บทที่ 4

การทดสอบระบบ

ในโครงงานนี้ใช้หลักการค้นหาหมวกนิรภัยด้วยวิธีการ YOLO ซึ่งขั้นตอนการทำงานหลักมี 2 ขั้นตอนคือ การสร้างโมเดลด้วยคอนโวลูชันนิวรอนเน็ตเวิร์กและการค้นหาหมวกนิรภัย ในบทที่ 4 จะมีหัวข้อดังนี้คือ

1.ชุดข้อมูลภาพเพื่อนำมาสร้างโมเดลและชุดข้อมูลที่ใช้ทดสอบโมเดล

2.ผลการสร้างโมเดลด้วยคอนโวลูชันนิวรอนเน็ตเวิร์กของ YOLO

3.ชุดข้อมูลภาพและวิดีโอที่ใช้ในการทดสอบโปรแกรม

4.ผลการตรวจจับหมวกนิรภัยของโปรแกรม

รายละเอียดแสดงดังหัวข้อต่อไปนี้

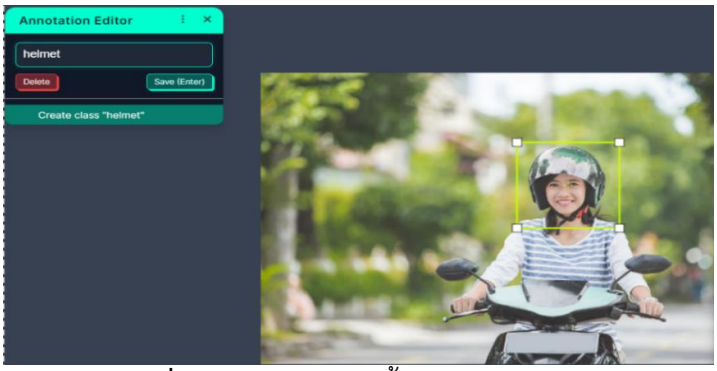
4.1 ข้อมูลภาพเพื่อนำมาสร้างโมเดลและชุดข้อมูลที่ใช้ทดสอบโมเดล

4.1.1 ชุดข้อมูลภาพ

4.1.1.1 ชุดข้อมูลภาพที่จะนำมาทำการสร้างโมเดลโดยจะใช้จำนวน 714 ภาพ ประกอบด้วย 499 ภาพสำหรับเทรน 111 ภาพสำหรับตรวจสอบความถูกต้อง 104 ภาพสำหรับทดสอบโมเดล

4.1.1.2 การเตรียมข้อมูลในการสร้างโมเดลเราได้ทำการเตรียมข้อมูลใน Roboflow ซึ่งใน Roboflow จะให้ตีกรอบวัตถุเพื่อที่จะกำหนดคลาสให้วัตถุที่เราตีกรอบ ดังภาพตัวอย่าง 4.1 และ 4.2

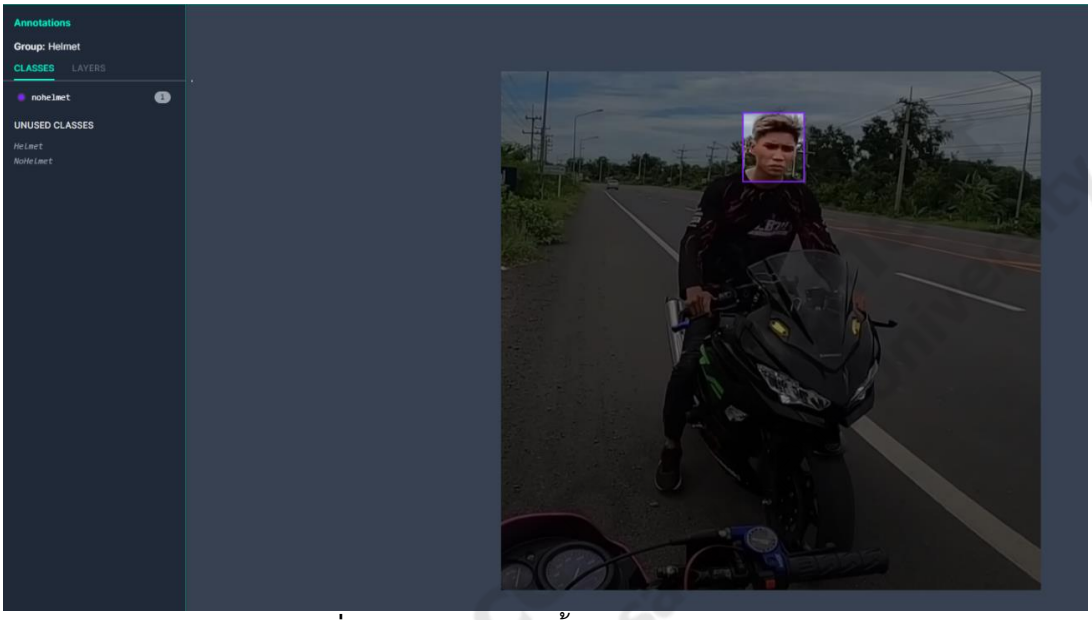
4.1.1.3 จากภาพประกอบที่ 4.1 การเตรียมชุดข้อมูลการเทรน คลาส helmet หรือคลาสสวมหมวกนิรภัย ด้วยการตีกรอบวัตถุแล้วกำหนดคลาส



ภาพประกอบที่ 4.1 การเตรียมชุดข้อมูลการเทรนคลาส helmet

Computer Science Department
Faculty of Information Technology Mahasarakham University

