

ระบบการตรวจสอบจำนวนลูกเทนนิสในกระบวนการผลิตลูกเทนนิส

THE QUANTITY OF TENNIS BALL INSPECTION SYSTEM IN TENNIS BALL  
PRODUCTION

นายสุรวุฒิ ธรรมเอื้อวิบูลย์

SURAWUT THAMIERWIBOON

ปริญญาานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2561

ระบบการตรวจสอบจำนวนลูกเทนนิสในกระบวนการผลิตลูกเทนนิส

THE QUANTITY OF TENNIS BALL INSPECTION SYSTEM IN TENNIS BALL  
PRODUCTION



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2561

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2561  
ภาควิชา วิศวกรรมอิเล็กทรอนิกส์  
คณะ วิศวกรรมศาสตร์  
เรื่อง สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ระบบการตรวจสอบจำนวนลูกเทนนิสในกระบวนการผลิตลูกเทนนิส  
THE QUANTITY OF TENNIS BALL INSPECTION SYSTEM IN TENNIS BALL  
PRODUCTION  
ผู้จัดทำ นายสุรุฒิ ธรรมเอื้อวิบูลย์ รหัสประจำตัว 58011366

ปริญญาานิพนธ์นี้ผ่านการตรวจสอบโดยอาจารย์ที่ปรึกษาแล้ว



(อาจารย์ชินภัทร นันทจิวารชัย)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญาานิพนธ์	ระบบการตรวจสอบจำนวนลูกเทนนิสในกระบวนการผลิตลูกเทนนิส
นักศึกษา	นายสุรวุฒิ ธรรมเอื้อวิบูลย์ รหัสนักศึกษา 58011366
ปริญญา	วิศวกรรมศาสตรบัณฑิต
ภาควิชา	วิศวกรรมอิเล็กทรอนิกส์
ปีการศึกษา	2561
อาจารย์ที่ปรึกษาปริญญาานิพนธ์	อาจารย์ชินภัทร นันทจิวารักษ์

## บทคัดย่อ

ปริญญาานิพนธ์ฉบับนี้เป็นการสร้างและออกแบบการแสดงผลในขั้นตอนหนึ่งของการจำลองการผลิตลูกเทนนิส โดยกำหนดให้แสดงผลจำลองการผลิตในขั้นตอนการตรวจสอบคุณภาพลูกเทนนิส หลังจากผ่านการประกอบลูกยางกับหน้าสักหลาด ได้ทำการออกแบบอินเตอร์เฟซบนเว็บเบราว์เซอร์ และแสดงผลบน Firebase โดยจะทำหน้าที่แสดงผลจำนวนลูกเทนนิสหลังผ่านกระบวนการประกอบลูกเทนนิส, จำนวนลูกเทนนิสที่ผ่านการตรวจสอบคุณภาพ, ค่า %Yield ของกระบวนการประกอบลูกเทนนิส โดยกำหนดเป้าหมายการผลิตในขั้นตอนประกอบลูกเทนนิสจะต้องได้ค่า %Yield ของลูกเทนนิสที่ผ่านการตรวจคุณภาพ 70% ขึ้นไป และถ้าหากได้ %Yield ต่ำกว่า 70% จะทำการแจ้งเตือนไปยังสมาร์ตโฟนของผู้ดูแลการผลิตที่กำหนดไว้โดยการส่งข้อความผ่านทาง Line Notify เราสามารถเรียกดูข้อมูลการผลิตล่าสุดได้ใน Fireboard และรูปแบบไฟล์ Text เพื่อให้การเข้าถึงปัญหาเป็นไปอย่างรวดเร็วมากยิ่งขึ้น การรับข้อมูลจะเป็นการใช้ Image processing ในการประมวลผลลอจิกเพื่อใช้นับจำนวนลูกเทนนิส

Thesis Title	The Quantity of Tennis ball Inspection system in Tennis ball Production
Student	Mr. Surawut thamierwiboon Student ID. 58011366
Degree	Bachelor of Engineering
Program	Electronics Engineering
Year	2018
Thesis Advisor	Mr. Chinnapat Nantajiwakornchai

## ABSTRACT

This thesis created and designed the display in one stage of tennis ball production. By specifying demonstrates the manufacturing in quality inspection process of tennis ball that after pass tennis ball assembling between rubber ball and felt which designed interfacing on raspberry pi and display result in firebase that display the quantity of tennis balls after pass the tennis ball assembling process, the quantity of tennis balls that have passed the quality checking and the yield percentage value of tennis ball assembling process. The production goal of tennis ball assembling process must be 70% Yield. If produced less than 70% Yield will notify to the production supervisor's smartphone by sending a message via Line Notify and we can browse the latest production data in text file. To access the problems more quickly, receiving data will use Image processing to process logic to count tennis ball.

## กิตติกรรมประกาศ

โครงการ “ระบบการตรวจสอบจำนวนลูกเทนนิสในกระบวนการผลิตลูกเทนนิส” สำเร็จลุล่วงได้ดีด้วยความกรุณาจากอาจารย์ชินภัทร นันทจิวารัชย์ อาจารย์ที่ปรึกษาที่ให้ความรู้ต่างๆ เกี่ยวกับความรู้พื้นฐานในการออกแบบการทำงานของชิ้นงานและคอยให้คำแนะนำตลอดการดำเนินงาน รวมถึงเอื้อเฟื้อสถานที่ในการปฏิบัติงาน

ขอขอบพระคุณผู้ปกครองของผู้จัดทำที่เป็นผู้สนับสนุนเงินทุนในการทำโครงการ รวมไปถึงเพื่อนนักศึกษาที่ให้ความแนะนำ รวมทั้งช่วยเหลือกันแก้ปัญหาที่เกิดขึ้นในระหว่างการทำโครงการนี้ ทั้งหมดทำให้โครงการสำเร็จลุล่วงไปด้วยดี

สุดท้ายนี้ผู้จัดทำหวังว่าโครงการนี้จะเป็นประโยชน์สำหรับผู้สนใจไม่มากก็น้อย

สุรวุฒิ ธรรมเอื้อวิบูลย์



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญภาพ.....	VI
บทที่ 1 บทนำ.....	1
1.1 ความสำคัญและที่มาของปัญหาที่ทำการศึกษา.....	1
1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา.....	1
1.3 สมมุติฐานการศึกษา.....	2
1.4 ขอบเขตการศึกษา.....	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	2
บทที่ 2 งานวิจัยที่เกี่ยวข้อง.....	3
2.1 หลักการและทฤษฎีของการประมวลผลภาพดิจิทัล.....	3
2.1.1 การปรับปรุงภาพ.....	6
2.1.2 ตัวกรองแบบเกาส์เซียน.....	7
2.1.3 ปริภูมิสี RGB และปริภูมิสี HSV.....	9
2.1.4 Morphology Image Processing.....	10
2.1.5 การหาขอบภาพ.....	11
2.2 Raspberry 3 Model B+.....	13
บทที่ 3 ขั้นตอนการดำเนินงาน.....	15
3.1 ศึกษาขั้นตอนการผลิตลูกเทนนิส.....	15
3.2 โฟลวชาร์ตของการแสดงผลที่ออกแบบ.....	18

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญ (ต่อ)

	หน้า
3.3 การออกแบบโครงสร้างอุปกรณ์จำลองการผลิตลูกเทนนิส.....	19
3.4 การเขียนโค้ดเพื่อสร้างลูกจิกให้เหมาะสำหรับการแสดงผล.....	22
บทที่ 4 ผลการดำเนินงาน.....	29
4.1 วิธีการดำเนินงาน.....	29
4.2 ผลการดำเนินงาน.....	29
บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ.....	36
5.1 สรุปผลการวิจัย.....	36
5.2 ข้อเสนอแนะ.....	36
บรรณานุกรม.....	37
ภาคผนวก.....	38

## สารบัญภาพ

ภาพที่	หน้า
2.1 การแปลงภาพแอนะล็อกให้เป็นภาพดิจิทัล.....	3
2.2 ภาพแบบ Binary หรือภาพขาว-ดำ.....	4
2.3 ระดับสีของ Grayscale ตามขนาดข้อมูลที่เก็บค่าสี.....	4
2.4 ความแตกต่างของ สีแสง (RGB) และสีวัตถุ (CMYK).....	5
2.5 วงล้อสีแบบ RGB.....	5
2.6 โมเดลสี HSV ในรูปแบบโคน (Cone).....	6
2.7 ภาพที่เก็บในแต่ละ channel ในระบบสี RGB และ HSV.....	6
2.8 ตัวอย่าง mask ขนาด 3x3.....	7
2.9 การ Smoothing โดยใช้ mean filtering โดยใช้ mask ที่มีขนาดต่างๆ กัน.....	7
2.10 ตัวกรองสัญญาณรบกวนแบบเกาส์เซียนแบบ 1 มิติ.....	8
2.11 ตัวกรองสัญญาณรบกวนแบบเกาส์เซียนแบบ 2 มิติ.....	8
2.12 ภาพที่มี salt & pepper noise ที่ส่วนเบี่ยงเบนเท่ากับ 0.02.....	9
2.13 ภาพที่ใช้ตัวกรองสัญญาณรบกวนแบบเกาส์เซียน.....	9
2.14 ปริภูมิสี RGB.....	9
2.15 ปริภูมิสี HSV.....	10
2.16 การเปลี่ยนแปลงลักษณะรูปร่างหรือโครงสร้างของภาพแบบการขยาย.....	11
2.17 การเปลี่ยนแปลงลักษณะรูปร่างหรือโครงสร้างของภาพแบบการร่อนขนาด.....	11
2.18 การหาขอบภาพ Canny Edge Detector.....	12
2.19 บอร์ด Raspberry Pi 3 Model B+.....	13
2.20 ชิพ BCM2837 และ heat sink กระจายความร้อนบอร์ด Raspberry Pi 3 Model B+.....	14
2.21 Power Management Integrated Circuit บนบอร์ด Raspberry Pi 3 Model B+.....	14
3.1 โครงสร้างอุปกรณ์สำหรับจำลองการผลิตลูกเทนนิส.....	19

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3.2 Front View ของอุปกรณ์สำหรับจำลองการผลิตลูกเทนนิส.....	19
3.3 Top View ของอุปกรณ์สำหรับจำลองการผลิตลูกเทนนิส.....	20
3.4 Side View ของอุปกรณ์สำหรับจำลองการผลิตลูกเทนนิส.....	20
3.5 ภาพและคำอธิบายอุปกรณ์สำหรับจำลองการผลิตลูกเทนนิส.....	21
3.6 ภาพอุปกรณ์สำหรับจำลองการผลิตลูกเทนนิส (ชิ้นงานจริง).....	21
3.7 ภาพแบบขนาดเฟรมปกติที่กล้องสามารถแสดงผลได้.....	22
3.8 ภาพในขนาดเฟรม 480x270 หลังจากกำหนดขนาดเฟรม.....	22
3.9 ภาพที่ได้หลังจากการกรองแบบเกาส์เซียน.....	23
3.10 ภาพที่ได้หลังจากการแปลงภาพเป็นลักษณะ HSV.....	23
3.11 การกำหนดช่วงสีที่เราต้องการให้ตรวจจับ.....	24
3.12 ภาพที่ได้หลังจากการวิเคราะห์หลังจากกำหนดค่าช่วงสี.....	24
3.13 ภาพที่ได้หลังจากการวิเคราะห์ Morphology.....	25
3.14 ภาพที่ได้หลังจากการวิเคราะห์หาขอบภาพ.....	25
3.15 ภาพที่ได้หลังจากการใช้คำสั่งสร้างเส้นตรง.....	26
3.16 ภาพการสร้างวงกลมในลักษณะเป็นจุดศูนย์กลางบน Morphology.....	26
3.17 ทดลอง print(count) นับจำนวนลูกเทนนิส.....	27
3.18 ข้อมูลที่ถูกบันทึกในไฟล์ ddptm.txt.....	28
4.1 การกำหนด Network Bridge ให้อยู่วงเดียวกัน.....	29
4.2 VNC viewer.....	30
4.3 การกำหนดค่าช่วงสี HSV ของลูกเทนนิสจากคำสั่ง Trackbar.....	30
4.4 การนำค่าช่วงสี HSV จากคำสั่ง Trackbar ไปใช้ตรวจจับสีลูกเทนนิส.....	31

## สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4.5 ภาพที่แสดงถึงหลักการนับลูกเทนนิส(1).....	31
4.6 ภาพที่แสดงถึงหลักการนับลูกเทนนิส(2).....	32
4.7 การแสดงผลจำนวนลูกเทนนิสและ %Yield บน Firebase.....	32
4.8 ค่า count, count2 ที่ถูกบันทึกไว้หลังจากใช้คำสั่ง stop code.....	33
4.9 ค่า count หลังจาก stop code แล้ว Run code ใหม่.....	33
4.10 ค่า count, count2 หลังจากสั่ง stop code แล้ว Run ใหม่.....	34
4.11 ค่า count, count2 ที่บันทึกเป็น txt หลัง stop code แล้ว Run code ใหม่.....	34
4.12 ค่า count, count2 บน Firebase หลัง stop code แล้ว Run code ใหม่.....	35

