

Chulalongkorn University

Chula Digital Collections

Chulalongkorn University Theses and Dissertations (Chula ETD)

2022

การพัฒนาส่วนต่อประสานกับผู้ใช้งานโปรแกรมระเบียบวิธีการนับผลของไวรัส ฮัดโนมิติ และ ปรับปรุงความแม่นยำให้กับระบบ

ศุภณัฐ เปล่งคำ
คณะวิศวกรรมศาสตร์

Follow this and additional works at: <https://digital.car.chula.ac.th/chulaetd>



Part of the [Applied Mechanics Commons](#), and the [Engineering Mechanics Commons](#)

Recommended Citation

เปล่งคำ, ศุภณัฐ, "การพัฒนาส่วนต่อประสานกับผู้ใช้งานโปรแกรมระเบียบวิธีการนับผลของไวรัสฮัดโนมิติ และ ปรับปรุงความแม่นยำให้กับระบบ" (2022). *Chulalongkorn University Theses and Dissertations (Chula ETD)*. 6526.
<https://digital.car.chula.ac.th/chulaetd/6526>

This Thesis is brought to you for free and open access by Chula Digital Collections. It has been accepted for inclusion in Chulalongkorn University Theses and Dissertations (Chula ETD) by an authorized administrator of Chula Digital Collections. For more information, please contact ChulaDC@car.chula.ac.th.

การพัฒนาส่วนต่อประสานกับผู้ใช้งานโปรแกรมระเบียบวิธีการนับพลาคของไวรัสอัตโนมัติ และ
ปรับปรุงความแม่นยำให้กับระบบ



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาระบบกายภาพที่เชื่อมประสานด้วยเครือข่ายไซเบอร์ ภาควิชาวิศวกรรมเครื่องกล
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2565

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Development of Automated viral plaque assay program And Accuracy Improvement



Mr. Suphanut Plengkham

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Cyber-Physical System

Department of Mechanical Engineering

FACULTY OF ENGINEERING

Chulalongkorn University

Academic Year 2022

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การพัฒนาส่วนต่อประสานกับผู้ใช้งานโปรแกรมระเบียบ วิธีการนับพลาคของไวรัสอัตโนมัติ และ ปรับปรุงความ แม่นยำให้กับระบบ
โดย	นายศุภณัฐ เปล่งคำ
สาขาวิชา	ระบบกายภาพที่เชื่อมประสานด้วยเครือข่ายไซเบอร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	รศ. ดร.กฤษฎา พนมเชิง

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่ง
ของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สุพจน์ เตชวรสินสกุล)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รศ. ดร.รัชทิน จันทร์เจริญ)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก
(รศ. ดร.กฤษฎา พนมเชิง)

..... กรรมการ
(รศ.ดร.อลงกรณ์ พิมพ์พิณ)

..... กรรมการภายนอกมหาวิทยาลัย
(ดร.กิตติกุล โกวิทางกูร)

ศุภณัฐ เปล่งคำ : การพัฒนาส่วนต่อประสานกับผู้ใช้งานโปรแกรมระเบียบวิธีการนับพลา
 ลาคของไวรัสอัตโนมัติ และ ปรับปรุงความแม่นยำให้กับระบบ. (Development of
 Automated viral plaque assay program And Accuracy Improvement) อ.ที่
 ปริญญาหลัก : รศ. ดร.กฤษฎา พนมเชิง

กระบวนการนับพลาของไวรัส หรือ Plaque Assays นั้นเป็นกระบวนการเฉพาะทางที่
 ต้องใช้ผู้เชี่ยวชาญทางไวรัสวิทยาในการตรวจวัด และเป็นพื้นฐานของการศึกษาพัฒนาการรักษาโรค
 และวัคซีน ซึ่งจำเป็นงานที่ท้าทายซับซ้อน และใช้เวลา ส่งผลให้เกิดความเหนื่อยล้าต่อผู้เชี่ยวชาญ
 ด้วยวิธีการนับหาปริมาณคลาสเตอร์ของไวรัสทางกายภาพ ดังนั้นเพื่อลดภาระงาน งานวิจัยนี้จึง
 ศึกษาและพัฒนา โปรแกรมการนับพลาอัตโนมัติ โดยได้ทำการสร้าง พัฒนา ปรับปรุง และ เพิ่ม
 ความเสถียรภาพ ให้กับระบบถ่ายภาพ โปรแกรม และ สร้างส่วนต่อประสานโปรแกรมการนับพลา
 ของไวรัสกับผู้ใช้งาน หรือ GUI (Graphic User Interface) สำหรับแสดงผลและใช้งานโปรแกรม
 ซึ่งตัวโปรแกรมการนับพลาของไวรัสได้ทำการพัฒนาอัลกอริทึม ด้วยกระบวนการวิเคราะห์ภาพ หรือ
 Image processing เช่น เทรชโฮลด์ (Threshold) มอร์โฟโลจี (Morphology) และ กระบวนการ
 แมชชีนเลิร์นนิ่งชนิดอันซูปเปอร์ไวส์ (Unsupervised Learning) ชนิด เค-มีน คลัสเตอร์ริง (K-
 mean clustering) ในส่วนของส่วนต่อประสานผู้ใช้งานได้ทำการพัฒนาโดย C# และมากไปกว่า
 นั้นได้ทำการสร้างอุปกรณ์ลดสิ่งรบกวนกับเครื่องถ่ายภาพอีกด้วย โดยผลลัพธ์จากโปรแกรมการ
 นับพลาของไวรัสอัตโนมัตินั้นสามารถ นับพลาของไวรัสในกระบวนการนับพลาของไวรัสได้มี
 ความถูกต้องถึง 94% อีกทั้ง สามารถลดภาระงานผู้เชี่ยวชาญในการตรวจหาปริมาณคลาสเตอร์ของ
 ไวรัสในกระบวนการนับพลาของไวรัสได้

สาขาวิชา ระบบกายภาพที่เชื่อมประสาน ลายมือชื่อนิสิต
 ด้วยเครือข่ายไซเบอร์

ปีการศึกษา 2565 ลายมือชื่อ อ.ที่ปรึกษาหลัก

6372116521 : MAJOR CYBER-PHYSICAL SYSTEM

KEYWORD: Plaque assay, unsupervised learning algorithms, k-mean clustering, image processing, Thresholding, Morphology, C#, graphic user interface

Suphanut Plengkham : Development of Automated viral plaque assay program And Accuracy Improvement. Advisor: Asst. Prof. Gridsada Phanomchoeng, Ph.D.

Plaque assay is a unique method and specialized measurement by virologists that provides a drug discovery until vaccine development. The approach of plaque assays is challenging and complicated and takes time to analyze by physical counting in well-plates, which causes abstract and stagnant to the expert. Hence, to diminish the load of the specialists to physical counting in well plates, this research has developed the software used for automated assays. The plaque assay uses unsupervised learning algorithms named k-mean clustering and image processing methods such as K-mean Clustering Thresholding and Morphology. A graphic user interface based on C# displays virtual plaque images. The system noise-canceling equipment is generated. The average correction of the Plaque assay automatic program is 94%. Moreover, it demonstrates its effectiveness in reducing experts' workload during experiments. The automated plaque assay program holds promise for facilitating and advancing virology research.

Field of Study: Cyber-Physical System

Student's Signature

Academic Year: 2022

Advisor's Signature

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สามารถสำเร็จลงได้อย่างสมบูรณ์ด้วยความกรุณาและความช่วยเหลือจาก รศ.ดร.กฤษฎา พนมเชิง อาจารย์ที่ปรึกษาวิทยานิพนธ์ ซึ่งท่านได้ให้การสนับสนุน ให้คำปรึกษา และ สละเวลาอันมีค่า ให้ข้าพเจ้าระหว่างการทำวิทยานิพนธ์มาโดยตลอด ข้าพเจ้าต้องขอขอบคุณ รศ.ดร.พญ.ศิวัชร บุญทรัพย์ยากร และ สมาชิก หน่วยปฏิบัติการวิจัยไวรัสวิทยาทางการแพทย์ประยุกต์ (Applied Medical Virology Research Unit) ภาควิชาจุลชีววิทยา คณะแพทยศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ทุกๆท่าน สำหรับให้คำปรึกษา ข้อมูลเกี่ยว กระบวนการนับพลาควของไวรัส (Plaque Assays) และ จัดเตรียมเพลทการทดลองสำหรับทำการศึกษาในงานวิจัยนี้ นอกจากนี้ ข้าพเจ้าขอขอบคุณ นายชยกร กู้เกียรติกุล ที่ให้คำปรึกษาและแนะนำในการทำงานของระบบโปรแกรมต่างๆ ซึ่งมีบทบาทสำคัญในการสนับสนุนความสำเร็จของวิทยานิพนธ์นี้ สุดท้ายนี้ข้าพเจ้า ขอขอบคุณครอบครัวของข้าพเจ้าที่ให้การสนับสนุนข้าพเจ้าในการศึกษาต่อในระดับมหาบัณฑิตศึกษา และอยู่เคียงข้างกันตลอดมาจนงานวิทยานิพนธ์นี้ประสบความสำเร็จได้

ศุภณัฐ เป่งขำ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	ค
บทคัดย่อภาษาอังกฤษ.....	ง
กิตติกรรมประกาศ.....	จ
สารบัญ.....	ฉ
สารบัญรูปรตาราง	ญ
สารบัญรูปภาพ.....	ฎ
บทที่1.....	1
บทนำ.....	1
1.1 ที่มาและความสำคัญ.....	1
1.2 วัตถุประสงค์	3
1.3 ขอบเขตของงานวิจัย.....	3
1.4 ประโยชน์ที่คาดว่าจะได้รับ.....	3
1.5 ขั้นตอนการวิจัย	3
บทที่ 2	5
อุปกรณ์และทฤษฎีที่เกี่ยวข้องในการศึกษา.....	5
2.1 ทฤษฎีการประมวลผลภาพ.....	5
2.1.1.1 กระบวนการจัดการ Morphology และ เชื่อมต่อวัตถุขอบเขต (Region Connectivity)	5
2.1.1.2 Morphological Algorithms พื้นฐาน.....	9
2.1.1.2.1 Boundary Extraction	9
2.1.1.2.2 Region Filling	10

2.1.1.2.3 Components Connection	12
2.1.2 Restoration Image Operation.....	13
2.1.2.1 การกำจัดสิ่งรบกวน.....	13
2.1.2.1.1 Arithmetic Mean Filter (Average Filter).....	14
2.1.2.1.2 Median Filter	14
2.1.2.1.2 Geometric Mean Filter	16
2.1.2.2 การปรับภาพให้มีความคมชัด	16
2.1.2.2.1 กระบวนการ Histogram Equalization	16
2.1.2.2.3 Lowpass Filtering.....	17
2.1.2.2.4 Gamma Correction.....	19
2.1.3 Image Segmentation Operation.....	20
2.1.3.1 Edge Detection	20
2.1.3.2 Threshold Segmentation	23
2.1.3.2.2 Binary Threshold	24
2.1.3.3 Hough Transform.....	25
2.1.3.4 Watershed method.....	28
2.2. Machine learning Operation.....	30
2.2.1 K-mean Clustering.....	30
2.2.1.1 Silhouette Method	36
2.3. เครื่องถ่ายภาพรูปแบบพลาสดิจิตอล	37
2.3.1 Hardware.....	37
2.3.1.1 ระบบการถ่ายภาพ	38
2.3.2 การใช้งาน เครื่องถ่ายภาพรูปแบบพลาสดิจิตอล	39
2.4 วรรณกรรมที่เกี่ยวข้องกับการนับด้วยกระบวนการวิเคราะห์ภาพ	41

2.6 เซลล์และไวรัสสำหรับกระบวนการนับพลาของไวรัส	43
บทที่ 3	44
วิธีดำเนินการพัฒนาและออกแบบโปรแกรมระเบียบวิธีการนับพลาของไวรัส	44
3.1 การออกแบบโปรแกรมหลักเพื่อวิเคราะห์ภาพเพื่อ นับจำนวนพลาของไวรัส	44
3.1.1 การพัฒนาโปรแกรม Algorithm ใน แบบที่ 1	45
3.1.2 การพัฒนาโปรแกรมหรือ Algorithm ใน แบบที่2.....	55
3.1.3 การกำหนดและการเลือกใช้ Algorithms ทั้งสอง	59
3.2 การออกแบบโปรแกรม Graphic User Interface สำหรับการนับพลาของไวรัส.....	59
3.3 การพัฒนาต่อเครื่องถ่ายภาพอัตโนมัติ.....	62
3.3.1 อุปกรณ์ป้องกันแสงจากภายนอก	62
3.3.2 ออกแบบชิ้นส่วนยึดกล้องถ่ายภาพ	64
บทที่ 4	65
การใช้งานโปรแกรมระเบียบวิธีการนับพลาของไวรัสอัตโนมัติ.....	65
4.1 วิธีการใช้งาน โปรแกรม ระเบียบวิธีการนับพลาอัตโนมัติ.....	65
บท 5.....	69
การวิเคราะห์ผล และการประเมินผล.....	69
5.1 วิธีการประเมินผลการนับพลาของไวรัส.....	69
5.1.1 ข้อกำหนดของความถูกต้องการนับปริมาณพลาของไวรัส	70
5.2 การวิเคราะห์ผล ความผันแปรของระบบการวัดของเครื่องถ่ายภาพอัตโนมัติและระเบียบการ นับพลาของไวรัส	71
5.2.1 ความแม่นยำ (Precision).....	71
5.2.1.1 การประเมินผลการนับพลาของไวรัสด้วยโปรแกรม Minitab	72
5.2.1.2 Gauge R and R ของโปรแกรมนับพลาของไวรัสอัตโนมัติ	73
5.3 การวิเคราะห์ผลและประเมินผลการนับพลาของไวรัสจากโปรแกรมนับพลาของไวรัส	75

5.3.1 ความเที่ยงตรง (Accuracy).....	75
5.3.2 การประเมินผลการนับพลาของไวรัสด้วยโปรแกรม Matlab.....	75
5.3.3 Bland – Altman	77
5.3.4 ผลลัพธ์จากการประเมินผลโปรแกรมการนับพลาของไวรัสอัตโนมัติ.....	78
5.4 การประเมินความพึงพอใจจากผู้ใช้งานโปรแกรมนับพลาอัตโนมัติ.....	83
บทที่ 6	85
สรุปผลงานวิจัย	85
6.1 ปัญหาที่พบ	86
6.2 แนวทางในการพัฒนาต่อ.....	86
ภาคผนวก.....	93
ภาคผนวก ก.....	94
ภาคผนวก ข.....	118
บรรณานุกรม.....	124
ประวัติผู้เขียน	126

สารบัญรูปรตาราง

	หน้า
ตารางที่ 1 แสดงผลลัพธ์การนับพลาของไวรัสอัตโนมัติ.....	78
ตารางที่ 2 ตรวจสอบรูปภาพนั้นว่าเข้าเกณฑ์หรือไม่.....	82



สารบัญรูปภาพ

หน้า

รูปที่ 1 รูปภาพที่ผ่านการ Threshold แบ่งแยกวัตถุแล้วเพื่อจะเข้ากระบวนการ Morphological Operation	6
รูปที่ 2 ภาพแสดง กระบวนการ Dilation ด้วยรัศมี 10	7
รูปที่ 3 ภาพแสดง กระบวนการDilation ด้วยรัศมี 31	7
รูปที่ 4 ภาพแสดงกระบวนการ Erosion	8
รูปที่ 5 ภาพแสดง กระบวนการOpening	8
รูปที่ 6 แสดงกระบวนการ Closing.....	9
รูปที่ 7 แสดงตัวอย่างกระบวนการ Boundary Extraction	10
รูปที่ 8 แสดงตัวอย่างกระบวนการ Region Filling	11
รูปที่ 9 แสดงตัวอย่างกระบวนการ Region Filling	11
รูปที่ 10 แสดงตัวอย่างกระบวนการ Components connection เมื่อนำวัตถุในรูปภาพมารวมกัน	12
รูปที่ 11แสดงตัวอย่างรูปภาพที่มีสิ่งรบกวน	13
รูปที่ 12 แสดงตัวอย่างรูปภาพที่มีสิ่งรบกวนที่ผ่าน Arithmetic Mean Filter	14
รูปที่ 13 แสดงตัวอย่างรูปภาพที่มีสิ่งรบกวนที่ผ่าน Median Filter	15
รูปที่ 14แสดงภาพเริ่มต้นก่อนเข้ากระบวนการ Histogram equalization.....	17
รูปที่ 15ภาพแสดงภาพเริ่มต้นหลังเข้ากระบวนการ Histogram equalization	17
รูปที่ 16 ภาพแสดงภาพเริ่มต้นก่อนเข้ากระบวนการ Lowpass Filtering	18
รูปที่ 17 ภาพแสดงภาพเริ่มต้นหลังเข้ากระบวนการ Lowpass Filtering	18
รูปที่ 18 ภาพแสดงภาพเริ่มต้นหลังเข้ากระบวนการ Gamma Correction ที่ค่า 0.5.....	19
รูปที่ 19 แสดงภาพเริ่มต้นหลังเข้ากระบวนการ Gamma Correction ที่ค่า 1.5.....	20
รูปที่ 20 แสดงภาพเริ่มต้นหลังเข้ากระบวนการ Edge Detection ด้วย Prewitt	21

รูปที่ 21 แสดงภาพเริ่มต้นหลังเข้ากระบวนการ Edge Detection ด้วย Sobel.....	22
รูปที่ 22 แสดงภาพที่ทำการจับวัตถุด้วยกระบวนการ Threshold Segmentation	23
รูปที่ 23แสดงภาพที่ทำการจับวัตถุด้วยกระบวนการ Binary Threshold.....	24
รูปที่ 24 แสดงภาพที่ทำการจับวัตถุด้วยกระบวนการ Binary Threshold	25
รูปที่ 25แสดงสมการเส้นตรงใน parameter space	26
รูปที่ 26แสดงจุดตัดบนสมการเส้นตรงสองเส้นเมื่อ a เป็นอนันต์	27
รูปที่ 27 แสดงภาพที่ทำการจับวัตถุด้วยกระบวนการ Hough Transform.....	28
รูปที่ 28แสดงภาพที่ทำการจับวัตถุด้วยกระบวนการ Watershed method ผ่านโปรแกรม Mv tech Halcon.....	29
รูปที่ 29แสดงภาพที่ทำการจับวัตถุด้วยกระบวนการ Watershed method ผ่านโปรแกรม Mat-lab	29
รูปที่ 30แสดงภาพ ขั้นตอน 3.1.1.....	31
รูปที่ 31แสดงภาพ L	31
รูปที่ 32แสดงภาพ a	32
รูปที่ 33แสดงภาพ b.....	32
รูปที่ 34แสดงภาพ ระยะทางจากแต่ละจุดข้อมูลไปยังแต่ละ จุดศูนย์กลาง	33
รูปที่ 35 แสดงภาพ จากค่าดัชนีพิกเซลในรูปภาพในรูปแบบ ของ 0-1.....	34
รูปที่ 36 แสดงวัตถุในรูปภาพของคลาสเตอร์ที่ 1 ซึ่งได้ทำการแยกตัวพลาของไวรัส จากพื้นหลัง... 35	
รูปที่ 37 แสดงวัตถุในรูปภาพของคลาสเตอร์ที่ 2 แยกตัวพื้นหลังออกจากพลาของไวรัส	35
รูปที่ 38 แสดงเครื่องถ่ายภาพรูปแบบพลาอัตโนมัติ.....	37
รูปที่ 39 แสดงเครื่องถ่ายภาพรูปแบบพลาอัตโนมัติและการเชื่อมต่อ	38
รูปที่ 40 แสดง UI ของเครื่องนับพลาอัตโนมัติ	39
รูปที่ 41แสดง UI ขณะที่ทำการ Run Program	40
รูปที่ 42 ภาพแสดง ขั้นตอนในการทำงานกระบวนการนับพลาของไวรัส Algorithm แบบที่1	45

รูปที่ 43 ภาพแสดง ชนิดของรูปภาพที่สามารถนำเข้ามาในโปรแกรม	46
รูปที่ 44 ภาพแสดง ขั้นตอนที่ 2 การจัดเตรียมรูปภาพ	47
รูปที่ 45 ภาพแสดง สาม ภาพ L a b	47
รูปที่ 46 ภาพแสดง ภาพที่นำมาทำการ Matching	48
รูปที่ 47 ภาพแสดง ภาพที่นำมาผ่าน กระบวนการ ลดสิ่งรบกวนโดยการผ่าน Mean Filter ขนาด 10 คูณ 10	49
รูปที่ 48 ภาพแสดงภาพที่นำมาเข้าสู่กระบวนการ Image Thresholding.....	49
รูปที่ 49 ภาพแสดง ภาพที่นำมาเข้าสู่กระบวนการ Morphology	50
รูปที่ 50 ภาพแสดงภาพที่นำมาเข้าสู่กระบวนการ Binary Threshold.....	51
รูปที่ 51 ภาพแสดงภาพจะเป็นการนับพลาของไวรัส จากกระบวนการ Binary Threshold	52
รูปที่ 52 ภาพแสดงภาพจะเป็นการนับพลาของไวรัส จากกระบวนการ K mean clustering	53
รูปที่ 53 แสดงภาพรวมจำนวนคลาสเตอร์ของพลา	54
รูปที่ 54 แสดงผลการนับพลาของไวรัส	54
รูปที่ 55 ภาพแสดง ขั้นตอนในการทำงานกระบวนการนับพลาของไวรัส Algorithm แบบที่ 2.....	55
รูปที่ 56 แสดง Region ใหม่ หลังจากทำการทำการ Binary Threshold.....	56
รูปที่ 57 แสดง ภาพหลังจากเข้ากระบวนการ K mean clustering.....	56
รูปที่ 58 แสดง Region หลังจากเข้ากระบวนการ K mean clustering.....	57
รูปที่ 59 แสดงการนับ Region หลังจากเข้ากระบวนการ K mean clustering	57
รูปที่ 60 แสดงการ Intersection ของ Region.....	58
รูปที่ 61 แสดงผลการนับพลาของไวรัส	58
รูปที่ 62 แสดง Graphic User interface	60
รูปที่ 63 ภาพแสดงพื้นที่ กราฟ จำนวนพลาของไวรัสในแต่ละหลุม	61
รูปที่ 64 แสดงแผนภาพการออกแบบอุปกรณ์ป้องกันแสงและขนาด	62
รูปที่ 65 แสดงเครื่องถ่ายภาพหลังและก่อนประกอบอุปกรณ์ป้องกันแสง	63

รูปที่ 66 แสดงเครื่องถ่ายภาพรูปที่ได้รับการปรับปรุงเสร็จ.....	63
รูปที่ 67 แสดงชิ้นส่วนยึดกล้องเมื่อติดตั้งกับเครื่องถ่ายภาพ	64
รูปที่ 68 ภาพแสดง โปรแกรม Window app from plaque.....	65
รูปที่ 69 ภาพแสดง วิธีการทำการอัปโหลดรูปภาพถ่ายพลาคของไวรัส.....	66
รูปที่ 70 ภาพแสดง วิธีการทำการเซตค่าพารามิเตอร์ต่างๆบนโปรแกรม.....	67
รูปที่ 71 ภาพแสดง โปรแกรม Window app from plaque เมื่อเสร็จสิ้นการทำงาน	67
รูปที่ 72 แสดงผลการนับพลาคของไวรัสแสดงไว้ใน Folder	68
รูปที่ 73 แสดงรูปภาพที่ไม่นำมาประมวลผล.....	70
รูปที่ 74 แสดงการนับพลาคของไวรัสเทียบระหว่างโปรแกรมและผู้เชี่ยวชาญ.....	71
รูปที่ 75 ผลการคำนวณค่าคุณสมบัติเชิงเส้นโดยโปรแกรม Minitab ของโปรแกรมระเบียบวิธีการ นับพลาคของไวรัส	72
รูปที่ 76 ผลการคำนวณค่าไบอัส Bias ของแต่ละค่าโดยโปรแกรม Minitab และกราฟแสดงการ กระจายตัวของข้อมูลการนับ	73
รูปที่ 77 แสดงผลการคำนวณการวิเคราะห์ผลด้วยวิธี Gauge R and R.....	74
รูปที่ 78 แสดงการเปรียบเทียบความแปรปรวนของค่าอ้างอิงที่หลากหลาย.....	75
รูปที่ 79 แสดงการกระจายตัวของข้อมูลบน Box plot.....	76
รูปที่ 80 แสดงค่า Bland – Altman ของโปรแกรมการนับพลาคอัตโนมัติ.....	77
รูปที่ 81 แสดงผลลัพธ์ของโปรแกรมการนับพลาคอัตโนมัติ ความถูกต้อง ความถูกต้องในช่วงอ้างอิง และ ค่าที่นับผิด	79
รูปที่ 82 แสดงผลการนับพลาคของไวรัสกับขอบเขตความถูกต้อง.....	80

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

ระเบียบวิธีการนับพลาควเป็นหนึ่งในกระบวนการวัดขั้นพื้นฐาน [1] เพื่อสำหรับการหาไวรัสที่คงเหลืออยู่ของจากทดลองในห้องปฏิบัติการไวรัสวิทยา กระบวนการระเบียบวิธีการนับ พลาควของไวรัสจะสามารถใช้ได้ในงานวิจัยที่ความซับซ้อนต่ำจนถึงการพัฒนาการรักษาและการพัฒนาวัคซีนที่มีความซับซ้อน [2] โดยในทางทฤษฎี เชื้อไวรัสจะถูกเลี้ยง ณ Cell monolayer ด้วย Semisolid medium ในสภาพแวดล้อมที่เหมาะสม ซึ่ง จะถูกจำกัดการแพร่ในเฉพาะแนวนอนเท่านั้นเมื่อหยอดสารไปบนเพลท ทดลองจากนั้นทำการย้อมสี เซลล์ที่ตายแล้ว(พลาควของไวรัส) จะถูกแสดงผ่านกล้องไมโครสโคป จะทิ้งเป็นคราบสีแดงและสีน้ำเงิน [3,4,5]

ในการนับ พลาควของไวรัสจะทำการนับการนับจากเพลทการทดลอง (Well plate) ซึ่งปกติจะทำการนับด้วยคน และเป็นกระบวนการที่ละเอียดอ่อน ซึ่งผู้ที่ทำการนับจำเป็นที่จะต้องเป็นผู้เชี่ยวชาญและมีประสบการณ์ ในการตรวจหา พลาควของไวรัส กระบวนการระเบียบวิธีการนับพลาควของไวรัสจะมีวิธีดำเนินการ หลายรูปแบบ มีตั้งแต่ ระดับต่ำ ถึง ปานกลางเช่น 6-, -12, -24 -Well Plate เป็นต้น กระบวนการที่กล่าวมาจำเป็นที่จะต้องใช้ผู้ที่มีความทักษะในการตรวจสอบหาพลาควในแต่ละครั้ง

โดยเมื่อไม่นานมานี้ได้มีงานวิจัยที่ทำการพัฒนาระบบกระบวนการระเบียบวิธีการนับพลาควของไวรัสอัตโนมัติ โดยใช้เชื้อไวรัสของโรคเริ่ม[6] และ โรคไข้เลือดออก [7] และมีการสร้าง Automated data acquisition and quantification methods โดยจะพัฒนาระบบอัตโนมัติโดยใช้เครื่อง Flatbed scanner [8] ซึ่งจะมีพื้นฐานจากการถ่ายภาพของพลาควใน Well plate โดยปัจจุบันได้มีความท้าทายในการพัฒนาระบบการนับพลาควอัตโนมัติ เนื่องจากในปัจจุบัน ได้มีการวางขายเครื่องตรวจนับและนับเซลล์ในระดับอุตสาหกรรมแล้ว ยกตัวอย่างเช่น Cellular Technology Limited developed a commercial Elispot and viral plaque-counting machine, ImmunoSpot CLT Analyzers [9] โดยเครื่องตรวจนับเซลล์อัตโนมัติเหล่านี้ ให้ภาพถ่ายผลลัพธ์จาก well plate ที่มีความคมชัดและรวดเร็ว ซึ่งจะสามารถนับพลาควของไวรัสได้โดยอัตโนมัติ และประมวลผลผ่านทางซอฟต์แวร์ แต่อย่างไรก็ตาม ซอฟต์แวร์ที่กล่าวถึงจะมุ่งเน้นไปทางการนับ ไวรัสชนิดของ Eli spot ซึ่งตัวซอฟต์แวร์ก็จะมีข้อดีที่ผู้ใช้งานจะสามารถปรับค่าต่างๆที่มีผลต่อการนับพลาควของ

ไวรัส เช่น ความเข้มข้น และขนาดในแต่ละ Well plate อย่างไรก็ตามก็ยังมีข้อเสียที่ การปรับค่าต่างๆ ให้เข้ากับการนับแต่ละครั้งนั้น ยังทำได้ค่อนข้างยาก และ ตัวเครื่องตรวจจับและนับเซลล์ในระดับ อุตสาหกรรมนี้มีราคาที่สูงมาก

สำหรับโปรแกรมการนับพลาควาผ่านโปรแกรมบนคอมพิวเตอร์ส่วนตัวนั้น จะสามารถทำได้ โดยได้โดยใช้โปรแกรมประมวลผลภาพต่างๆ เช่น ImageJ, OpenCV, Labview, R [7, 12-19]. โดยงานวิจัยของ คุณ S. Boonyasuppayakorn และคณะ เรื่อง Simplified dengue virus microwell plaque assay using an automated quantification program [7] ได้ทำการวิจัยผ่านโปรแกรม ImageJ เพื่อใช้งานกับ 96-well ในการนับพลาควาอัตโนมัติของ ไวรัสโรคไข้เลือดออก โดยใช้เครื่อง flatbed scanner เพื่อทำการสแกนตัว 96-well plate เพื่อจะได้รูปของ 96-well plate จากนั้นนำภาพ 96-well plate ทำการแบ่งแต่ละช่อง ภาพละ 1 ช่อง (Well) แต่อย่างไรก็ตามตัว เพลทการทดลองจะต้องทำการเพิ่มความเข้มข้นของของเหลวทึบแสงเช่น นม ก่อนจะทำการสแกน เพื่อให้ไวรัสในเพลทการทดลองสามารถมองเห็นได้ชัดเจนยิ่งขึ้น ในงานวิจัยบางส่วน[14-19] ได้ทำการพัฒนาโปรแกรมการนับพลาควาจากทฤษฎีการกระบวนประมวลผลภาพเช่นกัน โดยจะใช้ กระบวนการแบ่งวัตถุออกจากภาพ กระบวนการ Morphological และกระบวนการ Image Threshold แต่ในกระบวนการนับพลาควาของไวรัสบางชนิดนั้น การใช้ทฤษฎีการกระบวนประมวลผลภาพ อาจจะยังให้ผลลัพธ์ที่มีความแม่นยำเท่าที่ควร โดย ณ ปัจจุบันนั้นยังไม่มี การพัฒนาระบบกระบวนการระเบียบวิธีการนับพลาควาของไวรัสอัตโนมัติโดยใช้ กระบวนการ Machine learning ผสมผสานกับ กระบวนการประมวลผลภาพ (Image Processing) ซึ่งงานวิจัยนี้จะเป็นการพัฒนาต่อจากงานวิจัยต่อเนื่องจากงานวิจัย Machine-Learning-based Automated Quantification Machine for Virus Plaque Assay Counting [31] ซึ่งงานวิจัยก่อนหน้าได้ทำการ สร้างเครื่องนับพลาควาของไวรัสอัตโนมัติ โดยใช้ พื้นฐานการประมวลผลข้อมูล ด้วยกระบวนการ Machine learning เข้ามาซึ่งจะช่วยลดภาระในการทำงานของมนุษย์และเพิ่มความแม่นยำในการตรวจจับหาพลาควาของไวรัส โดยตัวเครื่องนั้นจะประกอบไปด้วยตัว Hardware ไว้ถ่ายรูปเพลทของไวรัส และ Software สำหรับนับจำนวนพลาควาของไวรัส ซึ่งจะใช้กระบวนการ K-mean clustering ซึ่งเป็นกระบวนการหนึ่งใน กระบวนการ Machine learning ชนิด Unsupervised Learning เพื่อช่วยให้ผู้เชี่ยวชาญและผู้ต้องการใช้งานนั้น ได้แบ่งเบาภาระงานในการทดลองการนับพลาควาของไวรัส แต่งานวิจัยยังมีส่วนให้พัฒนาต่อ โดยต้องการเพิ่มความแม่นยำในโปรแกรมนับพลาควาของไวรัสอัตโนมัติให้เพิ่มมากขึ้น และเป็นมิตรต่อผู้ใช้งาน

จากแนวทางในการพัฒนาต่อจึงได้เป็น ที่มาของงานวิจัยนี้ที่จะทำการพัฒนา Algorithm ต่อ และปรับปรุงเพิ่มความเร็วภาพในกับระบบ ให้มีความแม่นยำมากยิ่งขึ้น และ พัฒนา Graphic User interface ให้เป็นมิตรต่อผู้ใช้งานโดยผู้ใช้งานไม่จำเป็นจะต้องมีความรู้เกี่ยวกับการเขียน โปรแกรม

1.2 วัตถุประสงค์

1.2.1 พัฒนาโปรแกรมระเบียบวิธีการนับจำนวนพลาของไวรัสให้มีความแม่นยำมากยิ่งขึ้น

1.2.2 พัฒนาโปรแกรมซอฟต์แวร์สำหรับระเบียบวิธีการนับจำนวนพลาของไวรัสให้มีความ เสถียรภาพ ใช้งานง่าย และเพิ่มเติมฟังก์ชันการใช้งานเทียบจากอุปกรณ์ที่มีขายอยู่ในท้องตลาด

1.3 ขอบเขตของงานวิจัย

1.3.1 พัฒนาโปรแกรมระเบียบวิธีการนับจำนวนพลาของไวรัสให้มีความแม่นยำมากยิ่งขึ้น เทียบกับงานวิจัยก่อนหน้านี้โดยต่อยอดพัฒนาAlgorithmต่อจากงานวิจัยเดิม

1.3.2 พัฒนาโปรแกรมซอฟต์แวร์สำหรับระเบียบวิธีการนับจำนวนพลาของไวรัส โดยจะ พัฒนา Graphic User Interface ให้ง่ายต่อการใช้งาน และเพิ่มเติมฟังก์ชันในโปรแกรม

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 โปรแกรมระเบียบวิธีการนับจำนวนพลาของไวรัสสามารถนับจำนวนพลาของไวรัสมี ความถูกต้องและแม่นยำมากยิ่งขึ้น

1.4.2 โปรแกรมซอฟต์แวร์สำหรับระเบียบวิธีการนับจำนวนพลาของไวรัส มีความเสถียร ภาพและง่ายต่อการใช้งาน

1.5 ขั้นตอนการวิจัย

1.5.1 ศึกษางานวิจัยก่อนหน้านี้ การพัฒนาอุปกรณ์ถ่ายภาพอัตโนมัติและระเบียบวิธีการนับพ ลาคของไวรัส เพื่อนำมาศึกษาพัฒนาต่อยอดและ เพื่อเป็นประโยชน์และแนวทางในการพัฒนาต่อ

1.5.2 ศึกษางานวิจัยที่เกี่ยวข้อง เพื่อเป็นประโยชน์ในการพัฒนาโปรแกรมระเบียบวิธีการนับ จำนวนพลาของไวรัสอัตโนมัติ

1.5.3 ศึกษากระบวนการวิเคราะห์ภาพ โดยมุ่งเน้นไปทางการแบ่งแยกวัตถุและการปรับภาพ

1.5.4 ศึกษากระบวนการแมชชีนเลิร์นนิ่ง เพื่อเป็นประโยชน์แนวทางในการเพิ่มความแม่นยำในระบบ



บทที่ 2

อุปกรณ์และทฤษฎีที่เกี่ยวข้องในการศึกษา

ในการทำวิจัยนี้เป็นการคิดและออกแบบพัฒนาต่อยอดโปรแกรมสำหรับการนับพลาของไวรัส ซึ่งจะประกอบไปด้วย ทฤษฎีทางการวิเคราะห์ภาพ และการพัฒนาโปรแกรม

2.1 ทฤษฎีการประมวลผลภาพ

ทฤษฎีการประมวลผลภาพ เป็นวิธีการจัดการรูปภาพที่อยู่ในรูปแบบดิจิทัลด้วยวิธีการทางคอมพิวเตอร์ เพื่อเพิ่มประสิทธิภาพของรูปภาพให้มีความชัดเจน ความแม่นยำ และการแสดงผลภาพให้เป็นไปตามประสงค์ของผู้ที่ทำการศึกษา ข้อมูลนั้นๆ [22]

2.1.1 Morphological Operation

กระบวนการ Morphology นั้นเป็นกระบวนการหนึ่งของการประมวลผลภาพจะเป็นการทำงานในรูปแบบเชิงคณิตศาสตร์ ซึ่งจะใช้สำหรับหาโครงร่างหรือรูปร่างของวัตถุนั้นๆ โดยการทำงานจะแยกส่วนประกอบต่างๆออกจากภาพ ให้เป็นวัตถุเพื่อสามารถนำวัตถุนี้ไปวิเคราะห์ต่อไป ซึ่งในงานวิจัยนี้ได้นำกระบวนการ Morphology มาหารูปร่างพลาของไวรัสนั่นเอง [23]

2.1.1.1 กระบวนการจัดการ Morphology และ เชื่อมต่อวัตถุขอบเขต (Region Connectivity)

ในกระบวนการ Morphology จะเป็นการกระทำต่อวัตถุในรูปภาพที่สนใจและ ที่ได้กำหนดไว้ในขอบเขต (Region) โดย ต้องกำหนดขอบเขตที่สนใจเพื่อนำมาวิเคราะห์ข้อมูล