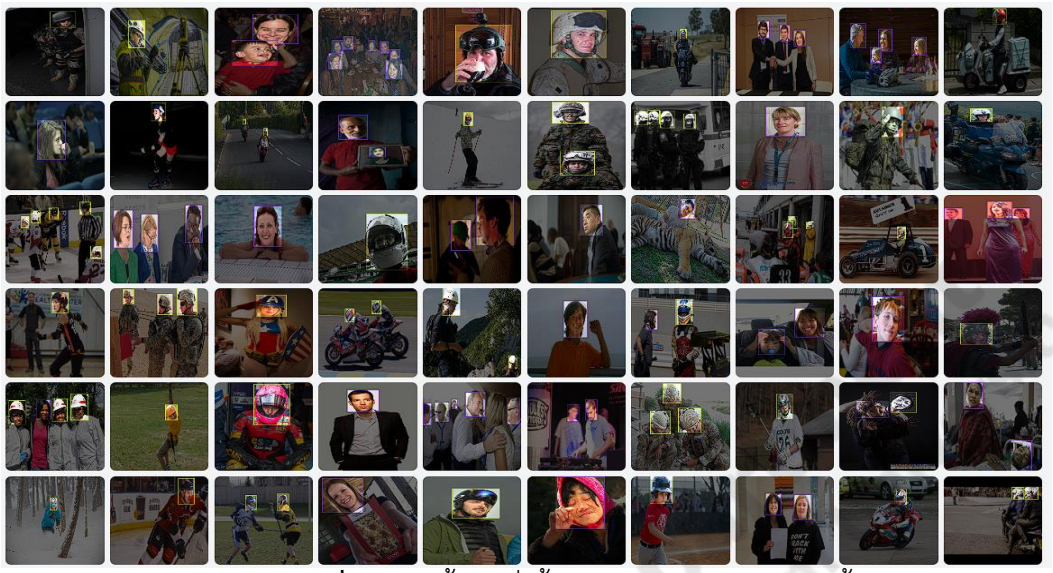
จากภาพประกอบที่ 4.2 การเตรียมชุดข้อมูลการเทรน คลาส nohelmet หรือ คลาสไม่สวมหมวกนิรภัย ด้วยการตีกรอบวัตถุแล้วกำหนดคลาส



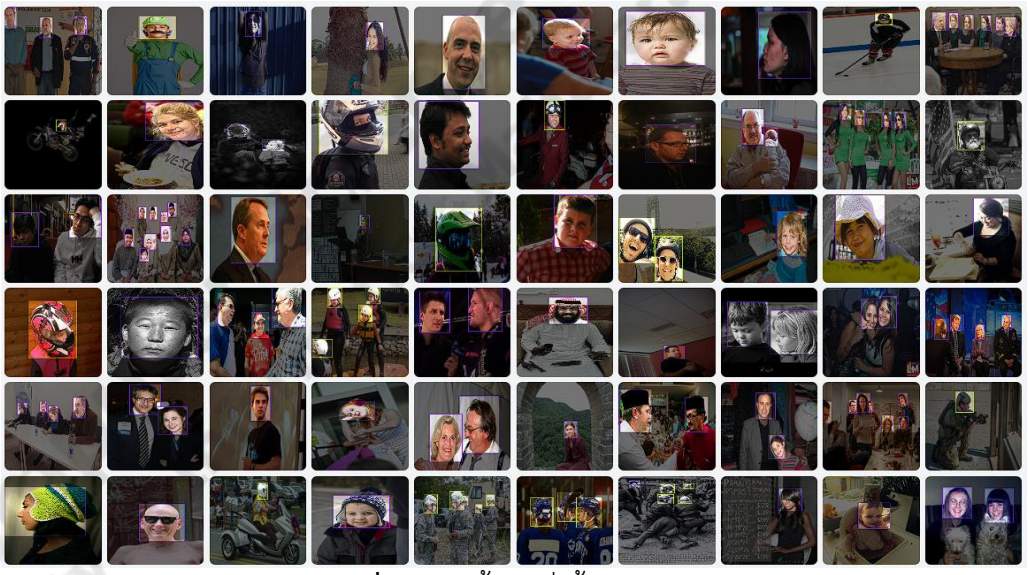
ภาพประกอบที่ 4.2 การเตรียมชุดข้อมูลการเทรนคลาส nohelmet

เมื่อตีกรอบชุดข้อมูลเสร็จจะได้ ชุดข้อมูลสำหรับเทรนดังภาพประกอบที่ 4.3 ชุดข้อมูลสำหรับตรวจสอบความถูกต้องดังภาพประกอบที่ 4.4 และ ชุดข้อมูลสำหรับทดสอบดังภาพประกอบที่ 4.5

