การจำแนกชนิดเพลี้ยแป้งมันสำปะหลังด้วยการประมวลผลภาพ

(Classification of Mealy Bug by Image Processing)

จัดทำโดย

นางสาววิยะดา มะลิวัลย์ รหัสนักศึกษา B5203888

นายเจนวิทย์ พลเยี่ยม รหัสนักศึกษา B5224289

นางสาวประพันธ์ฤดี ปิตาระเต รหัสนักศึกษา B5241804

อาจารย์ที่ปรึกษาโครงงาน

อาจารย์ ดร.บุญส่ง สุตะพันธ์

รายงานนี้เป็นส่วนหนึ่งของการศึกษาวิชา 427499 โครงงานวิศวกรรมโทรคมนาคม

หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคม หลักสูตรปรับปรุง พ.ศ. 2545

สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

ประจำภาคการศึกษาที่ 3 ปีการศึกษา2555

การจำแนกชนิดเพลี้ยแป้งมันสำปะหลังด้วยการประมวลผลภาพ

คณะกรรมการสอบโครงงาน

(อ.ดร.บุญส่ง สุตะพันธ์)
กรรมการ/อาจารย์ที่ปรึกษาโครงงาน

(ผู้ช่วยศาสตราจารย์ ดร.พีระพงษ์ อุฑารสกุล)
กรรมการ

(ผู้ช่วยศาสตราจารย์ ดร. วิภาวี หัตถกรรม)
กรรมการ

มหาวิทยาลัยเทคโนโลยีสุรนารี

มหาวิทยาลัยเทคโนโลยีสุรนารี อนุมัติให้นับรายงานโครงงานฉบับนี้ เป็นส่วนหนึ่งของการศึกษาระดับปริญญาตรี สาขาวิชาวิศวกรรมโทรคมนาคม รายวิชา 427499 โครงงานวิศวกรรมโทรคมนาคม ประจำปีการศึกษา 2555

บทคัดย่อ

(Abstract)

เพลี้ยแป้งเป็นศัตรูพืชที่สำคัญของการปลูกมันสำปะหลังในจังหวัดนครราชสีมา เพลี้ยแป้งมันสำปะหลังที่พบในประเทศไทยมี 4 ชนิด ประกอบด้วยเพลี้ยแป้งสีชมพู เพลี้ยแป้งสีเขียว เพลี้ยแป้งลาย และเพลี้ยแป้งสีเทา แต่ละชนิดมีความรุนแรงในการระบาดแตกต่างกัน เกษตรกรส่วนใหญ่ไม่มีข้อมูลชนิดของเพลี้ยหรือไม่สามารถจำแนกชนิดเพลี้ยแป้งดังกล่าวด้วยตนเองได้ ทำให้การป้องกัน การกำจัด รวมทั้งการเฝ้าระวังอาจไม่สอดคล้องกับชนิดของเพลี้ย โครงงานนี้เสนอวิธีการจำแนกชนิดของเพลี้ยแป้งมันสำปะหลัง โดยการประมวลผลจากภาพถ่าย โดยได้ทดลองศึกษาประสิทธิของเทคนิคการประมวลผลภาพ 2 ชนิด ประกอบด้วย เทคนิคการวิเคราะห์เนื้อภาพ (Texture Analysis) และเทคนิคการวิเคราะห์ค่าสี (RGB Analysis)

เทคนิคการวิเคราะห์เนื้อภาพ (Texture Analysis) ใช้หลักการการวิเคราะห์ภาพความสัมพันธ์ของพิกเซลที่ติดกันของค่าระดับสีที่เรียกว่า Gray-level Co-Occurrence Matrix (GLCM) ซึ่งประกอบไปด้วย 8 สมการผลการศึกษาพบว่าทั้ง 8 สมการ ไม่มีสมการใดเลยสามารถแยกชนิดเพลี้ยแป้งมันสำปะหลังได้อย่างชัดเจน

เทคนิคการวิเคราะห์ค่าสี (RGB Analysis)สร้างฐานข้อมูลจุดสีของภาพที่จากภาพเพลี้ยแป้งมันสำปะหลังที่ได้จากอินเตอร์เนต เมื่อต้องการจำแนกเพลี้ยแป้งมันสำปะหลังที่ไม่ทราบชนิดจะใช้การหาระยะทางสีที่น้อยที่สุดจากจุดสีของเพลี้ยแป้งมันสำปะหลังที่ไม่ทราบชนิดกับจุดสีของเพลี้ยแป้งมันสำปะหลังในฐานข้อมูลผลการทดสอบการจำแนกชนิดเพลี้ยแป้งมันสำปะหลังด้วยเทคนิคนี้พบว่าสามารถจำแนกเพลี้ยแป้งสีชมพูออกจากเพลี้ยแป้งชนิดอื่นๆ ได้ถูกต้อง 100%เมื่อใช้ภาพจากอินเตอร์เนตในการทดสอบ แต่ไม่สามารถจำแนกชนิดของเพลี้ยแป้งลาย เพลี้ยแป้งสีเขียว และเพลี้ยแป้งเทาได้เทคนิคการวิเคราะห์ค่าสีจึงเหมาะสมสำหรับการจำแนกชนิดเพลี้ยแป้งสีชมพูเท่านั้น

นอกจากนี้ในโครงงานนี้ยังได้ทดสอบประสิทธิภาพของเทคนิคการวิเคราะห์ค่าสีโดยใช้ภาพถ่ายเพลี้ยแป้งจากแปลงของเกษตรและจากท่อนพันธ์ โดยใช้กล้องจุลทรรศน์ตั้งโต๊ะ (Stereomicroscope) และกล้องจุลทรรศน์บนโทรศัพท์พกพา (Mobile microscope) ในการถ่ายภาพเพลี้ยแป้งที่ใช้ในกรทดสอบเป็นเพลี้ยแป้งเทาทั้งหมด จากผลการทดลองพบว่า เมื่อใช้ภาพถ่ายกล้อง

จุลทรรศน์บนโทรศัพท์พกพานั้นเทคนิคการวิเคราะห์ค่าสีให้ผลสอดคล้องกับกรณีการใช้ภาพถ่ายจากอินเตอร์เนต โดยไม่ทำนายภาพเพลี้ยแป้งเทาเป็นเพลี้ยสีชมพูเลย ส่วนใช้ภาพถ่ายกล้องจุลทรรศน์ตั้งโต๊ะนั้น กลับพบว่ามีจำแนกเพลี้ยแป้งเทาผิดเป็นเพลี้ยแป้งชมพูเป็นจำนวนมาก ทั้งนี้อาจเป็นเพราะขณะถ่ายภาพเพลี้ยแป้งไม่ได้ปรับพารามิเตอร์ของกล้องถ่ายภาพให้สอดคล้องกับชนิดของแหล่งกำเนิดแสง จึงถ้าให้ภาพถ่ายมีสีที่แตกต่างจากภาพในฐานข้อมูล

กิตติกรรมประกาศ

(Acknowledgement)

จากการที่ผู้จัดทำรายงานได้รับมอบหมายให้ทำโครงงานเรื่อง การจำแนกชนิดเพลี้ยแป้งมันสำปะหลังด้วยการประมวลผลภาพ (Classification of Mealy Bug by Image Processing)ส่งผลให้ผู้จัดทำรายงานได้รับความรู้และประสบการณ์ต่างๆ เกี่ยวกับการประมวลผลภาพ (Image Processing)และการเขียนโปรแกรมด้วยโปรแกรม Borland C++ Builder 6.0 และ MATLAB เป็นอย่างมาก บัดนี้โครงงานดังกล่าวพร้อมทั้งรายงานได้สำเร็จลงแล้ว ทั้งนี้ด้วยคำปรึกษาและสนับสนุนจาก

1. ดร. บุญส่ง สุตะพันธ์อาจารย์ที่ปรึกษาโครงงาน
2. นางสาวศรินทร์ วัชรบุศราคำ ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ
3. อาคารเครื่องมือ 1 ที่อนุญาตให้ใช้กล้องจุลทรรศน์แบบตั้งโต๊ะ
4. นางสาวปนัดดา สมัยกลาง ที่อนุญาตให้ใช้กล้องจุลทรรศน์แบบพกพา

ข้าพเจ้าใคร่ขอขอบพระคุณผู้ที่มีส่วนเกี่ยวข้องทุกท่านที่มีส่วนร่วมในการให้ข้อมูลและเป็นที่ปรึกษาในการทำรายงานฉบับนี้จนเสร็จสมบูรณ์ ตลอดจนให้การดูแลและให้ความเข้าใจเกี่ยวกับพื้นฐานการใช้งานโปรแกรม ซึ่งข้าพเจ้าขอขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้ด้วย

นางสาววิยะดา มะลิวัลย์

นายเจนวิทย์ พลเยี่ยม

นางสาวประพันธ์ฤดี ปิตาระเต

มหาวิทยาลัยเทคโนโลยีสุรนารี

สารบัญ

หน้า

บทที่1 บทนำ

1.1 วัตถุประสงค์ของโครงงาน 2

1.2 ขอบเขตการศึกษาโครงงาน 3

1.3 ประโยชน์ที่คาดว่าจะได้รับ 3

บทที่2การประมวลผลภาพด้วยเทคนิคการวิเคราะห์เนื้อภาพ

2.1 หลักในการดำเนินงาน 4

2.2 ข้อมูลที่ใช้ในการดำเนินงาน 4

2.3 การคัดกรองภาพเบื้องต้น (Pre-Processing) 5

2.4เทคนิคและกระบวนการที่ใช้ในการศึกษาข้อมูล 6

2.5 ผลการศึกษาสมการวิเคราะห์เนื้อภาพ 11

บทที่ 3 การประมวลผลภาพด้วยเทคนิคการวิเคราะห์ค่าสี RGB

3.1 หลักในการดำเนินงาน 16

3.2 ข้อมูลที่ใช้ในการดำเนินงาน 17

3.3 การคัดกรองภาพเบื้องต้น (Pre-Processing) 17

3.4 เทคนิคและกระบวนการที่ใช้ในการศึกษาข้อมูล 18

3.5 ผลการทดสอบประสิทธิภาพของโปรแกรมโดยใช้ภาพถ่ายจากอินเตอร์เน็ต 24

บทที่ 4 การทดสอบประสิทธิภาพของโปรแกรมประมวลผลภาพ โดยใช้ภาพถ่ายจาก Stereo Microscope

4.1 การถ่ายภาพด้วยกล้องจุลทรรศน์ (Stereo Microscope) 26

4.2ทดสอบถ่ายภาพด้วยกล้องจุลทรรศน์ (Stereo Microscope) 27

4.3การทดสอบประสิทธิภาพโดยใช้เทคนิคการวิเคราะห์ค่าสี RGB (RGB Analysis) 29

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 5ประสิทธิภาพโปรแกรมจากภาพถ่ายเพลี้ยแป้งมันสำปะหลังด้วยกล้อง Mobile Microscope

5.1 แหล่งปลูกมันที่เก็บตัวอย่างเพลี้ยแป้งมันสำปะหลังมา 32

5.2 การถ่ายภาพเพลี้ยแป้งมันสำปะหลังด้วย Mobile Microscope 33

บทที่ 6 สรุป 35

ภาคผนวก ก 37

ภาคผนวก ข 43

ภาคผนวก ค 49

ภาคผนวก ง 52

ภาคผนวก จ 56

ภาคผนวก ฉ 60

ภาคผนวก ช 63

ตัวอย่างการใช้งานโปรแกรม 82

ภาคผนวก ซ 84

ตัวอย่างการใช้งานโปรแกรม 89

บรรณานุกรม 91

ประวัติผู้เขียน 94

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 1

บทนำ

มันสำปะหลังเป็นพืชทนแล้ง ในประเทศไทยจึงสามารถปลูกมันสำปะหลังได้ตลอดทั้งปี แต่มีบางช่วงฤดูของการปลูกที่ทำให้มันสำปะหลังได้ผลผลิตต่ำ อันเนื่องมาจากถูกรบกวนจากแมลงศัตรูพืชที่เรียกกันว่า “เพลี้ยแป้งมันสำปะหลัง ” เพลี้ยแป้งมันสำปะหลังมี 4 ชนิดด้วยกัน ได้แก่ เพลี้ยแป้งลาย ,เพลี้ยแป้งสีเทา ,เพลี้ยแป้งสีเขียวและเพลี้ยแป้งสีชมพู ในเพลี้ยแป้งมันสำปะหลังทั้ง 4 ชนิดนี้ เพลี้ยแป้งสีชมพูถือว่าเป็นศัตรูพืชที่ร้ายแรงที่สุด เพราะสร้างความเสียหายทำให้ผลผลิตของมันสำปะหลังต่ำ แต่เพลี้ยแป้งมันสำปะหลังทั้ง 4 ชนิดนี้ก็ล้วนแต่ส่งผลให้ผลผลิตของมันสำปะหลังต่ำทั้งนั้น อีกทั้งยังสร้างความเสียหายให้แก่เกษตรกรเป็นจำนวนมาก โดยมีข้อมูลจาก ศูนย์ส่งเสริมและพัฒนาอาชีพการเกษตร (พืชสวน) จ.นครราชสีมา อ.เมือง จ. นครราชสีมาสำรวจพบว่ามีการระบาดของเพลี้ยแป้งตั้งแต่เดือนเมษายน พ .ศ. 2552เป็นต้นมา และแจ้งผลการสำรวจ ในปี 2553 ซึ่งการระบาดของเพลี้ยแป้งนี้ได้สร้างความเสียหายให้แก่พื้นที่เพาะปลูกมันสำปะหลังของเกษตรกรทั่วประเทศกว่า 600,000 ไร่ คิดเป็นมูลค่าความเสียหายกว่า 700 ล้านบาทจึงมีความจำเป็นอย่างยิ่งที่จะต้องทราบว่าเพลี้ยแป้งมันสำปะหลังที่ระบาดอยู่ในแปลงปลูกมันสำปะหลังของเกษตรกรนั้นเป็นเพลี้ยแป้งมันสำปะหลังชนิดไหน เพื่อจะสามารถกำจัดเพลี้ยแป้งมันสำปะหลังชนิดนั้นได้อย่างถูกวิธีพร้อมทั้งหาวิธีป้องกันไม่ให้แพร่ระบาดไปทั่วทั้งแปลงปลูกและป้องกันไม่ให้แพร่ระบาดไปยังแปลงปลูกข้างเคียงอีกด้วย

การจำแนกชนิดของเพลี้ยแป้งมันสำปะหลังจึงเป็นปัญหาหลักที่สำคัญต่อเกษตรกรที่ปลูกมันสำปะหลังบางราย เพราะเกษตรกรบางรายนั้นเมื่อพบปัญหาเพลี้ยแป้งระบาดในแปลงปลูกของตัวเองก็จะไม่ทราบว่าเป็นเพลี้ยแป้งชนิดไหน ทำให้ไม่รู้วิธีการกำจัดและวิธีป้องกันการแพร่ระบาดที่ถูกวิธี ส่งผลให้มันสำปะหลังเกิดความเสียหาย ผลผลิตต่ำ จึงจำเป็นอย่างยิ่งที่ เกษตรกรที่ปลูกมันสำปะหลังควรจะจำแนกชนิดของเพลี้ยแป้งมันสำปะหลังได้ทุกชนิด เพื่อที่ว่าเมื่อพบการแพร่ระบาดของเพลี้ยแป้งในแปลงปลูกจะสามารถกำจัดและป้องกันเพลี้ยแป้งเหล่านั้นได้อย่างถูกวิธี

มหาวิทยาลัยเทคโนโลยีสุรนารี

โครงงานนี้เสนอการจำแนกชนิดเพลี้ยแป้งมันสำปะหลังด้วยการประมวลผลภาพ2 เทคนิค ประกอบด้วย เทคนิคการวิเคราะห์เนื้อภาพ (Texture Analysis) และเทคนิคการวิเคราะห์ค่าสี RGB (RGB Analysis)เหตุผลที่ใช้เทคนิคการวิเคราะห์เนื้อภาพเนื่องจากเพลี้ยแป้งมันสำปะหลังทั้ง 4 ชนิดมีรูปร่างและโครงสร้างภายนอกแตกต่างกัน จึงน่าจะมีความเป็นไปได้ที่จะจำแนกชนิดของเพลี้ยแป้งมันสำปะหลังได้ ในส่วนของเทคนิคการวิเคราะห์ค่าสี RGBจะใช้หลักการหาต้นแบบสีของภาพเพลี้ยแป้งมันสำปะหลังเนื่องจากเล็งเห็นว่าเพลี้ยแป้งมันสำปะหลังบางชนิด เช่น เพลี้ยแป้งสีชมพู และเพลี้ยแป้งสีเขียวมีสีแตกต่างจากเพลี้ยแป้งมันสำปะหลังชนิดอื่นๆ จึงเป็นไปได้ที่จะใช้ในการจำแนกชนิดเพลี้ยแป้งได้

อย่างไรก็ตามทั้งเทคนิคการวิเคราะห์เนื้อภาพ (Texture Analysis) และเทคนิคการวิเคราะห์ค่าสีRGB (RGB Analysis)อาจจะมีประสิทธิภาพในการจำแนกชนิดเพลี้ยแป้งมันสำปะหลังต่างกัน จึงจำเป็นต้องมีการทดสอบประสิทธิภาพของเทคนิค โดยใช้ภาพถ่ายเพลี้ยแป้งมันสำปะหลังจากแปลงปลูกมันสำปะหลังของเกษตรกร ในโครงงานนี้จะใช้ภาพถ่ายเพลี้ยแป้งมันสำปะหลัง โดยใช้อุปกรณ์ 2 ชนิด คือ กล้องจุลทรรศน์แบบตั้งโต๊ะ (Stereo microscop) และกล้องจุลทรรศน์แบบพกพา(Mobile microscop)

1.1วัตถุประสงค์ของโครงงาน

1.พัฒนาโปรแกรมประมวลผลภาพสำหรับใช้จำแนกชนิดของเพลี้ยแป้งมันสำปะหลัง

2.ทดสอบประสิทธิภาพการจำแนกชนิดเพลี้ยแป้งมันสำปะหลัง ของโปรแกรมคอมพิวเตอร์ดังกล่าว โดยใช้ภาพถ่ายเพลี้ยแป้งมันสำปะหลังที่ได้จากแปลงปลูกมันสำปะหลังของเกษตรกร

1.2ขอบเขตการศึกษาโครงงาน

1.กำหนดภาพถ่ายที่นำมาเป็นข้อมูลภาพสำหรับการจำแนกชนิดของเพลี้ยแป้งมันสำปะหลังนั้นจะต้องมีการคัดกรองข้อมูลภาพเฉพาะส่วนที่เอื้อประโยชน์ต่อการจำแนกชนิดของเพลี้ยแป้งมันสำปะหลังก่อน เช่น การตัดภาพเพื่อให้ได้เพลี้ยแป้งเพียงหนึ่งตัว หรือต้องเป็นภาพที่มีความคมชัดพอสมควร เป็นต้น

2. การคัดกรองข้อมูลภาพในเบื้องต้นจะใช้คนดำเนินการก่อนที่จะเข้าสู่การประมวลผลภาพด้วยเทคนิคการวิเคราะห์เนื้อภาพและเทคนิคการวิเคราะห์ค่าสีRGB

3. โปรแกรมประมวลผลภาพเพื่อจำแนกชนิดเพลี้ยแป้งมันสำปะหลังสามารถใช้จำแนกชนิดของเพลี้ยแป้งมันสำปะหลังได้อย่างน้อย 1 ชนิด

1.3ประโยชน์ที่คาดว่าจะได้รับ

1. ได้โปรแกรม เกี่ยวกับ การประมวลผลภาพเพลี้ยแป้ง มันสำปะหลัง ด้วยเทคนิคการวิเคราะห์เนื้อภาพและเทคนิคการวิเคราะห์ค่าสีRGB

2. ได้เรียนรู้การใช้งานโปรแกรม Borland C++ Builder 6.0, การใช้งานโปรแกรม MATLAB ในหมวดของ Image processing toolbox และการเก็บข้อมูลภาคสนามที่ต้องอาศัยผู้เชี่ยวชาญในการให้ข้อมูล

3. ได้เรียนรู้การทำงานเป็นกลุ่ม และได้ประสบการณ์ในการประยุกต์ใช้ความรู้เพื่อแก้ปัญหาจริงของท้องถิ่น รวมทั้งได้ความเข้าใจเกี่ยวกับปัญหาของเกษตรกรผู้ปลูกมันสำปะหลัง

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่2

การประมวลผลภาพด้วยเทคนิคการวิเคราะห์เนื้อภาพ

การประมวลผลภาพด้วยเทคนิคการวิเคราะห์เนื้อภาพ (Texture Analysis)เป็นเทคนิคแรกที่ผู้จัดทำโครงงานนำมาศึกษาสำหรับใช้ในการจำแนกชนิดเพลี้ยแป้งมันสำปะหลัง ซึ่งทำการประมวลผลการจำแนกในโปรแกรมที่ชื่อว่า Borland C++ Builder 6.0ที่เลือกใช้เทคนิคนี้เพราะเพลี้ยแป้งมันสำปะหลังมีลักษณะพื้นผิวบนตัวของมันแตกต่างกันอย่างเห็นได้ชัดเจน โดยขั้นตอนการประมวลผลภาพด้วยเทคนิคการวิเคราะห์เนื้อภาพจะนำเสนอตามหลักการดังต่อไปนี้

2.1 หลักในการดำเนินงาน

เพลี้ยแป้งที่พบในมันสำปะหลังทั้ง 4 ชนิด ได้แก่ เพลี้ยแป้งลาย, เพลี้ยแป้งสีเทา, เพลี้ยแป้งสีเขียวและเพลี้ยแป้งสีชมพู ดังที่กล่าวไว้ในข้างต้นแล้วนั้น โดยในแต่ละชนิดมีลักษณะพื้นผิวแตกต่างกันด้วยข้อแตกต่างนี้จะเป็นตัวแปรสำคัญที่โปรแกรมประมวลผลนี้น่าจะสามารถทำการประมวลผลเพื่อจำแนกชนิดเพลี้ยแป้งมันสำปะหลังด้วยเทคนิคการวิเคราะห์เนื้อภาพ (Texture Analysis)ได้

2.2ข้อมูลที่ใช้ในการดำเนินงาน

ในการศึกษาครั้งนี้จะใช้ภาพถ่ายที่มีอยู่ในอินเตอร์เน็ต เป็นข้อมูลสำหรับการศึกษา เรียกภาพถ่ายเหล่านี้ว่า **“ข้อมูลภาพ”**ตัวอย่างข้อมูลภาพแสดงต่อไปนี้

มหาวิทยาลัยเทคโนโลยีสุรนารี

(ก)

(ข)

(ค)

(ง)

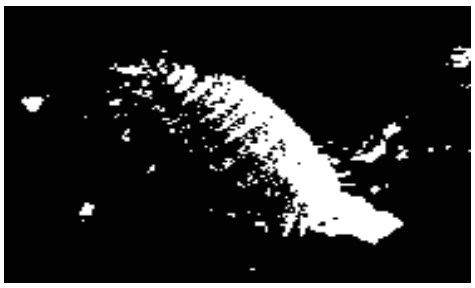
รูปที่ 2.1ภาพถ่ายเพลี้ยแป้งมันลำปะหลัง4 ชนิด (ก) เพลี้ยแป้งสีเขียว (ข) เพลี้ยแป้งลาย (ค) เพลี้ยแป้งสีชมพู และ (ง) เพลี้ยแป้งสีเทา

จากข้อมูลภาพ รูปที่ 2.1จะต้องดำเนินการเพื่อคัดกรองภาพให้เด่นชัดขึ้นต่อไป

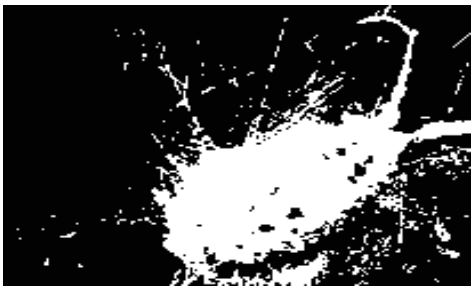
2.3การคัดกรองภาพเบื้องต้น(Pre-Processing)

เนื่องจาก ตัวอย่าง ข้อมูลภาพจากรูปที่ 2.1มีลักษณะเด่นที่ยังไม่ชัดเจนพอสำหรับนำไปวิเคราะห์เนื้อภาพจึงต้องปรับข้อมูลภาพที่มีอยู่ให้มีลักษณะที่เด่นชัดขึ้น โดยนำข้อมูลภาพนั้นมาประมวลผลด้วยวิธีการทางคอมพิวเตอร์เพื่อทำให้ข้อมูลภาพเหล่านั้นเป็นภาพสีเทา (Gray) โดยการนำค่า RGB ของแต่ละพิกเซลมารวมกันแล้วหารด้วย 3 ผลลัพธ์ที่ออกมาจะเป็นค่าแสดงระดับสีเทาของพิกเซลนั้นๆ ถ้าผลลัพธ์มีค่าระดับสีเทาน้อยกว่าหรือเท่ากับ 180 จะกำหนดให้พิกเซลนั้นมีค่าเป็น “ 0 ”(หรือเป็นสีดำ) แต่ถ้าผลลัพธ์มีค่ามากกว่า 180 จะกำหนดให้พิกเซลนั้นมีค่าเท่ากับ “ 1 ” (สีขาว) ข้อมูลภาพที่ผ่านกระบวนการดังกล่าวจะถูกนำไปใช้สำหรับการวิเคราะห์เนื้อภาพแสดงดังรูปที่ 2.2

