

การประยุกต์ใช้โครงข่ายประสาทเพื่อตรวจจับเซลล์หลายชนิดผ่านกล้องจุลทรรศน์เชิงซ้อนด้วยตัวรับรู้

ภาพ



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

ปีการศึกษา 2562

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

THE APPLICATION OF NEURAL NETWORK TO DETECT MULTIPLE CELLS VIA  
COMPOUND MICROSCOPE WITH IMAGE SENSOR



A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Master of Engineering in Electrical Engineering

Department of Electrical Engineering

FACULTY OF ENGINEERING

Chulalongkorn University

Academic Year 2019

Copyright of Chulalongkorn University

หัวข้อวิทยานิพนธ์	การประยุกต์ใช้โครงข่ายประสาทเพื่อตรวจจับเซลล์หลายชนิดผ่านกล้องจุลทรรศน์เชิงซ้อนด้วยตัวรับรู้ภาพ
โดย	นายณัฏกร เกษมสำราญ
สาขาวิชา	วิศวกรรมไฟฟ้า
อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก	ผู้ช่วยศาสตราจารย์ ดร.สุรีย์ พุ่มรินทร์
อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม	รองศาสตราจารย์ ดร.นรรธิน วัฒนมงคล

---

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้หัวข้อวิทยานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของ  
ของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

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

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ  
(รองศาสตราจารย์ ดร.วันเฉลิม โปธา)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก  
(ผู้ช่วยศาสตราจารย์ ดร.สุรีย์ พุ่มรินทร์)

..... อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม  
(รองศาสตราจารย์ ดร.นรรธิน วัฒนมงคล)

..... กรรมการภายนอกมหาวิทยาลัย  
(รองศาสตราจารย์ ดร.สัญญา มิตรเอม)

ณัฏกร เกษมสำราญ : การประยุกต์ใช้โครงข่ายประสาทเพื่อตรวจจับเซลล์หลายชนิดผ่าน  
กล้องจุลทรรศน์เชิงซ้อนด้วยตัวรับรูปภาพ. ( THE APPLICATION OF NEURAL  
NETWORK TO DETECT MULTIPLE CELLS VIA COMPOUND MICROSCOPE  
WITH IMAGE SENSOR) อ.ที่ปรึกษาหลัก : ผศ. ดร.สุรีย พุ่มรินทร์, อ.ที่ปรึกษาร่วม :  
รศ. ดร.นรรรัตน์ วัฒนมงคล

สำหรับงานทางด้านการตรวจหาและนับจำนวนเซลล์ภายในห้องปฏิบัติการ จะให้  
ความสำคัญในการวิเคราะห์จำนวนเซลล์ เช่น การตรวจเลือด ปัสสาวะ อุจจาระ เสมหะ เป็นต้น จาก  
การมองด้วยตาผ่านกล้องจุลทรรศน์นานนับชั่วโมงติดต่อกันและเป็นลักษณะงานทำซ้ำจะส่งผลให้  
เกิดอาการกล้ามเนื้อตึงตัวจนก่อให้เกิดความคลาดเคลื่อนได้ จึงมีแนวคิดในการสร้างอุปกรณ์ที่ช่วยลด  
ระยะเวลาในการนับคัดแยกเซลล์ขนาดเล็กที่มีความแม่นยำอย่างอัตโนมัติ ซึ่งส่งผลให้การ  
วินิจฉัยโรคทำได้รวดเร็วมากขึ้นเป็น ชุดกล้องอัจฉริยะ “ไมโครซิสตีซีเอ็น” สำหรับกล้องจุลทรรศน์  
 (“MicrosisDCN” intelligent camera for microscope : Microbes Diagnosis with Deep  
Convolutional Neural Network) สำหรับแยกชนิดและนับจำนวนเซลล์ขนาดเล็กด้วยโครงข่าย  
ประสาท เป็นชุดอุปกรณ์สำหรับสวมชุดกล้องเข้ากับท่อเลนส์ใกล้ตา (Eyepiece lens tube) ของ  
กล้องจุลทรรศน์เชิงซ้อน (Compound microscope) สามารถบ่งบอกจำนวนเซลล์ขนาดเล็กด้วย  
โครงข่ายประสาทที่นับได้ในพื้นที่มาตรฐานการมองเห็นของชุดกล้อง จากพื้นที่ขอบเขตการมองเห็น  
ของตัวรับรูปภาพภายในชุดกล้องมีหน่วยเป็น 11.89 40X “field images” to equal standard  
area หรือ 11.9 คูณจำนวนเซลล์ต่อ HPF (High Power Field) ระบบมีความสามารถในการ  
จำแนกเซลล์ขนาดเล็ก 3 คลาส ได้แก่ เม็ดเลือดแดง (Red blood cell : RBC) เม็ดเลือดขาว  
(White blood cell : WBC) และเกล็ดเลือด (Platelets) ที่มีค่า Mean Average Precision  
(mAP) สูงถึง 0.8681 หรือร้อยละ 86.81 และค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean  
Absolute Error : MAE) ของ RBC 1.06 WBC 0.06 และ Platelets 4.23

สาขาวิชา วิศวกรรมไฟฟ้า  
ปีการศึกษา 2562

ลายมือชื่อนิสิต .....  
ลายมือชื่อ อ.ที่ปรึกษาหลัก .....  
ลายมือชื่อ อ.ที่ปรึกษาร่วม .....

# # 6170169721 : MAJOR ELECTRICAL ENGINEERING

KEYWORD: Image processing, Compound microscope, Convolutional neural network (CNN), Camera kit

Natthakorn Kasamsumran : THE APPLICATION OF NEURAL NETWORK TO DETECT MULTIPLE CELLS VIA COMPOUND MICROSCOPE WITH IMAGE SENSOR. Advisor: Asst. Prof. SUREE PUMRIN, Ph.D. Co-advisor: Assoc. Prof. Norrarat Wattanamongkhol, Ph.D.

For the work in the detection and counting of cells in the laboratory, the researchers will focus on the analysis of the number of cells, for example, blood tests, urine, sputum, etc., with the eyesight through a microscope. Those researchers spend hours of continuous work and repetitive work, which can result in eye fatigue that can lead to discrepancies. Therefore, there is the idea of creating a device that reduces the time required for automatic counting of small cells with accuracy, called "MicrosisDCN" intelligent camera for microscope: Microbes Diagnosis with Deep Convolutional Neural Network. It can mount a camera kit into the eyepiece lens tube of a compound microscope. The device can classify and count small cells using neural networks in the standard area of the camera set from the field of view of the image sensor within the camera unit. The unit is 11.89 40X "field images" to the same standard area or 11.9 times the number of cells per HPF (High Power Field). The system can classify three classes: Red blood cell (RBC), White blood cell (WBC), and Platelets with mean average precision (mAP) up to 0.8681 or 86.81 percent. And with the mean absolute error (MAE) of RBC 1.06 WBC 0.06 and Platelets 4.23.

Field of Study: Electrical Engineering

Student's Signature .....

Academic Year: 2019

Advisor's Signature .....

Co-advisor's Signature .....

## กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้มีความตั้งใจที่นำเสนอผลการวิจัยในการพัฒนาอุปกรณ์สหวิทยาการ ที่นำองค์ความรู้ทางด้านวิศวกรรมมารังสรรค์อุปกรณ์ที่สามารถช่วยให้มนุษย์ทำงานได้ประสิทธิภาพสูงขึ้น ข้าพเจ้าขอขอบคุณบุคคลผู้มีส่วนเกี่ยวข้องที่ทำให้โครงการนี้สำเร็จได้ ประกอบด้วย ผศ.ดร.สุรียะ พุ่มรินทร์ และ รศ.ดร.นรรธิน วัฒนมงคล อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วมจุฬาลงกรณ์มหาวิทยาลัย และมหาวิทยาลัยบูรพา รศ.ดร.วันเฉลิม โปธา อาจารย์ภาควิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ ผศ.น.สพ.ดร.ภัทรรัฐ จันทรฉายทอง ภาควิชาจุลชีววิทยา คณะสัตวแพทยศาสตร์ ที่ให้โอกาสข้าพเจ้าได้ศึกษาวิธีการที่เกี่ยวข้องและปัญหาที่พบในการปฏิบัติงานเพื่อนำมาเป็นที่มาของวิทยานิพนธ์ และบิดามารดา เพื่อน ๆ พี่ ๆ น้อง ๆ ทั้งที่เป็นนิสิตและผู้ที่ทำงานแล้ว ได้มีส่วนร่วมให้วิทยานิพนธ์นี้สำเร็จลุล่วง

องค์ความรู้จากการทำวิทยานิพนธ์สามารถพัฒนาต่อยอดการพัฒนางานวิจัย ทุนพัฒนาสิ่งประดิษฐ์ กองทุนรัชดาภิเษกสมโภช สำนักบริหารวิจัย CU\_GI\_62\_18\_21\_03 ได้รับทุนอุดหนุนโครงการการแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย ครั้งที่ 22 จากสำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ เข้าร่วมโครงการ Young Technopreneur รุ่นที่ 8 ประจำปี 2562 โดยศูนย์บ่มเพาะธุรกิจเทคโนโลยี สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ กับกลุ่มบริษัทสามารถคอร์ปอเรชั่น โครงการ Startup Thailand League 2019 โดยสำนักงานนวัตกรรมแห่งชาติ และได้รับรางวัลติดดาวระดับ 5 ดาว ในการเขียน Concept paper จากการคัดเลือกผลงานที่มีการนำเสนอแนวคิดนวัตกรรมที่โดดเด่น ในกลุ่มการพัฒนาเทคโนโลยีปัญญาประดิษฐ์ อุปกรณ์อัจฉริยะ พลังงานและสิ่งแวดล้อม ในกิจกรรมการเพิ่มศักยภาพและมาตรฐานบุคลากรอุดมศึกษา บ่มเพาะและแลกเปลี่ยนเรียนรู้เพื่อพัฒนาสิ่งประดิษฐ์และนวัตกรรม สำนักงานการวิจัยแห่งชาติ (วช.)

ข้าพเจ้าขอขอบคุณ Cooler Master แห่งประเทศไทย ที่สนับสนุน Graphics processing unit (GPU) รุ่น Nvidia GeForce 1080Ti 2 ชุด เพื่อใช้ในการประมวลผลทางการเรียนรู้ของเครื่อง บริษัท สเปซเมด จำกัด ตัวแทนจัดจำหน่ายกล้องจุลทรรศน์ Olympus แห่งประเทศไทย ที่สนับสนุนกล้องจุลทรรศน์ใช้แสงแบบเชิงประกอบ Olympus CX33 สำหรับการถ่ายภาพและทดสอบชุดกล้องและบริษัท คิว เวฟ ซิสเต็มส์ จำกัด ที่อำนวยความสะดวกและสนับสนุนชุดกล้อง EagleEye Smart Camera เพื่อนำมาพัฒนาและปรับปรุงระบบโปรแกรมใช้กับชุดกล้อง MicrosisDCN ได้อย่างสมบูรณ์

## สารบัญ

	หน้า
.....	ค
บทคัดย่อภาษาไทย.....	ค
.....	ง
บทคัดย่อภาษาอังกฤษ.....	ง
กิตติกรรมประกาศ.....	จ
สารบัญ.....	ฉ
สารบัญตาราง.....	ฅ
สารบัญภาพ.....	ฉ
บทที่ 1 บทนำ.....	1
1.1 ที่มาและความสำคัญของปัญหา.....	1
1.2 วัตถุประสงค์ของงานวิจัย.....	2
1.3 ขอบเขตงานวิจัย.....	3
1.4 วิธีการดำเนินการวิจัย.....	4
1.5 ประโยชน์ที่คาดว่าจะได้รับ.....	4
บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง.....	5
2.1 คุณลักษณะของปัญญาประดิษฐ์.....	5
2.2 ชนิดของโครงข่ายประสาท.....	7
2.3 ทฤษฎีทางด้านการเรียนรู้ของเครื่อง.....	10
2.4 เทคโนโลยีที่นำมาใช้ในงานวิจัย.....	13
2.5 วิธีการประเมินผลความแม่นยำของโครงข่ายประสาท.....	18
2.5.1 ค่าความแม่นยำ.....	20

2.5.2 ค่าความเที่ยงตรง.....	20
2.5.3 ค่าการเรียกคืน.....	20
2.5.4 ค่าคะแนน F1 และ F เบต้า.....	20
2.5.5 การบ่งบอกความสามารถของโมเดลโครงข่ายประสาท.....	21
2.5.6 การบ่งบอกความสามารถด้วยค่า Average Precision (AP).....	22
2.5.7 การบ่งบอกความสามารถด้วยค่า Mean Average Precision (mAP).....	24
2.6 หลักการเกิดภาพของกล้องจุลทรรศน์.....	27
2.7 คุณลักษณะของกล้องจุลทรรศน์ชนิดไบรท์ฟิลด์.....	39
2.8 รายละเอียดองค์ประกอบเซลล์ขนาดเล็ก.....	41
2.9 งานวิจัยที่เกี่ยวข้องกับการนับเซลล์ขนาดเล็ก.....	46
บทที่ 3 ระเบียบวิธีวิจัย.....	49
3.1 การพัฒนาระบบชุดกล้องสำหรับโครงงานวิจัย.....	49
3.2 การพัฒนาชุดเลนส์และหน่วยกำลังขยายภาพ.....	58
3.2.1 การคำนวณขนาดพื้นที่ขอบเขตการมองเห็นของกล้องจุลทรรศน์.....	58
3.2.2 การคำนวณขนาดพื้นที่ขอบเขตการมองเห็นของตัวรับรู้อภาพ.....	60
3.2.3 การบ่งบอกจำนวนเซลล์ขนาดเล็กด้วยหน่วยกำลังขยายภาพ.....	63
3.3 การออกแบบโครงข่ายประสาทสำหรับตรวจจับเซลล์ขนาดเล็ก.....	68
3.3.1 การสร้างชุดข้อมูล (Dataset) และการติดป้าย (Labeling).....	68
3.3.2 กรรมวิธีการขยายชุดภาพ (Image augmentation).....	72
3.3.3 กรรมวิธีพัฒนาโครงข่ายประสาทสำหรับตรวจจับเซลล์ขนาดเล็ก.....	78
บทที่ 4 ผลการทำงานของระบบ.....	87
4.1 การขยายชุดภาพและพารามิเตอร์ของโมเดล.....	87
4.1.1 ผลลัพธ์การขยายชุดภาพเซลล์เม็ดเลือดแดง (RBC).....	91
4.1.2 ผลลัพธ์การขยายภาพเซลล์เม็ดเลือดขาว (WBC) และเกล็ดเลือด (Platelets).....	95



4.1.3 พารามิเตอร์ของโมเดลโครงข่ายประสาทที่ใช้ในชุดกล้อง .....	99
4.2 ผลลัพธ์การฝึกโมเดลแบบไม่ปรับพารามิเตอร์.....	102
4.3 ผลลัพธ์การฝึกโมเดลปรับพารามิเตอร์แบบที่ 1 .....	111
4.4 ผลลัพธ์การฝึกโมเดลปรับพารามิเตอร์แบบที่ 2 .....	120
บทที่ 5 บทสรุปผลงานวิจัย.....	129
5.1 สรุปผลการวิจัย.....	129
5.2 แนวทางการพัฒนาการวิจัย .....	130
5.3 ปัญหาและอุปสรรค.....	130
ภาคผนวก ก. บทความวิจัยที่ได้ตีพิมพ์ใน iEECON 2020.....	131
ภาคผนวก ข. ชุดคำสั่งติดตั้ง CUDA Toolkit และ cuDNN .....	135
ภาคผนวก ค. ชุดคำสั่งติดตั้ง OPENCV และ TensorFlow 1.15.....	140
ภาคผนวก ง. ชุดคำสั่งที่ใช้ในวิทยานิพนธ์ .....	150
ภาคผนวก จ. วิธีการใช้งานชุดกล้อง MicrosisDCN กับกล้องจุลทรรศน์.....	162
ภาคผนวก ฉ. คุณสมบัติของชุดกล้อง EagleEYE Smart Camera .....	166
ภาคผนวก ช. คุณสมบัติกล้องจุลทรรศน์และชุดเลนส์.....	170
ภาคผนวก ซ. ตารางค่าการสูญเสียและความแม่นยำ.....	176
บรรณานุกรม.....	188
ประวัติผู้เขียน .....	193

## สารบัญตาราง

	หน้า
ตารางที่ 1 การแบ่งข้อมูลตามระดับชั้น (Threshold) .....	21
ตารางที่ 2 การคำนวณค่าความแม่นยำ ความเที่ยงตรง การเรียกคืน และคะแนน F1 .....	22
ตารางที่ 3 ความสัมพันธ์ค่าความเที่ยงตรง (Precision) และค่าการเรียกคืน (Recall).....	23
ตารางที่ 4 ค่าดัชนีหักเหของตัวกลางชนิดต่าง ๆ และค่าความหนาแน่นโดยประมาณ .....	30
ตารางที่ 5 คุณสมบัติการเกิดภาพของเลนส์รวมแสง และเลนส์กระเจิงแสง .....	32
ตารางที่ 6 กำลังการขยายและค่า Numerical Aperture (N.A) .....	38
ตารางที่ 7 ค่ามาตรฐานของเม็ดเลือดแดง เม็ดเลือดขาว และเกล็ดเลือด [22] .....	46
ตารางที่ 8 เปรียบเทียบคุณสมบัติอุปกรณ์ที่ใช้ในการวิจัย .....	57
ตารางที่ 9 แสดงค่าเส้นผ่านศูนย์กลางของขอบเขตการมองเห็น .....	59
ตารางที่ 10 แสดงค่าพื้นที่ของขอบเขตการมองเห็น .....	60
ตารางที่ 11 ขนาดความกว้างและความยาวของตัวรับรู้ภาพ (Image sensor size) .....	61
ตารางที่ 12 สรุปค่าพารามิเตอร์ของชุดกล้องที่ได้จากการคำนวณและการวัดได้จริง .....	67
ตารางที่ 13 เปรียบเทียบ Average Precision (AP) ของโมเดลต่าง ๆ [38] .....	80
ตารางที่ 14 ผลแสดงพิกัดของพื้นที่สี่เหลี่ยมปิดล้อมต่อวัตถุภายในชุดข้อมูลภาพแบบ CSV (1).....	93
ตารางที่ 15 ผลแสดงพิกัดของพื้นที่สี่เหลี่ยมปิดล้อมต่อวัตถุภายในชุดข้อมูลภาพแบบ CSV (2).....	97
ตารางที่ 16 ค่าพารามิเตอร์ Anchor มาตรฐานของ RetinaNet .....	101
ตารางที่ 17 ค่าพารามิเตอร์ Anchor แบบที่ 1 สำหรับฝึกโมเดลโครงข่ายประสาท .....	101
ตารางที่ 18 ค่าพารามิเตอร์ Anchor แบบที่ 2 สำหรับฝึกโมเดลโครงข่ายประสาท .....	101
ตารางที่ 19 ผลลัพธ์ค่า MAE RMSE R2 และ mAP ของโมเดลแบบไม่ปรับพารามิเตอร์ .....	102
ตารางที่ 20 ผลเปรียบเทียบจำนวนที่นับได้จริงและพยากรณ์ได้ของโมเดลแบบไม่ปรับพารามิเตอร์ (1) .....	107

ตารางที่ 21 ผลเปรียบเทียบจำนวนที่นับได้จริงและพยากรณ์ได้ของโมเดลแบบไม่ปรับพารามิเตอร์ (2)	108
.....	
ตารางที่ 22 ผลเปรียบเทียบจำนวนที่นับได้จริงและพยากรณ์ได้ของโมเดลแบบไม่ปรับพารามิเตอร์ (3)	109
.....	
ตารางที่ 23 ผลเปรียบเทียบจำนวนที่นับได้จริงและพยากรณ์ได้ของโมเดลแบบไม่ปรับพารามิเตอร์ (4)	110
.....	
ตารางที่ 24 ผลลัพธ์ค่า MAE RMSE R2 และ mAP ของโมเดลปรับพารามิเตอร์แบบที่ 1	111
.....	
ตารางที่ 25 ผลเปรียบเทียบจำนวนที่นับจริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 1 (1)	114
.....	
ตารางที่ 26 ผลเปรียบเทียบจำนวนที่นับจริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 1 (2)	115
.....	
ตารางที่ 27 ผลเปรียบเทียบจำนวนที่นับจริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 1 (3)	116
.....	
ตารางที่ 28 ผลเปรียบเทียบจำนวนที่นับจริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 1 (4)	117
.....	
ตารางที่ 29 ผลลัพธ์ค่า MAE RMSE R2 และ mAP ของโมเดลปรับพารามิเตอร์แบบที่ 2	120
.....	
ตารางที่ 30 ผลเปรียบเทียบจำนวนที่นับจริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 2 (1)	123
.....	
ตารางที่ 31 ผลเปรียบเทียบจำนวนที่นับจริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 2 (2)	124
.....	
ตารางที่ 32 ผลเปรียบเทียบจำนวนที่นับจริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 2 (3)	125
.....	
ตารางที่ 33 ผลเปรียบเทียบจำนวนที่นับจริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 2 (4)	126
.....	
ตารางที่ 34 ค่าการสูญเสียและค่าความแม่นยำจากการฝึกโมเดลไม่ปรับพารามิเตอร์	176
.....	
ตารางที่ 35 ค่าการสูญเสียและค่าความแม่นยำจากการฝึกโมเดลไม่ปรับพารามิเตอร์ (ต่อ)	177

ตารางที่ 36 ตารางค่าการสูญเสียและค่าความแม่นยำจากการตรวจสอบความถูกต้องของโมเดลไม่  
ปรับพารามิเตอร์..... 178

ตารางที่ 37 ค่าการสูญเสียและค่าความแม่นยำจากการตรวจสอบความถูกต้องของโมเดลไม่ปรับ  
พารามิเตอร์ (ต่อ) ..... 179

ตารางที่ 38 ค่าการสูญเสียและค่าความแม่นยำจากการฝึกโมเดลปรับพารามิเตอร์แบบที่ 1 ..... 180

ตารางที่ 39 ค่าการสูญเสียและค่าความแม่นยำจากการฝึกโมเดลปรับพารามิเตอร์แบบที่ 1 (ต่อ).. 181

ตารางที่ 40 ค่าการสูญเสียและค่าความแม่นยำจากการตรวจสอบความถูกต้องของโมเดลปรับ  
พารามิเตอร์แบบที่ 1 ..... 182

ตารางที่ 41 ค่าการสูญเสียและค่าความแม่นยำจากการตรวจสอบความถูกต้องของโมเดลปรับ  
พารามิเตอร์แบบที่ 1 (ต่อ)..... 183

ตารางที่ 42 ค่าการสูญเสียและค่าความแม่นยำจากการฝึกโมเดลปรับพารามิเตอร์แบบที่ 2 ..... 184

ตารางที่ 43 ค่าการสูญเสียและค่าความแม่นยำจากการฝึกโมเดลปรับพารามิเตอร์แบบที่ 2 (ต่อ).. 185

ตารางที่ 44 ค่าการสูญเสียและค่าความแม่นยำจากการตรวจสอบความถูกต้องของโมเดลปรับ  
พารามิเตอร์แบบที่ 2 ..... 186

ตารางที่ 45 ค่าการสูญเสียและค่าความแม่นยำจากการตรวจสอบความถูกต้องของโมเดลปรับ  
พารามิเตอร์แบบที่ 2 (ต่อ)..... 187

## สารบัญภาพ

	หน้า
ภาพที่ 1 แบบร่างระบบชุดกล้อง MicrosisDCN ที่ใช้ในวิทยานิพนธ์.....	5
ภาพที่ 2 ชนิดของโครงข่ายประสาท (Neural Network) ที่ใช้งานในปัจจุบัน [8].....	8
ภาพที่ 3 การเปรียบเทียบเซลล์ประสาท และโครงสร้างของโครงข่ายประสาท .....	9
ภาพที่ 4 แนวคิดของโครงข่ายประสาทแบบคอนโวลูชัน (CNN) [10].....	10
ภาพที่ 5 การสร้างผังการแยกคุณลักษณะของจุดภาพโดยใช้ตัวกรอง .....	11
ภาพที่ 6 วิธีการทำ Stride และ Stride padding เพื่อให้ได้ผังการแยกคุณลักษณะ.....	12
ภาพที่ 7 วิธีการสร้างผังการแยกคุณลักษณะด้วยวิธีการ Pooling .....	13
ภาพที่ 8 ผังแสดงโครงสร้างของ Tensorflow core [11] .....	14
ภาพที่ 9 สัญลักษณ์ของ Keras และ Tensorflow [12].....	14
ภาพที่ 10 ภาพอธิบายลำดับชั้น (Layer) ของโครงข่ายประสาทคอนโวลูชัน (CNN) บน Keras.....	16
ภาพที่ 11 แสดงขั้นตอนการทำ Fully connected โดยใช้การคูณเมทริกซ์ผลคูณแบบดอท .....	17
ภาพที่ 12 สัญลักษณ์ของ OpenCV (Open Source Computer Vision Library) [13].....	17
ภาพที่ 13 ผังแสดงการจัดกลุ่มข้อมูลแบบซัฟเซท (ซ้าย) และ Confusion matrix (ขวา).....	18
ภาพที่ 14 ผังแสดงความสัมพันธ์ระหว่างความเที่ยงตรง และความแม่นยำ.....	19
ภาพที่ 15 การบ่งบอกความสามารถของโมเดลโครงข่ายประสาทด้วยวิธี Interpolated.....	23
ภาพที่ 16 ภาพอธิบายการหักเหของแสงด้วยทฤษฎีอนุภาคแสงของนิวตัน เส้น DF คือแกน ระนาบ (Principle axis) และ HI คือแกนตั้งฉากเลนส์ (Lens axis) [17].....	28
ภาพที่ 17 ภาพแสดงคุณลักษณะของเลนส์รวมแสงและเลนส์กระจายแสง [17] .....	30
ภาพที่ 18 แผนภาพอธิบายคุณสมบัติของเลนส์รวมแสง (Converging lens) [17].....	31
ภาพที่ 19 แบบร่างของชุดเลนส์ประกอบภายในกล้องจุลทรรศน์ชนิดไบรท์ฟิลด์ [17] .....	34
ภาพที่ 20 เลนส์รวมแสง 2 ชิ้นวางตัวบนแกนระนาบแสดงภาพวัตถุจริง [17].....	34

ภาพที่ 21	แผนภาพจำลองการเลี้ยวเบนของแสงของฟรอนฮอฟเฟอ (Fraunhofer diffraction) [18]	36
ภาพที่ 22	แผนภาพแสดงการกระเจิงแสงผ่านตัวกลางอากาศ [18] และ Immersion oil [19]	37
ภาพที่ 23	การเกิดมุมของความสามารถในการรวมแสงของเลนส์วัตถุ (N.A.) [19]	38
ภาพที่ 24	องค์ประกอบชิ้นส่วนภายในของกล้องจุลทรรศน์แบบเชิงซ้อน [21]	39
ภาพที่ 25	ผังการสร้างโลหิต (Hematopoiesis) ที่แสดงการแบ่งชนิดของเซลล์ [22]	41
ภาพที่ 26	ลักษณะของเม็ดเลือดแดงปกติ (Normal erythrocyte) และรูปร่างผิดปกติ [22]	42
ภาพที่ 27	ลักษณะของเม็ดเลือดขาว Neutrophil แบบ Segmented (ซ้าย) และ Band (ขวา)	43
ภาพที่ 28	ลักษณะของเม็ดเลือดขาว Eosinophil แบบ Myelocyte (ซ้าย) และ Mature (ขวา)	43
ภาพที่ 29	ลักษณะของเม็ดเลือดขาว Basophil	44
ภาพที่ 30	ลักษณะของเม็ดเลือดขาว Monocyte	44
ภาพที่ 31	ลักษณะของเม็ดเลือดขาว Lymphocyte	45
ภาพที่ 32	ลักษณะการกระจายตัวของเกล็ดเลือด (Platelet) ร่วมกับเม็ดเลือดแดงและขาว	45
ภาพที่ 33	กล้องจุลทรรศน์ตาเดียว (Monocular) ตาคู่ (Binocular) และสามตา (Trinocular)	49
ภาพที่ 34	การพัฒนาระบบชุดกล้องสำหรับโครงการงานวิจัยสำหรับกล้องจุลทรรศน์แบบ Binocular	50
ภาพที่ 35	ผังเปรียบเทียบคุณลักษณะของอุปกรณ์ที่แตกต่างกัน	51
ภาพที่ 36	การบันทึกภาพด้วยกล้องดิจิทัลผ่านตัวแปลงเลนส์ขนาด 23.2 มิลลิเมตร	51
ภาพที่ 37	แผนภาพแสดงระยะภาพของเลนส์แบบ C mount และ CS mount	52
ภาพที่ 38	การทดสอบตัวรับรูปภาพผ่านเครื่องคอมพิวเตอร์ ณ ห้องปฏิบัติการ ภาควิชาจุลชีววิทยา คณะสัตวแพทยศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย	52
ภาพที่ 39	โมดูลกล้อง Raspberry Pi ถ่ายทอดสัญญาณผ่าน CSI cable สวมกับท่อเลนส์ใกล้ตา	53
ภาพที่ 40	บอร์ดสมองกลฝังตัว Raspberry Pi 3 B+ และวิธีการเชื่อมต่อกล้องผ่าน CSI Cable	54
ภาพที่ 41	ชุดกล้อง EagleEYE Smart Camera สำหรับงานด้าน Machine Vision และการเปลี่ยนชุดเลนส์ไปเป็น เลนส์สวมท่อส่งภาพแนวตั้ง และเลนส์กำลังขยาย 0.5X แบบ C mount	55
ภาพที่ 42	ระบบปฏิบัติการ Linux RT รองรับโปรแกรม LabVIEW และระบบปฏิบัติการ Raspbian ที่ถูกติดตั้งชุดคำสั่งที่ใช้ในโครงการงานวิจัย	55

ภาพที่ 43 การปรับแต่งระบบปฏิบัติการ Raspbian พร้อมติดตั้งชุดคำสั่งโครงข่ายประสาทใหม่	55
ภาพที่ 44 การเปรียบเทียบชุดกล้องดั้งเดิม และชุดกล้องที่ถูกพัฒนาขึ้นใหม่	56
ภาพที่ 45 แสดงเส้นผ่านศูนย์กลางไดอะแฟรมเลนส์ (ซ้าย) และ Field of View (ขวา) [33]	59
ภาพที่ 46 การเปรียบเทียบพื้นที่ขอบเขตการมองเห็น FN20 กับขนาดพื้นที่ตัวรับรู้อุปกรณ์ [33]	61
ภาพที่ 47 การเปรียบเทียบขนาดพื้นที่ระหว่าง ขอบเขตการมองเห็น และขอบเขตตัวรับรู้อุปกรณ์	63
ภาพที่ 48 การสอบเทียบการวัดขนาดด้วย Ocular micrometer และ Stage micrometer	64
ภาพที่ 49 แผ่นสไลด์สอบเทียบ (Calibration slide) สำหรับกล้องจุลทรรศน์	65
ภาพที่ 50 แสดงพื้นที่ตัวรับรู้อุปกรณ์ที่เกิดจากสอบเทียบจริงด้วยเลนส์ใกล้วัตถุ 40X	66
ภาพที่ 51 แสดงความยาวตัวรับรู้อุปกรณ์ (ซ้าย) และความสูงตัวรับรู้อุปกรณ์ (ขวา)	66
ภาพที่ 52 เครื่องมือบรรณนิทัศน์ภาพ Colabeler บนระบบปฏิบัติการ Windows 10	68
ภาพที่ 53 การใช้เครื่องมือ Colabeler วางพื้นที่สี่เหลี่ยมปิดล้อมเม็ดเลือดแดง	69
ภาพที่ 54 การใช้เครื่องมือ Colabeler วางพื้นที่สี่เหลี่ยมปิดล้อมเม็ดเลือดขาว และเกล็ดเลือด	69
ภาพที่ 55 ผังภาพรวมของการนำชุดข้อมูลไปใช้ฝึกโครงข่ายประสาทของชุดกล้อง MicrosisDCN	72
ภาพที่ 56 ภาพถ่ายเซลล์เม็ดเลือดแดง กับภาพถ่ายที่ถูกบรรณนิทัศน์ (BBOX)	73
ภาพที่ 57 ผังขั้นตอนการทำงานของกรรมวิธีการขยายชุดภาพ (Image augmentation)	74
ภาพที่ 58 ภาพถ่ายเซลล์เม็ดเลือดแดงที่ผ่านการขยายชุดภาพด้วยฟังก์ชัน Grayscale	75
ภาพที่ 59 ภาพถ่ายเซลล์เม็ดเลือดแดงที่ผ่านการขยายชุดภาพด้วยฟังก์ชัน GaussianBlur	75
ภาพที่ 60 ภาพถ่ายเซลล์เม็ดเลือดแดงที่ผ่านการขยายชุดภาพด้วยฟังก์ชัน AdditiveGaussianNoise	75
ภาพที่ 61 ภาพถ่ายเซลล์เม็ดเลือดแดงที่ผ่านการขยายชุดภาพด้วยฟังก์ชัน MultiplyHueAndSaturation (Hue)	76
ภาพที่ 62 ภาพถ่ายเซลล์เม็ดเลือดแดงที่ผ่านการขยายชุดภาพด้วยฟังก์ชัน MultiplyHueAndSaturation (Saturation)	76
ภาพที่ 63 ภาพถ่ายเซลล์เม็ดเลือดแดงที่ผ่านการขยายชุดภาพด้วยฟังก์ชัน AddToHueAndSaturation (ทั้งคู่)	76

ภาพที่ 64 ภาพถ่ายเซลล์เม็ดเลือดแดงที่ผ่านการขยายชุดภาพด้วยฟังก์ชัน MultiplyBrightness ... 77

ภาพที่ 65 ภาพถ่ายเซลล์เม็ดเลือดแดงที่ผ่านการขยายชุดภาพด้วยฟังก์ชัน GammaContrast ..... 77

ภาพที่ 66 ภาพถ่ายเซลล์เม็ดเลือดแดงที่ผ่านการขยายชุดภาพด้วยฟังก์ชัน SigmoidContrast..... 77

ภาพที่ 67 ลักษณะโครงข่ายประสาท ResNet50, Xception, Inception V3, VGG19, VGG16 [37]  
..... 79

ภาพที่ 68 การแสดงวิธีคำนวณพื้นที่ Intersection over union (IoU) ..... 81

ภาพที่ 69 โครงสร้างภายในโครงข่ายประสาท RetinaNet โดย Tsung-Yi Lin และคณะ [38]..... 81

ภาพที่ 70 แนวคิดของการทำ Transfer learning ของโครงข่ายประสาทของระบบที่เสนอ ..... 82

ภาพที่ 71 ขั้นตอนการฝึกโมเดลโครงข่ายประสาทสำหรับตรวจจับเซลล์ขนาดเล็ก..... 83

ภาพที่ 72 การแปลงข้อมูล BBOX Dataframe จากไฟล์ XML ไปเป็น CSV ..... 84

ภาพที่ 73 การฝึกโมเดลโครงข่ายประสาทด้วย Nvidia GeForce GTX 1080Ti 2 ชุด..... 85

ภาพที่ 74 ผลการทำงานชุดคำสั่งตรวจสอบภาพ และเก็บเข้าตัวแปร images ตามจำนวนภาพ..... 87

ภาพที่ 75 ผลการทำงานแสดงภาพที่อยู่ในตัวแปรชนิด Array ชื่อว่า images ..... 88

ภาพที่ 76 ผลการทำงานแสดงไฟล์บรรณนิทัศน์ XML ที่อยู่ในแฟ้ม images/outputs ..... 88

ภาพที่ 77 ข้อมูลป้ายตามจำนวนวัตถุที่ถูกสกัดจากไฟล์บรรณนิทัศน์ไปเป็น CSV และ Dataframe 89

ภาพที่ 78 การแสดงตำแหน่งพื้นที่สี่เหลี่ยมปิดล้อมบนภาพเซลล์เม็ดเลือดแดงภายในชุดภาพ ..... 91

ภาพที่ 79 ตัวอย่างชุดข้อมูลภาพเซลล์เม็ดเลือดแดง จากกล้องจุลทรรศน์ก่อนได้รับการขยายชุดภาพ  
..... 92

ภาพที่ 80 ตัวอย่างชุดข้อมูลภาพเซลล์เม็ดเลือดแดง จากกล้องจุลทรรศน์หลังได้รับการขยายชุดภาพ  
..... 92

ภาพที่ 81 ตัวอย่างไฟล์บรรณนิทัศน์ XML ที่มีข้อมูลการติดป้ายวัตถุของชุดภาพที่ได้รับการขยาย (1)  
..... 93

ภาพที่ 82 การแสดงตำแหน่งพื้นที่สี่เหลี่ยมปิดล้อมบนภาพเซลล์เม็ดเลือดขาวและเกล็ดเลือดภายใน  
ชุดภาพ..... 95



ภาพที่ 83 ตัวอย่างชุดข้อมูลภาพเซลล์เม็ดเลือดขาวและเกล็ดเลือด จากกล้องจุลทรรศน์ก่อนได้รับการขยายชุดภาพ.....	96
ภาพที่ 84 ตัวอย่างชุดข้อมูลภาพเซลล์เม็ดเลือดขาวและเกล็ดเลือด จากกล้องจุลทรรศน์หลังได้รับการขยายชุดภาพ.....	96
ภาพที่ 85 ตัวอย่างไฟล์บรรณนิทัศน์ XML ที่มีข้อมูลการติดป้ายวัตถุของชุดภาพที่ได้รับการขยาย (2) .....	97
ภาพที่ 86 หน้าจอชุดคำสั่งภาษา Python เข้าถึงโมเดลโครงข่ายประสาทเพื่อนับเซลล์ขนาดเล็ก ....	99
ภาพที่ 87 ชุดภาพพร้อมหมายเลขสำหรับแสดงผลเปรียบเทียบของโมเดลแบบไม่ปรับพารามิเตอร์	103
ภาพที่ 88 กราฟแสดงความแม่นยำ mAP ของโมเดลแบบไม่ปรับพารามิเตอร์ .....	103
ภาพที่ 89 กราฟแสดง Training loss และ Validation loss ของโมเดลแบบไม่ปรับพารามิเตอร์.	104
ภาพที่ 90 กราฟแสดง Training accuracy และ Validation accuracy ของโมเดลแบบไม่ปรับพารามิเตอร์.....	104
ภาพที่ 91 ผลลัพธ์การนับจำนวนเซลล์เม็ดเลือดแดงที่นับได้จริงและพยากรณ์ได้ของโมเดลแบบไม่ปรับพารามิเตอร์.....	105
ภาพที่ 92 ผลลัพธ์การนับจำนวนเซลล์เม็ดเลือดขาวและเกล็ดเลือดที่นับได้จริงและพยากรณ์ได้ของโมเดลแบบไม่ปรับพารามิเตอร์.....	106
ภาพที่ 93 ชุดภาพพร้อมหมายเลขสำหรับแสดงผลเปรียบเทียบของโมเดลปรับพารามิเตอร์แบบที่ 1 .....	112
ภาพที่ 94 กราฟแสดงความแม่นยำ mAP ของโมเดลปรับพารามิเตอร์แบบที่ 1 .....	112
ภาพที่ 95 กราฟแสดง Training loss และ Validation loss ของโมเดลปรับพารามิเตอร์แบบที่ 1 .....	113
ภาพที่ 96 กราฟแสดง Training accuracy และ Validation accuracy ของโมเดลปรับพารามิเตอร์แบบที่ 1 .....	113
ภาพที่ 97 ผลลัพธ์การนับจำนวนเซลล์เม็ดเลือดแดงที่นับได้จริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 1 .....	118

ภาพที่ 98 ผลลัพธ์การนับจำนวนเซลล์เม็ดเลือดขาวและเกล็ดเลือดที่นับได้จริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 1 .....	119
ภาพที่ 99 ชุดภาพพร้อมหมายเลขสำหรับแสดงผลเปรียบเทียบของโมเดลปรับพารามิเตอร์แบบที่ 2 .....	121
ภาพที่ 100 กราฟแสดงความแม่นยำ mAP ของโมเดลปรับพารามิเตอร์แบบที่ 2 .....	121
ภาพที่ 101 กราฟแสดง Training loss และ Validation loss ของโมเดลปรับพารามิเตอร์แบบที่ 2 .....	122
ภาพที่ 102 กราฟแสดง Training accuracy และ Validation accuracy ของโมเดลปรับพารามิเตอร์แบบที่ 2 .....	122
ภาพที่ 103 ผลลัพธ์การนับจำนวนเซลล์เม็ดเลือดแดงที่นับได้จริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 2 .....	127
ภาพที่ 104 ผลลัพธ์การนับจำนวนเซลล์เม็ดเลือดขาวและเกล็ดเลือดที่นับได้จริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 2 .....	128
ภาพที่ 105 หน้าเว็บไซต์สำหรับดาวโหลด CUDA Toolkit version 10.2 .....	135
ภาพที่ 106 หน้าเว็บไซต์สำหรับดาวโหลด CUDA Toolkit version 10.0 .....	136
ภาพที่ 107 หน้าต่างที่ถูกเรียกใช้จากคำสั่ง nvidia-smi แสดงคุณสมบัติของ GPU .....	137
ภาพที่ 108 หน้าเว็บไซต์สำหรับดาวโหลด cuDNN version 7.6.5 .....	138
ภาพที่ 109 หน้าต่าง Terminal แสดงการทดสอบ cuDNN ด้วยคำสั่ง mnistCUDNN .....	139
ภาพที่ 110 หน้าต่าง Terminal แสดงการติดตั้ง dependencies ที่จำเป็นของ OpenCV .....	140
ภาพที่ 111 การติดตั้งสภาพแวดล้อมจำลองของ Python และ PyPI .....	141
ภาพที่ 112 การใช้โปรแกรม Nano ในการแก้ไขชุดคำสั่งภายในไฟล์ .bashrc .....	142
ภาพที่ 113 หน้าต่าง Terminal แสดงการติดตั้งคลังชุดคำสั่งส่วนเสริมด้วย pip install .....	143
ภาพที่ 114 ผลการ compile ชุดคำสั่ง OpenCV ส่วนที่ 1 .....	144
ภาพที่ 115 ผลการ compile ชุดคำสั่ง OpenCV ส่วนที่ 2 .....	145
ภาพที่ 116 ผลการ compile ชุดคำสั่ง OpenCV ส่วนที่ 3 .....	145

ภาพที่ 117 ผลการติดตั้ง OpenCV รุ่น 4.2.0 โดยสมบูรณ์.....	146
ภาพที่ 118 หน้าต่าง Terminal แสดงผลการติดตั้ง TensorFlow รุ่น 1.15.....	147
ภาพที่ 119 ผลการแสดง session เมื่อติดตั้ง TensorFlow อย่างสมบูรณ์.....	148



## บทที่ 1 บทนำ

### 1.1 ที่มาและความสำคัญของปัญหา

เซลล์ขนาดเล็กจัดเป็นหน่วยพื้นฐานทางชีวภาพ ซึ่งมีความสำคัญต่อมนุษย์เป็นอย่างยิ่ง มีคุณสมบัติและก่อให้เกิดโทษ จึงต้องมีการคัดแยกและนับจำนวน เพื่อศึกษาและวิจัยทั่วทุกมุมโลก การนับเซลล์ขนาดเล็กแบบมาตรฐาน ผู้เชี่ยวชาญจะนำตัวอย่าง (Specimen) เติมลงบนแผ่นสไลด์ (Slide glass) หรือฮีโมไซโตมิเตอร์ (Hemocytometer) เพื่อนับจำนวนส่องภายใต้กล้องจุลทรรศน์ ซึ่งต้องอาศัยประสบการณ์ในการระบุชนิดของเซลล์ด้วยตนเอง จะต้องนับจำนวนด้วยมือที่ละเซลล์ (Manual cell counting) ก่อนจดบันทึกบนกระดาษแล้วนำไปคำนวณหาจำนวนที่แท้จริง ในปี 2559 จากการศึกษาของ Washington University School of Medicine in St. Louis ค้นพบการต้านยาปฏิชีวนะของวัณโรคในห้องปฏิบัติการ (Drug-resistant tuberculosis reversed in lab) ที่เกิดจากเชื้อจุลินทรีย์ *Mycobacterium tuberculosis* มีผู้เสียชีวิตจากวัณโรคสูงถึง 1.5 ล้านคน [1] จากผลกระทบดังกล่าว กลุ่มนักศึกษาระดับปริญญาตรี Harvard University ได้ให้ความสำคัญ โดยการประยุกต์ใช้องค์ความรู้ทางด้านปัญญาประดิษฐ์ (Artificial intelligence) และการเรียนรู้ของเครื่อง (Machine learning) เพื่อช่วยในการพยากรณ์ภาวะต้านยาปฏิชีวนะของวัณโรคที่มีความแม่นยำสูงถึงร้อยละ 94 หนึ่งในนักศึกษากล่าวว่า “ปัญญาประดิษฐ์จะมีส่วนในการวินิจฉัยทางคลินิกโดยการสังเคราะห์ข้อมูลจำนวนมาก และทำให้ตัดสินใจได้อย่างรวดเร็ว ช่วยให้แพทย์สามารถทำการวินิจฉัยและรักษาโรคได้อย่างรวดเร็ว ที่ไม่ถูกจำกัดเฉพาะวัณโรครวมถึงโรคอื่น ๆ ด้วย” [2]

สำหรับการวินิจฉัยการเกิดโรคจากการติดเชื้อทั้งในมนุษย์และสัตว์ จะมีวิธีการที่มีการนำตัวอย่างสารคัดหลั่งไปตรวจ เพื่อเป็นข้อมูลประกอบการรักษา เช่น การตรวจเลือด ปัสสาวะ อุจจาระ เสมหะ เป็นต้น หนึ่งในวิธีการตรวจหาการติดเชื้อสามารถวิเคราะห์จากปริมาณเซลล์เม็ดเลือด ซึ่งห้องปฏิบัติการทางเทคนิคการแพทย์ จะให้ความสำคัญในการวิเคราะห์จำนวนเซลล์ รวมถึงปริมาณความเข้มข้นของสารต่าง ๆ เรียกว่าการตรวจความสมบูรณ์ของเม็ดเลือด (Complete blood count : CBC) ซึ่งประกอบไปด้วย จำนวนเม็ดเลือดแดง (Red blood cell : RBC) เม็ดเลือดขาว (White blood cell : WBC) และเกล็ดเลือด (Platelets) ในทางการวินิจฉัยระดับคลินิกของห้องปฏิบัติการ วิธีการตรวจค่าความสมบูรณ์ของเม็ดเลือด (Peripheral blood film : PBF) ผ่านกล้องจุลทรรศน์ได้รับความนิยมสูงถึงร้อยละ 70 [3] สำหรับศึกษาเซลล์ขนาดเล็กนั้น อุปกรณ์ที่สำคัญอย่างยิ่งคือ กล้องจุลทรรศน์ (Microscope) ซึ่งเป็นเครื่องมือที่ใช้ขยายภาพของวัตถุขนาดเล็ก เช่น กล้องจุลทรรศน์เชิงซ้อน (Compound microscope) เป็นกล้องที่ใช้กันอย่างแพร่หลาย ประกอบด้วยชุดเลนส์ใกล้ตา (Eyepiece lens) และเลนส์ใกล้วัตถุ (Objective lens) มีกำลังขยายตั้งแต่ 40-1000 เท่า [4]

สำหรับงานทางด้านการตรวจหาความสมบูรณ์ของเม็ดเลือด จะต้องมองด้วยตาผ่านกล้องจุลทรรศน์นานติดต่อกันหลายชั่วโมง และเป็นลักษณะงานทำซ้ำ (Routine) ทำให้ส่งผลให้เกิดอาการล้าสายตาจนก่อให้เกิดความคลาดเคลื่อนได้ (Human error) ผู้วิจัยจึงมีแนวคิดในการสร้างอุปกรณ์ที่ช่วยลดระยะเวลาในการนับคัดแยกที่มีความแม่นยำ ซึ่งส่งผลให้การวินิจฉัยโรคทำได้รวดเร็วมากขึ้น เรียกระบบโดยภาพรวมว่าชุดกล้องอัจฉริยะ “ไมโครซิสตีซีเอ็น” สำหรับกล้องจุลทรรศน์ (“MicrosisDCN” intelligent camera for microscope : Microbes Diagnosis with Deep Convolutional Neural Network) สำหรับแยกชนิดและนับจำนวนเซลล์ขนาดเล็กด้วยโครงข่ายประสาท (Neural network) เป็นชุดอุปกรณ์สำหรับสวมชุดกล้องเข้ากับท่อเลนส์ใกล้ตา (Eyepiece lens tube) ของกล้องจุลทรรศน์ทั่วไป เพื่อถ่ายภาพนำไปเข้ากระบวนการประมวลผลภาพ (Image processing) ด้วยโครงข่ายประสาทที่ถูกฝึกให้จำแนกภาพเซลล์ขนาดเล็กขนาดเล็กจากชุดข้อมูล (Datasets) จึงทำให้ชุดกล้องมีขีดความสามารถจำแนกและนับจำนวนเซลล์ได้

## 1.2 วัตถุประสงค์ของงานวิจัย

1. เพื่อพัฒนาชุดกล้องขนาดเล็กสำหรับสวมเข้ากับท่อเลนส์ใกล้ตา พัฒนาขั้นตอนวิธี (Algorithm) ที่ปรับประยุกต์ใช้ในการจำแนก และนับจำนวนเซลล์ขนาดเล็ก เพื่อลดระยะเวลาการระบุชนิดของเซลล์ เพิ่มความแม่นยำในการคัดแยก และนับจำนวนเซลล์ผ่านกล้องจุลทรรศน์เชิงซ้อน (Compound microscope)
2. เพื่อออกแบบและฝึกโครงข่ายประสาทแบบ Convolution Neural Network (CNN) ด้วยเครื่องมือแบบ Open source สำหรับการคัดแยกและนับจำนวนเซลล์ของเซลล์เม็ดเลือด (Blood Cells) จำนวน 3 ชนิด ได้แก่ เม็ดเลือดแดง (Red blood cell : RBC) เม็ดเลือดขาว (White blood cell : WBC) และเกล็ดเลือด (Platelets)
3. เพื่อประยุกต์ใช้การประมวลผลภาพ (Image processing) และการขยายชุดภาพ (Augmentation) ในการสร้างชุดข้อมูลภาพถ่ายเซลล์เม็ดเลือดจากกล้องจุลทรรศน์แบ่งออกเป็นชุดสำหรับการฝึก (Training set) ชุดสำหรับทดสอบ (Test set) และชุดตรวจสอบความถูกต้อง (Validation set)

### 1.3 ขอบเขตงานวิจัย

1. ศึกษาวิธีการใช้งานกล้องจุลทรรศน์สำหรับงานวิจัย ซึ่งเป็นอุปกรณ์ช่วยขยายวัตถุที่มีขนาดเล็กที่ดวงตามนุษย์ไม่สามารถมองเห็น ผู้วิจัยได้เลือกกล้องจุลทรรศน์เชิงซ้อน (Compound microscope) เพราะเป็นชนิดที่ได้รับความนิยม และใช้งานอย่างแพร่หลาย ยี่ห้อ Olympus CX33 Microscope แบบ Trinocular ที่มีกำลังขยายตั้งแต่ 40 ถึง 1000 เท่าของขนาดวัตถุ ในการศึกษา นั้น ได้เข้าใจถึงวิธีการปรับกำลังขยาย (Magnification adjustment) อย่างหยาบและละเอียด การปรับรูผ่านแสง (Diaphragm) การทำความสะอาดดูแลรักษากล้องจุลทรรศน์ให้พร้อมใช้งาน เพราะเลนส์ของกล้องจุลทรรศน์จะไวต่อเชื้อราที่สร้างความเสียหายกับเลนส์โดยตรง
2. การนำชุดกล้อง Eagle Eye Smart Camera Development Kit ที่มีชุดสมองกลฝังตัว Raspberry Pi Computer Module 3+ พร้อมชุดแผงวงจรรวมทั้งภาคจ่ายไฟ และตัวรับรู้ภาพความละเอียด 5 ล้านจุดภาพ ผู้วิจัยได้เลือกชุดกล้องดังกล่าวเนื่องจากชุดกล้องนี้ถูกออกแบบมาใช้งานทางด้าน Machine vision และรวมวงจรส่วนต่าง ๆ ที่จำเป็นต่อการวิจัยไว้ครบถ้วน และมีเสถียรภาพในการทำงานที่ดี ผู้วิจัยได้ปรับปรุงในส่วนของคุณภาพของระบบปฏิบัติการให้รองรับกับระบบโครงข่ายประสาท การติดตั้งชุดเลนส์สำหรับสวมเข้ากับท่อเลนส์ใกล้ตาของกล้องจุลทรรศน์ขนาด 23.2 30 หรือ 30.5 มิลลิเมตร
3. พัฒนาขั้นตอนวิธี (Algorithm) สำหรับการคัดแยกและนับจำนวนเซลล์ของเซลล์เม็ดเลือดด้วยภาษา Python การใช้คลังชุดคำสั่ง OpenCV สำหรับการถ่ายภาพเซลล์เม็ดเลือด และประมวลผลภาพผ่านตัวรับรู้ภาพ การใช้คลังชุดคำสั่ง TensorFlow และ Keras สำหรับการสร้างแบบจำลองโครงข่ายประสาทสำหรับนำไปใช้กับชุดกล้อง และการฝึกโครงข่ายประสาทจากการประมวลผลด้วย Multi-GPU ซึ่งต้องมีความแม่นยำไม่น้อยกว่าร้อยละ 80
4. สร้างชุดข้อมูลภาพถ่ายเซลล์เม็ดเลือดจากกล้องจุลทรรศน์ (Blood cell dataset) รวมการขยายชุดภาพ (Augmentation) ไม่น้อยกว่า 10,000 ภาพที่ได้รับการติดป้าย (Labeling) ประกอบด้วย เม็ดเลือดแดง (Red blood cell : RBC) เม็ดเลือดขาว (White blood cell : WBC) และเกล็ดเลือด (Platelets) ซึ่งถูกแบ่งออกเป็นชุดสำหรับการฝึก (Training set) ร้อยละ 70 ชุดสำหรับทดสอบ (Test set) ร้อยละ 20 และชุดตรวจสอบความถูกต้อง (Validation set) ร้อยละ 10

#### 1.4 วิธีการดำเนินการวิจัย

1. ศึกษาการใช้กล้องจุลทรรศน์ และกลวิธีในการตัดแยกกับนับจำนวนเซลล์ขนาดเล็ก
2. คัดเลือกชนิดเซลล์สำหรับทำวิจัย และศึกษาคุณลักษณะของภาพ
3. ออกแบบชุดกล้อง สร้างวงจรสมองกลฝังตัว และชุดคำสั่งที่เกี่ยวข้อง
4. ศึกษาวิธีการทำงานของระบบโครงข่ายประสาท และทดสอบการทำงาน
5. สร้างขั้นตอนวิธี และฝึกการเรียนรู้ของโครงข่ายประสาทจากภาพถ่ายเซลล์ต่าง ๆ
6. ทดสอบความแม่นยำในการตรวจนับผ่านคอมพิวเตอร์ผ่าน Multi-GPU
7. ทดสอบการตรวจนับผ่านชุดกล้อง และการปลูกถ่ายโครงข่ายประสาท
8. สรุปผลการดำเนินการ และทำบทสรุปการวิจัย เพื่อปิดเล่มวิทยานิพนธ์
9. ตีพิมพ์เผยแพร่ผลงานทางวิชาการระดับนานาชาติในฐานข้อมูลที่ได้รับการยอมรับ

#### 1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. ได้รับชุดกล้อง MicrosisDCN ความละเอียดไม่ต่ำกว่า 5 ล้านพิกเซลที่ถูกพัฒนาขั้นตอนวิธีให้สามารถตัดแยก และนับจำนวนเซลล์ขนาดเล็กผ่านกล้องจุลทรรศน์เชิงซ้อน (Compound microscope) โดยการสวมเข้ากับท่อเลนส์ใกล้ตาขนาด 23.2 30 หรือ 30.5 มิลลิเมตร ด้วยชุดเลนส์ที่รองรับเกลียวแบบ C Mount
2. ได้โมเดลโครงข่ายประสาทที่ได้รับการฝึกสมบูรณ์แล้ว นำไปใช้งานบนชุดกล้อง มีความสามารถในการตัดแยกและนับจำนวนเซลล์ขนาดเล็ก จำนวน 3 ชนิด ได้แก่ เม็ดเลือดแดง (Red blood cell) เม็ดเลือดขาว (White blood cell) และเกล็ดเลือด (Platelets) ซึ่งมีความแม่นยำ (Accuracy) ไม่น้อยกว่าร้อยละ 80
3. ได้ชุดข้อมูลภาพถ่ายเซลล์เม็ดเลือดจากกล้องจุลทรรศน์ (Blood Cell Dataset) รวมการขยายชุดภาพ (Augmentation) ไม่น้อยกว่า 10,000 ภาพ แบ่งออกเป็นชุดสำหรับการฝึก (Training set) ร้อยละ 70 ชุดสำหรับทดสอบ (Test set) ร้อยละ 20 และชุดตรวจสอบความถูกต้อง (Validation set) ร้อยละ 10

## บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง



ภาพที่ 1 แบบร่างระบบชุดกล้อง MicrosisDCN ที่ใช้ในวิทยานิพนธ์

### 2.1 คุณลักษณะของปัญญาประดิษฐ์

ปัญญาประดิษฐ์ (Artificial Intelligence : AI) เป็นเทคโนโลยีการสร้างความสามารถให้แก่เครื่องจักรและคอมพิวเตอร์ ด้วยขั้นตอนวิธีการ (Algorithm) และเครื่องมือ (Tools) เพื่อสร้างชุดคำสั่งที่มีความฉลาด สามารถทำงานเสมือนความสามารถของมนุษย์ที่ซับซ้อนได้ เช่น จดจำ แยกแยะ ให้เหตุผล ตัดสินใจ คาดการณ์ สื่อสารกับมนุษย์ เป็นต้น ในบางกรณีอาจไปถึงขั้นเรียนรู้ได้ด้วยตนเอง [5] ความคิดในการพัฒนาปัญญาประดิษฐ์เกิดขึ้นในช่วงทศวรรษ 90 จากกลุ่มนักวิจัยชั้นแนวหน้า โดยมีเนื้อหาครอบคลุมประกอบด้วย คอมพิวเตอร์อัตโนมัติ (Automatic computers) การเขียนโปรแกรมคอมพิวเตอร์โดยใช้ภาษาคอมพิวเตอร์จะทำได้อย่างไร (How can a computer be programmed to use a machine language) ระบบโครงข่ายประสาท (Neural network) และการพัฒนาด้วยตนเอง (Self-improvement) ซึ่งเป็นผลให้ระบบคอมพิวเตอร์มีความฉลาดในการทำงานมากขึ้น ชนิดของปัญญาประดิษฐ์ (Type of artificial intelligence) แบ่งออกได้เป็น 3 กลุ่ม ได้แก่ 1.) Artificial Narrow Intelligence (ANI) หรือ Weak AI เป็นระบบที่ถูกสร้างขึ้นมาใช้งานเฉพาะด้าน เช่น SIRI ใน iOS หรือ Alexa ของ Amazon เป็นผู้ช่วยในการหาข้อมูลต่าง ๆ แก่ลูกค้า 2.) Artificial General Intelligence (AGI) หรือ Strong AI เป็นปัญญาประดิษฐ์ที่มีความสามารถใกล้เคียงกับมนุษย์ (Human-Level AI) สามารถใช้ตรรกะบนพื้นฐานของเหตุและผล เพื่อวางแผนแก้ไขปัญหา ตลอดจนการเรียนรู้จากประสบการณ์ในอดีต เช่น รถยนต์ระบบขับอัตโนมัติของ



Tesla เป็นต้น และ 3.) Artificial Super Intelligence (ASI) โดย Bostrom จาก Harvard University กล่าวว่าปัญญาประดิษฐ์ชนิดนี้ เรียกได้ว่าเป็น Superintelligence หรือเครื่องจักรทรงภูมิปัญญา (Machine superintelligence) ที่มีขีดความสามารถในการประมวลผลด้วยความเร็วสูง สามารถบูรณาการองค์ความรู้ในศาสตร์ต่าง ๆ มาใช้ โกล้เคียงเชิงสติปัญญาเหนือมนุษย์ แต่ถ้าหากแบ่งตามความสัมพันธ์ สามารถแบ่งออกเป็น 3 กลุ่ม ได้แก่ 1.) ปัญญาประดิษฐ์ (Artificial intelligence) เป็นระบบที่เรียนรู้โดยใช้เหตุและผล มีความสามารถในการปรับตัว 2.) การเรียนรู้ของเครื่อง (Machine learning) เป็นรูปแบบการเรียนรู้ของเครื่องโดยใช้ขั้นตอนวิธี (Algorithm) เพื่อพัฒนาขีดความสามารถจากการสังเคราะห์ข้อมูลที่ได้จากประสบการณ์โดยตรง ซึ่งมีเทคนิคย่อย เช่น โดยมีเทคนิคย่อย คือ Supervised, Unsupervised, Semi-supervised, Reinforcement และ 3.) การเรียนรู้เชิงลึก (Deep learning) เป็นส่วนย่อยของการเรียนรู้ของเครื่องที่ว่าด้วยเรื่องการนำโครงข่ายประสาท (Neural network) มาใช้จัดการชุดข้อมูลต่าง ๆ

ระดับการเรียนรู้ของ AI ตามบริบทของการเรียนรู้ของเครื่อง สามารถแบ่งได้เป็น 3 ระดับ ได้แก่ ระดับที่ 1 Machine Learning ชุดของขั้นตอนวิธี (Algorithm) ที่เครื่องใช้เพื่อเรียนรู้ชุดข้อมูลและประสบการณ์ ในการพัฒนาความสามารถในการตัดสินใจเพื่อสร้างความฉลาดให้แก่ตัวเอง โดยผู้พัฒนาจะเขียนโปรแกรมให้เครื่องเรียนรู้จากข้อมูลด้วยตนเองตามวิธีที่ผู้พัฒนากำหนดไว้ ระดับที่ 2 Machine Intelligence ชุดของขั้นตอนวิธีขั้นสูงที่เครื่องใช้เพื่อเรียนรู้จากประสบการณ์ เช่น Deep Learning แนวโน้มเทคโนโลยีในปัจจุบัน เป็นต้น โดยระดับนี้มีประสิทธิภาพสูงกว่า Machine Learning และต้องการข้อมูลในการเรียนรู้เพื่อพัฒนาความฉลาดมากขึ้น ซึ่งระดับความสามารถในการเรียนรู้ของปัญญาประดิษฐ์ในปัจจุบันจะอยู่ในระดับนี้เป็นส่วนใหญ่ ระดับที่ 3 Machine Consciousness เป็นการออกแบบให้เครื่องสามารถเรียนรู้ได้ด้วยประสบการณ์ของตนเอง โดยไม่ต้องอาศัยข้อมูลภายนอกที่มนุษย์ป้อนเข้าไปให้ ซึ่งเป็นเทคนิคระดับที่สูงที่สุดของปัญญาประดิษฐ์ในปัจจุบัน

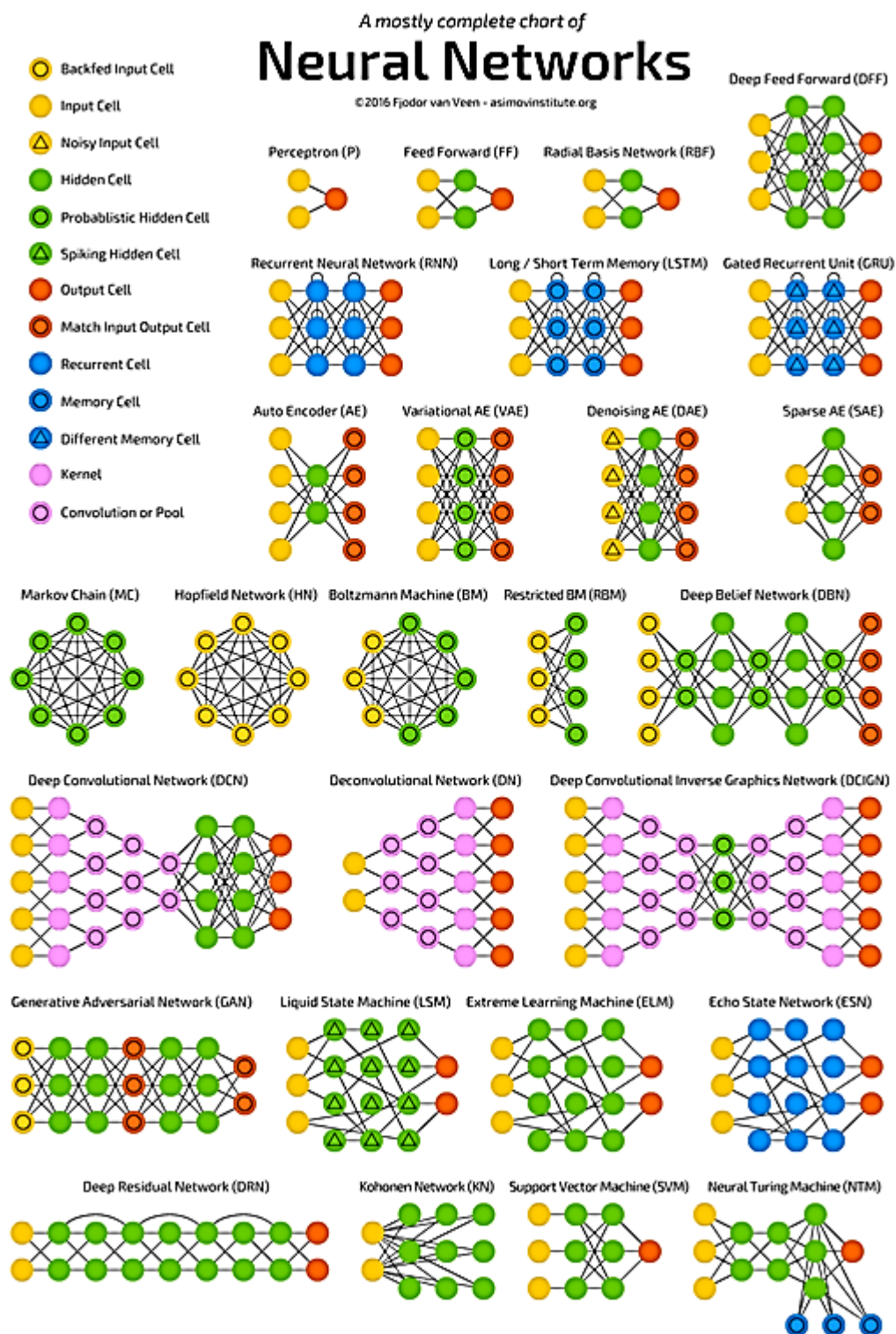
การเรียนรู้ของเครื่องจะมุ่งเน้นไปที่การพัฒนาชุดคำสั่งคอมพิวเตอร์ที่สามารถเข้าถึงข้อมูลและเทคนิคย่อยของการเรียนรู้ของเครื่อง เพื่อให้เข้าใจถึงระบบการทำงานของชุดคำสั่งในงานวิจัยนี้ จึงสามารถจำแนกออกเป็น 4 รูปแบบ ได้แก่

1. การเรียนรู้แบบมีผู้สอน (Supervised learning) เป็นการนำชุดข้อมูลสำหรับการฝึกนั้น (Training set) มาแยกประเภทผลลัพธ์ด้วยการติดป้าย (Labeling) แล้วจึงนำข้อมูลที่ติดป้ายแล้วไปใช้ในการฝึกของเครื่องด้วยขั้นตอนวิธีการสำหรับสร้างโมเดลโครงข่ายประสาทที่ใช้ในการพยากรณ์ผลลัพธ์ (Prediction) เมื่อได้โมเดลที่ผ่านการฝึกแล้ว จะนำไปทดสอบกับชุดข้อมูลตรวจสอบความถูกต้อง (Validation set) เพื่อให้พยากรณ์ผ่านแบบจำลอง (Predictive model) ว่าผลลัพธ์คืออะไร ความแม่นยำเป็นอย่างไร

2. การเรียนรู้แบบไม่มีผู้สอน (Unsupervised learning) เป็นขั้นตอนวิธีการเรียนรู้แบบไม่มีการควบคุมชุดข้อมูลแบบการเรียนรู้แบบมีผู้สอน โดยไม่มีการจัดประเภทหรือติดป้ายกำกับข้อมูล การเรียนรู้ด้วยวิธีดังกล่าวจะเกิดการอนุมานข้อมูลที่ได้รับ โดยการทำความเข้าใจกับชุดข้อมูลเพื่อนำไปสู่การอนุมานผลลัพธ์โดยใช้วิธีสำรวจข้อมูล
3. การเรียนรู้แบบกึ่งมีผู้สอน (Semi-supervised learning) เป็นการเรียนรู้ด้วยขั้นตอนวิธีแบบกึ่งควบคุมที่ผสมผสานการเรียนรู้แบบมีผู้สอน (Supervised) และแบบไม่มีผู้สอน (Unsupervised) อันเนื่องมาจากชุดข้อมูลที่มีทั้งป้ายกำกับและไม่ถูกทำป้ายกำกับ จะทำให้การเรียนรู้ของเครื่องแบบนี้สามารถปรับปรุงความแม่นยำในการเรียนรู้ของเครื่องได้ดีขึ้น
4. การเรียนรู้แบบเสริมแรง (Reinforcement learning) หรือการเรียนรู้แบบเสริมกำลัง โดยกำหนดเป้าหมายให้แก่เครื่อง เรียกว่า “Reinforcement Signal” เพื่อให้เครื่องสร้างทางเลือกในการตัดสินใจหลายรูปแบบ ขึ้นตามสถานะแวดล้อมสถานการณ์ที่แตกต่างกัน หลังจากนั้นเครื่องจะเก็บข้อมูลการตัดสินใจในแต่ละทางเลือกเพื่อเรียนรู้ผลลัพธ์ (Weight) และความเอนเอียง (Bias) ที่เกิดขึ้น จึงทำให้เกิดการประมวลผลด้วยการหาทางเลือกที่มีประสิทธิภาพที่สุดในการบรรลุเป้าหมาย

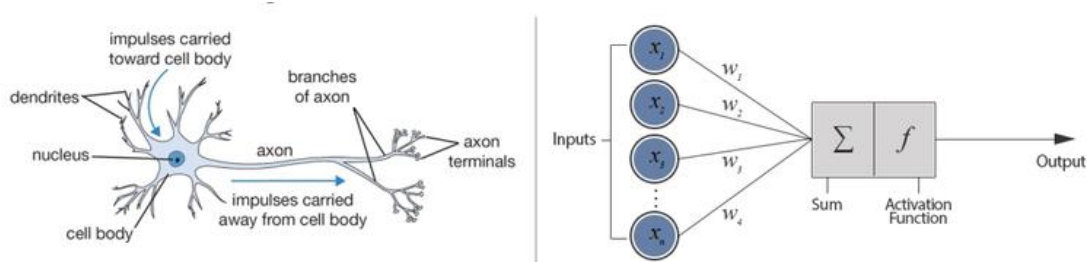
## 2.2 ชนิดของโครงข่ายประสาท

ในส่วนนี้จะเป็นการกล่าวถึงชนิดของโครงข่ายประสาทที่จะนำไปสู่กระบวนการที่ใช้ โครงข่ายประสาทที่พื้นฐานที่สุดคือ Perceptron (P) [6] ซึ่งเป็นพื้นฐานของโครงข่ายเริ่มแรก ทำงานได้เพียงป้อนข้อมูลเข้าผ่านชั้น Input มากกว่า 2 เส้นทาง จากนั้นทำการรวมผลรวมเข้าด้วยกันผ่านฟังก์ชันทางคณิตศาสตร์และให้ผลลัพธ์ออกทางชั้น Output สำหรับโครงข่ายประสาทที่มีความซับซ้อนเพิ่มขึ้น จะเป็นโครงข่ายประสาทแบบไปข้างหน้าหรือ Feed Forward (FF) ถือเป็นโครงข่ายประสาทที่สามารถอธิบายได้ชัดเจนที่สุด เนื่องจากมีเส้นทางข้อมูลไปในทิศทางเดียวเหมือนกับ Perception แตกต่างตรงที่จะมีชั้น Hidden อยู่ในภายในโครงข่ายที่ทำให้มีความซับซ้อนในการประมวลผลมากขึ้น โครงข่ายประสาทแบบวนซ้ำหรือ Recurrent Neural Network (RNN) [7] เป็นโครงข่ายประสาทแรกๆ ที่มีการให้ Feedback ข้อมูลภายในชั้น Hidden ของเส้นทางต่างๆ โดยที่ในแต่ละชั้นจะสามารถคงค่าข้อมูลไว้ภายในเซลล์ต่าง ๆ ได้ (Hold information or Store) ส่งผลให้กระบวนการรับข้อมูลทำได้อย่างเป็นลำดับขั้น (Data sequences) และให้ผลลัพธ์ทางชั้น Output อย่างเป็นลำดับ โครงข่ายประสาทชนิดนี้จะนิยมนำไปใช้ทางด้านข้อมูล เช่น การวิเคราะห์หลักทรัพย์ ความปลอดภัยของข้อมูลทางการเงิน และงานทางด้าน Time sequences เช่น วิเคราะห์สัญญาณในแบบโดเมนเวลา (Time domain) และโดเมนความถี่ (Frequency domain) และการวิเคราะห์สัญญาณเสียงต่าง ๆ



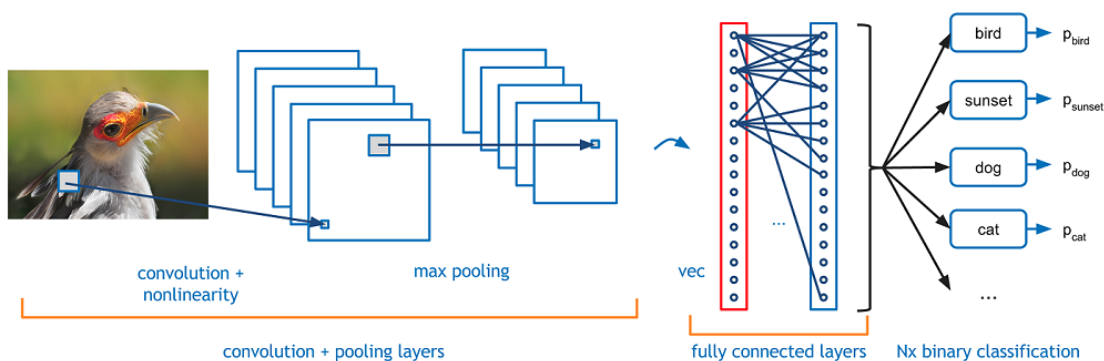
ภาพที่ 2 ชนิดของโครงข่ายประสาท (Neural Network) ที่ใช้งานในปัจจุบัน [8]

หากจะเปรียบเทียบคุณลักษณะของโครงข่ายประสาท จะเหมือนกับเซลล์ประสาท (Neural cells) ภายในสมองที่ทำหน้าที่ในการรับและส่งกระแสประสาทจาก Dendrite ไปยัง Axon โดยเกิดการประมวลผลภายในเซลล์ และเส้นทางเปรียบเสมือน Myelin sheath ที่หุ้ม Axon ไว้ การประมวลผล จะรับข้อมูลจากชั้นที่อยู่ก่อนหน้า และให้ผลลัพธ์ส่งต่อไปยังชั้นถัดไป ซึ่งในแต่ละชั้นจะสามารถดำเนินการทศวรรษทางคณิตศาสตร์ (Mathematics operation) การถ่วงค่าน้ำหนัก (Weight) หรือการหาความเอนเอียง (Bias) โดยไม่ยึดติดกันในแต่ละชั้น เพื่อเลือกเส้นทางข้อมูลที่เหมาะสมที่สุด ทั้งหมดที่กล่าวถึงนี้จะเรียกโดยรวมว่าโมเดลโครงข่ายประสาท



ภาพที่ 3 การเปรียบเทียบเซลล์ประสาท และโครงสร้างของโครงข่ายประสาท

ในส่วนถัดมาเป็นจะกล่าวถึงชนิดโครงข่ายประสาทที่นำมาใช้ในงานวิจัยนี้ โครงข่ายประสาทแบบคอนโวลูชัน หรือ Convolutional neural network (CNN) [9] เป็นการจำลองมาจากการมองเห็นภาพของมนุษย์ที่มองเห็นที่เป็นส่วยย่อย ๆ จากนั้นนำกลุ่มของพื้นที่ย่อย ๆ มาผสานกัน เพื่อตัดสินใจว่าสิ่งที่มองเห็นนั้นคืออะไร โดยจะมีการแยกคุณลักษณะของพื้นที่ย่อยนั้น เช่น เส้นขอบของภาพ สีที่ตัดกันระหว่างวัตถุ ซึ่งมนุษย์จะรับรู้ว่ามีส่วนไหนที่สีตัดกัน จากความสนใจในพื้นที่ และบริเวณรอบวัตถุประกอบกัน ในการทำงานจะใช้การคำนวณเชิงคณิตศาสตร์ในการระบุว่าวัตถุชิ้นนั้นคืออะไร ด้วยคอนโวลูชันเชิงพื้นที่ (Spatial convolution) จากภาพถ่ายที่ได้ และถูกนำมาประมวลผลภาพเริ่มจากการกำหนดค่าเคอร์เนล (Kernel) ที่ช่วยดึงคุณลักษณะที่ใช้ในการรู้จำวัตถุออกมา ซึ่งตัวกรองจะถูกทาบบลงในจุดภาพ (Pixel) แรกของภาพนั้น จากนั้นจะถูกเลื่อนไปทาบบนจุดภาพอื่นในภาพจนครบทุกจุดภาพ และใช้ Max pooling หาค่าสูงสุดในบริเวณที่ตัวกรองทาบบอยู่มาเป็นผลลัพธ์ โดยจะเตรียมตัวกรองในลักษณะเดียวกับการแยกคุณลักษณะ (Feature extraction) มาทาบบนข้อมูลแล้วเลือกค่าที่สูงสุดบนตัวกรองนั้นมาเป็นผลลัพธ์ใหม่ และจะเลื่อนตัวกรองไปตาม Stride ที่กำหนดไว้ ซึ่งกระบวนการทั้งหมด จะใช้กับภาพของเซลล์ที่ได้จากชุดกล้องที่ถูกติดตั้งไว้ และฝึกให้โครงข่ายประสาทคัดแยกและนับจำนวนได้ มีการปรับค่าน้ำหนักในการฝึกให้โครงข่ายประสาทสามารถแสดงผลที่ออกมา ซึ่งกระบวนการฝึกจะถูกทำซ้ำ เพื่อปรับค่าน้ำหนักหลาย ๆ รอบ เพื่อให้ความแม่นยำ (Accuracy) สูงขึ้น และมีค่าการสูญเสีย (Loss) ลดลง



ภาพที่ 4 แนวคิดของโครงข่ายประสาทแบบคอนโวลูชัน (CNN) [10]

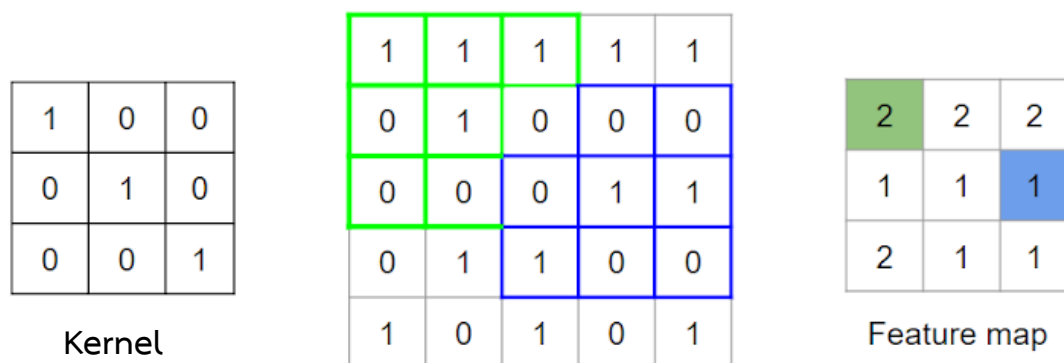
### 2.3 ทฤษฎีทางการเรียนรู้ของเครื่อง

การเรียนรู้ของเครื่อง (Machine learning) คือชุดคำสั่งขั้นตอนวิธีที่มีการทำงานคล้ายคลึง การเลียนแบบโครงข่ายประสาท (Neuron network) ภายในสมองของมนุษย์ ถูกสร้างขึ้นโดยการนำโครงข่ายประสาทหลาย ๆ ชั้น มาต่อกัน ซึ่งชั้นแรกทำหน้าที่ในการรับข้อมูลจากชั้น Input และชั้นสุดท้าย แล้วให้ผลลัพธ์การประมวลผลออกทางชั้น Output ในจำนวนชั้นย่อยภายในโครงข่ายประสาทจะเรียกว่าชั้น Hidden ระบบของชุดกล้อง MicroSDCN จะใช้เป็นโครงข่ายประสาทแบบคอนโวลูชัน (CNN) ที่ถูกจำลองมาจากการมองเห็นภาพของมนุษย์ที่มองพื้นที่เป็นส่วย่อย ๆ จากนั้นนำกลุ่มของพื้นที่ย่อย ๆ มาผสมกัน เพื่อตัดสินใจว่าสิ่งที่มองเห็นนั้นคืออะไร โดยจะมีการแยกคุณลักษณะของพื้นที่ย่อยนั้น เช่น เส้นขอบของภาพ สีที่ตัดกันระหว่างวัตถุ ซึ่งมนุษย์จะรับรู้ส่วนไหนที่สีตัดกัน ในการทำงานจะใช้การคำนวณเชิงคณิตศาสตร์ในการระบุว่าวัตถุนั้นคืออะไร โดยการคำนวณตามแนวคิดนี้ใช้หลักการเดียวกับคอนโวลูชันเชิงพื้นที่ (Spatial convolution) จากภาพถ่ายที่ได้และถูกนำมาประมวลผลภาพ (Image processing) เริ่มจากการกำหนดค่าเคอร์เนล (Kernel) หรือตัวกรอง (Filter) ที่ช่วยดึงคุณลักษณะที่ใช้ในการรู้จำวัตถุออก ซึ่งตัวกรองจะถูกทาบบลงในพิกเซลแรกของภาพข้อมูลเข้า จากนั้นจะถูกเลื่อนไปทาบบนพิกเซลอื่นในภาพทีละพิกเซลจนครบทุกพิกเซล และใช้ Max pooling หาค่าสูงสุดในบริเวณที่ตัวกรองทาบบอยู่มาเป็นผลลัพธ์ โดยจะเตรียมตัวกรองในลักษณะเดียวกับการแยกคุณลักษณะของข้อมูลของ CNN มาทาบบนข้อมูลแล้วเลือกค่าที่สูงสุดบนตัวกรองนั้นมาเป็นผลลัพธ์ใหม่ และจะเลื่อนตัวกรองไปตาม Stride ที่กำหนดไว้ กระบวนการทั้งหมดจะใช้กับภาพของเซลล์ขนาดเล็กที่ได้จากชุดกล้องที่ถูกติดตั้งไว้ และฝึกให้โครงข่ายประสาทสามารถคัดแยกและนับจำนวนเซลล์ขนาดเล็กได้

ขั้นตอนการสร้างโมเดลโครงข่ายประสาทสำหรับการเรียนรู้ของเครื่อง จะประกอบไปด้วย 3 ขั้นตอน คือ 1.) Feature extraction คือการนำชุดข้อมูล (Dataset) ที่สนใจมาเป็นปัญหาตั้งต้นในการสร้างโมเดลมาแปลงเป็นข้อมูลเชิงตัวเลขสำหรับการประมวลผล โดยอาศัยกระบวนการทางคณิตศาสตร์เข้าช่วย อย่างเช่นโครงข่ายประสาทแบบคอนโวลูชัน (CNN) จะใช้คอนโวลูชันเชิงพื้นที่ (Spatial convolution) ด้วยตัวกรอง (Filter) 2.) Regularization เป็นการจัดการความซับซ้อนของข้อมูล เพื่อนำไปสู่การสร้างโมเดลที่มีประสิทธิภาพ ซึ่งจะมีกระบวนการสำหรับปรับแต่งข้อมูล เช่น การลดหรือเพิ่ม Noise เป็นต้น 3.) Cross-Validation ชุดโมเดลที่สร้างขึ้นจะต้องได้รับการตรวจสอบความถูกต้อง โดยการนำชุดข้อมูลที่แยกส่วนจากชุดข้อมูลฝึกมาทำการพยากรณ์ผลลัพธ์ (Prediction) แล้วให้ผลถูกต้องแม่นยำมากที่สุด

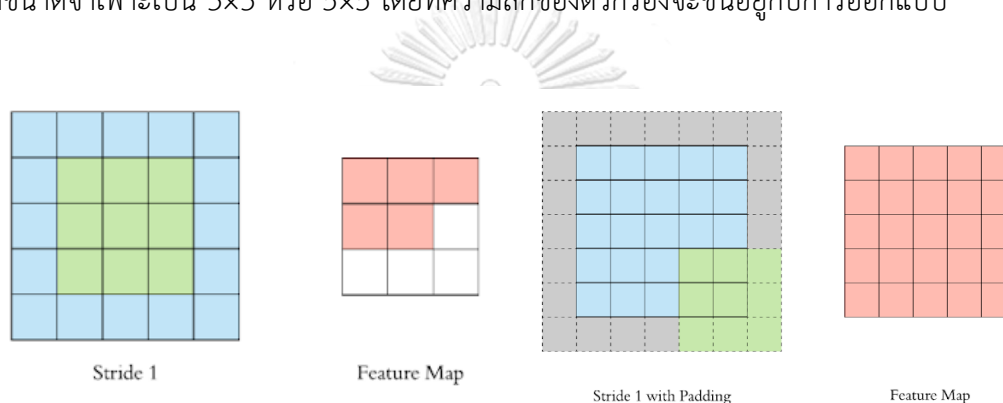
การแยกคุณลักษณะของข้อมูล (Feature extraction) จะทำในพื้นที่ย่อยของภาพ เช่น การตัดกันของเส้นขอบ ความแตกต่างของค่าสี โดยการใช้ตัวกรอง (Filter) ในการจำแนกและดึงลักษณะที่สนใจให้แสดงออกมา ซึ่งในทางปฏิบัติสามารถใช้ตัวกรองมากกว่าหนึ่งตัว ในการกรองลักษณะเฉพาะของภาพบนพื้นที่กว้างของข้อมูลให้ได้คุณลักษณะในการพิจารณามากขึ้น สำหรับตัวกรองภาพในเชิงเลขนั้น จะเป็น 2-Dimension ตามขนาดของพื้นที่ที่ถูกพิจารณา เช่น หากต้องการพิจารณาเส้นทแยงสีดำในภาพ ตัวกรองขนาด  $3 \times 3$  จะเป็นลักษณะดังภาพ บริเวณตรงกลางตัวกรองนั้นจะเป็นค่าผลรวมของข้อมูลที่ถูกทับกับตัวกรองบนจุดภาพ (Pixel) ของข้อมูล เรียกว่า Anchor กล่าวคือจากขนาดตัวกรอง  $3 \times 3$  เมื่อทับลงบนภาพขนาด 9 จุดภาพแล้ว จะรวมผลลัพธ์ในแต่ละช่องกับตัวกรองทั้ง 9 ช่อง และค่าออกมา จากนั้นจะเลื่อนตัวกรองไปยังจุดภาพถัดไปจนกระทั่งครบทุกจุดภาพจนไม่สามารถเลื่อนต่อไปได้อีก ทำให้ได้ผังคุณลักษณะ (Feature map)

### จุฬาลงกรณ์มหาวิทยาลัย



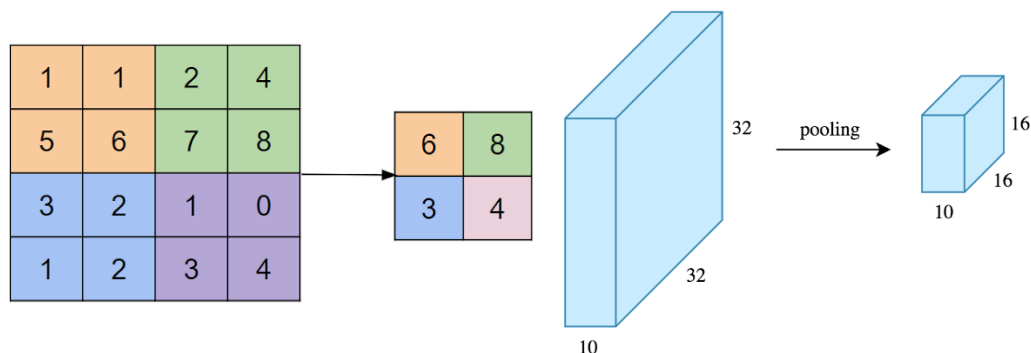
ภาพที่ 5 การสร้างผังการแยกคุณลักษณะของจุดภาพโดยการใช้ตัวกรอง

ซึ่งกระบวนการเลื่อนจะถูกเรียกว่า Stride โดยการกำหนดว่าจะให้เลื่อนตัวกรองคอนโวลูชันเป็นจำนวนกี่ step โดยปกติแล้วจะให้เลื่อนทีละ 1 step แต่ถ้าหากเลื่อนทีละ 2 step จะทำให้ฟังก์ชันคุณลักษณะมีขนาดลดลงเหลือ  $2 \times 2$  แทน สำหรับการทำคอนโวลูชันโดยให้ขนาดของฟังก์ชันคุณลักษณะเท่ากับขนาดของพื้นที่จุดภาพสามารถทำได้โดยเพิ่มพื้นที่สีเทาด้านข้าง และได้เติมค่า 0 หรือค่าอื่นๆ เข้าไป เรียกว่า Padding อย่างไรก็ตามการนำเสนอถึงวิธีการทำคอนโวลูชันในเชิง 2-Dimension เป็นตัวอย่างเพื่อให้เข้าถึงวิธีการ ในทางปฏิบัติภาพในหนึ่งจุดภาพจะประกอบด้วยแม่สี 3 ค่า ได้แก่ สีแดง (Red) สีเขียว (Green) และสีน้ำเงิน (Blue) จึงต้องทำคอนโวลูชันในเชิง 3-Dimension ประกอบด้วยขนาดความสูง (Height) ความยาว (Width) และความลึก (Depth) โดยที่ตัวกรองคอนโวลูชันนั้น จะมีขนาดจำเพาะเป็น  $3 \times 3$  หรือ  $5 \times 5$  โดยที่ความลึกของตัวกรองจะขึ้นอยู่กับการออกแบบ