## บทที่ 1

### บทนำ

#### 1.1 ความสำคัญและที่มาของปัญหาที่ทำการวิจัย

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์ สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ มีนโยบายให้ทางนักศึกษา นำความรู้ที่ได้รับจากการศึกษาและประสบการณ์มาประดิษฐ์อุปกรณ์อิเล็กทรอนิกส์ที่สามารถใช้งานได้จริง เป็นการแสดงผลถึงทักษะความสามารถของนักศึกษา และเป็นการทำให้นักศึกษาสามารถนำความรู้ที่ได้ไปใช้ได้ในอนาคตได้อย่างถูกต้องและเหมาะสม

ในช่วงสองทศวรรษที่ผ่านมาเทคโนโลยีคอมพิวเตอร์และการสื่อสารเข้ามามีบทบาทในชีวิตประจำวันของเราอย่างมาก เทคโนโลยีคอมพิวเตอร์และการสื่อสารที่ได้รับการพัฒนาขึ้นจึงมีบทบาททางด้านอุตสาหกรรมด้วยเช่นกัน อุตสาหกรรมเกิดการเติบโตอย่างรวดเร็วจากการพัฒนาทางเทคโนโลยี ผมจึงพิจารณาที่จะออกแบบการแสดงผลการผลิตของอุตสาหกรรมหนึ่งที่สามารถแสดงผลผ่านสมาร์ตโฟนได้ และได้นำอุตสาหกรรมการผลิตลูกเทนนิสมาเป็นต้นแบบ ซึ่งผมออกแบบการแสดงผลการผลิตลูกเทนนิสในขั้นตอนการตรวจสอบจำนวนลูกเทนนิสที่ได้คุณภาพภายหลังจากผ่านการประกอบหน้าสักหลาดกับลูกยาง โดยนำกล้องและหลักการทาง image processing มาใช้ในการตรวจนับลูกเทนนิส การแสดงผลการผลิตลูกเทนนิสจะแสดงผลเป็นจำนวนลูกก่อนผ่านการตรวจสอบคุณภาพ (หลังขั้นตอนการประกอบหน้าสักหลาดกับลูกยาง), แสดงผลจำนวนหลังผ่านการตรวจสอบคุณภาพ และค่า %Yield ในขั้นตอนการประกอบหน้าสักหลาดกับลูกยาง และมีการส่งข้อความแจ้งเตือนผ่านทางแอปพลิเคชันไลน์เมื่อค่า %Yield ต่ำกว่า 70% ในทุกเวลา 1 ชั่วโมง ทำให้ข้อมูลเข้าถึงผู้ดูแลการผลิตนี้ได้เร็วขึ้น ซึ่งจะช่วยให้สามารถสังเกตเห็นปัญหาและเข้าถึงปัญหาการผลิตได้รวดเร็วมากยิ่งขึ้น

#### 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1.2.1 สามารถนำความรู้จากภาคทฤษฎีและปฏิบัติที่ได้รับจากการศึกษาโครงการนี้มาผนวกใช้ในการปฏิบัติงานจริงได้อย่างถูกต้องและเหมาะสม

1.2.2 ศึกษาหลักการของ Image Processing และสามารถนำหลักการนี้ไปประยุกต์ใช้ได้

1.2.3 ศึกษาการโค้ด RaspberryPi ด้วยภาษา python และการเขียนคำสั่งแสดงผลบน Firebase realtime database

1.2.4 ศึกษาการทำงานของ RaspberryPi 3 Model B+

1.2.5 อุปกรณ์สามารถตรวจนับจำนวนลูกเทนนิสได้, สามารถแสดงผลบน Firebase ได้ และสามารถส่งข้อความแจ้งเตือนไปยังแอปพลิเคชัน Line ได้

### 1.3 สมมุติฐานการศึกษา

เมื่อลูกเทนนิสเคลื่อนที่บนอุปกรณ์จำลองขั้นตอนการตรวจสอบคุณภาพลูกเทนนิส กล้องจะสามารถตรวจจับลูกเทนนิสและนับจำนวนได้ Fireboard สามารถแสดงผลจำนวนลูกเทนนิสและค่า %Yield ได้ สามารถส่งข้อความแจ้งเตือนผ่านทางแอปพลิเคชันไลน์ได้เมื่อค่า %Yield ต่ำกว่า 70% ต่อชั่วโมง และสามารถเรียกดูข้อมูลการผลิตได้ในรูปแบบไฟล์ Text และคาดหวังว่าความรู้ที่ได้จากการทำโปรเจกต์นี้สามารถนำไปประยุกต์ใช้และเป็นแนวทางบนบทบาทหน้าที่ในอนาคตได้

### 1.4 ขอบเขตการศึกษา

ทำการศึกษา, ออกแบบ และสร้างการแสดงผลจากการจำลองการผลิตลูกเทนนิสในชั้นกระบวนการประกอบลูกเทนนิสและการตรวจสอบคุณภาพลูกเทนนิส ระบบการแสดงผลมีคุณสมบัติดังนี้

- 1.4.1 สามารถออกแบบการแสดงผลผ่าน fireboard ได้อย่างเหมาะสม
- 1.4.2 สามารถเก็บค่าข้อมูลการนับในรูปแบบไฟล์ Text และสามารถเปิดข้อมูลได้ตลอดเวลา
- 1.4.3 เมื่ออุปกรณ์อยู่ในสภาวะหยุดทำงานจากการขาดไฟเลี้ยง ข้อมูลจำนวนก่อนหน้าจะไม่หายไป เมื่อจ่ายไฟเลี้ยงอีกครั้งสามารถนับจำนวนต่อจากค่าเดิมได้

### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 สามารถนำอุปกรณ์ที่ออกแบบไปใช้งานได้ตามความเหมาะสมของตัวอุปกรณ์
- 1.5.2 สามารถนำความรู้ที่ได้จากการศึกษาไปใช้งานได้จริง
- 1.5.3 สามารถนำโครงงานนี้ไปพัฒนาต่อเพื่อการใช้งานได้เพิ่มเติม
- 1.5.4 สามารถนำความรู้ที่ได้รับศึกษาการทำงานของ RaspberryPi 3 Model B+ และหลักการเกี่ยวกับ Image Processing ไปพัฒนาต่อได้ในอนาคต

## บทที่ 2

### งานวิจัยที่เกี่ยวข้อง

#### 2.1 หลักการและทฤษฎีของการประมวลผลภาพดิจิทัล

การประมวลผลภาพดิจิทัลเกี่ยวข้องกับการแปลงข้อมูลรูปที่เป็นสัญญาณแอนะล็อกให้อยู่ในรูปของสัญญาณดิจิทัล เพื่อใช้ในการประมวลผลผ่านทางคอมพิวเตอร์ได้ และยังสามารถนำมาใช้ในการลดปัญหาของภาพ เช่น ลดสัญญาณรบกวนภายในภาพ เป็นต้น

ในการแปลงภาพให้เป็นสัญญาณดิจิทัลนั้น ระบบจะนำรูปที่รับเข้ามาไปคำนวณ โดยกระบวนการ Sampling และ Quantization และส่งข้อมูลออกมาในรูปแบบดิจิทัล คอมพิวเตอร์จะเก็บข้อมูลภาพลงหน่วยความจำ โดยการจองหน่วยความจำภายในเครื่องในรูปแบบของอาร์เรย์ โดยค่าในแต่ละช่องของอาร์เรย์แสดงถึงคุณสมบัติต่างๆ ของรูปที่จุดพิกเซลนั้นๆ และตำแหน่งของช่องอาร์เรย์ก็เป็นตัวกำหนดตำแหน่งของจุดพิกเซลภายในภาพด้วย

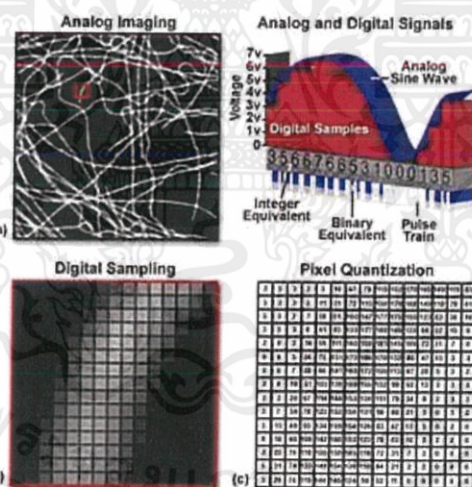


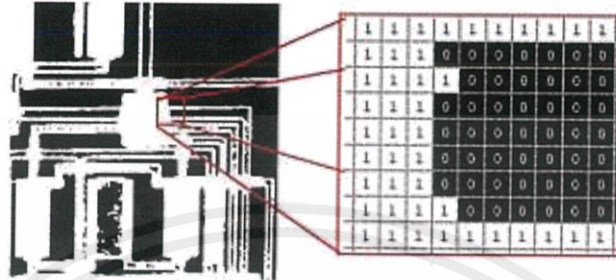
Figure 1

ภาพที่ 2.1 การแปลงภาพแอนะล็อกให้เป็นภาพดิจิทัล

ภาพดิจิทัลที่ได้จะมีรูปแบบการเก็บเป็นเมทริกซ์ ซึ่งจะมีการจัดเก็บภาพแต่ละชนิดต่างกันขึ้นอยู่กับระบบสีของภาพดังกล่าว โดยแบ่งชนิดของภาพได้ดังนี้

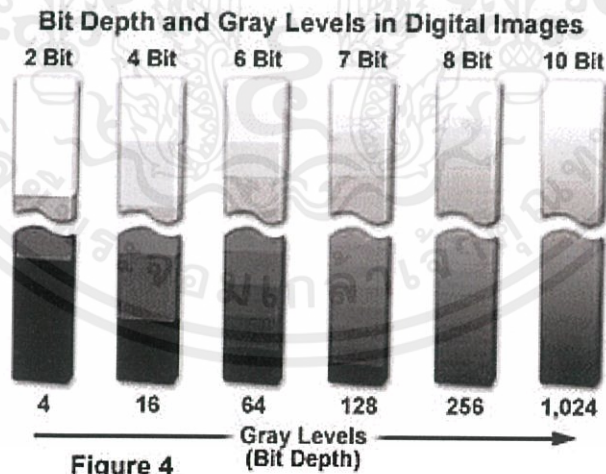
Binary image หรือภาพขาว-ดำ ในทางดิจิทัลหมายถึงว่ามีเพียง 2 สถานะคือ 0 และ 1 ซึ่งภาพในารี่ก็จะมีแค่ความเข้ม 2 ค่าเท่านั้นคือ 0 และ 1 หมายความว่า พิกเซลใดที่มีค่าเป็น 0 ก็จะมีหมายความว่าพิกเซลนั้นจะแสดงสีดำ พิกเซลใดที่มีค่าเป็น 1 ก็จะมีหมายความว่าพิกเซลนั้นจะแสดงสีขาว ซึ่ง

การแปลงภาพเกรย์สเกลเป็นภาพไบนารีนั้นจะต้องกำหนดค่าความเข้มของสีเทาที่ต้องการอ้างอิงหรือเรียกว่าค่าเทอร์ชโฮล (Threshold Value) ซึ่งค่านี้จะถูกกำหนดโดยผู้ใช้หรือว่าการใช้อัลกอริทึมในการหาค่า threshold โดยอัตโนมัติก็ได้



ภาพที่ 2.2 ภาพแบบ Binary หรือภาพขาว-ดำ

ระบบสี Grayscale เป็นช่วงของเฉดสีเทา ซึ่งแตกต่างกับภาพขาว-ดำที่มีเพียง 2 สี คือขาวกับดำ สีใน grayscale นี้แสดงถึงความเข้มของสี (Intensity) ในระดับต่างๆ โดยสีดำเป็นส่วนที่มีความเข้มของสีน้อย และสีขาวจะมีความเข้มของสีมาก จำนวนระดับของสีขึ้นอยู่กับขนาดของบิตที่ใช้เก็บค่าสี โดยทั่วไปแล้วจะเก็บข้อมูลสีประเภทนี้ด้วยข้อมูลขนาด 8 บิต หรือ 1 ไบต์ ซึ่งจะให้ความละเอียดของสีที่ 256 เฉดสี

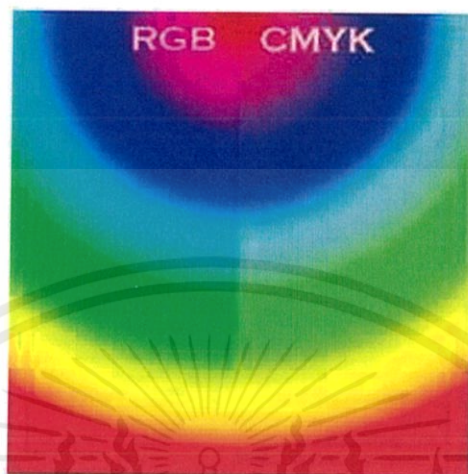


ภาพที่ 2.3 ระดับสีของ Grayscale ตามขนาดข้อมูลที่เก็บค่าสี

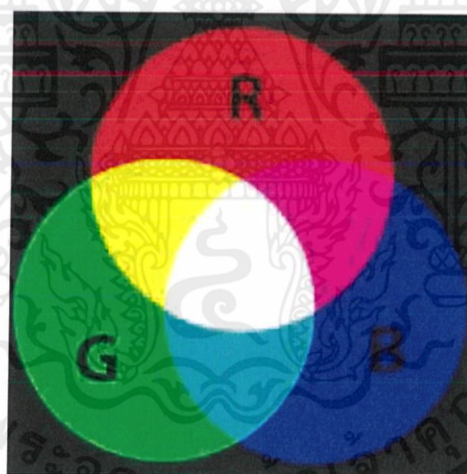
ระบบสี RGB เป็นระบบสีที่ประกอบด้วยแม่สีหลักสามสีคือ แดง(Red), เขียว(Green) และน้ำเงิน(Blue) RGB นั้นเป็นระบบสีแสง และเป็นแบบ Additive คือ ถ้าไม่มีสีใดเลยจะมองเห็นเป็นสีดำ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และในทางกลับกันหากมีครบทุกสีจะมองเป็นสีขาว จะต่างกับระบบสีแบบ Subtractive หรือระบบสีแบบ CMYK ที่เป็นสีที่เกิดจากการสะท้อนหรือเรียกกันทั่วไปว่าสีวัตถุ