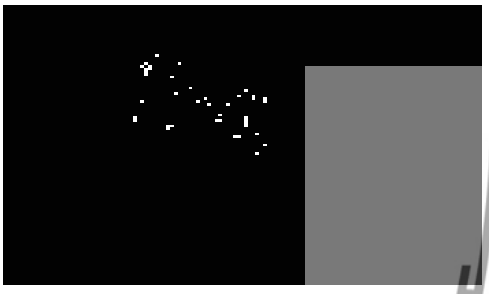
มหาวิทยาลัยเทคโนโลยีสุรนารี



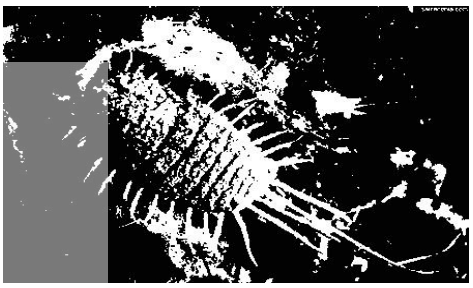
(ก)



(ข)



(ค)



(ง)

รูปที่ 2.2 ภาพเพลี้ยแป้งมันสำปะหลังที่ผ่านการปรับแต่งภาพ (Pre-processing image)

2.4เทคนิคและกระบวนการที่ใช้ในการศึกษาข้อมูล

การวิเคราะห์ข้อมูลภาพเพื่อจำแนกชนิดของเพลี้ยแป้งมันสำปะหลังนั้นจะใช้เทคนิคการวิเคราะห์เนื้อภาพ (Texture Analysis) เทคนิคนี้อาศัยการวิเคราะห์เมตริกซ์ความสัมพันธ์ของระดับสีเทา หรือที่เรียกว่า **"Grey-Level Co-Occurrence Matrix (GLCM)"**ที่ประกอบไปด้วยสมการสำหรับการคำนวณเนื้อภาพ (Texture Calculations) 8 สมการ ดังนี้

2.4.1 Homogeneity

สมการสำหรับคำนวณค่า Homogeneity คือ

$$\text{Homogeneity} = \sum_{i,j=0}^{N-1} \frac{P_{i,j}}{1+(i-j)^2} \qquad \text{..... (2.1)}$$

iคือ ระดับสีเทา ตั้งแต่ 0 ถึง 255 (พิกเซลอ้างอิง)

jคือ ระดับสีเทา ตั้งแต่ 0 ถึง 255 (พิกเซลใกล้เคียง)

Nคือ จำนวนของระดับสีเทาบนวัตถุที่สนใจ

$P_{i,j}$ คือ ความน่าจะเป็นของพิกเซลตำแหน่งแถวที่ i หลักที่ j

ถ้าข้อมูลภาพมีค่าระดับสีเทาที่สม่ำเสมอตลอดทั้งภาพ ค่า Homogeneity จะมีค่ามาก แต่ถ้าข้อมูลภาพมีค่าระดับสีเทาที่กระจัดกระจายแตกต่างกันตลอดทั้งภาพ ค่า Homogeneity จะให้ค่าต่ำ แต่ความสัมพันธ์นี้สามารถนำมาประยุกต์ใช้โดยเลือกพิจารณาเฉพาะบริเวณที่สนใจได้เป็นอย่างดี คือ ถ้าบริเวณที่สนใจมีระดับสีที่สม่ำเสมอ ค่า Homogeneity ที่ได้ก็จะมากด้วยเช่นกัน

2.4.2 Contrast

สมการสำหรับคำนวณค่า Contrast คือ

$$\text{Contrast} = \sum_{i,j=0}^{N-1} P_{i,j}\,(i-j)^2 \qquad \text{..... (2.2)}$$

Contrast เป็นการเปรียบเทียบระหว่างสีขาวที่มีค่ามากที่สุดกับสีดำที่มีค่าน้อยที่สุด Contrast ที่มีค่ามาก หมายความว่าข้อมูลภาพที่กำลังพิจารณาอยู่นั้นมีสีขาวเป็นส่วนประกอบของภาพอยู่เยอะ ในทางกลับกันถ้า Contrast มีค่าน้อย นั่นก็แสดงว่า ข้อมูลภาพที่กำลังพิจารณาอยู่มีสีดำเป็นส่วนประกอบของภาพมากกว่า

2.4.3 Dissimilarity

สมการสำหรับคำนวณค่า Dissimilarity คือ

$$\text{Dissimilarity} = \sum_{i,j=0}^{N-1} P_{i,j}\,|i-j| \qquad \text{..... (2.3)}$$

มหาวิทยาลัยเทคโนโลยีสุรนารี

Dissimilarity เป็นค่าที่บ่งบอกความแตกต่างของภาพว่าข้อมูลภาพที่กำลังพิจารณาอยู่นั้นในแต่ละพิกเซลมีความแตกต่างกันมากน้อยแค่ไหน ถ้า Dissimilarity มีค่ามากแสดงว่า $P_{i,j}$ มีค่าระดับสีเทาที่ซ้ำๆกันตามค่า i และ jนั่นคือภาพมีลักษณะคล้ายๆกัน ถ้า Dissimilarity มีค่าน้อยๆแสดงว่าภาพมีลักษณะที่แตกต่างกันอย่างเห็นได้ชัด ทั้งนี้จะเห็นความแตกต่างที่ชัดเจนขึ้นถ้าใช้ข้อมูลภาพหลายๆข้อมูล

2.4.4 Mean

สมการสำหรับคำนวณค่า Mean คือ

$$\text{Mean} = \sum_{i,j=0}^{N-1} P_{i,j}\, i \qquad \text{..... (2.4)}$$

Mean เป็นค่าเฉลี่ยของภาพโดยที่ $P_{i,j}$ แปรผันโดยตรงกับเมตริกซ์แถวที่ **i**

2.4.5 Standard Deviation

สมการสำหรับคำนวณค่า Standard Deviation คือ

$$\text{Standard Deviation} = \sum_{i,j=0}^{N-1} P_{i,j}\,(i - Mean)^2 \qquad \text{..... (2.5)}$$

Standard Deviation คือค่าความแปรปรวน เป็นการหาลักษณะการกระจายของความน่าจะเป็นซึ่งเป็นคุณสมบัติทางทฤษฎีความน่าจะเป็นโดยทั่วไปที่ต้องคำนึงถึงเมื่อมีการพิจารณาฟังก์ชันการกระจายความน่าจะเป็นในที่นี้เราต้องการดูลักษณะการกระจายของคู่จุดภาพที่มีระดับโทนสี (i, j) ที่เทียบจากค่าเฉลี่ย

มหาวิทยาลัยเทคโนโลยีสุรนารี

2.4.6 Entropy

สมการสำหรับคำนวณค่า Entropy คือ

$$\text{Entropy} = \sum_{i,j=0}^{N-1} P_{i,j}(-lnP_{i,j}) \qquad \text{..... (2.6)}$$

Entropy ใช้เพื่อแสดงลักษณะพื้นผิวของภาพ ซึ่ง ค่า Entropy จะมากหรือน้อยขึ้นอยู่กับ $P_{i,j}$ ถ้า $P_{i,j}$ มีค่ามาก Entropyจะมีค่าน้อยลง แต่ถ้า $P_{i,j}$ มีค่าน้อย ค่า Entropy จะมีค่าสูง ความสัมพันธ์นี้อธิบายได้ว่า Entropy ที่ให้ค่าออกมาสูงๆแสดงว่าภาพมีค่าระดับสีเทาที่แตกต่างกันเป็นจำนวนมาก แต่ถ้า Entropy มีค่าน้อยๆ แสดงว่าภาพมีค่าระดับสีเทาที่ซ้ำๆกัน

2.4.7 Angular Second Moment

สมการสำหรับคำนวณค่า Angular Second Moment คือ

$$\text{Angular Second Moment} = \sum_{i,j=0}^{N-1} {P_{i,j}}^2 \qquad \text{..... (2.7)}$$

Angular Second Moment (ASM) เป็นค่าที่สามารถบ่งบอกได้ว่าข้อมูลภาพที่กำลังพิจารณาอยู่นั้นมีลักษณะการเปลี่ยนแปลงของค่าระดับสีเทาเป็นไปในทิศทางใด ASM จะมีค่าสูงเมื่อ $P_{i,j}$ มีค่ามาก นั่นหมายความว่าภาพมีค่าระดับสีเทาที่ซ้ำๆกันตามค่า i และ j เป็นจำนวนมาก แต่ถ้าASM มีค่าน้อย แสดงว่าภาพมีค่าระดับสีเทาที่แตกต่างกันขึ้นกับค่า i และ j

2.4.8 Correlation

สมการสำหรับคำนวณค่า Correlation คือ

$$\text{Correlation} = \sum_{i,j=0}^{N-1} \frac{(i-\mu_i)(j-\mu_j)}{\sqrt{{\sigma_i}^2+{\sigma_j}^2}} \qquad \text{..... (2.8)}$$

มหาวิทยาลัยเทคโนโลยีสุรนารี

Correlation เป็นการวิเคราะห์ภาพโดยอาศัยความเป็นเชิงเส้น (Linear Dependence) ของค่าระดับสีเทาซึ่งเกิดจากการเปรียบเทียบค่าระดับสีเทาระหว่างพิกเซลอ้างอิงและพิกเซลใกล้เคียง ถ้าข้อมูลภาพที่นำมาพิจารณามีลักษณะเด่นๆจำนวนมากและอยู่ติดๆกัน Correlation จะมีค่าน้อย แต่ถ้าข้อมูลภาพที่นำมาพิจารณามีลักษณะเด่นๆที่ชัดเจน Correlation จะมีค่ามาก

เมื่อ

$$\mu_{\boldsymbol{i}} = \boldsymbol{Mean(i)} = \sum_{\cdot}^{\boldsymbol{N-1}} P_{i,j} \;\; \boldsymbol{i}$$

$$\mu_{\boldsymbol{j}} = \boldsymbol{Mean(j)} = \sum_{\boldsymbol{i,j=0}}^{\boldsymbol{N-1}} P_{i,j} \;\; \boldsymbol{j}$$

$$\boldsymbol{\sigma_i} = \boldsymbol{Standard\ Deviation(i)} = \sum_{\boldsymbol{i,j=0}}^{\boldsymbol{N-1}} P_{i,j}\,(\boldsymbol{i - Mean})^2$$

$$\boldsymbol{\sigma_j} = \boldsymbol{Standard\ Deviation(j)} = \sum_{\boldsymbol{i,j=0}}^{\boldsymbol{N-1}} P_{i,j}\,(\boldsymbol{j - Mean})^2$$

$\mu_{\boldsymbol{i}}$คือ ค่าเฉลี่ยระดับสีเทาพิกเซลอ้างอิง

$\mu_{\boldsymbol{j}}$คือ ค่าเฉลี่ยระดับสีเทาพิกเซลใกล้เคียง

$\boldsymbol{\sigma_i}$คือ ความเบี่ยงเบนมาตรฐานค่าระดับสีเทาพิกเซลอ้างอิง

$\boldsymbol{\sigma_j}$คือ ความเบี่ยงเบนมาตรฐานค่าระดับสีเทาพิกเซลใกล้เคียง

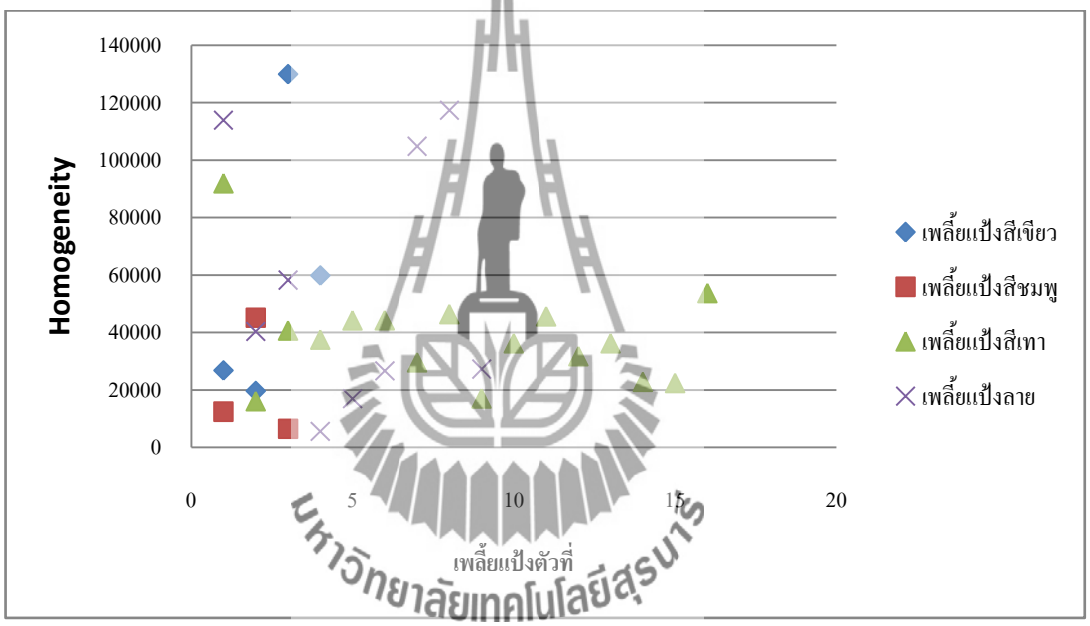
มหาวิทยาลัยเทคโนโลยีสุรนารี

การสร้างเมตริกซ์ความสัมพันธ์ของระดับสีเทา จะทำการสร้างเมตริกซ์ขนาด 256 x 256 โดยค่าที่จะใส่เข้าไปในเมตริกซ์แต่ละช่องนั้นหาได้จาก การเปรียบเทียบค่าระดับสีเทาระหว่าง 2 พิกเซล ได้แก่ พิกเซลอ้างอิง และ พิกเซลใกล้เคียง จากนั้นจะกำหนดให้ เมตริกซ์แถวที่ i เท่ากับค่าระดับสีเทาในพิกเซลอ้างอิง และ กำหนดให้เมตริกซ์หลักที่ j เท่ากับค่าระดับสีเทาในพิกเซลใกล้เคียง เมื่อทราบตำแหน่งช่องของเมตริกซ์ขนาด 256 x 256 แล้ว เมตริกซ์ตำแหน่งนั้นจะมีค่าเพิ่มขึ้นทีละ 1 ดำเนินการเปรียบเทียบเช่นเดียวกันนี้ไปตลอดทั้งภาพ ผลรวมของค่าในเมตริกซ์ขนาด256 x 256 จะมีค่าเท่ากับกับพื้นที่ของภาพที่กำลังพิจารณาอยู่ ค่าแต่ละค่าที่อยู่ในเมตริกซ์ขนาด 256 x 256 เรียกว่า **“ความน่าจะเป็นของพิกเซลตำแหน่งแถวที่ i และตำแหน่งหลักที่ j**หรือ

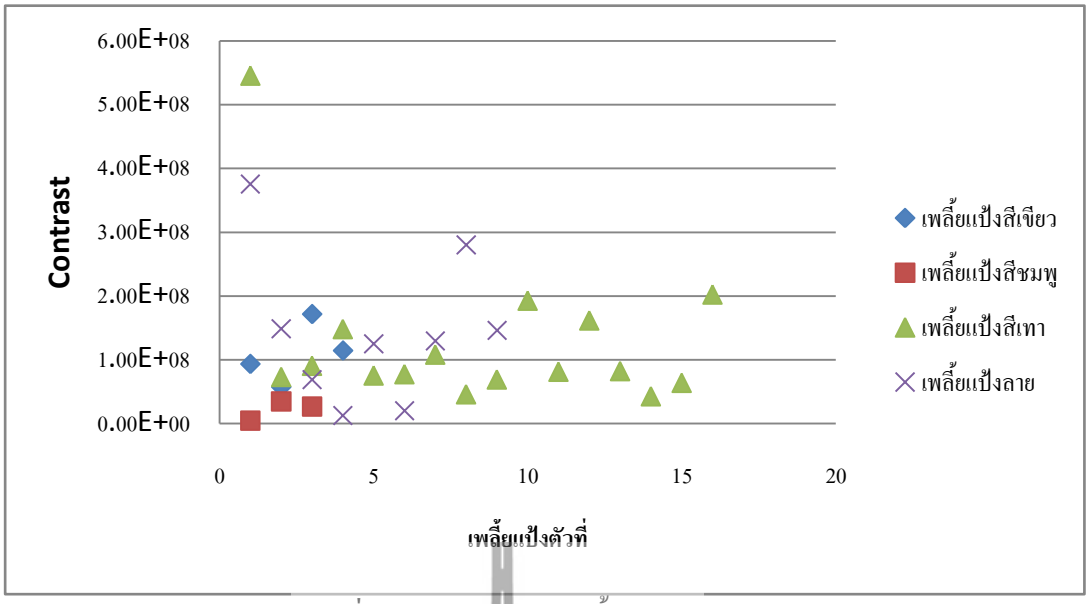
$P_{i,j}$" โดยที่$P_{i,j}$ ที่ได้นั้นจะนำไปใช้ในสมการสำหรับการคำนวณเนื้อภาพทั้ง 8 สมการข้างต้นถ้า $P_{i,j}$ มีค่ามาก แสดงว่าการเปรียบเทียบค่าระดับสีเทาระหว่าง 2 พิกเซล มีค่าระดับสีเทาที่ซ้ำๆกันตามค่า i และ j แต่ถ้าถ้า $P_{i,j}$ มีค่าน้อยๆแสดงว่าค่าระดับสีเทาระหว่าง 2พิกเซลที่ทำการเปรียบเทียบนั้นมีค่าระดับสีเทาที่แตกต่างกันขึ้นกับค่า i และ j

2.5ผลการศึกษาสมการวิเคราะห์เนื้อภาพ

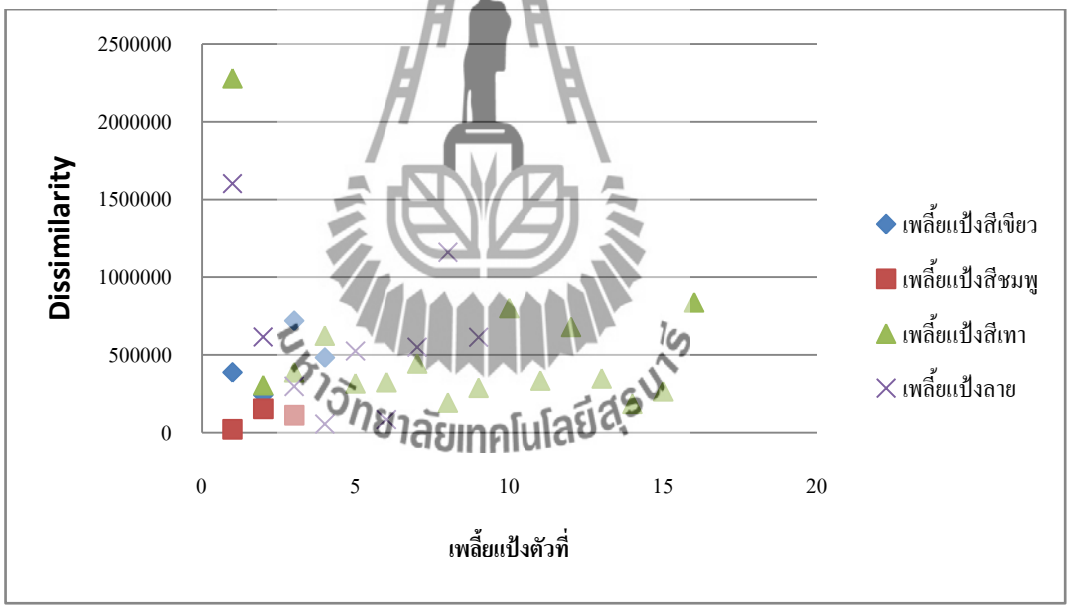
ใช้ข้อมูลภาพจากอินเตอร์เน็ต จำนวน 32 ภาพ โดยเป็นเพลี้ยแป้งสีเทา 16 ภาพเพลี้ยแป้งลาย 9 ภาพเพลี้ยแป้งสีชมพู 3 ภาพ และเพลี้ยแป้งสีเขียว 4 ภาพข้อมูลภาพต้นแบบทั้งหมดอยู่ในภาคผนวก ก ซึ่งผลการวิเคราะห์ภาพตามสมการที่ (1)-(8) แสดงดังกราฟรูปที่ 2.3-รูปที่ 2.10



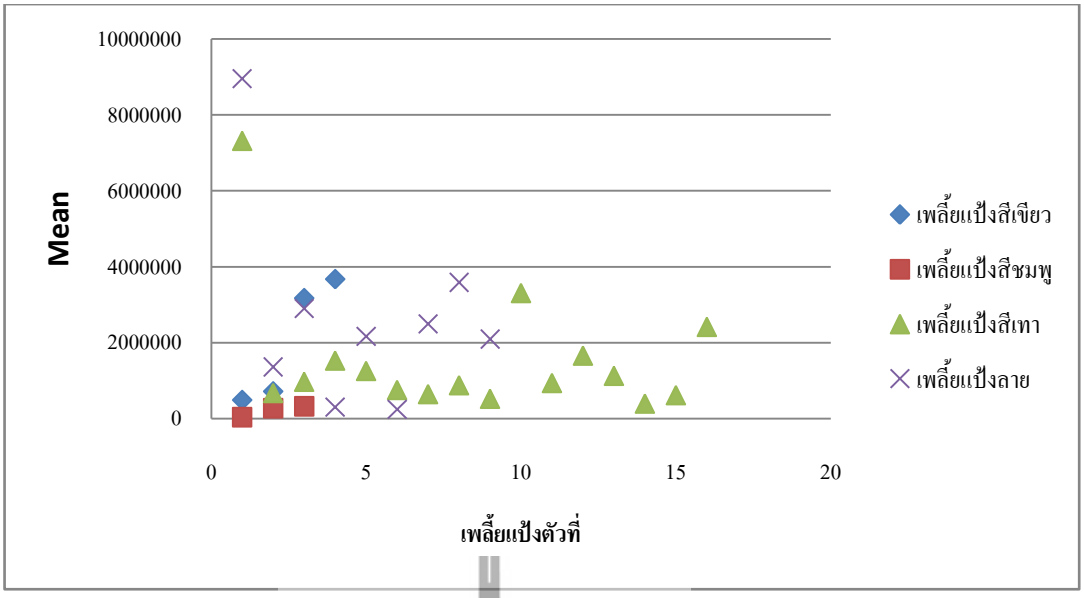
รูปที่ 2.3 Homogeneity ของเพลี้ยแป้งแต่ละชนิด



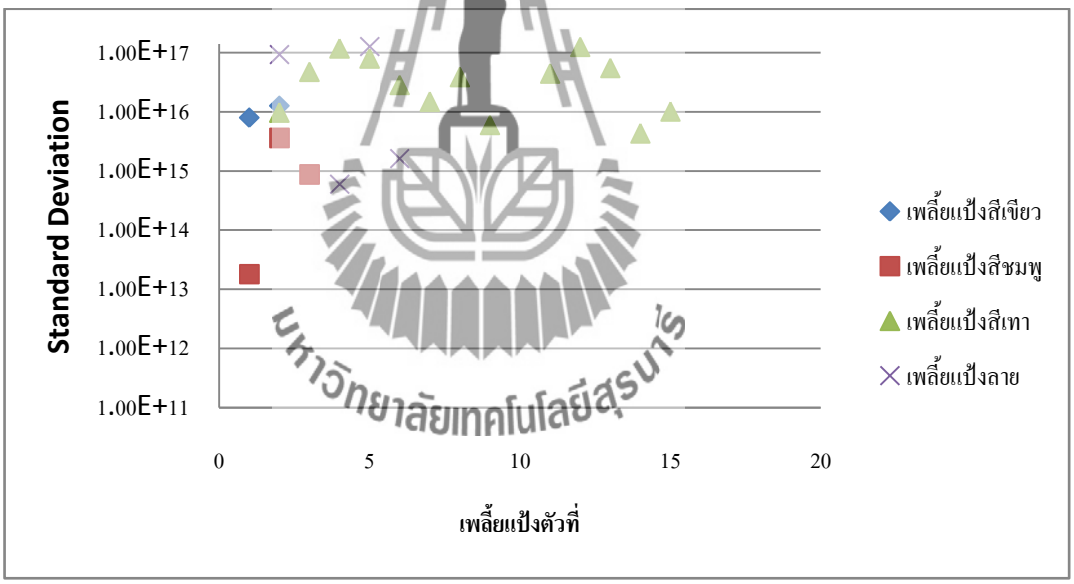
รูปที่ 2.4 Contrast ของเพลี้ยแป้งแต่ละชนิด