ภาพที่ 6 วิธีการทำ Stride และ Stride padding เพื่อให้ได้ฟังก์ชันการแยกคุณลักษณะ

หลังจากผ่านการทำคอนโวลูชันจนเสร็จแล้ว จะต้องทำการลดขนาด (Reduce dimension) เพื่อลดจำนวนของตัวแปรที่เกี่ยวข้องลง เรียกว่า Pooling ส่งผลให้การใช้ระยะเวลาในการฝึก Model ลดลง เกิด Overfitting ลดลงเช่นกัน และยังสามารถลดจำนวนชั้น (Layer) ของโครงข่ายประสาทตามขนาดที่ลดลงของฟังก์ชันคุณลักษณะ (Feature map) โดยการลดขนาดความสูง (Height) ความยาว (Width) แต่คงค่าความลึก (Depth) ไว้เท่าเดิม สำหรับการทำ Pooling จะมีวิธีการที่น่าสนใจ คือ Mean pooling และ Max pooling ซึ่ง Max pooling เป็นตัวกรองที่ได้รับความนิยมมากที่สุด มีหลักการโดยการหาค่าสูงสุดในบริเวณที่ตัวกรองถูกทาบบอยู่ และเลือกค่าสูงสุดดังกล่าวมาเป็นผลลัพธ์ของฟังก์ชันคุณลักษณะ ซึ่งจะมีการเลื่อนตามค่า Step ที่กำหนดไว้ไปเรื่อย ๆ เช่นเดียวกับการแยกคุณลักษณะของข้อมูล ขนาดของตัวกรอง Max pooling จะถูกเรียกว่า Pool size หรือ Windows size ตัวอย่างนี้จะแสดงให้เห็นการใช้ Max pooling ขนาด  $2 \times 2$  หากสังเกตขนาดข้อมูลกับขนาดตัวกรองเลื่อน 2 step จะไม่เกิดการซ้อนทับ (Overlapping) เพราะขนาดข้อมูลเป็น  $4 \times 4$  แต่ในทางปฏิบัติแล้วขนาดของชั้น Pooling จะมีขนาดใหญ่ เช่น  $32 \times 32 \times 10$  เมื่อทำการลดขนาดโดยใช้ Max pooling จะทำให้ขนาดลดลงเหลือ  $16 \times 16 \times 10$  โดยที่ความลึกจะไม่เปลี่ยนแปลง



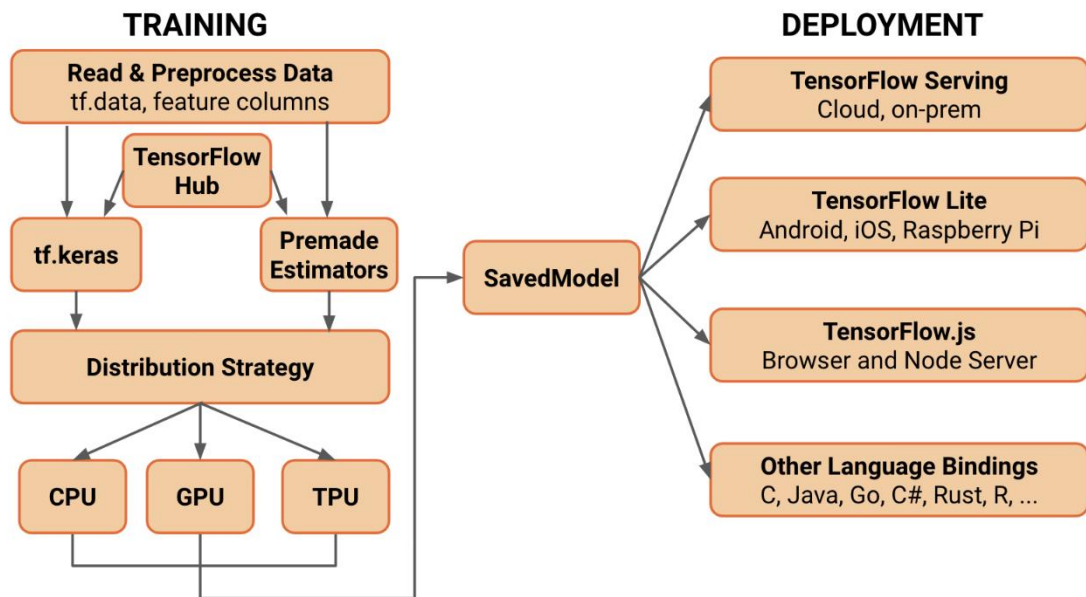
ภาพที่ 7 วิธีการสร้างฟังก์ชันแยกคุณลักษณะด้วยวิธีการ Pooling

## 2.4 เทคโนโลยีที่นำมาใช้ในงานวิจัย

การพัฒนากระบวนการเรียนรู้ของเครื่อง (Machine learning) ในชุดกล้อง MicrosisDCN จะดำเนินการบน TensorFlow ซึ่งเป็นคลังชุดคำสั่งสำหรับใช้พัฒนาการเรียนรู้ของเครื่อง ที่พัฒนาโดยกูเกิลแบบ Open source [11] รองรับการพัฒนาด้วยภาษา Python และ C/C++ มาพร้อมกับ TensorBoard ซึ่งเป็นชุดคำสั่งจำลองการทำงานของกระบวนการฝึก (Training) ของ TensorFlow ตัวอย่างของการนำ TensorFlow ไปใช้ ได้แก่ Image captioning, Translation, Object detection, Search engine เป็นต้น ทำงานได้โดยการรับข้อมูลเป็นอาร์เรย์หลายมิติ หรือที่เรียกกันว่า Tensors มีหน้าที่จัดเรียงลำดับการประมวลผลเป็น Flowchart ข้อมูลที่ถูกป้อนไปจะผ่านกระบวนการจนออกมาเป็นผลลัพธ์ มีการรวบรวม API (Application Programming Interface) ต่าง ๆ สร้างสถาปัตยกรรมแบบการเรียนรู้เชิงลึก (Deep learning) เช่น CNN นอกจากนี้ยังปรับสเกลประมวลผลระหว่าง CPU และ GPU ได้

องค์ประกอบของ Tensorflow ประกอบด้วย 3 ส่วน ได้แก่ 1.) Tensors มีหน้าที่จัดเรียงลำดับการประมวลผลเป็น Flowchart ข้อมูลที่ถูกป้อนไป เช่น การเก็บค่าหรือส่งผ่านค่าระหว่าง Node ของโครงข่ายประสาทให้ได้ค่า Weight หรือ Bias ผ่านกระบวนการทางคณิตศาสตร์ เช่น การปฏิบัติการทางเวกเตอร์ (Vector operation) การปฏิบัติการทางเมทริกซ์ (Matrix operation) เป็นต้น การคำนวณทั้งหมดจะเกิดขึ้นภายใน Graph จำนวนค่าที่ได้ต่อหนึ่ง Tensor จะถูกเรียกว่า Shape ซึ่งมีขนาดที่แตกต่างกันออกไป 2.) Operation node กลุ่มของชั้นของโครงข่ายประสาทที่เกิดการประมวลผลข้อมูลด้วย CPU หรือ GPU 3.) Graph เป็นลำดับการประมวลผลที่รวม Operation node ทั้งหมดเชื่อมระหว่างกันและกัน ทำให้เกิดการประมวลผลอย่างต่อเนื่อง

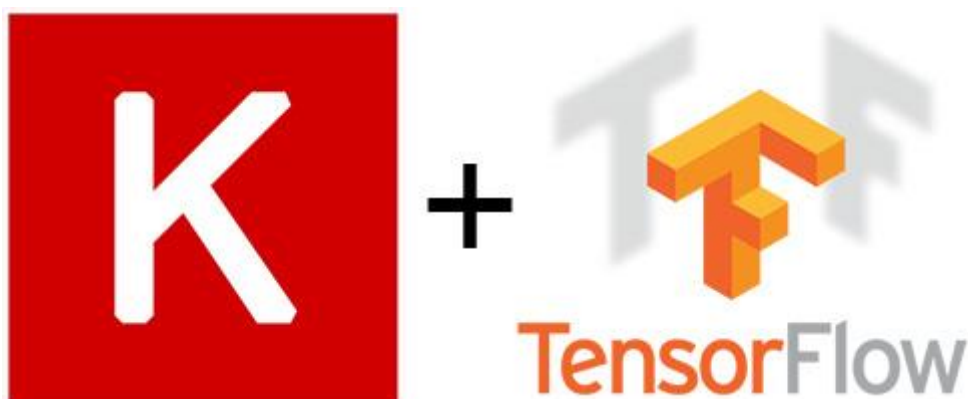




ภาพที่ 8 ฝั่งแสดงโครงสร้างของ Tensorflow core [11]

ข้อได้เปรียบของ Tensorflow คือการรวม API (Application Programming Interface) ของโปรแกรมต่าง ๆ เข้าไว้ด้วยกัน ส่งผลทำให้การสร้างระบบการเรียนรู้เชิงลึก (Deep learning) และการเรียนรู้ของเครื่อง (Machine learning) ทำได้ง่าย อย่างไรก็ตาม Tensorflow ต้องการอาศัยระยะเวลาการเรียนรู้พอสมควร จึงมีการพัฒนา Platform ที่ช่วยให้การพัฒนาโครงข่ายประสาททำได้ง่ายมากยิ่งขึ้น สามารถปรับแต่งโมเดลโครงข่ายประสาทให้ตรงความต้องการ ตลอดจนสามารถพัฒนาลำดับชั้น (Layer) ของโครงข่ายประสาทได้ เรียกว่า Keras

จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY



ภาพที่ 9 สัญลักษณ์ของ Keras และ Tensorflow [12]

Keras เป็นคลังชุดคำสั่งแบบเปิดเผยแพร่คำสั่งทางด้านโครงข่ายประสาทที่พัฒนาได้ด้วยภาษา Python เป็นหนึ่งใน API ระดับสูงที่รองรับการคำนวณโครงข่ายประสาทแบบหลายแบ็คเอนด์ (Backend computation with multiple engine) [12] และสามารถทำงานครบวงจรการทำงานของ Tensorflow ได้โดยตรง จุดประสงค์ของการพัฒนา Keras เพื่อตอบสนองความเป็นมิตรต่อผู้ใช้ (User friendliness) การใช้งานแบบแยกส่วนได้ (Modularity) และพัฒนาขยายออกได้ (Easy extensibility) นอกจากนี้ Keras ยังรองรับโครงข่ายประสาท RNN (Recurrent neural network) และ CNN (Convolutional neural network) รวมถึงตัวกรองชนิดต่าง ๆ เช่น Dropout, Pooling, Batch normalization เป็นต้น Keras สามารถพัฒนาการเรียนรู้เชิงลึกบนระบบที่หลากหลาย เช่น การใช้โมเดลโครงข่ายประสาทบน Smartphone (iOS and Android), Web application, Java Virtual Machine และ Tensor processing units (TPU) ร่วมกับ CUDA (Compute Unified Device Architecture) ซึ่งเป็น API ที่พัฒนาขึ้นโดยบริษัท Nvidia เพื่อให้สามารถเข้าถึงศักยภาพการประมวลผลด้วย GPU (Graphic processing unit) โดยปกตินั้นการประมวลผลทางด้านโครงข่ายประสาทจะทำผ่าน CPU ที่มีจำนวนแกน (Core) จำกัดเป็นแบบ Sequential serial processing ในขณะที่ GPU จะมีจำนวนแกนขนาดเล็กเป็นจำนวนมาก เพื่อการประมวลผลแบบกระจาย (Parallel processing) หลายชุดด้วยคำสั่งเดียว Single Instruction Multiple Data stream (SIMD)

ตัวอย่างการพัฒนา Convolution neural network ด้วย Keras นั้นสามารถพัฒนาด้วยภาษา Python ได้โดยตรง โครงสร้างของ CNN นี้ประกอบด้วย Convolution + Max pooling 4 ชั้น ตามด้วย Fully connected 2 ชั้น ฝั่ง Input จะเป็นข้อมูลรูปภาพขาเข้า และ Output จะเป็นผลลัพธ์ที่ให้ค่าแบบ 0 หรือ 1 ออกมา ซึ่งมีโปรแกรมภาษา Python ดังต่อไปนี้

```

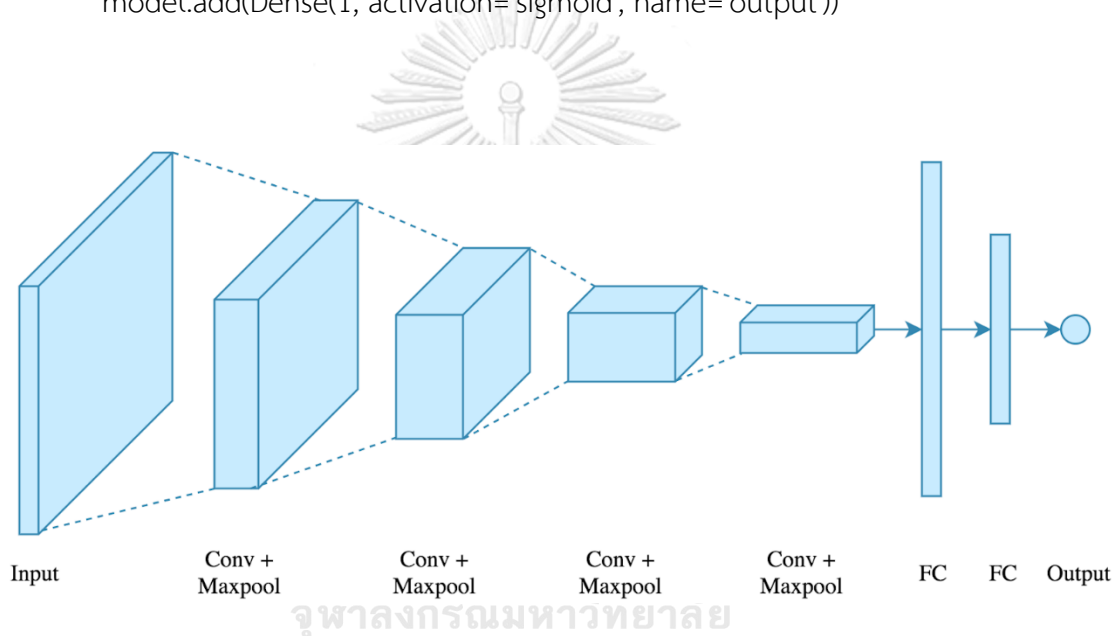
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', padding='same',
name='conv_1',
input_shape=(150, 150, 3)))
model.add(MaxPooling2D((2, 2), name='maxpool_1'))
model.add(Conv2D(64, (3, 3), activation='relu', padding='same',
name='conv_2'))
model.add(MaxPooling2D((2, 2), name='maxpool_2'))
model.add(Conv2D(128, (3, 3), activation='relu', padding='same',
name='conv_3'))

```

```

model.add(MaxPooling2D((2, 2), name='maxpool_3'))
model.add(Conv2D(128, (3, 3), activation='relu', padding='same',
name='conv_4'))
model.add(MaxPooling2D((2, 2), name='maxpool_4'))
model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(512, activation='relu', name='dense_1'))
model.add(Dense(128, activation='relu', name='dense_2'))
model.add(Dense(1, activation='sigmoid', name='output'))

```



ภาพที่ 10 ภาพอธิบายลำดับชั้น (Layer) ของโครงข่ายประสาทคอนโวลูชัน (CNN) บน Keras

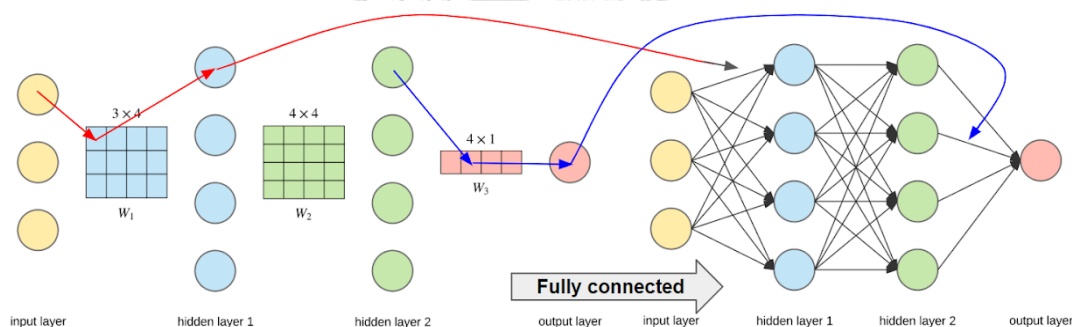
ส่วนประกอบของโครงข่ายประสาทสามารถอธิบายได้เป็น 4 ส่วน

1.) Conv2D เป็นคำสั่งที่ใช้ในการสร้างชั้นคอนโวลูชัน (Convolution layer) ภายในสามารถกำหนดค่าตัวแปร (Parameter) ลำดับแรกเป็นจำนวนของตัวกรอง (Filter count) กำหนดเป็น 32 ตัวกรอง ลำดับที่สองเป็นขนาดของตัวกรอง (Filter size) กำหนดขนาดเป็น 3x3 ลำดับที่สามเป็นการเลือก Padding ซึ่งจะมี 2 แบบ คือ Same หมายถึง Padding ตามจำนวนขอบ (Edge) และ Valid คือ ไม่ทำ Padding กำหนดเป็น Valid การเลื่อนของตัวกรองถูกกำหนดเป็น 1 step ตามค่ามาตรฐาน ลำดับที่ 4 เป็นชื่อของชั้นนี้ กำหนดเป็น conv\_1 ลำดับสุดท้ายเป็นขนาดของ Input กำหนดเป็นขนาดความสูง (Height) เป็น 150 ความยาว (Width) เป็น 150 และความลึก (Depth) เป็น 3 ตามค่าจุดภาพที่ได้กล่าวมาก่อนหน้านี้

2.) MaxPooling2D เป็นคำสั่งที่ใช้ในการสร้างชั้น Max pooling ภายในสามารถกำหนดค่าตัวแปรได้เฉพาะขนาดของ Pool size หรือ Windows size มักนิยมกำหนดเป็นขนาด  $2 \times 2$

3.) Flatten เป็นชุดคำสั่งในการสร้างชั้นสำหรับรับ Output ที่ได้จากชั้นคอนโวลูชันส่งต่อไปชั้น Fully connected ซึ่งเป็นกระบวนการเชื่อมโยงแต่ละ Node ของโครงข่ายประสาทเข้าด้วยกัน โดยสมบูรณ์ ใน Keras สามารถใช้คำสั่ง Dense ที่ปรากฏอยู่ในโปรแกรมภาษา Python อธิบายถึงหลักการของ Fully connected ตัวอย่างเช่น การเชื่อมต่อระหว่าง 3 node และ 4 node ทำได้โดยการคูณเมทริกซ์ Input vector ขนาด  $1 \times 3$  กับ Weight  $3 \times 4$  ของเมทริกซ์  $W_1$  ผลลัพธ์เชิง Dot product จะปรากฏเป็นปลายทางของ Node สีน้ำเงินในชั้น Hidden 1 และทำเช่นนี้ จนกระทั่งถึงชั้น Output

4.) Dropout เป็นส่วนที่มีความสำคัญในการปรับแต่งโครงข่ายประสาท เป็นการละทิ้ง Node ที่ความสำคัญต่ำหรือไม่มีความจำเป็น เพื่อให้โครงข่ายประสาทมีความแม่นยำสูงขึ้น (Accuracy) และป้องกันการเกิด Overfitting ในระหว่างการฝึกโครงข่ายประสาทที่สร้างขึ้นได้



ภาพที่ 11 แสดงขั้นตอนการทำ Fully connected โดยใช้การคูณเมทริกซ์ผลคูณแบบดอท

CHULALONGKORN UNIVERSITY

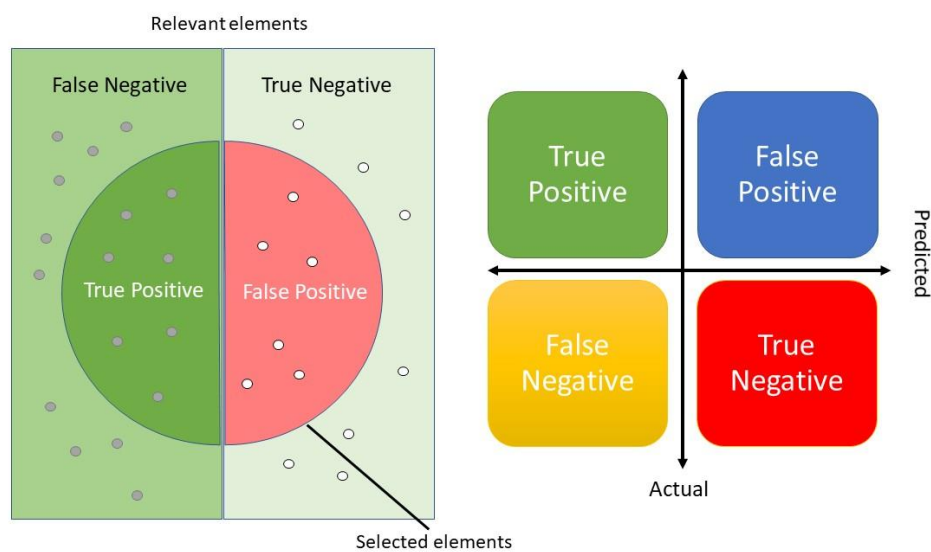


ภาพที่ 12 สัญลักษณ์ของ OpenCV (Open Source Computer Vision Library) [13]

ส่วนสำคัญที่ทำให้โครงข่ายประสาทสามารถทำงานได้นั้น คือ การถ่ายผ่านตัวรับรู้ภาพ (Image sensor) ชุดคำสั่งที่มีบทบาทในการพัฒนาอุปกรณ์ MicrosisDCN ได้แก่ OpenCV (Open Source Computer Vision Library) เป็นคลังชุดคำสั่งสำหรับการเขียนโปรแกรมที่มีความเกี่ยวข้องกับงานทางด้านคอมพิวเตอร์วิทัศน์ (Computer vision) และการประมวลผลภาพ (Image processing) พัฒนาโดย Intel บนภาษา C และ C++ [13] รวมถึงมี API ให้ทำงานร่วมกับภาษาอื่น ๆ เช่น Matlab, Ruby, Perl และ Python จึงสามารถผสมการทำงานร่วมกับการเรียนรู้ของเครื่องได้เป็นอย่างดี มีตัวอย่างการประยุกต์ใช้ที่หลากหลาย เช่น ชุดเครื่องมือเชิง 2 มิติและ 3 มิติ (2D and 3D feature toolkits), ระบบรู้จำใบหน้า (Facial recognition system), การจดจำท่าทาง (Gesture recognition) เป็นต้น



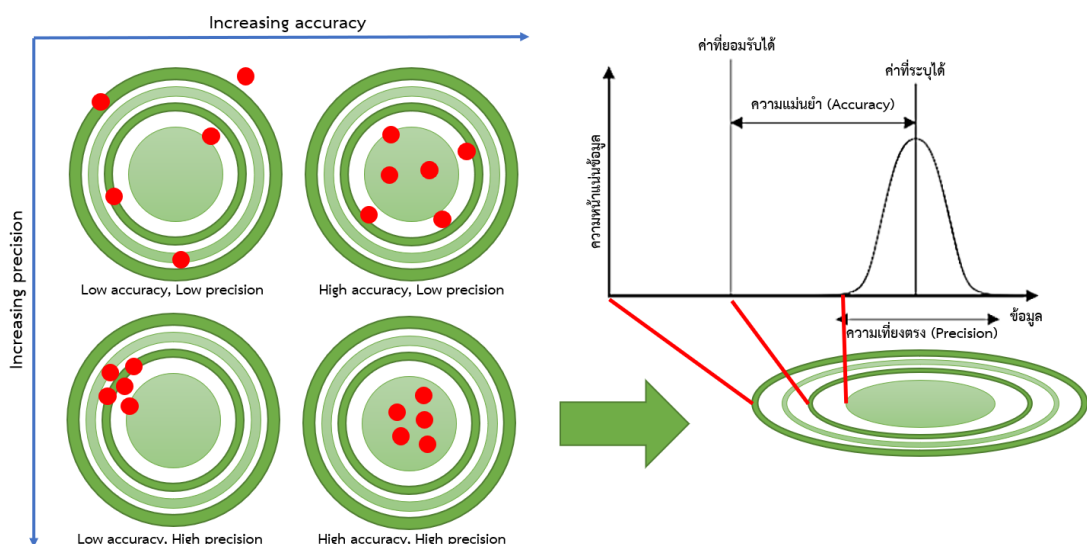
## 2.5 วิธีการประเมินผลความแม่นยำของโครงข่ายประสาท



ภาพที่ 13 ผังแสดงการจัดกลุ่มข้อมูลแบบซัพเซต (ซ้าย) และ Confusion matrix (ขวา)

วิธีการประเมินผลความแม่นยำของโครงข่ายประสาท หรือโมเดลการเรียนรู้ของเครื่อง จะมีหลายรูปแบบ วิธีการคำนวณความแม่นยำและความคลาดเคลื่อนนั้น จะใช้ข้อมูลที่ได้จากค่าจริง (Actual) และค่าพยากรณ์ (Predicted) นำมาแบ่งตามผลการพยากรณ์จัดกลุ่ม (Classifier) จากข้อมูลลงใน Confusion matrix คือเครื่องมือในการประเมินผลการจัดกลุ่ม แบ่งออกเป็น 4 กลุ่ม ได้แก่ True positive (TP), False positive (FP), False negative (FN) และ True negative (TN)

- 1.) True positive (TP) คือ จำนวนผลการพยากรณ์อย่างถูกต้อง เมื่อข้อมูลนั้นมีค่าเป็นบวก แล้วคลาสระบุว่าเป็นค่าบวก ตัวอย่างเช่น ความเป็นจริง “ขโมยขึ้นบ้าน” กล้องวงจรปิด “พบขโมย” ผลลัพธ์ “ป้องกันทรัพย์สินไว้ได้” แสดงว่าข้อมูลจัดกลุ่มเป็นคลาสบวกอย่างถูกต้อง
- 2.) False positive (FP) คือ จำนวนผลการพยากรณ์ไม่ถูกต้อง เมื่อข้อมูลนั้นมีค่าเป็นลบ แต่คลาสระบุว่าเป็นค่าบวก ทั้งที่จริงแล้วเป็นค่าลบ ตัวอย่างเช่น ความเป็นจริง “ไม่มีคนเดินผ่าน” กล้องวงจรปิด “พบขโมย” ผลลัพธ์ “แจ้งความเท็จ” แสดงว่าข้อมูลจัดกลุ่มเป็นเป็นคลาสบวกผิด
- 3.) False negative (FN) คือ จำนวนผลการพยากรณ์ไม่ถูกต้อง เมื่อข้อมูลนั้นมีค่าเป็นบวก แต่คลาสระบุว่าเป็นค่าลบ ทั้งที่จริงแล้วเป็นค่าบวก ตัวอย่างเช่น ความเป็นจริง “ขโมยขึ้นบ้าน” กล้องวงจรปิด “ไม่มีคนเดินผ่าน” ผลลัพธ์ “ของหายทั้งบ้าน” แสดงว่าข้อมูลจัดกลุ่มเป็นคลาสนลบผิด
- 4.) True negative (TN) คือ จำนวนผลการพยากรณ์อย่างถูกต้อง เมื่อข้อมูลนั้นมีค่าเป็นลบ แล้วคลาสระบุว่าเป็นค่าลบ ตัวอย่างเช่น ความเป็นจริง “ไม่มีคนเดินผ่าน” กล้องวงจรปิด “ไม่มีคนเดินผ่าน” ผลลัพธ์ “สถานการณ์ปกติ” แสดงว่าข้อมูลจัดกลุ่มเป็นคลาสนลบได้อย่างถูกต้อง



ภาพที่ 14 ผังแสดงความสัมพันธ์ระหว่างความเที่ยงตรง และความแม่นยำ

จากผังการแบ่งข้อมูลเป็น 4 กลุ่มด้วย Confusion matrix ในภาพที่ 13 จึงสามารถอธิบายความสัมพันธ์ในการหาค่าค่าเที่ยงตรง (Precision) และความแม่นยำ (Accuracy) ค่าการเรียกคืน (Recall) และค่าคะแนน F1 และ F เบต้า โดยใช้ค่าจากทั้ง 4 กลุ่ม คือ TP, FP, FN และ TN

### 2.5.1 ค่าความแม่นยำ

ค่าความแม่นยำ (Accuracy) เปิดวิธีการบ่งบอกความสามารถในการพยากรณ์ได้อย่างถูกต้องสนใจผลการพยากรณ์ของคลาสบวกและคลาสลบ ว่ามีค่าเป็นเท่าไรของกลุ่มข้อมูลทั้งหมด หากค่า Accuracy มีค่าสูงเข้าใกล้ 1 แสดงว่าโมเดลโครงข่ายประสาทนั้นมีความแม่นยำสูง

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

### 2.5.2 ค่าความเที่ยงตรง

ค่าความเที่ยงตรง (Precision) เป็นวิธีการบ่งบอกแนวโน้มการรวมกลุ่มกันของค่าพยากรณ์ที่ถูกต้องของคลาสบวก ว่ามีค่าเป็นเท่าไรของกลุ่มข้อมูลคลาสบวกทั้งหมด หากค่า Precision มีค่าเข้าใกล้ 1 แสดงว่าโมเดลโครงข่ายประสาทนั้นมีความเที่ยงตรงในการพยากรณ์ค่าของคลาสบวกได้ดี

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

### 2.5.3 ค่าการเรียกคืน

ค่าการเรียกคืน (Recall) เป็นวิธีการบ่งบอกความสามารถในการพยากรณ์ได้ถูกต้องของคลาสบวก หากค่า Recall มีค่าเข้าใกล้ 1 แสดงว่าโมเดลโครงข่ายประสาทนั้นมีความถี่ของการพยากรณ์ถูกต้องสูง นั่นคือหากถูกระบุว่าเป็นพบขโมย ความเป็นจริงต้องเจอขโมย

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

### 2.5.4 ค่าคะแนน F1 และ F เบต้า

ค่าคะแนน F1 (F1-score or F-measure) เป็นวิธีการบ่งบอกความสามารถของโมเดลโครงข่ายประสาทด้วยค่าเฉลี่ยแบบประสานกัน (Harmonic mean) ระหว่างค่าความเที่ยงตรง (Precision) กับค่าการเรียกคืน (Recall) หากค่า  $F_1$  มีค่าสูง แสดงว่าโมเดลโครงข่ายประสาทมีความสามารถในการพยากรณ์ได้อย่างถูกต้องและเที่ยงตรง

$$F_1 = \frac{2}{\text{precision}^{-1} + \text{recall}^{-1}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (4)$$

หากต้องการดูความสามารถในการพยากรณ์ที่สนใจแนวโน้มไปทางค่าการเรียกคืน หรือแนวโน้มไปทางค่าความเที่ยงตรง สามารถกำหนดค่าเบต้า  $\beta$  เช่น หากต้องการแนวโน้มไปทางค่าความเที่ยงตรงให้  $\beta < 1$  หากต้องการแนวโน้มไปทางค่าการเรียกคืนให้  $\beta > 1$  แต่ถ้าค่าเบต้า  $\beta = 0$  จะให้ค่าความเที่ยงตรงเพียงอย่างเดียว

$$F_\beta = \frac{(1 + \beta^2) \cdot (\text{precision} \cdot \text{recall})}{(\beta^2 \cdot \text{precision} + \text{recall})} \quad (5)$$

#### 2.5.5 การบ่งบอกความสามารถของโมเดลโครงข่ายประสาท

เพื่อให้เข้าใจถึงการบ่งบอกความสามารถของโมเดลโครงข่ายประสาท กำหนดข้อมูลตัวอย่างเป็นจำนวน 100 ข้อมูล ค่าเริ่มต้นแบ่งออกเป็นกลุ่มที่พยากรณ์ถูกต้องครึ่งหนึ่ง และพยากรณ์ผิดครึ่งหนึ่ง และแบ่งระดับชั้น (Threshold) จาก 0.0 ถึง 1.0 เป็นทั้งหมด 10 ชั้น โดยค่าแต่ละระดับชั้นจะมีแบ่งตามผลการพยากรณ์จัดกลุ่ม ตามตารางที่ 1

ตารางที่ 1 การแบ่งข้อมูลตามระดับชั้น (Threshold)

Threshold	True positive	False positive	True negative	False negative
0.0	50	50	0	0
0.1	49	48	2	1
0.2	46	43	8	3
0.3	62	36	18	8
0.4	57	24	23	10
0.5	42	16	29	13
0.6	34	11	32	23
0.7	21	8	39	32
0.8	12	3	42	43
0.9	7	2	47	44
1.0	0	0	50	50



จากนั้นคำนวณค่าความแม่นยำ (Accuracy) ด้วยสมการที่ 1 ค่าความเที่ยงตรง (Precision) ด้วยสมการที่ 2 ค่าการเรียกคืน (Recall) ด้วยสมการที่ 3 และค่าคะแนน F1 ด้วยสมการที่ 4 ทำให้ได้ค่าออกมาในตารางที่ 2 ซึ่งจะเห็นได้ว่าการแบ่งระดับขั้นที่ 0.5 โมเดลโครงข่ายประสาทจะมีความสามารถในการพยากรณ์ดีที่สุด จากหลักการเลือกค่าที่เข้าใกล้ 1 มากที่สุด มีค่าความแม่นยำสูงสุด ค่าความเที่ยงตรงเป็นอันดับสอง ค่าการเรียกคืนอยู่ที่ 0.76 และค่าคะแนน F1 เป็นอันดับสอง

ตารางที่ 2 การคำนวณค่าความแม่นยำ ความเที่ยงตรง การเรียกคืน และคะแนน F1

Threshold	Accuracy	Precision	Recall	F <sub>1</sub>
0.0	0.50	0.50	1.00	0.67
0.1	0.51	0.51	0.98	0.67
0.2	0.54	0.52	0.94	0.67
0.3	0.65	0.63	0.89	0.74
0.4	0.70	0.70	0.85	0.77
0.5	0.71	0.72	0.76	0.74
0.6	0.66	0.76	0.60	0.67
0.7	0.60	0.72	0.40	0.51
0.8	0.54	0.80	0.22	0.34
0.9	0.54	0.78	0.14	0.23
1.0	0.50	-	0.00	-

#### 2.5.6 การบ่งบอกความสามารถด้วยค่า Average Precision (AP)

วิธีการบ่งบอกความสามารถของโมเดลโครงข่ายประสาทที่มีการใช้งานอย่างแพร่หลาย เช่น การตรวจจับวัตถุ (Object detection) ของ PASCAL VOC2015 [14] จะเป็นการบอกด้วยค่า Average Precision (AP) ด้วยการหาพื้นที่ใต้โค้ง (Under curve area) ระหว่าง ค่าความเที่ยงตรง (Precision) และ ค่าการเรียกคืน (Recall) มีค่าอยู่ระหว่าง 0 ถึง 1 หากค่า AP มีค่าเข้าใกล้ 1 แสดงว่าโมเดลโครงข่ายประสาทนั้นสามารถในการพยากรณ์ได้อย่างถูกต้องแม่นยำ

$$AP = \int_0^1 p(r)dr = \sum_{i=1}^n P(i)\Delta Re(i) \quad (6)$$

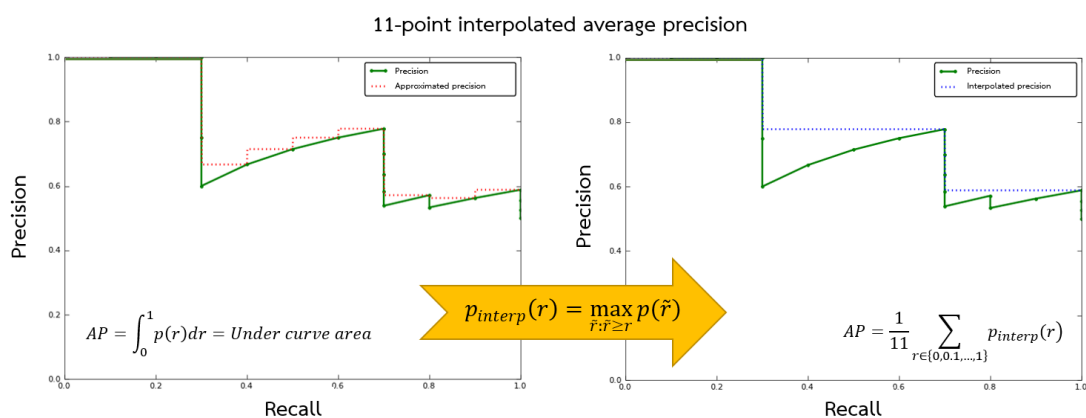
$$AP = \sum_{i=1}^n Precision_i(Recall_i - Recall_{i-1}) \quad (7)$$

เมื่อ  $AP$  คือพื้นที่ใต้โค้งของค่าความเที่ยงกับค่าการเรียกคืน  $p(r)dr$  คือฟังก์ชันความสัมพันธ์ระหว่างค่าความเที่ยงกับค่าการเรียกคืน  $n$  คือจำนวนจุดตัดข้อมูลในช่วงกราฟ  $P(i)$  คือช่วงค่าความเที่ยงตรงในพื้นที่ใต้โค้ง และ  $\Delta Re(i)$  คือการเปลี่ยนแปลงของค่าการเรียกคืน

ตารางที่ 3 ความสัมพันธ์ค่าความเที่ยงตรง (Precision) และค่าการเรียกคืน (Recall)

จุดตัดข้อมูล	ค่าความเที่ยงตรง	ค่าการเรียกคืน	ส่วนต่างค่าการเรียกคืน
จุดที่ 1	1.00	0.2	0.2
จุดที่ 2	1.00	0.4	0.2
จุดที่ 3	0.66	0.4	0.0
จุดที่ 4	0.75	0.6	0.2
จุดที่ 5	0.60	0.6	0.0
จุดที่ 6	0.66	0.8	0.2
จุดที่ 7	0.57	0.8	0.0
จุดที่ 8	0.50	0.8	0.0
จุดที่ 9	0.44	0.8	0.0
จุดที่ 10	0.50	1.0	0.2

จากตารางที่ 3 สามารถนำค่าความเที่ยงตรงและส่วนต่างค่าการเรียกคืน คำนวณค่า Average Precision ด้วยสมการที่ 6 จะได้ว่า  $AP = (1 \times 0.2) + (1 \times 0.2) + (0.66 \times 0.0) + (0.75 \times 0.2) + (0.6 \times 0.0) + (0.66 \times 0.2) + (0.57 \times 0) + (0.5 \times 0) + (0.44 \times 0) + (0.5 \times 0.2) = 0.782$  หรือ Average Precision (AP) มีค่าเป็นร้อยละ 78.20



ภาพที่ 15 การบ่งบอกความสามารถของโมเดลโครงข่ายประสาทด้วยวิธี Interpolated

ในการบ่งบอกความสามารถของโมเดลโครงข่ายประสาทนั้น จากภาพที่ 15 ทางซ้ายจะเห็นว่าค่าความเที่ยงตรง ณ ตำแหน่งของค่าการเรียกคืนใดใด จะมีค่าความเที่ยงตรงที่แตกต่างกันไป หากต้องการคำนวณหาพื้นที่ใต้โค้งจะต้องทำที่ระหว่างการเปลี่ยนแปลงค่าการเรียกคืน จึงนำเสนอวิธีการประมาณค่า Average Precision แบบเฉลี่ย โดยการปรับค่าความเที่ยงตรงให้เท่ากับค่าความเที่ยงตรง ณ ตำแหน่งก่อนหน้าของค่าการเรียกคืน จะทำให้ได้ภาพที่ 15 ทางขวาที่คำนวณพื้นที่ใต้โค้งได้ง่ายกว่า เรียกว่า Interpolated average precision

$$AP = \frac{1}{11} \sum_{r \in \{0, 0.1, \dots, 1\}} p_{interp}(r) \quad (8)$$

$$p_{interp}(r) = \max_{\tilde{r}: \tilde{r} \geq r} p(\tilde{r}) \quad (9)$$

เมื่อ  $AP$  คือผลรวมของพื้นที่ใต้โค้ง ณ ตำแหน่งของค่าการเรียกคืนตั้งแต่ 0.0 ถึง 1.0 และ  $p_{interp}(r)$  คือ การปรับค่าความเที่ยงตรงให้เท่ากับค่าความเที่ยงตรง ณ ตำแหน่งก่อนหน้าของค่าการเรียกคืน จากที่กล่าวมาการคำนวณนี้ เป็นการหาค่าของข้อมูลเพียงคลาสเดียวเท่านั้น

#### 2.5.7 การบ่งบอกความสามารถด้วยค่า Mean Average Precision (mAP)

เนื่องจากระบบที่ใช้โครงข่ายประสาทในการจำแนกข้อมูลออกเป็นกลุ่มนั้น จะนิยมแยกออกเป็นมากกว่า 1 คลาส เช่น เม็ดเลือดแดง (Red blood cell : RBC) เม็ดเลือดขาว (White blood cell : WBC) และเกล็ดเลือด (Platelets) ในการบ่งบอกว่าโมเดลโครงข่ายประสาทที่ใช้มีความสามารถในการจำแนกข้อมูลแต่ละคลาสได้แม่นยำแค่ไหน จึงจำเป็นต้องนำค่า Average Precision (AP) ของแต่ละคลาสมาหาค่าเฉลี่ย หากค่า mAP มีค่าเข้าใกล้ 1 แสดงว่าโมเดลโครงข่ายประสาทนั้นมีความสามารถในการพยากรณ์ได้อย่างถูกต้องแม่นยำในทุกคลาส

$$mAP = \frac{1}{n_{classes}} \sum AP_{classes} \quad (10)$$

ในการฝึกโมเดลของโครงข่ายประสาทนั้น นอกจากจะสนใจค่าความแม่นยำแล้ว (Accuracy) อีกสิ่งหนึ่งที่ต้องให้ความสำคัญ คือ ค่าสูญเสีย (Loss) ซึ่งโดยปกติแล้วหากค่าความแม่นยำของโมเดลที่ดีควรเข้าใกล้ร้อยละ 100 และมีค่าสูญเสียต่ำที่สุด (Loss minimize) หนึ่งในวิธีการสามารถใช้ในการเรียนรู้ที่ถดถอย (Regression) มาช่วยวัดผลการพยากรณ์อย่างต่อเนื่อง เพื่อดูค่าความคลาดเคลื่อนของการพยากรณ์ (Error) เช่น การสร้างฟังก์ชันการสูญเสีย (Loss function) โดยใช้การถดถอยเชิงเส้น

(Linear regression) เพื่อค่าความคลาดเคลื่อนระหว่างค่าจริงกับค่าพยากรณ์ เป็นต้น

#### 2.5.8 ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย

ค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error : MAE) เป็นวิธีการบ่งบอกค่าความคลาดเคลื่อนของข้อมูลโดยภาพรวม หากค่า MAE มีค่าต่ำแสดงว่าโมเดลโครงข่ายประสาทนั้นมีความแม่นยำสูง

$$MAE = \frac{1}{n} \sum_{i=1}^n |p_i - a_i| \quad (11)$$

เมื่อ  $MAE$  คือ ค่าความคลาดเคลื่อนสัมบูรณ์  $n$  คือ จำนวนข้อมูล  $p_i$  คือค่าพยากรณ์ และ  $a_i$  คือค่าจริง

#### 2.5.9 ค่าร้อยละความคลาดเคลื่อนสัมบูรณ์เฉลี่ย

ค่าร้อยละความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Percentage Error : MAPE) เป็นวิธีการบ่งบอกค่าความคลาดเคลื่อนให้อยู่ในรูปของร้อยละ หากค่า MAPE มีค่าร้อยละต่ำแสดงว่าโมเดลโครงข่ายประสาทนั้นมีความแม่นยำสูง

$$MAPE = \left( \frac{1}{n} \sum_{i=1}^n \frac{|a_i - p_i|}{|a_i|} \right) \times 100 \quad (12)$$

เมื่อ  $MAPE$  คือ ค่าร้อยละความคลาดเคลื่อนสัมบูรณ์  $n$  คือ จำนวนข้อมูล  $p_i$  คือค่าพยากรณ์ และ  $a_i$  คือค่าจริง

#### 2.5.10 ค่าความคลาดเคลื่อนกำลังสองเฉลี่ย

ค่าความคลาดเคลื่อนกำลังสองเฉลี่ย (Mean Squared Error : MSE) เป็นวิธีการบ่งบอกค่าความคลาดเคลื่อน จากการนำค่าความคลาดเคลื่อนที่คิดจากผลต่างค่าพยากรณ์และค่าจริงกำลังสอง หากค่า MSE มีค่าต่ำแสดงว่าโมเดลโครงข่ายประสาทนั้นมีความแม่นยำสูง

$$MSE = \frac{1}{n} \sum_{i=1}^n (p_i - a_i)^2 \quad (13)$$

เมื่อ  $MSE$  คือ ค่าความคลาดเคลื่อนกำลังสองเฉลี่ย  $n$  คือ จำนวนข้อมูล  $p_i$  คือค่าพยากรณ์ และ  $a_i$  คือค่าจริง

#### 2.5.11 ค่ารากที่สองของค่าความคลาดเคลื่อนเฉลี่ย

ค่ารากที่สองของค่าความคลาดเคลื่อนเฉลี่ย (Root Mean Squared Error : RMSE) เป็นวิธีการบ่งบอกค่าความคลาดเคลื่อนที่เป็นวิธีการมาตรฐาน สำหรับการบ่งบอกค่าทางสถิติ หากค่า RMSE มีค่าต่ำแสดงว่าโมเดลโครงข่ายประสาทนั้นมีความแม่นยำสูง

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_i - a_i)^2} \quad (14)$$

เมื่อ  $RMSE$  คือ ค่ารากที่สองของค่าความคลาดเคลื่อนเฉลี่ย  $MSE$  คือ ค่าความคลาดเคลื่อนกำลังสองเฉลี่ย  $n$  คือ จำนวนข้อมูล  $p_i$  คือค่าพยากรณ์ และ  $a_i$  คือค่าจริง

#### 2.5.12 ค่าสัมประสิทธิ์สหสัมพันธ์ และค่าสัมประสิทธิ์แสดงการตัดสินใจ

ค่าสัมประสิทธิ์สหสัมพันธ์ (Correlation coefficient :  $R$ ) เป็นวิธีการบ่งบอกความสัมพันธ์ระหว่างตัวแปร 2 ตัว คือ ค่าพยากรณ์และค่าจริง หากค่า  $R$  มีค่าเข้าใกล้ 1 แสดงว่าโมเดลโครงข่ายประสาทนั้นมีความแม่นยำสูง แต่หากเข้าใกล้ -1 แสดงว่ามีความแม่นยำต่ำ นอกจากนี้ยังสามารถหาค่าสัมประสิทธิ์แสดงการตัดสินใจ (Coefficient of determination :  $R^2$ ) เป็นวิธีการบ่งบอกค่าความแม่นยำจากสัดส่วนของค่าความแปรปรวนของข้อมูลแจกแจง (Explained variance) และค่าความแปรปรวนของข้อมูลรวม (Total variance) หากค่า  $R^2$  มีค่าเข้าใกล้ 1 แสดงว่าโมเดลโครงข่ายประสาทมีความแม่นยำสูง และบ่งบอกว่าค่าความคลาดเคลื่อนต่ำ

$$R = \frac{\sum_{i=1}^n (p_i \cdot a_i)}{\sqrt{\sum_{i=1}^n a_i^2} \cdot \sqrt{\sum_{i=1}^n p_i^2}} \quad (15)$$

$$R^2 = \frac{(\sum_{i=1}^n (p_i \cdot a_i))^2}{(\sum_{i=1}^n a_i^2) \cdot (\sum_{i=1}^n p_i^2)} \quad (16)$$

เมื่อ  $R$  คือ ค่าสัมประสิทธิ์สหสัมพันธ์  $R^2$  คือ ค่าสัมประสิทธิ์แสดงการตัดสินใจ  $n$  คือ จำนวนข้อมูล  $p_i$  คือค่าพยากรณ์ และ  $a_i$  คือค่าจริง

## 2.6 หลักการเกิดภาพของกล้องจุลทรรศน์

กล้องจุลทรรศน์จัดเป็นอุปกรณ์ที่ทำหน้าที่ขยายรายละเอียดของวัตถุขนาดเล็กให้มีความชัดเจนมากยิ่งขึ้น สามารถแบ่งออกเป็น 2 กลุ่ม ได้แก่ กล้องจุลทรรศน์เชิงซ้อน (Compound microscope) ประกอบด้วย กล้องจุลทรรศน์ชนิดไบรท์ฟิลด์ (Bright field microscope) กล้องจุลทรรศน์ชนิดดาร์คฟิลด์ (Dark field microscope) กล้องจุลทรรศน์ฟลูออเรสเซนซ์ (Fluorescence microscope) กล้องจุลทรรศน์วิภูภาค (Phase contrast microscope) เป็นต้น และกล้องจุลทรรศน์อิเล็กตรอน (Electron microscope) ประกอบด้วย กล้องจุลทรรศน์อิเล็กตรอนแบบส่องผ่าน (Transmission electron microscope : TEM) และกล้องจุลทรรศน์อิเล็กตรอนแบบส่องกราด (Scanning electron microscope : SEM) [15] ซึ่งจะกล่าวถึงกล้องจุลทรรศน์เชิงซ้อนชนิดไบรท์ฟิลด์ที่นำมาใช้ในการทำวิจัยเป็นหลัก

กล้องจุลทรรศน์ชนิดไบรท์ฟิลด์เป็นกล้องแบบพื้นฐานที่สุด กล้องจุลทรรศน์ที่เป็นรากฐานถูกคิดค้นโดย van Leeuwenhoke ปี ค.ศ.1958 จากนั้นมีการพัฒนาใช้เลนส์ขยายภาพเพียงชั้นเดียว เกิดเป็นภาพเสมือนของวัตถุระหว่างระนาบวัตถุจนถึงผิวหน้าเลนส์ และส่งไปยังจอตาเป็นภาพเสมือนหัวกลับ (Virtual inverted image) [16] จากนั้นจึงมีการพัฒนากล้องจุลทรรศน์เชิงซ้อนที่ใช้ระบบสองเลนส์ขยาย ประกอบด้วยเลนส์ใกล้ตา (Eyepiece lens) และเลนส์ใกล้วัตถุ (Objective lens) เลนส์ใกล้วัตถุจะทำหน้าที่ขยายวัตถุที่ปรากฏอยู่หน้าเลนส์ เช่น แผ่นสไลด์ (Slide glass) แผ่นสไตน์นับเม็ดเลือด (Hemocytometer) เป็นต้น เกิดเป็นภาพจริงหัวกลับ (Real inverted image) ที่ถูกขยายแล้ว จากนั้นภาพที่ได้จะถูกขยายด้วยเลนส์ใกล้ตา ก่อนส่งภาพเข้าสู่กระจกตา (Cornea) และเลนส์ตาจะรวมภาพที่ได้จากเลนส์ใกล้ตาทั้ง 2 ส่วนกลับมาเป็นภาพเสมือนหัวกลับที่จอตา (Retina) ซึ่งระยะระหว่างเลนส์ใกล้ตากับเลนส์ใกล้วัตถุอยู่ในช่วง 160 ถึง 170 มิลลิเมตร (Gage, 1891) นอกจากนี้ยังมีระบบเลนส์ที่กำเนิดภาพเป็นระยะอนันต์ (Infinity) เมื่อวัตถุถูกจัดตำแหน่งไว้หน้าเลนส์ใกล้วัตถุ

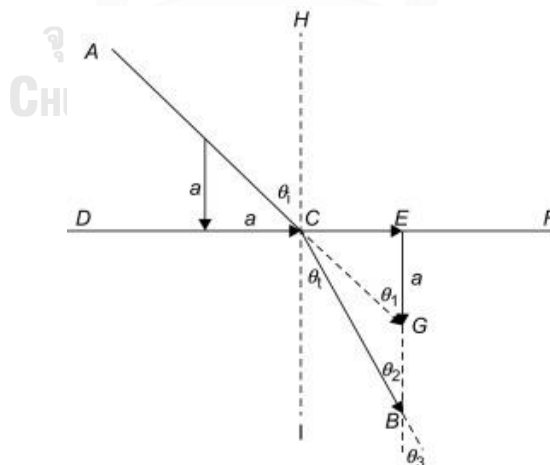
เนื่องจากชุดเลนส์ที่ใช้ภายในกล้องจุลทรรศน์เป็นเลนส์ระนาบโค้ง (Curved surface) ที่ทำจากวัสดุแก้วหรือพอลิเมอร์เฉพาะทาง ฉะนั้นการเดินทางของแสงจะไม่เป็นเส้นตรงอันเนื่องมาจากการหักเหของแสงผ่านตัวกลางที่มีดัชนีหักเห (Index of refraction) แตกต่างกัน อธิบายได้ด้วยความสัมพันธ์ทางคณิตศาสตร์ระหว่างมุมตกกระทบ (Angle of incidence)  $\theta_i$  กับมุมหักเห (Angle of transmission)  $\theta_t$  ถูกค้นพบโดย Willebrord Snel van Royen ปี ค.ศ. 1621 จึงเรียกว่ากฎของสเนล (The Snel-Descartes law) นิยามว่า เมื่อแสงเดินทางผ่านอากาศเข้าสู่ตัวกลางที่มีความหนาแน่น อัตราส่วนของมุมไซน์ของมุมตกกระทบ  $\sin \theta_i$  กับมุมไซน์ของมุมหักเห  $\sin \theta_t$  มีค่าคงที่  $n$  เป็นค่าคงที่ของการหักเหในตัวกลางนั้น ๆ

$$\sin \theta_i / \sin \theta_t = n \quad (17)$$

ค่าดัชนีหักเห (Index of refraction) เป็นคุณลักษณะของตัวกลางที่กำหนดจากความหนาแน่นของวัตถุที่แตกต่างกัน ช่วงปี ค.ศ. 1979 Isaac Newton ได้ประยุกต์ใช้กฎการเคลื่อนที่ (Law of motion) เมื่อแสงตกกระทบบนผิวหน้าสองส่วน DF ระหว่างตัวกลางที่มีความหนาแน่นต่างกัน อนุภาคของแสงจะถูกเร่งโดยตัวกลางที่มีความหนาแน่นสูงกว่า เมื่ออนุภาคของแสงมีองค์ประกอบความเร็วตั้งฉากกับผิวหน้าสองส่วนเพิ่มขึ้น สมมติว่าอนุภาคแสงเดินทางจาก A ไป B กระแทบเข้ากับผิวหน้าสองส่วนของอากาศกับแก้ว (Air-glass interface) ที่ตำแหน่ง C เนื่องจากอนุภาคแสงถูกดึงดูดจากตัวกลางที่เป็นแก้ว ส่งผลให้องค์ประกอบของความเร็วแสงตั้งฉาก EB ไปยังผิวหน้าสองส่วนถูกเร่ง ในขณะที่องค์ประกอบของแสงเชิงขนาน CE ไปยังผิวหน้าสองส่วนไม่ถูกเปลี่ยนแปลง อนุภาคของแสงจึงเดินทางจาก C ไป B แทนที่จะเดินทางจาก C ไป G เพราะระยะจาก CG และ CB นั้นแปรผันตามความเร็วของอนุภาคแสงที่ตกกระทบบนและหักเหผ่านตัวกลางต่างชนิดกัน จึงสรุปได้ว่า ความเร็วของแสงที่อยู่ในแก้ว  $v_i$  มีค่ามากกว่าความเร็วของแสงที่อยู่ในอากาศ  $v_a$

$$\sin \theta_1 = CE/CG \quad (18)$$

$$\sin \theta_2 = CE/CB \quad (19)$$



ภาพที่ 16 ภาพอธิบายการหักเหของแสงด้วยทฤษฎีอนุภาคแสงของนิวตัน เส้น DF คือแกนระนาบ (Principle axis) และ HI คือแกนตั้งฉากเลนส์ (Lens axis) [17]

จะพบว่าของมุมที่กระทำต่อเส้นตั้งฉาก  $\theta_2 = \theta_3$  มีค่าเท่ากันเป็นมุมตรงข้าม และทำให้เส้นทางเดินแสงขนาดกันทั้งสองเส้น CB เมื่อเปรียบเทียบกับสมการกฎของสเนลกำหนดให้  $\theta_i = \theta_1$  และ  $\theta_t = \theta_3$  จึงสามารถพิสูจน์ว่าค่าคงที่  $n$  นั้น จะขึ้นอยู่กับความหนาแน่นผิวหน้าสองส่วนของตัวกลางนั้น ๆ ซึ่งเป็นผลมาจากการเร่งหรือความเร็วของอนุภาคแสงที่เดินทางผ่านตัวกลาง

$$\sin \theta_i / \sin \theta_t = \sin \theta_1 / \sin \theta_3 \quad (20)$$

$$\frac{\sin \theta_i}{\sin \theta_t} = \frac{CE}{CG} \cdot \frac{CB}{CE} = \frac{CB}{CG} = n \quad (21)$$

จากคุณสมบัติของแสงเป็นทั้งอนุภาคและคลื่น จึงสามารถติดค่าดัชนีหักเห  $n$  จากอัตราส่วนของความเร็วแสงในสุญญากาศ (Velocity of light in a vacuum)  $c$  มีค่าเป็น  $2.99792458 \times 10^8$  เมตรต่อวินาที (m/s) กับความเร็วของแสงในตัวกลางที่สนใจ (Velocity of light in the medium)

$$n_i = c/v_i \quad (22)$$

เมื่อ  $n_i$  คือค่าดัชนีหักเหของตัวกลางชนิด  $i$  และ  $v_i$  คือค่าความเร็วของแสงในตัวกลางชนิด  $i$  ในหน่วยเมตรต่อวินาที (m/s) ซึ่งจะแสดงค่าดัชนีหักเหของตัวกลางชนิดต่าง ๆ และความสัมพันธ์ของความหนาแน่นของตัวกลางในหน่วยกิโลกรัมต่อลูกบาศก์เมตร ( $\text{kg}/\text{m}^3$ ) จากตารางที่ 4 จะพบว่าค่าดัชนีหักเหของแสงในสุญญากาศมีค่าเป็น 1.00000 และในอากาศมีค่าเป็น 1.00027 ในขณะที่ตัวกลางที่มีความหนาแน่นสูงขึ้นจะมีค่าดัชนีหักเหที่สูงตาม จึงสามารถกำหนดค่าดัชนีหักเหมาตรฐาน  $n_i=1$  เพื่อใช้ในการคำนวณหามุมตกกระทบ และมุมวิฤติได้ด้วยสมการ

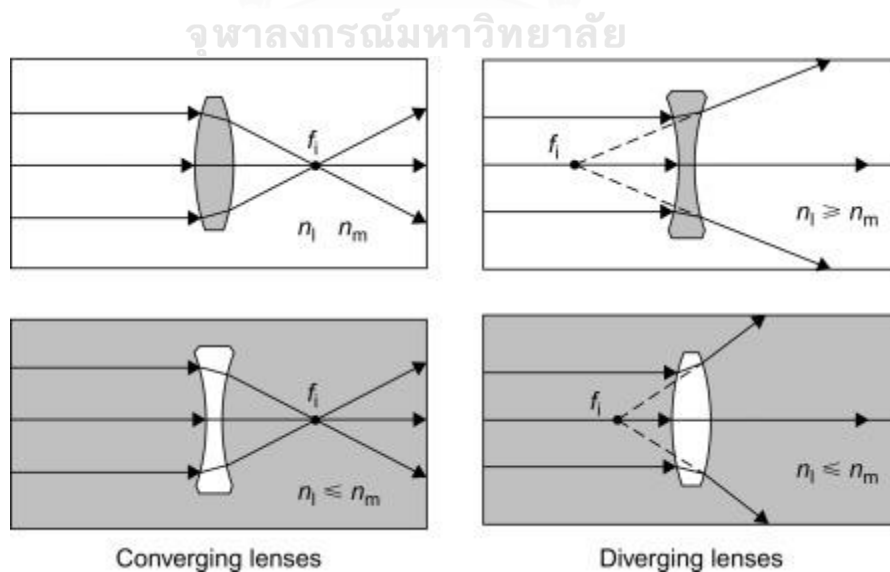
$$n_i \sin \theta_i = n_t \sin \theta_t \quad (23)$$



ตารางที่ 4 ค่าดัชนีหักเหของตัวกลางชนิดต่าง ๆ และค่าความหนาแน่นโดยประมาณ

ตัวกลาง	ดัชนีหักเห ( $n$ )	ความหนาแน่น ( $\text{kg/m}^3$ )
สุญญากาศ (Vacuum)	1.00000	0
อากาศ (Air)	1.00027	1.25
น้ำ (Water)	1.33300	1000
แก้ว (Glass)	1.51500	2600
เพชร (Diamond)	2.42000	3500

จากหลักการหักเหของแสงในตัวกลางชนิดต่าง ๆ จึงก่อให้เกิดการพัฒนาเลนส์จากวัสดุจำพวกแก้วมาสร้างเป็นปริซึมรูปร่างถั่ว (Lentil-shaped) เมื่อแสงเดินทางผ่านอากาศมาตกกระทบที่ผิวหน้าของเลนส์ แสงจะเกิดการหักเห กรณีที่แสงตกกระทบบดิ่งฉากกับส่วนที่หนาสุดของเลนส์หรือบริเวณตรงกลาง แสงจะพุ่งไปเป็นเส้นตรงไม่เกิดการหักเหขึ้น แต่กรณีที่แสงตกทำมุมตกกระทบบิดขึ้น หากใช้กฎของสเนลพิสูจน์ จะทำให้ทราบถึงมุมที่แสงนั้นหักเหไป ซึ่งจะเกิดเส้นทางเดินแสงในมุมที่ต่างกันหลาย ๆ เส้นทาง เมื่อแสงทั้งหมดมาบรรจบกันเกิดจุดตัดหลังเลนส์ขึ้น จุดนั้นจะเรียกว่า จุดโฟกัส  $f_i$  และระยะจากจุดโฟกัสจนถึงผิวหน้าของเลนส์จะเรียกว่า ระยะโฟกัส ซึ่ง Lentil-shaped นี้เป็นที่มาของเลนส์นูน (Convex lens) หรือเลนส์รวมแสง (Converging lens) ทำหน้าที่รวมแสง และเลนส์ที่ทำหน้าที่กระเจิงแสงเกิดจุดโฟกัสเสมือนบริเวณหน้าเลนส์ คือ เลนส์เว้า (Concave lens) หรือเลนส์กระเจิงแสง (Diverging lens) ในภาพที่ 17

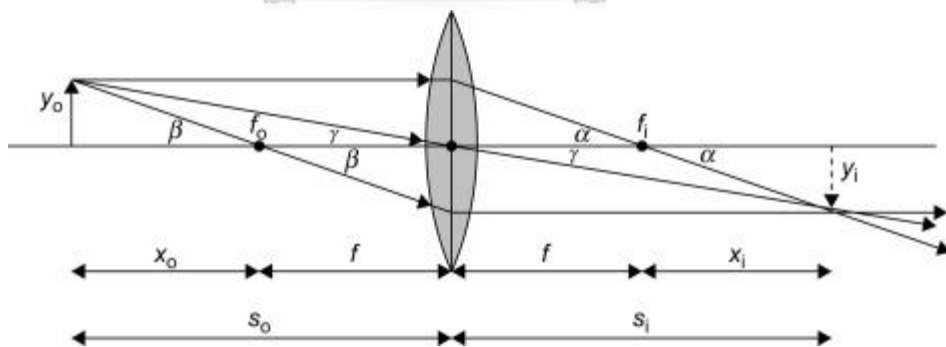


ภาพที่ 17 ภาพแสดงคุณลักษณะของเลนส์รวมแสงและเลนส์กระเจิงแสง [17]

จากภาพที่ 17 กรณีพื้นหลังสีขาวคือตัวกลางที่มีค่าดัชนีหักเหต่ำกว่าเลนส์ พื้นหลังสีเทา คือตัวกลางที่มีค่าดัชนีหักเหมากกว่าเลนส์ วิธีการหาความโค้งและการหักเหแสงของเลนส์สามารถหาได้จากความยาวโฟกัส (Focal length)  $f$  ที่มีความสัมพันธ์กับรัศมีความโค้งของเลนส์ เมื่อ  $R_1$  คือรัศมีความโค้งผิวเลนส์ด้านแรก  $R_2$  คือรัศมีความโค้งผิวเลนส์ด้านที่สอง รวมถึงดัชนีหักเหของเลนส์  $n_l$  และดัชนีหักเหของตัวกลาง เช่น อากาศ หรือน้ำมัน  $n_m$  ได้จากสมการ

$$\frac{1}{f} = \left( \left( \frac{n_l}{n_m} \right) - 1 \right) \left( \frac{1}{R_1} - \frac{1}{R_2} \right) \quad (24)$$

องค์ประกอบของความยาวโฟกัสของเลนส์มี 2 ส่วน คือ ความยาวโฟกัสวัตถุ  $f_o$  และความยาวโฟกัสภาพ  $f_i$  หากวางวัตถุไว้ที่ตำแหน่งเท่ากับความยาวโฟกัสหน้าเลนส์รวมแสง จะปรากฏภาพเสมือนหลังเลนส์นูนเป็นระยะอนันต์ ในทำนองเดียวกัน หากวางจุดกำเนิดแสงไว้ที่เดียวกันบนแกนระนาบเลนส์ (Principle axis) ที่  $f_o$  แสงจะเดินทางเป็นเส้นตรงขนานกับแกนไปยังอีกด้านหนึ่งของเลนส์ อีกกรณีหนึ่งหากวางวัตถุไว้ที่ระยะอนันต์ด้านหน้าเลนส์นูนภาพเสมือนที่ตำแหน่งเดียวกับระยะโฟกัสภาพ  $f_i$  เพื่อให้การอธิบายเข้าใจง่ายขึ้น จากภาพที่ 18 เป็นการอธิบายหลักการเกิดภาพโดยใช้เส้นรังสี 3 เส้นลากผ่านเลนส์ในตำแหน่งต่าง ๆ



ภาพที่ 18 แผนภาพอธิบายคุณสมบัติของเลนส์รวมแสง (Converging lens) [17]

ตารางที่ 5 อธิบายคุณสมบัติของการเกิดภาพ ตำแหน่งการเกิดภาพ ทิศทางของภาพ และขนาดของภาพที่เกิดขึ้นจากเลนส์รวมแสง และเลนส์กระจายแสง โดยตำแหน่งการเกิดภาพที่เกิดจากเลนส์รวมแสง (Converging lens) จะเหมือนกับกระจกเว้า (Concave mirror) และเลนส์นูน (Convex lens) ในทำนองเดียวกันภาพที่เกิดจากเลนส์กระจายแสง (Diverging lens) จะเหมือนกับกระจกนูน (Convex mirror) และเลนส์เว้า (Concave lens)

ตารางที่ 5 คุณสมบัติการเกิดภาพของเลนส์รวมแสง และเลนส์กระจายแสง

ตำแหน่งวัตถุ	ชนิด	ตำแหน่งภาพ	ทิศทาง	ขนาด
<b>ภาพที่เกิดจากเลนส์รวมแสง (Converging lens)</b>				
$\infty > s_o > 2f$	ภาพจริง	$f < s_i < 2f$	กลับ	เล็กกว่าวัตถุ
$s_o = 2f$	ภาพจริง	$s_i = 2f$	กลับ	เท่าวัตถุ
$f < s_o < 2f$	ภาพจริง	$\infty > s_i > 2f$	กลับ	ถูกขยาย
$s_o = f$		$\infty$		
$s_o < f$	ภาพเสมือน	$ s_i  > s_o$	ตั้งตรง	ถูกขยาย
<b>ภาพที่เกิดจากเลนส์กระจายแสง (Diverging lens)</b>				
ตำแหน่งใดใด	ภาพเสมือน	$ s_i  <  f $	ตั้งตรง	เล็กกว่าวัตถุ

จากคุณสมบัติของเลนส์ในตารางที่ 5 สมการคำนวณเลนส์บาง (Thin lens) หรือสมการเลนส์ของเกาส์ (Gaussian lens) และสมการกำลังการขยายของเลนส์ (Magnification lens) เพื่ออธิบายคุณสมบัติการเกิดภาพของเลนส์ ซึ่งตัวแปรสามารถอ้างอิงกับภาพที่ 18 การเกิดภาพที่มาจากเลนส์รวมแสงตามความยาวโฟกัส  $f$  วัตถุหน้าเลนส์มีขนาด  $y_o$  จะมีระยะห่างจากเลนส์เป็น  $s_o$  และภาพที่เกิดขึ้นหลังเลนส์มีขนาด  $y_i$  จะมีระยะห่างจากเลนส์เป็น  $s_i$

$$\frac{1}{f} = \frac{1}{s_i} + \frac{1}{s_o} \quad (25)$$

$$m_T = \frac{y_i}{y_o} = -\frac{s_i}{s_o} \quad (26)$$

$$s_o = f_o + x_o \quad (27)$$

$$s_i = f_i + x_i \quad (28)$$

เมื่อค่า  $S_i$  เป็นบวกภาพที่เกิดขึ้นหลังเลนส์จะเป็นภาพจริงหัวกลับ (Real inverted image) และถ้าค่า  $S_i$  เป็นลบภาพที่เกิดขึ้นหลังเลนส์จะเป็นภาพเสมือนหัวตั้ง (Virtual erect image) ในกรณีที่สนใจกำลังการขยายของเลนส์ จะคิดได้จากค่าสัมบูรณ์ของ  $m_T$  ซึ่งค่า  $m_T$  เป็นบวกจะเกิดภาพหัวตั้ง และถ้าเป็นลบจะเกิดภาพหัวกลับ ในการพิสูจน์สมการเลนส์ของเก๊าส์ ทำการจัดรูปสมการที่ 27 และ 28 ใหม่ จะได้ว่า

$$x_o = s_o - f_o \quad (29)$$

$$x_i = s_i - f_i \quad (30)$$

จากคุณสมบัติของเลนส์นูนสองด้าน (Biconvex) หรือ เลนส์เว้าสองด้าน (Biconcave) ทำให้ระยะโฟกัสหน้าเลนส์และหลังเลนส์เท่ากัน จึงสามารถแทนสมการอัตราส่วนในสมการที่ 31 และแทนค่าสมการที่ 29 และ 30 ทำการจัดรูปสมการจะได้ว่า

$$\frac{x_i}{f_i} = \frac{x_o}{f_o} \quad (31)$$

$$\frac{(s_i - f_i)}{f_i} = \frac{f_o}{(s_o - f_o)} \quad (32)$$

$$\frac{(s_i - f)}{f} = \frac{f}{(s_o - f)} \quad (33)$$

$$f^2 = (s_o - f)(s_i - f) \quad (34)$$

$$f^2 - f^2 = s_o s_i - s_o f - s_i f = 0 \quad (35)$$

$$\therefore s_o s_i = s_o f + s_i f = f(s_o + s_i) \quad (36)$$

$$(s_o s_i) \left(\frac{1}{f}\right) \left(\frac{1}{s_o s_i}\right) = f(s_o + s_i) \left(\frac{1}{f}\right) \left(\frac{1}{s_o s_i}\right) \quad (37)$$

$$\frac{(s_o s_i)}{f(s_o s_i)} = \left(\frac{1}{f} \cdot \frac{f}{1}\right) \left(\frac{1}{s_o s_i} \cdot \frac{s_o s_i}{1}\right) \quad (38)$$

$$\therefore \frac{1}{f} = \frac{1}{s_i} + \frac{1}{s_o} \quad (39)$$

จากภาพที่ 18 จะเห็นภาพสามเหลี่ยมคล้ายที่มีความสัมพันธ์ระหว่างค่ามุมต่าง ๆ มุม  $\alpha$   $\beta$  และ  $\gamma$  เมื่อสังเกตจะพบว่าความสัมพันธ์ของมุม  $\gamma$  กับวัตถุหน้าเลนส์มีขนาด  $y_o$  และระยะห่างจากเลนส์  $S_o$  จะมีค่าเท่ากับมุม มุม  $\gamma$  กับวัตถุหลังเลนส์มีขนาด  $y_i$  และระยะห่างจากเลนส์  $S_i$

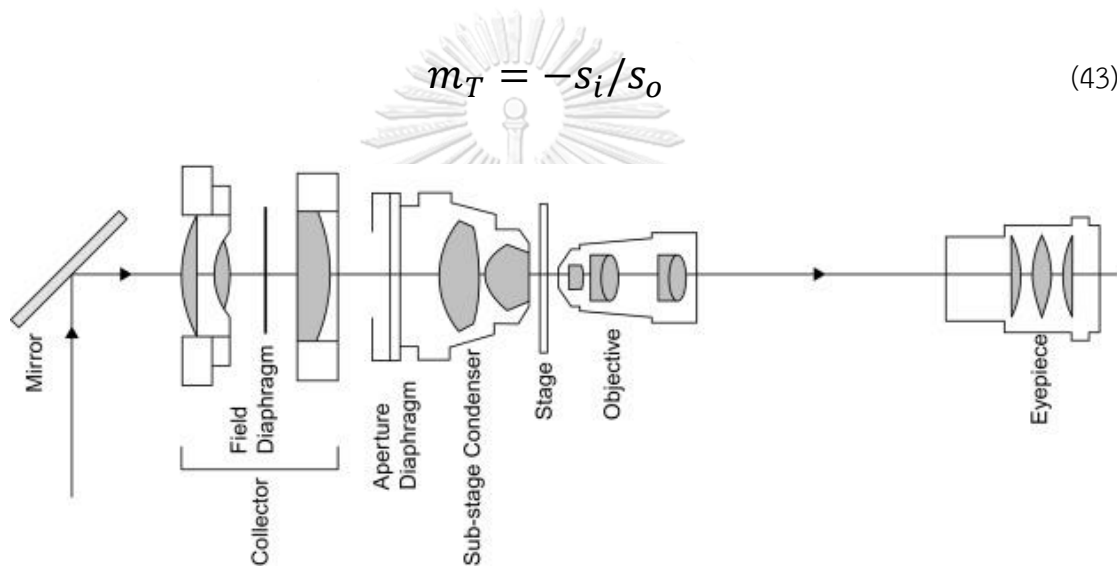
$$\tan \alpha = \frac{y_o}{f_i} = \frac{y_i}{x_i} \quad (40)$$

$$\tan \beta = \frac{y_o}{x_o} = \frac{y_i}{f_o} \quad (41)$$

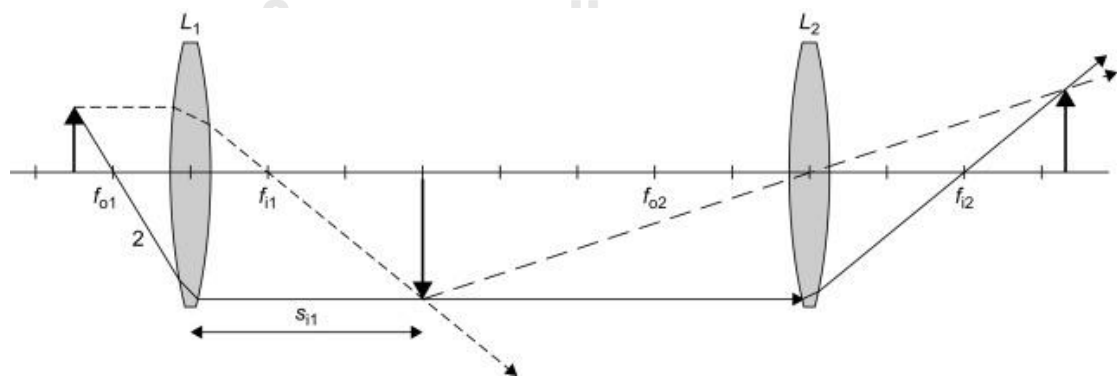
$$\tan \gamma = \frac{y_o}{s_o} = \frac{y_i}{s_i} \quad (42)$$

เมื่อ  $y_i/y_o = s_i/s_o$  ทำให้สามารถพิสูจน์สมการกำลังการขยายของเลนส์ (Transverse magnification) ได้ว่า

$$m_T = -s_i/s_o \quad (43)$$



ภาพที่ 19 แบบร่างของชุดเลนส์ประกอบภายในกล้องจุลทรรศน์ชนิดไบรท์ฟิลด์ [17]



ภาพที่ 20 เลนส์รวมแสง 2 ชั้นวางตัวบนแกนระนาบแสดงภาพวัตถุจริง [17]

กล้องจุลทรรศน์ในปัจจุบันมีเลนส์ภายในจำนวนหลายชิ้นอยู่ภายใน เลนส์ใกล้ตา (Eyepiece lens) เลนส์ใกล้วัตถุ (Objective lens) แท่นขยายภาพ (Collector) และแท่นรวมแสง (Substage

condenser) เรียกว่า เลนส์ประกอบ (Compound lenses) จากตัวอย่างในภาพที่ 17 ซึ่งเป็นหนึ่งในหลักการของชนิดเลนส์ภายใน จึงสามารถคำนวณลักษณะการเกิดภาพได้ โดยใช้วิธีดูแนวเส้นทางแสงหรือเส้นรังสีภายในเลนส์ประกอบ จากสมการคำนวณเลนส์บาง (Thin lens) เมื่อเลนส์ 2 ชั้นวางแยกจากกันด้วยระยะทางที่มากกว่าผลรวมของความยาวโฟกัส ให้อนุมานว่าเกิดภาพจริงจากเลนส์ชั้นแรกทำหน้าที่เป็นวัตถุจริงของเลนส์ชั้นที่สอง จึงสามารถคำนวณตำแหน่งภาพที่เกิดขึ้นจากเลนส์ประกอบได้จากสมการ

$$s_i = \frac{f_2 d - [f_1 f_2 s_o / (s_o - f_1)]}{d - f_2 - [f_1 s_o / (s_o - f_1)]} \quad (44)$$

เมื่อ  $f_1$  เป็นความยาวโฟกัสของเลนส์ชั้นแรก  $f_2$  เป็นความยาวโฟกัสของเลนส์ชั้นที่สอง  $d$  เป็นระยะห่างระหว่างเลนส์ทั้งสองชั้น จากสมการกำลังการขยายของเลนส์ (Transverse magnification) ของเลนส์แต่ละชั้น จึงสามารถคิดกำลังการขยายรวมของชุดเลนส์ได้ โดยการนำกำลังการขยายมาคูณกันโดยตรง

$$m_{Total} = (m_{T1})(m_{T2}) \quad (45)$$

$$m_{Total} = \frac{f_1 s_i}{d(s_o - f_1) - s_o f_1} \quad (46)$$

เมื่อค่า  $m_{Total}$  มีค่าเป็นบวกจะเกิดภาพหัวตั้ง หากค่าเป็นลบจะเป็นภาพหัวกลับ และ ความยาวโฟกัสด้านหน้าเลนส์ (Front focal length)  $f_{Ff}$  และความยาวโฟกัสด้านหลังเลนส์ (Back focal length)  $f_{Bf}$  ได้จากสมการ

$$f_{Ff} = \frac{f_1(d-f_2)}{d-(f_1+f_2)} \quad (47)$$

$$f_{Bf} = \frac{f_2(d-f_1)}{d-(f_1+f_2)} \quad (48)$$

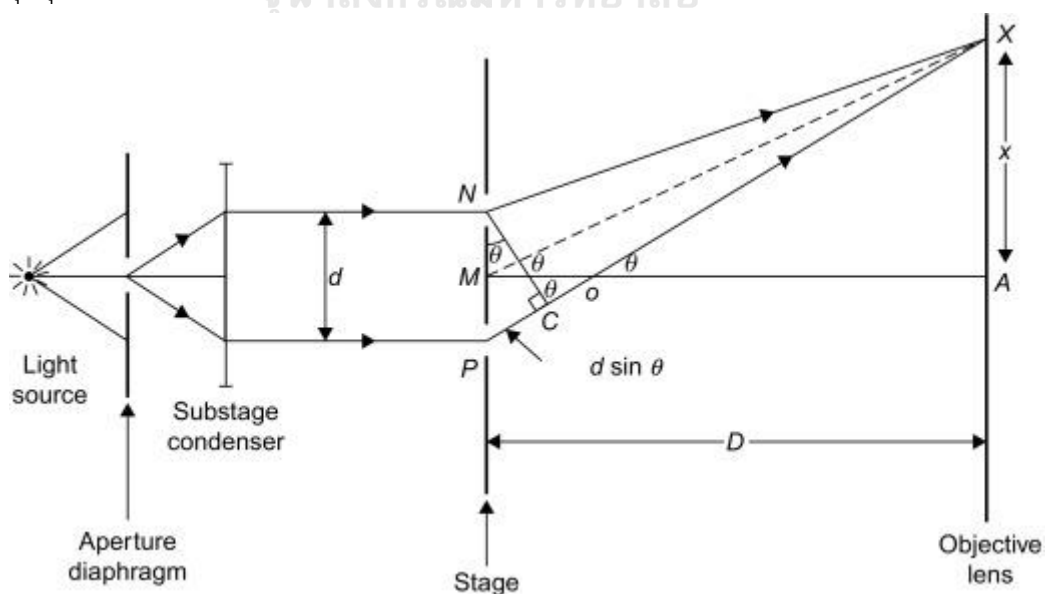
ถ้าระยะห่างระหว่างเลนส์สองชั้นเป็น 0 ทำให้ความยาวโฟกัสด้านหน้าเลนส์ และด้านหลังเลนส์มีค่าเท่ากัน จึงสามารถคำนวณระยะโฟกัสได้จากสมการ โดยที่  $f_1$  เป็นความยาวโฟกัสของเลนส์ชั้นแรก  $f_2$  เป็นความยาวโฟกัสของเลนส์ชั้นที่สอง

$$\frac{1}{f} = \frac{1}{f_1} + \frac{1}{f_2} \quad (49)$$

ส่วนกลับของความยาวโฟกัส สามารถเรียกอีกอย่างหนึ่งว่า กำลังการหักเหแสงของเลนส์ (Diopter หรือ Dioptric power) มีหน่วยเป็น D ถ้ากำลังการหักเหแสงของเลนส์มีค่ามาก จะทำให้ความยาวโฟกัสลดลง สามารถคิดค่ากำลังการหักเหแสงรวมได้จากสมการ

$$D_{Total} = D_1 + D_2 ; D_i = 1/f_i \quad (50)$$

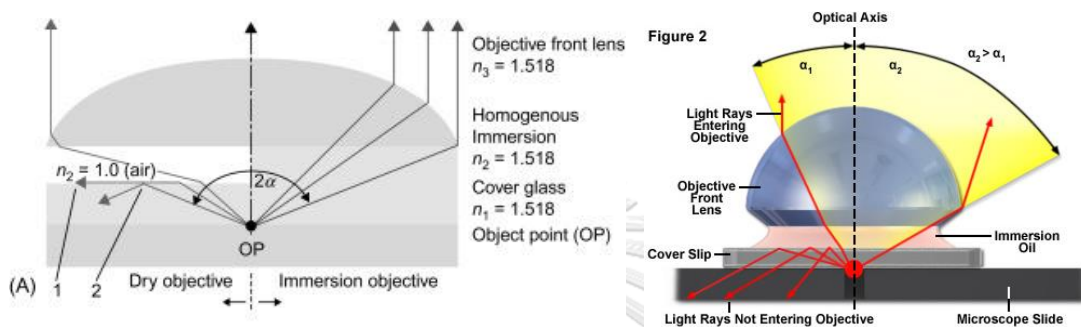
อีกค่าหนึ่งที่น่าสนใจในการบ่งบอกความสามารถของกล้องจุลทรรศน์ คือ ระดับความละเอียด (Resolving power or Resolution) เป็นขีดความสามารถที่กล้องจุลทรรศน์จะให้การมองเห็นอย่างชัดเจนภายใต้กำลังขยายที่ใช้ ผู้ใช้จะสามารถแยกวัตถุได้อย่างชัดเจน โดยไม่จำกัดกับขนาดของวัตถุ และความสว่างของพื้นหลังที่เหมาะสม นอกจากความยาวโฟกัสและกำลังการขยายแล้วนั้น ระดับความละเอียด แล้วยังเกิดจากข้อจำกัดของรูรับแสงเชิงมุม (Angular aperture) ของเลนส์ใกล้วัตถุ ซึ่งเป็นสิ่งหนึ่งที่ต้องให้ความสนใจ จากแผนภาพจำลองการเลี้ยวเบนของแสงของฟรอนฮอฟ (Fraunhofer diffraction) กล้องจุลทรรศน์จะไม่สามารถให้ความละเอียดวัตถุที่มีขนาดเล็กไปกว่าระยะ  $d$  พฤติกรรมการเลี้ยวเบนของแสง (Diffraction) ที่เกิดจากการแทรกสอด ถ้าให้ช่องแคบ  $N, M$  เป็นแหล่งกำเนิดแสงห่างกันเป็นระยะ  $d$  เมื่อแสงเดินทางจากช่องแคบมาถึงผิวหน้าเลนส์ใกล้วัตถุที่จุด  $X$  ด้วยระยะทางที่ต่างกัน ทำให้ได้ค่าผลต่างของระยะทางที่แสงเดินทางมาได้เป็น



ภาพที่ 21 แผนภาพจำลองการเลี้ยวเบนของแสงของฟรอนฮอฟ (Fraunhofer diffraction) [18]

$$NX - PX = d \sin \theta = n\lambda \quad (51)$$

$$d = \frac{\lambda}{n \sin \theta} \quad (52)$$

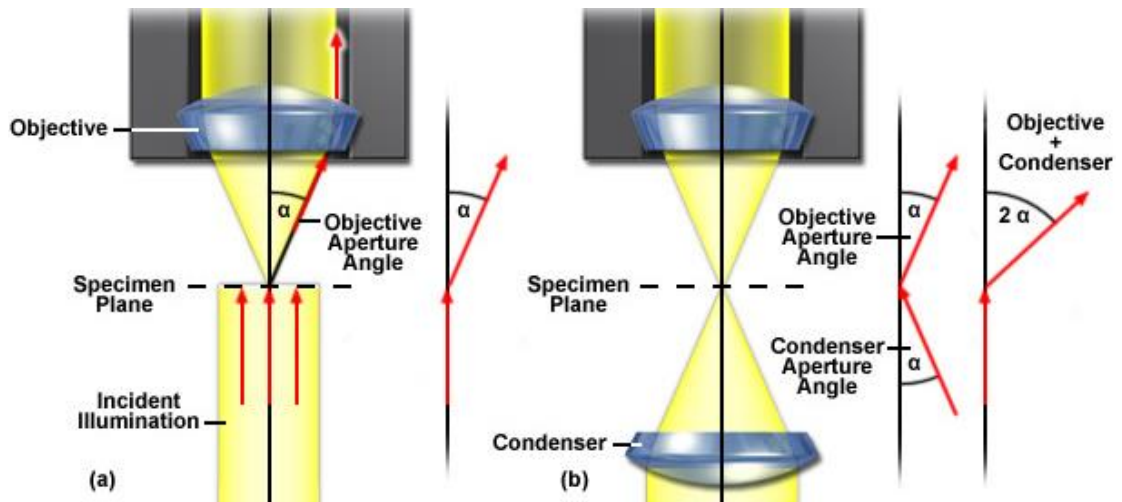


ภาพที่ 22 แผนภาพแสดงการกระเจิงแสงผ่านตัวกลางอากาศ [18] และ Immersion oil [19]

เนื่องจากมุม  $\theta$  มีค่าน้อยมาก จึงทำให้  $\sin \theta \approx \tan \theta = x/D$  และในบางผู้ผลิตกล้องจุลทรรศน์ได้เรียกมุม  $\theta$  แทนด้วยสัญลักษณ์  $\alpha$  or  $\mu$  ในสมการที่ 33 และ 34 ความสามารถในการรวมแสงของเลนส์วัตถุ (Numerical Aperture or N.A.) ถูกนิยามโดย Ernst Karl Abbe ในปี ค.ศ.1881 โดยเรียก  $n \sin \theta$  ใหม่ว่า N.A. หรือ NA และจากการพิสูจน์ค่าดัชนีหักเหของตัวกลางด้วยกฎของสเนลในสมการที่ 7 จึงส่งผลให้ค่า  $n$  ในสมการที่ 34 คือค่าดัชนีหักเหของตัวกลางระหว่างผิวหน้าเลนส์วัตถุ (Objective lens) กับแผ่นกระจกปิดสไลด์ (Cover slip) ซึ่งอากาศ  $n = 1$  และน้ำมัน Immersion oil มีค่า  $n = 1.25$  ถึง  $1.50$

$$\text{Numerical Aperture (N.A.)} = \frac{\lambda}{d} \quad (53)$$





ภาพที่ 23 การเกิดมุมของความสามารถในการรวมแสงของเลนส์วัตถุ (N.A.) [19]