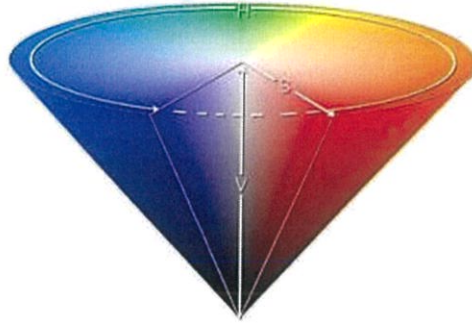
ภาพที่ 2.4 ความแตกต่างของ สีแสง (RGB) และสีวัตถุ (CMYK)



ภาพที่ 2.5 วงล้อสีแบบ RGB

ระบบสี HSV (Hue, Saturation, Value) หรือ HSB (Hue, Saturation, Brightness) เป็นระบบสีที่นิยมใช้กันในหมุ่นักแต่งภาพ เนื่องจากเป็นระบบสีที่ใกล้เคียงกับความคิดของมนุษย์ได้ดีกว่าระบบสี RGB โดย Hue คือ สีของภาพ, Saturation คือ ปริมาณความอึมตัวของสี ถ้ามีค่านี้นมากภาพจะมีสีสด ถ้ามีน้อยภาพจะยังมีสีน้อยลง จนในที่สุดจะกลายเป็นรูปที่ลักษณะแบบ Grayscale และ Value หรือ Brightness เป็นค่าที่แสดงถึงปริมาณความสว่างของภาพ หากมีค่ามากภาพจะยังมีความสว่างมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.6 โมเดลสี HSV ในรูปแบบโคน (Cone)



ภาพที่ 2.7 รูปที่เก็บในแต่ละ channel ในระบบสี RGB และ HSV

### 2.1.1 การปรับปรุงภาพ

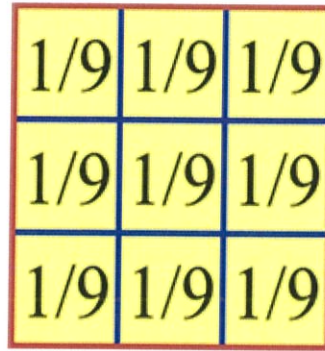
กระบวนการปรับปรุงคุณภาพของภาพเป็นการปรับปรุงภาพให้เหมาะสมกับการประมวลผลต่างๆ เนื่องจากภาพที่รับเข้ามานั้นอาจมีสัญญาณรบกวน จึงต้องปรับปรุงคุณภาพของภาพก่อนที่จะนำภาพนั้นไปประมวลผลทางคอมพิวเตอร์วิชันต่อไป ซึ่งจะทำให้ประมวลผลได้ดีขึ้น โดยทั่วไปการปรับปรุงภาพทำได้ทั้งใน spatial domain และ frequency domain แต่ในโครงการนี้ใช้การปรับปรุงภาพใน spatial domain ซึ่งมีจุดประสงค์เพื่อ smooth ภาพ การ smoothing นั้นทำได้โดยใช้การ convolution ซึ่งเป็นการนำ mask กวาดไปบนภาพที่ละจุดภาพจากซ้ายไปขวา และจากบนลงล่าง ตามสมการ

$$w(x, y) * f(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x - s, y - t)$$

โดยที่  $w(x, y)$  คือ mask ขนาด  $m \times n$

$f(x, y)$  คือ input

$$a = \frac{(m-1)}{2} \text{ และ } b = \frac{(n-1)}{2}$$



ภาพที่ 2.8 ตัวอย่าง mask ขนาด 3x3

พิจารณารูปข้างล่าง จะพบว่าขนาดของ mask ส่งผลต่อภาพ output เป็นอย่างมาก



รูปที่ 2.9 การ Smoothing โดยใช้ mean filtering โดยใช้ mask ที่มีขนาดต่างๆ กัน

### 2.1.2 ตัวกรองแบบเกาส์เซียน

เป็นตัวกรองสัญญาณรบกวนที่มีคุณลักษณะคล้ายระฆังคว่ำ ใช้สำหรับลดสัญญาณรบกวน (Noise) และลบความคมของรูปภาพ

สมการตัวกรองสัญญาณรบกวนแบบเกาส์เซียนแบบ 1 มิติ มีสมการดังนี้

$$G(x) = \sqrt{\frac{a}{\pi}} \times e^{-a \times x^2}$$

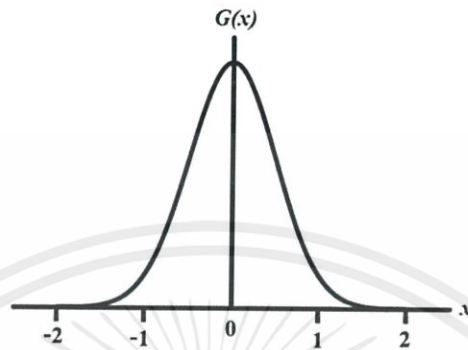
หรือเขียนได้จากการใช้พารามิเตอร์ส่วนเบี่ยงเบน คือ

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

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ  $x =$  ค่าตัวแปรในแกน  $x$

$\sigma =$  ค่าส่วนเบี่ยงเบน



ภาพที่ 2.10 ตัวกรองสัญญาณรบกวนแบบเกาส์เซียนแบบ 1 มิติ

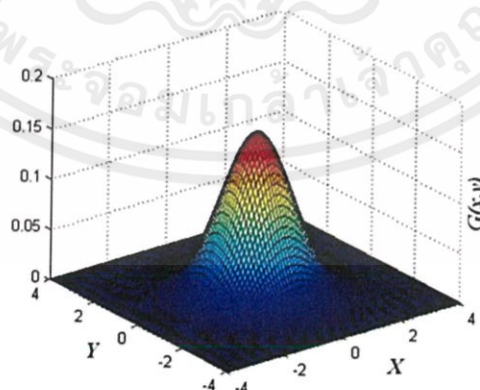
สมการตัวกรองสัญญาณรบกวนแบบเกาส์เซียนแบบ 2 มิติมีสมการดังนี้

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

เมื่อ  $x =$  ค่าตัวแปรในแกน  $x$

$y =$  ค่าตัวแปรในแกน  $y$

$\sigma =$  ค่าส่วนเบี่ยงเบน

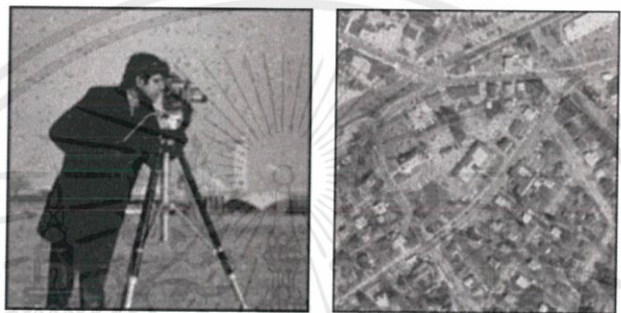


ภาพที่ 2.11 ตัวกรองสัญญาณรบกวนแบบเกาส์เซียนแบบ 2 มิติ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



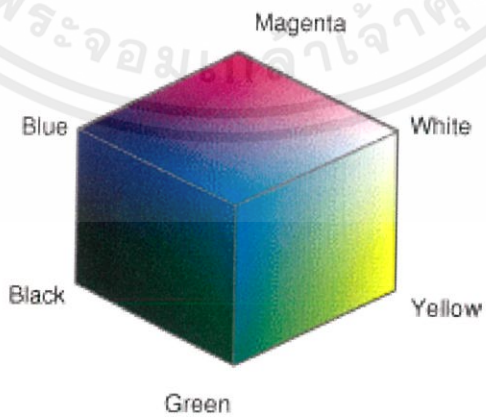
ภาพที่ 2.12 ภาพที่มี salt & pepper noise ที่ส่วนเบี่ยงเบนเท่ากับ 0.02



ภาพที่ 2.13 ภาพที่ใช้ตัวกรองสัญญาณรบกวนแบบเกาส์เซียน

### 2.1.3 ปริภูมิสี RGB และปริภูมิสี HSV (HSV Color space and HSV Color space)

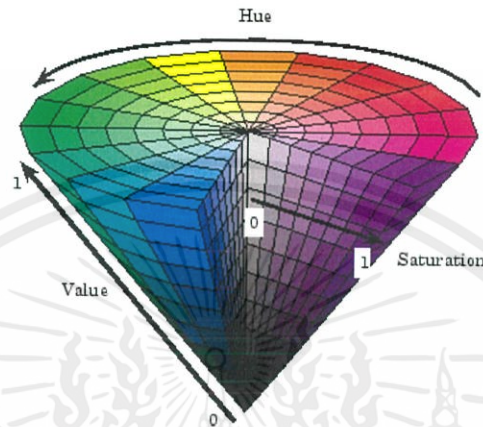
ปริภูมิสี RGB เป็นปริภูมิสีที่ใช้กันอย่างแพร่หลายในระบบภาพดิจิทัลคอมพิวเตอร์ โดยแทนค่าสีของจุดภาพแต่ละจุดภาพด้วยเวกเตอร์สามมิติ ซึ่งแทนค่าสีปฐมภูมิได้แก่ สีแดง, สีเขียว และ สีน้ำเงิน โดยมีช่วงของค่าสีระหว่าง 0-255 ซึ่งปริภูมิสี RGB จะมีลักษณะเป็นรูปลูกบาศก์ ดังรูป



ภาพที่ 2.14 ปริภูมิสี RGB

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริภูมิสี HSV ใช้หลักการแยกความสว่างออกจากเนื้อสีของจุดภาพ โดยปริภูมิสี HSV นั้นแทนค่าด้วยเวกเตอร์สามมิติ ซึ่งประกอบด้วย H แทนค่าเนื้อสี (Hue), S แทนค่าความอิ่มตัวของสี (Saturation) และ V แทนค่าความสว่างของแสง (Value) ซึ่งปริภูมิสี HSV จะมีลักษณะเป็นรูปกรวยดังรูป



ภาพที่ 2.15 ปริภูมิสี HSV

การแปลงปริภูมิสี RGB มาเป็นปริภูมิสี HSV ทำได้โดย

$$H = \begin{cases} \delta, & B \leq G \\ 2\pi - \delta, & \text{otherwise} \end{cases}$$

$$\text{โดยที่ } \delta = \cos^{-1} \left( \frac{(R-G)+(R-B)}{2\sqrt{(R-G)^2+(R-B)(G-B)}} \right)$$

$$S = 1 - 3 \cdot \frac{\text{Min}(R, G, B)}{R + G + B}$$

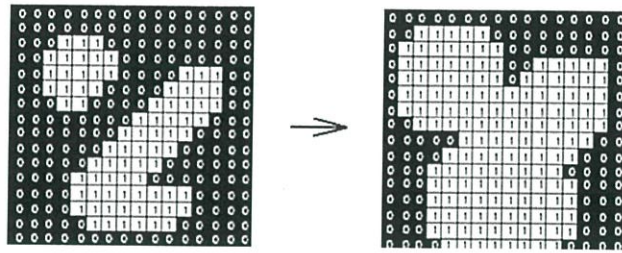
$$V = \frac{R + G + B}{3}$$

#### 2.1.4 Morphology Image Processing

เป็นการประมวลผลภาพโดยการเปลี่ยนแปลงลักษณะรูปร่างหรือโครงสร้างของภาพ โอเปอเรชันพื้นฐานโดยทั่วไปได้แก่ การ Dilation, Erosion และ Skeleton โดยการ Dilation คือการขยายภาพโดยมีสัดส่วนเท่ากันทั่วทั้งภาพ การ Erosion คือการย่อภาพ ส่วนการทำ Skeleton เป็นการหาโครงสร้างหลักของวัตถุ