รูปที่ 2.5 Dissimilarity ของเพลี้ยแป้งแต่ละชนิด

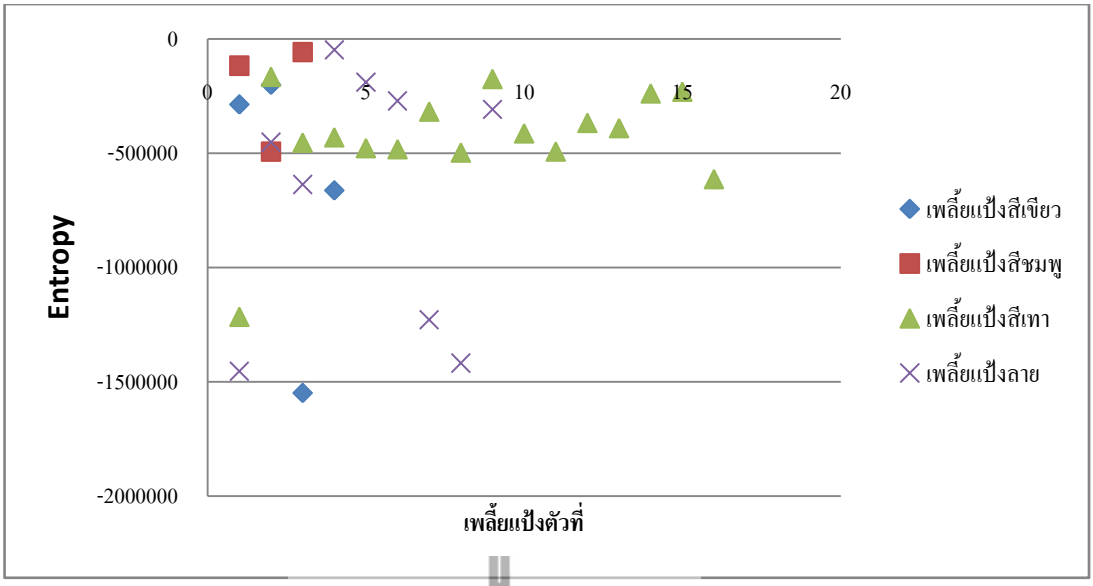


รูปที่ 2.6 Mean ของเพลี้ยแป้งแต่ละชนิด

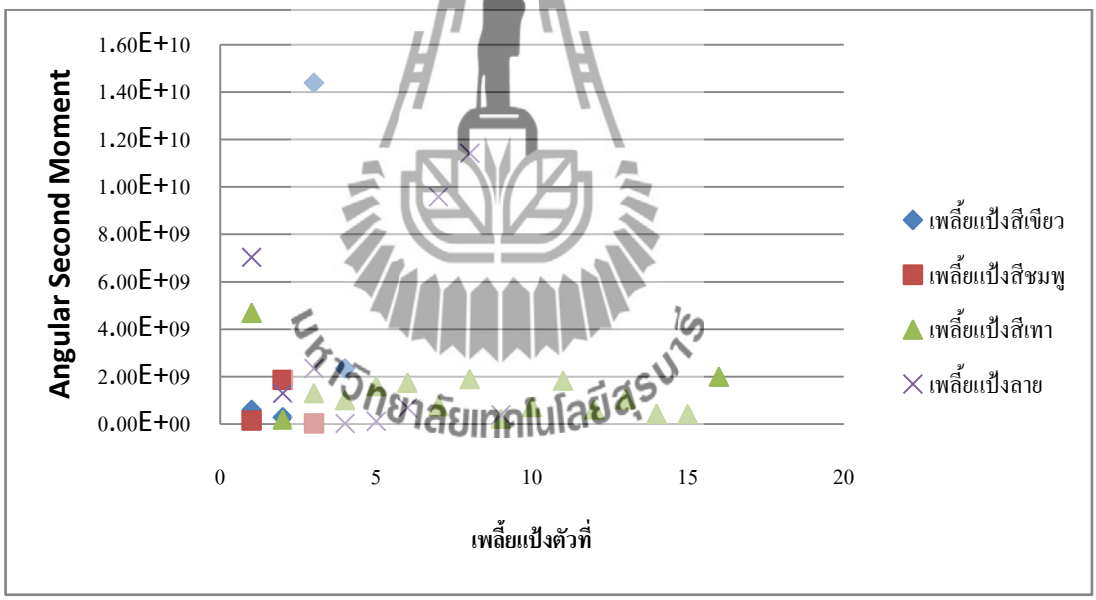


รูปที่ 2.7 Standard Deviation ของเพลี้ยแป้งแต่ละชนิด

มหาวิทยาลัยเทคโนโลยีสุรนารี

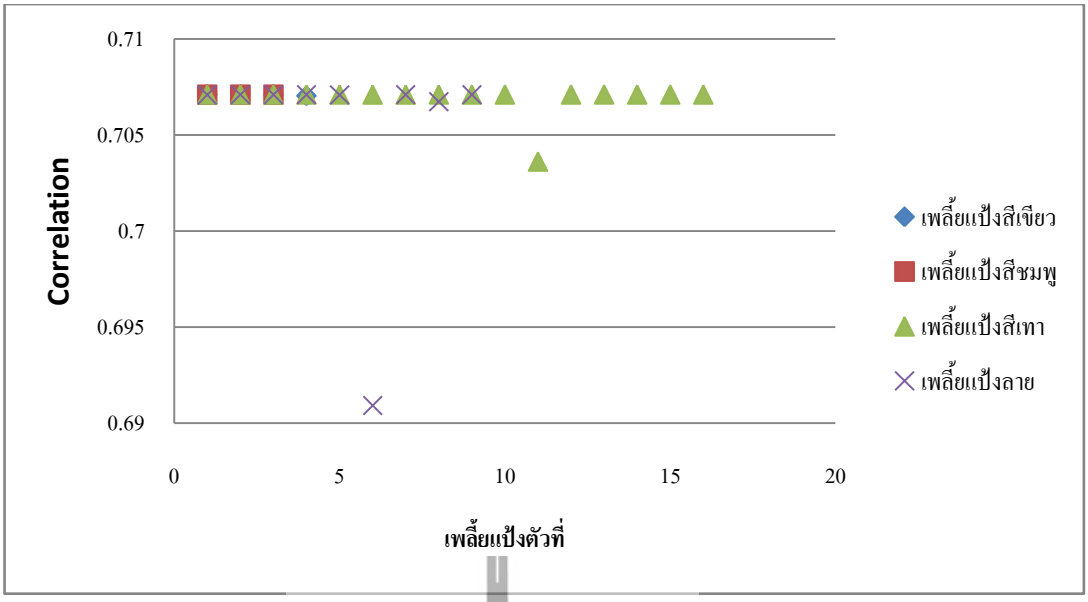


รูปที่ 2.8 Entropy ของเพลี้ยแป้งแต่ละชนิด



รูปที่ 2.9Angular Second Moment ของเพลี้ยแป้งแต่ละชนิด

มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 2.10 Correlation ของเพลี้ยแป้งแต่ละชนิด

จากข้อมูลในรูปที่ 2.3 -รูปที่ 2.10จะเห็นว่าสมการวิเคราะห์เนื้อภาพทั้ง 8 สมการไม่มีสมการใดเลยที่จะแยกแยะเพลี้ยแป้งมันสำปะหลังทั้ง 4 ชนิดได้อย่างชัดเจน เนื่องจากกราฟแสดงตำแหน่งข้อมูลที่กระจัดกระจาย จึงไม่สามารถบอกได้ว่าเพลี้ยแป้งชนิดไหนมีค่าการกระจายตัวที่ได้จากสมการทั้ง 8 สมการเท่าไร ส่งผลให้การประมวลผลภาพด้วยเทคนิคการวิเคราะห์เนื้อภาพ (Texture Analysis) ยังให้ผลลัพธ์ของการจำแนกไม่เด่นชัดพอสำหรับเพลี้ยแป้งมันสำปะหลังบางชนิด ถ้าดูจากกราฟแสดงผลการวิเคราะห์เนื้อภาพด้วยสมการคำนวณเนื้อภาพแล้ว พบว่า Standard Deviation ให้ความเหมาะสมมากที่สุดสำหรับการจำแนกชนิดเพลี้ยแป้งมันสำปะหลังแต่จะพบว่า Standard Deviation ของเพลี้ยแป้งสีชมพูและเพลี้ยแป้งสีเขียวมีค่าใกล้เคียงกันมาก ค่าความใกล้เคียงนี้อาจจะส่งผลให้การจำแนกผิดเพี้ยนไป ซึ่งไม่เป็นผลดีต่อเกษตรกรผู้ปลูกมันสำปะหลังแน่นอน จึงทำให้การจำแนกชนิดเพลี้ยแป้งมันสำปะหลังด้วย Texture Analysis ไม่สามารถแยกแยะชนิดเพลี้ยแป้งมันสำปะหลังได้

บทที่3

การประมวลผลภาพด้วยเทคนิคการวิเคราะห์ค่าสี RGB

การประมวลผลภาพด้วยเทคนิคการวิเคราะห์ค่าสี RGB (RGB Analysis) เป็นเทคนิคที่นำมาจำแนกชนิดเพลี้ยแป้งมันสำปะหลัง จึงมีความจำเป็นอย่างยิ่งที่ต้องใช้การประมวลผลภาพด้วยเทคนิคการวิเคราะห์ค่าสี RGB (RGB Analysis)มาช่วยจำแนกชนิดเพลี้ยแป้งมันสำปะหลังอีกวิธีหนึ่ง เพราะนอกจากลายบนตัวของเพลี้ยแป้งมันสำปะหลังแต่ละชนิดจะแตกต่างกันแล้ว เพลี้ยแป้งมันสำปะหลังบางชนิดยังมีสีที่ปรากฏบนตัวเพลี้ยแป้งมันสำปะหลังอีกด้วย เช่น เพลี้ยแป้งสีชมพู เป็นต้น

3.1 หลักในการดำเนินงาน

เพลี้ยแป้งมันสำปะหลัง 4 ชนิด ถ้าสังเกตจากสีของตัวเพลี้ยแป้งพบว่า เพลี้ยแป้งสีเขียว เพลี้ยแป้งลาย และ เพลี้ยแป้งสีเทา มีสีขาวปะปนอยู่เยอะพอสมควร แต่เพลี้ยแป้งสีชมพูปรากฏสีชมพูอ่อนๆอย่างเห็นได้ชัด ด้วยข้อแตกต่างด้านสีนี้จะเป็นข้อแตกต่างที่สำคัญที่จะใช้จำแนกชนิดเพลี้ยแป้งมันสำปะหลัง การจำแนกชนิดเพลี้ยแป้งมันสำปะหลังด้วยเทคนิคนี้สามารถใช้ภาพถ่ายเพลี้ยแป้งมันสำปะหลังที่มีหลายๆตัวในหนึ่งภาพได้

จุดมุ่งหมายหลักของการประมวลผลภาพด้วยเทคนิคนี้มุ่งเป้าหมายของการจำแนกชนิดเพลี้ยแป้งมันสำปะหลังไปที่เพลี้ยแป้งสีชมพู เพราะนอกจากจะมีสีบนตัวที่ปรากฏชัดเจนแล้ว ในประเทศไทยยังพบว่าเพลี้ยแป้งสีชมพูเป็นศัตรูพืชที่ร้ายแรงที่สุด จึงมีความจำเป็นอย่างยิ่งที่เกษตรกรผู้ปลูกมันสำปะหลังควรจะทราบว่าเพลี้ยแป้งมันสำปะหลังที่ระบาดในแปลงปลูกนั้นเป็นเพลี้ยแป้งสีชมพูหรือไม่ ถ้าใช่ก็ควรรีบกำจัดให้เร็วที่สุด การประมวลผลภาพด้วยเทคนิคการวิเคราะห์ค่าสี RGB (RGB Analysis) จึงเหมาะอย่างยิ่งที่จะเป็นอีกตัวช่วยที่จะสามารถทำให้เกษตรกรผู้ปลูกมันสำปะหลังจำแนกชนิดเพลี้ยแป้งมันสำปะหลังได้

มหาวิทยาลัยเทคโนโลยีสุรนารี

3.2 ข้อมูลที่ใช้ในการดำเนินงาน

ข้อมูลที่ใช้ในการดำเนินงานเป็นภาพถ่ายที่มาจากแหล่งข้อมูลทางอินเตอร์เน็ตซึ่งเป็นข้อมูลภาพ โดยจะแบ่งข้อมูลภาพออกเป็น 2 ส่วน ส่วนแรกเป็นข้อมูลภาพที่เรียกว่า **“ข้อมูลภาพต้นแบบ”** โดยที่ภาพต้นแบบจะเป็นฐานข้อมูลสำหรับใช้ในการวิเคราะห์เพื่อจำแนกชนิดของภาพเพลี้ยแป้งมันสำปะหลังที่ไม่ทราบชนิด ส่วนที่สองคือข้อมูลภาพเพลี้ยแป้งมันสำปะหลังที่ไม่ทราบชนิด เรียกว่า **“ภาพอินพุต”**

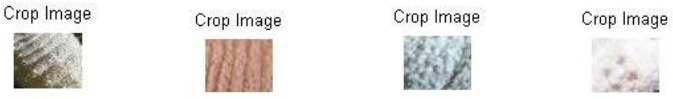
ภาพต้นแบบที่เลือกใช้เป็นฐานข้อมูลเพื่อการจำแนกชนิดเพลี้ยแป้งมันสำปะหลังแสดงด้วยตาราง 3.1: แสดงภาพต้นแบบเพลี้ยแป้งมันสำปะหลังที่ใช้เป็นฐานข้อมูลมีจำนวน 32 ภาพ แบ่งเป็น เพลี้ยแป้งสีเทา 16 ภาพ เพลี้ยแป้งลาย 9 ภาพ เพลี้ยแป้งสีชมพู 3 ภาพ และเพลี้ยแป้งสีเขียว 4 ภาพ โดยภาพทั้งหมดอยู่ในภาคผนวก ก

3.3 การคัดกรองภาพเบื้องต้น (Pre-Processing)

3.3.1 การคัดกรองภาพต้นแบบ

ภาพต้นแบบที่แสดงไว้ในข้างต้นต้องนำมาคัดเลือกเอาเฉพาะบางส่วนของตัวเพลี้ยแป้งมันสำปะหลังเท่านั้น เนื่องจากเราจำแนกชนิดเพลี้ยแป้งมันสำปะหลังด้วยเทคนิคการวิเคราะห์ค่าสี RGB เราจึงสนใจบริเวณใดบริเวณหนึ่งของตัวเพลี้ยแป้งมันสำปะหลังเท่านั้น เรียกบริเวณที่สนใจนั้นว่า **“บริเวณที่สนใจของภาพต้นแบบ”** ของภาพเพลี้ยแป้งมันสำปะหลังทั้ง 4 ชนิดที่จะใช้เป็นภาพต้นแบบสำหรับเป็นฐานข้อมูลในการจำแนกชนิดเพลี้ยแป้งมันสำปะหลังนั้นจะเป็นภาพถ่ายที่มีขนาดเท่าใดก็ได้ แต่เมื่อนำมาผ่านการคัดกรองภาพเบื้องต้นบริเวณที่สนใจของภาพต้นแบบจะต้องมีขนาดเท่ากันทั้งหมดเพื่อที่ว่าการจำแนกชนิดเพลี้ยแป้งมันสำปะหลังจะเป็นรูปแบบเดียวกันทั้งหมด [ภาพต้นแบบพร้อมด้วยบริเวณที่สนใจของภาพต้นแบบของเพลี้ยแป้งมันสำปะหลังทั้ง 4 ชนิด แสดงไว้ในภาคผนวก ข]การเลือกบริเวณสนใจของภาพต้นแบบ แสดงดังตัวอย่างในรูปที่ 3.1

มหาวิทยาลัยเทคโนโลยีสุรนารี



(ก) (ข) (ค) (ง)

รูปที่ 3.1 การเลือกบริเวณที่สนใจของภาพต้นแบบ (ก) เพลี้ยแป้งสีเขียว (ข) เพลี้ยแป้งสีชมพู (ค) เพลี้ยแป้งสีเทา และ (ง) เพลี้ยแป้งลาย

3.3.2การคัดกรองภาพอินพุต

ภาพอินพุตหรือภาพเพลี้ยแป้งมันสำปะหลังที่ไม่ทราบชนิดจะถูกนำมาคัดกรองภาพเช่นเดียวกับการคัดกรองภาพต้นแบบ เราจึงสนใจบริเวณใดบริเวณหนึ่งของตัวเพลี้ยแป้งมันสำปะหลังเท่านั้น เรียกบริเวณที่สนใจนั้นว่า **"บริเวณที่สนใจของภาพอินพุต "** ในการเลือกบริเวณที่สนใจของภาพอินพุตสามารถทำได้ 2 แบบ

1.แบบไม่กำหนดขนาด เป็นการเลือกบริเวณที่สนใจของภาพอินพุตจากจุดที่สนใจโดยตรงโดยไม่มีการกำหนดขนาดภาพที่จะได้ ผลลัพธ์การจำแนกชนิดเพลี้ยแป้งมันสำปะหลังจะมีความแม่นยำมากกว่าถ้าขนาดบริเวณที่สนใจของภาพอินพุตใกล้เคียงกับขนาดบริเวณที่สนใจของภาพต้นแบบ แต่ความแม่นยำจะน้อยลงถ้าขนาดบริเวณที่สนใจของภาพอินพุตแตกต่างมากอย่างเห็นได้ชัดเมื่อเทียบกับขนาดบริเวณที่สนใจของภาพต้นแบบ

2.แบบกำหนดขนาด การกำหนดขนาดให้กับบริเวณที่สนใจของภาพอินพุตจะต้องเท่ากันกับขนาดของบริเวณที่สนใจของภาพต้นแบบ การกำหนดขนาดจะช่วยให้การประมวลผลภาพด้วยเทคนิคการวิเคราะห์ค่าสี RGB อยู่ในหลักการเดียวกันกับฐานข้อมูลของบริเวณที่สนใจของภาพต้นแบบ ผลลัพธ์การจำแนกชนิดเพลี้ยแป้งมันสำปะหลังก็จะมีความถูกต้องและแม่นยำกว่าแบบไม่กำหนดขนาด

3.4 เทคนิคและกระบวนการที่ใช้ในการศึกษาข้อมูล

หลังงจากดำเนินการผ่านการคัดกรองภาพเบื้องต้นแล้ว ทั้งบริเวณที่สนใจของภาพต้นแบบเพลี้ยแป้งมันสำปะหลังทั้ง 4 ชนิดและบริเวณที่สนใจของภาพอินพุต จะถูกนำมาหาค่าเฉลี่ยของสี R, G และ B วิธีการหาค่าเฉลี่ยสีแสดงด้วยสมการต่อไปนี้

$$R_{avg} = \frac{\sum_{m=1}^{n} R}{m \times n} \quad \text{..... (3.1)}$$

$$G_{avg} = \frac{\sum_{m=1}^{n} G}{m \times n} \quad \text{..... (3.2)}$$

$$B_{avg} = \frac{\sum_{m=1}^{n} B}{m \times n} \quad \text{..... (3.3)}$$

เมื่อ m, nเท่ากับจำนวนพิกเซลทั้งหมดของภาพ

Rคือ ค่าที่บอกความเป็นสีแดงของพิกเซลหนึ่งๆ

Gคือ ค่าที่บอกความเป็นสีเขียวของพิกเซลหนึ่งๆ

Bคือ ค่าที่บอกความเป็นสีน้ำเงินของพิกเซลหนึ่งๆ

R_{avg} คือ ค่าเฉลี่ยสีแดงบริเวณที่สนใจของภาพหนึ่งๆ

G_{avg} คือ ค่าเฉลี่ยสีเขียวบริเวณที่สนใจของภาพหนึ่งๆ

B_{avg} คือ ค่าเฉลี่ยสีน้ำเงินบริเวณที่สนใจของภาพหนึ่งๆ

จากภาพต้นแบบที่ใช้เป็นฐานข้อมูลทั้งหมดจำนวน 32 ภาพ เราจะนำบริเวณที่สนใจของภาพต้นแบบขนาด 50x40pixels มาคำนวณด้วยสมการสำหรับหาค่าเฉลี่ยของสี R, G และ Bจากนั้นจะนำค่าเฉลี่ยสี R, G และ B ของบริเวณที่สนใจของภาพต้นแบบเพลี้ยแป้งมันสำปะหลังทั้ง 4 ชนิด และบริเวณที่สนใจของภาพอินพุต ซึ่งได้จากสมการด้านบนมาผ่านกระบวนการเพื่อทำให้บริเวณที่สนใจของเพลี้ยแป้งมันสำปะหลังเหล่านั้นไม่ขึ้นกับความเข้มของแสงหรือเรียกว่าการ “Normalized RGB”ภาพถ่ายเพลี้ยแป้งมันสำปะหลังแต่ละภาพที่ถูกนำมาประมวลผลมีความเป็นไปได้สูงที่ความเข้มแสงขณะถ่ายภาพต่างกันบ้างหรืออาจจะถ่ายด้วยระยะเวลาที่แตกต่างกันไป ภาพถ่ายที่ได้จึงมีความเข้มแสงไม่เท่ากัน การประมวลผลด้วยเทคนิคการวิเคราะห์ค่าสี RGBนี้จึงต้องทำให้บริเวณที่สนใจของภาพเพลี้ยแป้งมันสำปะหลังไม่ขึ้นกับความเข้มแสง ซึ่งสามารถกระทำได้ด้วยสมการดังต่อไปนี้

$$R_{\widetilde{B}} = \frac{R_{avg}}{R_{avg} + G_{avg} + B_{avg}} \quad \text{..... (3.4)}$$

$$G_{\widetilde{B}} = \frac{G_{avg}}{R_{avg} + G_{avg} + B_{avg}} \quad \text{..... (3.5)}$$

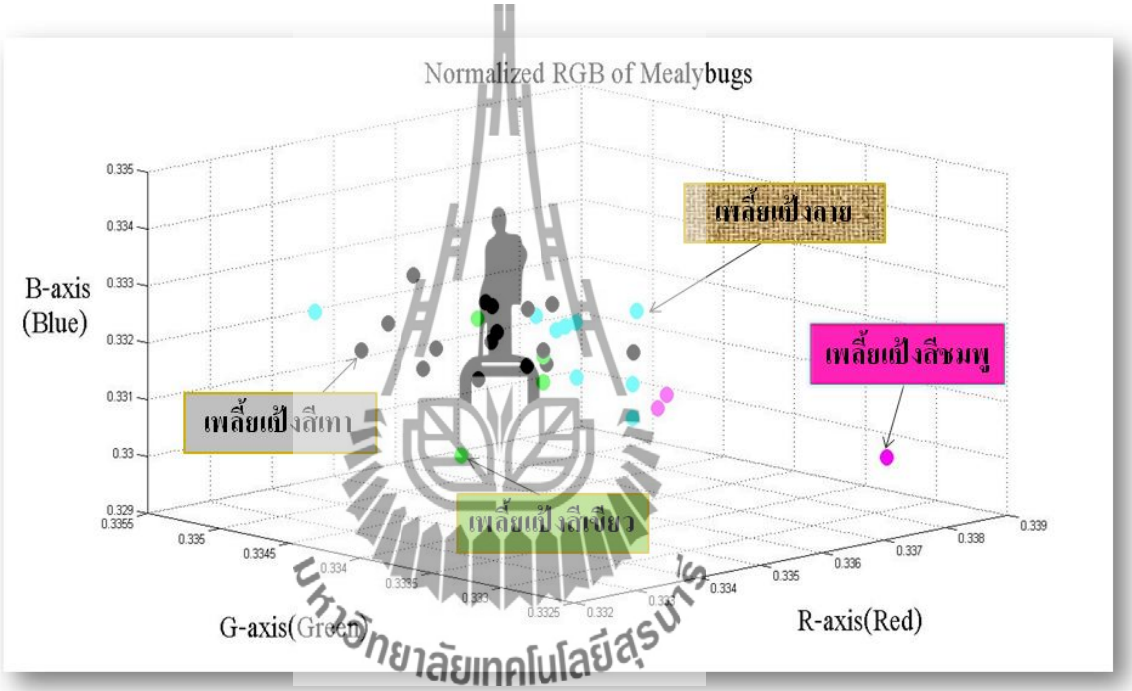
$$B_{\widetilde{B}} = \frac{B_{avg}}{R_{avg} + G_{avg} + B_{avg}} \quad \text{..... (3.6)}$$

เมื่อ $R_{\bar{B}}$ คือ ค่าเฉลี่ยสีแดงบริเวณที่สนใจของภาพหนึ่งๆที่ไม่ขึ้นกับความเข้มแสง

$G_{\bar{B}}$ คือ ค่าเฉลี่ยสีเขียวบริเวณที่สนใจของภาพหนึ่งๆที่ไม่ขึ้นกับความเข้มแสง

$B_{\bar{B}}$ คือ ค่าเฉลี่ยสีน้ำเงินบริเวณที่สนใจของภาพหนึ่งๆที่ไม่ขึ้นกับความเข้มแสง

