดยมีขั้นตอนการทำงานดังนี้

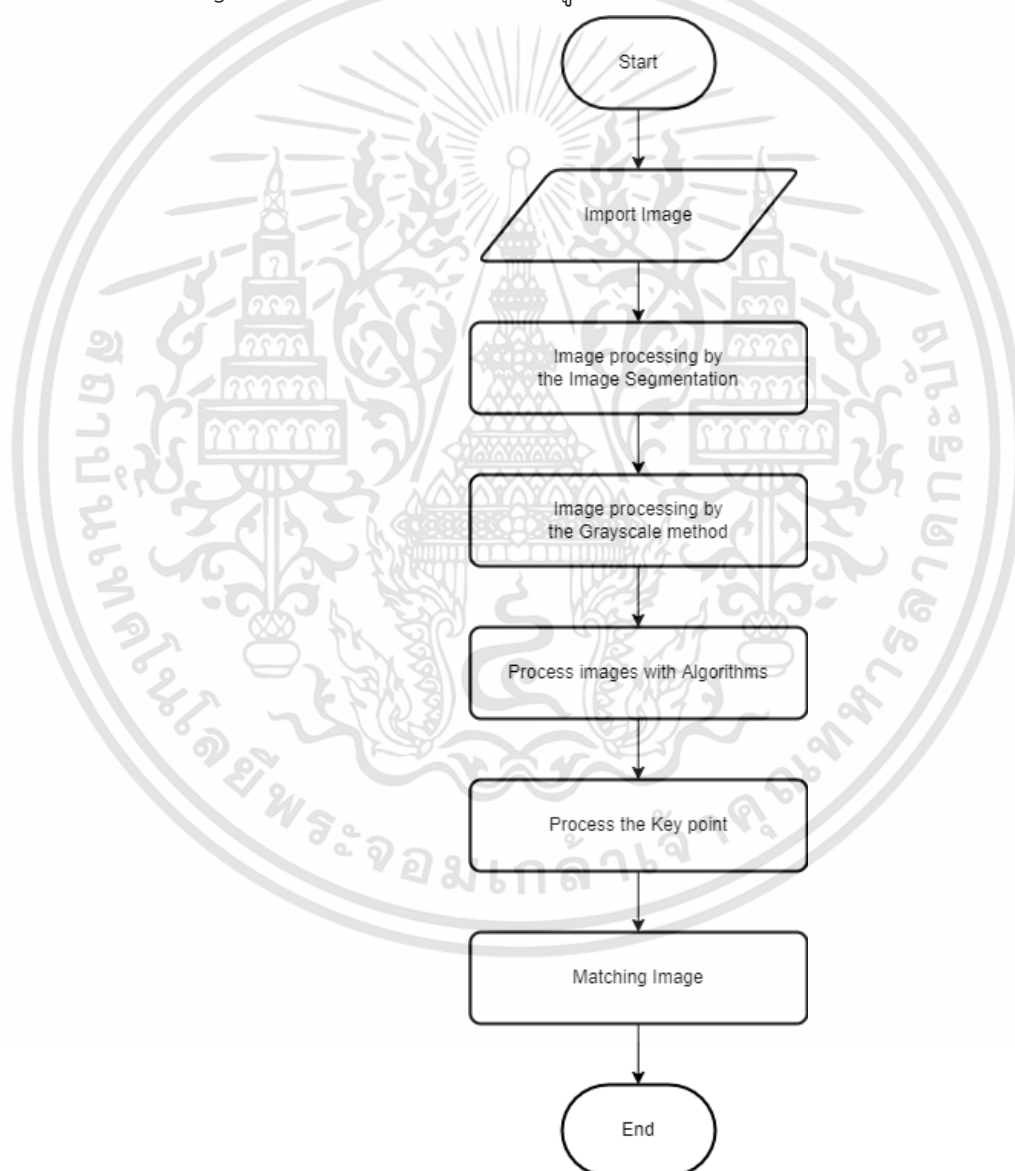
- 1) ทำ Gaussian Blur เพื่อทำให้ค่า Pixel รูปภาพที่ได้มีค่าเฉลี่ยในระดับค่าสีเดียวกัน หรือใกล้เคียงกัน เพื่อลด Noise ที่เกิดขึ้น
- 2) นำทฤษฎี Edge Detection มา Convert Image ให้เป็นขอบ โดยใช้วิธีการตรวจจับขอบที่เรียกว่า วิธีแคนนี่ (Canny method)
- 3) เขียนเงื่อนไขของค่าสีในรูปภาพเพื่อสลับส่วนของเส้นให้ชัดเจน
- 4) ทำการ Morphological (Dilation) ขยายเส้นเพื่อต่อส่วนที่ขาดหายไป ในระหว่างขั้นตอนและทำให้ขอบของเส้นหน้าท้องกลับมาเป็นเส้นเดียว
- 5) ขั้นตอนในการทำให้เส้นที่ได้มีขนาดความกว้างเป็น 1 pixel



รูปที่ 3.2 ขั้นตอนในการแปลงภาพเป็นเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.2 การใช้โปรแกรม MATLAB ในการวิเคราะห์รูปภาพพิจารณาจาก Gray Scale ขั้นตอนทั้งหมดจะไล่เรียงตั้งแต่เป็นรูปภาพ Raw Image ที่เป็นไฟล์ชนิดนามสกุล .png ที่ได้มาจากอุปกรณ์ในส่วนแรก และนำไปผ่านกระบวนการวิธีการประมวลผลภาพตามทฤษฎีของแต่ละอัลกอริทึม เพื่อให้ได้จำนวนและตำแหน่งของ key point รูปแบบนั้น ๆ เพื่อนำข้อมูลดังกล่าวไปใช้ในการเปรียบเทียบและตัดสินใจระบุอัตลักษณ์จะเชื่อมต่อไป แสดงแผนผังการทำงานของอัลกอริทึม ดังรูปที่ 3.3 โดยใช้อัลกอริทึมจากการศึกษาทั้งหมด 8 ตัว ได้แก่ 1) ORB 3) BRISK 4) Harris 5) KAZE 6) SURF 7) MinEigen 8) MSER ในการวิเคราะห์รูปถ่ายหน้าห้องจระเข้

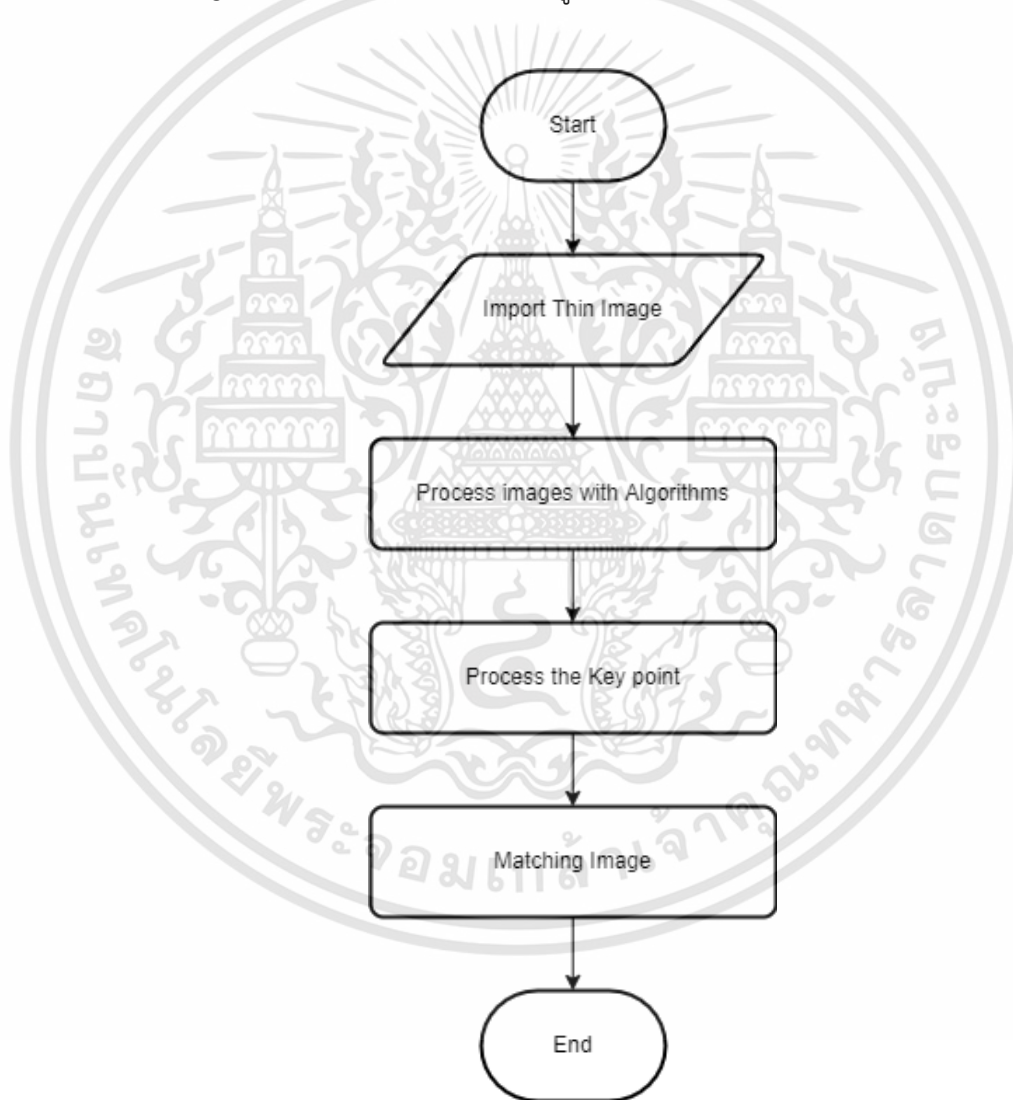


รูปที่ 3.3 แผนผังการทำงานของโปรแกรมวิเคราะห์รูปภาพ Gray Scale

ของลายหน้าห้องของจระเข้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

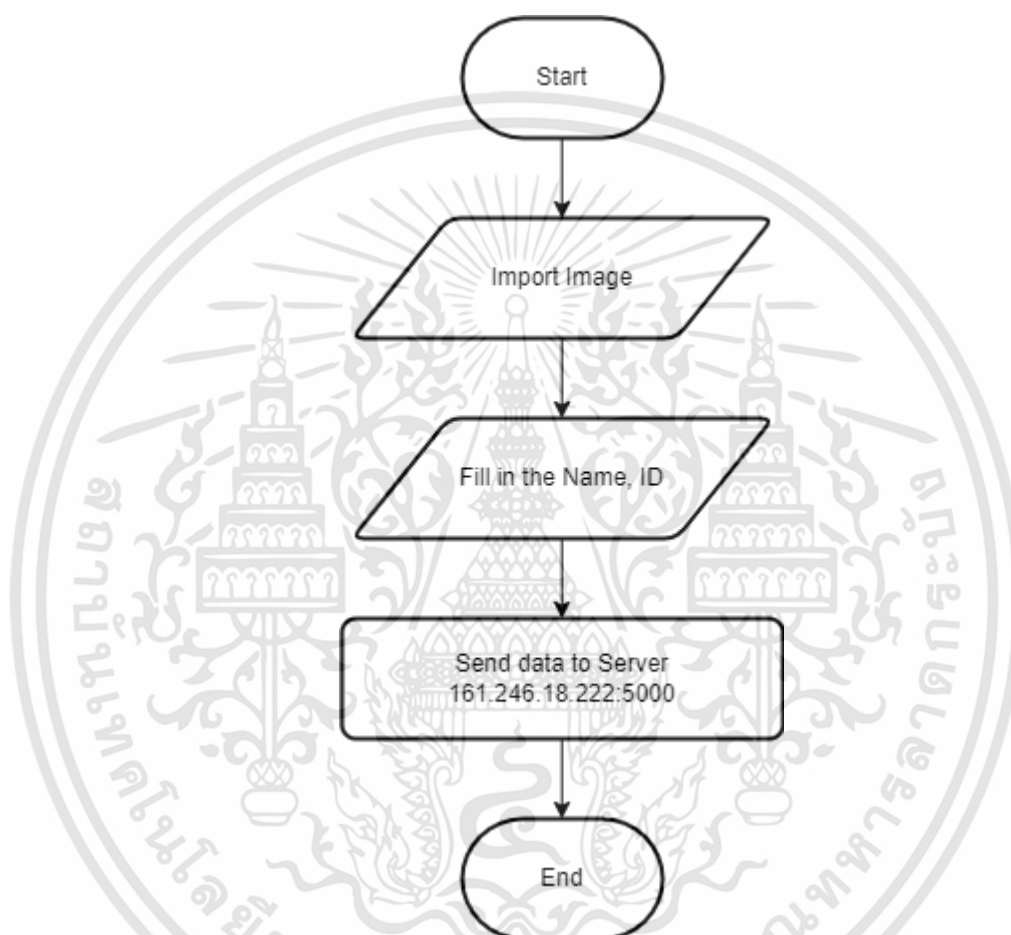
3.1.2.2 การใช้โปรแกรม MATLAB ในการวิเคราะห์รูปภาพ พิจารณาจาก Thin Image ขั้นตอนทั้งหมดจะไล่เรียงตั้งแต่เป็นรูปภาพ Raw Image ที่เป็นไฟล์ชนิดนามสกุล .png ที่ได้มาจากอุปกรณ์ในส่วนแรก และนำไปผ่านกระบวนการวิธีการประมวลผลภาพตามทฤษฎีของแต่ละอัลกอริทึม เพื่อให้ได้จำนวนและตำแหน่งของ key point รูปแบบนั้น ๆ เพื่อนำข้อมูลดังกล่าวไปใช้ในการเปรียบเทียบและตัดสินใจระบุอัตลักษณ์จะเชื่อมต่อไป แสดงแผนผังการทำงานของอัลกอริทึม ดังรูปที่ 3.4 โดยใช้อัลกอริทึมจากการศึกษาทั้งหมด 8 ตัว ได้แก่ 1) ORB 3) BRISK 4) Harris 5) KAZE 6) SURF 7) MinEigen 8) MSER ในการวิเคราะห์รูปถ่ายหน้าห้องจระเข้



รูปที่ 3.4 แผนผังการทำงานของโปรแกรมวิเคราะห์รูปภาพ Thin Image
ของลายหน้าห้องของจระเข้

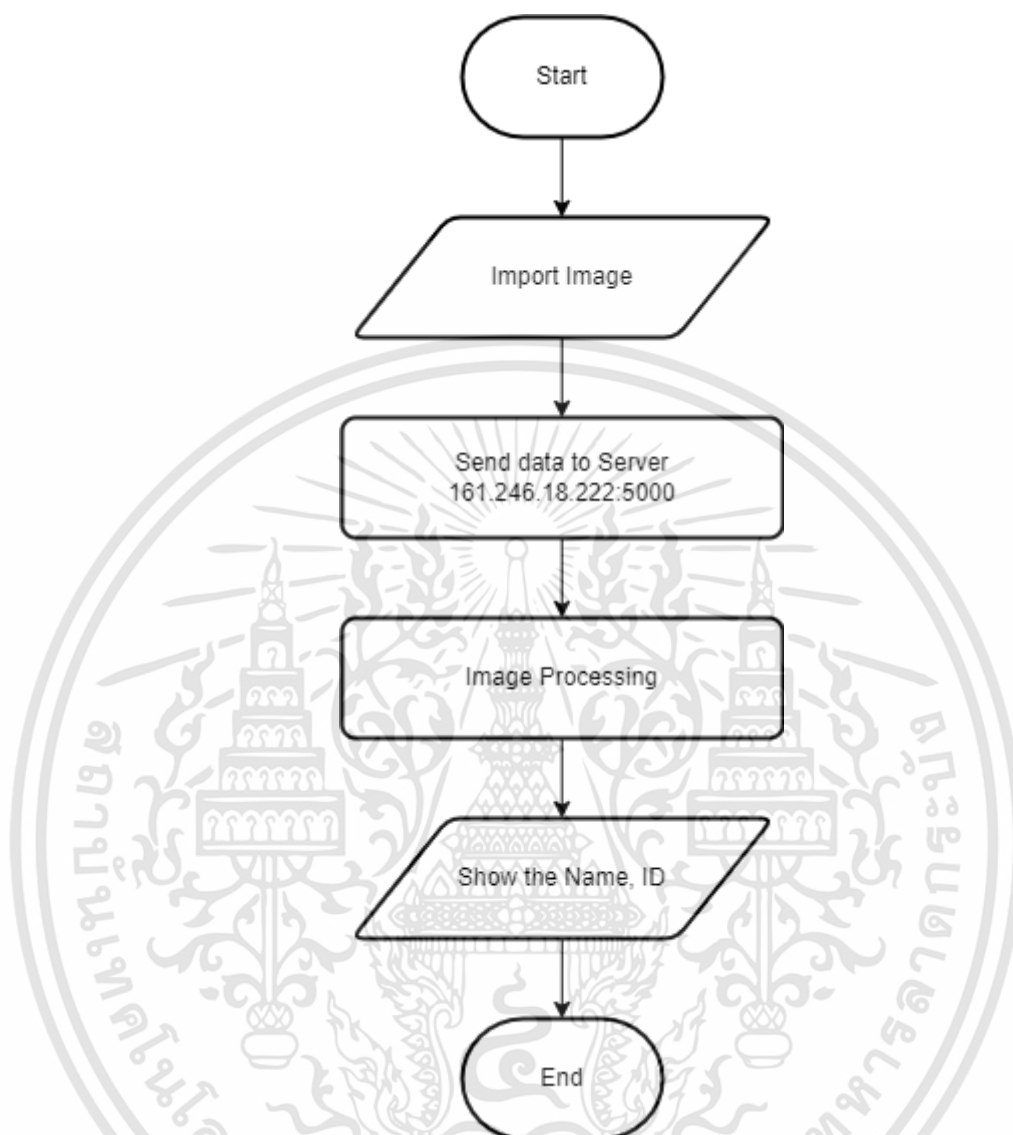
3.1.3 ออกแบบโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ (Mobile Application)

3.1.3.1 ฟังก์ชัน Register ผู้จัดทำได้ทำการออกแบบโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่สำหรับการลงทะเบียนจะเข้าด้วยการส่งรูปถ่ายหน้าห้องจะเข้าพร้อมระบุชื่อและหมายเลขไอดีไปเก็บยังเซิร์ฟเวอร์



รูปที่ 3.5 แผนผังการทำงานของโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ฟังก์ชัน Register

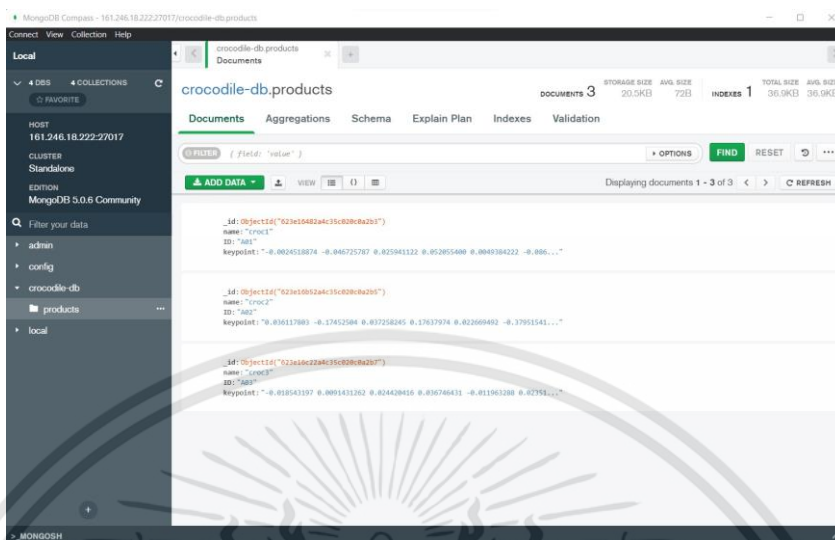
3.1.3.2 ฟังก์ชัน Identify ผู้จัดทำได้ทำการออกแบบโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่สำหรับการพิสูจน์เอกลักษณ์จะเข้าด้วยการส่งรูปถ่ายหน้าห้องจะเข้าไปยังเซิร์ฟเวอร์แล้วผ่านกระบวนการ image processing เพื่อระบุเอกลักษณ์จะเข้าแล้วส่งหมายเลขไอดีมาแสดงยังโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่



รูปที่ 3.6 แผนผังการทำงานของโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ฝั่ง Identify

3.1.4 ออกแบบฐานข้อมูลเก็บรูปภาพบนเซิร์ฟเวอร์

ผู้จัดทำได้ทำการออกแบบฐานข้อมูลที่ใช้เก็บข้อมูลรูปภาพมาบันทึกไว้บนเซิร์ฟเวอร์ ส่วนตัวที่มีหมายเลข IP คือ 161.246.18.222 และ Port ที่ใช้คือ 5000 และได้ใช้งาน Node.js ร่วมกับ Framework เพื่อทำงานเป็นเว็บเซิร์ฟเวอร์และจัดการกระทำต่าง ๆ กับฐานข้อมูลที่ใช้เป็น MongoDB โดยค่าที่บันทึกที่ฐานข้อมูลนี้จะเป็นข้อมูลเฉพาะตัวของจะเข้อันได้แก่ ID คือ หมายเลขประจำตัวจะเข้ , name คือ ชื่อแทนตัวจะเข้, key point คือ ค่าเอกลักษณ์ที่ได้จากการประมวลผลหน้าท้องจะเข้ แสดงดังรูปที่ 3.7



รูปที่ 3.7 ฐานข้อมูลที่ใช้เป็น MongoDB บันทึกค่า

3.2 เครื่องมือที่ใช้ในการทดลอง

ในปริญญานิพนธ์นี้ มีอุปกรณ์และเครื่องมือที่ใช้ในการทดลอง ดังนี้

3.2.1 โปรแกรม MATLAB

โดยการทดลองอัลกอริทึมแต่ละตัวนั้น จะทำการทดลองโดยใช้โปรแกรม MATLAB ในการวิเคราะห์เพื่อตรวจสอบว่าอัลกอริทึมตัวใดมีความเหมาะสมในการจับคู่ (Matching) จะระเข้มากที่สุด

3.3 การจัดเก็บผลการทดลอง

3.3.1 ทดสอบการแปลงรูปภาพจาก Physical ให้เหลือเพียงเส้นหน้าท้องที่สนใจ

3.3.2 ทดสอบอัลกอริทึมในการวิเคราะห์รูปภาพ โดยพิจารณาจาก Gray Scale

3.3.2.1 ORB

3.3.2.2 FAST

3.3.2.3 BRISK

3.3.2.4 Harris

3.3.2.5 KAZE

3.3.2.6 SURF

3.3.2.7 MinEigen

3.3.2.8 MSER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3 ทดสอบอัลกอริทึมในการวิเคราะห์รูปภาพ โดยพิจารณาจาก Thin Image

3.3.3.1 ORB

3.3.3.2 FAST

3.3.3.3 BRISK

3.3.3.4 Harris

3.3.3.5 KAZE

3.3.3.6 SURF

3.3.3.7 MinEigen

3.3.3.8 MSER

3.3.4 คำนวณหาเปอร์เซ็นต์ความถูกต้องการ Matching ทุกอัลกอริทึม

3.3.5 ทดสอบอัลกอริทึม KAZE ในการประมวลผลสำหรับโปรแกรมประยุกต์บน

โทรศัพท์เคลื่อนที่ (Mobile Application)

ทำการทดสอบโดยใช้อัลกอริทึม KAZE ในการประมวลผลเนื่องจากมีเปอร์เซ็นต์ความถูกต้องในการ Matching มากที่สุด

3.3.6 ทดสอบโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ (Mobile Application)

3.3.6.1 ฝั่ง Register ส่งรูปภาพ Name และ ID ไปเก็บยังเซิร์ฟเวอร์

3.3.6.2 ฝั่ง Identify ส่งรูปภาพแล้วรอการยืนยันว่าเป็นหมายเลขไอดี และชื่อของหน้าห้องจะระเซ่ตัวใด

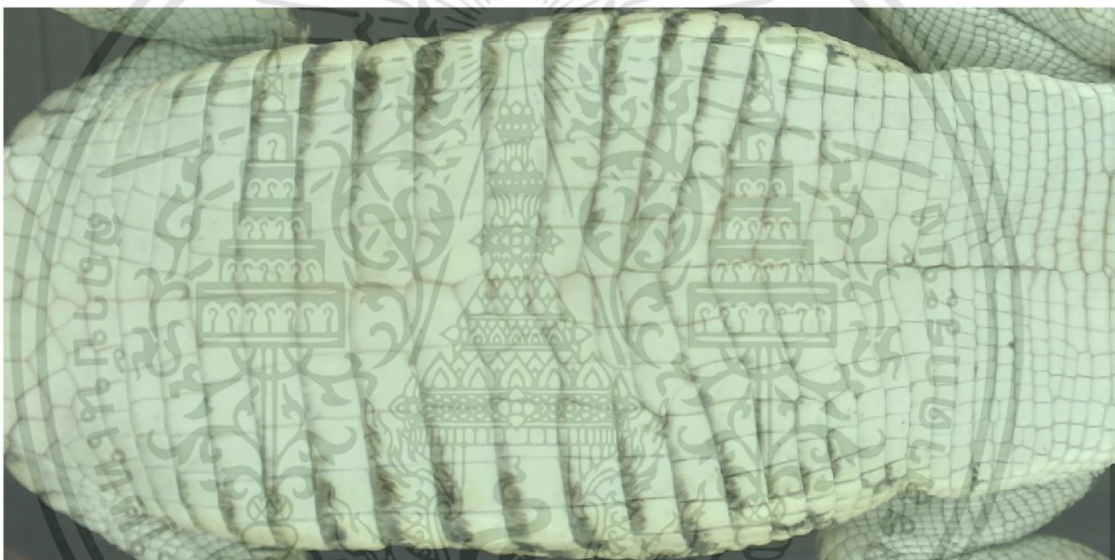
บทที่ 4

ผลการทดลอง

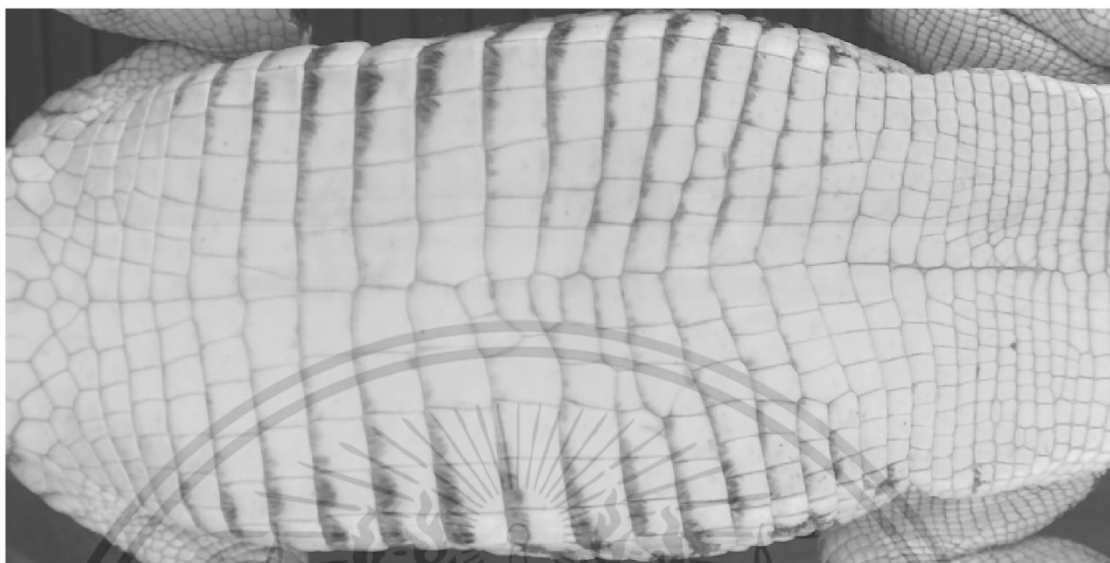
สำหรับการทดลองระบบการพิสูจน์เอกลักษณ์จะเข้าได้จัดเก็บผลการทดลอง โดยมีการแบ่งการทดลองและจัดเก็บผลการทดลองออกเป็นส่วน ๆ ดังนี้