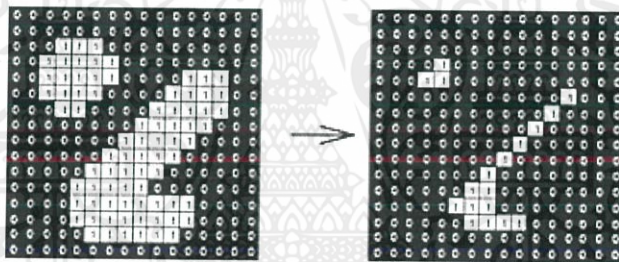
การขยาย (Dilation) จะพิจารณาข้อมูลภาพซึ่งเป็นภาพขาว-ดำ เป็นการขยายภาพให้ใหญ่ขึ้นเพื่อเพิ่มสีให้กับวัตถุที่แสดงผลในขั้นตอนสุดท้าย ซึ่งการขยายวัตถุจะทำได้โดยการกำหนดส่วนประกอบโครงสร้าง (Structuring element) และนำส่วนประกอบโครงสร้างไปกระจายบนข้อมูลภาพตามลำดับตลอดทั้งภาพ โดยเมื่อจุดเริ่มต้นของส่วนประกอบโครงสร้างหรือจุดกำเนิดตรงกับตำแหน่งข้อมูลภาพที่เท่ากับ 1 จะทำการยูเนียนส่วนประกอบโครงสร้าง เข้ากับข้อมูลภาพดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 2.16 การเปลี่ยนแปลงลักษณะรูปร่างหรือโครงร่างของภาพแบบการขยาย

การกร่อนขนาด (Erosion) เป็นการกร่อนขนาดบริเวณขอบของวัตถุ ซึ่งการกร่อนมีวิธีคล้ายกับการขยายคือ สร้างส่วนประกอบโครงร่างขึ้นมาแล้วนำไปกระจายตามข้อมูลภาพ โดยจะเลื่อนไปทุกตำแหน่งเปรียบเทียบกับข้อมูลภาพ ถ้าข้อมูลมีค่าเหมือนกับส่วนประกอบโครงร่างจะทำการกำหนดค่าข้อมูลภาพที่ตรงกับตำแหน่งที่ตรงกับจุดเริ่มต้นหรือจุดกำเนิดของส่วนประกอบโครงร่างให้เท่ากับ 1



ภาพที่ 2.17 การเปลี่ยนแปลงลักษณะรูปร่างหรือโครงร่างของภาพแบบการกร่อนขนาด

### 2.1.5 การหาขอบภาพ (Edge Detection)

ขอบภาพ คือเส้นที่แบ่งระหว่างวัตถุกับพื้นหลัง หรือ แยกวัตถุสองวัตถุออกจากกัน ทำให้เห็นลักษณะ, รูปร่าง และรายละเอียดที่ชัดเจนยิ่งขึ้นจากพื้นหลัง การหาขอบภาพจะเป็นการประมวลผลจากพิกเซลข้างเคียงโดยวัดจากการเปลี่ยนแปลงความเข้มแสงที่เกิดขึ้น กล่าวคือขอบภาพจะเห็นได้ชัดถ้าค่าการเปลี่ยนแปลงของความเข้มแสง (Intensity) มีค่ามาก ในทางตรงกันข้ามหากค่าการเปลี่ยนแปลงของความเข้มแสงมีค่าน้อย ขอบภาพก็จะเห็นได้ไม่ชัดเจน

Gradient method วิธีนี้จะหาขอบโดยการหาจุดต่ำสุดและจุดสูงสุดในรูปของอนุพันธ์อันดับหนึ่งของภาพ โดยจุดที่เป็นขอบจะอยู่ในส่วนที่เหนือค่า threshold จึงอาจทำให้เส้นขอบที่ได้มีลักษณะหนา ตัวอย่างวิธีการหาขอบของกลุ่มนี้ เช่น Roberts, Prewitt, Sobel และ Canny เป็นต้น

Laplacian method จะหาขอบโดยใช้อนุพันธ์อันดับ 2 โดยใช้จุดที่ค่า  $y$  เป็น 0 (Zero crossing) ซึ่งวิธีนี้จะใช้เวลาในการคำนวณมากกว่า Gradient method ตัวอย่างวิธีการหาขอบของกลุ่มนี้ เช่น Laplacian of Gaussian และ Marrs-Hildreth เป็นต้น

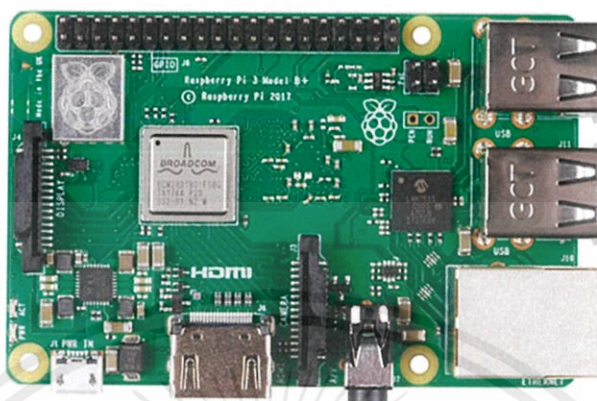
Canny Edge Detector คือวิธีการหาขอบของภาพโดยใช้ algorithm หลายขั้นตอนเพื่อให้ได้ขอบของภาพในหลายๆช่วง สร้างโดย John F. Canny ขั้นตอนในการทำมีดังต่อไปนี้

1. Gaussian Filter ทำการ blur ภาพ โดยการใช้ Gaussian Filter เพื่อทำการลด noise ของภาพต้นฉบับ
2. Gradient Operation ซึ่งในที่นี้จะใช้วิธี Sobel เพื่อทำการหา Gradient Direction และ Gradient Magnitude
3. Non-Maximum เป็นขั้นตอนที่จะทำการแบ่งชนิดของแต่ละ pixel โดยดูจาก Gradient Direction หลังจากนั้นนำค่า Gradient Magnitude ของ pixel ซ้ำงเคียงมาเปรียบเทียบเพื่อคัดเลือกขอบของภาพ ทำให้ได้ขอบที่คมชัดและบางยิ่งขึ้น
4. Double Threshold คือ การกำหนดค่า threshold สองค่า ได้แก่ Low threshold และ high threshold เพื่อทำการคัดเลือก pixel ที่เป็นเส้นขอบ



ภาพที่ 2.18 การหาขอบภาพ Canny Edge Detector

## 2.2 RaspberryPi 3 Model B+



ภาพที่ 2.19 บอร์ด Raspberry Pi 3 Model B+

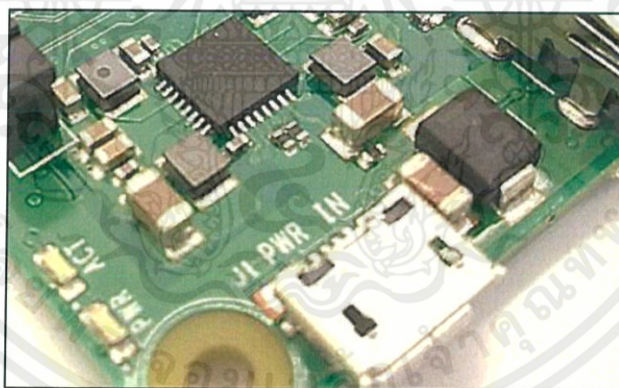
ราสเบอร์รี่พายเป็นเครื่องคอมพิวเตอร์ขนาดเล็ก ซึ่งมีขนาดเท่ากับบัตรเครดิต และมีราคาที่ถูกมากเมื่อเทียบกับคอมพิวเตอร์ส่วนบุคคล เราสามารถต่อราสเบอร์รี่พายนี้เข้ากับจอคอมพิวเตอร์หรือจอทีวีที่รองรับ HDMI หรือถ้าไม่มีพอร์ต HDMI ก็สามารถต่อผ่านสายสัญญาณวิดีโอปกติได้เช่นกัน แต่ความละเอียดอาจจะต่ำกว่า ราสเบอร์รี่พายนี้รองรับเมาส์และคีย์บอร์ดผ่าน USB port ราสเบอร์รี่พายจำเป็นต้องติดตั้งระบบปฏิบัติการ โดยระบบปฏิบัติการที่นิยมใช้คือ ระบบปฏิบัติการ Raspbian เพราะเป็นระบบปฏิบัติการที่ถูกสนับสนุนโดยตรงจากทาง Raspberry Pi Foundation

Raspberry Pi 3 Model B+ ใช้ซีพียูเป็น ARM Cortex-A53 ที่มีแกนประมวลผลจำนวน 4 แกน (Quad-Core) ทำงานที่ความถี่ 1.4 GHz ต่อแกน ใช้ BCM2837B0 แทน BCM2837 เดิม ซึ่งมาพร้อมฝาครอบทำหน้าที่เป็น heatsink ช่วยกระจายความร้อน มีส่วนช่วยทำให้รันความถี่สัญญาณนาฬิกาสูงขึ้นได้ไหวเนื่องจากความร้อนลดลง และบนบอร์ดประกอบด้วยอุปกรณ์ USB Host, Ethernet RJ-45, Audio Out และ Video Out, Full-Size HDMI, micro USB power, Camera CSI 15-pin connector, Display DSI 15-pin connector, GPIO 40-pin header ใช้ชิพ LAN7515 ซึ่งเป็น Gigabit Ethernet ทำความเร็วสูงสุดทำได้ประมาณ 300 Mbps ผลจากข้อจำกัดของตัวเองและคอขวดที่ USB ที่เข้า CPU มี Wi-Fi และ Bluetooth อยู่บนชิพตัวเดียวกัน โดยใช้ชิพ CYW43455 ของ Cypress ซึ่งจะอยู่ภายใต้ฝาครอบโลหะป้องกันสัญญาณรบกวนเพิ่มเสถียรภาพในการทำงาน



ภาพที่ 2.20 ชิพ BCM2837 และ heat sink กระจายความร้อนบอร์ด Raspberry Pi 3 Model B+

รองรับ Wi-Fi แบบ Dual-Band ทั้ง 2.4 GHz 802.11n และ 5 GHz 802.11ac อินเทอร์เน็ตผ่านทาง SDIO ส่วน Bluetooth เป็น Low Energy มาตรฐาน BLE 4.2 อินเทอร์เน็ตผ่านทาง UART รวมทั้งปรับมาใช้เสา/สายอากาศเป็นลายวงจรร (PCB Trace) ส่วนของการจัดการพลังงานได้มีการปรับปรุงให้ทำงานได้ดี จากการใช้ชิพจัดการพลังงาน Power Management Integrated Circuit: PMIC จาก Max Linear เบอร์ MXL7704 มีวงจร Buck Converter ควบคุมแรงดัน 4 ชุด ทำให้ตัวชิพที่ตรวจสอบ, ควบคุม และจัดการพลังงานได้ดียิ่งขึ้นผ่านอินเทอร์เน็ต I2C ตัว



ภาพที่ 2.21 Power Management Integrated Circuit บนบอร์ด Raspberry Pi 3 Model B+

## บทที่ 3

### ขั้นตอนการดำเนินงาน

#### 3.1 ศึกษาขั้นตอนการผลิตลูกเทนนิส

จากการศึกษาขั้นตอนการผลิตลูกเทนนิสพบว่าการผลิตลูกเทนนิสนั้นมีขั้นตอนดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