Morphological Operation จะเป็นการทำงานในรูปแบบเชิงคณิตศาสตร์ ในงานวิจัยนี้ได้ทำการนำกระบวนการทาง Morphology มาใช้สี่รูปแบบ ก็คือ การขยายวัตถุ Dilation การกัดกร่อนวัตถุ Erosion การทำปรับผิวของวัตถุด้วยวิธีลบคมจากด้านนอก Opening การทำปรับผิวของวัตถุด้วยวิธีลบคมจากด้านใน Closing โดยจะอธิบายในรูปแบบของสมการคณิตศาสตร์ได้ว่า [23]

$$\text{การขยายวัตถุ Dilation : } A \oplus B = A \cup b : b \in B; \quad (2.1)$$

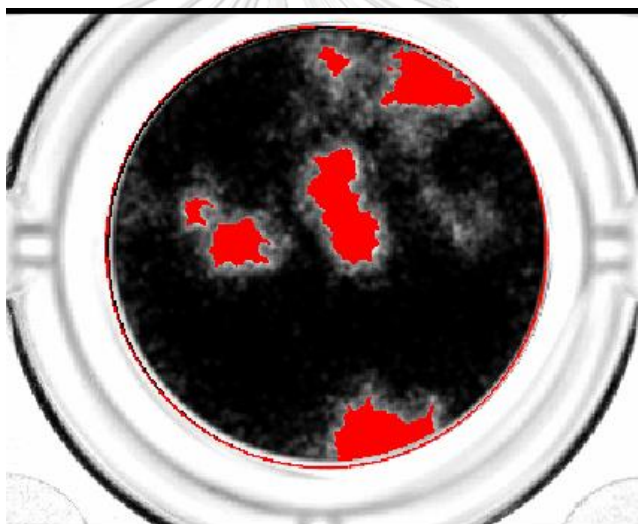
$$\text{การกัดกร่อนวัตถุ Erosion : } A \ominus B = A \cap b : b \in B; \quad (2.2)$$

โดยกำหนดให้ A เป็น รูปภาพที่จะนำมาวิเคราะห์ B เป็น โครงร่างสัดส่วนที่จะนำมากระทำต่อรูปภาพ A และ สัญลักษณ์ \cup และ \cap แสดงถึงกระบวนการ Union และ Intersect

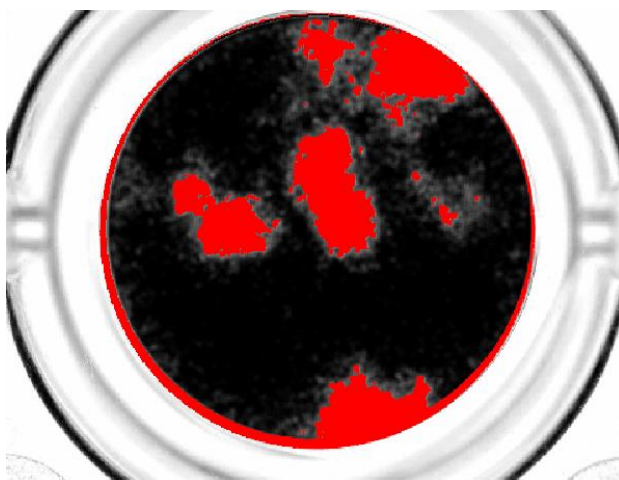
$$\text{Opening } (A \ominus B) = (A \ominus B) \oplus B \quad (2.3)$$

$$\text{Closing } (A \odot B) = (A \oplus B) \ominus B \quad (2.4)$$

การทำงานของกระบวนการ Opening และ Closing จะเป็นการต่อยอดการทำงานจาก กระบวนการ Dilation และ Erosion โดยกระบวนการ Opening จะเป็นการนำวัณณในภาพที่กำหนดไว้ในขอบเขต มาเข้ากระบวนการ Erosion ก่อนแล้วจึงนำไปเข้ากระบวนการ Dilation ในทางกลับกัน กระบวนการ Closing จะเป็นการนำวัณณแบบเดียวกันแต่จะเข้ากระบวนการ Dilation ก่อนแล้วจึงไปยัง กระบวนการ Closing นั่นเอง [23]



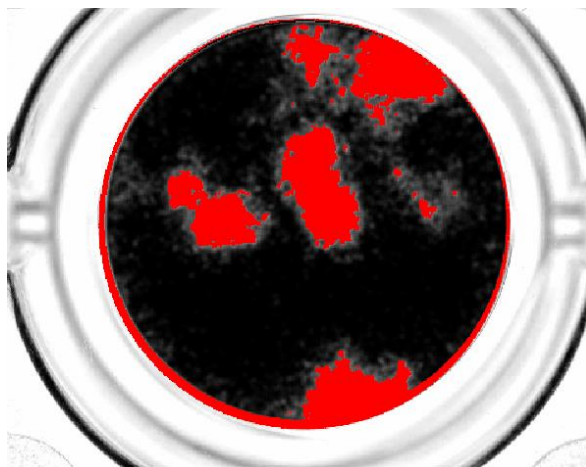
รูปที่ 1 รูปภาพที่ผ่านการ Threshold แบ่งแยกวัณณแล้วเพื่อจะเข้ากระบวนการ Morphological Operation



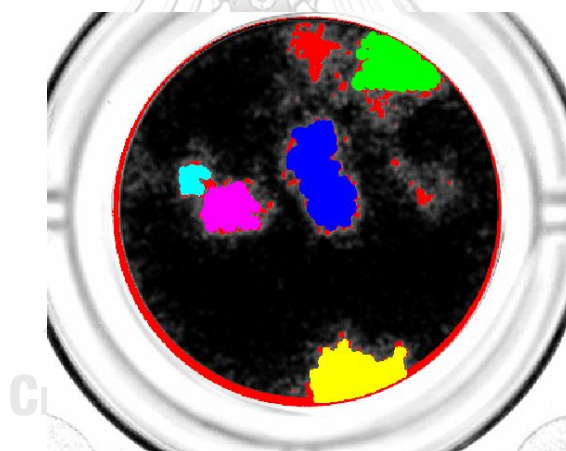
รูปที่ 2 ภาพแสดง กระบวนการ Dilation ด้วยรัศมี 10



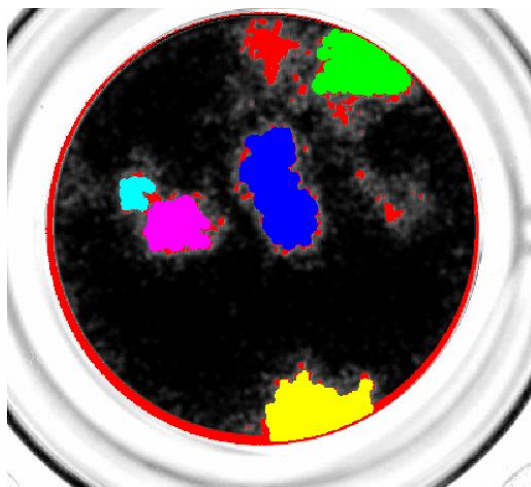
รูปที่ 3 ภาพแสดง กระบวนการDilation ด้วยรัศมี 31



รูปที่ 4 ภาพแสดงกระบวนการ Erosion



รูปที่ 5 ภาพแสดง กระบวนการ Opening

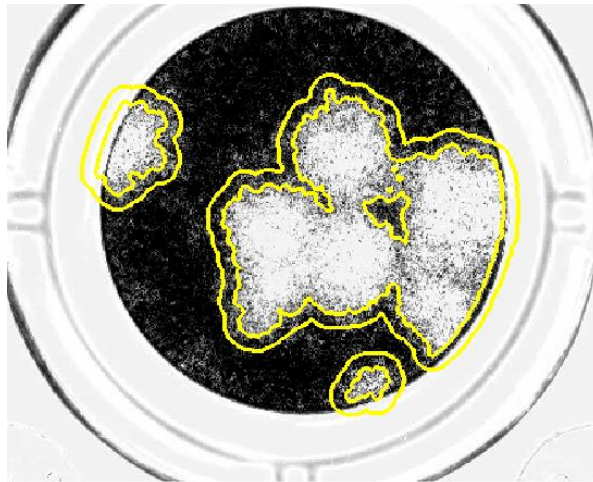


รูปที่ 6 แสดงกระบวนการ Closing

2.1.1.2 Morphological Algorithms พื้นฐาน

2.1.1.2.1 Boundary Extraction

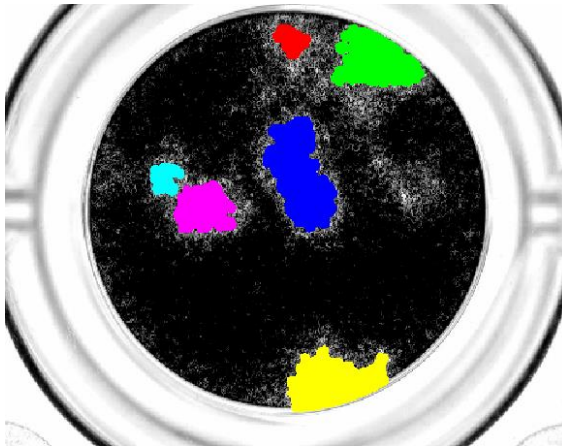
กระบวนการ Boundary Extraction เป็นการนำ ทฤษฎี Morphological Operation มากระทำต่อวัตถุเพื่อต้องการจะแยกขอบเขตของวัตถุนั้นโดยจะทำการ Erosion พิกเซลที่ขอบวัตถุด้วย Structuring element จากนั้นนำวัตถุก่อน Erosion มาทำการลบออก จากนั้นจะได้ขอบของวัตถุนั้นๆ สามารถอธิบายเป็น สมการคณิตศาสตร์ได้ว่า $B(A) = A - (A \ominus B)$ โดย $B(A)$ จะแทนขอบเขตของวัตถุ A และ B เป็น Structuring element [22]



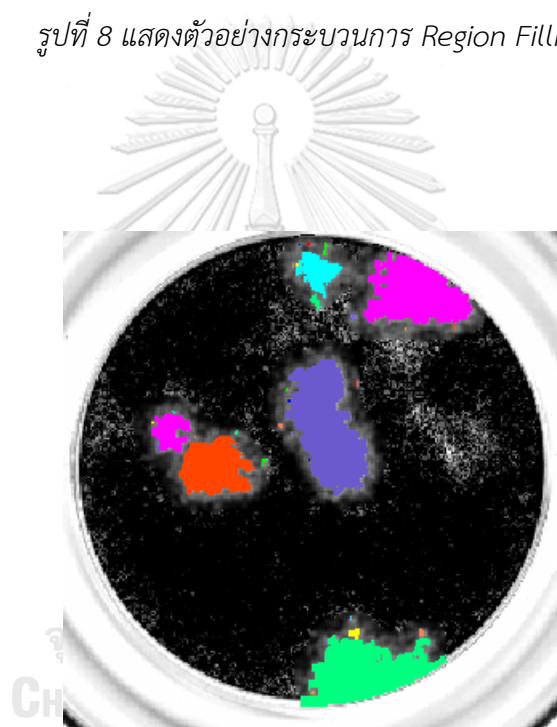
รูปที่ 7 แสดงตัวอย่างกระบวนการ Boundary Extraction

2.1.1.2.2 Region Filling

กระบวนการ Region Filling จะเป็นกระบวนการทำงานที่ตรงข้ามกับ Boundary Extraction โดยจะทำการเติมเต็มขอบเขตของภาพหรือวัตถุนั้นๆ (วัตถุมีรู) สามารถอธิบายตามสมการได้ดังนี้ $R_k = (R_{k-1} \oplus B) \cap A^c$ โดยกำหนดให้ R_{k-1} เป็นขอบเขตที่ต้องการเติมเต็ม (วัตถุมีรู) โดยมีจำนวนพิกเซล K และ B เป็น Structuring element และ A เป็นรูปภาพเริ่มต้นที่มีวัตถุ R_k อยู่ โดยกระบวนการนั้นจะทำการ Dilation ระหว่าง R_{k-1} และ Structuring element (B) จากนั้นไปอินเตอร์เซกกับ รูปภาพตั้งต้น A^c (เฉพาะส่วนที่อยากจะเติมเต็ม) และหยุดกระบวนการเมื่อ R_k ขอบเขตที่เติมเต็มเสร็จแล้วมีค่าเท่ากับ R_{k-1} [22]



รูปที่ 8 แสดงตัวอย่างกระบวนการ Region Filling

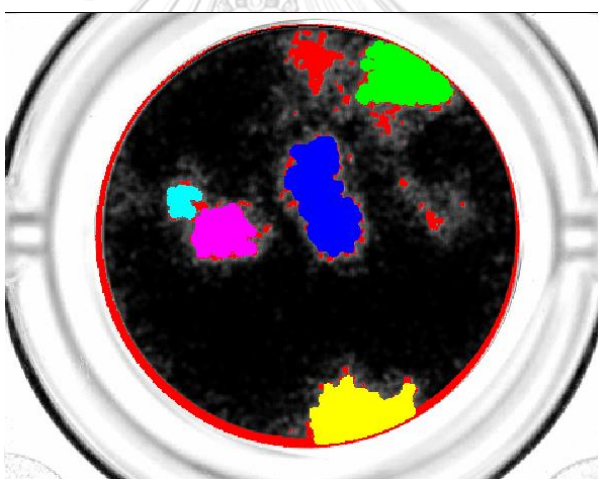


รูปที่ 9 แสดงตัวอย่างกระบวนการ Region Filling

2.1.1.2.3 Components Connection

กระบวนการนี้เป็นการเชื่อมต่อวัตถุที่สนใจให้เชื่อมต่อกันหรือทำให้วัตถุสองชิ้นขึ้นไปสามารถเชื่อมต่อกันเป็นวัตถุชิ้นเดียว กระบวนการ Connection จะเหมือนกับกระบวนการ Region Filling แต่จะต่างกันตรงที่ กระบวนการ Dilation ของวัตถุ ในกระบวนการนี้ จะรวมกับภาพดั้งเดิมทั้งชิ้น ไม่ใช่แค่ส่วนประกอบของชิ้นนั้นๆ สามารถอธิบายเป็นสมการได้ว่า

$R_k = (R_{k-1} \oplus B) \cap A$ โดย R_k เป็นขอบเขตที่เชื่อมต่อกันแล้วมีจำนวนพิกเซล K โดย เริ่มจาก R_{k-1} เป็นขอบเขตของวัตถุที่ยังไม่ได้เชื่อมกันโดย จะทำการ Dilation กับ Structuring element (B) จากนั้นไปอินเตอร์เซกกับ รูปภาพดั้งเดิม [22]



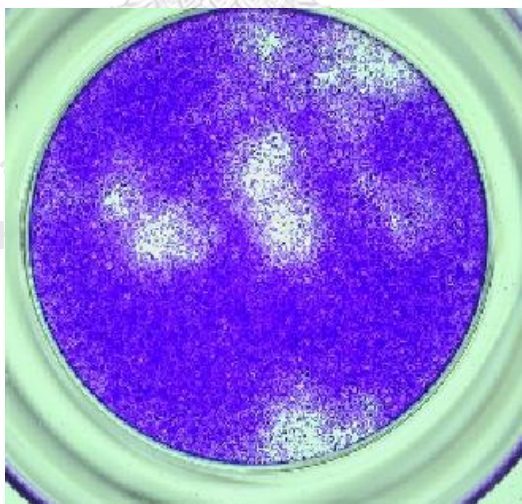
รูปที่ 10 แสดงตัวอย่างกระบวนการ Components connection เมื่อนำวัตถุในรูปภาพมารวมกัน

2.1.2 Restoration Image Operation

ในการนำรูปภาพเข้ามาในกระบวนการ ทฤษฎีการประมวลผลภาพนั้น รูปภาพตั้งต้น จะประกอบไปด้วยสิ่งรบกวนและความไม่คมชัด จึงจำเป็นต้องนำรูปภาพตั้งต้นเหล่านี้ไปดำเนินการปรับแต่งความคมชัดและลดสิ่งรบกวน ก่อนเข้าการประมวลผลภาพ เพื่อจะทำให้ผลลัพธ์มีประสิทธิภาพที่ดียิ่งขึ้น โดยการ Restoration Image (กระบวนการปรับปรุงภาพ) ในที่นี้จะอ้างถึงเฉพาะกระบวนการที่ใช้ในงานวิจัยนี้เท่านั้น โดยจะแบ่งออกเป็นสองส่วน คือ การกำจัดสิ่งรบกวน และการปรับภาพให้มีความคมชัด [22]

2.1.2.1 การกำจัดสิ่งรบกวน

สิ่งรบกวน (Noise) เป็นสัญญาณที่ขึ้นแบบสุ่ม จะเกิดขึ้นเมื่อการถ่ายภาพมีแสงไม่เพียงพอ หรือ เกิดการสั่นของกล้องถ่ายรูป รวมไปถึงกล้องอาจจะมีความคมชัดไม่เพียงพอ โดยเมื่อถ่ายภาพมานั้น ภาพส่วนใหญ่จะมี สิ่งรบกวนในภาพ เมื่อสังเกตภาพนั้นแล้ว เราจะสามารถกำจัดสิ่งรบกวนนั้นได้ตามวิธีที่เหมาะสมต่อสิ่งรบกวนนั้นๆ [22]

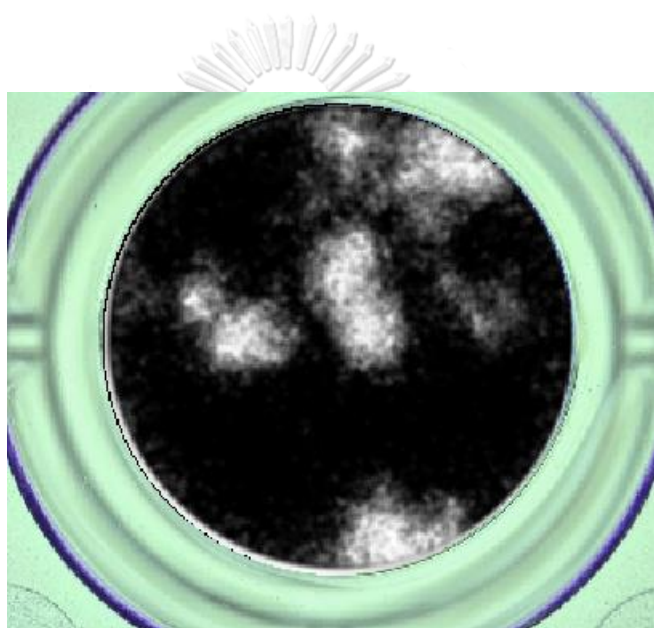


รูปที่ 11 แสดงตัวอย่างรูปภาพที่มีสิ่งรบกวน

2.1.2.1.1 Arithmetic Mean Filter (Average Filter)

Arithmetic Mean Filter เป็นฟิลเตอร์ชนิดหนึ่งเพื่อลดสิ่งรบกวน เช่นกัน โดยฟิลเตอร์ชนิดนี้จะทำงานได้ดีกับรูปภาพที่มีความคมชัดต่ำหรือ เบลอนั่นเอง สามารถอธิบายเป็นสมการได้ว่า[22]

$$\bar{x} = \frac{(x_1 + x_2 + \dots + x_n)}{n} \quad (2.5)$$



รูปที่ 12 แสดงตัวอย่างรูปภาพที่มีสิ่งรบกวนที่ผ่าน Arithmetic Mean Filter

2.1.2.1.2 Median Filter

Median filtering คือฟิลเตอร์ที่ช่วยกำจัดสิ่งรบกวน (Noise) โดยจะนำพิกเซลส์แต่ละตัวมาทำการหาค่ากลาง โดยฟิลเตอร์ชนิดนี้ส่วนใช้จะใช้กำจัด Salt and Pepper noise ในรูปภาพ สามารถอธิบายเป็นสมการได้ว่า [22]

$$f(x, y) = \text{median}\{g(i, j)\} \quad (2.6)$$

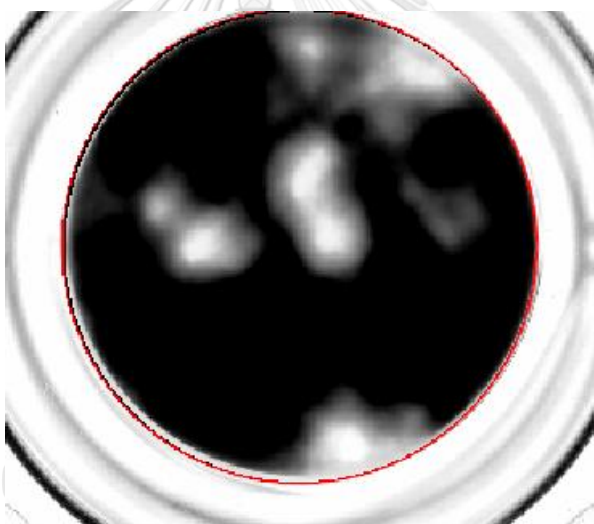
โดย

$$\text{median} ; \sigma^2 = \frac{1}{(i)(j)} \sum_{i=0}^{i-1} \sum_{j=0}^{j-1} (x(i, j) - \bar{x})^2 \quad (2.7)$$

$x(i, j)$ เป็นภาพเริ่มต้น มีจำนวนพิกเซล i ในแกน x และ จำนวนพิกเซล j ในแกน y

σ เป็นส่วนเบี่ยงเบนมาตรฐาน

\bar{x} เป็นค่าเฉลี่ยของ $x(i, j)$



รูปที่ 13 แสดงตัวอย่างรูปภาพที่มีสิ่งรบกวนที่ผ่าน Median Filter

2.1.2.1.2 Geometric Mean Filter

Geometric Mean Filter เป็นฟิลเตอร์อีกหนึ่งชนิดที่นำมาทำการลดสิ่งรบกวนในรูปภาพได้ โดยฟิลเตอร์ชนิดนี้จะทำให้รูปภาพ มีความเรียบเนียนมากยิ่งขึ้น แต่ข้อเสียคืออาจจะทำให้รูปภาพเสียรายละเอียดบางอย่างไป โดยการทำงานของฟิลเตอร์นี้จะทำการนำเกรย์เลเวลล์พิกเซลทั้งหมดของ รูปมาค่าการคูณกันและทำการยกกำลังด้วย ขนาดของฟิลเตอร์ที่ต้องการ ดังสมการ [22]

$$G(x, y) = [\prod_{i,j} S(i, j)]^{\frac{1}{mn}} \quad (2.8)$$

2.1.2.2 การปรับภาพให้มีความคมชัด

เนื่องรูปภาพพลาคของไวรัสที่นำเข้ามาเพื่อทำการวิเคราะห์ จากการถ่ายรูปจากเพลตของ ไวรัส รูปภาพที่ได้มา ยังไม่มีความชัดเพียงพอ จึงจำเป็นต้องใช้ทฤษฎีการปรับภาพให้มีความคมชัด ยิ่งขึ้น [22]

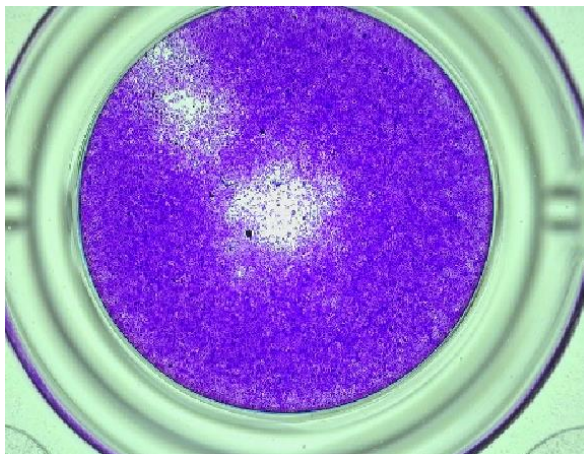
2.1.2.2.1 กระบวนการ Histogram Equalization

การปรับภาพสีเริ่มต้นให้มีความคมชัดนั้น จะสามารถทำได้ด้วย กระบวนการ Histogram equalization ซึ่งจะดูความหนาแน่นของของพิกเซลในรูปภาพในรูปแบบของกราฟ Histogram เมื่อได้กราฟแล้วจะนำแต่ละพิกเซลมาจัดเรียงใหม่ตามสมการ

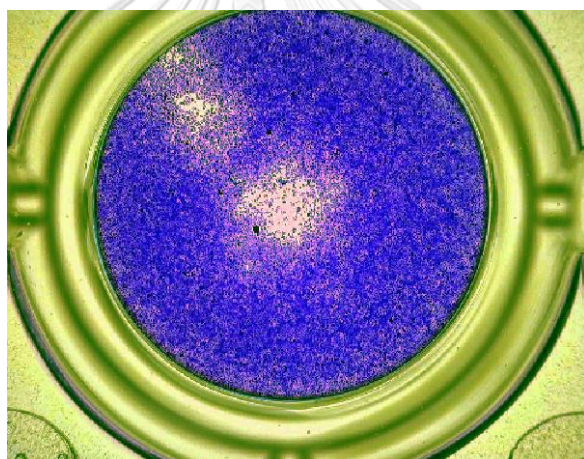
จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

$$S_k = \sum_{j=0}^k \frac{n_j}{n} (L - 1) \quad (2.9)$$

เมื่อ k มีค่าตั้งแต่ $\{0, 1, \dots, (L-1)\}$ และ n คือจำนวนพิกเซลในรูปภาพนั้นๆ เมื่อได้ค่า S_k ใหม่แล้วจึง นำมา สร้างกราฟ Histogram ผลลัพธ์จะแสดงให้เห็นว่ากราฟ Histogram ของพิกเซลในรูปภาพจะ มีความสูงที่ใกล้เคียงกัน และไม่รวมตัวอยู่ในจุดเดียว [22]



รูปที่ 14 แสดงภาพเริ่มต้นก่อนเข้ากระบวนการ Histogram equalization



รูปที่ 15 ภาพแสดงภาพเริ่มต้นหลังเข้ากระบวนการ Histogram equalization

2.1.2.2.3 Lowpass Filtering

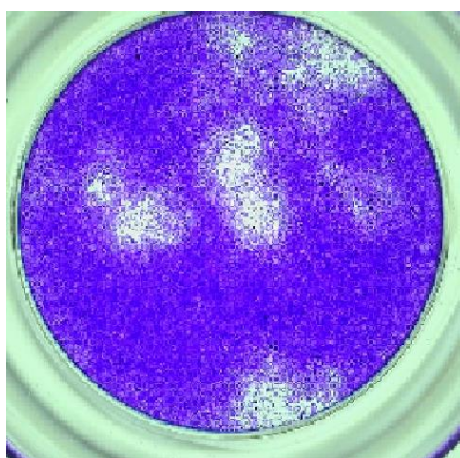
Lowpass Filtering คือฟิลเตอร์ชนิดหนึ่ง เมื่อนำพิกเซลจากภาพสีเริ่มต้นผ่านการด้วยกรองฟิลเตอร์นี้แล้วภาพจะมีความคมชัดยิ่งขึ้นและกำจัดสิ่งรบกวนไปในตัว โดยฟิลเตอร์จะเป็นแบบ 2 มิติ และจะสนใจพิกเซลที่มีสัญญาณความถี่ต่ำให้สามารถผ่านตัวกรองไปได้โดย แบ่งการทำงานในแต่ละช่องสี R G B โดยขนาดของฟิลเตอร์นั้นจะมีการใช้ที่แตกต่างกันออกไป เช่น เป็นเมทริกซ์ขนาด 3×3 หรือ 11×11 สามารถอธิบายการทำงานได้ดังสมการด้านล่าง [22]

$$y(k, l) = \sum_n h_r(n) \sum_m h_c(m) x(k - m, l - n) \quad (2.10)$$

โดย $x(m, n)$ เป็นภาพเริ่มต้นมีจำนวนพิกเซล m ในแกน x และ n จำนวนพิกเซลในแกน y

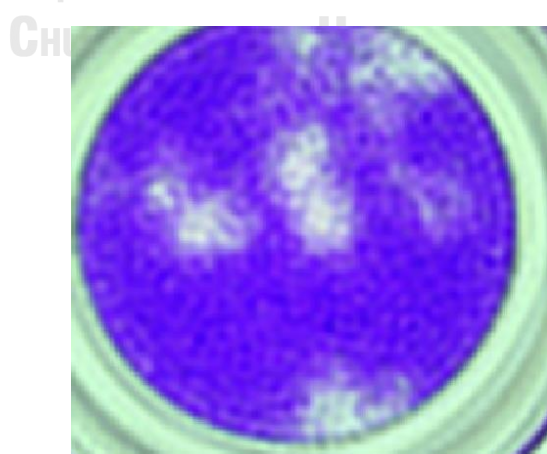
h_r เป็น ฟิลเตอร์(แถว) และ h_c เป็นคอลัมน์ฟิลเตอร์(หลัก) เรียกอีกอย่างได้ว่า

Averaging filter



รูปที่ 16 ภาพแสดงภาพเริ่มต้นก่อนเข้ากระบวนการ Lowpass Filtering

จุฬาลงกรณ์มหาวิทยาลัย



รูปที่ 17 ภาพแสดงภาพเริ่มต้นหลังเข้ากระบวนการ Lowpass Filtering

2.1.2.2.4 Gamma Correction

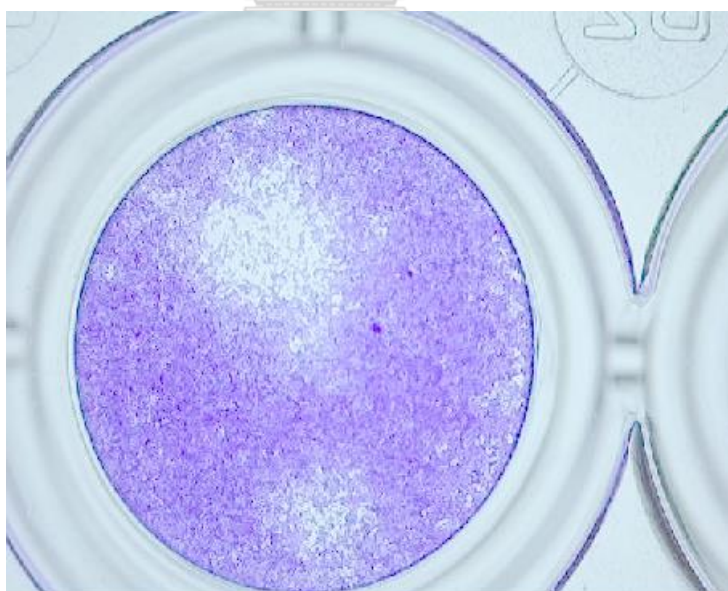
Gamma Correction เป็นอีกหนึ่งวิธีการในการปรับภาพให้มีความคมชัด เนื่องจากภาพเริ่มต้นในบางภาพนั้นความหนาแน่นของพิกเซลที่มีความ Nonlinearity เพื่อเป็นการทำให้พิกเซลมีความ linear จึงทำการยกกำลังค่าพิกเซลแต่ละตัวในภาพเริ่มต้น ด้วยค่าเกรมม่า (γ) [22] ดังสมการ 2.11

$$m' = m^\gamma \quad (2.11)$$

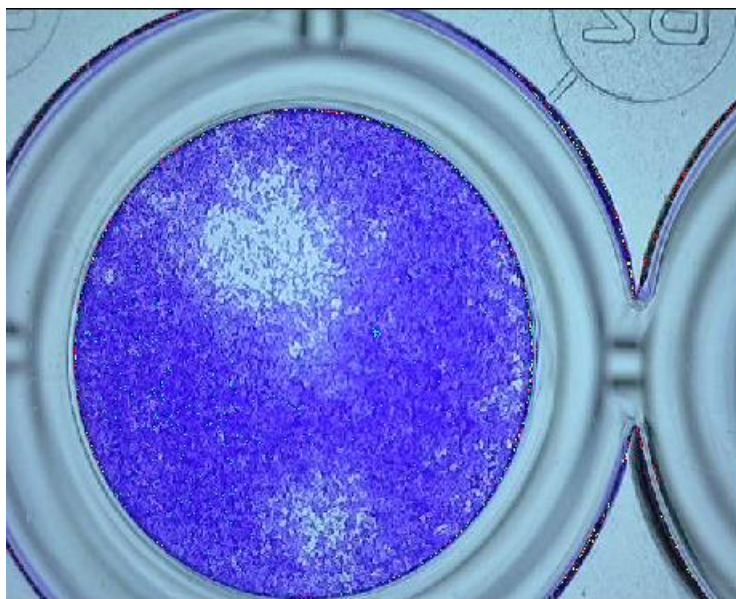
โดย m เป็นพิกเซลของภาพเริ่มต้น

m' เป็น พิกเซลที่ผ่านการยกกำลังโดยค่าเกรมม่าแล้ว

m	0	0.1	0.2	0.3	0.5	0.6	0.7	0.9	1
$m^{0.5}$	0	0.316	0.447	0.547	0.707	0.774	0.83	0.948	1
$m^{1.5}$	0	0.031	0.089	0.164	0.353	0.4647	0.585	0.853	1



รูปที่ 18 ภาพแสดงภาพเริ่มต้นหลังเข้ากระบวนการ Gamma Correction ที่ค่า 0.5



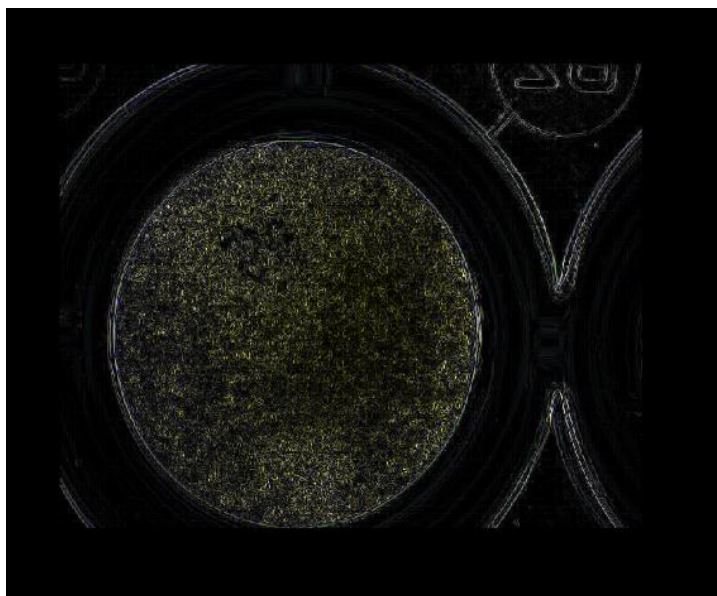
รูปที่ 19 แสดงภาพเริ่มต้นหลังเข้ากระบวนการ Gamma Correction ที่ค่า 1.5

2.1.3 Image Segmentation Operation

หลังจากที่ผ่านกระบวนการ ปรับแต่งภาพและ กำจัดสิ่งรบกวน แล้วถึงส่วนที่จะต้องแบ่ง ส่วนประกอบของภาพว่าสิ่งต่างๆในภาพประกอบด้วยอะไรบ้างด้วยกระบวนการนี้จะเป็นการแบ่ง ส่วนประกอบของภาพ [22]

2.1.3.1 Edge Detection

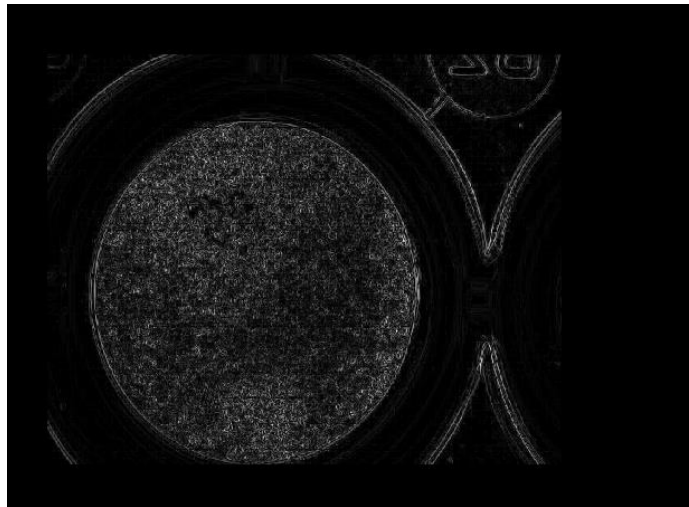
Edge Detection เป็นการกำหนดขอบเขตของพื้นที่ หรือวัตถุที่สนใจในรูปภาพ ซึ่ง Edge เกิดจากเส้นของสองพื้นผิวมาบรรจบกัน หรือเป็นการแตกต่างของความหนาแน่นของพิกเซลใน รูปภาพ ซึ่ง จะสามารถนิยามได้ว่าขอบเขตของวัตถุนั้นๆเรียกว่า Edge โดย พื้นฐานวิธีทั่วไปในการทำ Edge Detection จะเป็นการใช้ High pass filtering ซึ่งจะประกอบไปด้วย Prewitt และ Sobel Operator [22]



รูปที่ 20 แสดงภาพเริ่มต้นหลังเข้ากระบวนการ Edge Detection ด้วย Prewitt

$$\text{Prewitt} = \begin{bmatrix} -1 & 0 & -1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} ; \text{Gradient in n -direction} \quad (2.12)$$

$$\text{Prewitt} = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} ; \text{Gradient in m -direction} \quad (2.13)$$



รูปที่ 21 แสดงภาพเริ่มต้นหลังเข้ากระบวนการ Edge Detection ด้วย Sobel

$$\text{Sobel} = \begin{bmatrix} -1 & -1 & -1 \\ -2 & 0 & 2 \\ -1 & 1 & 1 \end{bmatrix} ; \text{Gradient in n -direction} \quad (2.14)$$