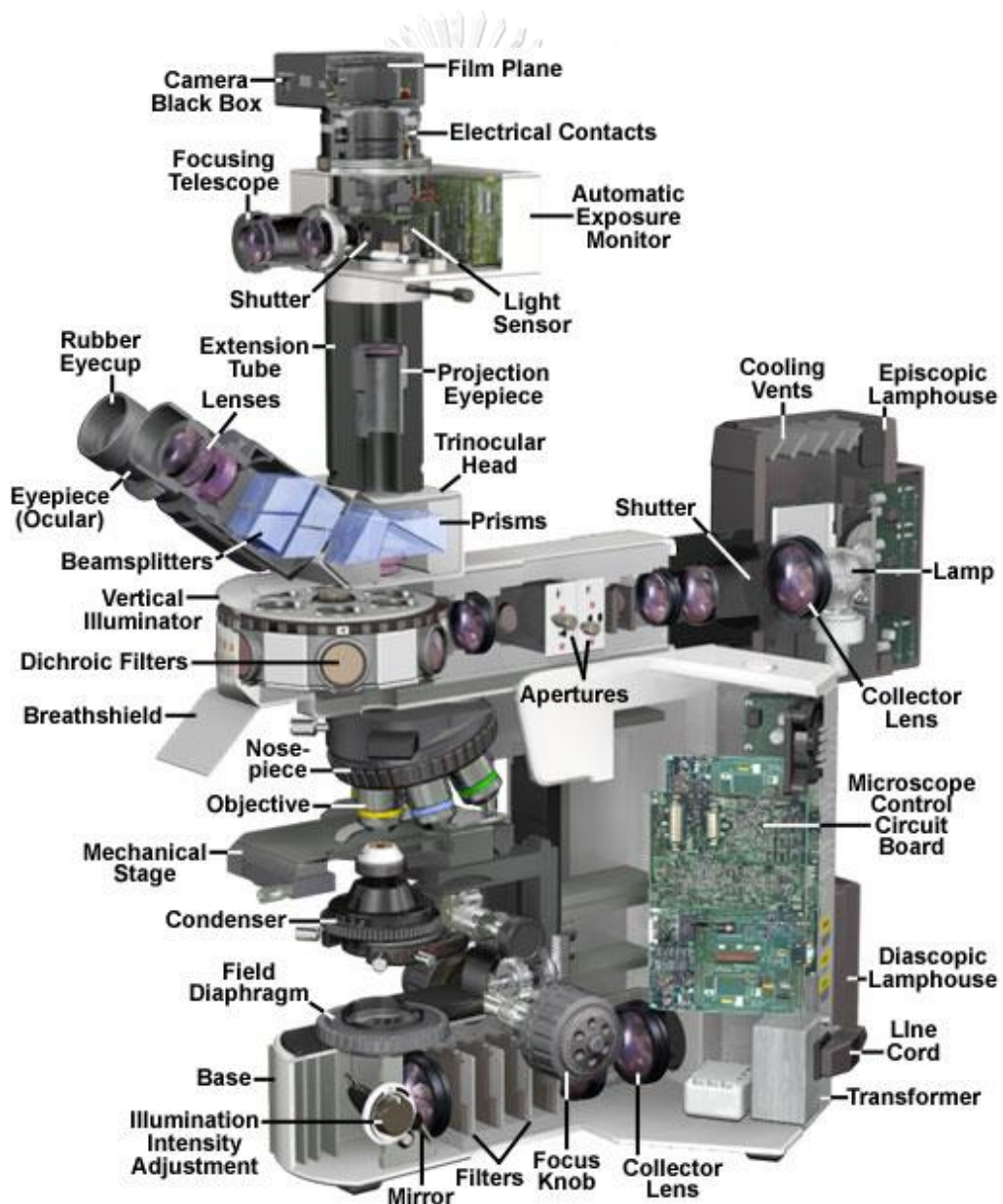
ผู้ผลิตกล้องจุลทรรศน์ส่วนใหญ่จะให้กำลังการขยายของเลนส์ (Transverse magnification) และความสามารถในการรวมแสงของเลนส์วัตถุ (Numerical Aperture or N.A.) ยิ่งค่า N.A. สูง ความคมชัดของวัตถุจะยิ่งสูงขึ้น ตามตารางที่ 6

ตารางที่ 6 กำลังการขยายและค่า Numerical Aperture (N.A)

กำลังการขยาย	Plan Achromat	Plan Fluorite	Plan Apochromat
0.5X	0.025	N/A	N/A
1X	0.04	N/A	N/A
2X	0.06	0.08	0.10
4X	0.10	0.13	0.20
10X	0.25	0.30	0.45
40X	0.65	0.75	0.95
60X	0.75	0.85	0.95
100X (Oil)	1.25	1.30	1.40

## 2.7 คุณลักษณะของกล้องจุลทรรศน์ชนิดโปรทไฟลด์

กล้องจุลทรรศน์ชนิดโปรทไฟลด์ (Bright field microscope) จัดเป็นหนึ่งในกล้องจุลทรรศน์เชิงซ้อน (Compound microscope) นิยมใช้งานในห้องปฏิบัติการทั่วไป เป็นอุปกรณ์ที่ทำหน้าที่ขยายรายละเอียดของวัตถุขนาดเล็กให้มีความชัดเจนมากยิ่งขึ้น ระบบเลนส์ทำหน้าที่ขยาย 2 ชุด คือ เลนส์ใกล้ตา (Eyepiece lens) และเลนส์ใกล้วัตถุ (Objective lens) หลักการทำงาน เมื่อแสงไฟจากแหล่งกำเนิด (Lamp) จะถูกรวมผ่านเลนส์รวมแสง (Condenser) ก่อนไปตกกระทบบนวัตถุตัวอย่างบนแผ่นสไลด์ที่วางไว้บนแท่นวาง (Mechanical stage) และส่งผ่านไปยังเลนส์ใกล้วัตถุเพื่อทำหน้าที่ขยายภาพและส่งผ่านไปยังตาของมนุษย์ต่อไป [20]



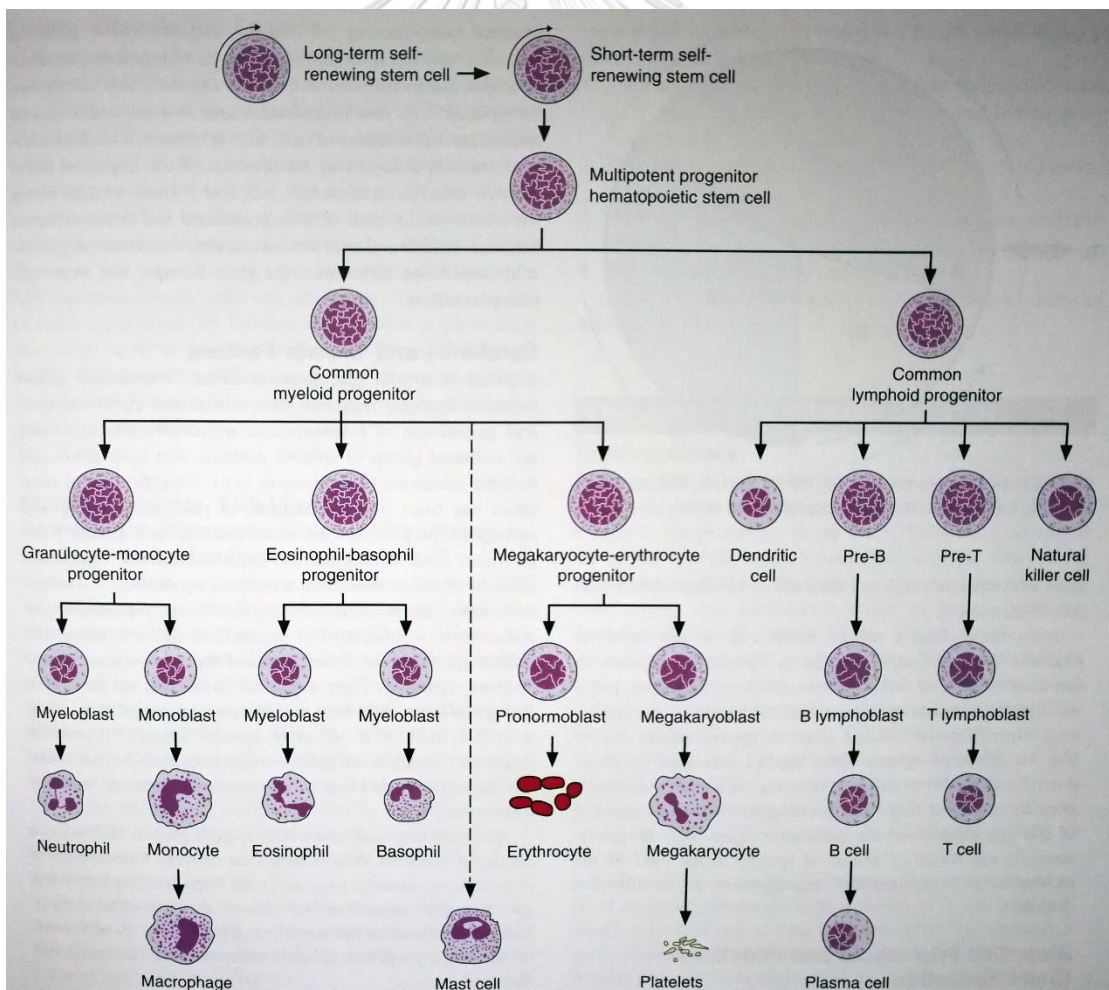
ภาพที่ 24 องค์ประกอบชิ้นส่วนภายในของกล้องจุลทรรศน์แบบเชิงซ้อน [21]

องค์ประกอบของกล้องจุลทรรศน์แบบเชิงซ้อนแสดงในภาพที่ 24 ประกอบด้วยชิ้นส่วนเลนส์ชิ้นส่วนทางกล และชิ้นส่วนทางไฟฟ้า แตกต่างกันไปตามรุ่นและคุณสมบัติของกล้องจุลทรรศน์แต่ละรุ่น ผู้วิจัยจึงขอกำหนดถึงองค์ประกอบหลัก ๆ คือ

- 1.) ส่วนฐาน (Base) เป็นส่วนด้านล่างสุดสำหรับวางกับพื้นโต๊ะห้องปฏิบัติการ และเป็นส่วนรับน้ำหนักทั้งหมดของตัวกล้อง ภายในประกอบด้วยวงจรไฟฟ้า (Circuit board) และแหล่งกำเนิดแสง (Illumination source) สำหรับส่องผ่านแสง
- 2.) ส่วนแขน (Arm) เป็นลำตัวกล้องสำหรับยึดติดอุปกรณ์ส่วนบนเข้ากับส่วนฐาน ใช้สำหรับสอดมือเพื่อยกกล้องจุลทรรศน์เคลื่อนย้าย
- 3.) แหล่งกำเนิดแสง (Illumination source) เป็นหลอดไฟที่มีหน้าหน้าที่ให้แสงสว่าง
- 4.) ม่านรูปรับแสง (Field diaphragm) ทำหน้าที่ปรับม่านกันแสงให้มีความเหมาะสมกับการใช้งาน แต่ถ้าเป็นกล้องจุลทรรศน์บางรุ่นจะใช้เป็นวงจรอิเล็กทรอนิกส์ปรับลดความเข้มแสงกับแหล่งกำเนิดแสงโดยตรง
- 5.) เลนส์รวมแสง (Condenser) เป็นชุดเลนส์รวมแสงที่ทำหน้าที่บีบลำแสง ก่อนไปตกกระทบบนวัตถุตัวอย่างบนแผ่นสไลด์ให้มีความเหมาะสมกับ ความสามารถในการรวมแสงของเลนส์วัตถุ (Numerical Aperture or N.A.)
- 6.) แท่นวาง (Mechanical stage) เป็นแท่นโลหะสำหรับวางวัตถุตัวอย่างบนแผ่นสไลด์ มีรูตรงกลางที่ยอมให้ลำแสงจากเลนส์รวมแสงส่องไปยังวัตถุ มีชุดจับยึดแผ่นสไลด์ และกลไกการเคลื่อนที่ในแนวแกน X และ Y เพื่อให้สามารถย้ายตำแหน่งวัตถุขณะส่องกล้อง
- 7.) วงล้อปรับระยะ (Adjustment knob) ส่วนการเคลื่อนที่ในแนวแกน Z สำหรับปรับความคมชัดของภาพ จากการปรับระยะระหว่างผิวหน้าเลนส์ใกล้วัตถุจนถึงแผ่นกระจกปิดสไลด์ ประกอบด้วยวงล้อปรับระยะ 2 แบบ คือ วงล้อปรับระยะอย่างหยาบ (Coarse adjustment knob) และวงล้อปรับระยะแบบละเอียด (Fine adjustment knob)
- 8.) แท่นเลนส์ใกล้วัตถุ (Nosepiece) เป็นแผ่นโลหะจานหมุน สำหรับติดตั้งชุดเลนส์ใกล้วัตถุ
- 9.) เลนส์ใกล้วัตถุ (Objective lens) เลนส์สำหรับขยายภาพวัตถุตัวอย่าง ประกอบด้วยชุดเลนส์ 4-6 ชุด แต่ชุดหลักๆ ที่มีในกล้องจุลทรรศน์เชิงซ้อนทั่วไป ได้แก่ 4X, 10X, 40X และ 100X โดยที่ X หมายถึงกำลังการขยายของเลนส์ (Transverse magnification) และมีการระบุค่าความสามารถในการรวมแสงของเลนส์วัตถุ (Numerical Aperture or N.A.)
- 10.) ส่วนหัวกล้อง (Trinocular head or Binocular head) เป็นส่วนถ่ายทอดภาพไปยังเลนส์ใกล้ตา (Eyepiece lens) และท่อส่งภาพแนวตั้ง (Projection tube) สำหรับสวมกับตัวรับรูปภาพ (Image sensor) ของกล้องถ่ายภาพดิจิทัล (Digital camera)

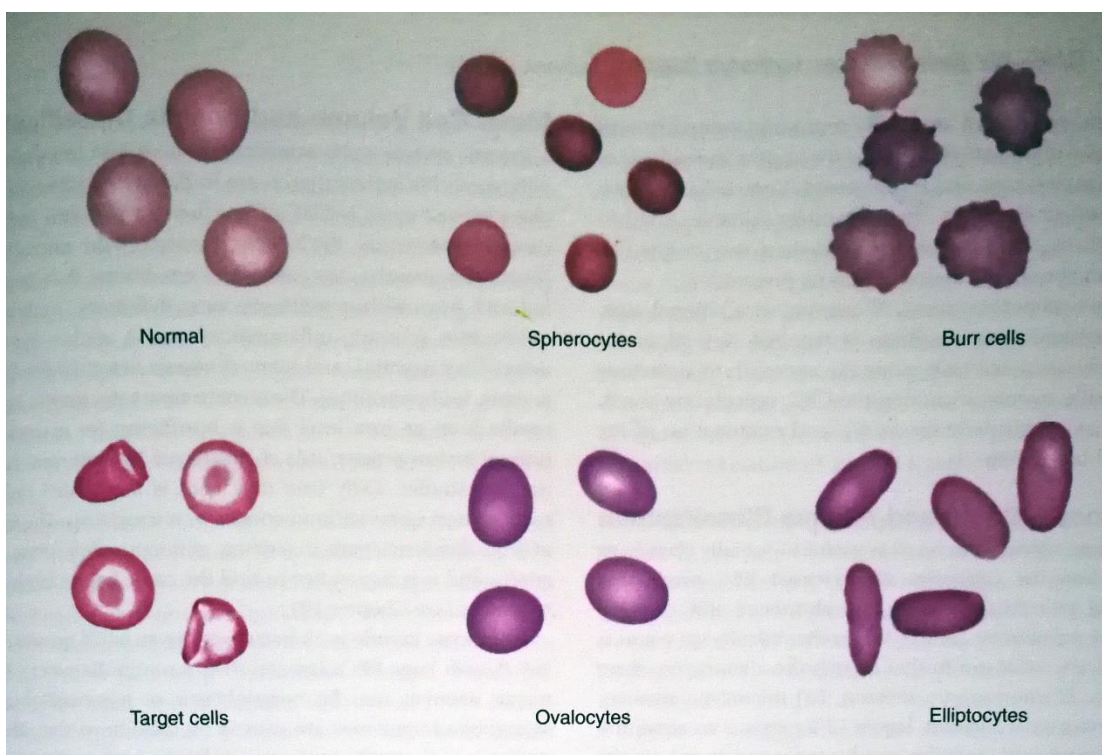
## 2.8 รายละเอียดองค์ประกอบเซลล์ขนาดเล็ก

การศึกษาเซลล์ขนาดเล็กหรือจุลชีพนั้น จำเป็นต้องใช้กล้องจุลทรรศน์ในการศึกษาและวินิจฉัยคุณลักษณะของสิ่งมีชีวิต ตัวอย่างเซลล์จุลชีพ ได้แก่ เซลล์โพรคาริโอต (Prokaryote) มีขนาด 0.2-10 ไมโครเมตร มีเยื่อหุ้มเซลล์ (Cell membrane) แต่ไม่มีเยื่อหุ้มนิวเคลียส (Nucleus membrane) เช่น แบคทีเรีย (Bacteria) สาหร่ายสีเขียวแกมน้ำเงิน (Cyanobacteria) เป็นต้น และเซลล์ยูคาริโอต (Eukaryote) มีขนาด 10-1000 ไมโครเมตร มีองค์ประกอบภายในเซลล์ที่ซับซ้อนกว่า และมีเยื่อหุ้มนิวเคลียส เช่น เซลล์สัตว์ (Animal cell) เซลล์พืช (Plant cell) เซลล์โพรโทซัว (Protozoa) รา (Fung) และสาหร่าย (Algae) เป็นต้น เซลล์เม็ดเลือดจัดเป็นเซลล์ขนาดเล็กภายในร่างกายมนุษย์และสัตว์ที่มีความสำคัญต่อการดำรงชีวิตและการวินิจฉัยโรค เช่น การตรวจหาการติดเชื้อสามารถดูได้จากปริมาณเม็ดเลือดชนิดต่าง ๆ ประกอบไปด้วย จำนวนเม็ดเลือดแดง (Red blood cell : RBC หรือ Erythrocyte) เม็ดเลือดขาว (White blood cell : WBC หรือ Leukocyte) และเกล็ดเลือด (Platelets หรือ Thrombocyte) [22]



ภาพที่ 25 ฝั่งการสร้างโลหิต (Hematopoiesis) ที่แสดงการแบ่งชนิดของเซลล์ [22]

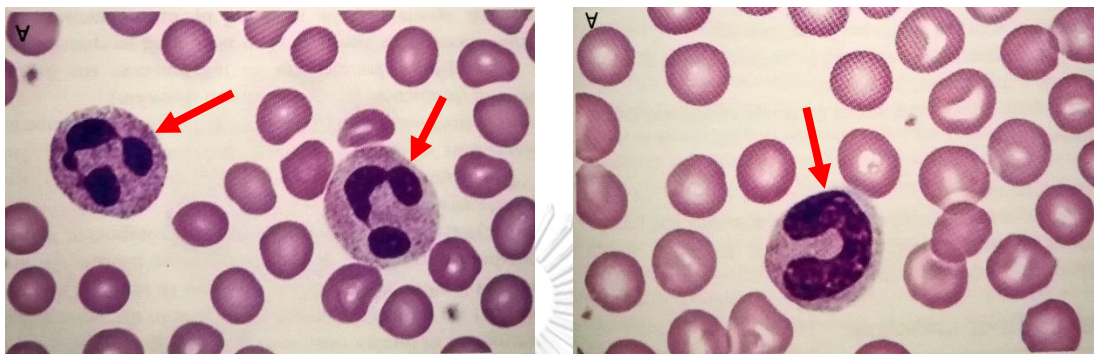
เม็ดเลือดแดง (Red blood cell : RBC หรือ Erythrocyte) มีจำนวนมากที่สุดในองค์ประกอบเลือด มีรูปร่างกลมตรงกลางเว้าเข้าหากัน (Biconcave disc) ไม่มีนิวเคลียส มีขนาดเส้นผ่านศูนย์กลาง 7-8 ไมโครเมตร ความหนาประมาณ 1.5-2.5 ไมโครเมตร สีชมพูออกแดง บริเวณกลางจะยอมไม่ติดสีในกระบวนการย้อมสี (Stained blood smear) ภายในมี Hemoglobin (Hb) ทำหน้าที่ลำเลียงออกซิเจนและคาร์บอนไดออกไซด์ สำหรับกระบวนการหายใจระดับเซลล์



ภาพที่ 26 ลักษณะของเม็ดเลือดแดงปกติ (Normal erythrocyte) และรูปร่างผิดปกติ [22]

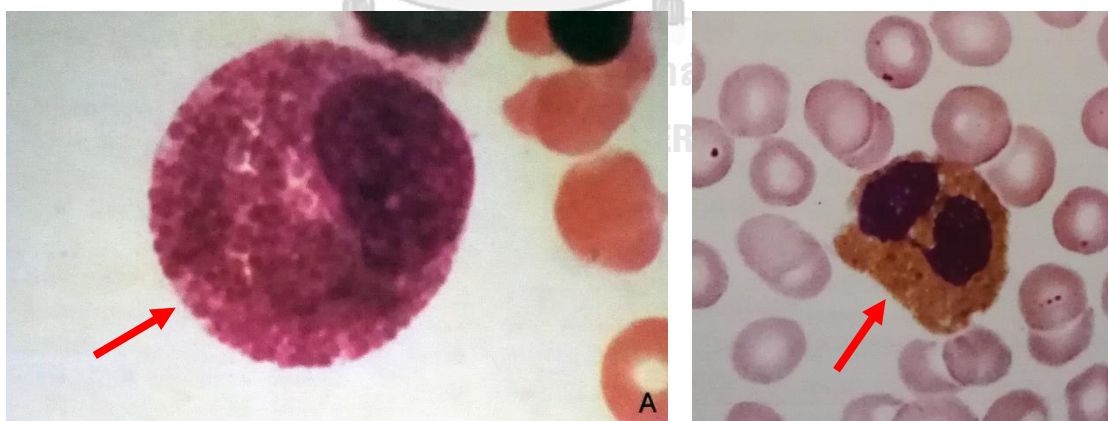
เม็ดเลือดขาว (White blood cell : WBC หรือ Leukocyte) มีจำนวนรองลงมาในองค์ประกอบเลือด มีรูปร่างแตกต่างกันไปตามลักษณะรูปร่างของเซลล์ เม็ดเลือดขาวทุกชนิดจะมีนิวเคลียสอยู่ภายใน ทำหน้าที่ด้านภูมิคุ้มกันทั้งสิ่งแปลกปลอมที่ก่อให้เกิดโรค และสารก่อให้เกิดอาการแพ้ เม็ดเลือดขาวแบ่งออกเป็น 2 กลุ่ม คือ กลุ่มที่มีแกรนูโล (Granulocyte) หรือนิวเคลียสแบบหลายกลุ่ม (Polymorphonuclear) ประกอบด้วย นิวโทรฟิล (Neutrophil) อีโอซิโนฟิล (Eosinophil) และเบโซฟิล (Basophil) ซึ่งมีลักษณะนิวเคลียสแยกออกเป็นพู (Lobes) และไม่มีแกรนูโล (Agranulocyte) หรือนิวเคลียสแบบเดี่ยว (Mononuclear) ประกอบด้วย โมโนไซต์ (Monocyte) และลิมโฟไซต์ (Lymphocyte)

นิวโทรฟิล (Neutrophil) เป็นเม็ดเลือดขาวที่พบมากที่สุดในระบบภูมิคุ้มกัน ร้อยละ 50 ถึง 70 ทำหน้าที่กลืนกินสิ่งแปลกปลอม เช่น แบคทีเรีย ปรสิต โดยวิธีการโอบล้อมสิ่งแปลกปลอมไปย่อยสลายภายในเซลล์ (Phagocytosis) มีขนาดเส้นผ่านศูนย์กลาง 9-15 ไมโครเมตร มีนิวเคลียส 2-5 พู จึงเป็นเม็ดเลือดขาวชนิดแรก ๆ ที่สามารถตรวจพบได้ในกระบวนการอักเสบ



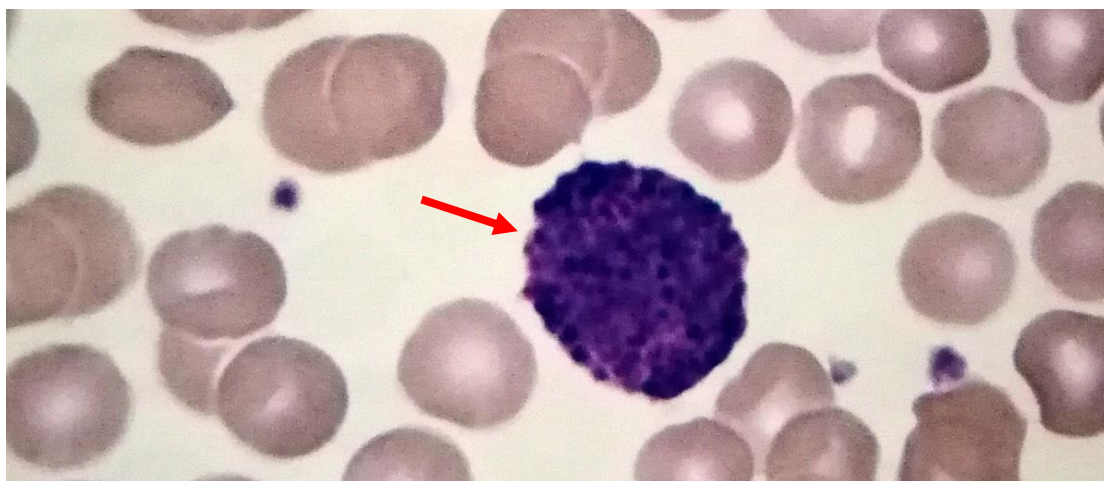
ภาพที่ 27 ลักษณะของเม็ดเลือดขาว Neutrophil แบบ Segmented (ซ้าย) และ Band (ขวา)

อีโอซิโนฟิล (Eosinophil) เป็นเม็ดเลือดขาวที่พบได้ประมาณร้อยละ 1-3 ของระบบภูมิคุ้มกัน มีหน้าที่เกี่ยวกับตอบสนองต่อสารที่ก่อให้เกิดอาการแพ้ (Allergic) และการติดเชื้อจากปรสิต (Parasitic helminths) มีขนาดเส้นผ่านศูนย์กลาง 9-15 ไมโครเมตร เท่ากับนิวโทรฟิล มีนิวเคลียส 1 พู ในระยะเซลล์ต้นกำเนิด (Myelocyte) และ 2 พู ในระยะเต็มวัย (Mature)



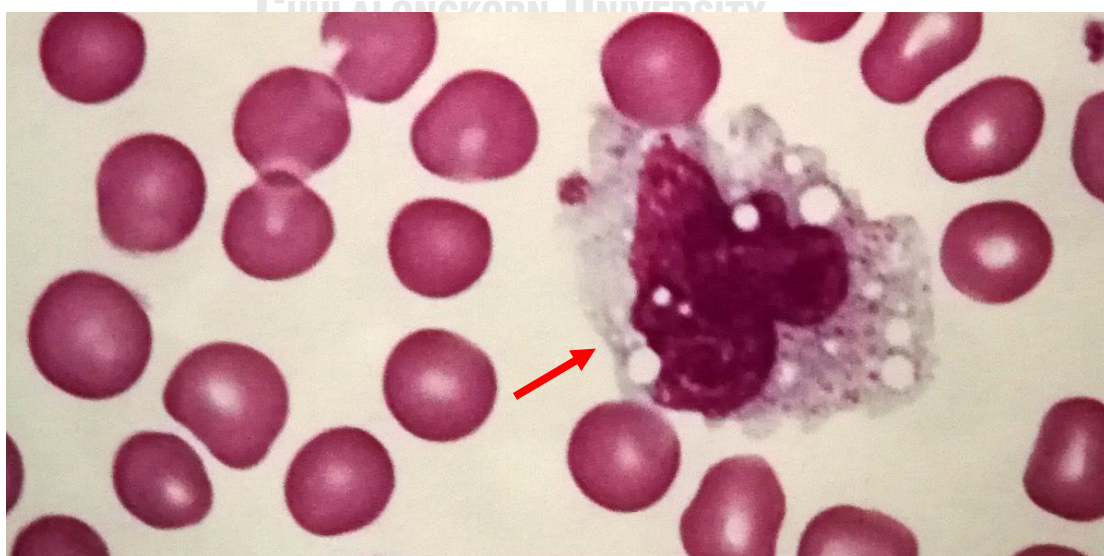
ภาพที่ 28 ลักษณะของเม็ดเลือดขาว Eosinophil แบบ Myelocyte (ซ้าย) และ Mature (ขวา)

เบโซฟิล (Basophil) เป็นเม็ดเลือดขาวที่พบได้ร้อยละ 0-2 ของระบบภูมิคุ้มกัน มีหน้าที่กระตุ้นเกล็ดเลือดไม่ให้แข็งตัว (Platelet activating) ด้วยสาร Heparin สร้างสารก่อภูมิแพ้ให้ร่างกาย การเกิดการอักเสบ (Histamine) มีขนาดเส้นผ่านศูนย์กลาง 10-16 ไมโครเมตร มีนิวเคลียสกระจายตัวอยู่ภายในเซลล์เป็นแบบ Basophilic granules



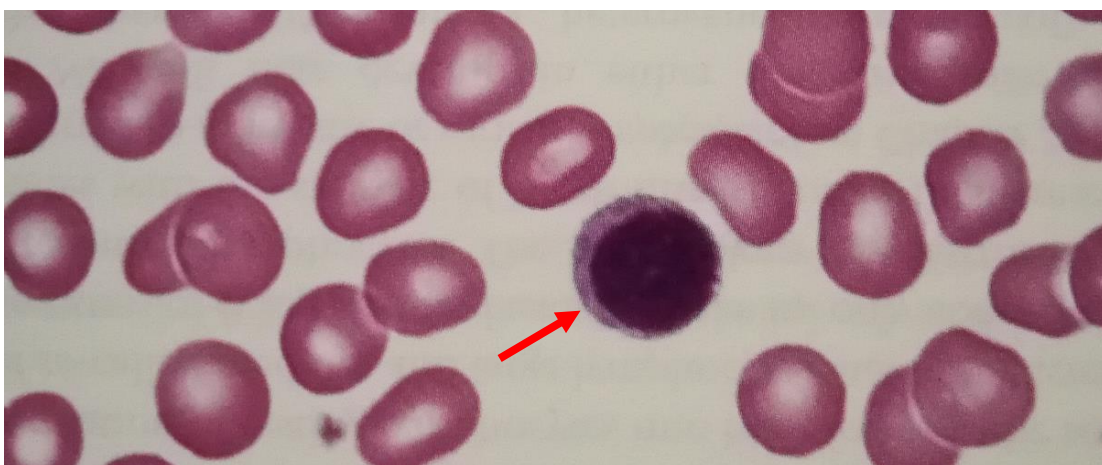
ภาพที่ 29 ลักษณะของเม็ดเลือดขาว Basophil

โมโนไซต์ (Monocyte) เป็นเม็ดเลือดขาวที่มีขนาดใหญ่ที่สุด มีขนาดเส้นผ่านศูนย์กลาง 12-20 ไมโครเมตร พบได้ร้อยละ 2-11 ของระบบภูมิคุ้มกัน ทำหน้าที่กำจัดสิ่งแปลกปลอม และการอักเสบภายในเนื้อเยื่อและอวัยวะ (Inflammatory macrophages) นิวเคลียสรูปร่างคล้ายउँงเท้าสัตว์ หรือเม็ดถั่ว มีช่องว่างภายในนิวเคลียสที่แสดงชัดเจนแตกต่างจากลิมโฟไซต์ (Lymphocyte)



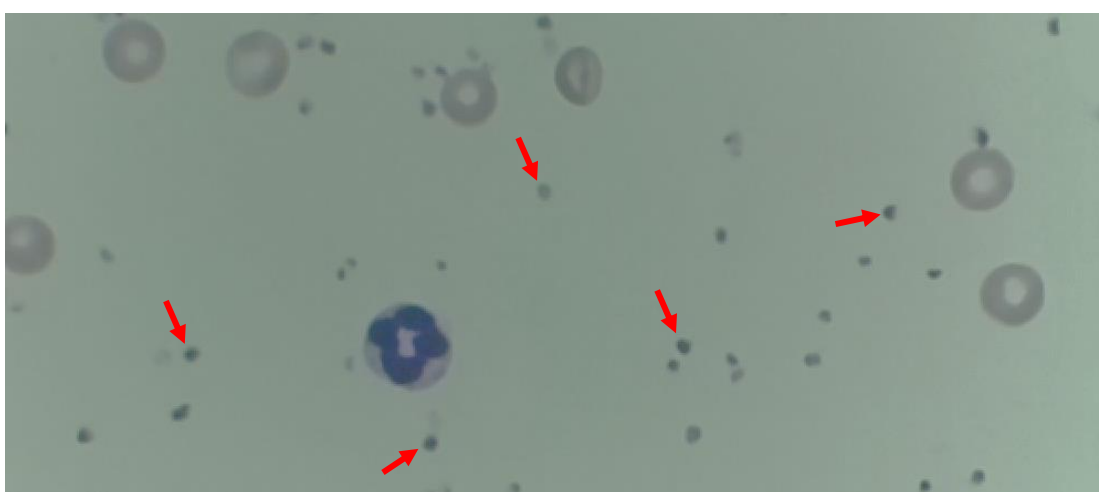
ภาพที่ 30 ลักษณะของเม็ดเลือดขาว Monocyte

ลิมโฟไซต์ (Lymphocyte) เป็นเม็ดเลือดที่มีจำนวนรองจากนิวโทรฟิล (Neutrophil) พบได้ประมาณร้อยละ 18-42 ของระบบภูมิคุ้มกัน ทำหน้าที่เกี่ยวกับภูมิคุ้มกันเปลี่ยนสภาพไปเป็น T-cell และ B-cell ในการกระตุ้น Antibody ลิมโฟไซต์สามารถพบได้ 2 ขนาด คือ ลิมโฟไซต์ขนาดใหญ่ มีเส้นผ่านศูนย์กลาง 12-18 ไมโครเมตร มีขนาดใหญ่กว่าเม็ดเลือดแดง และลิมโฟไซต์ขนาดเล็ก มีขนาดเส้นผ่านศูนย์กลาง 7-8 ไมโครเมตร มีขนาดเล็กกว่าเม็ดเลือดแดง



ภาพที่ 31 ลักษณะของเม็ดเลือดขาว Lymphocyte

เกล็ดเลือด (Platelets หรือ Thrombocyte) เป็นเซลล์เม็ดเลือดชนิดไม่มีสี ไม่มีนิวเคลียส และมีขนาดเล็กที่สุด มีเส้นผ่านศูนย์กลางประมาณ 2-4 ไมโครเมตร มีรูปร่างทรงกลมแบนและทรงรีแบน เกิดจากการสลายตัวของเซลล์ Megakaryocyte ย่อมติตสีชมพูอ่อนในกระบวนการย้อมสี มีหน้าที่ช่วยในการทำให้เลือดแข็งตัวในบริเวณที่เกิดบาดแผลและเนื้อเยื่อฉีกขาด



ภาพที่ 32 ลักษณะการกระจายตัวของเกล็ดเลือด (Platelet) ร่วมกับเม็ดเลือดแดงและขาว



ตารางที่ 7 ค่ามาตรฐานของเม็ดเลือดแดง เม็ดเลือดขาว และเกล็ดเลือด [22]

ชนิด	เพศชาย		เพศหญิง		หน่วย
	ร้อยละ	จำนวน	ร้อยละ	จำนวน	
Erythrocytes		4.60-6.00		4.00-5.40	$\times 10^6/\mu\text{L}$
Neutrophils	50-70	2.3-8.1	50-70	2.3-8.1	$\times 10^3/\mu\text{L}$
Lymphocytes	18-42	0.8-4.8	18-42	0.8-4.8	$\times 10^3/\mu\text{L}$
Monocytes	2-11	0.45-1.3	2-11	0.45-1.3	$\times 10^3/\mu\text{L}$
Eosinophils	1-3	0-0.4	1-3	0-0.4	$\times 10^3/\mu\text{L}$
Basophils	0-2	0-0.1	0-2	0-0.1	$\times 10^3/\mu\text{L}$
Platelets	150-450		150-450		$\times 10^3/\mu\text{L}$

## 2.9 งานวิจัยที่เกี่ยวข้องกับการนับเซลล์ขนาดเล็ก

สำหรับการประมวลผลภาพเพื่อใช้ในการวิเคราะห์ และการจำแนกชนิดของวัตถุ นั้น มีการใช้เทคนิคต่าง ๆ หลายรูปแบบ โดยมีวัตถุประสงค์ให้การวิเคราะห์ภาพมีความง่ายมากขึ้น วิธีการนับเซลล์ในรูปแบบที่ปรากฏจะอาศัยวิธีการประมวลผลภาพด้วยวิธีการทางขั้นตอนวิธีและสมการทางคณิตศาสตร์ เช่น การจำแนกความแตกต่างของนิวเคลียส (Nucleus) ไซโทพลาซึม (Cytoplasm) ลักษณะรูปร่างของเซลล์ (Cell morphology) โดยการแบ่งส่วนภาพ (Image segmentation) ด้วยขั้นตอนวิธี Canny edge, Otsu's method, clustering method ตลอดจนการประยุกต์ใช้โครงข่ายประสาท (Neural network) และซัพพอร์ตเวกเตอร์แมชชีน (Support Vector Machine : SVM) ในการจำแนกเซลล์จากภาพ แต่ละวิธีการจะใช้ขั้นตอนที่คล้ายกัน คือ การแบ่งส่วนภาพ การดึงคุณลักษณะ และการจำแนกประเภท

Qingmin Liao et al. [23] นำเสนอวิธีการการแบ่งส่วนภาพอย่างแม่นยำ (Accurate segmentation) จากการวิเคราะห์รูปร่างของเซลล์เม็ดเลือดขาวด้วยวิธีการติดป้าย (Label) ในแต่ละจุดภาพ (Pixel) รอบบริเวณนิวเคลียส จนเกิดการแบ่งขอบอย่างชัดเจน กระบวนการดังกล่าวเป็นวิธีการที่ง่าย ให้ประสิทธิภาพดีกับเซลล์ที่มีรูปร่างกลม และมีการแบ่งขอบอย่างชัดเจน หากเป็นเซลล์ที่รูปร่างแตกต่างจากวงกลม เช่น วงรี และเซลล์ที่ไซโทพลาซึมแตก จะไม่สามารถจำแนกได้

Kan Jiang et al. [24] บทความวิจัยได้นำเสนอวิธีการแบ่งส่วนเซลล์เม็ดเลือดขาวด้วยตัวกรองมิติขนาดและระยะทาง (Scale-space filtering) และการจัดกลุ่มจุดผกผัน (Watershed clustering) เพื่อใช้ในการแก้ปัญหาการจำแนกเซลล์ที่มีความแตกต่างกันของรูปร่าง สี ขอบ และตำแหน่ง โดยจะแยกส่วนการประมวลผลออกเป็น 2 ส่วน ส่วนแรกแยกกลุ่มเซลล์เม็ดเลือดขาวออก

จากกลุ่มอื่น จากนั้นจึงปรับคุณสมบัติภาพด้วยตัวกรองมิติขนาดและระยะทาง เพื่อดึงคุณลักษณะของนิวเคลียสออกจากภาพ ส่วนของไซโทพลาซึมจะถูกแบ่งด้วยการจัดกลุ่มจุดผกผัน และแสดงออกโดยใช้ฮิสโตแกรมปริภูมิสามมิติแบบ HSV (Hue-Saturation-Value histogram : 3-D HSV histogram) จนสามารถแยกลักษณะรูปร่างของเซลล์ออกมาได้ (Morphological operations)

Fang Yi et al. [25] บทความวิจัยเริ่มนำโครงข่ายประสาทมาใช้ในการตัดแยกเซลล์เม็ดเลือดขาว โดยมีการปรับปรุงขั้นตอนวิธี (Algorithm) และรูปแบบการสุ่มเลือก (Uniform sampling) เพื่อลดจำนวนข้อมูลที่ใช้ในการฝึกโมเดลด้วยการหาค่าที่เหมาะสมแบบกลุ่มอนุภาค (Particle swarm optimization : PSO) ซึ่งเป็นวิธีการคำนวณอย่างชาญฉลาด (Computational intelligence) ที่ถูกนำมาใช้ในการฝึกโมเดลจนส่งผลให้ระยะเวลาในการฝึกโมเดล และระยะเวลาการทำงานของโครงข่ายประสาทลดลง ในขณะที่ความแม่นยำใกล้เคียงกัน

J. Wu et al. [26] เนื่องจากการแบ่งส่วนเซลล์ของเม็ดเลือดขาวด้วยวิธีการจำแนกจากขอบนั้น ยังมีความไม่สมบูรณ์เมื่อเส้นขอบติดกันไม่สมบูรณ์ เช่น การใช้การจัดกลุ่มจุดผกผัน (Watershed clustering) ในการแยกส่วนของเซลล์ จึงนำวิธีการ Otsu's และฮิสโตแกรมปริภูมิสามมิติแบบ HIS (Hue-Saturation-Intensity histogram : 3-D HSI histogram) มาแบ่งช่วงสี HIS ในปริภูมิสามมิติ พิกัดทรงกระบอก ซึ่งประกอบด้วย Green, Yellow, Red, Magenta, Blue, and Cyan พบว่าในองค์ประกอบ Hue จะแยกเซลล์เม็ดเลือดขาวออกมาได้ และ Saturation สามารถแยกนิวเคลียสออกจากเซลล์ได้ และเมื่อใช้การแบ่งขอบด้วยวิธีการ Otsu's จึงทำให้ผลลัพธ์การแบ่งส่วนได้ชัดเจน ซึ่งวิธีการนี้เหมาะสมเฉพาะการสกัดคุณลักษณะรูปร่างนิวเคลียสเพียงอย่างเดียว

L. B. Dorini et al. [27] จากปัญหาความซับซ้อนของการแบ่งส่วนภาพของเซลล์ที่เกิดจากความไม่แน่นอนของภาพที่ถ่ายจากกล้องจุลทรรศน์ จึงมีการนำเสนอวิธีการการแปลงกลุ่มจุดผกผัน (Watershed transform) เพื่อใช้สำหรับแยกส่วนภาพของนิวเคลียสและไซโทพลาซึมของเซลล์เม็ดเลือดขาว และต้องใช้ตัวดำเนินการสลับมิติขนาดกับระยะทาง (Scale-space toggle operator) เพื่อให้รูปร่างของเซลล์เป็นวงกลมปกติ วิธีการนี้จึงยังพบปัญหาในการจำแนกเซลล์ เมื่อนิวเคลียสและไซโทพลาซึมที่รูปร่างแตกต่างจากวงกลม และขนาดที่แปรผันตามชนิดของเซลล์เม็ดเลือดขาว และมีการวิจัยที่ใช้วิธีการคล้ายกันของ W. Gao et al. [28] ใช้วิธีการเปลี่ยนแปลงเวฟเล็ตซับซ้อน (Complex wavelet transform) ในการแยกบริเวณที่สนใจ (Region-of-interest : ROI) ของภาพเซลล์แยกในการปรับค่าไล่ระดับพื้นผิว (Textural gradient) และ Adaptive threshold จนนำไปสู่การทำแยกส่วนแบบจุดผกผัน

N. Theera-Umporn et al. [29] บทความวิจัยนี้จะเน้นในการจำแนกเซลล์ของภายในไขกระดูก (bone marrow) ที่มีความซับซ้อนของลักษณะรูปร่างของเซลล์มากกว่าเซลล์ที่อยู่ในเลือดโดยตรง อันเนื่องมาจากความหนาแน่นของเซลล์สูงกว่ามาก จึงทำให้การแบ่งส่วนของเซลล์ทำได้ยาก

ขึ้น คณะผู้วิจัยใช้วิธีการทางคณิตศาสตร์ด้วยการแยกรูปร่างของเซลล์จากนิวเคลียสเพียงอย่างเดียว ด้วยวิธีการ K-Fold cross validation คือการแย่งข้อมูลออกเป็น K ส่วนเท่าๆ กัน เพื่อใช้ในการฝึก และทดสอบโมเดล ได้ความสามารถจำแนกชนิดของเซลล์ถึงร้อยละ 77 เมื่อกำหนด K=5 ให้กับ โครงข่ายประสาทรวมกับการจำแนกด้วยวิธี Naive Bayes classifiers

M. Sajjad et al. [30] ได้นำเสนอวิธีการใช้ K-means ในการสร้างขั้นตอนวิธีการจำแนก เซลล์ขนาดเล็ก ซึ่งความเร็วในการทำงานขึ้นกับค่า K กับความหนาแน่นของชนิดของเซลล์ (class intensity) โดยขั้นตอนแรกจะปรับภาพสีไปเป็นภาพสีเชิง HSI ก่อนใช้ค่า K-means ในการสกัด จำแนกเซลล์เม็ดเลือดขาวออกจากภาพรวมกับใช้ซัพพอร์ตเวกเตอร์แมชชีน (Support vector machine : SVM) ความแตกต่างที่เพิ่มขึ้นจากระบบที่เคยมีก่อนหน้านี้ คือการนำไปใช้งานบนระบบ สมาร์ทโฟน (Smartphone) แต่ยังไม่จัดเป็นการจำแนกด้วยโครงข่ายประสาท

R. Agrawal et al. [31] บทความวิจัยนี้ได้นำเสนอขั้นตอนวิธีการตรวจหาเซลล์มะเร็งเม็ด เลือดขาวชนิด Acute myeloid leukemia (AML), Acute lymphoblastic leukemia (ALL) และ Myeloma ที่มีลักษณะรูปร่างของเซลล์ที่แตกต่างจากเซลล์ปกติทั่วไป คณะผู้วิจัยได้นำภาพจากกล้อง จุลทรรศน์มาแบ่งส่วนภาพด้วยระบบสัญญาณสี YCbCr แสดงภาพโดยใช้ความสว่างและความต่างของ สี ซึ่ง Y คือความสว่าง (luminance) Cb คือสีน้ำเงินที่ถูกตัดความสว่างออกไป และ Cr คือสีแดงที่ถูก ตัดความสว่างออกไป ก่อนนำภาพไปเข้ากระบวนการต่าง ๆ ทั้งการแจกแจงแบบเกาสเซียน (Gaussian distribution) แยกส่วนของภาพด้วยวิธีการ Otsu's และการปรับค่า K-means จน สุดท้ายใช้ Gray level co-occurrence matrix (GLCM) สำหรับดึงคุณลักษณะของภาพเซลล์ สำหรับการจำแนกด้วยโครงข่ายประสาทแบบคอนโวลูชัน (Convolutional neural network : CNN) ให้ความแม่นยำสูงถึงร้อยละ 97.3

จะเห็นได้ว่างานวิจัยที่มีมาก่อนหน้านี้ได้นำเสนอถึงกลวิธีในการจำแนกเซลล์ขนาดเล็ก เช่น เซลล์เม็ดเลือดขาว ด้วยวิธีการประมวลผลภาพ (Image processing) และการใช้โครงข่ายประสาท (Neural network) มาช่วยให้วิธีการระบุชนิดของเซลล์ทำได้รวดเร็วและแม่นยำ ซึ่งในวิทยานิพนธ์นี้ จะนำเสนอวิธีตรวจจับเซลล์ขนาดเล็กด้วยโครงข่ายประสาทที่ไม่จำกัดเฉพาะเซลล์เม็ดเลือดขาว หาก ยังตรวจจับเซลล์เม็ดเลือดแดง และเกล็ดเลือด ผ่านชุดอุปกรณ์สำหรับสวมเข้ากับท่อเลนส์ใกล้ตา (Eyepiece lens tube) ของกล้องจุลทรรศน์ได้โดยตรง

## บทที่ 3 ระเบียบวิธีวิจัย

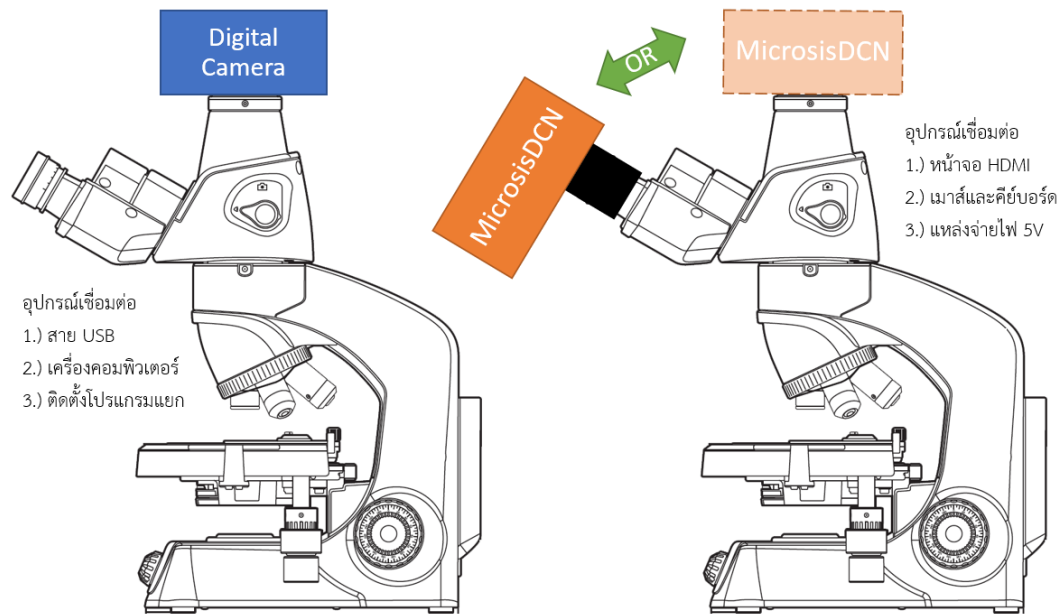
### 3.1 การพัฒนาระบบชุดกล้องสำหรับโครงการวิจัย

การศึกษาเซลล์ขนาดเล็ก อุปกรณ์ที่สำคัญอย่างยิ่งคือกล้องจุลทรรศน์ (Microscope) ซึ่งเป็นเครื่องมือที่ขยายภาพของวัตถุขนาดเล็ก ในที่นี้จะให้ความสำคัญกับกล้องจุลทรรศน์เชิงซ้อน (Compound microscope) ซึ่งเป็นกล้องที่ใช้กันอย่างแพร่หลาย ตัวอย่างเช่น งานทางด้าน การตรวจหาความสมบูรณ์ของเม็ดเลือดนั้น จะต้องมองด้วยตาผ่านกล้องจุลทรรศน์นานนับชั่วโมง ติดต่อกัน จะส่งผลให้เกิดอาการล้าสายตาจนก่อให้เกิดความคลาดเคลื่อนได้ จะมีวิธีการที่มีการนำตัวอย่างสารคัดหลั่งไปตรวจเพื่อเป็นข้อมูลประกอบการรักษา เช่น การตรวจเลือด ปัสสาวะ อุจจาระ เสมหะ เป็นต้น หนึ่งในวิธีการตรวจหาการติดเชื้อสามารถดูได้จากปริมาณเม็ดเลือด ภายในห้องปฏิบัติการทางเทคนิคการแพทย์จะให้ความสำคัญในการวิเคราะห์จำนวนเซลล์ รวมถึงปริมาณ ความเข้มข้นของสารต่าง ๆ เรียกว่าการตรวจความสมบูรณ์ของเม็ดเลือด (Complete blood count : CBC) ผ่านกล้องจุลทรรศน์ ประกอบไปด้วย จำนวนเม็ดเลือดแดง (Red blood cell : RBC) เม็ดเลือดขาว (White blood cell : WBC) และเกล็ดเลือด (Platelets)

ผู้วิจัยจึงมีแนวคิดในการสร้างอุปกรณ์ที่ช่วยลดระยะเวลาในการนับคัดแยกที่มีความแม่นยำ ซึ่งส่งผลให้การวินิจฉัยโรคทำได้รวดเร็วมากขึ้น เรียกระบบโดยภาพรวมว่าชุดกล้องอัจฉริยะ “ไมโครซิสดีซีเอ็น” สำหรับกล้องจุลทรรศน์ (“MicrosisDCN” intelligent camera for microscope : Microbes Diagnosis with Deep Convolutional Neural Network) สำหรับแยกชนิดและนับจำนวนเซลล์ขนาดเล็กด้วยโครงข่ายประสาท (Neural network) เป็นชุดอุปกรณ์สำหรับสวมชุดกล้องเข้ากับท่อเลนส์ใกล้ตา (Eyepiece lens tube) ของกล้องจุลทรรศน์ทั่วไป เพื่อนำไปเข้ากระบวนการประมวลผลภาพ (Image processing) ด้วยโครงข่ายประสาทที่ถูกฝึกให้จำแนกภาพเซลล์ขนาดเล็ก



ภาพที่ 33 กล้องจุลทรรศน์ตาเดียว (Monocular) ตาคู่ (Binocular) และสามตา (Trinocular)



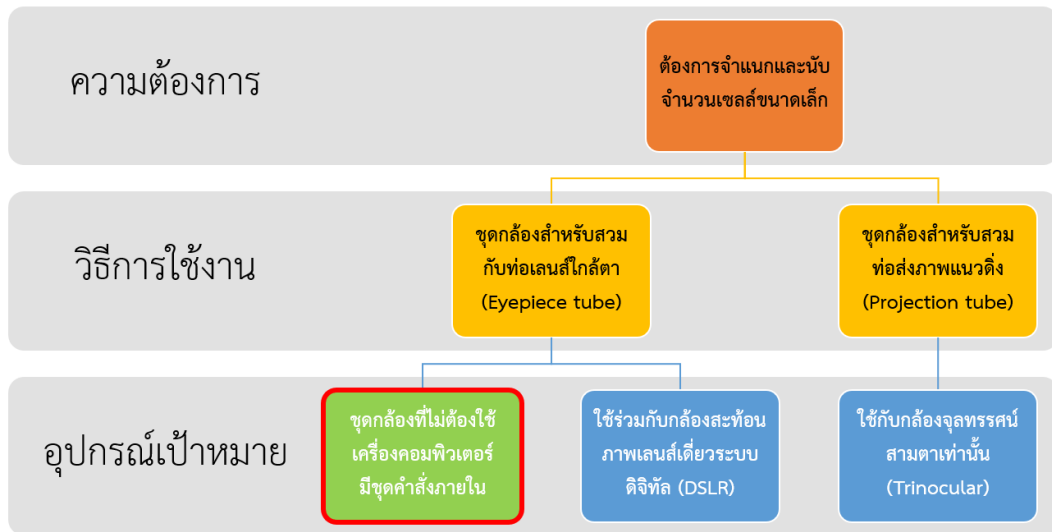
ชุดกล้องดิจิทัลถ่ายทอดสัญญาณภาพสำหรับ Trinocular

ชุดกล้อง MicrosisDCN ที่พัฒนาขึ้นสำหรับ Binocular

### ภาพที่ 34 การพัฒนาระบบชุดกล้องสำหรับโครงงานวิจัยสำหรับกล้องจุลทรรศน์แบบ Binocular

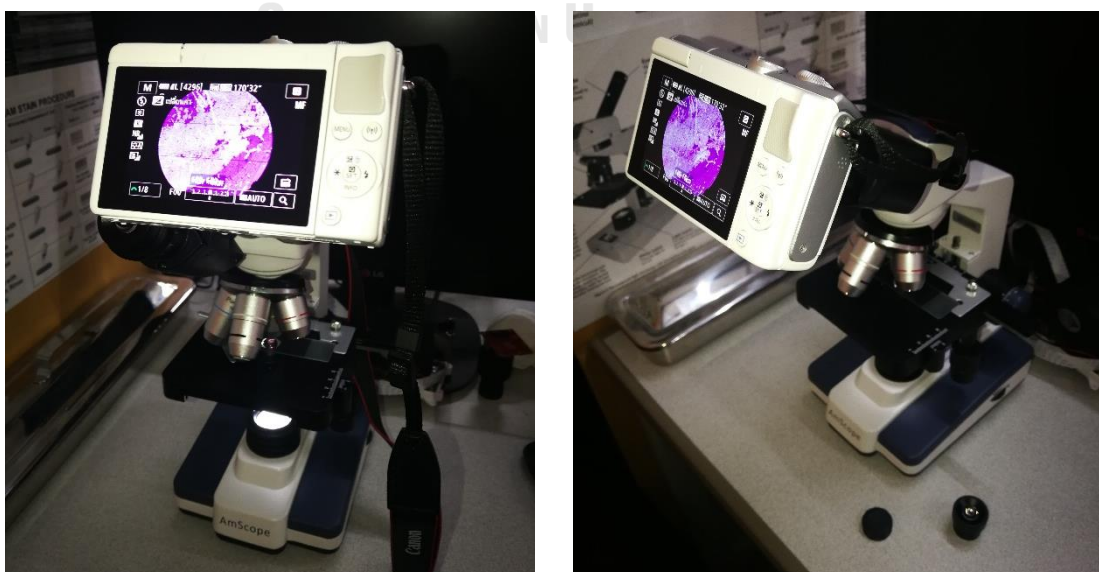
กล้องจุลทรรศน์เชิงซ้อน (Compound microscope) แยกออกเป็น 3 แบบ ได้แก่ กล้องจุลทรรศน์ตาเดียว (Monocular microscope) กล้องจุลทรรศน์ตาคู่ (Binocular) และกล้องจุลทรรศน์สามตา (Trinocular microscope) ตามภาพที่ 34 ซึ่งกล้องจุลทรรศน์ตาเดียวและตาคู่นั้นมีใช้อย่างแพร่หลายในห้องปฏิบัติการระดับสถานศึกษา จนถึงระดับห้องปฏิบัติการวิจัยชั้นสูง ผู้ใช้สามารถส่องดูวัตถุตัวอย่างผ่านเลนส์ใกล้ตาได้โดยตรง สำหรับกล้องจุลทรรศน์สามตาจะมีใช้งานในห้องปฏิบัติการเฉพาะทาง เนื่องด้วยราคาที่สูงกว่ากล้องจุลทรรศน์แบบตาเดียวและตาคู่ ความซับซ้อนของชุดเลนส์และปริซึมในการส่องผ่านแสง โดยปกติแล้วแสงที่ผ่านจากเลนส์ใกล้วัตถุจะเดินทางผ่านชุดเลนส์และปริซึมหักเหแสงเข้าสู่กระบอกเลนส์ตา (Eyepiece tube) ก่อนตกกระทบเข้ากับเลนส์ใกล้ตา (Eyepiece lens) เพื่อขยายภาพให้ตามนุษย์สามารถมองเห็นได้

กล้องจุลทรรศน์สามตานี้จะมีชุดเลนส์และปริซึมที่แตกต่างออกไป เพราะกล้องชนิดนี้นอกจากจะหักเหแสงเข้าสู่กระบอกเลนส์ตาแล้ว ยังสามารถหักเหแสงเข้าสู่ท่อส่งภาพแนวตั้ง (Projection tube) ส่งผ่านแสงไปตกกระทบบนตัวรับรูปภาพ (Image sensor) ของกล้องถ่ายภาพดิจิทัล (Digital camera) สำหรับถ่ายทอดสัญญาณไปแสดงผลบนจอภาพ หรือเครื่องคอมพิวเตอร์ ที่ต้องใช้โปรแกรมเฉพาะทางในการเชื่อมต่ออุปกรณ์ และจำกัดเฉพาะรุ่นที่รองรับเท่านั้น ผู้ใช้ไม่จำเป็นต้องส่องกล้องจากเลนส์ใกล้ตา และยังสามารถนำภาพที่ได้ไปใช้งานทางการประมวลผลภาพ (Image processing) หรือการสังเกตชนิดของวัตถุแบบทันทีทันใด (Real time observation)



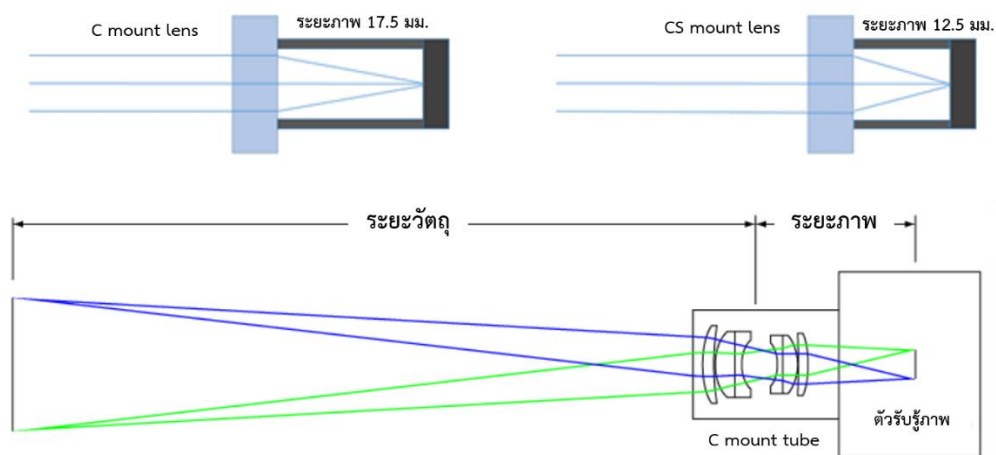
ภาพที่ 35 ผังเปรียบเทียบคุณลักษณะของอุปกรณ์ที่แตกต่างกัน

จากเหตุผลที่กล่าวมานั้น ผู้วิจัยมีแนวคิดที่จะพัฒนาชุดกล้องที่สามารถเข้ากับท่อเลนส์ใกล้ตาของกล้องจุลทรรศน์ตาเดี่ยวและตาคู่ทั่วไปได้ วิธีการได้มาซึ่งภาพจากกล้องจุลทรรศน์ในระบบดั้งเดิมจะใช้กล้องดิจิทัลเชื่อมต่อกับกล้องจุลทรรศน์สามตาผ่านทางท่อส่งภาพแนวตั้ง และยังสามารถใช้ตัวแปลงเลนส์ขนาด 23.2 มิลลิเมตร ชนิด C mount เป็นเกลียวชนิดเดียวกันกับชุดถ่ายทอดสัญญาณภาพจากกล้องจุลทรรศน์สามตา ต่อเข้ากับกล้องดิจิทัลในการถ่ายภาพได้ ข้อจำกัดของการใช้งานลักษณะนี้ เนื่องจากไม่ได้ใช้ชุดเลนส์ปกติของกล้องดิจิทัล จึงไม่สามารถถ่ายภาพได้โดยตรง จะต้องใช้โปรแกรมเชื่อมต่อกับกล้องดิจิทัลจากผู้ผลิตควบคุมการทำงาน มีเฉพาะกล้องสะท้อนภาพเลนส์เดี่ยวระบบดิจิทัล (Digital single lens reflex : DSLR) ที่สามารถใช้งานได้



ภาพที่ 36 การบันทึกภาพด้วยกล้องดิจิทัลผ่านตัวแปลงเลนส์ขนาด 23.2 มิลลิเมตร

เลนส์ที่มีเกลียวชนิด C mount เป็นเลนส์ที่พบได้มากในกล้องถ่ายภาพยนตร์ 16 มิลลิเมตร กล้องวงจรปิด (CCTV) และชุดถ่ายภาพต้อสัญญาณภาพจากกล้องจุลทรรศน์ เกลียวชนิด C mount ประกอบด้วยเกลียวตัวผู้และตัวเมียที่เข้าคู่กัน มีเส้นผ่านศูนย์กลาง 25.4 มิลลิเมตร หรือ 1 นิ้ว ขนาดเกลียว 32 เกลียวต่อนิ้ว ตามมาตรฐาน ANSI B1.1 สำหรับ Unified screw threads ในหมวด 1-32-UN-2A แยกเป็น 2 แบบ คือ C mount มีระยะภาพถึงผิวหน้าตัวรับรูภาพ 17.5 มิลลิเมตร และ CS mount มีระยะภาพถึงผิวหน้าตัวรับรูภาพ 12.5 มิลลิเมตร



ภาพที่ 37 แผนภาพแสดงระยะภาพของเลนส์แบบ C mount และ CS mount



ภาพที่ 38 การทดสอบตัวรับรูภาพผ่านเครื่องคอมพิวเตอร์ ณ ห้องปฏิบัติการ ภาควิชาจุลชีววิทยา คณะสัตวแพทยศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

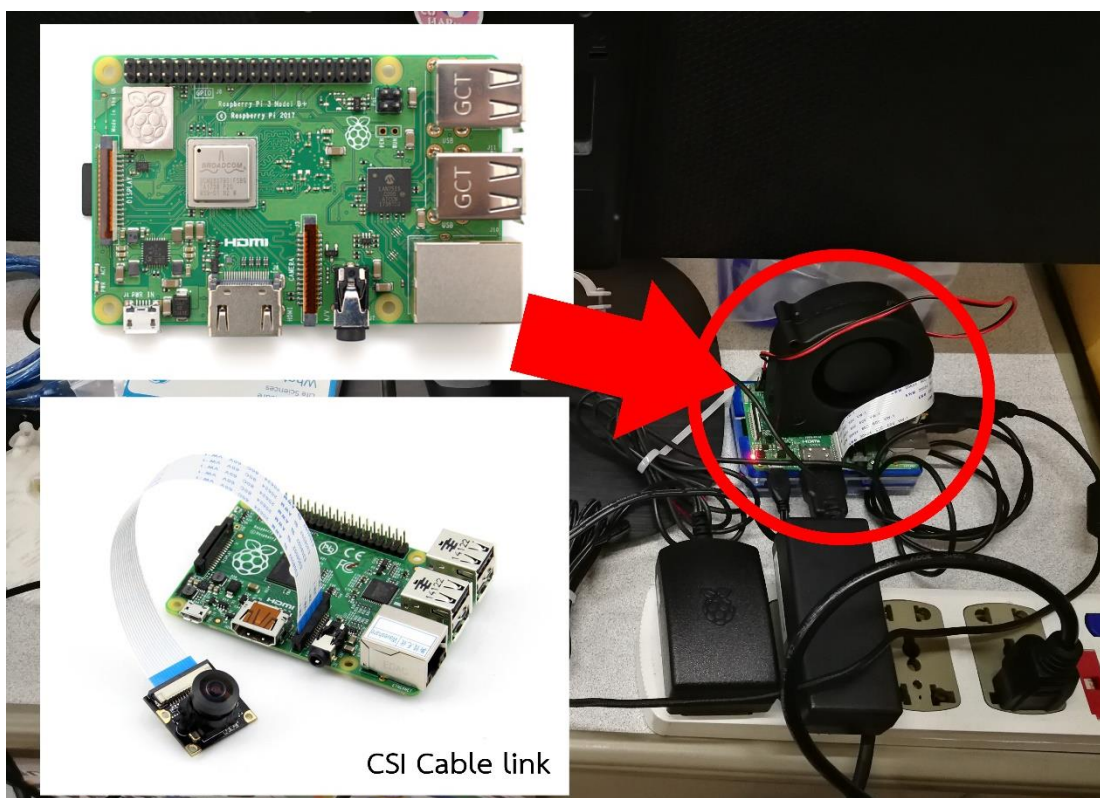
วิธีการถัดมา คือ การใช้ตัวรับรู้ภาพ (Image sensor) ของกล้องถ่ายภาพดิจิทัล (Digital camera) สำหรับถ่ายทอดสัญญาณไปแสดงผลบนโทรทัศน์ ผ่านสายสัญญาณภาพแบบอนาล็อก RCA โดยวิธีนี้จะทำได้เพียงถ่ายทอดภาพสีเคลื่อนไหวเพียงอย่างเดียว และมีการใช้กล้องดิจิทัลถ่ายทอดสัญญาณภาพผ่านสาย USB ไปยังเครื่องคอมพิวเตอร์ ที่มีโปรแกรมควบคุมการทำงานด้านการถ่ายภาพและประมวลผลภาพพร้อมด้วย เช่น DinoCapture, MICAM, OpenMV เป็นต้น ในภาพที่ 37 ผู้วิจัยได้นำตัวรับรู้ภาพเชื่อมต่อกับกล้องจุลทรรศน์สามตา จะเห็นว่านอกจากต้องใช้พื้นที่สำหรับวางเครื่องคอมพิวเตอร์ รวมถึงผู้ใช้จะต้องมีความเข้าใจกับโปรแกรมด้านการถ่ายภาพ และต้องจำแนกและนับจำนวนด้วยตาผ่านหน้าจอ ซึ่งมีวิธีการที่ไม่แตกต่างจากการส่องกล้องด้วยตาเปล่า

แนวคิดของผู้วิจัยมองว่า จะทำอย่างไรถึงจะรวมระบบการถ่ายภาพและระบบควบคุมการทำงานไว้ภายในชุดกล้องเพียงชุดเดียว จึงนำระบบสมองกลฝังตัว (Embedded system) ที่สามารถติดตั้งระบบปฏิบัติการ (Operating system) ที่เหมาะสมกับการพัฒนารองรับการทำงานผ่านโครงข่ายประสาท คือ Raspberry Pi เป็นบอร์ดคอมพิวเตอร์ขนาดเล็กที่ประกอบด้วย CPU GPU และ RAM สามารถเชื่อมต่อหน้าจอสแสดงผล เมมส์และคีย์บอร์ดได้ สามารถนำมาประยุกต์ใช้พัฒนาโครงงานทางอิเล็กทรอนิกส์ การเขียนโปรแกรม การใช้งานการเรียนรู้ของเครื่องและโครงข่ายประสาท รองรับการทำงานระบบปฏิบัติการลินุกซ์ (Linux operating system)



ภาพที่ 39 โมดูลกล้อง Raspberry Pi ถ่ายทอดสัญญาณผ่าน CSI cable สวมกับท่อเลนส์ใกล้ตา



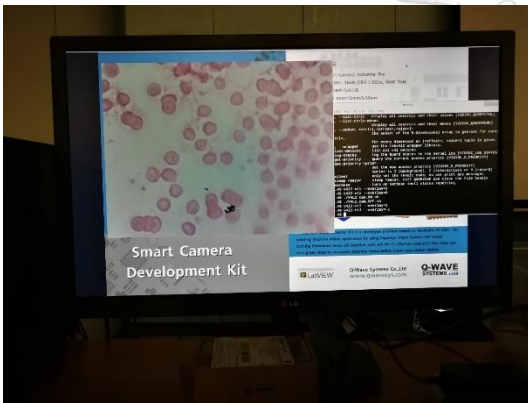


ภาพที่ 40 บอร์ดสมองกลฝังตัว Raspberry Pi 3 B+ และวิธีการเชื่อมต่อกล้องผ่าน CSI Cable

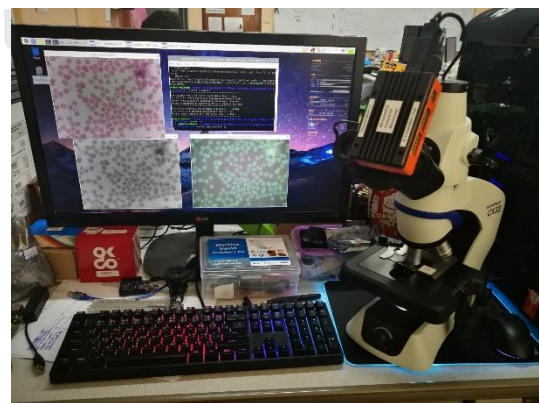
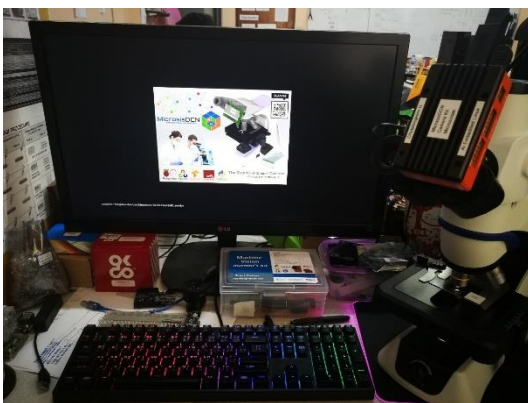
ในภาพที่ 39 จะเห็นการใช้ Raspberry Pi camera module ตัวรับรู้ภาพ OV5647 ความละเอียด 5 ล้านจุดภาพ (Megapixel) ขนาดความกว้าง CCD  $\frac{1}{4}$  นิ้ว ชนิดเกลียว 12 มิลลิเมตร (M12) สวมเข้ากับเลนส์กำลังขยาย 0.5X แบบ C mount เข้ากับตัวแปลง M12 to C mount สวมเข้ากับท่อเลนส์ใกล้ตาของกล้องจุลทรรศน์ โยงสายด้วยสายแพสำหรับการเชื่อมต่อแบบ CSI (Camera serial interface) เข้ากับ Raspberry Pi 3 B+ แสดงในภาพที่ 40 ซึ่งบอร์ดใช้ CPU Broadcom BCM2837B0 Quad-Core ARM Cortex-A53 (ARMv8) 1.4GHz มี RAM ชนิด LPDDR2 SDRAM 1GB ภาครับสัญญาณ Wifi และ Bluetooth ใช้ Cypress CYW43455 Dual-Band Wifi 2.4GHz / 5GHz Bluetooth 4.2 BLE มี USB 4 Port Gigabit LAN 1 Port แสดงผลผ่าน HDMI ออกทางหน้าจอได้ในตัวเดียว [32] จะเห็นได้ว่าขนาดของอุปกรณ์กะทัดรัดมากกว่าเครื่องคอมพิวเตอร์ แต่ยังพบความไม่สะดวกในการติดตั้ง และโยงสาย CSI ระหว่างโมดูลกล้องกับตัวบอร์ด Raspberry Pi จึงมีความจำเป็นต้องปรับปรุงให้สามารถรวมโมดูลกล้อง และบอร์ดเข้าไว้ด้วยกัน ผู้วิจัยจึงเลือกใช้ EagleEYE Smart Camera (EY-PRO-32) ที่พัฒนาโดย คิว เวฟ ซิสเต็มส์ จำกัด มาประยุกต์ และปรับแต่งระบบปฏิบัติการเฉพาะสำหรับใช้งาน



ภาพที่ 41 ชุดกล้อง EagleYE Smart Camera สำหรับงานด้าน Machine Vision และการเปลี่ยนชุดเลนส์ไปเป็น เลนส์สวมท่อส่งภาพแนวตั้ง และเลนส์กำลังขยาย 0.5X แบบ C mount



ภาพที่ 42 ระบบปฏิบัติการ Linux RT รองรับโปรแกรม LabVIEW และระบบปฏิบัติการ Raspbian ที่ถูกติดตั้งชุดคำสั่งที่ใช้ในโครงการวิจัย จุฬาลงกรณ์มหาวิทยาลัย



ภาพที่ 43 การปรับแต่งระบบปฏิบัติการ Raspbian พร้อมติดตั้งชุดคำสั่งโครงข่ายประสาทใหม่

ในภาพที่ 41 เป็นการนำ EagleEYE Smart Camera (EY-PRO-32) ซึ่งกล้องเครื่องจักรวิทัศน์ (Machine vision) ทางอุตสาหกรรมที่ติดตั้งระบบปฏิบัติการ Linux RT (Realtime) ภายในมีชุดสมองกลฝังตัว Raspberry Pi Computer Module 3+ พร้อมชุดแผงวงจรรวมทั้งภาคจ่ายไฟ และตัวรับรู้ภาพ OV5647 ความละเอียด 5 ล้านจุดภาพ ผู้วิจัยได้เลือกชุดกล้องดังกล่าวมาปรับปรุงชุดเลนส์ สำหรับสวมเข้ากับเลนส์กำลังขยาย 0.5X แบบ C mount ขนาด 23.2 30 หรือ 30.5 มิลลิเมตร เนื่องจากชุดกล้องนี้ถูกออกแบบมา ได้รวมวงจรส่วนต่าง ๆ ที่จำเป็นต่อการวิจัยไว้ครบถ้วน และปรับเปลี่ยนระบบปฏิบัติการให้เหมาะสมต่อการใช้งานกับกล้องจุลทรรศน์ ทำให้อุปกรณ์มีรูปร่างเหมาะสมต่อการใช้งาน และเพิ่มเสถียรภาพจากการใช้ SD-Card 32GB เป็น eMMC 32GB

ในภาพที่ 41 ผู้วิจัยได้นำระบบปฏิบัติการ Raspbian รุ่น Raspberry Pi OS (32-bit) with desktop and recommended software มาติดตั้งชุดคำสั่งที่จำเป็น เช่น OpenCV, Tensorflow, PiCamera และอื่น ๆ ที่จะทำให้ระบบโครงข่ายประสาทสามารถทำงานได้บนอุปกรณ์ดังกล่าว และจัดทำเป็น Image files ต้นฉบับที่ชื่อว่า MicrosisDCN ver 1.0e สามารถนำไปใช้บน Raspberry Pi Model B 3, Model B 3+, Raspberry Pi 4 ที่มี RAM ตั้งแต่ 1 GB เนื่องจากชุดกล้องสามารถเชื่อมต่อ Internet ผ่าน LAN Port ได้ ผู้พัฒนาจึงปรับเปลี่ยนชื่อเชื่อมต่อ USB ให้สามารถใส่ WiFi Adapter สำหรับการเชื่อมต่อสัญญาณไร้สาย เช่น เครือข่ายอินเทอร์เน็ตมหาวิทยาลัย WPA2-Enterprise ที่จะต้องแก้ไข wpa\_supplicant ให้เชื่อมต่อแบบรับ Certification จากรบบได้ และการควบคุมทำได้โดยการใช้ Wireless mouse & keyboard ในรุ่นทั่วไปผ่านการเชื่อมต่อ USB



ภาพที่ 44 การเปรียบเทียบชุดกล้องดั้งเดิม และชุดกล้องที่ถูกพัฒนาขึ้นมาใหม่

ตารางที่ 8 เปรียบเทียบคุณสมบัติอุปกรณ์ที่ใช้ในการวิจัย

	 <p>CSI Cable link</p>	
ชนิดอุปกรณ์	ชุดกล้องต่อตัวรับรู้ภาพผ่าน CSI ไป Raspberry Pi 3 B+ Board	ชุดกล้อง EagleEye สวมเลนส์ 0.5X Raspberry Pi Computer Module 3+
ระบบปฏิบัติการ	Raspberry Pi OS (Raspbian)	Linux RT (เดิม) / Raspbian (ใหม่)
ตัวรับรู้ภาพ	Image sensor OV5647 with CSI	Image sensor OV5647 onboard
ชนิดเกลียวเลนส์	C mount / CS mount (ปรับแต่ง)	C mount / CS mount (ในตัว)
ตัวประมวลผล	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.4GHz	Broadcom BCM2837B0, Cortex-A53 64-bit SoC @ 1.2GHz
ชนิดแรม	1GB LPDDR2 SDRAM	1GB LPDDR2 SDRAM
ความจุ	ขึ้นอยู่กับขนาดของ Micro-SD Card	32GB eMMC Flash memory
นับเวลา	ไม่มี / ใช้เวลา NTP จากอินเทอร์เน็ต	Real-Time clock onboard (RTC)
Watchdog	ไม่มี	มี
พอร์ตเชื่อมต่อ	<ul style="list-style-type: none"> <li>- HDMI socket</li> <li>- 3.5mm analogue audio jack</li> <li>- 4 ports USB 2.0</li> <li>- Ethernet (Lan port)</li> <li>- Camera Serial Interface (CSI)</li> <li>- Display Serial Interface (DSI)</li> </ul>	<ul style="list-style-type: none"> <li>- mini-HDMI socket</li> <li>- Single ports USB 2.0</li> <li>- Ethernet (Lan port)</li> </ul>
ไฟเลี้ยง	5V 2.1A via micro USB	12-24V via DC Adaptor
วงจรป้องกัน	Diode / IC Regulator	<ul style="list-style-type: none"> <li>- Polarity protection</li> <li>- Short circuit</li> <li>- Over voltage and current</li> <li>- Thermal shutdown</li> </ul>
ขนาดวงจร	82 x 56 x 19.5 มิลลิเมตร	67.6 x 31.1 x 3.7 มิลลิเมตร

### 3.2 การพัฒนาชุดเลนส์และหน่วยกำลังขยายภาพ

#### 3.2.1 การคำนวณขนาดพื้นที่ขอบเขตการมองเห็นของกล้องจุลทรรศน์

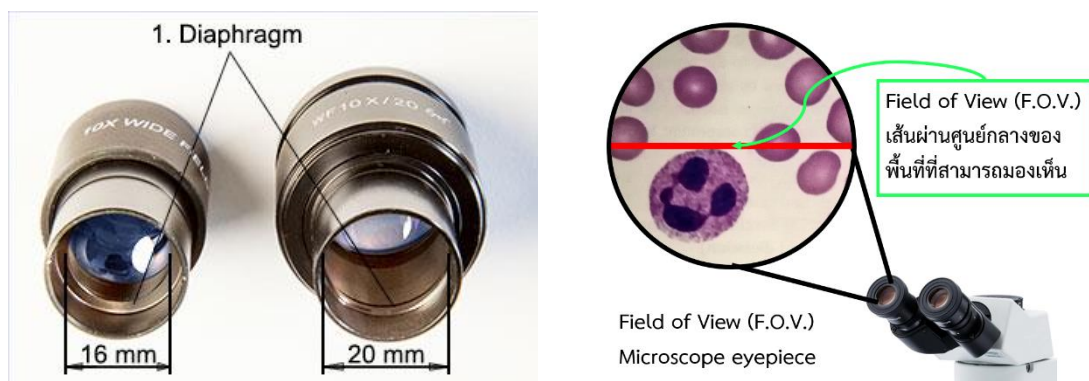
กล้องจุลทรรศน์เชิงซ้อน (Compound microscope) ยี่ห้อ Olympus CX33 Microscope แบบ Trinocular ที่มีกำลังขยายตั้งแต่ 40 ถึง 1000 เท่าของขนาดวัตถุ ซึ่งประกอบด้วยเลนส์ใกล้ตา (Eyepiece lens) กำลังขยาย 10X ขอบเขตการมองเห็น (Field of View : FOV) 20 และมีเลนส์ใกล้วัตถุ (Objective lens) กำลังขยาย 4X, 10X, 40X และ 100X (immersion oil) ในการถ่ายภาพและทดลองการจำแนกด้วยโครงข่ายประสาท จะกำหนดกำลังขยายของเลนส์ใกล้วัตถุไว้ที่ 40X เป็นการเลือกใช้กำลังขยายรวม 400 เท่า ที่ให้รายละเอียดและความคมชัดของเซลล์ขนาดเล็ก ได้แก่ เม็ดเลือดแดง เม็ดเลือดขาว และเกล็ดเลือดได้ชัดเจน ทั้งการส่องผ่านเลนส์ใกล้ตา และการถ่ายภาพผ่านจตุรรับรู้อภาพของชุดกล้อง โดยไม่จำเป็นต้องใช้กำลังขยายรวม 1000 เท่า จากเลนส์ใกล้วัตถุ 100X ที่จะต้องใส่ Immersion oil สำหรับปรับค่าดัชนีหักเหของแสง จากสมการกำลังการขยายของเลนส์ (Transverse magnification) สามารถคิดกำลังการขยายรวมได้ว่า

$$m_T = m_{obj} \cdot m_{eye} \quad (54)$$

$$m_{cam} = m_{obj} \cdot m_{adapter} \quad (55)$$

เมื่อ  $m_T$  เป็นผลการคูณระหว่างกำลังขยายของเลนส์ใกล้วัตถุ  $m_{obj}$  กับกำลังขยายของเลนส์ใกล้ตา  $m_{eye}$  ได้จากสมการ แต่ชุดกล้อง MicrosisDCN จะใช้เลนส์กำลังขยาย 0.5X แบบ C mount ส่งภาพไปยังตัวรับรู้อภาพเปรียบเสมือนเลนส์ใกล้ตา จึงคำนวณกำลังการขยายชุดกล้อง  $m_{cam}$  จากผลการคูณระหว่างกำลังขยายของเลนส์ใกล้วัตถุ  $m_{obj}$  กับกำลังขยายของเลนส์สวมท่อเลนส์ใกล้ตา  $m_{adaptor}$  มีค่าเป็น 0.5 ทำให้ได้ ค่ากำลังการขยายของเลนส์กล้องจุลทรรศน์  $m_T = 40 \times 10 = 400$  เท่า และค่ากำลังการขยายชุดกล้อง  $m_{cam} = 40 \times 0.5 = 20$  เท่า

การคำนวณขนาดภาพจริงที่ปรากฏบนตัวรับรู้อภาพ มีหลักการเดียวกับการคำนวณขนาดภาพจริงของกล้องจุลทรรศน์ที่ปรากฏในดวงตาผ่านการขยายของเลนส์ใกล้ตา ซึ่งหาได้จาก ขอบเขตการมองเห็น (Field of View : F.O.V.) คือ พื้นที่ที่สามารถมองเห็นวัตถุภายในขอบเขตที่มองเห็นของกล้องจุลทรรศน์ จะขึ้นอยู่กับขนาดของไดอะแฟรมเลนส์ (Diaphragm of eyepiece lens) หรือเลขขอบเขต (Field Number : FN) โดยมักจะระบุไว้บนตัวเลนส์ เช่น 10X/20 หมายถึง เลนส์ใกล้ตามีกำลังขยาย 10 เท่า และมีเส้นผ่านศูนย์กลางไดอะแฟรมเลนส์ 20 มิลลิเมตร



ภาพที่ 45 แสดงเส้นผ่านศูนย์กลางไดอะแฟรมเลนส์ (ซ้าย) และ Field of View (ขวา) [33]

จากคุณสมบัติของขนาดของไดอะแฟรมเลนส์ จึงทำให้สามารถคำนวณหาเส้นผ่านศูนย์กลางของขอบเขตการมองเห็น  $D_{FV}$  และขนาดพื้นที่ขอบเขตการมองเห็น  $A_{FV}$  ได้จากสมการ

$$D_{FV} = \frac{FN}{m_{obj} \cdot m_{tube}} \quad (56)$$

$$A_{FV} = \pi \left( \frac{D_{FV}}{2} \right)^2 \quad (57)$$

เมื่อ  $FN$  เป็นเลขขอบเขตของเลนส์ใกล้ตา  $m_{obj}$  เป็นกำลังขยายของเลนส์ใกล้วัตถุ  $m_{tube}$  เป็นค่าปัจจัยการขยายของท่อเลนส์ ในกรณีที่มีชุดเลนส์ปรับเปลี่ยนกำลังขยาย (Magnification changer) หากไม่มีให้แทนค่าเป็น 1

ตารางที่ 9 แสดงค่าเส้นผ่านศูนย์กลางของขอบเขตการมองเห็น

กำลังขยายเลนส์ใกล้วัตถุ	เลนส์ใกล้ตา FN18	เลนส์ใกล้ตา FN20	เลนส์ใกล้ตา FN22
2X	9.00 mm	10.00 mm	11.00 mm
4X	4.50 mm	5.00 mm	5.50 mm
10X	1.80 mm	2.00 mm	2.20 mm
20X	0.90 mm	1.00 mm	1.10 mm
40X	0.45 mm	0.50 mm	0.55 mm
50X	0.36 mm	0.40 mm	0.44 mm
60X	0.30 mm	0.33 mm	0.37 mm
100X	0.18 mm	0.20 mm	0.22 mm

ตารางที่ 10 แสดงค่าพื้นที่ของขอบเขตการมองเห็น

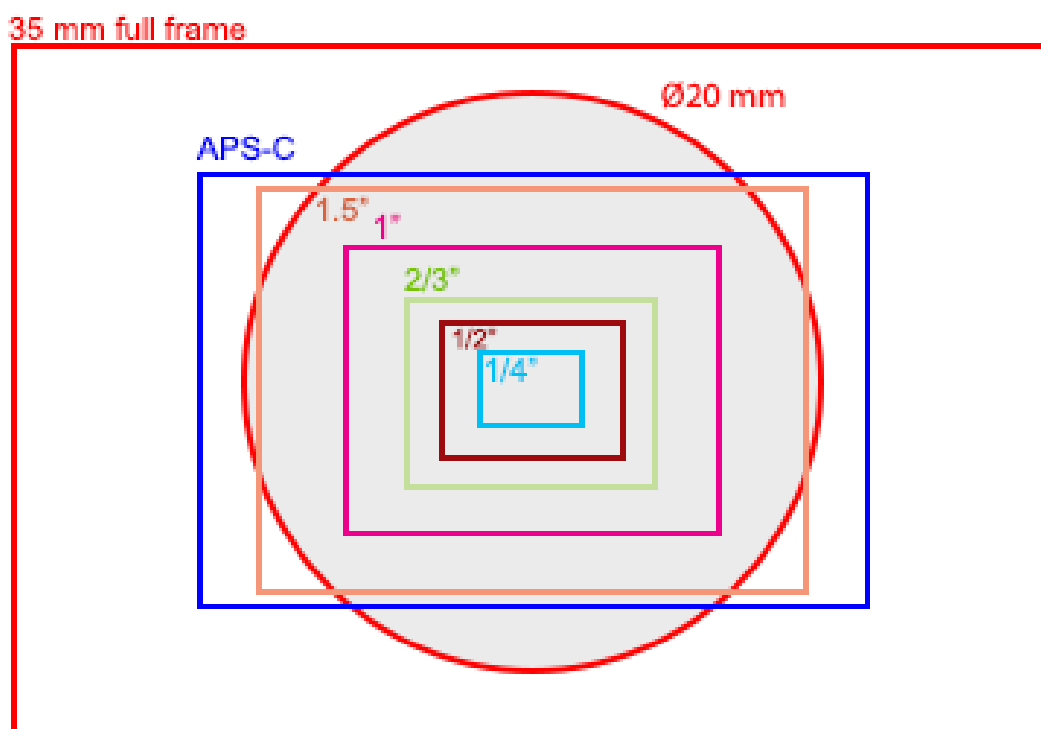
กำลังขยายเลนส์ใกล้วัตถุ	เลนส์ใกล้ตา FN18	เลนส์ใกล้ตา FN20	เลนส์ใกล้ตา FN22
2X	63.58500 mm <sup>2</sup>	78.50000 mm <sup>2</sup>	94.98500 mm <sup>2</sup>
4X	15.89625 mm <sup>2</sup>	19.62500 mm <sup>2</sup>	23.74625 mm <sup>2</sup>
10X	2.54340 mm <sup>2</sup>	3.14000 mm <sup>2</sup>	3.79940 mm <sup>2</sup>
20X	0.63585 mm <sup>2</sup>	0.78500 mm <sup>2</sup>	0.94985 mm <sup>2</sup>
40X	0.15896 mm <sup>2</sup>	0.19625 mm <sup>2</sup>	0.23746 mm <sup>2</sup>
50X	0.10174 mm <sup>2</sup>	0.12560 mm <sup>2</sup>	0.15198 mm <sup>2</sup>
60X	0.07065 mm <sup>2</sup>	0.08705 mm <sup>2</sup>	0.10573 mm <sup>2</sup>
100X	0.02543 mm <sup>2</sup>	0.03140 mm <sup>2</sup>	0.03799 mm <sup>2</sup>