ภาพประกอบที่ 4.3 ชุดข้อมูลที่ใช้สำหรับเทรนโมเดล

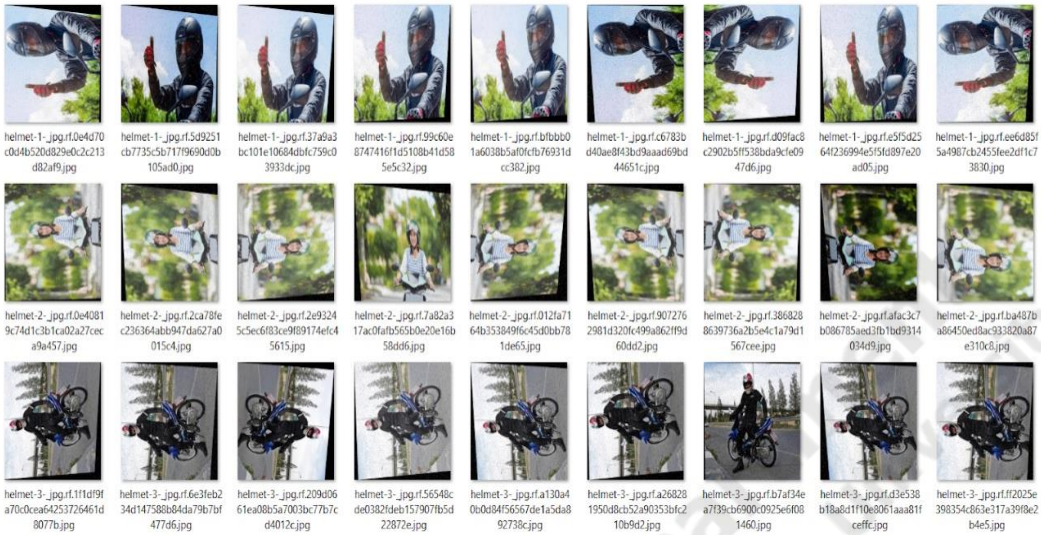


ภาพประกอบที่ 4.4 ชุดข้อมูลที่ใช้สำหรับตรวจสอบความถูกต้อง



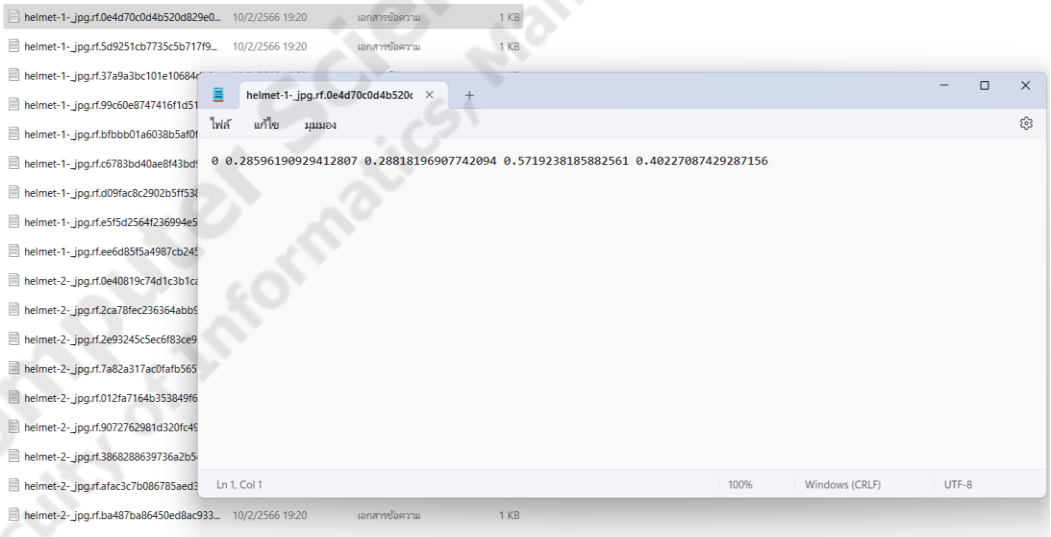
ภาพประกอบที่ 4.5 ชุดข้อมูลที่ใช้สำหรับทดสอบโมเดล

เมื่อเรากำหนดคลาสของชุดข้อมูลเสร็จเรียบร้อยแล้วทำการ Export ข้อมูลใน Project Roboflow ออกมา จะได้ไฟล์ images และ labels ไปใช้ในขั้นตอนสร้างโมเดล แสดงดังภาพประกอบที่ 4.6 และ 4.7 ตามลำดับ



ภาพประกอบที่ 4.6 Images

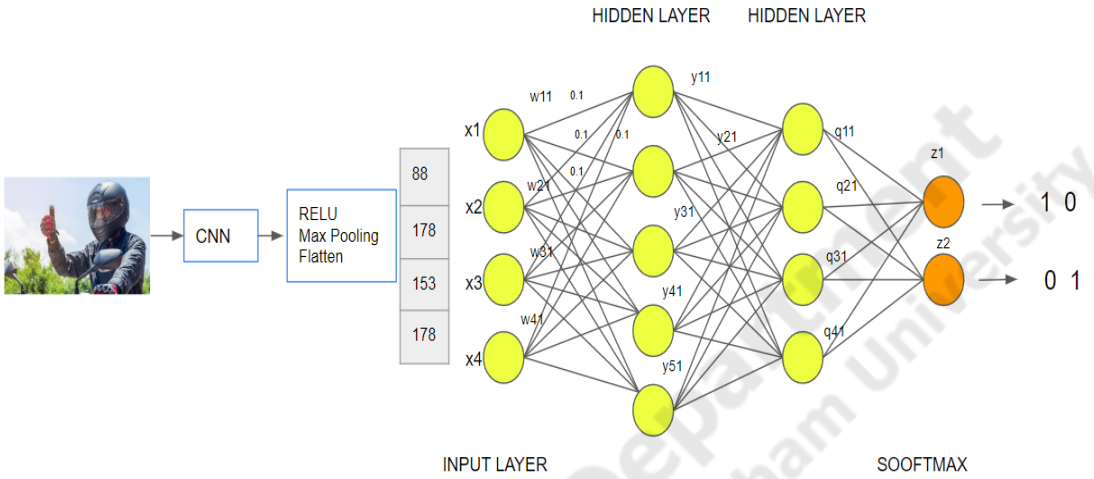
ข้อมูลในไฟล์ labels จะประกอบไปด้วยตัวเลข 5 จำนวนคือ 1.คลาส 2.ตำแหน่ง X 3. ตำแหน่ง Y 4.ความกว้าง W 5.ความสูง H ตามลำดับ แสดงดังภาพประกอบที่ 4.7