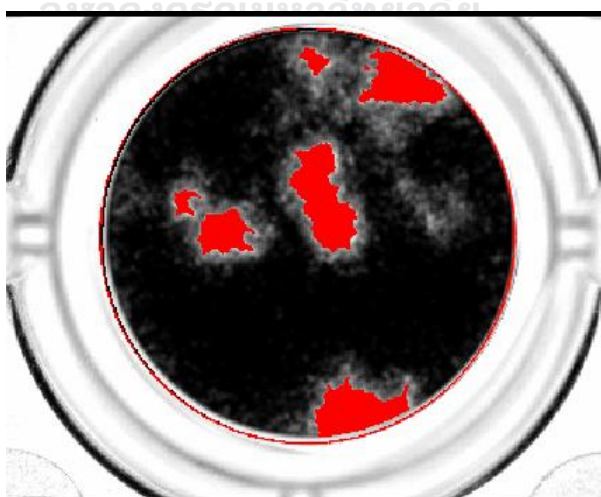
$$\text{Sobel} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} ; \text{Gradient in m -direction} \quad (2.15)$$

2.1.3.2 Threshold Segmentation

กระบวนการ Threshold เป็นกระบวนการในการแบ่งวัตถุในรูปภาพบนพื้นฐานในช่วงของความหนาแน่นของพิกเซลนั้นๆว่ามีความหนาแน่นอยู่ในช่วงไหน โดยรูปภาพนั้นจะถูกแปลงเป็นภาพขาวดำเมื่อทำการ Threshold เสร็จรูปภาพจะมีค่า Gray scale อยู่ระหว่าง 0 – 256 levels โดยส่วนใหญ่ Thresholding จะทำการแบ่งตัววัตถุที่สนใจออกจากพื้นหลัง โดยจะสามารถแบ่งได้ตามสมการดังนี้ [22]

$$x(m,n) = \begin{cases} Obj & \text{if } x(m,n) > T \\ B & \text{if } x(m,n) \leq T \end{cases} \quad (2.16)$$

โดย *Obj* คือวัตถุที่ต้องการจะแยก และ *B* คือ พื้นหลัง สามารถอธิบายได้ว่าถ้า พิกเซลไหนมีค่ามากกว่าค่า *T* จะเป็นวัตถุจะเป็นสีขาวในภาพขาวดำ และ พิกเซลที่มีค่าน้อยกว่า *T* จะกลายเป็นพื้นหลังสีดำนั่นเองโดยถ้าไม่มีการกำหนดขอบเขตที่จะ Threshold จะเรียกได้อีกอย่างว่า Global Thresholding จะสามารถแสดงเป็น Histogram ของจำนวนพิกเซลเพื่อให้เข้าใจในการทำงานมากยิ่งขึ้น



รูปที่ 22 แสดงภาพที่ทำการจับวัตถุด้วยกระบวนการ Threshold Segmentation

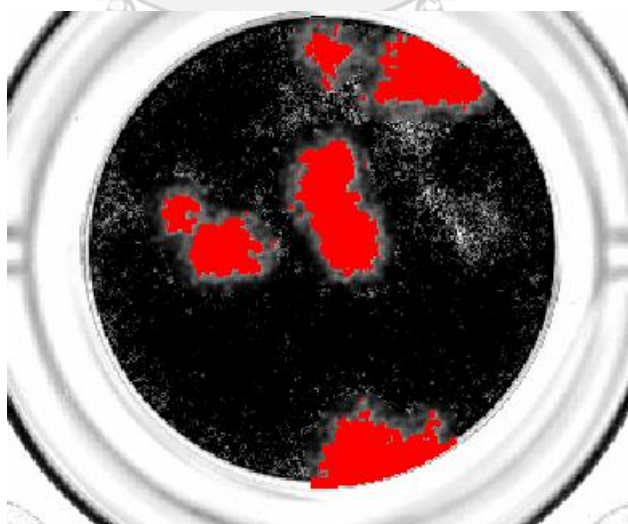
2.1.3.2.2 Binary Threshold

หลังจากทำการแบ่งวัตถุจากรูปภาพเป็นภาพขาวดำ (Binary Image) ซึ่งเป็นจะทำการแบ่งวัตถุได้ดีเมื่อรูปภาพเป็น Single channel โดย Binary Threshold จะเป็นการทำ Global Thresholding และใช้แบ่งแยก Region ได้ดี ซึ่งจะสามารถได้ผลดีกับรูปภาพที่มีความสว่างของพื้นหลังที่สูง โดยจะแบ่งออกเป็นสองวิธีคือ Max separability และ Smooth Histogram โดย Max separability จะสามารถเป็นสมการได้ดังนี้ [22]

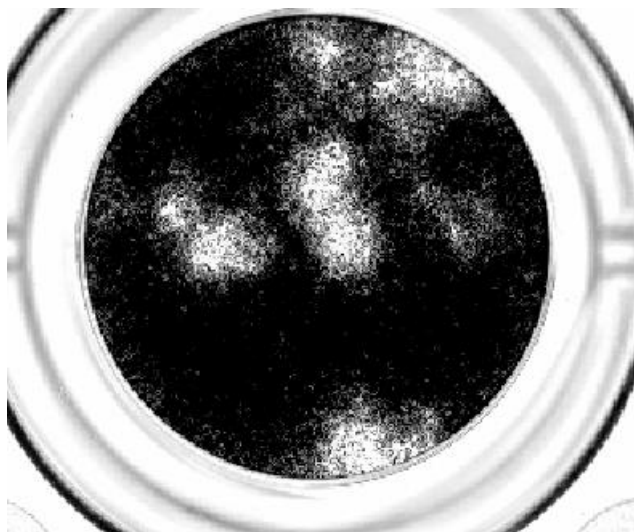
$$\begin{cases} \text{lightdark} ; \text{if } \text{light} \geq T \\ \text{lightdark} ; \text{if } \text{dark} < T \end{cases} \quad (2.17)$$

และ Smooth Histogram จะสามารถอธิบายได้ดังสมการ

$$\begin{cases} \text{lightDark} = \text{light} \geq T \\ \text{lightDark} = \text{dark} < T \end{cases} \quad (2.18)$$



รูปที่ 23 แสดงภาพที่ทำการจับวัตถุด้วยกระบวนการ Binary Threshold



รูปที่ 24 แสดงภาพที่ทำการจับวัตถุด้วยกระบวนการ Binary Threshold

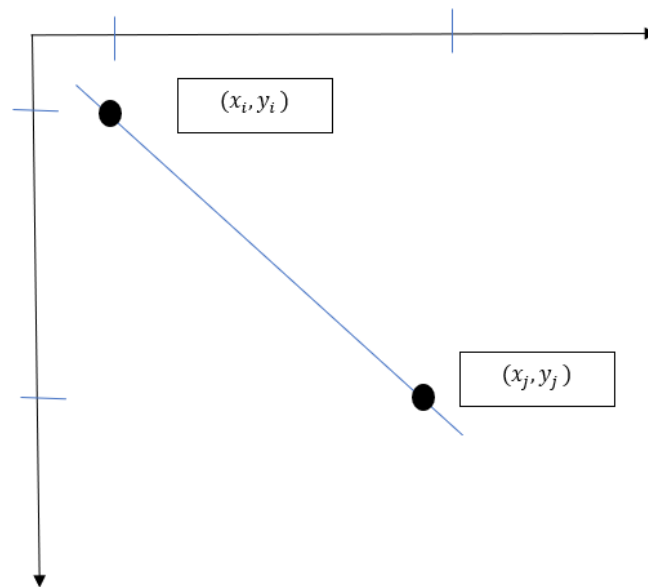
2.1.3.3 Hough Transform

การแปลงฮัฟนั้นเป็นการนำ Edge point มาเชื่อมต่อกันเพื่อหารูปร่างของวัตถุนั้นๆ โดยถ้าจุดๆนั้นวางอยู่บนเส้นโค้งเส้นเดียวกันก็จะสามารถแบ่งเป็นวัตถุนั้นได้ โดยจะกำหนดให้รูปภาพมีจำนวนจุดเท่ากับ n พิกเซลส์ ในการทำฮัฟนั้นจะต้องการหาซัพเซตของจุด n ว่าจุดๆนั้นเป็นขอบของวัตถุหรือเป็นเส้นตรงหรือไม่ โดยจะมีวิธีการดังนี้ [22]

การหาเส้นตรงและมุมบนเส้นตรงจากแปลงฮัฟ

หาเส้นตรงทุกเส้นที่มีในรูปภาพว่าเส้นตรงนั้นลากผ่านซัพเซตของจุด n หรือไม่ ถ้าเส้นตรงนั้นลากผ่านจุด จะนำจุดที่อยู่ใกล้เคียงกันมารวมกลุ่มเป็นคู่ โดยถ้าจุดสองจุดนั้นอยู่บนเส้นตรงเดียวกัน [1] ทั้งสองจุดจะตัดกัน บน Parameter Space หรือ Hough Space นั้นเอง จะสามารถอธิบายได้ เป็นสมการเส้นตรงใน parameter space ว่า

$$y_i = ax_i + b \quad (2.19)$$



รูปที่ 25 แสดงสมการเส้นตรงใน parameter space

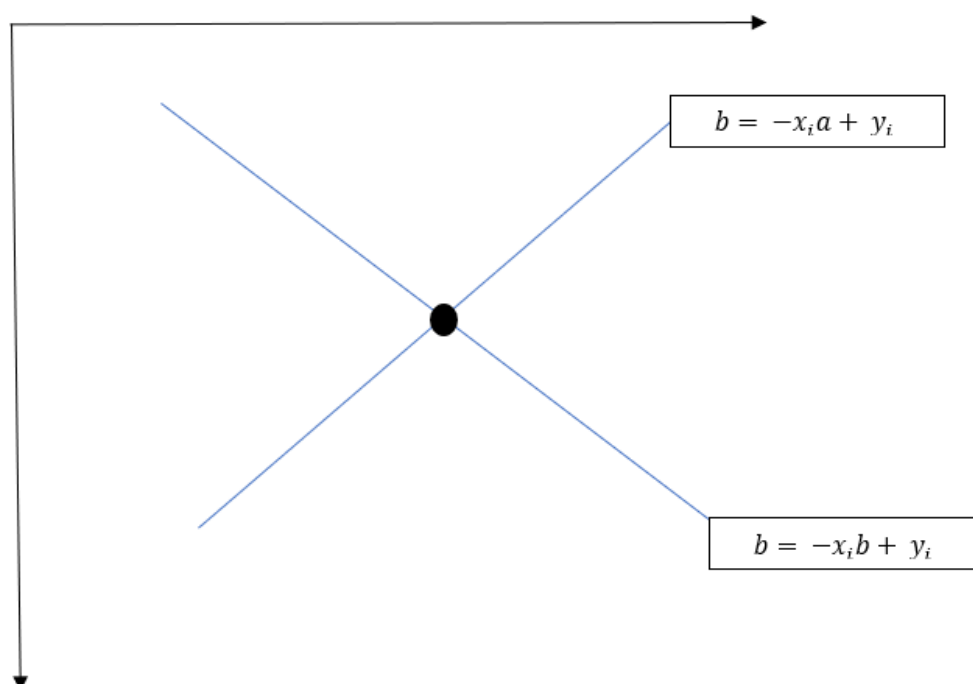
x_i และ y_i แทน จุดที่ตำแหน่ง x, y

a แทนความชัน b แทนจุดตัด

$b = -x_i + y_i$ สมการเส้นตรงบน Parameter Space

จากนั้นนำเส้นตรงจากสมการเส้นตรงบน Parameter Space มา plot โดยจะเก็บค่า a และ b สูงสุดต่ำสุดไว้ใน Accumulator space [22]

แทนที่จะเป็นจุดที่ต้องการหาแต่เป็นเส้นตรง (a เป็นอนันต์) ก็สามารถทำให้เป็นเป็นเส้นตรงบน Parameter Space ได้จากสมการ



รูปที่ 26 แสดงจุดตัดบนสมการเส้นตรงสองเส้นเมื่อ a เป็นอนันต์

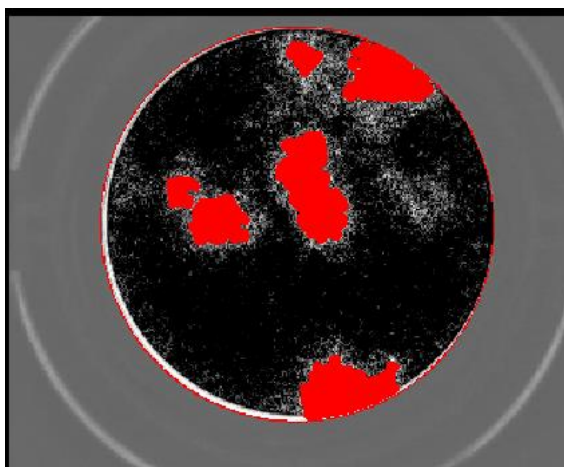
CHULALONGKORN UNIVERSITY

$$x \cos \theta + y \sin \theta = r$$

(2.20)

โดย r เป็น ระยะที่สั้นที่สุด

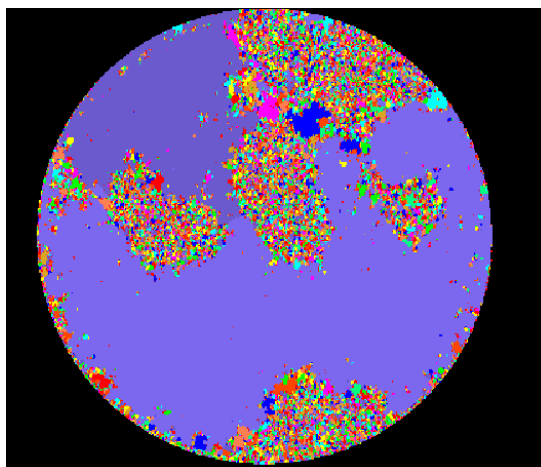
θ เป็น มุมระหว่าง r กับแกน x



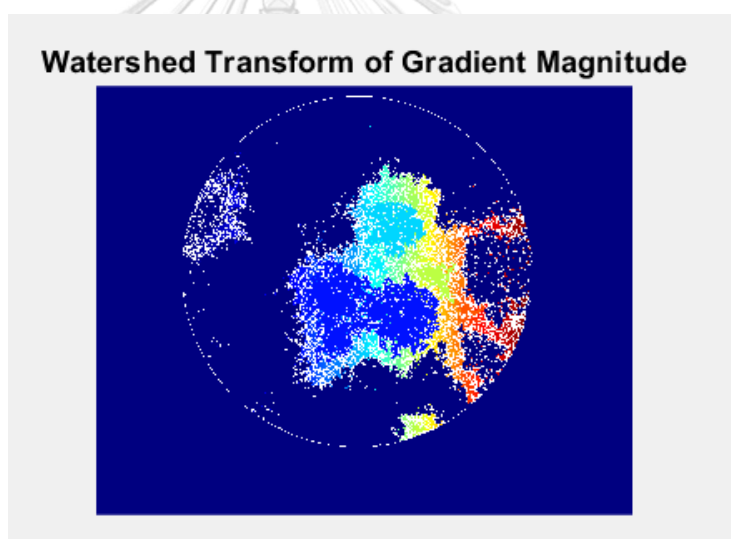
รูปที่ 27 แสดงภาพที่ทำการจับวัตถุด้วยกระบวนการ Hough Transform

2.1.3.4 Watershed method

Watershed method เป็นการแบ่งขอบเขตของวัตถุแบบการสร้าง Region เพิ่มขึ้นมา ซึ่งจะแตกต่างจากวิธีทั้งหมดซึ่ง จะใช้การสร้างจากลักษณะทางการภาพของจุดที่มีความคล้ายกัน คล้ายกับการสร้างเขื่อนเก็บน้ำ โดยจะทำการสร้างตัวเชื่อมก่อน (Watershed line) จากนั้นเติมน้ำลงไปในหลุม (Catchment basins) เมื่อเติมน้ำเต็มจะเห็นเป็นสันเขื่อน (Watershed line) แบ่งเป็นวัตถุๆไป [22]



รูปที่ 28 แสดงภาพที่ทำการจับวัตถุด้วยกระบวนการ Watershed method ผ่านโปรแกรม Mv tech
Halcon



รูปที่ 29 แสดงภาพที่ทำการจับวัตถุด้วยกระบวนการ Watershed method ผ่านโปรแกรม Mat-lab

2.2. Machine learning Operation

ในการแบ่งตัวพลาของไวรัสออกจากกันนั้น มีความท้าทายในการแยกพลาของไวรัสออกจากกันเมื่อพลาของไวรัสเกิดซ้อนทับกันหลายตัวจึงไม่สามารถจะทำการแบ่งวัตถุในรูปภาพด้วยวิธีการประมวลผลภาพแบบปกติได้ จึงเลือกใช้ หลักการทาง Machine learning มาทำการแบ่งพลาในส่วนนี้

โดย ทฤษฎี Machine learning ที่นำมาศึกษานั้นจะเป็นการทำนายข้อมูลแบบ Unsurprised learning โดยเป็นการแบ่งกลุ่มของคลาสเตอร์ที่มีวัตถุความคล้ายคลึงกัน เช่น คลาสเตอร์ของพลาของไวรัส และ มีความนิยมมากในการวิเคราะห์วัตถุที่สนใจจากรูปภาพชีวการแพทย์ [24,25]

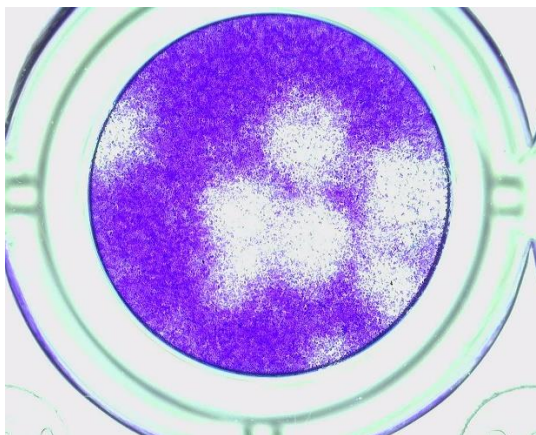
2.2.1 K-mean Clustering

Clustering เป็นการจัดกลุ่มคลาสเตอร์ซึ่งเป็นการรวมตัวของจุดพิกเซล ที่มีลักษณะใกล้เคียงกันนำมาอยู่รวมกัน ซึ่งในแต่ละจุดจะมีจำนวนพิกเซลแตกต่างกันไป ซึ่งจะใช้เป็นหลักเกณฑ์ในการแบ่งกลุ่ม คลาสเตอร์ K โดยอาศัยหลักการ Euclidean distance เป็นการเปรียบเทียบระหว่างจุดข้อมูล เพื่อหา Expectation-Maximization. กับจุดศูนย์กลางของคลาสเตอร์ โดยเมื่อ ระยะจากจุดข้อมูลนั้นอยู่ใกล้กับจุดศูนย์กลางคลาสเตอร์ใดก็จะนำไปจัดอยู่ในคลาสเตอร์นั้นๆ K-mean Clustering หรือในทาง Machine learning จะรู้จักในชื่อว่า K-mean Algorithm [26]

$$J = \sum_{i=1}^m \sum_{k=1}^K w_{ik} \|x^i - \mu_k\|^2 \quad (2.21)$$

โดยจะมีขั้นตอนดังนี้

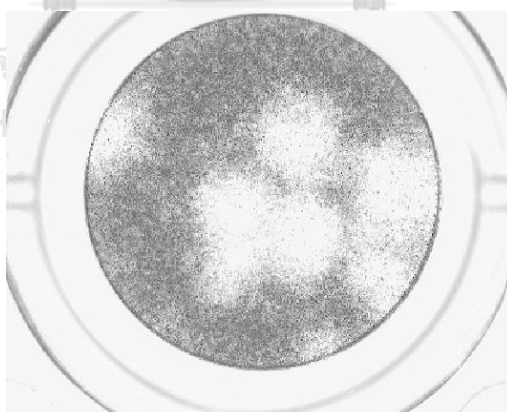
3.1.1 การเตรียมรูปภาพเพื่อจะนำไปใช้ใน กระบวนการ K-mean Clustering



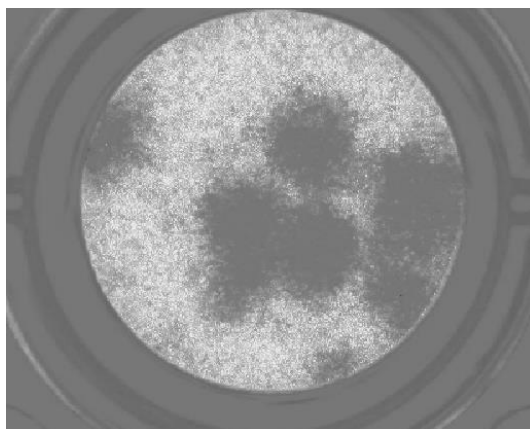
รูปที่ 30 แสดงภาพ ขั้นตอน 3.1.1

3.1.1.1. นำภาพที่ต้องการแบ่งคลัสเตอร์มาเปลี่ยนส่วนใหญ่จะเป็นภาพสี RGB มาแปลงเป็นภาพบน CIE L*a*b Colour space

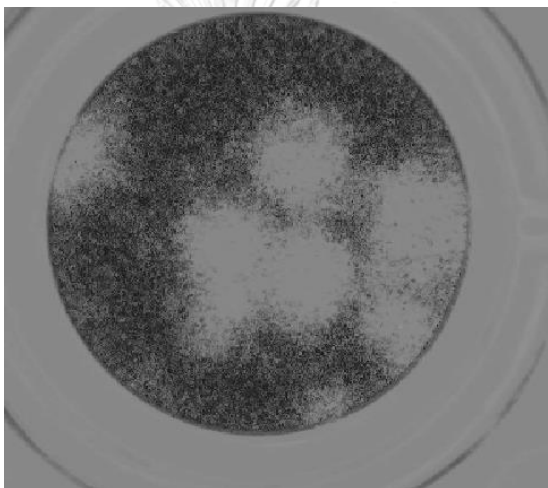
3.1.1.2. เมื่อได้ภาพที่เป็น CIE L*a*b Colour space จะใช้เฉพาะ a และ b



รูปที่ 31 แสดงภาพ L



รูปที่ 32 แสดงภาพ a



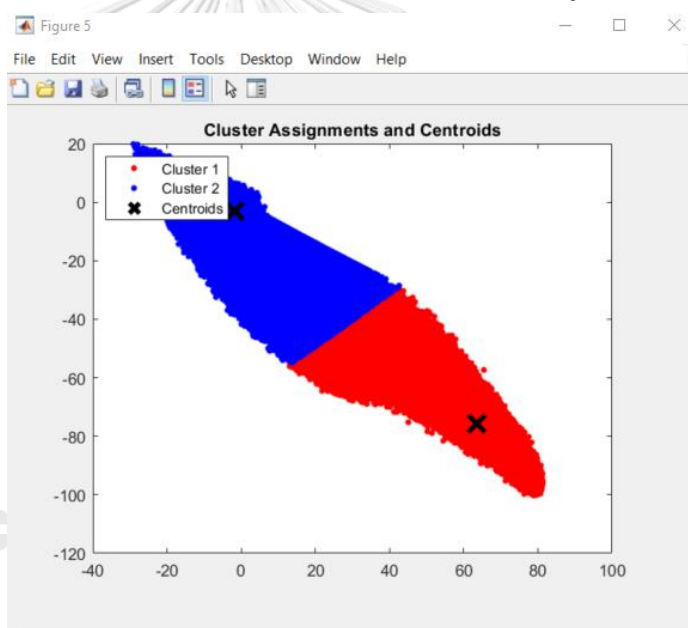
รูปที่ 33 แสดงภาพ b

3.1.2 ขั้นตอนกระบวนการ K mean Clustering

1. ระบุจำนวนคลัสเตอร์ที่ต้องการจะแบ่งกลุ่ม (ค่า K)
2. กำหนดจุด Data point (x_i) จำนวน Data point ในคลัสเตอร์ (w_{ik}) และกำหนด ศูนย์กลาง (μ_k) ในแต่ละคลัสเตอร์ โดยจะเริ่มจากการสุ่มตำแหน่งขึ้นมาในรูปภาพ

$$\mu_k = \frac{\sum_{i=1}^m w_{ik} x_i}{\sum_{i=1}^m w_{ik}} \quad (2.22)$$

3. คำนวณหาระยะทางจากแต่ละจุดไปยังแต่ละ จุดศูนย์กลาง



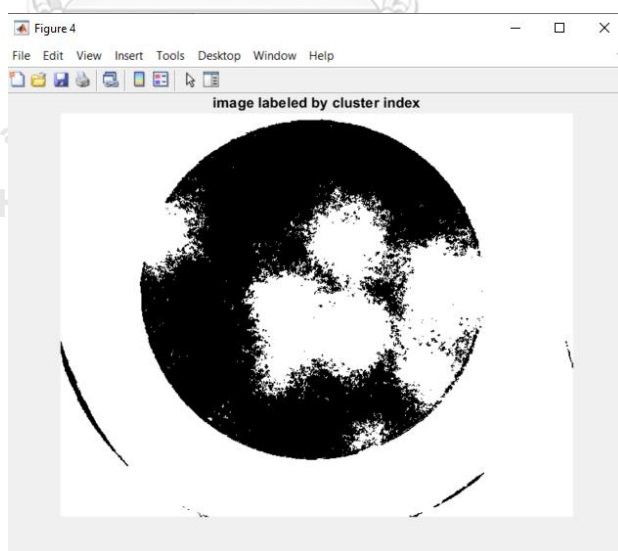
รูปที่ 34 แสดงภาพ ระยะทางจากแต่ละจุดข้อมูลไปยังแต่ละ จุดศูนย์กลาง

4. จัดกลุ่มแต่ละคลาสเตอร์บนพื้นฐานของ ระยะจากจุดของข้อมูลไปยังจุดศูนย์กลางนั้นๆ (Minimum Distance)
5. คำนวณจุดศูนย์กลางคลาสเตอร์ใหม่อีกครั้ง ว่าที่ทำการสุมขึ้นมาอยู่ในตำแหน่งที่ถูกต้องหรือไม่
6. คำนวณหาระยะทางจากแต่ละจุดไปยังแต่ละ จุดศูนย์กลาง(ใหม่) อีกครั้ง

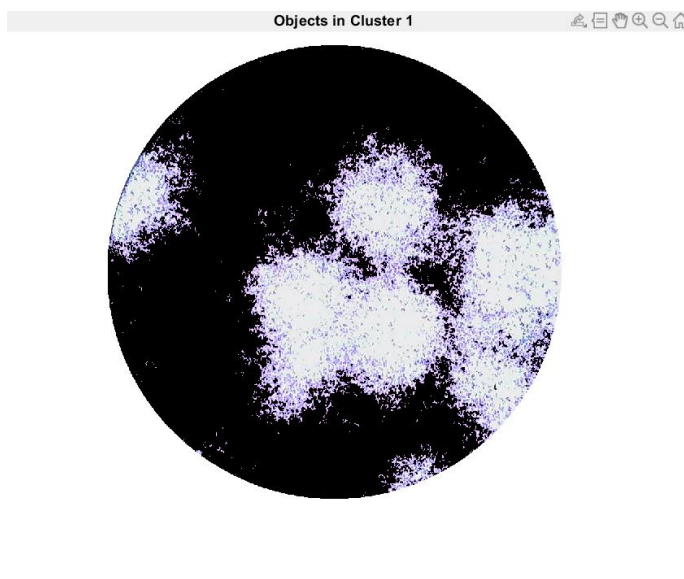
$$\frac{1}{m_k} \sum_{i=1}^{m_k} \|x^i - \mu_{ck}\|^2 \quad (2.23)$$

7. ทำขั้นตอนที่ 5 – 6 ซ้ำกระทั่งตำแหน่งของจุดศูนย์กลางไม่มีการเปลี่ยนแปลง

การทดลองนำ ภาพพลาของไวรัสผ่านการ K-mean Clustering จะแบ่งตัวพลาของไวรัสออกจากพื้นหลังได้อย่างชัดเจน



รูปที่ 35 แสดงภาพ จากค่าดัชนีพิกเซลในรูปภาพในรูปแบบ ของ 0-1



รูปที่ 36 แสดงวัตถุในรูปภาพของคลัสเตอร์ที่ 1 ซึ่งได้ทำการแยกตัวพลาของไวรัส จากพื้นหลัง



รูปที่ 37 แสดงวัตถุในรูปภาพของคลัสเตอร์ที่ 2 แยกตัวพื้นหลังออกจากพลาของไวรัส

2.2.1.1 Silhouette Method

เพื่อให้เป็นการแน่ใจว่าการแบ่งคลาสเตอร์ในกระบวนการ K-mean Clustering นั้นมีความถูกต้องแม่นยำมากยิ่งขึ้น Silhouette Method จึงเป็นอีกหนึ่งกระบวนการที่จะเป็นการตรวจสอบการแบ่งคลาสเตอร์นั้นๆ [27] โดย จะกำหนด

1. คำนวณระยะทางเฉลี่ยจากทุกจุดข้อมูลไปยังแต่ละจุดข้อมูล ในคลาสเตอร์ ในที่นี้กำหนดระยะทางเฉลี่ยเป็น a_i
2. คำนวณระยะทางเฉลี่ยจากทุกจุดข้อมูลไปยังแต่ละจุดข้อมูลของคลาสเตอร์ที่อยู่ใกล้เคียงกัน ในที่นี้กำหนดระยะทางเฉลี่ยเป็น b_i
3. นำค่า a_i และ b_i มาคำนวณหาค่า coefficient จากสมการ

$$\frac{b_i - a_i}{\max(a_i, b_i)} = c \quad (2.24)$$

ถ้าค่า c เท่ากับ 0 แสดงว่า แต่ละคลาสเตอร์อยู่ติดกันเกินไป

ถ้าค่า c เท่ากับ 1 แสดงว่า แต่ละคลาสเตอร์อยู่ห่างกันเกินไป

ถ้าค่า c เท่ากับ -1 แสดงว่า การจัดกลุ่มของคลาสเตอร์นั้นผิด

เพราะฉะนั้น ค่า c ควรมีค่าอยู่ระหว่าง 0 ถึง 1 แต่ควรจะให้ใกล้ 1 เพื่อจะได้คลาบบคลุ่มการแบ่งกลุ่มของ คลาสเตอร์ที่ถูกต้อง

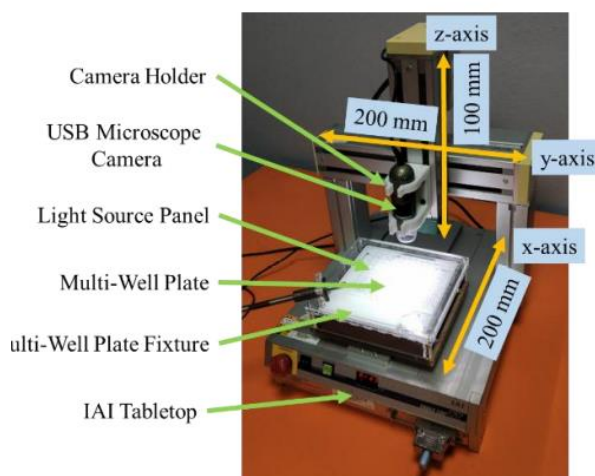
2.3. เครื่องถ่ายภาพแบบพลาออตโนเมตริก

เครื่องถ่ายภาพแบบพลาออตโนเมตริก ได้ถูกออกแบบขึ้นเพื่อทำการถ่ายภาพพลาออตโนเมตริกของไวรัสในแต่ หลุม 96-Well-Plate และ แสดงผลภาพ โดยงานวิจัยนี้เป็นงานวิจัยต่อเนื่อง โดยจะทำการใช้งาน และ เก็บภาพถ่ายพลาออตโนเมตริก จากเครื่องถ่ายภาพแบบพลาออตโนเมตริก ที่ได้มีการพัฒนามาจาก งานวิจัยก่อนหน้า [31] โดยในหัวข้อนี้จะเป็นการอธิบายอุปกรณ์ต่างๆ ของเครื่องและวิธีการใช้งาน

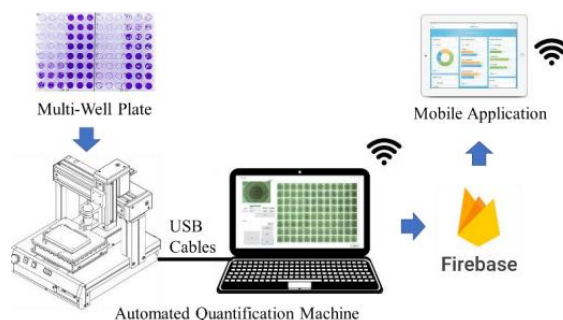
2.3.1 Hardware

เครื่องถ่ายภาพแบบพลาออตโนเมตริก นั้นถูกสร้างขึ้นเพื่อลดภาระการทำงาน ของทำการทดลอง ในห้องปฏิบัติการ ดังนั้น เครื่องถ่ายภาพแบบพลาออตโนเมตริก จะมีการติดตั้งที่ค่อนข้างง่าย วัสดุที่ใช้จะ ราคาไม่แพง ง่ายต่อการเคลื่อนย้าย และ ทำการถ่ายภาพจากแผ่น Well plate ได้ทั้ง ภาพเดี่ยว หรือ หลายๆภาพได้ในการสั่งงานหนึ่งครั้ง โดยส่วนประกอบของ Hard ware จะประกอบไปด้วย

1. IAI Tabletop Model TT-A3-I-2020-10B-SP. [20]
2. USB microscope camera
3. Dino-Lite AM4113T [21]
4. Adjustable light source panel
5. Illumination source LED surface ขนาด 150 mm x 150 mm



รูปที่ 38 แสดงเครื่องถ่ายภาพแบบพลาออตโนเมตริก



รูปที่ 39 แสดงเครื่องถ่ายภาพรูปแบบพลาตอัตโนมัติและการเชื่อมต่อ

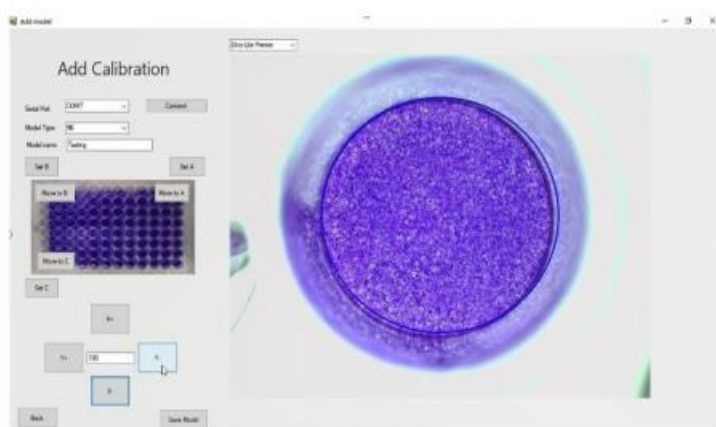
จากภาพ 2-38 จะแสดงเครื่องถ่ายภาพรูปแบบพลาตอัตโนมัติและการเชื่อมต่อ โดยการทำงาน จะเริ่มจากตัว IAI Tabletop จะทำงานโดย หุ่นยนต์ Cartesian ซึ่งจะสามารถเคลื่อนที่ได้ 3 แกน (แกน x-, y- และ z) โดยมีพื้นที่ในการทำงาน 200 มม. x 200 มม. x 100 มม. โดยค่าการระบุตำแหน่ง ด้วยทำซ้ำ จะมีความแม่นยำไม่เกิน ± 0.02 มม. โดยที่แกน x ของตัว IAI Tabletop จะติดตั้ง Multi-well plate fixture และ Light source panel โดยที่ แกน z จะติดตั้งที่ยึดกล้องถ่ายภาพและ กล้อง USB Microscope โดยเพื่อความแม่นยำของระบบ ตัวแกน z จะอยู่นิ่งเพื่อถ่ายภาพโดยและจะสามารถ ทำการขยับ ขึ้นลงได้ ก็ต่อเมื่อต้องการปรับระยะโฟกัสเท่านั้น โดย การทำงานของหุ่นยนต์ Cartesian IAI Tabletop จะทำการขยับตัว Multi-well plate ไปตามระนาบ x-y โดยการจัดตำแหน่งอัตโนมัติ และ Adjustable light source panel จะถูกติดตั้งบน แกน x ของหุ่นยนต์ และ การให้แสงกับระบบนั้น จะใช้เทคนิค Backlight technique [31] เพื่อเพิ่มความคมชัดให้กับรูปภาพ

2.3.1.1 ระบบการถ่ายภาพ

การถ่ายภาพเพื่การทดลองในแต่ละหลุมนั้น ระบบถ่ายภาพจะถูกติดตั้ง ณ แกน Z ของหุ่นยนต์ Cartesian โดย ตัวกล้องถ่ายภาพ เป็นกล้องชนิด USB Microscope camera ชนิด Dino-Lite AM4113T [21] เป็นกล้องถ่ายภาพสี พร้อมกับติดตั้งไฟ LED มาในตัว ซึ่งมีความละเอียด 1280×1024 พิกเซล และมีกำลังขยาย $10\times-50\times, 220\times$ โดย ณ กำลังขยายที่ต่ำกล้องจะสามารถถ่ายภาพขนาดเต็มของตัว Six-well-plate และ เมื่อกล้อง ณ กำลังขยายปานกลางจะสามารถถ่ายภาพขนาดเต็มของตัว 96-well-plate