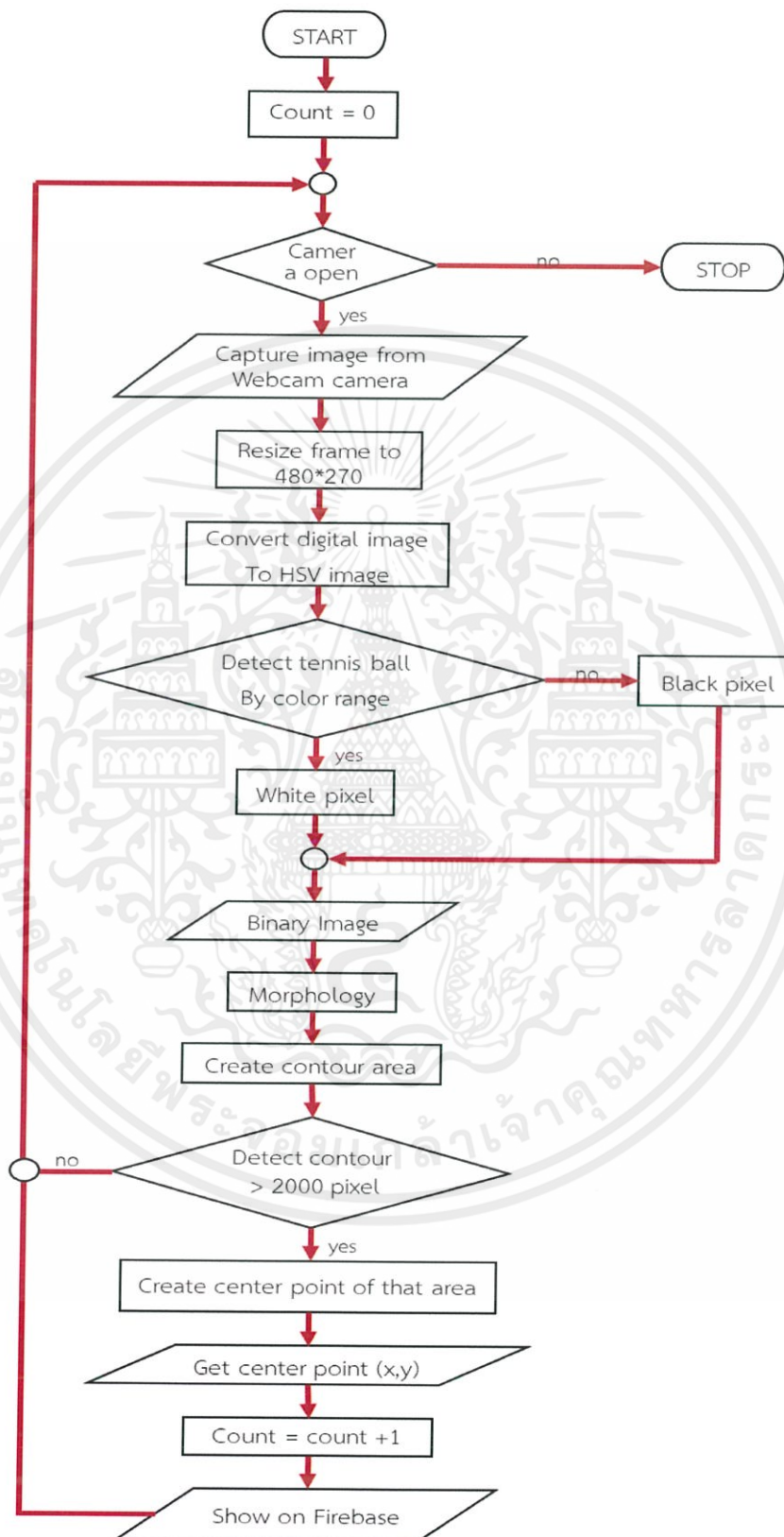
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่ทราบขั้นตอนกระบวนการผลิตลูกเทนนิสแล้ว ข้าพเจ้าเลือกนำขั้นตอนการตรวจสอบลูกเทนนิสหลังจากผ่านกระบวนการประกอบหน้าสักหลาดบนลูกยางมาออกแบบการแสดงผลจำนวนแบบเรียลไทม์ ซึ่งการแสดงผลนี้มีการเก็บบันทึกข้อมูลเป็นจำนวนของลูกเทนนิสที่ผ่านขั้นตอนประกอบหน้าสักหลาดบนลูกยางและเก็บข้อมูลจำนวนของลูกเทนนิสที่ผ่านการตรวจสอบแล้ว และมีการคำนวณบันทึกค่า %Yield ของขั้นตอนนี้ไว้ และมีการแจ้งเตือนการผลิตในทุกๆ ชั่วโมงที่มีค่า %Yield ต่ำกว่า 70% ซึ่งจะมีขั้นตอนการทำงานดังบล็อกไดอะแกรมในหัวข้อถัดไป



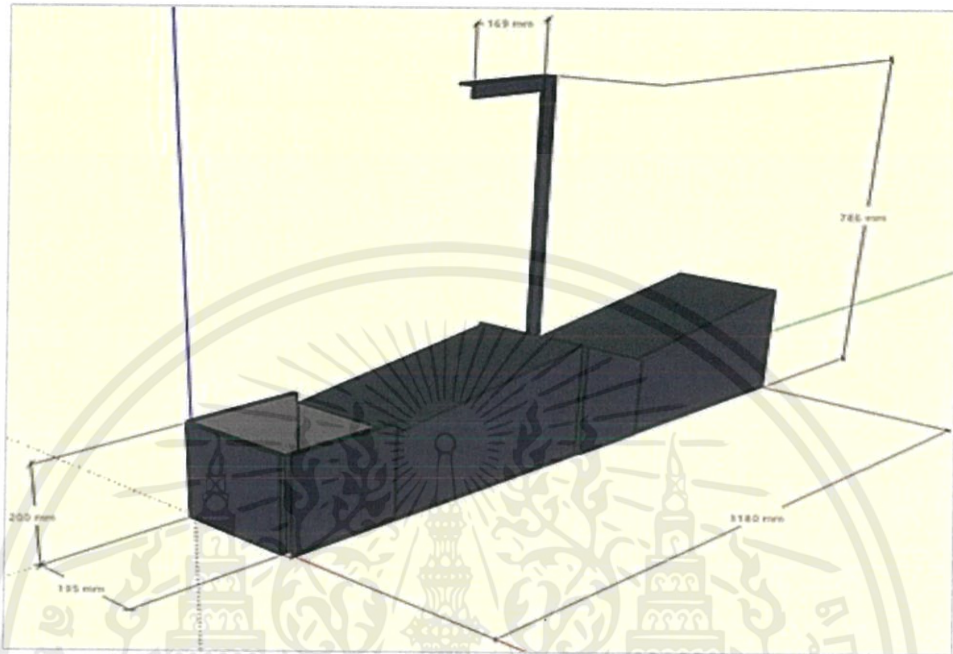
เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.2 โฟลวชาร์ตของการแสดงผลที่ออกแบบ

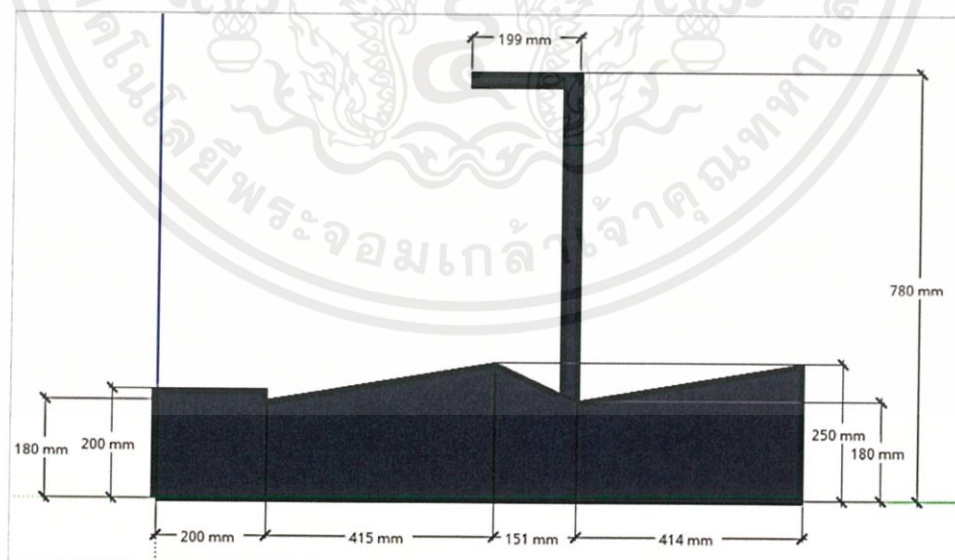


เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.3 ออกแบบโครงสร้างอุปกรณ์จำลองการผลิตลูกเทนนิสในขั้นตอนการตรวจสอบลูกเทนนิสหลังจากผ่านขั้นตอนการประกอบผ้าสักหลาดกับลูกยาง

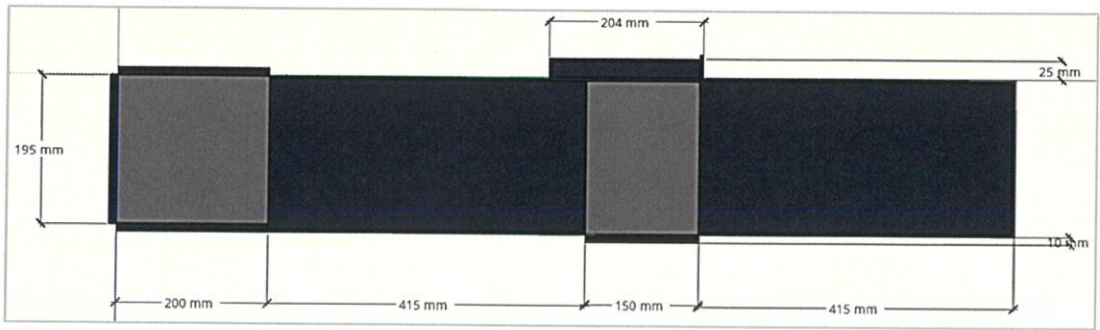


ภาพที่ 3.1 โครงสร้างอุปกรณ์สำหรับจำลองการผลิตลูกเทนนิสในขั้นตอนการตรวจสอบลูกเทนนิสหลังจากผ่านขั้นตอนการประกอบผ้าสักหลาดกับลูกยาง

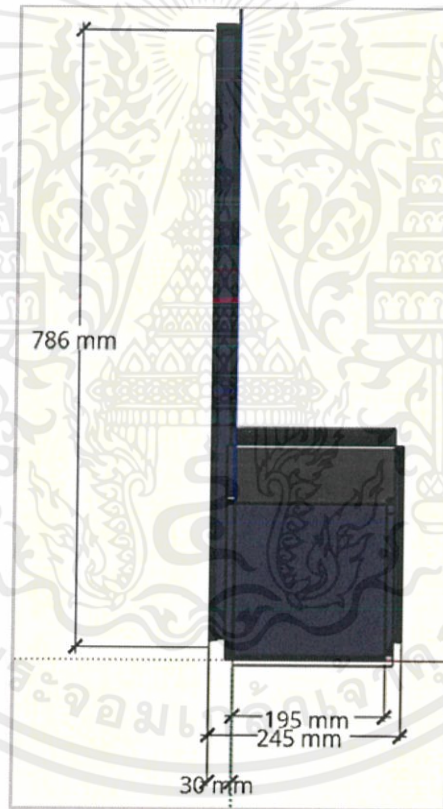


ภาพที่ 3.2 Front View ของอุปกรณ์สำหรับจำลองการผลิตลูกเทนนิสในขั้นตอนการตรวจสอบลูกเทนนิสหลังจากผ่านขั้นตอนการประกอบผ้าสักหลาดกับลูกยาง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



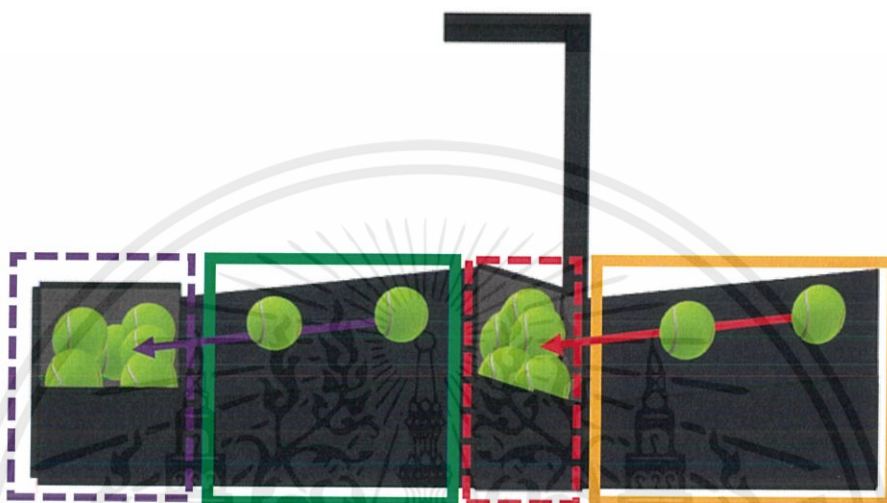
ภาพที่ 3.3 Top View ของอุปกรณ์สำหรับจำลองการผลิตลูกเทนนิสในขั้นตอนการตรวจสอบลูกเทนนิสหลังจากผ่านขั้นตอนการประกอบผ้าสักหลาดกับลูกยาง



ภาพที่ 3.4 Side View ของอุปกรณ์สำหรับจำลองการผลิตลูกเทนนิสในขั้นตอนการตรวจสอบลูกเทนนิสหลังจากผ่านขั้นตอนการประกอบผ้าสักหลาดกับลูกยาง

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ส่วนที่จำลองสายพานลำเลียงลูกเทนนิสหลังจากผ่านขั้นตอนการประกอบผ้าสักหลาดกับลูกยาง
- ส่วนที่จำลองสายพานลำเลียงลูกเทนนิสหลังจากผ่านการตรวจสอบคุณภาพลูกเทนนิส
- ส่วนที่จำลองขั้นตอนการตรวจสอบคุณภาพลูกเทนนิส
- ส่วนที่จำลองขั้นตอนการบรรจุลูกเทนนิสลงกล่องสำหรับจัดจำหน่าย



ภาพที่ 3.5 ภาพและคำอธิบายอุปกรณ์สำหรับจำลองการผลิตลูกเทนนิสในขั้นตอนการตรวจสอบลูกเทนนิสหลังจากผ่านเข้าอัดแรงดัน



ภาพที่ 3.6 ภาพอุปกรณ์สำหรับจำลองการผลิตลูกเทนนิสในขั้นตอนการตรวจสอบลูกเทนนิสหลังจากผ่านขั้นตอนการประกอบผ้าสักหลาดกับลูกยาง (ชิ้นงานจริง)