### 3.2.2 การคำนวณขนาดพื้นที่ขอบเขตการมองเห็นของตัวรับรูภาพ

จากการหาเส้นผ่านศูนย์กลางของขอบเขตการมองเห็น และการหาขนาดพื้นที่ขอบเขตการมองเห็นตามกำลังขยายเลนส์ใกล้วัตถุ ที่เลนส์ใกล้วัตถุกำลังขยาย 40X และเลนส์ใกล้ตากำลังขยาย 10X มีเลขขอบเขตเป็น 20 จึงได้ค่าเส้นผ่านศูนย์กลางของขอบเขตการมองเห็นด้วยสมการ 56 เป็น 0.5 มิลลิเมตร หรือ 500 ไมโครเมตร และขนาดพื้นที่ขอบเขตการมองเห็นด้วยสมการ 57 เป็น 0.19625 ตารางมิลลิเมตร หรือ 196250 ตารางไมโครเมตร เป็นพื้นที่วงกลม

ในชุดกล้อง MicrosisDCN ตัวรับรูภาพ OV5647 ความละเอียด 5 ล้านจุดภาพ ชนิด CMOS (Complementary metal-oxide semiconductor) ขนาดตัวรับรูภาพ 1/4 นิ้ว มีหลักการทำงานเมื่อแสงเดินทางผ่านชุดเลนส์ตกกระทบบนตัวรับรูภาพเป็นเซลล์อิเล็กทรอนิกส์ไวแสงเรียกว่า โฟโตไซด์ (Photosites) ทำหน้าที่เปลี่ยนแสงความเข้มแสงสีแดง สีเขียว และสีน้ำเงิน (RGB) ไปเป็นสัญญาณทางไฟฟ้า โดยผ่านชุดวงจร A/D (Analog digital converter) เพื่อแปลงสัญญาณเชิงอุปมาน (Analog signal) ไปเป็นสัญญาณเชิงเลข (Digital signal) เพื่อนำไปประมวลผลบนชุดสมองกลฝังตัว Raspberry Pi Computer Module 3+ ซึ่งตัวรับรูภาพจะทำหน้าที่เป็นตัวรับภาพเสมือนการทำงานของเลนส์ใกล้ตา จากขนาดตัวรับรูภาพ OV5647 1/4 นิ้ว มีความกว้าง 3.6 มิลลิเมตร ความสูง 2.7 มิลลิเมตร เป็นพื้นที่สี่เหลี่ยมผืนผ้า จึงสามารถหาพื้นที่ตัวรับรูภาพ  $A_{sen}$  ได้จากการคูณระหว่างความกว้างและความสูงมีค่าเป็น 9.72 ตารางมิลลิเมตร

$$A_{sen} = W_{sen} \cdot H_{sen} \quad (58)$$



ภาพที่ 46 การเปรียบเทียบพื้นที่ขอบเขตการมองเห็น FN20 กับขนาดพื้นที่ตัวรับรู้อภาพ [33]

ตารางที่ 11 ขนาดความกว้างและความยาวของตัวรับรู้อภาพ (Image sensor size)

ขนาดตัวรับรู้อภาพ (Size)	ความกว้าง (Width)	ความสูง (Height)
1/6 นิ้ว	2.4 mm	1.8 mm
1/4 นิ้ว (OV5647)	3.6 mm	2.7 mm
1/3 นิ้ว	4.8 mm	3.6 mm
1/2 นิ้ว	6.4 mm	4.8 mm
2/3 นิ้ว	8.8 mm	6.6 mm
1 นิ้ว	13.2 mm	8.8 mm
APS-C	22.3 mm	14.9 mm
APS-H	28.0 mm	19.0 mm
35mm full frame (DSLR)	36.0 mm	24.0 mm

จากภาพที่ 46 จะเห็นว่าพื้นที่ตัวรับรู้อภาพขนาด 1/4 นิ้ว มีขนาดเล็กกว่าพื้นที่ขอบเขตการมองเห็นของกล้องจุลทรรศน์ ส่งผลให้การแสดงภาพที่เกิดขึ้นบนชุดกล้องมีขนาดพื้นที่ที่เล็กกว่าพื้นที่ที่มองด้วยตาผ่านเลนส์ใกล้ตา ตามที่ได้คำนวณก่อนหน้านั้น กำลังการขยายของเลนส์กล้องจุลทรรศน์



$m_T = 400$  เท่า และค่ากำลังการขยายชุดกล้อง  $m_{cam} = 20$  เท่า ที่กำลังขยายของเลนส์ใกล้วัตถุ 40X จากขนาดพื้นที่ขอบเขตการมองเห็นของเลนส์ใกล้ตา 10X/FN20 มีค่าเป็น 0.19625 ตาราง มิลลิเมตร เป็นพื้นที่วงกลม จึงสามารถคำนวณขนาดพื้นที่ขอบเขตตัวรับรูปร่างได้จากสมการ

$$W_{real} = \frac{W_{sen}}{m_{cam}} \quad (59)$$

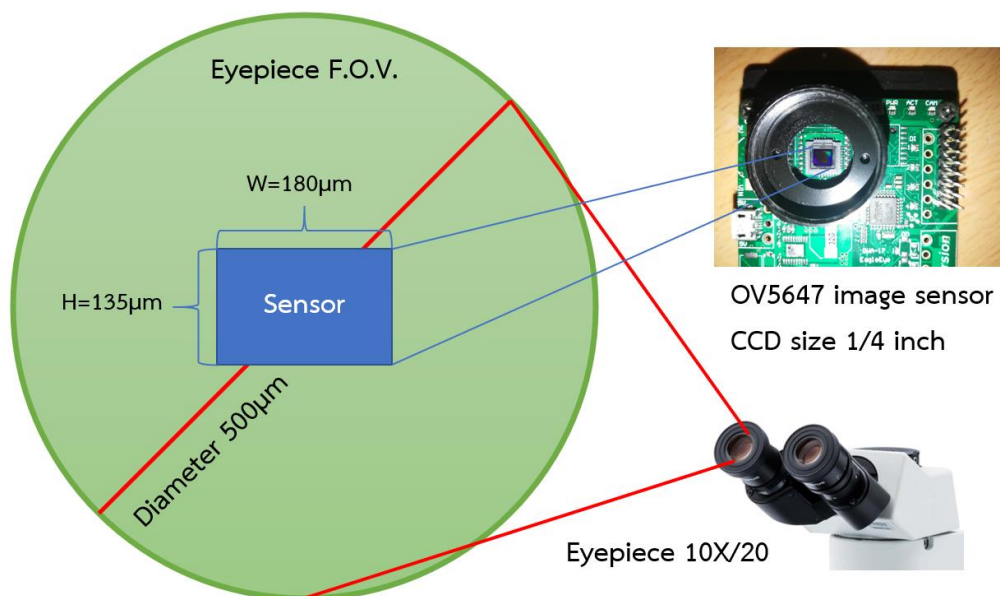
$$H_{real} = \frac{H_{sen}}{m_{cam}} \quad (60)$$

$$A_{real} = W_{real} \cdot H_{real} \quad (61)$$

$$A_r = \frac{AFV}{A_{real}} \quad (62)$$

เมื่อ  $W_{real}$  เป็นความกว้างของขอบเขตตัวรับรูปร่าง  $H_{real}$  เป็นความสูงของขอบเขตตัวรับรูปร่างปรากฏ  $W_{sen}$  เป็นความกว้างของตัวรับรูปร่าง  $H_{sen}$  เป็นความสูงของตัวรับรูปร่าง  $m_{cam}$  เป็นกำลังการขยายชุดกล้อง  $A_{real}$  เป็นขนาดของพื้นที่ที่ตัวรับรูปร่าง ซึ่งเป็นขนาดพื้นที่จริงที่ชุดกล้องสามารถแสดงภาพได้  $A_r$  เป็นอัตราส่วนของพื้นที่การมองเห็นของชุดกล้อง มีค่าเป็นกี่ ส่วนของพื้นที่ที่ตัวรับรูปร่าง

ตัวอย่างการพิสูจน์การคำนวณขนาดพื้นที่ขอบเขตตัวรับรูปร่าง จากการหาขนาดพื้นที่ขอบเขตการมองเห็นตามกำลังขยายเลนส์ใกล้วัตถุในสมการที่ 57 ที่เลนส์ใกล้วัตถุกำลังขยาย 40X และเลนส์ใกล้ตากำลังขยาย 10X มีเลขขอบเขตเป็น 20 ทำให้  $AFV$  มีค่าเป็น 0.19625 ตาราง มิลลิเมตร เป็นพื้นที่วงกลม จากนั้นคำนวณหาความกว้างของขอบเขตตัวรับรูปร่าง  $W_{real} = 3.6/20 = 0.18$  มิลลิเมตร หรือ 180 ไมโครเมตร และ ความสูงของขอบเขตตัวรับรูปร่าง  $H_{real} = 2.7/20 = 0.135$  มิลลิเมตร หรือ 135 ไมโครเมตร ทำให้สามารถคำนวณพื้นที่ที่ตัวรับรูปร่าง  $A_{real} = 0.18 \times 0.135 = 0.0243$  ตารางมิลลิเมตร สุดท้ายจึงหาค่าอัตราส่วนของพื้นที่การมองเห็นของชุดกล้องได้จาก  $A_r = 0.19625/0.0243 = 8.07613$  หรือมีค่าประมาณ 8.1 ส่วนของพื้นที่ขอบเขตการมองเห็น ทำให้จำนวนเซลล์ขนาดเล็กที่ปรากฏในภาพบนชุดกล้องต้องคูณด้วย 8.1 ส่วน จึงจะได้เท่ากับจำนวนเซลล์ขนาดเล็กที่ตามองเห็นผ่านเลนส์ใกล้ตาที่มีกำลังขยายเป็น 400 เท่า ซึ่งเป็นการการบ่งบอกจำนวนเซลล์ขนาดเล็กด้วยวิธีการแบบ Mitotic count [34] ที่ใช้ในการคัดกรองเซลล์มะเร็งในมนุษย์และในสัตว์



ภาพที่ 47 การเปรียบเทียบขนาดพื้นที่ระหว่าง ขอบเขตการมองเห็น และขอบเขตตัวรับรู้ปรากฏ

### 3.2.3 การบ่งบอกจำนวนเซลล์ขนาดเล็กด้วยหน่วยกำลังขยายภาพ

Mitotic count เป็นการนับจำนวนเซลล์ขนาดเล็กที่มองเห็นผ่านการส่องกล้องจุลทรรศน์ ได้ถูกนำมาใช้อย่างแพร่หลายในการระบุจำนวนเซลล์ขนาดเล็กในมนุษย์และในสัตว์ ทั้งการประมาณจำนวนเซลล์ที่ผิดปกติ หรือแม้แต่เซลล์ที่ปะปนมาในตัวอย่างปัสสาวะ สำหรับหน่วยที่ใช้ในการบ่งบอกจำนวนเซลล์ขนาดเล็กจะใช้เป็น จำนวนเซลล์ขนาดเล็กที่ปรากฏต่อพื้นที่กำลังขยายสูง (Per high power field : /HPF) โดยที่พื้นที่กำลังขยายสูงจะเลือกใช้เลนส์ใกล้วัตถุกำลังขยาย 40X และเลนส์ใกล้ตากำลังขยาย 10X กำลังขยายรวม 400 เท่า ในกรณีที่ต้องการหาจำนวนเซลล์ขนาดเล็กที่แน่นอนควรส่องเซลล์ตัวอย่างเป็นจำนวน 10 พื้นที่ที่แตกต่างกัน ดังนั้นจึงมีหน่วยที่บ่งบอกอีกหน่วยหนึ่ง เรียกว่า จำนวนเซลล์ขนาดเล็กที่ปรากฏต่อ 10 พื้นที่กำลังขยายสูง (Per 10 high power field : /10HPF) จากสมการที่ 62 จึงสามารถปรับปรุงหน่วยการระบุจำนวนเซลล์ใหม่ได้ว่า

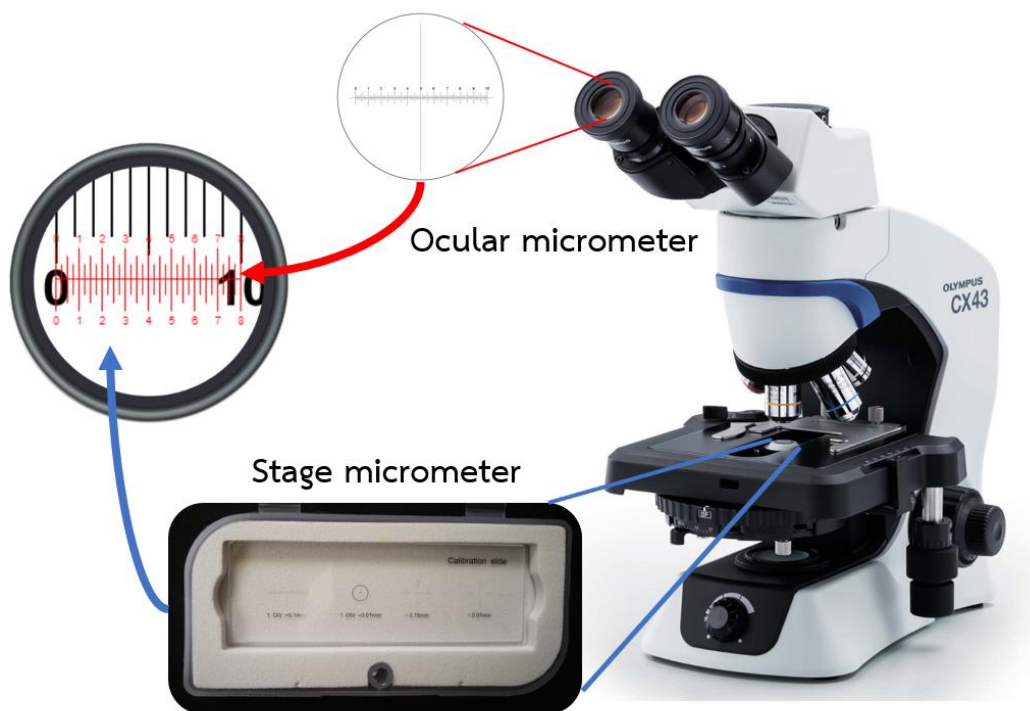
$$A_{std} = \frac{A_{std}}{A_{real}} \times n_{HPF} \quad (63)$$

$A_{std}$  เป็นอัตราส่วนของพื้นที่มาตรฐานการมองเห็นของชุดกล้อง  $A_{std}$  เป็นขนาดพื้นที่ขอบเขตการมองเห็นมาตรฐาน คือ เลนส์ใกล้วัตถุกำลังขยาย 40X และเลนส์ใกล้ตากำลังขยาย 10X มีเลขขอบเขตเป็น 20 มีค่าเป็น 0.19625 ตารางมิลลิเมตร  $A_{real}$  เป็นขนาดของพื้นที่ตัวรับรู้ปรากฏ  $n_{HPF}$  เป็นจำนวนพื้นที่กำลังขยายสูง

สำหรับการระบุหน่วยการนับแบบ Mitotic count สำหรับภาพถ่ายจากกล้องดิจิทัล สามารถระบุเป็นหน่วยเต็มรูปแบบ คือ  $A_{std}$  40X “field images” to equal standard area เช่นชุดกล้องนี้จะมีค่า  $A_{std}$  เป็น 81.0 40X “field images” to equal standard area หากต้องการระบุหน่วยอย่างย่อในรูป จำนวนเซลล์ขนาดเล็กที่ปรากฏต่อ 10 พื้นที่กำลังขยายสูง (Per 10 high power field : /10HPF) สามารถทำได้โดย

$$n_{10HPF} = n_{count} \times A_{std} \quad (64)$$

$n_{10HPF}$  เป็นจำนวนเซลล์ขนาดเล็กที่นับได้ใน 10 พื้นที่มาตรฐานการมองเห็นของชุดกล้อง เช่น 10 เซลล์/10HPF  $n_{count}$  เป็นจำนวนเซลล์ขนาดเล็กที่นับได้ในพื้นที่มาตรฐานการมองเห็นของชุดกล้อง

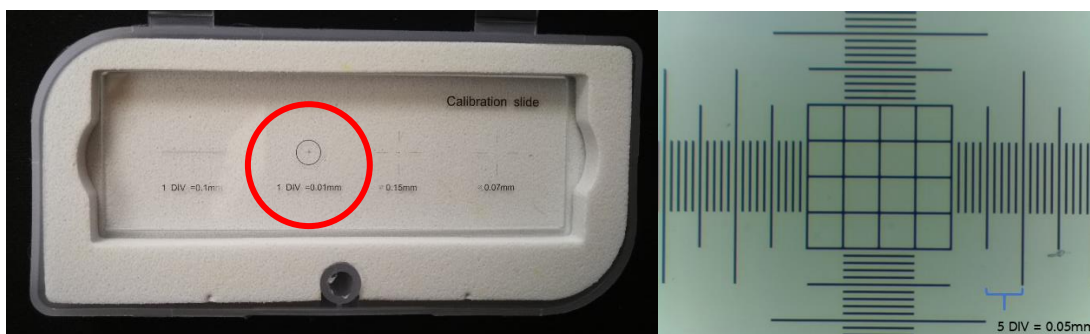


ภาพที่ 48 การสอบเทียบการวัดขนาดด้วย Ocular micrometer และ Stage micrometer

การสอบเทียบระยะการมองภาพ (Reticle calibration) เป็นวิธีการหนึ่งที่จะช่วยให้การวัดขนาดของวัตถุภายในการขยายของกล้องจุลทรรศน์มีความถูกต้อง อาศัยการวัดสเกลขนาดเล็กด้วยไมโครมิเตอร์ (Micrometer) เป็นอุปกรณ์สำหรับวัดขนาดวัตถุขนาดเล็กในระดับมิลลิเมตร และไมโครเมตร สำหรับกล้องจุลทรรศน์จะใช้ไมโครมิเตอร์ 2 ส่วน คือ

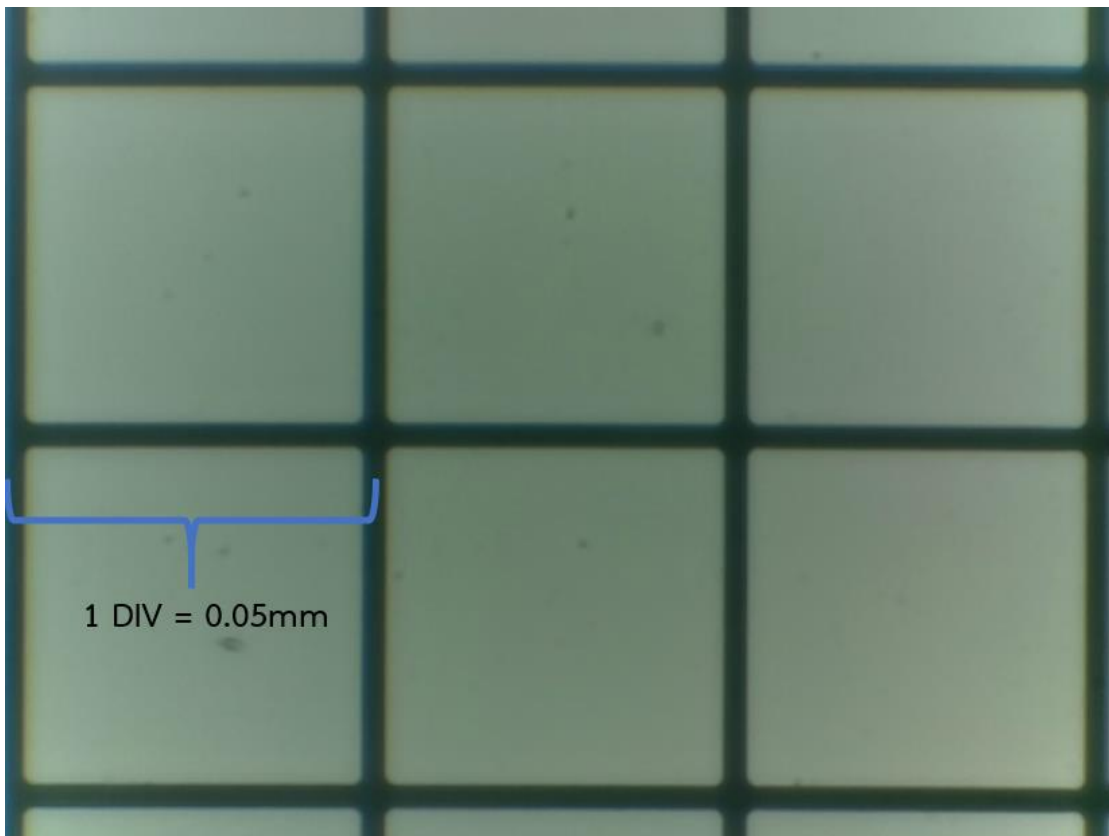
- 1.) Ocular micrometer หรือ Eyepiece micrometer เป็นแผ่นพลาสติกหรือแก้ววงกลมติดตั้งภายในท่อเลนส์ใกล้ตา (Ocular tube) บนตัวแผ่นจะระบุขีดสเกลไว้ (Scale) พร้อมตัวเลขกำกับในช่วงระยะห่าง 10 ช่อง ซึ่งระยะห่างระหว่างช่องนี้จะใช้เทียบตำแหน่งให้ตรงกับขีดสเกลของ Stage micrometer สำหรับสอบเทียบ
- 2.) Stage micrometer หรือ แผ่นสไลด์สอบเทียบ (Calibration slide) มีลักษณะเป็นแผ่นกระจกขนาดเท่าแผ่นสไลด์ บนตัวแผ่นจะระบุขีดสเกล (Scale) พร้อมตัวเลขกำกับในช่วงระยะห่าง 10 ช่อง ซึ่งระยะห่างระหว่างช่องของแผ่นสไลด์สอบเทียบจะมีระยะกำกับ เช่น 1 ช่อง = 0.1 มิลลิเมตร (1 DIV = 0.1mm) 1 ช่อง = 0.01 มิลลิเมตร (1 DIV = 0.01mm) เพื่อใช้เป็นอุปกรณ์สอบเทียบว่าขีดสเกลของ Ocular micrometer 1 ช่อง มีความกว้างเป็นกี่มิลลิเมตร หรือไมโครเมตร

สำหรับการสอบเทียบระยะการมองภาพของชุดกล้อง จะเปลี่ยนจากการใช้ Ocular micrometer เป็นการถ่ายภาพที่ถูกถ่ายจากชุดกล้อง ขนาดของภาพถ่ายจะมีความกว้าง และความยาวเท่ากับความกว้างตัวรับรู้อุปกรณ์ และความยาวตัวรับรู้อุปกรณ์ตามลำดับ ในภาพที่ 48 แสดงภาพที่ถูกถ่ายให้เห็นถึงขีดสเกลของแผ่นสไลด์สอบเทียบที่มีระยะห่างระหว่างช่องเป็น 1 ช่อง = 0.01 มิลลิเมตร หรือ 10 ไมโครเมตร (1 DIV = 0.01mm) หากเป็นความกว้างของพื้นที่ตารางบริเวณตรงกลางแผ่นสไลด์สอบเทียบจะมีระยะห่างระหว่างด้านเป็น 0.05 มิลลิเมตร หรือ 50 ไมโครเมตร

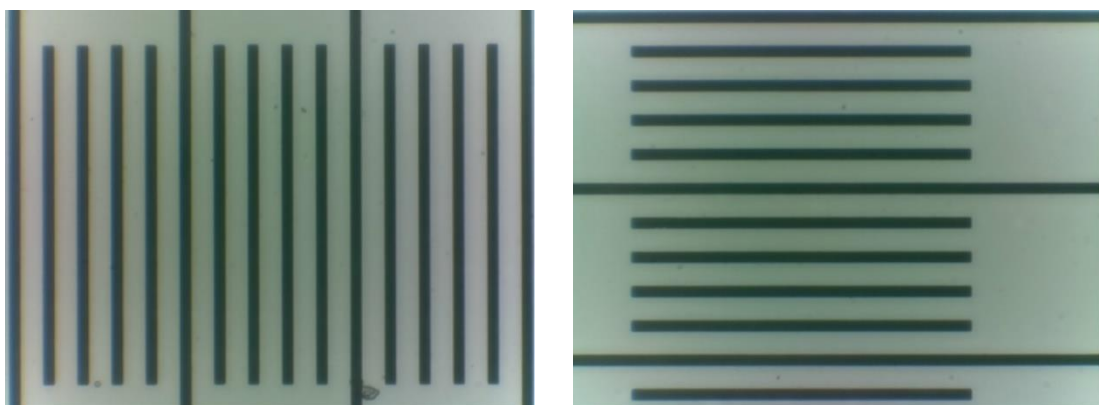


ภาพที่ 49 แผ่นสไลด์สอบเทียบ (Calibration slide) สำหรับกล้องจุลทรรศน์

การคำนวณหาความกว้างของขอบเขตตัวรับรู้ปรากฏ  $W_{real}$  มีค่าเป็น 0.18 มิลลิเมตร หรือ 180 ไมโครเมตร และความสูงของขอบเขตตัวรับรู้ปรากฏ  $H_{real}$  มีค่า 0.135 มิลลิเมตร หรือ 135 ไมโครเมตร จากภาพถ่ายของชุดกล้องในภาพที่ 49 และ 50 แสดงให้เห็นว่าความกว้างของตัวรับรู้ปรากฏที่วัดได้จริงเป็น 15 ช่อง จึงมีความกว้างเป็น 0.15 มิลลิเมตร และความสูงของตัวรับรู้ปรากฏที่วัดได้จริงเป็น 11 ช่อง จึงมีความสูงเป็น 0.11 มิลลิเมตร



ภาพที่ 50 แสดงพื้นที่ตัวรับรู้ปรากฏที่เกิดจากสอเทียบจริงด้วยเลนส์ใกล้วัตถุ 40X



ภาพที่ 51 แสดงความยาวตัวรับรู้ปรากฏ (ซ้าย) และความสูงตัวรับรู้ปรากฏ (ขวา)

เมื่อความกว้างของตัวรับรู้อภาพปรากฏที่วัดได้จริง และความสูงของตัวรับรู้อภาพปรากฏที่วัดได้จริง แตกต่างจากค่าที่คำนวณได้จากขนาดความกว้างและความยาวของตัวรับรู้อภาพในตารางที่ 11 ส่งผลให้การคำนวณพื้นที่ที่ตัวรับรู้อภาพมีค่าเปลี่ยนแปลงไป จึงต้องมีการคำนวณใหม่ได้ว่า  $A_{real} = 0.15 \times 0.11 = 0.0165$  ตารางมิลลิเมตร ทำให้ค่าอัตราส่วนของพื้นที่การมองเห็นของชุดกล้องมีค่าเป็น  $A_r = 0.19625/0.0165 = 11.89394$  หรือมีค่าประมาณ 11.89 ส่วนของพื้นที่ที่ขอบเขตการมองเห็น ทำให้จำนวนเซลล์ขนาดเล็กที่ปรากฏในภาพบนชุดกล้องต้องคูณด้วย 11.89 ส่วน จึงจะได้เท่ากับจำนวนเซลล์ขนาดเล็กที่ตามองเห็นผ่านเลนส์ใกล้ตาที่มีกำลังขยายเป็น 400 เท่า สำหรับการระบุหน่วยการนับแบบ Mitotic count สำหรับภาพถ่ายจากชุดกล้องนี้ จึงมีค่าเป็น 11.89 40X “field images” to equal standard area ซึ่งจะนำค่าพารามิเตอร์ไปเป็นส่วนหนึ่งของชุดคำสั่งภายในระบบโครงข่ายประสาทต่อไป

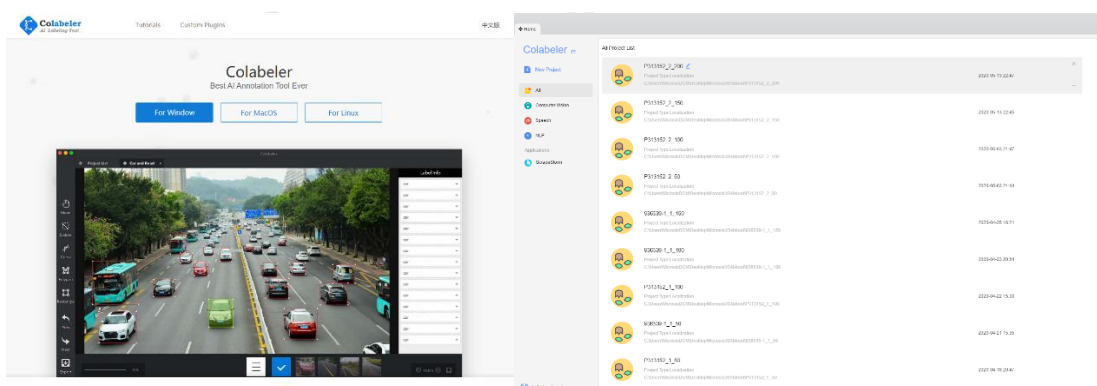
ตารางที่ 12 สรุปค่าพารามิเตอร์ของชุดกล้องที่ได้จากการคำนวณและการวัดได้จริง

ชนิดพารามิเตอร์	ค่าการคำนวณ	ค่าการวัดจริง
ความกว้างตัวรับรู้อภาพ	0.18 mm	0.15 mm
ความยาวตัวรับรู้อภาพ	0.135 mm	0.11 mm
พื้นที่ตัวรับรู้อภาพปรากฏ	0.0243 mm <sup>2</sup>	0.0165 mm <sup>2</sup>
อัตราส่วนของพื้นที่การมองเห็นของชุดกล้อง	$0.19625/0.0243 = 8.07613 = 8.1$	$0.19625/0.0165 = 11.89394 = 11.89$
ระบุหน่วยการนับแบบ Mitotic count	81.0 40X “field images” to equal standard area	118.9 40X “field images” to equal standard area
จำนวนเซลล์ขนาดเล็กที่นับได้ในพื้นที่มาตรฐานการมองเห็นของชุดกล้อง	8.1n /HPF *** n = จำนวนเซลล์	11.9n /HPF *** n = จำนวนเซลล์
จำนวนเซลล์ขนาดเล็กที่ปรากฏต่อ 10 พื้นที่กำลังขยายสูงของชุดกล้อง	81.0n /10HPF *** n = จำนวนเซลล์	118.9n /10HPF *** n = จำนวนเซลล์

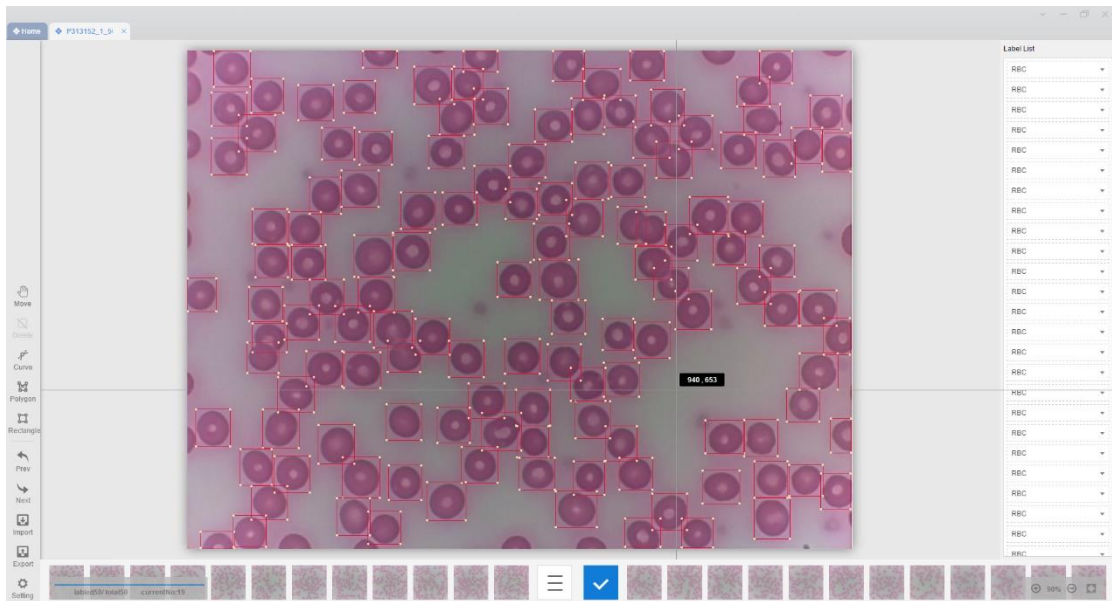
### 3.3 การออกแบบโครงข่ายประสาทสำหรับตรวจจับเซลล์ขนาดเล็ก

#### 3.3.1 การสร้างชุดข้อมูล (Dataset) และการติดป้าย (Labeling)

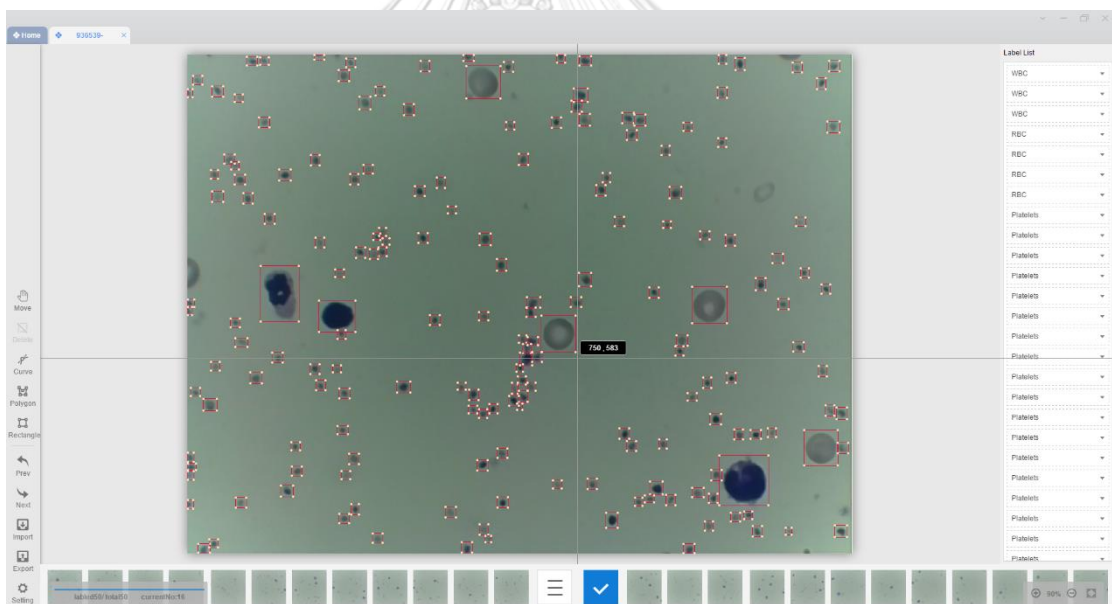
ระบบโครงข่ายประสาทที่ใช้ในชุดกล้อง มีการออกแบบให้ทำงานด้วยการเรียนรู้ของเครื่องแบบการเรียนรู้แบบมีผู้สอน (Supervised learning) เป็นการนำชุดข้อมูลสำหรับการฝึกนั้น (Training set) มาแยกประเภทผลลัพธ์ด้วยการติดป้าย (Labeling) แล้วจึงนำข้อมูลที่ติดป้ายแล้ว (Pre-labeled training data) ไปใช้ในการฝึกโมเดลต่อไป โดยการใช้เครื่องมือบรรณนิทัศน์ภาพ (Image annotation tool) เพื่อสร้างตำแหน่งของภาพโดยการวางพื้นที่สี่เหลี่ยมปิดล้อมวัตถุ (Bounding box : BBOX) คือพื้นที่สี่เหลี่ยมที่ถูกทาบบนวัตถุที่ต้องการระบุตำแหน่ง หรือพื้นที่จริงชั้นพื้นฐาน (Ground truth box) เพื่อติดป้ายระบุว่าวัตถุสิ่งนั้นเป็นคลาสชื่ออะไร ในระบบนี้จะถูกแยกประเภทออกเป็น 3 คลาส ได้แก่ ภาพเม็ดเลือดแดง กำหนดชื่อคลาสเป็น RBC ภาพเม็ดเลือดขาว กำหนดชื่อคลาสเป็น WBC และภาพเกล็ดเลือด กำหนดชื่อคลาสเป็น Platelets แล้วระบุเป็นตำแหน่งจุดภาพ 4 ตำแหน่ง คือ  $X_{min}$ ,  $Y_{min}$ ,  $X_{max}$  และ  $Y_{max}$  เรียกว่า Pascal VOC bounding box [14] สำหรับเครื่องมือบรรณนิทัศน์ภาพมีหลายเครื่องมือ เช่น Labelbox, Imglabel, DataTurks, LabelMe และ Colabeler เป็นต้น ทางผู้วิจัยเลือกใช้เครื่องมือ Colabeler จากเว็บไซต์ <http://www.colabeler.com/> เนื่องจากรองรับหลายระบบปฏิบัติการ สามารถติดป้ายภาพโดยที่ไม่ต้องเชื่อมต่อเครือข่าย มีปุ่มคีย์ลัดที่ช่วยอำนวยความสะดวกในการทำงาน บรรณนิทัศน์ได้ทั้งวางพื้นที่สี่เหลี่ยมปิดล้อม วางพื้นที่หลายเหลี่ยมปิดล้อม (Polygon bounding) และวางพื้นที่ขอบโค้งวัตถุ (Curve bounding) และสามารถบันทึกพิกัดพื้นที่ปิดล้อมได้หลายรูปแบบ เช่น Pascal VOC, XML, JSON และ MondoDB



ภาพที่ 52 เครื่องมือบรรณนิทัศน์ภาพ Colabeler บนระบบปฏิบัติการ Windows 10



ภาพที่ 53 การใช้เครื่องมือ Colabeler วางพื้นที่สี่เหลี่ยมปิดล้อมเม็ดเลือดแดง



ภาพที่ 54 การใช้เครื่องมือ Colabeler วางพื้นที่สี่เหลี่ยมปิดล้อมเม็ดเลือดขาว และเกล็ดเลือด

รูปแบบการสร้างไฟล์บันทึกพิกัดพื้นที่ปิดล้อมได้ หรือไฟล์บรรณนิทัศน์ (Annotation file) นั้นจะประกอบไปด้วยรายการตำแหน่งวัตถุที่ถูกวางพื้นที่สี่เหลี่ยมปิดล้อมในภาพนั้น ๆ 1 ไฟล์ต่อ 1 ภาพ เช่น หากในภาพหนึ่งมีวัตถุ 3 สิ่งที่ต้องการตรวจนับ จะทำให้มีบรรณนิทัศน์ทั้งหมด 3 คลาส จากรูปแบบการเก็บพิกัดพื้นที่ปิดล้อมแบบ Pascal VOC bounding box จะถูกบันทึกไฟล์นามสกุล XML (Extensible markup language) ที่ตัวอย่างแสดงไว้ดังนี้

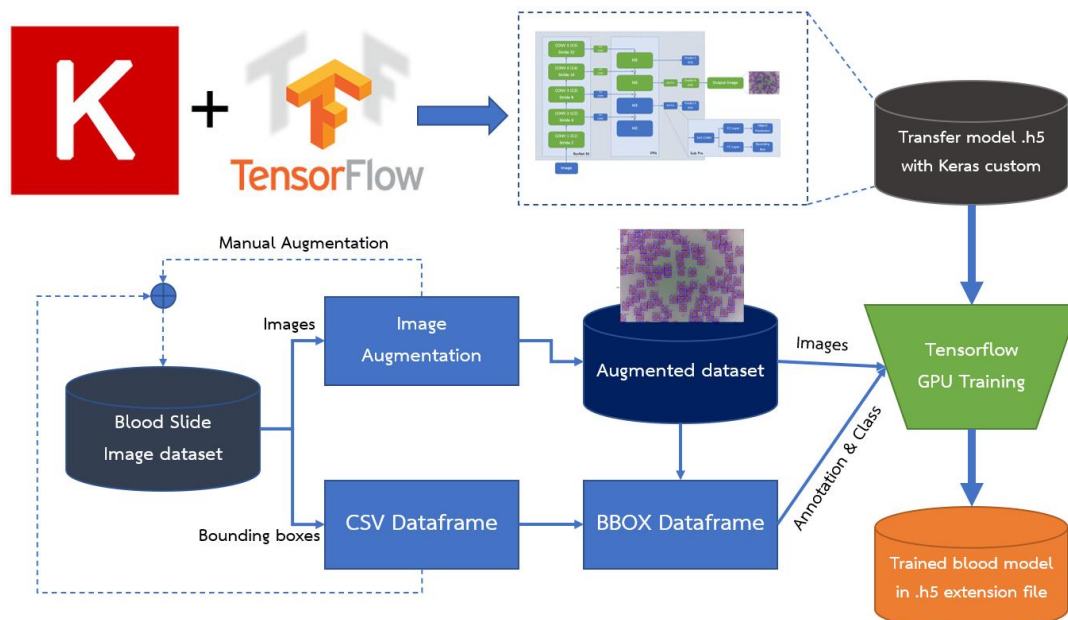


```
<?xml version="1.0" ?>
<annotation>
  <folder>P313152_1_50</folder>
  <filename>2020-04-16_15-50-36_CU.jpg</filename>
  <path>C:\MicrosisUSAblood\P313152_1_50\2020-04-16_15-50-36_CU.jpg</path>
  <source>
    <database>Unknown</database>
  </source>
  <size>
    <width>1280</width>
    <height>960</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  <object>
    <name>RBC</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>43</xmin>
      <ymin>1</ymin>
      <xmax>113</xmax>
      <ymin>53</ymin>
    </bndbox>
  </object>
  <object>
    ...
  </object>
  ...
</annotation>
```

- 1.) `<?xml version="1.0" ?>` เป็นการบ่งบอกว่าไปเป็นไฟล์ชนิด XML เรียกว่า XML declaration ใช้ในการระบุรุ่นของ XML specification และ Encoding declaration
- 2.) `<annotation> ... </annotation>` เป็นแท็ก (Tag) สำหรับบอกว่าเป็นบรรณนิทัศน์
- 3.) `<folder> ... </folder>` เป็นแท็กระบุชื่อแฟ้มที่ใช้เก็บภาพและบรรณนิทัศน์
- 4.) `<filename> ... </filename>` เป็นแท็กระบุชื่อไฟล์ภาพ
- 5.) `<path> ... </path>` เป็นแท็กระบุว่าไฟล์ภาพถูกเก็บไว้ตำแหน่งใดของคอมพิวเตอร์
- 6.) `<source> <database> ... </database> </source>` เป็นแท็กระบุแหล่งที่มาชุดข้อมูล
- 7.) `<size> ... </size>` เป็นแท็กที่เก็บขนาดของภาพ ประกอบด้วย ความกว้าง `<width> ... </width>` ความสูง `<height> ... </height>` และความลึก `<depth> ... </depth>` หากเป็นภาพขาวดำค่าความลึกจะเป็น 1 หากเป็นภาพสี ค่าความลึกจะเป็น 3
- 8.) `<segmented> ... </segmented>` ใช้บ่งบอกตำแหน่งพิกัด X และ Y ของจุดยอดพื้นที่หลายเหลี่ยมปิดล้อมรอบวัตถุที่ถูกแบ่งออกเป็น ส่วน (Segmentation mask)
- 9.) `<object> ... </object>` เป็นแท็กที่เก็บรายละเอียดของวัตถุ หากมีจำนวนวัตถุที่ถูกติดป้ายจำนวนหลายตำแหน่ง จะมีแท็กวัตถุตามจำนวนวัตถุที่ถูกติดป้าย ซึ่งภายในจะมีแท็กย่อยประกอบไปด้วย `<name>`, `<pose>`, `<truncated>`, `<difficult>` และ `<bndbox>`
- 10.) `<name> ... </name>` เป็นแท็กชื่อตามคลาสที่ติดป้าย เช่น RBC, WBC และ Platelets
- 11.) `<pose>Unspecified</pose>` เป็นแท็กระบุท่าทางของวัตถุที่ติดป้าย
- 12.) `<truncated>0</truncated>` เป็นแท็กระบุว่ากรอบพื้นที่สี่เหลี่ยมปิดล้อมนั้นสอดคล้องกับขอบเขตของวัตถุในภาพหรือไม่ หากวัตถุนั้นมีขอบเขตที่มองเห็นได้บางส่วนจะเป็น 1 หากวัตถุนั้นมีขอบเขตที่มองเห็นได้โดยสมบูรณ์จะเป็น 0
- 13.) `<difficult>0</difficult>` เป็นแท็กระบุความยากในการระบุคลาสของวัตถุนั้น ๆ หากวัตถุนั้นระบุได้ยากจะเป็น 1 หากวัตถุนั้นระบุได้ง่ายจะเป็น 0
- 14.) `<bndbox> ... </bndbox>` เป็นแท็กระบุพิกัดพื้นที่สี่เหลี่ยมปิดล้อมด้วยพิกัด X และ Y ในภาพ ประกอบด้วย `<xmin> ... </xmin>` `<ymin> ... </ymin>` `<xmax> ... </xmax>` `<ymax> ... </ymax>` ซึ่งจะมีค่าไม่เกิดความกว้างและความสูงของภาพ

ขั้นตอนนี้จะทำให้ได้ชุดข้อมูลภาพถ่ายเซลล์เม็ดเลือดจากกล้องจุลทรรศน์ (Blood Cell Dataset) ซึ่งเป็นภาพที่ได้รับการติดป้าย (Labeling) ประกอบด้วย เม็ดเลือดแดง (Red blood cell : RBC) เม็ดเลือดขาว (White blood cell : WBC) และเกล็ดเลือด (Platelets) และถูกแบ่งออกเป็นชุดสำหรับการฝึก (Training set) ร้อยละ 70 ชุดสำหรับทดสอบ (Test set) ร้อยละ 20 และชุดตรวจสอบความถูกต้อง (Validation set) ร้อยละ 10

### 3.3.2 กรรมวิธีการขยายชุดภาพ (Image augmentation)

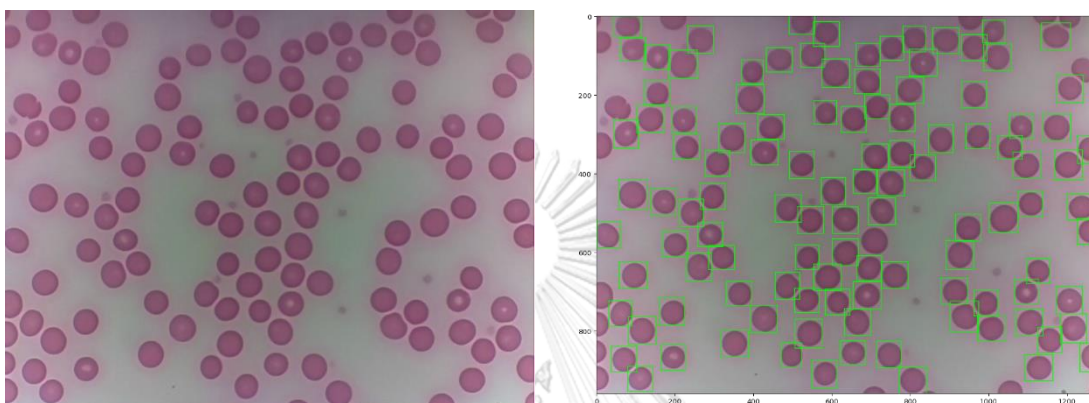


ภาพที่ 55 ผังภาพรวมของการนำชุดข้อมูลไปใช้ฝึกโครงข่ายประสาทของชุดกล้อง MicrosisiDCN

การฝึกความสามารถในการจดจำของโครงข่ายประสาท เพื่อนำมาใช้ตรวจจับหรือคัดแยกวัตถุที่ต้องการออกจากภาพนั้น จะต้องอาศัยชุดข้อมูลภาพที่ถูกติดป้ายแล้วมาใช้ในการฝึก ซึ่งประสิทธิภาพในการทำงานและความแม่นยำในการคัดแยกวัตถุของโมเดล จะขึ้นอยู่กับปริมาณข้อมูลที่ใช้ในการฝึก เช่นเดียวกับการเอาภาพอักษร ก. ให้เด็กดูสำหรับคัดลายมือ ครู 1 คนจะเขียนลายมือได้ 1 แบบ ถ้าจะเอาลายมือ 20 แบบ ต้องเรียกครู 20 คนมาเขียนจะทำให้มีภาระงานหนักขึ้น เหมือนการนำภาพมาติดป้ายด้วยมือ ถ้าหากได้ครู 1 คนสร้างแบบลายมือได้ 20 แบบ แล้วนำมาให้เด็กดู เด็กที่เห็นภาพ 20 แบบ ย่อมอ่านได้ดีกว่าเด็กเห็นภาพเพียงแค่ 1 แบบ ในทำนองเดียวกันหากโมเดลโครงข่ายประสาทถูกฝึกด้วยภาพจำนวนหนึ่ง โมเดลอาจตรวจจับหรือคัดแยกเฉพาะวัตถุที่เรียนรู้จากชุดข้อมูล ในกรณีที่นำชุดกล้องไปส่องกับเซลล์เม็ดเลือดที่ย้อมสีมาไม่ดี หรือการปรับความสว่าง ความคมชัดของกล้องจุลทรรศน์ที่ไม่ได้ จะส่งผลการความแม่นยำในการคัดแยกเซลล์เม็ดเลือดแดง เม็ดเลือดขาว และเกล็ดเลือด นั้นลดลง

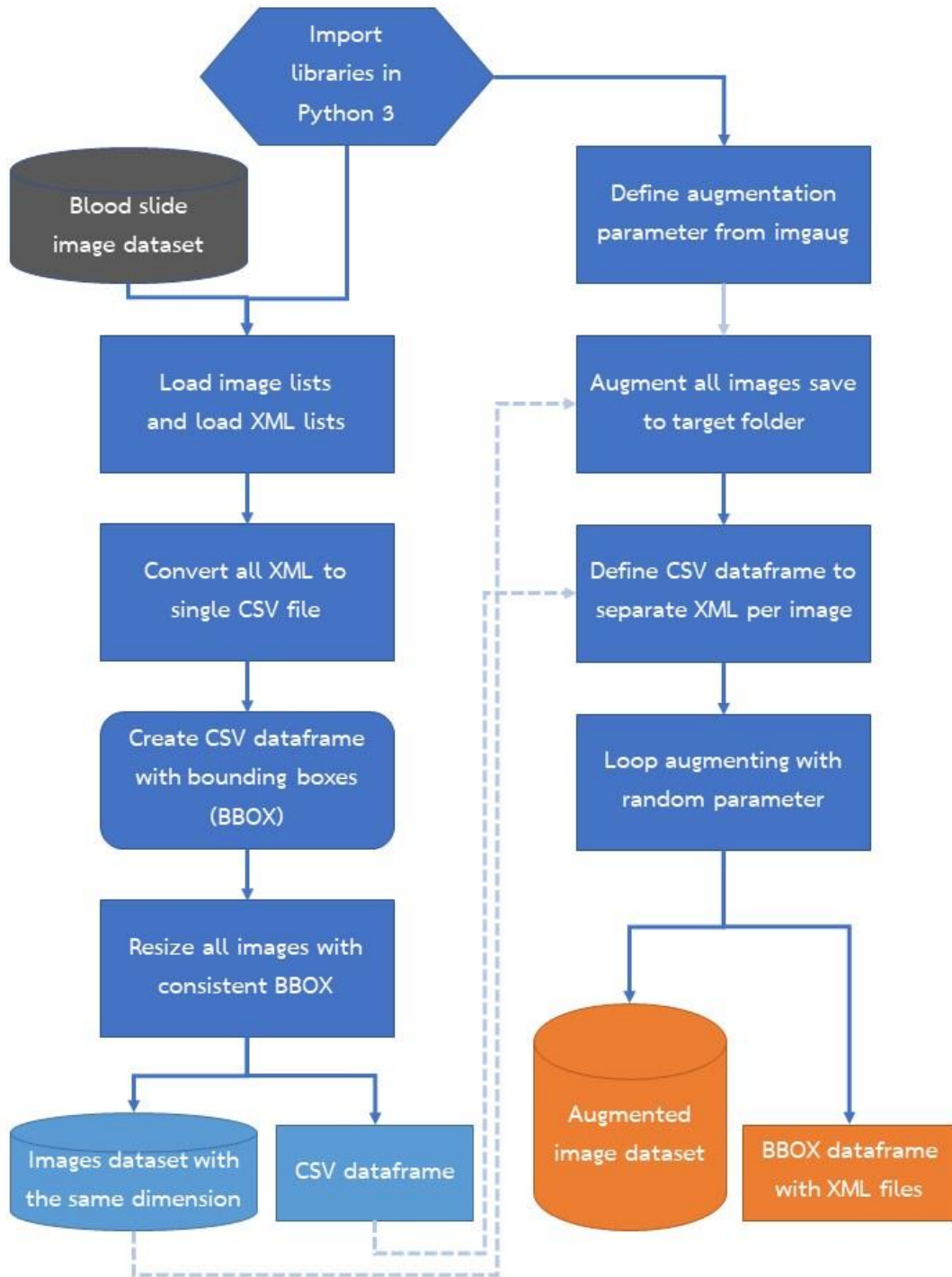
ในการขยายชุดภาพของชุดข้อมูลที่ผ่านมาการติดป้ายแล้วนั้น ผู้วิจัยจึงต้องเลือกชุดคำสั่งที่สามารถนำมาพัฒนาได้ คือ imgaug เป็นคลังชุดคำสั่ง (Library) ที่ใช้ในการขยายชุดภาพ (Image augmentation) สำหรับนำมาใช้เป็นชุดข้อมูลสำหรับฝึกโครงข่ายประสาท ซึ่ง imgaug นี้มีการสนับสนุนกรรมวิธีการขยายชุดภาพที่หลากหลาย [35] มาใช้สร้างชุดข้อมูลภาพถ่ายเซลล์เม็ดเลือดจาก

กล้องจุลทรรศน์ โดยการตัดแปลงจากภาพเดิมที่มีอยู่ แต่ต้องไม่เปลี่ยนแปลงจนคุณลักษณะภาพเปลี่ยนแปลงไปจากเดิมมากเกินไป ส่งผลให้ประสิทธิภาพในการฝึกโมเดลโครงข่ายประสาท และความแม่นยำลดลง นอกจากนี้การขยายชุดภาพจะช่วยลดการเกิด Overfitting ของโมเดล หรือการตอบสนองต่อการรบกวนจำนวนมาก จนนำการรบกวนที่ทำให้เกิดข้อมูลไม่ถูกต้องไปเป็นผลในการเรียนรู้ ส่งผลให้ประสิทธิภาพการตรวจจับและคัดแยกแยะไปจากเดิม

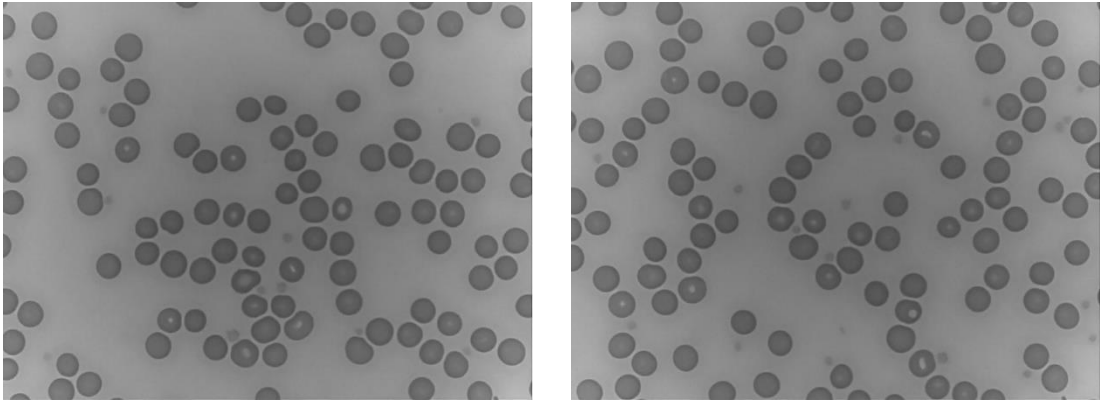


ภาพที่ 56 ภาพถ่ายเซลล์เม็ดเลือดแดง กับภาพถ่ายที่ถูกบรณนนิทัศน์ (BBOX)

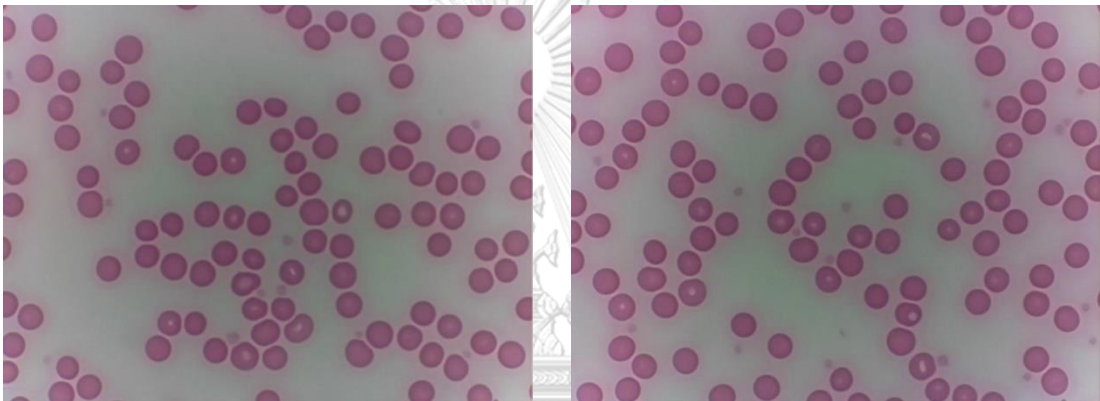
การขยายชุดภาพที่ผ่านการติดป้าย และไฟล์ XML เก็บพิกัดพื้นที่ปิดล้อมแบบ Pascal VOC bounding box จะพบปัญหาว่า สามารถขยายชุดภาพได้ แต่ไม่สามารถขยายชุดไฟล์ XML ตามชุดภาพที่เพิ่มขึ้นได้ วิธีการดั้งเดิมคือการปรับปรุงไฟล์ XML ด้วยมือ จัดเป็นการขยายชุดข้อมูลด้วยมือ (Manual augmentation) ที่ต้องใช้เวลาและคนจำนวนมาก ประกอบกับการใช้ TensorFlow ในการฝึกโมเดลโครงข่ายประสาท API จะอ่านพิกัดพื้นที่ปิดล้อมจากไฟล์ XML ตามรูปแบบของ Pascal VOC ไปแปลงเป็นไฟล์ CSV (Comma Separated Value) ให้อยู่ในรูปของ Dataframe เมื่ออยู่ในรูปดังกล่าวแล้ว ถ้าต้องการปรับปรุงพิกัดพื้นที่ปิดล้อม หรือแก้ไขเฉพาะภาพทำได้ยาก เพราะไฟล์ XML จะบันทึกพิกัดและข้อมูลภาพต่อ 1 ภาพ ในขณะที่ไฟล์ CSV จะบันทึกพิกัดและข้อมูลภาพทั้งแฟ้มไว้ในไฟล์เดียว เหตุนี้ทำให้นักพัฒนาส่วนหนึ่งไม่สามารถนำชุดภาพที่ผ่านการขยายไปใช้ในการฝึกโมเดลได้อย่างเต็มที่ ทางผู้วิจัยจึงได้พัฒนาชุดคำสั่งในการบันทึกไฟล์ XML ให้เท่ากับจำนวนชุดภาพที่ถูกขยาย (CSV to XML dataframe) ซึ่งสามารถนำชุดข้อมูลไปใช้ฝึกโครงข่ายประสาทใน TensorFlow ต่อไป มีผังการทำงานแสดงในภาพที่ 57 ตัวอย่างชุดภาพที่ถูกขยายแล้วแสดงในภาพที่ 58 จนถึงภาพที่ 66



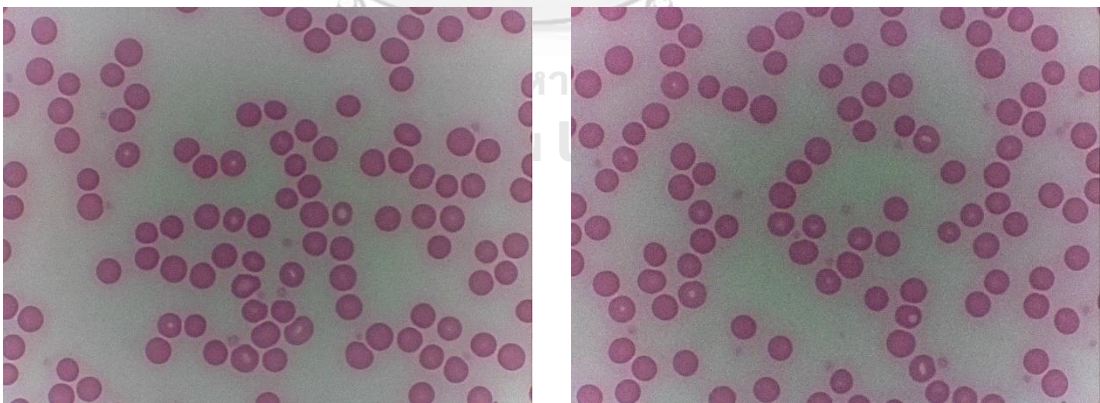
ภาพที่ 57 ผังขั้นตอนการทำงานของกรรมวิธีการขยายชุดภาพ (Image augmentation)



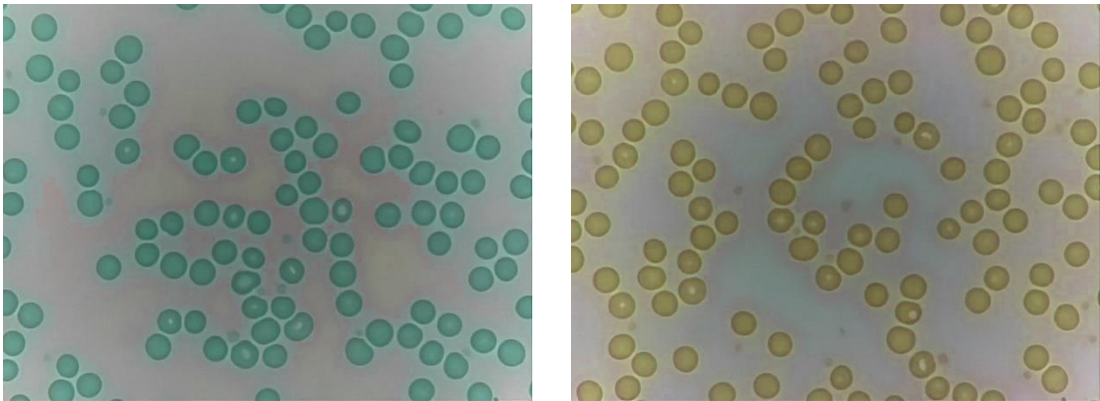
ภาพที่ 58 ภาพถ่ายเซลล์เม็ดเลือดแดงที่ผ่านการขยายชุดภาพด้วยฟังก์ชัน Grayscale



ภาพที่ 59 ภาพถ่ายเซลล์เม็ดเลือดแดงที่ผ่านการขยายชุดภาพด้วยฟังก์ชัน GaussianBlur

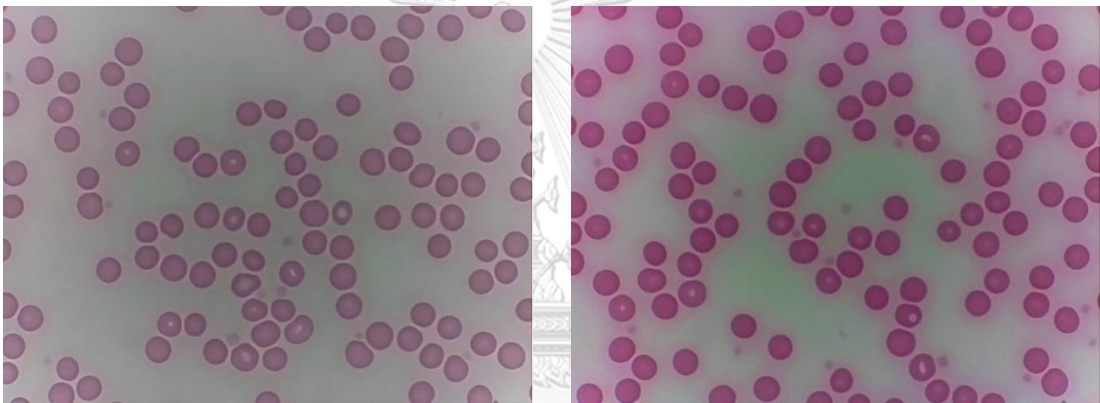


ภาพที่ 60 ภาพถ่ายเซลล์เม็ดเลือดแดงที่ผ่านการขยายชุดภาพด้วยฟังก์ชัน AdditiveGaussianNoise



ภาพที่ 61 ภาพถ่ายเซลล์เม็ดเลือดแดงที่ผ่านการขยายชุดภาพด้วยฟังก์ชัน

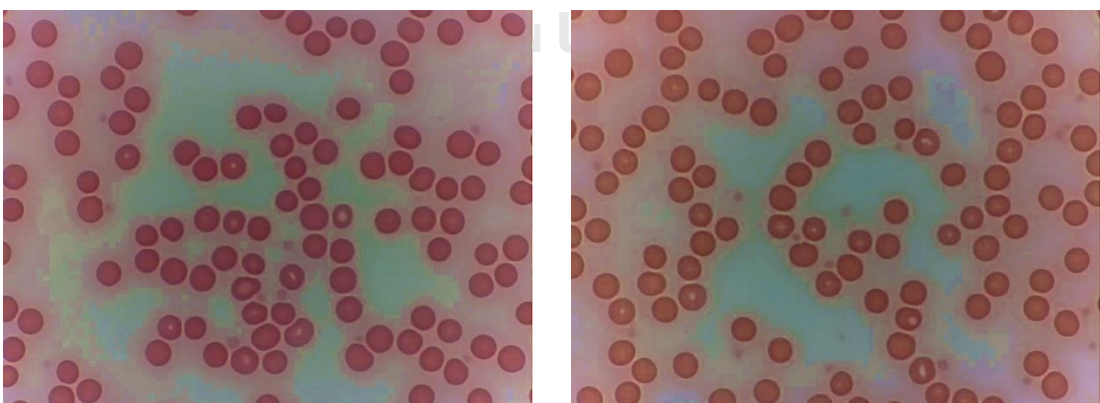
MultiplyHueAndSaturation (Hue)



ภาพที่ 62 ภาพถ่ายเซลล์เม็ดเลือดแดงที่ผ่านการขยายชุดภาพด้วยฟังก์ชัน

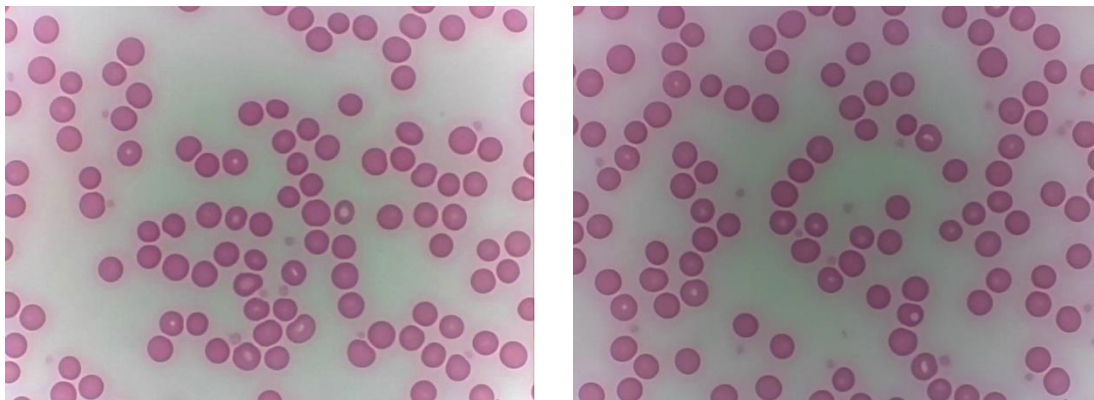
MultiplyHueAndSaturation (Saturation)

จุฬาลงกรณ์มหาวิทยาลัย

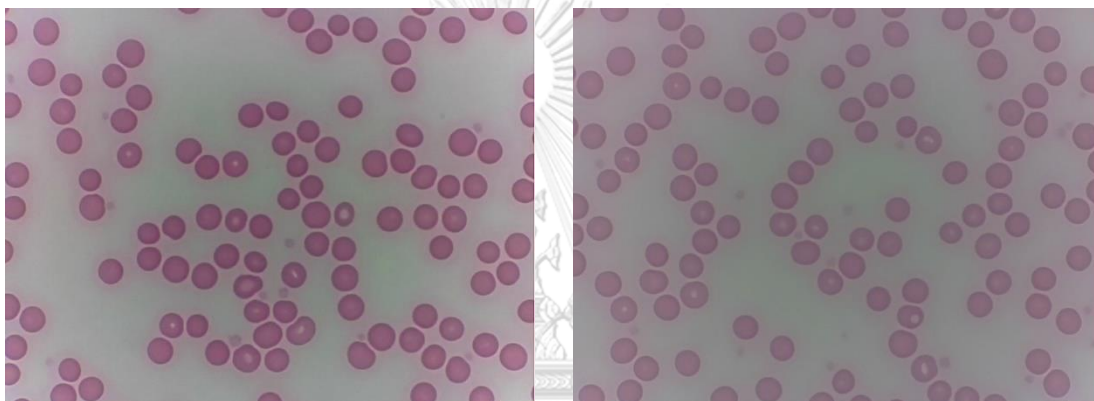


ภาพที่ 63 ภาพถ่ายเซลล์เม็ดเลือดแดงที่ผ่านการขยายชุดภาพด้วยฟังก์ชัน

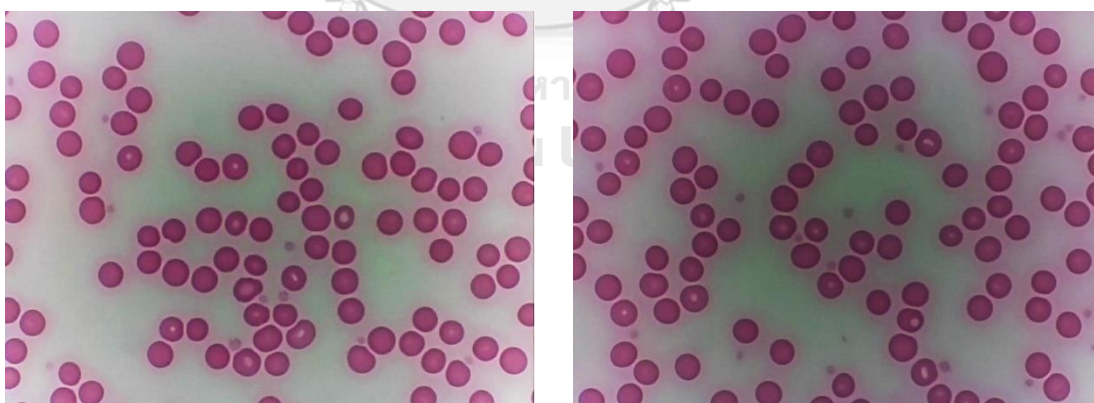
AddToHueAndSaturation (ทั้งคู่)



ภาพที่ 64 ภาพถ่ายเซลล์เม็ดเลือดแดงที่ผ่านการขยายชุดภาพด้วยฟังก์ชัน MultiplyBrightness



ภาพที่ 65 ภาพถ่ายเซลล์เม็ดเลือดแดงที่ผ่านการขยายชุดภาพด้วยฟังก์ชัน GammaContrast



ภาพที่ 66 ภาพถ่ายเซลล์เม็ดเลือดแดงที่ผ่านการขยายชุดภาพด้วยฟังก์ชัน SigmoidContrast

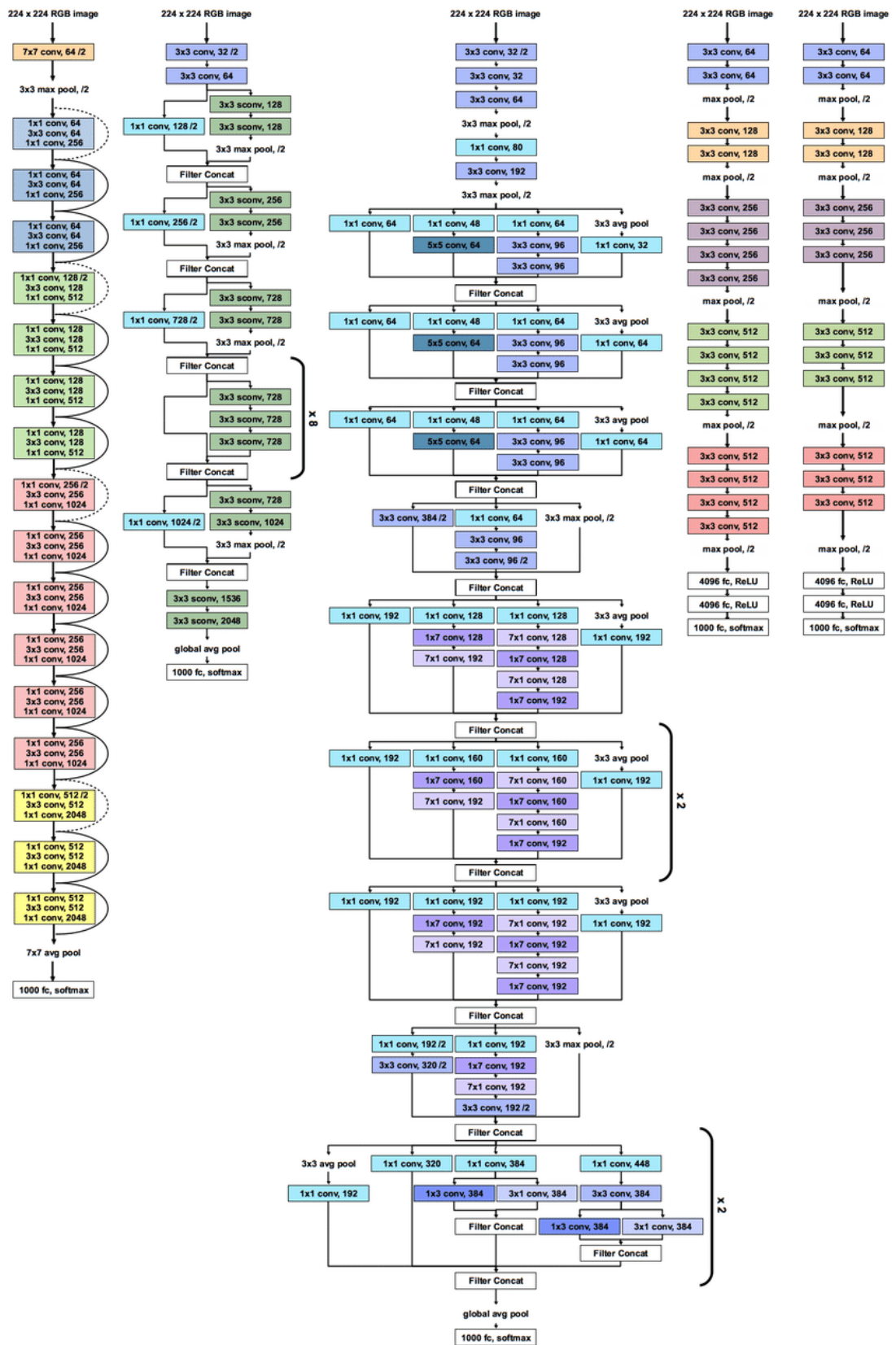


### 3.3.3 กรรมวิธีพัฒนาโครงข่ายประสาทสำหรับตรวจจับเซลล์ขนาดเล็ก

การจำแนกวัตถุจากภาพด้วยโครงข่ายประสาทนั้น เป็นการทำให้คอมพิวเตอร์มีความสามารถในการรับรู้ และประมวลผลเพื่อจำแนกวัตถุว่าสิ่งนั้นคืออะไร สามารถตอบสนองต่อภาพที่ถูกป้อนข้อมูลเชิงเลข เสมือนหลักการทำงานของสมองมนุษย์ผ่านทาง การมองเห็นภาพด้วยตา เรียกว่า คอมพิวเตอร์วิทัศน์ (Computer vision) ประเภทงานทางคอมพิวเตอร์วิทัศน์แบ่งออกเป็น 4 ด้าน

- 1.) การจัดหมวดหมู่วัตถุ (Object classification) เป็นการจำแนกวัตถุจากภาพว่าเป็นวัตถุชนิดใดในภาพนั้น ๆ ซึ่งจะระบุเป็นคลาสโดยภาพรวมเพียงคลาเดียว
- 2.) การจัดหมวดหมู่วัตถุและระบุตำแหน่ง (Classification and localization) เป็นการจำแนกวัตถุจากภาพ พร้อมระบุตำแหน่งว่าเป็นวัตถุชนิดใด ซึ่งจะระบุเป็นคลาเดียวในภาพโดยใช้พื้นที่สี่เหลี่ยมปิดล้อม (Bounding box)
- 3.) การตรวจจับวัตถุ (Object detection) เป็นการจำแนกวัตถุมากกว่าหนึ่งคลาจากภาพ พร้อมระบุตำแหน่งของแต่ละคลาในภาพนั้น ๆ อาจเป็นคลาเดียวกันหรือคลาต่างกัน โดยใช้พื้นที่สี่เหลี่ยมปิดล้อม (Bounding box)
- 4.) การแบ่งส่วนภาพ (Image segmentation) เป็นการจำแนกวัตถุมากกว่าหนึ่งคลาในภาพ พร้อมระบุตำแหน่งของแต่ละคลาในภาพนั้น ๆ ด้วย วา่งพื้นที่หลายเหลี่ยมปิดล้อม (Polygon bounding) หรือวางพื้นที่ขอบโค้งวัตถุ (Curve bounding)

ในช่วงปี พ.ศ.2558 มีการแข่งขันชื่อว่า Large Scale Visual Recognition Challenge 2015 (ILSVRC2015) ในการใช้ชุดข้อมูลภาพ (Image dataset) เพื่อให้ให้นักวิจัยและผู้พัฒนาโมเดลโครงข่ายประสาทนำชุดข้อมูลภาพไปใช้ในการทดสอบ เพื่อหาโครงข่ายประสาทที่มีประสิทธิภาพสูงสุด จากการระดมทุนและนักวิจัยในการเตรียมชุดข้อมูลภาพมากกว่า 1.4 ล้านภาพ แบ่งออกเป็นวัตถุที่ได้รับการติดป้ายทั้งหมด 1000 คลาส ซึ่งทุกภาพจะถูกระบุคลาที่ต้องไว้วางใจว่าวัตถุนั้นคืออะไร ชุดข้อมูลภาพนี้เรียกว่า ImageNet จึงเกิดการพัฒนากการตรวจจับวัตถุในภาพ (Object detection) ด้วยโครงข่ายประสาทรูปแบบต่าง ๆ เช่น VGG Net, R-CNN, Fast-RCNN, Faster-RCNN, GoogleNet หรือ Inception, ResNet, RetinaNet เป็นต้น จากงานวิจัย Deep Residual Learning for Image Recognition [36] ถูกอ้างอิงใน Google scholar 47,481 ครั้ง ช่วง มกราคม พ.ศ.2559 จนถึง มิถุนายน พ.ศ.2563 นำเสนอการแก้ปัญหา Vanishing gradient คือ ปัญหาที่เกิดขึ้นในระหว่างการฝึกโมเดลโครงข่ายประสาท เมื่อขนาดของความลาด (Gradient) ลดลงเรื่อย ๆ จนเท่ากับ 0 จะส่งผลให้ค่าน้ำหนัก (Weight) ไม่ถูกปรับเป็นค่าปัจจุบันและโมเดลไม่ถูกฝึกต่อ มักเกิดกับโครงข่ายประสาทที่มีลำดับชั้นลึกมาก แก้ไขได้โดยการทางลัด (Shortcut) ข้ามลำดับชั้นภายในของโครงข่ายประสาท เช่น การใส่ฟังก์ชันกระตุ้น (Activation function) ReLU แทน Sigmoid เป็นต้น เรียกโครงข่ายประสาทชนิดนี้ว่า Residual Network (ResNet)



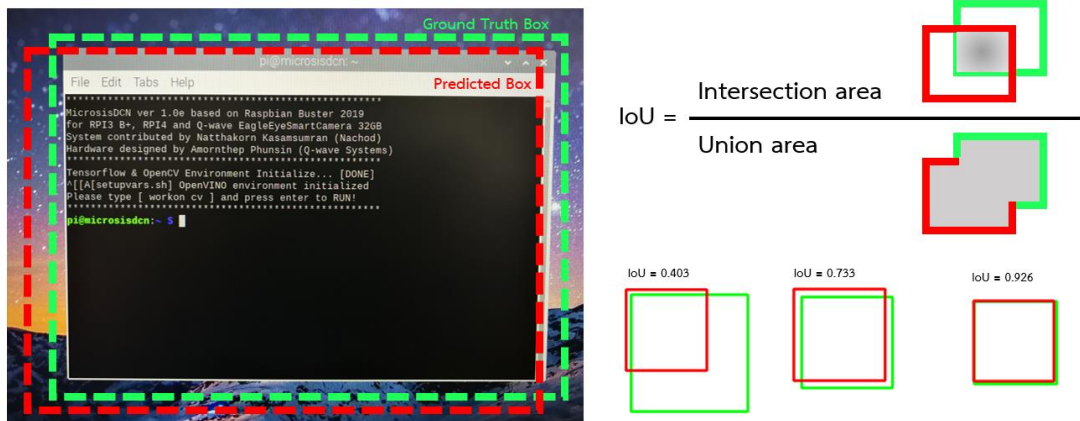
ภาพที่ 67 ลักษณะโครงสร้างข่ายประสาท ResNet50, Xception, Inception V3, VGG19, VGG16 [37]

ถัดมางานวิจัย Focal Loss for Dense Object Detection [38] ถูกอ้างอิงใน Google scholar 3,403 ครั้ง ช่วง มกราคม พ.ศ.2561 จนถึงมิถุนายน พ.ศ.2563 เป็นการปรับปรุงโครงสร้างของโมเดลโครงข่ายประสาท Residual Network (ResNet) ใช้ร่วมกับ Feature Pyramid Network (FPN) และ Focal loss เข้ามาแก้ไขปัญหาความแตกต่างของคลาส (Class imbalance) ที่มีจำนวนมาก โดยการสนใจค่าน้ำหนักที่ถูกจัดหมวดหมู่ของแต่ละคลาส หากจัดหมวดหมู่จำแนกได้อย่างถูกต้องในชั้นคอนโวลูชันแล้ว จะให้ความสำคัญกับคลาสนั้นลดลง ส่งผลให้ค่าการสูญเสียลดลง และการพยากรณ์ผลลัพธ์ถูกต้อง ต่อมา Hans Gaiser และคณะได้เผยแพร่คลังชุดคำสั่ง Keras RetinaNet ที่ช่วยให้นักพัฒนาสามารถปรับปรุงและฝึกโมเดลโครงข่ายประสาท RetinaNet ได้บน Keras ผ่านเครื่องมือ Jupyter Notebook เป็นโปรแกรมใช้ในการเขียนภาษา Python ผ่านหน้า Browser เช่น Google Chrome ทดแทนการเขียนโปรแกรมด้วย Text Editor ชุดคำสั่งที่เกี่ยวข้องสามารถศึกษาเพิ่มเติมได้ที่ <https://github.com/fizyr/keras-retinanet> และยังสามารถนำไปใช้ได้โดยไม่มีค่าใช้จ่ายตามสัญญา Apache License 2.0

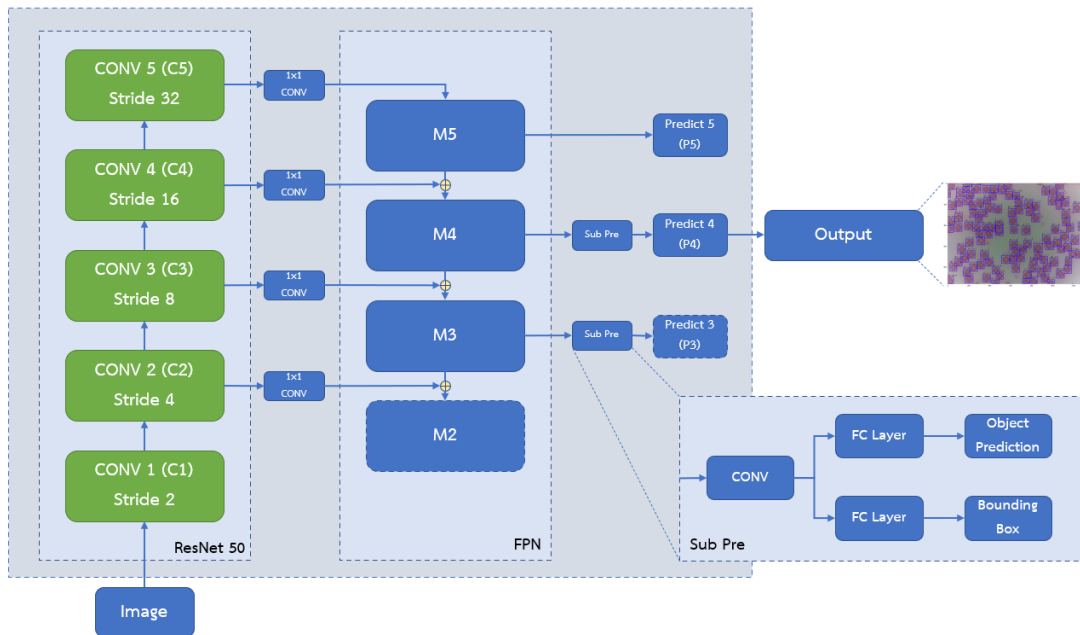
ตารางที่ 13 เปรียบเทียบ Average Precision (AP) ของโมเดลต่าง ๆ [38]

Model	Backbone	AP	AP <sub>50</sub>	AP <sub>75</sub>	AP <sub>S</sub>	AP <sub>M</sub>	AP <sub>L</sub>
Faster R-CNN	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN with FPN	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN with TDM	Inception-ResNet	36.8	55.7	39.2	16.2	39.8	52.1
YOLOv2	DarkNet-19	21.6	44.0	19.2	5.0	22.4	35.5
SSD513	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
RetinaNet	ResNet-101-FPN	40.8	61.1	44.1	24.1	44.2	51.2

ในการพิจารณาความสามารถของโมเดลในการพยากรณ์อย่างถูกต้องด้วยค่า AP จะต้องพิจารณาร่วมกับ Intersection over union (IoU) เป็นอัตราส่วนระหว่างพื้นที่ของพื้นที่สี่เหลี่ยมปิดล้อม (Bounding box) ที่เกิดซ้อนทับของพื้นที่จริงขั้นพื้นฐาน (Ground truth box) และพื้นที่จากการพยากรณ์ (Predicted box) กับพื้นที่รวมของทั้งหมด จากตารางที่ 13 AP คือค่าร้อยละ AP ที่ IoU=0.50 : 0.05 : 0.95 (Primary challenge metric in COCO) AP<sub>50</sub> คือ ค่าร้อยละ AP ที่ IoU=0.50 (PASCAL VOC metric) AP<sub>75</sub> คือค่าร้อยละ AP ที่ IoU=.75 (Strict metric in COCO) AP<sub>S</sub> คือ ค่าร้อยละ AP ของวัตถุขนาดเล็ก (Small object) AP<sub>M</sub> คือ ค่าร้อยละ AP ของวัตถุขนาดกลาง (Medium object) และ AP<sub>L</sub> คือค่าร้อยละ AP ของวัตถุขนาดใหญ่ (Large object)



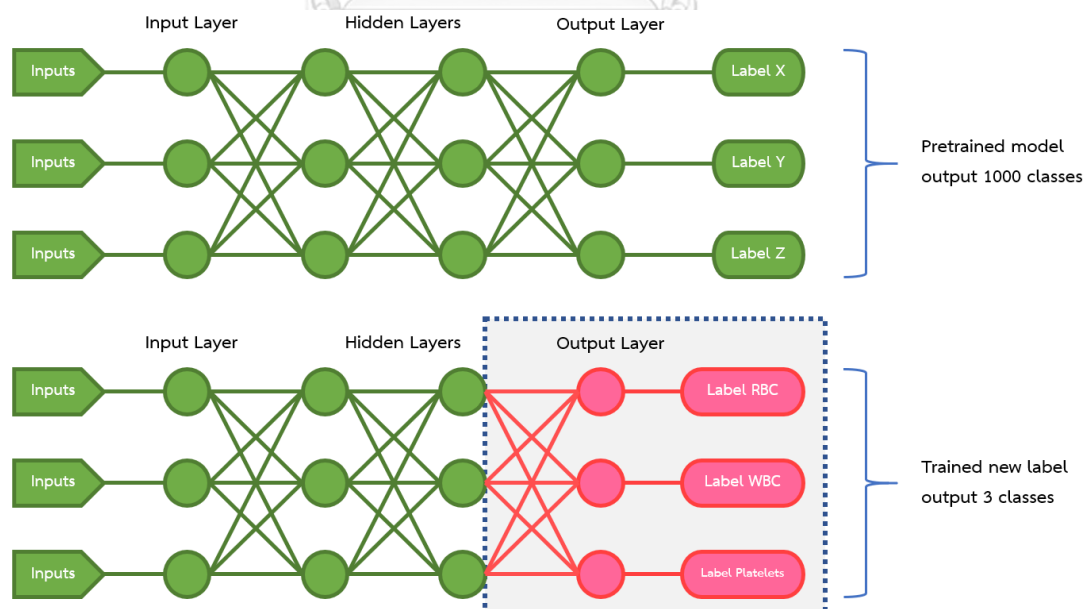
ภาพที่ 68 การแสดงวิธีคำนวณพื้นที่ Intersection over union (IoU)