ภาพประกอบที่ 4.7 Lables

ขั้นตอนการสร้างโมเดล YOLO (Training)มีขั้นตอนการทำงานที่สำคัญคือ การสกัดคุณลักษณะเด่นของภาพด้วยคอนโวโลชันนิวรอลเน็ตเวริ์ก (Convolution neuron networks:CNNS) และสร้างโมเดลด้วย Fully connected layer: FC หลักการทำงานของ Fully-Connected Layer เป็นชั้นที่เชื่อมโยงระหว่าง Feature Map และ Output แบบสมบูรณ์ นั่น

หมายถึง ทุกๆ Neuron ที่อยู่ในชั้นสุดท้ายของ Feature Mapจะถูกนำไปเปลี่ยนรูป (Reshape) หรือเรียกว่า Flattenเพื่อส่งไปคำนวณต่อในชั้นถัดไปแสดงดังภาพประกอบที่ 4.8



ภาพประกอบที่ 4.8 ขั้นตอนการสร้างโมเดล

4.2 ผลการสร้างโมเดลด้วยคอนโวลูชันนิรอนเน็ตเวิร์ก YOLO

ผลการสร้างโมเดล เราได้ทำการสร้างโมเดล 5 ตัว โดยเริ่มต้นตัวแรกด้วยการเทรนจำนวน 100 รอบแสดงดังตารางที่ 4.1 ตัวที่สองเทรน 200 รอบ แสดงดังตารางที่ 4.2 ตัวที่สามเทรน 300 รอบแสดงดังตารางที่ 4.3 ตัวที่สี่เทรน 400 รอบแสดงดังตารางที่ 4.4 และ ตัวที่ห้าเทรน 500 รอบ แสดงดังตารางที่ 4.5 เพื่อหาโมเดลที่ดีที่สุด โดยสังเกตุจากค่า Precision,Recall,MAP มีค่าที่สูง และเริ่มคงที่แล้ว

P หรือ Precision คือความแน่นอนในการวัดประสิทธิภาพเป็นการใกล้เคียงกันของค่าที่ หาได้จากการทดลองหลายๆ ครั้งในสิ่งของสิ่งเดียวกัน

Recall คือ จำนวนที่ทำนายถูกกี่ตัวเพื่อเป็นการวัดความถูกต้องของโมเดล

MAP เป็นการหาค่าเฉลี่ยของความแม่นยำของ Precision และ Recall

```
100 epochs completed in 0.658 hours.
Optimizer stripped from runs/detect/train2/weights/last.pt, 6.2MB
Optimizer stripped from runs/detect/train2/weights/best.pt, 6.2MB

Validating runs/detect/train2/weights/best.pt...
Ultralytics YOLOv8.0.0 🚀 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)
Fusing layers...
Model summary: 168 layers, 3006038 parameters, 0 gradients, 8.1 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 23/23 [00:10<00:00,  2.23it/s]
                   all        714       1194      0.911      0.832      0.929       0.69
                helmet        714        384      0.905      0.792      0.899      0.677
              nohelmet        714        810      0.917      0.873      0.959      0.703
Speed: 0.1ms pre-process, 1.5ms inference, 0.0ms loss, 1.6ms post-process per image
Saving runs/detect/train2/predictions.json...
Results saved to runs/detect/train2
```

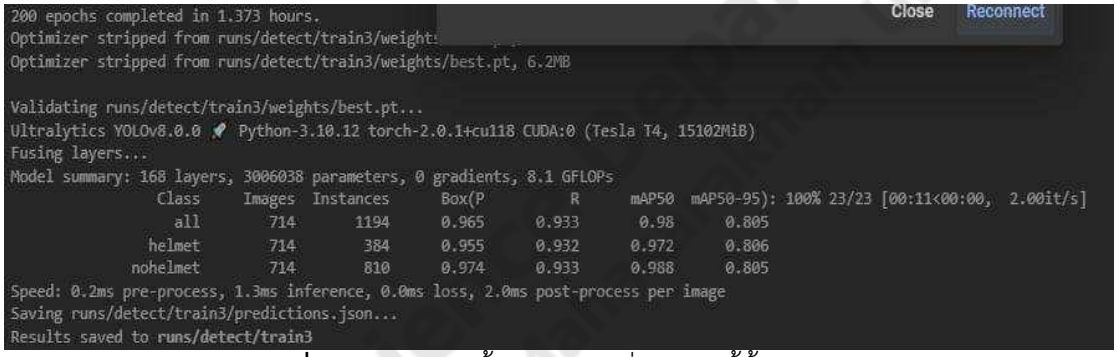
ภาพประกอบที่ 4.9 ผลการสร้างโมเดลตัวที่ 1 เรียนรู้ด้วยจำนวน 100 รอบ

Computer Science Department
Faculty of Informatics, Mahasarakham University

ตารางที่ 4.1 โดยเรียนรู้ด้วยจำนวน 100 รอบ

คลาส	Precision	Recall(R)%	MAP
ทั้งหมด	91.1	83.2	69.0
หมวกนิรภัย	90.5	79.2	67.7
ไม่สวมหมวกนิรภัย	91.7	87.3	70.3

จากการสังเกตุค่า Precision Recall MAP จึงทดลองเพิ่มจำนวนรอบเป็น 200 รอบ เพื่อดูความเปลี่ยนแปลงของค่า Precision Recall MAP ว่ายังสามารถเพิ่มขึ้นได้อีกหรือไม่