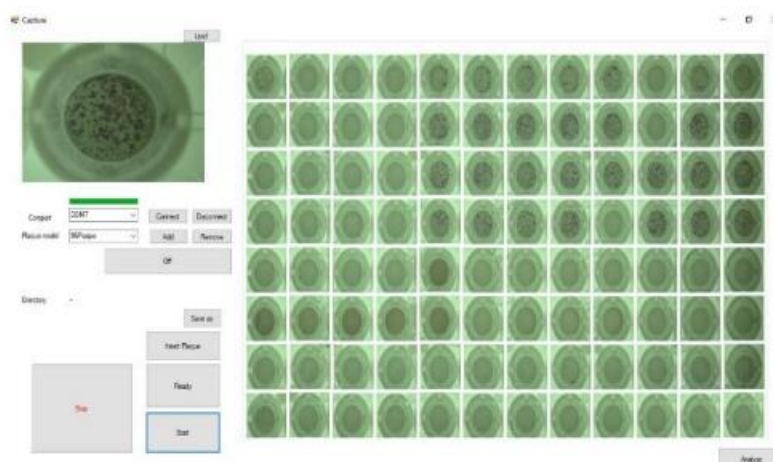
2.3.2 การใช้งาน เครื่องถ่ายภาพแบบพลาโคอัตโนมัติ

ในการควบคุม Hardware ต่างๆในเครื่อง เครื่องถ่ายภาพแบบพลาโคอัตโนมัติ จะทำการเชื่อมต่อตัวเครื่องกับตัวคอมพิวเตอร์ ผ่านสาย USB และควบคุมผ่านทาง Application ซึ่งได้ออกแบบตัว UI และ UX เพื่อให้งานต่อการใช้งาน โดยพัฒนาขึ้นโดยใช้ภาษา C# แสดงผ่านหน้าต่าง Application ตัว Application จะทำการสื่อสารกับ IAI Tabletop เพื่อทำการควบคุมการเคลื่อนที่และตำแหน่งของ multi-well plate และได้เชื่อมต่อกับกล้อง USB microscope camera โดยผ่าน DNVIT SDK เพื่อควบคุมแหล่งกำเนิดแสง [21] โดยรูปภาพจะถูกบันทึกเป็น รูปแบบ PNG format โดยหน้าต่าง Application นั้นจะประกอบไปด้วยฟังก์ชัน 1. Calibration routine 2. image acquisition 3. Firebase database [22]



รูปที่ 40 แสดง UI ของเครื่องนับพลาโคอัตโนมัติ

เนื่องจากการจะเริ่มใช้อุปกรณ์ในแต่ละครั้งมีความไม่แน่นอนเนื่องจากตำแหน่งของแกนอาจจะไม่ตรงกันจึงต้องทำการ calibrate ตัว IAI Tabletop โดยจะต้องทำการ calibration โดยมีขั้นตอนดังนี้ 1. ควบคุม IAI Tabletop โดยการควบคุม ผ่านตัว UI เพื่อค้นหาตำแหน่ง กึ่งกลางหลุม A1 A12 และ H1 2. กดปุ่ม calibration routine button จากนั้นตัวเครื่องจะทำการคำนวณหาพิกัดและแกนของ multi-well plate fixture โดยอัตโนมัติ



รูปที่ 41แสดง UI ขณะที่จะทำการ Run Program

ในฟังก์ชัน Image acquisition ซึ่งเป็นฟังก์ชันในการถ่ายภาพ โดยได้อนุญาตให้ผู้ใช้งานถ่ายภาพทั้งหมด หรือ สามารถเลือก จำนวนหลุมที่จะถ่าย หรือจะระบุเป็นหลุมก็ได้ เช่น A1, B2 และ H12 จากนั้นเครื่องก็จะขยับและทำการถ่ายรูปอัตโนมัติ

โดยประสิทธิภาพนั้นจะสามารถถ่ายภาพสำหรับ 96-well-plate เครื่องสามารถจับภาพและแสดงภาพทั้งหมดได้ภายในเวลาไม่ถึง 1 นาที เมื่อจับภาพได้แล้ว ซอฟต์แวร์ที่ใช้ Machine Learning สำหรับการนับผลของไวรัสในรูปภาพ แต่ระบบนั้นอาจจะมีความยากในการใช้งาน ยังมีแนวทางในการพัฒนาในอนาคต เช่น เช่น ทำเว็บแอปพลิเคชันให้ผู้ใช้งานมีความสะดวกสบายมากขึ้น

2.4 วรรณกรรมที่เกี่ยวข้องกับการนับด้วยกระบวนการวิเคราะห์ภาพ

การพัฒนาโปรแกรมการนับพลาของไวรัสเป็นการโครงการพัฒนาต่อเนื่อง ในเบื้องต้นจะเป็นการสืบค้นวรรณกรรมที่เกี่ยวข้อง โดยงานวิจัยที่นำมาจะเกี่ยวกับการศึกษาการประมวลผลภาพ และการแบ่งรูปภาพ ของเซลล์สิ่งมีชีวิตเป็นหลัก ซึ่งแต่ละรูปร่างจะมีลักษณะแตกต่างกันออกไป เพื่อนำองค์ความรู้ไปศึกษาพัฒนาโปรแกรมการนับไวรัสของพลาได้อย่างถูกต้อง ซึ่งมีรายละเอียดดังต่อไปนี้

[24] งานวิจัยของคุณ Fatima และคณะ ได้ทำการศึกษาเกี่ยวกับผลลัพธ์จากการนับเซลล์มะเร็งที่ตายแล้วโดยวิธี K-mean clustering แล้วเปรียบเทียบกับผลการนับจากผู้เชี่ยวชาญ โดยมีวิธีการดังนี้ (1) นำภาพถ่ายเข้ามา ในโปรแกรมในรูปแบบของ RGB ซึ่งในภาพจะประกอบไปด้วยเซลล์ที่ยังมีชีวิตและเซลล์ตายแล้ว (2) แยกเซลล์ทั้งคู่ออกจากพื้นหลัง โดยการใช้ทฤษฎี K-mean clustering แบ่งรูปภาพออกเป็นสี่คลัสเตอร์ ได้แก่ พื้นหลัง เซลล์ที่ยังมีชีวิต เซลล์ที่ตายแล้ว และ ขอบของเซลล์ที่มีชีวิต (3) กำจัดสิ่งรบกวนในภาพ Noise โดยใช้ Averaging filter และ 2D Median filter เพื่อกำจัด Salt and pepper noise จากนั้นใช้กระบวนการ Opening เพื่อจัดการสิ่งรบกวนที่เหลือ (4) ทำการนับเซลล์ จาก ทฤษฎี Hough transform เพื่อหาขนาดของเซลล์ พบว่า การนับเซลล์ที่มะเร็งที่ตายแล้วแยกออกจากเซลล์ที่มีชีวิต ให้ผลที่มีความเที่ยงตรง ได้มีการแนะนำว่าสามารถพัฒนาความเที่ยงตรงได้อีก เช่น ถ่ายภาพให้ชัดและชัดเจนยิ่งขึ้น หรือ แบ่งคลัสเตอร์ให้ละเอียดยิ่งขึ้น

[28]งานวิจัยของคุณ S.S.Savkare และคณะได้ทำการศึกษาการแบ่งเซลล์เม็ดเลือด ออกจากภาพไมโครสโคป โดยงานวิจัยนี้ต้องการนับเซลล์เม็ดเลือดขาว ออกจากภาพถ่าย ชีวการแพทย์จากเครื่องไมโครสโคป ซึ่งในภาพถ่ายชีวการแพทย์ จะประกอบไปด้วยส่วนประกอบหลายอย่าง โดยมีวิธีการดังนี้ (1) แยกส่วนประกอบของภาพถ่ายเป็น ภาพ เขียว แดง น้ำเงิน (2) ใช้ Median filter ขนาดสามคูณสาม เพื่อกำจัดสิ่งรบกวน Noise (3) เทียบภาพที่ได้กับภาพสี (4) ใช้ Laplacian filter (5) ให้ทำซ้ำขั้นตอนทั้งหมด พบว่าวิธีการขั้นตอนข้างต้นใช้สำหรับกำจัดสิ่งรบกวนและจะได้ผลดีกับภาพที่มีความคมชัดต่ำ หลังจากนั้นจึงใช้ทฤษฎี K-mean clustering โดยทำตามขั้นตอนดังนี้ (1) แปลงภาพเป็น L a b Space (2) โดยทำการแบ่งคลัสเตอร์เป็นสองคลัสเตอร์ (3) ตัวคลัสเตอร์ K จะรวมกลุ่มด้วยระยะทางที่ใกล้ที่สุด (4) (5) K-mean ระบุความละเอียดในแต่ละข้อมูล (6) เปลี่ยนรูปภาพมาใน HSV Space (7) แยกความเข้มสีใน Colour space (8) เปลี่ยน Saturation plane เป็น ภาพ Binary ด้วย Global thresholds (9) เพิ่มผลลัพธ์กลับไปยังขั้นตอนที่ 4 และ 7 หลังจากทำการแบ่งภาพโดยวิธีการ K-mean clustering พบว่ายังมีเซลล์ที่ซ้อนทับกันจึงไม่สามารถนับได้จึงมีการใช้ ทฤษฎี

Sobel edge detector และ Watershed transform เพื่อแบ่งเซลล์เม็ดเลือดที่ซ้อนทับกันจากนั้นจึงมาทำการนับเซลล์อีกที พบว่า การนับเซลล์นั้นได้ประสิทธิภาพถึง 95 เปอร์เซ็นต์แต่ผลจะไม่ดีเนื่องจากบางรูปเซลล์จะเป็นคราบ

[29] งานวิจัยคุณ Nancy M. Salem ได้ทำการศึกษาการแบ่ง Segmentation scheme for white blood cell ซึ่งเซลล์ที่นำมาวิเคราะห์คือเซลล์เม็ดเลือดขาวโดยจะทำการแบ่งออกจากกันด้วยกระบวนการ K-mean clustering โดยรูปภาพถ่ายของเซลล์เม็ดเลือดนั้นจะเป็นรูปภาพสี RGB และนำมาแปลงเป็นรูปภาพบน L^*a^*b Colour space จากนั้นนำรูปเฉพาะ a และ b ไปวิเคราะห์ต่อ เพื่อนำไปแยกเซลล์เม็ดเลือดขาวด้วยกระบวนการ K-mean clustering โดยรูปภาพเม็ดเลือดขาวเหล่านี้จะได้มาจาก Data set ที่มีอยู่แล้ว ผลการวิจัยนั้นพบว่า การตรวจจับและแยกเซลล์เม็ดเลือดขาวออกจากภาพถ่ายเพื่อนำมาวิเคราะห์ข้อมูลด้วยกระบวนการ K-mean clustering นั้นสามารถทำได้จริง

[30] งานวิจัยคุณ Benjamin X.Guan ได้ทำการศึกษากระบวนการตรวจจับหาสเต็มเซลล์ของมนุษย์และขอบเขตของเซลล์โดยจุดประสงค์หลักของงานวิจัยนี้จะใช้กระบวนการ K-mean clustering พร้อมกับวิธีการ Weighted Entropy โดยจะเริ่มจากการนำภาพที่มีความคมชัดและมีเซลล์อยู่เพื่อนำมาวิเคราะห์ โดยรูปภาพสเต็มเซลล์ส่วนใหญ่ นั้น จะประกอบไปด้วยความหนาแน่นของพอกเซลล์ที่สูง จะประกอบด้วยค่า Entropy ที่สูงเช่นกัน โดยค่า Entropy เหล่านี้จะมีประโยชน์ในการหาสเต็มเซลล์ โดยกระบวนการจะเริ่มจากการ Weighted Entropy ให้ค่ามีค่าเฉลี่ยยิ่งขึ้นจากนั้นนำค่ามา Optimization ผลลัพธ์ของการศึกษาจะพบว่ากระบวนการ Weighted Entropy สามารถเพิ่มประสิทธิภาพกระบวนการ K-mean clustering ได้จริง และเพิ่มความแม่นยำได้มากขึ้น

2.6 เซลล์และไวรัสสำหรับกระบวนการนับพลาควาของไวรัส

กระบวนการนับพลาควาของไวรัส (Plaque Assays) เพื่อที่จะนำมาวิเคราะห์ข้อมูลในงานวิจัยนี้ ได้ทำการสร้างตัวอย่าง โดย ศูนย์วิจัยไวรัสวิทยา จุฬาลงกรณ์มหาวิทยาลัย โดยอ้างอิง ส่วนผสมของ สารละลายต่างๆ ชนิดของไวรัส และ ขั้นตอนการทำ Plaque Assays อ้างอิงจากงานวิจัยของคุณ S. Boonyasuppayakorn และ คณะ เรื่อง Simplified dengue virus microwell plaque assay using an automated quantification program [7]. โดยจะใช้ความเข้มข้นของสารละลาย Medium ตั้งแต่ 10^{-1} ไปจนถึง 10^{-6} รวมทั้งหมด $50 \mu\text{L}$ จะทำการผสมกับ เซลล์ของไวรัสชนิด LLC/MK2. (ATCC®CCL-7) จำนวน 1×10^4 cells ต่อหลุม ($50 \mu\text{L}$) หลังจากนั้นจะพักไว้ให้เซลล์ไวรัสเติบโตเป็นเวลา 5-10 วัน จนกระทั่ง มองเห็นได้ชัดเจนภายใต้กล้องจุลทรรศน์ เพื่อนำไป วิเคราะห์จำนวนพลาควาต่อไป โดยจะประกอบไปด้วยเพลทการทดลองชนิด 96-well plate จำนวน 10 ชุด สามารถให้ตัวอย่างได้ จำนวน 606 ตัวอย่าง ซึ่งจากงานวิจัยข้างต้น [7] ได้กล่าวถึงขนาดพลาควาของ ไวรัสที่เล็กที่สุดเป็นหนึ่งคลาสเตอร์ อยู่ที่ 1-2 มิลลิเมตรต่อหนึ่งคลาสเตอร์นั่นเอง

บทที่ 3

วิธีดำเนินการพัฒนาและออกแบบโปรแกรมระเบียบวิธีการนับพลาควของไวรัส

ในงานวิจัยนี้เป็นการต่อยอดและพัฒนา โปรแกรม ต่อเนื่องจากงานวิจัย การพัฒนาอุปกรณ์ถ่ายภาพอัตโนมัติและระเบียบวิธีการนับพลาควของไวรัส จากแนวทางการศึกษาที่พัฒนาไปนั้น จึงได้ทำการพัฒนาต่อ โดยได้ทำการพัฒนา Algorithm ของโปรแกรม และ สร้าง Graphic User interface เพื่อความสะดวกต่อการใช้ผู้ใช้งานในการใช้งานโปรแกรม โดยได้ทำการเพิ่มฟังก์ชันการใช้งาน โดยฟังก์ชันการใช้งานพื้นฐานเหล่านี้นำรูปแบบมาจาก เครื่องตรวจจับและนับเซลล์ในระดับอุตสาหกรรม Cellular Technology Limited developed a commercial Elispot and viral plaque-counting machine, ImmunoSpot CLT Analyzers [9] โดยนำหลักการบางส่วนมาปรับใช้ในการพัฒนา และได้ทำการปรับโครงสร้างของโปรแกรมมาขึ้นใหม่เพื่อจุดประสงค์ที่ ต้องการให้ระบบระเบียบวิธีการการนับพลาควของไวรัสอัตโนมัติมีความแม่นยำมากยิ่งขึ้นและ มีความทันสมัย ให้การใช้นั้นมีความเป็นมิตรต่อผู้ใช้งานมากยิ่งขึ้น โดยจะแบ่งเป็นสองส่วนในการพัฒนา คือ 1. ตัวโปรแกรมหลักซึ่งจะใช้พัฒนา Algorithm Coding และ Graphic User Interface สำหรับผู้ใช้งาน 2. เครื่องถ่ายภาพ

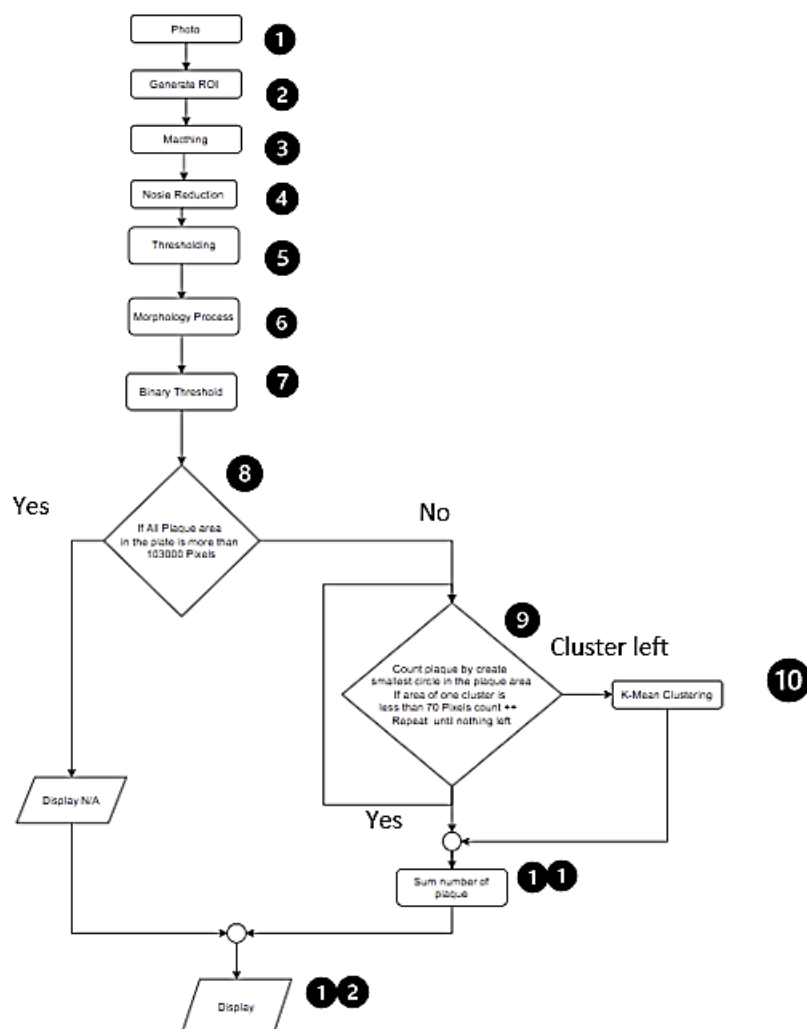
3.1 การออกแบบโปรแกรมหลักเพื่อวิเคราะห์ภาพเพื่อ นับจำนวนพลาควของไวรัส

นำภาพจากการถ่ายภาพบน 96 – well plate เป็นภาพเริ่มต้น จากการศึกษาข้อมูลเบื้องต้น ได้ตัดสินใจเลือกใช้กระบวนการขั้นตอนในการ ตรวจหาพลาควของไวรัสจากภาพ และวิเคราะห์ข้อมูลในภาพ เพื่อจะนำไปนับจำนวนพลาควของไวรัสต่อไป โดย การพัฒนานี้จะพัฒนามโปรแกรม MVTEC HALCON ซึ่งเป็นโปรแกรมเป็นซอฟต์แวร์การประมวลผลภาพและวิเคราะห์ภาพที่มีความสามารถในการสร้างและพัฒนาโปรแกรมแอปพลิเคชันเชิงวิชาการและอุตสาหกรรมสำหรับการวิเคราะห์ข้อมูลได้หลายประเภท เช่น Deep learning Machine Learning และ Image Processing นั้นเอง โดยจะใช้คำสั่งฟังก์ชันพื้นฐานเกี่ยวกับ Image Processing เป็นหลักเพื่อวิเคราะห์ภาพถ่ายพลาควของไวรัส

จากการทำการศึกษาพบว่ายังไม่มีการสร้าง Algorithm โปรแกรมนับพลาควของไวรัสอัตโนมัติ ด้วยกระบวนการ Machine Learning ผสมกับ กระบวนการ Image Processing ในการสร้าง กระบวนการนับการนับพลาควของไวรัสอัตโนมัติ โดยได้ทำการนับพลาควของไวรัสด้วย 2 กระบวนการ จากนั้นนำผลมารวมกัน ในลักษณะ Hybrid โดยการพัฒนาโปรแกรมนั้นได้ทำการพัฒนา Algorithm ขึ้นมาจำนวนสองรูปแบบ (2 versions) ซึ่งจะแตกต่างกันเรื่องลำดับขั้นตอนการจัดการข้อมูล ดังนี้

3.1.1 การพัฒนาโปรแกรม Algorithm ใน แบบที่ 1

แบบที่1 จะมีขั้นตอนการทำงานดังนี้



รูปที่ 42 ภาพแสดง ขั้นตอนในการทำงานกระบวนการนับพลาควของไวรัส Algorithm แบบที่1

กระบวนการนับพลาไคในแบบที่ 1 นั้น จะมีลำดับขั้นตอนการทำงานดังนี้ เพื่อให้ง่ายต่อการอธิบาย จึงมีเลขกำกับบอกขั้นตอนไว้ ดังรูปที่ 42

ขั้นตอนที่ 1 จะเป็นการนำรูปถ่ายพลาไคของไวรัสในแต่ละหลุม เข้ามายังโปรแกรม MVTec HALCON โดยรูปภาพนั้นจะมารวมเป็นไฟล์

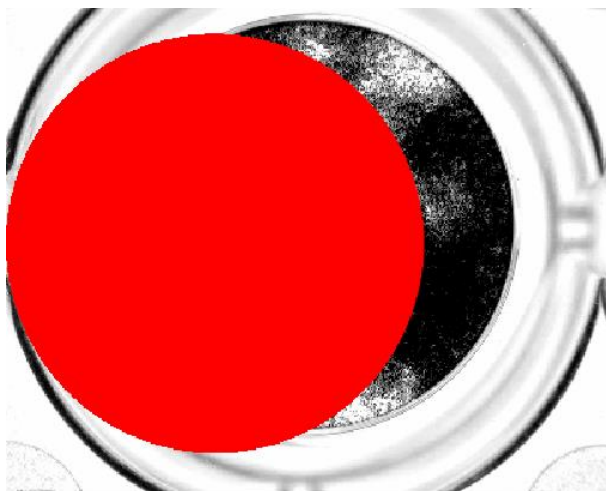
tif|tiff|gif|bmp|jpg|jpeg|jp2|png|pcx|pgm|ppm|pbm|xwd|ima|hobj

และกำหนดรูปภาพ มาสเตอร์ที่จะมาเป็นต้นแบบในการหาพื้นที่ของหลุมเพื่อกำหนดขอบเขตในการวิเคราะห์ข้อมูล

```
list_files ('C:/Users/Supha/OneDrive/เดสก์ท็อป/Plaque Update/D1-c5-c6', ['files','follow_links'], ImageFiles)
tuple_regexp_select (ImageFiles, ['\\.(tif|tiff|gif|bmp|jpg|jpeg|jp2|png|pcx|pgm|ppm|pbm|xwd|ima|hobj)$','ignore
n := []
```

รูปที่ 43 ภาพแสดง ชนิดของรูปภาพที่สามารถนำเข้ามาในโปรแกรม

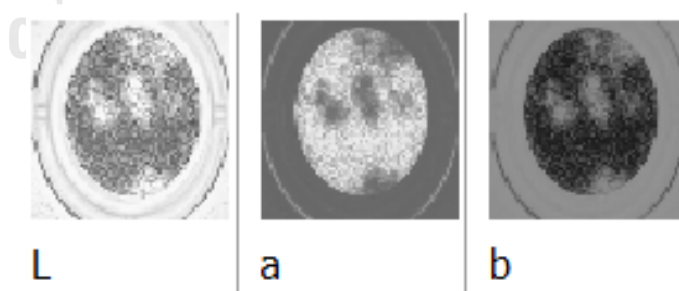
ขั้นตอนที่ 2 จะเป็นการจัดเตรียมรูปภาพ โดยเริ่มจากการจัดเตรียม Region of Interest ในรูปภาพเพื่อ ใช้ในการ Matching โดยจะเป็นการทำให้ตำแหน่งในการวิเคราะห์ภาพ นั้นอยู่ในตำแหน่ง เดียวกันทั้ง 96 รูป โดยจะมีพื้นฐาน จากการหาขอบของ 96 – well Plate หรือ ขอบของหลุมทดลอง และเทียบกับรูปมาสเตอร์ที่ได้กำหนดไว้ในขั้นตอนที่ 1 จากนั้น เป็นการ แปลง รูปภาพให้อยู่บน Color Space ที่ต้องการ



รูปที่ 44 ภาพแสดง ขั้นตอนที่ 2 การจัดเตรียมรูปภาพ

2.1 ทำการสร้างวงกลมขนาดใกล้เคียงกับตัวหลุม 96 – well Plate เพื่อ เป็นพื้นฐานในการ ทำ Matching จากนั้นหาพื้นที่ของวงกลมเพื่อ กำหนดขอบเขตที่จะวิเคราะห์

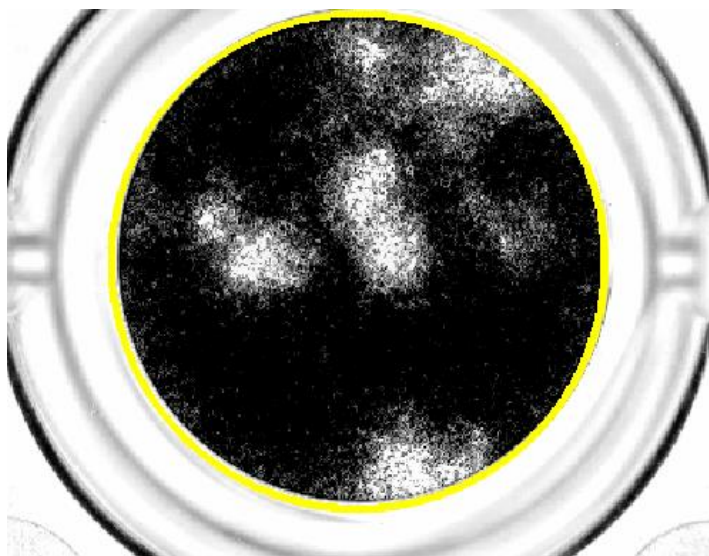
2.2 ทำการแบ่งสีในรูปภาพเพื่อจะให้ง่ายต่อการวิเคราะห์โดยแยกสีในรูปภาพเป็นสามรูปภาพ พื้นฐานบนสีแต่ละสี เป็นภาพ R G B ซึ่งจะประกอบด้วยสีแดง เขียวและฟ้าตามลำดับ จากนั้นเปลี่ยน จากภาพ R G B เป็นภาพ Lab ซึ่งจะประกอบเป็นสามภาพ L a b ซึ่งจะประกอบไปด้วย ภาพที่มี พื้นฐานจากความสว่าง L ภาพที่มีค่าพื้นฐานของสีจากค่าสีเขียวถึงสีแดง a และ ภาพที่มีพื้นฐานของสี จากค่าสีฟ้าถึงค่าสีเหลือง b



รูปที่ 45 ภาพแสดง สาม ภาพ L a b

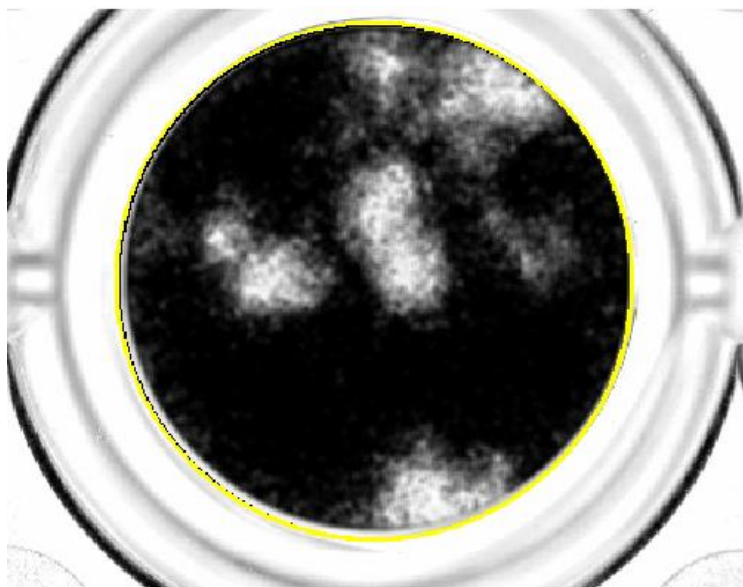
2.3 นำภาพมาลบกัน โดย ตั้งด้วย ภาพ G ลบ ด้วยภาพ a เพื่อจะได้ภาพที่มีเงดสี ที่มีความง่ายต่อการวิเคราะห์นั่นเอง

ขั้นตอนที่ 3 นำภาพที่เตรียมไว้มาทำการ Matching เพื่อจะได้พื้นที่ที่จะวิเคราะห์เหมือนกันในทุกภาพ



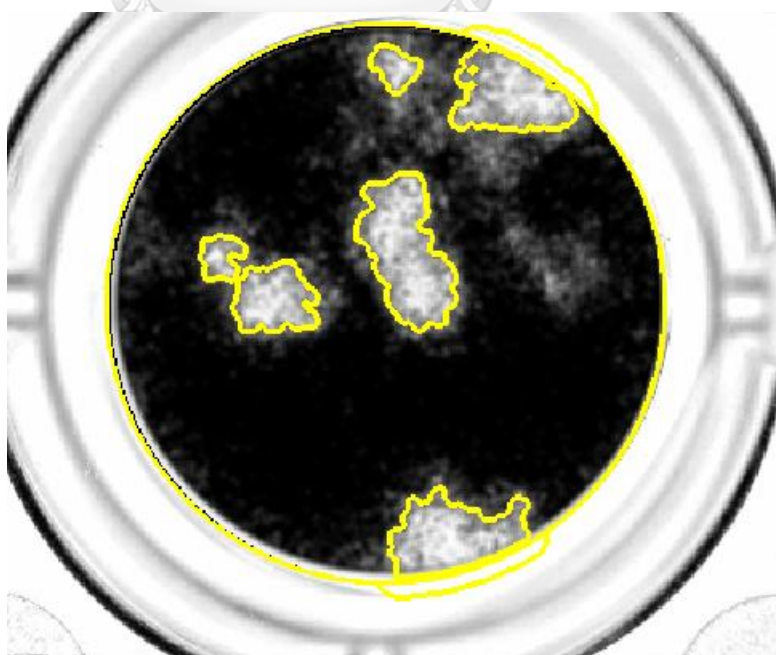
รูปที่ 46 ภาพแสดง ภาพที่นำมาทำการ Matching

ขั้นตอนที่ 4 หลังจากได้ทำการกำหนดขอบเขตที่จะวิเคราะห์และจึงเข้าสู่กระบวนการ ลดสิ่งรบกวนโดยการผ่าน Mean Filter ขนาด 10 คูณ 10



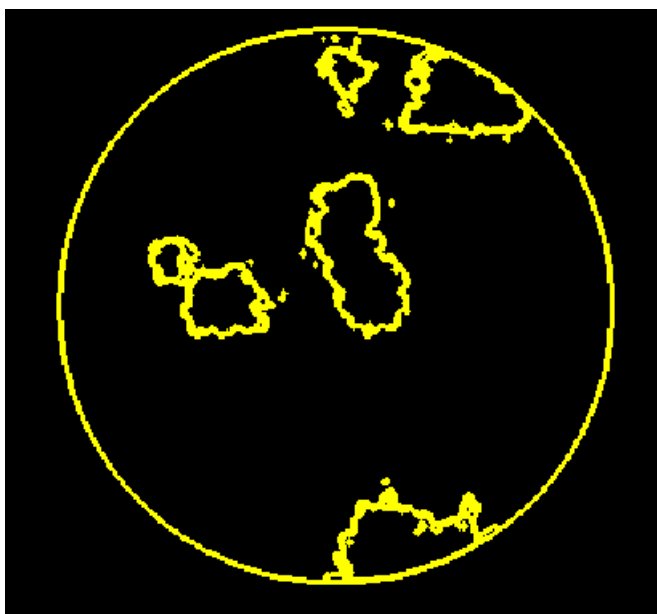
รูปที่ 47 ภาพแสดง ภาพที่นำมาผ่าน กระบวนการ ลดสิ่งรบกวนโดยการผ่าน Mean Filter ขนาด 10 คูณ 10

ขั้นตอนที่ 5 เข้าสู่กระบวนการ Image Segmentation รอบที่ 1 จะเป็นการใช้ ทฤษฎี Image Thresholding เพื่อทำการแยก ตัวพลาของไวรัสออกจากพื้นหลัง และกำหนดว่านี่คือพลาของไวรัส



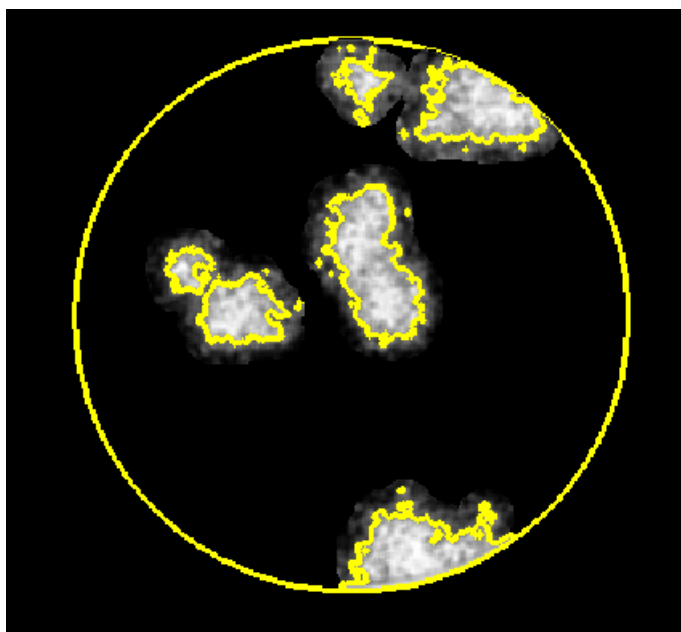
รูปที่ 48 ภาพแสดงภาพที่นำมาเข้าสู่กระบวนการ Image Thresholding

ขั้นตอนที่ 6 ทำการแยกพลาซที่อยู่ติดกันและกำจัดสิ่งรบกวนภาพในตัวพลาซของไวรัสด้วยกระบวนการ Morphology ซึ่งกล่าวไว้อย่างละเอียดแล้วในบทที่2 โดยจะมีกระบวนการ Erosion Dilation Opening Closing ตามลำดับ



รูปที่ 49 ภาพแสดง ภาพที่นำมาเข้าสู่กระบวนการ Morphology

ขั้นตอนที่ 7 ทำ Image Segmentation อีก หนึ่งครั้งโดยใช้ Binary Threshold โดยกำหนด Max Separately Light เพื่อกำหนดให้สีขาวซึ่งเป็นพลาของไวรัสเป็นพื้นที่จะนำไปวิเคราะห์ต่อไป

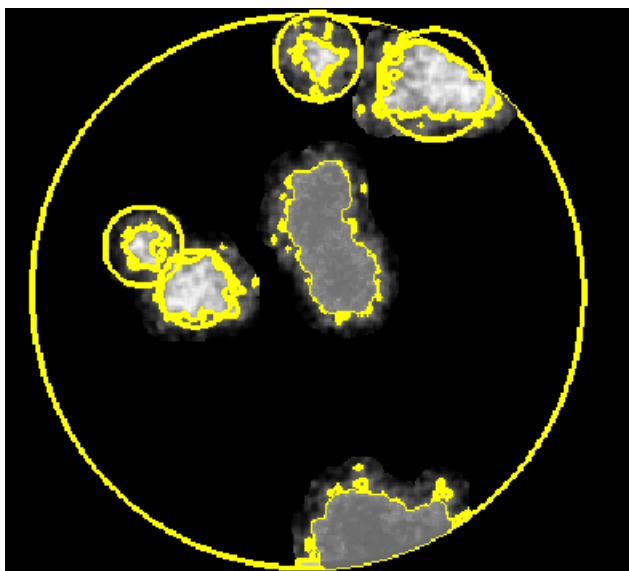


รูปที่ 50 ภาพแสดงภาพที่นำมาเข้าสู่กระบวนการ Binary Threshold

ขั้นตอนที่ 8 ทำการหาพื้นที่ของพลาในรูปว่ามีขนาดกี่พิกเซล โดยถ้าพลาของไวรัสนั้นมีพื้นที่มากกว่า 103000 พิกเซลนั้น โปรแกรมจะกำหนดว่านับไม่ได้และแสดงผล N/A แต่

ถ้าจำนวนพิกเซลในแต่ละคลาสเตอร์ของพลาคนั้นมีค่าน้อยกว่า 10300 จะนำไปทำการนับต่อไป

ขั้นตอนที่ 9 จะเป็นการนับพลาของไวรัสที่ทำการ Segmentation จาก Binary Threshold โดยจะเห็นว่าพลาของไวรัสนั้นได้ทำการแบ่งไว้แล้ว โดยจะทำการนับจากพื้นที่ของคลาสเตอร์นั้นๆ โดยจะทำการหาพื้นที่และสร้างวงกลมโดยถ้า วงกลมนั้นมีรัศมีเล็กกว่า 70 พิกเซลจะได้เป็นหนึ่งในคลาสเตอร์ ทำจนครบจำนวนที่เราทำการ Segment ไว้



รูปที่ 51 ภาพแสดงภาพจะเป็นการนับพลาของไวรัส จากกระบวนการ Binary Threshold

จากภาพ 51 จะเห็นว่าคลาสเตอร์ของพลาจะยังคงเหลืออยู่และยังไม่ได้นับ เป็นเพราะว่าคลาสเตอร์เหล่านี้ มีการซ้อนทับกันมากกว่าหนึ่งคลาสเตอร์จึงไม่สามารถทำการ Segmentation ด้วยวิธี Binary Threshold โดยจะมีการ Segmentation ต่อ โดยนำพื้นที่ ที่ยังไม่ได้ทำการนับไปแบ่งเป็นพื้นที่ย่อย ต่อใน กระบวนการ K mean clustering แต่ถ้านับครบแล้วไม่มีพื้นที่ที่เกินกำหนด ก็จะเก็บผลและข้ามไปขั้นตอน 11 ทันที

ขั้นตอนที่ 10 K mean segmentation จะเป็นกระบวนการที่จะทำการแบ่งคลาสเตอร์ของพลาที่ซ้อนทับกัน