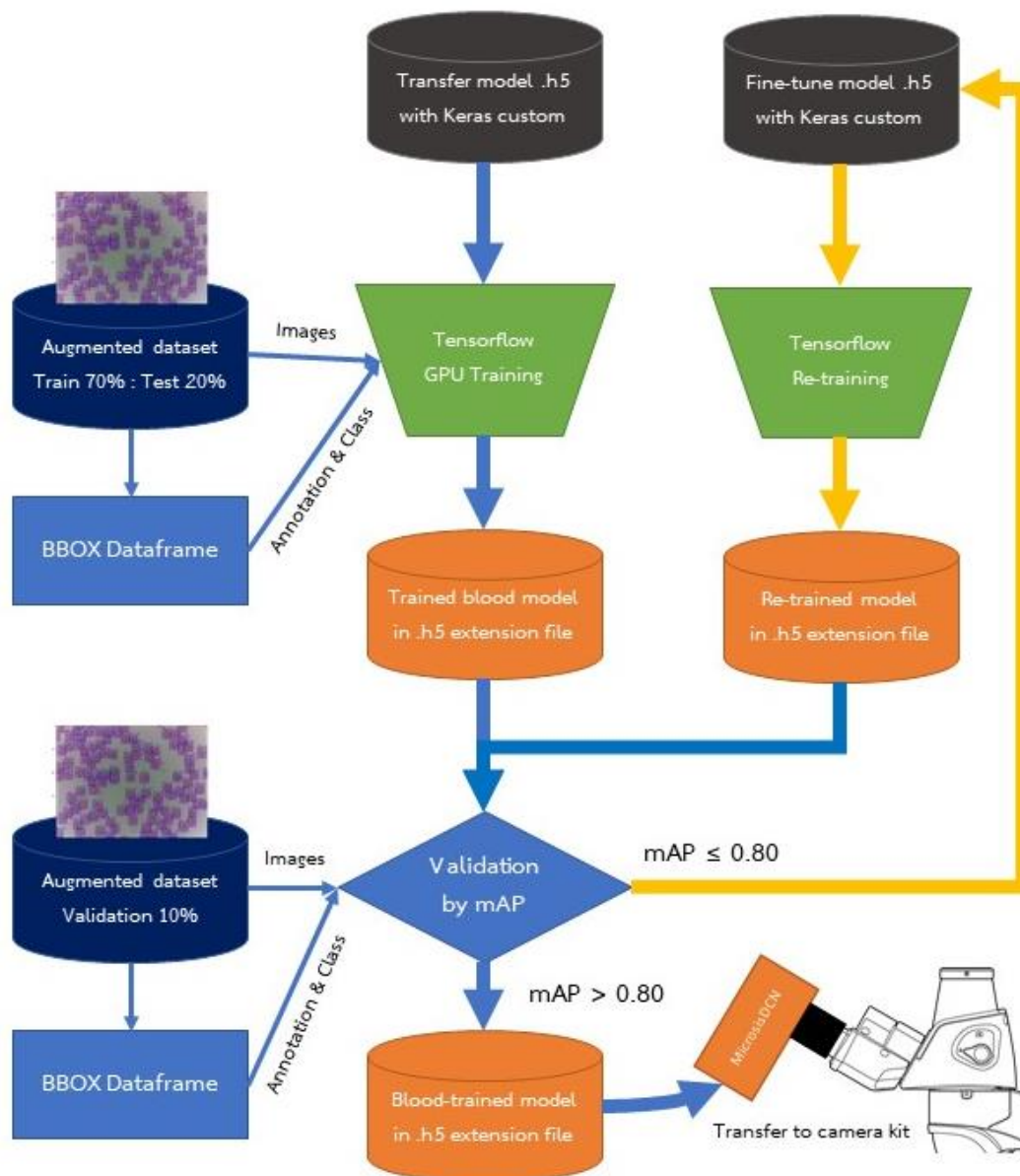
ภาพที่ 69 โครงสร้างภายในโครงข่ายประสาท RetinaNet โดย Tsung-Yi Lin และคณะ [38]

จากตารางที่ 13 จะเห็นได้ว่าความแม่นยำในการพยากรณ์ของ RetinaNet สูงกว่า Faster R-CNN ที่ผู้วิจัยได้พัฒนาก่อนหน้านี้ สามารถดูได้เพิ่มเติมใน ภาคผนวก ก. บทความวิจัยที่ได้ตีพิมพ์ใน iEECON 2020 ซึ่งจะเป็นการฝึกโดยใช้ Tensorflow โดยตรง ปัญหาที่พบผู้วิจัยไม่สามารถเข้าถึงโครงสร้างภายในโครงข่ายประสาทเพื่อทำการปรับแต่งให้เหมาะสมต่อระบบ จึงจำเป็นต้องเปลี่ยนโครงข่ายประสาทที่สามารถพัฒนาได้บน Keras จึงทำให้ผู้วิจัยเลือกโมเดลโครงข่ายประสาท RetinaNet มาใช้ในการพัฒนาโครงข่ายประสาทภายในชุดกล้อง MicrosisDCN ด้วยการนำ Transfer learning คือการนำโมเดลโครงข่ายประสาทที่ถูกฝึกและปรับปรุงลำดับชั้นให้เหมาะสมต่อ

การนำไปใช้งาน เช่น Conv2D, Dense, Dropout และอื่น ๆ มาฝึกด้วยชุดข้อมูลภาพใหม่ที่ยังไม่ได้ รับการฝึกมาก่อน จนมีความสามารถในการพยากรณ์ข้อมูลในคลาสที่ถูกฝึกได้ กับชุดข้อมูลภาพถ่าย เซลล์เม็ดเลือดจากกล้องจุลทรรศน์ (Blood cell dataset) ที่ผู้วิจัยสร้างขึ้นก่อนหน้านี้ ไปฝึกบน เครื่องคอมพิวเตอร์ส่วนบุคคลด้วยคำสั่ง Keras RetinaNet ตัวอย่างเช่น python keras\_retinanet/bin/train.py --weights ./model/blood\_model.h5 --freeze-backbone --random-transform --multi-gpu X --multi-gpu-force --batch-size XX --epochs XXX --steps XXX --weighted-average --compute-val-loss csv annotations.csv classes.csv ซึ่งใน ไฟล์คำสั่งนั้นจะถูกปรับแต่งให้เหมาะสมกับการฝึกด้วย Multiple-GPU เนื่องจากคำสั่ง เดิมนั้นรองรับเฉพาะ CPU และ GPU แบบเดี่ยวเท่านั้น ทำให้ได้รับโมเดลโครงข่ายประสาทรูปของ ไฟล์ .h5 ซึ่งเป็นสกุลไฟล์ในรูปแบบ Hierarchical Data Format (HDF) ที่ฝึกเสร็จสิ้นไปติดตั้งบน ระบบสมองกลฝังตัว (Embedded system) ของชุดกล้อง EagleEYE Smart Camera ทำงานด้วย ระบบปฏิบัติการ Raspbian ที่ได้รับการติดตั้งชุดคำสั่งที่จำเป็น เช่น OpenCV, Tensorflow, PiCamera และอื่น ๆ สำหรับการเรียกใช้งานโมเดลโครงข่ายประสาทสามารถทำงานได้บนอุปกรณ์ ดังกล่าวผ่านการเปิดหน้าต่าง Terminal บนระบบปฏิบัติการ จากนั้นจึงเข้าถึง Virtual environment ที่ได้ติดตั้งไว้ด้วยคำสั่ง `~/bashrc` และ `workon cv` จากนั้นจึงเรียกใช้งานคำสั่ง ภาษา Python เช่น `python blood_detection.py` บนการทำงานของชุดกล้องต่อไป



ภาพที่ 70 แนวคิดของการทำ Transfer learning ของโครงข่ายประสาทของระบบที่เสนอ



ภาพที่ 71 ขั้นตอนการฝึกโมเดลโครงข่ายประสาทสำหรับตรวจจับเซลล์ขนาดเล็ก

ขั้นตอนการฝึกโมเดลโครงข่ายประสาท จะนำชุดข้อมูลภาพถ่ายเซลล์เม็ดเลือดจากกล้องจุลทรรศน์ที่ผ่านการขยายชุดภาพ และติดป้ายบ่งบอกคลาสของเซลล์ขนาดเล็กจำนวน 3 คลาส ประกอบด้วย เม็ดเลือดแดง ชื่อคลาส RBC เม็ดเลือดขาว ชื่อคลาส WBC และเกล็ดเลือด ชื่อคลาส Platelets แบ่งออกเป็นชุดสำหรับการฝึก (Training set) ร้อยละ 70 ชุดสำหรับทดสอบ (Test set) ร้อยละ 20 และชุดตรวจสอบความถูกต้อง (Validation set) ร้อยละ 10 โดยที่ชุดสำหรับการฝึกและชุดสำหรับทดสอบจะถูกนำไปใช้ในขั้นตอนการฝึกโมเดลโครงข่ายประสาทด้วย Tensorflow ผ่าน Multiple-GPU และชุดตรวจสอบความถูกต้องจะนำไปใช้ในการขั้นตอนการประเมินผล หากค่า

Mean Average Precision (mAP) หรือความแม่นยำในการพยากรณ์ของทั้ง 3 คลาสมีค่าน้อยกว่าร้อยละ 80 จะต้องปรับแต่งและฝึกใหม่ให้มีค่าน้ำหนักที่เหมาะสมต่อชุดข้อมูลภาพเซลล์เม็ดเลือด และทำการฝึกโมเดลใหม่ซ้ำจนกว่าความแม่นยำในการพยากรณ์ทั้ง 3 คลาสมีค่ามากกว่าร้อยละ 80 จึงจะสามารถนำโมเดลโครงข่ายประสาทไปใช้งานได้

ชุดคำสั่งจะเข้าถึงตำแหน่งไฟล์ภาพ JPG และไฟล์ XML เก็บพิกัดพื้นที่ปิดล้อมแบบ Pascal VOC bounding box แปลงให้อยู่ในรูปของ BBOX Dataframe ประกอบไปด้วยตำแหน่งไฟล์ภาพที่เก็บในแฟ้ม พิกัดพื้นที่สี่เหลี่ยมปิดล้อมรอบวัตถุ  $X_{min}$ ,  $Y_{min}$ ,  $X_{max}$  และ  $Y_{max}$  และชื่อคลาสที่ถูกติดป้าย จากนั้นจึงแปลงเป็นไฟล์ชื่อว่า annotations.csv สำหรับใช้ในการฝึกโมเดลโครงข่ายประสาท และไฟล์ชื่อว่า classes.csv ใช้สำหรับบ่งบอกชื่อคลาสด้วยตัวเลข เช่น WBC = 0, Platelets = 1 และ RBC = 2 สำหรับใช้ระบุว่าวัตถุนั้นชื่อคลาสอะไร และนำตัวเลขดังกล่าวไปใช้กับชุดคำสั่งพยากรณ์ภาพต่อไป แสดงให้เห็นในภาพที่ 72

```
In [3]: annotations = pd.read_csv("annotations.csv")
        annotations.head()
```

```
Out[3]:
```

	./images/2020-04-16_20-36-04_CU.jpg	1087	85	1156	151	RBC
0	./images/2020-04-16_20-36-04_CU.jpg	909	17	991	89	RBC
1	./images/2020-04-16_20-36-04_CU.jpg	1167	53	1240	124	RBC
2	./images/2020-04-16_20-36-04_CU.jpg	1162	130	1233	193	RBC
3	./images/2020-04-16_20-36-04_CU.jpg	1200	143	1264	215	RBC
4	./images/2020-04-16_20-36-04_CU.jpg	1186	283	1249	352	RBC

```
In [4]: classes = pd.read_csv("classes.csv")
        classes.head()
```

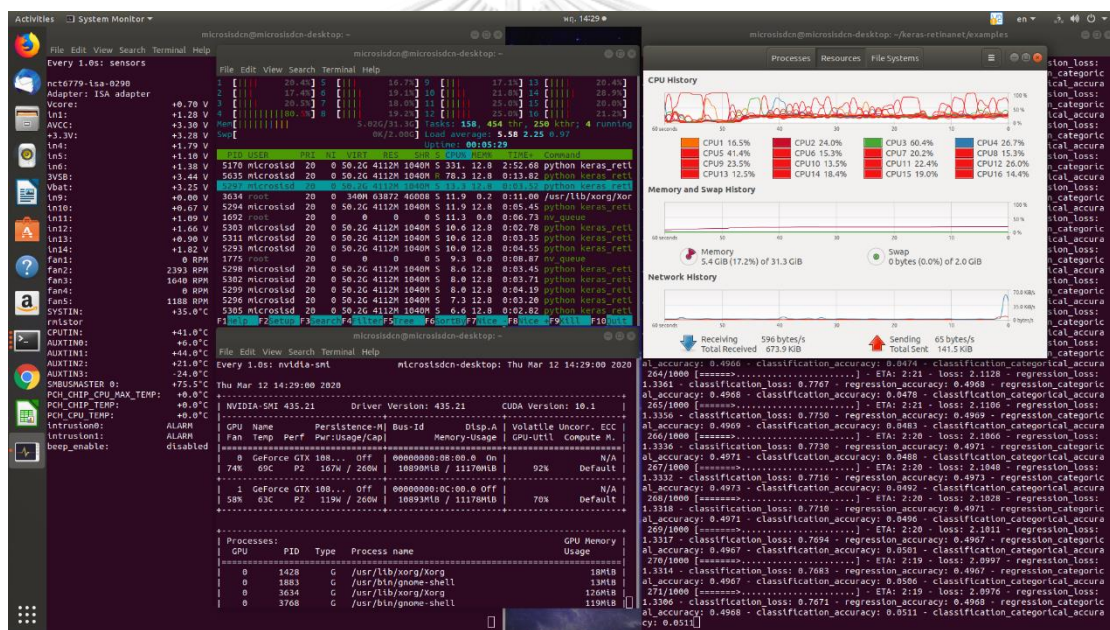
```
Out[4]:
```

	WBC	0
0	Platelets	1
1	RBC	2

ภาพที่ 72 การแปลงข้อมูล BBOX Dataframe จากไฟล์ XML ไปเป็น CSV

เมื่อได้ไฟล์ annotations.csv และ classes.csv แล้วจะต้องตรวจสอบความถูกต้องของการบรรณนิทัศน์ตำแหน่งพื้นที่สี่เหลี่ยมปิดล้อมก่อนทำการฝึกโมเดลโครงข่ายประสาท สามารถทำได้ 2 วิธี คือ วิธีแรกตรวจสอบตั้งแต่การสร้างชุดข้อมูลและการติดป้าย ซึ่งค่าของ  $X_{min}$ ,  $Y_{min}$ ,  $X_{max}$  และ  $Y_{max}$  จะต้องไม่ติดลบ หรือมีค่ามากกว่าความกว้างและความยาวของภาพนั้น ๆ ประกอบกับการตรวจสอบด้วยชุดคำสั่งซ้ำ วิธีการที่สอง คือการใช้ชุดคำสั่ง Debugging ของคลังชุดคำสั่งในการตรวจสอบความถูกต้อง ตัวอย่างเช่น python keras\_retinanet/bin/debug.py csv annotations.csv classes.csv หากตำแหน่งใดที่ทำการบรรณนิทัศน์ถูกต้องจะแสดงเป็นสีเขียวในภาพที่แสดงขึ้นในหน้าจอ

กระบวนการฝึกโมเดลโครงข่ายประสาทนั้น จะต้องเตรียมระบบปฏิบัติการให้พร้อมต่อการใช้งาน จะต้องมีการติดตั้งชุดคำสั่งองค์ประกอบที่เกี่ยวข้อง และติดตั้งองค์ประกอบพื้นฐานที่ทำให้เข้าใช้งาน GPU สามารถดูเพิ่มเติมใน ภาคผนวก ข. ชุดคำสั่งติดตั้ง CUDA TOOLKIT และ CUDNN กับ ภาคผนวก ค. ชุดคำสั่งติดตั้ง OPENCV และ TENSORFLOW 1.15 ผู้วิจัยได้อธิบายขั้นตอนไว้โดยละเอียด จากนั้นจึงเข้าสู่ขั้นตอนการฝึกโมเดลโครงข่ายประสาท และการปรับปรุงลำดับลำดับชั้นภายในโครงข่ายประสาท สามารถใช้ชุดคำสั่งที่อยู่ภายใน Tensorflow และ Keras ที่สามารถปรับปรุงได้ทั้ง VGG16, VGG19, Inception v3, Keras และอื่น ๆ จากส่วนหนึ่งของชุดคำสั่งที่ใช้ในการสร้างโมเดลโครงข่ายประสาทจากรูปแบบ RetinaNet จะแยกเป็นส่วนของโครงข่ายประสาทหลัก ResNet50 และ Feature Pyramid Network (FPN) ในการดึงคุณลักษณะของภาพจากชั้น feedforward และพยากรณ์ตำแหน่งของคลาส



ภาพที่ 73 การฝึกโมเดลโครงข่ายประสาทด้วย Nvidia GeForce GTX 1080Ti 2 ชุด

คอมพิวเตอร์ที่ใช้ประกอบด้วยระบบปฏิบัติการ : Ubuntu 18.04 LTS and Microsoft Windows 10 หน่วยประมวลผลกลาง CPU AMD Ryzen 7 Series 3800X 3.9Ghz 8 cores 14 threads หน่วยประมวลผลกราฟิก GPU NVIDIA GeForce GTX 1080Ti 11Gb จำนวน 2 ชุดบอร์ดหลัก Motherboard Asrock AM4 Asrock X470 Master SLI/AC หน่วยความจำหลัก RAM KLEVV CRAS X RGB DDR4 3200Mhz ความจุ 32GB หน่วยความจำสำรอง ฮาร์ดดิสก์ WD BLACK WD1003FZEX ความจุ 1TB และ SSD WD BLACK SN750 NVMe M.2 ความจุ 512GB



คำสั่งชุดคำสั่งสำหรับฝึกโมเดลโครงข่ายประสาทจะประกอบด้วย 3 ส่วน ได้แก่ ชุดคำสั่งสำหรับฝึก train.py ตามด้วย arguments สำหรับกำหนดพารามิเตอร์เฉพาะ ประกอบด้วย --weights ./models/blood\_model.h5 กำหนดตำแหน่งของโมเดลภายในแฟ้มที่ต้องการนำมาฝึก --tensorboard-dir ./tensorboard กำหนดตำแหน่งแฟ้มเก็บเหตุการณ์ที่เกิดขึ้นระหว่างการฝึก (Log files) --freeze-backbone กำหนดให้ฝึกเจาะจงไปที่ Backbone ของโครงข่ายประสาท ซึ่งจะเป็น ResNet50 --random-transform กำหนดให้เข้าถึงตำแหน่งไฟล์ภาพที่เก็บในแฟ้มและพิกัดพื้นที่สี่เหลี่ยมปิดล้อมรอบวัตถุอย่างสุ่ม ไม่เป็นลำดับที่ระบุในไฟล์ CSV --epochs XXX กำหนดจำนวนรอบการฝึกของข้อมูลที่นำมาฝึกโมเดลจนครบ --steps XXX กำหนดจำนวนขั้นย่อยของรอบการฝึก --weighted-average กำหนดให้คำนวณค่า mAP จากการเฉลี่ยค่าน้ำหนักของความเที่ยงตรงในแต่ละคลาส --compute-val-loss กำหนดให้คำนวณ Validation loss ในระหว่างการฝึก และ csv annotations.csv classes.csv กำหนดตำแหน่งของไฟล์ CSV ของชุดข้อมูลภาพ นอกจากนี้ยังมี arguments เพิ่มเติมสำหรับการเข้าถึงการฝึกด้วย GPU ประกอบด้วย --multi-gpu X กำหนดจำนวน GPU ที่มีในคอมพิวเตอร์ ซึ่งจะต้องตรงกันกับจำนวนที่แสดงในชุดคำสั่ง nvidia-smi ใช้แสดงข้อมูลของหน่วยประมวลผลกราฟิก --multi-gpu-force กำหนดให้ Tensorflow ใช้ GPU ในการฝึก --batch-size X กำหนดการแบ่งกลุ่มชุดข้อมูลสำหรับการฝึกมีจำนวนเท่ากับ GPU

ส่วนถัดมาชุดคำสั่งสำหรับแปลงโมเดลโครงข่ายประสาทให้เป็นโมเดลอนุมาน (Inference model) ที่สามารถนำไปใช้พยากรณ์ข้อมูลที่ไม่เคยเจอในการฝึกมาก่อนได้ convert\_model.py ตามด้วย arguments สำหรับกำหนดพารามิเตอร์ประกอบด้วย ./model/blood\_model\_{จำนวน epoch}.h5 กำหนดตำแหน่งของโมเดลภายในแฟ้มที่ถูกฝึกเสร็จตามรอบ epoch และ ./model/blood\_model\_inference.h5 กำหนดตำแหน่งที่ต้องการบันทึกโมเดลอนุมานสำหรับนำไปใช้ในชุดทดลอง สุดท้ายชุดคำสั่งประเมินความสามารถของโมเดลอนุมานกับชุดตรวจสอบความถูกต้อง (Validation set) เพื่อดูค่า Mean Average Precision (mAP) หรือความแม่นยำในการพยากรณ์ของทั้ง 3 คลาส evaluate.py ตามด้วย arguments สำหรับกำหนดพารามิเตอร์ประกอบด้วย csv annotations.csv classes.csv กำหนดตำแหน่งของไฟล์ CSV ของชุดตรวจสอบความถูกต้อง และ ./model/blood\_model\_inference.h5 กำหนดตำแหน่งที่ต้องการบันทึกโมเดลอนุมาน

## บทที่ 4 ผลการทำงานของระบบ

### 4.1 การขยายชุดภาพและพารามิเตอร์ของโมเดล

จากการใช้คลังชุดคำสั่ง `imgaug` ในการขยายชุดภาพ (Image augmentation) มาใช้สร้างชุดข้อมูลภาพถ่ายเซลล์เม็ดเลือดจากกล้องจุลทรรศน์ โดยการดัดแปลงจากภาพเดิมที่มีอยู่จากการติดป้าย (Labeling) จากปัญหาที่พบในการขยายชุดภาพนั้น สามารถขยายชุดภาพได้ แต่ไม่สามารถขยายชุดไฟล์ XML เก็บพิกัดพื้นที่ปิดล้อมแบบ Pascal VOC bounding box (Pascal VOC BBOX) ตำแหน่งตามชุดภาพที่เพิ่มขึ้นได้ ทางผู้วิจัยจึงได้พัฒนาชุดคำสั่งในการบันทึกไฟล์บรรณนิพจน์ให้เท่ากับจำนวนชุดภาพที่ถูกขยาย (CSV to XML dataframe)

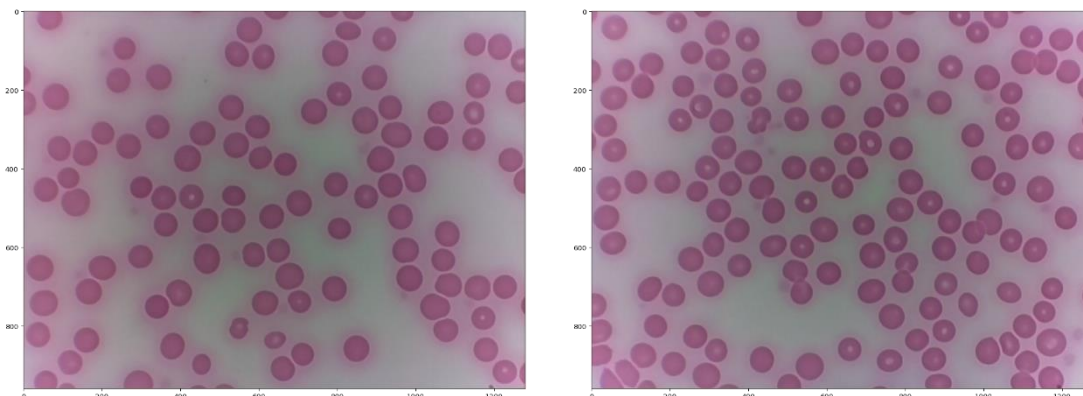
ชุดคำสั่งจะทำการตรวจสอบภาพในแฟ้ม datasets ที่ชื่อว่า images และไฟล์ภาพเป็นนามสกุล .jpg ด้วยฟังก์ชัน `glob.glob('PATHFILE')` จะทำการค้นหาไฟล์ตามเงื่อนไขที่ระบุไว้จนตรงทั้งแฟ้มใน PATHFILE คือ images เช่นเจอไฟล์ 2020-04-18\_15-32-29\_CU.jpg ชุดคำสั่งจะอ่านไฟล์ภาพด้วยชุดคำสั่ง `imageio.imread(file)` จากคลังชุดคำสั่ง `imageio` ที่จะทำให้ค่าออกมาเป็น NumPy array ชนิดตัวแปรแบบ `uint8` ซึ่งปกติตัวแปรชนิด NumPy array จะเป็น `int64` หรือ `float64` ประกอบด้วยพารามิเตอร์ ได้แก่ ความสูง (Height) ความกว้าง (Width) ช่อง (Channel) ในระบบสี RGB (Red, Green and Blue) จะมีค่าเป็น 3 หรือ RGB Colorspace แต่อีกวิธีการหนึ่งที่สามารถใช้งานกรณีที่น่าจะเข้าคลังชุดคำสั่ง `OpenCV` จะใช้เป็นชุดคำสั่ง `cv2.imread(file)` แทนชุดคำสั่ง `images.append()` จะนำข้อมูลภาพที่อ่านได้เก็บไว้ในตัวแปรชนิด Array ชื่อ images ต่อไปในระหว่างที่ชุดคำสั่งทำงานจะแสดงชื่อไฟล์ภาพภายในแฟ้ม images และ dimension ของภาพที่อ่านได้ และจำนวนภาพที่อยู่ในแฟ้มทั้งหมด แสดงผลลัพธ์ในภาพที่ 74 ผู้ใช้สามารถเรียกภาพที่นำเข้ามาแสดงผลด้วยชุดคำสั่ง `ia.imshow(images[x])` โดยที่ x คือตำแหน่งของตัวแปร images หากในแฟ้ม images มี 50 ภาพ ค่า x จะมีตั้งแต่ 0 ถึง 49 ภาพที่ 75 แสดงตัวอย่างภาพที่ `images[2]` และ `images[7]`

```

Filename : 2020-04-18_15-32-23_CU.jpg Dimension is (960, 1280, 3)
Filename : 2020-04-18_15-32-17_CU.jpg Dimension is (960, 1280, 3)
Filename : 2020-04-18_15-36-55_CU.jpg Dimension is (960, 1280, 3)
Filename : 2020-04-18_15-32-32_CU.jpg Dimension is (960, 1280, 3)
Filename : 2020-04-18_15-36-58_CU.jpg Dimension is (960, 1280, 3)
Filename : 2020-04-18_15-31-59_CU.jpg Dimension is (960, 1280, 3)
Filename : 2020-04-18_15-32-03_CU.jpg Dimension is (960, 1280, 3)
Filename : 2020-04-18_15-31-14_CU.jpg Dimension is (960, 1280, 3)
Filename : 2020-04-18_15-32-06_CU.jpg Dimension is (960, 1280, 3)
Original datasets have 100 images

```

ภาพที่ 74 ผลการทำงานชุดคำสั่งตรวจสอบภาพ และเก็บเข้าตัวแปร images ตามจำนวนภาพ



ภาพที่ 75 ผลการทำงานแสดงภาพที่อยู่ในตัวแปรชนิด Array ชื่อว่า images

สำหรับชุดคำสั่งนำเข้าไฟล์บรรณนิทัศน์ XML จะคล้ายกับชุดคำสั่งก่อนหน้านี้ ซึ่งจะทำการตรวจสอบไฟล์ในแฟ้ม datasets ที่ชื่อว่า images/outputs และไฟล์ภาพเป็นนามสกุล .xml ด้วยฟังก์ชัน glob.glob ('PATHFILE') จะทำการค้นหาไฟล์ตามเงื่อนไขที่ระบุไว้จนตรงทั้งแฟ้มใน PATHFILE คือ images/outputs เช่นเจอไฟล์ 2020-04-18\_15-31-12\_CU.xml เมื่อชุดคำสั่งทำงานจะแสดงชื่อไฟล์ภาพภายในแฟ้ม images/outputs และจำนวนไฟล์ XML ที่อยู่ในแฟ้มทั้งหมด แสดงผลลัพธ์ในภาพที่ 76

```

outputs/2020-04-18_15-37-21_CU.xml
outputs/2020-04-18_15-38-26_CU.xml
outputs/2020-04-18_15-37-58_CU.xml
outputs/2020-04-18_15-32-08_CU.xml
outputs/2020-04-18_15-31-59_CU.xml
outputs/2020-04-18_15-38-57_CU.xml
Original XML have 100 files

```

ภาพที่ 76 ผลการทำงานแสดงไฟล์บรรณนิทัศน์ XML ที่อยู่ในแฟ้ม images/outputs

หลังจากตรวจสอบไฟล์ภาพและไฟล์บรรณนิทัศน์เรียบร้อยแล้ว ชุดคำสั่งถัดมาส่วนของชุดคำสั่งแปลงข้อมูลในไฟล์บรรณนิทัศน์ XML ไปเป็น CSV และ Dataframe โดยจะทำการอ่านไฟล์บรรณนิทัศน์ XML ทั้งหมดในแฟ้ม datasets ที่ชื่อว่า images/outputs และแยกข้อมูลโดยการค้นหาแท็ก <object> ... </object> ตามจำนวนการติดป้ายวัตถุภายในภาพ เช่น เซลล์ขนาดเล็กถูกติดป้าย 100 ป้าย จะทำการวนลูป 100 ครั้งจนครบ และภายในรอบ (Loop) จะสกัดข้อมูลที่สำคัญประกอบด้วย <filename> ... </filename> เป็นแท็กระบุชื่อไฟล์ภาพ <size> ... </size> เป็น

แท็กที่เก็บขนาดของภาพ ความกว้าง `<width> ... </width>` ความสูง `<height> ... </height>` `<name> ... </name>` เป็นแท็กชื่อตามคลาสที่ติดป้าย เช่น RBC, WBC และ Platelets `<bndbox> ... </bndbox>` เป็นแท็กระบุพิกัดพื้นที่สี่เหลี่ยมปิดล้อมด้วยพิกัด X และ Y ในภาพ ประกอบด้วย `<xmin> ... </xmin>` `<ymin> ... </ymin>` `<xmax> ... </xmax>` `<ymax> ... </ymax>` จัดลงในตัวแปรที่แบ่งออกเป็นคอลัมภ์ คือ ['filename', 'width', 'height', 'class', 'xmin', 'ymin', 'xmax', 'ymax'] บันทึกเป็นไฟล์ชื่อ `csv_labels.csv` และสร้างเป็น dataframe จากการอ่านไฟล์ CSV แสดงผลลัพธ์ในภาพที่ 77

Out[7]:

	filename	width	height	class	xmin	ymin	xmax	ymax
0	2020-04-18_15-32-39_CU.jpg	1280	960	RBC	1212	44	1274	122
1	2020-04-18_15-32-39_CU.jpg	1280	960	RBC	1198	116	1268	186
2	2020-04-18_15-32-39_CU.jpg	1280	960	RBC	1142	2	1216	74
3	2020-04-18_15-32-39_CU.jpg	1280	960	RBC	1116	60	1194	142
4	2020-04-18_15-32-39_CU.jpg	1280	960	RBC	1076	116	1152	188
...	...	...	...	...	...	...	...	...
13471	2020-04-18_15-38-57_CU.jpg	1280	960	RBC	616	792	694	870
13472	2020-04-18_15-38-57_CU.jpg	1280	960	RBC	196	808	270	880
13473	2020-04-18_15-38-57_CU.jpg	1280	960	RBC	204	872	278	940
13474	2020-04-18_15-38-57_CU.jpg	1280	960	RBC	144	892	212	956
13475	2020-04-18_15-38-57_CU.jpg	1280	960	RBC	96	852	166	930

13476 rows x 8 columns

ภาพที่ 77 ข้อมูลป้ายตามจำนวนวัตถุที่ถูกสกัดจากไฟล์บรรณนิทัศน์ไปเป็น CSV และ Dataframe

Dataframe ที่ได้จะถูกนำไปใช้ในการขยายชุดภาพที่มีฟังก์ชันชื่อว่า `image_aug` และชุดคำสั่งในการบันทึกไฟล์บรรณนิทัศน์ให้เท่ากับจำนวนชุดภาพที่ถูกขยายที่มีฟังก์ชันชื่อว่า `csv_to_xml` และภายในฟังก์ชัน `image_aug` จะรับ 5 arguments ประกอบด้วย `df` คือ Dataframe ของจากการอ่านไฟล์ CSV ที่ถูกแปลงก่อนหน้านี้ `images_path` คือ แฟ้มที่เก็บไฟล์ภาพชุดข้อมูลต้นฉบับ `aug_images_path` คือ แฟ้มปลายทางสำหรับบันทึกชุดภาพที่ถูกขยายตามพารามิเตอร์ที่ถูกกำหนดไว้ `images_prefix` คือ คำอุปสรรคหน้าชื่อไฟล์ภาพที่ผ่านการขยายชุดภาพ และ `aug_params` คือ พารามิเตอร์สำหรับคลังชุดคำสั่ง `imgaug` ซึ่งกระบวนการขยายชุดภาพจะแปรผันตามพารามิเตอร์ที่กำหนดไว้ โดยหนึ่งภาพจะสามารถเกิด augmentation 2 แบบจาก 9 แบบอย่างสุ่ม จากคำสั่ง `iaa.SomeOf(2, [พารามิเตอร์ที่กำหนด])` เช่น ภาพที่เกิดจาก

augmentation ด้วยฟังก์ชัน GaussianBlur และ MultiplyBrightness เป็นต้น ภาพที่ได้จะถูกบันทึกเป็นไฟล์ภาพใหม่ที่มีคำอุปสรรคนำหน้า เช่น aug0\_2020-04-18\_15-31-12\_CU.jpg และตำแหน่งพิกัดพื้นที่ปิดล้อมรอบวัตถุได้ไฟล์บรรณนิทัศน์ใหม่ชื่อว่า aug\_csv\_labels.csv ที่สัมพันธ์กับไฟล์ภาพที่ถูกขยายทั้งหมดภายในฟังก์ชัน csv\_to\_xml จะรับ 4 arguments ประกอบด้วย csv\_path คือตำแหน่งไฟล์บรรณนิทัศน์ aug\_csv\_labels.csv images\_path คือแฟ้มที่เก็บชุดภาพที่ถูกขยายเรียบร้อยแล้ว labels\_path คือแฟ้มปลายทางสำหรับบันทึกไฟล์ XML ใหม่ตามจำนวนภาพที่ถูกขยาย และ folder กำหนดชื่อแฟ้มที่เก็บชุดข้อมูลภาพในแท็ก <folder> ... </folder> ซึ่งกระบวนการในฟังก์ชันนี้ จะทำการบันทึกไฟล์บรรณนิทัศน์ให้เท่ากับจำนวนชุดภาพที่ถูกขยายทั้งหมดในแฟ้มปลายทาง

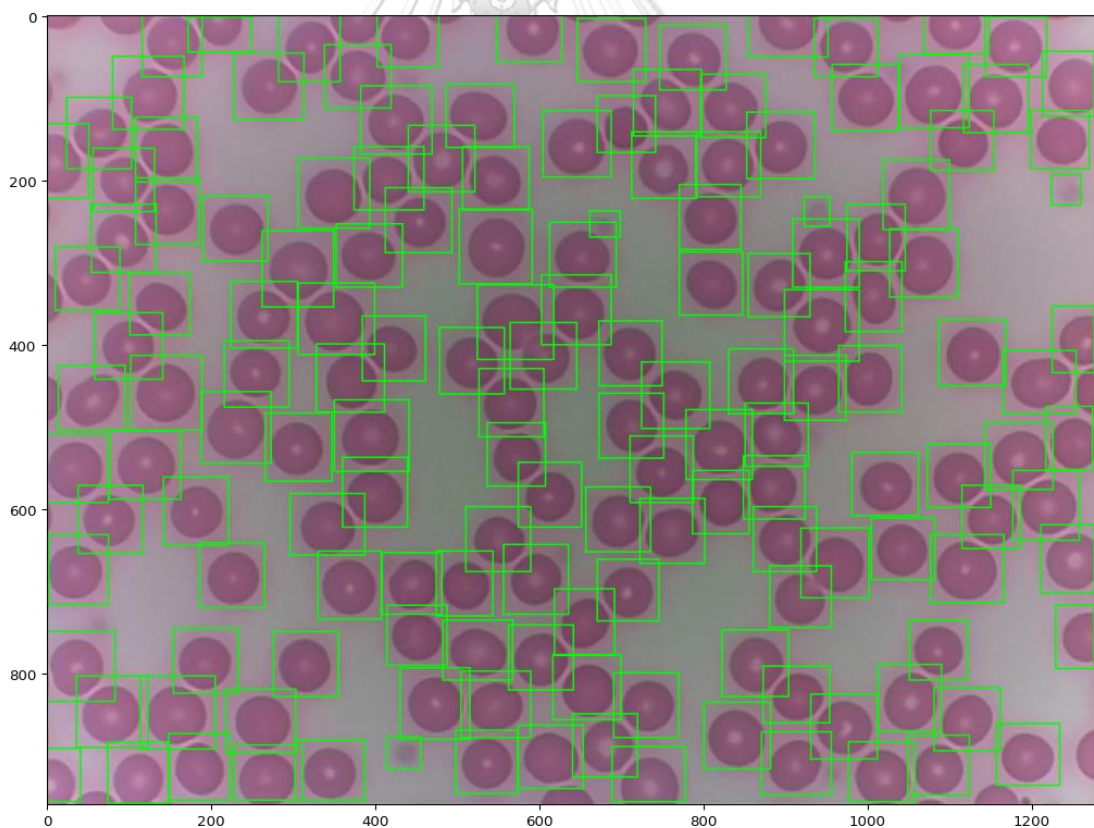


```
# การกำหนดพารามิเตอร์สำหรับคลังชุดคำสั่ง imgaug เพื่อใช้ในฟังก์ชัน img_aug
aug_params = iaa.SomeOf(2, [
    iaa.Grayscale(alpha=1.0),
    iaa.GaussianBlur(sigma=(1.0, 3.0)),
    iaa.AdditiveGaussianNoise(scale=(0.03*255, 0.05*255)),
    iaa.MultiplyHueAndSaturation(mul_hue=(0.5, 1.5)),
    iaa.MultiplyHueAndSaturation(mul_saturation=(0.5, 1.5)),
    iaa.AddToHueAndSaturation((-10, 10), per_channel=True),
    iaa.MultiplyBrightness((0.9, 1.1)),
    iaa.GammaContrast((0.5, 1.0)),
    iaa.SigmoidContrast(gain=(3, 10), cutoff=(0.4, 0.6))])
```

สำหรับการขยายชุดภาพและบันทึกไฟล์บรรณนิทัศน์อัตโนมัติ ผู้ใช้สามารถกำหนดจำนวนรอบในตัวแปร augment\_loop เช่น 5 รอบ หากชุดข้อมูลภาพมี 1,000 ภาพ ชุดข้อมูลที่ถูกขยายจะมีทั้งหมด 5,000 ภาพ ถูกบันทึกไว้ในแฟ้ม aug\_images และไฟล์ XML ที่ได้จะถูกบันทึกในแฟ้ม outputs ภายในแฟ้ม aug\_images อีกทีหนึ่ง ผลลัพธ์ของการใช้ชุดคำสั่งดังกล่าวจะอยู่ในหัวข้อผลลัพธ์การขยายชุดภาพเซลล์เม็ดเลือดแดง (RBC) และ ผลลัพธ์การขยายภาพเซลล์เม็ดเลือดขาว (WBC) และเกล็ดเลือด (Platelets) แสดงผลในภาพที่ 78 ถึง 85

#### 4.1.1 ผลลัพธ์การขยายชุดภาพเซลล์เม็ดเลือดแดง (RBC)

```
# ชุดคำสั่งแสดงผลภาพจากไฟล์ .jpg และตำแหน่งพื้นที่ปิดล้อมรอบวัตถุจากไฟล์ .xml
images_df.to_csv('csv_labels.csv', index=False)
grouped = images_df.groupby('filename')
group_df = grouped.get_group('2020-04-18_15-32-39_CU.jpg')
group_df = group_df.reset_index()
group_df = group_df.drop(['index'], axis=1)
bb_array = group_df.drop(['filename', 'width', 'height', 'class'], axis=1).values
image = imageio.imread('images/2020-04-18_15-32-39_CU.jpg')
bbox = BoundingBoxesOnImage.from_xyxy_array(bb_array, shape=image.shape)
ia.imshow(bbox.draw_on_image(image, size=2))
```



ภาพที่ 78 การแสดงตำแหน่งพื้นที่สี่เหลี่ยมปิดล้อมบนภาพเซลล์เม็ดเลือดแดงภายในชุดภาพ



ภาพที่ 79 ตัวอย่างชุดข้อมูลภาพเซลล์เม็ดเลือดแดง จากกล้องจุลทรรศน์ก่อนได้รับการขยายชุดภาพ



ภาพที่ 80 ตัวอย่างชุดข้อมูลภาพเซลล์เม็ดเลือดแดง จากกล้องจุลทรรศน์หลังได้รับการขยายชุดภาพ



ภาพที่ 81 ตัวอย่างไฟล์บรรณนิทัศน์ XML ที่มีข้อมูลการติดป้ายวัตถุของชุดภาพที่ได้รับการขยาย (1)

ตารางที่ 14 ผลแสดงพิกัดของพื้นที่สี่เหลี่ยมปิดล้อมต่อวัตถุภายในชุดข้อมูลภาพแบบ CSV (1)

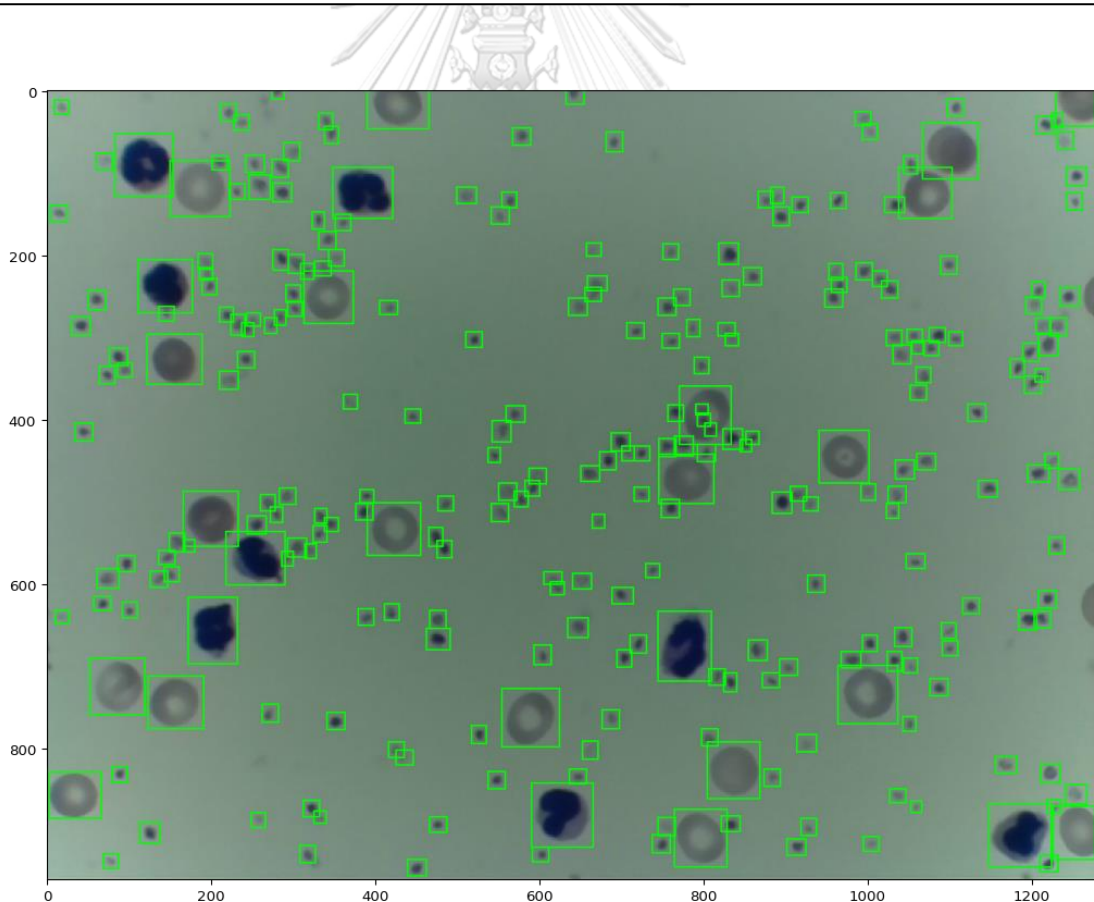
Filename	Width	Height	Class	X <sub>min</sub>	Y <sub>min</sub>	X <sub>max</sub>	Y <sub>max</sub>
2020-04-18_15-32-39_CU.jpg	1280	960	RBC	1212	44	1274	122
2020-04-18_15-32-39_CU.jpg	1280	960	RBC	1198	116	1268	186
2020-04-18_15-32-39_CU.jpg	1280	960	Platelets	414	878	454	916
2020-04-18_15-32-39_CU.jpg	1280	960	Platelets	922	221	952	256
2020-04-18_15-32-39_CU.jpg	1280	960	Platelets	661	238	697	269
...	...	...	...	...	...	...	...
2020-04-18_15-32-39_CU.jpg	1280	960	Platelets	1223	194	1258	230
2020-04-18_15-38-31_CU.jpg	1280	960	RBC	0	652	62	724
2020-04-18_15-38-31_CU.jpg	1280	960	RBC	0	580	42	662
2020-04-18_15-38-31_CU.jpg	1280	960	RBC	14	464	96	550
2020-04-18_15-38-31_CU.jpg	1280	960	RBC	2	402	66	486
2020-04-18_15-32-19_CU.jpg	1280	960	Platelets	433	785	466	830



```
<!--รายละเอียดตัวอย่างไฟล์บรรณนิทัศน์ชุดข้อมูลภาพ RBC ที่ถูกขยาย -->
<?xml version="1.0" ?>
<annotation>
<folder>MicrosisDCN_Datasets</folder>
<filename>aug0_2020-04-18_15-38-22_CU.jpg</filename>
<path>aug_images/aug0_2020-04-18_15-38-22_CU.jpg</path>
<source>
  <database>Engineering Chulalongkorn</database>
</source>
<size>
  <width>1280</width>
  <height>960</height>
  <depth>3</depth>
</size>
  <segmented>0</segmented>
  <object>
    <name>RBC</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>1158.0</xmin>
      <ymin>50.0</ymin>
      <xmax>1230.0</xmax>
      <ymax>122.0</ymax>
    </bndbox>
  </object>
  ...
</annotation>
```

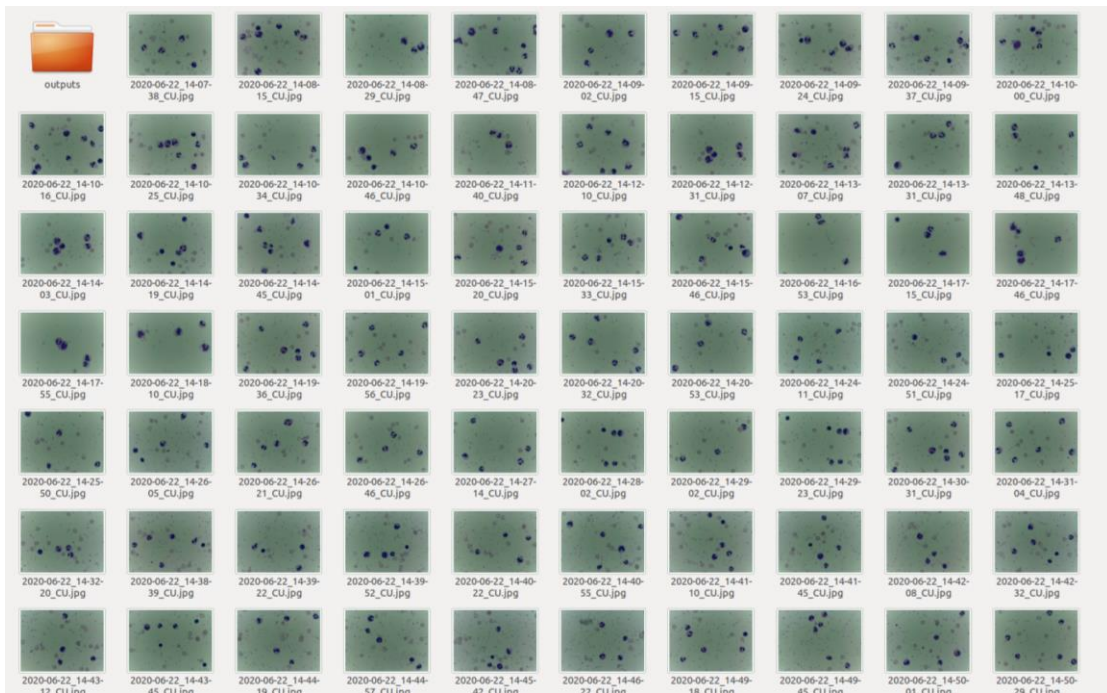
#### 4.1.2 ผลลัพธ์การขยายภาพเซลล์เม็ดเลือดขาว (WBC) และเกล็ดเลือด (Platelets)

```
# ชุดคำสั่งแสดงผลภาพจากไฟล์ .jpg และตำแหน่งพื้นที่ปิดล้อมรอบวัตถุจากไฟล์ .xml
images_df.to_csv('csv_labels.csv', index=False)
grouped = images_df.groupby('filename')
group_df = grouped.get_group('2020-06-22_16-06-25_CU.jpg')
group_df = group_df.reset_index()
group_df = group_df.drop(['index'], axis=1)
bbox_array = group_df.drop(['filename', 'width', 'height', 'class'], axis=1).values
image = imageio.imread('images/2020-06-22_16-06-25_CU.jpg')
bbox = BoundingBoxesOnImage.from_xyxy_array(bbox_array, shape=image.shape)
ia.imshow(bbox.draw_on_image(image, size=2))
```

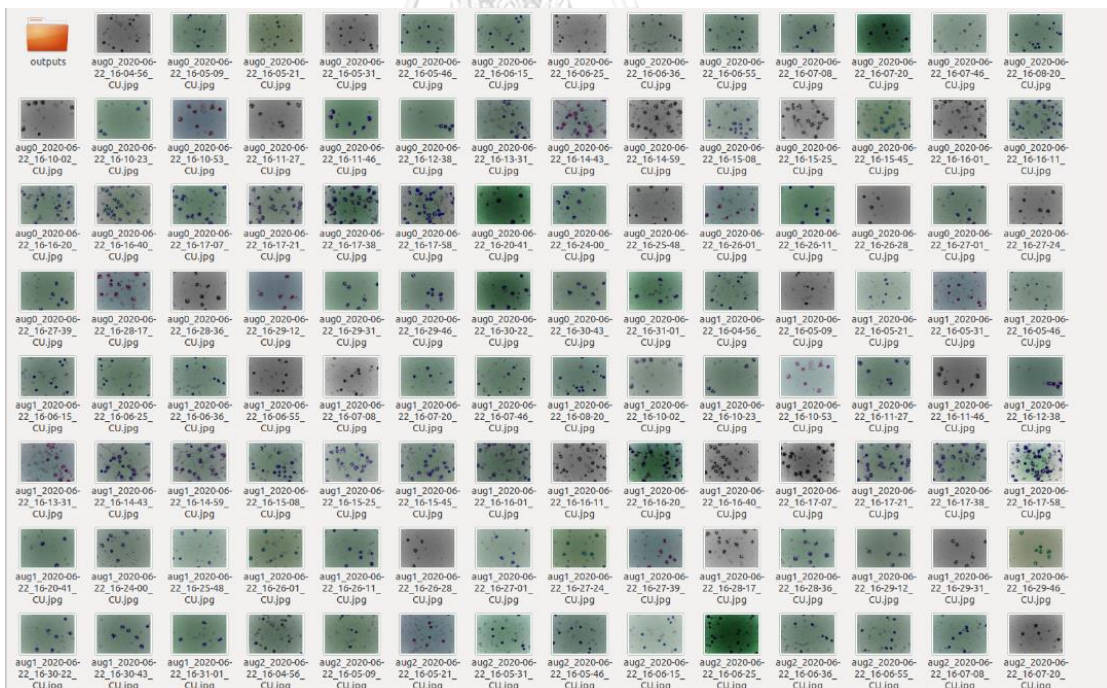


ภาพที่ 82 การแสดงตำแหน่งพื้นที่สี่เหลี่ยมปิดล้อมบนภาพเซลล์เม็ดเลือดขาวและเกล็ดเลือดภายใน

ชุดภาพ



ภาพที่ 83 ตัวอย่างชุดข้อมูลภาพเซลล์เม็ดเลือดขาวและเกล็ดเลือด จากกล้องจุลทรรศน์ก่อนได้รับการขยายชุดภาพ



ภาพที่ 84 ตัวอย่างชุดข้อมูลภาพเซลล์เม็ดเลือดขาวและเกล็ดเลือด จากกล้องจุลทรรศน์หลังได้รับการขยายชุดภาพ



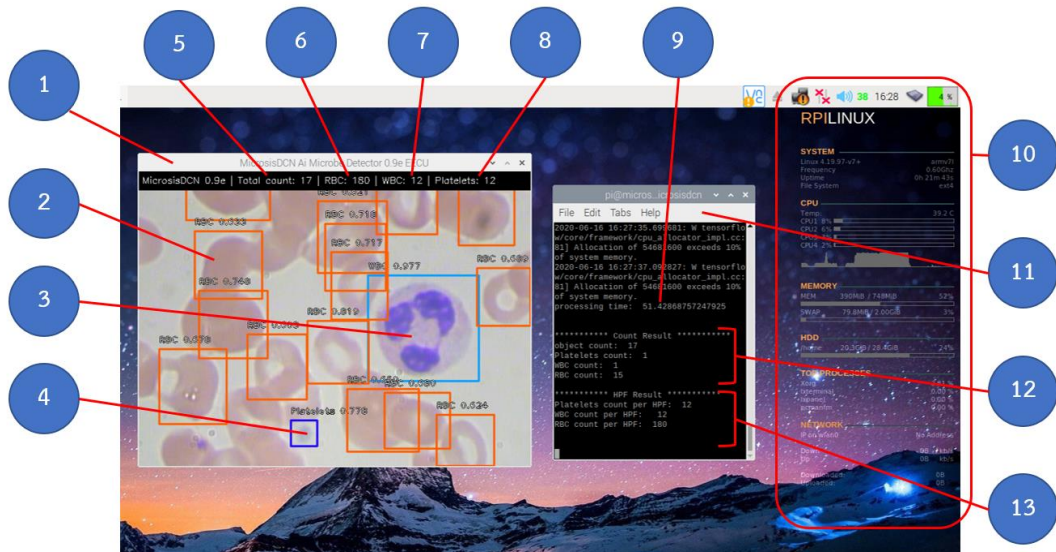
ภาพที่ 85 ตัวอย่างไฟล์บรรณนิทัศน์ XML ที่มีข้อมูลการติดป้ายวัตถุของชุดภาพที่ได้รับการขยาย (2)

ตารางที่ 15 ผลแสดงพิกัดของพื้นที่สี่เหลี่ยมปิดล้อมต่อวัตถุภายในชุดข้อมูลภาพแบบ CSV (2)

Filename	Width	Height	Class	X <sub>min</sub>	Y <sub>min</sub>	X <sub>max</sub>	Y <sub>max</sub>
2020-06-22_16-28-17_CU.jpg	1280	960	Platelets	1229	875	1245	893
2020-06-22_16-28-17_CU.jpg	1280	960	Platelets	1246	879	1265	904
2020-06-22_16-31-01_CU.jpg	1280	960	WBC	157	543	287	656
2020-06-22_16-31-01_CU.jpg	1280	960	WBC	674	161	794	296
2020-06-22_16-31-01_CU.jpg	1280	960	WBC	686	652	807	781
...	...	...	...	...	...	...	...
2020-06-22_16-31-01_CU.jpg	1280	960	RBC	701	488	766	552
2020-06-22_16-31-01_CU.jpg	1280	960	RBC	844	590	914	664
2020-06-22_16-31-01_CU.jpg	1280	960	Platelets	22	52	54	79
2020-06-22_16-31-01_CU.jpg	1280	960	Platelets	113	16	142	44
2020-06-22_16-31-01_CU.jpg	1280	960	Platelets	301	33	334	66
2020-06-22_16-31-01_CU.jpg	1280	960	Platelets	502	4	528	27

```
<!-- รายละเอียดตัวอย่างไฟล์บรรณนิทัศน์ชุดข้อมูลภาพ WBC และ Platelets ที่ถูกขยาย-->
<?xml version="1.0" ?>
<annotation>
<folder>MicrosisDCN_Datasets</folder>
<filename>aug4_2020-06-22_16-27-24_CU.jpg</filename>
<path>aug_images/aug4_2020-06-22_16-27-24_CU.jpg</path>
<source>
  <database>Engineering Chulalongkorn</database>
</source>
<size>
  <width>1280</width>
  <height>960</height>
  <depth>3</depth>
</size>
  <segmented>0</segmented>
  <object>
    <name>Platelets</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
      <xmin>94.0</xmin>
      <ymin>68.0</ymin>
      <xmax>113.0</xmax>
      <ymin>87.0</ymin>
    </bndbox>
  </object>
  ...
</annotation>
```

#### 4.1.3 พารามิเตอร์ของโมเดลโครงข่ายประสาทที่ใช้ในชุดกล้อง



ภาพที่ 86 หน้าจอชุดคำสั่งภาษา Python เข้าถึงโมเดลโครงข่ายประสาทเพื่อนับเซลล์ขนาดเล็ก

เมื่อได้โมเดลอนุমানจากการฝึกบนคอมพิวเตอร์แล้ว จะนำไปติดตั้งในชุดกล้อง MicrosisDCN ที่เป็นอุปกรณ์สำหรับสวมชุดกล้องเข้ากับท่อเลนส์ใกล้ตา (Eyepiece lens tube) วิธีการใช้งานชุดกล้อง MicrosisDCN สามารถดูเพิ่มเติมใน ภาคผนวก ง. วิธีการใช้งานชุดกล้อง MICROSIDCN กับกล้องจุลทรรศน์ ส่วนหนึ่งของชุดคำสั่งในหน้าถัดไป ภายในมีชุดสมองกลฝังตัว Raspberry Pi Computer Module 3+ ในการเรียกใช้งานโมเดลโครงข่ายประสาท สามารถทำได้โดยการเปิดหน้าต่าง Terminal บนระบบปฏิบัติการ จากนั้นจึงเข้าถึง Virtual environment ที่ได้ติดตั้งไว้ด้วยคำสั่ง `~/bashrc` และ `workon cv` แล้วเรียกใช้งานคำสั่งภาษา Python เช่น `python blood_detection.py` จะปรากฏในภาพที่ 86 ส่วนประกอบมีดังต่อไปนี้

- 1.) หน้าต่างชุดคำสั่งแสดงผลที่ทำงานด้วยคำสั่งชุดคำสั่ง OpenCV แสดงภาพเซลล์ที่นับได้
- 2.) เซลล์เม็ดเลือดแดงที่ถูกระบุตำแหน่งด้วยโครงข่ายประสาท แสดงชื่อคลาสและค่า mAP
- 3.) เซลล์เม็ดเลือดขาวที่ถูกระบุตำแหน่งด้วยโครงข่ายประสาท แสดงชื่อคลาสและค่า mAP
- 4.) เกล็ดเลือดที่ถูกระบุตำแหน่งด้วยโครงข่ายประสาท แสดงชื่อคลาสและค่า mAP
- 5.) จำนวนเซลล์ขนาดเล็กที่ถูกนับได้ทั้งหมด ในภาพจากกล้องจุลทรรศน์
- 6.) จำนวนเม็ดเลือดแดงที่นับได้ในพื้นที่มาตรฐานการมองเห็นของชุดกล้อง (HPF)
- 7.) จำนวนเม็ดเลือดขาวที่นับได้ในพื้นที่มาตรฐานการมองเห็นของชุดกล้อง (HPF)
- 8.) จำนวนเกล็ดเลือดที่นับได้ในพื้นที่มาตรฐานการมองเห็นของชุดกล้อง (HPF)

- 9.) ระยะเวลาการประมวลทั้งหมดตั้งแต่เรียกใช้งานโมเดลถึงนับจำนวนในหน่วยวินาที
- 10.) แยกเครื่องมือแสดงสถานการณ์ทำงานของชุดกล้อง เช่น CPU RAM และ Process
- 11.) หน้าต่าง Terminal ในระบบปฏิบัติการ Raspbian บน Virtual environment
- 12.) จำนวนเซลล์ขนาดเล็กที่นับได้จริง ในภาพจากกล้องจุลทรรศน์แยกเป็นรายคลาส
- 13.) จำนวนเซลล์ขนาดเล็กที่นับได้ในพื้นที่มาตรฐานการมองเห็นของชุดกล้อง (HPF) แยกเป็นรายคลาส

การทำงานของโมเดลโครงข่ายประสาท RetinaNet ที่ประกอบด้วย Residual Network 50 (ResNet50) และ Feature Pyramid Network (FPN) จากการพัฒนาโครงข่ายประสาทภายในชุดกล้อง MicroSisDCN ด้วยการทำ Transfer learning นอกจากการฝึกด้วยชุดข้อมูลภาพใหม่ที่ยังไม่ได้รับการฝึกมาก่อนด้วยชุดข้อมูลภาพถ่ายเซลล์เม็ดเลือดจากกล้องจุลทรรศน์ (Blood Cell Dataset) ยังมีการปรับพารามิเตอร์ในชั้นคอนโวลูชัน (CONV) ใน ResNet50 เพื่อให้การแยกคุณลักษณะของข้อมูล (Feature extraction) จะทำในพื้นที่ย่อยของภาพโดยการใช้ตัวกรอง จำแนกและดึงลักษณะที่สนใจให้แสดงออกมา ด้วยการปรับ Anchor ให้เหมาะสมกับขนาดป้ายของวัตถุในภาพจนได้ฟังก์ชันคุณลักษณะที่จำแนกวัตถุออกได้ ซึ่งการฝึกโมเดลโครงข่ายประสาทจะถูกแบ่งออกเป็น 3 แบบ ได้แก่ การฝึกโมเดลแบบไม่มีการปรับพารามิเตอร์ การฝึกโมเดลปรับพารามิเตอร์แบบที่ 1 และการฝึกโมเดลปรับพารามิเตอร์แบบที่ 2

การแยกคุณลักษณะของข้อมูล (Feature extraction) จะทำในพื้นที่ย่อยของภาพที่มาจากรูปข้อมูลภาพที่ได้รับการขยายจะมีขนาดความกว้าง 1280 ความสูง 960 จุดภาพ การสร้างตัวกรอง (Kernel) ในการจำแนกและดึงลักษณะที่สนใจให้แสดงออกมาสามารถปรับให้เหมาะสมกับขนาดของวัตถุที่อยู่ในพื้นที่สี่เหลี่ยมปิดล้อมได้ หรือการปรับพารามิเตอร์ของ Anchor ประกอบด้วย Sizes เป็นขนาดของพื้นที่จุดภาพที่ต้องการแยกคุณลักษณะของข้อมูลตามจำนวนชั้นคุณลักษณะ เช่น ค่า 512 จะมีขนาดเป็น  $512 \times 512$  จุดภาพ ซึ่งใน ResNet50 จะมีทั้งหมด 5 ชั้น Strides เป็นขนาดของกระบวนการเลื่อนตัวกรองคอนโวลูชันมีขนาดพื้นที่จุดภาพเล็กกว่า Sizes ตามจำนวนชั้นคุณลักษณะในโครงข่ายประสาท เช่น ค่า 128 จะมีขนาดเป็น  $128 \times 128$  จุดภาพ Ratios เป็นอัตราส่วนของฟังก์ชันคุณลักษณะ (Feature map) ที่ใช้ต่อหนึ่งช่วงพื้นที่ของ Anchor คิดได้จากความสูงของฟังก์ชันคุณลักษณะหารด้วยความกว้างของฟังก์ชันคุณลักษณะ Scales เป็นขนาดสเกลของฟังก์ชันคุณลักษณะ (Feature map) ที่ใช้ต่อหนึ่งช่วงพื้นที่ของ Anchor เมื่อนำค่าของ Ratios คูณด้วย Scales จะได้จำนวน Anchor ที่เกิดขึ้นในการหาตำแหน่งของวัตถุในพื้นที่สี่เหลี่ยมปิดล้อม ซึ่งทั้ง Ratios และ Scales จะใช้ค่ามาตรฐาน (Default anchor parameters)

ตารางที่ 16 ค่าพารามิเตอร์ Anchor มาตรฐานของ RetinaNet

Sizes	32	64	128	256	512
Strides	8	16	32	64	128
Ratios	0.5	1	2	3	
Scales	1	1.2	1.6		

ตารางที่ 17 ค่าพารามิเตอร์ Anchor แบบที่ 1 สำหรับฝึกโมเดลโครงข่ายประสาท

Sizes	16	32	64	128	256
Strides	8	16	32	64	128
Ratios	0.5	1	2	3	
Scales	1	1.2	1.6		

ตารางที่ 18 ค่าพารามิเตอร์ Anchor แบบที่ 2 สำหรับฝึกโมเดลโครงข่ายประสาท

Sizes	8	16	32	64	128
Strides	4	8	16	32	64
Ratios	0.5	1	2	3	
Scales	1	1.2	1.6		

การฝึกโมเดลโครงข่ายประสาททั้ง 3 แบบ จะใช้ชุดคำสั่งดังต่อไปนี้ คือ ชุดคำสั่งสำหรับฝึกโมเดลโครงข่ายประสาท `python train.py --weights ./snapshots/blood_model.h5 --config config.ini --compute-val-loss --tensorboard-dir ./tensorboard_log --multi-gpu 2 --multi-gpu-force --batch-size 2 --epochs 50 --steps 1000 csv annotations.csv classes.csv --val-annotations val_annotations.csv` โดยที่การฝึกโมเดลแบบไม่มีการปรับพารามิเตอร์ จะตัด arguments `--config config.ini` ออก ชุดคำสั่งสำหรับชุดคำสั่งสำหรับแปลงโมเดลโครงข่ายประสาทให้เป็นโมเดลอนุมาน (Inference model) `python /convert_model.py --config config.ini ./snapshots/blood_model_csv_XX.h5 ./snapshots/blood_model_csv_XX_convert.h5` และชุดคำสั่งประเมินความสามารถของโมเดลอนุมาน กับชุดตรวจสอบความถูกต้อง (Validation set) เพื่อดูค่า Mean Average Precision (mAP) ในการพยากรณ์ของทั้ง 3 คลาส `python /evaluate.py csv val_annotations.csv classes.csv ./snapshots/blood_model_csv_XX_convert.h5` โดยที่ XX มีค่าตั้งแต่ 01 ถึง 50 ตามจำนวน epochs ของการฝึก



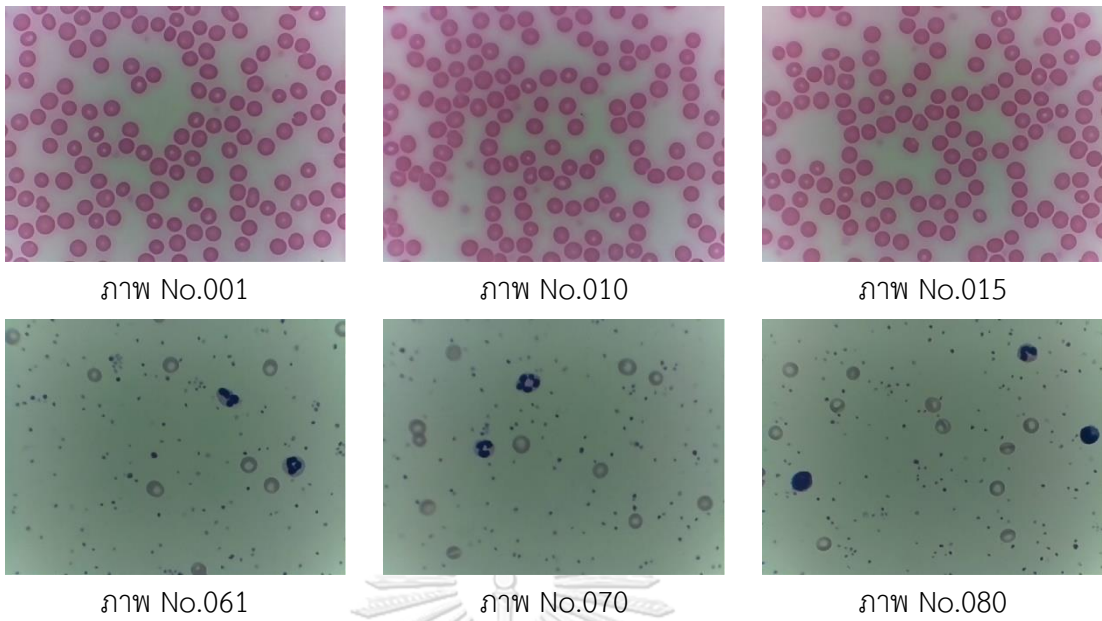
#### 4.2 ผลลัพธ์การฝึกโมเดลแบบไม่ปรับพารามิเตอร์

จากการฝึกโมเดลโครงข่ายประสาทแบบไม่ปรับพารามิเตอร์ตามค่าในตารางที่ 16 ครบจำนวน 50 Epochs แล้วนั้น จากกราฟแสดงค่า mAP (Mean Average Precision) ของ 3 คลาส ได้แก่ เซลล์เม็ดเลือดแดง (RBC) เซลล์เม็ดเลือดขาว (WBC) และเกล็ดเลือด (Platelets) มีค่าสูงสุดที่ 0.7356 และมีค่าต่ำสุดที่ 0.6794 ซึ่งความแม่นยำในการพยากรณ์ทั้ง 3 คลาสมีค่าน้อยกว่า 0.8000 หรือร้อยละ 80 ผู้วิจัยจึงได้ทดสอบเปรียบเทียบภาพที่ถูกระบุจำนวนที่นับได้จริงจากตามมนุษย์ และภาพที่ถูกระบุจำนวนที่พยากรณ์ได้จากชุดกล้องที่เป็นภาพเดียวกันจำนวน 100 ภาพ ได้ผลลัพธ์ปรากฏในตารางที่ 20 ถึง 23 เพื่อนำมาหาค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (Mean Absolute Error : MAE) บ่งบอกความคลาดเคลื่อนโดยภาพรวม ค่ารากที่สองของค่าความคลาดเคลื่อนเฉลี่ย (Root Mean Squared Error : RMSE) บ่งบอกความคลาดเคลื่อนทางสถิติมาตรฐาน และค่าสัมประสิทธิ์แสดงการตัดสินใจ (Coefficient of Determination :  $R^2$ ) บ่งบอกความแปรปรวนของค่าจริงกับค่าพยากรณ์

จากค่า MAE และ RMSE ของเซลล์เม็ดเลือดแดงและเซลล์เม็ดเลือดขาวมีค่าต่ำ แสดงว่าโมเดลโครงข่ายประสาทนั้นมีความแม่นยำในการตรวจจับเซลล์ทั้งสองชนิดได้ดี แต่ค่า MAE และ RMSE ของเกล็ดเลือดนั้นมีค่าสูง ส่งผลให้ความแม่นยำในการตรวจจับเกล็ดเลือดนั้นทำได้ไม่ดี ซึ่งสอดคล้องกับค่า  $R^2$  ของเซลล์เม็ดเลือดแดงและเซลล์เม็ดเลือดขาวนั้นมีค่าเข้าใกล้ 1 ในขณะที่ของเกล็ดเลือดมีค่าที่ 0.156402 แสดงว่าความสัมพันธ์ของพื้นที่ของพื้นที่สี่เหลี่ยมปิดล้อม (Bounding box) ที่เกิดซ้อนทับของพื้นที่จริงชั้นพื้นฐาน (Ground truth box) และพื้นที่จากการพยากรณ์ (Predicted box) นั้นไม่ซ้อนทับกันพอดี หรือระบุชนิดของคลาสไม่ได้ ผลลัพธ์ของโมเดลโครงข่ายประสาทถูกแสดงอ้างอิงจากภาพที่ 87 ภาพการพยากรณ์เซลล์เม็ดเลือดแดงแสดงในภาพที่ 91 และเซลล์เม็ดเลือดขาวกับเกล็ดเลือดแสดงในภาพที่ 92 จะเห็นว่าโมเดลโครงข่ายประสาทนี้สามารถตรวจจับเซลล์เม็ดเลือดแดงและเม็ดเลือดขาวได้ แต่จะไม่สามารถตรวจจับเกล็ดเลือดได้

ตารางที่ 19 ผลลัพธ์ค่า MAE RMSE  $R^2$  และ mAP ของโมเดลแบบไม่ปรับพารามิเตอร์

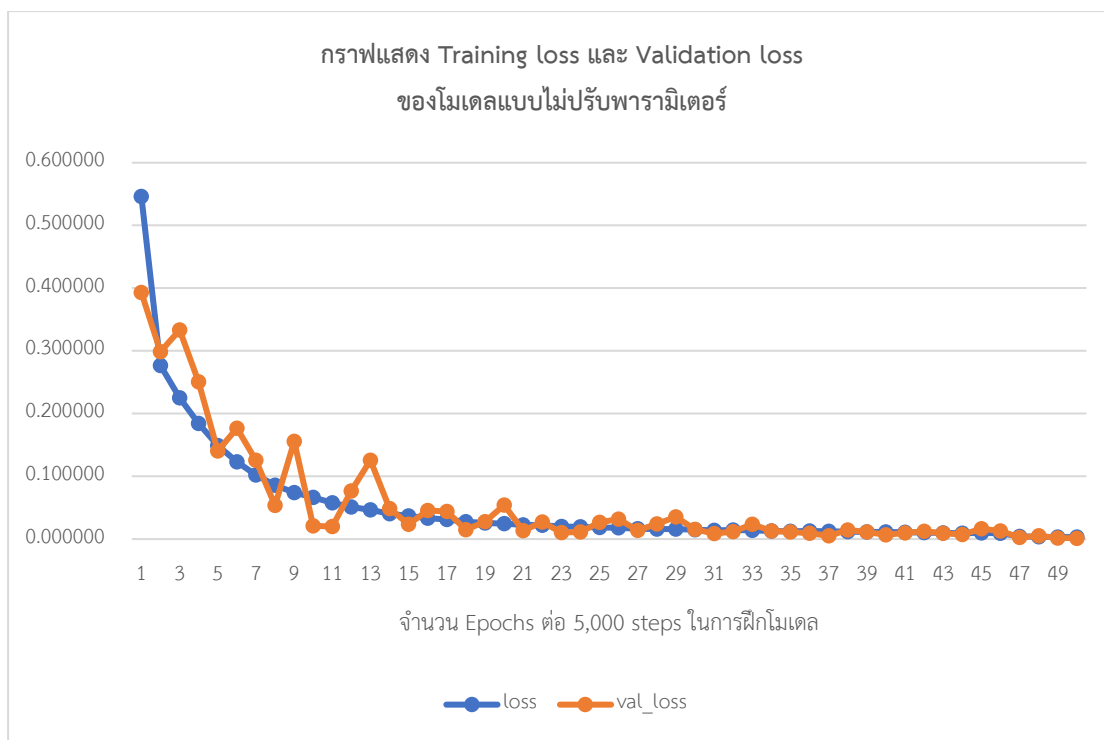
ชื่อคลาส	RBC	WBC	Platelets
MAE	6.09	0.12	58.91
RMSE	8.497647	0.489898	96.78414
$R^2$	0.998511	0.972978	0.156402
mAP	0.6794 ถึง 0.7356		



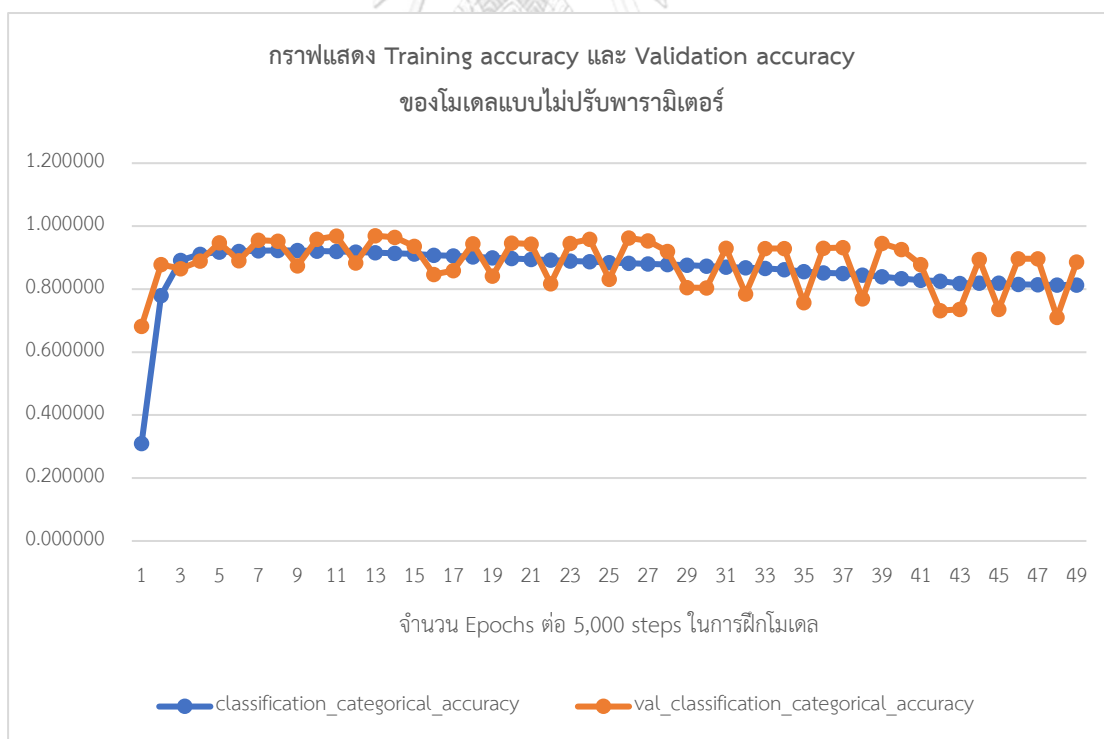
ภาพที่ 87 ชุดภาพพร้อมหมายเลขสำหรับแสดงผลเปรียบเทียบของโมเดลแบบไม่ปรับพารามิเตอร์



ภาพที่ 88 กราฟแสดงความแม่นยำ mAP ของโมเดลแบบไม่ปรับพารามิเตอร์

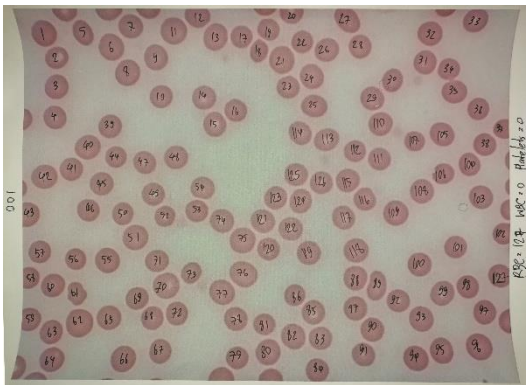


ภาพที่ 89 กราฟแสดง Training loss และ Validation loss ของโมเดลแบบไม่ปรับพารามิเตอร์



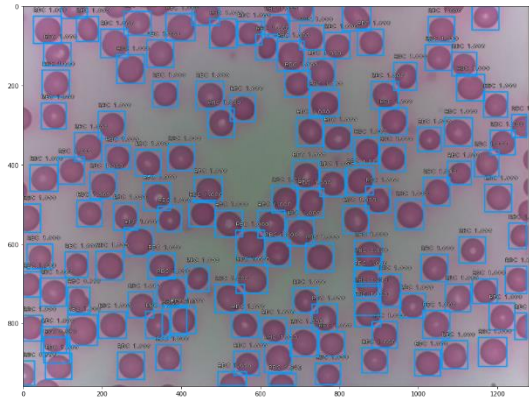
ภาพที่ 90 กราฟแสดง Training accuracy และ Validation accuracy ของโมเดลแบบไม่ปรับพารามิเตอร์

ภาพที่นับด้วยตามนุษย์

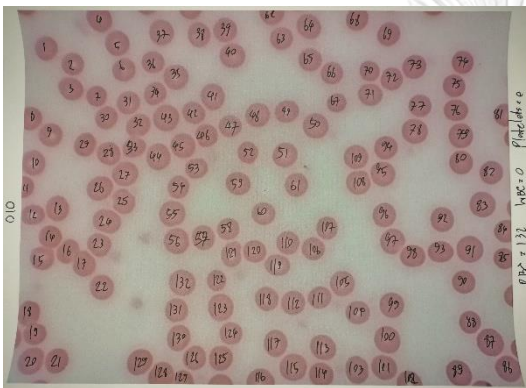


ภาพ No. 001

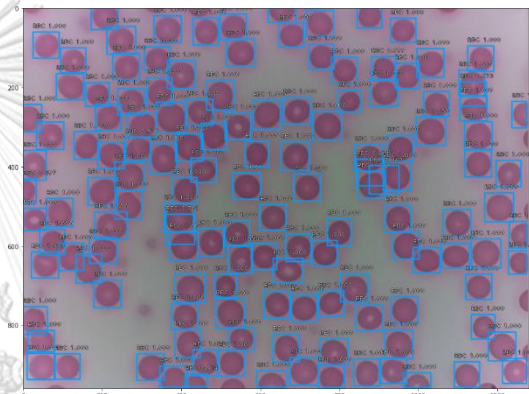
ภาพที่นับด้วยโครงข่ายประสาท



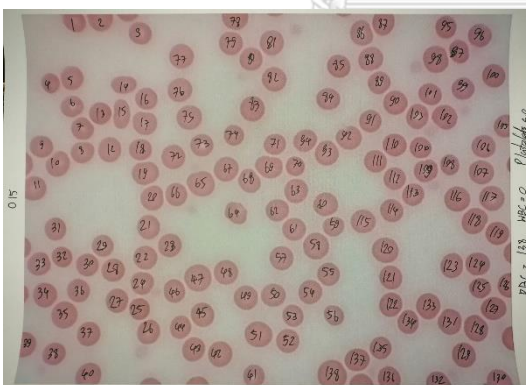
ภาพ No. 001



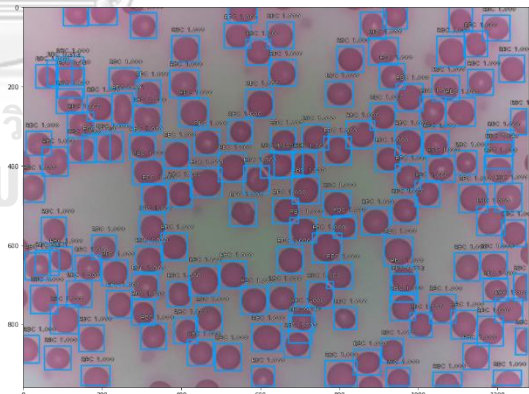
ภาพ No. 010



ภาพ No. 010



ภาพ No. 015

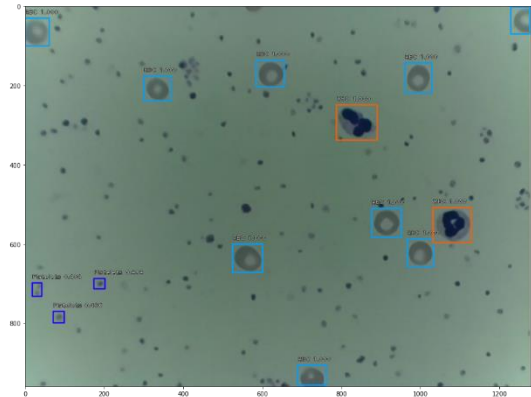
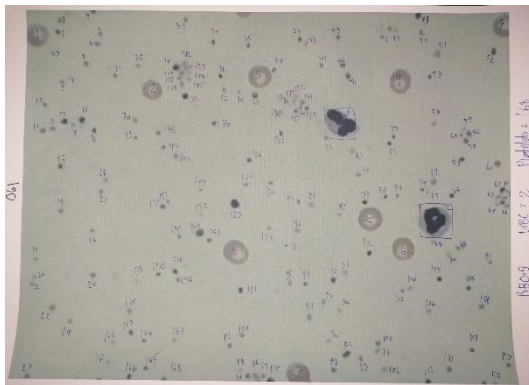


ภาพ No. 015

ภาพที่ 91 ผลลัพธ์การนับจำนวนเซลล์เม็ดเลือดแดงที่นับได้จริงและพยากรณ์ได้ของโมเดลแบบไม่ปรับพารามิเตอร์

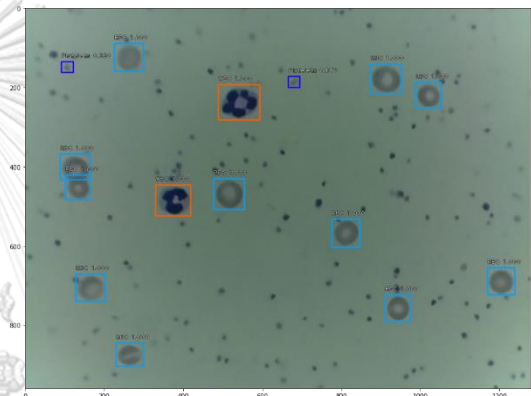
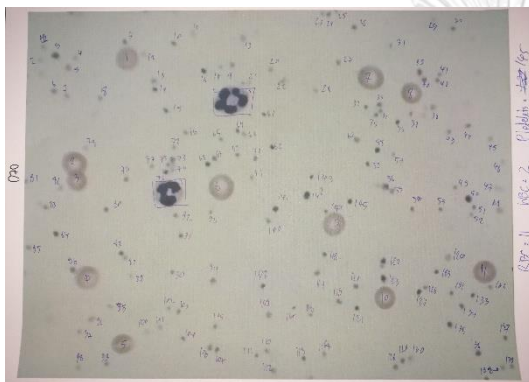
ภาพที่นับด้วยตามนุษย์

ภาพที่นับด้วยโครงข่ายประสาท



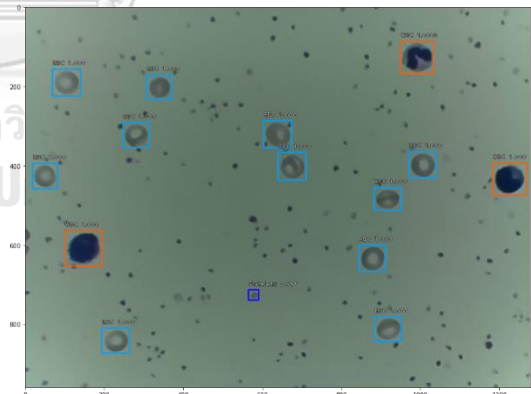
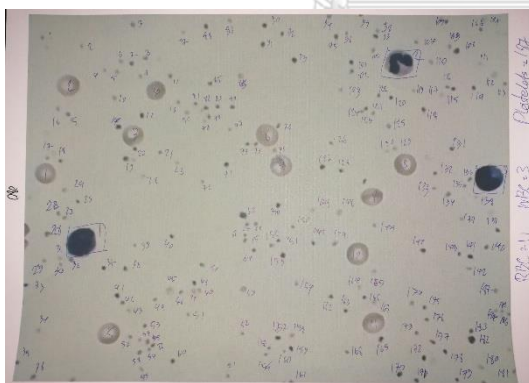
ภาพ No. 061

ภาพ No. 061



ภาพ No. 070

ภาพ No. 070



ภาพ No. 080

ภาพ No. 080

ภาพที่ 92 ผลลัพธ์การนับจำนวนเซลล์เม็ดเลือดขาวและเกล็ดเลือดที่นับได้จริงและพยากรณ์ได้ของโมเดลแบบไม่ปรับพารามิเตอร์

ตารางที่ 20 ผลเปรียบเทียบจำนวนที่นับได้จริงและพยากรณ์ได้ของโมเดลแบบไม่ปรับพารามิเตอร์ (1)

ภาพ No.	จำนวนที่นับจริง			จำนวนที่พยากรณ์ได้			ค่าผิดพลาดสัมบูรณ์		
	RBC	WBC	Platelets	RBC	WBC	Platelets	RBC	WBC	Platelets
001	127	0	0	136	0	0	9	0	0
002	142	0	0	154	0	0	12	0	0
003	124	0	0	131	0	0	7	0	0
004	110	0	0	116	0	0	6	0	0
005	146	0	0	155	0	0	9	0	0
006	125	0	0	130	0	0	5	0	0
007	134	0	0	142	0	0	8	0	0
008	136	0	0	141	0	0	5	0	0
009	133	1	0	141	1	0	8	0	0
010	132	0	0	136	0	0	4	0	0
011	127	0	0	132	0	0	5	0	0
012	125	0	0	131	0	0	6	0	0
013	138	0	0	148	0	0	10	0	0
014	125	0	0	134	0	0	9	0	0
015	138	0	0	147	0	0	9	0	0
016	120	0	0	130	0	0	10	0	0
017	118	0	0	125	0	0	7	0	0
018	125	1	0	138	1	0	13	0	0
019	116	0	0	124	0	1	8	0	1
020	126	0	0	129	0	0	3	0	0
021	124	0	0	137	0	0	13	0	0
022	120	0	0	129	0	0	9	0	0
023	130	0	0	138	0	0	8	0	0
024	131	1	0	142	1	0	11	0	0
025	123	0	0	142	0	1	19	0	1

ตารางที่ 21 ผลเปรียบเทียบจำนวนที่นับได้จริงและพยากรณ์ได้ของโมเดลแบบไม่ปรับพารามิเตอร์ (2)