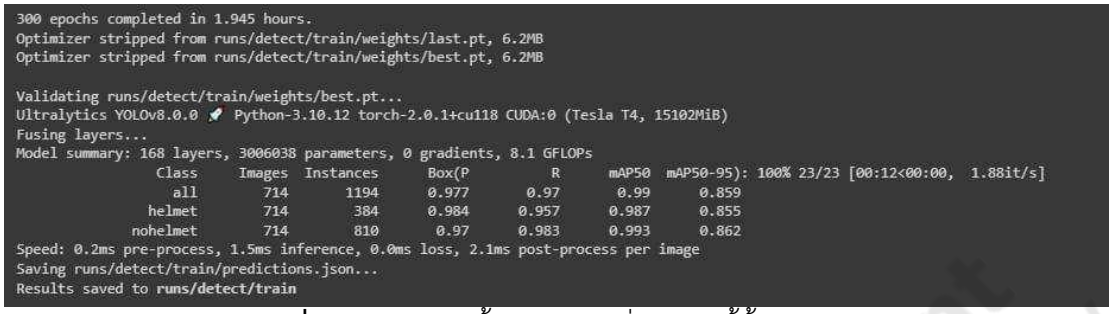


ภาพประกอบที่ 4.10 ผลการสร้างโมเดลตัวที่ 2 เรียนรู้ด้วยจำนวน 200 รอบ

ตารางที่ 4.2 โดยเรียนรู้ด้วยจำนวน 200 รอบ

คลาส	Precision	Recall(R)%	MAP
ทั้งหมด	96.5	93.3	80.5
หมวกนิรภัย	95.5	93.2	80.6
ไม่สวมหมวกนิรภัย	97.4	93.3	80.5

จากการสังเกตุค่า Precision Recall MAP จึงทดลองเพิ่มจำนวนรอบเป็น 300 รอบ เพื่อดูความเปลี่ยนแปลงของค่า Precision Recall MAP ว่ายังสามารถเพิ่มขึ้นได้อีกหรือไม่

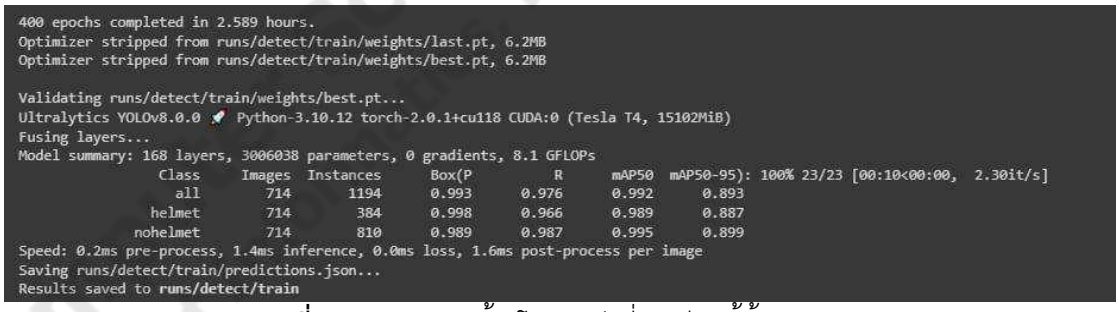


ภาพประกอบที่ 4.11 ผลการสร้างโมเดลตัวที่ 3 เรียนรู้ด้วยจำนวน 300 รอบ

ตารางที่ 4.3 โดยเรียนรู้ด้วยจำนวน 300 รอบ

คลาส	Precision	Recall(R)%	MAP
ทั้งหมด	97.7	97.0	85.9
หมวกนิรภัย	98.4	95.7	85.5
ไม่สวมหมวกนิรภัย	97	98.3	86.2

จากการสังเกตุค่า Precision Recall MAP จึงทดลองเพิ่มจำนวนรอบเป็น 400 รอบ เพื่อดูความเปลี่ยนแปลงของค่า Precision Recall MAP ว่ายังสามารถเพิ่มขึ้นได้อีกหรือไม่



ภาพประกอบที่ 4.12 ผลการสร้างโมเดลตัวที่ 4 เรียนรู้ด้วยจำนวน 400 รอบ

ตารางที่ 4.4 โดยเรียนรู้ด้วยจำนวน 400 รอบ

คลาส	Precision	Recall(R)%	MAP
ทั้งหมด	99.3	97.6	89.3
หมวกนิรภัย	96.6	98.9	88.7
ไม่สวมหมวกนิรภัย	98.7	99.5	89.9

จากการสังเกตุค่า Precision Recall MAP จึงทดลองเพิ่มจำนวนรอบเป็น 500 รอบ เพื่อดูความเปลี่ยนแปลงของค่า Precision Recall MAP ว่ายังสามารถเพิ่มขึ้นได้อีกหรือไม่

Computer Science Department Faculty of Science Mahasarakham University

```
500 epochs completed in 3.013 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 6.2MB
Optimizer stripped from runs/detect/train/weights/best.pt, 6.2MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.0.0 🚀 Python-3.10.12 torch-2.0.1+cu118 CUDA:0 (Tesla T4, 15102MiB)
Fusing layers...
Model summary: 168 layers, 3006038 parameters, 0 gradients, 8.1 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 23/23 [00:09<00:00,
                   all        714       1194      0.995      0.985      0.995      0.915
                helmet        714        384      0.992      0.977      0.994      0.911
              nohelmet        714        810      0.997      0.993      0.995       0.92
Speed: 0.3ms pre-process, 1.3ms inference, 0.0ms loss, 1.4ms post-process per image
Saving runs/detect/train/predictions.json...
Results saved to runs/detect/train
```