4.1 การทดสอบการแปลงรูปภาพจาก Physical ให้เหลือเพียงเส้นหน้าท้องที่สนใจ

ในการทดลองผู้จัดทำรูปภาพจะเข้ามาแปลงเป็นภาพ Grayscale ที่แสดงถึงการแบ่งส่วนแต่ละพิกเซลในภาพ เพื่อง่ายต่อการใช้งานในขั้นตอนถัดไป แสดงดังรูปที่ 4.1 และรูปที่ 4.2



รูปที่ 4.1 รูปภาพหนังหน้าท้องจระเข้อินพุด



รูปที่ 4.2 รูปภาพ Grayscale

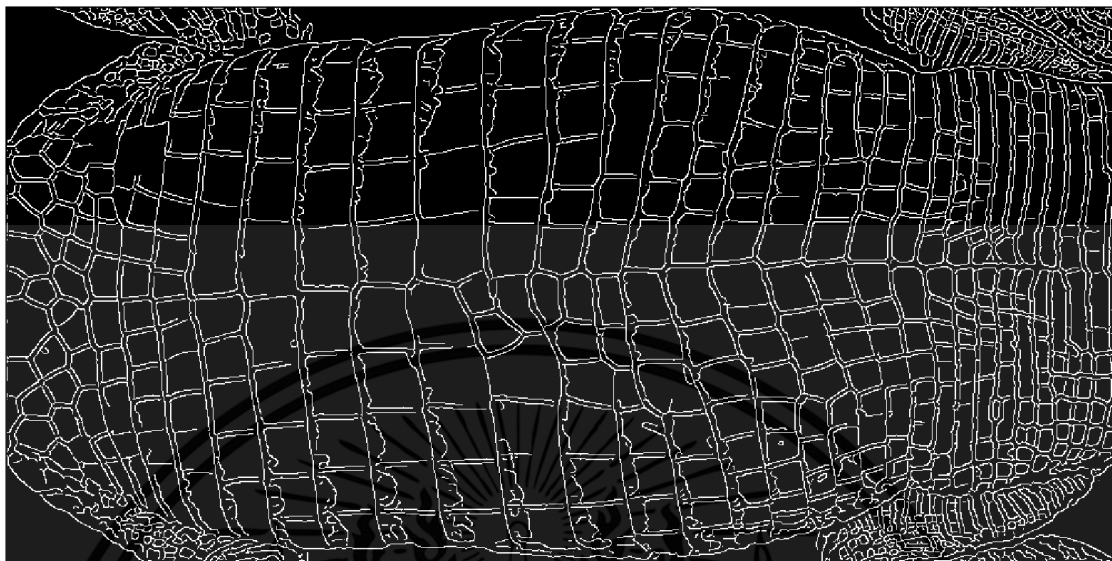
นำมาประมวลผลภาพแบบเกาส์เซียนเบลอ (Gaussian Blur) เพื่อกำจัดสัญญาณรบกวนของภาพ ทำให้รูปภาพลายหน้าท้องจะเข้มมีระดับสีเดียวกันหรือใกล้เคียง แสดงดังรูปที่ 4.3



รูปที่ 4.3 รูปภาพหน้าท้องจระเข้หลังการทำ Gaussian Blur

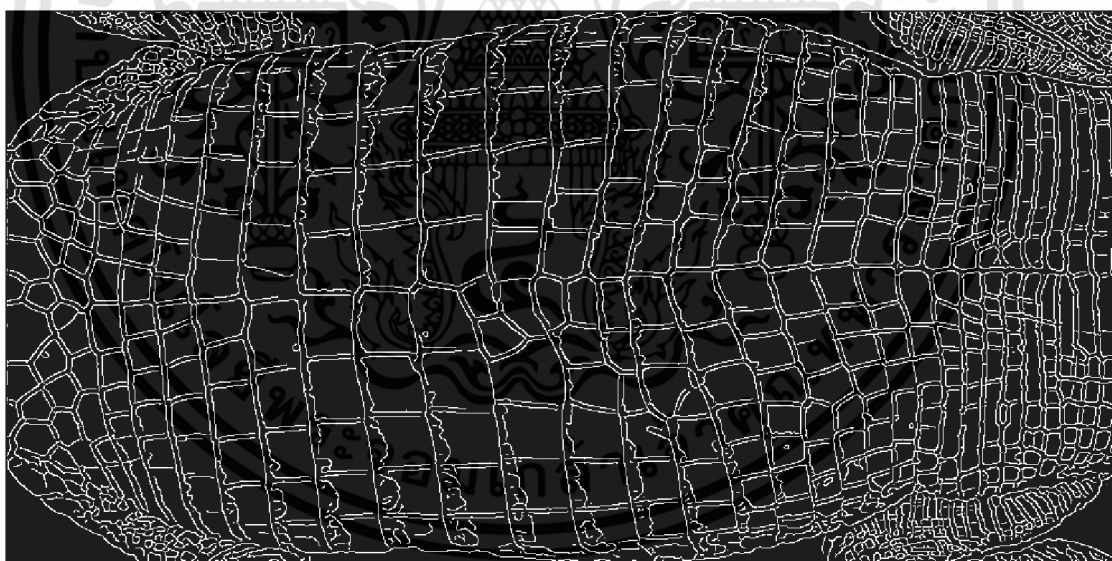
ผู้จัดทำได้ใช้กระบวนการตรวจจับขอบ (Edge Detection) ที่เรียกว่า วิธีแคนนี่ โดยการเปรียบเทียบขนาดของการเปลี่ยนแปลงค่าสีของพิกเซลนั้นกับพิกเซลข้างเคียงที่อยู่ในทิศทางของการเปลี่ยนแปลงค่าสี และการใช้ค่าเกณฑ์สองระดับ (ค่าเกณฑ์สูง และค่าเกณฑ์ต่ำ) เพื่อกำหนดพิกเซลที่เป็นขอบเข้มและขอบไม่เข้ม แสดงดังรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 Canny Edge Detection

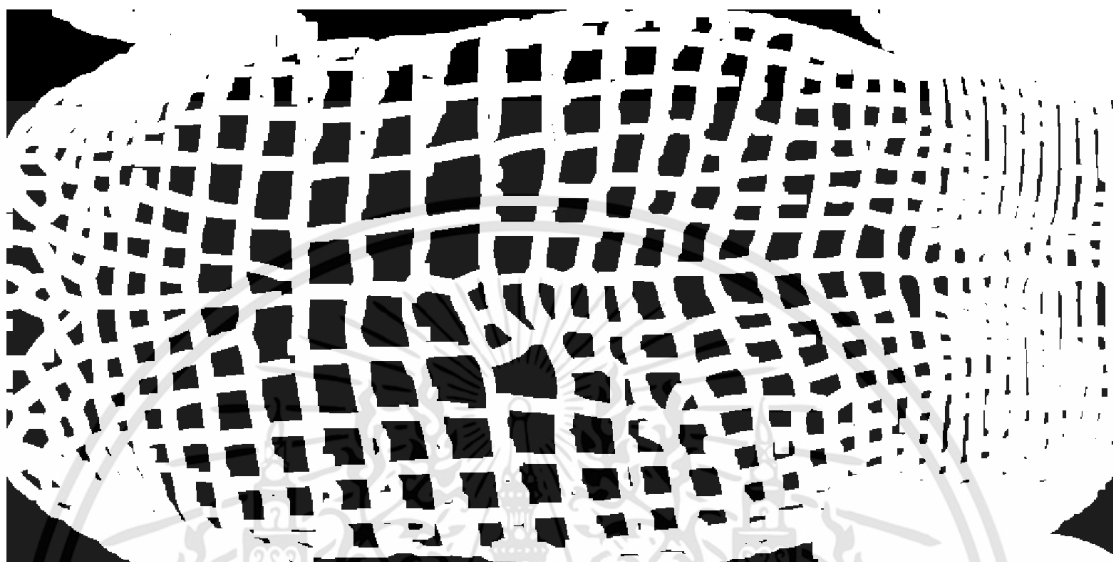
นำมาผ่านกระบวนการ Thresholding โดยการสร้างเงื่อนไขของค่าสีในรูปภาพ เพื่อสลับส่วนของเส้นให้ชัดเจน แสดงดังรูปที่ 4.5



รูปที่ 4.5 รูปที่ผ่านกระบวนการ Thresholding

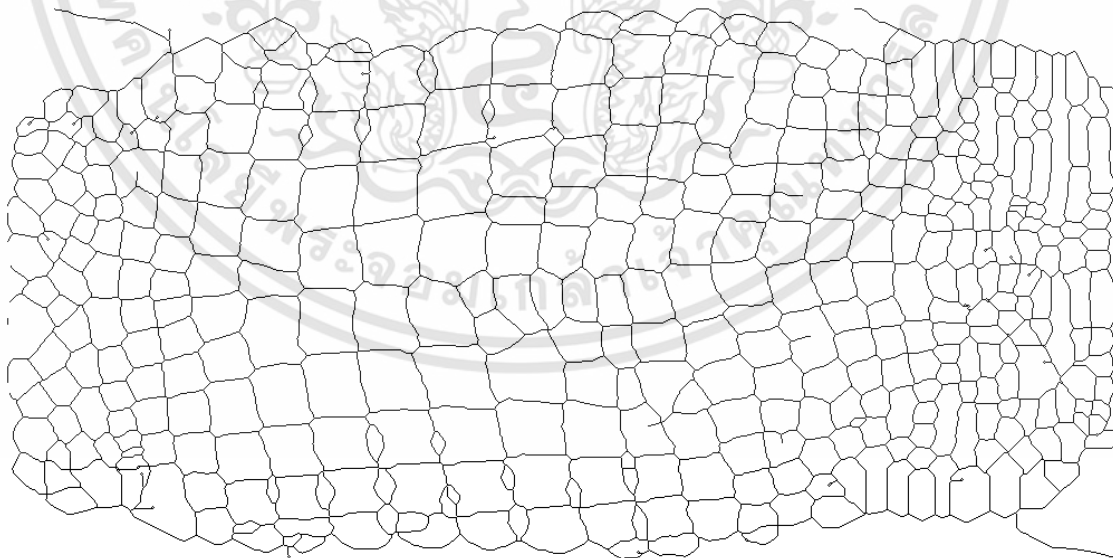
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำมาทำ Morphological (Dilation) เป็นการขยายเส้นลายหน้าท้องจระเข้เพื่อต่อส่วนที่ขาดหายไป ทำให้ขอบของเส้นลายหน้าท้องจระเข้กลับมาเป็นเส้นเดียว แสดงดังรูปที่ 4.6



รูปที่ 4.6 รูปภาพที่ผ่านการ Morphological (Dilation)

นำมาทำเป็น Thin Image ให้เส้นลายหน้าท้องจระเข้มีขนาดความกว้างเป็น 1 pixel เพื่อง่ายต่อการหา key point แสดงดังรูปที่ 4.7



รูปที่ 4.7 รูปภาพที่ผ่านการ Thinned

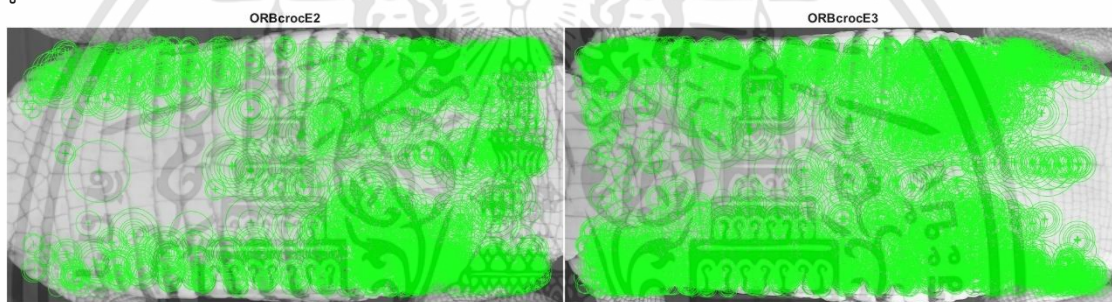
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดสอบอัลกอริทึมในการวิเคราะห์รูปภาพ โดยพิจารณาจาก Gray Scale

ผู้จัดทำได้นำภาพ Grayscale ลายหน้าห้องพระเข้ที่แสดงถึงการแบ่งส่วนแต่ละพิกเซลในภาพมาทดลองหา key point และ Matching จำนวน 3 ตัว โดยพระเข้ในช่วงวัย 3 เดือน แทนเป็น E2, A2 และ D1 และ ในช่วงวัย 6 เดือนแทนเป็น E3, A3 และ D2 เพื่อทดลองว่าอัลกอริทึมตัวใดมีการ Matching ได้ดีที่สุด

4.2.1 ORB

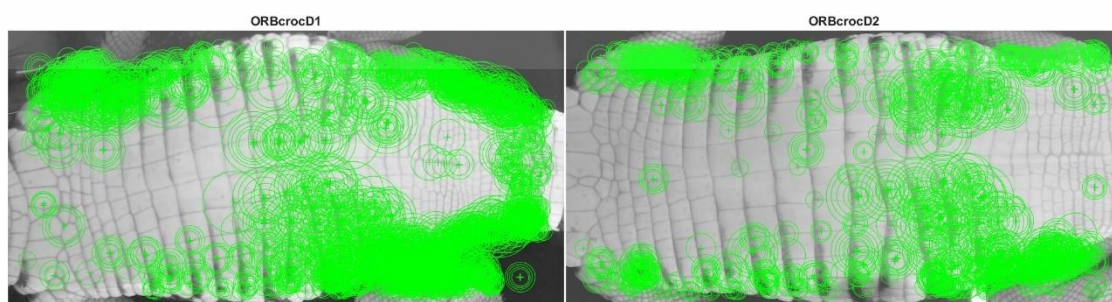
การหา key point ของภาพใช้อัลกอริทึม ORB โดย key point ที่ได้สามารถหาได้สูงสุด 5359 key point โดยจัดให้ภาพให้อยู่ในบริเวณหน้าต่างของพระเข้ แสดงดังรูปที่ 4.8 รูปที่ 4.9 และรูปที่ 4.10



รูปที่ 4.8 การหา key point ภาพ Grayscale พระเข้ E2/E3 ของอัลกอริทึม ORB



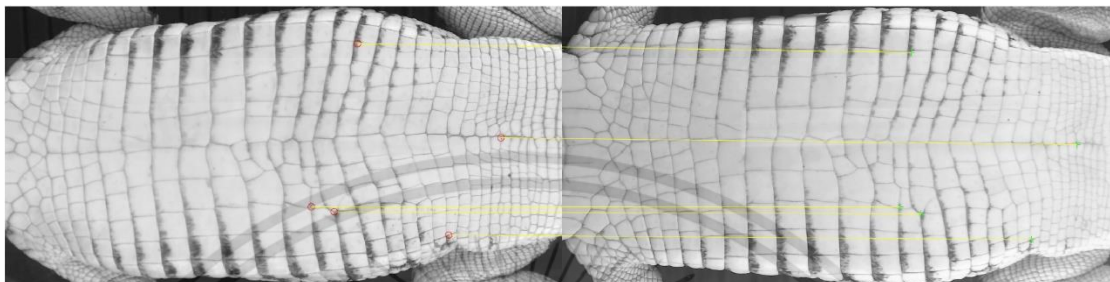
รูปที่ 4.9 การหา key point ภาพ Grayscale พระเข้ A2/A3 ของอัลกอริทึม ORB



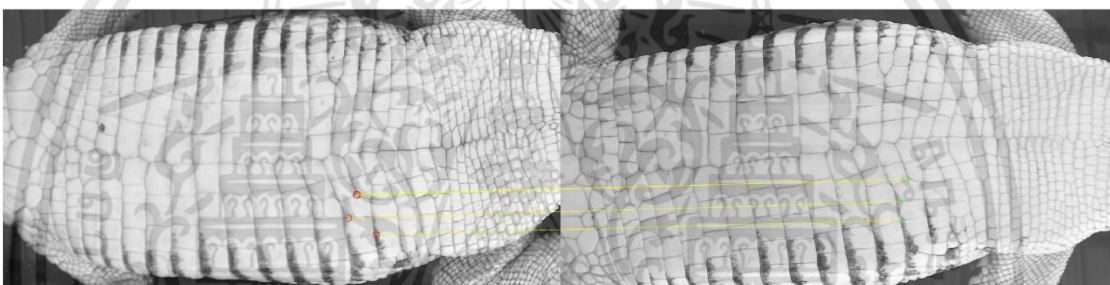
รูปที่ 4.10 การหา key point ภาพ Grayscale พระเข้ D1/D2 ของอัลกอริทึม ORB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

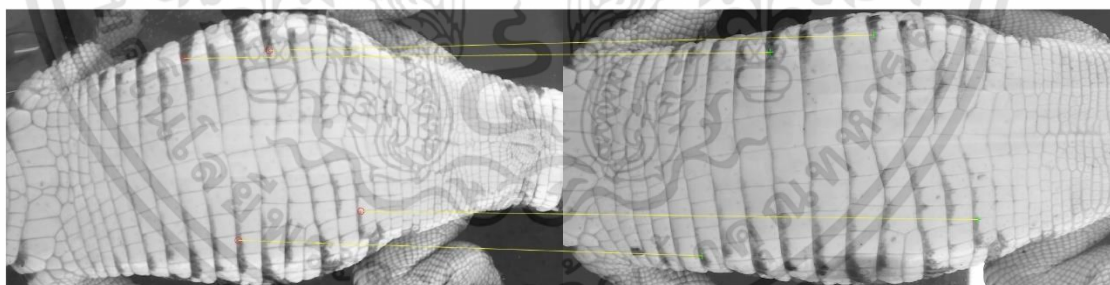
จากนั้นนำอัลกอริทึม ORB มาใช้ในการเปรียบเทียบภาพหลายหน้าห้องจระเข้เพื่อแสดงภาพที่มีการ Matching แสดงดังรูปที่ 4.11 รูปที่ 4.12 และรูปที่ 4.13



รูปที่ 4.11 การหา Matching ภาพ Grayscale จระเข้ E2/E3 ของอัลกอริทึม ORB



รูปที่ 4.12 การหา Matching ภาพ Grayscale จระเข้ A2/A3 ของอัลกอริทึม ORB

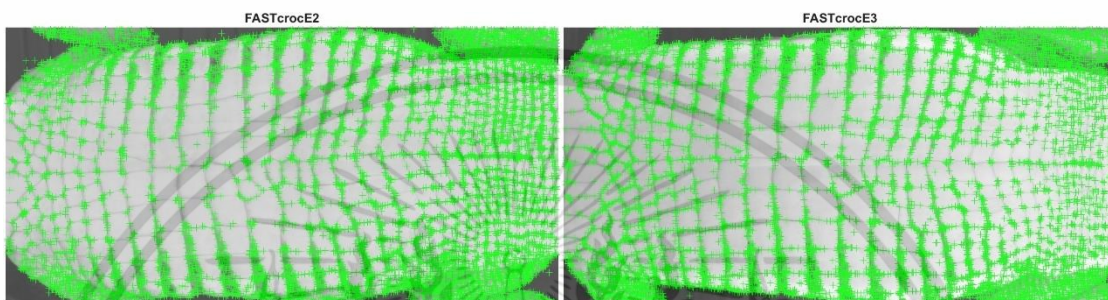


รูปที่ 4.13 การหา Matching ภาพ Grayscale จระเข้ D1/D2 ของอัลกอริทึม ORB

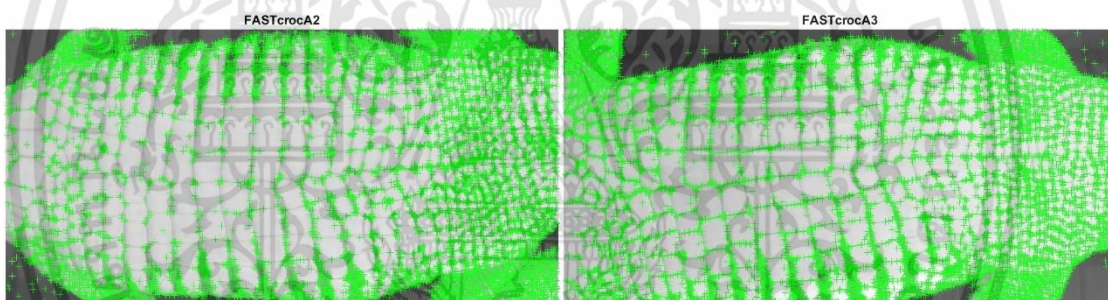
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2 FAST

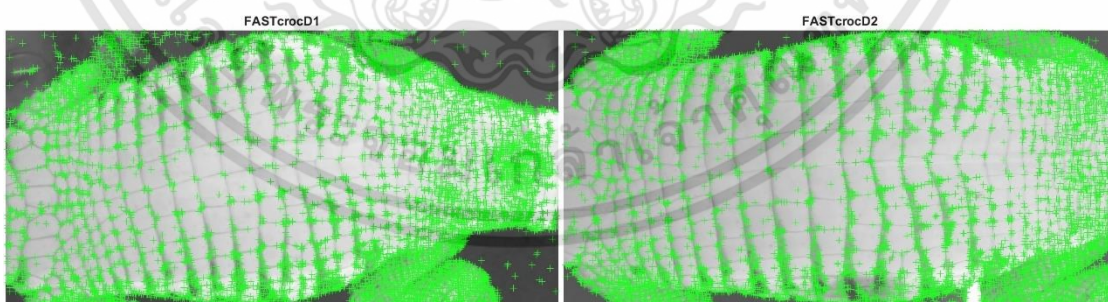
การหา key point ของภาพใช้อัลกอริทึม FAST โดย key point ที่ได้สามารถหาได้สูงสุด 8042 key point โดยจัดให้ภาพให้อยู่ในบริเวณหน้าท้องของจระเข้ แสดงดังรูปที่ 4.14 รูปที่ 4.15 และรูปที่ 4.16



รูปที่ 4.14 การหา key point ภาพ Grayscale จระเข้ E2/E3 ของอัลกอริทึม FAST



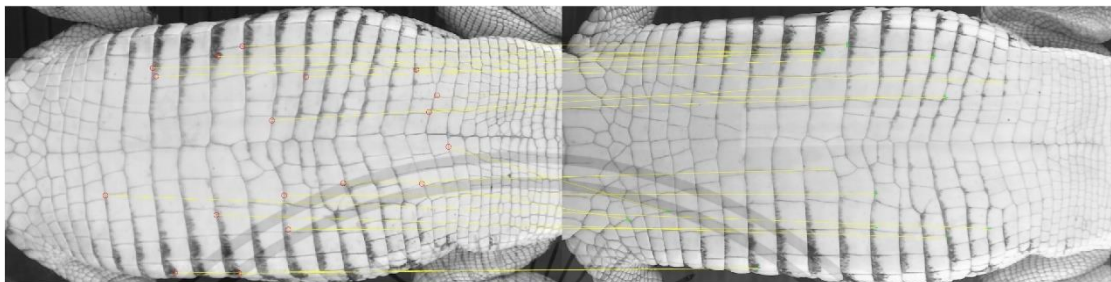
รูปที่ 4.15 การหา key point ภาพ Grayscale จระเข้ A2/A3 ของอัลกอริทึม FAST



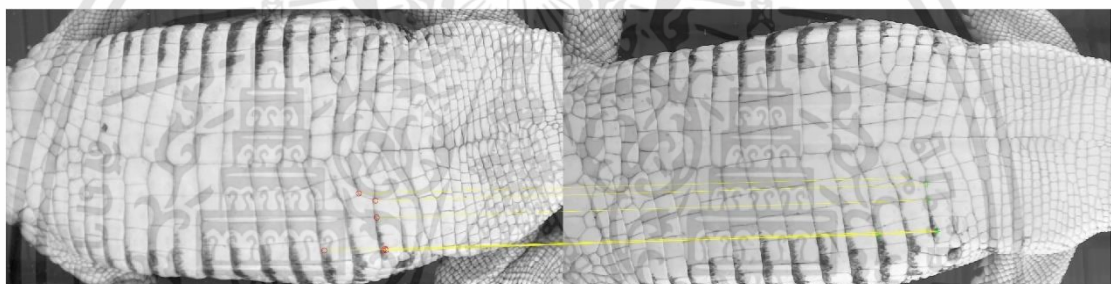
รูปที่ 4.16 การหา key point ภาพ Grayscale จระเข้ D1/D2 ของอัลกอริทึม FAST

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

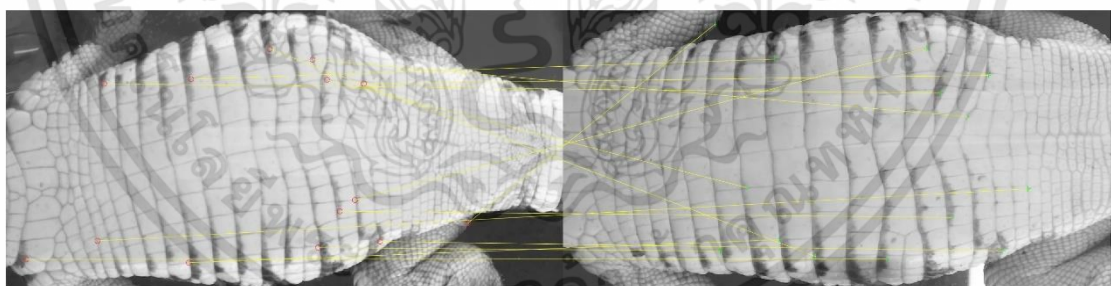
จากนั้นนำอัลกอริทึม FAST มาใช้ในการเปรียบเทียบภาพหลายหน้าหน้าท้องจระเข้เพื่อแสดงภาพที่มีการ Matching แสดงดังรูปที่ 4.17 รูปที่ 4.18 และรูปที่ 4.19