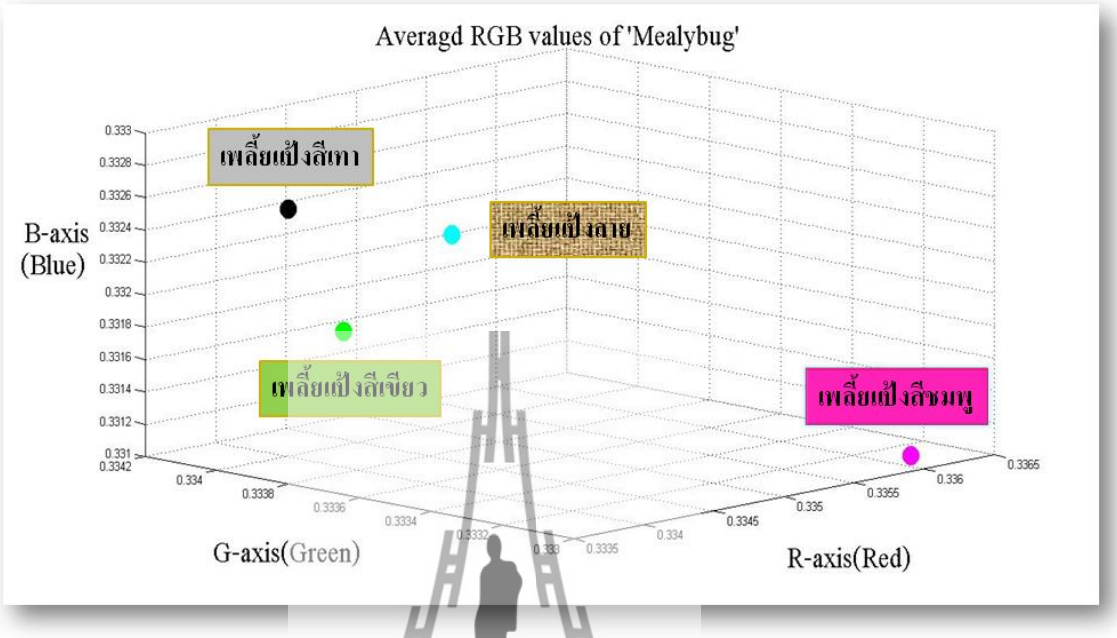
เมื่อนำบริเวณที่สนใจของภาพต้นแบบที่ใช้เป็นฐานข้อมูลทั้ง 32 ภาพมาผ่านสมการที่ทำให้ภาพไม่ขึ้นกับความเข้มแสงและนำค่าที่ได้มาพล็อตกราฟ 3 มิติดัง รูปที่ 3.2ค่าสีแต่ละภาพของบริเวณที่สนใจของภาพต้นแบบเพลี้ยแป้งมันสำปะหลังทั้ง 4 ชนิดที่พล็อตในรูปแกนสี R, G และ B โดยกำหนดให้แกน x แสดงแกนของ R(red), แกน y แสดงแกนของ G(green)และแกน z แสดงแกนของ B(blue)



รูปที 3.2 ค่าสีแต่ละภาพของบริเวณทีสนใจของภาพต้นแบบเพลียแป้งมันสำปะหลังทัง4 ชนิดที พล็อตในรูปแกนสี R, G และ B

จากรูปที่ 3.2จะเห็นว่าเพลี้ยแป้งสีเทา เพลี้ยแป้งสีเขียว และเพลี้ยแป้งลาย มีกลุ่มข้อมูลค่าสีที่อยู่ในบริเวณเดียวกันและบางชนิดมีค่าสีซ้อนทับกันบางส่วน แต่มีกลุ่มข้อมูลของเพลี้ยแป้งสีชมพูที่แยกบริเวณออกมาอย่างชัดเจน ไม่ปะปนอยู่ในกลุ่มของเพลี้ยแป้งทั้ง 3 ชนิดเลย ซึ่งข้อมูลข้างต้นนี้แสดงให้เห็นว่ามีความเป็นไปได้สูงที่จะสามารถจำแนกชนิดเพลี้ยแป้งสีชมพูออกจากเพลี้ยแป้งชนิดอื่นๆ ได้ ดังนั้นเทคนิคการประมวลผลภาพด้วย RGB Analysisจากข้อมูลในรูปที่ 3.2จะพบว่าจุดสีที่เป็นข้อมูลของเพลี้ยแป้งแต่ละชนิดนั้นอยู่กันอย่างกระจัดกระจาย เพื่อให้ได้ตัวแทนของจุดสีสำหรับใช้เป็นฐานข้อมูลในการจำแนกชนิดเพลี้ยแป้งมันสำปะหลังต่อไปนี้เราจะ

นำจุดสีของเพลี้ยแป้งชนิดเดียวกันมาหาค่าเฉลี่ยเพื่อหาตัวแทนจุดสีของเพลี้ยแป้งแต่ละชนิด โดยค่าเฉลี่ยจุดสีของเพลี้ยแป้งทั้ง 4 ชนิดแสดงด้วย รูปที่ 3.3ค่าสีเฉลี่ยของบริเวณที่สนใจของภาพต้นแบบเพลี้ยแป้งมันสำปะหลังทั้ง4 ชนิดต่อไปนี้



รูปที่ 3.3ค่าสีเฉลี่ยของบริเวณที่สนใจของภาพต้นแบบเพลี้ยแป้งมันสำปะหลังทั้ง4 ชนิด

จากรูปที่ 3.3จุดสีแต่ละจุดจะเป็นตัวแทนของเพลี้ยแป้งมันสำปะหลัง 4 ชนิด จะเห็นว่าเพลี้ยแป้งสีเทา เพลี้ยแป้งสีเขียว และเพลี้ยแป้งลาย ยังคงมีกลุ่มข้อมูลค่าสีที่อยู่ในบริเวณเดียวกันอย่างเห็นได้ชัดเจนและเพลี้ยแป้งสีชมพูก็ยังคงแสดงตำแหน่งความแตกต่างเมื่อเทียบกับเพลี้ยแป้ง 3 ชนิดได้ชัดเจนเช่นเดิม นั่นแสดงว่าการประมวลผลภาพด้วยเทคนิคการวิเคราะห์ค่าสี RGB Analysis อาจจะมีประสิทธิภาพถ้าใช้ในการจำแนกเพลี้ยแป้งสีชมพู

กระบวนการสุดท้ายที่จะให้คำตอบว่าภาพอินพุตเป็นเพลี้ยแป้งชนิดใดคือการคำนวณหาค่าความใกล้เคียงโดยการเทียบค่าระหว่างค่าเฉลี่ยสีบริเวณที่สนใจของภาพอินพุตกับค่าเฉลี่ยสีบริเวณที่สนใจของภาพต้นแบบของเพลี้ยแป้งมันสำปะหลังทั้ง 4 ชนิด ถ้าค่าความใกล้เคียงใดมีค่าน้อยที่สุดแสดงว่าภาพอินพุตนั้นเป็นชนิดเดียวกันกับภาพต้นแบบเพลี้ยแป้งมันสำปะหลังที่ถูกนำมาเปรียบเทียบนั้นสมการที่ใช้แสดงดังนี้

มหาวิทยาลัยเทคโนโลยีสุรนารี

$$D_{input \leftrightarrow green} = \sqrt{(R_{\tilde{B}u} - R_{\tilde{B}1})^2 + (G_{\tilde{B}u} - G_{\tilde{B}1})^2 + (B_{\tilde{B}u} - B_{\tilde{B}1})^2} \quad (3.7)$$

$$D_{input \leftrightarrow lay} = \sqrt{(R_{\tilde{B}u} - R_{\tilde{B}2})^2 + (G_{\tilde{B}u} - G_{\tilde{B}2})^2 + (B_{\tilde{B}u} - B_{\tilde{B}2})^2} \quad (3.8)$$

$$D_{input \leftrightarrow pink} = \sqrt{(R_{\tilde{B}u} - R_{\tilde{B}3})^2 + (G_{\tilde{B}u} - G_{\tilde{B}3})^2 + (B_{\tilde{B}u} - B_{\tilde{B}3})^2} \quad (3.9)$$

$$D_{input \leftrightarrow gray} = \sqrt{(R_{\tilde{B}u} - R_{\tilde{B}4})^2 + (G_{\tilde{B}u} - G_{\tilde{B}4})^2 + (B_{\tilde{B}u} - B_{\tilde{B}4})^2} \quad (3.10)$$

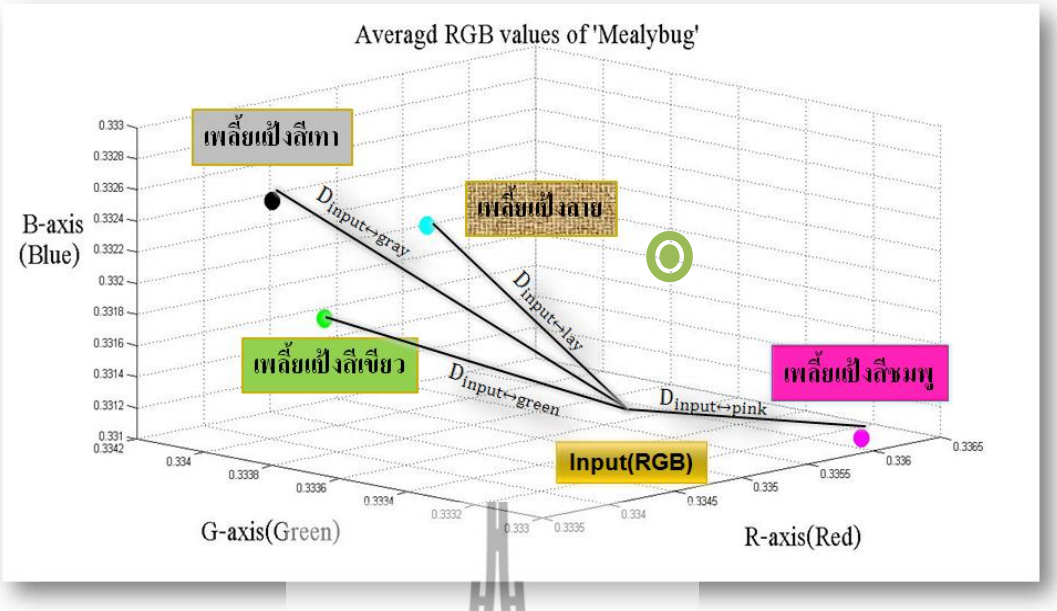
เมื่อ $D_{input \leftrightarrow green}$ คือ ค่าความใกล้เคียงโดยการเทียบค่าระหว่างค่าเฉลี่ยสีบริเวณที่สนใจของภาพอินพุตกับค่าเฉลี่ยสีบริเวณที่สนใจของภาพเพลี้ยแป้งสีเขียว

$D_{input \leftrightarrow lay}$ คือ ค่าความใกล้เคียงโดยการเทียบค่าระหว่างค่าเฉลี่ยสีบริเวณที่สนใจของภาพอินพุตกับค่าเฉลี่ยสีบริเวณที่สนใจของภาพเพลี้ยแป้งลาย

$D_{input \leftrightarrow pink}$ คือ ค่าความใกล้เคียงโดยการเทียบค่าระหว่างค่าเฉลี่ยสีบริเวณที่สนใจของภาพอินพุตกับค่าเฉลี่ยสีบริเวณที่สนใจของภาพเพลี้ยแป้งสีชมพู

$D_{input \leftrightarrow gray}$ คือ ค่าความใกล้เคียงโดยการเทียบค่าระหว่างค่าเฉลี่ยสีบริเวณที่สนใจของภาพอินพุตกับค่าเฉลี่ยสีบริเวณที่สนใจของภาพเพลี้ยแป้งสีเทา

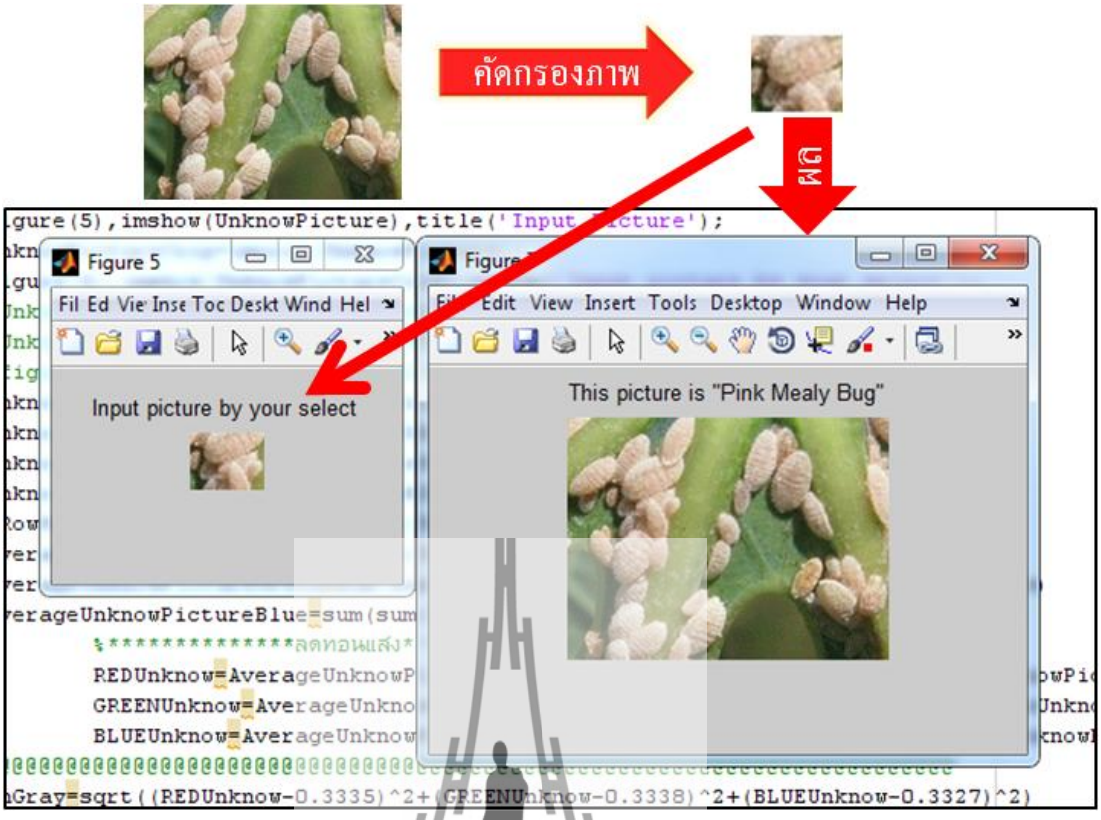
สมการที่ใช้หาค่าความใกล้เคียงโดยการเทียบค่าระยะทาง ระหว่างค่าเฉลี่ยสีบริเวณที่สนใจของภาพอินพุตกับค่าเฉลี่ยสีบริเวณที่สนใจของภาพต้นแบบทั้ง 4 ชนิดสามารถเปรียบเทียบได้กับการหาระยะทางระหว่างจุดสี RGB ในแนวแกน x แกน y และแกน z เมื่อกำหนดให้ R คือ แกน x, G คือ แกน y และ B คือแกน z ยกตัวอย่าง เช่น เมื่อมีภาพอินพุตดังแสดงในรูปที่ 3.1และคำนวณค่าเฉลี่ยสี R, G และ B แล้ว ก็จะได้จุดสีของภาพอินพุตที่แสดงด้วย รูปที่ 3.4แสดงระยะห่างระหว่างจุดสี RGBของภาพอินพุตเทียบกับฐานข้อมูลของเพลี้ยแป้งมันสำปะหลังทั้ง 4 ชนิดถ้าภาพอินพุตมีระยะห่างจากจุดอ้างอิงใดน้อยที่สุด จะกำหนดให้คำตอบคือเพลี้ยแป้งมันสำปะหลังชนิดนั้นๆ



รูปที่ 3.4แสดงระยะห่างระหว่างจุดสี RGBของภาพอินพุตเทียบกับฐานข้อมูลของเพลี้ยแป้งมันสำปะหลังทั้ง 4ชนิด

เมื่อมีภาพอินพุตที่ต้องการทราบชนิดถูกนำไปประมวลผลภาพด้วยโปรแกรม MATLAB โปรแกรมจะแสดงหน้าต่างโชว์ภาพอินพุตให้ผู้ใช้เลือกบริเวณใดบริเวณหนึ่งบนตัวเพลี้ยแป้ง จากนั้นโปรแกรมประมวลผลก็จะนำภาพที่ผู้ใช้เลือกไปวิเคราะห์หาว่าภาพอินพุตนั้นเป็นชนิดใด ดังรูปที่ 3.5

มหาวิทยาลัยเทคโนโลยีสุรนารี



รูปที่ 3.5การจำแนกชนิดเพลี้ยแป้งมันสำปะหลังจากโปรแกรม MATLABที่พัฒนาขึ้น

3.5ผลการทดสอบประสิทธิภาพของโปรแกรมโดยใช้ภาพถ่ายจากอินเตอร์เน็ต

ภาพอินพุตที่ใช้ทดสอบมี 32 ภาพ (ดูจากภาคผนวก ค) โดยแยกเป็นเพลี้ยแป้งสีชมพู 11 ภาพ เพลี้ยแป้งสีเทา 3 ภาพและเพลี้ยแป้งลาย 3 ภาพ (เพลี้ยแป้งสีเขียว ไม่มีข้อมูลภาพ)ผลการทดสอบแสดงในตารางที่ 3.2

มหาวิทยาลัยเทคโนโลยีสุรนารี

ตาราง 3.2: ผลการทดสอบประสิทธิภาพของโปรแกรม

ชนิดเพลี้ยแป้งฯ	จำนวนภาพทั้งหมด	จำแนกถูกต้อง(ภาพ)	จำแนกผิด (ภาพ)	% ความถูกต้อง
เพลี้ยแป้งสีชมพู	11	11	0	100%
เพลี้ยแป้งสีเทา	3	0	3	0%
เพลี้ยแป้งลาย	3	2	1	66.67%

จากการทดสอบกับภาพเพลี้ยแป้งมันสำปะหลังทั้ง 3ชนิด จำนวน 17 ภาพ พบว่าเทคนิค RGB Analysis เหมาะสมที่จะนำมาใช้แยกแยะชนิดเพลี้ยแป้ง สีชมพูได้ เพราะสามารถแยกเพลี้ยแป้งสีชมพูออกจากชนิดอื่นได้ ถูกต้อง 100% แต่อย่างไรก็ตามโปรแกรมดังกล่าวไม่สามารถแยกแยะเพลี้ยแป้งสีเทาและเพลี้ยแป้งลายได้ ในการใช้งานจริง อาจจะเลือกใช้เทคนิค RGB Analysis ในการจำแนกเพลี้ยแป้งสีชมพูออกมา กรณีที่ไม่ใช่เพลี้ยแป้งสีชมพู อาจจะต้องพัฒนาเทคนิคการประมวลผลแบบอื่นๆ ต่อไป

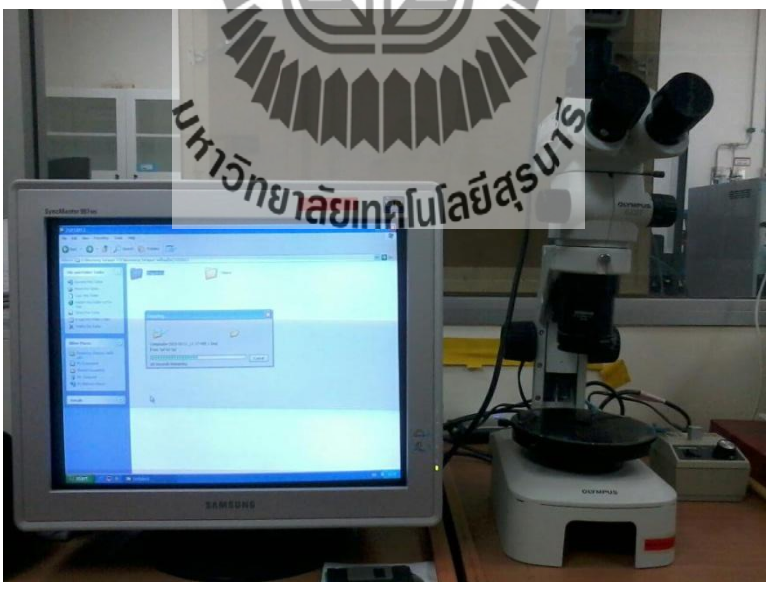
บทที่ 4

การทดสอบประสิทธิภาพของโปรแกรมประมวลผลภาพโดยใช้ภาพถ่ายจากStereo Microscope

ในบทนี้จะกล่าวถึงการทดสอบประสิทธิภาพของโปรแกรมโดยภาพที่ถ่ายจากกล้องจุลทรรศน์(Stereo Microscope) ที่มีคุณภาพในการ จับภาพด้วยความคมชัดและมีความละเอียดสูง กล้องชนิดนี้จะเป็นกล้องที่มีกำลังขยายไม่สูงมากนักซึ่งโดยปกติจะมีกำลังขยายไม่เกิน 100 เท่า แต่สามารถปรับ Zoom กำลังขยายได้ จึงทำให้บางครั้งถูกเรียกว่า Stereo Zoom Microscope ดังนั้นจะนำภาพถ่ายมาใช้เทคนิคการวิเคราะห์เนื้อภาพ (Texture Analysis) และเทคนิคการวิเคราะห์ค่าสี RGB (RGB Analysis) มาจำแนกชนิดของเพลี้ยแป้งมันสำปะหลัง

4.1 การถ่ายภาพด้วยกล้องจุลทรรศน์ (Stereo Microscope)

การถ่ายภาพเพลี้ยแป้งมันสำปะหลังด้วยกล้องจุลทรรศน์ (Stereo Microscope) เป็นการถ่ายภาพที่ผู้จัดทำได้เล็งเห็นว่าเป็นกล้องที่มีคุณภาพในการจับภาพที่มีความคมชัดลักษณะจะเป็นดังรูป



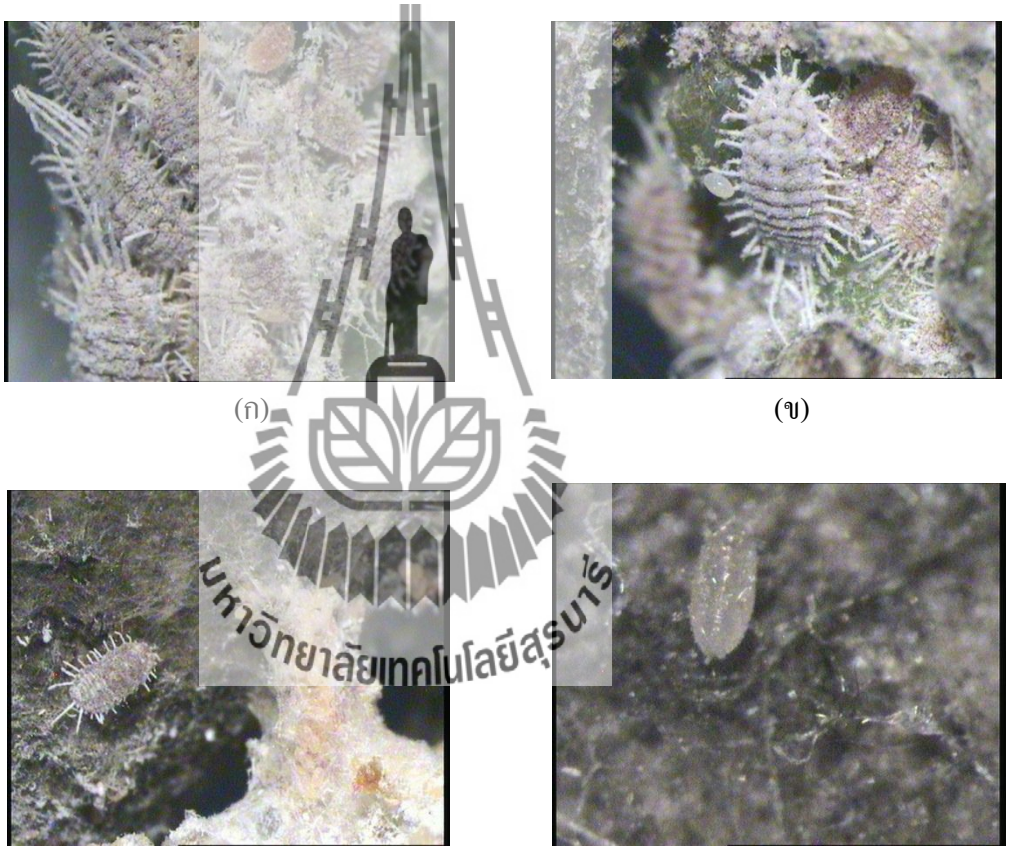
รูปที่ 4.1กล้องจุลทรรศน์ (Stereo Microscope)

(อุปกรณ์ที่ใช้ในการถ่ายภาพ: จากอาคารศูนย์เครื่องมือ1 (F1))

มหาวิทยาลัยเทคโนโลยีสุรนารี

4.2 ทดสอบถ่ายภาพด้วยกล้องจุลทรรศน์ (Stereo Microscope)

เพลี้ยแป้งมันสำปะหลังที่นำมาทดสอบ จะแบ่งออกเป็น 2 กลุ่มคือ เพลี้ยแป้งที่ได้จากการนำท่อนพันธุ์มามัดรวมกัน (การซุง) และเพลี้ยแป้งที่ได้จากแปลง ปลูกการทดสอบถ่ายภาพเพลี้ยแป้งมันสำปะหลังด้วยกล้อง จุลทรรศน์(Stereo Microscope) ที่เก็บตัวอย่างได้จากการนำท่อนพันธุ์มัดรวมกัน (การซุง) ซึ่งจะทำให้เพลี้ยแป้งขยายพันธุ์ได้รวดเร็วและมีปริมาณมาก ลักษณะที่สังเกตเห็นได้คือ เพลี้ยแป้งจะเกาะกันเป็นกลุ่มๆ ตามกิ่ง ตามลำต้น ดังแสดงในรูปที่ 4.2