ภาพประกอบที่ 4.13 ผลการสร้างโมเดลตัวที่ 5 เรียนรู้ด้วยจำนวน 500 รอบ

ตารางที่ 4.5 โดยเรียนรู้ด้วยจำนวน 500 รอบ

คลาส	Precision	Recall(R)%	MAP
ทั้งหมด	99.5	98.5	91.5
หมวกนิรภัย	99.2	97.7	91.1
ไม่สวมหมวกนิรภัย	99.7	99.3	92.0

จากการสังเกตุค่า Precision,Recall,MAP เป็นค่าที่สูงและคงที่แล้วจึงไม่ทำการทดลองเพิ่มจำนวนรอบต่อ

ตารางที่ 4.6 แสดงผลสรุปการทดสอบโมเดลแต่ละรอบการเทรน

จำนวนรอบการเทรน	Recall(R)%	MAP	Precision	เวลา(ชั่วโมง)
100	83.2	69.0	91.1	0.658
200	93.3	80.5	96.5	1.373
300	97.0	85.9	97.7	1.945
400	99.2	89.3	99.3	2.589
500	98.5	91.5	99.5	3.013

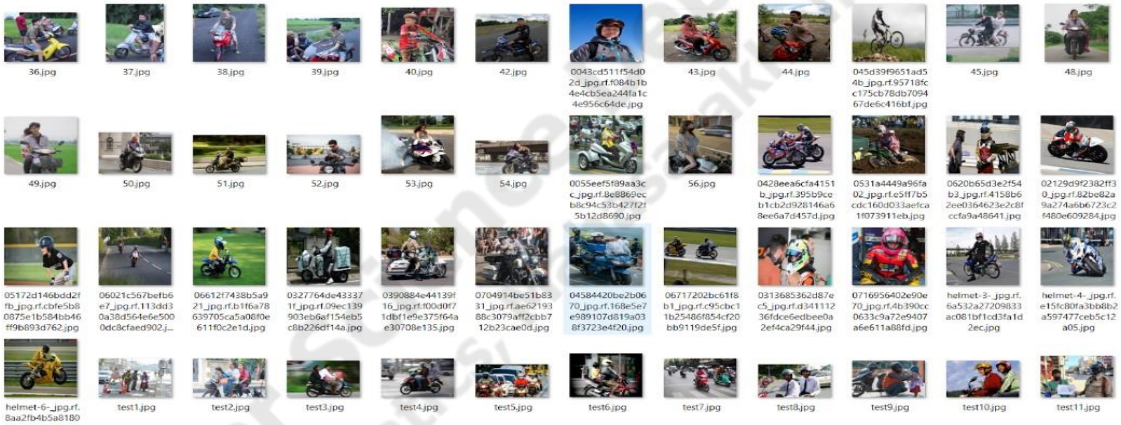
จากตารางที่ 4.6 แสดงค่า Recall ค่า MAP ค่า Precision และเวลาของการเทรนของแต่ละรอบตั้งแต่ 100 รอบ ถึง 500 รอบ ว่ามีประสิทธิภาพและความแตกต่างกันมากน้อยเพียงใดจำนวนรอบการเทรนสูงจะใช้เวลาในการเทรนที่นาน และโมเดลที่เรานำมาใช้เป็นโมเดลที่มีการเรียนรู้ 500 รอบ เพราะมีค่า MAP ,Precision,Recall ที่สูง และคงที่

Computer Science Department
Faculty of Informatics, Mahasarakham University

4.3 ชุดข้อมูลภาพและวิดีโอที่ใช้ในการทดสอบโปรแกรม

ในการทดสอบโปรแกรมเราใช้ ชุดข้อมูลประเภทรูปภาพจำนวน 100 ภาพ เป็นภาพ JPG ซึ่งแต่ละภาพอาจประกอบไปด้วย บุคคลสวมหมวกนิรภัยและไม่สวมหมวกนิรัย และวิดีโอจำนวน 10 วิดีโอ การเก็บข้อมูลวิดีโอ เราทำการบันทึกวิดีโอบริเวณ มหาวิทยาลัยมหาสารคาม วิดีโอละ 1-2 นาที ที่ FPS ของวิดีโอ 30 FPS เนื่องจากการทดสอบวิดีโอที่บันทึกมาด้วย 60 FPS โปรแกรมจะทำงานได้ช้า

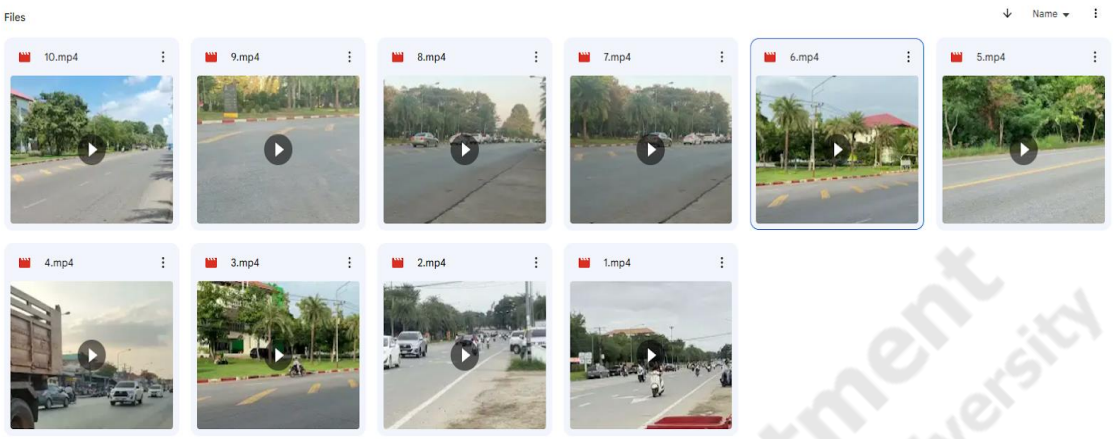
จากภาพประกอบที่ 4.14 ชุดข้อมูลประเภทรูปภาพจำนวน 100 ภาพ เพื่อที่จะนำไปทดสอบโปรแกรม