ภาพ No.	จำนวนที่นับจริง			จำนวนที่พยากรณ์ได้			ค่าผิดพลาดสัมบูรณ์		
	RBC	WBC	Platelets	RBC	WBC	Platelets	RBC	WBC	Platelets
026	125	0	0	131	0	0	6	0	0
027	121	0	0	132	0	0	11	0	0
028	131	0	0	140	0	0	9	0	0
029	137	0	0	149	0	0	12	0	0
030	136	0	0	148	0	0	12	0	0
031	129	0	0	138	0	0	9	0	0
032	123	0	0	133	1	0	10	1	0
033	127	0	0	131	1	0	4	1	0
034	130	0	0	139	0	0	9	0	0
035	130	0	0	137	0	0	7	0	0
036	130	0	0	141	0	0	11	0	0
037	144	1	0	161	0	0	17	1	0
038	135	0	0	145	0	0	10	0	0
039	133	0	0	139	0	0	6	0	0
040	134	0	0	148	0	0	14	0	0
041	132	0	0	135	0	0	3	0	0
042	136	0	0	147	0	0	11	0	0
043	145	0	0	154	0	0	9	0	0
044	138	0	0	151	0	0	13	0	0
045	131	0	0	136	0	0	5	0	0
046	145	0	0	157	1	0	12	1	0
047	155	0	0	170	0	0	15	0	0
048	137	0	0	147	0	0	10	0	0
049	150	0	0	166	1	0	16	1	0
050	132	0	0	142	0	0	10	0	0

ตารางที่ 22 ผลเปรียบเทียบจำนวนที่นับได้จริงและพยากรณ์ได้ของโมเดลแบบไม่ปรับพารามิเตอร์ (3)

ภาพ No.	จำนวนที่นับจริง			จำนวนที่พยากรณ์ได้			ค่าผิดพลาดสัมบูรณ์		
	RBC	WBC	Platelets	RBC	WBC	Platelets	RBC	WBC	Platelets
051	115	0	0	121	0	0	6	0	0
052	103	0	0	109	0	0	6	0	0
053	114	0	0	118	1	0	4	1	0
054	121	0	0	131	0	0	10	0	0
055	125	0	0	128	0	0	3	0	0
056	133	2	0	153	2	0	20	0	0
057	145	0	0	165	0	0	20	0	0
058	118	0	0	128	0	0	10	0	0
059	124	0	0	137	0	0	13	0	0
060	131	0	0	142	0	0	11	0	0
061	9	2	168	9	2	3	0	0	165
062	11	3	175	11	3	2	0	0	173
063	5	4	173	5	4	9	0	0	164
064	14	7	177	15	7	1	1	0	176
065	12	3	143	13	3	0	1	0	143
066	12	1	161	12	1	1	0	0	160
067	5	1	155	5	1	3	0	0	152
068	7	1	129	7	1	2	0	0	127
069	7	1	57	7	1	7	0	0	50
070	11	2	145	11	2	2	0	0	143
071	9	2	159	9	2	0	0	0	159
072	10	1	172	10	1	0	0	0	172
073	9	2	171	9	2	2	0	0	169
074	9	2	172	9	2	1	0	0	171
075	12	1	150	12	1	2	0	0	148



ตารางที่ 23 ผลเปรียบเทียบจำนวนที่นับได้จริงและพยากรณ์ได้ของโมเดลแบบไม่ปรับพารามิเตอร์ (4)

ภาพ No.	จำนวนที่นับจริง			จำนวนที่พยากรณ์ได้			ค่าผิดพลาดสัมบูรณ์		
	RBC	WBC	Platelets	RBC	WBC	Platelets	RBC	WBC	Platelets
076	11	2	197	11	2	0	0	0	197
077	12	1	176	13	1	0	1	0	176
078	3	3	154	4	3	1	1	0	153
079	12	9	204	14	9	0	2	0	204
080	11	3	187	11	3	1	0	0	186
081	1	5	14	31	1	5	30	4	9
082	6	6	102	5	6	0	1	0	102
083	8	4	93	8	4	0	0	0	93
084	11	4	131	11	4	0	0	0	131
085	9	4	146	9	4	0	0	0	146
086	9	6	119	9	6	0	0	0	119
087	12	4	170	11	4	0	1	0	170
088	8	5	107	9	5	0	1	0	107
089	10	6	139	10	6	0	0	0	139
090	11	8	133	11	8	0	0	0	133
091	11	5	115	11	5	0	0	0	115
092	13	5	131	13	6	0	0	1	131
093	10	5	164	11	5	0	1	0	164
094	6	3	108	6	3	0	0	0	108
095	13	8	195	13	8	0	0	0	195
096	8	4	191	8	4	0	0	0	191
097	9	9	198	10	9	0	1	0	198
098	11	8	214	10	8	0	1	0	214
099	12	6	163	12	5	0	0	1	163
100	3	6	78	3	6	5	0	0	73

### 4.3 ผลลัพธ์การฝึกโมเดลปรับพารามิเตอร์แบบที่ 1

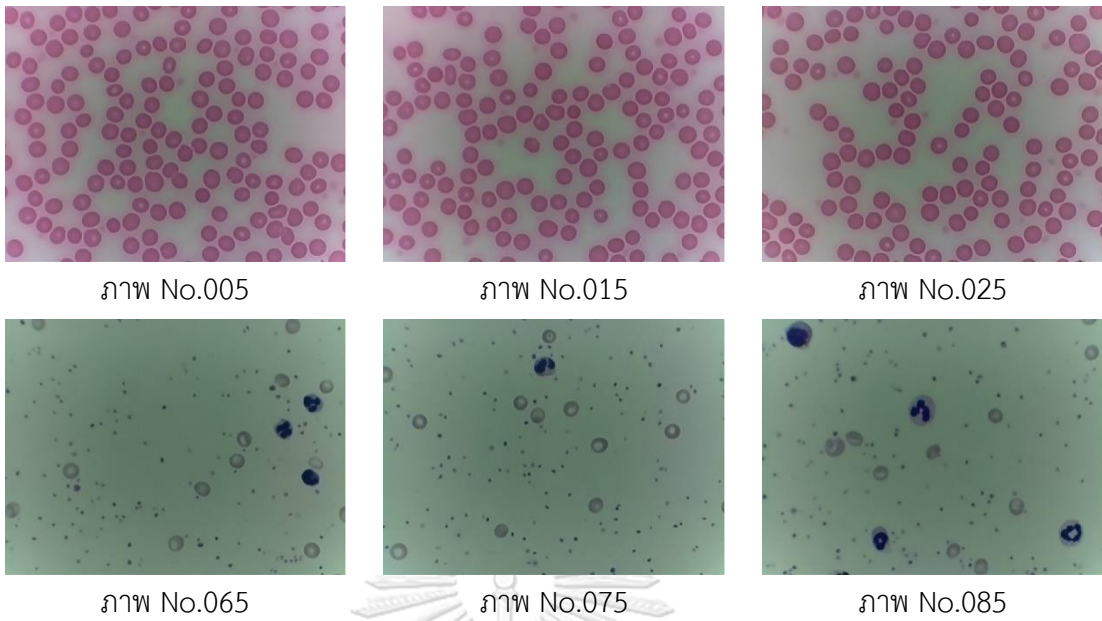
ในการฝึกโมเดลแบบไม่ปรับพารามิเตอร์ความแม่นยำในการพยากรณ์ทั้ง 3 คลาสมีค่าน้อยกว่า 0.8000 หรือร้อยละ 80 ซึ่งโมเดลโครงข่ายประสาทสามารถตรวจจับเซลล์เม็ดเลือดแดงและเซลล์เม็ดเลือดขาวได้ มีค่าความคลาดเคลื่อนสัมบูรณ์เฉลี่ย (MAE) 6.09 และ 0.12 ตามลำดับ แต่ของเกล็ดเลือดมีค่าเป็น 58.91 ส่งผลให้ตรวจจับเกล็ดเลือดไม่ได้ จึงมีความจำเป็นต้องปรับปรุงโมเดลโครงข่ายประสาทให้สามารถตรวจจับและนับจำนวนได้ครบทั้ง 3 คลาส ด้วยการปรับ Anchor ให้เหมาะสมกับขนาดป้ายของวัตถุในภาพจนได้ฟังก์ชันลักษณะที่จำแนกวัตถุออกได้ จากค่าพารามิเตอร์ Anchor มาตรฐานในตารางที่ 16 จะพบว่า Sizes เป็นขนาดของพื้นที่จุดภาพที่ต้องการแยกคุณลักษณะของข้อมูลตามจำนวนชั้นคุณลักษณะที่เล็กที่สุดมีค่า 32 จะมีขนาดเป็น 32×32 จุดภาพ ทำให้วัตถุที่ติดป้ายที่มีขนาดเล็กกว่า 32×32 จุดภาพ จะไม่ถูกสร้างฟังก์ชันลักษณะทำให้ไม่สามารถตรวจจับได้

จากงานวิจัยของ M. Zlocha et. al. ได้เสนอวิธีการปรับ Anchor ให้เหมาะสมกับวัตถุขนาดเล็กในภาพถ่าย CT Scan โดยการใช้ Anchor optimization [39] ผู้วิจัยจึงปรับค่าพารามิเตอร์ Anchor แบบที่ 1 สำหรับฝึกโมเดลโครงข่ายประสาท แสดงในตารางที่ 17 จากนั้นจึงเริ่มการฝึกโมเดลโครงข่ายประสาทใหม่จนครบจำนวน 50 Epochs จากกราฟแสดงค่า mAP (Mean Average Precision) ของ 3 คลาส มีค่าสูงสุดที่ 0.8681 และมีค่าต่ำสุดที่ 0.7305 ซึ่งความแม่นยำในการพยากรณ์ทั้ง 3 คลาสมีค่าสูงขึ้นกว่าครั้งก่อน มีค่ามากกว่า 0.8000 หรือร้อยละ 80 ผู้วิจัยจึงได้ทดสอบเปรียบเทียบภาพที่ถูกระบุจำนวนที่นับได้จริงจากตามนุษย์ และภาพที่ถูกระบุจำนวนที่พยากรณ์ได้จากชุดกล้องที่เป็นภาพเดียวกันจำนวน 100 ภาพ ได้ผลลัพธ์ปรากฏในตารางที่ 25 ถึง 28

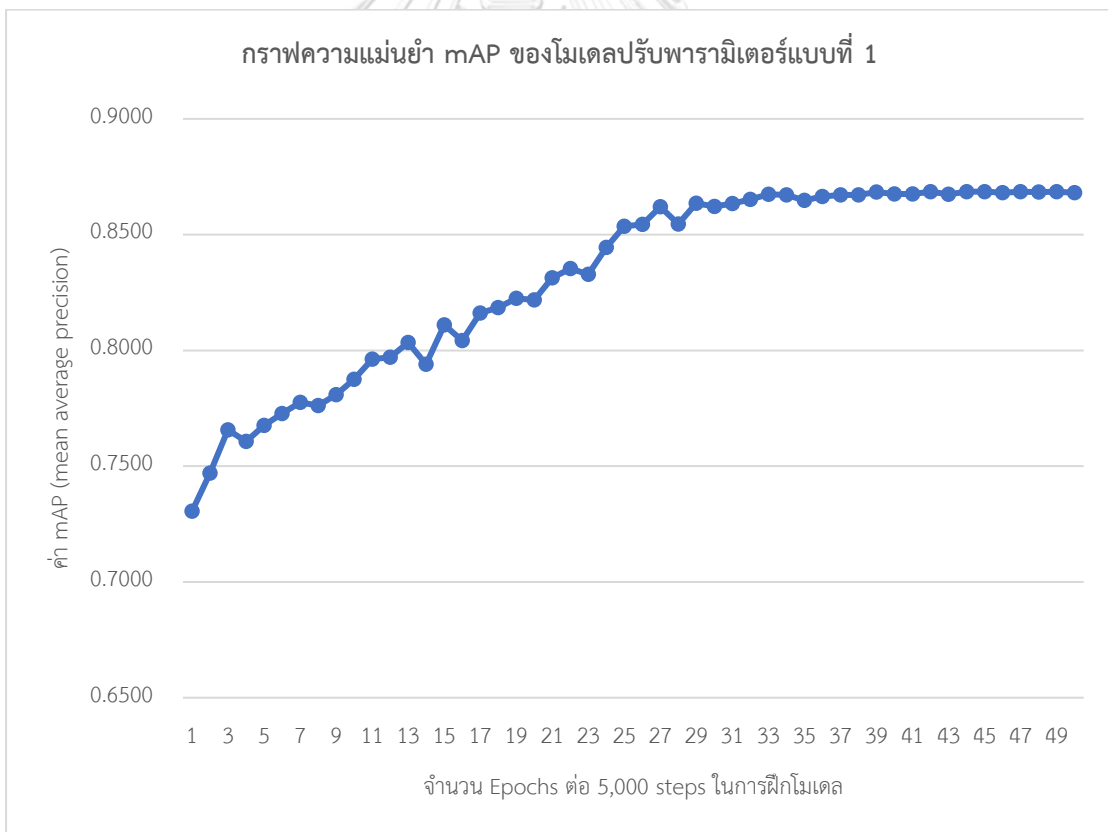
จากค่า MAE และ RMSE ทั้ง 3 คลาสมีค่าต่ำ แสดงว่าโมเดลโครงข่ายประสาทนั้นมีความแม่นยำในการตรวจจับเซลล์ทั้ง 3 ชนิดได้ดี ซึ่งสอดคล้องกับค่า  $R^2$  ของทั้ง 3 คลาสนั้นมีค่าเข้าใกล้ 1 ผลลัพธ์ของโมเดลโครงข่ายประสาทถูกแสดงอ้างอิงจากภาพที่ 93 ภาพการพยากรณ์เซลล์เม็ดเลือดแดงแสดงในภาพที่ 97 และเซลล์เม็ดเลือดขาวกับเกล็ดเลือดแสดงในภาพที่ 98 จะเห็นว่าโมเดลโครงข่ายประสาทนี้สามารถตรวจจับเซลล์ได้ครบทุกคลาสสามารถนำไปใช้กับชุดกล้องได้

ตารางที่ 24 ผลลัพธ์ค่า MAE RMSE  $R^2$  และ mAP ของโมเดลปรับพารามิเตอร์แบบที่ 1

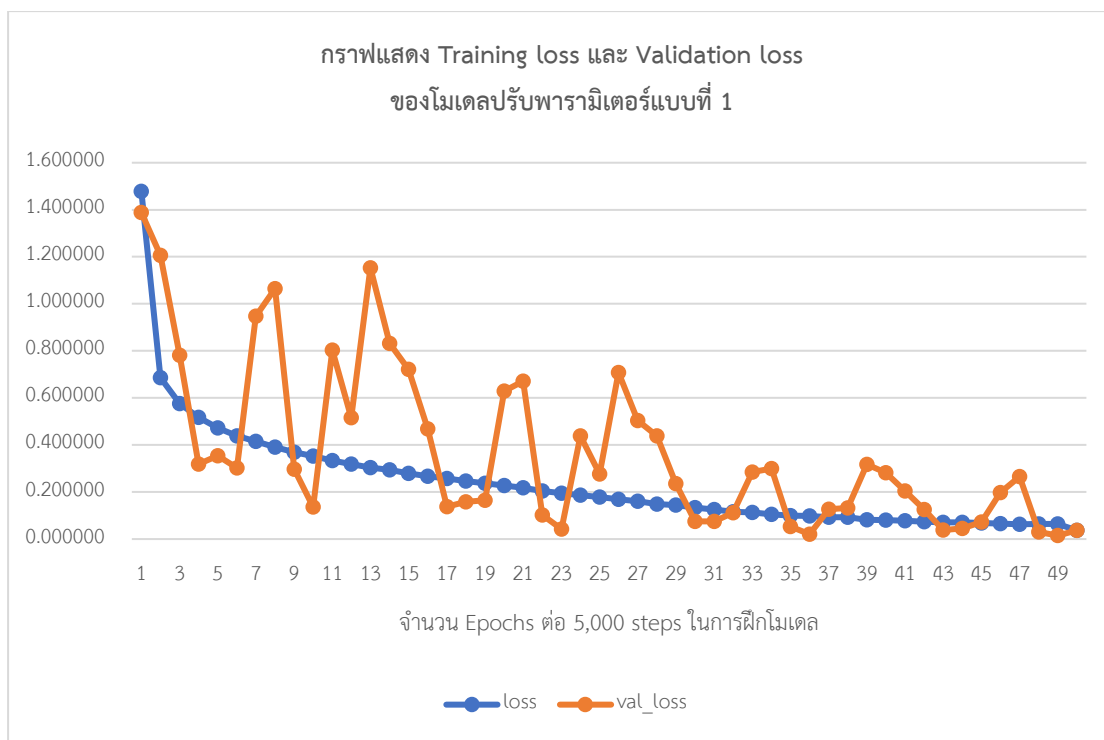
ชื่อคลาส	RBC	WBC	Platelets
MAE	1.06	0.06	4.23
RMSE	1.643168	0.244949	9.101099
$R^2$	0.999800	0.993290	0.995952
mAP	0.7305 ถึง 0.8681		



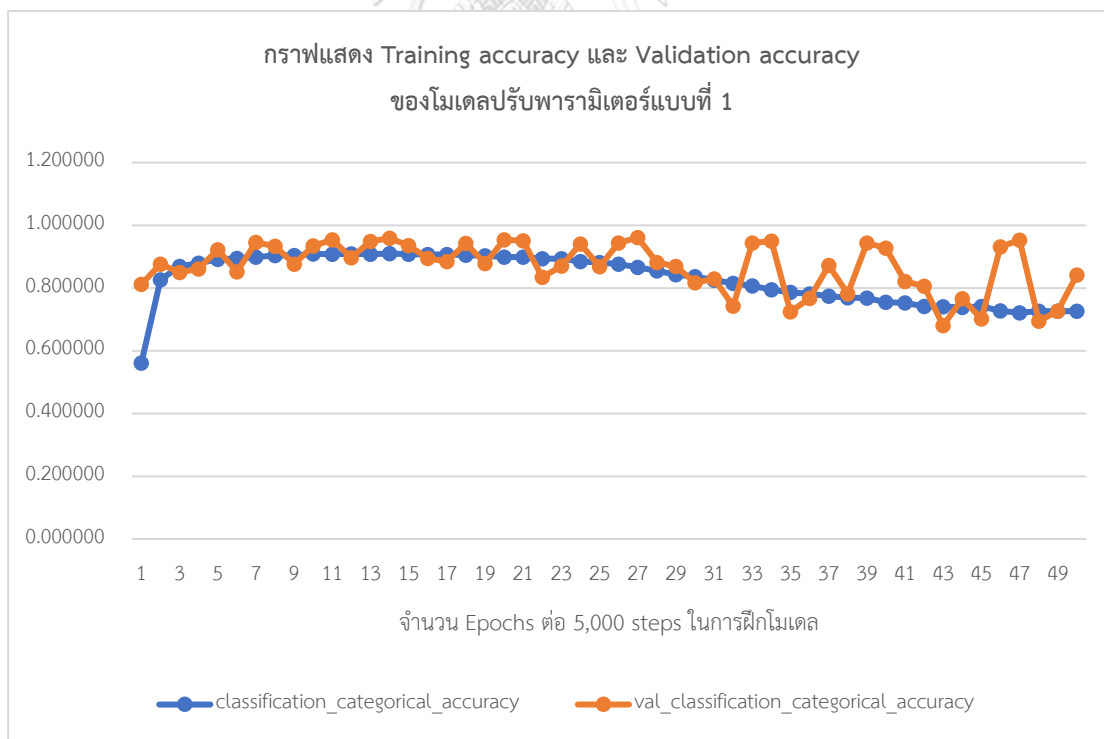
ภาพที่ 93 ชุดภาพพร้อมหมายเลขสำหรับแสดงผลเปรียบเทียบของโมเดลปรับพารามิเตอร์แบบที่ 1



ภาพที่ 94 กราฟแสดงค่าความแม่นยำ mAP ของโมเดลปรับพารามิเตอร์แบบที่ 1



ภาพที่ 95 กราฟแสดง Training loss และ Validation loss ของโมเดลปรับพารามิเตอร์แบบที่ 1



ภาพที่ 96 กราฟแสดง Training accuracy และ Validation accuracy ของโมเดลปรับพารามิเตอร์แบบที่ 1

ตารางที่ 25 ผลเปรียบเทียบจำนวนที่นับจริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 1 (1)

ภาพ No.	จำนวนที่นับจริง			จำนวนที่พยากรณ์ได้			ผิดพลาดสัมบูรณ์ (Error)		
	RBC	WBC	Platelets	RBC	WBC	Platelets	RBC	WBC	Platelets
001	127	0	0	128	0	0	1	0	0
002	142	0	0	143	0	0	1	0	0
003	124	0	0	126	0	0	2	0	0
004	110	0	0	109	0	0	1	0	0
005	146	0	0	147	0	0	1	0	0
006	125	0	0	125	0	0	0	0	0
007	134	0	0	131	0	0	3	0	0
008	136	0	0	139	0	0	3	0	0
009	133	1	0	131	1	0	2	0	0
010	132	0	0	129	0	0	3	0	0
011	127	0	0	128	0	0	1	0	0
012	125	0	0	124	0	0	1	0	0
013	138	0	0	141	0	0	3	0	0
014	125	0	0	124	0	0	1	0	0
015	138	0	0	139	0	0	1	0	0
016	120	0	0	120	0	0	0	0	0
017	118	0	0	119	0	0	1	0	0
018	125	1	0	125	0	0	0	1	0
019	116	0	0	118	0	0	2	0	0
020	126	0	0	126	0	0	0	0	0
021	124	0	0	125	0	0	1	0	0
022	120	0	0	122	0	0	2	0	0
023	130	0	0	129	0	0	1	0	0
024	131	1	0	134	1	0	3	0	0
025	123	0	0	125	0	0	2	0	0

ตารางที่ 26 ผลเปรียบเทียบจำนวนที่นับจริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 1 (2)

ภาพ No.	จำนวนที่นับจริง			จำนวนที่พยากรณ์ได้			ผิดพลาดสัมบูรณ์ (Error)		
	RBC	WBC	Platelets	RBC	WBC	Platelets	RBC	WBC	Platelets
026	125	0	0	125	0	0	0	0	0
027	121	0	0	123	0	0	2	0	0
028	131	0	0	131	0	0	0	0	0
029	137	0	0	140	0	0	3	0	0
030	136	0	0	136	0	0	0	0	0
031	129	0	0	129	0	0	0	0	0
032	123	0	0	123	0	0	0	0	0
033	127	0	0	129	0	0	2	0	0
034	130	0	0	131	0	0	1	0	0
035	130	0	0	134	0	0	4	0	0
036	130	0	0	130	0	0	0	0	0
037	144	1	0	145	0	0	1	1	0
038	135	0	0	135	0	0	0	0	0
039	133	0	0	135	0	0	2	0	0
040	134	0	0	135	0	0	1	0	0
041	132	0	0	129	0	0	3	0	0
042	136	0	0	140	0	0	4	0	0
043	145	0	0	142	0	0	3	0	0
044	138	0	0	140	0	0	2	0	0
045	131	0	0	131	0	0	0	0	0
046	145	0	0	146	0	0	1	0	0
047	155	0	0	156	0	0	1	0	0
048	137	0	0	138	0	0	1	0	0
049	150	0	0	152	0	0	2	0	0
050	132	0	0	135	0	0	3	0	0

ตารางที่ 27 ผลเปรียบเทียบจำนวนที่นับได้จริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 1 (3)

ภาพ No.	จำนวนที่นับได้จริง			จำนวนที่พยากรณ์ได้			ผิดพลาดสัมบูรณ์ (Error)		
	RBC	WBC	Platelets	RBC	WBC	Platelets	RBC	WBC	Platelets
051	115	0	0	115	0	0	0	0	0
052	103	0	0	106	0	0	3	0	0
053	114	0	0	116	0	0	2	0	0
054	121	0	0	123	0	0	2	0	0
055	125	0	0	128	0	0	3	0	0
056	133	2	0	134	1	0	1	1	0
057	145	0	0	151	0	0	6	0	0
058	118	0	0	121	0	0	3	0	0
059	124	0	0	125	0	0	1	0	0
060	131	0	0	135	0	0	4	0	0
061	9	2	168	9	2	201	0	0	33
062	11	3	175	11	3	177	0	0	2
063	5	4	173	5	4	184	0	0	11
064	14	7	177	16	6	211	2	1	34
065	12	3	143	12	3	154	0	0	11
066	12	1	161	13	1	173	1	0	12
067	5	1	155	5	1	164	0	0	9
068	7	1	129	7	1	144	0	0	15
069	7	1	57	7	1	61	0	0	4
070	11	2	145	11	2	150	0	0	5
071	9	2	159	9	2	165	0	0	6
072	10	1	172	10	1	169	0	0	3
073	9	2	171	9	2	174	0	0	3
074	9	2	172	9	2	173	0	0	1
075	12	1	150	12	1	158	0	0	8

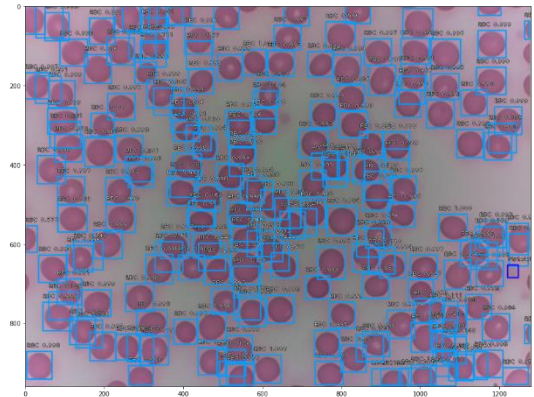
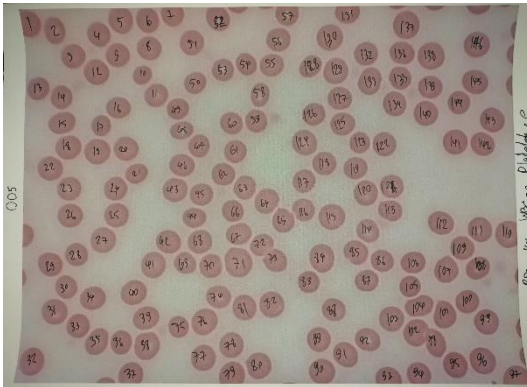
ตารางที่ 28 ผลเปรียบเทียบจำนวนที่นับจริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 1 (4)

ภาพ No.	จำนวนที่นับจริง			จำนวนที่พยากรณ์ได้			ผิดพลาดสัมบูรณ์ (Error)		
	RBC	WBC	Platelets	RBC	WBC	Platelets	RBC	WBC	Platelets
076	11	2	197	11	2	229	0	0	32
077	12	1	176	12	1	186	0	0	10
078	3	3	154	3	3	151	0	0	3
079	12	9	204	12	9	234	0	0	30
080	11	3	187	11	3	216	0	0	29
081	1	5	14	1	5	11	0	0	3
082	6	6	102	5	6	103	1	0	1
083	8	4	93	8	4	103	0	0	10
084	11	4	131	11	4	132	0	0	1
085	9	4	146	9	4	143	0	0	3
086	9	6	119	9	6	116	0	0	3
087	12	4	170	11	4	171	1	0	1
088	8	5	107	9	5	119	1	0	12
089	10	6	139	10	6	148	0	0	9
090	11	8	133	11	8	155	0	0	22
091	11	5	115	11	5	121	0	0	6
092	13	5	131	13	6	139	0	1	8
093	10	5	164	11	5	159	1	0	5
094	6	3	108	6	3	115	0	0	7
095	13	8	195	13	8	208	0	0	13
096	8	4	191	8	4	194	0	0	3
097	9	9	198	10	9	204	1	0	6
098	11	8	214	10	8	224	1	0	10
099	12	6	163	12	5	172	0	1	9
100	3	6	78	3	6	108	0	0	30



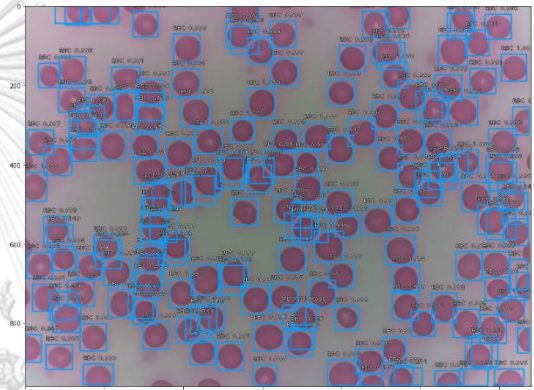
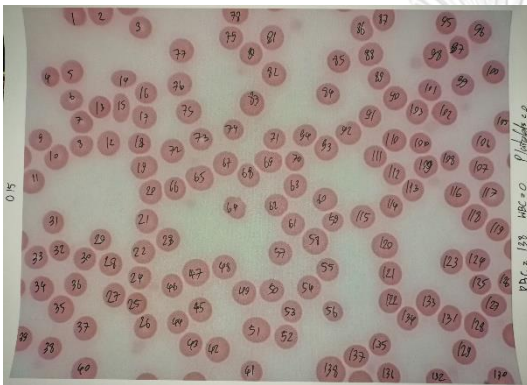
ภาพที่นับด้วยตามมนุษย์

ภาพที่นับด้วยโครงข่ายประสาท



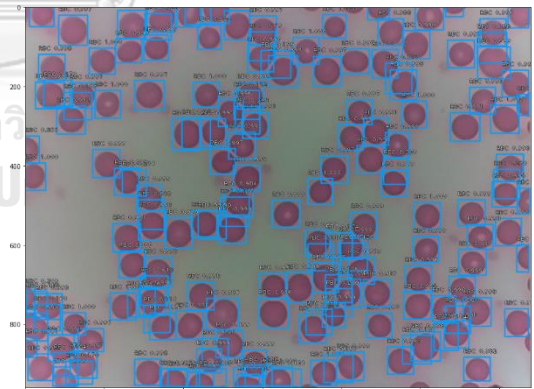
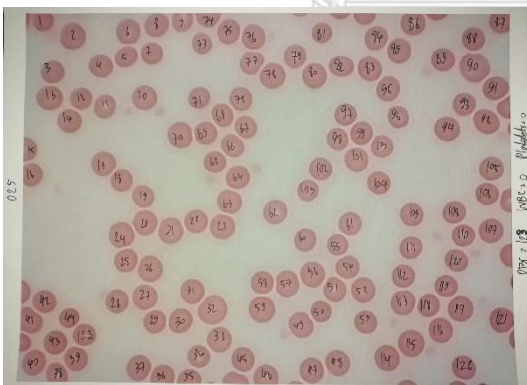
ภาพ No. 005

ภาพ No. 005



ภาพ No. 015

ภาพ No. 015

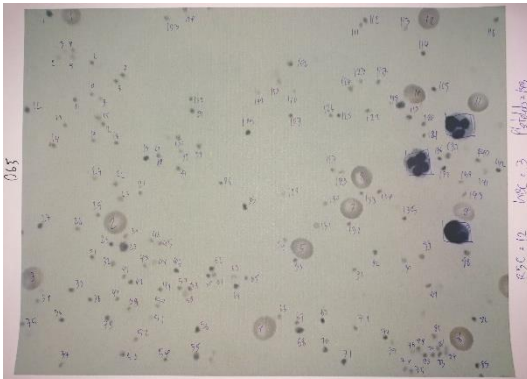


ภาพ No. 025

ภาพ No. 025

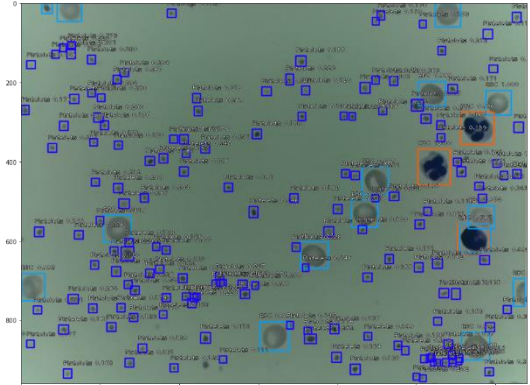
ภาพที่ 97 ผลลัพธ์การนับจำนวนเซลล์เม็ดเลือดแดงที่นับได้จริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 1

ภาพที่นับด้วยตามมนุษย์

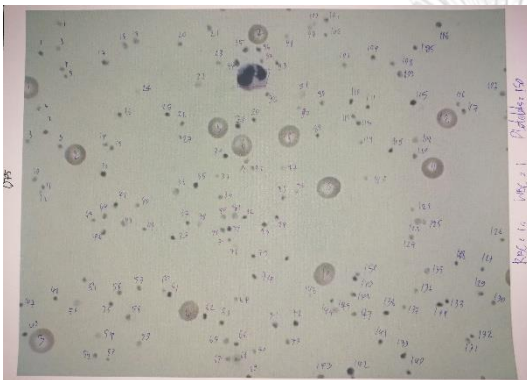


ภาพ No. 065

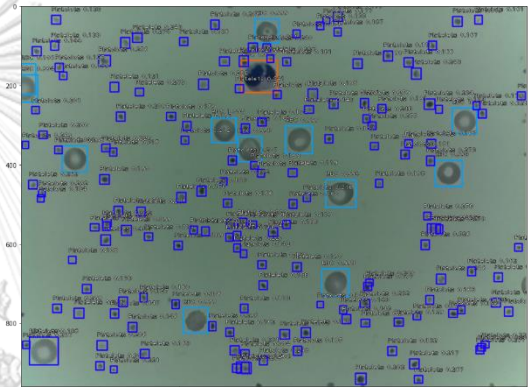
ภาพที่นับด้วยโครงข่ายประสาท



ภาพ No. 065



ภาพ No. 075



ภาพชุดที่ 075



ภาพ No. 085



ภาพ No. 085

ภาพที่ 98 ผลลัพธ์การนับจำนวนเซลล์เม็ดเลือดขาวและเกล็ดเลือดที่นับได้จริงและพยากรณ์ได้ของ  
โมเดลปรับพารามิเตอร์แบบที่ 1

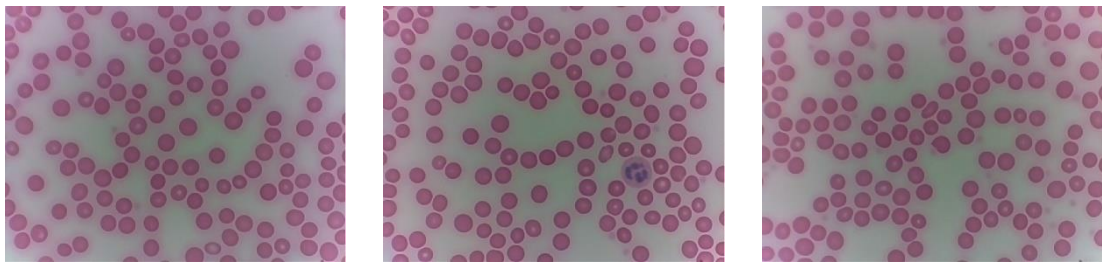
#### 4.4 ผลลัพธ์การฝึกโมเดลปรับพารามิเตอร์แบบที่ 2

เนื่องด้วยการฝึกโมเดลแบบปรับพารามิเตอร์แบบที่ 1 ส่งผลให้ความแม่นยำในการพยากรณ์ ทั้ง 3 คลาสเพิ่มขึ้นจากเดิม ผู้วิจัยจึงใช้ Anchor optimization ในการปรับ Anchor ให้มีขนาดเล็กกว่าเดิม เพื่อให้พารามิเตอร์ในชั้นคอนโวลูชัน (CONV) สามารถแยกคุณลักษณะของข้อมูล (Feature extraction) ได้ละเอียดมากยิ่งขึ้น ค่าพารามิเตอร์ Anchor มาตรฐานในตารางที่ 16 จะพบว่า Sizes เป็นขนาดของพื้นที่จุดภาพที่ต้องการแยกคุณลักษณะของข้อมูลตามจำนวนชั้นคุณลักษณะที่เล็กที่สุดมีค่า 32 จะมีขนาดเป็น  $32 \times 32$  จุดภาพ ผู้วิจัยจึงปรับ ค่าพารามิเตอร์ Anchor แบบที่ 2 สำหรับฝึกโมเดลโครงข่ายประสาท แสดงในตารางที่ 18 จากนั้นจึงเริ่มการฝึกโมเดลโครงข่ายประสาทใหม่จนครบจำนวน 50 Epochs จากกราฟแสดงค่า mAP (Mean Average Precision) ของ 3 คลาส มีค่าสูงสุดที่ 0.2293 และมีค่าต่ำสุดที่ 0.0037 ซึ่งความแม่นยำในการพยากรณ์ทั้ง 3 คลาสมีค่าน้อยกว่า 0.8000 หรือร้อยละ 80 และน้อยกว่าการฝึกโมเดลปรับพารามิเตอร์แบบที่ 1 ผู้วิจัยจึงได้ทดสอบเปรียบเทียบภาพที่ถูกระบุจำนวนที่นับได้จริงจากตามนุษย์ และภาพที่ถูกระบุจำนวนที่พยากรณ์ได้จากชุดกล้องที่เป็นภาพเดียวกันจำนวน 100 ภาพ ได้ผลลัพธ์ปรากฏในตารางที่ 30 ถึง 33

จากค่า MAE และ RMSE ทั้ง 3 คลาสมีค่าสูง แสดงว่าโมเดลโครงข่ายประสาทนั้นมีความแม่นยำในการตรวจจับเซลล์ทั้ง 3 ชนิดได้ไม่ดี และค่า ซึ่งสอดคล้องกับค่า  $R^2$  ของเซลล์เม็ดเลือดแดงมีค่าเข้าใกล้ 1 ส่วนเซลล์เม็ดเลือดขาวมีค่าต่ำ และเกล็ดเลือดหาค่าไม่ได้ เพราะไม่มีจำนวนที่พยากรณ์ได้เกิดขึ้น ผลลัพธ์ของโมเดลโครงข่ายประสาทถูกแสดงอ้างอิงจากภาพที่ 99 ภาพการพยากรณ์เซลล์เม็ดเลือดแดงแสดงในภาพที่ 103 และเซลล์เม็ดเลือดขาวกับเกล็ดเลือดแสดงในภาพที่ 104 จะเห็นว่าโมเดลโครงข่ายประสาทนี้ ไม่สามารถตรวจจับเซลล์ได้ครบทุกคลาส พื้นที่ของพื้นที่สี่เหลี่ยมปิดล้อม (Bounding box) ที่เกิดซ้อนทับของพื้นที่จริงขึ้นพื้นฐาน (Ground truth box) และพื้นที่จากการพยากรณ์ (Predicted box) นั้นไม่ถูกต้อง และไม่สามารถระบุตำแหน่งของเซลล์ขนาดเล็กใดได้เลย

ตารางที่ 29 ผลลัพธ์ค่า MAE RMSE  $R^2$  และ mAP ของโมเดลปรับพารามิเตอร์แบบที่ 2

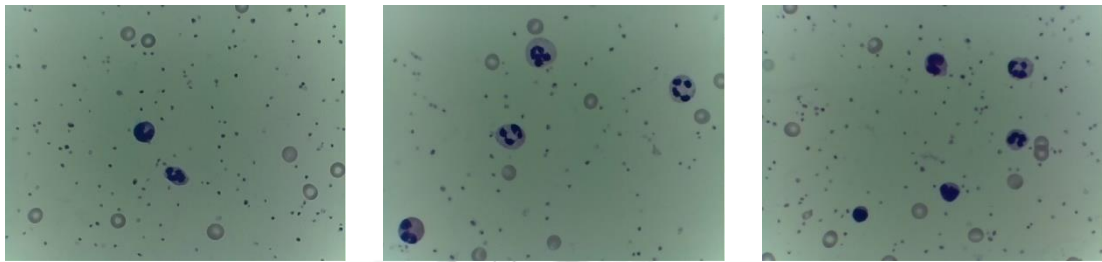
ชื่อคลาส	RBC	WBC	Platelets
MAE	48.9	1.36	59.36
RMSE	61.02098	2.52866	97.36755
$R^2$	0.971038	0.329402	หาค่าไม่ได้
mAP	0.0037 ถึง 0.2239		



ภาพ No.012

ภาพ No.024

ภาพ No.036

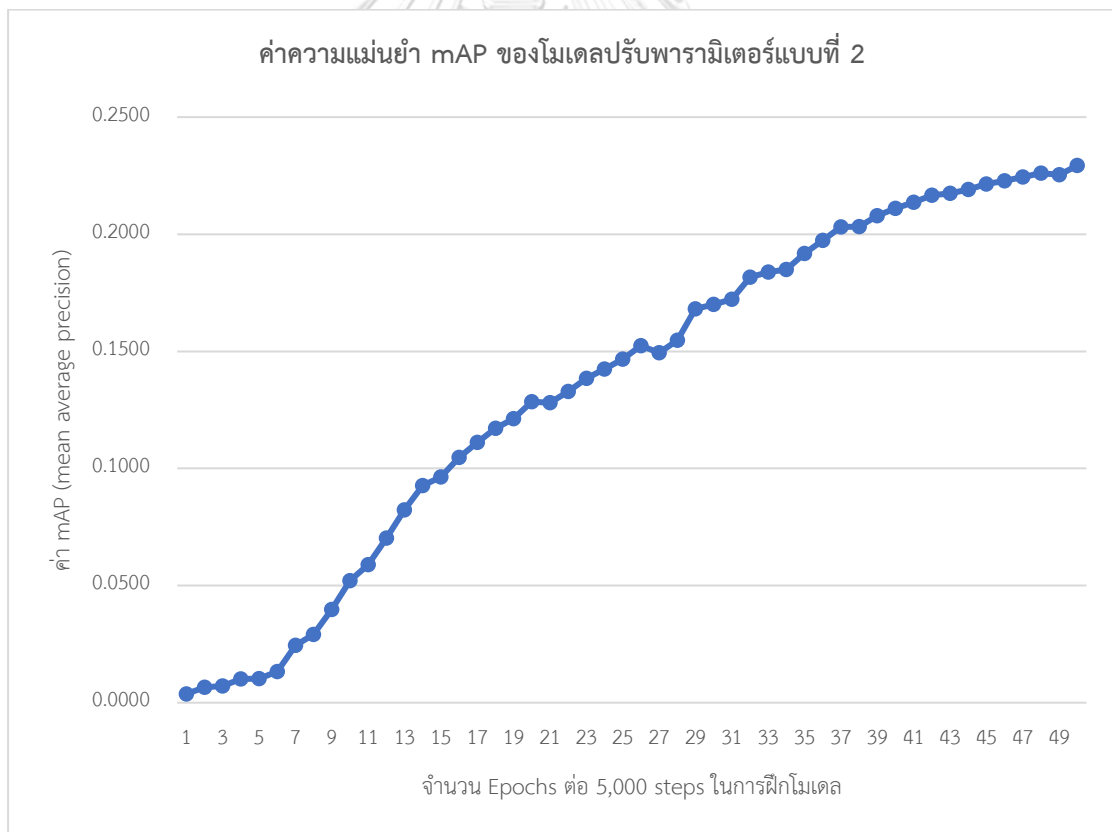


ภาพ No.073

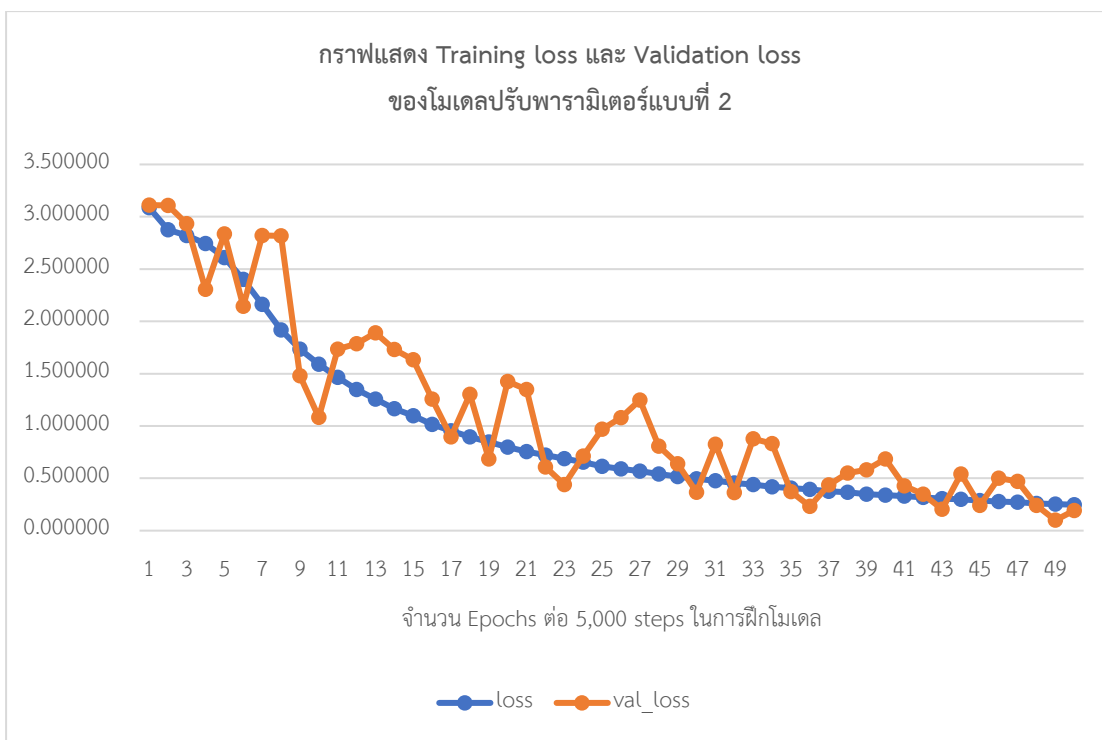
ภาพ No.083

ภาพ No.093

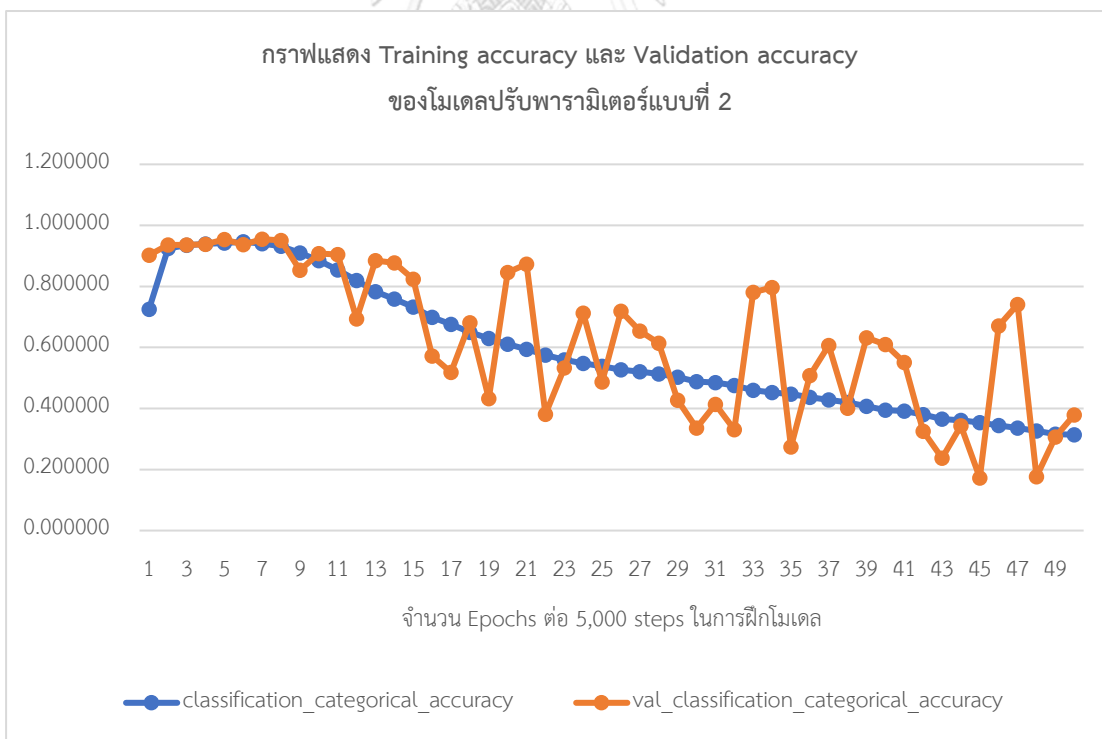
ภาพที่ 99 ชุดภาพพร้อมหมายเลขสำหรับแสดงผลเปรียบเทียบของโมเดลปรับพารามิเตอร์แบบที่ 2



ภาพที่ 100 กราฟแสดงความแม่นยำ mAP ของโมเดลปรับพารามิเตอร์แบบที่ 2



ภาพที่ 101 กราฟแสดง Training loss และ Validation loss ของโมเดลปรับพารามิเตอร์แบบที่ 2



ภาพที่ 102 กราฟแสดง Training accuracy และ Validation accuracy ของโมเดลปรับพารามิเตอร์แบบที่ 2

ตารางที่ 30 ผลเปรียบเทียบจำนวนที่นับจริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 2 (1)

ภาพ No.	จำนวนที่นับจริง			จำนวนที่พยากรณ์ได้			ผิดพลาดสัมบูรณ์ (Error)		
	RBC	WBC	Platelets	RBC	WBC	Platelets	RBC	WBC	Platelets
001	127	0	0	55	0	0	72	0	0
002	142	0	0	54	0	0	88	0	0
003	124	0	0	55	0	0	69	0	0
004	110	0	0	53	0	0	57	0	0
005	146	0	0	46	0	0	100	0	0
006	125	0	0	67	0	0	58	0	0
007	134	0	0	53	0	0	81	0	0
008	136	0	0	48	0	0	88	0	0
009	133	1	0	62	0	0	71	1	0
010	132	0	0	56	0	0	76	0	0
011	127	0	0	49	0	0	78	0	0
012	125	0	0	48	0	0	77	0	0
013	138	0	0	31	0	0	107	0	0
014	125	0	0	45	0	0	80	0	0
015	138	0	0	50	0	0	88	0	0
016	120	0	0	53	0	0	67	0	0
017	118	0	0	55	0	0	63	0	0
018	125	1	0	39	0	0	86	1	0
019	116	0	0	53	0	0	63	0	0
020	126	0	0	45	0	0	81	0	0
021	124	0	0	46	0	0	78	0	0
022	120	0	0	46	0	0	74	0	0
023	130	0	0	53	0	0	77	0	0
024	131	1	0	60	0	0	71	1	0
025	123	0	0	47	0	0	76	0	0

ตารางที่ 31 ผลเปรียบเทียบจำนวนที่นับจริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 2 (2)

ภาพ No.	จำนวนที่นับจริง			จำนวนที่พยากรณ์ได้			ผิดพลาดสัมบูรณ์ (Error)		
	RBC	WBC	Platelets	RBC	WBC	Platelets	RBC	WBC	Platelets
026	125	0	0	54	0	0	71	0	0
027	121	0	0	62	0	0	59	0	0
028	131	0	0	62	0	0	69	0	0
029	137	0	0	37	0	0	100	0	0
030	136	0	0	45	0	0	91	0	0
031	129	0	0	50	0	0	79	0	0
032	123	0	0	32	0	0	91	0	0
033	127	0	0	40	0	0	87	0	0
034	130	0	0	43	0	0	87	0	0
035	130	0	0	52	0	0	78	0	0
036	130	0	0	60	0	0	70	0	0
037	144	1	0	57	0	0	87	1	0
038	135	0	0	53	0	0	82	0	0
039	133	0	0	36	0	0	97	0	0
040	134	0	0	72	0	0	62	0	0
041	132	0	0	59	0	0	73	0	0
042	136	0	0	58	0	0	78	0	0
043	145	0	0	56	0	0	89	0	0
044	138	0	0	48	0	0	90	0	0
045	131	0	0	66	0	0	65	0	0
046	145	0	0	54	0	0	91	0	0
047	155	0	0	58	0	0	97	0	0
048	137	0	0	52	0	0	85	0	0
049	150	0	0	58	0	0	92	0	0
050	132	0	0	70	0	0	62	0	0

ตารางที่ 32 ผลเปรียบเทียบจำนวนที่นับจริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 2 (3)

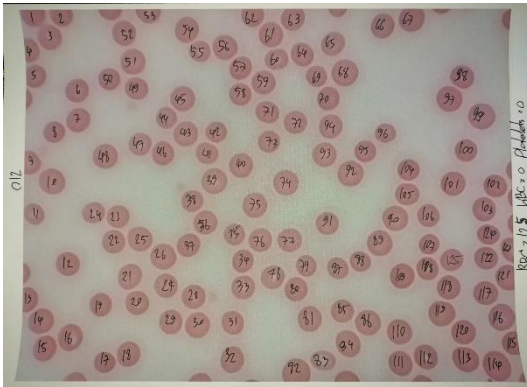
ภาพ No.	จำนวนที่นับจริง			จำนวนที่พยากรณ์ได้			ผิดพลาดสัมบูรณ์ (Error)		
	RBC	WBC	Platelets	RBC	WBC	Platelets	RBC	WBC	Platelets
051	115	0	0	54	0	0	61	0	0
052	103	0	0	46	0	0	57	0	0
053	114	0	0	42	0	0	72	0	0
054	121	0	0	49	0	0	72	0	0
055	125	0	0	59	0	0	66	0	0
056	133	2	0	62	0	0	71	2	0
057	145	0	0	61	0	0	84	0	0
058	118	0	0	50	0	0	68	0	0
059	124	0	0	53	0	0	71	0	0
060	131	0	0	48	0	0	83	0	0
061	9	2	168	4	0	0	5	2	168
062	11	3	175	0	0	0	11	3	175
063	5	4	173	2	0	0	3	4	173
064	14	7	177	5	0	0	9	7	177
065	12	3	143	6	0	0	6	3	143
066	12	1	161	1	0	0	11	1	161
067	5	1	155	7	1	0	2	0	155
068	7	1	129	0	0	0	7	1	129
069	7	1	57	0	1	0	7	0	57
070	11	2	145	4	0	0	7	2	145
071	9	2	159	3	0	0	6	2	159
072	10	1	172	3	0	0	7	1	172
073	9	2	171	1	0	0	8	2	171
074	9	2	172	0	0	0	9	2	172
075	12	1	150	1	0	0	11	1	150



ตารางที่ 33 ผลเปรียบเทียบจำนวนที่นับจริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 2 (4)

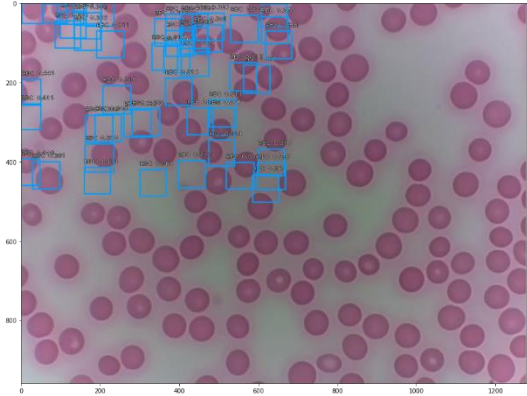
ภาพ No.	จำนวนที่นับจริง			จำนวนที่พยากรณ์ได้			ผิดพลาดสัมบูรณ์ (Error)		
	RBC	WBC	Platelets	RBC	WBC	Platelets	RBC	WBC	Platelets
076	11	2	197	1	0	0	10	2	197
077	12	1	176	1	0	0	11	1	176
078	3	3	154	4	0	0	1	3	154
079	12	9	204	14	0	0	2	9	204
080	11	3	187	4	0	0	7	3	187
081	1	5	14	1	3	0	0	2	14
082	6	6	102	0	6	0	6	0	102
083	8	4	93	5	2	0	3	2	93
084	11	4	131	9	0	0	2	4	131
085	9	4	146	4	1	0	5	3	146
086	9	6	119	4	2	0	5	4	119
087	12	4	170	8	0	0	4	4	170
088	8	5	107	6	2	0	2	3	107
089	10	6	139	4	0	0	6	6	139
090	11	8	133	6	4	0	5	4	133
091	11	5	115	8	0	0	3	5	115
092	13	5	131	5	0	0	8	5	131
093	10	5	164	8	1	0	2	4	164
094	6	3	108	1	1	0	5	2	108
095	13	8	195	5	3	0	8	5	195
096	8	4	191	5	0	0	3	4	191
097	9	9	198	5	0	0	4	9	198
098	11	8	214	4	0	0	7	8	214
099	12	6	163	4	0	0	8	6	163
100	3	6	78	4	5	0	1	1	78

ภาพที่นับด้วยตามนุษย์

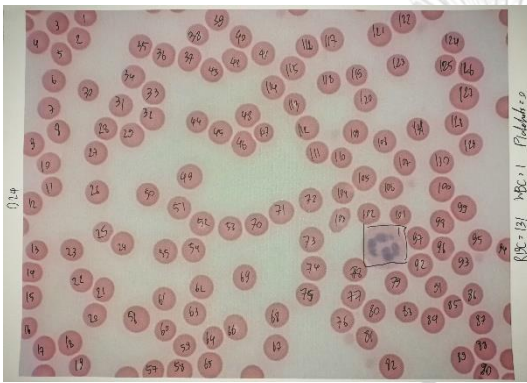


ภาพที่ 012

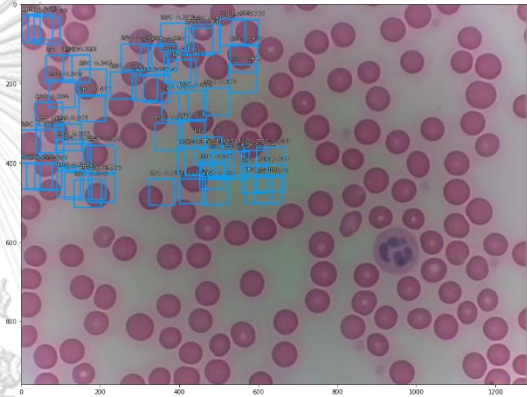
ภาพที่นับด้วยโครงข่ายประสาท



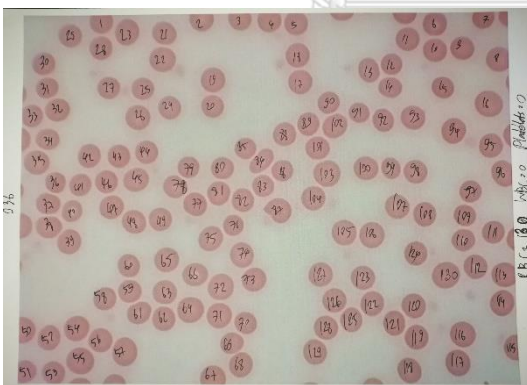
ภาพที่ 012



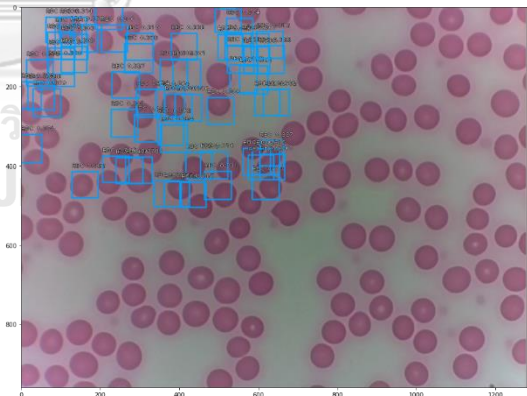
ภาพที่ 024



ภาพที่ 024



ภาพที่ 036



ภาพที่ 036

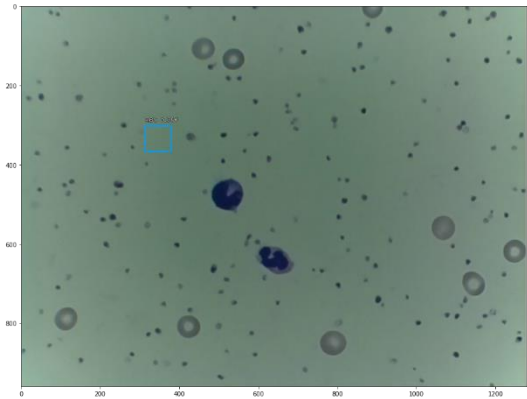
ภาพที่ 103 ผลลัพธ์การนับจำนวนเซลล์เม็ดเลือดแดงที่นับได้จริงและพยากรณ์ได้ของโมเดลปรับพารามิเตอร์แบบที่ 2

ภาพที่นับด้วยตามนุษย์

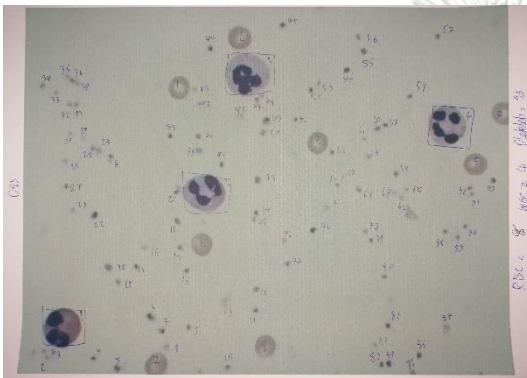


ภาพที่ 073

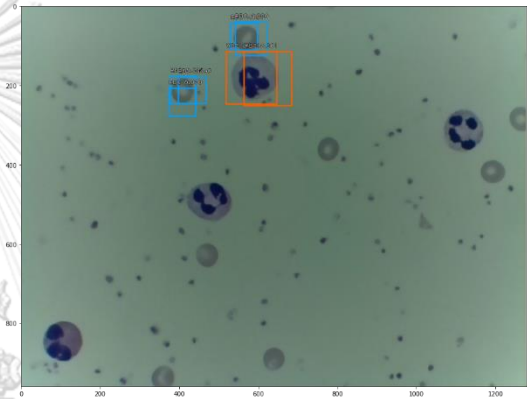
ภาพที่นับด้วยโครงข่ายประสาท



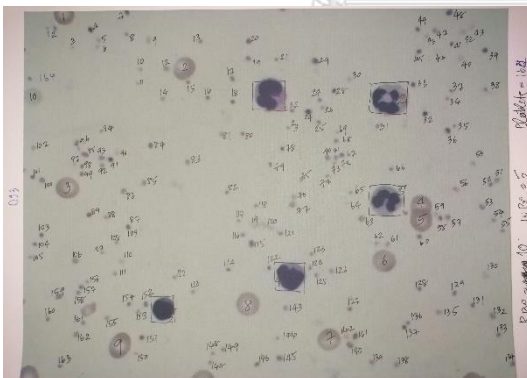
ภาพที่ 073



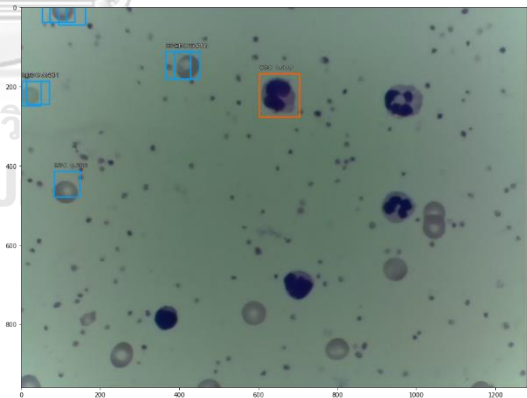
ภาพที่ 083



ภาพที่ 083



ภาพที่ 093



ภาพที่ 093

ภาพที่ 104 ผลลัพธ์การนับจำนวนเซลล์เม็ดเลือดขาวและเกล็ดเลือดที่นับได้จริงและพยากรณ์ได้ของ  
โมเดลปรับพารามิเตอร์แบบที่ 2

## บทที่ 5 บทสรุปผลงานวิจัย

### 5.1 สรุปผลการวิจัย

วิทยานิพนธ์ฉบับนี้ ได้นำเสนอการสร้างอุปกรณ์ที่ช่วยลดระยะเวลาในการนับคัดแยกที่มีความแม่นยำ ซึ่งส่งผลให้การวินิจฉัยโรคทำได้รวดเร็วมากขึ้น เรียกระบบโดยภาพรวมว่าชุดกล้องอัจฉริยะ “ไมโครซิสดีซีเอ็น” สำหรับกล้องจุลทรรศน์ (“MicrosisDCN” intelligent camera for microscope : Microbes Diagnosis with Deep Convolutional Neural Network) สำหรับแยกชนิดและนับจำนวนเซลล์ขนาดเล็กด้วยโครงข่ายประสาท เป็นชุดอุปกรณ์สำหรับสวมชุดกล้องเข้ากับท่อเลนส์ใกล้ตา (Eyepiece lens tube) ของกล้องจุลทรรศน์ทั่วไป สามารถบ่งบอกจำนวนเซลล์ขนาดเล็กที่นับได้ในพื้นที่มาตรฐานการมองเห็นของชุดกล้อง จากพื้นที่ขอบเขตการมองเห็นด้วย 11.89 ส่วน จึงจะได้เท่ากับจำนวนเซลล์ขนาดเล็กที่ตามองเห็นผ่านเลนส์ใกล้ตาที่มีกำลังขยายเป็น 400 เท่า และมีหน่วย Mitotic count มีค่าเป็น 11.89 40X “field images” to equal standard area หรือ 11.9 คูณจำนวนเซลล์ต่อ HPF (High Power Field)

ในส่วนของการขยายชุดภาพมาใช้สร้างชุดข้อมูลภาพถ่ายเซลล์เม็ดเลือดจากกล้องจุลทรรศน์ ผู้วิจัยได้พัฒนาชุดคำสั่งในการบันทึกไฟล์บรรณนิทัศน์ให้เท่ากับจำนวนชุดภาพที่ถูกขยายเพื่อแก้ไขปัญหาที่ไม่สามารถขยายชุดไฟล์ XML เก็บพิกัดพื้นที่ปิดล้อมแบบตำแหน่งตามชุดภาพที่เพิ่มขึ้นได้ และสามารถนำชุดข้อมูลไปใช้ฝึกโครงข่ายประสาทใน TensorFlow นอกเหนือจากการฝึกในวิทยานิพนธ์ฉบับนี้ ถัดมาการพัฒนาโครงข่ายประสาทภายในชุดกล้อง MicrosisDCN จากการฝึกด้วยชุดข้อมูลภาพใหม่ที่ยังไม่ได้รับการฝึกมาก่อนด้วยชุดข้อมูลภาพถ่ายเซลล์เม็ดเลือดจากกล้องจุลทรรศน์ (Blood Cell Dataset) ด้วยการปรับพารามิเตอร์แบ่งการฝึกโมเดลโครงข่ายประสาทจะถูกแบ่งออกเป็น 3 แบบ ได้แก่ การฝึกโมเดลแบบไม่มีการปรับพารามิเตอร์ การฝึกโมเดลปรับพารามิเตอร์แบบที่ 1 และการฝึกโมเดลปรับพารามิเตอร์แบบที่ 2 พบการฝึกโมเดลปรับพารามิเตอร์แบบที่ 1 มีผลลัพธ์เรียงตามชื่อคลาส เซลล์เม็ดเลือดแดง (RBC) เซลล์เม็ดเลือดขาว (WBC) และเกล็ดเลือด (Platelets) มีค่า mAP สูงสุดที่ 0.8681 ประกอบกับผลเปรียบเทียบจำนวนที่นับจริงและพยากรณ์ได้สามารถตรวจจับเซลล์ได้ครบทุกคลาส ในขณะที่การฝึกโมเดลแบบไม่ปรับพารามิเตอร์ มีผลลัพธ์ค่า mAP สูงสุดที่ 0.7356 สามารถตรวจจับเซลล์เม็ดเลือดแดงและเม็ดเลือดขาวได้ แต่จะไม่สามารถตรวจจับเกล็ดเลือดได้ และการฝึกโมเดลปรับพารามิเตอร์แบบที่ 2 มีผลลัพธ์ค่า mAP สูงสุดที่ 0.2239 โดยที่ไม่สามารถระบุตำแหน่งของเซลล์ขนาดเล็กได้ จึงทำให้การฝึกโมเดลปรับพารามิเตอร์แบบที่ 1 ในรูปแบบโมเดลอนุมาน (Inference model) สามารถนำไปใช้กับชุดกล้อง MicrosisDCN บนชุดสมองกลฝังตัว Raspberry Pi Computer Module 3+ และตัวรับรูปภาพ OV5647 ความละเอียด 5 ล้านจุดภาพ ผ่านหน้าจอบูตคำสั่งภาษา Python ได้โดยตรง

## 5.2 แนวทางการพัฒนาการวิจัย

ชุดกล้อง MicrosisDCN มีความแตกต่างจากชุดถ่ายถอดสัญญาณจากกล้องจุลทรรศน์ที่ทำหน้าที่เพียงถ่ายถอดสัญญาณภาพไปแสดงผลเท่านั้น ต้องนำไปใช้กับกล้องจุลทรรศน์เฉพาะรุ่นแบบ Trinocular รวมถึงต้องใช้โปรแกรมเฉพาะด้านที่จะต้องซื้อเพิ่มเติมจากชุดกล้อง ซึ่งเป็น Optional kit แต่ชุดกล้องที่พัฒนาขึ้นจัดเป็น Camera kit พร้อมใช้งานสามารถสวมเข้ากับท่อเลนส์ใกล้ตา (Eyepiece tube) ขนาด 23.2 30.0 และ 30.2 มิลลิเมตร ที่มาพร้อมระบบโครงข่ายประสาทที่มีความสามารถในการตรวจจับ และนับจำนวนของเซลล์ขนาดเล็กอย่างอัตโนมัติ สามารถเชื่อมต่อการใช้งานกับหน้าจอและชุดเมาส์คีย์บอร์ดได้ โดยไม่ต้องอาศัยคอมพิวเตอร์จากภายนอก แต่หากต้องการนำชุดโครงข่ายประสาทไปใช้งานบนคอมพิวเตอร์สามารถติดตั้งบนระบบปฏิบัติการได้ทั้ง Linux และ Windows ชุดข้อมูลของระบบที่ใช้ในการตรวจจับสามารถนำมาทำการฝึกเองภายหลังในกรณีที่ต้องการนำไปตรวจจับเซลล์ขนาดเล็กเฉพาะทาง หรือเชื้อที่เกิดการอุบัติใหม่ของโรคนั้น ๆ เพื่อเป็นชุดข้อมูลสำหรับองค์กรใช้สำหรับงานวิจัยเฉพาะด้าน

แผนการดำเนินงานต่อไปที่มีการใช้โครงข่ายประสาทรูปแบบอื่น ๆ ในฝึกด้วยชุดข้อมูลภาพเดียวกัน เช่น COCO, Mobilenet, SSD, YOLO, ResNet ด้วย Keras เพื่อทดสอบหาความสามารถในการจำแนกและนับจำนวนเซลล์ขนาดเล็กจากภาพถ่ายที่ได้จากกล้องจุลทรรศน์ดูจากผลลัพธ์ mAP (Mean Average Precision) ต่อไป และความสามารถในการตรวจจับเซลล์ขนาดเล็กที่หลายหลายมากขึ้น (Multiple class detection) เช่น parasite, protozoa, scrod เป็นต้น สำหรับชุดกล้อง MicrosisDCN คาดว่าจะปรับปรุงในส่วนของชุดเลนส์ที่ใช้สวมเข้ากับท่อเลนส์ใกล้ตา ให้สามารถเชื่อมต่อเข้ากับช่องรับภาพสำหรับกล้องจุลทรรศน์แบบ Trinocular เพื่อให้สามารถนำไปใช้งานกับกล้องจุลทรรศน์ได้หลากหลายมากยิ่งขึ้น

## 5.3 ปัญหาและอุปสรรค

เนื่องจากโครงข่ายประสาทที่กำลังพัฒนาอยู่มีความแม่นยำ (Accuracy) ประมาณร้อยละ 80 ถึง 90 จึงทำให้ผลการตรวจจับและนับจำนวนก่อให้เกิดความคลาดเคลื่อนจากกรณี เช่น เซลล์ขนาดเล็กซ้อนทับกัน (Object overlapping) จึงมีความจำเป็นที่ต้องพัฒนาโครงข่ายประสาทให้มีขีดความสามารถมากขึ้น เช่น การทำขยายชุดภาพเพิ่มเติม การปรับแต่งโมเดลโครงข่ายประสาทให้มีประสิทธิภาพมากขึ้น หรืออาจเลือกต้นแบบโมเดลที่เหมาะสมต่อการทำงานภายใต้ทรัพยากรของระบบที่จำกัด และการเตรียมชุดข้อมูลที่เหมาะสมถ่ายภาพได้อย่างชัดเจน และมีกระบวนการ Pre-processing ก่อนเข้าสู่การฝึก รวมถึงการเพิ่มจำนวน Epochs ให้มากขึ้นกว่าเดิม

ภาคผนวก ก. บทความวิจัยที่ตีพิมพ์ใน IECON 2020

# Applying Faster R-CNN for Hematocytes Detection on Compound Microscope with Image Sensor Device and Multiple GPU Computation

Natthakorn Kasamsunran  
Department of Electrical Engineering,  
Faculty of Engineering,  
Chulalongkorn University,  
Bangkok, Thailand.  
natthakorn.engr@gmail.com

Patrarat Chanchaithong  
Department of Veterinary Microbiology,  
Faculty of Veterinary Science,  
Chulalongkorn University,  
Bangkok, Thailand.  
patrarat.c@chula.ac.th

Norarat Wattanamongkhon  
Department of Electrical Engineering,  
Faculty of Engineering,  
Burapha University,  
Chonburi, Thailand.  
norarat@eng.buu.ac.th

Wanchalerm Pora  
Department of Electrical Engineering,  
Faculty of Engineering,  
Chulalongkorn University,  
Bangkok, Thailand.  
wanchalerm.p@chula.ac.th

Suree Pumrin  
Department of Electrical Engineering,  
Faculty of Engineering,  
Chulalongkorn University,  
Bangkok, Thailand.  
suree.p@chula.ac.th

**Abstract**— In this paper, we demonstrate a machine learning algorithm with multiple GPU processing in hematocytes detection. Object detection in compound microscopy images presents a specific task. Microscopy image import directly to a specimen slide under the compound microscope by an image sensor device. We propose Faster R-CNN with a customizing model applying cutting-edge object detection systems. The MicrosisDCN (Microbes Diagnosis with Deep Convolutional Neural Network) deploys Faster R-CNN in python script configures to access multiple GPUs computation with 7168 CUDA cores of dual GPUs with Linux command options: worker\_replicas and num\_clones equal to several graphics processing units. The datasets consist of hematocytes extracted from raw slides under a microscope. These images separate the cells of interest into three groups: red blood cell (RBCs), white blood cell (WBCs), and platelets. The training dataset consists of 80 percent of the 40,000 images. And the testing dataset consists of 20 percent of the 40,000 images. Our algorithm also provides the result of mean average precision (mAP) and enables multiple GPU training models in Tensorflow and OpenCV. The mAP is the average of average precision (AP) with intersection-over-union (IoU) in measuring the score of object detection accuracy. If the mAP score approaches 1.0, it indicates high accuracy. Our 9,000 steps valuation algorithm model by the python script estimates the mAP in 3 groups: RBCs, WBCs, and platelets as about 0.9147, 0.9664, and 0.9548. The optimum would be at nearly 12,000 steps because the algorithm estimates the mAP model as 0.96 in all types. The experiment aims to verify the neural network model using a compound microscope.

**Keywords**—Object detection, Microscopy image, Hematocytes.

## I. INTRODUCTION

A microorganism or microbe found in the environment has a capacity for a type of identity under the microscope. In a practical routine, the general method is commonly operated by specialists. They investigate the smear of specimen slides under a microscope based on their experience evaluation. This process may be accurate in many cases. It takes much time and prone to human error. Object detection with exact microbes in the microscope image has been evaluated in machine learning models with lots of interest, especially for clinical and laboratory applications [1]. So, the image dataset will capacitate the research community to create a universal standard detector with one platform. An artificial intelligent

industrial report suggests easily interface enabling laboratory staff to evaluate multiple cell phenotypes used in the machine learning model [2]. The suitable choice would approve an algorithm for microbe classification and counting. And it would provide accurate quantitative results without human error and reduce the time in progress [3].

In the diagnosis of microbial diseases, it can be determined by changes in the number of hematocytes or blood cells. For example, higher white blood cells may be caused by microbial infections. Therefore, selecting an object detection method for collecting blood cells would be an interest in the research community. Object detection in compound microscopy images presents a specific task. Microscopy images of blood cells have variations in magnification of objective lens and light expose from the microscope, in boundary, shape, density, and color. Due to variations in specimen preparation or smear method, it may include objects of the unknown category [4]. However, using machine learning to classify blood cells will assist experts in simplifying the process. Many researchers classify the cells of interest into three groups: erythrocytes also known as red blood cells or RBCs, leukocytes also known as white blood cells or WBCs and thrombocytes also known as platelets [4],[5]. Recently, an open-library for deep learning and machine learning development has many sources provided in GitHub. In this paper, we select to apply the library called TensorFlow and implement our machine learning using NVIDIA's CUDA toolkit for the NVIDIA GeForce graphics card. TensorFlow [6],[7],[8] can be executed with nominal modification on a wide variety of environment systems (e.g. Raspberry Pi, Laptop computer, Smartphone) and specific computational devices both of multi-cores of the central processing unit (CPU) or graphics processing unit (GPU).

In this paper, we present a machine learning algorithm with multiple GPU processing in hematocytes detection. The microscopy image also imports directly to a specimen slide under the compound microscope by an image sensor device. Finally, we propose Faster R-CNN [16] of cutting-edge object detection systems.

## II. RELATE WORK FOR HEMATOCYTES DETECTION

### A. The concept for Hematocytes Detection

First, we design a camera kit named MicrosisDCN (Microbes Diagnosis with Deep Convolutional Neural

This research is funded by Chulalongkorn University : CU\_GL\_62\_18\_21\_03

Network) by computer-aided design software (CAD). Then we build a strong plastic chassis that contains all the circuits together into a ready to use completed set. It supports c mount threads for an eyepiece microscope, comfortably. That thread includes 23.2 mm, 30 mm, and 30.5 mm screw sizes. The above camera kits must be able to withstand extreme shock and high temperatures, including a manual, an installation diagram. Next, our algorithm tests on a selection of sample hematocytes on the smear slide from the Wright-Giemsa Stain (e.g. [11],[12]). This sample is considered as the starting hematocytes. After that, the blood cells and impurities will be examined by looking through the eyepiece at a magnification of 40 to 1,000 times within one area of vision. The structure of a neural network is designed based on the principles of object detection libraries available in OpenCV [8], integrating with Tensorflow [6] and testing camera and microscopes as Fig 1. Then we compare it with the camera set to operate under neural network that determines cell points and counts.

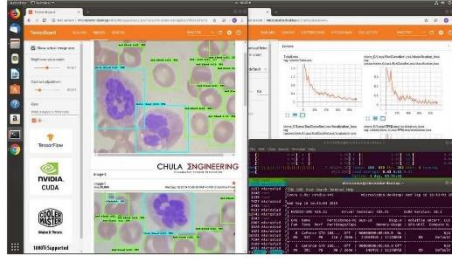


Figure 1. Our training evaluation the Faster R-CNN model at 86,896 steps.

Finally, the algorithm set computes the significance and weight of the image processing in OpenCV. Tensorflow also deploys neural network training to get a training model. The neural network learns hematocytes pattern using the image captured from a set of cameras through a microscope of more than 20,000 images with tracking coordinates  $X_{min}$ ,  $Y_{min}$ ,  $X_{max}$ , and  $Y_{max}$  in PASCAL VOC format. Our prototype works on a personal computer and a Raspberry Pi 4 with a processed train model. We transplant it into a Linux embedded system.

#### B. The comparison of different types of neural networks.

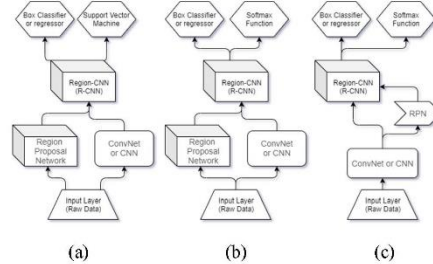


Figure 2. (a) The comparison of neural network diagram in R-CNN [15], (b) Fast R-CNN [16] and (c) Faster R-CNN [17].

The convolutional neural network (CNN) [12] has the feature to separate the image into a manifold area and then classifies each area into a variety of classes. There is a limitation in demand with lots of areas to make accurate predictions and hence high computation time. R-CNN [14]

has a searching feature to select an area. It separates approximately 2,000 areas from each image with 40-50 seconds of prediction time. There is a limitation requirement of high computational time in every area. Since it sends area to the CNN one at a time in various triple models for making predictions. In the next part, Fast R-CNN [15] has a feature that each image will be sent to CNN only once. It will retrieve the feature map searching in these maps to create predictions with the R-CNN triple model combination. The time of this prediction is 2 seconds per image. There is a limitation to slowly area searching with lone term computation time. And Faster R-CNN [16] is used for this research. It has a feature to replace the selective explore process with region proposal network (RPN) [13]. The RPN makes our algorithm faster than the previous methods. The time of this prediction is 0.2 seconds per image, but there are restrictions related to the system performance depends on system operation. The object proposal in RPN takes time since different systems are running one after the previous status.

#### III. OVERVIEW OF THE FASTER R-CNN MODEL WORKFLOW.

Our system, MicrosisDCN (Microbes Diagnosis with Deep Convolutional Neural Network), applies Faster R-CNN as the base of the model in a neural network. The neural network has components that are associated with the diagram in Fig. 2. We develop an algorithm for training both modules with common features. The determination of the positive object sample has intersection-over-union (IoU) over 0.7 for positive samples and below 0.3 for negative samples [17]. In our system, it sets to 0.5. Convolution of feature map estimates multiple areas of different ratios and scales at the midpoint. The proposed region network trains an object detection model. Next, Faster R-CNN is used to start RPN training and share hidden layers (e.g. convolutional layers, softmax layers) for RPN layers and detect function network at the same time. Finally, the algorithm repeats training for PRN and Faster R-CNN to adjust the accuracy and precision of the model.

##### A. Bounding box classifier.

The principle of using a bounding box classifier in Faster R-CNN relates to the same principle in Fast R-CNN. It improves the performance of localization for a class detection in the boundary prediction [14]. The predicted bounding box classifies training in pairs of coordinate  $\{p, g\}$ . The parameter  $\{p, g\}$  is configured to take scale-invariant transformation between centers of the bounding box and log-space transformation between widths and heights of the bounding box. All the transformation takes  $p$  to estimate ground truth bounding  $g$ ,

$$\hat{g}_x = p_w d_x(p) + p_x, \quad (1)$$

$$\hat{g}_y = p_h d_y(p) + p_y, \quad (2)$$

$$\hat{g}_w = p_w e^{d_w(p)}, \quad (3)$$

$$\hat{g}_h = p_h e^{d_h(p)}, \quad (4)$$

where  $p = (p_x, p_y, p_w, p_h)$  is a coordinate of the center of the proposal in image pixel (width and height),  $g = (g_x, g_y, g_w, g_h)$  is coordinate of ground truth bounding,  $d_k(p)$  where  $k = (x, y, w, h)$  is the predicted transformation,  $\hat{g}_k$  where  $j = (x, y, w, h)$  is the correction of predicted box estimated using the coordinate of the center of the proposal  $p$  and the predicted transformation, where  $e$  is the exponential function of  $d_k(p)$ .

Optimize four parameters in  $d_x(p)$ ,  $d_y(p)$ ,  $d_w(p)$ ,  $d_h(p)$  can solve the equation by minimizing the sum of squared error loss with regularization ridge regression, as shown in (5).

$$w = \sum_{i \in \{x,y,w,h\}} (t_i - d_i(p))^2 + \lambda \|w\|^2, \quad (5)$$

where  $w$  is the regularized least-squares objective optimization,  $\lambda$  is the parameters of learnable vector model,  $d_i(p)$  is the predicted transformation and  $t_i$  is ground truth transformations.

The regression applying then transformation is that all equation and parametric paired  $\{p, g\}$  are assigned as

$$t_x = (g_x - p_x)/p_w, \quad (6)$$

$$t_y = (g_y - p_y)/p_h, \quad (7)$$

$$t_w = \log(g_w/p_w), \quad (8)$$

$$t_h = \log(g_h/p_h), \quad (9)$$

where  $t = (t_x, t_y, t_w, t_h)$  is ground truth transformations with scale-invariant and log-space transformation,  $p = (p_x, p_y, p_w, p_h)$  is a coordinate of the center of the proposal in width and height,  $g = (g_x, g_y, g_w, g_h)$  is coordinate of ground truth bounding.

#### B. Mean average precision (mAP).

The average precision (AP) is widespread in measuring the score of object detection accuracy. It scores the average precision value for recall score between 0 to 1 or 0% to 100% in percentage score [18]. It also solves the precision-recall curve plotting by the area under a curve. Precision defines the accuracy of our predictions. The recall finds all the positive samples in predictions. The mean average precision (mAP) is the average of average precision (AP) with intersection-over-union (IoU).  $AP@.5$  means the AP with  $IoU=0.5$ . This is the veritable metric for object detection algorithms in our system. For calculating the average of the maximum precision score, we divide the recall score from 0 to 1.0 by  $[0.0, 0.1, \dots, 1.0]$  into 11 points on the curve. That precise equation is

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}, \quad (10)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}, \quad (11)$$

$$F_1\text{-score} = \frac{precision \cdot recall}{precision + recall}, \quad (12)$$

$$AP = \frac{1}{11} \sum_{r \in \{0.0, 0.1, \dots, 1.0\}} AP_r, \quad (13)$$

where true positive is a correct value of the correct label, false positive is an incorrect value of the correct label, false negative is an incorrect value of the incorrect label, and  $AP_r$  is a summation of 11 points recall scores in precision-recall under area graph plotting.

### IV. MODEL TRAINING AND STAGE PROCESS.

#### A. Datasets preprocessing

The datasets consist of hematocytes extracted from raw slide under our microscope and image slide from the public sources [11],[12]. These images separate the cells of interest into three groups: red blood cell (RBCs), white blood cell (WBCs), and platelets. The training dataset consists of 80 percent of the slide images. And the testing dataset consists of 20 percent of the remainder as Fig. 3. All 40,000 images in a dataset have determined by tracking bounding box

coordinates in PASCAL VOC format (CSV files). In the TensorFlow sub-folder, there is a file named `xml_to_csv.py` and generate `tfrecord.py`. The first script converts the XML format to CSV format. Then the second script creates the TensorFlow record format (TFRecords). When the process, it sends `.record` files in the training folder.

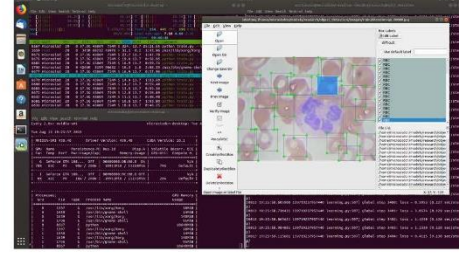


Figure 3. The hematocytes datasets being a bounding box to image labeled (RBC, WBC, and platelets) using labelling program.

#### B. Multi-GPU computation for training model.

The TensorFlow object detection classifier can be used with various pre-trained models. A Faster R-CNN model with Inception v2 architecture (`faster_rcnn_inception_v2.config`) is our selected model. This model is the basic model for a `faster_rcnn_hematocytes` model in our system. The algorithm generates the result of mean average precision (mAP). To train the hematocytes model. We apply the `train.py` in the legacy folder. Then open a command line terminal and type command in Fig. 4. This configuration enables multiple GPU training models in Tensorflow with Linux command options: `--worker_replicas` and `--num_clones` via a number of graphics processing units (GPUs) on a computer. The command `faster_rcnn_hematocytes.config` in a text editor is configured with an option, `batch_size` equal to a number of GPUs. Finally, the command sequence in Fig. 4.

```
[user](cv):~$ python legacy/train.py
--train_dir=training
--pipeline_config_path=training/
  faster_rcnn_hematocytes.config
--logtostderr --worker_replicas=2 --
num_clones=2 --ps_tasks=1
[user](cv):~$ tensorboard --logdir=training/
[user](cv):~$ python legacy/eval.py
--logtostderr
--checkpoint_dir=training
--eval_dir=eval
--pipeline_config_path=training/
  faster_rcnn_hematocytes.config
[user](cv):~$ tensorboard --logdir=eval/
```

Figure 4. The training and evaluating command using a python virtual environment.

#### C. Hematocytes detection system with a microscope.

Our system specification consists of OS Ubuntu 18.04.3 LTS, CPU AMD Ryzen 5 2600 3.4Ghz 6 cores 12 threads, GPU NVIDIA GeForce GTX 1080Ti 11Gb (Total 7168 CUDA cores with 2 GPU cards), Motherboard Asrock AM4 Asrock X470 Master SLI/AC, RAM KLEVV CRAS X RGB DDR4 3200Mhz 32GB, Storage WD BLACK WD1003FZEX 1TB & WD BLACK SN750 NVMe M.2 512GB, Cooler Master Power supply Masterwatt Maker 1500W BT, MasterCase Maker 5t, CPU Fan V8 GTS, MasterKeys Pro L, MasterMouse Pro L, RGB Hard Gaming Mousepad and Olympus CX33 compound microscope as shown in Fig. 5.



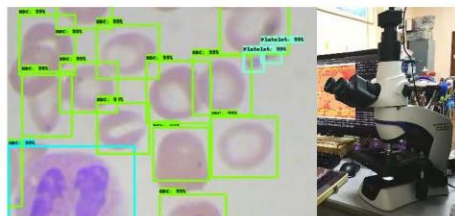


Figure 5. Our detecting blood cells system on smear slides through a microscope with a trinocular camera (prototype).

## V. RESULT AND CONCLUSION

Since the model checkpoint at 9,000 steps, the evaluation algorithm proposed by the `eval.py` script estimated the mAP for RBCs, WBCs, and platelets at about 0.9147, 0.9664 and 0.9548, respectively. In the case of 12,000 steps, our algorithm estimates mAP model at 0.96 as shown in Table I. The optimum selection would be at nearly 12,000 steps. When the procedure in the training step checks with hematocytes datasets, this may represent a false detection among RBCs and WBCs of overtraining at about 30,000 steps of the checkpoint and due to low-quality slide preparation or a blur microscope lens magnification. The average computation time per step of AMD Ryzen 5 2600 3.4Ghz 6 cores 12 threads training is 210 to 240 milliseconds (0.2 sec/step). On the other hand, the average time per step of GPU NVIDIA GeForce GTX 1080Ti with 2 cards is 110 to 180 milliseconds (0.1 sec/step).

TABLE I. THE EVALUATION RESULT IN SYSTEM.