รูปที่ 4.17 การหา Matching ภาพ Grayscale จระเข้ E2/E3 ของอัลกอริทึม FAST



รูปที่ 4.18 การหา Matching ภาพ Grayscale จระเข้ A2/A3 ของอัลกอริทึม FAST

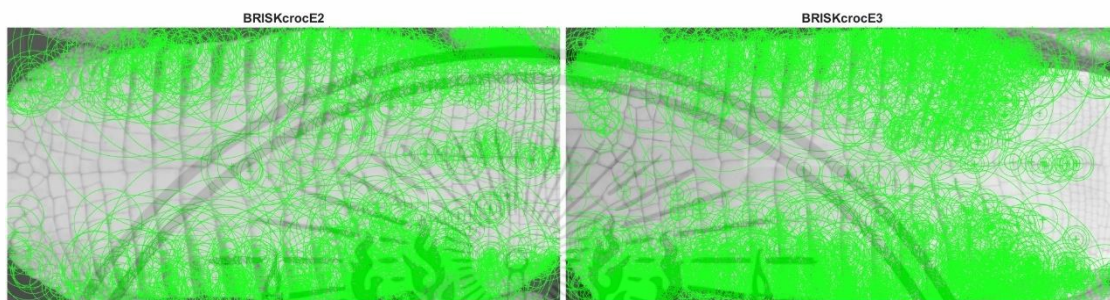


รูปที่ 4.19 การหา Matching ภาพ Grayscale จระเข้ D1/D2 ของอัลกอริทึม FAST

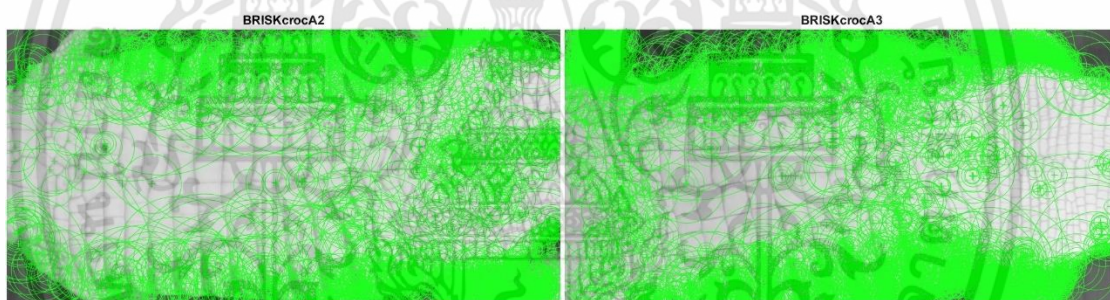
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.3 BRISK

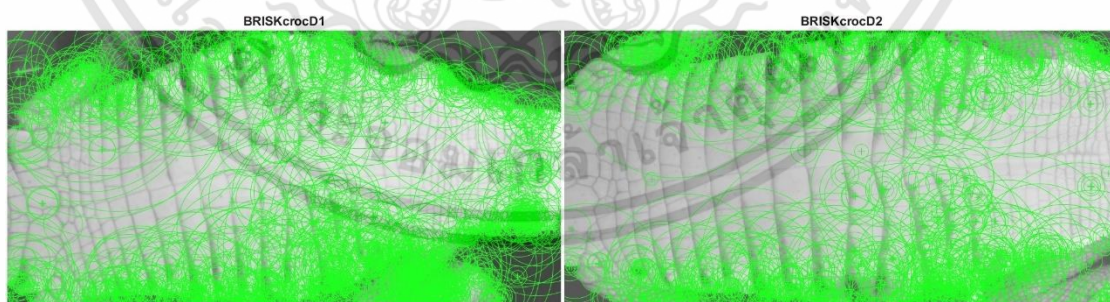
การหา key point ของภาพใช้อัลกอริทึม BRISK โดย key point ที่ได้สามารถหาได้สูงสุด 2643 key point โดยจัดให้ภาพให้อยู่ในบริเวณหน้าต่างของกระเซ่ แสดงดังรูปที่ 4.20 รูปที่ 4.21 และรูปที่ 4.22



รูปที่ 4.20 การหา key point ภาพ Grayscale กระเซ่ E2/E3 ของอัลกอริทึม BRISK

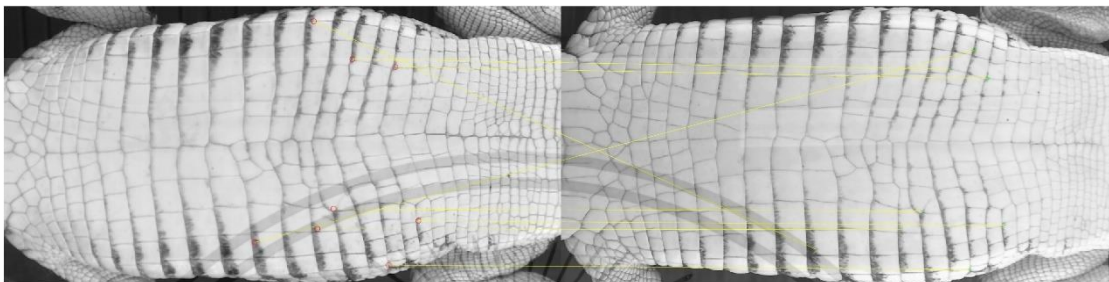


รูปที่ 4.21 การหา key point ภาพ Grayscale กระเซ่ A2/A3 ของอัลกอริทึม BRISK

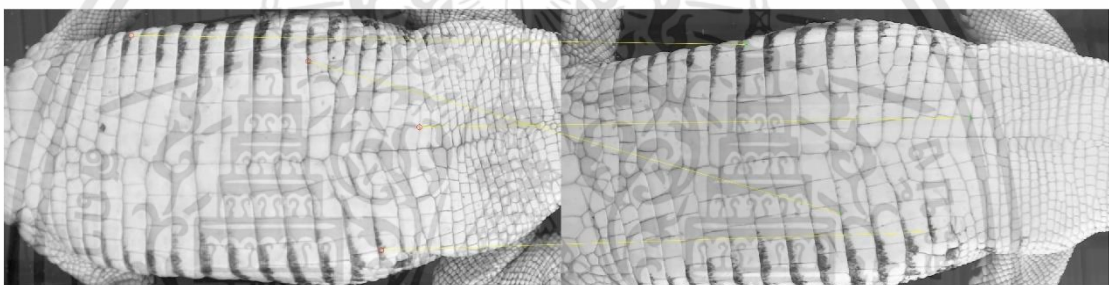


รูปที่ 4.22 การหา key point ภาพ Grayscale กระเซ่ D1/D2 ของอัลกอริทึม BRISK

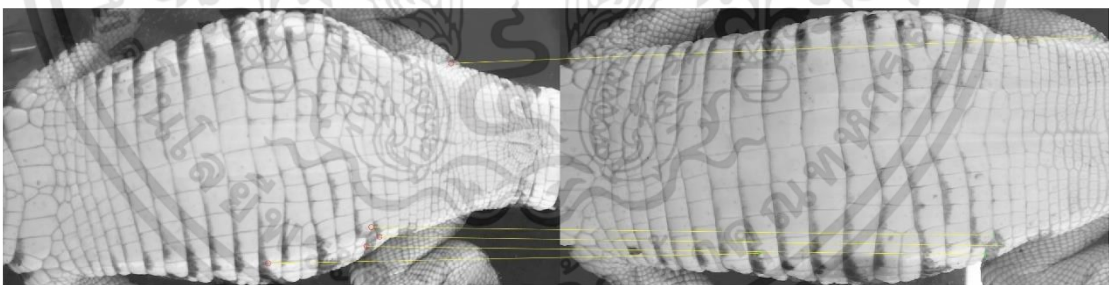
จากนั้นนำอัลกอริทึม BRISK มาใช้ในการเปรียบเทียบภาพลายหน้าห้องจระเข้เพื่อแสดงภาพที่มีการ Matching แสดงดังรูปที่ 4.23 รูปที่ 4.24 และรูปที่ 4.25



รูปที่ 4.23 การหา Matching ภาพ Grayscale จระเข้ E2/E3 ของอัลกอริทึม BRISK



รูปที่ 4.24 การหา Matching ภาพ Grayscale จระเข้ A2/A3 ของอัลกอริทึม BRISK

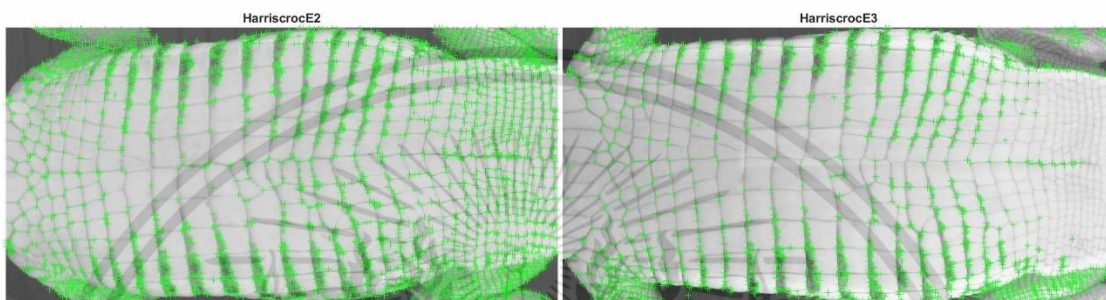


รูปที่ 4.25 การหา Matching ภาพ Grayscale จระเข้ D1/D2 ของอัลกอริทึม BRISK

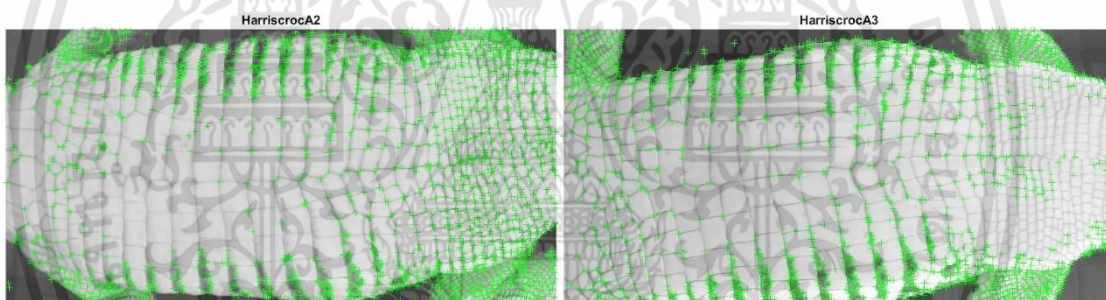
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.4 Harris

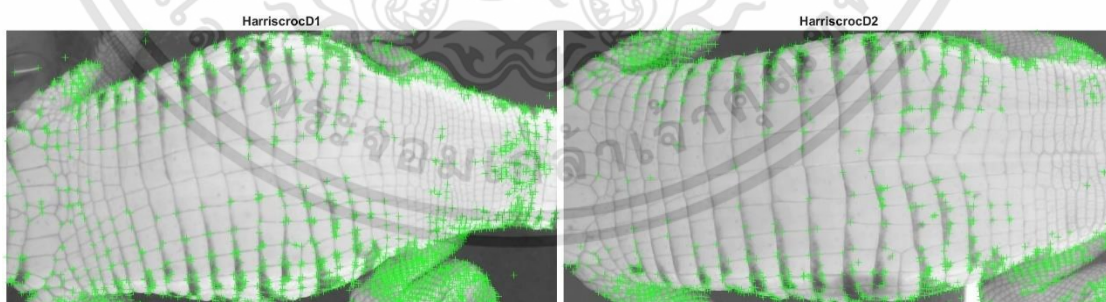
การหา key point ของภาพใช้อัลกอริทึม Harris โดย key point ที่ได้สามารถหาได้สูงสุด 2676 key point โดยจัดให้ภาพให้อยู่ในบริเวณหน้าต่างของจระเข้ แสดงดังรูปที่ 4.26 รูปที่ 4.27 และรูปที่ 4.28



รูปที่ 4.26 การหา key point ภาพ Grayscale จระเข้ E2/E3 ของอัลกอริทึม Harris



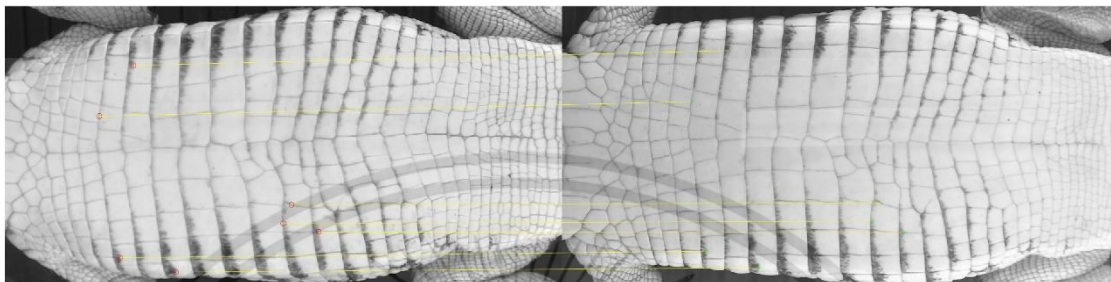
รูปที่ 4.27 การหา key point ภาพ Grayscale จระเข้ A2/A3 ของอัลกอริทึม Harris



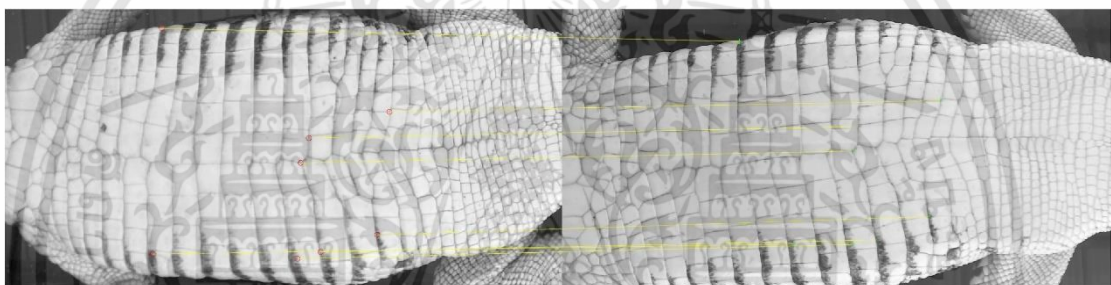
รูปที่ 4.28 การหา key point ภาพ Grayscale จระเข้ D1/D2 ของอัลกอริทึม Harris

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

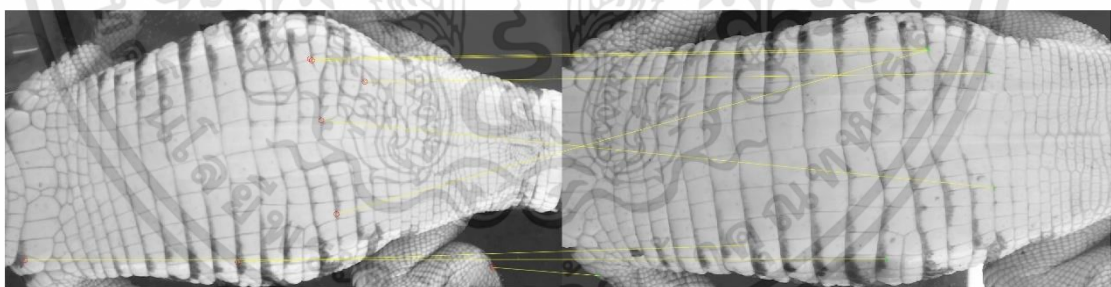
จากนั้นนำอัลกอริทึม Harris มาใช้ในการเปรียบเทียบภาพลายหน้าท้องจระเข้เพื่อแสดงภาพที่มีการ Matching แสดงดังรูปที่ 4.29 รูปที่ 4.30 และรูปที่ 4.31



รูปที่ 4.29 การหา Matching ภาพ Grayscale จระเข้ E2/E3 ของอัลกอริทึม Harris



รูปที่ 4.30 การหา Matching ภาพ Grayscale จระเข้ A2/A3 ของอัลกอริทึม Harris

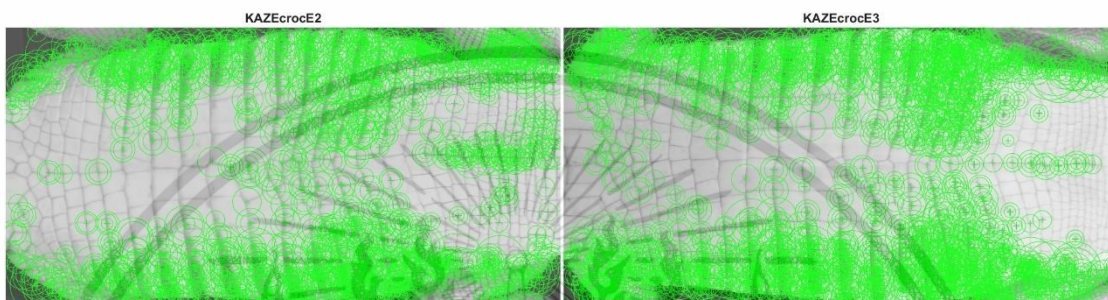


รูปที่ 4.31 การหา Matching ภาพ Grayscale จระเข้ D1/D2 ของอัลกอริทึม Harris

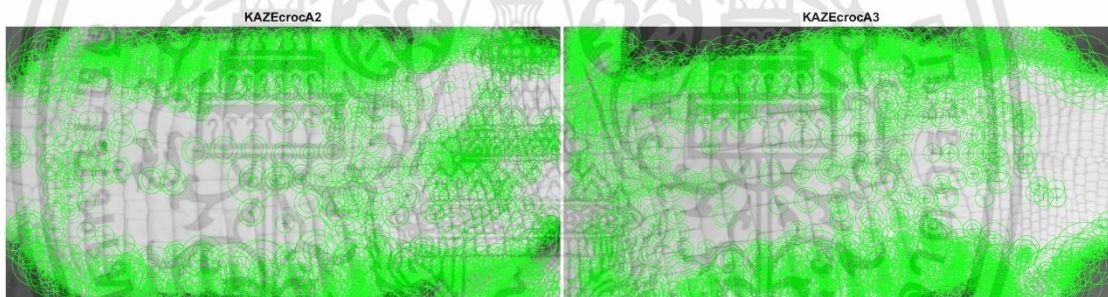
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.5 KAZE

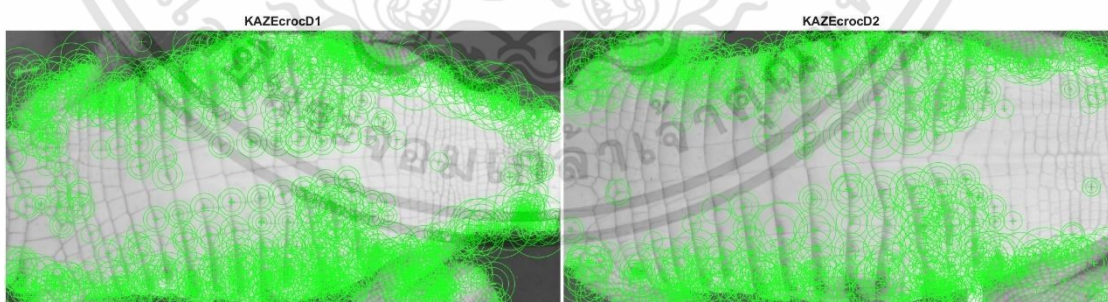
การหา key point ของภาพใช้อัลกอริทึม KAZE โดย key point ที่ได้สามารถหาได้สูงสุด 2240 key point โดยจัดให้ภาพให้อยู่ในบริเวณหน้าต่างของจระเข้ แสดงดังรูปที่ 4.32 รูปที่ 4.35 และรูปที่ 4.34



รูปที่ 4.32 การหา key point ภาพ Grayscale จระเข้ E2/E3 ของอัลกอริทึม KAZE



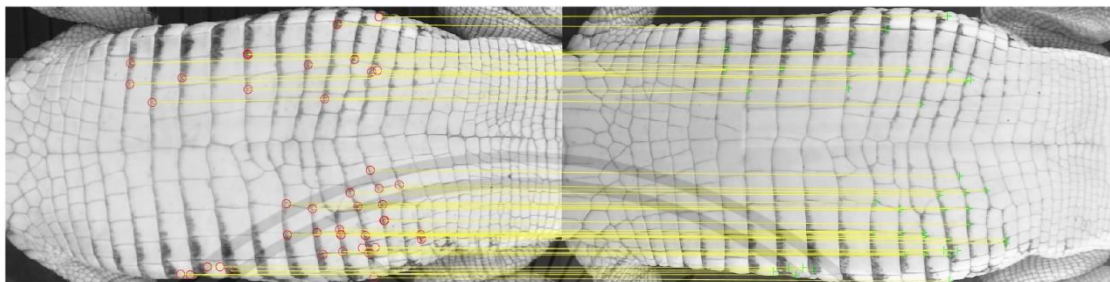
รูปที่ 4.33 การหา key point ภาพ Grayscale จระเข้ A2/A3 ของอัลกอริทึม KAZE



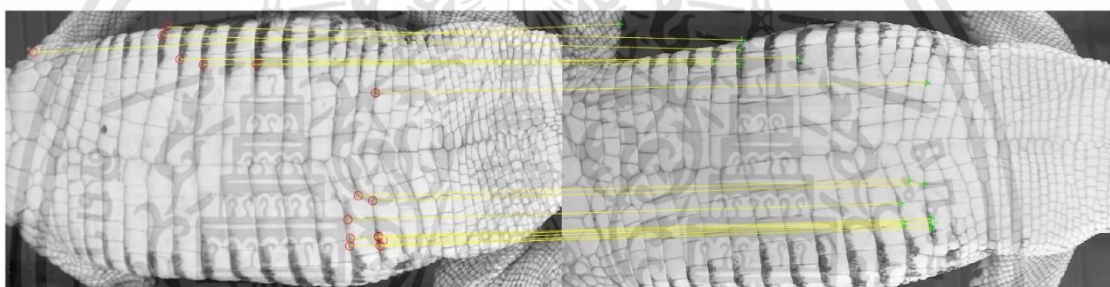
รูปที่ 4.34 การหา key point ภาพ Grayscale จระเข้ D1/D2 ของอัลกอริทึม KAZE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

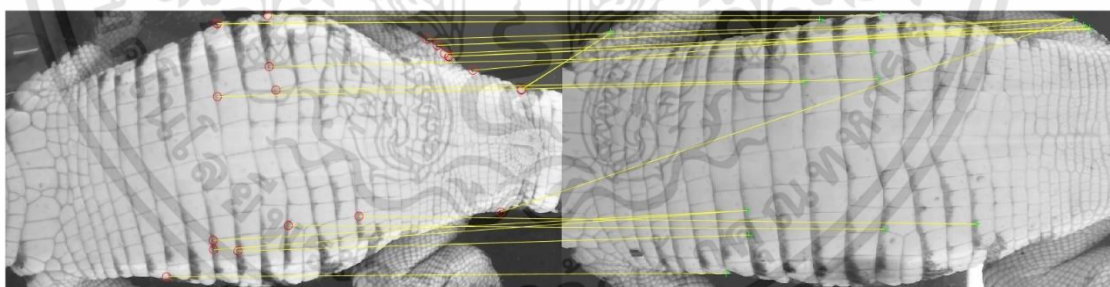
จากนั้นนำอัลกอริทึม KAZE มาใช้ในการเปรียบเทียบภาพลายหน้าท้องจระเข้เพื่อแสดงภาพที่มีการ Matching แสดงดังรูปที่ 4.35 รูปที่ 4.36 และรูปที่ 4.37



รูปที่ 4.35 การหา Matching ภาพ Grayscale จระเข้ E2/E3 ของอัลกอริทึม KAZE



รูปที่ 4.36 การหา Matching ภาพ Grayscale จระเข้ A2/A3 ของอัลกอริทึม KAZE

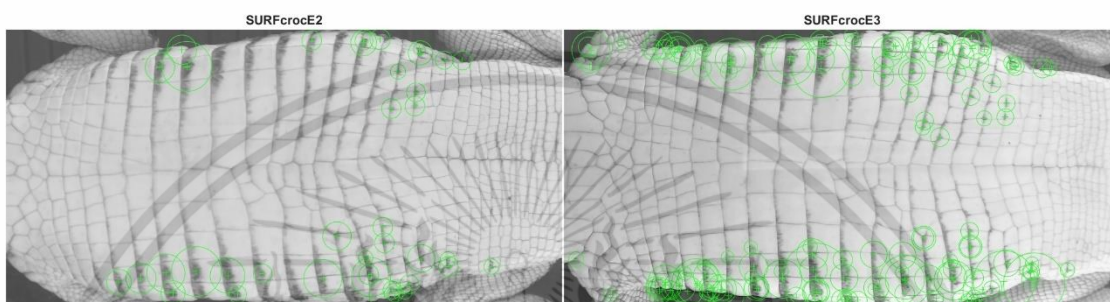


รูปที่ 4.37 การหา Matching ภาพ Grayscale จระเข้ D1/D2 ของอัลกอริทึม KAZE

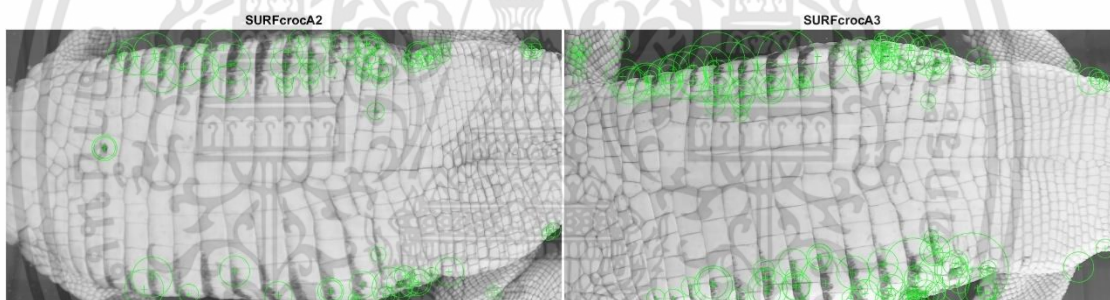
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.6 SURF

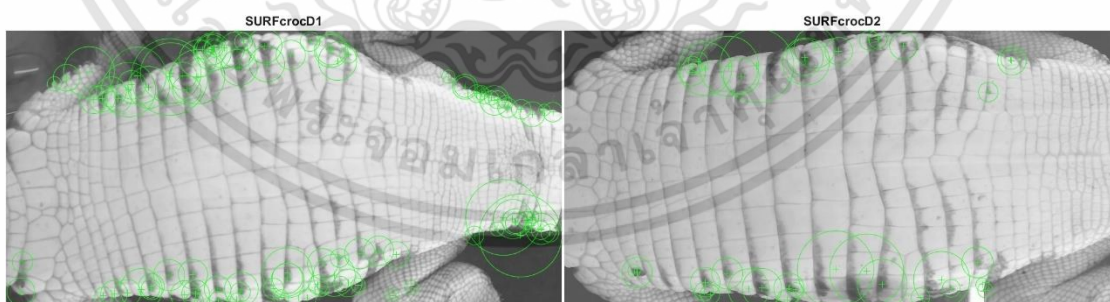
การหา key point ของภาพใช้อัลกอริทึม SURF โดย key point ที่ได้สามารถหาได้สูงสุด 142 key point โดยจัดให้ภาพให้อยู่ในบริเวณหน้าต่างของจระเข้ แสดงดังรูปที่ 4.38 รูปที่ 4.39 และรูปที่ 4.40



รูปที่ 4.38 การหา key point ภาพ Grayscale จระเข้ E2/E3 ของอัลกอริทึม SURF



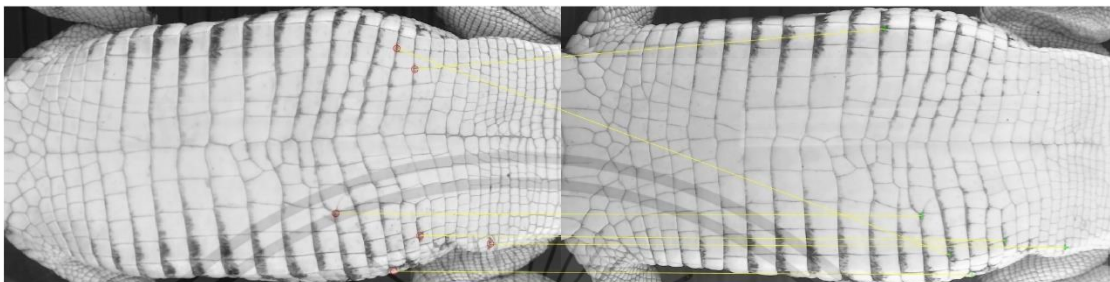
รูปที่ 4.39 การหา key point ภาพ Grayscale จระเข้ A2/A3 ของอัลกอริทึม SURF