10.1 คำนวณหาพื้นที่หาขนาดของพื้นที่เพื่อที่จะทำ K mean segmentation

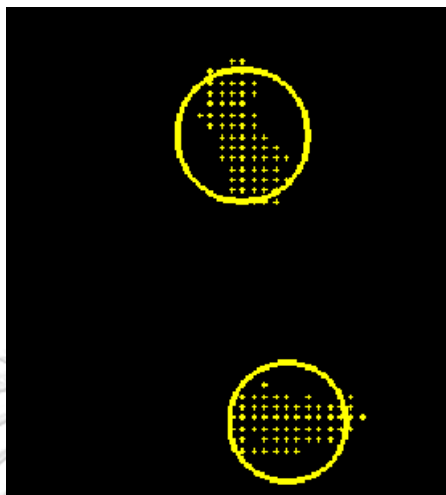
10.2 จากนั้นทำ K mean ครั้งแรกเพื่อ ทำการแยกพลาออกจากพื้นหลังและเพื่อกำหนดพื้นที่ที่จะทำการแบ่งคลาสเตอร์ ให้ $K = 2$ โดยจัดกลุ่มข้อมูลจากระยะทาง

10.3 ทำการสร้าง Grid ขึ้นมาเพื่อเป็นตัวช่วยในการแบ่งคลาสเตอร์ โดยในการ Kmean ครั้งแรกจะยังหา จุดศูนย์กลางได้ไม่มีแม่นยำจึงให้จุด Grid เป็นเหมือนพื้นที่ในการจัดกลุ่มคลาสเตอร์

10.4 เข้ากระบวนการ K mean clustering อีกครั้ง โดย คราวนี้จะทำการนับพลาของไวรัสที่มีการซ้อนทับกันทำการ K mean clustering โดยจะทำการสุ่มจุดศูนย์กลางคลาสเตอร์ขึ้นมาก่อน จากนั้นจึงทำการหาจุดศูนย์กลางใหม่ ทำซ้ำไปเรื่อยๆจนจุดศูนย์กลางของคลาสเตอร์ ไม่เปลี่ยนแปลง

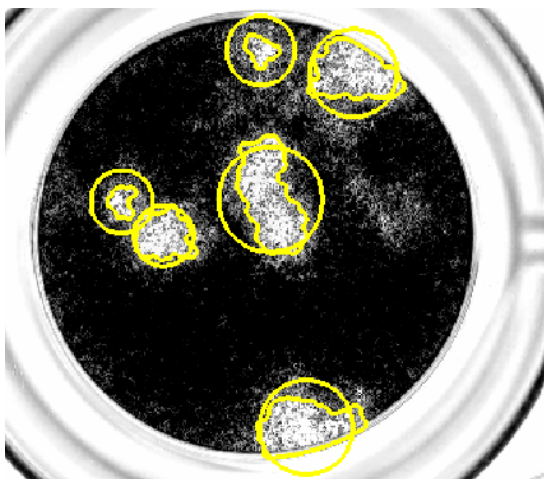
จนครอบคลุมพื้นที่พลาคนั้นๆ โดยจะกำหนดให้พื้นที่ของคลาสเตอร์นั้นๆมีรัศมีค่าเท่ากับ 110 พิกเซล

10.4 จากนั้นทำการเก็บค่าจำนวนคลาสเตอร์



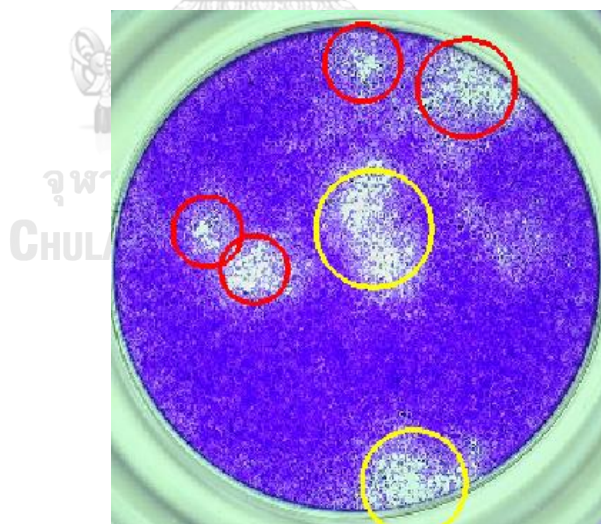
รูปที่ 52 ภาพแสดงภาพจะเป็นการนับพลาคของไวรัส จากกระบวนการ K mean clustering

ขั้นตอนที่ 11 รวมจำนวนคลัสเตอร์ของพลาแคในรูปภาพจากการ binary threshold และ K mean clustering มารวมกัน



รูปที่ 53 แสดงภาพรวมจำนวนคลัสเตอร์ของพลาแค

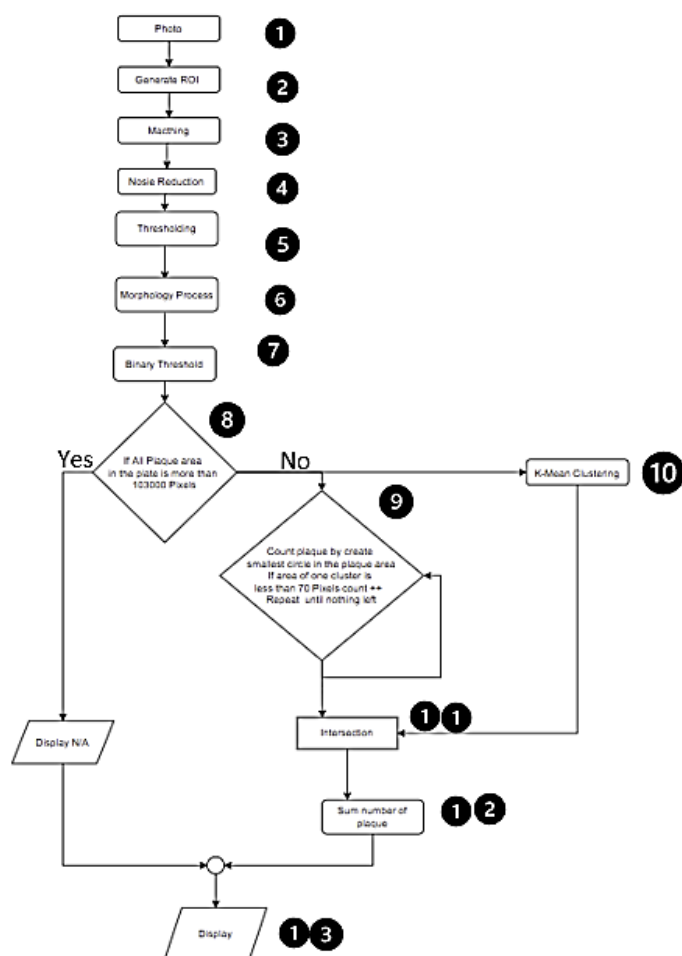
ขั้นตอนที่ 12 แสดงผล



รูปที่ 54 แสดงผลการนับพลาแคของไวรัส

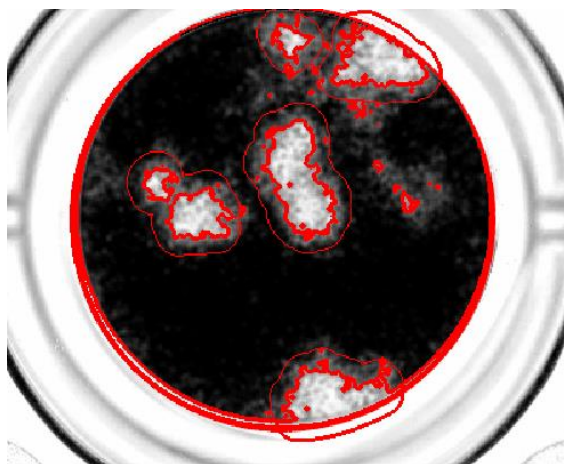
3.1.2 การพัฒนาโปรแกรมหรือ Algorithm ใน แบบที่2

แบบที่ 2 ซึ่งจะมีการทำงานคล้ายแบบที่ 1 ในขั้นตอนที่ 1 ถึง ขั้นตอนที่ 8 ซึ่งจะเป็นการเตรียมรูปภาพก่อนจะนำไปทำการวิเคราะห์หาคลาสเตอร์พลาของไวรัส จึงจะขอก้าวถึงขั้นตอนที่ 9 เป็นต้นไป จะมีการทำงานดังนี้



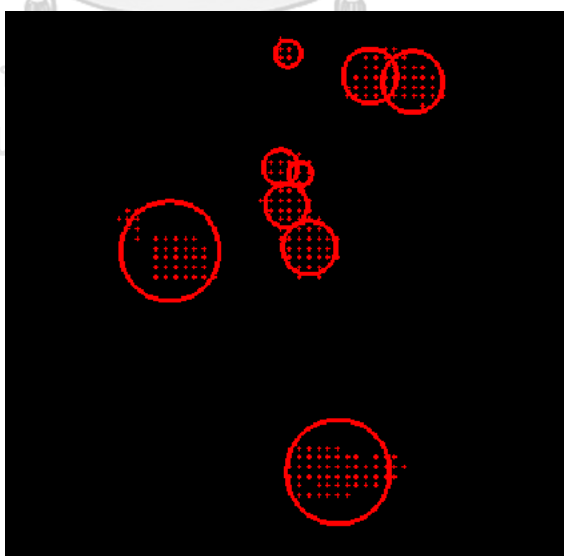
รูปที่ 55 ภาพแสดง ขั้นตอนในการทำงานกระบวนการนับพลาของไวรัส Algorithm แบบที่ 2

ขั้นตอนที่ 9 หลักจากที่ได้ทำการแยกตัวพลาสมของไวรัสออกจากพื้นหลังได้แล้ว จากขั้นตอนที่ 7 โดยวิธีการ Binary Threshold ได้เป็น Region ใหม่ และ นำคลาสเตอร์พลาสมของไวรัส ทั้งหมดใน Region เข้ากระบวนการ K mean clustering



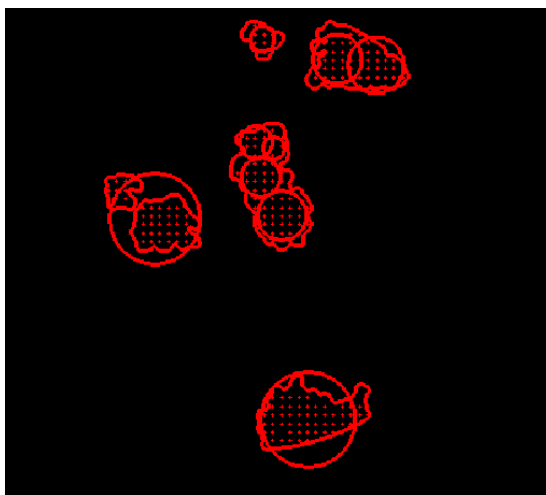
รูปที่ 56 แสดง Region ใหม่ หลักจากการทำการ Binary Threshold

ขั้นตอนที่ 10 K mean segmentation จะเป็นกระบวนการที่จะทำการแบ่งคลาสเตอร์ ของพลาสมทั้งหมดในภาพ โดยขั้นตอนจะคล้ายกับขั้นตอนที่ 10 ในแบบที่ 1



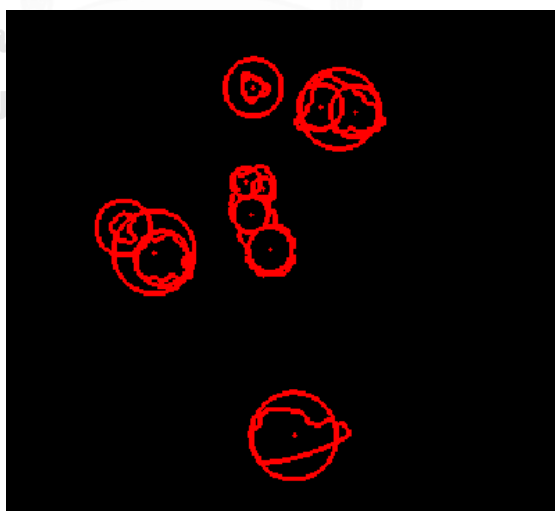
รูปที่ 57 แสดง ภาพหลังจากเข้ากระบวนการ K mean clustering

ขั้นตอนที่ 11 เมื่อทำการ K mean clustering เสร็จเป็นที่เรียบร้อยแล้วจะได้ Region ที่ประกอบไปด้วย พลาคของไวรัสทั้งหมด



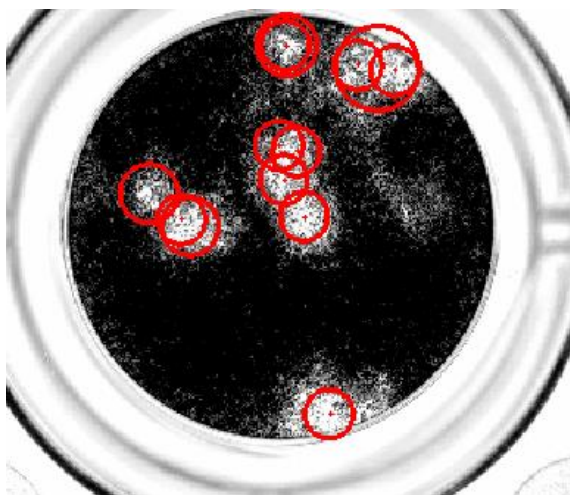
รูปที่ 58 แสดง Region หลังจากเข้ากระบวนการ K mean clustering

ขั้นตอนที่ 12 จากนั้นเพื่อเป็นการเพิ่มความแม่นยำ เนื่องจากการนับพลาคของไวรัสที่ไม่ได้ซ้อนทับกันด้วยวิธี binary threshold มีความคลอบคลุมคลาสเตอร์นั้นๆได้ดี จึงทำการนับเฉพาะคลาสเตอร์ของพลาคที่ไม่ได้ซ้อนทับกัน ได้มาเป็น อีกหนึ่ง Region



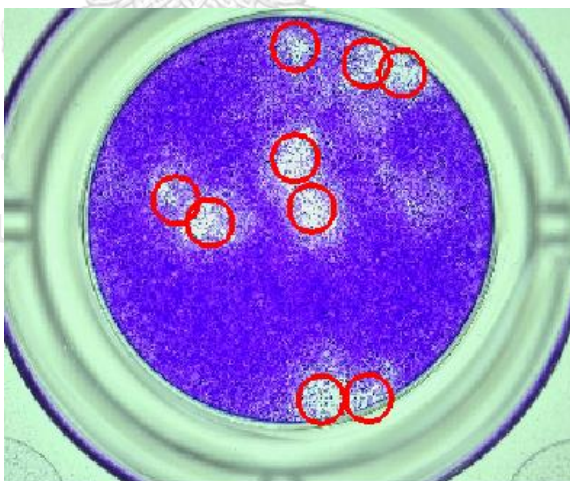
รูปที่ 59 แสดงการนับ Region หลังจากเข้ากระบวนการ K mean clustering

ขั้นตอนที่ 13 นำ Region จากการแบ่งด้วย K mean clustering มา Intersection กับ Region ที่ทำการแบ่งจาก binary threshold ว่าในส่วนไหนมีพื้นที่ทับกัน ก็จะให้นับเป็น คลาสเตอร์ 1



รูปที่ 60 แสดงการ Intersection ของ Region

ขั้นตอนที่ 14 ทำการรวมและแสดงผล



รูปที่ 61 แสดงผลการนับพลาซของไวรัส

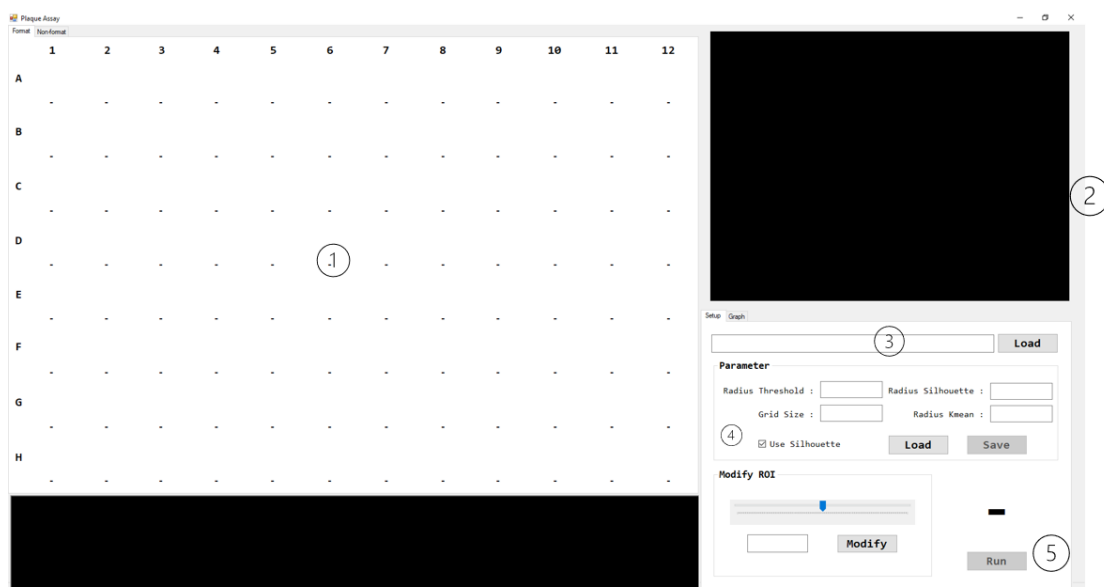
3.1.3 การกำหนดและการเลือกใช้ Algorithms ทั้งสอง

จากการพัฒนา Algorithms ระบบระเบียบวิธีการนับพลาของไวรัสอัตโนมัติ ทั้งสองนั้น จากงานวิจัยก่อนหน้านี้ [31] ว่าการนับพลาของไวรัสในแต่ละหลุมนั้นจะ ต้องนับ และแสดงผล ระหว่างนั้น ไม่เกิดข้อผิดพลาดขณะที่โปรแกรมทำงานอยู่ (No Bugs and No Error) ได้ทุกรูปภาพไม่ว่า ในรูปภาพ นั้นจะมีจำนวนพลาของไวรัสตั้งแต่ 0 ไปจนถึงนับไม่ได้ รูปแบบ Algorithm แบบที่ 1 สามารถทำตามเงื่อนไขนี้ได้ทั้งหมด แต่สำหรับ Algorithms แบบที่ 2 ไม่สามารถทำตามเงื่อนไขได้โดยไม่สามารถอ่านภาพ ได้ทั้งหมดซึ่งมีบางภาพ ทำให้ Algorithms เกิด Error เช่น ภาพที่ไม่มีพลาอยู่ในเฟลททดลอง เป็นต้น จึงทำการนำ Algorithm แบบที่ 1 มาใช้ในการพัฒนา ส่วนเชื่อมต่อกับผู้ใช้งาน (Graphic User Interface) ต่อไป

3.2 การออกแบบโปรแกรม Graphic User Interface สำหรับการนับพลาของไวรัส

โปรแกรม การนับพลาอัตโนมัตินั้นโดยปกติจะต้องทำการเปิดผ่านโปรแกรม Mv tech Halcon จึงยากต่อการใช้งานและยากต่อการเปิดโปรแกรมและเช็คค่าพารามิเตอร์ต่างๆ โดยรวมคือ ยากต่อการใช้งานในแต่ละครั้ง โดยเพื่อให้การแสดงผลมีความชัดเจน และง่ายต่อการใช้งาน จึงได้ทำการเชื่อมต่อโปรแกรมการนับพลาอัตโนมัติเข้ากับ Application ซึ่งโดยทำการสร้างหน้าต่าง Window แสดงการทำงาน และแสดงผลลัพท์ในแต่ละครั้ง โดยแสดงผ่านตัว Graphic User Interface ซึ่งทำการพัฒนา Application ด้วย .Netframework ซึ่งเป็น Framework พื้นฐานที่ทำการพัฒนาโปรแกรม ด้วย C# โดยตัว .Netframework มีการจัดการข้อมูลและเชื่อมต่อไปยังบริการและอุปกรณ์ต่างๆ ซึ่งทำให้ C# .NET เป็นภาษาโปรแกรมมิ่งที่เหมาะสมสำหรับการพัฒนาโปรแกรมแอปพลิเคชันที่มีความซับซ้อนและใช้งานร่วมกับระบบที่มีขนาดใหญ่

โดยเมื่อ อัปโหลดไฟล์ภาพและ กดปุ่ม Run บน Graphic User Interface บนตัว Application โปรแกรมการนับพลาอัตโนมัติก็จะทำงานทันทีด้วย Algorithm แบบที่ 1 โดยใน Application จะมีตัว Graphic User Interface ลักษณะนี้

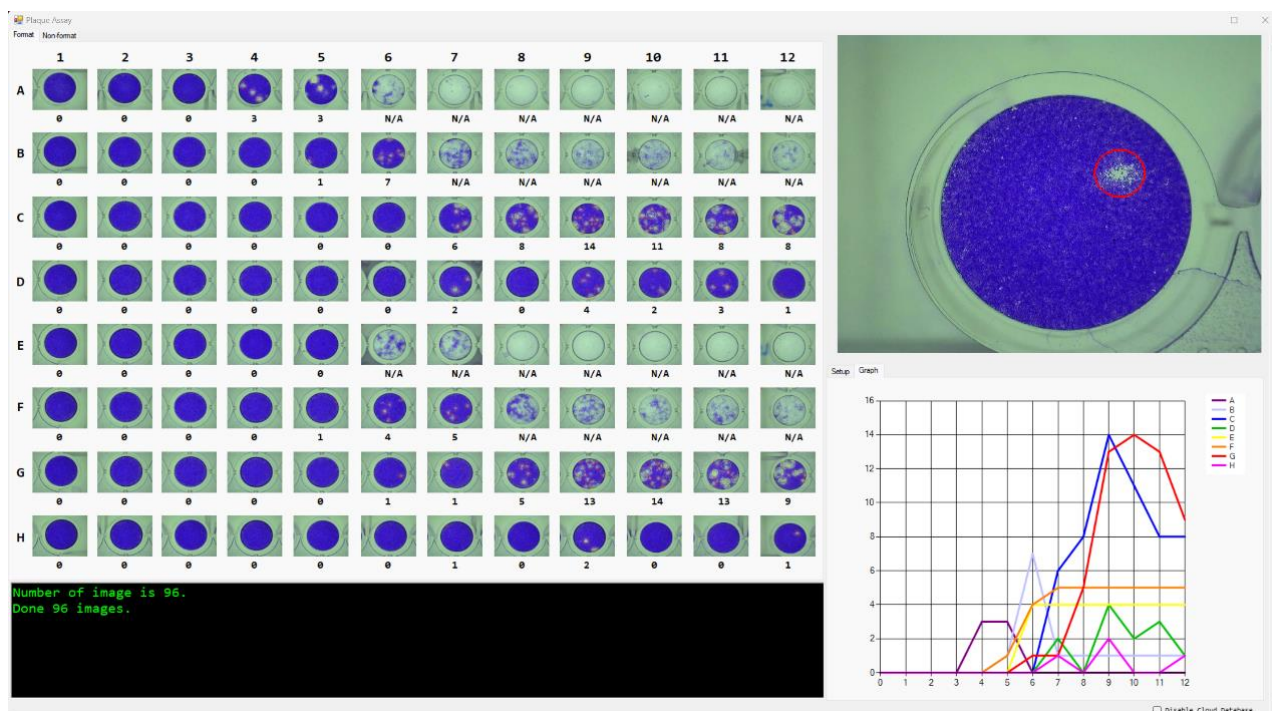


รูปที่ 62 แสดง Graphic User interface

ในหน้าต่างจะประกอบไปด้วย ส่วนประกอบดังนี้
ก่อนเริ่มโปรแกรม

- 1.พื้นที่แสดงผลการนับพลาของไวรัสในรูปแบบของ 96 – well Plate
- 2.พื้นที่แสดงภาพการนับพลาในขณะนั้น
- 3.พื้นที่สำหรับการอัปโหลดรูปภาพที่ทำการถ่ายจาก 96- well Plate
- 4.พื้นที่สำหรับการปรับพารามิเตอร์ต่างๆ ให้เหมาะสมกับรูปภาพถ่ายในแต่ละครั้ง
- 5.ปุ่มสำหรับ Run โปรแกรม

เมื่อสิ้นสุดการ Run Program หน้าต่างส่วน พื้นที่สำหรับการปรับพารามิเตอร์ จะเปลี่ยนเป็นพื้นที่แสดงข้อมูลดัง รูป 63 โดยวิธีการใช้งานจะถูกกล่าวถึงในบทที่ 4

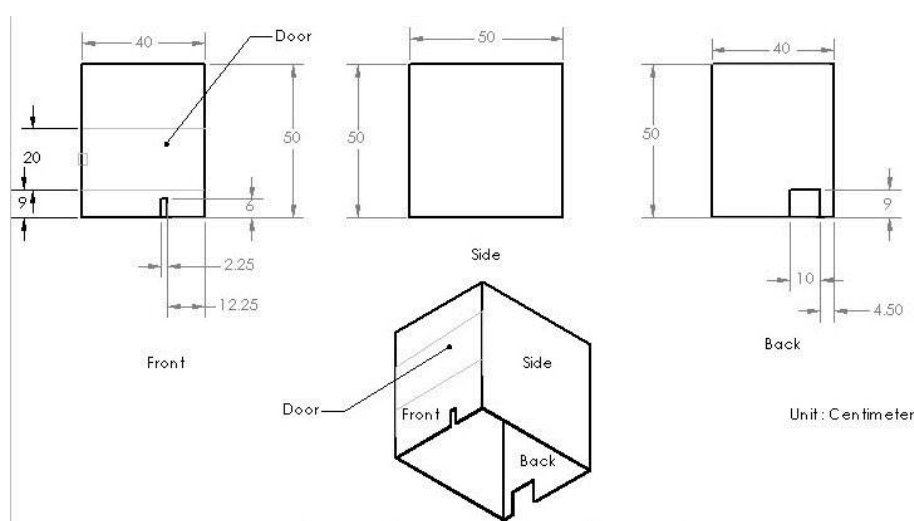


รูปที่ 63 ภาพแสดงพื้นที่ กราฟ จำนวนพลาของไวรัสในแต่ละหลุม

3.3 การพัฒนาต่อเครื่องถ่ายภาพอัตโนมัติ

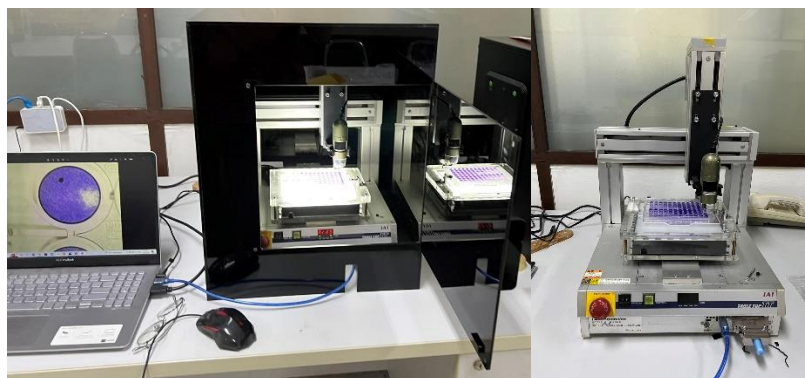
3.3.1 อุปกรณ์ป้องกันแสงจากภายนอก

จากการทดลองพบว่าแสงจากภายนอกเป็นปัจจัยหนึ่งที่ทำให้เกิดสิ่งรบกวนในรูปภาพ จึงได้ทำการออกแบบและสร้างอุปกรณ์ป้องกันแสงจากภายนอก โดยวัสดุทำจาก อะคริลิก สีดำทึบแสงทั้งสี่ด้าน โดยด้านหน้า ประกอบไปด้วยประตูเปิด ปิด สำหรับการวางเฟลทของไวรัสเพื่อทำการถ่ายรูป และ ด้านหลังได้มีช่องสำหรับ สายไฟ และ สายสื่อสาร

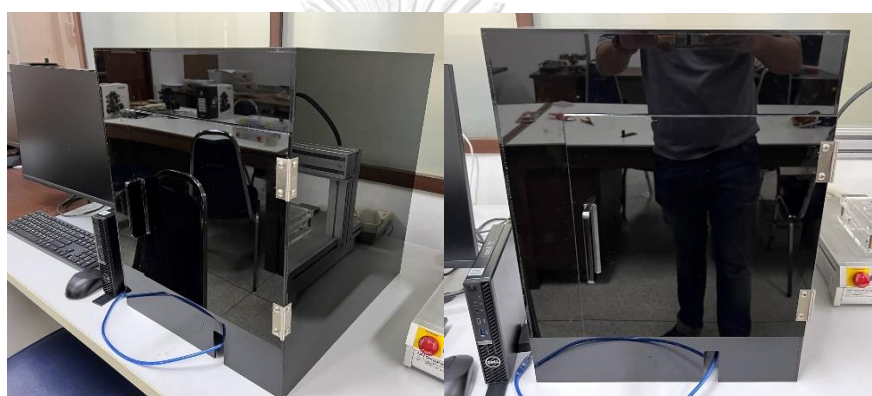


รูปที่ 64 แสดงแผนภาพการออกแบบอุปกรณ์ป้องกันแสงและขนาด

อุปกรณ์ป้องกันแสงนั้นมีขนาด 40x50x50 กว้างยาวและสูง ประกอบเข้ากับเครื่องถ่ายรูป ด้วยวิธีการวางครอบเครื่องถ่ายรูป เพื่อป้องกันแสงดังรูป 65 และ รูป 66 แสดงเครื่องถ่ายรูปที่ได้ถูกพัฒนาเพื่อให้กระบวนการถ่ายภาพดียิ่งขึ้นโดยการลดเสียงรบกวนจากแสงภายนอก และ ทำให้มีรูปร่างหน้าตาเครื่องถ่ายรูปดียิ่งขึ้น



รูปที่ 65 แสดงเครื่องถ่ายภาพหลังและก่อนประกอบอุปกรณ์ป้องกันแสง

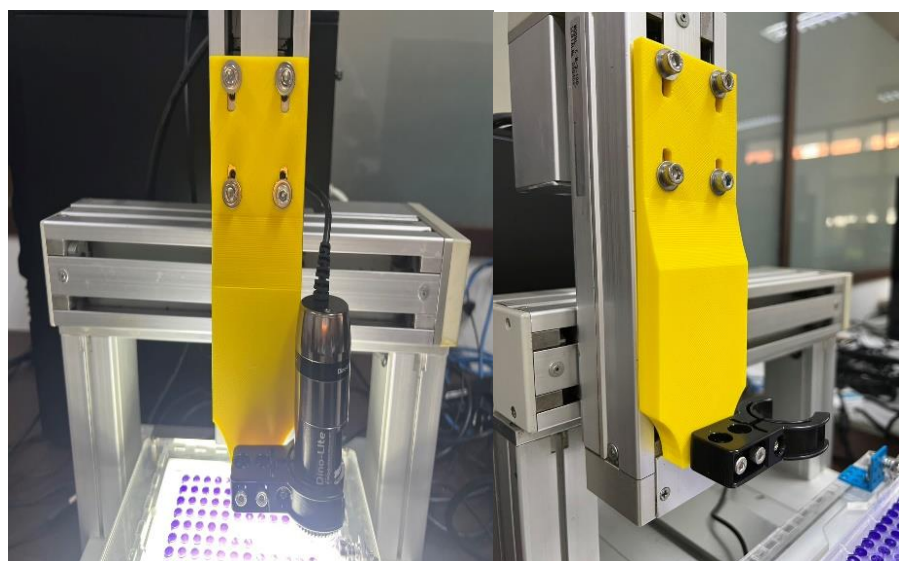


รูปที่ 66 แสดงเครื่องถ่ายภาพรูปที่ได้รับการปรับปรุงเสร็จ

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

3.3.2 ออกแบบชิ้นส่วนยึดกล้องถ่ายภาพ

เนื่องจากชิ้นส่วนที่ยึดกล้องเดิมที่ติดตั้งมาจากงานวิจัยก่อนหน้านี้ [31] นั้นประกอบด้วยชิ้นส่วนหลายชิ้นส่งผลทำให้ ยากในการนำมาพัฒนาและจัดหาชิ้นส่วนให้ครบ จึงได้ทำการออกแบบชิ้นส่วนที่ยึดกล้องกับเครื่องถ่ายภาพ โดยการออกแบบนั้นได้ออกแบบให้มีความแข็งแรงมั่นคงและเป็นชิ้นเดียวกันรวมเป็นชิ้นเดียว ดังรูปที่ 67



รูปที่ 67 แสดงชิ้นส่วนยึดกล้องเมื่อติดตั้งกับเครื่องถ่ายภาพ

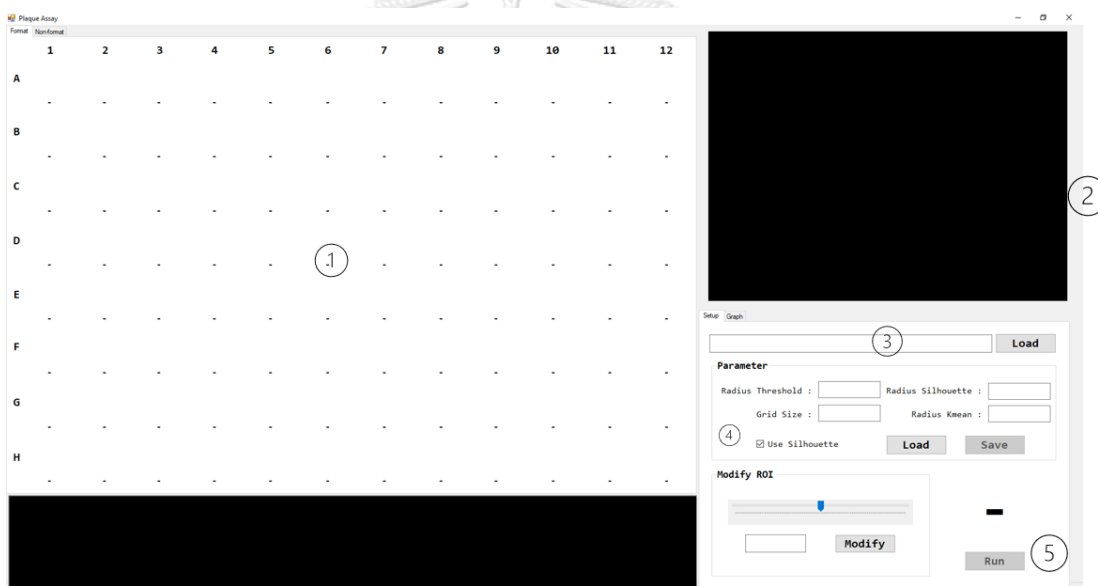
บทที่ 4

การใช้งานโปรแกรมระเบียบวิธีการนับพลาควของไวรัสอัตโนมัติ

ในการวิจัยนี้เป็นงานวิจัยที่พัฒนาต่อเนื่องได้ทำการปรับปรุงออกแบบและสร้างโปรแกรมระเบียบวิธีการนับ พลาควของไวรัสอัตโนมัติ เพื่อให้มีการใช้งานที่ง่ายยิ่งขึ้น จึงได้สร้างส่วนต่อประสานกับผู้ใช้งานหรือ Graphic User Interface (GUI) ซึ่งจะใช้ Algorithm วิธีในแบบที่ 1 ที่กล่าวไว้ในบทที่ 3 สำหรับนับพลาควของไวรัส โดยโปรแกรมนั้นจะมีวิธีการใช้งานดังนี้

4.1 วิธีการใช้งาน โปรแกรม ระเบียบวิธีการนับพลาควอัตโนมัติ

1. เปิดโปรแกรม Window app from plaque ขึ้นมา จะได้หน้าต่างขึ้นมาดังรูปภาพ

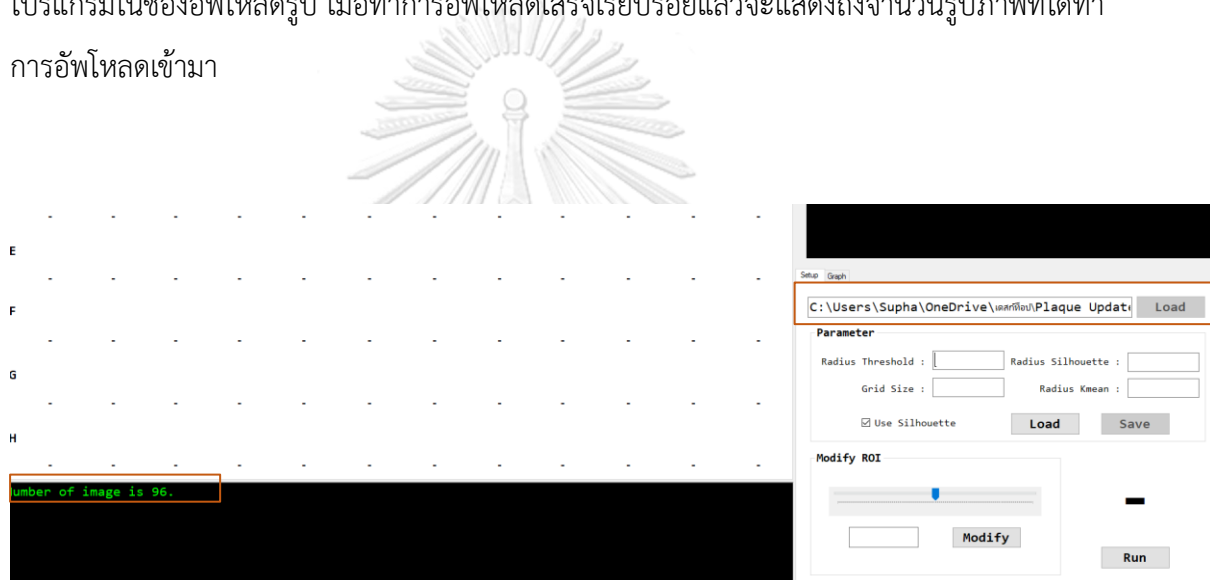


รูปที่ 68 ภาพแสดง โปรแกรม Window app from plaque

ในหน้าต่างจะประกอบไปด้วย ส่วนประกอบดังนี้

- 1.พื้นที่แสดงผลการนับพลาของไวรัสในรูปแบบของ 96 – Well Plate
- 2.พื้นที่แสดงภาพการนับพลาในขณะนั้น
- 3.พื้นที่สำหรับการอัปโหลดรูปภาพที่ทำการถ่ายจาก 96 - Well Plate
- 4.พื้นที่สำหรับการปรับพารามิเตอร์ต่างๆ ให้เหมาะสมกับรูปภาพถ่ายในแต่ละครั้ง
- 5.ปุ่มสำหรับ Run โปรแกรม