Global steps per checkpoint	System Accuracy of Hematocytes Detection		
	RBCs (mAP)	WBCs (mAP)	Platelets (mAP)
3,000 Steps	0.8573	0.8712	0.9027
6,000 Steps	0.8656	0.8944	0.9198
9,000 Steps	0.9147	0.9358	0.9548
12,000 Steps	0.9692	0.9664	0.9703
25,000 Steps	0.9721	0.9741	0.9726
86,000 Steps	0.9869	0.9762	0.9784

This paper represents the training model using python script configures to access multiple GPU computation (with 7168 CUDA cores). It aims to verify the neural network model with a compound microscope. As a future plan, we will compare multiple neural network models for comprehensive accuracy evaluation and improve camera kits into a portable system in camera complete set on Linux embedded board. The testing guideline should be further investigated with multiple datasets (e.g. parasite, protozoa, scrod) as training models for a variety of laboratory applications.

## ACKNOWLEDGMENT

This research is funded by Chulalongkorn University : CU GI 62\_18\_21\_03. The research could not be completed without the support of the "EECU Master Study and Research Support Scholarship for Collaboration Network." Thanks to all the advisors, specialists, COOLER MASTER Co., Ltd. in Thailand for supporting the GPUs, and SPACEMED Co., Ltd. for supporting the microscope.

## REFERENCES

- [1] J. Hung, D. Ravel, S. C.P. Lopes, G. Rangel, O. A. Nery, B. Malleret, F. Nosten, M. V.G. Lacerda, M. U. Ferreira, L. Rénia, M. T. Duraisingh, F. T.M. Costa, M. Marti, and A. E. Carpenter, "Applying Faster R-CNN for Object Detection on Malaria Images," arXiv:1804.09548v2 [cs.CV], Mar 2019.
- [2] CBInsights, "The AI Industry Series: Top Healthcare AI Trends To Watch", 2019 [Online]. Available: <https://www.cbinsights.com/research/report/ai-trends-healthcare/>. [Accessed: 10-Sep-2019].
- [3] S. Mavandadi, S. Feng, F. Yu, S. Dimitrov, K. Nielsen-Saines, W. R. Prescott and A. Ozcan, "A mathematical framework for combining decisions of multiple experts toward accurate and remote diagnosis of malaria using tele-microscopy," PLoS One. 2012; 7(10): e46192. DOI: 10.1371/journal.pone.0046192, Oct 2012.
- [4] M. Khodashesan, H. Ebrahimpour-komleh and A. M. Nickfarjam, "White Blood Cell Detection and Counting based on Genetic Algorithm," Advances in Science and Engineering Technology International Conferences (ASET), Dubai, United Arab Emirates, DOI: 10.1109/ICASET.2019.8714455, May 2019.
- [5] Y. Xue and N. Ray, "Cell Detection in Microscopy Images with Deep Convolutional Neural Network and Compressed Sensing," arXiv:1708.03307v3 [cs.CV], Feb 2018.
- [6] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégasm, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu and X. Zheng, "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems," arXiv:1603.04467v2 [cs.DC], Nov 2015.
- [7] Y. J. Mo, J. Kim, J.-K. Kim, A. Mohaisen W. Lee, "Performance of Deep Learning Computation with TensorFlow Software Library in GPU-Capable Multi-Core Computing Platforms," The 9th International Conference on Ubiquitous and Future Networks (ICUFN), Milan, Italy, 2017, pp 240-242.
- [8] P.N. Druzhkov, V.L. Erukhimov, N.Y. Zolotykh, E.A. Kozinov, V.D. Kustikova, I.B. Meerov and A.N. Polovinkin, "New Object Detection Features in the OpenCV Library," Pattern Recognition and Image Analysis, 2011, Vol. 21, No. 3, pp. 384-386. DOI:10.1134/S1054661811020271, Sep 2011.
- [9] T. Lui, S. Fang, Y. Zhao, P. Wang and J. Zhang, "Implementation of Training Convolutional Neural Networks," arXiv:1506.01195v2 [cs.CV], Jun 2015.
- [10] M. M. Alam and M. T. Islam, "Machine learning approach of automatic identification and counting of blood cells," IET Healthcare Technology Letters, 2019, Vol. 6, Iss. 4, pp. 103-108. DOI: 10.1049/hlt.2018.5098, Aug 2019.
- [11] S. Rajaraman, S. Jaeger and SK. Antani, "Performance evaluation of deep neural ensembles toward malaria parasite detection in thin-blood smear images," PeerJ. :7:e6977. DOI:10.7717/peerj.6977, May 2019.
- [12] A. S. Razavian, H. Azizpour, J. Sullivan and S. Carlsson, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition," The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Ohio, USA, 2014, pp. 806-813.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," arXiv:1406.4729v4 [cs.CV], Jun 2014.
- [14] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," arXiv:1311.2524v5 [cs.CV], Oct 2014.
- [15] R. Girshick, "Fast R-CNN," arXiv:1504.08083v2 [cs.CV], Sep 2015.
- [16] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," arXiv:1506.01497v3 [cs.CV], Jan 2016.
- [17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Coumapeau, M. Brucher, M. Perrot and E. Duchesnay, "Scikit-learn: Machine Learning in Python," Journal of Machine Learning Research 12 (2011), pp. 2825-2830.
- [18] M. Everingham, S.M. A. Eslami, L.V. Gool, C. K.I. Williams, J. Winn, A. Zisserman, "The Pascal Visual Object Classes Challenge: A Retrospective," International Journal of Computer Vision (2015), Vol. 111, Iss. 1, pp. 98-136, DOI: 10.1007/s11263-014-0733.

## ภาคผนวก ข. ชุดคำสั่งติดตั้ง CUDA TOOLKIT และ CUDNN

การเตรียมระบบปฏิบัติการให้พร้อมต่อการใช้งานนั้น จะต้องมีการติดตั้งชุดคำสั่งองค์ประกอบที่เกี่ยวข้อง เพื่อให้สามารถใช้งานได้อย่างสมบูรณ์ ซึ่งในส่วนนี้จะกล่าวถึงวิธีการติดตั้งเครื่องคอมพิวเตอร์ให้สามารถใช้งานในการฝึกโมเดล รวมถึงการพัฒนาโปรแกรมบนระบบปฏิบัติการ Ubuntu 18.04 หากเป็นระบบปฏิบัติการ Windows จะต้องติดตั้งโปรแกรม Anaconda มาช่วยให้สภาพแวดล้อมจำลอง (Virtual environment) สามารถทำงานได้ ในการฝึกโมเดลให้กับการเรียนรู้ของเครื่องนั้น จะใช้หน่วยประมวลผลจาก GPU ของ Nvidia เนื่องด้วยให้ความเร็วในการประมวลผลได้รวดเร็วกว่า CPU จึงต้องติดตั้งองค์ประกอบพื้นฐานที่ทำให้เข้าถึงการใช้งานหน่วยประมวลผลกราฟิก คือ Nvidia Driver สำหรับควบคุมการทำงาน CUDA (Compute Unified Device Architecture) ซึ่งเป็นแพลตฟอร์มสำหรับการประมวลผลแบบคู่ขนาน และ cuDNN (NVIDIA CUDA Deep Neural Network library) เป็นแพลตฟอร์มเสริมของ CUDA สำหรับงานการเรียนรู้เชิงลึก ติดตั้งคลังชุดคำสั่งสำหรับ CUDA dependencies ที่จำเป็นต้องใช้ในการติดตั้งของไฟล์ .deb

```
~$ sudo apt-get install build-essential dkms
```

```
~$ sudo apt-get install freeglut3 freeglut3-dev libxi-dev libxmu-dev
```

ทำการดาวน์โหลดไฟล์ติดตั้ง CUDA Toolkit จากเว็บไซต์ <https://developer.nvidia.com/cuda-zone> ให้ดาวน์โหลดทั้ง version 10.2 สำหรับติดตั้ง Nvidia Driver และ 10.0 สำหรับ TensorFlow version 1.15

The screenshot shows the NVIDIA website's download page for CUDA Toolkit 10.2. It features a 'Select Target Platform' section with a grid of buttons for different operating systems, architectures, distributions, versions, and installer types. Below this, there is a section for downloading the installer for Linux Ubuntu 18.04 x86\_64, which includes a 'Base Installer' section with a list of terminal commands for installation.

```

$ wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1804/x86_64/cuda-ubuntu1804.pin
$ sudo mv cuda-ubuntu1804.pin /etc/apt/preferences.d/cuda-repository-pin-600
$ wget http://developer.download.nvidia.com/compute/cuda/10.2/Prod/local_installers/cuda-repo-ubuntu1804-10-2-local-10.2.89-440.33_01_1.0-1_amd64.deb
$ sudo dpkg -i cuda-repo-ubuntu1804-10-2-local-10.2.89-440.33_01_1.0-1_amd64.deb
$ sudo apt-key add /var/cuda-repo-10-2-local-10.2.89-440.33.01/7fa2af88.pub
$ sudo apt-get update
  
```

ภาพที่ 105 หน้าเว็บไซต์สำหรับดาวน์โหลด CUDA Toolkit version 10.2

```

~$ wget https://developer.download.nvidia.com/compute/cuda/repos
  /ubuntu1804/x86_64/cuda-ubuntu1804.pin\

~$ sudo mv cuda-ubuntu1804.pin /etc/apt/preferences.d/cuda-repository-
  pin-600

~$ wget http://developer.download.nvidia.com/compute/cuda/10.2/Prod
  /local_installers/cuda-repo-ubuntu1804-10-2-local-10.2.89-
  440.33.01_1.0-1_amd64.deb

~$ sudo dpkg -i cuda-repo-ubuntu1804-10-2-local-10.2.89-440.33.01_1.0
  -1_amd64.deb

~$ sudo apt-key add /var/cuda-repo-10-2-local-10.2.89-440.33.01
  /7fa2af80.pub

```

เมื่อดาวน์โหลด CUDA Toolkit version 10.2 เรียบร้อยแล้วให้พิมพ์คำสั่งเพื่อติดตั้ง Nvidia driver version 440.33.01 และ CUDA Toolkit ทั้งหมด จากนั้นจึงค่อยดาวน์โหลด CUDA Toolkit version 10.0 ติดตั้งเฉพาะส่วนของคลังชุดคำสั่งสำหรับงานการเรียนรู้เชิงลึก โดยไม่ติดตั้ง Nvidia driver ซ้ำ

```

~$ sudo apt-get update

~$ sudo apt-get -y install cuda

```

The screenshot shows the NVIDIA CUDA Toolkit 10.0 Archive website. The main heading is "CUDA Toolkit 10.0 Archive". Below it, there's a "Select Target Platform" section with a filter for "Operating System" set to "Linux". Under "Architecture", "x86\_64" is selected. Under "Distribution", "Ubuntu" is selected. Under "Version", "10.0" is selected. Under "Installer Type", "deb [local]" is selected. The page then shows "Download Installers for Linux Ubuntu 18.04 x86\_64". It lists two installers: "Base Installer" (1.4 GB) and "Patch 1 [Released May 10, 2019]" (2.3 MB). Below the "Base Installer" link, there are installation instructions:

1. 'sudo dpkg -i cuda-repo-ubuntu1804-10-0-local-10.0.130-410.48\_1.0-1\_amd64.deb'
2. 'sudo apt-key add /var/cuda-repo-10-0-local-10.0.130-410.48\_1.0-1\_amd64.pub'
3. 'sudo apt-get update'
4. 'sudo apt-get install cuda'

Other installation options are available in the form of meta-packages. For example, to install all the library packages, replace "cuda" with the "cuda-libraries-10-0" meta package. For more information on all the available meta packages click [here](#).

ภาพที่ 106 หน้าเว็บไซต์สำหรับดาวน์โหลด CUDA Toolkit version 10.0

```

:~$ sudo dpkg -i cuda-repo-ubuntu1804-10-0-local-10.0.130-410.48_1.0
-1_amd64.deb
:~$ sudo apt-key add /var/cuda-repo-10-0-local-10.0.130-410.48/7fa2af80.pub
:~$ sudo apt-get update
:~$ sudo apt-get install cuda-toolkit-10-0 cuda-tools-10-0 cuda-runtime-10-0
cuda-compiler-10-0 cuda-libraries-10-0 cuda-libraries-dev-10-0

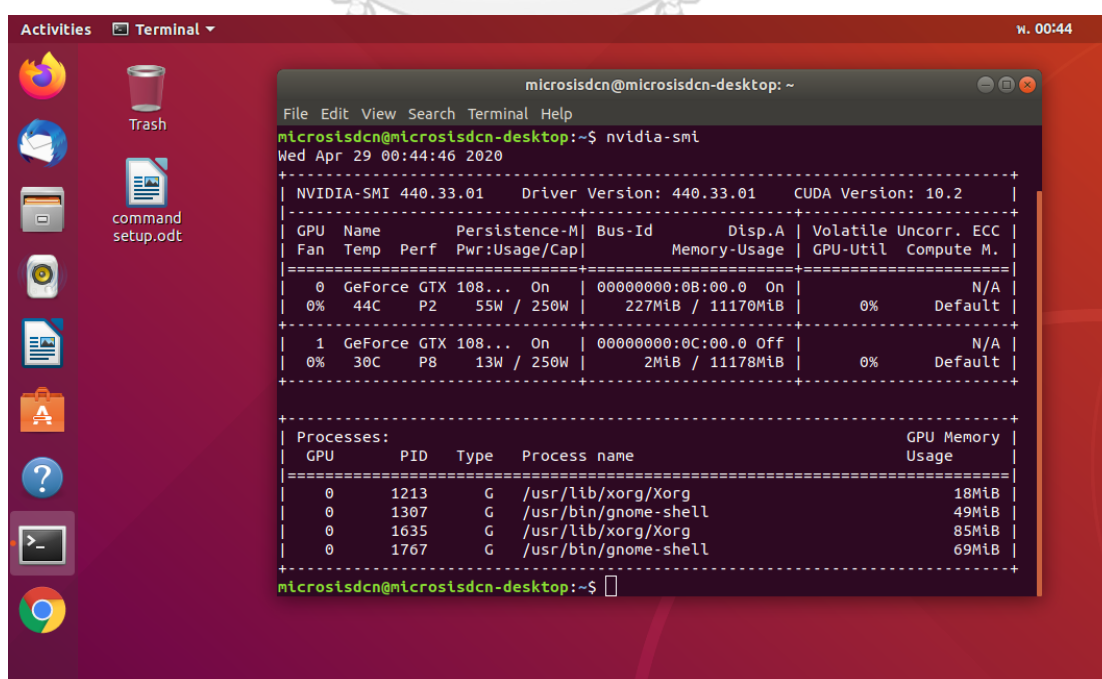
```

หากติดตั้งทั้ง Nvidia driver และ CUDA Toolkit เสร็จสมบูรณ์แล้ว ให้ทำการรีสตาร์ทเครื่องคอมพิวเตอร์ และเปิดหน้าต่าง Terminal เพิ่มคำสั่งเพื่อตรวจสอบว่าได้ติดตั้งครบถ้วนแล้ว คลังชุดคำสั่ง nvidia-smi จะเรียกข้อมูลของหน่วยประมวลผลกราฟิกมีอยู่ในเครื่องคอมพิวเตอร์ ตัวอย่างนี้แสดงให้เห็นว่ามี GPU อยู่ทั้งหมด 2 หน่วย คือ Nvidia GeForce GTX 1080 Ti ที่มีข้อมูลความเร็วพัดลม อุณหภูมิ กำลังไฟฟ้าที่ใช้ขณะนั้น ความจุของหน่วยความจำ ร้อยละการประมวลผลของ GPU ส่วนถัดมา Processes คืองานที่ GPU กำลังประมวลผลอยู่ สามารถจำแนกได้ตาม PID ที่แสดงอยู่ขณะนั้น และแบ่งได้ตามหมายเลขของ GPU คือ 0 และ 1

```

:~$ nvidia-smi

```



```

microsisdcn@microsisdcn-desktop: ~
File Edit View Search Terminal Help
microsisdcn@microsisdcn-desktop:~$ nvidia-smi
Wed Apr 29 00:44:46 2020
+-----+
| NVIDIA-SMI 440.33.01    Driver Version: 440.33.01    CUDA Version: 10.2     |
+-----+-----+-----+-----+-----+-----+
| GPU  Name      Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
+-----+-----+-----+-----+-----+-----+
| 0   GeForce GTX 108...  On      | 00000000:0B:00.0  On      |         N/A         |
| 0%   44C    P2    55W / 250W | 227MiB / 11170MiB |    0%      Default   |
+-----+-----+-----+-----+-----+-----+
| 1   GeForce GTX 108...  On      | 00000000:0C:00.0  Off     |         N/A         |
| 0%   30C    P8    13W / 250W |  2MiB / 11178MiB |    0%      Default   |
+-----+-----+-----+-----+-----+-----+
+-----+-----+-----+-----+-----+
| Processes:                                                       GPU Memory |
|  GPU       PID    Type   Process name                               Usage      |
+-----+-----+-----+-----+-----+-----+
|    0       1213    G     /usr/lib/xorg/Xorg                          18MiB      |
|    0       1307    G     /usr/bin/gnome-shell                       49MiB      |
|    0       1635    G     /usr/lib/xorg/Xorg                          85MiB      |
|    0       1767    G     /usr/bin/gnome-shell                       69MiB      |
+-----+-----+-----+-----+-----+-----+
microsisdcn@microsisdcn-desktop:~$ █

```

ภาพที่ 107 หน้าต่างที่ถูกเรียกใช้จากคลังชุดคำสั่ง nvidia-smi แสดงคุณสมบัติของ GPU

The image shows a web browser window displaying the 'cuDNN Download' page. The page title is 'cuDNN Download' and it includes a link to the 'I Agree To the Terms of the cuDNN Software License Agreement'. Below this, there are sections for downloading cuDNN v7.6.5 (November 18th, 2019) for CUDA 10.2, and links to libraries for Windows, Mac, Linux, Ubuntu, and RedHat/CentOS (x86\_64 architecture). The terminal window shows the following commands and output:

```

microstddcn@microsoft-desktop:~$ sudo mv cuda-ubuntu1804.pin /etc/apt/preferences.d/cuda-repository-pin-600
microstddcn@microsoft-desktop:~$ wget http://developer.download.nvidia.com/compute/cuda/10.2/Prod/Local_installers/cuda-repo-ubuntu1804-10-2-local-10.2.89-440.33.01_1.0-1_and64-deb
--2020-04-29 00:07:58-- http://developer.download.nvidia.com/compute/cuda/10.2/Prod/Local_installers/cuda-repo-ubuntu1804-10-2-local-10.2.89-440.33.01_1.0-1_and64-deb
Resolving developer.download.nvidia.com (developer.download.nvidia.com)... 152.199.39.144
Connecting to developer.download.nvidia.com (developer.download.nvidia.com)[152.199.39.144]:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1896278608 (1.8G) [application/x-deb]
Saving to: 'cuda-repo-ubuntu1804-10-2-local-10.2.89-440.33.01_1.0-1_and64.deb'
cuda-repo- 62%[=====] 1.11G 3.39MB/s eta 9m 4s

```

ภาพที่ 108 หน้าเว็บไซต์สำหรับดาวน์โหลด cuDNN version 7.6.5

ส่วนถัดมาจะเป็นการติดตั้งส่วนเสริมของ CUDA Toolkit สำหรับนำมาใช้งานทางด้านการเรียนรู้ของเครื่องด้วย cuDNN (NVIDIA CUDA Deep Neural Network library) ซึ่งจะต้องสมัครสมาชิกกับ NVIDIA Developer ก่อนจึงจะสามารถดาวน์โหลดชุดไฟล์สำหรับติดตั้ง โดยจะต้องดาวน์โหลดทั้ง 3 ส่วน คือ cuDNN Runtime Library for Ubuntu18.04 (Deb), cuDNN Developer Library for Ubuntu18.04 (Deb) และ cuDNN Code Sample and User Guide for Ubuntu18.04 จากคำสั่งด้านล่างนี้

```

~:$ wget https://developer.nvidia.com/compute/machine-learning/cudnn
  /secure/7.6.5.32/Production/10.2_20191118/Ubuntu18_04-x64
  /libcudnn7_7.6.5.32-1+cuda10.2_amd64.deb
~:$ wget https://developer.nvidia.com/compute/machine-learning/cudnn
  /secure/7.6.5.32/Production/10.2_20191118/Ubuntu18_04-x64
  /libcudnn7-dev_7.6.5.32-1+cuda10.2_amd64.deb
~:$ wget https://developer.nvidia.com/compute/machine-learning/cudnn
  /secure/7.6.5.32/Production/10.2_20191118/Ubuntu18_04-x64
  /libcudnn7-doc_7.6.5.32-1+cuda10.2_amd64.deb
~:$ sudo dpkg -i libcudnn7_7.6.5.32-1+cuda10.2_amd64.deb
~:$ sudo dpkg -i libcudnn7-dev_7.6.5.32-1+cuda10.2_amd64.deb
~:$ sudo dpkg -i libcudnn7-doc_7.6.5.32-1+cuda10.2_amd64.deb

```

หลังจากติดตั้งคลังชุดคำสั่ง cuDNN เสร็จสิ้นแล้วสามารถตรวจสอบว่า CUDA ทำงานสมบูรณ์หรือไม่นั้น สามารถเรียกการทำงานของหนึ่งในคลังชุดคำสั่ง คือ mnistCUDA จากคำสั่งด้านล่าง หากทำการติดตั้งเรียบร้อยแล้วจะแสดงผลว่า Test passed!

```
~$ cp -r /usr/src/cudnn_samples_v7/ ~.
```

```
~$ cd ~/cudnn_samples_v7/mnistCUDA.
```

```
~$ make clean && make.
```

```
~$ ./mnistCUDA
```

```

microsidscn@microsidscn-desktop:~/cudnn_sample_v7/mnistCUDA
$ nvidia-smi
Wed Apr 29 08:49:07 2020
+-----+
| NVIDIA-SMI 440.33.01    Driver Version: 440.33.01    CUDA Version: 10.2     |
|-----+-----+
| GPU Name                   Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|  Memory-Usage | GPU-Util  Compute M. |
|-----+-----+-----+
| 0  GeForce GTX 108...    On          | 00000000:0B:00:0  On      | N/A              |
| 4%   47C    P8     13W / 250W  | 489MiB / 11170MiB | 0%         Default   |
|-----+-----+-----+
| 1  GeForce GTX 108...    On          | 00000000:0C:00:0  Off     | N/A              |
| 0%   30C    P8     14W / 250W  |  2MiB / 11170MiB | 0%         Default   |
|-----+-----+-----+
| Processes:                                                       GPU Memory |
|  GPU   PID    Type   Process name      Usage    |
|-----+-----+-----+
| 0      1213   G      /usr/lib/xorg/Xorg      18MiB |
| 0      1307   G      /usr/bin/gnome-shell   49MiB |
| 0      1635   G      /usr/lib/xorg/Xorg      22MiB |
| 0      1767   G      /usr/bin/gnome-shell   108MiB |
+-----+-----+
microsidscn@microsidscn-desktop:~$
$ cd ~/cudnn_sample_v7/mnistCUDA
$ make clean && make
Cleaning...
Compiling...
$ ./mnistCUDA
Testing single precision
Loading Image data/one_28x28.pgm
Performing forward propagation ...
Testing cudnnGetConvolutionForwardAlgorithm ...
Fastest algorithm is Algo 1
Testing cudnnFindConvolutionForwardAlgorithm ...
**** CUDNN_STATUS_SUCCESS for Algo 0: 0.018336 time requiring 0 memory
**** CUDNN_STATUS_SUCCESS for Algo 1: 0.033792 time requiring 3464 memory
**** CUDNN_STATUS_SUCCESS for Algo 2: 0.049856 time requiring 5760 memory
**** CUDNN_STATUS_SUCCESS for Algo 5: 0.052596 time requiring 28308 memory
**** CUDNN_STATUS_SUCCESS for Algo 7: 0.066560 time requiring 285744 memory
Resulting weights from Softmax:
0.000000 0.999939 0.000000 0.000000 0.000051 0.000000 0.000012 0.000017 0.000010 0.000000
Loading Image data/three_28x28.pgm
Performing forward propagation ...
Resulting weights from Softmax:
0.000000 0.000000 0.000000 0.999928 0.000000 0.000071 0.000000 0.000000 0.000000 0.000000
Loading Image data/five_28x28.pgm
Performing forward propagation ...
Resulting weights from Softmax:
0.000000 0.000000 0.000000 0.000002 0.000000 0.999920 0.000154 0.000000 0.000012 0.000006
Result of classification: 1 3 5
Test passed!

Testing half precision (math in single precision)
Loading Image data/one_28x28.pgm
Performing forward propagation ...
Testing cudnnGetConvolutionForwardAlgorithm ...
Fastest algorithm is Algo 1
Testing cudnnFindConvolutionForwardAlgorithm ...
**** CUDNN_STATUS_SUCCESS for Algo 0: 0.015360 time requiring 0 memory
**** CUDNN_STATUS_SUCCESS for Algo 1: 0.028672 time requiring 3464 memory
**** CUDNN_STATUS_SUCCESS for Algo 2: 0.035840 time requiring 28800 memory
**** CUDNN_STATUS_SUCCESS for Algo 5: 0.049152 time requiring 28308 memory
**** CUDNN_STATUS_SUCCESS for Algo 4: 0.065536 time requiring 287360 memory
Resulting weights from Softmax:
0.000001 1.000000 0.000001 0.000000 0.000051 0.000001 0.000012 0.000017 0.000010 0.000001
Loading Image data/three_28x28.pgm
Performing forward propagation ...
Resulting weights from Softmax:
0.000000 0.000000 0.000000 1.000000 0.000000 0.000071 0.000000 0.000000 0.000000 0.000000
Loading Image data/five_28x28.pgm
Performing forward propagation ...
Resulting weights from Softmax:
0.000000 0.000000 0.000000 0.000002 0.000000 1.000000 0.000154 0.000000 0.000012 0.000006
Result of classification: 1 3 5
Test passed!
microsidscn@microsidscn-desktop:~/cudnn_sample_v7/mnistCUDA$

```

ภาพที่ 109 หน้าต่าง Terminal แสดงการทดสอบ cuDNN ด้วยคำสั่ง mnistCUDA

## ภาคผนวก ค. ชุดคำสั่งติดตั้ง OPENCV และ TENSORFLOW 1.15

```

microsisdcn@microsisdcn-desktop: ~
File Edit View Search Terminal Help
libffms2-4 libfwup1 libgpac4 libprotobuf-lite10 libprotoc10 libraw1394-dev libraw1394-tools
libwayland-egl1-mesa
Use 'sudo apt autoremove' to remove them.
0 upgraded, 0 newly installed, 0 to remove and 8 not upgraded.
microsisdcn@microsisdcn-desktop:~$ sudo apt-get -y install libjpeg-dev libpng-dev libtiff-dev libjpeg
8-dev libtiff5-dev libtiff-dev libavcodec-dev libavformat-dev libswscale-dev libdc1394-22-dev libxine
2-dev libv4l-dev libgtk-3-dev libgtk2.0-dev libtbb-dev qt5-default libatlas-base-dev libmp3lame-dev l
ibtheora-dev libvorbis-dev libxvidcore-dev libx264-dev libopencore-amrnb-dev libopencore-amrwb-dev li
bavresample-dev x264 v4l-utils libprotobuf-dev protobuf-compiler libgoogle-glog-dev libgflags-dev lib
gphoto2-dev libeigen3-dev libhdf5-dev doxygen
Reading package lists... Done
Building dependency tree
Reading state information... Done
doxygen is already the newest version (1.8.13-10).
libgphoto2-dev is already the newest version (2.5.16-2).
libgtk2.0-dev is already the newest version (2.24.32-1ubuntu1).
libjpeg-dev is already the newest version (8c-2ubuntu8).
libjpeg8-dev is already the newest version (8c-2ubuntu8).
libmp3lame-dev is already the newest version (3.100-2).
libtheora-dev is already the newest version (1.1.1+dfsg.1-14).
libv4l-dev is already the newest version (1.14.2-1).
libvorbis-dev is already the newest version (1.3.5-4.2).
libatlas-base-dev is already the newest version (3.10.3-5).
libgflags-dev is already the newest version (2.2.1-1).
libgoogle-glog-dev is already the newest version (0.3.5-1).

```

ภาพที่ 110 หน้าต่าง Terminal แสดงการติดตั้ง dependencies ที่จำเป็นของ OpenCV

ส่วนสำคัญในการพัฒนาระบบ คือ คลังชุดคำสั่งสำหรับประมวลผลภาพ OpenCV และ สำหรับพัฒนาระบบการเรียนรู้เชิงลึกและการเรียนรู้ของเครื่องทั้งหมด TensorFlow จะทำการติดตั้งสภาพแวดล้อมจำลองของ Python (Python virtual environment) เพื่อป้องกันปัญหาของการอัปเดตคลังชุดคำสั่ง หรือความผิดพลาดในการแก้ไขของค์ประกอบของระบบที่จะส่งผลให้โปรแกรมไม่สามารถทำงานได้ ติดตั้งคลังชุดคำสั่งสำหรับ OpenCV dependencies

### CHULALONGKORN UNIVERSITY

```

:~$ sudo apt-get -y install build-essential checkinstall cmake unzip pkg-config
yasm git gfortran

```

```

:~$ sudo apt-get -y install libjpeg-dev libpng-dev libtiff-dev libjpeg8-dev
libtiff5-dev libtiff-dev libavcodec-dev libavformat-dev libswscale-dev
libdc1394-22-dev libxine2-dev libv4l-dev libgtk-3-dev libgtk2.0-dev
libtbb-dev qt5-default libatlas-base-dev libmp3lame-dev
libtheora-dev libvorbis-dev libxvidcore-dev libx264-dev
libopencore-amrnb-dev libopencore-amrwb-dev libavresample-dev
x264 v4l-utils libprotobuf-dev protobuf-compiler libgoogle-glog-dev
libgflags-dev libgphoto2-dev libeigen3-dev libhdf5-dev doxygen

```

```

:~$ sudo apt-get install python3-dev python3-pip python3-testresources
python-pydot python-pydot-ng python3-tk graphviz

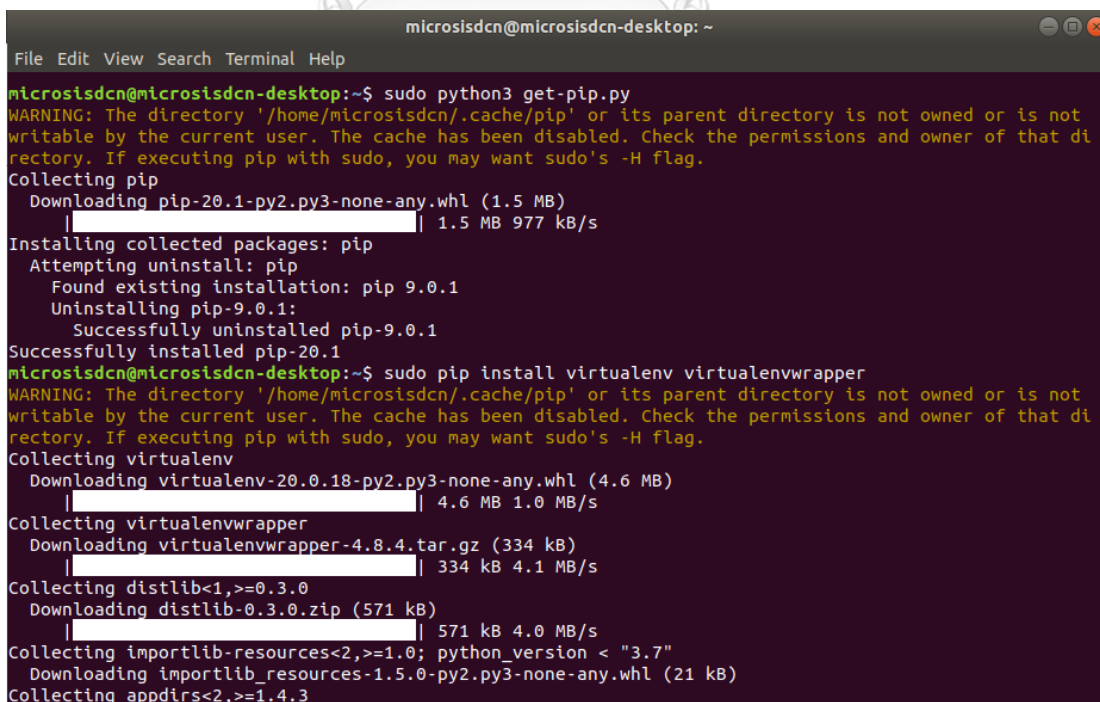
```

ทำการดาวน์โหลดคลังชุดคำสั่ง OpenCV (Open Source Computer Vision Library) จากเว็บไซต์ <https://github.com/opencv/> ซึ่งจะมี 2 repositories คือ opencv ที่เป็น main package หลักและ opencv\_contrib ซึ่งเป็นชุดคำสั่งเสริมสำหรับ opencv ตัวอย่าง เช่น คลังชุดคำสั่งที่ดึง CUDA มาใช้งาน cudacodec, cudafilter เป็นต้น จากนั้นจึงติดตั้งชุดคำสั่งสำหรับสร้างสภาพแวดล้อมจำลองของ Python และการเรียกใช้งาน PyPI (Python Package Index) ในการช่วยการติดตั้งส่วนเสริมของ Python ด้วยคำสั่ง pip หรือ easy\_install

```

:~$ git clone https://github.com/opencv/opencv.git
:~$ git clone https://github.com/opencv/opencv_contrib.git
:~$ wget https://bootstrap.pypa.io/get-pip.py
:~$ sudo python3 get-pip.py
:~$ sudo pip install virtualenv virtualenvwrapper
:~$ sudo rm -rf ~/get-pip.py ~/.cache/pip

```



```

microsisdcn@microsisdcn-desktop: ~
File Edit View Search Terminal Help
microsisdcn@microsisdcn-desktop:~$ sudo python3 get-pip.py
WARNING: The directory '/home/microsisdcn/.cache/pip' or its parent directory is not owned or is not
writable by the current user. The cache has been disabled. Check the permissions and owner of that di
rectory. If executing pip with sudo, you may want sudo's -H flag.
Collecting pip
  Downloading pip-20.1-py2.py3-none-any.whl (1.5 MB)
    |-----| 1.5 MB 977 kB/s
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 9.0.1
    Uninstalling pip-9.0.1:
      Successfully uninstalled pip-9.0.1
  Successfully installed pip-20.1
microsisdcn@microsisdcn-desktop:~$ sudo pip install virtualenv virtualenvwrapper
WARNING: The directory '/home/microsisdcn/.cache/pip' or its parent directory is not owned or is not
writable by the current user. The cache has been disabled. Check the permissions and owner of that di
rectory. If executing pip with sudo, you may want sudo's -H flag.
Collecting virtualenv
  Downloading virtualenv-20.0.18-py2.py3-none-any.whl (4.6 MB)
    |-----| 4.6 MB 1.0 MB/s
Collecting virtualenvwrapper
  Downloading virtualenvwrapper-4.8.4.tar.gz (334 kB)
    |-----| 334 kB 4.1 MB/s
Collecting distlib<1,>=0.3.0
  Downloading distlib-0.3.0.zip (571 kB)
    |-----| 571 kB 4.0 MB/s
Collecting importlib-resources<2,>=1.0; python_version < "3.7"
  Downloading importlib_resources-1.5.0-py2.py3-none-any.whl (21 kB)
Collecting appdirs<2,>=1.4.3

```

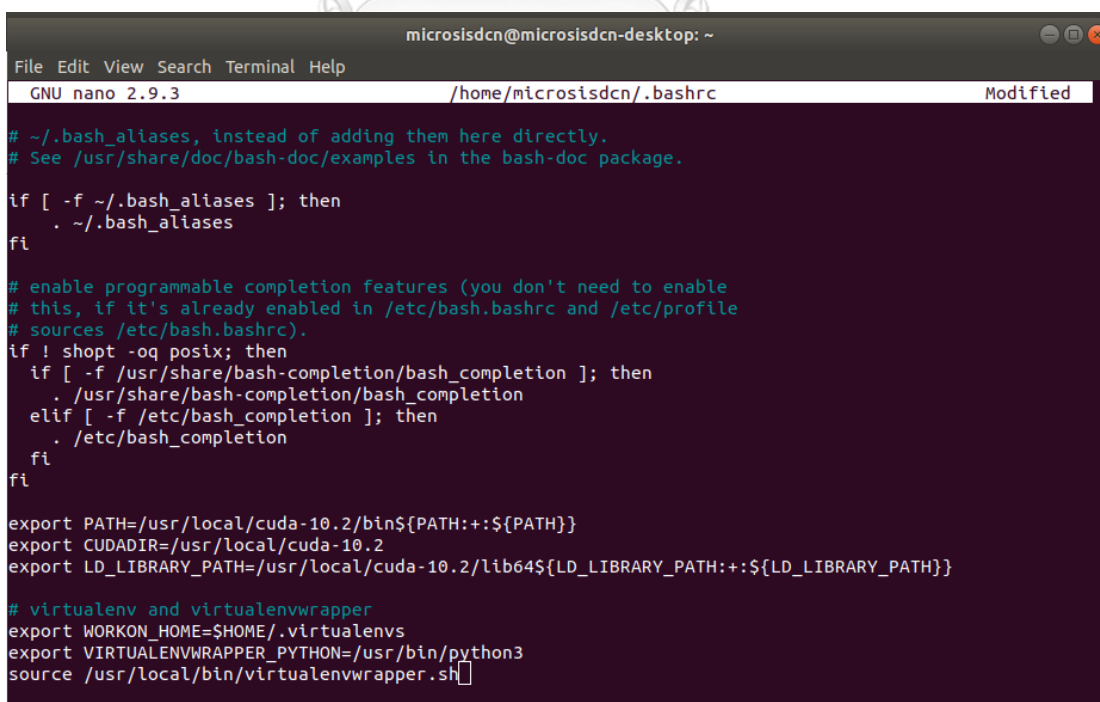
ภาพที่ 111 การติดตั้งสภาพแวดล้อมจำลองของ Python และ PyPI



เนื่องจากระบบปฏิบัติการ Ubuntu จะต้องทราบถึงเส้นทางการเข้าถึงไฟล์ข้อมูล (Files directory path หรือ Path) เพื่อให้สามารถเรียกคำสั่งที่ช่วยให้สภาพแวดล้อมจำลองของ Python ทำงานได้อย่างสมบูรณ์ ทำให้ต้องแก้ไขไฟล์ BASH (Bourne-Again Shell) ที่ทำหน้าที่รับคำสั่งจากผู้ใช้งานส่งต่อไปยัง OS Kernel ด้วยการพิมพ์คำสั่งในหน้าต่าง terminal ว่า `~$ sudo nano ~/.bashrc` และเพิ่มชุดคำสั่งด้านล่าง หลังจากบันทึกการแก้ไขไฟล์ `.bashrc` และทุกครั้งที่จะเริ่มใช้งานจะต้องพิมพ์คำสั่ง `~$ source ~/.bashrc` ทุกครั้ง การสร้างสภาพแวดล้อมจำลองของ Python ในครั้งแรกนั้น จะต้องกำหนดว่า ชื่อสภาพแวดล้อมจำลอง และจะใช้ Python Interpreter รุ่นใด เมื่อสร้างเสร็จแล้วต้องการเข้าสู่สภาพแวดล้อมจำลอง สามารถพิมพ์คำสั่ง `workon [ชื่อ]` ได้

```
# virtualenv and virtualenvwrapper
export WORKON_HOME=$HOME/.virtualenvs
export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
source /usr/local/bin/virtualenvwrapper.sh

~$ mkvirtualenv cv -p python3
~$ workon cv
```



```
microsisdcn@microsisdcn-desktop: ~
File Edit View Search Terminal Help
GNU nano 2.9.3 /home/microsisdcn/.bashrc Modified
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
  . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi

export PATH=/usr/local/cuda-10.2/bin${PATH:+:${PATH}}
export CUDADIR=/usr/local/cuda-10.2
export LD_LIBRARY_PATH=/usr/local/cuda-10.2/lib64${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}

# virtualenv and virtualenvwrapper
export WORKON_HOME=$HOME/.virtualenvs
export VIRTUALENVWRAPPER_PYTHON=/usr/bin/python3
source /usr/local/bin/virtualenvwrapper.sh
```

ภาพที่ 112 การใช้โปรแกรม Nano ในการแก้ไขชุดคำสั่งภายในไฟล์ `.bashrc`

เมื่อติดตั้งและเข้าสู่สภาพแวดล้อมจำลอง Python ได้แล้ว ให้สังเกตจะมี (cv) ปรากฏขึ้น ซึ่งทุกการติดตั้งคลังชุดคำสั่งหรือส่วนเริ่มด้วย PyPI (Python Package Index) จะเกิดขึ้นภายในนี้ โดยไม่ไปกระทบกับระบบหลักภาพนอก ในการติดตั้ง OpenCV นั้นจะมีการเรียกใช้งานคลังชุดคำสั่ง NumPy (Numeric Python) เป็นโมดูลส่วนเสริมทางคณิตศาสตร์ และตรรกะการคำนวณ เพราะการประมวลผลภาพจะต้องมีการคำนวณ เช่น คำนวณเมทริกซ์ของภาพ เป็นต้น นอกจากนี้ยังสามารถติดตั้งคลังชุดคำสั่งที่จำเป็นต่อการติดตั้ง TensorFlow ในส่วนถัดไปได้ เช่น SciPy โมดูลส่วนเริ่มสำหรับงานทางวิทยาการข้อมูลและการแสดงผลแบบ visualization เป็นต้น

```
(cv) :~$ pip install numpy scipy pandas h5py scikit-learn scikit-image wheel
QtPy matplotlib Cython imgaug imutils jupyterlab notebook
```



```
Stored in directory: /home/microsisdcn/.cache/pip/wheels/b4/cb/f1/d142b3bb45d488612cf3943d8a1db090e
b95e6687045ba61d1
Successfully built imutils tornado prometheus-client pandocfilters pyrsistent backcall
Installing collected packages: numpy, scipy, six, h5py, joblib, scikit-learn, decorator, networkx, pi
llow, cycloper, pyparsing, python-dateutil, kiwisolver, matplotlib, PyWavelets, imageio, scikit-image,
QtPy, Cython, opencv-python, Shapely, imgaug, imutils, MarkupSafe, Jinja2, tornado, pyzmq, ipython-ge
nutils, traitlets, jupyter-core, entrypoints, pyrsistent, attrs, zipp, importlib-metadata, jsonschema
, nbformat, testpath, pandocfilters, webencodings, bleach, mistune, defusedxml, pygments, nbconvert,
jupyter-client, ptyprocess, pexpect, parso, jedi, backcall, wcwidth, prompt-toolkit, pickleshare, ipy
thon, ipykernel, terminado, Send2Trash, prometheus-client, notebook, urllib3, idna, chardet, certifi,
requests, json5, jupyterlab-server, jupyterlab
Successfully installed Cython-0.29.17 MarkupSafe-1.1.1 PyWavelets-1.1.1 QtPy-1.9.0 Send2Trash-1.5.0 S
hapely-1.7.0 attrs-19.3.0 backcall-0.1.0 bleach-3.1.4 certifi-2020.4.5.1 chardet-3.0.4 cycloper-0.10.0
decorator-4.4.2 defusedxml-0.6.0 entrypoints-0.3 h5py-2.10.0 idna-2.9 imageio-2.8.0 imgaug-0.4.0 impo
rtlib-metadata-1.6.0 imutils-0.5.3 ipykernel-5.2.1 ipython-7.13.0 ipython-genutils-0.2.0 jedi-0.17.0
Jinja2-2.11.2 joblib-0.14.1 json5-0.9.4 jsonschema-3.2.0 jupyter-client-6.1.3 jupyter-core-4.6.3 jupy
terlab-2.1.1 jupyterlab-server-1.1.1 kiwisolver-1.2.0 matplotlib-3.2.1 mistune-0.8.4 nbconvert-5.6.1
nbformat-5.0.6 networkx-2.4 notebook-6.0.3 numpy-1.18.3 opencv-python-4.2.0.34 pandocfilters-1.4.2 pa
rso-0.7.0 pexpect-4.8.0 pickleshare-0.7.5 pillow-7.1.2 prometheus-client-0.7.1 prompt-toolkit-3.0.5 p
typrocess-0.6.0 pygments-2.6.1 pyparsing-2.4.7 pyrsistent-0.16.0 python-dateutil-2.8.1 pyzmq-19.0.0 r
equests-2.23.0 scikit-image-0.16.2 scikit-learn-0.22.2.post1 scipy-1.4.1 six-1.14.0 terminado-0.8.3 t
estpath-0.4.4 tornado-6.0.4 traitlets-4.3.3 urllib3-1.25.9 wcwidth-0.1.9 webencodings-0.5.1 zipp-3.1.
0
WARNING: You are using pip version 20.0.2; however, version 20.1 is available.
You should consider upgrading via the '/home/microsisdcn/.virtualenvs/cv/bin/python -m pip install --
upgrade pip' command.
(cv) microsisdcn@microsisdcn-desktop:~$
```

ภาพที่ 113 หน้าต่าง Terminal แสดงการติดตั้งคลังชุดคำสั่งส่วนเสริมด้วย pip install

จากที่ได้ดาวโหลดชุดคำสั่งของ OpenCV มานั้นยังอยู่ในรูปของ source code ที่ยังไม่ถูก compile จึงต้อง compile ให้เป็นชุดคำสั่งที่สามารถติดตั้งลงในระบบได้ ให้สร้างแฟ้ม build ภายในแฟ้ม OpenCV จากนั้นจึงเรียกคำสั่ง cmake ซึ่งเป็นโปรแกรมสร้าง build script ตามที่ได้ตั้งค่าไว้ ระหว่างที่กำลังทำงานสามารถสังเกต Error ได้ หาก compile เสร็จสิ้นจะแสดงข้อความ Configuring done และ Generating done ตามลำดับ

```
(cv) :~$ cd ~/opencv
```

```
(cv) :~$ mkdir build && cd build
```

```
(cv) :~$ cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D INSTALL_PYTHON_EXAMPLES=ON \
-D OPENCV_GENERATE_PKGCONFIG=ON \
-D WITH_CUDA=ON \ -D OPENCV_DNN_CUDA=ON \
-D INSTALL_C_EXAMPLES=OFF \
-D OPENCV_ENABLE_NONFREE=ON \
-D OPENCV_EXTRA_MODULES_PATH=~/.opencv_contrib/modules \
-D PYTHON_EXECUTABLE=~/.virtualenvs/cv/bin/python \
-D BUILD_EXAMPLES=ON ..
```

เพื่อความถูกต้อง ควรตรวจสอบ logging ที่แสดงช่วงสุดท้ายของการ compile ว่ามีคำสั่งชุดคำสั่งหรือส่วนเสริมใดที่สามารถใช้งานได้ Interpreter รุ่นใด และ build script นี้ถูกสร้างขึ้นเพื่อติดตั้งบนสภาพแวดล้อมจำลองของ Python หรือไม่ ดูได้จาก Python (for build)

```

Terminal
microstsdcn@microstsdcn-desktop:~/opencv/build

-- General configuration for OpenCV 4.3.0-dev =====
-- Version control:
--   4.3.0-201-gc72262f28
--
-- Extra modules:
--   Location (extra): /home/microstsdcn/opencv_contrib/modules
--   Version control (extra): 4.3.0-33-g99de623
--
-- Platform:
--   Timestamp: 2020-04-29T03:18:50Z
--   Host: Linux 5.3.0-46-generic x86_64
--   CMake: 3.10.2
--   Cmake generators: Unix Makefiles
--   Cmake build tool: /usr/bin/make
--   Configuration: RELEASE
--
-- CPU/HW Features:
--   Baseline: SSE SSE2 SSE3
--   requested: SSE3
--   Dispatched code generation: SSE4_1 SSE4_2 FP16 AVX AVX2 AVXS12_SKX
--   requested: SSE4_1 SSE4_2 AVX FP16 AVX2 AVXS12_SKX
-- SSE4_1 (16 files): + SSSE3 SSE4_1
-- SSE4_2 (2 files): + SSSE3 SSE4_1 POPCNT SSE4_2
-- FP16 (1 files): + SSSE3 SSE4_1 POPCNT SSE4_2 FP16 AVX
-- AVX (5 files): + SSSE3 SSE4_1 POPCNT SSE4_2 AVX
-- AVX2 (30 files): + SSSE3 SSE4_1 POPCNT SSE4_2 FP16 FMA3 AVX AVX2
-- AVXS12_SKX (6 files): + SSSE3 SSE4_1 POPCNT SSE4_2 FP16 FMA3 AVX AVX2 AVX_512F AVXS12_COMMON AVXS12_SKX
--
-- C/C++:
--   Built as dynamic libst: YES
--   C++ standard: 11
--   C++ Compiler: /usr/bin/c++ (ver 7.5.0)
-- C++ flags (Release): -fsigned-char -W -Wall -Werror-return-type -Werror-non-virtual-dtor -Werror-address -Werror-sequence-point -Wformat -Werror-format-security -Wmissing-declarations -Wund
e
ef -Winit-self -Wpointer-arith -Wshadow -Wuninitialized -Winit-self -Wsuggest-override -Wno-delete-non-virtual-dtor -Wno-comment -Wimplicit-fallthrough=3 -Wno-strict-overflow -fdiagnostics-sh
ow-option -Wno-long-long -pthread -font-frame-pointer -ffunction-sections -fdata-sections -msse -msse2 -msse3 -fvisibility-hidden -fvisibility-inlines-hidden -O3 -DDEBUG -DDEBUG
-- C++ flags (Debug): -fsigned-char -W -Wall -Werror-return-type -Werror-non-virtual-dtor -Werror-address -Werror-sequence-point -Wformat -Werror-format-security -Wmissing-declarations -Wund
e
ef -Winit-self -Wpointer-arith -Wshadow -Wuninitialized -Winit-self -Wsuggest-override -Wno-delete-non-virtual-dtor -Wno-comment -Wimplicit-fallthrough=3 -Wno-strict-overflow -fdiagnostics-sh
ow-option -Wno-long-long -pthread -font-frame-pointer -ffunction-sections -fdata-sections -msse -msse2 -msse3 -fvisibility-hidden -g -O0 -DDEBUG -DDEBUG
-- C Compiler: /usr/bin/cc
-- C flags (Release): -fsigned-char -W -Wall -Werror-return-type -Werror-address -Werror-sequence-point -Wformat -Werror-format-security -Wmissing-declarations -Wstrict-
prototypes -Wundef -Winit-self -Wpointer-arith -Wshadow -Wuninitialized -Winit-self -Wno-comment -Wimplicit-fallthrough=3 -Wno-strict-overflow -fdiagnostics-show-option -Wno-long-long -pthread -font-
e
-poller -ffunction-sections -fdata-sections -msse -msse2 -msse3 -fvisibility-hidden -O3 -DDEBUG -DDEBUG
-- C flags (Debug): -fsigned-char -W -Wall -Werror-return-type -Werror-address -Werror-sequence-point -Wformat -Werror-format-security -Wmissing-declarations -Wstrict-
prototypes -Wundef -Winit-self -Wpointer-arith -Wshadow -Wuninitialized -Winit-self -Wno-comment -Wimplicit-fallthrough=3 -Wno-strict-overflow -fdiagnostics-show-option -Wno-long-long -pthread -font-
e
-poller -ffunction-sections -fdata-sections -msse -msse2 -msse3 -fvisibility-hidden -g -O0 -DDEBUG -DDEBUG
-- Linker flags (Release): -Wl,--exclude-libs,libippcv.a -Wl,--exclude-libs,libippw.a -Wl,--gc-sections -Wl,--as-needed
-- Linker flags (Debug): -Wl,--exclude-libs,libippcv.a -Wl,--exclude-libs,libippw.a -Wl,--gc-sections -Wl,--as-needed
-- Scache: NO
-- Precompiled headers: NO
-- Extra dependencies: pthread cudart_static -lphread dl rt nppc nppial nppic nppicon nppid1 nppif nppit nppin nppist nppisu nppitc npps cublas cudnn cufft -L/usr/local/cuda-10.2/lib64
-- 3rdparty dependencies:
--
-- OpenCV modules:
-- To be built: alphamat aruco brseq b1oinstired call3d ccallb core cudaarithm cudabgsegm cudacodec cudafeatures2d cudafilters cudainnproc cudalegacy cudaobjdetect cudaoptflow cudast
reeo cudawarping cudve datasets dnn dnn_objdetect dnn_superres dnn_face_feature_extractor dnn_fcn_fast freetype fuzzy_gapi hdf hdf4 highgui img_hash imgcodecs imgproc ltenarity transform t11m_descriptor ml_objdetect opt

```

ภาพที่ 114 ผลการ compile ชุดคำสั่ง OpenCV ส่วนที่ 1

```

Activities Terminal
microcsidcn@microcsidcn-desktop: ~/opencv/build

-- Disabled by dependency: world
-- Unavailable: cnn_3dobj cvv java js matlab ovis python2 vtz
-- Applications: tests perf_tests examples apps
-- Documentation: NO
-- Non-free algorithms: YES

-- GUI:
-- GTK+: YES (ver 3.22.30)
-- GThread: YES (ver 2.56.4)
-- GEXE: NO
-- VTK support: NO

-- Media I/O:
-- ZLIB: /usr/lib/x86_64-linux-gnu/libz.so (ver 1.2.11)
-- JPEG: /usr/lib/x86_64-linux-gnu/libjpeg.so (ver 80)
-- HEIF: build (ver encoder: 0x020f)
-- PNG: /usr/lib/x86_64-linux-gnu/libpng.so (ver 1.6.37)
-- TIFF: /usr/lib/x86_64-linux-gnu/libtiff.so (ver 42 / 4.0.9)
-- JPEG 2000: build Jasper (ver 1.900.1)
-- OpenEXR: build (ver 2.3.0)
-- YES
-- SUNRASTER: YES
-- PNM: YES
-- PPM: YES

-- Video I/O:
-- DC1394: YES (2.2.5)
-- FFmpeg: YES
-- avcodec: YES (57.107.100)
-- avformat: YES (57.83.100)
-- avutil: YES (55.78.100)
-- swscale: YES (4.8.100)
-- avresample: YES (3.7.0)
-- GStreamer: NO
-- v4l/v4l2: YES (Linux/videodev2.h)

-- Parallel framework: pthreads
-- Trace: YES (with Intel ITT)

-- Other third-party libraries:
-- Intel IPP: 2020.0.8 Gold [2020.0.0]
-- Intel IPP IW: /home/microcsidcn/opencv/build/3rdparty/ippicv/ippicv_lnx/icv
-- sources (2020.0.8)
-- Lapack: at: /home/microcsidcn/opencv/build/3rdparty/ippicv/ippicv_lnx/lw
-- Eigen: YES (ver 3.3.4)
-- Custom HAL: NO
-- Protobuf: build (3.5.1)
-- NVIDIA CUDA: YES (ver 10.2, CUFFT CUBLAS)

```

ภาพที่ 115 ผลการ compile ชุดคำสั่ง OpenCV ส่วนที่ 2

```

Activities Terminal
microcsidcn@microcsidcn-desktop: ~/opencv/build

-- DC1394: YES (2.2.5)
-- FFmpeg: YES
-- avcodec: YES (57.107.100)
-- avformat: YES (57.83.100)
-- avutil: YES (55.78.100)
-- swscale: YES (4.8.100)
-- avresample: YES (3.7.0)
-- GStreamer: NO
-- v4l/v4l2: YES (Linux/videodev2.h)

-- Parallel framework: pthreads
-- Trace: YES (with Intel ITT)

-- Other third-party libraries:
-- Intel IPP: 2020.0.8 Gold [2020.0.0]
-- Intel IPP IW: /home/microcsidcn/opencv/build/3rdparty/ippicv/ippicv_lnx/icv
-- sources (2020.0.8)
-- Lapack: at: /home/microcsidcn/opencv/build/3rdparty/ippicv/ippicv_lnx/lw
-- Eigen: YES (ver 3.3.4)
-- Custom HAL: NO
-- Protobuf: build (3.5.1)
-- NVIDIA CUDA: YES (ver 10.2, CUFFT CUBLAS)
-- NVIDIA GPU arch: 30 35 37 50 52 56 60 61 70 75
-- NVIDIA PTX arches:
-- cuDNN: YES (ver 7.6.5)
-- OpenCL: YES (no extra features)
-- Include path: /home/microcsidcn/opencv/3rdparty/include/opencl/1.2
-- Link libraries: Dynamic load

-- Python 3:
-- Interpreter: /home/microcsidcn/virtualenvs/cv/bin/python3 (ver 3.6.9)
-- Libraries: /usr/lib/x86_64-linux-gnu/libpython3.6m.so (ver 3.6.9)
-- numpy: /home/microcsidcn/virtualenvs/cv/lib/python3.6/site-packages/numpy/core/include (ver 1.18.3)
-- Install path: lib/python3.6/site-packages/cv2/python3.6
-- Python (for build): /home/microcsidcn/virtualenvs/cv/bin/python3

-- Java:
-- ant: NO
-- JNI: NO
-- Java wrappers: NO
-- Java tests: NO

-- Install to: /usr/local

-- Configuring done
-- Generating done
-- Build files have been written to: /home/microcsidcn/opencv/build
*** (cv) microcsidcn@microcsidcn-desktop:~/opencv/build

```

ภาพที่ 116 ผลการ compile ชุดคำสั่ง OpenCV ส่วนที่ 3

เมื่อ compile เสร็จเรียบร้อยแล้ว ถัดไปจะทำการติดตั้ง OpenCV โดยใช้คำสั่ง make ตามด้วย -j จำนวนแกนของ CPU จากนั้นใช้คำสั่ง ldconfig เชื่อมโยง cache ที่จำเป็นของ OpenCV

```
(cv) :~$ make -j16
```

```
(cv) :~$ sudo make install
```

```
(cv) :~$ sudo ldconfig
```

ขั้นตอนการตรวจสอบว่า OpenCV ถูกติดตั้งโดยสมบูรณ์และสามารถใช้งานได้บนสภาพแวดล้อมจำลองของ Python สามารถเรียกโปรแกรม Python เพื่อเขียนคำสั่งตรวจสอบรุ่นของ OpenCV ซึ่งในรุ่นปัจจุบัน คือ 4.2.0

```
(cv) :~$ workon cv
```

```
(cv) :~$ python
```

```
>>> import cv2
```

```
>>> cv2.__version__
```

```
'4.2.0'
```



```
-- Installing: /usr/local/share/opencv4/samples/python/lk_homography.py
-- Installing: /usr/local/share/opencv4/samples/python/lk_track.py
-- Installing: /usr/local/share/opencv4/samples/python/logpolar.py
-- Installing: /usr/local/share/opencv4/samples/python/morphology.py
-- Installing: /usr/local/share/opencv4/samples/python/mosse.py
-- Installing: /usr/local/share/opencv4/samples/python/mouse_and_match.py
-- Installing: /usr/local/share/opencv4/samples/python/mser.py
-- Installing: /usr/local/share/opencv4/samples/python/opencv_version.py
-- Installing: /usr/local/share/opencv4/samples/python/opt_flow.py
-- Installing: /usr/local/share/opencv4/samples/python/peopledetect.py
-- Installing: /usr/local/share/opencv4/samples/python/plane_ar.py
-- Installing: /usr/local/share/opencv4/samples/python/plane_tracker.py
-- Installing: /usr/local/share/opencv4/samples/python/qrcode.py
-- Installing: /usr/local/share/opencv4/samples/python/squares.py
-- Installing: /usr/local/share/opencv4/samples/python/stereo_match.py
-- Installing: /usr/local/share/opencv4/samples/python/stitching.py
-- Installing: /usr/local/share/opencv4/samples/python/stitching_detailed.py
-- Installing: /usr/local/share/opencv4/samples/python/text_skewness_correction.py
-- Installing: /usr/local/share/opencv4/samples/python/texture_flow.py
-- Installing: /usr/local/share/opencv4/samples/python/tst_scene_render.py
-- Installing: /usr/local/share/opencv4/samples/python/turing.py
-- Installing: /usr/local/share/opencv4/samples/python/video.py
-- Installing: /usr/local/share/opencv4/samples/python/video_threaded.py
-- Installing: /usr/local/share/opencv4/samples/python/video_v4l2.py
-- Installing: /usr/local/share/opencv4/samples/python/watershed.py
(cv) microsisdcn@microsisdcn-desktop:~/opencv/build$ sudo ldconfig
(cv) microsisdcn@microsisdcn-desktop:~/opencv/build$ cd ~/.virtualenvs/cv/lib/python3.6/site-packages
/
(cv) microsisdcn@microsisdcn-desktop:~/.virtualenvs/cv/lib/python3.6/site-packages$ ln -s /usr/local/
bin/      cuda-10.2/  games/    lib/      sbin/     src/
cuda/    etc/             include/  man/     share/
(cv) microsisdcn@microsisdcn-desktop:~/.virtualenvs/cv/lib/python3.6/site-packages$ ln -s /usr/local/
bin/      cuda-10.2/  games/    lib/      sbin/     src/
cuda/    etc/             include/  man/     share/
(cv) microsisdcn@microsisdcn-desktop:~/.virtualenvs/cv/lib/python3.6/site-packages$ python
Python 3.6.9 (default, Apr 18 2020, 01:56:04)
[GCC 8.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> cv2.__version__
'4.2.0'
>>> □
```

ภาพที่ 117 ผลการติดตั้ง OpenCV รุ่น 4.2.0 โดยสมบูรณ์

การติดตั้ง TensorFlow นั้นสามารถเรียกใช้คำสั่งของ PyPI เพื่อเรียกคลังชุดคำสั่งมาติดตั้งลงในสภาพแวดล้อมจำลองของ Python ได้ทันที โดยการจัดการ package ผ่านคำสั่ง pip install ซึ่งรุ่นของ TensorFlow รุ่น 1.X จะแบ่งเป็น TensorFlow สำหรับ CPU ประมวลผล และ TensorFlow-GPU สำหรับ GPU ประมวลผล หากติดตั้ง TensorFlow 2.X จะรองรับทั้ง CPU และ GPU ในการติดตั้งครั้งเดียว

(cv) :~\$ pip install -U tensorflow-gpu==1.15

```

microslidcn@microslidcn-desktop:~$ pip install -U tensorflow-gpu==1.15
Successfully built gast termcolor absl-py wrapt
Installing collected packages: numpy, tensorflow-estimator, grpcio, setuptools, markdown, werkzeug, absl-py, protobuf, tensorboard, gast, h5py, keras-applications, google-pasta, opt-einsum, termcolor, keras-preprocessing, astor, keras, tensorflow-gpu
WARNING: The scripts f2py, f2py3 and f2py3_6 are installed in '/home/microslidcn/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
WARNING: The scripts easy_install and easy_install-3.6 are installed in '/home/microslidcn/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
WARNING: The script tensorboard is installed in '/home/microslidcn/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
WARNING: The scripts estimator_dkt_converter, freeze_graph, saved_model_cli, tensorboard, tf_upgrade_v2, trtite_convert, toco and toco_from_protos are installed in '/home/microslidcn/.local/bin' which is not on PATH.
Consider adding this directory to PATH or, if you prefer to suppress this warning, use --no-warn-script-location.
Successfully installed absl-py-9.0 astor-0.8.1 gast-0.2.2 google-pasta-0.2.0 grpcio-1.28.1 h5py-2.10.0 keras-applications-1.0.8 keras-preprocessing-1.1.0 markdown-3.2.1 numpy-1.18.3 opt-einsum-3.2.1 pro
tobuf-3.11.3 setuptools-46.1.3 tensorboard-1.15.0 tensorflow-estimator-1.15.1 tensorflow-gpu-1.15.0 termcolor-1.1.0 werkzeug-1.0.1 wrapt-1.12.1
microslidcn@microslidcn-desktop:~$ pip install -U tensorflow-gpu==1.15
Defaulting to user installation because normal site-packages is not writable
Requirement already up-to-date: tensorflow-gpu==1.15 in ./local/lib/python3.6/site-packages (1.15.0)
Requirement already satisfied, skipping upgrade: tensorflow-estimator==1.15.1 in ./local/lib/python3.6/site-packages (from tensorflow-gpu==1.15) (1.15.1)
Requirement already satisfied, skipping upgrade: gast==0.2.2 in ./local/lib/python3.6/site-packages (from tensorflow-gpu==1.15) (0.2.2)
Requirement already satisfied, skipping upgrade: six==1.10.0 in /usr/lib/python3/dist-packages (from tensorflow-gpu==1.15) (1.10.0)
Requirement already satisfied, skipping upgrade: keras-applications==1.0.8 in ./local/lib/python3.6/site-packages (from tensorflow-gpu==1.15) (1.0.8)
Requirement already satisfied, skipping upgrade: wrapt==1.11.1 in ./local/lib/python3.6/site-packages (from tensorflow-gpu==1.15) (1.11.1)
Requirement already satisfied, skipping upgrade: absl-py==0.7.0 in ./local/lib/python3.6/site-packages (from tensorflow-gpu==1.15) (0.9.4)
Requirement already satisfied, skipping upgrade: numpy<2.0,>=1.16.0 in ./local/lib/python3.6/site-packages (from tensorflow-gpu==1.15) (1.18.3)
Requirement already satisfied, skipping upgrade: wheel==0.26 in /usr/lib/python3/dist-packages (from tensorflow-gpu==1.15) (0.30.0)
Requirement already satisfied, skipping upgrade: keras-preprocessing==1.0.8 in ./local/lib/python3.6/site-packages (from tensorflow-gpu==1.15) (1.1.0)
Requirement already satisfied, skipping upgrade: tensorboard<1.16.0,>=1.15.0 in ./local/lib/python3.6/site-packages (from tensorflow-gpu==1.15) (1.15.0)
Requirement already satisfied, skipping upgrade: astor==0.6.0 in ./local/lib/python3.6/site-packages (from tensorflow-gpu==1.15) (0.8.1)
Requirement already satisfied, skipping upgrade: termcolor==1.1.0 in ./local/lib/python3.6/site-packages (from tensorflow-gpu==1.15) (1.1.0)
Requirement already satisfied, skipping upgrade: grpcio==1.8.6 in ./local/lib/python3.6/site-packages (from tensorflow-gpu==1.15) (1.28.1)
Requirement already satisfied, skipping upgrade: google-pasta==0.1.0 in ./local/lib/python3.6/site-packages (from tensorflow-gpu==1.15) (0.2.0)
Requirement already satisfied, skipping upgrade: opt-einsum==2.3.2 in ./local/lib/python3.6/site-packages (from tensorflow-gpu==1.15) (3.2.1)
Requirement already satisfied, skipping upgrade: protobuf==3.6.1 in ./local/lib/python3.6/site-packages (from tensorflow-gpu==1.15) (3.11.3)
Requirement already satisfied, skipping upgrade: h5py in ./local/lib/python3.6/site-packages (from keras-applications==1.0.8->tensorflow-gpu==1.15) (2.10.0)
Requirement already satisfied, skipping upgrade: tensorflow==1.15.0 in ./local/lib/python3.6/site-packages (from tensorflow-gpu==1.15) (1.15.0)
Requirement already satisfied, skipping upgrade: tensorflow-gpu==1.15 in ./local/lib/python3.6/site-packages (from tensorflow-gpu==1.15) (1.15.0)
Requirement already satisfied, skipping upgrade: tensorflow==1.15.0 in ./local/lib/python3.6/site-packages (from tensorflow-gpu==1.15) (1.15.0)
microslidcn@microslidcn-desktop:~$ pip install keras
Defaulting to user installation because normal site-packages is not writable
Collecting keras
  Downloading keras-2.3.1-py2.py3-none-any.whl (377 kB)
Requirement already satisfied: keras-applications==1.0.8 in ./local/lib/python3.6/site-packages (from keras) (1.0.8)
Requirement already satisfied: h5py in ./local/lib/python3.6/site-packages (from keras) (2.10.0)
Requirement already satisfied: six==1.9.0 in /usr/lib/python3/dist-packages (from keras) (1.11.0)
Requirement already satisfied: tensorflow==1.15.0 in ./local/lib/python3.6/site-packages (from keras) (1.15.0)
Requirement already satisfied: tensorflow-gpu==1.15 in ./local/lib/python3.6/site-packages (from keras) (1.15.0)
Collecting scipy==0.14
Using cached scipy-1.4.1-cp36-cp36m-manylinux1_x86_64.whl (26.1 MB)
Requirement already satisfied: numpy==1.9.1 in ./local/lib/python3.6/site-packages (from keras) (1.18.3)
Installing collected packages: scipy, keras
Successfully installed keras-2.3.1 scipy-1.4.1
microslidcn@microslidcn-desktop:~$

```

ภาพที่ 118 หน้าต่าง Terminal แสดงผลการติดตั้ง TensorFlow รุ่น 1.15

จากกระบวนการติดตั้งด้วยคำสั่ง pip install -U tensorflow-gpu==1.15 นั้น จะดึงคลังชุดคำสั่งและส่วนเสริมของระบบ เช่น gast, six, wrapt, protobuf และอื่น ๆ เมื่อติดตั้ง TensorFlow เรียบร้อยแล้ว สามารถทดสอบการติดตั้งโดยเข้าไปที่โปรแกรม Python และใช้คำสั่งดังต่อไปนี้

(cv) :~\$ python

```
>>> import tensorflow as tf
```

```
>>> hello = tf.constant('Hello, TensorFlow!')
```

```
>>> sess = tf.Session()
```

```
>>> print(sess.run(hello))
```

```

Activities Terminal
microsidcn@microsidcn-desktop: /
File Edit View Search Terminal Help
(cv) microsidcn@microsidcn-desktop:~/python$ python
Python 3.6.9 (default, Apr 18 2020, 01:55:04)
[GCC 8.4.0] on Linux
Type "help", "copyright", "credits" or "license()" for more information.
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
2020-04-29 11:29:57.535349: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcuda.so.1
2020-04-29 11:29:57.563185: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:983] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2020-04-29 11:29:57.563651: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1618] Found device 0 with properties:
name: GeForce GTX 1080 TI major: 6 minor: 1 memoryClockRate(GHz): 1.721
pciBusID: 0000:0c:00:0
2020-04-29 11:29:57.564399: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:983] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2020-04-29 11:29:57.564399: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1618] Found device 1 with properties:
name: GeForce GTX 1080 TI major: 6 minor: 1 memoryClockRate(GHz): 1.721
pciBusID: 0000:0c:00:0
2020-04-29 11:29:57.564543: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcudart.so.10.0
2020-04-29 11:29:57.565067: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcublas.so.10.0
2020-04-29 11:29:57.565892: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcubrand.so.10.0
2020-04-29 11:29:57.566562: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcusolver.so.10.0
2020-04-29 11:29:57.567097: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcusparse.so.10.0
2020-04-29 11:29:57.568990: I tensorflow/stream_executor/platform/default/dso_loader.cc:44] Successfully opened dynamic library libcudnn.so.7
2020-04-29 11:29:57.569115: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:983] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2020-04-29 11:29:57.569591: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:983] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2020-04-29 11:29:57.570281: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:983] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2020-04-29 11:29:57.570752: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:983] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2020-04-29 11:29:57.571391: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1746] Adding visible gpu devices: 0, 1
2020-04-29 11:29:57.571624: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to use: AVX2 FMA
2020-04-29 11:29:57.592755: I tensorflow/core/platform/profile_utils/cpu_utils.cc:94] CPU Frequency: 3892925000 Hz
2020-04-29 11:29:57.592755: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x52715b0 initialized for platform Host (this does not guarantee that XLA will be used). Devices:
2020-04-29 11:29:57.592755: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
2020-04-29 11:29:57.708450: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:983] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2020-04-29 11:29:57.721951: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:983] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2020-04-29 11:29:57.722745: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x5314070 initialized for platform CUDA (this does not guarantee that XLA will be used). Devices:
2020-04-29 11:29:57.722770: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (0): Host, Default Version
2020-04-29 11:29:57.722770: I tensorflow/compiler/xla/service/service.cc:176] StreamExecutor device (1): GeForce GTX 1080 TI, Compute Capability 6.1
2020-04-29 11:29:57.724619: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:983] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2020-04-29 11:29:57.726381: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1618] Found device 0 with properties:
name: GeForce GTX 1080 TI major: 6 minor: 1 memoryClockRate(GHz): 1.721
pciBusID: 0000:0c:00:0
2020-04-29 11:29:57.726449: I tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:983] successful NUMA node read from SysFS had negative value (-1), but there must be at least one NUMA node, so returning NUMA node zero
2020-04-29 11:29:57.727145: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1618] Found device 1 with properties:
name: GeForce GTX 1080 TI major: 6 minor: 1 memoryClockRate(GHz): 1.721
pciBusID: 0000:0c:00:0

```

ภาพที่ 119 ผลการแสดง session เมื่อติดตั้ง TensorFlow อย่างสมบูรณ์

ขั้นตอนสุดท้าย คือการติดตั้งคลังชุดคำสั่ง Keras เพื่อให้ TensorFlow สามารถดึงการทำงานของคุณคลัง และส่วนเสริมต่าง ๆ มาใช้งานได้ เมื่อติดตั้งเสร็จสิ้นแล้วให้สร้างไฟล์ภาษา Python และนำคำสั่งในตัวอย่างไปส่งงาน เพื่อทดสอบว่าการติดตั้ง Keras เสร็จสมบูรณ์

```
(cv) ~$ pip install keras progressbar keras-resnet IPython[all]
```

```
(cv) ~$ sudo nano mnist_mip.py
```

```

from __future__ import print_function
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout
from keras.optimizers import RMSprop
batch_size = 128
num_classes = 10
epochs = 20
(x_train, y_train), (x_test, y_test) = mnist.load_data()

```

```
x_train = x_train.reshape(60000, 784)
x_test = x_test.reshape(10000, 784)
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255
print(x_train.shape[0], 'train samples')
print(x_test.shape[0], 'test samples')
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)
model = Sequential()
model.add(Dense(512, activation='relu', input_shape=(784,)))
model.add(Dropout(0.2))
model.add(Dense(512, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(10, activation='softmax'))
model.summary()
model.compile(loss='categorical_crossentropy',
              optimizer=RMSprop(),
              metrics=['accuracy'])
history = model.fit(x_train, y_train,
                   batch_size=batch_size,
                   epochs=epochs,
                   verbose=1,
                   validation_data=(x_test, y_test))
score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

```
(cv) :~$ python mnist_mip.py
```



## ภาคผนวก ง. ชุดคำสั่งที่ใช้ในวิทยานิพนธ์

```
# ชุดคำสั่งที่ใช้ในการแปลงข้อมูล BBOX Dataframe (ส่วนหนึ่งในชุดคำสั่ง)
import ...
IMAGE_PATH = './images'
ANNOTATIONS = 'annotations.csv'
CLASSES = 'classes.csv'
annotations = []
classes = set([])
for xml_file in [f for f in os.listdir(IMAGE_PATH) if f.endswith(".xml")]:
    tree = ET.parse(os.path.join(IMAGE_PATH, xml_file))
    root = tree.getroot()
    file_name = None
    for elem in root:
        if elem.tag == 'filename':
            file_name = os.path.join(IMAGE_PATH, elem.text)
        if elem.tag == 'object':
            obj_name = None
            coords = []
            for subelem in elem:
                if subelem.tag == 'name':
                    obj_name = subelem.text
                if subelem.tag == 'bndbox':
                    for subsubelem in subelem:
                        coords.append(subsubelem.text)
            item = [file_name] + coords + [obj_name]
            annotations.append(item)
            classes.add(obj_name)
# ต่อหน้าถัดไป
```

```

with open(ANNOTATIONS, 'w') as f:
    writer = csv.writer(f)
    writer.writerows(annotations)
with open(CLASSES, 'w') as f:
    for i, line in enumerate(classes):
        f.write('{}\n'.format(line,i))
# แสดง CSV ที่ได้จากการแปลงคลาสจาก XML
classes = pd.read_csv("classes.csv")
classes.head()
# แสดง CSV ที่ได้จากการแปลงข้อมูลทั้งหมดจาก XML
annotations = pd.read_csv("annotations.csv")
annotations.head()

```

```

# ชุดคำสั่งที่ทำงานบนชุดกล้อง MicrosisDCN (ส่วนหนึ่งในชุดคำสั่ง)
import ...
def get_session():
    config = tf.compat.v1.ConfigProto()
    config.gpu_options.allow_growth = False
    return tf.Session(config=config)
keras.backend.tensorflow_backend.set_session(get_session())
model = models.load_model("./blood_model_inference.h5",
    backbone_name='resnet50')
print(model.summary())
labels_to_names = {0: 'WBC', 1: 'Platelets', 2: 'RBC'}
with picamera.PiCamera() as camera:
    cap=picamera.array.PiRGBArray(camera)
    camera.resolution = (1280, 960)
    camera.capture(cap,format="rgb")

# ต่อหน้าถัดไป

```

```
camera.start_preview()
time.sleep(3)
image = cap.array
draw = image.copy()
draw = cv2.cvtColor(draw, cv2.COLOR_BGR2RGB)
image = preprocess_image(image)
image, scale = resize_image(image)
start = time.time()
boxes, scores, labels = model.predict_on_batch(np.expand_dims(image, axis=0))
p_time = time.time() - start
print("processing time: ", p_time)
boxes /= scale
current_object = 0
plate_count = 0
WBC_count = 0
RBC_count = 0
score = 0
HPF_plate = 0
HPF_WBC = 0
HPF_RBC = 0
for box, score, label in zip(boxes[0], scores[0], labels[0]):
    if score < 0.5:
        break
    current_object += 1
    color = label_color(label)
    bbox = box.astype(int)
    draw_box(draw, bbox, color=color)
    caption = "{} {:.3f}".format(labels_to_names[label], score*100)
    draw_caption(draw, b, caption)
    if label == 0:
# ต่อนหน้าถัดไป
```

```
    plate_count += 1
    if label == 1:
        WBC_count += 1
    if label == 2:
        RBC_count += 1
HPF_plate = plate_count
HPF_WBC = WBC_count
HPF_RBC = RBC_count
font = cv2.FONT_HERSHEY_SIMPLEX
draw = cv2.cvtColor(draw, cv2.COLOR_RGB2BGR)
cv2.rectangle(draw, (0,0), (1280, 30), (0, 0, 0), -1)
cv2.putText(draw, "MicrosisDCN 0.9e | Total count: " + str(current_object) +
            " | RBC: " + str(HPF_RBC) + " | WBC: " + str(HPF_WBC) + " | Platelets: " +
            str(HPF_plate), (5, 19), font, 0.5, (255, 255, 255), 1)
print(" ")
print("***** Count Result *****")
print("object count: ", current_object)
print("Platelets count: ", plate_count)
print("WBC count: ", WBC_count)
print("RBC count: ", RBC_count)
print(" ")
print("***** HPF Result *****")
print("Platelets count per HPF: ", HPF_plate )
print("WBC count per HPF: ", HPF_WBC)
print("RBC count per HPF: ", HPF_RBC)
print(" ")
cv2.imshow('MicrosisDCN Ai Microbe Detector 0.9e EECU', draw)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

```

# สร้างโมเดลโครงข่ายประสาท ResNet50 เป็น backbone (ส่วนหนึ่งในชุดคำสั่ง)
def ResNet_50(input_shape, classes):
    X_input = Input(input_shape)
    X = ZeroPadding2D((3, 3))(X_input)
    # Stage 1
    X = Conv2D(64, (7, 7), strides=(2, 2), name='conv1',
              kernel_initializer=glorot_uniform(seed=0))(X)
    X = BatchNormalization(axis=3, name='bn_conv1')(X)
    X = Activation('relu')(X)
    X = MaxPooling2D((3, 3), strides=(2, 2))(X)
    # Stage 2
    X = conv_block(X, f=3, filters=[64, 64, 256], stage=2, block='a', s=1)
    X = identity_block(X, 3, [64, 64, 256], stage=2, block='b')
    X = identity_block(X, 3, [64, 64, 256], stage=2, block='c')
    # Stage 3
    X = conv_block(X, f = 3, filters = [128, 128, 512], stage = 3,
                  block='a', s = 2)
    X = identity_block(X, 3, [128, 128, 512], stage=3, block='b')
    X = identity_block(X, 3, [128, 128, 512], stage=3, block='c')
    X = identity_block(X, 3, [128, 128, 512], stage=3, block='d')
    # Stage 4
    X = conv_block(X, f = 3, filters = [256, 256, 1024], stage = 4,
                  block='a', s = 2)
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='b')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='c')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='d')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='e')
    X = identity_block(X, 3, [256, 256, 1024], stage=4, block='f')
# ต่อหน้าถัดไป

```

```

# Stage 5
X = convolutional_block(X, f = 3, filters = [512, 512, 2048], stage = 5,
                        block='a', s = 2)
X = identity_block(X, 3, [512, 512, 2048], stage=5, block='b')
X = identity_block(X, 3, [512, 512, 2048], stage=5, block='c')
X = AveragePooling2D((7,7), name="avg_pool")(X)
X = Flatten()(X)
X = Dense(classes, activation='softmax', name='fc' + str(classes),
          kernel_initializer = glotot_uniform(seed=0))(X)
model = Model(inputs = X_input, outputs = X, name='ResNet50')
return model
# สร้าง Backbone ตามจำนวนคลาส 3 คลาส ให้ขนาดภาพกว้าง 224 สูง 224 จุดภาพ
resnet_backbone = ResNet_50(input_shape = (224, 224, 3), classes = 3)
model = ResNet50(input_shape = (224, 224, 3), Resnet_backbone, 3)
model.compile(optimizer='adam', loss='categorical_crossentropy',
              metrics=['accuracy'])
model.save('blood_model.h5', backbone_name='resnet50' )
# *** จุดคำสั่งภายใน retinanet.py ยกมาเฉพาะส่วนสำคัญ
# สร้าง Feature Pyramid Network (FPN) ไปเชื่อมกับ ResNet50
def __create_pyramid_features(C3, C4, C5, feature_size=256):
    P5 = keras.layers.Conv2D(feature_size, kernel_size=1, strides=1,
                              padding='same', name='C5_reduced')(C5)
    P5_upsampled = layers.UpSampleLike(name='P5_upsampled')(P5, C4)
    P5 = keras.layers.Conv2D(feature_size, kernel_size=3, strides=1,
                              padding='same', name='P5')(P5)
    # add P5 elementwise to C4
    P4 = keras.layers.Conv2D(feature_size, kernel_size=1, strides=1,
                              padding='same', name='C4_reduced')(C4)
# ต่อหน้าถัดไป

```

```

P4 = keras.layers.Add(name='P4_merged')([P5_upsampled, P4])
P4_upsampled = layers.UpsampleLike(name='P4_upsampled')([P4, C3])
P4 = keras.layers.Conv2D(feature_size, kernel_size=3, strides=1,
                          padding='same', name='P4')(P4)
# add P4 elementwise to C3
P3 = keras.layers.Conv2D(feature_size, kernel_size=1, strides=1,
                          padding='same', name='C3_reduced')(C3)
P3 = keras.layers.Add(name='P3_merged')([P4_upsampled, P3])
P3 = keras.layers.Conv2D(feature_size, kernel_size=3, strides=1,
                          padding='same', name='P3')(P3)
# "P6 is obtained via a 3x3 stride-2 conv on C5"
P6 = keras.layers.Conv2D(feature_size, kernel_size=3, strides=2,
                          padding='same', name='P6')(C5)
# "P7 is computed by applying ReLU followed by a 3x3 stride-2 conv on P6"
P7 = keras.layers.Activation('relu', name='C6_relu')(P6)
P7 = keras.layers.Conv2D(feature_size, kernel_size=3, strides=2,
                          padding='same', name='P7')(P7)
return [P3, P4, P5, P6, P7]
# สร้างโมเดลโครงข่ายประสาทด้วยคลัสซุคคำสั่ง Keras RetinaNet (ส่วนหนึ่งในชุดคำสั่ง)
def retinanet(inputs, backbone_layers, num_classes, num_anchors = None,
              create_pyramid_features = __create_pyramid_features,
              submodels = None, name = 'retinanet'):
    num_anchors = AnchorParameters.default.num_anchors()
    C3, C4, C5 = backbone_layers
    features = create_pyramid_features(C3, C4, C5)
    pyramids = __build_pyramid(submodels, features)
    return keras.models.Model(inputs=inputs, outputs=pyramids, name=name)
# ....

```

```

# ประกาศคำสั่งชุดคำสั่งที่เกี่ยวข้องกับการขยายภาพและบันทึกไฟล์บรรณนิทัศน์ XML
import imgaug as ia
ia.seed(1)
from imgaug.augmentables.bbs import BoundingBox, BoundingBoxesOnImage
from imgaug import augmenters as iaa
import imageio, re, os, glob, shutil, csv
import pandas as pd
import numpy as np
import xml.etree.ElementTree as ET

# ชุดคำสั่งตรวจสอบภาพใน datasets และเก็บเข้าตัวแปร images ตามจำนวนภาพ
images = []
for index, file in enumerate(glob.glob('images/*.jpg')):
    images.append(imageio.imread(file))
    print('Filename : {} Dimension is {}'.format(file[7:], images[index].shape))
print('Original datasets have {} images'.format(len(images)))

# ประกาศคำสั่งชุดคำสั่งที่เกี่ยวข้องกับการแสดงภาพและไฟล์บรรณนิทัศน์ XML
ia.imshow(images[2])
ia.imshow(images[7])
for index, file in enumerate(glob.glob('images/outputs/*.xml')):
    print(file[7:])

```



```
# ชุดคำสั่งแปลงข้อมูลในไฟล์บรรณนิทัศน์ XML ไปเป็น CSV และ Dataframe
def xml_to_csv(path):
    data_list = []
    for xml_file in glob.glob(path + '/*.xml'):
        tree = ET.parse(xml_file)
        root = tree.getroot()
        for element in root.findall('object'):
            value = (root.find('filename').text, int(root.find('size')[0].text),
                    int(root.find('size')[1].text), element[0].text, int(element[4][0].text),
                    int(element[4][1].text), int(element[4][2].text),
                    int(element[4][3].text))
            data_list.append(value)
    label_info = ['filename', 'width', 'height', 'class', 'xmin', 'ymin', 'xmax', 'ymax']
    xml_dataframe = pd.DataFrame(data_list, columns= label_info)
    return xml_dataframe

dataframe = xml_to_csv('images/outputs/')
dataframe.to_csv(('csv_labels.csv'), index=None)
print('Successfully converted XML to CSV dataframe.')
dataframe
```

```

# ชุดคำสั่งสำหรับขยายชุดภาพจาก Dataframe ที่แปลงมาจากไฟล์บรรณนิทัศน์ XML
def image_aug(df, images_path, aug_images_path, image_prefix, aug_params):
    aug_bbox = pd.DataFrame(columns=['filename','width','height','class','xmin',
        'ymin', 'xmax', 'ymax'])
    grouped = df.groupby('filename')
    for filename in df['filename'].unique():
        group_df = grouped.get_group(filename)
        group_df = group_df.reset_index()
        group_df = group_df.drop(['index'], axis=1)
        image = imageio.imread(images_path+filename)
        bb_array = group_df.drop(['filename', 'width', 'height', 'class'], axis=1).values
        bbox = BoundingBoxesOnImage.from_xyxy_array(bb_array,
            shape=image.shape)
        image_aug, bbox_aug = aug_params(image=image, bounding_boxes=bbox)
        bbox_aug = bbox_aug.remove_out_of_image()
        bbox_aug = bbox_aug.clip_out_of_image()
        if re.findall('Image...', str(bbox_aug)) == ['Image()']:
            pass
        else:
            imageio.imwrite(aug_images_path+image_prefix+filename, image_aug)
            info_df = group_df.drop(['xmin', 'ymin', 'xmax', 'ymax'], axis=1)
            for index, _ in info_df.iterrows():
                info_df.at[index, 'width'] = image_aug.shape[1]
                info_df.at[index, 'height'] = image_aug.shape[0]
            info_df['filename'] = info_df['filename'].apply(lambda x: image_prefix+x)
            bbox_df = bbox_obj_to_df(bbox_aug)
            aug_df = pd.concat([info_df, bbox_df], axis=1)
            aug_bbox = pd.concat([aug_bbox, aug_df])
    aug_bbox = aug_bbox.reset_index()
    aug_bbox = aug_bbox.drop(['index'], axis=1)
    return aug_bbox

```

**# ชุดคำสั่งในการบันทึกไฟล์บรรณนิทัศน์ XML จาก Dataframe ที่เกิดจากการขยายชุดภาพ**

```
def csv_to_xml(csv_path, images_path, labels_path, folder):
    path = open(csv_path, 'r')
    csv_reader = csv.reader(path)
    header = next(csv_reader)
    original_name = None
    for index in csv_reader:
        filename = index[0]
        if filename == original_name:
            object = ET.SubElement(annotation, 'object')
            ET.SubElement(object, 'name').text = index[3]
            ET.SubElement(object, 'pose').text = 'Unspecified'
            ET.SubElement(object, 'truncated').text = '0'
            ET.SubElement(object, 'difficult').text = '0'
            bndbox = ET.SubElement(object, 'bndbox')
            ET.SubElement(bndbox, 'xmin').text = index[4]
            ET.SubElement(bndbox, 'ymin').text = index[5]
            ET.SubElement(bndbox, 'xmax').text = index[6]
            ET.SubElement(bndbox, 'ymax').text = index[7]
        else:
            if original_name is not None:
                labels_file = original_filename.replace('.jpg', '.xml')
                tree = ET.ElementTree(annotation)
                tree.write(labels_path + labels_file)
            annotation = ET.Element('annotation')
            ET.SubElement(annotation, 'folder').text = folder
            ET.SubElement(annotation, 'filename').text = filename
            ET.SubElement(annotation, 'path').text = images_path + filename
            source = ET.SubElement(annotation, 'source')
            ET.SubElement(source, 'database').text = 'Engineering Chulalongkorn'
```

# ต่อหน้าถัดไป

```

# ชุดคำสั่งในการบันทึกไฟล์บรรณนิทัศน์ให้เท่ากับจำนวนชุดภาพที่ถูกขยายตามจำนวนรอบ
augment_loop = 5
for i in range(augment_loop):
    augmented_dataframe = image_aug(dataframe, 'images/', 'aug_images/',
        'aug{}_'.format(i), aug_params)
    augmented_dataframe.to_csv('aug{}_images.csv'.format(i), index=False)
    csv_to_xml(csv_path='aug{}_images.csv'.format(i), images_path='aug_images/',
        labels_path='aug_images/outputs/', folder='MicrosisDCN_Datasets')
    os.remove('aug{}_images.csv'.format(i))
print('Successfully augmented images to new datasets.')