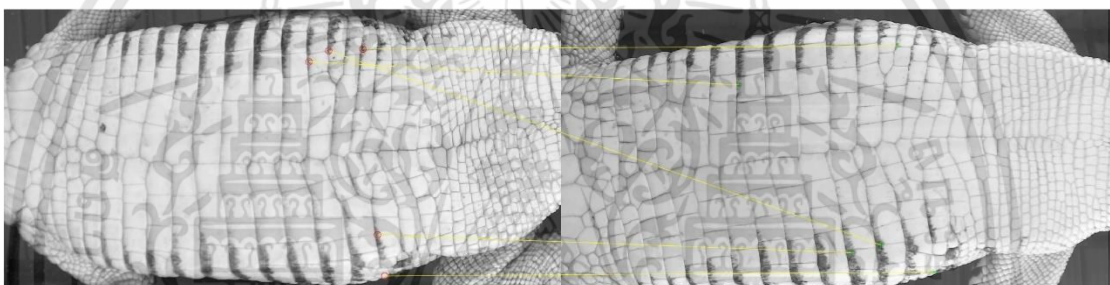
รูปที่ 4.40 การหา key point ภาพ Grayscale จระเข้ D1/D2 ของอัลกอริทึม SURF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

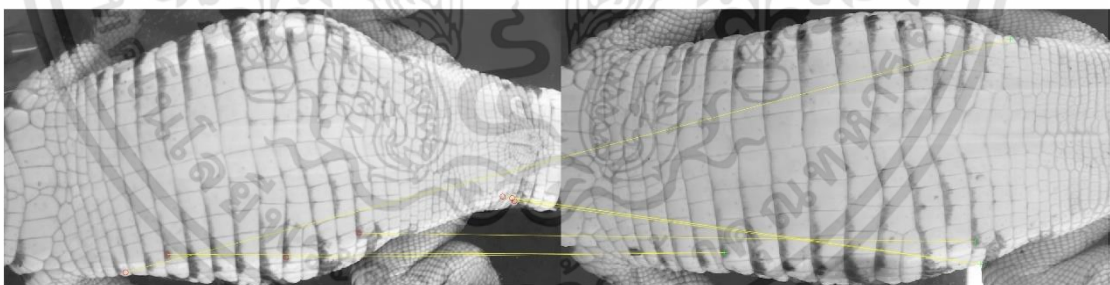
จากนั้นนำอัลกอริทึม SURF มาใช้ในการเปรียบเทียบภาพหลายหน้าทั้งองระเซ้เพื่อแสดงภาพที่มีการ Matching แสดงดังรูปที่ 4.41 รูปที่ 4.42 และรูปที่ 4.43



รูปที่ 4.41 การหา Matching ภาพ Grayscale จระเซ้ E2/E3 ของอัลกอริทึม SURF



รูปที่ 4.42 การหา Matching ภาพ Grayscale จระเซ้ A2/A3 ของอัลกอริทึม SURF

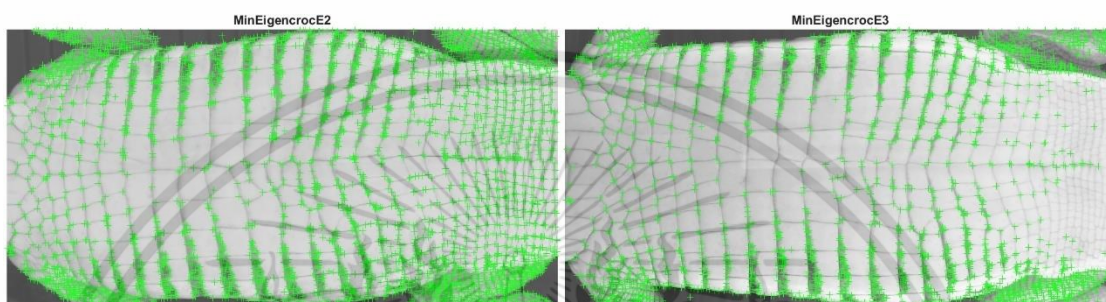


รูปที่ 4.43 การหา Matching ภาพ Grayscale จระเซ้ D1/D2 ของอัลกอริทึม SURF

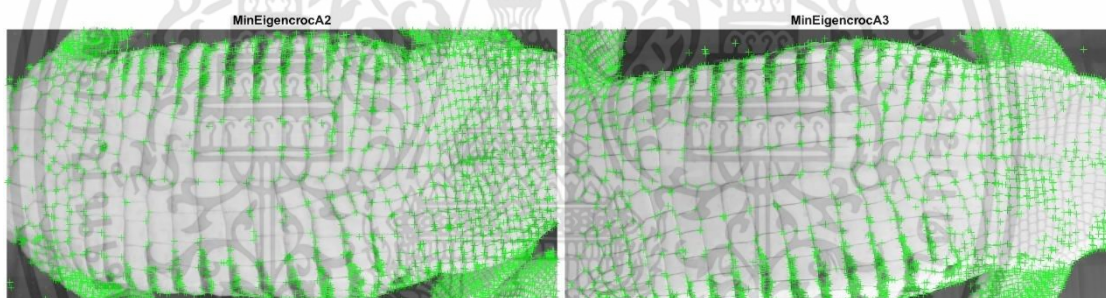
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.7 MinEigen

การหา key point ของภาพใช้อัลกอริทึม MinEigen โดย key point ที่ได้สามารถหาได้สูงสุด 3222 key point โดยจัดให้ภาพให้อยู่ในบริเวณหน้าต่างของจระเข้ แสดงดังรูปที่ 4.44 รูปที่ 4.45 และรูปที่ 4.46



รูปที่ 4.44 การหา key point ภาพ Grayscale จระเข้ E2/E3 ของอัลกอริทึม MinEigen



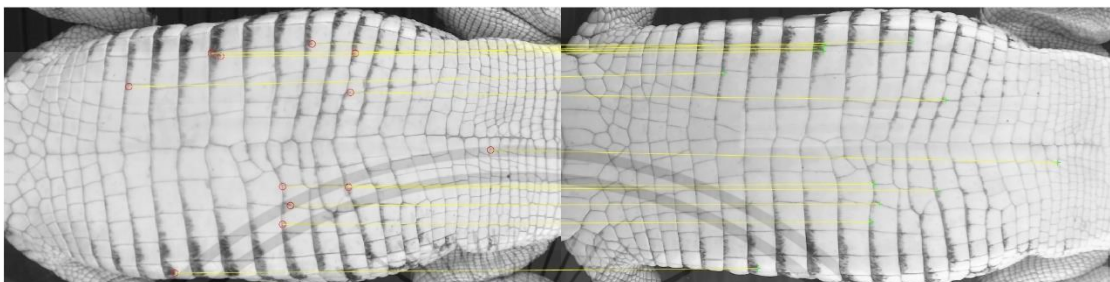
รูปที่ 4.45 การหา key point ภาพ Grayscale จระเข้ A2/A3 ของอัลกอริทึม MinEigen



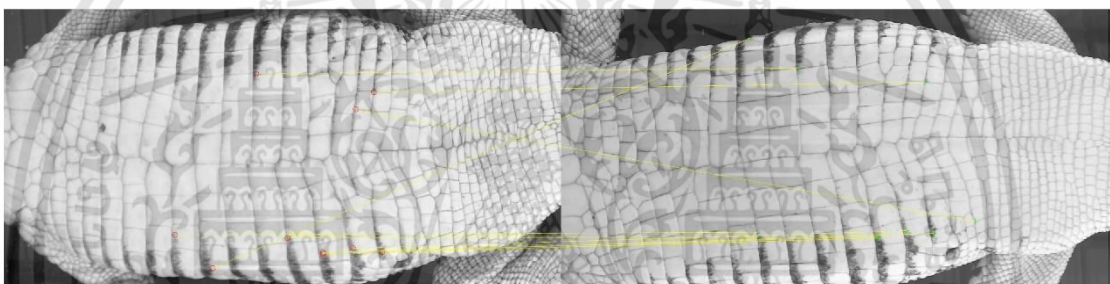
รูปที่ 4.46 การหา key point ภาพ Grayscale จระเข้ D1/D2 ของอัลกอริทึม MinEigen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

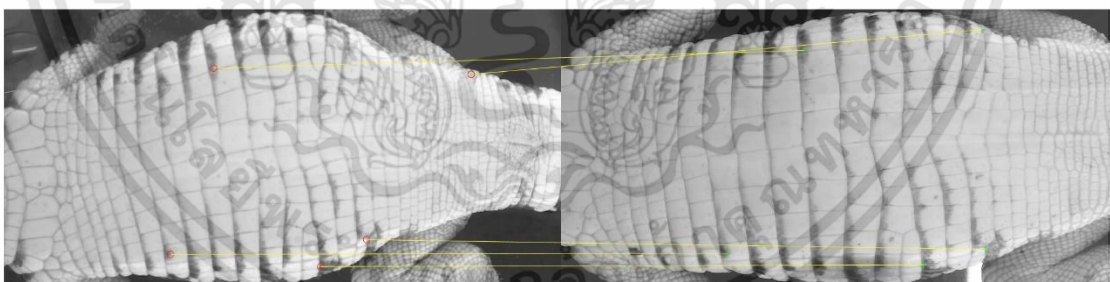
จากนั้นนำอัลกอริทึม MinEigen มาใช้ในการเปรียบเทียบภาพลายหน้าท้องจระเข้ เพื่อแสดงภาพที่มีการ Matching แสดงดังรูปที่ 4.47 รูปที่ 4.48 และรูปที่ 4.49



รูปที่ 4.47 การหา key point ภาพ Grayscale จระเข้ A2/A3 ของอัลกอริทึม MinEigen



รูปที่ 4.48 การหา Matching ภาพ Grayscale จระเข้ A2/A3 ของอัลกอริทึม MinEigen

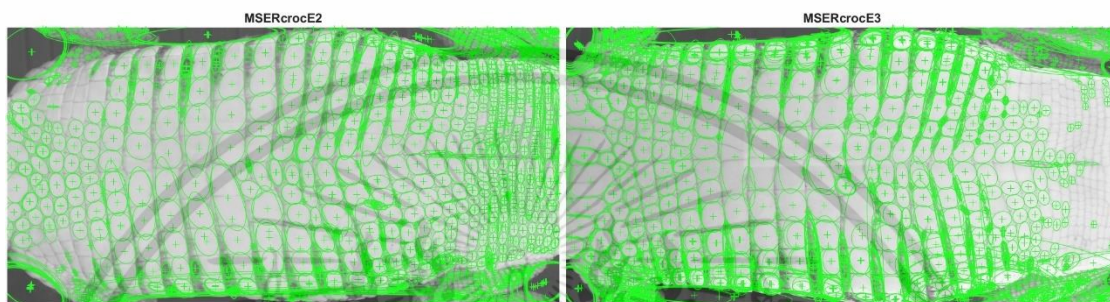


รูปที่ 4.49 การหา Matching ภาพ Grayscale จระเข้ D1/D2 ของอัลกอริทึม MinEigen

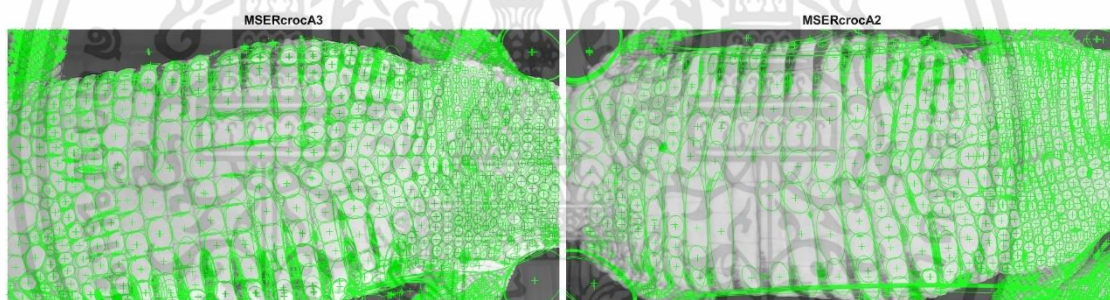
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.8 MSER

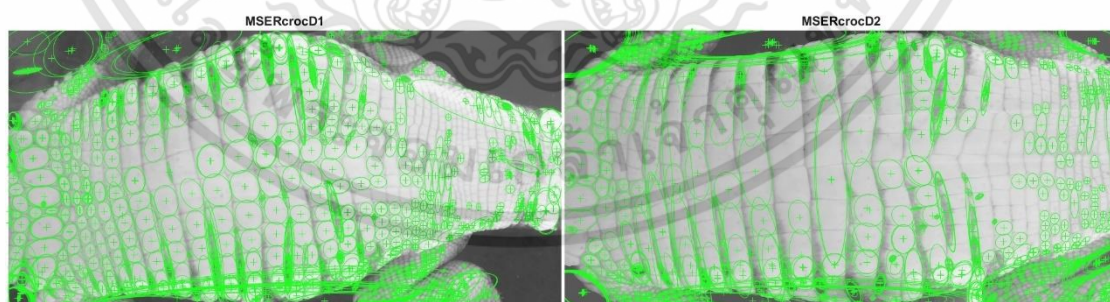
การหา key point ของภาพใช้อัลกอริทึม MSER โดย key point ที่ได้สามารถหาได้สูงสุด 2812 key point โดยจัดให้ภาพให้อยู่ในบริเวณหน้าต่างของจระเข้ แสดงดังรูปที่ 4.50 รูปที่ 4.51 และรูปที่ 4.52



รูปที่ 4.50 การหา key point ภาพ Grayscale จระเข้ E2/E3 ของอัลกอริทึม MSER



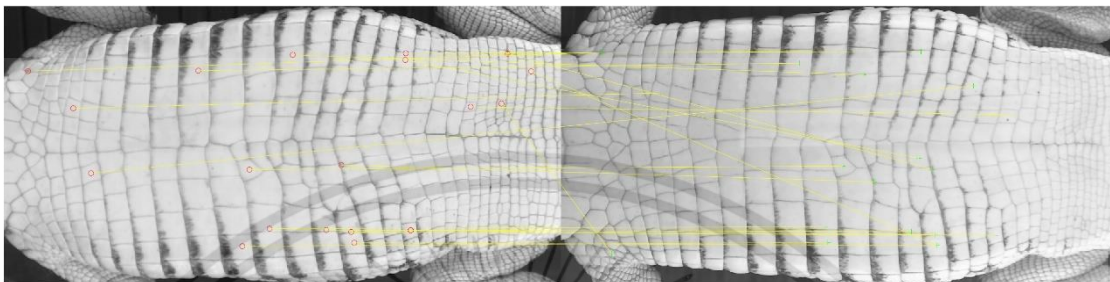
รูปที่ 4.51 การหา key point ภาพ Grayscale จระเข้ A2/A3 ของอัลกอริทึม MSER



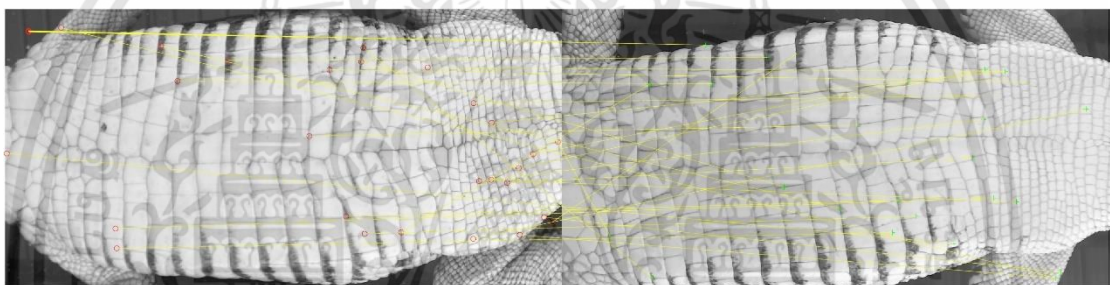
รูปที่ 4.52 การหา key point ภาพ Grayscale จระเข้ D1/D2 ของอัลกอริทึม MSER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

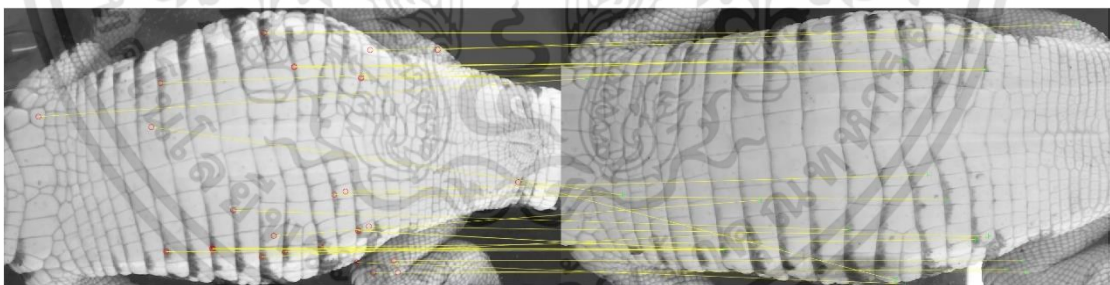
จากนั้นนำอัลกอริทึม MSER มาใช้ในการเปรียบเทียบภาพหลายหน้าหน้าท้องจระเข้เพื่อแสดงภาพที่มีการ Matching แสดงดังรูปที่ 4.53 รูปที่ 4.54 และรูปที่ 4.55



รูปที่ 4.53 การหา Matching ภาพ Grayscale จระเข้ E2/E3 ของอัลกอริทึม MSER



รูปที่ 4.54 การหา Matching ภาพ Grayscale จระเข้ A2/A3 ของอัลกอริทึม MSER



รูปที่ 4.55 การหา Matching ภาพ Grayscale จระเข้ D1/D2 ของอัลกอริทึม MSER

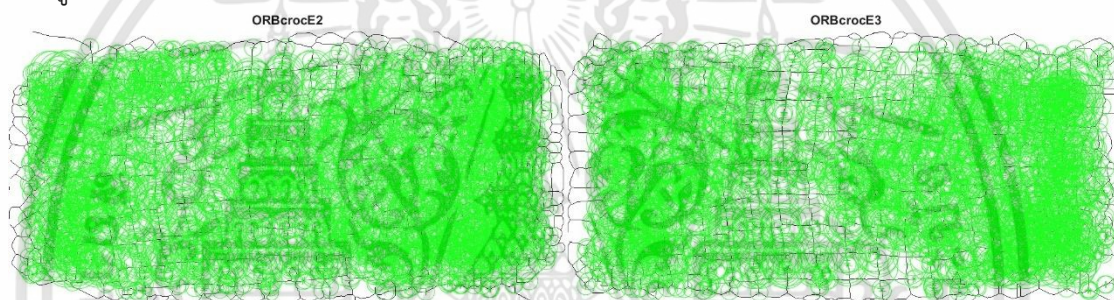
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 การทดสอบอัลกอริทึมในการวิเคราะห์รูปภาพ โดยพิจารณาจาก Thin Image

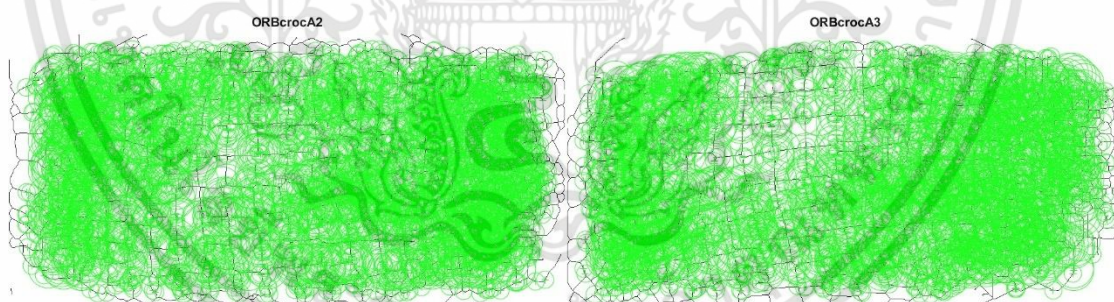
ผู้จัดทำได้นำภาพ Thin ลายหน้าห้องจระเข้ เพื่อให้เส้นลายหน้าห้องจระเข้มีขนาดความกว้างเป็น 1 pixel จำนวน 3 ตัว โดยจระเข้ในช่วงวัย 3 เดือนแทนเป็น E2, A2 และ D1 และ ในช่วงวัย 6 เดือนแทนเป็น E3, A3 และ D2 เพื่อทดลองว่าอัลกอริทึมตัวใดมีการ Matching ได้ดีที่สุด

4.3.1 ORB

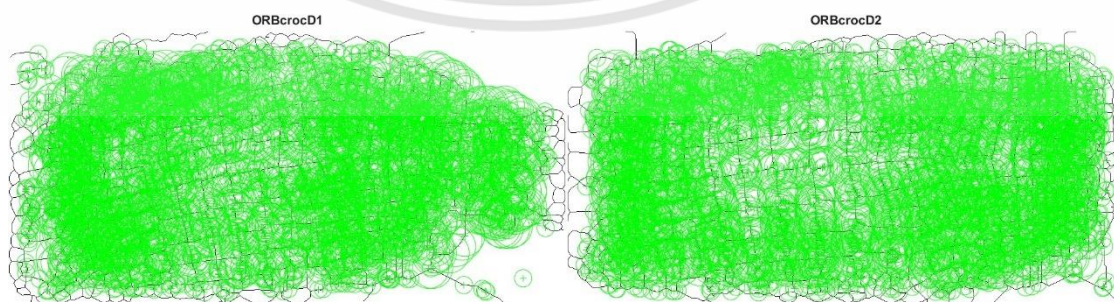
การทำ key point ของภาพใช้อัลกอริทึม ORB โดย key point ที่ได้สามารถหาได้สูงสุด 2655 key point โดยจัดให้ภาพให้อยู่ในบริเวณหน้าห้องของจระเข้ แสดงดังรูปที่ 4.56 รูปที่ 4.57 และรูปที่ 4.58



รูปที่ 4.56 การหา key point ภาพ Thin จระเข้ E2/E3 ของอัลกอริทึม ORB

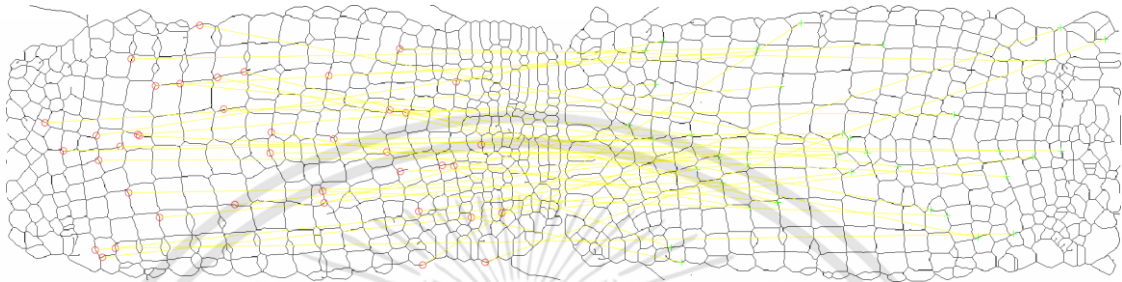


รูปที่ 4.57 การหา key point ภาพ Thin จระเข้ A2/A3 ของอัลกอริทึม ORB

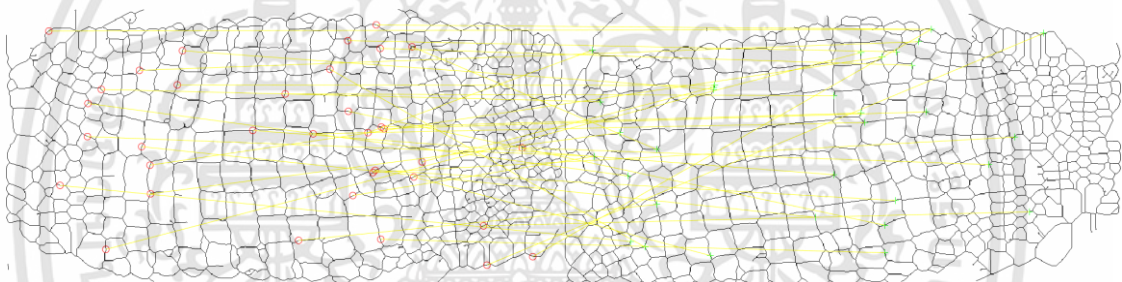


รูปที่ 4.58 การหา key point ภาพ Thin จระเข้ D1/D2 ของอัลกอริทึม ORB

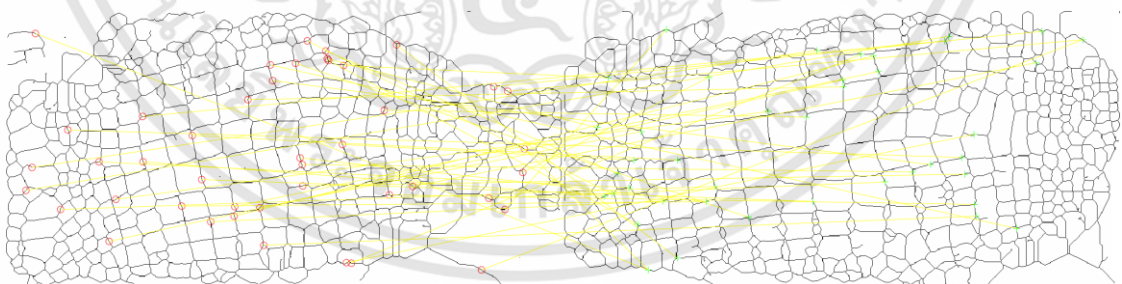
จากนั้นนำอัลกอริทึม ORB มาใช้ในการเปรียบเทียบภาพหลายหน้าห้องกระจกเพื่อแสดงภาพที่มีการ Matching แสดงดังรูปที่ 4.59 รูปที่ 4.60 และรูปที่ 4.61



รูปที่ 4.59 การหา Matching ภาพ Thin กระจก E2/E3 ของอัลกอริทึม ORB



รูปที่ 4.60 การหา Matching ภาพ Thin กระจก A2/A3 ของอัลกอริทึม ORB

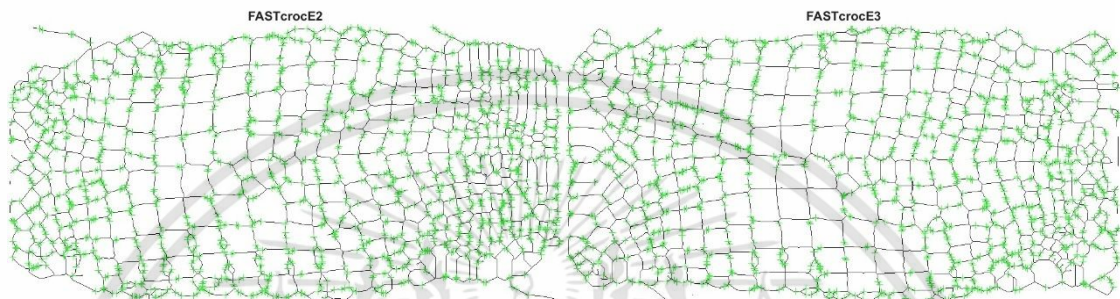


รูปที่ 4.61 การหา Matching ภาพ Thin กระจก D1/D2 ของอัลกอริทึม ORB

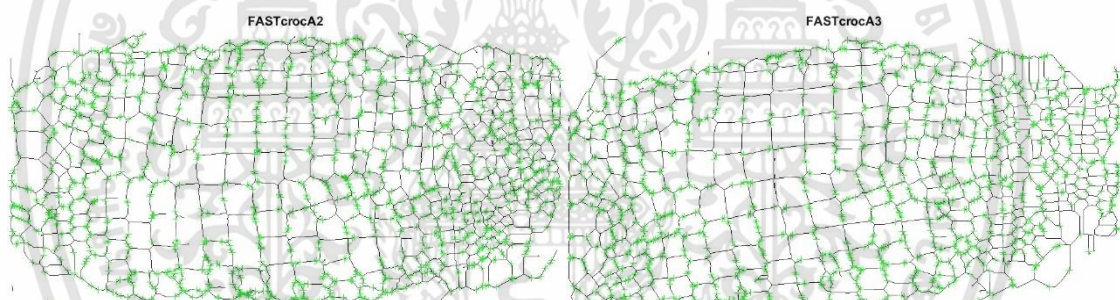
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 FAST