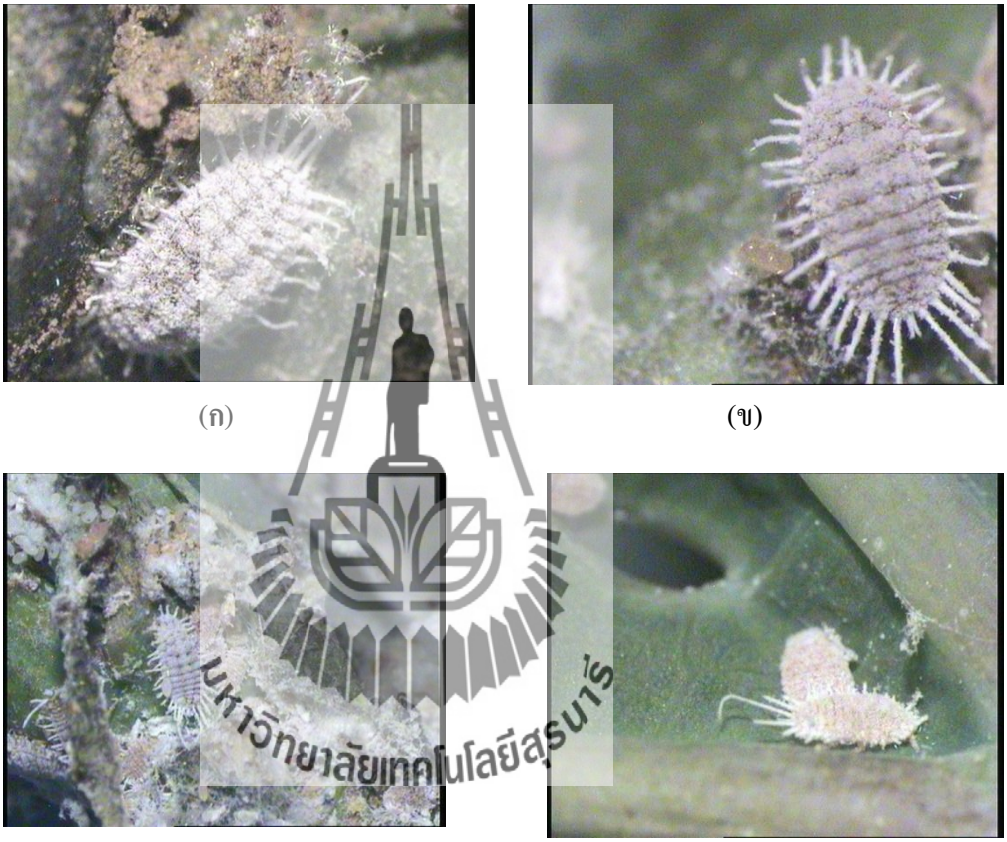
(ก) (ข)

(ค) (ง)

รูปที่ 4.2 เพลี้ยแป้งจากท่อนพันธ์มัดรวมกัน (ก) เพลี้ยแป้งที่เกาะกันเป็นกลุ่มๆ (ข) เพลี้ยแป้งอยู่ที่เปลือกไม้ (ค) ไข่ของเพลี้ยแป้ง และ (ง) ตัวอ่อนของเพลี้ยแป้ง

มหาวิทยาลัยเทคโนโลยีสุรนารี

การทดสอบถ่ายภาพเพลี้ยแป้งด้วยกล้องจุลทรรศน์ (Stereo Microscope) ที่เก็บตัวอย่างได้จากแปลงจะพบว่าเพลี้ยแป้งจะไม่เกาะกันเป็นกลุ่มๆจนแน่นเกินไ ป เนื่องจากแปลงมันสำปะหลังจะปลูกโดยมีระยะห่างเท่าๆกัน ทำให้เพลี้ยแป้งพัดปลิวไปตามทิศทางของลมเมื่อทำการถ่ายภาพจะได้ภาพที่เป็นรูปทรงและคมชัดมาก และส่วนมากจะอยู่จุดละๆ น้อยๆตัว ไม่ว่าเพลี้ยแป้งจะอยู่บริเวณไหน ก็สามารถที่จะจับภาพได้ อย่างชัดเจน ทำให้เราสะดวกสบายในการนำไปใช้วิเคราะห์ผลในขั้นตอนต่อไป



รูปที่ 4.3 เพลี้ยแป้งมันสำปะหลังจากแปลงปลูก (ก) เพลี้ยแป้งที่อาศัยอยู่ตามกิ่ง (ข) เพลี้ยแป้งที่อาศัยอยู่ตรงลำต้น (ค) เพลี้ยแป้งที่อาศัยอยู่ใกล้ๆ ยอด และ (ง) เพลี้ยแป้งที่อาศัยอยู่ตามใบ

มหาวิทยาลัยเทคโนโลยีสุรนารี

4.3ผลการทดสอบประสิทธิภาพโดยใช้เทคนิคการวิเคราะห์ค่าสี RGB (RGB Analysis)

เพลี้ยแป้งมันสำปะหลังที่ได้จากการถ่ายภาพด้วยกล้องจุลทรรศน์ (Stereo Microscope) และใช้ตัวอย่างจากการนำท่อนพันธุ์มัดรวมกัน (การซุง) ภาพที่ถ่ายได้ทั้งหมด 109 ภาพ โดยเป็นชนิดเพลี้ยแป้งสีเทาทั้งหมด เนื่องจากไม่สามารถหาตัวอย่างเพลี้ยแป้งชนิดอื่นได้ขณะทำโครงงานนี้ เราจะนำไปทดสอบประสิทธิภาพโปรแกรม โดยใช้เทคนิคการวิเคราะห์ค่าสี RGB (RGB Analysis) ยกตัวอย่าง20 ภาพ แสดงในภาคผนวก ง

เพลี้ยแป้งมันสำปะหลังที่ได้จากการถ่ายภาพด้วยกล้องจุลทรรศน์ (Stereo Microscope) และใช้ตัวอย่างจากแปลง ภาพที่ถ่ายได้ทั้งหมด 29 ภาพ โดยเป็นชนิดเพลี้ยแป้งสีเทาทั้งหมด ไม่สามารถหาเพลี้ยแป้งชนิดอื่นๆ ได้ในระหว่างการทำโครงงานนี้ เราจะนำไปทดสอบประสิทธิภาพโปรแกรม โดยใช้เทคนิคการวิเคราะห์ค่าสี RGB (RGB Analysis)ซึ่งยกตัวอย่าง 20 ภาพ แสดงในภาคผนวก จ

ผลการจำแนกชนิดเพลี้ยแป้งมันสำปะหลังโดยใช้โปรแกรมที่พัฒนาแสดงในตารางที่ 4.3

ตาราง 4.3 ผลการจำแนกชนิดเพลี้ยแป้งมันสำปะหลังโดยใช้โปรแกรมที่พัฒนา

แหล่งที่มา	ภาพทดสอบ	ผลการจำแนก			
		เพลี้ยแป้งสีชมพู	เพลี้ยแป้งสีเขียว	เพลี้ยแป้งลาย	เพลี้ยแป้งสีเทา
ท่อนพันธ์	เพลี้ยแป้งสีเทา 109 ภาพ	95 ภาพ (87.12%)	11 ภาพ (10.09%)	1 ภาพ (0.92%)	- (0%)
แปลงปลูก	เพลี้ยแป้งสีเทา 25 ภาพ	23 ภาพ (92%)	1 ภาพ (4%)	1 ภาพ (4%)	- (0%)

จากการทดสอบประสิทธิภาพด้วยเทคนิคการวิเคราะห์ค่าสี RGB (RGB Analysis) พบว่าโปรแกรมไม่สามารถจำแนกชนิดเพลี้ยแป้งสีเทาได้ถูกต้อง และยังมีการจำแนกชนิดเพลี้ยแป้งสีเทา

เป็นเพลี้ยแป้งสีชมพู ซึ่งไม่สอดคล้องกับวัตถุประสงค์ของการใช้เทคนิค RGB Analysis ที่ต้องการจำแนกเพลี้ยแป้งสีชมพูออกจากกลุ่มเพลี้ยแป้งชนิดอื่นๆ การจำแนกที่ผิดพลาดของโปรแกรมอาจเกิดจากที่ กลุ่มโครงงานมีฐานข้อมูลน้อยเกินไป และ ภาพที่ถ่ายได้จากกล้องจุลทรรศน์ (Stereo Microscope)อาจจะเกิดเนื่องจากไม่ได้ตั้งค่าการถ่ายภาพของกล้องดิจิตอลให้สอดคล้องกับแหล่งกำเนิดแสงของกล้องจุลทรรศน์ ทำให้ภาพเพลี้ยแป้งสีเทามีโทนออกแดงเหลืองตามแหล่งกำเนิดแสง (หลอดทังสเตนโลเจน) คล้ายกับเพลี้ยแป้งสีชมพู

จากผลการทดลองใช้ภาพถ่ายจากกล้องจุลทรรศน์ พบว่าโปรแกรม RGB Analysis จำแนกผิดพลาดค่อนข้างมาก ขณะที่การทดสอบโปรแกรมโดยใช้ภาพถ่ายจากอินเตอร์เน็ต กลับให้ผลที่ถูกต้องแม่นยำในการจำแนกเพลี้ยแป้งสีชมพู แสดงให้เห็นว่า เทคนิคการจำแนกชนิดโดยใช้สีจะขึ้นกับเงื่อนไขการถ่ายภาพค่อนข้างมาก

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 5

ประสิทธิภาพโปรแกรมจากภาพถ่ายเพลี้ยแป้งมันสำปะหลังด้วยกล้อง Mobile Microscope

โครงงานนี้เปรียบเสมือนเป็นหน่วยงานที่ทำหน้าที่จำแนกชนิดเพลี้ยแป้งมันสำปะหลังด้วยเทคนิคการประมวลผลภาพเพื่อวิเคราะห์เนื้อภาพแล้วจำแนกชนิดของเพลี้ยแป้งมันสำปะหลัง เมื่อเกษตรกรได้เก็บข้อมูลภาพของเพลี้ยแป้งมันสำปะหลังแล้วส่งมาให้หน่วยงานนี้ หน่วยงานนี้ก็จะทำการโปรแกรมประมวลผลภาพจำแนกระบุชนิดของเพลี้ยแป้งมันสำปะหลัง แล้วส่งผลข้อมูลชนิดของเพลี้ยแป้งมันสำปะหลังกลับไปยังเกษตรกรได้ทราบผล

รูปที่5.1 แสดงกระบวนการทำงาน

มหาวิทยาลัยเทคโนโลยีสุรนารี

การเก็บข้อมูลภาพเพลี้ยแป้งมันสำปะหลังซึ่งก็คือการถ่ายภาพเพลี้ยแป้งมันสำปะหลังนั้นเอง การถ่ายภาพเพลี้ยแป้งมันสำปะหลังจะใช้กล้องถ่ายรูปทั่วไปในปัจจุบันถ่ายภาพเพลี้ยแป้งมันสำปะหลังไม่ได้ เพราะว่าเพลี้ยแป้งมันสำปะหลังมีขนาดเล็กมากและกล้องถ่ายรูปทั่วไปในปัจจุบันไม่ได้ถูกออกแบบมาให้ถ่ายรูปในระยะจุดโฟกัสที่ใกล้มากๆได้ แต่เนื่องจากได้มีผู้คิดค้นอุปกรณ์ใช้งานร่วมกับโทรศัพท์มือถือที่สามารถถ่ายภาพเพลี้ยแป้งมันสำปะหลังผ่านกล้องโทรศัพท์มือถือขึ้นมาแล้ว จึงได้ขอยืมอุปกรณ์มาใช้ทำการทดลองถ่ายภาพเพลี้ยแป้งมันสำปะหลังเพื่อวัดผลประสิทธิภาพการประมวลผลภาพด้วยโปรแกรมและประสิทธิภาพการถ่ายภาพเพลี้ยแป้งมันสำปะหลังด้วยโทรศัพท์มือถือ

(ก)

(ข)

รูปที่ 5.2 (ก) อุปกรณ์ที่ใช้ติดบนกล้องโทรศัพท์มือถือ (ข) ติดตั้งอุปกรณ์กับโทรศัพท์มือถือ ผู้คิดค้นอุปกรณ์ คือ นางสาวปนัดดา สมัยกลาง สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์ชั้นปีที่ 3

มหาวิทยาลัยเทคโนโลยีสุรนารี

5.1 แหล่งปลูกมันที่เก็บตัวอย่างเพลี้ยแป้งมันสำปะหลังมา

เพลี้ยแป้งมันสำปะหลังได้จากแปลงปลูก 2 แปลง คือ แปลงปลูกของนาย สงบ แทวกระโทก อำเภอจักราช จังหวัดนครราชสีมา และแปลงปลูกของ นายวงศ์ สายสนอง อำเภอโนนแดง จังหวัดบุรีรัมย์ พบเฉพาะเพลี้ยแป้งมันสำปะหลังสีเทา ไม่พบเพลี้ยแป้งชนิดอื่นเลยระหว่างการทำโครงงานนี้

5.2การถ่ายภาพเพลี้ยแป้งมันสำปะหลังด้วย Mobile Microscope

เมื่อได้เก็บตัวอย่างเพลี้ยแป้งมันสำปะหลังมาจากแปลงปลูกแล้ว ควรคัดเพลี้ยแป้งมันสำปะหลังมาวางไว้ตัวเดียวบนใบมันสำปะหลัง เพื่อภาพที่ถ่ายออกมาได้นั้นมีสีแตกต่างกันอย่างชัดเจนระหว่างสีของตัวเพลี้ยแป้งมันสำปะหลังกับสีพื้นซึ่งก็คือสีของใบมันสำปะหลังที่มีสีเขียว และกล้องมือถือจะสามารถโฟกัสภาพเพลี้ยแป้งมันสำปะหลังตัวเดียวได้ละเอียดและง่ายกว่าถ่ายภาพเพลี้ยแป้งมันสำปะหลังที่อยู่เป็นกลุ่มตัวอย่างแสดงด้วยรูปที่ 5.3

(ก)

(ข)

รูปที่ 5.3 (ก) เก็บตัวอย่างเพลี้ยแป้งสีเทา (ข) ภาพเพลี้ยแป้งสีเทาที่ถ่ายได้

การถ่ายภาพเพลี้ยแป้งมันสำปะหลังด้วย Mobile Microscopeมีหลายปัจจัยในการรบกวนการถ่ายภาพเพลี้ยแป้งมันสำปะหลัง ปัจจัยในการรบกวนนี้ทำให้ภาพที่ถ่ายออกมาออกมามีคุณภาพความชัดของภาพน้อย เช่น แสง ลม การเคลื่อนที่ของตัวเพลี้ย การสั่นของมือผู้ถ่ายและความละเอียดของกล้องมือถือ เป็นต้น ตัวอย่างภาพเพลี้ยแป้งมันสำปะหลังถ่ายที่เงื่อนไขการถ่ายภาพต่างๆ แสดงในรูปที่ 5.4

(ก)

(ข)

(ค)

(ง)

รูปที่ 5.4 ภาพเพลี้ยแป้งต่างลักษณะ (ก) มีการสั่นของมือผู้ถ่าย (ข) มีแสงสว่างมาก (ค) มีการเคลื่อนไหวของตัวเพลี้ยแป้งมันสำปะหลัง (ง) มีปัจจัยการรบกวนน้อย

จากการได้ทดสอบภาพถ่ายเพลี้ยแป้งมันสำปะหลังสีเทาด้วย Mobile Microscopeจำนวน 20 ภาพ แสดงในภาคผนวก ฉ ไม่พบว่าโปรแกรมประมวลผลภาพด้วยเทคนิค RGB Analysis ประมวลผลภาพได้เป็นเพลี้ยแป้งสีเทา แต่พบว่าโปรแกรมประมวลผลภาพเพลี้ยแป้งสีเทาเป็นเพลี้ยแป้งสีเขียว7 ภาพ และเพลี้ยแป้งลาย 13 ภาพ อย่างไรก็ตามโปรแกรมประมวลผลภาพด้วยเทคนิค RGB Analysisไม่ได้ประมวลผลภาพเพลี้ยแป้งสีเทาเป็นเพลี้ยแป้งสีชมพูเลย แสดงว่าเทคนิคการประมวลผลภาพนี้อาจจะเหมาะกับการจำแนกชนิดเพลี้ยแป้งสีชมพู ในกรณีการถ่ายภาพด้วย Mobile Microscop โดยใช้ภาพถ่ายจากกล้องจุลทรรศน์ (Stereo Microscop) กลับพบว่ามีการจำแนกผิด โดยจำแนกเพลี้ยแป้งสีเทาเป็นเพลี้ยแป้งสีชมพู ทั้งสองกรณีให้ผลต่างกันมาก อาจจะเป็นเพราะแหล่งกำเนิดแสงและการตั้งค่าในการถ่ายภาพ ซึ่งกล้อง Mobile microscop ใช้แหล่งกำเนิดแสงจากหลอดไฟฟลูออเรสเซนต์ และปกติกล้องจะปรับค่าอัตโนมัติ สีของภาพถ่ายจึงจะใกล้เคียงกับรูปจากอินเตอร์เน็ตมากกว่า ขณะที่กล้องจุลทรรศน์ Stereo Microscop ใช้แหล่งกำเนิดแสงแบบทังสเตนฮาโลเจน ซึ่งจะมีโทนสีช่วงแดงออกมามาก ถ้าการตั้งค่ากล้องถ่ายรูปไม่เหมาะสมอาจทำให้ภาพสีเปลี่ยนไป

มหาวิทยาลัยเทคโนโลยีสุรนารี

บทที่ 6

สรุป

โครงงานนี้นำเสนอการจำแนกชนิดเพลี้ยแป้งมันสำปะหลังด้วยการประมวลผลภาพ โดยใช้เทคนิคในการประมวลผลภาพ 2 เทคนิคคือ เทคนิคการวิเคราะห์เนื้อภาพ (Texture Analysis) และเทคนิคการวิเคราะห์ค่าสี RGB (RGB Analysis)

เทคนิคการวิเคราะห์เนื้อภาพ ด้วยการวิเคราะห์ความสัมพันธ์ของพิกเซลที่ติดกันของค่าระดับสีที่เรียกว่า Gray-level Co-occurrence Matrix (GLCM) และใช้ข้อมูลภาพจากการคำนวณค่า Homogeneity, Contrast, Dissimilarity, Mean, Standard Deviation, Entropy, Angular Second Moment และ Correlation ภาพที่นำมาวิเคราะห์เนื้อภาพจะผ่านกระบวนการคัดกรองเบื้องต้น (Pre-processing) เพื่อทำให้ภาพปรากฏลักษณะเด่นขึ้นมาอย่างชัดเจน จากนั้นจึงนำภาพดังกล่าวไปวิเคราะห์เนื้อภาพด้วยการคำนวณเนื้อภาพ (Texture) ผลจากการคำนวณปรากฏว่าไม่มีพารามิเตอร์ใดจากทั้ง 8 พารามิเตอร์ดังกล่าวที่สามารถใช้จำแนกชนิดเพลี้ยแป้งมันสำปะหลังได้อย่างชัดเจน

เทคนิคการวิเคราะห์ค่าสี RGB เป็นเทคนิคที่ต้องใช้ข้อมูลภาพต้นแบบของเพลี้ยแป้งมันสำปะหลังทั้ง 4 ชนิดมาเป็นฐานข้อมูลสำหรับใช้ในการวิเคราะห์เพื่อจำแนกชนิดของภาพเพลี้ยแป้งมันสำปะหลังที่ไม่ทราบชนิด โดยอาศัยวิธีการทำให้ภาพเพลี้ยแป้งมันสำปะหลังที่เป็นต้นแบบและภาพเพลี้ยแป้งมันสำปะหลังที่ไม่ทราบชนิดไม่ขึ้นอยู่กับความเข้มแสง (Brightness) หลักการคือ นำภาพแต่ละภาพมาหาค่าเฉลี่ยของสี R (red), G (green) และ B (blue) แล้วนำค่าเฉลี่ยแต่ละค่าที่ได้ เข้าสู่กระบวนการทำให้ภาพไม่ขึ้นอยู่กับความเข้มแสง จากนั้นนำค่าเฉลี่ยของภาพดังกล่าวของเพลี้ยแป้งมันสำปะหลังที่ไม่ทราบชนิดมาจำแนกชนิด โดยหาระยะทางสีเทียบกับค่าเฉลี่ยของภาพต้นแบบของเพลี้ยแป้งมันสำปะหลังทั้ง 4 ชนิด ถ้าระยะทางสีมีค่าความใกล้เคียงภาพต้นแบบใดที่สุด แสดงว่าเพลี้ยแป้งมันสำปะหลังที่ไม่ทราบชนิดนั้นเป็นชนิดเดียวกันกับภาพต้นแบบเพลี้ยแป้งมันสำปะหลังที่ถูนำมาเปรียบเทียบนั้น

ผลการทดสอบประสิทธิภาพของโปรแกรมโดยใช้ข้อมูลภาพจากอินเตอร์เน็ต 17 ภาพในการทดลอง พบว่าเทคนิคการวิเคราะห์ค่าสี RGB สามารถทำนายเพลี้ยแป้งสีชมพูได้ถูกต้อง 100% แต่ไม่สามารถจำแนกเพลี้ยแป้งมันสำปะหลังชนิดอื่นๆ ได้

เมื่อนำภาพถ่ายจากกล้องจุลทรรศน์ (Stereo Microscope) มาทดสอบ ผลการทดสอบกลับตรงกันข้ามกับการทดสอบโดยใช้ภาพจากอินเตอร์เน็ต กล่าวคือ โปรแกรมทำนายเพลี้ยแป้งสีเทาเป็นเพลี้ยแป้งสีชมพูในอัตราส่วนที่สูง เมื่อใช้ภาพถ่ายจากกล้อง Mobile Microscope มาทดสอบกลับพบว่า แม้จะทำนายเพลี้ยแป้งสีเทาไม่ถูกทั้งหมด แต่ไม่มีการทำนายภาพถ่ายว่าเป็นเพลี้ยแป้งสีชมพูเลย

การที่ใช้อุปกรณ์ถ่ายภาพเพลี้ยแป้ง 2 ชนิด ที่ต่างกันมาทดสอบประสิทธิภาพของโปรแกรมและให้ผลต่างกันนั้น แสดงว่าโปรแกรมที่ใช้เทคนิคการวิเคราะห์ค่าสี RGB ขึ้นอยู่กับเงื่อนไขการถ่ายภาพค่อนข้างมาก กล้อง Mobile Microscope ถ่ายโดยใช้แสงจากหลอดไฟฟลูออเรสเซนต์ โดยไม่มีการปรับตั้งค่ากล้องในโทรศัพท์ อาจจะให้สีแตกต่างกันจากกล้องจุลทรรศน์ (Stereo Microscope) ที่ใช้หลอดทังสเตนฮาโลเจนเป็นแหล่งกำเนิดแสง และผู้ใช้ต้องปรับค่าการถ่ายภาพของกล้องให้สอดคล้องกัน ถ้าการปรับค่าดังกล่าวทำได้ไม่ถูกต้อง จะทำให้ภาพเกิดสีที่แตกต่างจากที่ควรจะเป็น ในอนาคตจึงควรจะมีการพัฒนาเทคนิคการประมวลผลภาพเพิ่มเติม เพื่อให้สามารถจำแนกภาพเพลี้ยแป้งมันสำปะหลังได้อย่างถูกต้องต่อไป

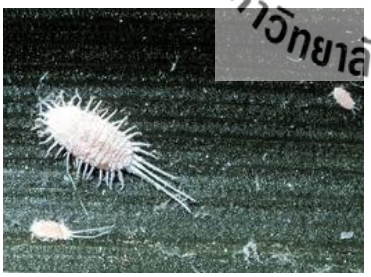
มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคผนวก ก

ตาราง 2.1 แสดงภาพต้นแบบเพลี้ยแป้งมันสำปะหลังที่ใช้เป็นฐานข้อมูล

ภาพต้นแบบเพลี้ยแป้งมันสำปะหลังที่ใช้เป็นฐานข้อมูล	ชื่อชนิดภาพต้นแบบเพลี้ยแป้งมันสำปะหลัง
	เพลี้ยแป้งสีเขียว
	เพลี้ยแป้งสีเขียว
	เพลี้ยแป้งสีเขียว
	เพลี้ยแป้งสีเขียว
	เพลี้ยแป้งสีชมพู

มหาวิทยาลัยเทคโนโลยีสุรนารี

	เพลี้ยแป้งสีชมพู
	เพลี้ยแป้งสีชมพู
	เพลี้ยแป้งสีเทา
	เพลี้ยแป้งสีเทา
	เพลี้ยแป้งสีเทา
	เพลี้ยแป้งสีเทา

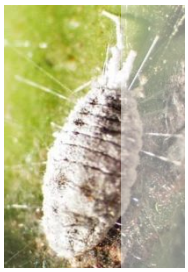
มหาวิทยาลัยเทคโนโลยีสุรนารี

	เพลี้ยแป้งสีเทา
	เพลี้ยแป้งสีเทา
	เพลี้ยแป้งสีเทา
	เพลี้ยแป้งสีเทา
	เพลี้ยแป้งสีเทา
	เพลี้ยแป้งสีเทา

มหาวิทยาลัยเทคโนโลยีสุรนารี

	เพลี้ยแป้งสีเทา
	เพลี้ยแป้งสีเทา
	เพลี้ยแป้งสีเทา
	เพลี้ยแป้งสีเทา
	เพลี้ยแป้งสีเทา
	เพลี้ยแป้งสีเทา

มหาวิทยาลัยเทคโนโลยีสุรนารี

	เพลี้ยแป้งลาย
	เพลี้ยแป้งลาย
	เพลี้ยแป้งลาย
	เพลี้ยแป้งลาย
	เพลี้ยแป้งลาย
	เพลี้ยแป้งลาย

มหาวิทยาลัยเทคโนโลยีสุรนารี

	เพลี้ยแป้งลาย
	เพลี้ยแป้งลาย
	เพลี้ยแป้งลาย

มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคผนวก ข

ตาราง 3.1 ภาพต้นแบบพร้อมด้วยบริเวณที่สนใจของภาพต้นแบบของเพลี้ยแป้งมันสำปะหลัง ทั้ง 4 ชนิด