```



```

# การกำหนดพารามิเตอร์สำหรับคลังชุดคำสั่ง imgaug เพื่อใช้ในฟังก์ชัน img_aug
aug_params = iaa.SomeOf(2, [
    iaa.Grayscale(alpha=1.0),
    iaa.GaussianBlur(sigma=(1.0, 3.0)),
    iaa.AdditiveGaussianNoise(scale=(0.03*255, 0.05*255)),
    iaa.MultiplyHueAndSaturation(mul_hue=(0.5, 1.5)),
    iaa.MultiplyHueAndSaturation(mul_saturation=(0.5, 1.5)),
    iaa.AddToHueAndSaturation((-10, 10), per_channel=True),
    iaa.MultiplyBrightness((0.9, 1.1)),
    iaa.GammaContrast((0.5, 1.0)),
    iaa.SigmoidContrast(gain=(3, 10), cutoff=(0.4, 0.6))]

```

### ภาคผนวก จ. วิธีการใช้งานชุดกล้อง MICROSISDCN กับกล้องจุลทรรศน์



- 1.) หลังจากติดตั้งระบบปฏิบัติการเสร็จสิ้นแล้ว ประกอบชุดเลนส์กำลังขยาย 0.5X เข้ากับชุดกล้อง ถ้าขนาดท่อเลนส์ใกล้ตาเป็น 23.2 มิลลิเมตร สามารถสวมได้ทันที แต่หากเป็นขนาดอื่นควรสวมแหวนรองตามขนาดที่เหมาะสม เช่น 30 หรือ 30.5 มิลลิเมตร



- 2.) ใช้ไขควงหมุนคลายสกรูที่ยึดเลนส์ใกล้ตาออก สามารถเลือกใช้ข้างซ้ายหรือข้างขวาได้ตามความถนัดของแต่ละบุคคล เนื่องจากเลนส์ใกล้ตาที่เหลืออีกข้างหนึ่ง ผู้ใช้สามารถใช้ส่องเพื่อปรับระยะโฟกัสของกล้องให้ชัดเจน เหมาะสมตามกำลังขยายของเลนส์ใกล้วัตถุ



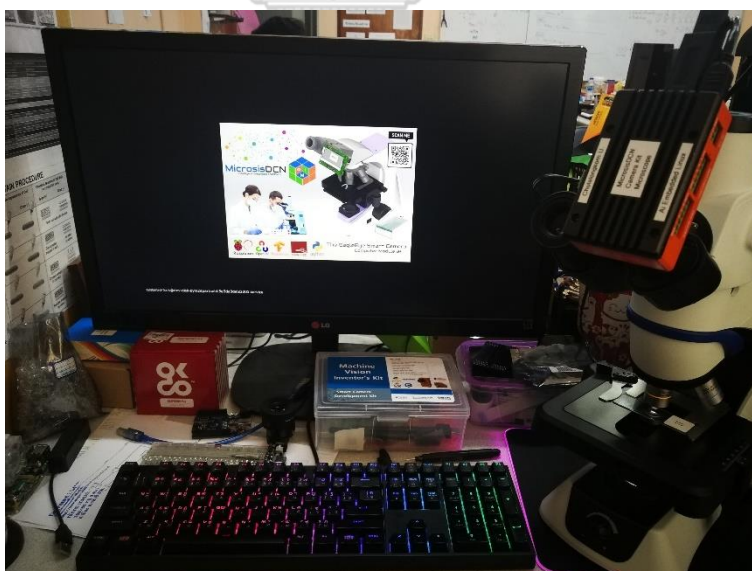
- 3.) สวมชุดกล้องเข้ากับกล้องจุลทรรศน์ให้แน่นสนิท จากนั้นใช้ไขควงหมุนสกรูยึดให้พอแน่น จึงจะสามารถใช้งานได้โดยที่ชุดกล้องไม่ขยับไปมา ค่าแนะนำเพิ่มเติมให้หันจุดเชื่อมต่อไปยังฝั่งที่เหมาะสม สำหรับเชื่อมต่อสายส่งสัญญาณภาพ และชุดเมาส์คีบอร์ดไร้สายต่อไป



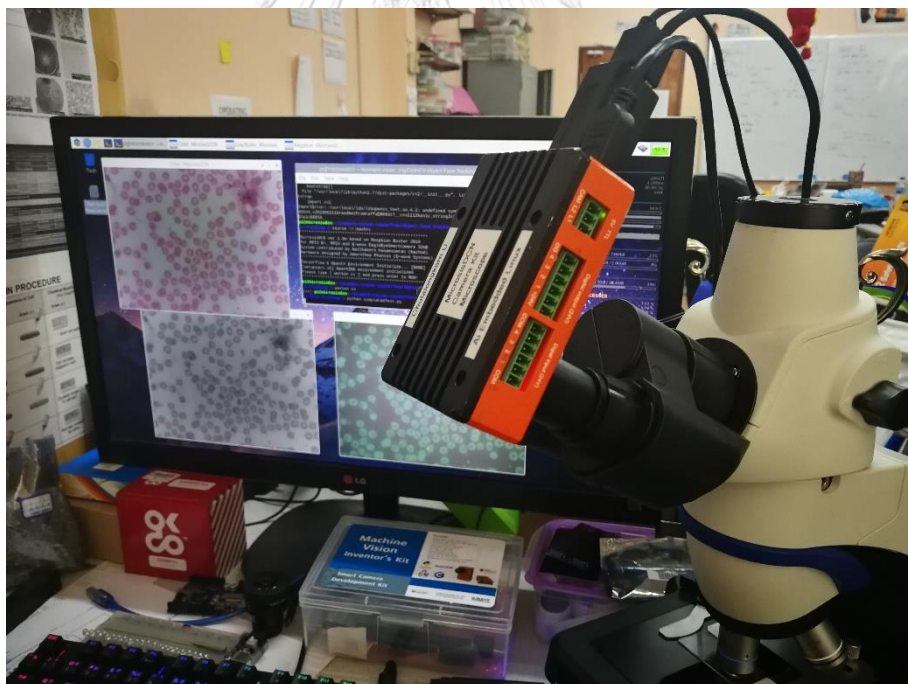
- 4.) สำหรับช่องเชื่อมต่อนั้นจะประกอบ 3 ส่วน ได้แก่ Ethernet port สำหรับเชื่อมต่อสายแลนในการต่อกับเครือข่ายภายนอกในการปรับปรุงระบบปฏิบัติการ USB host port เป็นช่องสำหรับเชื่อมต่ออุปกรณ์ที่ใช้ USB เช่น Wireless mouse & keyboard, USB hub สำหรับขยายช่องเชื่อมต่ออุปกรณ์ และ mini-HDMI port สำหรับเชื่อมต่อสายสัญญาณภาพแบบ HDMI เข้ากับจอแสดงผลที่รองรับ เช่น จอคอมพิวเตอร์ และโทรทัศน์อัจฉริยะ



- 5.) เชื่อมต่อแหล่งจ่ายไฟเลี้ยงระบบที่ได้จาก AC to DC Adaptor ชุดกล้อง MicroisDCN รุ่นนี้ ใช้แหล่งจ่ายไฟ 24 โวลต์ กระแส 2 แอมป์ ซึ่งจุดเชื่อมต่อนี้จะมี 24V สำหรับไฟบวก GND สำหรับไฟลบ และ E สำหรับเชื่อมสายดินในกรณีที่น่าไปใช้งานทางด้านอุตสาหกรรม และ สำหรับรุ่นถัดไปจะพัฒนาเป็นแบบใช้ Micro USB 5 โวลต์ 3 แอมป์



- 6.) หากเชื่อมต่ออุปกรณ์ต่าง ๆ ครบถ้วน รวมถึงแหล่งจ่ายไฟแล้ว ชุดกล้องจะบูต (System booting) ระบบขึ้นมาอย่างอัตโนมัติ และจะปรากฏภาพดังรูป ซึ่งแสดงว่าระบบกำลังเตรียมชุดคำสั่งและทรัพยากรที่จำเป็นต่อการใช้งานขึ้นมา



- 7.) เมื่อเข้าสู่หน้าจอ Desktop ของระบบ ให้เข้าไปที่หน้าต่าง Command Line หรือ Terminal เพื่อเริ่มพิมพ์ชุดคำสั่งสำหรับเรียก Virtual environment และระบบโครงข่ายประสาท เพื่อรับภาพจากกล้องจุลทรรศน์ และส่งไปประมวลผลต่อไป ผู้ใช้จะต้องนำแผ่นสไลด์ที่มีตัวอย่างเซลล์ขนาดเล็กที่ต้องการจำแนก และนับจำนวนสไลด์เลนส์ไกล้วถัดไป

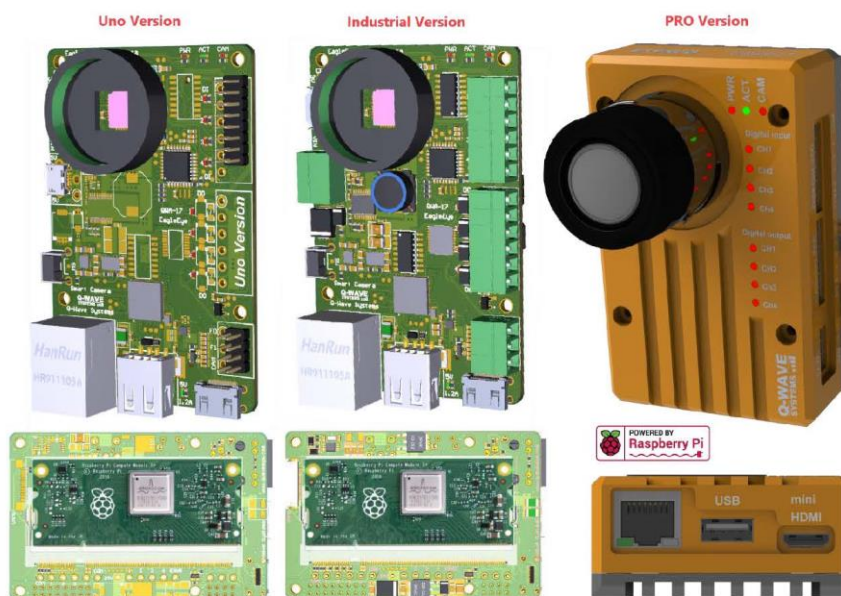


## ภาคผนวก ฉ. คุณสมบัติของชุดกล้อง EAGLEEYE SMART CAMERA

**Q-WAVE**  
SYSTEMS

EAGLE\_EYE\_SMART-CAMERA REV.1.0\_EN (2019)

### EagleEYE Smart Camera



[https://github.com/QWaveSystems/QwaveSmartCamera\\_EagleEYE](https://github.com/QWaveSystems/QwaveSmartCamera_EagleEYE)

Product Datasheet  
Developer Manual  
Getting Start Guide

## Revision History

Revision	Date	Comment	Editor
A 1.0.0	22/11/2019	Initial version	Amornthep Phunsin

## Order Part Number

Product	P/N	Price USD/THB
EagleEYE Uno Board	EY-UNO	
EagleEYE Industrial Board	EY-IND	
Full Size Heat-sink	EY-HSK	
EagleEYE Uno+ Heat sink+"CM3+ 32GB"	EY-UNO-32	
EagleEYE Industrial+ Heat sink+"CM3+ 32GB"	EY-IND-32	
EagleEYE Pro "CM3+ 32GB"	EY-PRO-32	
EagleEYE Pro Developer Kit - Raspberry Pi CM3+ 32GB - EagleEYE Flasher CM3+ board - x1 Power supply 24V 2A - x5 Lens CS mount 4/6/8/12/5-50mm - x1 RGB 40 LEDS (WS2812) - x1 mini HDMI to HDMI cable 1m - x1 LAN 100Mbps cable 1m - x1 USB 2.0 hub 3 port Slim version - x1 USB 2.0 WiFi 2.4GHz 802.11b/g/n (150Mbit/s) - x1 Din rail clipper	EY-PRO-KIT	
EagleEYE Uno Developer Kit - Raspberry Pi CM3+ 32GB - EagleEYE Flasher CM3+ board - x1 Power supply 5V 2.5A (micro USB) - x5 Lens CS mount 4/6/8/12/5-50mm - x1 RGB 40 LEDS (WS2812) - x1 mini HDMI to HDMI cable 1m - x1 LAN 100Mbps cable 1m - x1 USB 2.0 hub 3 port Slim version - x1 USB 2.0 WiFi 2.4GHz 802.11b/g/n (150Mbit/s)	EY-UNO-KIT	

## Target Application

Q-Wave Systems Co.,Ltd 65/2 Moo1 Bung Sriracha Chonburi 20230 Thailand,  
 Email : amornthep@qwavesys.com

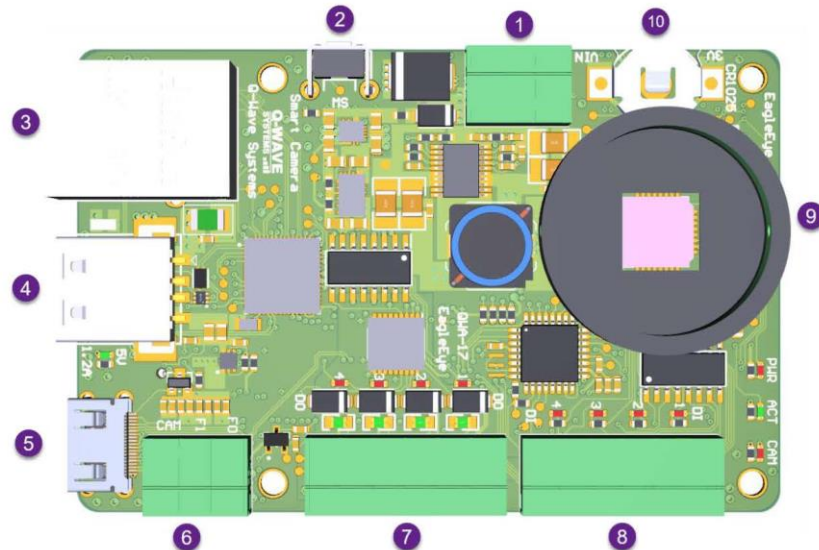
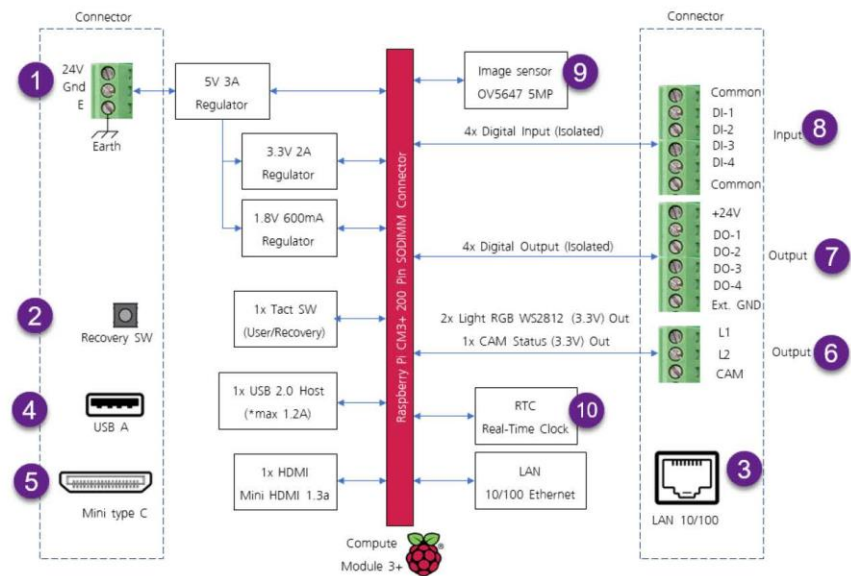
- Embedded vision
- industrial machine vision application
- Prototype computer vision algorithm
- Image processing analysis
- AI vision machine learning
- Drones navigation systems
- Robots vision, AGV (Automated Guided Vehicles)
- ADAS (Advanced Driver Assistance System)

## Product Specification

Specification	Industrial/Pro Version	Uno Version
CPU	Raspberry Pi Compute Module 3+ (CM3+) via SODIMM connector, CPU Broadcom BCM2837B0 1.2GHz quad core cortex-A53 processor, 1GB LPDDR2 RAM, eMMC 8GB/16GB/32GB flash	
Camera Sensor	5MP OV5647 1/4inchs CMOS RAW image sensor (CSI-2)	
Camera Resolution	QSXGA 2592x1944 (max), Video QVGA 320x240 @120fps (max)	
Lens Configuration	CS mount lens	
Video Output	1x Mini HDMI Port (HDMI V1.3a)	
Networking	1x 10/100M Ethernet	
USB Host	1x USB 2.0 host port up to 1.2A	
Camera Status	1x 3.3V Output (CAM)	
Light RGB Output	2x 3.3V Output (L1,L2) *Required external +5V 3A (RGB WS2812)	
Thermal Solution	Full size heat-sink (98x61mm)	
RTC	Real-Time clock onboard	-
HW Watchdog	Yes (onboard MCU)	-
Voltage Input	12V-24V Input (*min 25W)	5V 3A Input via uUSB
Circuit Protection	Polarity protection, Short circuit, Over voltage/current, Thermal shutdown	-
Temperature Range	-20 C – +85 C	0 C – +45 C
Digital Input	4x Isolated 4 Channel 24V input	4x 3.3V via GPIO header
Digital Output	4x Isolated 4 Channel 24V Output (Required external +24V supply)	4x 3.3V via GPIO header (*50mA total)
Dimension (W/L/H)	Board size 85x56x19.5mm	Board size 85x56x19.5mm
Weight	-	-
Dimension(heatsink)	with heat-sink 98x61x31mm	with heat-sink 98x61x28mm
Power Consumption	25W (max)	15W (max)
Software	Standard Raspbian OS, OpenCV, C++ and Python	

## Block Diagram

EagleEYE Industrial version



Pro Version (Industrial Version + Enclosure+ Heat-sink)

Q-Wave Systems Co.,Ltd 65/2 Moo1 Bung Sriracha Chonburi 20230 Thailand,  
Email : amornthep@qwavesys.com

ภาคผนวก ข. คุณสมบัติกล้องจุลทรรศน์และชุดเลนส์

**OLYMPUS**  
Your Vision, Our Future

Biological Microscope

**CX43/CX33**

CX3 Series

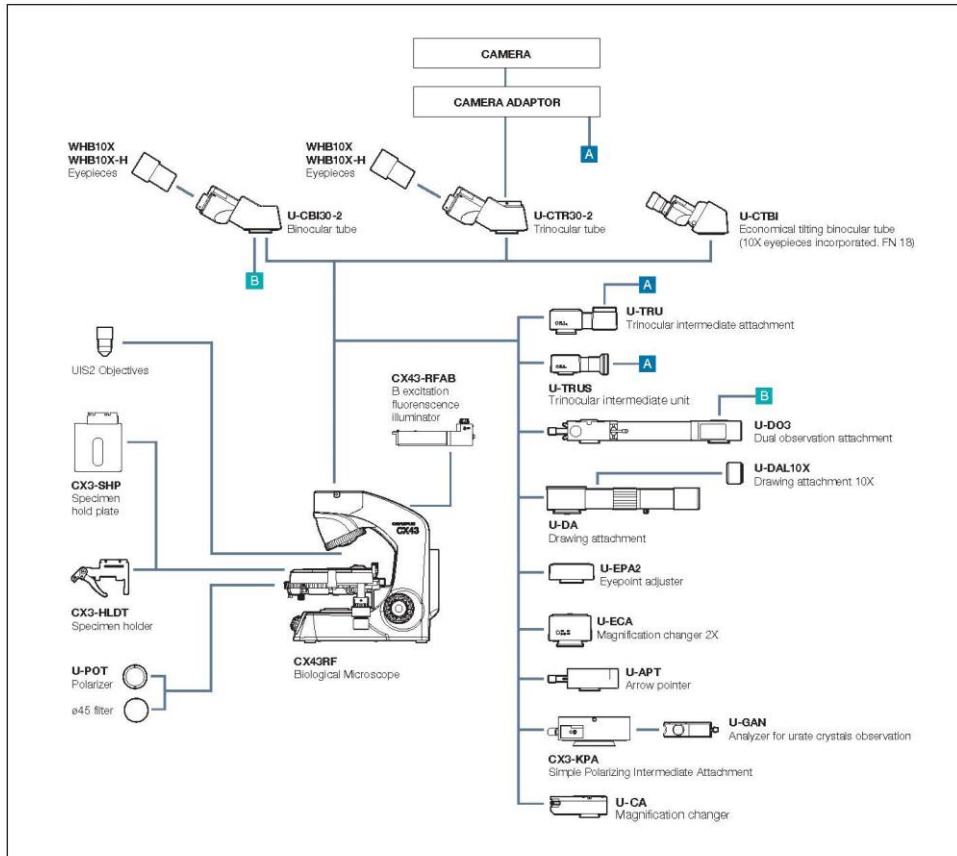
Comfortable, High-Throughput Routine Microscopy

**NEW**

**UIS2**  
World-leading optics

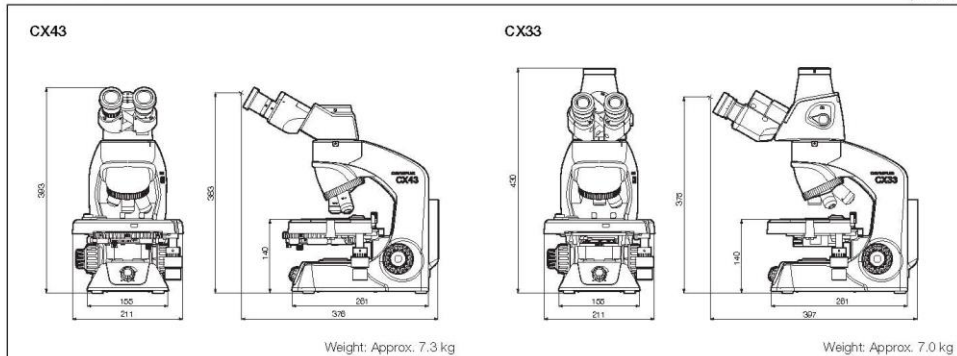


**CX43 System**



**Dimensions**

(Unit: mm)



**CX43 Specifications**

Optical System	UIS2 (universal infinity-corrected) optical system			
Illumination System	<ul style="list-style-type: none"> <li>Built-in transmitted illumination system</li> <li>Köhler illumination (fixed field diaphragm)</li> <li>LED power consumption 2.4 W (nominal value), precentered</li> </ul>			
Focusing	<ul style="list-style-type: none"> <li>Stage height movement (coarse movement stroke: 15 mm)</li> <li>Stroke per rotation for coarse adjustment knob: 36.8 mm, Focusing stopper</li> <li>Torque adjustment for coarse adjustment knob</li> <li>Fine focus knob (minimum adjustment gradations: 2.5 µm)</li> </ul>			
Revolving Nosepiece	Fixed quintuple nosepiece with inward tilt			
Stage	<ul style="list-style-type: none"> <li>Wire movement mechanical fixed stage, (W × D): 211 mm × 154 mm</li> <li>Traveling range (X × Y): 76 mm × 52 mm</li> <li>Single specimen holder (optional: double specimen holder, sheet holder)</li> <li>Specimen position scale</li> <li>Stage XY movement stopper</li> </ul>			
Observation Tube	Type (anti-fungal)	Binocular	Trinocular	Tilting binocular
	Eyepiece (anti-fungal)	10X Field Number (FN): 20	10X Field Number (FN): 20	10X Field Number (FN): 18
	Tube Inclination	30°	30°	30°–60°
	Light Path Selector	None	None (eyepiece/camera port = 50/50 fixed)	None
Interpupillary Distance Adjusting Range	48–75 mm			
Condenser	<ul style="list-style-type: none"> <li>Abbe condenser NA 1.25 with oil immersion</li> <li>Universal condenser with 7 turret positions: BF (4–100X), 2X, DF, Ph1, Ph2, Ph3, FL</li> <li>Condenser turret lock pin (BF only)</li> <li>Built-in aperture iris diaphragm</li> <li>AS lock pin</li> </ul>			
Observation Methods	Brightfield, simple polarization, fluorescence, phase contrast, darkfield			
Objectives	Plan achromat (UIS2), anti-fungal 2X NA 0.06 W.D. 5.8 mm 4X NA 0.1 W.D. 18.5 mm 10X NA 0.25 W.D. 10.6 mm 10XPH NA 0.25 W.D. 10.6 mm 20X NA 0.4 W.D. 1.2 mm 20XPH NA 0.4 W.D. 1.2 mm 40X NA 0.65 W.D. 0.6 mm 40XPH NA 0.65 W.D. 0.6 mm 60X NA 0.8 W.D. 0.2 mm 100XO NA 1.25 W.D. 0.13 mm 100XOPH NA 1.25 W.D. 0.15 mm 100XOI NA 1.25–0.6 W.D. 0.13 mm			
Fluorescence Light Source	Easily add an LED reflected fluorescence illuminator (peak excitation wavelength 470 nm: B excitation only), precentered			
Rated Voltage/Electric Current	AC 100–240 V 50/60 Hz 0.4 A			

**CX33 Specifications**

Optical System	Infinity optical system			
Illumination System	<ul style="list-style-type: none"> <li>Built-in transmitted illumination system</li> <li>Köhler illumination (fixed field diaphragm)</li> <li>LED power consumption 2.4 W (nominal value), precentered</li> </ul>			
Focusing	<ul style="list-style-type: none"> <li>Stage height movement (coarse movement stroke: 15 mm)</li> <li>Stroke per rotation for coarse adjustment knob: 36.8 mm, Focusing stopper</li> <li>Torque adjustment for coarse adjustment knob</li> <li>Fine focus knob (minimum adjustment gradations: 2.5 µm)</li> </ul>			
Revolving Nosepiece	Fixed quadruple nosepiece with inward tilt			
Stage	<ul style="list-style-type: none"> <li>Wire movement mechanical fixed stage, (W × D): 211 mm × 154 mm</li> <li>Traveling range (X × Y): 76 mm × 52 mm</li> <li>Single specimen holder (optional: double specimen holder, sheet holder)</li> <li>Specimen position scale</li> <li>Stage XY movement stopper</li> </ul>			
Observation Tube	<ul style="list-style-type: none"> <li>30° inclined trinocular tube (anti-fungal)</li> <li>Light path selector: eyepiece/camera port = 100/0 or 0/100</li> <li>Interpupillary distance adjusting range: 48–75 mm</li> <li>Eye point adjustment: 375.0–427.9 mm</li> </ul>			
Eyepieces (anti-fungal)	<ul style="list-style-type: none"> <li>10X Field Number (FN): 20</li> <li>15X Field Number (FN): 16 (optional)</li> </ul>			
Condenser	<ul style="list-style-type: none"> <li>Abbe condenser NA 1.25 with oil immersion</li> <li>Built-in aperture iris diaphragm</li> </ul>			
Observation Methods	Brightfield, darkfield			
Objectives	Plan achromat, anti-fungal 4X NA 0.1 W.D. 27.8 mm 10X NA 0.25 W.D. 8.0 mm 20X NA 0.4 W.D. 2.5 mm (optional) 40X NA 0.65 W.D. 0.6 mm 100X NA 1.25 W.D. 0.13 mm (optional)			
Rated Voltage/Electric Current	AC 100–240 V 50/60 Hz 0.4 A			

• **OLYMPUS CORPORATION** is ISO14001 certified.  
 • **OLYMPUS CORPORATION** is ISO9001 certified.  
 • **OLYMPUS CORPORATION** is ISO13485 certified.  
 • All company and product names are registered trademarks and/or trademarks of their respective owners.  
 • Specifications and appearances are subject to change without any notice or obligation on the part of the manufacturer.

[www.olympus-lifescience.com](http://www.olympus-lifescience.com)

**OLYMPUS**

**OLYMPUS CORPORATION**  
 3-3-1 Nishi-Shinjuku, Shinjuku-Ku, Tokyo 163-0914, Japan

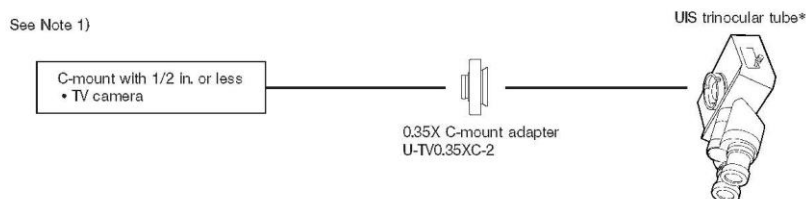
Printed in Japan N8600645-032017

## Instructions for the Low-Magnification C-Mount Adapter U-TV0.35XC-2

The U-TV0.35XC-2 is low-magnification TV adapter with C-mount which allows a digital imaging device such as a digital TV camera to capture wide-angle images. Use of telecentric optics helps reduce the occurrence of light deficiency in the peripheral sections.

In addition, the transmittance at the infrared frequency band is increased to 1,000 nm.

### 1 System Diagram



\* Can be used with an attachment having the same mounting structure as the UIS trinocular tube and straight photo tube.  
(U-TR30-2, U-TR30NIR, U-SWTR-3, U-DPT (port B), U-MPH, etc.)

#### Note 1) Restrictions on the TV camera

- TV camera cannot be used if its C-mount surface is located below the camera surface.
- TV camera may get in the way of the microscope operation if the camera's lateral size from the light axis exceeds 68 mm.
- When a TV camera having a larger CCD than specified is used, the image may lack brightness in the peripheral sections or a part of an image may be cut off.
- When the TV camera has high sensitivity or is not provided with automatic light control, the monitor image may become whitish. Should this happen, lower the light intensity level of the microscope.

### 2 Assembly

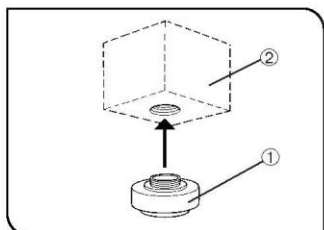


Fig. 1

1. Attach the C-mount adapter ① to the C-mount TV camera ② by screwing firmly. (Fig. 1)
2. Using the Allen screwdriver provided with the microscope, loosen the straight photo tube clamping screw ③, then fit the mount dovetail ④ of the C-mount adapter into the straight photo tube mount ⑤ of the trinocular tube. (Fig. 2)
3. Tighten the clamping screw ③ firmly. (Fig. 2)

★ For convenient confocality adjustment, set the C-mount adapter so that the LOCK and FOCUS screws face sideways.

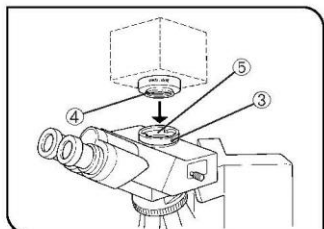


Fig. 2





### 3 Operation

#### Adjusting the Microscope

1. Turn on the microscope light source and adjust the required points of the microscope to make it ready for observation.
2. Set the light path of the UIS trinocular tube to the TV light path.

#### Adjusting the TV Camera and Monitor

Perform the adjustments such as color adjustment by referring to the instruction manuals of your TV camera and monitor.

★ The center of eyepiece and that of the monitor may not coincide correctly. This is a function of the CCD adjustment mechanism of the TV camera, not a malfunction.

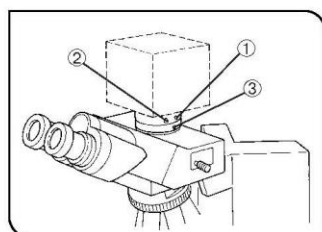


Fig. 3

#### Adjusting the Confocality Between the Observed Image and Monitored Image (Fig. 3)

Ⓞ The confocality adjustment requires the Allen wrench (for locking) provided with the adapter and the Allen screwdriver (for focusing) provided with the microscope.

★ The confocality adjustment range is  $\pm 0.25$  mm. If the adjustment of the adapter is not enough, please also adjust the focusing feature of the TV camera.

If the correct confocality cannot still be obtained, use another TV camera.

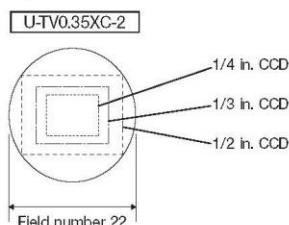
1. Look into the eyepiece and bring the specimen into focus.
2. Set the TV light path and switch to the monitor image.
3. Loosen the confocality adjustment screw (LOCK) Ⓞ using the Allen wrench.
4. While observing the monitor image, adjust focus by turning the confocality adjustment screw (FOCUS) Ⓜ slowly using the Allen screwdriver.
5. When correct focusing is obtained, hold the FOCUS screw position by keeping the Allen screwdriver inserted into it, and tighten the LOCK screw Ⓞ using the Allen wrench.

#### Rotating the Camera (Fig. 3)

Loosen the straight photo tube clamping screw Ⓞ.  
Rotate the TV camera and tighten the straight photo tube clamping screw Ⓞ firmly.

### 4 Imaging Field Areas

Ⓞ The following diagrams show the imaging field areas, which are determined by the field of view of the eyepiece (field number 22) and the size of the CCD seen through the C-mount adapter.



$$\text{Magnification on monitor} = \text{Objective magnification} \times \text{C-mount adapter magnification (0.35X)} \times \frac{\text{Monitor diagonal length}^*}{\text{CCD diagonal length}^*}$$

\* Differs depending on the manufacturer.

CCD reference: 1 in. TV camera → 16.16 mm, 2/3 in. → 11 mm, 1/2 in. → 8.08 mm, 1/3 in. → 6 mm, 1/4 in. → 4 mm

# OLYMPUS®



Printed on 100% recycled paper with soy ink.

Printed in Japan 2004 11 M 005-0

## Microscope Lens

### 0.5X L77 Microscope C Mount Ocular Adapter 23.2mm Electronic Eyepiece Reduction Lens

#### 0.5X Microscope lens for Microscope CCD Camera JT0506.0542



## Product Description

### Specification

Magnification: 0.5X

Mounting Size: 23.2mm (to connect binocular mount)

C-mount Thread: 25.4mm (to connect electronic eyepiece or CCD camera)

Suit Microscope: biological microscope and stereo microscope

### Warm Tips:

Generally, the binocular mount size of biological microscope is 23.2mm, the binocular mount size of stereo microscope is 30mm. If your microscope is stereo microscope, then you need to put another 23.2mm to 30mm adapter ring, then it will be OK. If you need this adapter ring, please contact us.

### Feature

0.5X magnification to get a more wide view of microscope

C-MOUNT adapter to connect CCD camera or digital eyepiece to microscope

## Detailed Image



**ภาคผนวก ข. ตารางค่าการสูญเสียและค่าความแม่นยำ**

ตารางที่ 34 ค่าการสูญเสียและค่าความแม่นยำจากการฝึกโมเดลไม่ปรับพารามิเตอร์

Epoch	Training loss	Classification loss	Classification accuracy	Classification categorical accuracy	mAP
0	0.546354	0.211964	0.309764	0.309764	0.6794
1	0.276496	0.022985	0.779393	0.779393	0.6955
2	0.225215	0.013785	0.891131	0.891131	0.7097
3	0.184216	0.009834	0.910587	0.910587	0.7100
4	0.149205	0.007553	0.917360	0.917360	0.7110
5	0.122888	0.006046	0.919347	0.919347	0.7122
6	0.102213	0.005112	0.921844	0.921844	0.7182
7	0.085997	0.004367	0.922571	0.922571	0.7122
8	0.074309	0.003824	0.922697	0.922697	0.7156
9	0.066571	0.003362	0.920479	0.920479	0.7266
10	0.057986	0.002938	0.919367	0.919367	0.7215
11	0.051161	0.002580	0.918210	0.918210	0.7228
12	0.046516	0.002222	0.916253	0.916253	0.7231
13	0.040328	0.001920	0.914046	0.914046	0.7356
14	0.037094	0.001667	0.911350	0.911350	0.7216
15	0.033325	0.001389	0.908054	0.908054	0.7093
16	0.031034	0.001182	0.905539	0.905539	0.7124
17	0.027595	0.001011	0.902165	0.902165	0.7253
18	0.025453	0.000862	0.899336	0.899336	0.7230
19	0.024408	0.000750	0.897608	0.897608	0.7110
20	0.022417	0.000611	0.894237	0.894237	0.7201
21	0.022310	0.000575	0.892514	0.892514	0.7141
22	0.020176	0.000462	0.889198	0.889198	0.7109
23	0.019391	0.000404	0.887104	0.887104	0.7075
24	0.018528	0.000374	0.884218	0.884218	0.7054

ตารางที่ 35 ค่าการสูญเสียและค่าความแม่นยำจากการฝึกโมเดลไม่ปรับพารามิเตอร์ (ต่อ)

Epoch	Training loss	Classification loss	Classification accuracy	Classification categorical accuracy	mAP
25	0.017739	0.000315	0.882214	0.882214	0.7017
26	0.016645	0.000284	0.879824	0.879824	0.7100
27	0.016086	0.000256	0.878217	0.878217	0.7053
28	0.015920	0.000223	0.875803	0.875803	0.7062
29	0.015134	0.000189	0.873243	0.873243	0.7041
30	0.014084	0.000176	0.869842	0.869842	0.6992
31	0.014627	0.000144	0.867929	0.867929	0.6989
32	0.013965	0.000136	0.866101	0.866101	0.6975
33	0.013108	0.000120	0.861399	0.861399	0.7000
34	0.012608	0.000140	0.856019	0.856019	0.6989
35	0.012898	0.000090	0.852230	0.852230	0.6991
36	0.012371	0.000091	0.849305	0.849305	0.6896
37	0.011630	0.000086	0.844473	0.844473	0.6955
38	0.011098	0.000079	0.839335	0.839335	0.6927
39	0.011125	0.000067	0.833849	0.833849	0.6952
40	0.010646	0.000065	0.828758	0.828758	0.6946
41	0.010357	0.000057	0.825486	0.825486	0.6906
42	0.010038	0.000066	0.817786	0.817786	0.6919
43	0.009559	0.000078	0.818680	0.818680	0.6945
44	0.009631	0.000048	0.819604	0.819604	0.6945
45	0.009533	0.000042	0.815624	0.815624	0.6964
46	0.004336	0.000031	0.814417	0.814417	0.6935
47	0.003452	0.000028	0.813159	0.813159	0.6929
48	0.003188	0.000026	0.812780	0.812780	0.6924
49	0.003062	0.000025	0.812539	0.812539	0.6924

ตารางที่ 36 ตารางค่าการสูญเสียและค่าความแม่นยำจากการตรวจสอบความถูกต้องของโมเดลไม่  
ปรับพารามิเตอร์

Epoch	Validation loss	Classification loss	Classification accuracy	Classification categorical accuracy	mAP
0	0.393049	0.084162	0.682283	0.682283	0.6794
1	0.298703	0.021737	0.877759	0.877759	0.6955
2	0.333216	0.061893	0.865059	0.865059	0.7097
3	0.250786	0.004545	0.889706	0.889706	0.7100
4	0.140566	0.004530	0.946961	0.946961	0.7110
5	0.176536	0.002066	0.890032	0.890032	0.7122
6	0.125619	0.018813	0.955861	0.955861	0.7182
7	0.053817	0.013462	0.952406	0.952406	0.7122
8	0.155525	0.001282	0.874074	0.874074	0.7156
9	0.020894	0.003961	0.958284	0.958284	0.7266
10	0.019949	0.004197	0.968212	0.968212	0.7215
11	0.076791	0.002647	0.883543	0.883543	0.7228
12	0.125451	0.043453	0.969557	0.969557	0.7231
13	0.048477	0.013693	0.964130	0.964130	0.7356
14	0.023646	0.002749	0.936470	0.936470	0.7216
15	0.045748	0.002720	0.846486	0.846486	0.7093
16	0.044061	0.000264	0.858542	0.858542	0.7124
17	0.014689	0.000726	0.943864	0.943864	0.7253
18	0.027757	0.000394	0.841075	0.841075	0.7230
19	0.054369	0.015872	0.946339	0.946339	0.7110
20	0.013171	0.001519	0.943444	0.943444	0.7201
21	0.027263	0.000200	0.817394	0.817394	0.7141
22	0.010305	0.000230	0.945609	0.945609	0.7109
23	0.011325	0.000255	0.958672	0.958672	0.7075
24	0.026443	0.000317	0.831675	0.831675	0.7054

ตารางที่ 37 ค่าการสูญเสียและค่าความแม่นยำจากการตรวจสอบความถูกต้องของโมเดลไม่ปรับพารามิเตอร์ (ต่อ)

Epoch	Validation loss	Classification loss	Classification accuracy	Classification categorical accuracy	mAP
25	0.031861	0.004575	0.962552	0.962552	0.7017
26	0.013860	0.001305	0.953453	0.953453	0.7100
27	0.023922	0.001927	0.920342	0.920342	0.7053
28	0.035326	0.000350	0.804636	0.804636	0.7062
29	0.015648	0.000060	0.803623	0.803623	0.7041
30	0.008609	0.000079	0.929897	0.929897	0.6992
31	0.011876	0.000064	0.785112	0.785112	0.6989
32	0.023646	0.003558	0.928900	0.928900	0.6975
33	0.012920	0.000161	0.929049	0.929049	0.7000
34	0.011496	0.000062	0.756663	0.756663	0.6989
35	0.009101	0.000061	0.930329	0.930329	0.6991
36	0.005188	0.000054	0.932323	0.932323	0.6896
37	0.014510	0.000098	0.769232	0.769232	0.6955
38	0.011313	0.000651	0.945473	0.945473	0.6927
39	0.006896	0.000196	0.926200	0.926200	0.6952
40	0.009910	0.000338	0.877991	0.877991	0.6946
41	0.012302	0.000141	0.731678	0.731678	0.6906
42	0.009213	0.000018	0.735552	0.735552	0.6919
43	0.007120	0.000015	0.894359	0.894359	0.6945
44	0.016377	0.000048	0.735962	0.735962	0.6945
45	0.012673	0.000321	0.896403	0.896403	0.6964
46	0.002755	0.000031	0.896136	0.896136	0.6935
47	0.005065	0.000025	0.710728	0.710728	0.6929
48	0.001522	0.000019	0.886634	0.886634	0.6924
49	0.001269	0.000006	0.912644	0.912644	0.6924

ตารางที่ 38 ค่าการสูญเสียและค่าความแม่นยำจากการฝึกโมเดลปรับพารามิเตอร์แบบที่ 1

Epoch	Training loss	Classification loss	Classification accuracy	Classification categorical accuracy	mAP
0	1.478483	0.345085	0.561042	0.561042	0.7305
1	0.686343	0.147006	0.826249	0.826249	0.7470
2	0.575307	0.122613	0.868887	0.868887	0.7656
3	0.516845	0.112121	0.879827	0.879827	0.7606
4	0.473200	0.103733	0.891428	0.891428	0.7675
5	0.438480	0.098911	0.894891	0.894891	0.7727
6	0.414763	0.095744	0.899013	0.899013	0.7776
7	0.391345	0.091941	0.903553	0.903553	0.7762
8	0.369758	0.089653	0.904445	0.904445	0.7809
9	0.353284	0.087635	0.908611	0.908611	0.7875
10	0.334237	0.084787	0.907611	0.907611	0.7963
11	0.318817	0.082038	0.908758	0.908758	0.7971
12	0.303272	0.080337	0.908049	0.908049	0.8034
13	0.293662	0.079269	0.909997	0.909997	0.7940
14	0.279632	0.076595	0.908360	0.908360	0.8110
15	0.267025	0.076186	0.906800	0.906800	0.8042
16	0.257320	0.073039	0.907433	0.907433	0.8162
17	0.246281	0.070641	0.904795	0.904795	0.8185
18	0.237755	0.069602	0.903229	0.903229	0.8225
19	0.228042	0.064832	0.899143	0.899143	0.8218
20	0.218272	0.063195	0.899253	0.899253	0.8313
21	0.204184	0.057181	0.893423	0.893423	0.8353
22	0.195166	0.056155	0.893553	0.893553	0.8328
23	0.187397	0.051894	0.884890	0.884890	0.8445
24	0.178154	0.048796	0.881238	0.881238	0.8536

ตารางที่ 39 ค่าการสูญเสียและค่าความแม่นยำจากการฝึกโมเดลปรับพารามิเตอร์แบบที่ 1 (ต่อ)

Epoch	Training loss	Classification loss	Classification accuracy	Classification categorical accuracy	mAP
25	0.168884	0.043555	0.875975	0.875975	0.8545
26	0.161229	0.039201	0.866488	0.866488	0.8620
27	0.148832	0.034190	0.855417	0.855417	0.8545
28	0.145174	0.033855	0.842271	0.842271	0.8635
29	0.133059	0.027790	0.836752	0.836752	0.8622
30	0.125367	0.023027	0.825224	0.825224	0.8634
31	0.116606	0.019170	0.815046	0.815046	0.8652
32	0.112944	0.016927	0.806939	0.806939	0.8674
33	0.105473	0.014490	0.794635	0.794635	0.8671
34	0.100087	0.012227	0.786507	0.786507	0.8648
35	0.097698	0.011076	0.781148	0.781148	0.8664
36	0.093189	0.010410	0.774218	0.774218	0.8671
37	0.092433	0.009701	0.769294	0.769294	0.8671
38	0.082026	0.007943	0.768655	0.768655	0.8683
39	0.080420	0.008150	0.755099	0.755099	0.8676
40	0.078535	0.006955	0.752456	0.752456	0.8676
41	0.074127	0.007454	0.742179	0.742179	0.8686
42	0.071258	0.006169	0.740466	0.740466	0.8674
43	0.070555	0.006069	0.738429	0.738429	0.8686
44	0.068665	0.006301	0.741228	0.741228	0.8686
45	0.066246	0.005468	0.726941	0.726941	0.8681
46	0.062713	0.005402	0.720808	0.720808	0.8686
47	0.064052	0.005553	0.726734	0.726734	0.8684
48	0.064277	0.005246	0.726872	0.726872	0.8686
49	0.036821	0.003664	0.726737	0.726737	0.8681



ตารางที่ 40 ค่าการสูญเสียและค่าความแม่นยำจากการตรวจสอบความถูกต้องของโมเดลปรับพารามิเตอร์แบบที่ 1

Epoch	Validation loss	Classification loss	Classification accuracy	Classification categorical accuracy	mAP
0	1.387762	0.357434	0.812598	0.812598	0.7305
1	1.206033	0.350856	0.876573	0.876573	0.7470
2	0.781084	0.179423	0.850292	0.850292	0.7656
3	0.319170	0.025700	0.861098	0.861098	0.7606
4	0.354003	0.053932	0.922095	0.922095	0.7675
5	0.302864	0.019545	0.852235	0.852235	0.7727
6	0.947408	0.254685	0.945866	0.945866	0.7776
7	1.065072	0.387375	0.934038	0.934038	0.7762
8	0.296848	0.019022	0.876889	0.876889	0.7809
9	0.137050	0.042048	0.934464	0.934464	0.7875
10	0.803216	0.225400	0.954051	0.954051	0.7963
11	0.516305	0.130676	0.896488	0.896488	0.7971
12	1.153359	0.408156	0.949156	0.949156	0.8034
13	0.832400	0.196625	0.959498	0.959498	0.7940
14	0.722132	0.252161	0.935622	0.935622	0.8110
15	0.468297	0.106723	0.894377	0.894377	0.8042
16	0.137646	0.006033	0.884173	0.884173	0.8162
17	0.158026	0.030879	0.942344	0.942344	0.8185
18	0.164331	0.006890	0.878707	0.878707	0.8225
19	0.628602	0.190740	0.953974	0.953974	0.8218
20	0.670804	0.228251	0.951088	0.951088	0.8313
21	0.102631	0.004381	0.834787	0.834787	0.8353
22	0.042529	0.009496	0.869986	0.869986	0.8328
23	0.437927	0.141481	0.941051	0.941051	0.8445
24	0.276683	0.066816	0.868197	0.868197	0.8536

ตารางที่ 41 ค่าการสูญเสียและค่าความแม่นยำจากการตรวจสอบความถูกต้องของโมเดลปรับพารามิเตอร์แบบที่ 1 (ต่อ)

Epoch	Validation loss	Classification loss	Classification accuracy	Classification categorical accuracy	mAP
25	0.708568	0.263798	0.943716	0.943716	0.8545
26	0.504059	0.083015	0.960936	0.960936	0.8620
27	0.437973	0.156245	0.881076	0.881076	0.8545
28	0.235737	0.039355	0.868842	0.868842	0.8635
29	0.074760	0.001736	0.816899	0.816899	0.8622
30	0.074500	0.001806	0.829620	0.829620	0.8634
31	0.111657	0.002574	0.743004	0.743004	0.8652
32	0.284421	0.051249	0.943905	0.943905	0.8674
33	0.299237	0.040580	0.949892	0.949892	0.8671
34	0.053826	0.001945	0.723925	0.723925	0.8648
35	0.020801	0.001400	0.767655	0.767655	0.8664
36	0.127524	0.020843	0.872430	0.872430	0.8671
37	0.132711	0.017847	0.781318	0.781318	0.8671
38	0.317949	0.035194	0.943362	0.943362	0.8683
39	0.282674	0.041359	0.926971	0.926971	0.8676
40	0.204753	0.025917	0.821477	0.821477	0.8676
41	0.125443	0.012715	0.806158	0.806158	0.8686
42	0.037757	0.000945	0.680128	0.680128	0.8674
43	0.045046	0.000879	0.766565	0.766565	0.8686
44	0.072457	0.001200	0.701912	0.701912	0.8686
45	0.198099	0.018087	0.931195	0.931195	0.8681
46	0.265539	0.023626	0.953271	0.953271	0.8686
47	0.030692	0.000744	0.694380	0.694380	0.8684
48	0.014706	0.000391	0.726238	0.726238	0.8686
49	0.037643	0.006943	0.841872	0.841872	0.8681

ตารางที่ 42 ค่าการสูญเสียและค่าความแม่นยำจากการฝึกโมเดลปรับพารามิเตอร์แบบที่ 2

Epoch	Training loss	Classification loss	Classification accuracy	Classification categorical accuracy	mAP
0	3.088294	0.693992	0.724982	0.724982	0.0037
1	2.874957	0.608275	0.924445	0.924445	0.0066
2	2.819309	0.591376	0.935390	0.935390	0.0071
3	2.743294	0.570361	0.939497	0.939497	0.0101
4	2.609475	0.546833	0.942477	0.942477	0.0102
5	2.401595	0.525900	0.946078	0.946078	0.0133
6	2.163316	0.498304	0.940596	0.940596	0.0245
7	1.918852	0.469835	0.931986	0.931986	0.0291
8	1.735844	0.444592	0.910357	0.910357	0.0398
9	1.590941	0.420611	0.884825	0.884825	0.0521
10	1.465767	0.396693	0.854422	0.854422	0.0589
11	1.349240	0.372372	0.819532	0.819532	0.0702
12	1.257773	0.350753	0.783199	0.783199	0.0823
13	1.165586	0.328390	0.758810	0.758810	0.0927
14	1.097456	0.311795	0.732395	0.732395	0.0964
15	1.016067	0.291286	0.699229	0.699229	0.1047
16	0.953407	0.275571	0.675777	0.675777	0.1111
17	0.895469	0.261327	0.649305	0.649305	0.1171
18	0.847583	0.247900	0.629304	0.629304	0.1212
19	0.799113	0.234683	0.610759	0.610759	0.1285
20	0.755661	0.223212	0.594302	0.594302	0.1280
21	0.721912	0.212417	0.575335	0.575335	0.1328
22	0.689313	0.204256	0.559212	0.559212	0.1385
23	0.654330	0.192859	0.547969	0.547969	0.1424
24	0.615838	0.181209	0.538340	0.538340	0.1466

ตารางที่ 43 ค่าการสูญเสียและค่าความแม่นยำจากการฝึกโมเดลปรับพารามิเตอร์แบบที่ 2 (ต่อ)

Epoch	Training loss	Classification loss	Classification accuracy	Classification categorical accuracy	mAP
25	0.591498	0.174739	0.527332	0.527332	0.1525
26	0.568489	0.168083	0.520155	0.520155	0.1494
27	0.541451	0.157622	0.513290	0.513290	0.1547
28	0.515302	0.150083	0.502542	0.502542	0.1681
29	0.494246	0.142551	0.488475	0.488475	0.1700
30	0.477905	0.136976	0.485197	0.485197	0.1722
31	0.456866	0.130729	0.475518	0.475518	0.1816
32	0.440572	0.124997	0.459730	0.459730	0.1839
33	0.420277	0.116493	0.452034	0.452034	0.1850
34	0.406416	0.112790	0.446859	0.446859	0.1918
35	0.393002	0.106281	0.436911	0.436911	0.1974
36	0.375356	0.102407	0.428823	0.428823	0.2031
37	0.366037	0.098456	0.419702	0.419702	0.2033
38	0.349636	0.092248	0.407091	0.407091	0.2079
39	0.338038	0.088280	0.394830	0.394830	0.2110
40	0.330138	0.085526	0.392040	0.392040	0.2137
41	0.317117	0.080051	0.379784	0.379784	0.2167
42	0.305660	0.077528	0.365059	0.365059	0.2175
43	0.298786	0.072810	0.361506	0.361506	0.2191
44	0.288062	0.069911	0.353562	0.353562	0.2214
45	0.278723	0.067934	0.344133	0.344133	0.2228
46	0.271974	0.064145	0.335803	0.335803	0.2245
47	0.260596	0.059845	0.326148	0.326148	0.2261
48	0.253188	0.057103	0.316137	0.316137	0.2254
49	0.247623	0.055571	0.313912	0.313912	0.2293

ตารางที่ 44 ค่าการสูญเสียและค่าความแม่นยำจากการตรวจสอบความถูกต้องของโมเดลปรับพารามิเตอร์แบบที่ 2

Epoch	Validation loss	Classification loss	Classification accuracy	Classification categorical accuracy	mAP
0	3.111521	0.745426	0.902466	0.902466	0.0037
1	3.109320	0.691437	0.935841	0.935841	0.0066
2	2.933176	0.595174	0.936570	0.936570	0.0071
3	2.306385	0.358443	0.937707	0.937707	0.0101
4	2.834537	0.707954	0.954340	0.954340	0.0102
5	2.145665	0.413663	0.937428	0.937428	0.0133
6	2.820024	0.611886	0.954897	0.954897	0.0245
7	2.816406	0.647737	0.950461	0.950461	0.0291
8	1.480814	0.325129	0.853396	0.853396	0.0398
9	1.083802	0.337868	0.907825	0.907825	0.0521
10	1.733700	0.587485	0.904393	0.904393	0.0589
11	1.786376	0.468763	0.693541	0.693541	0.0702
12	1.891225	0.619121	0.884657	0.884657	0.0823
13	1.732942	0.592202	0.876834	0.876834	0.0927
14	1.633862	0.466560	0.823874	0.823874	0.0964
15	1.257183	0.429140	0.571950	0.571950	0.1047
16	0.896687	0.201609	0.518895	0.518895	0.1111
17	1.303827	0.338124	0.681645	0.681645	0.1171
18	0.684887	0.211111	0.432141	0.432141	0.1212
19	1.425700	0.521661	0.846213	0.846213	0.1285
20	1.348031	0.466908	0.873197	0.873197	0.1280
21	0.609734	0.139106	0.381086	0.381086	0.1328
22	0.440651	0.117373	0.533487	0.533487	0.1385
23	0.711415	0.363265	0.712599	0.712599	0.1424
24	0.970069	0.316351	0.486869	0.486869	0.1466

ตารางที่ 45 ค่าการสูญเสียและค่าความแม่นยำจากการตรวจสอบความถูกต้องของโมเดลปรับพารามิเตอร์แบบที่ 2 (ต่อ)

Epoch	Validation loss	Classification loss	Classification accuracy	Classification categorical accuracy	mAP
25	1.079193	0.470609	0.719442	0.719442	0.1525
26	1.249426	0.526870	0.653929	0.653929	0.1494
27	0.806535	0.297760	0.614143	0.614143	0.1547
28	0.638288	0.244457	0.427347	0.427347	0.1681
29	0.368057	0.086857	0.335523	0.335523	0.1700
30	0.824946	0.227862	0.413610	0.413610	0.1722
31	0.362853	0.093581	0.330416	0.330416	0.1816
32	0.877530	0.346046	0.781096	0.781096	0.1839
33	0.831467	0.351490	0.796648	0.796648	0.1850
34	0.373922	0.069314	0.274081	0.274081	0.1918
35	0.233724	0.050204	0.508241	0.508241	0.1974
36	0.436846	0.191984	0.606361	0.606361	0.2031
37	0.549797	0.199791	0.401250	0.401250	0.2033
38	0.581352	0.250361	0.631966	0.631966	0.2079
39	0.685399	0.270740	0.609367	0.609367	0.2110
40	0.427647	0.154884	0.551375	0.551375	0.2137
41	0.349387	0.110729	0.325533	0.325533	0.2167
42	0.205457	0.039296	0.237682	0.237682	0.2175
43	0.540272	0.102553	0.343776	0.343776	0.2191
44	0.242796	0.054141	0.172519	0.172519	0.2214
45	0.501893	0.185961	0.670991	0.670991	0.2228
46	0.472205	0.187385	0.741361	0.741361	0.2245
47	0.240758	0.035380	0.177081	0.177081	0.2261
48	0.099279	0.013889	0.306654	0.306654	0.2254
49	0.191235	0.074095	0.379457	0.379457	0.2293

## บรรณานุกรม

1. Flentie, K., et al., *Chemical disarming of isoniazid resistance in Mycobacterium tuberculosis*. Proceedings of the National Academy of Sciences, 2019. **116**(21): p. 10510-10517.
2. Chen, M.L., et al., *Beyond multidrug resistance: Leveraging rare variants with machine and statistical learning models in Mycobacterium tuberculosis resistance prediction*. EBioMedicine, 2019. **43**: p. 356-369.
3. Adewoyin, A.S. and B. Nwogoh, *Peripheral blood film - a review*. Annals of Ibadan postgraduate medicine, 2014. **12**(2): p. 71-79.
4. Payne, D., *Use and limitations of light microscopy for diagnosing malaria at the primary health care level*. Bull World Health Organ, 1988. **66**(5): p. 621-6.
5. ปรีชาพล ชูศรี, จีรณา น้อยมณี, and เกษม พันธุ์สิน, เทคโนโลยีปัญญาประดิษฐ์ สำหรับงานบริหารงานและการบริการภาครัฐ. พศจิกายน 2562, ed. 1.0. 2019, บริษัท ส.พิจิตรการพิมพ์ จำกัด: สำนักงานพัฒนารัฐบาลดิจิทัล (องค์การมหาชน).
6. Rosenblatt, F.F., *The perceptron: a probabilistic model for information storage and organization in the brain*. Psychological review, 1958. **65** **6**: p. 386-408.
7. Elman, J.L., *Finding Structure in Time*. Cognitive Science, 1990. **14**(2): p. 179-211.
8. VEEN, F.V., *The Neural Network Zoo*. 2016, The Asimov Institute.
9. Y. Lecun, L.B., Y. Bengio and P. Haffner, *Gradient-based learning applied to document recognition*. Proceedings of the IEEE, 1998. **86**(11): p. 2278-2324.
10. จักรกริช ไกรเนตร and จักรกฤษณ์ ประดุงชนม์. เรียนรู้และทำความเข้าใจเรื่อง Convolutional Neural Network (CNN) คืออะไร. 2019 [cited 2020 30 April]; Available from: <https://www.glurgeek.com/education/ml-cnn/>.
11. Abadi, M., et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. arXiv e-prints, 2016.
12. Moolayil, J., *Learn Keras for Deep Neural Networks: A Fast-Track Approach to Modern Deep Learning with Python*. 2018: Apress. 200.
13. Bradski, G. and A. Kaehler, *Learning OpenCV: Computer Vision in C++ with the OpenCV Library*. 2013: O'Reilly Media, Inc. 575.

14. Everingham, M., et al., *The Pascal Visual Object Classes Challenge: A Retrospective*. International Journal of Computer Vision, 2015. **111**(1): p. 98-136.
15. Holik, A.S., *Optical Microscopy*, in *Encyclopedia of Materials: Science and Technology*, K.H.J. Buschow, et al., Editors. 2001, Elsevier: Oxford. p. 6458-6463.
16. Wayne, R., *Chapter 4 - Bright-Field Microscopy*, in *Light and Video Microscopy (Third Edition)*, R. Wayne, Editor. 2019, Academic Press. p. 95-116.
17. Wayne, R., *Chapter 2 - The Geometric Relationship Between Object and Image*, in *Light and Video Microscopy (Third Edition)*, R. Wayne, Editor. 2019, Academic Press. p. 17-50.
18. Wayne, R., *Chapter 3 - The Dependence of Image Formation on the Nature of Light*, in *Light and Video Microscopy (Third Edition)*, R. Wayne, Editor. 2019, Academic Press. p. 51-94.
19. Rudi Rottenfusser, E.E.W., Michael W. Davidson. *Education in Microscopy and Digital Imaging*. [cited 2020 14 May]; Available from: <http://zeiss-campus.magnet.fsu.edu/articles/basics/resolution.html>.
20. Castleman, K.R. and I.T. Young, *Chapter 2 - Fundamentals of Microscopy*, in *Microscope Image Processing*, Q. Wu, F.A. Merchant, and K.R. Castleman, Editors. 2008, Academic Press: Burlington. p. 11-25.
21. Davidson, M.A.M.W. *Olympus BX51 Microscope Cutaway Diagram*. 2001 [cited 2020 17 May]; Available from: <https://micro.magnet.fsu.edu/primer/anatomy/bx51cutaway.html>.
22. Rodak, B.F., G.A. Fritsma, and E.M. Keohane, *Hematology: Clinical Principles and Applications*. 2012: Elsevier Saunders.
23. Qingmin, L. and D. Yingying. *An accurate segmentation method for white blood cell images*. in *Proceedings IEEE International Symposium on Biomedical Imaging*. 2002.
24. Kan, J., L. Qing-Min, and D. Sheng-Yang. *A novel white blood cell segmentation scheme using scale-space filtering and watershed clustering*. in *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693)*. 2003.
25. Fang, Y., et al. *White Blood Cell Image Segmentation Using On-line Trained*



- Neural Network*. in *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*. 2005.
26. Wu, J., et al. *A novel color image segmentation method and its application to white blood cell image analysis*. in *2006 8th international Conference on Signal Processing*. 2006.
  27. Dorini, L.B., R. Minetto, and N.J. Leite. *White blood cell segmentation using morphological operators and scale-space analysis*. in *XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2007)*. 2007.
  28. Gao, W., Y. Tang, and X. Li. *Segmentation of Microscopic Images for Counting Leukocytes*. in *2008 2nd International Conference on Bioinformatics and Biomedical Engineering*. 2008.
  29. Theera-Umpon, N. and S. Dhompongsa, *Morphological Granulometric Features of Nucleus in Automatic Bone Marrow White Blood Cell Classification*. *IEEE Transactions on Information Technology in Biomedicine*, 2007. **11**(3): p. 353-359.
  30. Sajjad, M., et al., *Leukocytes Classification and Segmentation in Microscopic Blood Smear: A Resource-Aware Healthcare Service in Smart Cities*. *IEEE Access*, 2017. **5**: p. 3475-3489.
  31. Agrawal, R., et al. *Detection of White Blood Cell Cancer using Image Processing*. in *2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN)*. 2019.
  32. Gay, W., *Raspberry Pi Hardware Reference*. 2014: Apress. 248.
  33. Zolotkov, A. *Microscope Magnification Calculator*. 2017 [cited 2020 1 June 2020]; Available from: <https://www.translatorscafe.com/unit-converter/id-ID/calculator/microscope-magnification/>.
  34. Meuten, D., F. Moore, and J. George, *Mitotic Count and the Field of View Area: Time to Standardize*. *Veterinary Pathology*, 2016. **53**: p. 7-9.
  35. Jung, A.B., et al. *imgaug*. 2020 [cited 2020; 1 Feb 2020]. Available from: <https://github.com/aleju/imgaug>.
  36. He, K., et al. *Deep Residual Learning for Image Recognition*. arXiv e-prints, 2015. arXiv:1512.03385.
  37. Leonardo, M., et al., *Deep Feature-Based Classifiers for Fruit Fly Identification*

- (Diptera: Tephritidae). 2018. 41-47.
38. Lin, T., et al. *Focal Loss for Dense Object Detection*. in *2017 IEEE International Conference on Computer Vision (ICCV)*. 2017.
39. Zlocha, M., Q. Dou, and B. Glocker *Improving RetinaNet for CT Lesion Detection with Dense Masks from Weak RECIST Labels*. arXiv e-prints, 2019. arXiv:1906.02283.





จุฬาลงกรณ์มหาวิทยาลัย  
**CHULALONGKORN UNIVERSITY**

## ประวัติผู้เขียน

ชื่อ-สกุล	ณัทกร เกษมสำราญ
วัน เดือน ปี เกิด	23 มีนาคม พ.ศ.2535
สถานที่เกิด	จังหวัดชลบุรี
วุฒิการศึกษา	วิศวกรรมศาสตรบัณฑิต (วศ.บ.) สาขาวิศวกรรมไฟฟ้า คณะ วิศวกรรมศาสตร์ มหาวิทยาลัยบูรพา
ที่อยู่ปัจจุบัน	เลขที่ 188/85 หมู่ 3 ถนนเสม็ด-อ่างศิลา ตำบลเสม็ด อำเภอเมือง จังหวัด ชลบุรี รหัสไปรษณีย์ 20000
ผลงานตีพิมพ์	1.) N. Kasamsumran, N. Chaimak and N. Wattanamongkhon, "Coral Propagation by High Power LEDs Lamp with Automatic Control System," ICBB&ICCEMB 2016, 8-10 December 2016, Bangkok, Thailand. 2.) N. Kasamsumran, P. Chanchaithong, N. Wattanamongkhon, W. Pora and S. Pumrin, "Applying Faster R-CNN for Hematocytes Detection on Compound Microscope with Image Sensor Device and Multiple GPU Computation," 2020 8th International Electrical Engineering Congress (iEECON), Chiang Mai, Thailand, 2020, pp. 1-4. <a href="https://doi.org/10.1109/iEECON48109.2020.229490">https://doi.org/10.1109/iEECON48109.2020.229490</a> 3.) ณัทกร เกษมสำราญ, ศุทธิณีย์ เศรษฐวานิช, เสฎฐกรรณ์ อุปเสน, และนรรรัตน์ วัฒนมงคล, "The separation method of metal ions from PCB acid etching by NaHCO <sub>3</sub> to reduce the impact on the environment", การประกวดโครงงานวิศวกรรมเคมีและเคมีประยุกต์ระดับปริญญาตรีแห่งประเทศไทยครั้งที่ 5 ประจำปีการศึกษา 2559, 28 เมษายน 2559 4.) ณัทกร เกษมสำราญ, เวียงชัย คาระมาตย์, ปราการ ทัดติยกุล, และนรรรัตน์ วัฒนมงคล, "High Power LED Lamp Phototherapy for Neonatal Hyperbilirubinemia", งานประชุมวิชาการ ECTI-CARD 2017 ครั้งที่ 9 "การประยุกต์ใช้เทคโนโลยี เพื่อตอบสนองท้องถิ่นและภาคอุตสาหกรรม", 25-28 กรกฎาคม 2560

5.) ณิชกร เกษมสำราญ, เวียงชัย คาระมาตย์, และนรรรัตน์ วัฒนมงคล, “Measurement Instrument Data Health Tracking for Exercise by Bicycling,” งานประชุมวิชาการ ECTI-CARD 2017 ครั้งที่ 9 “การประยุกต์ใช้เทคโนโลยี เพื่อตอบสนองท้องถิ่นและภาคอุตสาหกรรม”, 25-28 กรกฎาคม 2560

6.) เวียงชัย คาระมาตย์, ณิชกร เกษมสำราญ, เซาว์นวัฒน์ เอื้อเพื่อ, และนรรรัตน์ วัฒนมงคล, “Measurement Device and Alarm the Quantity of Carbon Monoxide to Reduce the Chance of Death in Vehicle”, งานประชุมวิชาการ ECTI-CARD 2017 ครั้งที่ 9 “การประยุกต์ใช้เทคโนโลยี เพื่อตอบสนองท้องถิ่นและภาคอุตสาหกรรม”, 25-28 กรกฎาคม 2560,

7.) ประภาศรี เบญจศิริลักษณ์, ภูเบศร์ พอดี, ณิชกร เกษมสำราญ, ปราการ ทัดติยกุล, เซาว์นวัฒน์ เอื้อเพื่อ และนรรรัตน์ วัฒนมงคล, “LED Phototherapy Control for Neonatal Hyperbilirubinemia with IoT System,” งานประชุมวิชาการ ECTI-CARD 2018 ครั้งที่ 10 “การประยุกต์ใช้งานเทคโนโลยีเพื่อตอบสนองนโยบายประเทศ 4.0”, 26-29 มิถุนายน 2561

#### รางวัลที่ได้รับ

1.) รางวัล Gold Prize (เหรียญทอง) 43rd International Exhibition of Inventions of Geneva โดยรัฐบาลสวิสฯ (The Swiss Federal Government of the State, the City of Geneva) และองค์การทรัพย์สินทางปัญญาแห่งโลกหรือ WIPO (The World Intellectual Property Organization) วันที่ 15-19 เมษายน พ.ศ.2558 ณ Palexpo นครเจนีวา สมาพันธรัฐสวิส

2.) รางวัล Special Prize : JIPA Award สาขา Best invention of biotechnology มอบโดยตัวแทนจาก Japan Intellectual Property Association (JIPA) แห่งประเทศญี่ปุ่น และรางวัล Special Prize : Honor of Invention มอบโดยตัวแทนจาก Macao Innovation & Invention Association (MIIA) จากเขตบริหารพิเศษมาเก๊าแห่งสาธารณรัฐประชาชนจีน จากงานวันนักประดิษฐ์ ประจำปี 2558 จัดโดยสำนักงานคณะกรรมการวิจัยแห่งชาติ (วช.) ณ ศูนย์แสดงสินค้าและการประชุมอิมแพ็ค เมืองทองธานี

3.) รางวัลชนะเลิศ พร้อมโล่และเงินรางวัล 40,000 บาท จากโครงการ

“รางวัลนักคิดสิ่งประดิษฐ์รุ่นใหม่” ประจำปี 2558 ระดับอุดมศึกษา กลุ่ม  
เรื่องที่ 4 สิ่งประดิษฐ์เพื่อประโยชน์ต่อสิ่งแวดล้อมและพลังงาน ในงานวันนัก  
ประดิษฐ์ ประจำปี 2558 จัดโดยสำนักงานคณะกรรมการวิจัยแห่งชาติ (วช.)  
ณ ศูนย์แสดงสินค้าและการประชุมอิมแพ็ค เมืองทองธานี

พ.ศ.2559 (ค.ศ.2016)

4.) รางวัลรองชนะเลิศอันดับ 1 พร้อมเงินรางวัล 30,000 บาท จากโครงการ  
“รางวัลนักคิดสิ่งประดิษฐ์รุ่นใหม่” ประจำปี 2559 ระดับอุดมศึกษา ในงาน  
วันนักประดิษฐ์ ประจำปี 2559 กลุ่มเรื่องที่ 2 สิ่งประดิษฐ์ด้านวิทยาศาสตร์  
วิทยาศาสตร์การแพทย์ และการสาธารณสุข จัดโดยสำนักงานคณะกรรมการ  
วิจัยแห่งชาติ (วช.) วันที่ 2-6 กุมภาพันธ์ พ.ศ. 2559 ณ ศูนย์นิทรรศการและ  
การประชุมไบเทค (BITEC) บางนา

5.) รางวัลรองชนะเลิศอันดับ 2 ระดับปริญญาตรี พร้อมเงินรางวัล 5,000  
บาท และโล่ประกาศเกียรติคุณ จากงานแสดงผลงานทางวิชาการ และการ  
ประกวดสิ่งประดิษฐ์ทางด้านวิศวกรรมชีวการแพทย์ (Biomedical  
Engineering Innovation 2016) ระหว่างวันที่ 2-3 เมษายน พ.ศ.2559 ณ  
สำนักงานอธิการบดี มหาวิทยาลัยมหิดล วิทยาเขตศาลายา

6.) รางวัล Silver Award ร่วมแสดงผลงานวิจัยของมหาวิทยาลัยบูรพา ใน  
งานมหกรรมวิจัยแห่งชาติ 2559 (Thailand Research Expo 2016)  
ระหว่างวันที่ 17-21 สิงหาคม พ.ศ.2559 ณ โรงแรมเซนทารา แกรนด์  
บางกอกคอนเวนชันเซ็นเตอร์ เซ็นทรัลเวิร์ด และได้รับพระมหากรุณาธิคุณ

อันสูงยิ่ง จากสมเด็จพระเทพรัตนราชสุดาฯ สยามบรมราชกุมารี ทรง  
ทอดพระเนตรให้ความสนพระทัย และมีรับสั่งให้ “พัฒนาโคมโไฟแอลอีดีฯ  
ให้มีประสิทธิภาพจนถึงที่สุดและนำไปแจกจ่ายตามสถานพยาบาลในถิ่น  
ทุรกันดารห่างไกล” จัดโดยสำนักงานคณะกรรมการวิจัยแห่งชาติ (วช.)

7.) รางวัลรองชนะเลิศอันดับ 1 การเขียนข้อเสนอโครงการนวัตกรรมสาย  
อุดมศึกษา ประจำปี 2559 ในการประกวดสิ่งประดิษฐ์และนวัตกรรมสาย  
อุดมศึกษา งานมหกรรมวิจัยแห่งชาติ 2559 (Thailand Research Expo  
2016) ระหว่างวันที่ 17-21 สิงหาคม พ.ศ.2559 ณ โรงแรมเซนทารา แกร  
นด์ บางกอกคอนเวนชันเซ็นเตอร์ เซ็นทรัลเวิร์ด จัดโดยสำนักงาน  
คณะกรรมการวิจัยแห่งชาติ (วช.)

8.) รางวัลกลุ่มเด็กและเยาวชนดีเด่นแห่งชาติ สาขาสิ่งประดิษฐ์และ

นวัตกรรม ประจำปี 2559 จากผู้แทนพระองค์สมเด็จพระบรมโอรสาธิราชฯ สยามมกุฎราชกุมาร ในวันอังคารที่ 20 กันยายน พ.ศ.2559 ณ ห้องรอยัล จูบิลี อาคารชาเลนเจอร์ อิมแพค เมืองทองธานี จังหวัดนนทบุรี จัดโดยกรมกิจการเด็กและเยาวชน กระทรวงพัฒนาสังคมและความมั่นคงของมนุษย์ โดยเป็นผลงานที่ได้รับรางวัลสิ่งประดิษฐ์เพียงผลงานเดียวในจำนวนผลงานทั้งสิ้น 49 ผลงาน

9.) รางวัลนิสิตดีเด่นประจำปี 2560 เข้ารับเข้ารับเกียรติบัตรเนื่องในวันคล้ายวันสถาปนามหาวิทยาลัยบูรพา ครบรอบปีที่ 62 วันเสาร์ที่ 8

กรกฎาคม พ.ศ.2560 ณ หอประชุมธำรง บัวศรี มหาวิทยาลัยบูรพา

10.) รางวัลข้อเสนอด้านอวกาศดีเด่น ระดับเยาวชน ประจำปี 2560

โครงการ Asian Try Zero-G : National Space Experiment หัวข้อ The

water drop orbiting the balls with electrostatic force รับโล่รางวัล

จากรัฐมนตรีกระทรวงวิทยาศาสตร์และเทคโนโลยี ดร.อรรถกา สິบุญเรือง

งานมหกรรมวิทยาศาสตร์และเทคโนโลยีแห่งชาติประจำปี 2560 ณ ศูนย์

แสดงสินค้าและการประชุมอิมแพค เมืองทองธานี วันที่ 26 สิงหาคม

พ.ศ.2560 จัดโดย สำนักงานพัฒนาเทคโนโลยีอวกาศและภูมิสารสนเทศ

(องค์การมหาชน) และสำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ

11.) รางวัล "ทุนเยาวชนคุณภาพแห่งปี 2017" (Quality Youths

Scholarship of The Year 2017) พร้อมใบประกาศเกียรติคุณเพื่อยกย่อง

และเชิดชูเกียรติให้เป็น "เยาวชนคุณภาพแห่งปี 2017" เพื่อสร้างแรงจูงใจให้

เยาวชนรุ่นหลังได้เห็นคุณค่าของตนเองและการทำงานผลงานหรือกิจกรรม

เพื่อสาธารณประโยชน์ ณ ศูนย์ประชุม สถาบันวิจัยจุฬาภรณ์ จัดโดยมูลนิธิ

สภาวิทยาศาสตร์และเทคโนโลยีแห่งประเทศไทย (มสวท.)

กระทรวงวิทยาศาสตร์และเทคโนโลยี วันที่ 1 พฤศจิกายน พ.ศ.2560

12.) รางวัลพระราชทานในโครงการคัดเลือกนักเรียน นักศึกษา และ

สถานศึกษา เพื่อรับรางวัลพระราชทาน จากสมเด็จพระเทพรัตนราชสุดาฯ

สยามบรมราชกุมารี ประจำปีการศึกษา 2559 ณ ศาลาดุสิดาลัย สวน

จิตรลดา วันที่ 6 พฤศจิกายน พ.ศ.2560 โดยรางวัลพระราชทานแก่นักเรียน

นักศึกษา และสถานศึกษาได้จัดขึ้นเพื่อสนองพระราชปราชญ์ของ

พระบาทสมเด็จพระเจ้าอยู่หัว (รัชกาลที่ 9) ซึ่งได้มีกระแสพระราชดำรัสแก่

รัฐมนตรีว่าการกระทรวงศึกษาธิการ (ม.ล.ปิ่น มาลากุล) เมื่อ พ.ศ. 2506 ว่า

กระทรวงศึกษาธิการควรจะพิจารณาจัดรางวัลให้แก่นักเรียน ซึ่งเรียนดี ประพฤติดี และโรงเรียนที่จัดการศึกษาดี

13.) ผ่านเข้ารอบในการแข่งขัน Startup Thailand League ระดับภูมิภาค ได้รับเงินสนับสนุนเพื่อพัฒนาผลิตภัณฑ์ต้นแบบ ทีมละ 100,000 บาท จัดการแข่งขันที่มหาวิทยาลัยวลัยลักษณ์ เมื่อวันที่ 17 สิงหาคม พ.ศ.2560 ที่ เซนทรัล นครศรีธรรมราช มีทีมเข้าร่วมการแข่งขันทั้งหมด 26 ทีม จาก มหาวิทยาลัยบูรพา มหาวิทยาลัยพะเยา มหาวิทยาลัยสงขลานครินทร์ และ มหาวิทยาลัยวลัยลักษณ์ โดยทีมที่ผ่านเข้ารอบ คือ Health BaBy โมเดล ธุรกิจโคมไฟแก้ปัญหาเด็กทารกตัวเหลือง จัดโดยสำนักงานนวัตกรรมแห่งชาติ (NIA)

14.) รางวัลชนะเลิศ ในการแข่งขันการประกวดสิ่งประดิษฐ์และนวัตกรรม ด้าน IoT (Internet of Things) ระดับอุดมศึกษาหรือเทียบเท่า รับเงิน รางวัล 20,000 บาท เหรียญทอง และโล่เกียรติยศจาก รัฐมนตรีกระทรวง ดิจิทัลเพื่อเศรษฐกิจและสังคม ในงานมหกรรมดิจิทัลและเทคโนโลยีระดับ นานาชาติ Digital Thailand Big Bang 2017 วันที่ 21 กันยายน พ.ศ.2560 ณ อาคารชาเลนเจอร์ 1-2 ศูนย์แสดงสินค้าและการประชุมอิมแพค เมืองทองธานี

15.) รางวัลชนะเลิศ สาขาวิทยาศาสตร์และเทคโนโลยี รางวัลนวัตกรรมแห่งประเทศไทย (นวท.) ครั้งที่ 17 รับถ้วยพระราชทาน สมเด็จพระเจ้าพี่นางเธอ เจ้าฟ้ากัลยาณิวัฒนา กรมหลวงนราธิวาสราชนครินทร์ พร้อมทุนการศึกษา 50,000 บาท ภายในงาน i-INNOVATION THAILAND WEEK 2017 ระหว่างวันที่ 5-8 ตุลาคม พ.ศ.2560 ณ ภิรัชฮอลล์ ไบเทคบางนา และ การศึกษาดูงาน ณ สาธารณรัฐประชาชนจีนเป็นเวลา 1 สัปดาห์ โดย สมาคมวิทยาศาสตร์แห่งประเทศไทยฯ กระทรวงวิทยาศาสตร์และเทคโนโลยี สภาสมาคมวิทยาศาสตร์และเทคโนโลยีแห่งประเทศไทย สำนักงานกองทุน สนับสนุนการสร้างเสริมสุขภาพ (สสส.) และสำนักงานนวัตกรรมแห่งชาติ (NIA)

พ.ศ.2561 (ค.ศ.2018)

16.) รางวัลเหรียญทอง และรางวัลรองชนะเลิศอันดับ 2 พร้อมเงินรางวัล 25,000 บาท จากโครงการ “รางวัลนักคิดสิ่งประดิษฐ์รุ่นใหม่” ประจำปี 2561 ระดับอุดมศึกษา ในงานวันนักประดิษฐ์ ประจำปี 2561 กลุ่มเรื่องที่ 4



สิ่งประดิษฐ์เพื่อการแพทย์และสาธารณสุข จัดโดยสำนักงานคณะกรรมการวิจัยแห่งชาติ (วช.) วันที่ 2-6 กุมภาพันธ์ พ.ศ.2561 ณ ศูนย์นิทรรศการและการประชุมไบเทค (BITEC) บางนา

17.) รางวัลที่ 2 หัวข้อพิเศษ โปรแกรมเพื่อการประยุกต์ใช้งานสำหรับสื่อสารระหว่างสรรพสิ่ง (IoT) รับโล่พร้อมเงินรางวัล 40,000 บาท การแข่งขันพัฒนาโปรแกรมคอมพิวเตอร์แห่งประเทศไทย ครั้งที่ 20 (NSC 2018) จัดภายใต้งาน “มหกรรมประกวดเทคโนโลยีสารสนเทศแห่งประเทศไทย ครั้งที่ 17” (The Seventeenth Thailand IT Contest Festival 2018) ระหว่างวันที่ 14 - 16 มีนาคม พ.ศ.2561 เวลา 08.30-19.00 น. ณ ไอส์แลนด์ฮอลล์ ชั้น 3 ศูนย์การค้าแฟชั่นไอส์แลนด์ โดยศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC)

18.) รางวัลชนะเลิศในการแข่งขันการวัดทางไฟฟ้า เรื่อง “การวัดค่าความถูกต้องของฟังก์ชันเจเนอเรเตอร์ด้วยดิจิตอลมัลติมิเตอร์” ในการประชุมวิชาการ งานวิจัย และพัฒนาเชิงประยุกต์ ครั้งที่ 10 (ECTI-CARD 2018) ระหว่างวันที่ 26-29 มิถุนายน พ.ศ.2561 ณ ศูนย์วัฒนธรรมภาคเหนือตอนล่าง วังจันทร์ ริเวอร์วิว จังหวัดพิษณุโลก

19.) รางวัลชนะเลิศ โล่รางวัล เกียรติบัตร พร้อมเงินรางวัล ประเภท Quality Improvement : Innovation งาน Quality Forum 2018 : BDMS Excellent Quality Network ผลงาน เรื่อง “การควบคุมเครื่องส่องไฟแอลอีดีสำหรับรักษาทารกแรกเกิดภาวะตัวเหลืองด้วย ระบบไอโอที” เป็นตัวแทนร่วมระหว่างนักวิจัยจากโรงพยาบาลญาไท 2 และมหาวิทยาลัยบูรพา วันที่ 14 สิงหาคม 2561 ณ โรงพยาบาลกรุงเทพ จ.กรุงเทพฯ

20.) The Winner of Thailand ICT Awards 2108 รางวัลชนะเลิศการประกวดซอฟต์แวร์ดีเด่นแห่งชาติ (TICTA 2018) เมื่อวันที่ 16 สิงหาคม พ.ศ.2561 ได้รับคัดเลือกเป็นตัวแทนประเทศไทยไปแข่งขัน APICTA Awards 2018 ณ เมืองกวางเจา สาธารณรัฐประชาชนจีน จัดโดยสมาคมอุตสาหกรรมเทคโนโลยีสารสนเทศไทย กระทรวงดิจิทัลเพื่อเศรษฐกิจและสังคม และสำนักงานส่งเสริมเศรษฐกิจดิจิทัล (DEPA) ณ ห้องบอลรูม 2 ชั้น 7 โรงแรมโซฟิเทล สุขุมวิท กรุงเทพมหานคร

21.) รางวัลรองชนะเลิศอันดับ 1 การประกวดสิ่งประดิษฐ์นวัตกรรม ชับเคลื่อนไทยสู่ยุค 4.0 ระดับอุดมศึกษา ผลงาน CU-HAPiot

(Chulalongkorn High Agriculture Platform with IoT) เป็นต้นแบบระบบสมองกลฝังตัวไอโอทีแบบ environment box สำหรับการปลูกผักไฮโดรโปนิกส์ จากงานมหกรรมด้านดิจิทัลเทคโนโลยีระดับนานาชาติ Digital Thailand Big Bang 2018 เมื่อวันที่ 21 กันยายน 2561 ณ ชาเลนเจอร์ ฮอลล์ 1-3 อิมแพ็ค เมืองทองธานี โดยสำนักงานส่งเสริมเศรษฐกิจดิจิทัล (DEPA)

พ.ศ.2562 (ค.ศ.2019)

22.) รางวัลเหรียญทอง และรางวัลรองชนะเลิศอันดับ 2 พร้อมเงินรางวัล 50,000 บาท จากโครงการ “รางวัลนักคิดสิ่งประดิษฐ์รุ่นใหม่” ประจำปี 2562 ระดับอุดมศึกษา ในงานวันนักประดิษฐ์ ประจำปี 2562 ผลงาน CU-HAPiot (Chulalongkorn High Agriculture Platform with IoT) กลุ่มเครื่องมืออุปกรณ์อัจฉริยะ ระบบเครื่องกลที่ใช้อิเล็กทรอนิกส์ควบคุม ปัญญาประดิษฐ์ และเทคโนโลยีสมองผลฝังตัว จัดโดยสำนักงานคณะกรรมการวิจัยแห่งชาติ (วช.) วันที่ 2-6 กุมภาพันธ์ พ.ศ.2562 ณ ศูนย์นิทรรศการและการประชุมไบเทค (BITEC) บางนา

23.) โครงการ BagnosisDCN ผ่านการคัดเลือก 1 ใน 10 ผลงาน ได้รับเงินสนับสนุนทีมละ 6,000 บาท และเกียรติบัตร จาก SEIC-Space Experiment Ideas Contest กิจกรรมการแข่งขัน “ข้อเสนองานวิจัยการทดลองในสภาวะไร้แรงโน้มถ่วง” กิจกรรมนำเสนองานวิจัยการทดลองในสภาวะไร้แรงโน้มถ่วง ในงาน Thailand space week 2019 วันที่ 27-28 สิงหาคม พ.ศ.2562 ณ อิมแพคเมืองทองธานี

24.) โครงการ MicrosisDCN โมเดลธุรกิจในกลุ่มอุตสาหกรรมการแพทย์และบริการเพื่อผู้สูงอายุ (Medical & Aging Society) ผ่านเข้ารอบ 12 ทีมสุดท้าย ได้รับทุนพัฒนาผลงาน ทุนละ 30,000 บาท โครงการ Young Technopreneur 2019 จัดโดย กลุ่มสามารถคอร์ปอเรชั่น และศูนย์บ่มเพาะธุรกิจเทคโนโลยี (BIC) สำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (สวทช.)

25.) โครงการชุดกล้องอัจฉริยะ “ไมโครซิสตีซีเอ็น” สำหรับกล้องจุลทรรศน์ ได้รับเงินอุดหนุนการวิจัย สัญญาเลขที่ CU\_GI\_62\_18\_21\_13 ประเภททุนสิ่งประดิษฐ์ กองทุนรัชดาภิเษกสมโภช ปีงบประมาณ 2562 สำนักบริหารวิจัย จุฬาลงกรณ์มหาวิทยาลัย เป็นทุนไม่เกิน 1 ล้านบาท เพื่อส่งเสริมและ

สนับสนุนผู้ที่มีความสามารถในการคิดค้นสิ่งประดิษฐ์ที่เป็นประโยชน์ต่อการใช้งาน การเรียนการสอน ซึ่งสามารถนำไปผลิตเป็นเครื่องมือหรืออุปกรณ์ที่เป็นประโยชน์ต่อสังคม

พ.ศ.2563 (ค.ศ.2020)

26.) ผลงานเรื่อง "ชุดกล้อง "MicrosisDCN" วิเคราะห์จุลชีพด้วยโครงข่ายประสาทสำหรับสวมท่อเลนส์ใกล้ตาของกล้องจุลทรรศน์" ได้รับรางวัลตีพิมพ์ ดาวรางวัลระดับ 5 ดาว รางวัลละ 5,000 บาท พร้อมเกียรติบัตร ในการเขียน concept paper จากการคัดเลือกผลงานที่มีการนำเสนอแนวคิดนวัตกรรมที่โดดเด่น ในกลุ่มการพัฒนาเทคโนโลยีปัญญาประดิษฐ์อุปกรณ์อัจฉริยะ พลังงานและสิ่งแวดล้อม (Smart Devices, Mechatronics, Digital, Artificial Intelligence, Energy and Environment) กิจกรรมการเพิ่มศักยภาพและมาตรฐานบุคลากรอุดมศึกษา : บ่มเพาะและแลกเปลี่ยนเรียนรู้เพื่อพัฒนาสิ่งประดิษฐ์และนวัตกรรม สำนักงานการวิจัยแห่งชาติ (วช.) วันที่ 4 – 6 กุมภาพันธ์ 2563 ณ ห้อง MR214 - MR217 ชั้น 2 ศูนย์นิทรรศการและการประชุมไบเทค บางนา กรุงเทพฯ