2. ทำการอัปโหลดรูปภาพถ่ายพลาของไวรัส ที่ทำการถ่ายจาก 96 - Well Plateเข้ามาในโปรแกรมในช่องอัปโหลดรูป เมื่อทำการอัปโหลดเสร็จเรียบร้อยแล้วจะแสดงถึงจำนวนรูปภาพที่ได้ทำการอัปโหลดเข้ามา



จุฬาลงกรณ์มหาวิทยาลัย

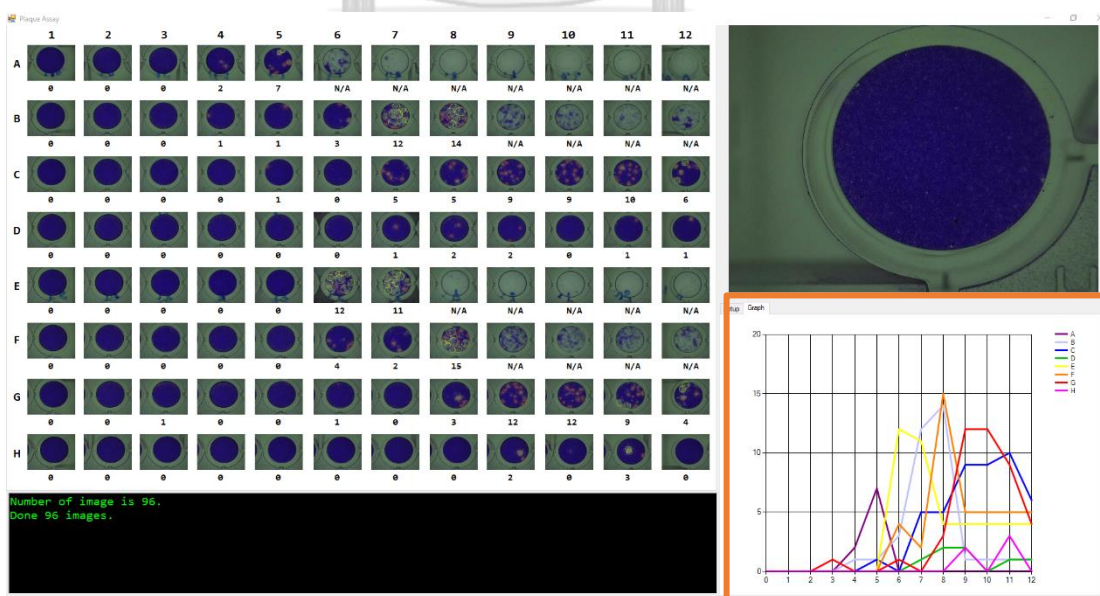
CHULALONGKORN UNIVERSITY

รูปที่ 69 ภาพแสดง วิธีการทำการอัปโหลดรูปภาพถ่ายพลาของไวรัส

3.ทำการ ใส่ค่าพารามิเตอร์ต่างๆที่ต้องการซึ่งในการถ่ายภาพในแต่ละครั้งนั้นจะมีรายละเอียดที่แตกต่างกันในส่วนนี้จึงสร้างให้ผู้ใช้งานปรับได้ตามความเหมาะสม หรือจะใช้ค่า Default โดยกดที่ปุ่ม Load จะทำการตั้งค่า ได้ดังนี้ จากนั้นกดปุ่ม Save

รูปที่ 70 ภาพแสดง วิธีการทำการเซตค่าพารามิเตอร์ต่างๆบนโปรแกรม

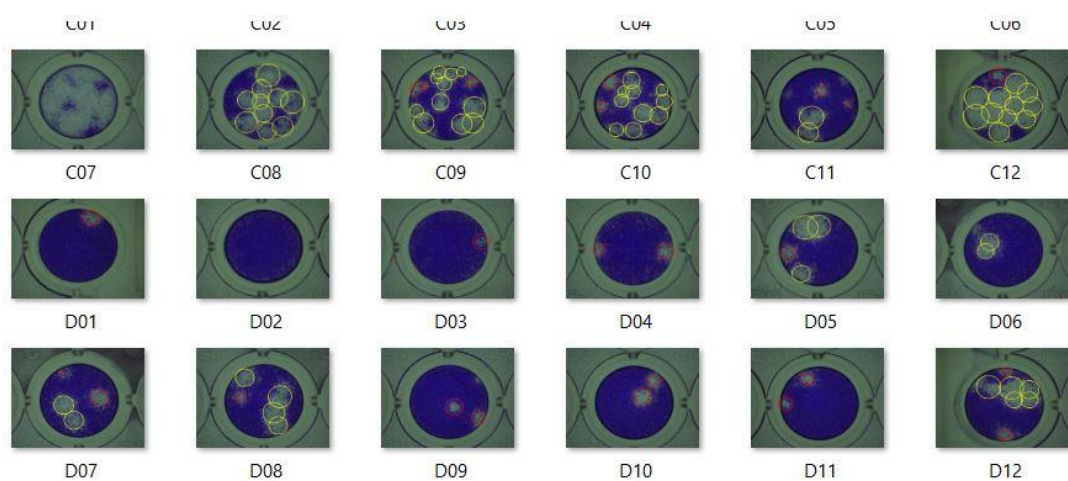
4. ทำการ Run โปรแกรม โดยการกดปุ่ม Run โปรแกรมจะเริ่มทำการนับ พลาของไวรัส โดยเมื่อทำการนับเสร็จสิ้นจะแสดงผลดังนี้ โดยถ้า จำนวนพลาของไวรัสมีพื้นที่เกินกว่าที่กำหนดไว้หรือไม่สามารถนับได้ จะแสดงเป็น N/A



รูปที่ 71 ภาพแสดง โปรแกรม Window app from plaque เมื่อเสร็จสิ้นการทำงาน

จากภาพ 71 จะเห็นได้ ว่าพื้นที่หมายเลข 4 ที่เป็นพื้นที่ปรับแก้พารามิเตอร์จะเปลี่ยนเป็นพื้นที่แสดงข้อมูล จำนวนพลาซมของไวรัสโดยจะทำการพลอตออกมาเป็นกราฟในกรอบสีแดงในภาพ 4-4 โดยจะแยกตามแถวแสดงเป็นสีแต่ละสีของผลการทดลอง ตั้งแต่ A-H

5.หลักจากโปรแกรมการนับพลาสมาอัตโนมัติ ทำงานเสร็จสิ้นนั้น รูปภาพที่ทำการวงและนับแล้วจะถูกบันทึกไว้ใน Folder Counted อัตโนมัติในแต่ละชุดภาพที่นำมานับ



รูปที่ 72 แสดงผลการนับพลาสมาของไวรัสแสดงไว้ใน Folder

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

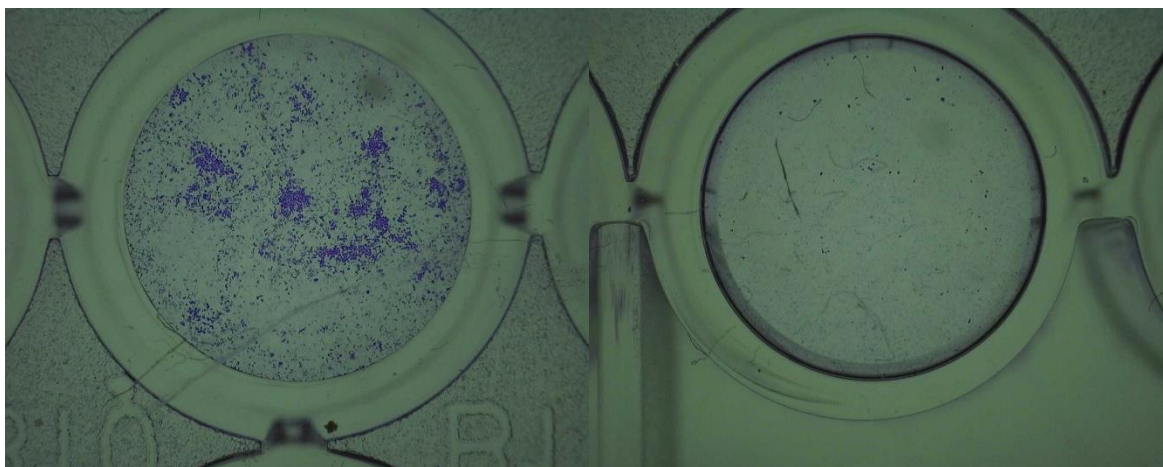
บท 5

การวิเคราะห์ผล และการประเมินผล

ระบบระเบียบวิธีการนับพลาของไวรัส (Plaque Assays) ถูกจัดเตรียมขึ้นโดย ศูนย์วิจัยไวรัสวิทยาทางการแพทย์ คณะแพทยศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย และ ทำการทดลองบนเพลทการทดลองชนิด 96 Well Plate โดยใช้เชื้อไวรัสโรคไข้เลือดออกสำหรับการทดลองเป็นจำนวน 10 Well Plate เพื่อทำการประเมินผลการนับพลาของไวรัสออตโนมิติ โดยผลการนับจำนวนพลาของไวรัส ทางผู้เชี่ยวชาญสามารถทำการคัดแยกเพลทการทดลองที่มีข้อมูล พลาของไวรัส ในช่วง 0-19 คลาสเตอร์ จำนวน 606 ตัวอย่าง เพื่อใช้ในการประเมินความสามารถโปรแกรมการนับพลาไวรัสออตโนมิติ

5.1 วิธีการประเมินผลการนับพลาของไวรัส

เพลทการทดลองกระบวนการนับพลาของไวรัส (Plaque Assays) ชนิด 96- Well Plate จำนวน 10 Well Plate นั้นได้ทำการถ่ายรูปทีละหลุม โดยเครื่องถ่ายรูปพลาออตโนมิติ จำนวน 960 รูป/ตัวอย่าง ซึ่งจะประกอบด้วย รูปภาพที่มีข้อมูลเพื่อนำไปประมวลผล และ รูปภาพที่ไม่นำมาประมวลผล โดยรูปภาพที่ไม่นำมาประมวลผลการนับพลาของไวรัสจะประกอบไปด้วย รูปภาพที่ 1. พื้นที่ของพลาเกินกำหนด มีจำนวนมากกว่า 19 คลาสเตอร์ (ซึ่งรายละเอียดนั้นการแยกนั้นได้กล่าวถึงในบทที่ 3) ดังรูป 73 ก 2. ในเพลททดลองไม่มี สารทดลองอยู่ (ช่องทดลองว่างเปล่า) ดังรูป 73 ข ฉะนั้นจากการคัดแยกรูปภาพโดยผู้เชี่ยวชาญนั้น ทำให้มีรูปภาพ สามารถนำไปประมวลผล โปรแกรมการนับพลาของไวรัสทั้งหมด 606 รูป ประกอบไปด้วยคลาสเตอร์พลาของไวรัส ในช่วง 0-19 คลาสเตอร์



ก

ข

รูปที่ 73 แสดงรูปภาพที่ไม่นำมาประมวลผล

5.1.1 ข้อกำหนดของความถูกต้องการนับปริมาณพลาของไวรัส

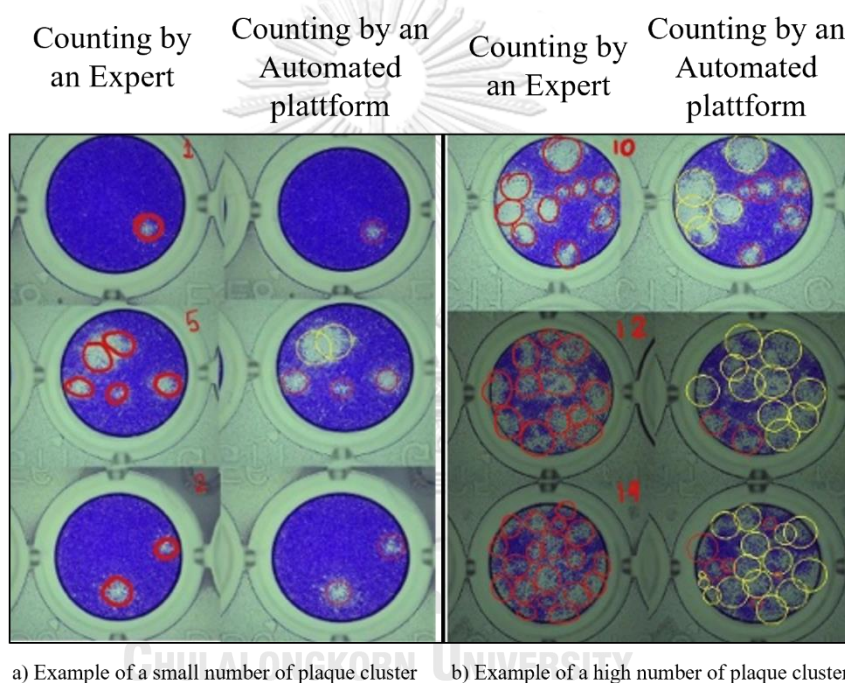
การนับพลาของไวรัสนั้นเป็นกระบวนการที่ค่อนข้างที่ละเอียดอ่อน ผู้เชี่ยวชาญจะสามารถระบุจำนวนของพลาที่ถูกต้อง โดย รูปภาพ 74 ได้ทำการแสดงให้เห็นถึงการนับพลาของไวรัสโดยโปรแกรมอัตโนมัติเทียบกับผู้เชี่ยวชาญ โดยเมื่อจำนวนพลาที่เพิ่มขึ้นก็ทำให้ความยากในการนับเพิ่มขึ้นเช่นกัน โดยพื้นที่พลาของไวรัสที่เล็กที่สุดของเชื้อไวรัส โรคไขเลือดออกที่มองเห็นได้นี้มีขนาด เส้นผ่านศูนย์กลางประมาณ 1-2 มิลลิเมตร [7]

เพราะฉะนั้น ทางผู้เชี่ยวชาญ ผศ.ดร.พญ.ศิวะพร บุญยทรัพย์ากร จึงได้กำหนดเกณฑ์การตัดสิน เรียกว่า ค่าอ้างอิงของปริมาณพลาของไวรัส โดยถ้าผลการนับจากโปรแกรมนับพลาของไวรัสนั้นอยู่ในค่าอ้างอิงนี้ ถือว่าผลการนับนั้นถูกต้อง มี 5 ช่วงดังนี้

1. ค่าอ้างอิงของปริมาณพลาของไวรัส ในช่วง 0 – 1 ผลการนับต้องอยู่ในช่วงบวกลบ 0
2. ค่าอ้างอิงของปริมาณพลาของไวรัส ในช่วง 2 – 3 ผลการนับต้องอยู่ในช่วงบวกลบ 1
3. ค่าอ้างอิงของปริมาณพลาของไวรัส ในช่วง 4 – 8 ผลการนับต้องอยู่ในช่วงบวกลบ 2
4. ค่าอ้างอิงในช่วงของปริมาณพลาของไวรัส 9 - 10 ผลการนับต้องอยู่ในช่วงบวกลบ 3
5. ค่าอ้างอิงในช่วงของปริมาณพลาของไวรัส 11 - 15 ผลการนับต้องอยู่ในช่วงบวกลบ 4
6. ค่าอ้างอิงของปริมาณพลาของไวรัส ตั้งแต่ 16 เป็นต้นไปต้องอยู่ในช่วงบวกลบ 5

เช่น จำนวนพลาของไวรัสในเพลท B12 มี พลาของไวรัสจำนวน 7 คลาสเตอร์ (ผลการนับโดยผู้เชี่ยวชาญ) แต่ผลการนับโดยโปรแกรมนับพลาของไวรัสอัตโนมัติได้ 6 คลาสเตอร์ ก็ถือว่าผลการนับถูกต้อง เพราะจาก ค่าอ้างอิง ปริมาณพลาของไวรัส มีปริมาณ 4-8 คลาสเตอร์ จะมีช่วงการนับที่ถูกต้องในช่วง บวก ลบ 2

จากเกณฑ์การตัดสินความถูกต้อง หรือ ค่าอ้างอิงจะเห็นได้ว่า เมื่อจำนวนพลาของไวรัสเมื่อมีปริมาณเพิ่มมากขึ้น ค่าช่วง บวกลบ ความถูกต้อง ช่วงก็จะกว้างขึ้น แต่อย่างไรก็ตาม การนำผลการนับจำนวนพลาของไวรัสไปใช้ในการวิจัยต่อ จะเน้น ปริมาณพลาของไวรัส ในปริมาณไม่เกิน 12 คลาสเตอร์มากกว่า [31]



รูปที่ 74 แสดงการนับพลาของไวรัสเทียบระหว่างโปรแกรมและผู้เชี่ยวชาญ

5.2 การวิเคราะห์ผล ความผันแปรของระบบการวัดของเครื่องถ่ายภาพอัตโนมัติและระเบียบการนับพลาของไวรัส

5.2.1 ความแม่นยำ (Precision)

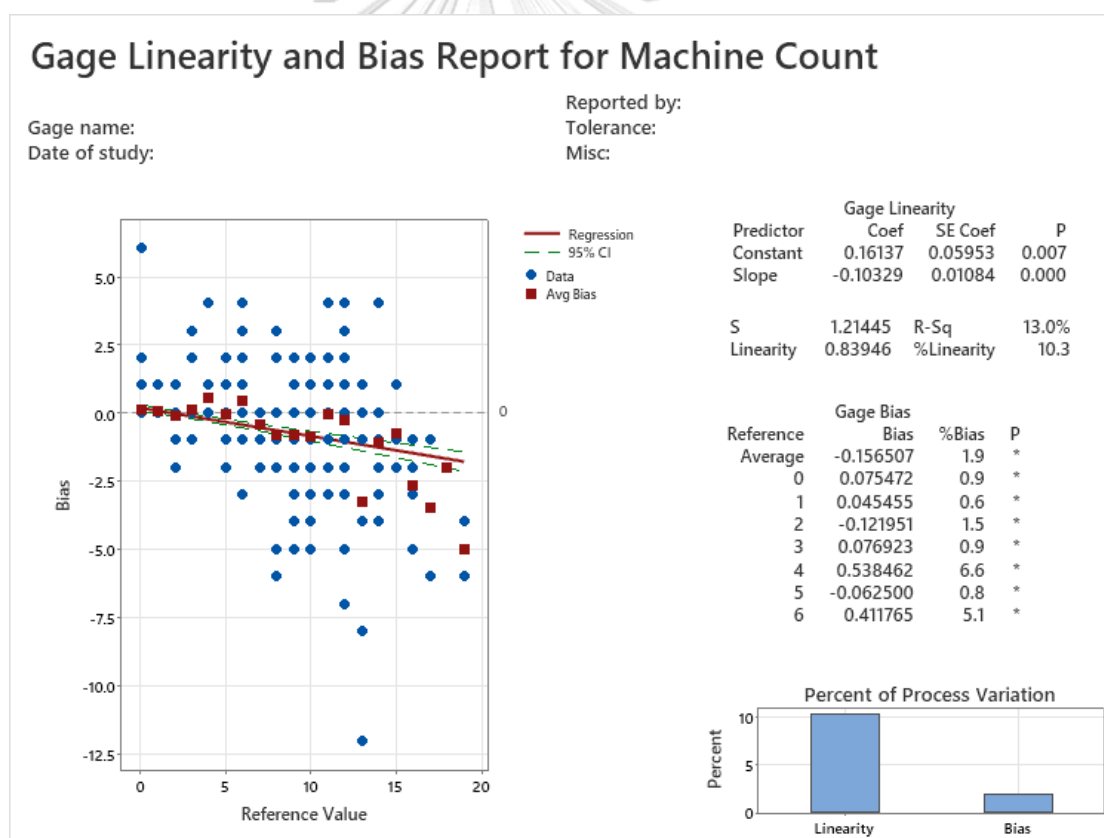
ความแม่นยำ (Precision) เป็นตัวบ่งชี้ระดับความสอดคล้องหรือการซ้ำกันของการวัดเดิมกันหลายครั้ง งานวิจัยนี้มีจุดประสงค์ที่จะทำการพัฒนาให้มีประสิทธิภาพที่ดีขึ้นจึงนำความสนใจ ของข้อมูลการวัดมาเป็นพื้นฐานในการวัดผลเบื้องต้น ซึ่งจะประกอบไปด้วย (1) ไบอัส Bias (2) คุณสมบัติเชิงเส้น Linearity และ (3) ความเสถียร Stability จากนั้นคำนวณหาค่าเฉลี่ย (\bar{x}) และค่าเบี่ยงเบนมาตรฐาน

(SD) ของผลลัพธ์ในแต่ละครั้งแต่ละจำนวนคลาสเตอร์ เพื่อหาความแม่นยำของโปรแกรมนับผลาคของไวรัสอัตโนมัติ

เนื่องจากโปรแกรมการนับ ผลาคของไวรัสนั้น ใช้ Algorithm Machine learning ชนิด Unsupervised Learning ผลการนับในแต่ละครั้งนั้นอาจจะออกมาไม่เท่ากันทุกครั้ง ฉะนั้นความค่าเสถียรภาพของผลการนับ ผลาคของไวรัสจากโปรแกรมนั้นจึงเป็นสิ่งที่สำคัญ

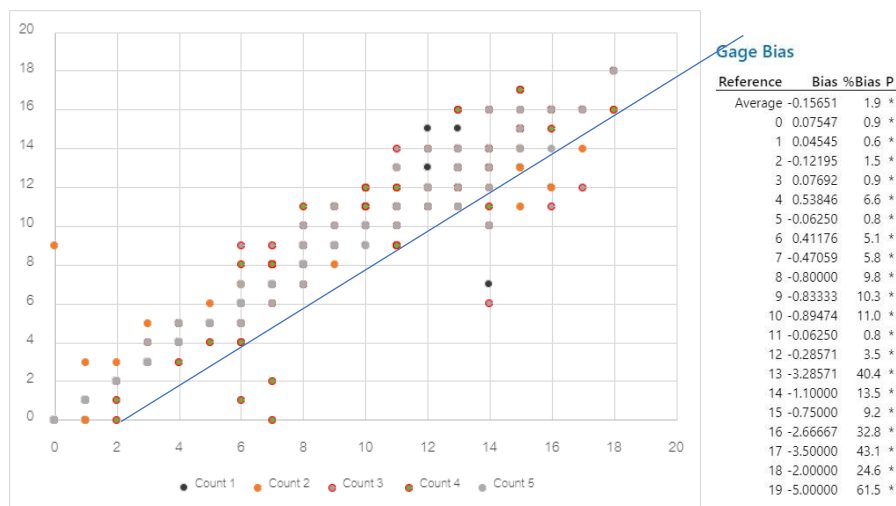
5.2.1.1 การประเมินผลการนับผลาคของไวรัสด้วยโปรแกรม Minitab

1. ทำการสุ่มเลือกชุด Well-plate มาทั้งหมด 5 ชุด ประกอบไปด้วยรูปภาพ ผลาคของไวรัสซึ่งมีข้อมูลผลาค ปรากฏอยู่จำนวน 300 รูป ซึ่งมีปริมาณอยู่ในช่วง 0-19 คลาสเตอร์ ทำการถ่ายรูปใหม่ 5 ครั้ง เป็น 5 ชุดข้อมูล ให้โปรแกรมนับ ซ้ำทั้งหมด 5 ครั้ง
2. ทำการใส่ข้อมูลในโปรแกรม Minitab แสดงผลดังรูป 5-1 ถึง 5-2



รูปที่ 75 ผลการคำนวณค่าคุณสมบัติเชิงเส้นโดยโปรแกรม Minitab ของโปรแกรมระเบียบวิธีการนับผลาคของไวรัส

ผลการคำนวณด้วยโปรแกรม Minitab พบว่าค่า คุณสมบัติเชิงเส้น Linearity เท่ากับ 0.83946 และ ค่า ไบอัส Bias ของแต่ละค่าดังแสดงในรูปที่ 75 จากผลการคำนวณวิเคราะห์คุณสมบัติเชิงเส้น (Linearity) มีค่า 10.3 %



รูปที่ 76 ผลการคำนวณค่าไบอัส Bias ของแต่ละค่าโดยโปรแกรม Minitab และกราฟแสดงการกระจายตัวของข้อมูลการนับ

จากนั้น คำนวณหาค่าเฉลี่ยเลขคณิต (\bar{x}) และค่าเบี่ยงเบนมาตรฐาน (SD) ของผลการนับจาก โปรแกรมอัตโนมัติ นับซ้ำทั้งหมด 5 ครั้ง เพื่อหาการกระจายตัวของผลการนับทั้ง 5 ครั้งได้ ค่าเฉลี่ยเลขคณิตเฉลี่ย (\bar{x} Average) เท่ากับ 2.970476 ค่าส่วนเบี่ยงเบนมาตรฐานเฉลี่ย (SD Average) เท่ากับ 2.953333 จากกราฟการกระจายตัวของข้อมูล ในรูปที่ 75 และ รูปที่ 76 มีการกระจายตัวของข้อมูลที่ไม่ได้กว้างมากถือว่ามีค่าความแม่นยำที่ตนเอง งานวิจัยนี้ได้ ทำการออกแบบ Algorithm ให้นับได้เฉพาะ 0-19 คลาสเตอร์เท่านั้นเนื่องจากจำนวนคลาสเตอร์ที่มากขึ้นจะเกิดข้อผิดพลาดได้ง่าย และไม่ได้จำเป็นต้องนำข้อมูลพลาคที่มากไปใช้ เพื่อความเสถียรภาพจึงได้ออกแบบ โปรแกรมให้มีความแม่นยำมากขึ้นจากงานวิจัยก่อนหน้า [32] ในช่วงนี้ โดยค่าไบอัสเฉลี่ย Average Bias ที่ได้นั้นคือ 1.9% คือผลการนับในแต่ละครั้งนั้นมีความ เสถียรภาพ มากยิ่งขึ้นกว่างานวิจัยก่อนหน้า [32] อย่างชัดเจน

5.2.1.2 Gauge R and R ของโปรแกรมนับพลาคของไวรัสอัตโนมัติ

จากที่กล่าวไปข้างต้น ว่า Algorithm ของโปรแกรมนับพลาคอัตโนมัติ นั้นใช้กระบวนการ Machine Learning ชนิด Unsupervised Learning ผลลัพธ์จึงอาจจะมีคลาดเคลื่อนได้แต่

ละครั้ง โดยกระบวนการ Gauge R and R (Gage R&R) เป็นกระบวนการทางสถิติที่ใช้ในการวัดความแม่นยำของเครื่องมือวัดหรืออุปกรณ์ที่ใช้ในกระบวนการผลิตหรือการทดสอบโดยการประเมินความแม่นยำของเครื่องมือวัดเป็นส่วนสำคัญในการตรวจสอบว่าเครื่องมือมีความแม่นยำในการวัดที่เพียงพอในการใช้งานในกระบวนการและสามารถใช้ในการวัดผลในลักษณะที่ถูกต้องหรือไม่ ในการหาความแม่นยำของระบบโปรแกรมการนับผลของไวรัสที่อ้างอิงจากงานวิจัยก่อนหน้า [32] ความแม่นยำ(Precision) จะสนใจเฉพาะความสามารถในการวัดซ้ำ (Repeatability) ของระบบเท่านั้น จะไม่คำนึงถึงความสามารถในการทำซ้ำ (Reproducibility) เนื่องจาก เครื่องถ่ายภาพผลของไวรัสอัตโนมัติมีใช้ถ่ายภาพเพียงเครื่องเดียวและใช้ คนคนเดียวในการทำ การทดลอง(คุมเครื่องและถ่ายภาพ) เพื่อต้องการลดความแปรปรวน (Variance) ในการวัด เพราะความแปรปรวนนั้นเกิดจากจำนวนคนที่ทำการทดลองและเครื่องถ่ายภาพ

ผลในการหาความสามารถในการทำซ้ำของโปรแกรมการนับผลของไวรัสที่ได้ทำการวิเคราะห์ด้วยโปรแกรม Matlab โดยได้ทำการ ทำการสุ่มเลือกชุด Well-plate มาทั้งหมด 5 ชุด ประกอบไปด้วยรูปภาพ ผลของไวรัสซึ่งมีข้อมูลผลปรากฏอยู่จำนวน 300 รูป ซึ่งมีปริมาณอยู่ในช่วง 0-19 คลาสเตอร์ ทำการถ่ายภาพใหม่ 5 ครั้ง เป็น 5 ชุดข้อมูล ให้โปรแกรมนับ ซ้ำทั้งหมด 5 ครั้ง เพื่อดูผลลัพธ์จากโปรแกรมว่ามีความสม่ำเสมอหรือไม่ แสดงได้ดังภาพ 79

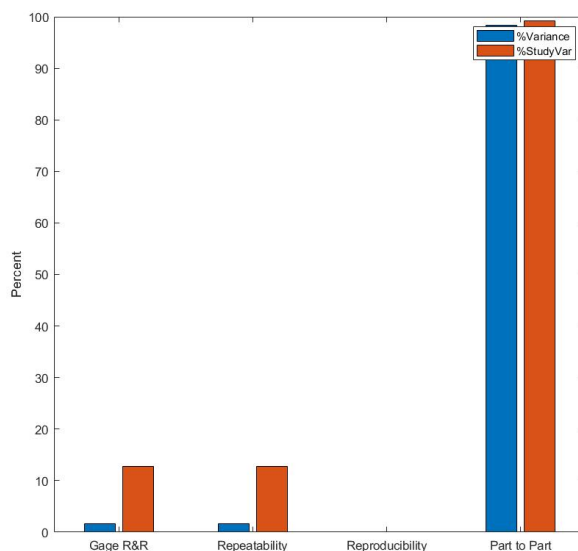
{ 'Source' }	{ 'Variance' }	{ '% Variance' }	{ 'sigma' }	{ '5.15*sigma' }	{ '% 5.15*sigma' }
{ 'Gage R&R' }	{ [0.3124] }	{ [1.6275] }	{ [0.5590] }	{ [2.8786] }	{ [12.7572] }
{ ' Repeatability' }	{ [0.3124] }	{ [1.6275] }	{ [0.5590] }	{ [2.8786] }	{ [12.7572] }
{ ' Reproducibility' }	{ [0] }	{ [0] }	{ [0] }	{ [0] }	{ [0] }
{ ' Operator' }	{ [0] }	{ [0] }	{ [0] }	{ [0] }	{ [0] }
{ 'Part' }	{ [18.8848] }	{ [98.3725] }	{ [4.3457] }	{ [22.3802] }	{ [99.1829] }
{ 'Total' }	{ [19.1972] }	{ [100] }	{ [4.3815] }	{ [22.5645] }	{ 0x0 char }

Number of distinct categories (NDC):11
 % of Gage R&R of total variations (PRR): 12.76
 Note: The last column of the above table does not have to sum to 100%

รูปที่ 77 แสดงผลการคำนวณการวิเคราะห์ผลด้วยวิธี Gauge R and R

จากภาพ 77 ผลลัพธ์จากกระบวนการ Gauge R and R ได้ค่า Total Gauge R and R เท่ากับ 1.6275% ซึ่งค่า Total Gauge R and R นั้นคือการรวมผลของ ค่าความสามารถในการวัดซ้ำ (Repeatability) และ ค่าความสามารถในการทำซ้ำ (Reproducibility)รวมกันต้องไม่เกิน 7% [33] ซึ่งผลลัพธ์ที่ได้นั้นอยู่ในเกณฑ์ที่ยอมรับ มากไปกว่านั้นค่า Total Variation จาก Part to Part ซึ่งเป็นการบอกความสม่ำเสมอของความแปรปรวน ซึ่งมีค่าถึง 98.37% ซึ่งหมายถึงสามารถทำซ้ำและได้ผลลัพธ์ที่คล้ายกัน

โดยค่าความสามารถในการแยกความแตกต่างของชิ้นงานหรือ Number of Distinct Categories (NDC) มีค่าเท่ากับ 11 ซึ่งเป็นผลลัพธ์ที่ดี เนื่องจากค่า NDC นั้นควรจะมีค่ามากกว่า 5 จึงจะบอกได้ว่าการวัดนี้มีความสามารถในการแยกความแตกต่างของชิ้นงานได้ [33]



รูปที่ 78 แสดงการเปรียบเทียบความแปรปรวนของค่าอ้างอิงที่หลากหลาย

จากภาพที่ 78 แสดงแผนภูมิการเปรียบเทียบของความหลากหลายของค่าอ้างอิงและความแปรปรวนจากความสามารถในการวัดซ้ำจะแสดงได้ว่าความหลากหลายของค่าอ้างอิงนั้นมีค่าความแปรปรวนมากกว่าค่าความสามารถในการวัดซ้ำซึ่งถือว่าดี

5.3 การวิเคราะห์ผลและประเมินผลการนับพลาของไวรัสจากโปรแกรมนับพลาของไวรัส

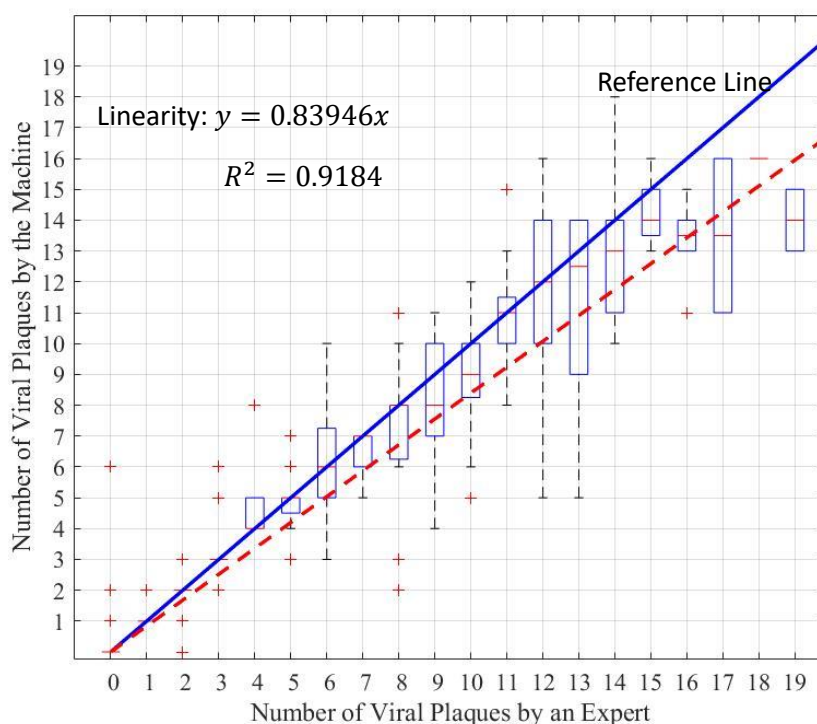
5.3.1 ความเที่ยงตรง (Accuracy)

ความเที่ยงตรง (Accuracy) คือค่า ที่ใช้ในการวัดความถูกต้องของการทดสอบหรือการวัด โดยระบุถึงสัดส่วนของผลลัพธ์ที่ถูกต้องทั้งหมดเมื่อเทียบกับผลลัพธ์ทั้งหมดที่ได้รับ ในงานวิจัยนี้ ในการประเมินความถูกต้องของจำนวนพลาในเพลทการทดลอง ได้ใช้กฎเกณฑ์ตาม ข้อกำหนดของความถูกต้องการนับปริมาณพลาของไวรัส ในช่วงของค่าอ้างอิง ที่กล่าวไปข้างต้นในหัวข้อ 5.1.1 ความว่า ถ้าผลการนับจากโปรแกรมนับพลาของไวรัสนั้นอยู่ในค่าอ้างอิงนี้ ถือว่าผลการนับนั้นถูกต้อง

5.3.2 การประเมินผลการนับพลาของไวรัสด้วยโปรแกรม Matlab

จากได้ทำการเก็บผลการนับพลาของไวรัสจากเพลทการทดลองทั้งหมด 10 เพลท ประกอบไปด้วยภาพชุดข้อมูลที่ใช้ได้ทั้งหมด 606 ภาพ ซึ่งมีจำนวน 0-19 คลาสเตอร์ ได้นำข้อมูลผล

การนับมา วิเคราะห์ผลลัพธ์ทั้งหมดด้วยโปรแกรม Matlab เพื่อต้องการแสดง คุณสมบัติเชิงเส้น (Linearity) บนแผนภูมิการกระจายตัวของข้อมูล Box plot ด้วยผลจากโปรแกรมระบบระเบียบวิธีการนับพลาของไวรัสอัตโนมัติ เทียบกับผลการนับพลาจากผู้เชี่ยวชาญ โดย