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

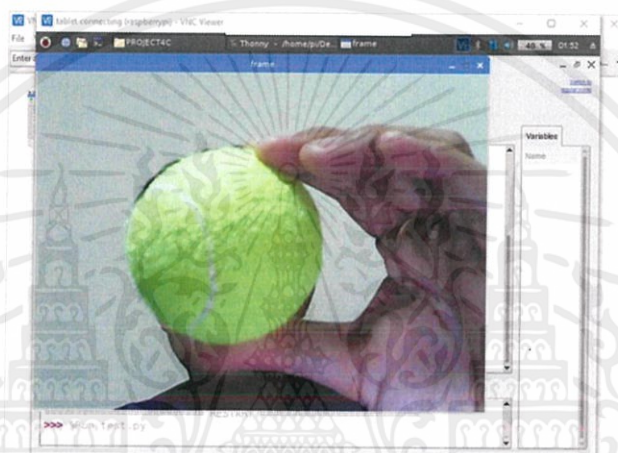
### 3.4 การเขียนโค้ดเพื่อสร้างลอจิกให้เหมาะสำหรับการแสดงผล

#### 3.4.1 การวิเคราะห์สีและลักษณะของลูกเทนนิสเพื่อนับจำนวน

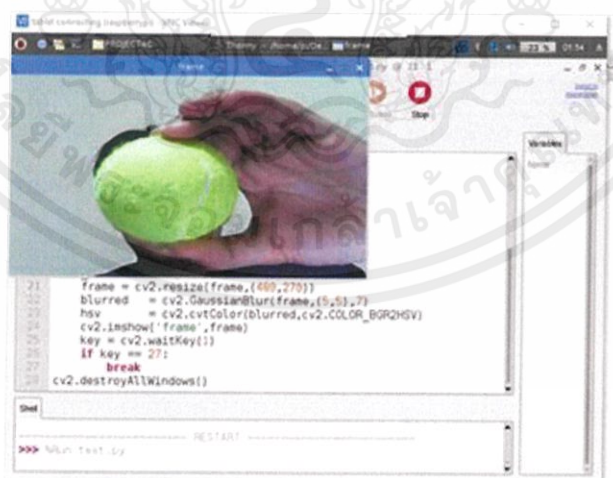
วิเคราะห์โดยการเขียนโปรแกรมด้วยภาษา Python และใช้ library OpenCV (Open Source Computer Vision)

##### 3.4.1.1 การลดขนาดเฟรม

เป็นการลดขนาดเฟรมที่ใช้ในการวิเคราะห์โดยใช้คำสั่ง `cv2.resize(frame,(480,270))`



ภาพที่ 3.7 ภาพแบบขนาดเฟรมปกติที่กล้องสามารถแสดงผลได้

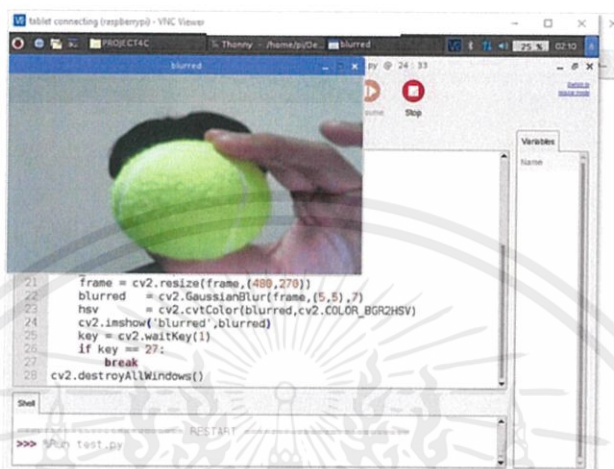


ภาพที่ 3.8 ภาพในขนาดเฟรม 480x270 หลังจากกำหนดขนาดเฟรม

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.1.2 ใช้การกรองแบบเกาส์เซียน

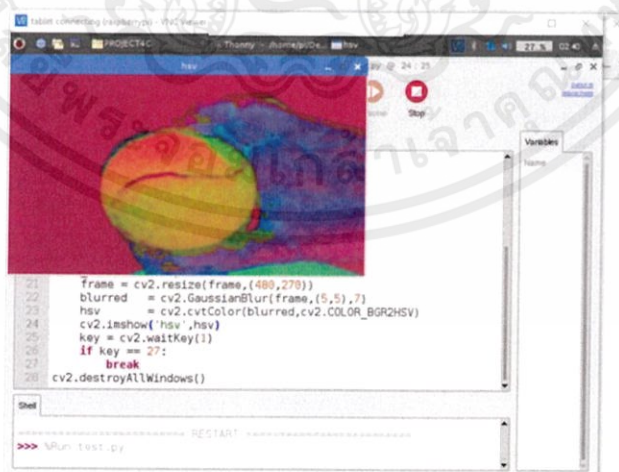
การกรองแบบเกาส์เซียนใช้สำหรับลดสัญญาณรบกวนและลดความคมของรูปภาพโดยใช้คำสั่ง `cv2.GaussianBlur(frame,(5,5),7)`



ภาพที่ 3.9 ภาพที่ได้หลังจากการกรองแบบเกาส์เซียน

### 3.4.1.3 แปลงรูปแบบภาพเป็น HSV

เป็นการแยกความสว่างออกจากเนื้อสีของภาพโดยนำเฟรมที่ได้จากการกรองแบบเกาส์เซียนมาเขียนในคำสั่ง `cv2.cvtColor(blurred,cv2.COLOR_BGR2HSV)`



ภาพที่ 3.10 ภาพที่ได้หลังจากการแปลงภาพเป็นลักษณะ HSV

### 3.4.1.4 การกำหนดช่วงสีที่เราต้องการให้ตรวจจับ

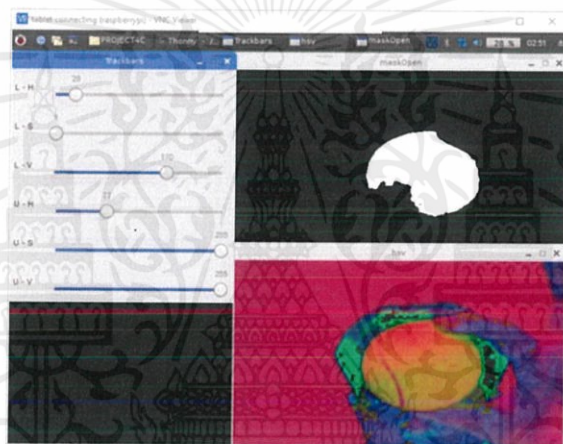
#### 3.4.1.4.1 สร้างเกจ H, S, V

ซึ่งใช้คำสั่ง `cv2.createTrackbar` เพื่อสร้างเกจ HSV ซึ่งใช้ในการควบคุมสีให้อยู่ในช่วงที่เราต้องการให้ตรวจจับ

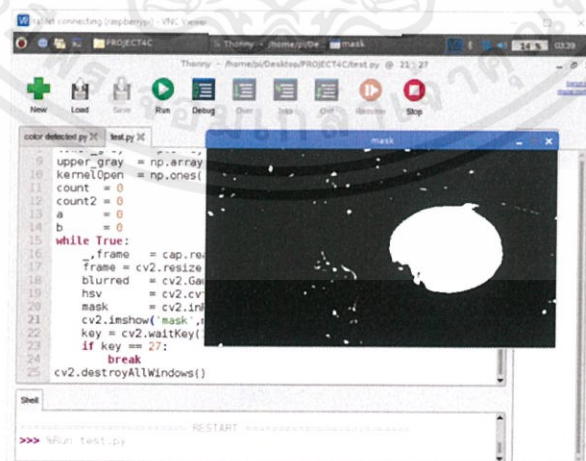
#### 3.3.1.4.2 การกำหนดค่าช่วงสี

หลังจากเราได้ช่วงสีที่เราต้องการ เราได้ค่าช่วงสีที่เราต้องการค่าที่ต่ำที่สุดเป็น (29, 1, 170) ค่าที่มากที่สุดเป็น (77, 255, 255) จะถูกนำมาใช้ในคำสั่ง

`lower_green = np.array([29,1,170])` และ `upper_green = np.array([77,255,255])`



ภาพที่ 3.11 การกำหนดช่วงสีที่เราต้องการให้ตรวจจับ

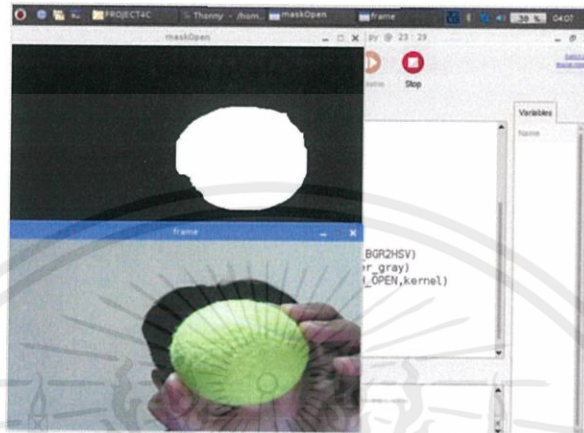


ภาพที่ 3.12 ภาพที่ได้หลังจากการวิเคราะห์หลังจากกำหนดค่าช่วงสี

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.1.5 วิเคราะห์ Morphology ให้เฟรมภาพหลังจากวิเคราะห์สีจากช่วงสี

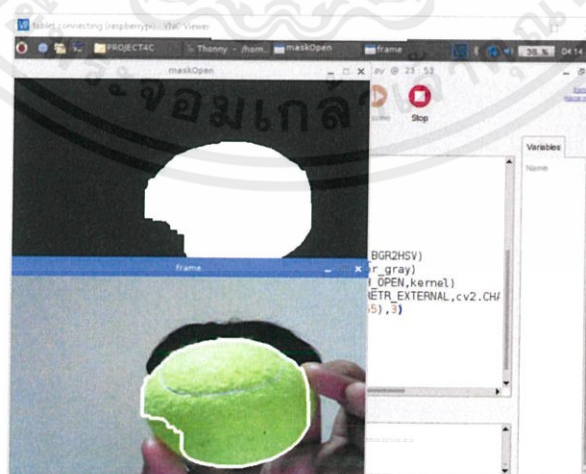
ทำการกำหนดค่า kernel = np.ones((3,3)) และใช้คำสั่ง cv2.morphologyEx เพื่อทำการดึงภาพหลังจากการวิเคราะห์สีจากช่วงสีมาสร้างภาพลักษณะ Morphology



ภาพที่ 3.13 ภาพที่ได้หลังจากการวิเคราะห์ Morphology

### 3.4.1.6 การหาขอบภาพ

การหาขอบภาพเราสามารถหาขอบภาพโดยใช้คำสั่ง cv2.drawContours( frame, conts, -1, (255,255,255), 3) ซึ่งเป็นค่าที่ได้จาก \_, conts, h = cv2.findContours (maskOpen, cv2.RETR\_EXTERNAL, cv2.CHAIN\_APPROX\_NONE)



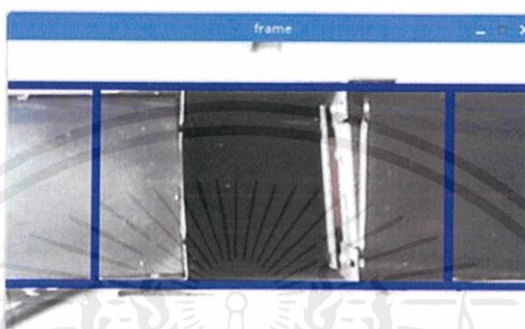
ภาพที่ 3.14 ภาพที่ได้หลังจากการวิเคราะห์หาขอบภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.1.7 สร้างเส้นเพื่อใช้ในการกำหนดเงื่อนไขในการนับ

ใช้คำสั่งสร้างเส้นตรง

```
cv2.line(frame, (0,40), (480,40), (0,255,0), 5)
cv2.line(frame, (0,220), (480,220), (0,255,0), 5)
cv2.line(frame, (70,40), (70,220), (0,255,0), 5)
cv2.line(frame, (400,40), (400,220), (0,255,0), 5)
```

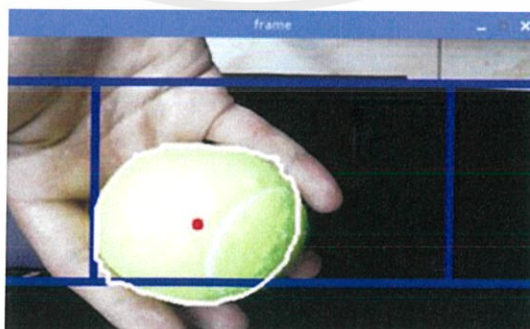


ภาพที่ 3.15 ภาพที่ได้หลังจากการใช้คำสั่งสร้างเส้นตรง

### 3.4.1.8 สร้างจุดกึ่งกลางบน Morphology ที่เราทำการวิเคราะห์ไว้

ใช้คำสั่ง for เพื่ออ่านค่าความยาวแต่ละส่วนบน Morphology ที่เราทำการวิเคราะห์ไว้ และเขียนคำสั่งสร้างวงกลมในลักษณะเป็นจุดศูนย์กลางบน Morphology กำหนดตัวแปร

```
for i in range(len(conts)):
    retval = cv2.contourArea(conts[ i ])
    x, y, w, h = cv2.boundingRect(conts[ i ])
    cv2.circle(frame, (int(x+w/2), int(y+h/2)), 5, (0,0,255), -1)
```



ภาพที่ 3.16 สร้างวงกลมในลักษณะเป็นจุดศูนย์กลางบน Morphology ที่เราทำการวิเคราะห์ไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 3.4.1.9 เขียนคำสั่งการนับและการวิเคราะห์เฉพาะส่วนที่กำหนดไว้