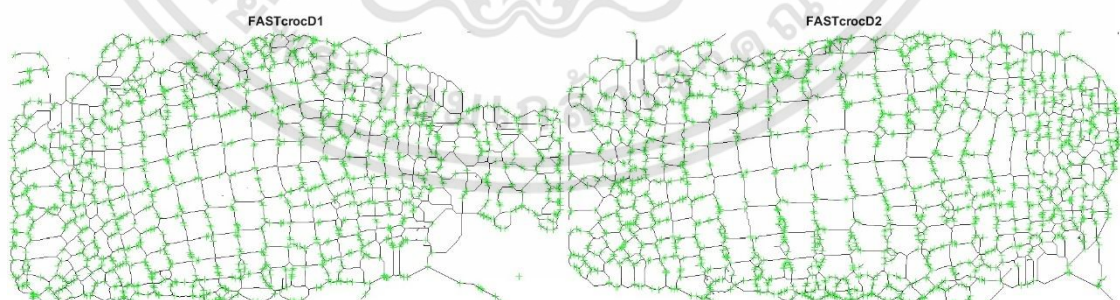
การหา key point ของภาพใช้อัลกอริทึม FAST โดย key point ที่ได้สามารถหาได้สูงสุด 1442 key point โดยจัดให้ภาพให้อยู่ในบริเวณหน้าต่างของจระเข้ แสดงดังรูปที่ 4.62 รูปที่ 4.63 และรูปที่ 4.64



รูปที่ 4.62 การหา key point ภาพ Thin จระเข้ E2/E3 ของอัลกอริทึม FAST

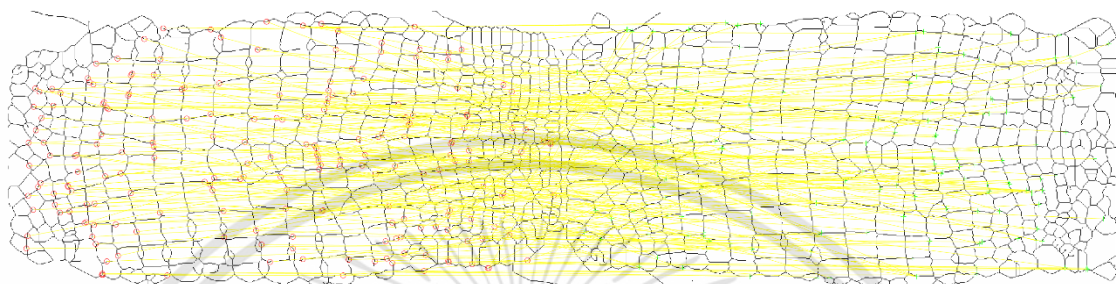


รูปที่ 4.63 การหา key point ภาพ Thin จระเข้ A2/A3 ของอัลกอริทึม FAST



รูปที่ 4.64 การหา key point ภาพ Thin จระเข้ D1/D2 ของอัลกอริทึม FAST

จากนั้นนำอัลกอริทึม FAST มาใช้ในการเปรียบเทียบภาพหลายหน้าหน้าท้องจระเข้เพื่อแสดงภาพที่มีการ Matching แสดงดังรูปที่ 4.65 รูปที่ 4.66 และรูปที่ 4.67



รูปที่ 4.65 การหา Matching ภาพ Thin จระเข้ E2/E3 ของอัลกอริทึม FAST



รูปที่ 4.66 การหา Matching ภาพ Thin จระเข้ A2/A3 ของอัลกอริทึม FAST

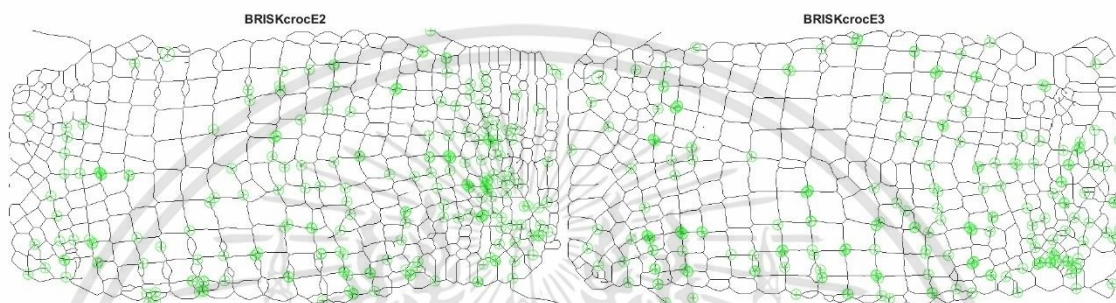


รูปที่ 4.67 การหา Matching ภาพ Thin จระเข้ D1/D2 ของอัลกอริทึม FAST

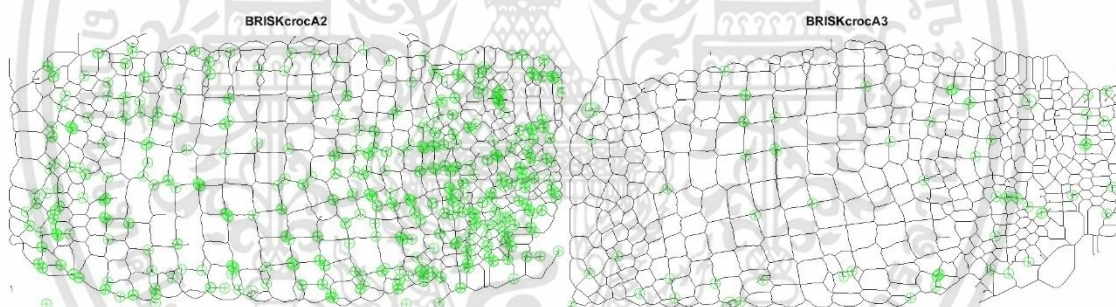
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.3 BRISK

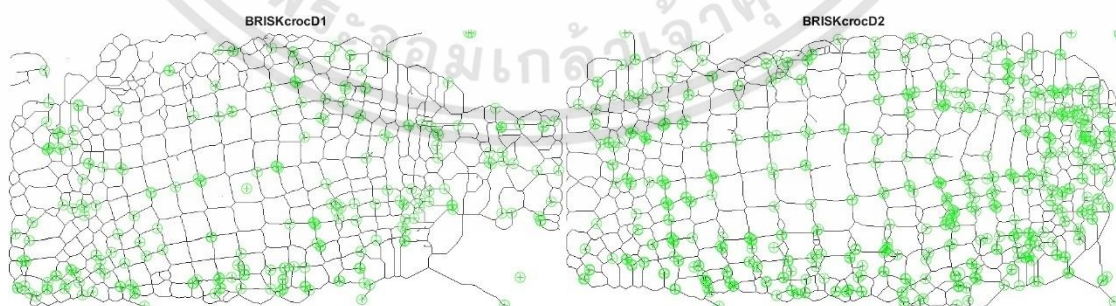
การทำ key point ของภาพใช้อัลกอริทึม BRISK โดย key point ที่ได้สามารถหาได้สูงสุด 519 key point โดยจัดให้ภาพให้อยู่ในบริเวณหน้าต่างของจระเข้ แสดงดังรูปที่ 4.68 รูปที่ 4.69 และรูปที่ 4.70



รูปที่ 4.68 การหา key point ภาพ Thin จระเข้ E2/E3 ของอัลกอริทึม BRISK

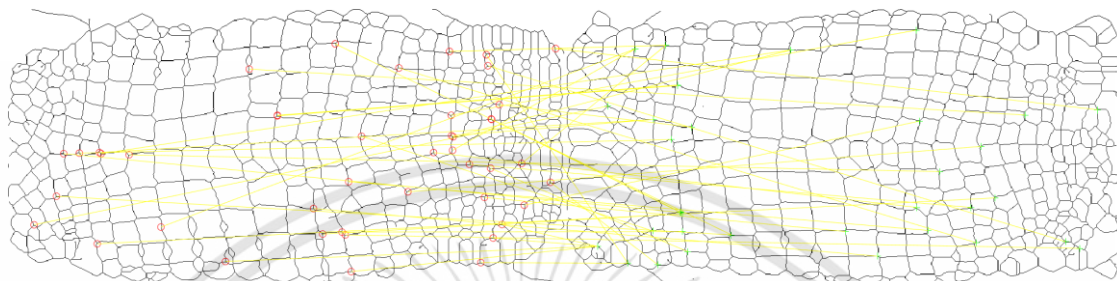


รูปที่ 4.69 การหา key point ภาพ Thin จระเข้ A2/A3 ของอัลกอริทึม BRISK



รูปที่ 4.70 การหา key point ภาพ Thin จระเข้ D1/D2 ของอัลกอริทึม BRISK

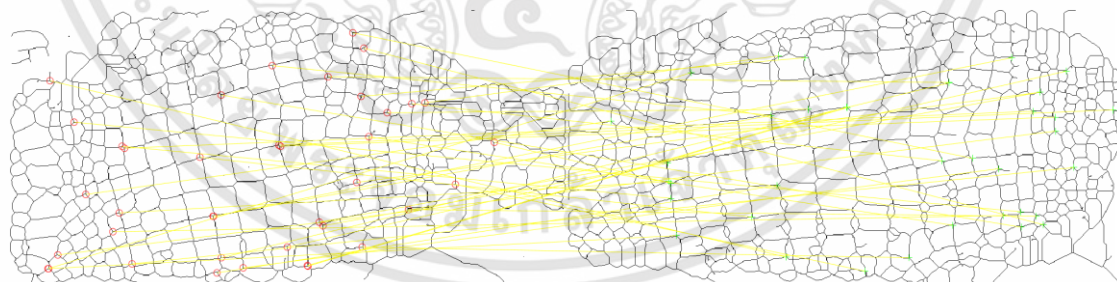
จากนั้นนำอัลกอริทึม BRISK มาใช้ในการเปรียบเทียบภาพลายหน้าห้องกระจกเพื่อแสดงภาพที่มีการ Matching แสดงดังรูปที่ 4.71 รูปที่ 4.72 และรูปที่ 4.73



รูปที่ 4.71 การหา Matching ภาพ Thin กระจก E2/E3 ของอัลกอริทึม BRISK



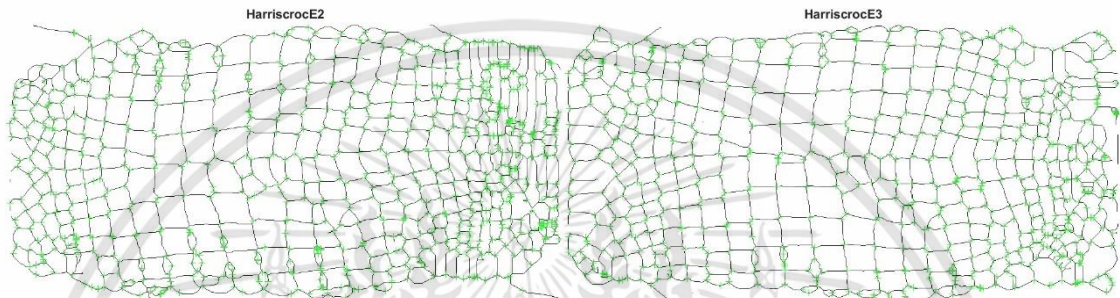
รูปที่ 4.72 การหา Matching ภาพ Thin กระจก A2/A3 ของอัลกอริทึม BRISK



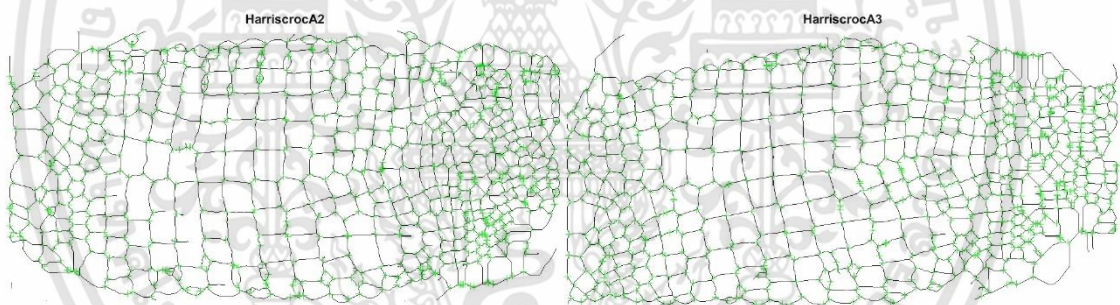
รูปที่ 4.73 การหา Matching ภาพ Thin กระจก D1/D2 ของอัลกอริทึม BRISK

4.3.4 Harris

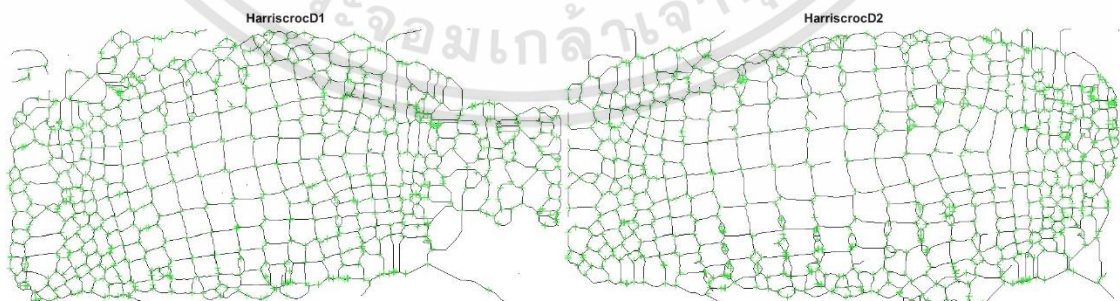
การหา key point ของภาพใช้อัลกอริทึม Harris โดย key point ที่ได้สามารถหาได้สูงสุด 784 key point โดยจัดให้ภาพให้อยู่ในบริเวณหน้าต่างของจระเข้ แสดงดังรูปที่ 4.74 รูปที่ 4.75 และรูปที่ 4.76



รูปที่ 4.74 การหา key point ภาพ Thin จระเข้ E2/E3 ของอัลกอริทึม Harris

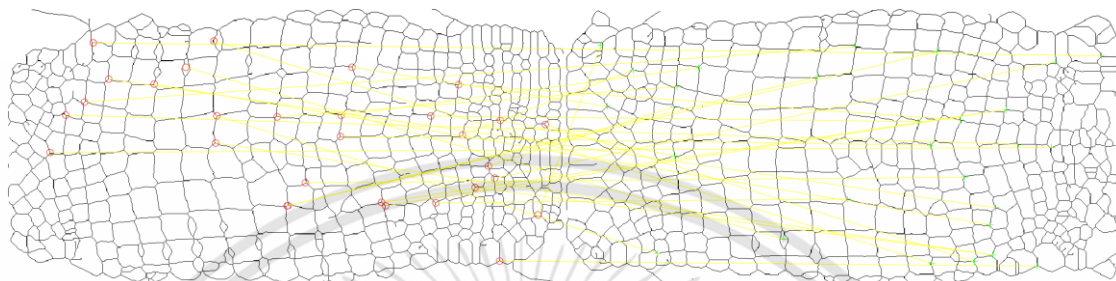


รูปที่ 4.75 การหา key point ภาพ Thin จระเข้ A2/A3 ของอัลกอริทึม Harris

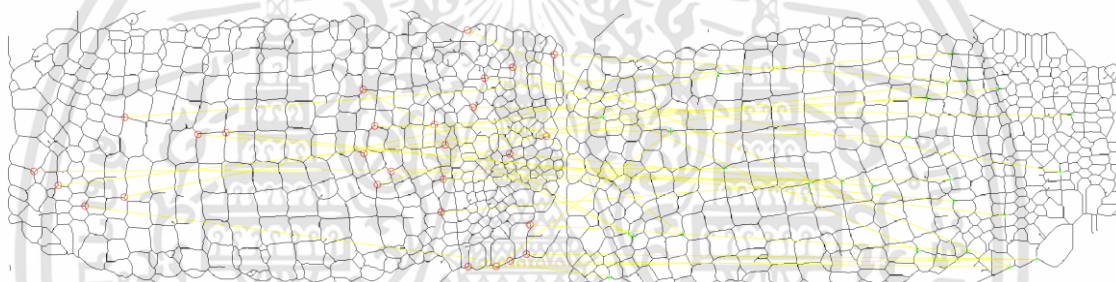


รูปที่ 4.76 การหา key point ภาพ Thin จระเข้ D1/D2 ของอัลกอริทึม Harris

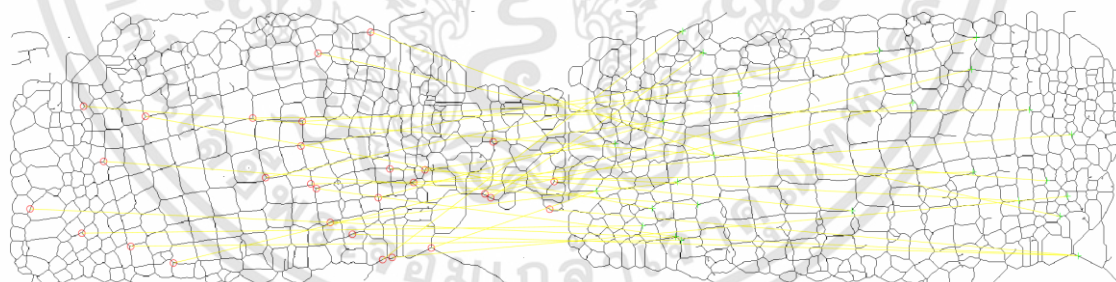
จากนั้นนำอัลกอริทึม Harris มาใช้ในการเปรียบเทียบภาพลายหน้าห้องจระเข้เพื่อแสดงภาพที่มีการ Matching แสดงดังรูปที่ 4.77 รูปที่ 4.78 และรูปที่ 4.79



รูปที่ 4.77 การหา Matching ภาพ Thin จระเข้ E2/E3 ของอัลกอริทึม Harris



รูปที่ 4.78 การหา Matching ภาพ Thin จระเข้ A2/A3 ของอัลกอริทึม Harris

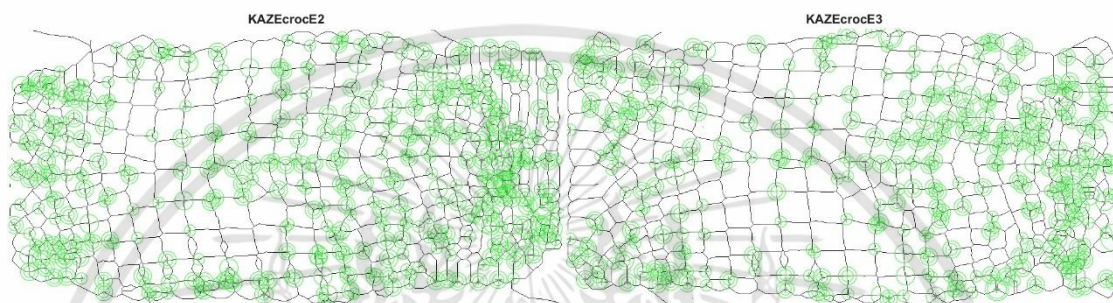


รูปที่ 4.79 การหา Matching ภาพ Thin จระเข้ D1/D2 ของอัลกอริทึม Harris

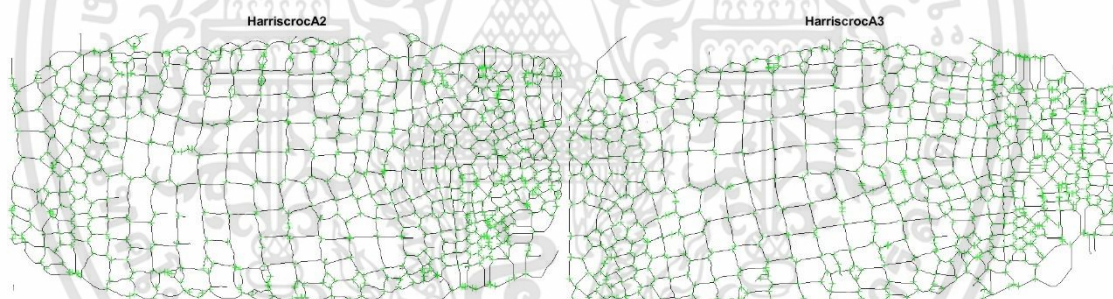
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.5 KAZE

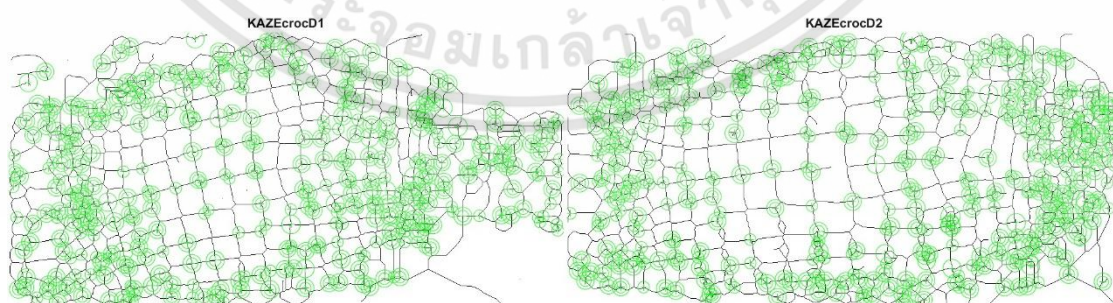
การหา key point ของภาพใช้อัลกอริทึม KAZE โดย key point ที่ได้สามารถหาได้สูงสุด 627 key point โดยจัดให้ภาพให้อยู่ในบริเวณหน้าต่างของจระเข้ แสดงดังรูปที่ 4.80 รูปที่ 4.81 และรูปที่ 4.82



รูปที่ 4.80 การหา key point ภาพ Thin จระเข้ E2/E3 ของอัลกอริทึม KAZE



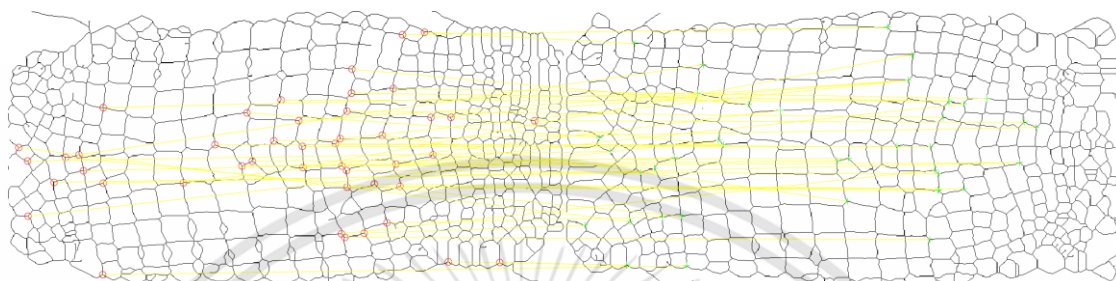
รูปที่ 4.81 การหา key point ภาพ Thin จระเข้ A2/A3 ของอัลกอริทึม KAZE



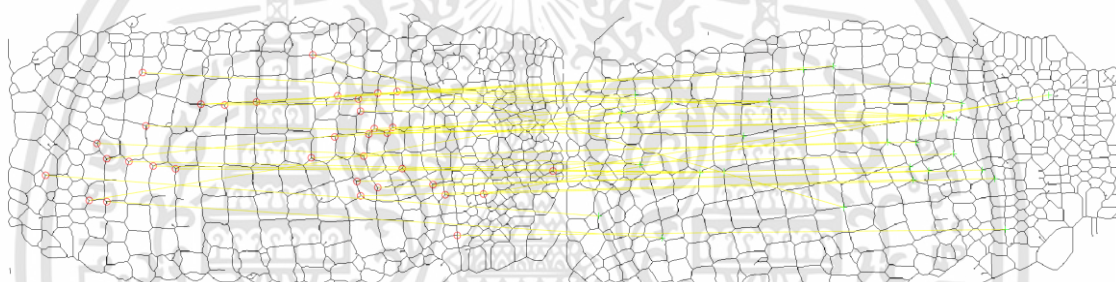
รูปที่ 4.82 การหา key point ภาพ Thin จระเข้ D1/D2 ของอัลกอริทึม KAZE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

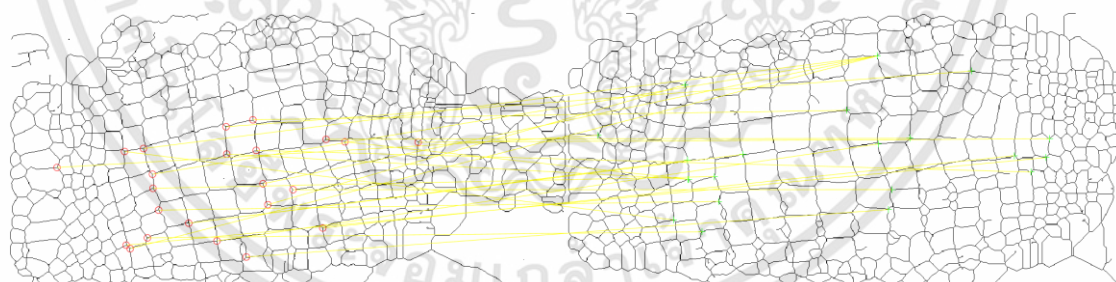
จากนั้นนำอัลกอริทึม KAZE มาใช้ในการเปรียบเทียบภาพลายหน้าห้องกระจกเพื่อแสดงภาพที่มีการ Matching แสดงดังรูปที่ 4.83 รูปที่ 4.84 และรูปที่ 4.85