รูปที่ 79 แสดงการกระจายตัวของข้อมูลบน Box plot

จากรูปที่ 79 แสดง Box plot ของจำนวนพลาของไวรัส โดย เครื่องหมาย + สีแดง แทน Outliners ของการวัด ซึ่งจะเห็นได้ว่ามีค่าน้อยกว่า 5 % โดยค่า Standard deviation (SD) ในช่วงจำนวนพลาตั้งแต่ 0-12 คลาสเตอร์ มีค่าน้อยกว่า 1 และค่า SD จะเพิ่มขึ้นเรื่อยๆเมื่อจำนวนพลาของไวรัสเพิ่มจำนวนขึ้น กรอบสีฟ้าแสดงผลการนับในช่วงในค่าอ้างอิง เทียบกับผลการนับจากผู้เชี่ยวชาญ เส้น สีฟ้า แสดงเป็นเส้นหลักสำหรับเปรียบเทียบค่า linearity (Reference Value) ซึ่งมีความชันเป็น 1 และ เส้นประสีแดง แสดงถึง Trend Line ของโปรแกรมการนับพลาไวรัสอัตโนมัติ เทียบ กับ ผลการนับโดยผู้เชี่ยวชาญ โดยสร้าง นำมาสร้าง the goodness-of-fit โดยสามารถนำสมการเส้นแดงประมาเขียนเป็นสมการเส้นตรงตามที่คำนวณจากโปรแกรม Minitab ได้สมการ

$$y = 0.83946x \quad (5.1)$$

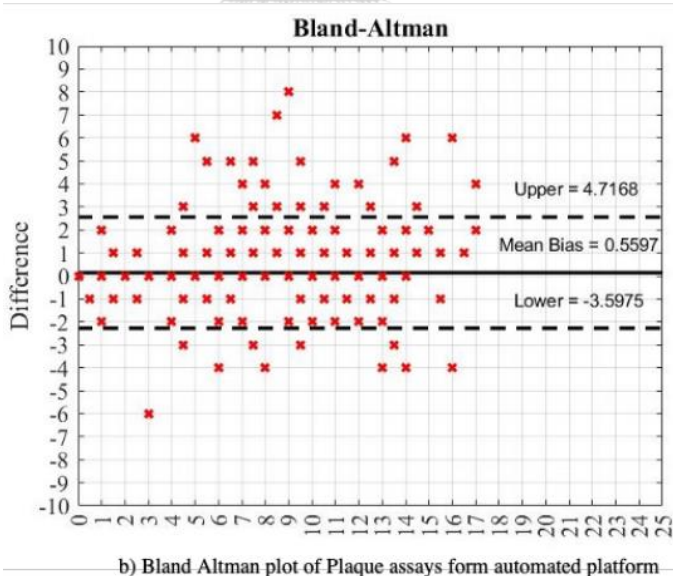
Goodness-of-fit

$$R^2 = 0.9184 \quad (5.2)$$

โดย y คือจำนวนพลาทของไวรัสที่นับโดยโปรแกรมการนับพลาทของไวรัส และ x คือ ผลการนับ พลาทของไวรัส โดย ผู้เชี่ยวชาญ ใน ช่วง 0-19 คลาสเตอร์

5.3.3 Bland – Altman

Bland-Altmanเป็นอีกหนึ่งกระบวนการในการประเมินผลประสิทธิภาพของการทำ Assays เปรียบเทียบกันทั้งสองชนิด [29] โดยในที่นี้ได้นำผลการนับพลาทของไวรัสโดย โปรแกรมระบบระเบียบวิธีการนับพลาทของไวรัสอัตโนมัติ เทียบกับผลการนับจากผู้เชี่ยวชาญโดยนำค่าอ้างอิงจากผู้เชี่ยวชาญมาหาค่าเฉลี่ยความคลาดเคลื่อน (Error mean bias) และค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) 4.7168 และ -3.5975 ตามลำดับ แสดงเป็นค่าขอบเขต Boundaries จากหัวข้อ 5.3.2 นำผลนับพลาทของไวรัสจากโปรแกรมอัตโนมัติที่อยู่ในค่าอ้างอิงแสดงลงไปเป็นจุดสีแดง ดังแสดงดังภาพ78 จะเห็นได้ว่าส่วนใหญ่จุดสีแดงนั้นอยู่ในขอบเขตที่กำหนดไว้ซึ่งผลลัพธ์นี้ถือว่าเป็นที่ยอมรับในการทำกระบวนการ Bland-Altman ที่โปรแกรมนับพลาทของไวรัสอัตโนมัติสามารถนับจำนวนพลาทของไวรัสได้อย่างมีประสิทธิภาพและประสิทธิผล



รูปที่ 80 แสดงค่า Bland – Altman ของโปรแกรมการนับพลาทอัตโนมัติ

5.3.4 ผลลัพธ์จากการประเมินผลโปรแกรมการนับพลาของไวรัสอัตโนมัติ

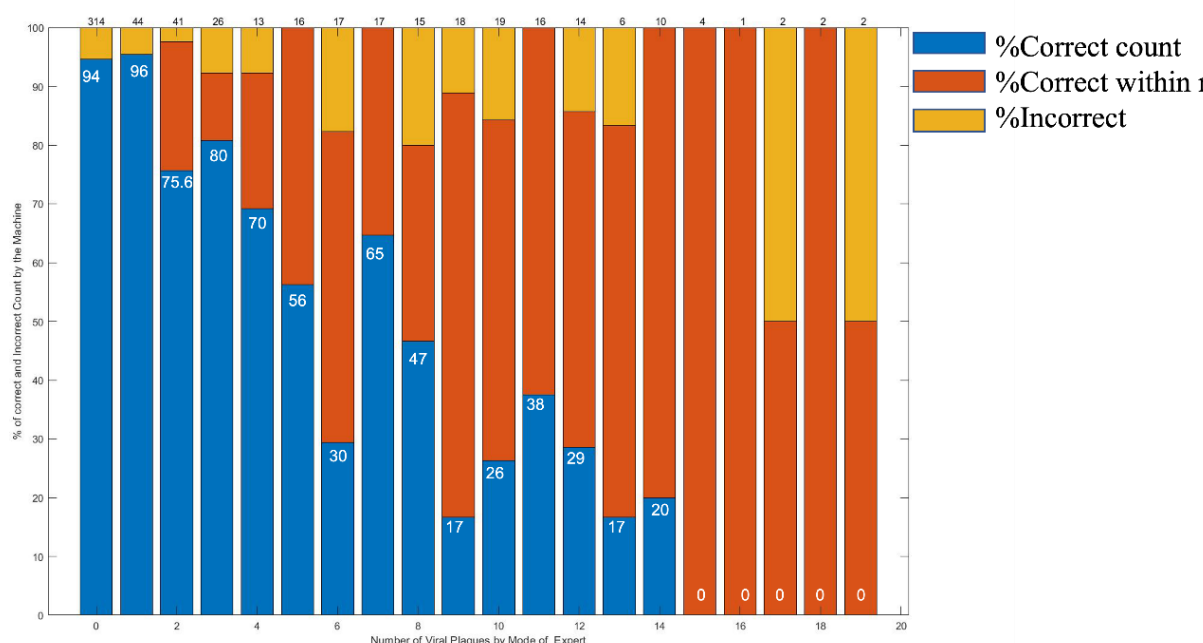
จากหัวข้อ 5.3.2 ถึง หัวข้อ 5.3.4 เป็นการนำผลลัพธ์การนับพลาอัตโนมัติ จากโปรแกรมนับพลาอัตโนมัติ เทียบกับค่าที่ถูกต้องจากค่าอ้างอิงที่ผู้เชี่ยวชาญกำหนดเกณฑ์การนับที่ถูกต้องไว้ เพื่อมาวิเคราะห์ แสดงผลในรูปแบบต่างๆไม่ว่าจะเป็น Box Plot เพื่อแสดง ผลลัพธ์ในรูปแบบกราฟและยังแสดงให้เห็นถึงความถูกต้องในการนับ ระบุว่าผลการนับพลาของไวรัสจากผู้เชี่ยวชาญเทียบกับผลการนับจากโปรแกรมนับพลาของไวรัสอัตโนมัติ ในกระบวนการ Bland – Altman ในหัวข้อนี้จะเป็นการแจกแจงผลการนับอย่างละเอียด ว่า ผลการนับในแต่ละคลาสเตอร์ ในตารางที่ 1

ตารางที่ 1 แสดงผลลัพธ์การนับพลาของไวรัสอัตโนมัติ

ค่าอ้างอิง	จำนวนภาพ	จำนวนภาพที่ เครื่องนับถูก ในค่าอ้างอิง	จำนวนภาพที่เครื่อง นับผิด ในค่าอ้างอิง	เปอร์เซ็นต์การนับ ถูกต้องในค่าอ้างอิง
0	318	301	17	94.542
1	44	42	2	95.454
2	41	40	1	97.560
3	26	24	4	92.307
4	13	12	4	92.307
5	16	16	7	100
6	17	14	3	82.352
7	17	17	0	100
8	15	12	3	80
9	18	16	2	88.888
10	19	16	4	84.210
11	16	16	0	100
12	14	12	2	85.714
13	6	5	1	83.333
14	10	10	0	100
15	4	4	0	100
16	6	6	0	100

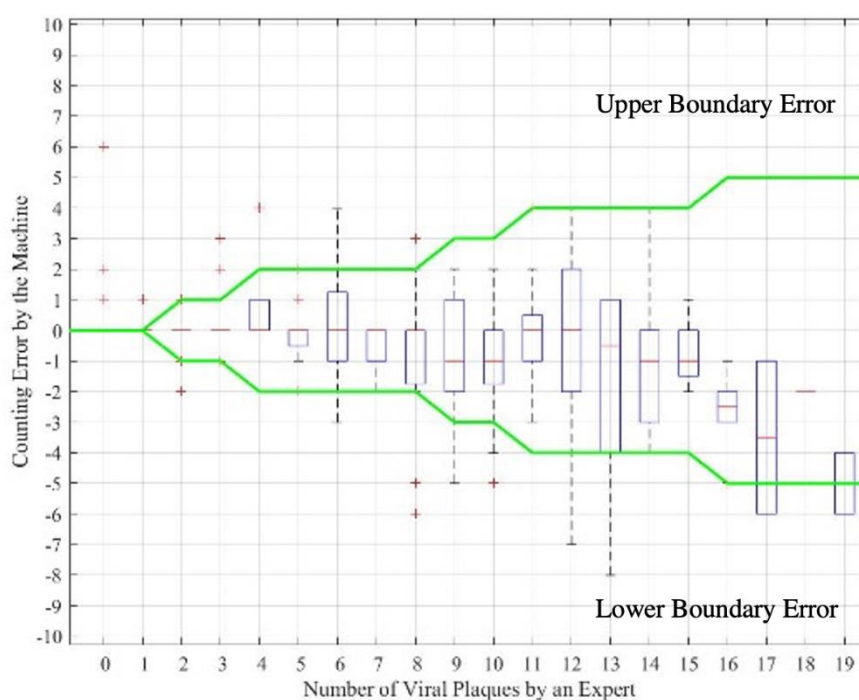
ค่า อ้างอิง	จำนวนภาพ	จำนวนภาพที่ เครื่องนับถูก ในค่าอ้างอิง	จำนวนภาพที่เครื่อง นับผิด ในค่าอ้างอิง	เปอร์เซ็นต์การนับ ถูกต้องในค่าอ้างอิง
17	2	1	1	50
18	2	2	0	100
19	2	1	1	50
รวม	606	567	52	93.5643 \approx 94%

จากตารางที่ 1 แสดงผลการนับพลาของไวรัสจากรูปภาพเพลาการทดลองทั้ง 606 ซึ่งทำการตรวจผลโดยผู้เชี่ยวชาญ สามารถสรุปได้ว่า นับถูกต้อง 567 รูปตามเกณฑ์การนับในหัวข้อ 5.1.1 นับผิดพลาด 52 รูป คิดเป็นเปอร์เซ็นต์ความถูกต้องอยู่ที่ 93.56% หรือ ประมาณ 94% โดยจากค่าผลลัพธ์สามารถสรุปได้ว่าการพัฒนาโปรแกรมระบบระเบียบวิธีการนับพลาของไวรัสนั้นมีประสิทธิภาพที่เพิ่มขึ้น และเป็นที่ยอมรับ



รูปที่ 81 แสดงผลลัพธ์ของโปรแกรมการนับพลาอัตโนมัติ ความถูกต้อง ความถูกต้องในช่วงอ้างอิง และ ค่าที่นับผิด

เพื่อให้การแสดงผลชัดเจนยิ่งขึ้นได้ทำการสร้าง Box plot ในการแสดงให้เห็นถึงผลลัพธ์การนับพลาของไวรัสจากโปรแกรมอัตโนมัติ โดยนำข้อมูลจากตารางที่ 1 มาสร้างกราฟ โดยจากรูปที่ 81 แสดงถึงผลการนับที่ถูกต้อง ผลการนับที่ถูกต้อง เป็นสีฟ้ารวมกับสีแดง และ ค่าที่นับผิด โดยจะแสดงเป็น เหลือง ตามลำดับ จาก Box plot เมื่อจำนวนพลาของไวรัสที่เพิ่มขึ้น ความถูกต้อง ก็สามารถนับได้ถูกต้องตามค่าอ้างอิงที่กำหนดไว้ โดยสามารถแสดงผลให้ชัดเจนยิ่งขึ้นในกราฟ รูปที่ 82 ดังนี้



รูปที่ 82 แสดงผลการนับพลาของไวรัสกับขอบเขตความถูกต้อง

จากรูป 82 กราฟเส้นสีเขียวแสดงให้เห็นถึงช่วงความถูกต้องตามเกณฑ์ในหัวข้อ 5.1.1 โดยนำผลการนับ พลาของไวรัสจากโปรแกรมนับพลาของไวรัสอัตโนมัติมาใส่ลงไปในกราฟ โดยนำค่า Outliner จากหัวข้อที่ 5.3.2 ว่าผลการนับนั้นสามารถอยู่ในค่าอ้างอิงที่มีความถูกต้องในทุกช่วง โดยผลการนับมีค่า Quartile ที่ หนึ่ง สอง และ สาม อยู่ในขอบเขตความถูกต้องทั้งหมด ในทุกค่าอ้างอิง 0-19 โดยจากงานวิจัยก่อนหน้า [31] ซึ่งผลลัพธ์จะออกจากเส้นค่าความถูกต้องเมื่อจำนวนพลาของไวรัสเพิ่มมากขึ้น แต่งานวิจัยนี้สามารถพัฒนาให้กระบวนการนับอัตโนมัติอยู่ในขอบเขตความถูกต้องได้ทั้งหมด

5.3.5 ผลการตรวจสอบภาพถ่ายว่าภาพนั้น เข้าเกณฑ์การนับได้หรือไม่

เนื่องจากการทดลองการทำ Plaque Assays (ระบบระเบียบวิธีการนับพลาควาของไวรัส) นั้น ประกอบไปด้วยหุ้การทดลองที่สารละลายทดลองเยอะเกินไป หรือ ไม่มีการทดลองในหุ้การนั้น จึงทำให้หุ้การทดลองนั้นไม่มี พลาควาของไวรัสอยู่ จึงต้องทำการตรวจสอบว่าแต่ละภาพสามารถนำมา วิเคราะห์ผลได้หรือไม่ อีกอย่างหนึ่งคือ ภาพที่มีจำนวนพลาควาของไวรัสมากเกินไป ทางผู้วิจัยได้ทำการ ปรึกษา กับผู้เชี่ยวชาญ อาจารย์ และ ผู้ทำงานวิจัยก่อนหน้า สามารถสรุปได้ว่าจำนวนพลาควาที่เพิ่มขึ้น ความผิดพลาดก็จะเพิ่มมากขึ้นเช่นกัน ไม่ว่าจะเป็นการนับจากผู้เชี่ยวชาญก็ตาม จึงทำการทดลอง Plaque Assays ให้จำนวนคลาสเตอร์พลาควาของไวรัสนั้นมีค่ามากที่สุดที่ 19 คลาสเตอร์ก็เพียงพอในการ นำผลลัพธ์จากโปรแกรมการนับพลาควาของไวรัส ไปศึกษาต่อ โดยงานวิจัยนี้ กำหนดให้จำนวน พลาควาของไวรัสในรูปภาพนั้นมีพื้นที่มากกว่า 103000 พิกเซลล์ หรือ มากกว่า 19 คลาสเตอร์นับไม่ได้ และ ให้พลาควาของไวรัสมีที่เล็กที่สุดอยู่มีพื้นที่ 110 พิกเซลล์โดยนำผลที่โปรแกรมแสดงผลว่าหรือนับไม่ ได้มาคำนวณหา Sensitivity Specificity Accuracy และ Precision ของโปรแกรม

โดยเริ่มจากการตรวจสอบว่าภาพถ่ายไวรัสในเพทลนั้นเข้าเกณฑ์การนับได้เพื่อนำไปเข้าสู่ โปรแกรมกระบวนการนับพลาควาของไวรัสหรือไม่(เพทลเปล่าและจำนวนพลาควาของไวรัสมากเกินไป)ซึ่ง ในโปรแกรมจะแสดงผลว่า N/A โดย จากหัวข้อ 5.1 จำนวนรูปภาพทั้งหมด 960 รูป 10 เพทลการ ทดลอง

ตารางที่ 2 ตรวจสอบรูปภาพนั้นว่าเข้าเกณฑ์หรือไม่

โปรแกรม \ คน	จำนวนรูปที่คนนับว่าผ่านเกณฑ์	จำนวนรูปที่คนนับว่าไม่ผ่านเกณฑ์
จำนวนที่โปรแกรมนับได้ว่าผ่านเกณฑ์	606	2
จำนวนที่โปรแกรมนับได้ว่าไม่ผ่านเกณฑ์	3	349

กำหนด : True Positive (TP) เป็น จำนวนที่โปรแกรมและคนนับได้ว่าผ่านเกณฑ์

True Negative (TN) เป็น จำนวนที่โปรแกรมและคนนับได้ว่าไม่ผ่านเกณฑ์

False Negative (FN) เป็น จำนวนที่โปรแกรมนับได้ว่าไม่ผ่านเกณฑ์และจำนวนรูปที่คนนับว่าผ่านเกณฑ์

False Positive (FP) เป็น จำนวนที่โปรแกรมนับได้ว่าไม่ผ่านเกณฑ์และจำนวนรูปที่คนนับว่าไม่ผ่านเกณฑ์

โดย ได้ทำการคำนวณค่า Sensitivity Specificity Accuracy Precision ของโปรแกรม และค่า Negative Predictive Value หรือ (NPV) และ Positive predictive Value หรือ (PPV) ตามสมการการคำนวณดังนี้

$$\text{Sensitivity} = TP / (TP + FN) = 606 / (606+3) = 0.9951 \quad (99.51\%)$$

$$\text{Specificity} = TN / (FP + TN) = 349 / (5+349) = 0.9943 \quad (99.43\%)$$

$$\text{Accuracy} = (TP + TN) / (P + N) = (606+349) / (606+3+5+349) = 0.9948 \quad (99.48\%)$$

$$\text{Precision} = \text{Positive predictive Value} = TP / (TP + FP) = 606 / (606+5) = 0.9967 \quad (99.67\%)$$

$$\text{Negative Predictive Value} = TN / (TN + FN) = 349 / (349+3) = 0.9915 \quad (99.15\%)$$

จากการคำนวณ ค่า Precision และค่า Positive predictive Value โปรแกรมการนับพลาควไวรัสอัตโนมัติสามารถตรวจจับการมีพลาควไวรัสในเพลทและตรวจสอบจำนวนไวรัสไม่เกิน 19 คลาสเตอร์ หรือ ตรวจสอบว่าเพลทการทดลองนั้นสามารถนับได้หรือไม่ มีความน่าเชื่อถือถึง 99.67% โดยสามารถทำการตรวจพบเพลทเปล่าและเพลทที่ไม่สามารถนับได้ ถึง 99.51% โดยโปรแกรมมีความเที่ยงตรงเท่ากับ 99.48% ในการคัดกรอง

5.4 การประเมินความพึงพอใจจากผู้ใช้งานโปรแกรมนับพลาควอัตโนมัติ

งานวิจัยนี้ได้พัฒนาโปรแกรมการพัฒนาส่วนต่อประสานโปรแกรมการนับพลาควอัตโนมัติ หรือ Graphic User Interface (GUI) เพื่อช่วยแบ่งเบาภาระผู้ทำการศึกษา หรือผู้เชี่ยวชาญทางไวรัสวิทยา ในการใช้โปรแกรมนับพลาควอัตโนมัติ ได้ดียิ่งขึ้น จากงานวิจัยก่อนหน้านี้ [32] ซึ่งจะเห็นได้ว่าในงานวิจัยก่อนหน้านี้ การใช้โปรแกรมนับพลาควอัตโนมัติ จะมีกระบวนการซับซ้อนหลายขั้นตอน ทำให้เสียเวลาในการนับเป็นอย่างมาก กว่าที่จะได้ผลลัพธ์ของการนับพลาควไวรัสแต่ละเพลทการทดลอง จึงได้มีวัตถุประสงค์การพัฒนางานวิจัยนี้ให้เกิดขึ้น โดยอิงจากปัญหาของผู้ใช้งานเป็นหลัก ว่าเกิดปัญหาอะไรในการทดลอง เพื่อนำปัญหาเหล่านั้นมาปรับปรุง และแก้ไข ให้ใช้งานได้ง่าย เป็นไปตามวัตถุประสงค์ของงานวิจัย และได้มีการเพิ่มเติมฟังก์ชันการใช้งานที่จำเป็นต่อการนับพลาควไวรัส ได้แก่ ความถูกต้องของโปรแกรมที่มีความเที่ยงตรงมากขึ้นถึง 6% จากงานวิจัยก่อนหน้านี้ [31] และมีความเสถียรภาพและมีความสม่ำเสมอ อีกทั้งสามารถปรับแก้ไขค่า Parameter ต่างๆ และการแสดงผลที่ชัดเจนควบคุมโปรแกรมได้ง่าย

โดยเพื่อต้องการผลความพอใจจากผู้ใช้ผู้ใช้งานจริง จึงได้ทำการติดตั้งเครื่องถ่ายภาพและโปรแกรมการนับพลาควไวรัสอัตโนมัติ ที่ศูนย์วิจัยทางไวรัสวิทยา คณะแพทยศาสตร์จุฬาลงกรณ์มหาวิทยาลัย เพื่อสอบถามความเห็นจากผู้ใช้งานโปรแกรมจริง พบว่า

เครื่องถ่ายภาพที่ใช้สามารถถ่ายภาพเพลทการทดลองได้อย่างชัดเจนและถูกต้องในการบันทึกภาพ นอกจากนี้ยังมีความสะดวกในการวางเพลทของไวรัสบนเครื่องถ่ายภาพ อย่างไรก็ตาม ยังมีความต้องการปรับปรุงในเรื่องของระยะเวลาในการถ่ายภาพที่เร็วขึ้นและความสะดวกสบายในการติดตั้งเครื่องถ่ายภาพ เพื่อให้การใช้งานมีประสิทธิภาพมากยิ่งขึ้น

ส่วนโปรแกรมการนับพลาควไวรัสอัตโนมัติ ส่วนต่อประสานผู้ใช้งาน (GUI) สามารถแสดงผลการนับได้อย่างถูกต้องและชัดเจน มีรูปแบบที่เข้าใจง่ายในการนำเสนอข้อมูลและแสดงผลลัพธ์ ระบบการใช้งานมีการออกแบบให้สามารถปรับค่าพารามิเตอร์ต่างๆ ได้ง่าย และมีการวางตำแหน่งฟังก์ชันการใช้งานที่เหมาะสม อย่างไรก็ตาม ควรพิจารณาปรับปรุงเพื่อลดเวลาในการรันโปรแกรมที่ใช้เวลานาน และเพิ่มความเร็วในการทำงาน ซึ่งอาจจะเกิดจากการปรับปรุงขั้นตอนในการทำงานของโปรแกรมเพื่อให้สามารถทำงานได้รวดเร็วขึ้น

โดยรวมแล้ว โปรแกรมการนับผลของไวรัสอัตโนมัติที่ศูนย์วิจัยไวรัสวิทยาคณะแพทยศาสตร์
จุฬาลงกรณ์มหาวิทยาลัยพัฒนาขึ้นมานั้น ได้รับความพึงพอใจจากผู้ใช้งานโดยรวมโดยมีผลลัพธ์ที่ดี



บทที่ 6

สรุปผลงานวิจัย

งานวิจัยนี้ได้ทำการพัฒนาโปรแกรมการนับพลาของไวรัสอัตโนมัติ และการปรับปรุงความแม่นยำให้กับระบบ ซึ่งรวมไปถึง ความเสถียรภาพ ความแม่นยำ และได้ทำการสร้างส่วนต่อประสานกับผู้ใช้งาน Graphic User Interface (GUI)

กระบวนการนับพลาของไวรัส หรือ Plaque Assays นั้นเป็นกระบวนการที่ยาก และซับซ้อน จำเป็นที่ต้องใช้ความเชี่ยวชาญ ในการทำการทดลองในแต่ละครั้ง งานวิจัยนี้จะช่วยลดภาระงานสำหรับผู้เชี่ยวชาญที่ ทำการทดลองการนับพลาของไวรัส ด้วยการแสดงผลภาพการนับ จำนวนคลาสเตอร์พลาของไวรัส ผ่านส่วนต่อประสานงานโปรแกรมการนับพลาอัตโนมัติ หรือ Graphic User Interface (GUI) ในส่วนของการปรับปรุงความแม่นยำให้กับระบบนั้น เพื่อต้องการตรวจจับคลาสเตอร์ของไวรัส และทำการนับจำนวนคลาสเตอร์บน 96 – Well plate ให้ถูกต้องมากยิ่งขึ้น ได้ใช้ทฤษฎีการประมวลผลภาพ หรือ Image Processing Method สำหรับการแยกคลาสเตอร์ พลาของไวรัสออกจากกัน และ นับพลาของไวรัสที่ไม่ซ้อนทับกัน เช่น กระบวนการ Thresholding และ กระบวนการ Morphology เป็นต้น มากไปกว่านั้น ได้นำกระบวนการแมชชีนเลิร์นนิง (Machine Learning) ชนิด K mean clustering เพื่อเพิ่มความแม่นยำ ในการแยกคลาสเตอร์ พลาของไวรัส ที่มีการซ้อนทับกันออกจากกัน

งานวิจัยนี้ได้ทำการปรับโครงสร้าง พัฒนา Algorithm และ แก้ไข กระบวนการต่างๆเข้าไปในโปรแกรมการนับพลาอัตโนมัติ ที่ถูกสร้างบนโปรแกรม Mv tech Halcon และ ทำการปรับโครงสร้างของโปรแกรมโดยมีความแปลกใหม่อยู่ที่การผสมผสานการนับพลาของไวรัสอัตโนมัติจากสองกระบวนการ Image Processing และ Machine Learning มีขั้นตอนดังนี้ (1) นำภาพถ่ายเข้ามา ซึ่งภาพถ่ายนั้น ได้มาจากเครื่องถ่ายภาพ โดยงานวิจัยนี้ได้ทำการปรับปรุงความแม่นยำ ให้กับเครื่องถ่ายภาพ โดยสร้างอุปกรณ์ป้องกันแสงจากภายนอก และ อุปกรณ์ยึดกล้องถ่ายภาพ เพื่อเพิ่มประสิทธิภาพให้กับรูปภาพ ก่อนจะเข้าสู่ขั้นตอนถัดไป (2) จัดเตรียม Region of Interest (3)ทำการ Matching (4) กระบวนการ ลดสิ่งรบกวนโดยการผ่าน Median Filter (5) แบ่งส่วนภาพด้วย Image Segmentation (6)เข้ากระบวนการ Morphology (7) เข้ากระบวนการ Binary Threshold (8) คัดแยกรูปภาพจากพื้นที่ของพลา (9) ทำการนับ จำนวนพลาของไวรัส โดยที่เหลือ หรือ พลาของไวรัสที่ซ้อนทับกัน จะทำการนับโดยกระบวนการ (10) K-mean Clustering (11) รวมผล (12) แสดงผลไปยังโปรแกรม

ผลการนับของโปรแกรมการนับพลาของไวรัสอัตโนมัติ ได้หาค่า ความถูกต้อง ความแม่นยำ ความเสถียรภาพของโปรแกรม โดยความถูกต้องของผลการนับพลาของไวรัส นั้น อยู่ในค่าอ้างอิงที่ กำหนดขึ้นโดยภาควิชาไวรัสวิทยา คณะแพทยศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย จากบทที่ 5 โดยค่าอ้างอิงนี้สามารถบอกเป็นนัยได้ว่า เมื่อจำนวนพลาของไวรัส น้อย ค่าช่วง Error gaps จะน้อย แต่เมื่อจำนวนพลาของไวรัสเพิ่มมากขึ้น ช่วง Error gap ก็จะเพิ่มขึ้นเช่นเดียวกัน

ผลการนับจากโปรแกรมการนับการพลาของไวรัสอัตโนมัติ มีความถูกต้องอยู่ที่ 94 % ซึ่งมากกว่างานวิจัยก่อนหน้านี้ ถึง 6% จากกระบวนการผลลัพธ์ Gauge R and R พบว่าผลลัพธ์จากโปรแกรมการนับพลาของไวรัส นั้นมีความเสถียรภาพและมีความสม่ำเสมอ เมื่อสอบถามความพึงพอใจจากผู้เชี่ยวชาญ ก็สามารถช่วยลดภาระงานได้จริง ไม่ว่าจะเป็นฟังก์ชันการแสดงรูปภาพที่ชัดเจน ฟังก์ชันการใช้งานแยกภาพถ่ายที่นับไม่ได้ ออก เพื่อนำผลไปทำการตรวจสอบได้ไวขึ้น และ ผลจากกราฟเมื่อโปรแกรมเสร็จสิ้น ก็ขอนำข้อมูลจากโปรแกรมการนับพลาของไวรัสอัตโนมัติ ไปศึกษาต่อไปได้ อย่างคร่าว

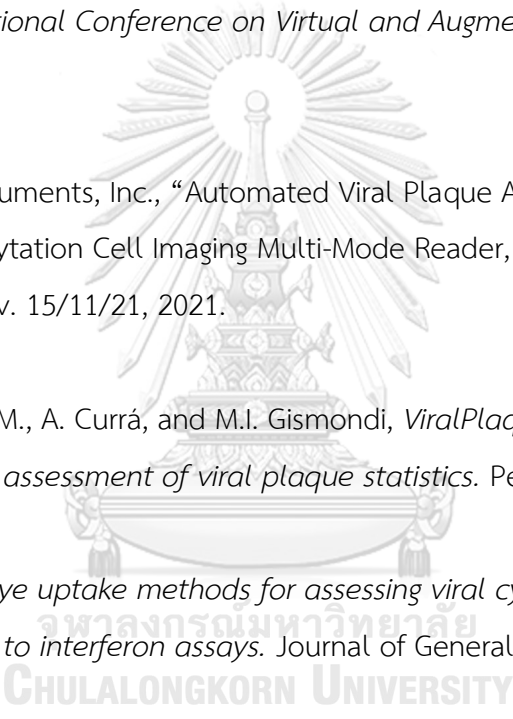
6.1 ปัญหาที่พบ

จากการทำงานวิจัยกระบวนการประมวลผลภาพนั้นภาพถ่ายมีความสำคัญต่อการประมวลผลเป็นอย่างมาก จัดอุปกรณ์เครื่องถ่ายภาพ และ แสง ในการถ่ายภาพเพลดการทดลองแต่ละครั้งควรจะทำในสิ่งแวดล้อมเดียวกัน และ เพลดการทดลองบางเพลดที่มีร่องรอยขีดข่วนเนื่องจากการจัดเก็บส่งผลให้ภาพนั้นไม่มีความคมชัดได้ จึงต้องทำการจัดเก็บเพลดการทดลองให้ดี อีกหนึ่งปัญหาที่พบคือ เวอร์ชันของโปรแกรมที่นำมาพัฒนาต่อจากงานวิจัยก่อนหน้านี้ [31-32] มีความต่างกันพอสมควรเนื่องจากการอัปเดตของโปรแกรมในแต่ละปีจึงต้องทำการปรับแต่งคำสั่งในโปรแกรมพอสมควร

6.2 แนวทางในการพัฒนาต่อ

จากงานวิจัยนี้สามารถพัฒนาต่อโดยการเพิ่มเติมฟังก์ชันการใช้งานเข้าไปในโปรแกรมการนับพลาของไวรัสและอาจจะสามารถพัฒนาให้โปรแกรมสามารถนับพลาของไวรัสมากกว่า 19 คลาสเตอร์ได้หรือสามารถนับเชื้อโรคชนิดอื่นได้ และยังสามารถนำ Algorithm 2ที่ได้ทำการพัฒนามาบางส่วนแล้วมาพัฒนาต่อได้ รวมไปถึงกระบวนการ Silhouette เพื่อพัฒนากระบวนการ K-mean Clustering ได้

บรรณานุกรม

- 
- [1] Delbruck, M., *The growth of bacteriophage and lysis of the host*. The journal of general physiology, 1940. 23(5): p. 643-660.
- [2] Premsattham, P., et al. *Development of Automated Platform for Image Capturing and Counting Algorithm for Viral Plaque*. in *Proceedings of the 2019 3rd International Conference on Virtual and Augmented Reality Simulations*. 2019.
- [3] BioTek Instruments, Inc., “Automated Viral Plaque Assay Workflow Using the Cytation Cell Imaging Multi-Mode Reader,” Application Bulletin, Rev. 15/11/21, 2021.
- [4] Cacciabue, M., A. Currá, and M.I. Gismondi, *ViralPlaque: a Fiji macro for automated assessment of viral plaque statistics*. PeerJ, 2019. 7: p. e7729.
- [5] Finter, N., *Dye uptake methods for assessing viral cytopathogenicity and their application to interferon assays*. Journal of General Virology, 1969. 5(3): p. 419-427.
- [6] P. Bhattarakosol, C. Yoosook, A. Cross, “Intratypic variation of herpes simplex virus type 2 isolates detected by monoclonal antibodies against viral glycoproteins,” Arch. Virol, vol. 115, pp. 89–100, 1990.
- [7] S. Boonyasuppayakorn, A. Suroengrit, P. Srivarangkul, W. Yuttithamnon, S. Pankaew, T. Saelee, E. Prompetchara, S. Salakij, P. Bhattarakosol, “Simplified dengue virus microwell plaque assay using an automated quantification program,” Journal of Virological Methods, vol. 237, pp. 25–31, 2016.

- [8] K. Sullivan, J. Kloess, C. Qian, D. Bell, A. Hay, Y.P. Lin, Y. Gu, “High throughput virus plaque quantitation using a flatbed scanner,” *Journal of Virological Methods*, Vol. 179, no. 1, pp. 81-89, 2012.
- [9] Cellular Technology Limited (CTL), ImmunoSpot CLT S6 Analyzers, <http://www.immunospot.com> (accessed 15 November 2021)
- [10] S.A. Smith, R. deAlwis, N. Kose, A.P. Durbin, S.S. Whitehead, A.M. deSilva, et al., “Human monoclonal antibodies derived from memory B cells following live attenuated dengue virus vaccination or natural infection exhibit similar characteristics,” *J Infect Dis.*, vol. 207, pp. 1898– 1908, 2013.
- [11] S. Sukupolvi-Petty, J.D. Brien, S.K. Austin, B. Shrestha, S. Swayne, K. Kahle, et al., “Functional Analysis of Antibodies against Dengue Virus Type4 Reveals Strain-Dependent Epitope Exposure That Impacts Neutralization and Protection,” *J Virol.*, Vol. 87, pp. 8826–8842, 2013.
- [12] WS. Rasband, “ImageJ. In: National Institutes of Health,” Bethesda, Maryland, USA.
- [13] RCoreTeam. “R: A language and environment for statistical computing. R Foundation for Statistical Computing.” Vienna, Austria; 2016.
- [14] Z. Cai Z, N. Chattopadhyay, W.J. Liu, C. Chan, J.P. Pignol, R.M. Reilly, “Optimized digital counting colonies of clonogenic assays using ImageJ software and customized macros: comparison with manual counting,” *International Journal of Radiation Biology*, Vol. 87, pp. 1135-1146, 2011.
- [15] M. Cacciabue, A. Currá, and M.I. Gismondi, “Viral Plaque: a Fiji macro for automated assessment of viral plaque statistics,” *PeerJ*. 2019; 7 (2019) e7729 <http://doi.org/10.7717/peerj.7729>, 2019.

- [16] Q. Geissmann, "OpenCFU, a new free and open-source software to count cell colonies and other circular objects.", PLOS ONE Vol. 8, no. 2, e54072 DOI 10.1371/journal.pone.0054072, 2013.
- [17] K. Sullivan, J. Kloess, C. Qian, D. Bell, A. Hay, Y.P. Lin, Y. Gu, "High throughput virus plaque quantitation using a flatbed scanner." Journal of Virological Methods, Vol. 179, pp. 81-89 DOI 10.1016/j.jviromet.2011.10.003, 2012.
- [18] L.C. Katzelnick, et., "Viridot: An automated virus plaque (immunofocus) counter for the measurement of serological neutralizing responses with application to dengue virus," PLoS Negl Trop Dis., vol. 12, no. 10, (2018) e0006862., 2018.
- [19] M.Moorman,A.Dong,"AutomatedViralPlaqueCountingUsingImage Segmentation and Morphological Analysis," 2012 IEEE International Symposium on Multimedia 2012. pp. 157-160, 2012.
- [20] IAI Robot (Thailand) Co., Ltd., TT Gantry Type, <https://www.iairobot.co.th/en/product/table/tt.php> (accessed 30 June 2021)
- [21] AnMo Electronics Corporation, <https://www.dino-lite.com> (accessed 30 June 2021)
- [22] Sundararajan, D. (2017). Digital image processing: a signal processing and algorithmic approach, Springer.
- [23] Chankong, T., et al. (2014). "Automatic cervical cell segmentation and classification in Pap smears." Computer methods and programs in biomedicine 113(2): 539-556.
- [24] Chobngam, F., et al. (2012). Preliminary results of death cell counting based on K-mean clustering. The 5th 2012 Biomedical Engineering International Conference, IEEE.