เมื่อจุดศูนย์กลางบน Morphology ที่เราทำการวิเคราะห์ไว้เคลื่อนที่ผ่าน  $x+w/2 > 68$  และ  $x+w/2 < 73$  และ ต้องมีค่า  $(\text{int}(y+h/2 < 220))$  ทำงานคำสั่ง `count = count+1` แปลงค่า `a = 1` ซึ่งค่า `a` เป็นค่าที่ใช้ในการกำหนดเพื่อป้องกันการนับจำนวนต่อเนื่อง

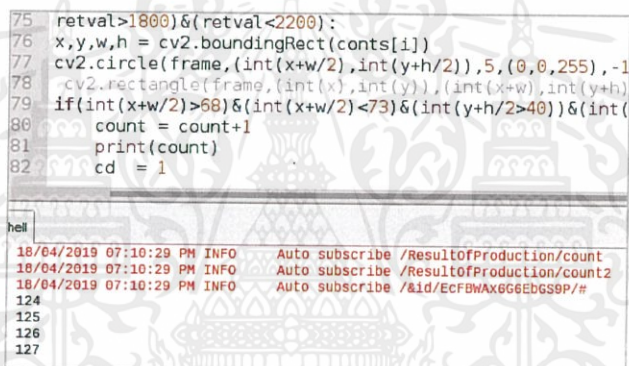
```
If (int(x+w/2)>68) & (int(x+w/2)<73) & (int(y+h/2>40)) & (int(y+h/2<220)) & (a==0):
```

```
count = count+1
```

```
a = 1
```

```
elif (int(x+w/2)>74) & (a==1):
```

```
a = 0
```



```
75 retval>1800)&(retval<2200):
76 x,y,w,h = cv2.boundingRect(conts[i])
77 cv2.circle(frame,(int(x+w/2),int(y+h/2)),5,(0,0,255),-1)
78 cv2.rectangle(frame,(int(x),int(y)),(int(x+w),int(y+h)
79 if(int(x+w/2)>68)&(int(x+w/2)<73)&(int(y+h/2>40))&(int(
80 count = count+1
81 print(count)
82 cd = 1

hell
18/04/2019 07:10:29 PM INFO Auto subscribe /ResultOfProduction/count
18/04/2019 07:10:29 PM INFO Auto subscribe /ResultOfProduction/count2
18/04/2019 07:10:29 PM INFO Auto subscribe /&id/Ecf0WAXG6E6GS9P/#
124
125
126
127
```

ภาพที่ 3.17 ทดลอง print(count) นับจำนวนลูกเทนนิส

### 3.4.2 การเขียนคำสั่ง text เพื่อการนำค่าจำนวนมาบันทึกผล

ในภาษา Python ใช้ฟังก์ชัน `open()` สำหรับเปิดไฟล์เพื่อวัตถุประสงค์ที่ต้องการ ซึ่งได้ใช้คำสั่ง `'a'` เพื่อเขียนข้อมูลต่อท้ายไฟล์เดิม ซึ่งจะเขียนค่า `count` (จำนวนลูกเทนนิสที่นับได้) ในไฟล์ `RTP.txt` เพื่อเก็บเป็นค่าข้อมูลอ้างอิงจำนวน และบันทึกข้อมูลค่า `count`, เวลา,วัน/เดือน/ปี ในไฟล์ `ddptm.txt` ดังรูป 3.18

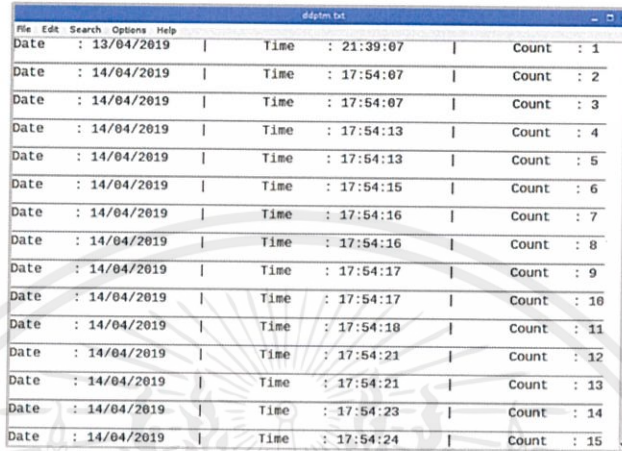
```
tc = open('/home/pi/Desktop/T.C./RTP.txt','a')
```

```
tc.write(str(count)+'\n')
```

```
tc.close()
```

```
dd = open('/home/pi/Desktop/DATA/ddptm.txt','a')
```

```
dd.write (str(time.strftime("Date\t: %d/%m/2019\t\tTime\t:
%H:%M:%S\t\tCount\t: "+str(count) + "\n_____
\n"))
dd.close()
```



Date	Time	Count
13/04/2019	21:39:07	1
14/04/2019	17:54:07	2
14/04/2019	17:54:07	3
14/04/2019	17:54:13	4
14/04/2019	17:54:13	5
14/04/2019	17:54:15	6
14/04/2019	17:54:16	7
14/04/2019	17:54:16	8
14/04/2019	17:54:17	9
14/04/2019	17:54:17	10
14/04/2019	17:54:18	11
14/04/2019	17:54:21	12
14/04/2019	17:54:21	13
14/04/2019	17:54:23	14
14/04/2019	17:54:24	15

ภาพที่ 3.18 ข้อมูลที่ถูกบันทึกในไฟล์ ddptm.txt

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### ผลการดำเนินงาน

#### 4.1 วิธีการดำเนินงาน

4.1.1 เชื่อมต่ออินเทอร์เน็ตให้กับ Raspberry Pi ซึ่งจะเชื่อมต่อทางพอร์ต LAN เข้ากับโน้ตบุ๊กที่เชื่อมกับอินเทอร์เน็ตไว้ และสร้าง Network Bridge ให้อยู่วงเดียวกัน ทำการเปิด VNC VIEWER เพื่อเช็คดูว่า Raspberry Pi เชื่อมต่อกับอินเทอร์เน็ตหรือไม่

4.1.2 ตรวจสอบกล้องและทำการปรับค่าสีที่ต้องการตรวจจับให้เหมาะสม

4.1.3 ลองนำลูกเทนนิสผ่านกล้อง และเช็คบน Firebase ว่าเกิดการนับและส่งข้อมูลตามที่โปรแกรมไว้หรือไม่

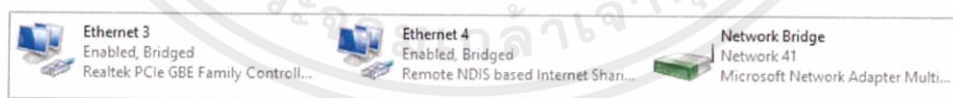
4.1.4 ใช้คำสั่ง Stop code เช็คไฟล์ Text บน Raspberry Pi ว่าเมื่อไม่มีการจ่ายไฟข้อมูลต่างๆ ที่บันทึกไว้จะถูกเก็บสมบูรณ์หรือไม่

4.1.5 Run Code อีกรอบเพื่อเช็คหาค่าจำนวนที่นับไว้ก่อน Stop code ถูกดึงมาใช้นับต่อหรือไม่

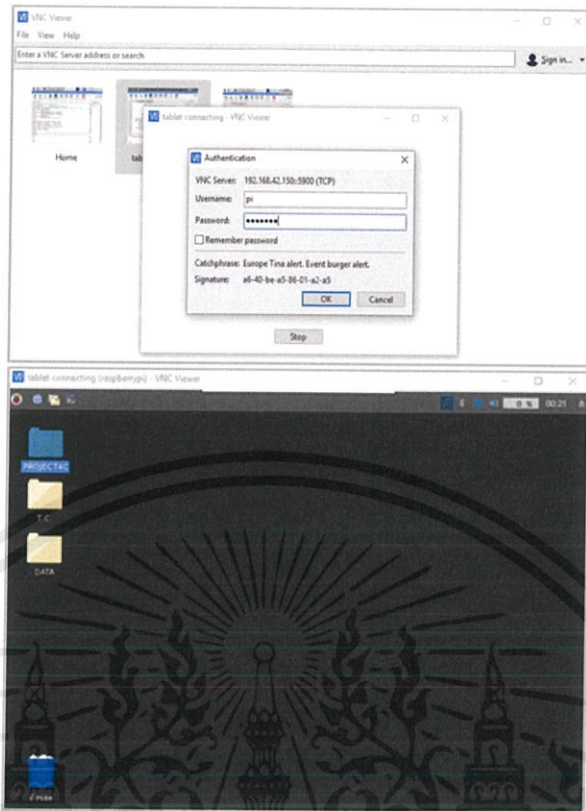
4.1.6 เช็คการแจ้งเตือนผ่านแอปพลิเคชันไลน์เมื่อค่า %Yield ต่ำกว่า 70%

#### 4.2 ผลการดำเนินงาน

4.2.1 การเชื่อมต่ออินเทอร์เน็ตให้กับ Raspberry Pi

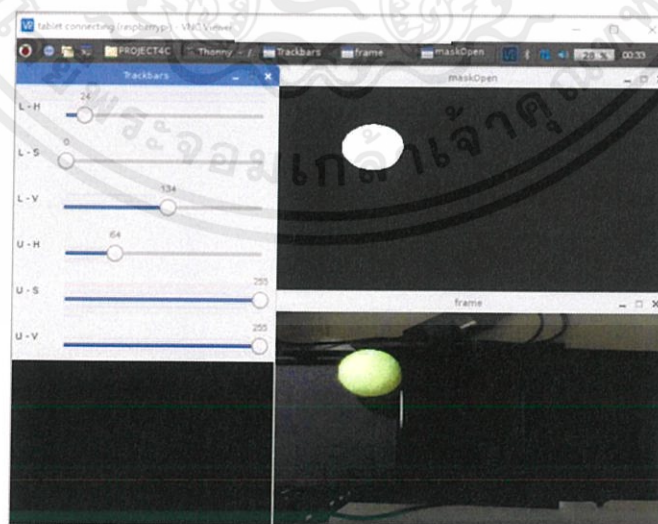


ภาพที่ 4.1 การกำหนด Network Bridge ให้อยู่วงเดียวกัน



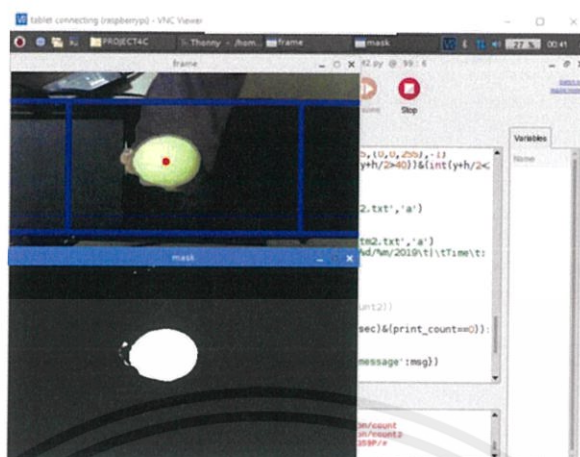
ภาพที่ 4.2 VNC viewer

4.2.2 ตรวจสอบกล่องและทำการปรับค่าสีที่ต้องการตรวจจับ (สีของลูกเทนนิส) ด้วย HSV  
Trackbar



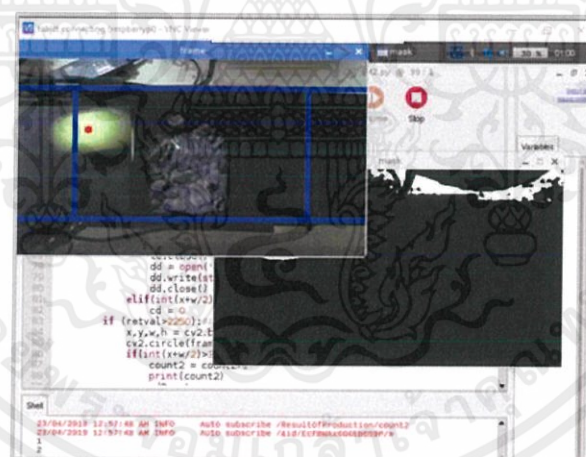
ภาพที่ 4.3 การกำหนดค่าช่วงสี HSV ของลูกเทนนิสจากคำสั่ง Trackbar

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



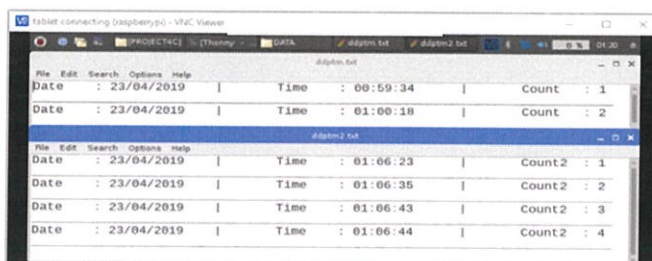
ภาพที่ 4.4 การนำค่าช่วงสี HSV จากคำสั่ง Trackbar ไปใช้ตรวจจับสีลูกเทนนิส