รูปที่ 4.83 การหา Matching ภาพ Thin กระจก E2/E3 ของอัลกอริทึม KAZE



รูปที่ 4.84 การหา Matching ภาพ Thin กระจก A2/A3 ของอัลกอริทึม KAZE

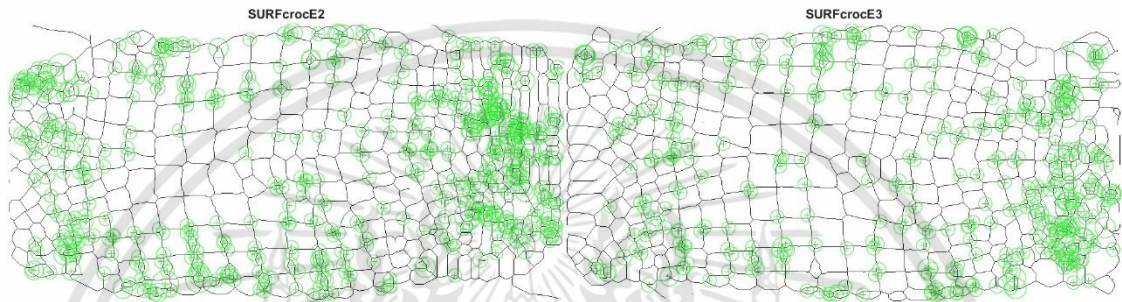


รูปที่ 4.85 การหา Matching ภาพ Thin กระจก D1/D2 ของอัลกอริทึม KAZE

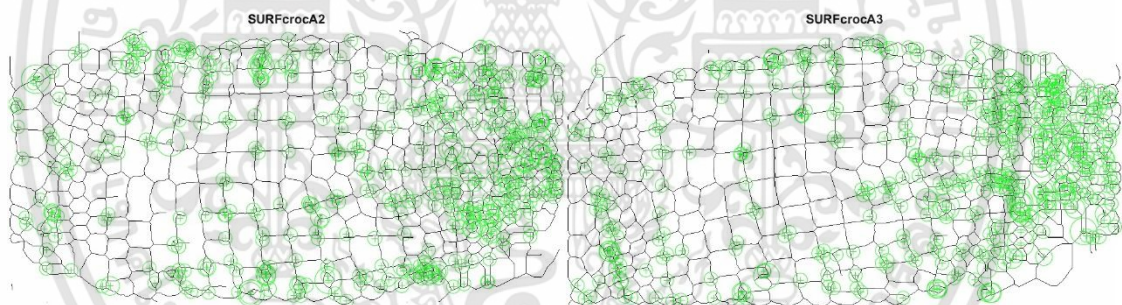
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.6 SURF

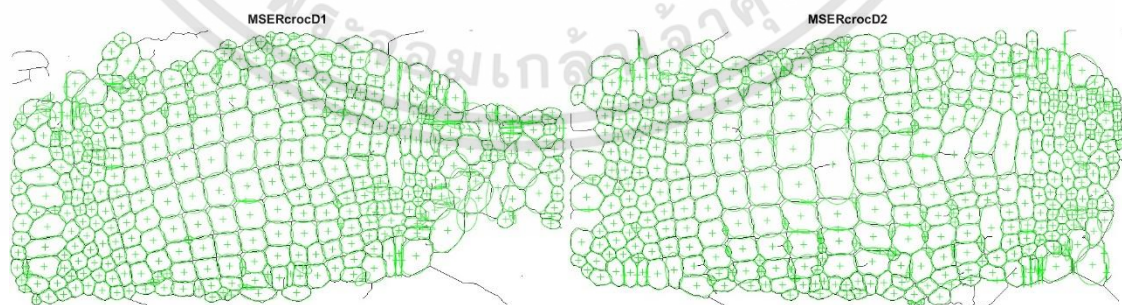
การหา key point ของภาพใช้อัลกอริทึม SURF โดย key point ที่ได้สามารถหาได้สูงสุด 502 key point โดยจัดให้ภาพให้อยู่ในบริเวณหน้าต่างของจระเข้ แสดงดังรูปที่ 4.86 รูปที่ 4.87 และรูปที่ 4.88



รูปที่ 4.86 การหา key point ภาพ Thin จระเข้ E2/E3 ของอัลกอริทึม SURF



รูปที่ 4.87 การหา key point ภาพ Thin จระเข้ A2/A3 ของอัลกอริทึม SURF

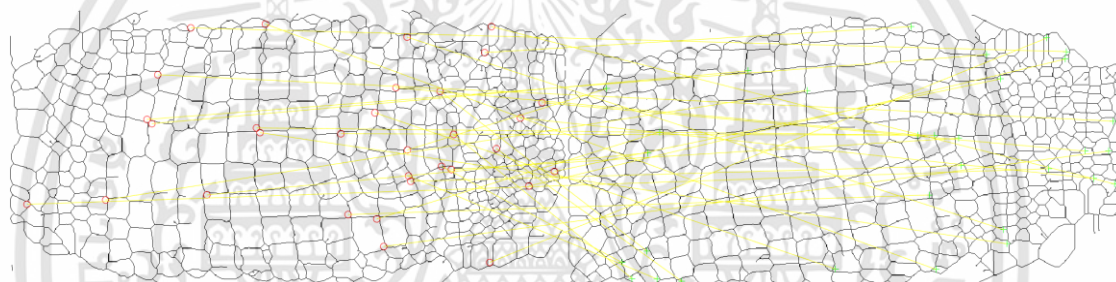


รูปที่ 4.88 การหา key point ภาพ Thin จระเข้ D1/D2 ของอัลกอริทึม SURF

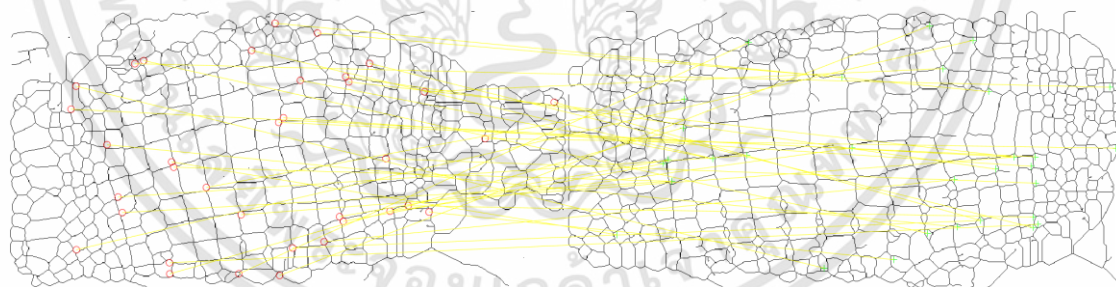
จากนั้นนำอัลกอริทึม SURF มาใช้ในการเปรียบเทียบภาพลายหน้าห้องกระจกเพื่อแสดงภาพที่มีการ Matching แสดงดังรูปที่ 4.89 รูปที่ 4.90 และรูปที่ 4.91



รูปที่ 4.89 การหา Matching ภาพ Thin กระจก E2/E3 ของอัลกอริทึม SURF



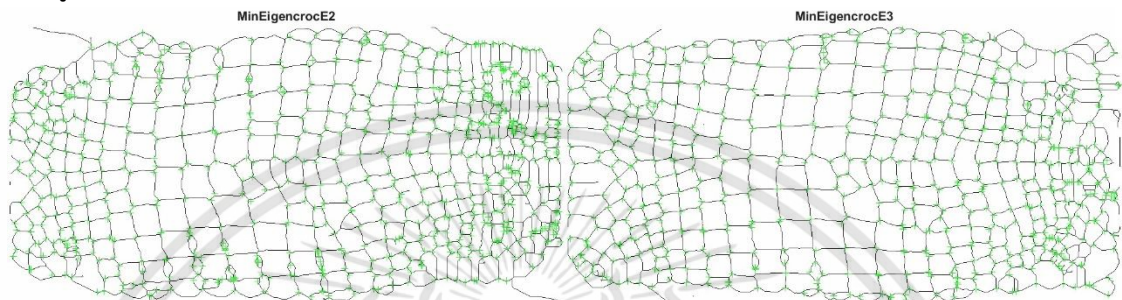
รูปที่ 4.90 การหา Matching ภาพ Thin กระจก A2/A3 ของอัลกอริทึม SURF



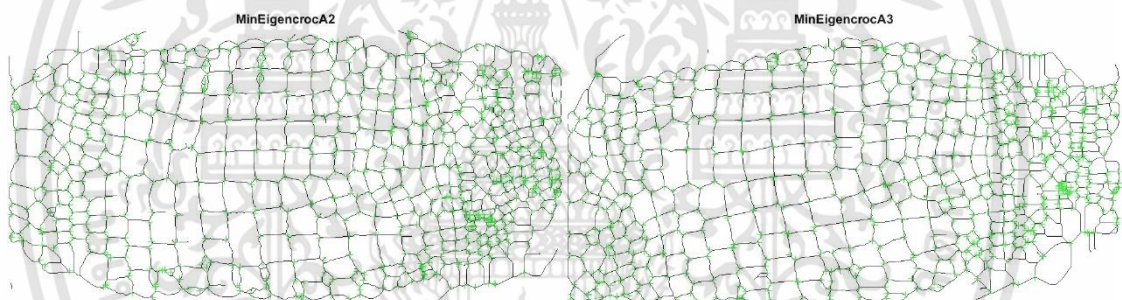
รูปที่ 4.91 การหา Matching ภาพ Thin กระจก D1/D2 ของอัลกอริทึม SURF

4.3.7 MinEigen

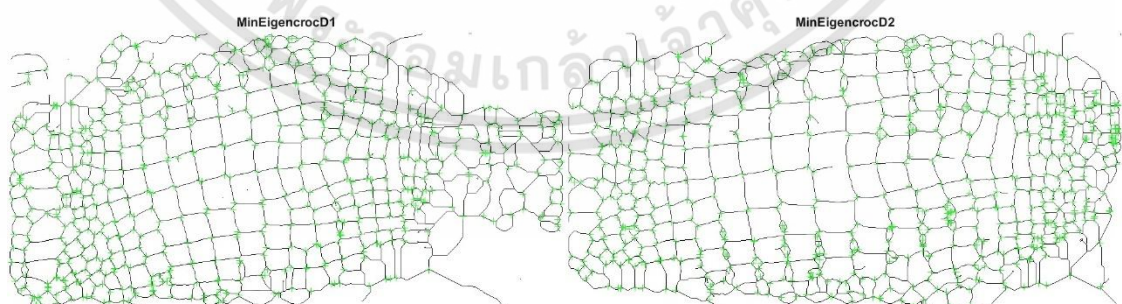
การหา key point ของภาพใช้อัลกอริทึม ORB โดย key point ที่ได้สามารถหาได้สูงสุด 751 key point โดยจัดให้ภาพให้อยู่ในบริเวณหน้าต่างของจระเข้ แสดงดังรูปที่ 4.92 รูปที่ 4.93 และรูปที่ 4.94



รูปที่ 4.92 การหา key point ภาพ Thin จระเข้ E2/E3 ของอัลกอริทึม MinEigen



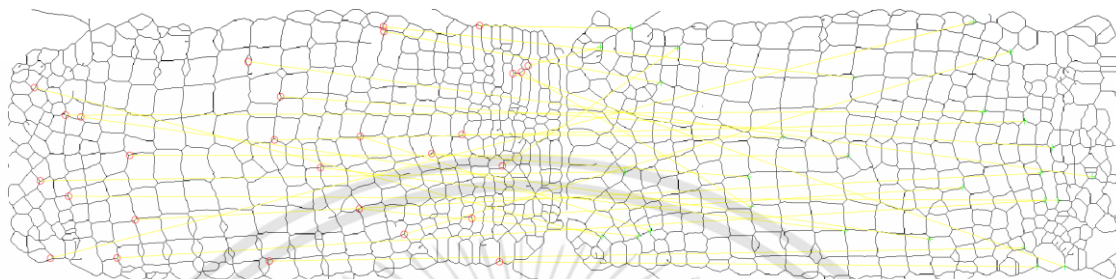
รูปที่ 4.93 การหา key point ภาพ Thin จระเข้ A2/A3 ของอัลกอริทึม MinEigen



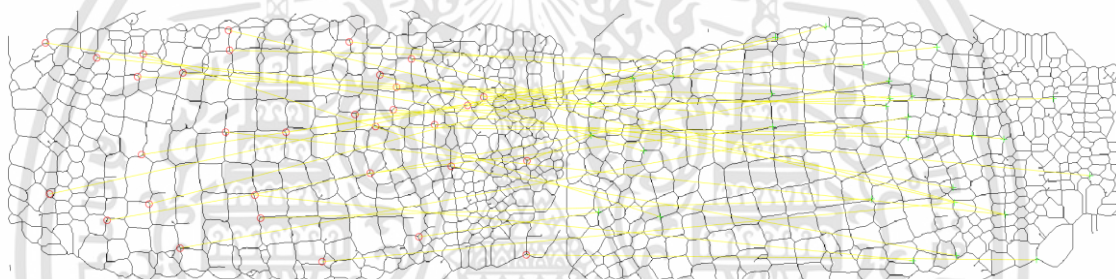
รูปที่ 4.94 การหา key point ภาพ Thin จระเข้ D1/D2 ของอัลกอริทึม MinEigen

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

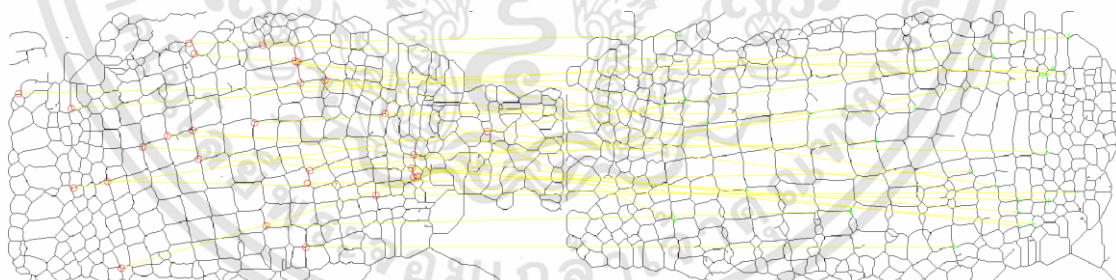
จากนั้นนำอัลกอริทึม SURF มาใช้ในการเปรียบเทียบภาพหลายหน้าห้องกระจกเพื่อแสดงภาพที่มีการ Matching แสดงดังรูปที่ 4.95 รูปที่ 4.96 และรูปที่ 4.97



รูปที่ 4.95 การหา Matching ภาพ Thin กระจก E2/E3 ของอัลกอริทึม MinEigen



รูปที่ 4.96 การหา Matching ภาพ Thin กระจก A2/A3 ของอัลกอริทึม MinEigen

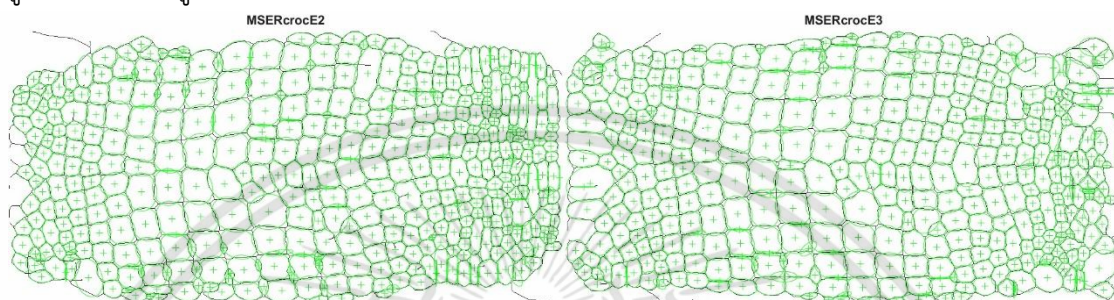


รูปที่ 4.97 การหา Matching ภาพ Thin กระจก D1/D2 ของอัลกอริทึม MinEigen

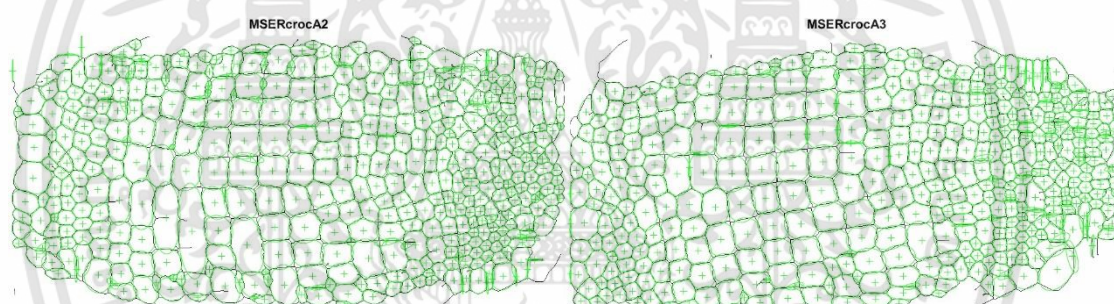
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.8 MSER

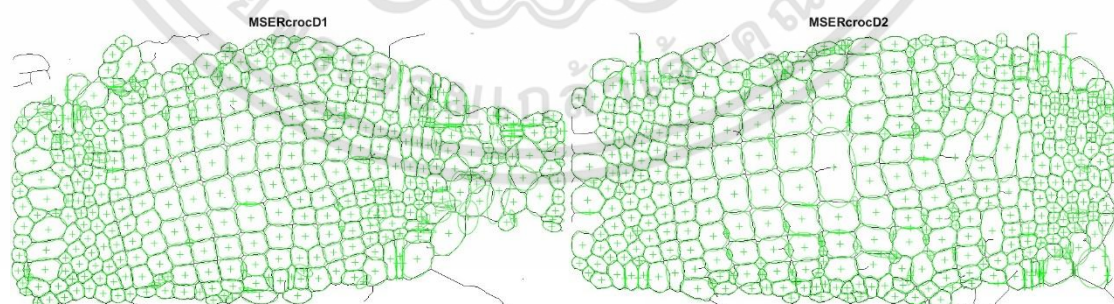
การทำ key point ของภาพใช้อัลกอริทึม MSER โดย key point ที่ได้สามารถหาได้สูงสุด 459 key point โดยจัดให้ภาพให้อยู่ในบริเวณหน้าต่างของจระเข้ แสดงดังรูปที่ 4.98 รูปที่ 4.99 และรูปที่ 4.100



รูปที่ 4.98 การหา key point ภาพ Thin จระเข้ E2/E3 ของอัลกอริทึม MSER



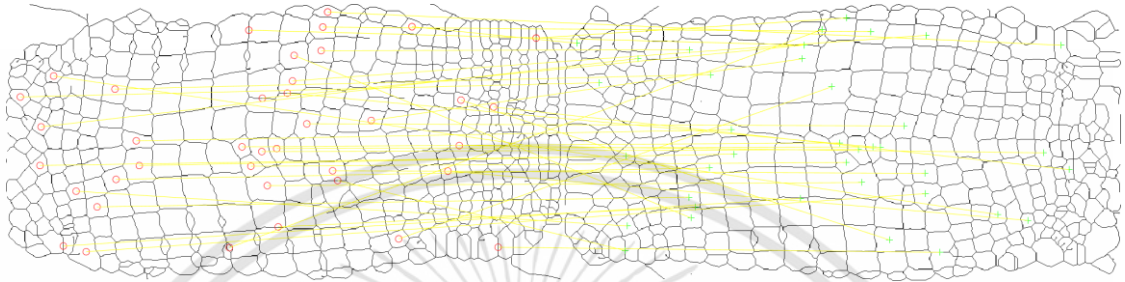
รูปที่ 4.99 การหา key point ภาพ Thin จระเข้ A2/A3 ของอัลกอริทึม MSER



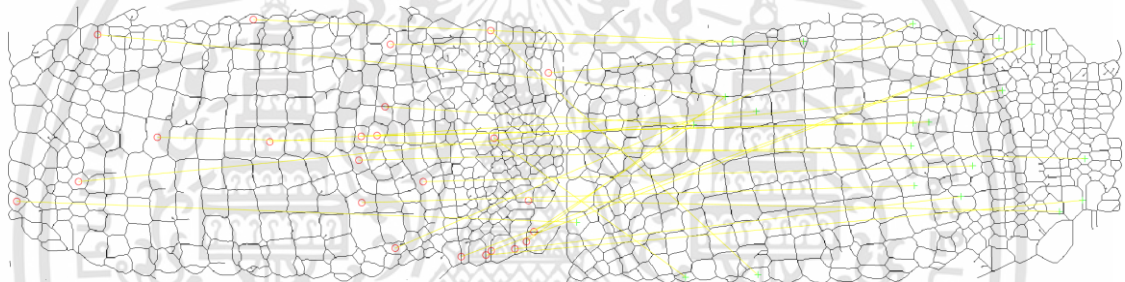
รูปที่ 4.100 การหา key point ภาพ Thin จระเข้ D1/D2 ของอัลกอริทึม MSER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

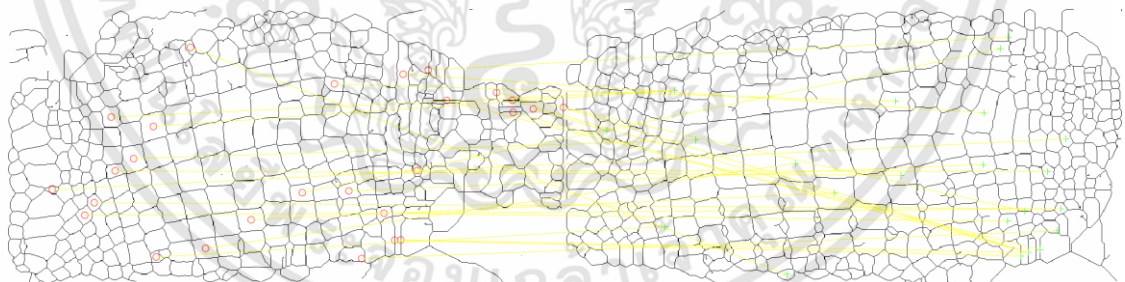
จากนั้นนำอัลกอริทึม SURF มาใช้ในการเปรียบเทียบภาพหลายหน้าห้องกระจกเพื่อแสดงภาพที่มีการ Matching แสดงดังรูปที่ 4.101 รูปที่ 4.102 และรูปที่ 4.103



รูปที่ 4.101 การหา Matching ภาพ Thin กระจก E2/E3 ของอัลกอริทึม MSER



รูปที่ 4.102 การหา Matching ภาพ Thin กระจก A2/A3 ของอัลกอริทึม MSER



รูปที่ 4.103 การหา Matching ภาพ Thin กระจก D1/D2 ของอัลกอริทึม MSER

4.4 การคำนวณหาเปอร์เซ็นต์ความถูกต้องการ Matching

ทดลองหาค่า key point และเปอร์เซ็นต์การ Matching ที่เหมาะสมแต่ละอัลกอริทึมของระยะเข้า 3 ตัว จากรูปภาพแบบ Grayscale ได้ผลตามตารางที่ 4.1 และตารางที่ 4.2

ตารางที่ 4.1 การหาค่า key point ของแต่ละอัลกอริทึมจากรูปภาพแบบ Grayscale

อัลกอริทึม	E2/E3		A2/A3		D1/D2	
	3 เดือน	6 เดือน	3 เดือน	6 เดือน	3 เดือน	6 เดือน
ORB	2577	3984	3615	5359	2438	1270
FAST	6753	6930	7294	8042	6209	5851
BRISK	1002	1965	1462	2643	903	509
Harris	2676	1572	2058	1772	1799	1193
KAZE	1648	2141	1989	2240	1615	1174
SURF	39	142	74	139	99	37
MinEigen	3222	1999	2631	2301	3114	2676
MSER	1643	2140	2087	2812	1232	1124

ตารางที่ 4.2 การหาค่าเปอร์เซ็นต์การ Matching ของแต่ละอัลกอริทึมจากรูปภาพแบบ Grayscale

อัลกอริทึม	ระยะเข้าตัวที่ 1		ระยะเข้าตัวที่ 2		ระยะเข้าตัวที่ 3		ค่าเฉลี่ย % ความถูกต้อง
	จำนวน Matching	% ความ ถูกต้อง	จำนวน Matching	% ความ ถูกต้อง	จำนวน Matching	% ความ ถูกต้อง	
ORB	5	80%	4	75%	4	50%	69.23%
FAST	18	33.33%	7	100%	14	28.57%	43.59%
BRISK	8	75%	4	50%	5	80%	70.59%
Harris	7	100%	8	75%	8	37.50%	69.57%
KAZE	44	93.18%	17	82.35%	22	72.73%	85.54%
SURF	6	66.67%	5	20%	7	28.57%	38.89%
MinEigen	12	83.33%	10	60%	5	80%	74.07%
MSER	24	58.33%	40	15%	35	68.57%	44.44%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดลองหาค่า key point และเปอร์เซ็นต์การ Matching ที่เหมาะสมแต่ละอัลกอริทึมของระยะ 3 ตัว จากรูปภาพแบบ Thin ได้ผลตามตารางที่ 4.3 และตารางที่ 4.4

ตารางที่ 4.3 การหาค่า key point ของแต่ละอัลกอริทึมจากรูปภาพแบบ Thin

อัลกอริทึม	E2/E3		A2/A3		D1/D2	
	3 เดือน	6 เดือน	3 เดือน	6 เดือน	3 เดือน	6 เดือน
ORB	2584	2335	2655	2446	2510	2667
FAST	1390	1237	1382	1231	1239	1442
BRISK	212	204	519	64	232	513
Harris	784	636	756	752	704	756
KAZE	582	511	614	593	591	627
SURF	502	374	437	459	423	489
MinEigen	751	600	741	721	599	677
MSER	449	413	459	459	409	435

ตารางที่ 4.4 การหาค่าเปอร์เซ็นต์การ Matching ของแต่ละอัลกอริทึมจากรูปภาพแบบ Thin

อัลกอริทึม	ระยะเข้ตัวที่ 1		ระยะเข้ตัวที่ 2		ระยะเข้ตัวที่ 3		ค่าเฉลี่ย % ความถูกต้อง
	จำนวน Matching	% ความ ถูกต้อง	จำนวน Matching	% ความ ถูกต้อง	จำนวน Matching	% ความ ถูกต้อง	
ORB	40	25%	35	28.57%	44	27.27%	26.89%
FAST	207	-	176	-	165	-	-
BRISK	44	18.18%	56	14.29%	39	17.95%	16.55%
Harris	30	16.67%	28	17.86%	29	3.45%	14.29%
KAZE	49	38.78%	39	20.51%	24	33.33%	40.18%
SURF	37	16.22%	32	3.13%	36	2.78%	7.62%
MinEigen	28	14.29%	32	15.63%	28	17.86%	15.91%
MSER	39	43.59%	24	33.33%	27	29.63%	36.67%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 การทดสอบอัลกอริทึม KAZE ในการประมวลผลสำหรับโปรแกรมประยุกต์บน โทรศัพท์เคลื่อนที่ (Mobile Application)

ในการทดสอบผู้จัดทำนำรูปภาพหน้าท้องจระเข้มาผ่านกระบวนการ Deep Learning ซึ่งจะเป็นการตัดส่วนภาพพื้นหลังออก เพื่อง่ายต่อการใช้งานในขั้นตอนถัดไป แสดงดังรูปที่ 4.104 และรูปที่ 4.105



รูปที่ 4.104 รูปภาพหน้าท้องจระเข้ก่อนกระบวนการ Deep Learning

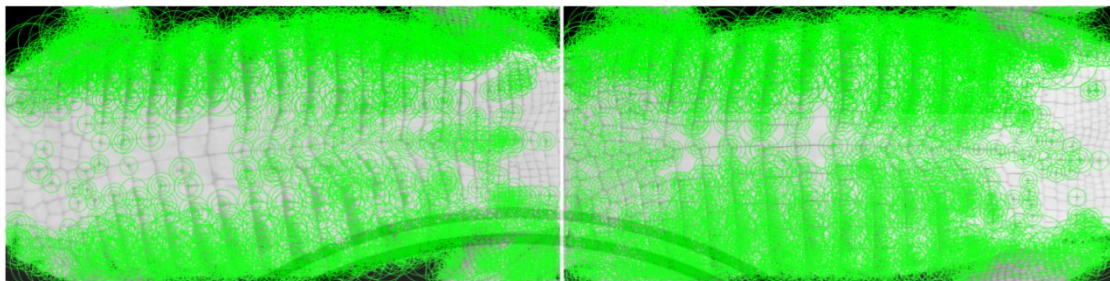


รูปที่ 4.105 รูปภาพหน้าท้องจระเข้หลังผ่านกระบวนการ Deep Learning

นำรูปภาพหน้าท้องจระเข้ที่ผ่านกระบวนการ Deep Learning ในการตัดภาพพื้น
หลังออกมาแปลงเป็นภาพ Grayscale และใช้อัลกอริทึม KAZE ในการประมวลผลเนื่องจากมี