- [25] Salem, N. M. (2014). Segmentation of white blood cells from microscopic images using K-means clustering. 2014 31st National Radio Science Conference (NRSC), IEEE.
- [26] Dabbura, I. (2018). "K-means Clustering: Algorithm, Applications, Evaluation Methods, and Drawbacks." Retrieved 25/5, 2021, from <https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a>
- [27] Kumar, S. (2020). "Silhouette Method — Better than Elbow Method to find Optimal Clusters." Retrieved 20/6, 2021, from <https://towardsdatascience.com/silhouette-method-better-than-elbow-method-to-find-optimal-clusters-378d62ff6891>.
- [28] Savkare, S. and S. Narote (2015). Blood cell segmentation from microscopic blood images. 2015 International Conference on Information Processing (ICIP), IEEE.
- [29] Salem, N. M. (2014). Segmentation of white blood cells from microscopic images using K-means clustering. 2014 31st National Radio Science Conference (NRSC), IEEE.
- [30] Guan, B. X., et al. (2013). Automatic cell region detection by k-means with weighted entropy. 2013 IEEE 10th International Symposium on Biomedical Imaging, IEEE.
- [31] G. Phanomchoeng et al., "Machine-learning-based automated quantification machine for virus plaque assay counting," PeerJ Computer Science, vol. 8, p. e878, 2022.

- [32] ป. เปรมสัต์ยธรรม, "การพัฒนอุปกรณ์ถ่ายภาพอัตโนมัติและระเบียบวิธีการนับผลาคของไวรัส," ปรินญาโท, วิศวกรรมเครื่องกล, จุฬาลงกรณ์มหาวิทยาลัย, วิทยานิพนธ์ (วศ.ม.)--จุฬาลงกรณ์มหาวิทยาลัย, 2561, 2561. [Online]. Available: <https://cuir.car.chula.ac.th/handle/123456789/74209>
- [33] R. K. Burdick, C. M. Borror, and D. C. Montgomery, Design and analysis of gauge R&R studies: making decisions with confidence intervals in random and mixed ANOVA models. SIAM, 2005.







ภาคผนวก ก.

ตารางที่ ก-1 แสดงผลการนับจากโปรแกรมนับผลาคของไวรัสอัตโนมัติเทียบกับค่าอ้างอิงของภาพ
ทั้งหมด 606 รูป

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
1	0	0
2	0	0
3	0	0
4	0	0
5	0	0
6	0	0
7	0	0
8	0	0
9	0	0
10	0	0
11	0	0
12	0	0
13	0	0
14	0	0
15	0	0
16	0	0
17	0	0
18	0	0
19	0	0
20	0	0
21	0	0
22	0	0
23	0	0
24	0	0

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
25	0	0
26	0	0
27	0	1
28	0	0
29	0	0
30	0	0
31	0	0
32	0	0
33	0	0
34	0	0
35	0	0
36	0	0
37	0	0
38	0	0
39	0	0
40	0	0
41	0	0
42	0	0
43	0	0
44	0	0
45	0	0
46	0	1
47	0	0
48	0	0
49	0	0
50	0	0

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
51	0	0
52	0	0
53	0	0
54	0	0
55	0	2
56	0	0
57	0	0
58	0	0
59	0	0
60	0	0
61	0	0
62	0	0
63	0	0
64	0	0
65	0	0
66	0	0
67	0	0
68	0	0
69	0	0
70	0	0
71	0	0
72	0	0
73	0	0
74	0	0
75	0	0
76	0	0

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
77	0	0
78	0	0
79	0	0
80	0	0
81	0	0
82	0	0
83	0	0
84	0	0
85	0	0
86	0	0
87	0	0
88	0	0
89	0	0
90	0	0
91	0	0
92	0	0
93	0	0
94	0	0
95	0	0
96	0	0
97	0	0
98	0	0
99	0	0
100	0	0
101	0	0
102	0	0

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
103	0	0
104	0	0
105	0	0
106	0	0
107	0	1
108	0	0
109	0	0
110	0	0
111	0	0
112	0	0
113	0	0
114	0	0
115	0	0
116	0	0
117	0	0
118	0	0
119	0	0
120	0	0
121	0	0
122	0	0
123	0	0
124	0	0
125	0	0
126	0	0
127	0	6
128	0	0

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
129	0	0
130	0	0
131	0	0
132	0	0
133	0	0
134	0	0
135	0	0
136	0	0
137	0	0
138	0	0
139	0	0
140	0	0
141	0	0
142	0	0
143	0	0
144	0	0
145	0	0
146	0	0
147	0	0
148	0	0
149	0	0
150	0	0
151	0	0
152	0	0
153	0	0
154	0	0

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
155	0	0
156	0	0
157	0	0
158	0	0
159	0	0
160	0	0
161	0	0
162	0	0
163	0	0
164	0	0
165	0	0
166	0	0
167	0	0
168	0	0
169	0	0
170	0	1
171	0	2
172	0	0
173	0	1
174	0	0
175	0	0
176	0	0
177	0	0
178	0	0
179	0	0
180	0	0

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
181	0	0
182	0	0
183	0	0
184	0	0
185	0	0
186	0	1
187	0	0
188	0	0
189	0	1
190	0	0
191	0	0
192	0	0
193	0	0
194	0	0
195	0	0
196	0	0
197	0	0
198	0	0
199	0	0
200	0	0
201	0	0
202	0	0
203	0	0
204	0	0
205	0	1
206	0	0

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
207	0	0
208	0	1
209	0	0
210	0	0
211	0	0
212	0	0
213	0	0
214	0	1
215	0	0
216	0	0
217	0	0
218	0	0
219	0	0
220	0	0
221	0	0
222	0	0
223	0	0
224	0	0
225	0	0
226	0	0
227	0	0
228	0	0
229	0	0
230	0	0
231	0	0
232	0	0

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
233	0	0
234	0	0
235	0	0
236	0	0
237	0	0
238	0	0
239	0	0
240	0	0
241	0	0
242	0	0
243	0	0
244	0	0
245	0	0
246	0	0
247	0	0
248	0	0
249	0	0
250	0	0
251	0	0
252	0	0
253	0	0
254	0	0
255	0	0
256	0	0
257	0	0
258	0	0

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
259	0	0
260	0	0
261	0	0
262	0	0
263	0	0
264	0	0
265	0	0
266	0	0
267	0	0
268	0	0
269	0	0
270	0	0
271	0	0
272	0	0
273	0	0
274	0	0
275	0	0
276	0	0
277	0	0
278	0	0
279	0	0
280	0	0
281	0	0
282	0	0
283	0	0
284	0	0

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
285	0	1
286	0	0
287	0	0
288	0	0
289	0	0
290	0	0
291	0	0
292	0	0
293	0	0
294	0	0
295	0	1
296	0	0
297	0	0
298	0	0
299	0	0
300	0	0
301	0	0
302	0	0
303	0	0
304	0	0
305	0	0
306	0	0
307	0	1
308	0	0
309	0	0
310	0	0

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
311	0	0
312	0	0
313	0	0
314	0	0
315	0	0
316	0	0
317	0	0
318	0	1
319	1	1
320	1	1
321	1	1
322	1	1
323	1	1
324	1	1
325	1	1
326	1	1
327	1	1
328	1	1
329	1	1
330	1	1
331	1	1
332	1	1
333	1	1
334	1	1
335	1	1
336	1	1

ลำดับที่	คำอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
337	1	1
338	1	1
339	1	1
340	1	1
341	1	1
342	1	1
343	1	1
344	1	1
345	1	1
346	1	1
347	1	1
348	1	1
349	1	2
350	1	1
351	1	2
352	1	1
353	1	1
354	1	1
355	1	1
356	1	1
357	1	1
358	1	1
359	1	1
360	1	1
361	1	1
362	1	1

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
363	2	2
364	2	2
365	2	2
366	2	2
367	2	1
368	2	1
369	2	1
370	2	1
371	2	1
372	2	2
373	2	2
374	2	0
375	2	2
376	2	2
377	2	2
378	2	2
379	2	2
380	2	2
381	2	2
382	2	2
383	2	3
384	2	2
385	2	3
386	2	2
387	2	1
388	2	2

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
389	2	2
390	2	2
391	2	2
392	2	2
393	2	2
394	2	2
395	2	3
396	2	2
397	2	2
398	2	2
399	2	2
400	2	2
401	2	2
402	2	2
403	2	2
404	3	5
405	3	3
406	3	3
407	3	2
408	3	3
409	3	6
410	3	3
411	3	3
412	3	3
413	3	3
414	3	3

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
415	3	3
416	3	2
417	3	3
418	3	3
419	3	2
420	3	3
421	3	3
422	3	3
423	3	3
424	3	3
425	3	3
426	3	3
427	3	3
428	3	3
429	3	3
430	4	4
431	4	8
432	4	4
433	4	4
434	4	4
435	4	5
436	4	4
437	4	5
438	4	4
439	4	4
440	4	4

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
441	4	4
442	4	5
443	5	5
444	5	4
445	5	5
446	5	5
447	5	4
448	5	5
449	5	6
450	5	5
451	5	7
452	5	6
453	5	4
454	5	5
455	5	3
456	5	5
457	5	5
458	5	5
459	6	5
460	6	7
461	6	7
462	6	10
463	6	5
464	6	8
465	6	7
466	6	8

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
467	6	6
468	6	5
469	6	3
470	6	9
471	6	6
472	6	6
473	6	6
474	6	6
475	6	5
476	7	7
477	7	7
478	7	6
479	7	7
480	7	5
481	7	7
482	7	7
483	7	7
484	7	7
485	7	5
486	7	6
487	7	7
488	7	7
489	7	6
490	7	6
491	7	7
492	7	7

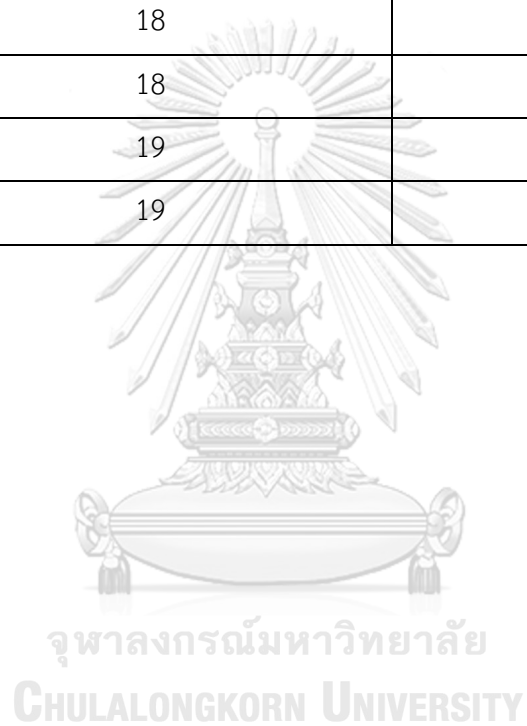
ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
493	8	10
494	8	6
495	8	11
496	8	8
497	8	8
498	8	7
499	8	8
500	8	7
501	8	8
502	8	8
503	8	8
504	8	3
505	8	6
506	8	2
507	8	8
508	9	9
509	9	10
510	9	8
511	9	8
512	9	7
513	9	6
514	9	5
515	9	4
516	9	9
517	9	8
518	9	8

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
519	9	11
520	9	10
521	9	8
522	9	10
523	9	7
524	9	9
525	9	10
526	10	9
527	10	10
528	10	11
529	10	9
530	10	9
531	10	6
532	10	12
533	10	6
534	10	8
535	10	9
536	10	9
537	10	10
538	10	12
539	10	7
540	10	10
541	10	11
542	10	10
543	10	10
544	10	5

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
545	11	10
546	11	11
547	11	8
548	11	11
549	11	11
550	11	10
551	11	9
552	11	13
553	11	15
554	11	11
555	11	11
556	11	10
557	11	13
558	11	11
559	11	9
560	11	12
561	12	12
562	12	10
563	12	7
564	12	16
565	12	16
566	12	5
567	12	13
568	12	12
569	12	15
570	12	9

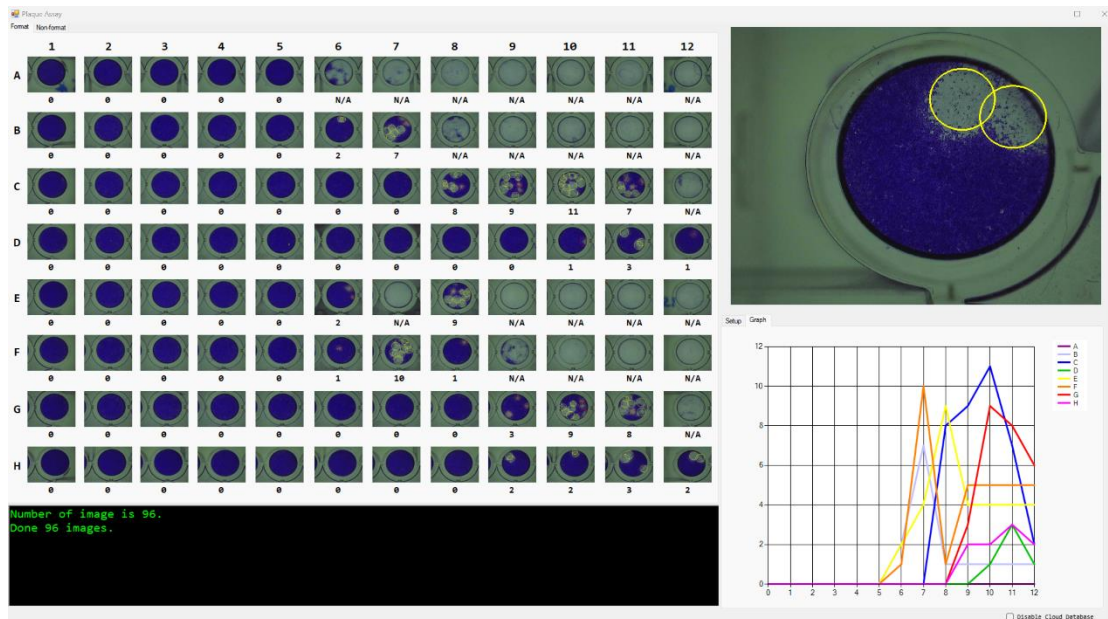
ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
571	12	14
572	12	12
573	12	12
574	12	11
575	13	9
576	13	13
577	13	5
578	13	14
579	13	12
580	13	14
581	14	11
582	14	14
583	14	11
584	14	13
585	14	13
586	14	18
587	14	10
588	14	12
589	14	14
590	14	13
591	15	14
592	15	14
593	15	13
594	15	16
595	16	15
596	16	14

ลำดับที่	ค่าอ้างอิง	ค่าที่นับโดยโปรแกรมอัตโนมัติ
597	16	14
598	16	11
599	16	13
600	16	13
601	17	11
602	17	16
603	18	16
604	18	16
605	19	15
606	19	13

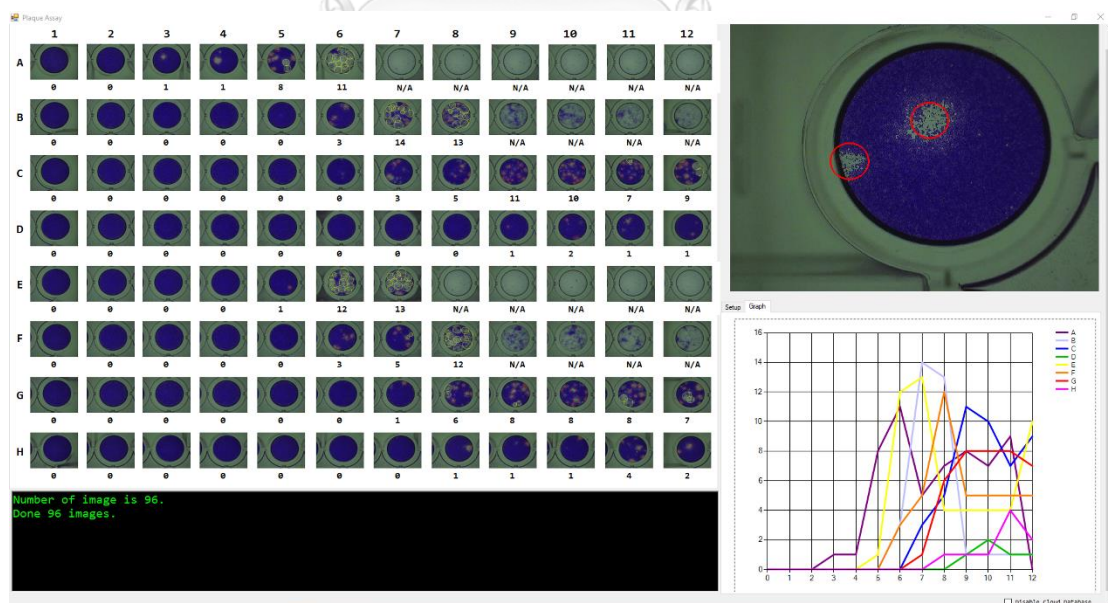


ภาคผนวก ข.

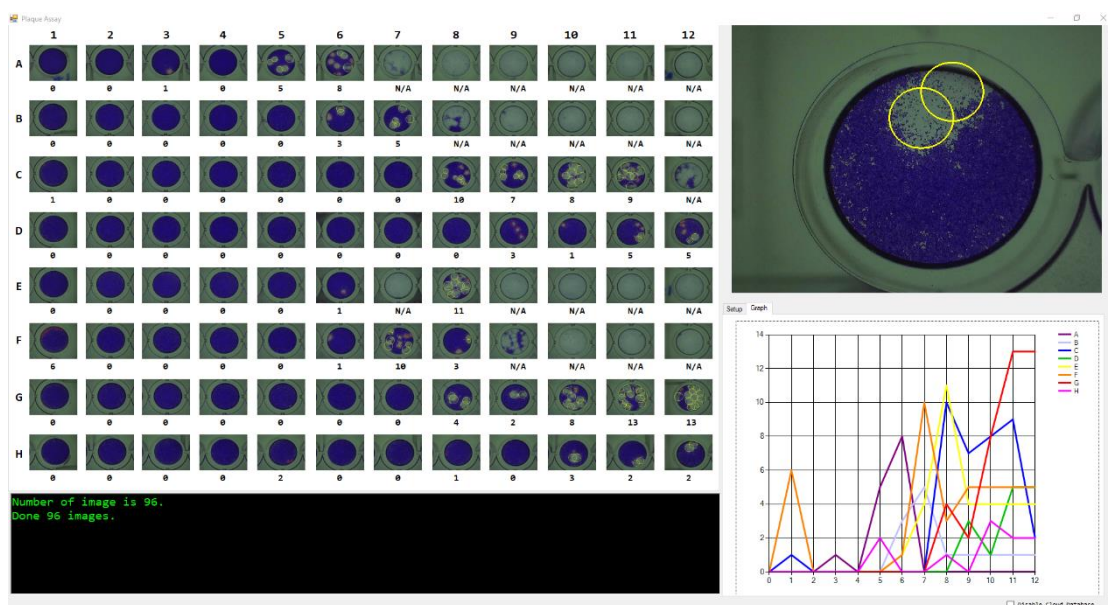
ผลการใช้งานส่วนต่อประสานต่อผู้ใช้งานของโปรแกรมนับพลาซของไวรัสอ์ตโนมิติเพื่อแสดงผลการนับของทั้ง 10 เฟลทการทดลอง



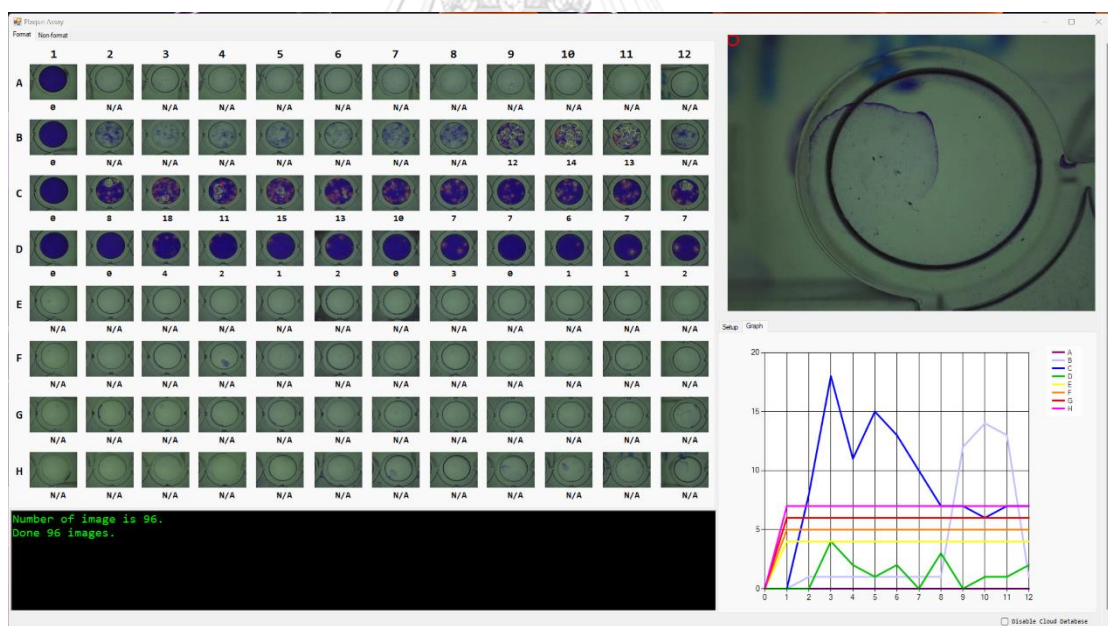
รูปที่ ข-1 แสดงผลการใช้งานโปรแกรมการนับพลาซของไวรัสอ์ตโนมิติด้วยเฟลทการทดลองที่1



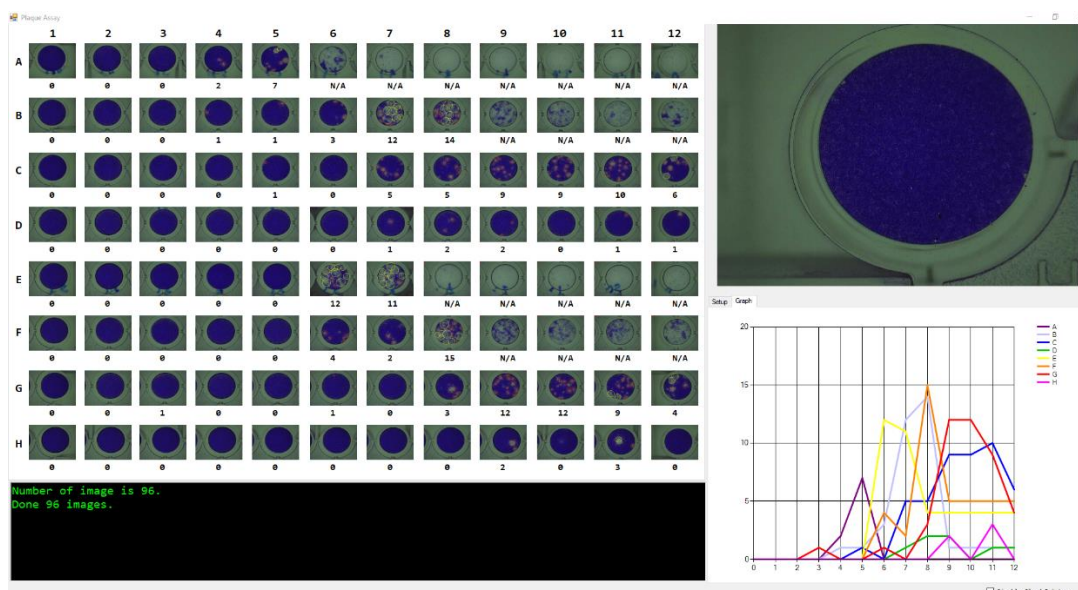
รูปที่ ข-2 แสดงผลการใช้งานโปรแกรมการนับพลาซของไวรัสอ์ตโนมิติด้วยเฟลทการทดลองที่2



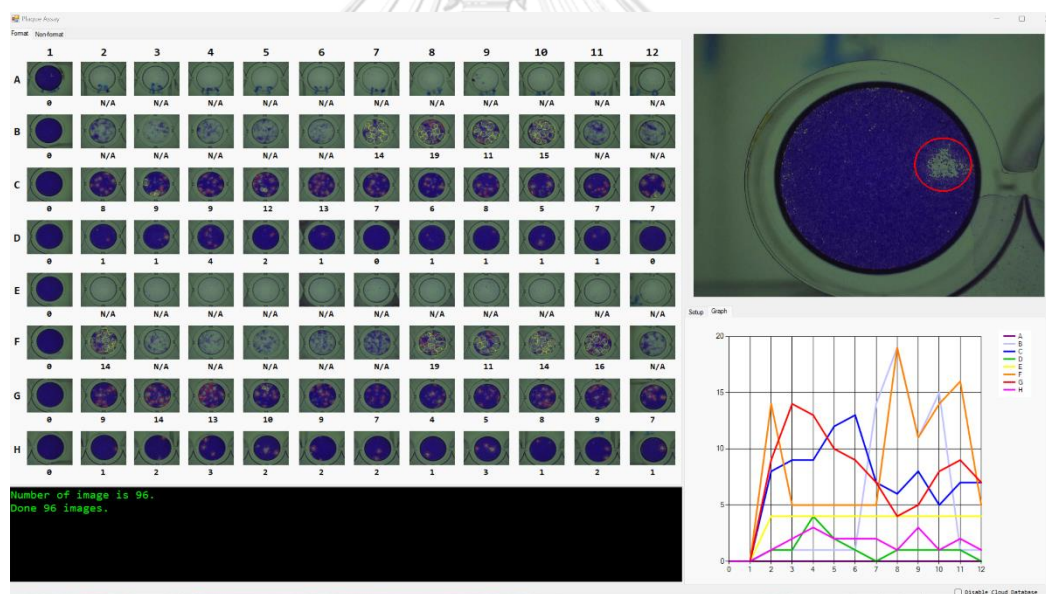
รูปที่ ข-3 แสดงผลการใช้งานโปรแกรมการนับพลาของไวรัสอิตโนมิติด้วยเพลทการทดลองที่3



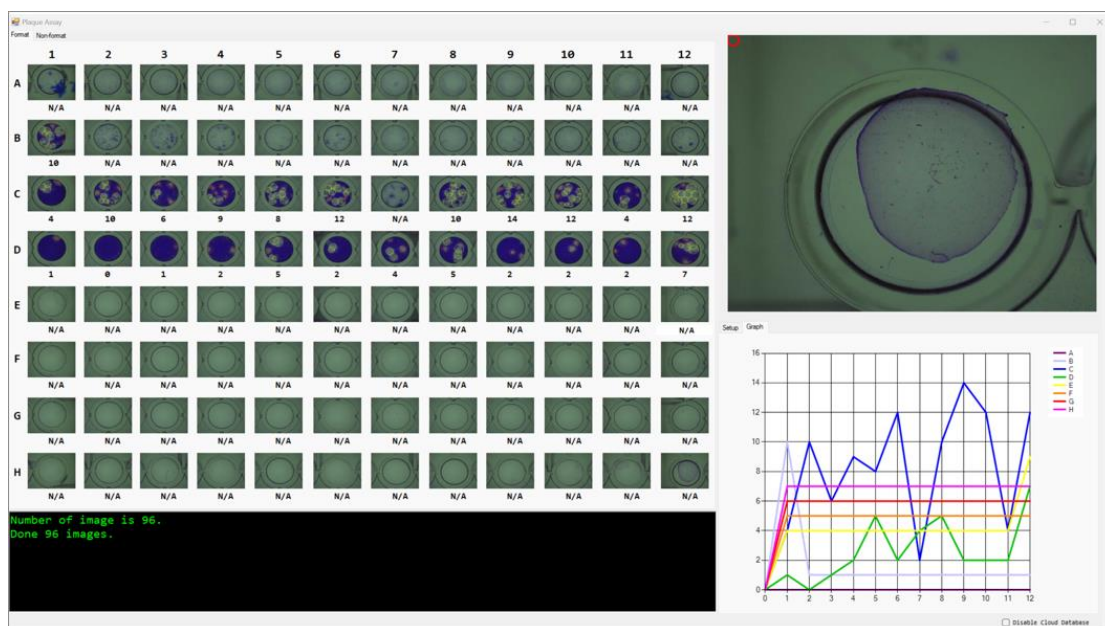
รูปที่ ข-4 แสดงผลการใช้งานโปรแกรมการนับพลาของไวรัสอิตโนมิติด้วยเพลทการทดลองที่4



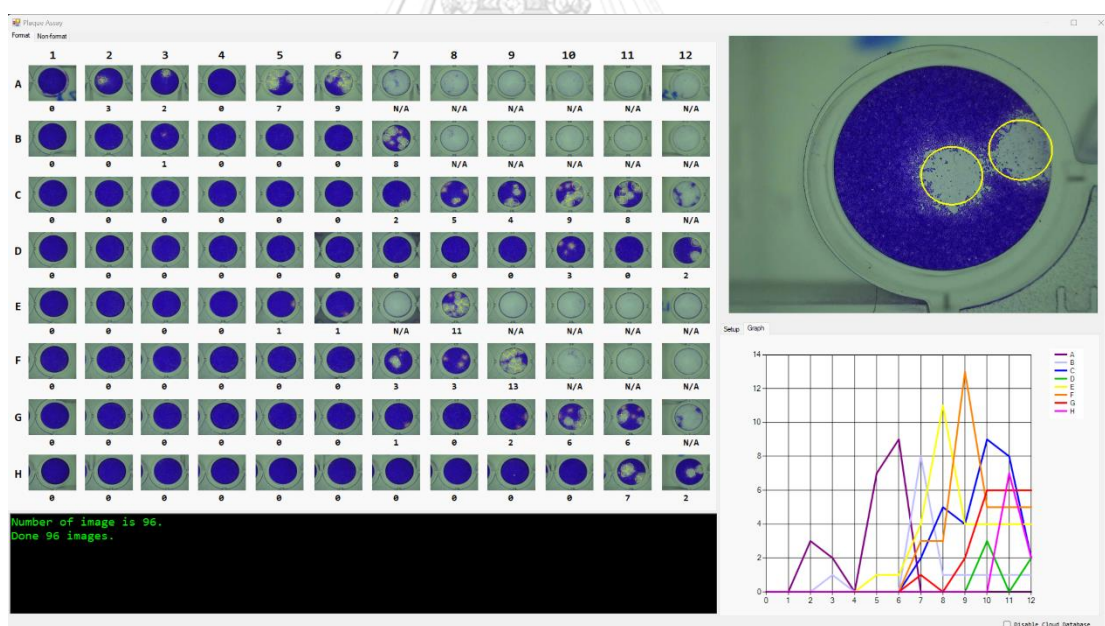
รูปที่ ข-5 แสดงผลการใช้งานโปรแกรมการนับพลาซของไวรัสฮัดโนมิติตัวด้วยเพลทการทดลองที่ 5



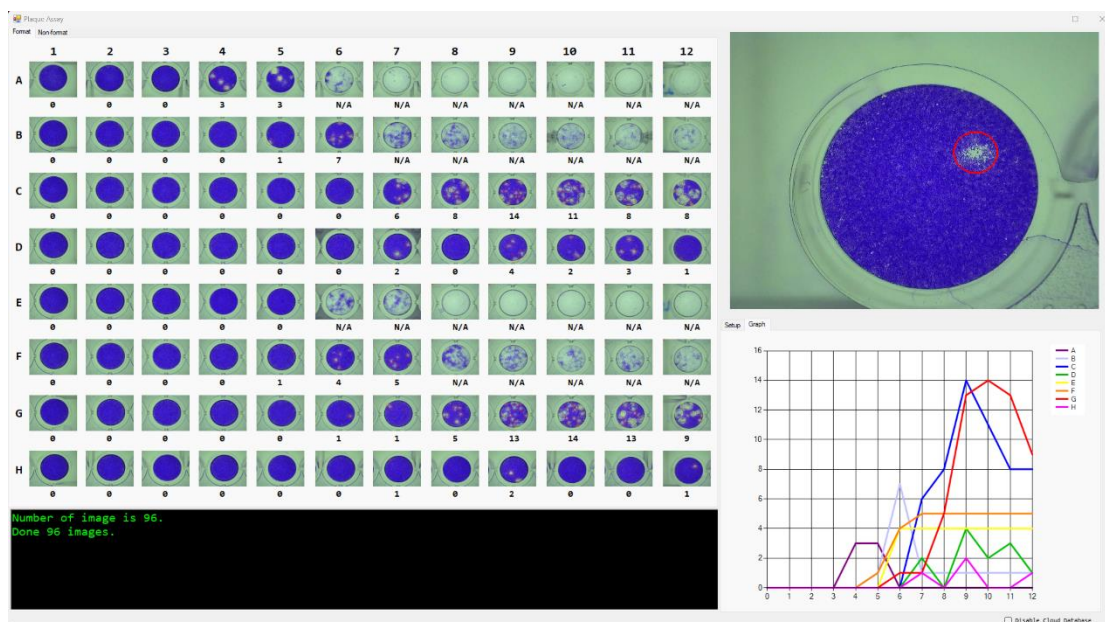
รูปที่ ข-6 แสดงผลการใช้งานโปรแกรมการนับพลาซของไวรัสฮัดโนมิติตัวด้วยเพลทการทดลองที่ 6



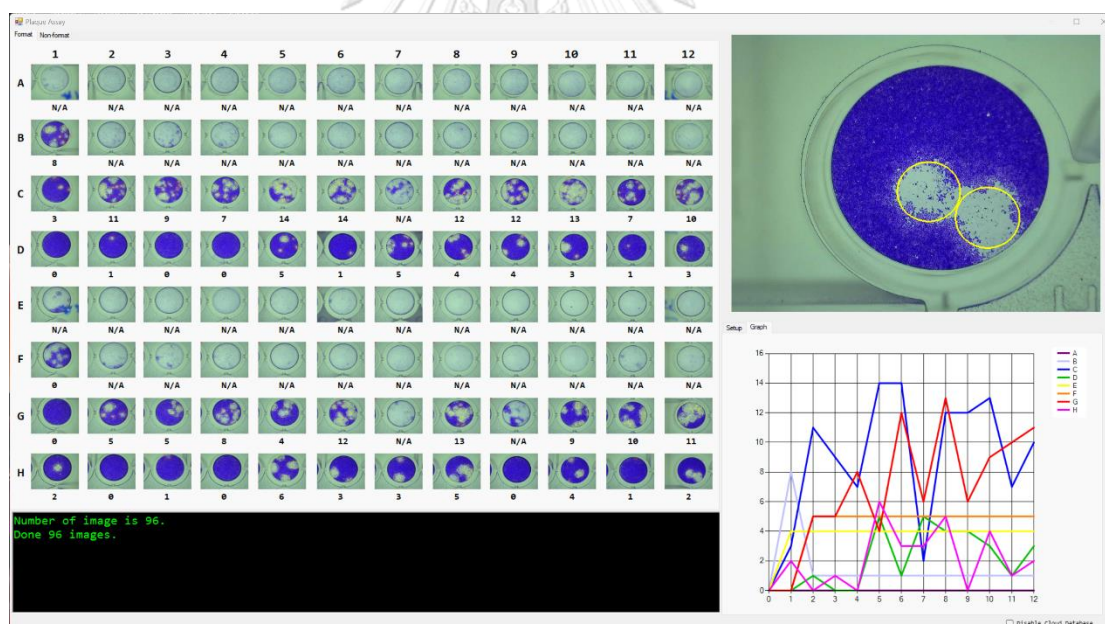
รูปที่ ข-7 แสดงผลการใช้งานโปรแกรมการนับพลาซของไวรัสฮัดโนมิตีด้วยเพลทการทดลองที่ 7



รูปที่ ข-8 แสดงผลการใช้งานโปรแกรมการนับพลาซของไวรัสฮัดโนมิตีด้วยเพลทการทดลองที่ 8



รูปที่ ข-9 แสดงผลการใช้งานโปรแกรมการนับพลาซของไวรัสอิตโนมิติด้วยเพลทการทดลองที่9



รูปที่ ข-10 แสดงผลการใช้งานโปรแกรมการนับพลาซของไวรัสอิตโนมิติด้วยเพลทการทดลองที่10



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

บรรณานุกรม



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

ประวัติผู้เขียน

ชื่อ-สกุล	ศุภณัฐ เปล่งคำ
วัน เดือน ปี เกิด	28 พฤษภาคม 2541
สถานที่เกิด	โรงพยาบาลบำรุงราษฎร์ กรุงเทพมหานคร
วุฒิการศึกษา	วศ.บ. คณะวิศวกรรมศาสตร์ ภาควิชาวิศวกรรมเครื่องกล มหาวิทยาลัย เกษตรศาสตร์ วิทยาเขต บางเขน
ที่อยู่ปัจจุบัน	357/1 ต.หน้าเมือง อ.เมือง จ.ปราจีนบุรี 25000
ผลงานตีพิมพ์	1. Development of an Automated Viral Plaque Assay Program and ELISpot Counting Program on 96- Well Plate Platform · Proc. of the International Conference on Electrical, Computer and Energy Technologies (ICECET 2022) 20- 22 July 2022, Prague-Czech Republic 2. [1] S. Plengkham, N. Busyakanistha, N. Promsuwan, N. Santiudommongkol, and R. Chanchaen, "Developing of Automatic Gasket Gluing Robot for a Tractor Company," in 2023 9th International Conference on Mechatronics and Robotics Engineering (ICMRE), 2023: IEEE, pp. 19-23.
รางวัลที่ได้รับ	Gold Medal Award 2022 Japan Design, Idea & Invention Expo. July1-3,2022