ภาพต้นแบบเพลี้ยแป้งมันสำปะหลังที่ใช้เป็นฐานข้อมูล	บริเวณที่สนใจของภาพต้นแบบ (50x40 pixel)	ชื่อชนิดภาพต้นแบบเพลี้ยแป้งมันสำปะหลัง
	Crop Image	เพลี้ยแป้งสีเขียว
	Crop Image	เพลี้ยแป้งสีเขียว
	Crop Image	เพลี้ยแป้งสีเขียว
	Crop Image	เพลี้ยแป้งสีเขียว

มหาวิทยาลัยเทคโนโลยีสุรนารี

	Crop Image	เพลี้ยแป้งสีชมพู
	Crop Image	เพลี้ยแป้งสีชมพู
	Crop Image	เพลี้ยแป้งสีชมพู
	Crop Image	เพลี้ยแป้งสีเทา
	Crop Image	เพลี้ยแป้งสีเทา
	Crop Image	เพลี้ยแป้งสีเทา

มหาวิทยาลัยเทคโนโลยีสุรนารี

	Crop Image	เพลี้ยแป้งสีเทา
	Crop Image	เพลี้ยแป้งสีเทา
	Crop Image	เพลี้ยแป้งสีเทา
	Crop Image	เพลี้ยแป้งสีเทา
	Crop Image	เพลี้ยแป้งสีเทา
	Crop Image	เพลี้ยแป้งสีเทา

มหาวิทยาลัยเทคโนโลยีสุรนารี

	Crop Image	เพลี้ยแป้งสีเทา
	Crop Image	เพลี้ยแป้งสีเทา
	Crop Image	เพลี้ยแป้งสีเทา
	Crop Image	เพลี้ยแป้งสีเทา
	Crop Image	เพลี้ยแป้งสีเทา
	Crop Image	เพลี้ยแป้งสีเทา

มหาวิทยาลัยเทคโนโลยีสุรนารี

	Crop Image	เพลี้ยแป้งสีเทา
	Crop Image	เพลี้ยแป้งลาย
	Crop Image	เพลี้ยแป้งลาย
	Crop Image	เพลี้ยแป้งลาย
	Crop Image	เพลี้ยแป้งลาย
	Crop Image	เพลี้ยแป้งลาย

มหาวิทยาลัยเทคโนโลยีสุรนารี

	Crop Image	เพลี้ยแป้งลาย
	Crop Image	เพลี้ยแป้งลาย
	Crop Image	เพลี้ยแป้งลาย
	Crop Image	เพลี้ยแป้งลาย

มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคผนวก ค

ตาราง 3.3ภาพต้นแบบเพลี้ยแป้งมันสำปะหลังจากอินเตอร์เน็ตที่ใช้ทดสอบประสิทธิภาพของโปรแกรมที่พัฒนาขึ้น

ภาพอินพุตเพลี้ยแป้งมันสำปะหลังที่ใช้ทดสอบโปรแกรม	ชื่อชนิดภาพอินพุตเพลี้ยแป้งมันสำปะหลัง
	เพลี้ยแป้งสีชมพู
	เพลี้ยแป้งสีชมพู
	เพลี้ยแป้งสีชมพู
	เพลี้ยแป้งสีชมพู
	เพลี้ยแป้งสีชมพู

มหาวิทยาลัยเทคโนโลยีสุรนารี

	เพลี้ยแป้งสีชมพู
	เพลี้ยแป้งสีชมพู
	เพลี้ยแป้งสีชมพู
	เพลี้ยแป้งสีชมพู
	เพลี้ยแป้งสีชมพู
	เพลี้ยแป้งสีชมพู

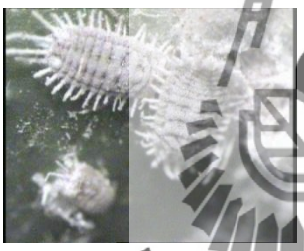
มหาวิทยาลัยเทคโนโลยีสุรนารี

	เพลี้ยแป้งสีเทา
	เพลี้ยแป้งสีเทา
	เพลี้ยแป้งสีเทา
	เพลี้ยแป้งลาย
	เพลี้ยแป้งลาย
	เพลี้ยแป้งลาย

มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคผนวก ง

ตาราง 4.1 ตัวอย่างที่ได้จากการนำท่อนพันธุ์มัดรวมกัน (การซุง)

จำนวน	ภาพอินพุต	ชนิด	ผลการทดสอบ
1.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
2.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
3.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีเขียว
4.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งลาย
5.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู

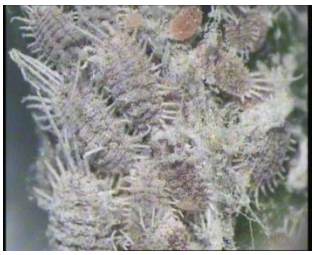
มหาวิทยาลัยเทคโนโลยีสุรนารี

6.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
7		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
8.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
9.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
10.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
11.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู

มหาวิทยาลัยเทคโนโลยีสุรนารี

12.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
13.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
14.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
15.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
16.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
17.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู

มหาวิทยาลัยเทคโนโลยีสุรนารี

18.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
19		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีเขียว
20		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู

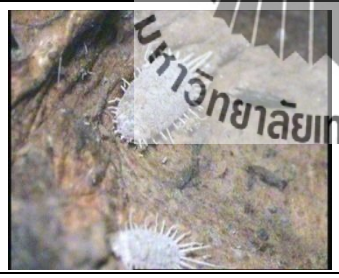
มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคผนวก จ

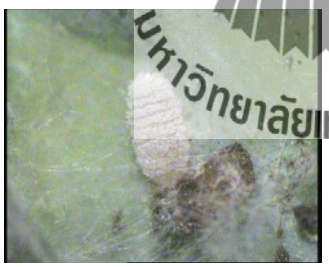
ตาราง 4.2 ตัวอย่างที่ได้จากการแปลงปลูกมันสำปะหลัง

จำนวน	ภาพอินพุต	ชนิด	ผลทดสอบ
1.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
2.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีเขียว
3.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
4.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
5.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งลาย

มหาวิทยาลัยเทคโนโลยีสุรนารี

6.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งชมพู
7.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
8.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
9.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
10.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
11.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู

มหาวิทยาลัยเทคโนโลยีสุรนารี

12.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
13.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
14.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
15		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
16		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
17.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู

มหาวิทยาลัยเทคโนโลยีสุรนารี

18		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
19.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู
20.		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีชมพู

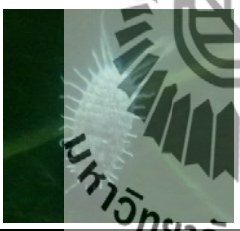
มหาวิทยาลัยเทคโนโลยีสุรนารี

ภาคผนวก ฉ

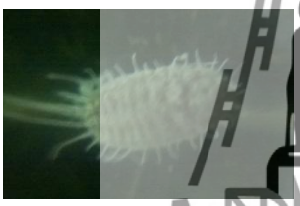
ตารางที่ 5.1ผลทดสอบประสิทธิภาพการจำแนกชนิดด้วยเทคนิค RGB Analysis โดยใช้ Mobile Microscop

ภาพที่	ภาพอินพุต	ข้อมูลที่มีอยู่	ผลการจำแนกชนิด
1		เพลี้ยแป้งสีเทา	เพลี้ยแป้งลาย
2		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีเขียว
3		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีเขียว
4		เพลี้ยแป้งสีเทา	เพลี้ยแป้งลาย
5		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีเขียว
6		เพลี้ยแป้งสีเทา	เพลี้ยแป้งลาย

มหาวิทยาลัยเทคโนโลยีสุรนารี

7		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีเขียว
8		เพลี้ยแป้งสีเทา	เพลี้ยแป้งเขียว
9		เพลี้ยแป้งสีเทา	เพลี้ยแป้งลาย
10		เพลี้ยแป้งสีเทา	เพลี้ยแป้งลาย
11		เพลี้ยแป้งสีเทา	เพลี้ยแป้งลาย
12		เพลี้ยแป้งสีเทา	เพลี้ยแป้งลาย
13		เพลี้ยแป้งสีเทา	เพลี้ยแป้งลาย

มหาวิทยาลัยเทคโนโลยีสุรนารี

14		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีเขียว
15		เพลี้ยแป้งสีเทา	เพลี้ยแป้งลาย
16		เพลี้ยแป้งสีเทา	เพลี้ยแป้งลาย
17		เพลี้ยแป้งสีเทา	เพลี้ยแป้งสีเขียว
18		เพลี้ยแป้งสีเทา	เพลี้ยแป้งลาย
19		เพลี้ยแป้งสีเทา	เพลี้ยแป้งลาย
20		เพลี้ยแป้งสีเทา	เพลี้ยแป้งลาย

มหาวิทยาลัยเทคโนโลยีสุรนารี

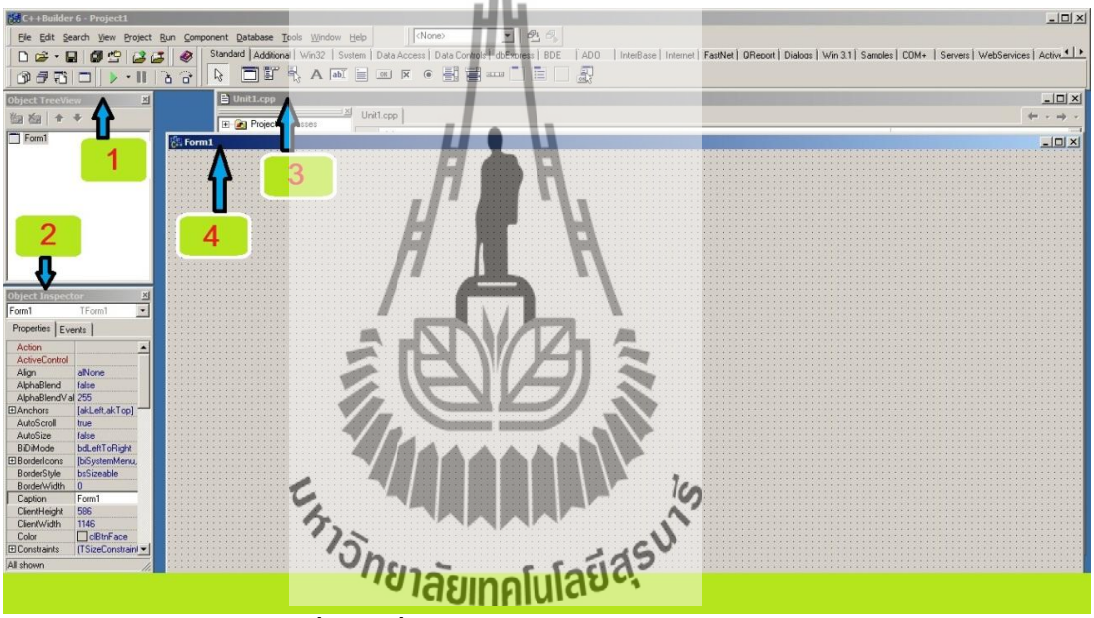
ภาคผนวก ช

การใช้งานโปรแกรม Borland C++ Builder 6 เพื่อประมวลผลภาพ

ด้วยเทคนิคการวิเคราะห์เนื้อภาพ

การจำแนกชนิดของเพลี้ยแป้งมันสำปะหลังที่อธิบายไว้ข้างต้น ใช้โปรแกรมที่ชื่อว่า Borland C++ Builder 6เป็นตัวช่วยในการจำแนกชนิดของเพลี้ยแป้ง เป็นวิธีที่เกี่ยวกับการวิเคราะห์เนื้อภาพ สามารถวิเคราะห์เนื้อภาพด้วยขั้นตอนในโปรแกรมดังนี้

ขั้นตอนที่ 1:เปิดโปรแกรม **Borland C++ Builder 6** ขึ้นมา ดังแสดงใน**รูปก-1**



รูปก-1: แสดงหน้าจอเริ่มต้นเมื่อทำการเปิดโปรแกรม Borland C++ Builder 6

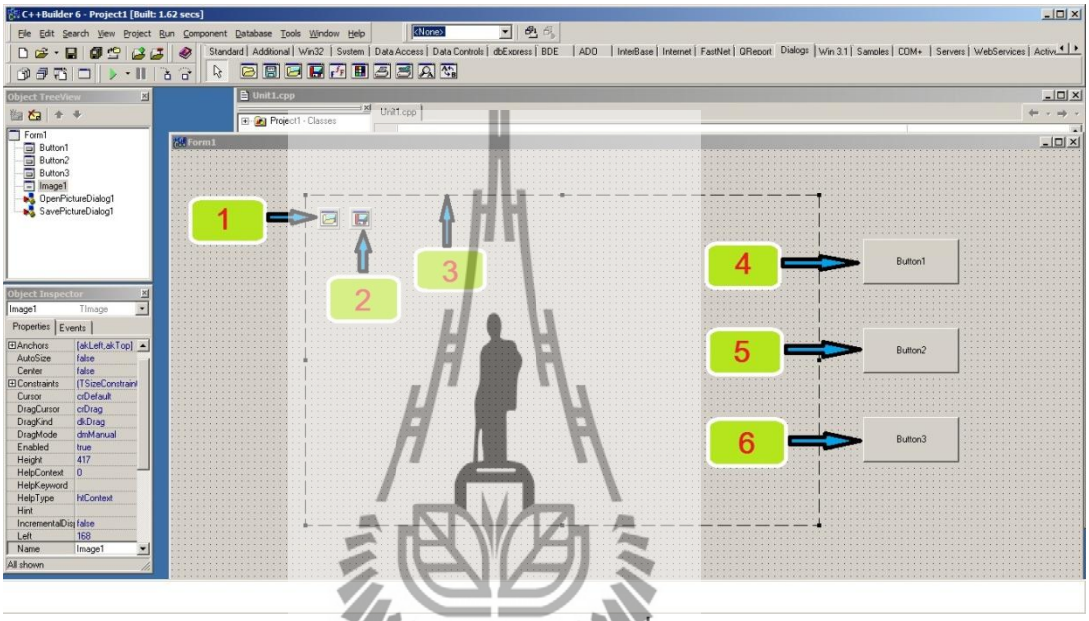
1:คือ **Object TreeView** เป็นหน้าต่างเครื่องมือแสดงการใช้งานของ Form (หมายเลข 4) ในกรณีที่เปิดใช้งานหลาย Form เช่น ผู้ใช้กำลังใช้งาน Form1 อยู่ Object TreeView ก็จะแสดง Form1 เป็นต้น

2:คือ **Object Inspector**เป็นหน้าต่างเครื่องมือที่ประกอบไปด้วย Properties และ Event สำหรับกำหนดคุณลักษณะและกำหนดการทำงานที่ให้เกิดขึ้นบน Form

3:คือ **Unit1.cpp**เป็นหน้าต่างสำหรับเขียน Code เพื่อกำหนดให้ Form1 ทำงานตามคำสั่งที่กำหนด กรณีผู้ใช้เปิด Form2 โปรแกรมจะสร้าง Unit2.cpp ขึ้นมาโดยอัตโนมัติ

4:คือ **Form** เป็นหน้าต่างสำหรับสร้าง Object ให้เกิดขึ้นตามวัตถุประสงค์ สร้างโดยการวาง Component ต่างๆลงบน Form เช่น Button, Radio Button, Image, Memo เป็นต้น

ขั้นตอนที่ 2: สร้าง Object ให้เกิดขึ้นบน Form โดยการวาง Component ต่างๆดัง **รูปก-2** และหลังจากวางComponent ต่างๆเป็นที่เรียบร้อยแล้ว ที่ Unit1.cpp จะปรากฏ code ที่โปรแกรมสร้างขึ้นให้โดยอัตโนมัติดัง**รูปก-3** นอกเหนือจากโปรแกรมจะสร้าง Unit1.cpp ให้โดยอัตโนมัติแล้ว Borland C++ Builder 6 ยังสร้าง Unit1.h ให้อัตโนมัติอีกด้วย ดัง**รูปก-4**



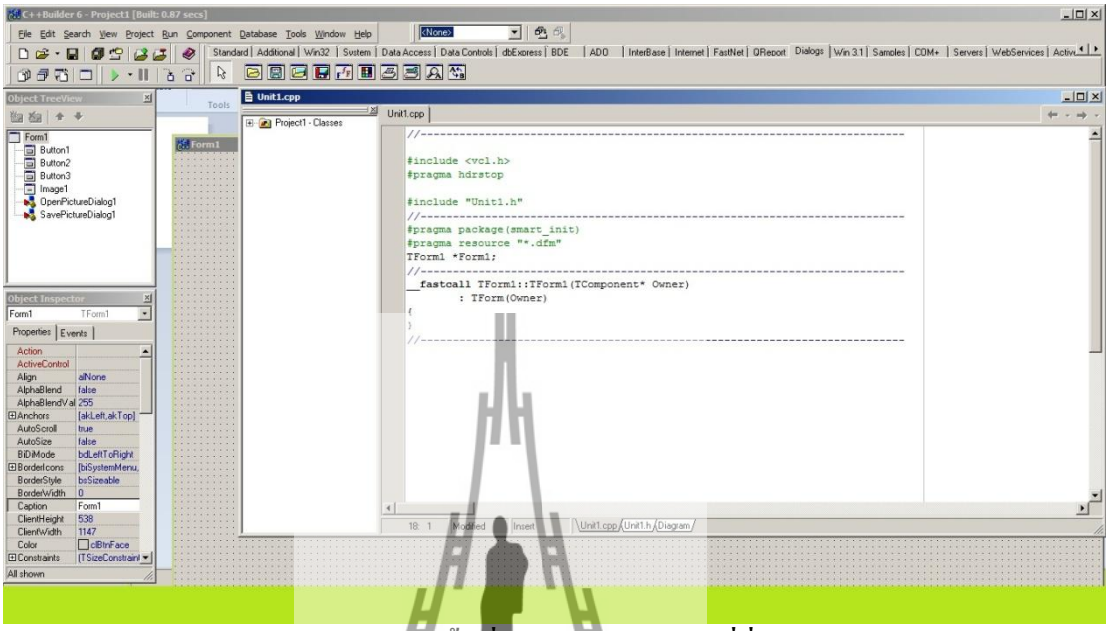
รูป ก-2: แสดง Component ต่าง ๆที่ถูกวางลงบนForm

1:คือ Component ที่ชื่อว่า **OpenDialogPicture** สัญลักษณ์ของ OpenDialogPicture คือ สามารถเรียกใช้ OpenDialogPicture จาก Menu bar เลือกไปที่ Dialogs จากนั้นคลิกเลือก OpenDialogPicture เพื่อนำมาวางบน Form

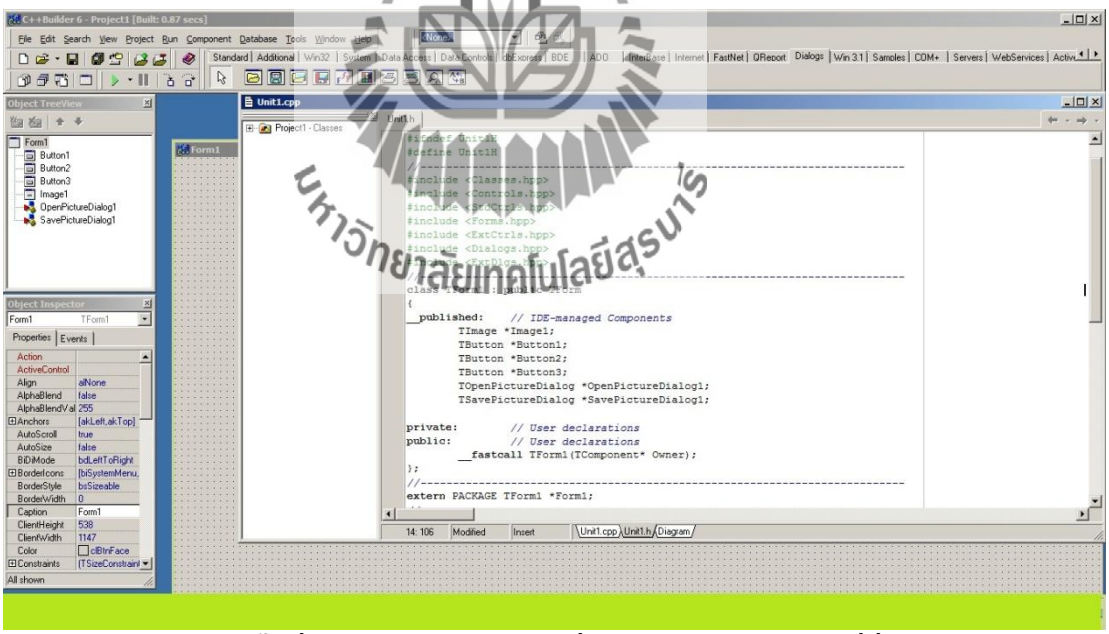
2:คือ Component ที่ชื่อว่า **SaveDialogPicture** สัญลักษณ์ของ SaveDialogPicture คือ สามารถเรียกใช้ SaveDialogPicture จาก Menu bar เลือกไปที่ Dialogs จากนั้นคลิกเลือก SaveDialogPicture เพื่อนำมาวางบน Form

3:คือ Component ที่ชื่อว่า**Image** สัญลักษณ์ของ Image คือ สามารถ เรียกใช้ Image จาก Menu bar เลือกไปที่ Additional จากนั้นคลิกเลือก Image เพื่อนำมาวางบน Form

4, 5 และ 6: คือ **Button1, Button2** และ**Button3** ตามลำดับ สัญลักษณ์ของ Button คือ สามารถเรียกใช้Button จาก Menu bar เลือกไปที่Standard จากนั้นคลิกเลือก Button เพื่อนำมาวางบน Form



รูป ก-3: แสดงพื้นที่สำหรับเขียน Code ที่ชื่อว่า Unit1.cpp



รูป ก-4: แสดงพื้นที่สำหรับประกาศตัวแปรที่จะใช้ในการเขียน code ที่ชื่อว่า Unit1.h

ขั้นตอนที่ 3: ดำเนินการเปลี่ยนชื่อ Button1 เป็น “Open Image” , Button2 เป็น “Matrix” และ Button3 เป็น “Feature” โดยดำเนินการเปลี่ยนดังต่อไปนี้

- ดับเบิ้ลคลิกที่ **Button1** โปรแกรมจะสลับหน้าต่างจาก Form1 เป็นUnit1.cpp ให้สำหรับเขียน Code ให้ผู้ใช้ทำการเขียน Code ดังนี้

  Button1->Caption= “Open Image”;

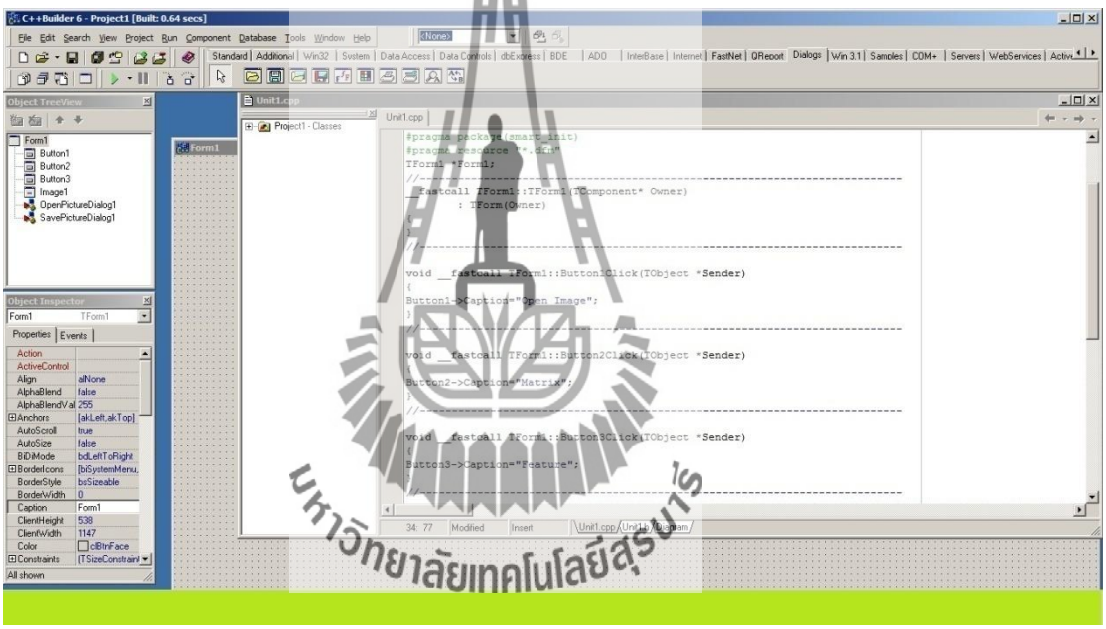
- กลับไปที่ Form1 ดับเบิ้ลคลิกที่ **Button2** โปรแกรมจะสลับหน้าต่างจาก Form1 เป็นUnit1.cpp ให้สำหรับเขียน Code ให้ผู้ใช้ทำการเขียน Code ดังนี้

  Button2->Caption= “Matrix”;

- กลับไปที่ Form1 ดับเบิ้ลคลิกที่ **Button3** โปรแกรมจะสลับหน้าต่างจาก Form1 เป็นUnit1.cpp ให้สำหรับเขียน Code ให้ผู้ใช้ทำการเขียน Code ดังนี้

  Button3->Caption= “Feature”