4.2.3 ลองนำลูกเทนนิสไปตรวจสอบบนอุปกรณ์จำลองขั้นตอนการตรวจสอบคุณภาพลูกเทนนิส หลังจากผ่านการประกอบลูกเทนนิส พบว่าสามารถนับจำนวนและแสดงผลบน Firebase ได้



ภาพที่ 4.5 เมื่อลูกเทนนิสผ่านเส้นตรงสีน้ำเงินซ้าย ค่า count จะถูกนับ ซึ่งเปรียบเสมือนลูกเทนนิสนี้ กำลังเข้าสู่กระบวนการตรวจสอบคุณภาพลูกเทนนิสหลังจากผ่านการประกอบลูกเทนนิส





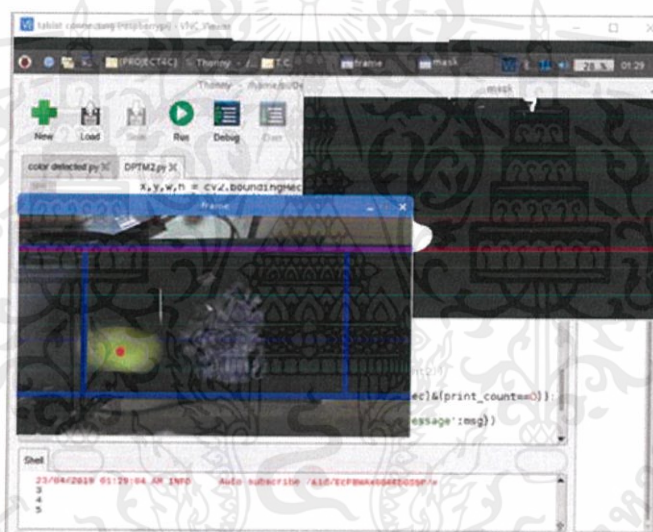
adapin.txt		
Date	Time	Count
: 23/04/2019	: 00:59:34	: 1
Date	Time	Count
: 23/04/2019	: 01:00:18	: 2

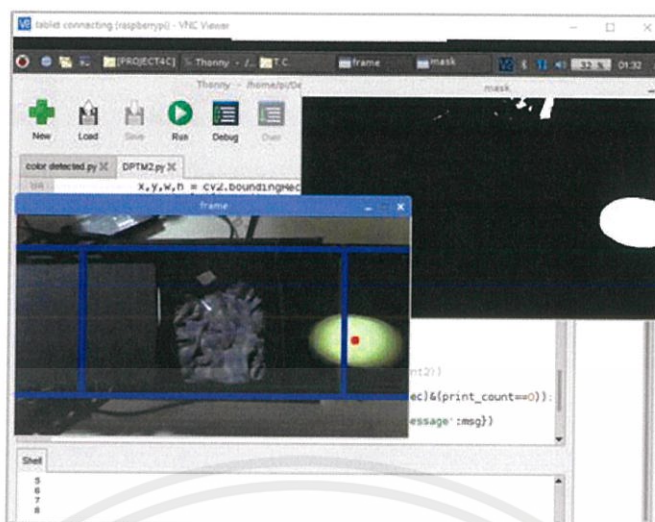
adapin2.txt		
Date	Time	Count2
: 23/04/2019	: 01:06:23	: 1
Date	Time	Count2
: 23/04/2019	: 01:06:35	: 2
Date	Time	Count2
: 23/04/2019	: 01:06:43	: 3
Date	Time	Count2
: 23/04/2019	: 01:06:44	: 4

ภาพที่ 4.8 ข้อมูลจำนวนของ count, count2 ที่ถูกบันทึกไว้หลังจากใช้คำสั่ง stop code ซึ่งบันทึกในรูปแบบไฟล์ text

4.2.5 Run Code อีกรอบเพื่อเช็คค่าจำนวนที่นับไว้ก่อน Stop code ถูกดึงมาใช้นับต่อหรือไม่



ภาพที่ 4.9 ข้อมูลจำนวนของ count ที่ถูกบันทึกไว้จะถูกนำมาใช้นับต่อหลังจาก stop code แล้ว Run code ใหม่



ภาพที่ 4.10 ข้อมูลจำนวนของ count2 ที่ถูกบันทึกไว้จะถูกนำมาใช้นับต่อหลังจาก stop code แล้ว  
Run code ใหม่

Date	Time	Count
23/04/2019	00:59:34	1
23/04/2019	01:00:18	2
23/04/2019	01:29:27	3
23/04/2019	01:29:45	4
23/04/2019	01:29:55	5

Date	Time	Count2
23/04/2019	01:06:23	1
23/04/2019	01:06:35	2
23/04/2019	01:06:43	3
23/04/2019	01:06:44	4
23/04/2019	01:31:32	5
23/04/2019	01:31:55	6
23/04/2019	01:31:58	7
23/04/2019	01:32:03	8

ภาพที่ 4.11 ข้อมูลจำนวนของ count, count2 ที่ถูกบันทึกไว้หลังจากสั่ง stop code แล้ว Run code ใหม่ ซึ่งบันทึกในรูปแบบไฟล์ text

The Quantity of Tennis ball Inspection system in The Production  
 การตรวจระบบจำนวนลูกเทนนิสในระบบการผลิตลูกเทนนิส

The Quantity of Tennis ball Inspection system Table

1:QTB-AA	2:QTB-AGC	3:Y-QC	4:Time-Date
25	4	16	05/05/2019 01:11:30
25	4	16	05/05/2019 01:11:00
25	4	16	05/05/2019 01:10:30

The Quantity of tennis ball after pass assembly between rubber ball and felt

The Quantity of tennis ball after pass Quality check

Yield of assembling between rubber ball and felt

ภาพที่ 4.12 ข้อมูลจำนวนของ count, count2 บน Firebase หลังจากสั่ง stop code แล้ว Run code ใหม่

## บทที่ 5

### สรุปผลการวิจัยและข้อเสนอแนะ

#### 5.1 สรุปผลการวิจัย

การตรวจจับลูกเทนนิสโดยใช้กล้องเว็บแคม โดยใช้หลักการตรวจจับสีในลักษณะกำหนดเป็นช่วงสี ซึ่งกำหนดค่าช่วงสีต่ำสุดและค่าช่วงสีสูงสุดจากค่าสี HSV ได้ภาพ binary image โดยพื้นที่สีขาวจะเป็นพื้นที่ที่เป็นสีที่เราทำการกำหนดช่วงสีได้สีดำคือสีที่ไม่อยู่ในช่วงสีนั้น และใช้คำสั่ง Morphology เพื่อลดค่าพิกเซลที่เป็นพื้นที่สีขาว จากการทดลองพบว่าเมื่อลูกเทนนิสเคลื่อนที่ผ่านกล้องตรวจจับติดไว้กับอุปกรณ์ลอจิกที่โปรแกรมไว้สามารถตรวจจับจำนวนลูกเทนนิสได้ และ Firebase สามารถแสดงผลจำนวนลูกเทนนิส, % Yield ในกระบวนการประกอบสักหลาดกับลูกยางได้ และสามารถเรียกดูข้อมูลจำนวนลูกเทนนิสในไฟล์ที่สร้างไว้บน raspberry pi ได้ รูปแบบของไฟล์จะเป็นรูปแบบ .txt และเมื่อขาดไฟเลี้ยงจะไม่เกิดการรีเซ็ตค่าการนับ และเมื่อมีไฟเลี้ยงอีกครั้งจะสามารถนำค่าจำนวนเดิมที่ถูกลบไว้ล่าสุดมานับต่อได้

#### 5.2 ข้อเสนอแนะ

จากการทดลองพบว่ากล้องที่ใช้มี frame rate 40 fps จึงทำให้นับจำนวนผิดพลาดเมื่อลูกเทนนิสเคลื่อนที่ด้วยความเร็วที่มาก และตัวกล้องจะมีระบบการเปิดอินฟราเรดอัตโนมัติเมื่ออยู่ในที่มืด ทำให้การตรวจจับสีผิดพลาดไปจึงต้องตั้งค่าช่วงสี HSV ใหม่บ่อยครั้งเมื่อทำการทดลอง

## บรรณานุกรม

- [1] Somkiat Udomhassakul, "Fundamentals of Digital Image Processing", May 2012.
- [2] Open Source Computer Vision, "Image processing method",  
<https://docs.opencv.org/3.2.0/>
- [3] "ความรู้พื้นฐานด้านการผลิต", 15 Dec 2008.  
<https://www.igpthai.org/CGSME/file/Corporate%20Governance.pdf>
- [4] Jakkrit Keawcharoen, " OpenCV กับการเคลื่อนที่ของObject ", February 2019  
<http://kiddyhacking.blogspot.com/2017/11/opencv-track.html>
- [5] Jirawatee, "รู้จัก Firebase Authentication", Jul 21 2016,  
<https://medium.com/firebasethailand>





เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### Code ที่ใช้ในการวิเคราะห์นับจำนวนลูกเทนนิสด้วย Image processing

```

import cv2
import numpy as np
import logging
import time
import requests
from firebase import firebase

cap = cv2.VideoCapture(0)
lower_green = np.array([14,0,185])
upper_green = np.array([100,255,255])
kernel = np.ones((3,3))
count = 0
count2 = 0
cd = 0
cd2 = 0
url = 'https://notify-api.line.me/api/notify'
token = 'u9S3WGeMKo8x16UsEkvjOkCzd6OCvBulhx2QQA1M5hW'
headers = {'content-type':'application/x-www-form-urlencoded','Authorization': 'Bearer '+token}
with open('/home/pi/Desktop/T.C./RTP.txt','r') as tc2:
    s = tc2.readlines()
    if len(s) > 0:
        last = s[len(s)-1].split('\n')
        count = int(last[0])
tc2.close()
with open('/home/pi/Desktop/T.C./RTP2.txt','r') as tc4:
    s2 = tc4.readlines()
    if len(s2) > 0:
        last2 = s2[len(s2)-1].split('\n')
        count2 = int(last2[0])
tc4.close()

while True:

```

```

_,frame = cap.read()
frame = cv2.resize(frame,(480,270))
blurred = cv2.GaussianBlur(frame,(5,5),7)
hsv = cv2.cvtColor(blurred,cv2.COLOR_BGR2HSV)
mask = cv2.inRange(hsv,lower_green,upper_green)
maskOpen = cv2.morphologyEx(mask,cv2.MORPH_OPEN,kernelOpen)
_,conts,h = cv2.findContours(maskOpen,cv2.RETR_EXTERNAL,cv2.CHAIN_APPROX_NONE)
cv2.drawContours(frame,conts,-1,(255,255,255),3)
cv2.line(frame,(0,40),(480,40),(0,255,0),5)
cv2.line(frame,(0,220),(480,220),(0,255,0),5)
cv2.line(frame,(70,40),(70,220),(0,255,0),5)
cv2.line(frame,(400,40),(400,220),(0,255,0),5)
for i in range(len(conts)):
    retval = cv2.contourArea(conts[i])
    if (retval>1700):
        x,y,w,h = cv2.boundingRect(conts[i])
        cv2.circle(frame,(int(x+w/2),int(y+h/2)),5,(0,0,255),-1)
        if(int(x+w/2)>68)&(int(x+w/2)<73)&(int(y+h/2>40))&(int(y+h/2<220))&(a==0):
            count = count+1
            cd = 1
            tc = open('/home/pi/Desktop/T.C./RTP.txt','a')
            tc.write(str(count)+'\n')
            tc.close()
            dd = open('/home/pi/Desktop/DATA/ddptm.txt','a')
            dd.write(str(time.strftime("Date\t: %d/%m/2019\t|\tTime\t:
%H:%M:%S\t|\tCount\t:
"+str(count)+"\n
_____
\n")))
            dd.close()
        elif(int(x+w/2)>74)&(cd==1):
            cd = 0
            if(int(x+w/2)>398)&(int(x+w/2)<403)&(int(y+h/2>40))&(int(y+h/2<220))&(b==0):
                count2 = count2+1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cd2 = 1
tc3 = open('/home/pi/Desktop/T.C./RTP2.txt','a')
tc3.write(str(count2)+'\n')
tc3.close()
dd2 = open('/home/pi/Desktop/DATA/ddptm2.txt','a')
dd2.write(str(time.strftime("Date\t: %d/%m/2019\t\tTime\t:
%H:%M:%S\t\tCount2\t:
"+str(count2)+"\n
\n")))
dd2.close()
elif(int(x+w/2)>404)&(b==1):
    b = 0
    yiel_total = 100*count2//count
    start_time = time.localtime()
    if ((start_time[5]%50==0)&(start_time[5]!=rec_time)): #4 MIN || 5 SEC
        rec_time=start_time[5]
        dt = str(time.strftime("%d/%m/19%H:%M:%S"))
        firebase.post('/Q-TB-P',{'1|QTB-AA':(count),'2|QTB-AQC':(count2),'3|Y-
QC':(yiel_total),'4|Time:Date':(dt)})
        if ((count2<50)&(count!=0)):
            print(yiel_total)
            msg = ("%Yield of Quality check less than 70 : Now %Yield =", yiel_total)
            r = requests.post(url,headers=headers,data={'message':msg})
    cv2.imshow('frame',frame)
    cv2.imshow('mask',mask)
    key = cv2.waitKey(1)
    if key == 27:
        break
cv2.destroyAllWindows()

```