ภาพประกอบที่ 4.14 ชุดข้อมูลทดสอบประเภทรูปภาพ

จากภาพที่ 4.15 เป็นชุดข้อมูลประเภทวีดีโอจำนวน 10 วีดีโอ ความยาว 1-2 นาที ในแต่ละวีดีโอประกอบด้วย ถนน รถยนต์ รถจักรยานยนต์ หมวกนิรภัย บุคคล เพื่อที่จะนำไปทำการทดสอบกับโปรแกรม

Computer Science Department
Faculty of Informatics Mahasarakham University



ภาพประกอบที่ 4.15 ชุดข้อมูลทดสอบประเภทวิดีโอ

4.4 ผลการตรวจจับหมวกนิรภัยของโปรแกรม

ในการวัดประสิทธิภาพการค้นหาหมวกนิรภัยของโปรแกรม ดังสมการ4.1,4.2,4.3,4.4 ตามลำดับ

4.4.1 Accuracy ความถูกต้อง หรือ ความแม่นยำ (accuracy) เป็นค่าที่บ่งบอกถึงความสามารถของเครื่องมือวัด (instrument) ในการอ่านค่าหรือแสดงค่าที่วัดได้เข้าใกล้ค่าจริง

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{(\text{TP} + \text{TN} + \text{FN} + \text{FP})} \quad 4.1$$

4.4.2 Precision ค่า precision คือความแน่นอนในการวัด เป็นการใกล้เคียงกันของค่าที่ หาได้จากการทดลองหลาย ๆ ครั้งในสิ่งของสิ่งเดียวกัน

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \quad 4.2$$

4.4.3 Recall ค่าความระลึก คือ จำนวนที่ทำนายถูกกี่ตัวเป็นการวัดความถูกต้องของโมเดล

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \quad 4.3$$

4.4.4 F1 Score คือค่าที่แสดงประสิทธิภาพโดยการนำค่าPrecision และ Recall มาคำนวณหาค่าเฉลี่ย

$$\text{F1} - \text{Score} = \frac{2(\text{Precision x Recall})}{(\text{Precision} + \text{Recall})} \quad 4.4$$

Computer Science Department
Faculty of Informatics, Mahasarakham University

4.4.5 mAP (mean Average Precision) คือหนึ่งในวัตถุประสงค์สำคัญในการประเมินประสิทธิภาพของระบบการตรวจจับวัตถุ (Object Detection) โดยเฉพาะในงาน Computer Vision และการเรียนรู้เชิงลึก (Deep Learning) ที่ใช้ในการตรวจจับวัตถุในภาพหรือวิดีโอ การวัดค่า mAP ช่วยในการประเมินความแม่นยำของระบบในการตรวจจับวัตถุว่ามีประสิทธิภาพอย่างไรบนชุดข้อมูลทดสอบ (Test dataset) โดยรวมค่าความแม่นยำที่ได้จากการตรวจจับทุกวัตถุในชุดข้อมูลและคำนวณค่าเฉลี่ย (average) ของค่า Precision สำหรับแต่ละวัตถุที่ตรวจจับได้

สูตรการคำนวนmAP

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP$$

โดยที่ AP คือ ค่าความแม่นยำเฉลี่ย

N คือ จำนวนของ Object

mAP คือ ตัวประเมินประสิทธิภาพความแม่นยำของระบบตรวจจับวัตถุ

โดยที่ TP คือ True Positive ผลเฉลยว่าใส่หมวกนิรภัยโปรแกรมตอบว่าใส่หมวกนิรภัย

TN คือ True Negative ผลเฉลยว่าไม่ใส่หมวกนิรภัยโปรแกรมตอบว่าไม่ใส่หมวกนิรภัย

FP คือ Fales Negative ผลเฉลยว่าใส่หมวกนิรภัยแต่โปรแกรมตอบว่าไม่ใส่หมวกนิรภัย

FN คือ False Positive ผลเฉลยว่าไม่ใส่หมวกนิรภัยแต่โปรแกรมตอบว่าใส่หมวกนิรภัย

4.4.6 วัดประสิทธิภาพข้อมูลประเภทภาพนิ่งทดลองชุดข้อมูลภาพทดสอบโปรแกรมจำนวน 100 ภาพ โดยมีผู้สวมหมวกนิรภัยทั้งหมด 83 คน ไม่สวมหมวกนิรภัยทั้งหมด 79 คน สามารถแสดงการค้นหาผู้สวมหมวกและผู้ที่ไม่สวมหมวกนิรภัยดังตารางที่ 4.7 ใน Confusion matrix images

ตารางที่ 4.7 Confusion matrix images

	หมวกนิรภัย	**ไม่ใส่หมวกนิรภัย**
หมวกกันนิรภัย	79	4
ไม่ใส่หมวกนิรภัย	3	76

จากตารางที่ 4.7 พบว่าผู้ใส่หมวกนิรภัยทั้งหมด 83คน โปรแกรมถูกต้อง 79 ไม่สวมหมวกนิรภัยทั้งหมด 79 โปรแกรมถูกต้อง 76