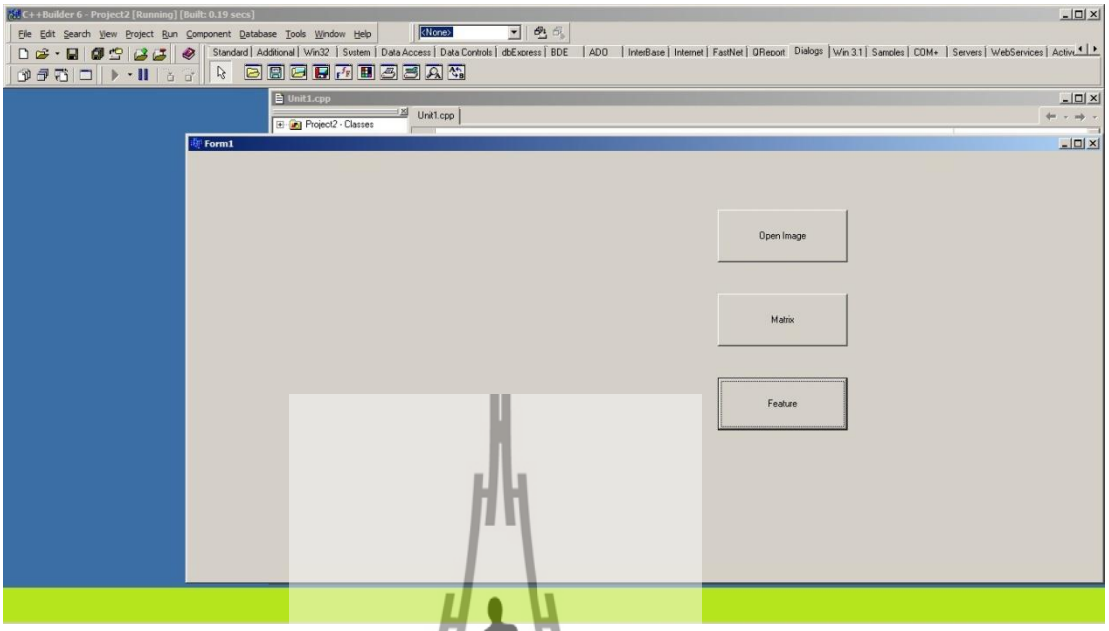
โดย Code ที่เขียนได้แสดงดังรูปก-5



รูป ก-5: แสดง code ที่ถูกเขียนขึ้นจากขั้นตอนที่ 3 สำหรับเปลี่ยนชื่อ Button

เมื่อดำเนินการตามขั้นตอนที่ 3 เรียบร้อยแล้วให้ทำการบันทึกงานที่สร้างไว้โดยไปที่ Menu bar เลือกไปที่ File จากนั้นเลือก Save All เพื่อบันทึกทุกอย่างที่ได้ทำลงบนโปรแกรม โดยโปรแกรมจะขึ้นหน้าต่างถามการบันทึกสองครั้งด้วยกัน ครั้งแรกจะเป็นการบันทึกในหน้าต่างของ Unit1.cpp และครั้งที่สองจะเป็นการบันทึกในหน้าต่างของ Project1 เมื่อบันทึกเสร็จเรียบร้อยแล้ว ต่อไปจะรันโปรแกรมด้วยสัญลักษณ์ และหยุดการรันโปรแกรมด้วยสัญลักษณ์ ผู้ใช้จะสังเกตเห็นว่าหลังจากรันโปรแกรมแล้วจะมีหน้าต่างของ Form1 ปรากฏขึ้นมา

โดยอัตโนมัติ และเมื่อคลิกที่ Button1, Button2 และ Button3 ตามลำดับ ผลลัพธ์ที่ได้แสดงใน รูปก-6



รูป ก-6: แสดงผลการรันโปรแกรมเพื่อเปลี่ยนชื่อของ Button

หมายเหตุ :อีกหนึ่งวิธีที่สามารถเปลี่ยนชื่อของ Button ได้โดยไม่ต้องกดรันโปรแกรมคือที่แถบเครื่องมือที่ชื่อว่า Object Inspector ให้เลือก Button1>>Properties>>Caption ในช่องว่างด้านขวาของ Captionให้พิมพ์ว่า Open Image สำหรับ Button1, Button2>>Properties>>Caption พิมพ์ว่า Matrix สำหรับ Button2และ Button3>>Properties>>Caption พิมพ์ว่า Feature สำหรับ Button3

มหาวิทยาลัยเทคโนโลยีสุรนารี

ขั้นตอนที่ 4: เขียน Code ให้กับ Button1 หรือOpen Image เพื่อให้โปรแกรมสามารถเปิดภาพได้เมื่อผู้ใช้คลิกปุ่มนี้ ดำเนินการเขียน Code ดังต่อไปนี้

- ให้ผู้ใช้เข้าไปที่ **Unit1.h** เนื่องจากจะมีการเปิดภาพที่มีนามสกุล **.jpeg**จึงต้องประกาศ header เพื่อแจ้งให้โปรแกรมทราบล่วงหน้าว่าจะมีการเปิดภาพนามสกุล .jpeg เพราะฉะนั้น Code ที่ต้องเพิ่มเข้าไปในส่วนนี้คือ

```
#include <jpeg.hpp>
```

ที่ **Unit1.h** ต่อไปจะใช้พื้นที่สำหรับประกาศตัวแปรที่จะใช้สำหรับเขียน Code ใน Unit1.cpp ซึ่งตัวแปรที่ต้องประกาศมีดังนี้

```
public:                 // User declarations
        __fastcall TForm1(TComponent* Owner);
        Graphics::TBitmap *bitmap1;
        TRect MyRect;
        int xsize, ysize;
        int *intenR, *intenG, *intenB;
        int **intenGray;
```

- จากนั้นให้เข้าไปที่ **Unit1.cpp** เพื่อไปใช้พื้นที่สำหรับกำหนดค่าให้กับตัวแปรที่ได้ประกาศไว้ใน Unit1.h โดยจะทำการกำหนดค่าตัวแปรภายใต้ฟังก์ชัน

```
void __fastcall TForm1::FormCreate(TObject *Sender)
```

ในส่วนนี้ห้ามผู้ใช้พิมพ์ Code ใดๆ เองเพราะโปรแกรมอาจจะไม่รู้จัก ซึ่งผู้ใช้สามารถเรียก Code นี้ได้จากแถบเครื่องมือชื่อ Object Inspectorเลือก Form1>>Event>>OnCreat ที่ช่องว่างด้านขวาของ OnCreate ให้ดับเบิ้ลคลิกที่ช่องว่างนั้น จะเห็นคำว่า FromCreate หลังจากนั้นโปรแกรมก็จะสร้าง Code ดังกล่าวให้โดยอัตโนมัติ ส่วน Code ที่เหลือผู้ใช้จะต้องดำเนินการพิมพ์เองดังนี้

```
void __fastcall TForm1::FormCreate(TObject *Sender)

{

 bitmap1 = new Graphics::TBitmap;

 MyRect = Rect(0, 0, Image1->Width, Image1->Height);

}
```

- ที่ **Unit1.cpp** ต่อไปจะดำเนินการใช้พื้นที่สำหรับเขียน Code เพื่อเปิดภาพที่มีนามสกุล **.jpeg**ให้ผู้ใช้กลับไปที่ Form1 แล้วดับเบิ้ลคลิกที่ Button1 หรือ Open Image โปรแกรมจะสร้างโค้ด

```
void __fastcall TForm1::Button1Click(TObject *Sender)
```

ให้โดยอัตโนมัติ จากนั้นก็สามารถดำเนินการเขียน Code สำหรับเปิดภาพนามสกุล .jpeg ได้ตาม Code ด้านล่างนี้

```
    intenGray = new int *[xsize];
    for(int i=0;i<xsize;i++)
    {
        intenGray[i] = new int [ysize];
    }
    Byte *ptr;
    for(int j=0; j< ysize; j++)
    {
        ptr  = (Byte *)(bitmap1->ScanLine[j]);
        for(int i=0; i< xsize; i++)
        {
            intenGray[i][j] = (ptr[3*i]+ptr[3*i+1]+ptr[3*i+2])/3;
        }
    }
    Image1->Canvas->StretchDraw(MyRect, bitmap1);
}
}
catch(...)
{
    ShowMessage("error reading image file");
}

   if(intenR != NULL)
    {
        delete[] intenR;
        delete[] intenG;
        delete[] intenB;
    }
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

ขั้นตอนที่ 5: เขียน Code ให้กับ Button2 หรือ Matrix เพื่อให้โปรแกรมสร้างเมทริกซ์ขนาด 256 x 256 สำหรับเก็บค่าสถิติที่เกิดจากการเปรียบเทียบค่าสี RGB ระหว่างพิกเซลใกล้เคียงจากซ้ายไปขวา ดำเนินการเขียน Codeด้วยขั้นตอนต่อไปนี้

- ให้ผู้ใช้เข้าไปที่ **Unit1.h** เพื่อไปใช้พื้นที่สำหรับประกาศตัวแปรที่จะใช้สำหรับเขียน Code ให้กับ Button2 ซึ่งจะทำการประกาศตัวแปรดังต่อไปนี้ *(ใช้พื้นที่ในการประกาศตัวแปรต่อจากขั้นตอนที่ 4)*

```
public:                 // User declarations
        __fastcall TForm1(TComponent* Owner);
        Graphics::TBitmap *bitmap1;
        TRect MyRect;
        int xsize, ysize;
        int *intenR, *intenG, *intenB;
        int **intenGray;
        float **Co_metrix;
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

- จากนั้นไปที่ **Unit1.cpp** เพื่อดำเนินการใช้พื้นที่สำหรับเขียน Codeเพื่อให้โปรแกรมสร้างเมทริกซ์ ขนาด 256 x 256 สำหรับเก็บค่าสถิติที่เกิดจากการเปรียบเทียบค่าสี RGB ระหว่างพิกเซลใกล้เคียงจากซ้ายไปขวา ให้ผู้ใช้กลับไปที่ Form1 แล้วดับเบิ้ลคลิกที่ Button2 หรือ Matrix โปรแกรมจะสร้างโค้ด

```
void __fastcall TForm1::Button2Click(TObject *Sender)
```

ให้โดยอัตโนมัติ จากนั้นก็เขียน Code ด้านล่างนี้เพื่อดำเนินการดังกล่าว

```
void __fastcall TForm1::Button2Click(TObject *Sender)
{
        Co_metrix = new float *[256];
        for(int i=0;i<256;i++)
        {
                Co_metrix[i] = new float [256];
        }
        for(int d=0;d<256;d++)
        {
                for(int c=0;c<256;c++)
                {
                        Co_metrix[c][d]=0;
                }
        }
for(int d=0;d<ysize;d++)
{
        for(int c=0;c<xsize-1;c++)
        {
                Co_metrix[intenGray[c][d]][intenGray[c+1][d]] ++;
        }
}
}
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

ขั้นตอนที่ 6: เขียน Code ต่อจากบรรทัดสุดท้ายของขั้นตอนที่ 5เพื่อให้โปรแกรมทำการคำนวณค่า **Homogeneity, Contrast, Dissimilarity, Mean, Standard Deviation, Entropy, Angular Second Moment** และ Correlationโดยทั้ง 8 ค่านี้จะเรียกรวมๆว่า "**All Feature**"สำหรับนำไปวิเคราะห์เนื้อภาพ โดยรายละเอียดของค่าต่างๆนั้นได้อธิบายไว้ในส่วนของเนื้อหาแล้ว

- ให้ผู้ใช้เข้าไปที่ **Unit1.h** เพื่อไปใช้พื้นที่สำหรับประกาศตัวแปรที่จะใช้สำหรับเขียน Code นี้ ซึ่งจะทำการประกาศตัวแปรดังต่อไปนี้*(ใช้พื้นที่ในการประกาศตัวแปรต่อจากขั้นตอนที่ 4และ 5)*

```
public:                 // User declarations
      __fastcall TForm1 (TComponent* Owner);
      Graphics::TBitmap *bitmap1;
       TRect MyRect;
       int xsize, ysize;
       int *intenR, *intenG, *intenB;
       int **intenGray;
       float **Co_metrix;
       float I1,I2,I3,I4,I5,I6,I7,I8;
void AllFeature();
       float Homogeneity();
       float Contrast();
       float Dissimilarity();
       float Mean();
       float StandardDeviation(float I4);
       float Entropy();
       float AngularSecondMoment();
       float Correlation(float I4,float I5);
          float
    sum,total,sum2,total2,sum3,total3,sum4,total4,sum5,total5,sum6,total6,sum7,total7,sum8,total8;
       int Change,Absolute,Result;
       float M_mean;
       float Standard;
       float M_meanJ;
       float StandardJ;
        float summ4,totall4,summ5,totall5;
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

- ในการคำนวณ**All Feature** นั้น จะมีการเรียกใช้ฟังก์ชันทางคณิตศาสตร์ ดังนั้นที่ **Unit1.h** จึงต้องประกาศ header เพื่อแจ้งให้โปรแกรมทราบล่วงหน้าว่าจะมีการเรียกใช้ฟังก์ชันทางคณิตศาสตร์ เพราะฉะนั้น Code ที่ต้องเพิ่มเข้าไปในส่วนนี้คือ

```
#include <math.h>
```

- จากนั้นไปที่ **Unit1.cpp** เพื่อดำเนินการใช้พื้นที่สำหรับเขียน Codeเพื่อให้โปรแกรมคำนวณค่าทั้ง 8 ค่านั้น หรือที่เรียกว่า **"All Feature"** แสดง Code ดังกล่าว ดังต่อไปนี้

```
void TForm1::AllFeature()
{
      I1=Homogeneity();
      I2=Contrast();
      I3=Dissimilarity();
      I4= Mean();
      I5=StandardDeviation(I4);
      I6=Entropy();
      I7=AngularSecondMoment();
      I8=Correlation(I4,I5);

}
//.....
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
float TForm1:: Homogeneity()

{

float Ho_mogen;

sum=0;

total=0;

Ho_mogen=0;


        for(int j=0;j<256;j++)

        {

            for(int i=0;i<256;i++)

            {

                    sum=(Co_metrix[i][j])/(1+((i-j)*(i-j)));

                    total=total+sum;


             }


        }

Ho_mogen=total;

return  Ho_mogen;

}


//.....................................................................................
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
float TForm1:: Dissimilarity()
{
float Dis_sim;
sum3=0;
total3=0;
Change=0;
Absolute=0;
Dis_sim=0;

        for(int j=0;j<256;j++)
        {
            for(int i=0;i<256;i++)
            {
                    Change=i-j;
                    Absolute=abs(Change);
                    sum3=(Co_metrix[i][j])*(Absolute);
                    total3=total3+sum3;
             }
        }
Dis_sim=total3;
return   Dis_sim;
}
//.....
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
 float TForm1:: Mean()
float TForm1::StandardDeviation(float I4)
{
sum5=0;
total5=0;
Standard=0;

            for(int j=0;j<256;j++)
            {
                  for(int i=0;i<256;i++)
                  {
                              sum5=(Co_metrix[i][j])*((i-I4)*(i-I4));
                              total5=total5+sum5;
                  }
            }
 Standard=total5;
return Standard;
}

//.....
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
float TForm1:: Entropy()
{
float En_tropy;
total6=0;
sum6=0;
En_tropy=0;
float X,ResultLOG;
            for(int j=0;j<256;j++)
            {
                  for(int i=0;i<256;i++)
                  {
                              X =Co_metrix[i][j];
                              if(X!=0)
                                 {
ResultLOG = log(X);
                    sum6=(Co_metrix[i][j])*(-1)*(ResultLOG);
            }
                              else
                              sum6=0;
                              total6=total6+sum6;
                  }
            }
 En_tropy=total6;
return  En_tropy;
}
//.....
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

```
float TForm1::Correlation(float I4,float I5)
{
float Cor_relation;
sum8=0;
total8=0;

summ4=0;
totall4=0;
M_meanJ=0;

summ5=0;
totall5=0;
StandardJ=0;

for(int j=0;j<256;j++)
{
      for(int i=0;i<256;i++)
      {
            summ4=(Co_metrix[i][j])*j;

            totall4=totall4+summ4;

            M_meanJ=totall4;

      }
}


for(int j=0;j<256;j++)
      {
            for(int i=0;i<256;i++)
            {
                  summ5=(Co_metrix[i][j])*((j-M_meanJ)*(j-M_meanJ));

                  totall5=totall5+summ5;

                  StandardJ=totall5;
            }

      }

for(int j=0;j<256;j++)
{
      for(int i=0;i<256;i++)
      {
            sum8=(Co_metrix[i][j])*(((i-I4)*(j- M_meanJ))/sqrt((I5*I5)+(StandardJ*StandardJ)));

          total8=total8+sum8;

      }
}

Cor_relation=total8;
return   Cor_relation;

}
//.....
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

ขั้นตอนที่ 7: เขียน Code ให้กับ Button3 หรือ Feature เป็นขั้นตอนสุดท้ายสำหรับการเขียน Code เพื่อการวิเคราะห์เนื้อภาพต่อจากนี้จะเป็น Code สำหรับเรียกดูค่าทั้ง 8 ค่า ใน All Feature อันได้แก่ Homogeneity, Contrast, Dissimilarity, Mean, Standard Deviation, Entropy, Angular Second Moment และ Correlation ดำเนินการดังต่อไปนี้

- ในการเรียกดูค่าทั้ง 8 ค่า ใน**All Feature** นั้น จะมีการเรียกใช้ฟังก์ชันที่เกี่ยวข้องต่างๆ ดังนั้นที่ **Unit1.h** จึงต้องประกาศ header เพื่อแจ้งให้โปรแกรมทราบล่วงหน้า Code ที่ต้องเพิ่มเข้าไปในส่วนนี้คือ

```
#include <iostream.h>
#include <fstream>
#include <stdio.h>
using namespace std;
```

- ให้ผู้ใช้เข้าไปที่ **Unit1.cpp** เพื่อไปใช้พื้นที่สำหรับเขียน Code ในการเรียกดูค่าทั้ง 8 ค่า ใน**All Feature** โดยไปที่ Form1 แล้วดับเบิ้ลคลิกที่ Button3 หรือ Feature โปรแกรมจะสร้างโค้ด

```
void __fastcall TForm1::Button3Click(TObject *Sender)
```

ให้โดยอัตโนมัติ จากนั้นก็เขียน Code ด้านล่างนี้เพื่อดำเนินการดังกล่าว

```
void __fastcall TForm1::Button3Click(TObject *Sender)
{
  AllFeature();
  ofstream SUT("AllFeature.txt");
   {
     SUT<<I1<<"   "<<I2<<"   "<<I3<<"   "<<I4<<"   "<<I5<<"   "<<I6<<"   "<<I7<<"   "<<I8;
   }
  SUT.close();
}
```

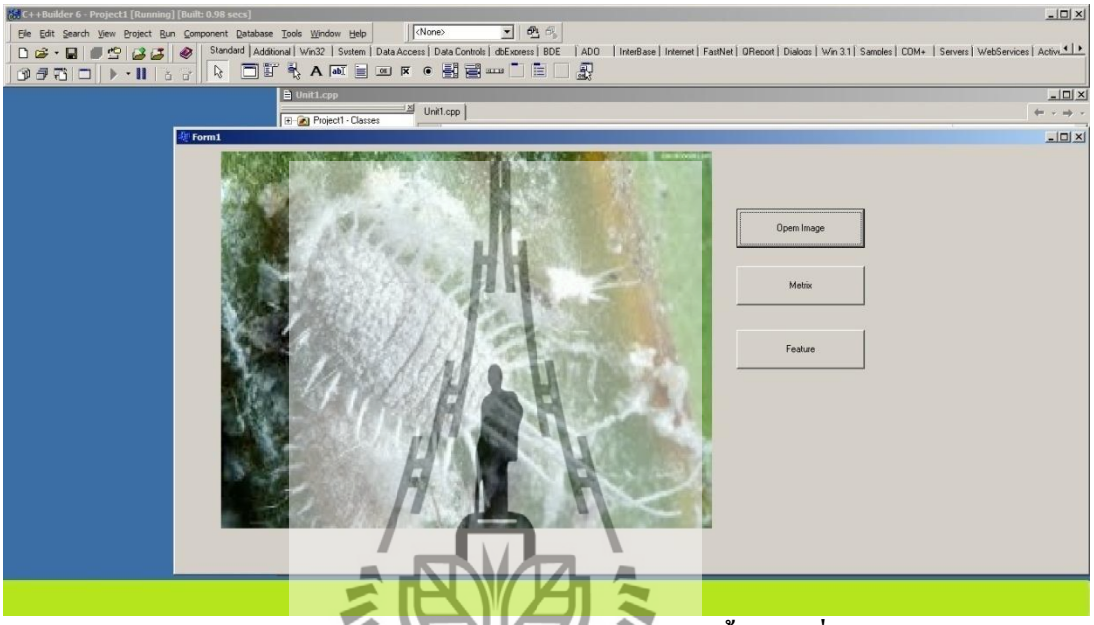
❖

ตัวอย่างการใช้งานโปรแกรม

❖ เมื่อดำเนินการเริ่มตั้งแต่ขั้นตอนที่ 1 จนสิ้นสุดที่ ขั้นตอนที่ 4 ให้บันทึกและรันโปรแกรม หลังจากนั้นจะมีหน้าต่างของ Form1 แสดงที่หน้าจอ

ให้คลิกที่ปุ่ม

เพื่อไปโหลดภาพที่มีนามสกุล **.jpeg**ผลลัพธ์ที่ได้แสดงด้วย



รูป ก-7:แสดงผลการดำเนินการจากขั้นตอนที่ 1-4

มหาวิทยาลัยเทคโนโลยีสุรนารี

❖ ถัดมาคือดำเนินการจากขั้นตอนที่ 5 จนถึงสุดท้ายโดยสิ้นสุดที่ ขั้นตอนที่ 7 ให้ทำการบันทึกและรันโปรแกรมใหม่อีกรอบ โดย

ครั้งที่1: คลิกที่

เพื่อโหลดภาพ

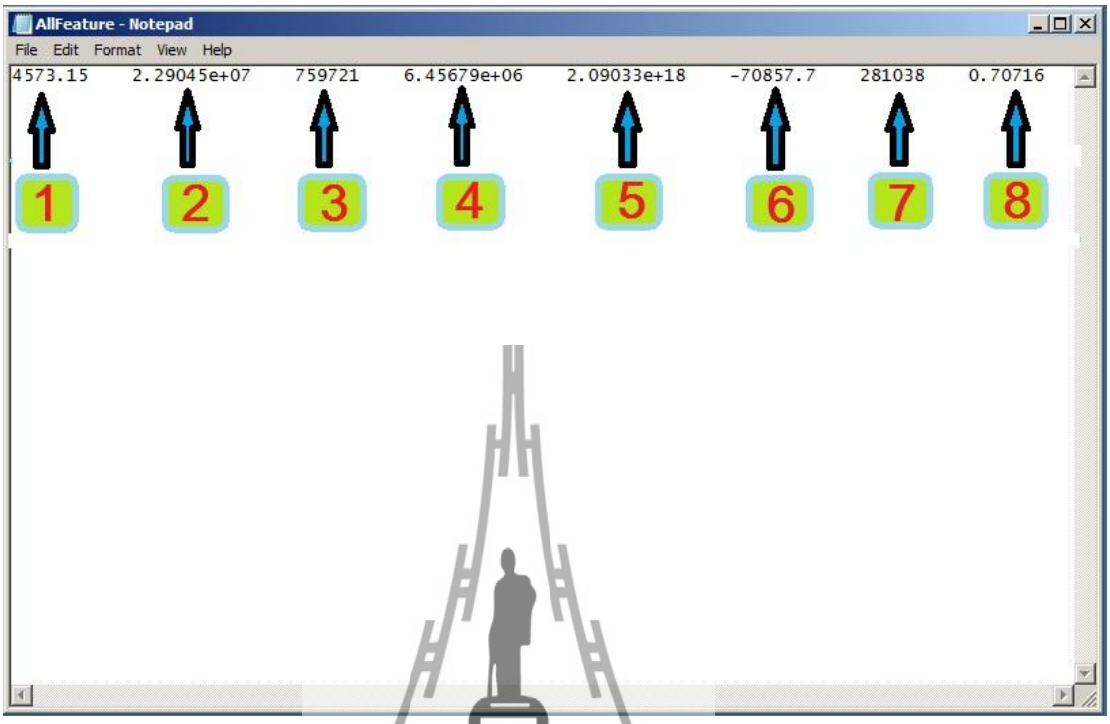
ครั้งที่2: คลิกที่

เพื่อสร้างเมทริกซ์ขนาด 256 x 256

ครั้งที่3: คลิกที่

เพื่อเรียกดูค่าใน All Feature

ผลลัพธ์จากการคลิกทั้ง 3 ครั้ง แสดงด้วย **รูป H** ผลลัพธ์นี้สามารถเข้าไปดูใน ตำแหน่งที่ผู้ใช้ไปเลือกภาพมาขณะคลิกปุ่ม **Open Image**