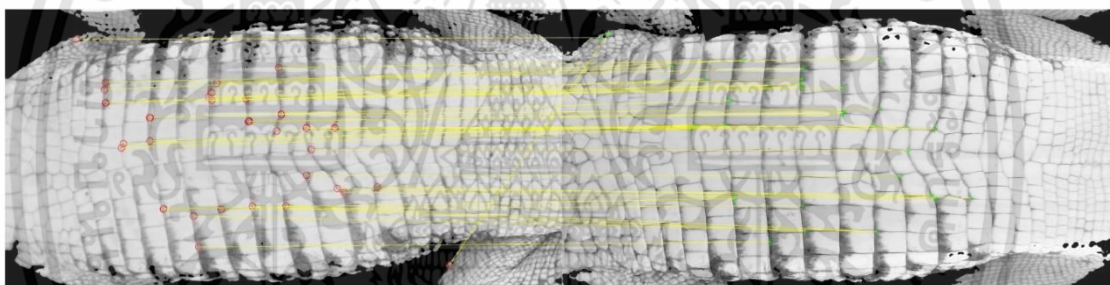
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เปอร์เซ็นต์ความถูกต้องในการ Matching มากที่สุด โดยทดสอบหา key point ของจระเข้ตัวเดียวกัน ในช่วงวัย 3 เดือนและช่วงวัย 6 เดือน แสดงดังรูปที่ 4.106



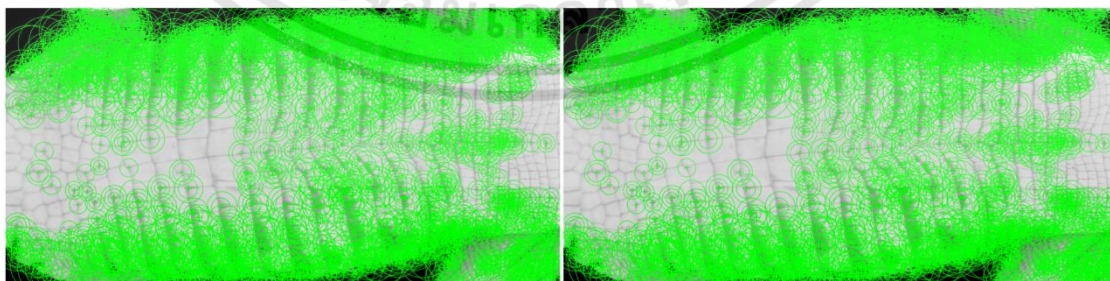
รูปที่ 4.106 key point ของจระเข้ตัวเดียวกันในช่วงวัย 3 เดือนและช่วงวัย 6 เดือน

ทดสอบการใช้อัลกอริทึม KAZE ในการเปรียบเทียบรูปภาพหลายหน้าห้องจระเข้ตัวเดียวกันในช่วงวัย 3 เดือนกับช่วงวัย 6 เดือน การ Matching แสดงดังรูปที่ 4.107



รูปที่ 4.107 การ Matching ของจระเข้ตัวเดียวกันในช่วงวัย 3 เดือนกับช่วงวัย 6 เดือน

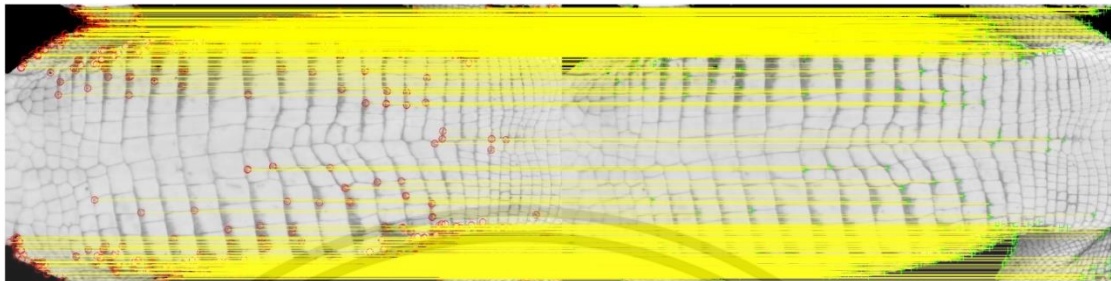
ทดสอบการใช้อัลกอริทึม KAZE ในการหา key point ของจระเข้ตัวเดียวกันในช่วงวัย 3 เดือน แสดงดังรูปที่ 4.108



รูปที่ 4.108 key point ของจระเข้ตัวเดียวกันในช่วงวัย 3 เดือน

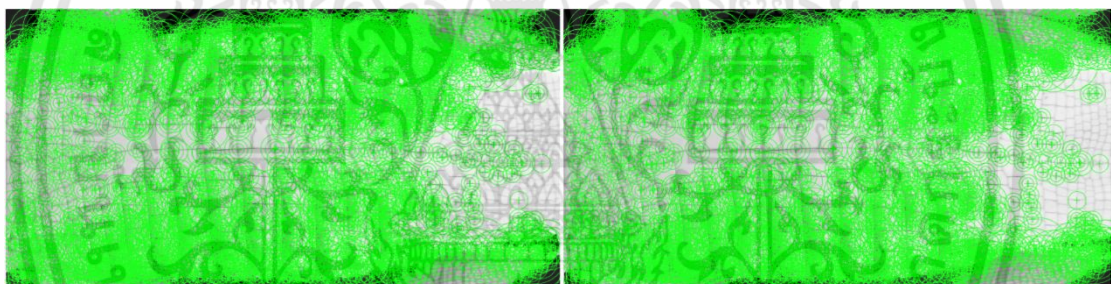
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดสอบการใช้อัลกอริทึม KAZE ในการเปรียบเทียบรูปภาพลายหน้าท้องจระเข้ตัวเดียวกันในช่วงวัย 3 เดือน การ Matching แสดงดังรูปที่ 4.109



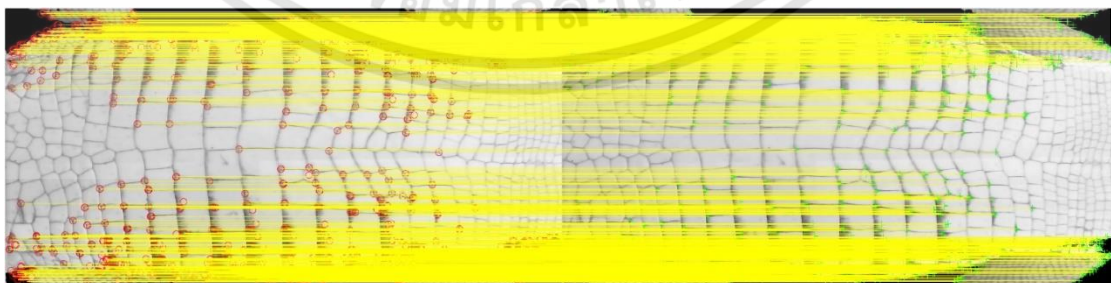
รูปที่ 4.109 การ Matching ของจระเข้ตัวเดียวกันในช่วงวัย 3 เดือน

ทดสอบการใช้อัลกอริทึม KAZE ในการหา key point ของจระเข้ตัวเดียวกันในช่วงวัย 6 เดือน แสดงดังรูปที่ 4.110



รูปที่ 4.110 key point ของจระเข้ตัวเดียวกันในช่วงวัย 6 เดือน

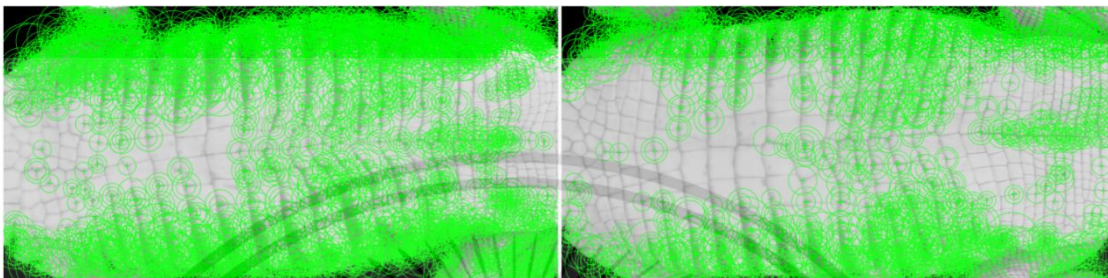
ทดสอบการใช้อัลกอริทึม KAZE ในการเปรียบเทียบรูปภาพลายหน้าท้องจระเข้ตัวเดียวกันในช่วงวัย 6 เดือน การ Matching แสดงดังรูปที่ 4.111



รูปที่ 4.111 การ Matching ของจระเข้ตัวเดียวกันในช่วงวัย 6 เดือน

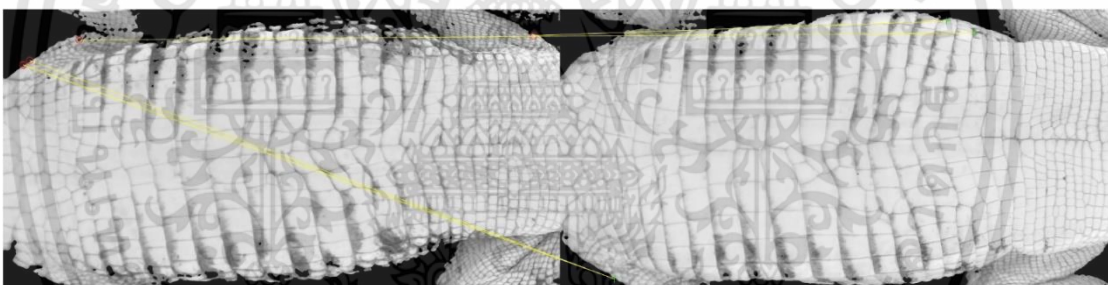
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดสอบการใช้อัลกอริทึม KAZE ในการหา key point ของกระเซঁคนละตัวกันในช่วงวัย 3 เดือน แสดงดังรูปที่ 4.112



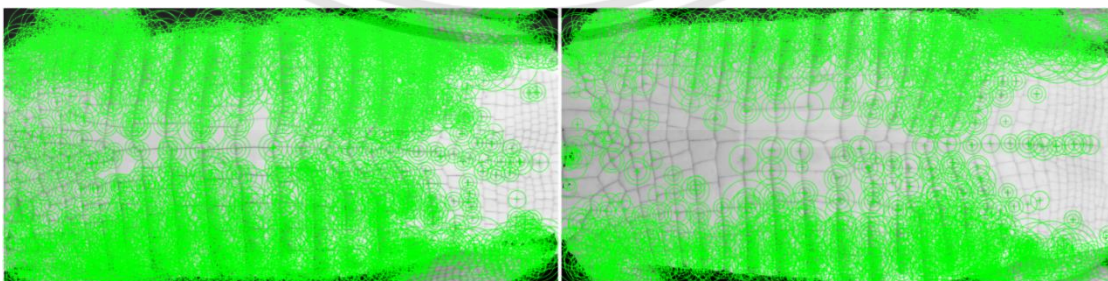
รูปที่ 4.112 key point ของกระเซঁคนละตัวกันในช่วงวัย 3 เดือน

ทดสอบการใช้อัลกอริทึม KAZE ในการเปรียบเทียบรูปภาพหลายหน้าห้องกระเซঁคนละตัวกันในช่วงวัย 3 เดือน การ Matching แสดงดังรูปที่ 4.113



รูปที่ 4.113 การ Matching ของกระเซঁคนละตัวกันในช่วงวัย 3 เดือน

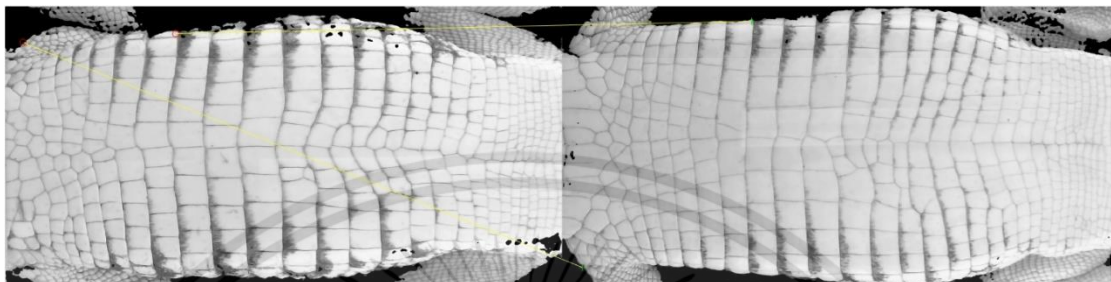
ทดสอบการใช้อัลกอริทึม KAZE ในการหา key point ของกระเซঁคนละตัวกันในช่วงวัย 6 เดือน แสดงดังรูปที่ 4.114



รูปที่ 4.114 key point ของกระเซঁคนละตัวกันในช่วงวัย 6 เดือนและช่วงวัย 6 เดือน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทดสอบการใช้อัลกอริทึม KAZE ในการเปรียบเทียบรูปภาพลายหน้าท้องกระเซ้คน
 ละตัวกันในช่วงวัย 6 เดือน การ Matching แสดงดังรูปที่ 4.115



รูปที่ 4.115 การ Matching ของกระเซ้คนละตัวกันในช่วงวัย 6 เดือน

ทดลองหาค่า key point และเปอร์เซ็นต์การ Matching ระหว่างกระเซ้ตัวเดียวกัน
 ในช่วงวัย 3 เดือนกับช่วงวัย 6 เดือน มีจำนวนกระเซ้ทั้งหมด 10 ตัว โดยใช้อัลกอริทึม KAZE ได้ผล
 ตามตารางที่ 4.5

ตาราง 4.5 การหาค่า key point และเปอร์เซ็นต์การ Matching โดยใช้อัลกอริทึม KAZE

กระเซ้ตัวที่	จำนวน Key point		จำนวน Matching	% ความถูกต้อง
	3 เดือน	6 เดือน		
กระเซ้ตัวที่ 1	1260	1439	19	94.74%
กระเซ้ตัวที่ 2	1564	1993	14	92.86%
กระเซ้ตัวที่ 3	1856	2431	38	94.74%
กระเซ้ตัวที่ 4	1426	2566	20	100%
กระเซ้ตัวที่ 5	2841	3460	44	100%
กระเซ้ตัวที่ 6	1723	1973	10	90%
กระเซ้ตัวที่ 7	1993	2612	54	100%
กระเซ้ตัวที่ 8	2345	3114	37	94.59%
กระเซ้ตัวที่ 9	1957	2542	55	100%
กระเซ้ตัวที่ 10	2961	3273	79	100%

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

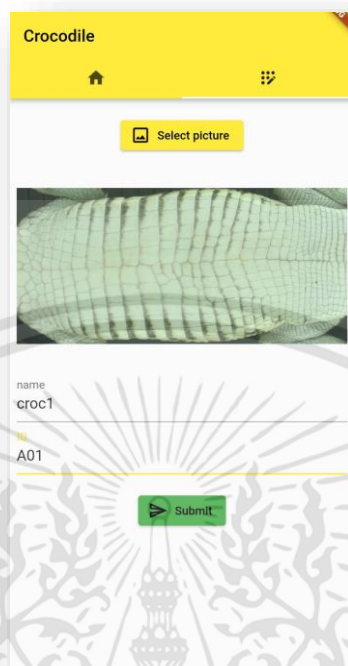
4.6 ทดสอบโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ (Mobile Application)

4.6.1 ฝั่ง Register ส่งรูปภาพ Name และ ID ไปเก็บยังเซิร์ฟเวอร์

ผู้จัดทำได้พัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่โดยใช้ชื่อว่า Crocodile เพื่อใช้ในการลงทะเบียน เมื่อเข้าโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ ผู้ใช้ต้องทำการเลือกรูปภาพหนึ่งหน้าห้องจะเซ็ที่ต้องการลงทะเบียน พร้อมระบุชื่อ croc1 และหมายเลขไอดี A01 แสดงดังรูปที่ 4.116 และรูปที่ 4.117



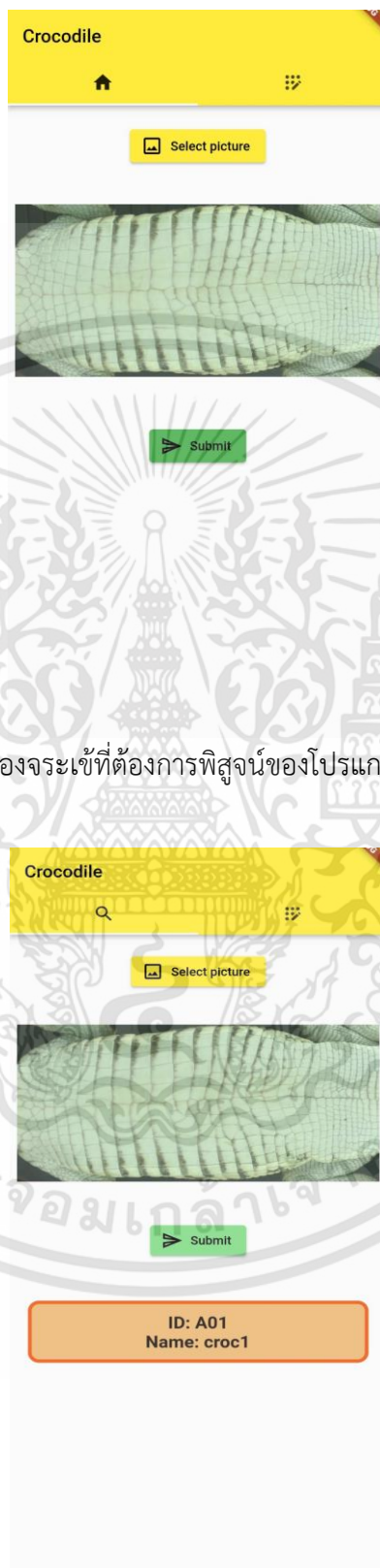
รูปที่ 4.116 หน้าหลักโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่โดยใช้ชื่อว่า Crocodile



รูปที่ 4.117 หน้าลงทะเบียนของโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่

4.6.2 ฟังก์ชัน Identify ส่งรูปภาพแล้วรอการยืนยันว่าเป็น ID ของหน้าท้องจระเข้ตัวใด

ผู้จัดทำได้พัฒนาโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ ผู้ใช้ต้องทำการเลือกรูปภาพหน้าท้องจระเข้ที่ต้องการพิสูจน์เอกลักษณ์จระเข้ไปยังเซิร์ฟเวอร์แล้วผ่านกระบวนการ image processing แล้วส่งหมายเลขไอดี A03 มาแสดงยังโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ แสดงดังรูปที่ 4.118 และรูปที่ 4.119



รูปที่ 4.118 รูปภาพหน้าจอห้องจระเข้ที่ต้องการพิสูจน์ของโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่

รูปที่ 4.119 หน้าแสดงผลรูปภาพหน้าจอห้องจระเข้ที่ต้องการพิสูจน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

โครงการโปรแกรมประยุกต์การพิสูจน์เอกลักษณ์จะเข้าโดยใช้การประมวลผลภาพมีเป้าหมายเพื่อออกแบบและพัฒนาอัลกอริทึมสำหรับการพิสูจน์เอกลักษณ์จะเข้าจากรูปลายหน้าห้องจะเข้า รวมถึงการออกแบบการฐานข้อมูลสำหรับการพิสูจน์เอกลักษณ์จะเข้า และออกแบบการสร้างโปรแกรมประยุกต์เพื่อระบุเอกลักษณ์จะเข้าจากรูปภาพ โดยใช้อัลกอริทึม 8 ตัวคือ ORB, FAST, BRISK, Harris, KAZE, SURF, MinEigen และ MSER ซึ่งพิจารณาจากรูปภาพ 2 แบบ คือ Grayscale และ Thin จำนวน 3 ตัว ในช่วงวัย 3 เดือนกับ 6 เดือน

1) ในส่วนของการวิเคราะห์รูปภาพแบบ grayscale โดยจากผลทดลองพบว่าอัลกอริทึม ORB เปอร์เซ็นต์ความถูกต้องการ Matching 69.23% อัลกอริทึม FAST เปอร์เซ็นต์ความถูกต้องการ Matching 43.59% อัลกอริทึม BRISK เปอร์เซ็นต์ความถูกต้องการ Matching 70.59% อัลกอริทึม Harris เปอร์เซ็นต์ความถูกต้องการ Matching 69.57% อัลกอริทึม KAZE เปอร์เซ็นต์ความถูกต้องการ Matching 85.54% อัลกอริทึม SURF เปอร์เซ็นต์ความถูกต้องการ Matching 38.89% อัลกอริทึม MinEigen เปอร์เซ็นต์ความถูกต้องการ Matching 74.07% และอัลกอริทึม MSER เปอร์เซ็นต์ความถูกต้องการ Matching 44.44%

2) ในส่วนของการวิเคราะห์รูปภาพแบบ Thin โดยจากผลทดลองพบว่าอัลกอริทึม ORB เปอร์เซ็นต์ความถูกต้องการ Matching 26.89% อัลกอริทึม FAST เปอร์เซ็นต์ความถูกต้องการ Matching ไม่สามารถหาได้ อัลกอริทึม BRISK เปอร์เซ็นต์ความถูกต้องการ Matching 16.55% อัลกอริทึม Harris เปอร์เซ็นต์ความถูกต้องการ Matching 14.29% อัลกอริทึม KAZE เปอร์เซ็นต์ความถูกต้องการ Matching 40.18% อัลกอริทึม SURF เปอร์เซ็นต์ความถูกต้องการ Matching 7.62% อัลกอริทึม MinEigen เปอร์เซ็นต์ความถูกต้องการ Matching 15.91% และอัลกอริทึม MSER เปอร์เซ็นต์ความถูกต้องการ Matching 36.67%

ซึ่งผลการทดลองถ้าเปรียบเทียบทั้ง 8 อัลกอริทึมจะพบว่าอัลกอริทึม KAZE รูปแบบ Grayscale และ Thin ให้ผลความน่าเชื่อถือมากที่สุด เนื่องจากเปอร์เซ็นต์ความถูกต้องการ Matching มีค่าที่ค่อนข้างสูงกว่าอัลกอริทึมตัวอื่น จึงนำอัลกอริทึม KAZE รูปแบบ Grayscale มาใช้ในการประมวลผลสำหรับโปรแกรมประยุกต์บนโทรศัพท์เคลื่อนที่ (Mobile Application) โดยจากผลการทดลองในจะเข้าจำนวน 10 ตัว มีค่าเฉลี่ยเปอร์เซ็นต์ความถูกต้องการ Matching 98.11%

5.2 ข้อเสนอแนะ

อัลกอริทึมที่ใช้พิสูจน์เอกลักษณ์ของจระเข้ควรมีความยืดหยุ่นในการใช้งานและมีความแม่นยำมากขึ้น และในส่วนของอัลกอริทึมที่พัฒนาขึ้นมาในการแปลงรูปภาพจาก Physical ให้เหลือเพียงเส้นหน้าท้องที่สนใจ ควรมีสัญญาณรบกวนลดลง เพื่อนำมาพัฒนาต่อในส่วนของการหา Key point และนำไป Matching



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

- [1] Shreenidhi Sudhakar. “Histogram Equalization”
<https://towardsdatascience.com/histogram-equalization-5d1013626e64>
- [2] Wikipedia. “Gaussian Blur” https://en.wikipedia.org/wiki/Gaussian_blur
- [3] A. Walker and E. Wolfart. “Laplacian/Laplacian of Gaussian”
- [4] A. Walker and E. Wolfart. “Dilation”
- [5] Manish Mathuria. "Fingerprint Matching using Ridge-End and Bifurcation Points."
https://www.academia.edu/2537762/Fingerprint_Matching_using_Ridge-End_and_Bifurcation_Point
- [6] Lukasz Wieclaw. "A minutiae-based matching algorithms in fingerprint Recognition systems." https://www.academia.edu/2508970/A_minutiae-based_matching_algorithms_in_fingerprint_recognition_systems
- [7] Deepanshu Tyagi. “Introduction to ORB (Oriented FAST and Rotated BRIEF).”
<https://medium.com/analytics-vidhya/introduction-to-orb-oriented-fast-and-rotated-brief-4220e8ec40cf>
- [8] Deepanshu Tyagi. “Introduction to Feature Detection and Matching.”
<https://medium.com/analytics-vidhya/introduction-to-feature-detection-and-matching-65e27179885d>
- [9] วิศวธน. ดีภักดี. “คุณลักษณะจากการทดสอบกลุ่มแบบเร่ง (FAST).” ข้อมูลเบื้องต้นเกี่ยวกับ FAST (คุณลักษณะจากการทดสอบกลุ่มแบบเร่ง) (ichi.pro)
- [10] C. Harris และ MJ Stephens เครื่องตรวจจับมุมและขอบแบบรวม ใน Alvey Vision Conference หน้า 147–152, 1988
- [11] Edward Rosten, Reid Porter และ Tom Drummond “เร็วขึ้นและดีขึ้น: วิธีการเรียนรู้ของเครื่องในการตรวจจับมุม” ใน IEEE Trans Pattern Analysis and Machine Intelligence, 2010, vol 32, pp. 105–119

บรรณานุกรม (ต่อ)

- [12] Edward Rosten และ Tom Drummond “การเรียนรู้ของเครื่องสำหรับการตรวจจับมุมความเร็วสูง” ใน European Conference on Computer Vision ครั้งที่ 9 ฉบับที่ 9 1, 2549, หน้า 430–443
- [13] R.Szeliski Computer Vision: อัลกอริทึมและแอปพลิเคชัน สปริงเกอร์, 2010.
- [14] Unknow. “MATLAB (แมตแล็บ) ภาษาคอมพิวเตอร์ระดับสูง คืออะไร และเกี่ยวข้องกับอย่างไรกับการเขียนแอปพลิเคชัน” <https://www.mindphp.com/4043-what-is-matlab.html>.
- [15] Wikipedia. “จระเข้”. <https://th.wikipedia.org/wiki/จระเข้>
H. Bay, T. Tuytelaars และ L. Van Gool SURF: เร่งคุณสมบัติที่แข็งแกร่ง ใน Proc. ECCV, หน้า 404– 417, 2549
- [16] Mos Programmer. “มาใช้ NodeJS กันเถอะ”
<https://medium.com/@mosprogramer/>
- [17] Chai Phonbopit. “MongoDB คืออะไร? + สอนวิธีใช้งานเบื้องต้น”
<https://devahoy.com/blog/2015/08/getting-started-with-mongodb/>
- [18] TPW. “ทำความรู้จักภาษา Dart ฉบับโปรแกรมเมอร์”
<https://www.centrilliontech.co.th/blog/2570/dart-101-introduction-for-programmers/>
- [19] Code Mobile company limited. “Flutter”
<https://www.codemobiles.co.th/online/course.php?id=flutter>
- [20] Wikipedia. “JSON”
<https://th.wikipedia.org/wiki/%E0%B9%80%E0%B8%88%E0%B8%8B%E0%B8%B1%E0%B8%99>
- [21] Natthawat Phongchit. “Convolutional Neural Network (CNN)”
<https://medium.com/@natthawatphongchit/>
- [22] ABB. “Deep Learning” <https://new.abb.com/news/detail/58004/deep-learning>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Main.dart

```

import 'package:crocodile/register.dart';
import 'package:flutter/material.dart';
import 'package:crocodile/search.dart';

void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.yellow,
      ),
      home: const MyHomePage(title: 'Crocodile'),
    );
  }
}

class MyHomePage extends StatefulWidget {
  const MyHomePage({Key? key, required this.title}) : super(key: key);
  final String title;

  @override
  State<MyHomePage> createState() => _MyHomePageState();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}

class _MyHomePageState extends State<MyHomePage> with TickerProviderStateMixin
{
  @override
  Widget build(BuildContext context) {
    return DefaultTabController(
      length: 2,
      child: Scaffold(
        appBar: AppBar(
          title: Text(widget.title),
          bottom: const TabBar(
            tabs: [
              Tab(icon: Icon(Icons.search)),
              Tab(icon: Icon(Icons.app_registration_rounded)),
            ],
          ),
        ),
        body: TabBarView(children: [SearchPage(), RegisterPage()]),
      ));
  }
}