รูป H:แสดงผลลัพธ์จากการเรียกดูค่าใน All Feature

มหาวิทยาลัยเทคโนโลยีสุรนารี

1 : แสดงถึงค่า**Homogeneity=4573.15**

2 : แสดงถึงค่า**Contrast =2.29045e+07=2.29045 x** e^{7}

3 : แสดงถึงค่า**Dissimilarity=759721**

4 : แสดงถึงค่า**Mean=6.45679e+06=6.45679 x** e^{6}

5 : แสดงถึงค่า**Standard Deviation=2.09033e+18=2.09033 x** e^{18}

6 : แสดงถึงค่า Entropy=**-70857.7**

7 : แสดงถึงค่า**Angular Second Moment=281038**

8 : แสดงถึงค่า**Correlation= 0.70716**

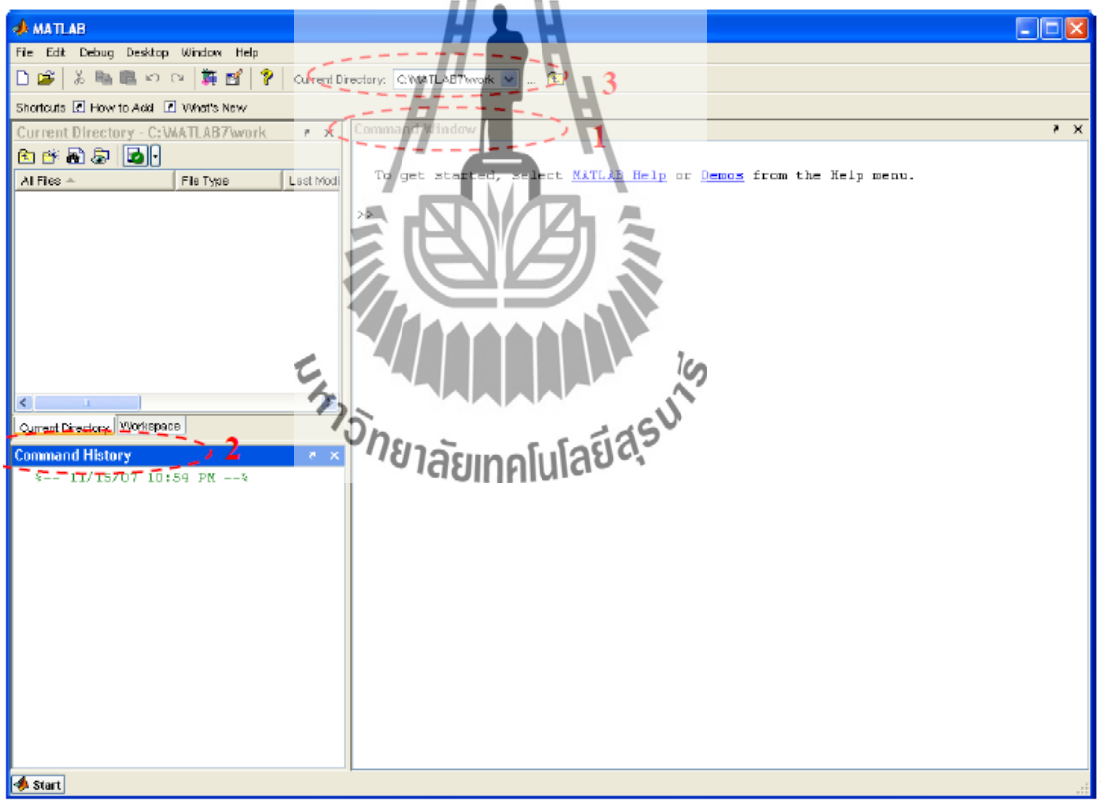
ภาคผนวก ซ

การใช้งานโปรแกรม MATLAB เพื่อประมวลผลภาพ

ด้วยเทคนิคการวิเคราะห์ค่าสี RGB

การจำแนกชนิดของเพลี้ยแป้งมันสำปะหลังด้วยเทคนิคการวิเคราะห์ค่าสี RGB ใช้โปรแกรม MATLAB ในหมวดของ Image Processing Toolbox เป็นตัวช่วยในการจำแนกชนิดของเพลี้ยแป้ง เป็นวิธีที่เกี่ยวกับการวิเคราะห์ค่าสีของภาพ สามารถวิเคราะห์ เพื่อจำแนกชนิดเพลี้ยแป้งมันสำปะหลังด้วยขั้นตอนในโปรแกรมดังนี้

ขั้นตอนที่ 1:เปิดโปรแกรม**MATLAB**ขึ้นมา ดังแสดงใน**รูปข-1**



รูป ข-1 แสดงหน้าต่างของโปรแกรม MATLAB เมื่อเริ่มต้นใช้งาน

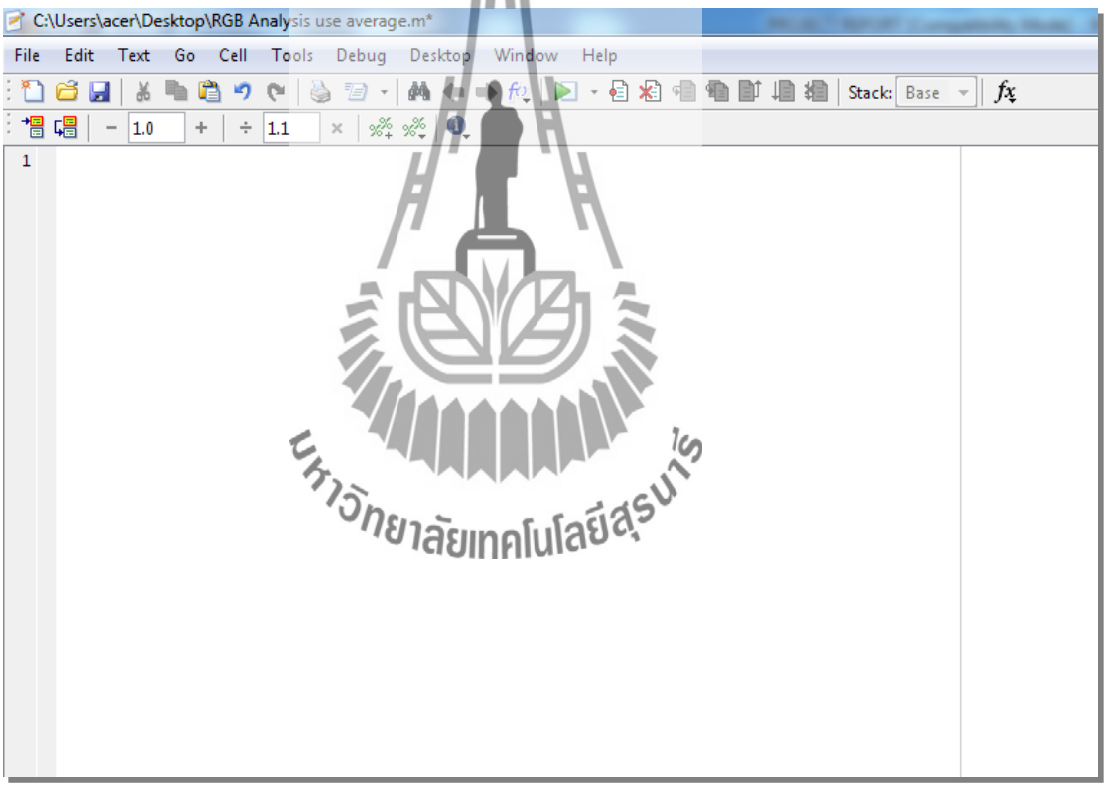
1: Command Window สำหรับเขียนโค้ดโปรแกรม และแสดงผลลัพธ์การรันโปรแกรม เมื่อกด Enter

มหาวิทยาลัยเทคโนโลยีสุรนารี

2: Command History สำหรับแสดงชุดคำสั่งที่ถูกใช้ไปแล้วใน Command Window และสามารถเรียกใช้ซ้ำอีกได้

3: Current Directory สำหรับบอกว่า MATLAB กำลังติดต่อกับแฟ้มใดในคอมพิวเตอร์ โดย MATLAB จะมองเห็นข้อมูลที่เก็บอยู่ที่ Directory ที่กำหนดเท่านั้น ถ้าต้องการให้ MATLAB มองเห็นแฟ้มอื่น ก็ให้เปลี่ยน Current Directory เป็นแฟ้มนั้นๆด้วยเช่นกัน

ขั้นตอนที่ 2: สร้าง M-file สำหรับเขียนโค้ดโปรแกรม ข้อดีของการเขียนโค้ดโปรแกรมใน M-file เพราะสามารบันทึกเพื่อเรียกใช้งานในครั้งต่อไปได้ ซึ่งถ้าเขียนโค้ดโปรแกรมใน Command Window จะไม่สามารถเก็บโค้ดโปรแกรมไว้ใช้งานในครั้งต่อไปได้เราจะสร้าง M-file ใน Notepad แล้วบันทึกเป็นนามสกุล **.m**ก็จะได้M-file ดัง**รูป ข-2**



รูป ข-2 แสดงหน้าต่าง M-file ที่บันทึก Notepad ด้วยนามสกุล .m

ขั้นตอนที่ 3: เขียนโค้ดโปรแกรมเพื่อจำแนกชนิดเพลี้ยแป้งมันสำปะหลังลงใน M-file ที่สร้างขึ้น รายละเอียดของโค้ดโปรแกรมแสดงดังต่อไปนี้

มหาวิทยาลัยเทคโนโลยีสุรนารี

- โค้ดโปรแกรมสำหรับอ่านภาพอินพุตเข้ามาเก็บไว้ในโปรแกรมแสดงด้วยโค้ดต่อไปนี้

```
UnknowPicture=imread('QQQQQQQQQQQQQQQ.jpg');
figure(5),imshow(UnknowPicture),title('Input Picture');
```

- โค้ดโปรแกรมสำหรับให้ผู้ใช้เลือกบริเวณที่สนใจของภาพอินพุตเพื่อนำมาวิเคราะห์ว่าเป็นเพลี้ยแป้งมันสำปะหลังชนิดใดแสดงด้วยโค้ดต่อไปนี้

```
UnknowPictureCrop=imcrop(UnknowPicture);
figure(5),imshow(UnknowPictureCrop),title('Input picture by your select');
```

- โค้ดโปรแกรมสำหรับสร้างสมการหาค่าเฉลี่ยสี RGB ของภาพอินพุตและเก็บค่าเฉลี่ยสีที่ได้ไว้ในตัวแปรที่สร้างขึ้นแสดงด้วยโค้ดต่อไปนี้

```
UnknowPictureRGB=UnknowPictureCrop;
UnknowPictureRed=UnknowPictureRGB(:,:,1);
UnknowPictureGreen=UnknowPictureRGB(:,:,2);
UnknowPictureBlue=UnknowPictureRGB(:,:,3);
[RowUnknow ColumnUnknow] = size(UnknowPictureCrop);
AverageUnknowPictureRed=sum(sum(UnknowPictureRed))/(RowUnknow*ColumnUnknow)
AverageUnknowPictureGreen=sum(sum(UnknowPictureGreen))/(RowUnknow*ColumnUnknow)
AverageUnknowPictureBlue=sum(sum(UnknowPictureBlue))/(RowUnknow*ColumnUnknow)
```

มหาวิทยาลัยเทคโนโลยีสุรนารี

➢ โค้ดโปรแกรมสำหรับสร้างสมการทำให้ภาพอินพุตไม่ขึ้นกับความเข้มแสงแสดงด้วยโค้ดต่อไปนี้

```
REDUnknow=AverageUnknowPictureRed/(AverageUnknowPictureRed+…
                AverageUnknowPictureGreen+AverageUnknowPictureBlue)
GREENUnknow=AverageUnknowPictureGreen/(AverageUnknowPictureRed+…
                AverageUnknowPictureGreen+AverageUnknowPictureBlue)
BLUEUnknow=AverageUnknowPictureBlue/(AverageUnknowPictureRed+…
                AverageUnknowPictureGreen+AverageUnknowPictureBlue)
```

➢ โค้ดโปรแกรมสำหรับสร้างสมการหาระยะห่างของจุดสี RGB ของภาพอินพุตเทียบกับฐานข้อมูลรูปที่ 3.3แสดงระยะห่างระหว่างจุดสี RGBของภาพอินพุตเทียบกับฐานข้อมูลของเพลี้ยแป้งมันสำปะหลังทั้ง 4ชนิดแสดงด้วยโค้ดต่อไปนี้

```
UnGray=sqrt((REDUnknow-0.3335)^2+(GREENUnknow-0.3338)^2+(BLUEUnknow-0.3327)^2)
UnGreen=sqrt((REDUnknow-0.3344)^2+(GREENUnknow-0.3340)^2+(BLUEUnknow-0.3317)^2)
UnPink=sqrt((REDUnknow-0.3359)^2+(GREENUnknow-0.3330)^2+(BLUEUnknow-0.3311)^2)
UnLay=sqrt((REDUnknow-0.3339)^2+(GREENUnknow-0.3335)^2+(BLUEUnknow-0.3326)^2)
```

➢ โค้ดโปรแกรมสำหรับแสดงผลถ้าค่าจุดสีของภาพอินพุตใกล้เคียงค่าจุดสีของฐานข้อมูลเพลี้ยแป้งมันสำปะหลังใน 4 ชนิดๆใดชนิดหนึ่งมากกว่า แสดงว่าภาพอินพุตเป็นเพลี้ยแป้งมันสำปะหลังชนิดเดียวกันกับฐานข้อมูลที่ใช้เปรียบเทียบแสดงด้วยโค้ดต่อไปนี้

```
if UnGray<UnGreen&UnGray<UnPink&UnGray<UnLay
    figure(7),imshow(UnknowPicture),title('This picture is "Gray Mealy Bug"');
else
if UnGreen<UnGray&UnGreen<UnPink&UnGreen<UnLay
      figure(7),imshow(UnknowPicture),title('This picture is "Green Mealy Bug"');
elseif UnPink<UnGray&UnPink<UnGreen&UnPink<UnLay
        figure(7),imshow(UnknowPicture),title('This picture is "Pink Mealy Bug"');
else
        figure(7),imshow(UnknowPicture),title('This picture is "Lay Mealy Bug"');
end
end
```

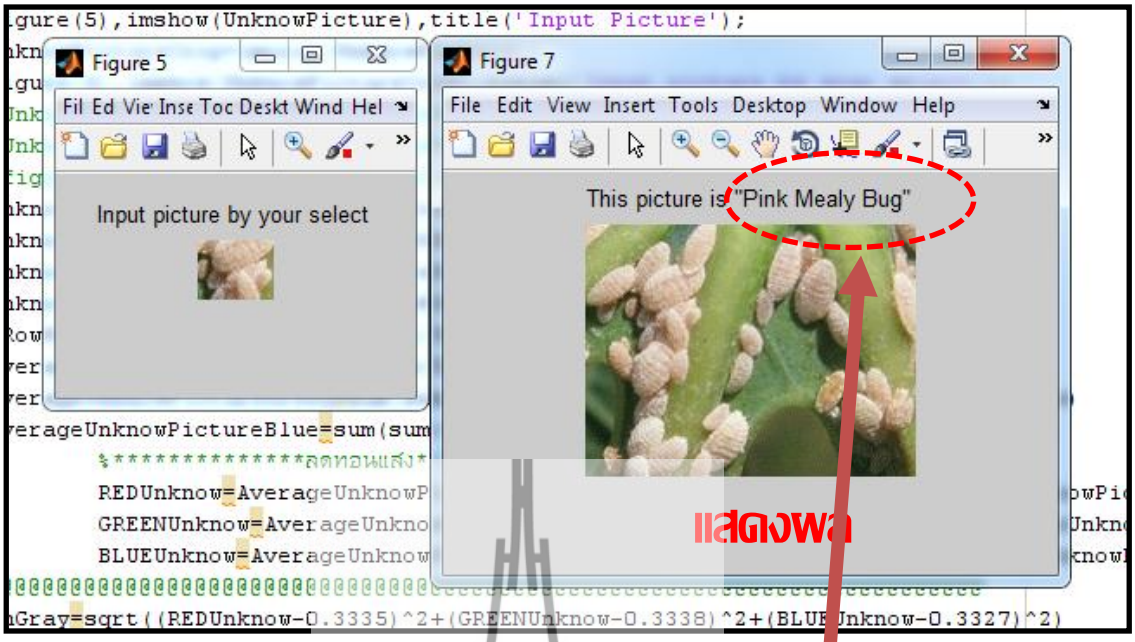
ตัวอย่างการใช้งานโปรแกรม

❖ ภาพอินพุตคือ

❖ ผู้ใช้เลือกบริเวณที่สนใจของภาพอินพุตได้ดังนี้

มหาวิทยาลัยเทคโนโลยีสุรนารี

❖ โปรแกรมแสดงผลลัพธ์การจำแนกชนิดเพลี้ยแป้งมันสำปะหลังด้วยภาพดังนี้



ภาพอินพุตนี้คือ Pink Mealy Bug หรือ เพลี้ยแป้งสีชมพู

มหาวิทยาลัยเทคโนโลยีสุรนารี

บรรณานุกรม

[1] จงรักษ์ จารุเนตร , **คลินิกเกษตรเคลื่อนที่:แนวทางการวินิจฉันโรคพืช** , เอกสารวิชาการศูนย์บริการวิชาการด้านพืชและปัจจัยการผลิตปราจีนบุรี , สำนักวิจัยและพัฒนาการเกษตรเขตที่ 6 กรมวิชาการเกษตร กระทรวงเกษตรและสหกรณ์.

[2] ชรัตน์ มงคลสวัสดิ์ และคนอื่นๆ , **การวิเคราะห์เนื้อภาพเพื่อจำแนกช่วงอายุของยางพาราด้วยภาพถ่ายจากดาวเทียม THEOS ลุ่มน้ำห้วยครอง** , ศูนย์ภูมิภาคเทคโนโลยีอวกาศและภูมิสารสนเทศ ภาคตะวันออกเฉียงเหนือ มหาวิทยาลัยขอนแก่น , ค้นเมื่อ 5 พฤษภาคม 2555 จาก URL:http:// negistda.kku.ac.th/ research26/ paper _re26.pdf

[3] ลิขสิทธิ์มหาวิทยาลัยเชียงใหม่,**บทที่ 2 ทฤษฎีที่เกี่ยวข้อง**, ค้นเมื่อ 5 พฤษภาคม 2555 จาก URL: http:// archive.lib.cmu.ac.th/full/T/2551/enco0951sp_ch2.pdf

[4] H. Al-Hiary, S. Bani-Ahmad, M. Reyalat, M.Braik and Z. ALRahamneh, **Fast and Accurate Detection and Classification of Plant Diseases**, Department of Information Technology, Al-Balqa'Applied University, Salt Campus, Jordan, (2011).

[5] **'เพลี้ยแป้ง 'ระบาดไม่หยุด! โคราชสูญ 3.5 แสนไร่กว่า 400 ล้าน** แหล่งที่มา http://www.manager.co.th/mgrWeekly/ViewNews.aspx?NewsID=9530000016766ค้นเมื่อ 18 พฤษภาคม 2555

[6] **เพลี้ยแป้ง...มหันตภัยต่อมันสำปะหลัง** แหล่งที่มา http://www.thaitapiocastarch.org/article20_th.asp ค้นเมื่อ 10มกราคม2556

[7] **รูปภาพเพลี้ยแป้งมันสำปะหลัง** แหล่งที่มา https://www.google.co.th/search?q=Striped+Mealybug&hl=th&tbm=isch&tbo=u&source=univ&sa=X&ei=YNZNUcLgEYbUrQel0IDoAQ&ved=0CDkQsAQ&biw=1366&bih=639ค้นเมื่อ 10 มกราคม2556

มหาวิทยาลัยเทคโนโลยีสุรนารี

[8] **รูปภาพเพลี้ยแป้งมันสำปะหลัง** แหล่งที่มา https://www.google.co.th/search?q=%E0%B9%80%E0%B8%9E%E0%B8%A5%E0%B8%B5%E0%B9%89%E0%B8%A2%E0%B9%81%E0%B8%9B%E0%B9%89%E0%B8%87%E0%B8%A1%E0%B8%B1%E0%B8%99%E0%B8%AA%E0%B9%8D%E0%B8%B2%E0%B8%9B%E0%B8%B0%E0%B8%AB%E0%B8%A5%E0%B8%B1%E0%B8%87&hl=th&tbm=isch&tbo=u&source=univ&sa=X&ei=qD83UcW8KYO3rAeKtoHgBA&ved=0CFkQsAQ&biw=1366&bih=639#imgrc=BrOK3_vDf_wJUM%3A%3BX5BUtB6LAscWaM%3Bhttp%253A%252F%252Fkhaosuankwang.khonkaen.doae.go.th%252Fimages%252FAgFair.jpg%3Bhttp%253A%252F%252Fkhaosuankwang.khonkaen.doae.go.th%252F%3B370%3B128ค้นเมื่อ 10มกราคม2556

[9] **รูปภาพเพลี้ยแป้งมันสำปะหลัง** แหล่งที่มา https://www.google.co.th/search?q=mealy+bug&hl=th&tbm=isch&tbo=u&source=univ&sa=X&ei=O2M3UZjPCI6zrAf844DgDw&sqi=2&ved=0CEYQsAQ&biw=1366&bih=639#imgrc=_ค้นเมื่อ 10มกราคม2556

[10] **รูปภาพเพลี้ยแป้งมันสำปะหลัง** แหล่งที่มา https://www.google.co.th/search?q=mealy+bug&hl=th&tbm=isch&tbo=u&source=univ&sa=X&ei=O2M3UZjPCI6zrAf844DgDw&sqi=2&ved=0CEYQsAQ&biw=1366&bih=639#imgrc=rPnDF2OSLx4iYM%3A%3Bx24FTDHxS0G0kM%3Bhttp%253A%252F%252Fwww.tindaraorchids.com%252Fimages%252Fsupplies%252Fmealybugs.jpg%3Bhttp%253A%252F%252Fwww.tindaraorchids.com%252Finsect-control.htm%3B250%3B174ค้นเมื่อ 10มกราคม2556

[11] **รูปภาพเพลี้ยแป้งมันสำปะหลัง** แหล่งที่มา http://www.google.co.th/imgres?imgurl=http://www.thaitapiocastarch.org/images/article/20/001.jpg&imgrefurl=http://www.thaitapiocastarch.org/article20_th.asp&h=345&w=409&sz=48&tbnid=_hbINubmhjFX9M:&tbnh=78&tbnw=92&zoom=1&usg=__KlmWSnVJPzW-wiNNtrh9UBGs2mQ=&docid=fs3Ffpor0K3wNM&hl=th&sa=X&ei=rUdLUfS9AcXYrQfTgoGACQ&sqi=2&ved=0CDYQ9QEwAQ&dur=2456ค้นเมื่อ 10มกราคม2556

มหาวิทยาลัยเทคโนโลยีสุรนารี

[12] **CIE 1931 color space**แหล่งที่มา http://en.wikipedia.org/wiki/CIE_1931_color_spaceค้นเมื่อ 2มีนาคม2556

[13]**C++ Builder**แหล่งที่มา http://www.google.co.th/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0CDQQFjAA&url=http%3A%2F%2Feisc.univalle.edu.co%2Fmaterias%2FProcesamiento_De_Imagenes_Digitales%2FLibro%2520de%2520Ayuda%2FIntroduction%2520to%2520Image%2520Processing%2520and%2520Computer%2520Vision.pdf&ei=0NZWUZ_KL4fmrAfloYGwAw&usg=AFQjCNG9fiDws0UizF9WN4-mR1Cxq0oQwQ&sig2=oSMzJi5nAiO3b2bg937tjQค้นเมื่อ 10มกราคม 2556

[14]**How To Install Borland C++ Builder "Windows 7"..**แหล่งที่มา http://www.youtube.com/watch?v=QX-tLoS99p8ค้นเมื่อ 2มีนาคม2556

[15]**Color theory**แหล่งที่มา http://en.wikipedia.org/wiki/Color_theoryค้นเมื่อ 18กันยายน 2555

[16]ดร.จาตุรงค์ตันติบัณฑิตคู่มือการใช้งาน **MATLAB เบื้องต้น** แหล่งที่มา http://www.google.co.th/url?sa=t&rct=j&q=&esrc=s&source=web&cd=3&sqi=2&ved=0CD4QFjAC&url=http%3A%2F%2Fsuanpalm3.kmutnb.ac.th%2Fteacher%2FFileDL%2Fsupot56255519003.pdf&ei=5SpXUczsBs7MrQfzy4CIDg&usg=AFQjCNGp4NJ1wkTUUzDyFUCkx-aaFlxQaQ&sig2=EaPC4ReaTNIrFU_SFDsOOAค้นเมื่อ 28มีนาคม 2556

มหาวิทยาลัยเทคโนโลยีสุรนารี

ประวัติผู้เขียน

นางสาววิยะดา มะลิวัลย์ เกิดเมื่อวันที่ 17 กุมภาพันธ์ พ.ศ. 25 34 ภูมิลำเนาอยู่ที่ ตำบลช่อระกา อำเภอบ้านเหลื่อม จังหวัดนครราชสีมา สำเร็จการศึกษาระดับมัธยมปลายจากโรงเรียนเนินสง่าวิทยา อำเภอเนินสง่า จังหวัดชัยภูมิ เมื่อปี พ.ศ. 2 551 ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

นายเจนวิทย์ พลเยี่ยม เกิดเมื่อวันที่ 1 มีนาคม พ.ศ. 2534ภูมิลำเนาอยู่ที่ตำบลกุดชุม อำเภอกุดชุม จังหวัดยโสธร สำเร็จการศึกษาระดับมัธยมปลายจากโรงเรียนยโสธรพิทยาคม เมื่อปี พ.ศ. 2551 ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

นางสาวประพันธ์ฤดี ปิตาระเต เกิดเมื่อวันที่ 13 มีนาคม พ.ศ. 25 32 ภูมิลำเนาอยู่ที่ ตำบลมวกเหล็ก อำเภอมวกเหล็ก จังหวัดสระบุรี สำเร็จการศึกษาระดับมัธยมปลายจากโรงเรียนมวกเหล็กวิทยา อำเภอมวกเหล็ก จังหวัดสระบุรี เมื่อปี พ.ศ. 25 51 ปัจจุบันเป็นนักศึกษาชั้นปีที่ 4 สาขาวิชาวิศวกรรมโทรคมนาคม สำนักวิชาวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีสุรนารี

มหาวิทยาลัยเทคโนโลยีสุรนารี