```

ฝั่ง register

```
import 'dart:convert';
import 'dart:io';

import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:image_picker/image_picker.dart';
import 'package:http/http.dart' as http;

class RegisterPage extends StatefulWidget {
  RegisterPage({Key? key}) : super(key: key);

  @override
  State<RegisterPage> createState() => _RegisterPageState();
}

class _RegisterPageState extends State<RegisterPage> {
  File? _image;
  final ImagePicker _picker = ImagePicker();

  Future<void> _selectImage() async {
    final image = await _picker.pickImage(source: ImageSource.gallery);

    if (image != null) {
      setState(() {
        _image = File(image.path);
      });
    }
  }

  void _reset() {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Navigator.popAndPushNamed(context, "/");
}

@override
Widget build(BuildContext context) {
  return SingleChildScrollView(
    child: Center(
      child: Padding(
        padding: const EdgeInsets.fromLTRB(0, 20, 0, 0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.start,
          children: <Widget>[
            ElevatedButton.icon(
              onPressed: _selectImage,
              icon: const Icon(
                Icons.image_outlined,
                size: 24.0,
              ),
              label: const Text('Select picture' ), // <-- Text
            ),
            _image == null
              ? const Text("please select picture register")
              : buildForm(_image!, _reset)
          ],
        ),
      ),
    ),
  );
}
}

```

```

Widget buildForm(File image, Function reset) {
  final _formKey = GlobalKey<FormState>();
  String _name = "";
  String _id = "";
  var url = Uri.parse(
    'http://161.246.18.222:5000/upload');

  return Padding(
    padding: const EdgeInsets.all(8.0),
    child: Form(
      key: _formKey,
      child: Column(
        children: [
          Image.file(
            image,
            height: 250,
          ),
          TextFormField(
            style: const TextStyle(fontSize: 18),
            decoration: const InputDecoration(
              labelText: "name", border: UnderlineInputBorder()),
            validator: (value) {
              if (value == null || value.isEmpty) {
                return 'Please enter Name';
              }
              return null;
            },
            onSaved: (value) {
              _name = value!;
            },
          ),
        ],
      ),
    ),
  );
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

TextField(
  style: const TextStyle(fontSize: 18),
  decoration: const InputDecoration(
    labelText: "ID", border: UnderlineInputBorder()),
  validator: (value) {
    if (value == null || value.isEmpty) {
      return 'Please enter ID';
    }
    return null;
  },
  onSave: (value) {
    _id = value!;
  },
),
Padding(
  padding: const EdgeInsets.symmetric(vertical: 20.0),
  child: ElevatedButton.icon(
    onPressed: () async {
      print("Click");
      if (_formKey.currentState!.validate()) {
        print("Click");
        _formKey.currentState?.save();
      }
      Map data = {
        'photo': await convertImage(image),
        'name': _name,
        'id': _id
      };

      var body = json.encode(data);

```

```

final response = await http.post(url,
  headers: {"Content-Type": "application/json"},
  body: body);
print("send");
if (response.statusCode == 201) {
  reset();
  await showSuccess();
} else {
  await showFail();
}
},
icon: const Icon(Icons.send_outlined),
style: ElevatedButton.styleFrom(
  primary: Color.fromARGB(255, 131, 221, 134),
),
label: const Text('Submit'),
),
],
));
}

```

```

Future<String> convertImage(File image) async {
  List<int> bytes = await image.readAsBytes();
  String img64 = base64Encode(bytes);
  return img64;
}

```

```

Future<void> showSuccess() async {
  await Fluttertoast.showToast(

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

msg: "Upload success",
toastLength: Toast.LENGTH_SHORT,
gravity: ToastGravity.BOTTOM,
backgroundColor: Colors.lightGreen[700],
timeInSecForLosWeb: 2,
fontSize: 16.0);
}

```

```

Future<void> showFail() async {
  await Fluttertoast.showToast(
    msg: "Upload Fail",
    toastLength: Toast.LENGTH_SHORT,
    gravity: ToastGravity.BOTTOM,
    backgroundColor: Colors.red[400],
    timeInSecForLosWeb: 2,
    fontSize: 16.0);
}

```

ฝั่ง Identity

```

import 'dart:convert';
import 'dart:io';

import 'package:flutter/material.dart';
import 'package:fluttertoast/fluttertoast.dart';
import 'package:image_picker/image_picker.dart';
import 'package:http/http.dart' as http;

class SearchPage extends StatefulWidget {
  SearchPage({Key? key}) : super(key: key);

  @override
  State<SearchPage> createState() => _SearchPageState();
}

class _SearchPageState extends State<SearchPage> {
  File? _image;
  final ImagePicker _picker = ImagePicker();

  Future<void> _selectImage() async {
    final image = await _picker.pickImage(source: ImageSource.gallery);

    if (image != null) {
      setState(() {
        _image = File(image.path);
      });
    }
  }

  void _reset() {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Navigator.popAndPushNamed(context, "/");
}

@override
Widget build(BuildContext context) {
  return SingleChildScrollView(
    child: Center(
      child: Padding(
        padding: const EdgeInsets.fromLTRB(0, 20, 0, 0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.start,
          children: <Widget>[
            ElevatedButton.icon(
              onPressed: _selectImage,
              icon: const Icon(
                Icons.image_outlined,
                size: 24.0,
              ),
              label: const Text('Select picture'), // <-- Text
            ),
            _image == null
              ? const Text("please select picture search")
              : buildForm(_image!, _reset)
          ],
        ),
      ),
    );
}
}

```

```

Widget buildForm(File image, Function reset) {
  final _formKey = GlobalKey<FormState>();
  var url = Uri.parse('http://161.246.18.222:5000/upload');

  return Padding(
    padding: const EdgeInsets.all(8.0),
    child: Form(
      key: _formKey,
      child: Column(
        children: [
          Image.file(
            image,
            height: 250,
          ),
          Padding(
            padding: const EdgeInsets.symmetric(vertical: 20.0),
            child: ElevatedButton.icon(
              onPressed: () async {
                if (_formKey.currentState!.validate()) {
                  _formKey.currentState?.save();

                  Map data = {
                    'photo': await convertImage(image),
                  };

                  var body = json.encode(data);

                  final response = await http.post(url,
                    headers: {"Content-Type": "application/json"},
                    body: body);
                }
              },
            ),
          ),
        ],
      ),
    ),
  );
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        if (response.statusCode == 201) {
            reset();
            await showSuccess();
        } else {
            await showFail();
        }
    }
},
icon: const Icon(Icons.send_outlined),
style: ElevatedButton.styleFrom(
    primary: Color.fromARGB(255, 115, 221, 118),
),
label: const Text('Submit'),
),
),
],
));
}

```

```

Future<String> convertImage(File image) async {
    List<int> bytes = await image.readAsBytes();
    String img64 = base64Encode(bytes);
    return img64;
}

```

```

Future<void> showSuccess() async {
    await Fluttertoast.showToast(
        msg: "Upload success",
        toastLength: Toast.LENGTH_SHORT,
    );
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

gravity: ToastGravity.BOTTOM,
backgroundColor: Colors.lightGreen[700],
timeInSecForlosWeb: 2,
fontSize: 16.0);
}

```

```

Future<void> showFail() async {

```

```

  await Fluttertoast.showToast(

```

```

    msg: "Upload Fail",

```

```

    toastLength: Toast.LENGTH_SHORT,

```

```

    gravity: ToastGravity.BOTTOM,

```

```

    backgroundColor: Colors.red[400],

```

```

    timeInSecForlosWeb: 2,

```

```

    fontSize: 16.0);
}

```

```

//Future<void> showResult() async {

```

```

  //var url = ('http://161.246.18.222:5000/upload');

```

```

  // var response = await http.get(url);

```

```

  //if (response.statusCode == 200) {

```

```

  //var data = json.decode

```

```

    //data .showToast(

```

```

    //)

```

```

  //}

```

```

// if (http.response())

```

```

// }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// if (response.statusCode == 200) {  
//   var data = json.decode(response.body) as  
// }  
// {  
//   "id" : 1,  
//   "name" : "doubleGreen"  
// }
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ORB

```
%Unique Pattern Create
```

```
clc;
```

```
clear;
```

```
R_Image1 = imread('assets/imageRaw/crocE2.png');
```

```
R_Image2 = imread('assets/imageRaw/crocE3.png');
```

```
IR_Image1 = rgb2gray(R_Image1);
```

```
IR_Image2 = rgb2gray(R_Image2);
```

```
%resize
```

```
RESIZE_CANNY = [500, 1000];
```

```
I_C1 = imresize(IR_Image1, RESIZE_CANNY);
```

```
I_C2 = imresize(IR_Image2, RESIZE_CANNY);
```

```
points1 = detectORBFeatures(I_C1,"ScaleFactor",1.2,"NumLevels",8);
```

```
points2 = detectORBFeatures(I_C2,"ScaleFactor",1.2,"NumLevels",8);
```

```
figure
```

```
subplot(1,2,1)
```

```
imshow(I_C1)
```

```
hold on
```

```
plot(points1)
```

```
title('ORBcrocE2')
```

```
subplot(1,2,2)
```

```
imshow(I_C2)
```

```
hold on
```

```
plot(points2)
```

```
title('ORBcrocE3')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

[features1,valid_points1] = extractFeatures(I_C1,points1);
[features2,valid_points2] = extractFeatures(I_C2,points2);
indexPairs = matchFeatures(features1,features2);
matchedPts1 = valid_points1(indexPairs(:,1));
matchedPts2 = valid_points2(indexPairs(:,2));
figure
showMatchedFeatures(I_C1,I_C2,matchedPts1,matchedPts2,'montage');

```

FAST

%Unique Pattern Create

```
clc;
```

```
clear;
```

```
R_Image1 = imread('assets/imageRaw/crocE2.png');
```

```
R_Image2 = imread('assets/imageRaw/crocE3.png');
```

```
IR_Image1 = rgb2gray(R_Image1);
```

```
IR_Image2 = rgb2gray(R_Image2);
```

%resize

```
RESIZE_CANNY = [500, 1000];
```

```
I_C1 = imresize(IR_Image1, RESIZE_CANNY);
```

```
I_C2 = imresize(IR_Image2, RESIZE_CANNY);
```

```
points1 = detectFASTFeatures(I_C1,"MinContrast",0.02,"MinQuality",0.02)
```

```
points2 = detectFASTFeatures(I_C2,"MinContrast",0.02,"MinQuality",0.02)
```

figure

```
subplot(1,2,1)
```

```
imshow(I_C1)
```

```
hold on
```

```
plot(points1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

title('FASTcrocE2')
subplot(1,2,2)
imshow(I_C2)
hold on
plot(points2);
title('FASTcrocE3')

[features1,valid_points1] = extractFeatures(I_C1,points1);
[features2,valid_points2] = extractFeatures(I_C2,points2);
indexPairs = matchFeatures(features1,features2);
matchedPts1 = valid_points1(indexPairs(:,1));
matchedPts2 = valid_points2(indexPairs(:,2));
figure
showMatchedFeatures(I_C1,I_C2,matchedPts1,matchedPts2,'montage');

BRISK
%Unique Pattern Create
clc;
clear;
R_Image1 = imread('assets/imageRaw/crocE2.png');
R_Image2 = imread('assets/imageRaw/crocE3.png');

IR_Image1 = rgb2gray(R_Image1);
IR_Image2 = rgb2gray(R_Image2);

%resize
RESIZE_CANNY = [500, 1000];
I_C1 = imresize(IR_Image1, RESIZE_CANNY);
I_C2 = imresize(IR_Image2, RESIZE_CANNY);

points1 = detectBRISKFeatures(I_C1,"MinContrast",0.1,"MinQuality",0.1,"NumOctaves",5)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
points2 = detectBRISKFeatures(I_C2,"MinContrast",0.1,"MinQuality",0.1,"NumOctaves",5)
```

```
figure
```

```
subplot(1,2,1)
```

```
imshow(I_C1)
```

```
hold on
```

```
plot(points1)
```

```
title('BRISKcrocE2')
```

```
subplot(1,2,2)
```

```
imshow(I_C2)
```

```
hold on
```

```
plot(points2)
```

```
title('BRISKcrocE3')
```

```
[features1,valid_points1] = extractFeatures(I_C1,points1);
```

```
[features2,valid_points2] = extractFeatures(I_C2,points2);
```

```
indexPairs = matchFeatures(features1,features2);
```

```
matchedPts1 = valid_points1(indexPairs(:,1));
```

```
matchedPts2 = valid_points2(indexPairs(:,2));
```

```
figure
```

```
showMatchedFeatures(I_C1,I_C2,matchedPts1,matchedPts2,'montage');
```

```
Harris
```

```
%Unique Pattern Create
```

```
clc;
```

```
clear;
```

```
R_Image1 = imread('assets/imageRaw/crocE2.png');
```

```
R_Image2 = imread('assets/imageRaw/crocE3.png');
```

```
IR_Image1 = rgb2gray(R_Image1);
```

```
IR_Image2 = rgb2gray(R_Image2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

%resize
RESIZE_CANNY = [500, 1000];
I_C1 = imresize(IR_Image1, RESIZE_CANNY);
I_C2 = imresize(IR_Image2, RESIZE_CANNY);

points1 = detectHarrisFeatures(I_C1,"MinQuality",0.001,"FilterSize",5)
points2 = detectHarrisFeatures(I_C2,"MinQuality",0.001,"FilterSize",5)

figure
subplot(1,2,1)
imshow(I_C1)
hold on
plot(points1);
title('HarriscrocE2')
subplot(1,2,2)
imshow(I_C2)
hold on
plot(points2);
title('HarriscrocE3')

[features1,valid_points1] = extractFeatures(I_C1,points1);
[features2,valid_points2] = extractFeatures(I_C2,points2);
indexPairs = matchFeatures(features1,features2);
matchedPts1 = valid_points1(indexPairs(:,1));
matchedPts2 = valid_points2(indexPairs(:,2));

figure
showMatchedFeatures(I_C1,I_C2,matchedPts1,matchedPts2,'montage');

KAZE
%Unique Pattern Create

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

clc;
clear;
R_Image1 = imread('assets/imageRaw/crocE2.png');
R_Image2 = imread('assets/imageRaw/crocE3.png');

IR_Image1 = rgb2gray(R_Image1);
IR_Image2 = rgb2gray(R_Image2);

%resize
RESIZE_CANNY = [500, 1000];
I_C1 = imresize(IR_Image1, RESIZE_CANNY);
I_C2 = imresize(IR_Image2, RESIZE_CANNY);

points1 = detectKAZEFeatures(I_C1,"Threshold",0.0005)
points2 = detectKAZEFeatures(I_C2,"Threshold",0.0005)

figure
subplot(1,2,1)
imshow(I_C1)
hold on
plot(points1);
title('KAZEcrocE2')
subplot(1,2,2)
imshow(I_C2)
hold on
plot(points2);
title('KAZEcrocE3')

[features1,valid_points1] = extractFeatures(I_C1,points1);
[features2,valid_points2] = extractFeatures(I_C2,points2);
indexPairs = matchFeatures(features1,features2);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

matchedPts1 = valid_points1(indexPairs(:,1));
matchedPts2 = valid_points2(indexPairs(:,2));
figure
showMatchedFeatures(I_C1,I_C2,matchedPts1,matchedPts2,'montage');

```

SURF

%Unique Pattern Create

```
clc;
```

```
clear;
```

```
R_Image1 = imread('assets/imageRaw/crocE2.png');
```

```
R_Image2 = imread('assets/imageRaw/crocE3.png');
```

```
IR_Image1 = rgb2gray(R_Image1);
```

```
IR_Image2 = rgb2gray(R_Image2);
```

%resize

```
RESIZE_CANNY = [500, 1000];
```

```
I_C1 = imresize(IR_Image1, RESIZE_CANNY);
```

```
I_C2 = imresize(IR_Image2, RESIZE_CANNY);
```

```
points1 = detectSURFFeatures(I_C1,"MetricThreshold",3000)
```

```
points2 = detectSURFFeatures(I_C2,"MetricThreshold",3000)
```

figure

```
subplot(1,2,1)
```

```
imshow(I_C1)
```

```
hold on
```

```
plot(points1);
```

```
title('SURFcrocE2')
```

```
subplot(1,2,2)
```

```
imshow(I_C2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

hold on
plot(points2);
title('SURFcrocE3')

[features1,valid_points1] = extractFeatures(I_C1,points1);
[features2,valid_points2] = extractFeatures(I_C2,points2);
indexPairs = matchFeatures(features1,features2);
matchedPts1 = valid_points1(indexPairs(:,1));
matchedPts2 = valid_points2(indexPairs(:,2));
figure
showMatchedFeatures(I_C1,I_C2,matchedPts1,matchedPts2,'montage');

MinEigen
%Unique Pattern Create
clc;
clear;
R_Image1 = imread('assets/imageRaw/crocE2.png');
R_Image2 = imread('assets/imageRaw/crocE3.png');

IR_Image1 = rgb2gray(R_Image1);
IR_Image2 = rgb2gray(R_Image2);

%resize
RESIZE_CANNY = [500, 1000];
I_C1 = imresize(IR_Image1, RESIZE_CANNY);
I_C2 = imresize(IR_Image2, RESIZE_CANNY);

corners1 = detectMinEigenFeatures(I_C1,"MinQuality",0.02,"FilterSize",5);
corners2 = detectMinEigenFeatures(I_C2,"MinQuality",0.02,"FilterSize",5);

```

```

figure
subplot(1,2,1)
imshow(I_C1)
hold on
plot(corners1)
title('MinEigencrocE2')
subplot(1,2,2)
imshow(I_C2)
hold on
plot(corners2)
title('MinEigencrocE3')

[features1,valid_corners1] = extractFeatures(I_C1,corners1);
[features2,valid_corners2] = extractFeatures(I_C2,corners2);
indexPairs = matchFeatures(features1,features2);
matchedCorners1 = valid_corners1(indexPairs(:,1));
matchedCorners2 = valid_corners2(indexPairs(:,2));
figure
showMatchedFeatures(I_C1,I_C2,matchedCorners1,matchedCorners2,'montage');

MSER
%Unique Pattern Create
clc;
clear;
R_Image1 = imread('assets/imageRaw/crocE2.png');
R_Image2 = imread('assets/imageRaw/crocE3.png');

IR_Image1 = rgb2gray(R_Image1);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IR_Image2 = rgb2gray(R_Image2);

%resize
RESIZE_CANNY = [500, 1000];
I_C1 = imresize(IR_Image1, RESIZE_CANNY);
I_C2 = imresize(IR_Image2, RESIZE_CANNY);

mserRegions1 = detectMSERFeatures(I_C1,"ThresholdDelta",1);
mserRegions2 = detectMSERFeatures(I_C2,"ThresholdDelta",1);

figure
subplot(1,2,1)
imshow(I_C1)
hold on
plot(mserRegions1)
title('MSERcrocE2')
subplot(1,2,2)
imshow(I_C2)
hold on
plot(mserRegions2)
title('MSERcrocE3')

[features1,points1] = extractFeatures(I_C1,mserRegions1);
[features2,points2] = extractFeatures(I_C2,mserRegions2);
indexPairs = matchFeatures(features1,features2);
matchedPts1 = points1(indexPairs(:,1));
matchedPts2 = points2(indexPairs(:,2));

figure
showMatchedFeatures(I_C1,I_C2,matchedPts1,matchedPts2,'montage');

```

Thin

```
%Unique Pattern Create
```

```
clc;
```

```
clear;
```

```
R_Imagae = imread('assets/imageRaw/crocE2.png');
```

```
%resize
```

```
RESIZE_CANNY = [500, 1000];
```

```
I_C = imresize(R_Imagae, RESIZE_CANNY);
```

```
figure(1);
```

```
imshow(I_C);
```

```
FUDGEFACTOR_CONSTANT = 0.8;
```

```
DIALATION_THRESHOLD_CONSTANT = 4;
```

```
GAUSSIAN_KERNEL_CONSTANT = [3,3];
```

```
% convert to grayscale
```

```
I = rgb2gray(I_C);
```

```
figure(2);
```

```
imshow(I);
```

```
%Gaussian Blur
```

```
H = fspecial('Gaussian',GAUSSIAN_KERNEL_CONSTANT); % Gaussian lowpass filter
```

```
G = imfilter(I,H);
```

```
figure(3);
```

```
imshow(G);
```

```
%Edge Detection
```

```
EDGE_THEROEM_CONST = 'Canny';
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Bw = edge(G,EDGE_THEROEM_CONST);
figure(4);
imshow(Bw);

%Threshold
[~,threshold] = edge(G,EDGE_THEROEM_CONST);
BWs = edge(G,EDGE_THEROEM_CONST,threshold*FUDGEFACTOR_CONSTANT);
figure(5);
imshow(BWs);

%dilation increase width of line from center(px)
BF = bwmorph(BWs,'dilate',DIALATION_THRESHOLD_CONSTANT); % Morphological
operations on binary image
figure(6);
imshow(BF);

%thin image from divide line width
thin_image=~bwmorph(BF,'thin',Inf);
imtool(thin_image

```

Deep Learning

```

clc;
clear;

dataDir = fullfile('D:\','Project','1','ReseachAlgorithm','deeplabv3plus');
imgDir = fullfile(dataDir,'imageRaw(resize)');
imds = imageDatastore(imgDir);

I = readimage(imds,1);
figure
imshow(I)

classes = [
    "bg"
    "croc"
];

labelIDs = helper.camvidPixelLabelIDs;

labelDir = fullfile(dataDir,'imageLabel(resize)');
pxds = pixelLabelDatastore(labelDir,classes,labelIDs);

C = readimage(pxds,1);
cmap = camvidColorMap;
B = labeloverlay(I,C,'ColorMap',cmap);
figure
imshow(B)
pixelLabelColorbar(cmap,classes);

tbl = countEachLabel(pxds)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

frequency = tbl.PixelCount/sum(tbl.PixelCount);
figure
bar(1:numel(classes),frequency)
xticks(1:numel(classes))
xticklabels(tbl.Name)
xtickangle(45)
ylabel('Frequency')

[imdsTrain, imdsVal, imdsTest, pxdsTrain, pxdsVal, pxdsTest] =
helper.partitionCamVidData(imds,pxds);

numTrainingImages = numel(imdsTrain.Files)
numValImages = numel(imdsVal.Files)
numTestingImages = numel(imdsTest.Files)

imageSize = [785 1566 3];

numClasses = numel(classes);

lgraph = deeplabv3plusLayers(imageSize, numClasses, "resnet18");

imageFreq = tbl.PixelCount ./ tbl.ImagePixelCount;
classWeights = median(imageFreq) ./ imageFreq

pxLayer =
pixelClassificationLayer('Name','labels','Classes',tbl.Name,'ClassWeights',classWeights);
lgraph = replaceLayer(lgraph,"classification",pxLayer);

analyzeNetwork(lgraph);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

dsVal = combine(imdsVal,pxdsVal);

options = trainingOptions('sgdm', ...
    'LearnRateSchedule','piecewise',...
    'LearnRateDropPeriod',10,...
    'LearnRateDropFactor',0.3,...
    'Momentum',0.9, ...
    'InitialLearnRate',1e-3, ...
    'L2Regularization',0.005, ...
    'ValidationData',dsVal,...
    'MaxEpochs',30, ...
    'MiniBatchSize',8, ...
    'Shuffle','every-epoch', ...
    'CheckpointPath', tempdir, ...
    'VerboseFrequency',2,...
    'Plots','training-progress',...
    'ValidationPatience', 4);

dsTrain = combine(imdsTrain, pxdsTrain);

xTrans = [-10 10];
yTrans = [-10 10];
dsTrain = transform(dsTrain, @(data)augmentImageAndLabel(data,xTrans,yTrans));

net = trainNetwork(dsTrain,lgraph,options);

pretrainedNetwork = fullfile(dataDir,'net.mat');
data = load(pretrainedNetwork);
net = data.net;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

l1 = readimage(imdsTest,1);
C1 = semanticseg(l1, net);

B1 = labeloverlay(l1,C1);
figure
imshow(B1)
pixelLabelColorbar(cmap, classes);

expectedResult = readimage(pxdsTest,1);
actual = uint8(C1);
expected = uint8(expectedResult);
figure
imshowpair(actual, expected)

iou = jaccard(C,expectedResult);
table(classes,iou)

pxdsResults = semanticseg(imdsTest,net, ...
    'MiniBatchSize',4, ...
    'WriteLocation',tempdir, ...
    'Verbose',false);

metrics = evaluateSemanticSegmentation(pxdsResults,pxdsTest,'Verbose',false);

metrics.DataSetMetrics

metrics.ClassMetrics

////////////////////////////////////

function labelIDs = camvidPixelLabelIDs()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
labelIDs = { ...
```

```
    % "bg"
```

```
    [
```

```
    127 127 127; ...
```

```
    ]
```

```
    % "croc"
```

```
    [
```

```
    153 217 234; ...
```

```
    ]
```

```
};
```

```
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
function pixelLabelColorbar(cmap, classNames)
```

```
    colormap(gca,cmap)
```

```
    c = colorbar('peer', gca);
```

```
    c.TickLabels = classNames;
```

```
    numClasses = size(cmap,1);
```

```
    c.Ticks = 1/(numClasses*2):1/numClasses:1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
c.TickLength = 0;
```

```
end
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////
```

```
function cmap = camvidColorMap()
```

```
cmap = [
```

```
    127 127 127 % bg
```

```
    153 217 234 % croc
```

```
];
```

```
cmap = cmap ./ 255;
```

```
end
```

```
////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////
```

```
////////////////////////////////////////////////////////////////
```

```
function data = augmentImageAndLabel(data, xTrans, yTrans)
```

```
% Augment images and pixel label images using random reflection and
```

```
% translation.
```

```
for i = 1:size(data,1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

tform = randomAffine2d(...
    'XReflection',true,...
    'XTranslation', xTrans, ...
    'YTranslation', yTrans);

% Center the view at the center of image in the output space while
% allowing translation to move the output image out of view.
rout = affineOutputView(size(data{i},1), tform, 'BoundsStyle', 'centerOutput');

% Warp the image and pixel labels using the same transform.
data{i,1} = imwarp(data{i,1}, tform, 'OutputView', rout);
data{i,2} = imwarp(data{i,2}, tform, 'OutputView', rout);
end
end

```