Computer Science Department Faculty of Informatics, Mahasarakham University

จากตาราง Confusion matrix images ที่ 4.7 นำมาคำนวณหาค่าความถูกต้องของ Accuracy , Precision, Recall, F1-Score แทนค่า TP= 79 TN = 76 FP = 4 FN = 3 คำนวณได้ดังนี้

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

$$\text{Accuracy} = \frac{79 + 76}{(79 + 76 + 3 + 4)}$$

$$\text{Accuracy} = 0.956$$

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

$$\text{Precision} = \frac{79}{(79 + 4)}$$

$$\text{Precision} = 0.951$$

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

$$\text{Recall} = \frac{79}{(79 + 3)}$$

$$\text{Recall} = 0.963$$

$$\text{F1} - \text{Score} = \frac{2(\text{Precision x Recall})}{(\text{Precision} + \text{Recall})}$$

$$\text{F1} - \text{Score} = \frac{2(0.951\ x\ 0.963)}{(0.951 + 0.963)}$$

$$\text{F1} - \text{Score} \approx 0.957$$

การคำนวนหาค่า mAP นำค่า helmetAP กับ nohelmetAP จากตำแหน่งที่ค้นหาผู้ที่สวมหมวกนิรภัยและผู้ที่ไม่ใส่หมวกนิรภัยมาคำนวณหาคำตอบ ดังนี้

mAP (Mean Average Precision)

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP$$

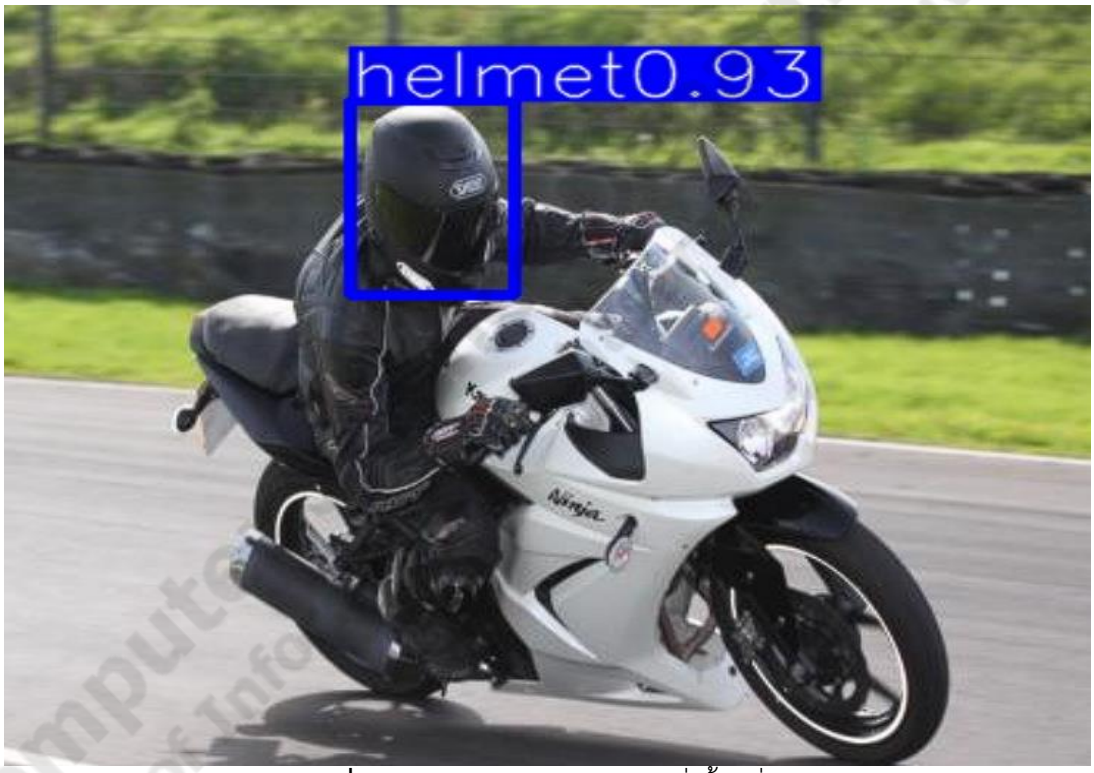
Computer Science Department
Faculty of Informatics, Mahasarakham University

การคำนวณหาค่า helmetAP จะทำการเอา AP(Average Precision) และ จำนวนภาพที่สวมหมวกนิรภัย (N = 79) มาหาค่าเฉลี่ยเพื่อที่จะนำไปคำนวณหาค่า mAP

วิธีการคำนวณ helmetAP

$$\text{helmetAP} = \frac{(0.93 + 0.91 + 0.94 + 0.97 + 0.76 + \cdots + 0.76)}{97}$$

$$\text{helmetAP} = 0.78$$



ภาพประกอบที่ 4.16 ภาพทดสอบโปรแกรมที่ใช้หาค่า helmetAP

การคำนวณหาค่า nohelmetAP จะทำการเอา AP(Average Precision) และ จำนวนภาพที่ไม่สวมหมวกนิรภัย (N = 76) มาหาค่าเฉลี่ยเพื่อที่จะนำไปคำนวณหาค่า mAP

วิธีการคำนวณ nohelmetAP

$$\text{nohelmetAP} = \frac{(0.85 + 0.88 + 0.90 + 0.89 + 0.80 + \cdots + 0.81)}{76}$$

$$\text{nohelmetAP} = 0.86$$

ภาพประกอบที่ 4.17 ภาพทดสอบโปรแกรมที่ใช้หาค่า nohelmetAP

การคำนวณหาค่า mAP

$$\text{mAP} = \frac{(helmetAP + nohelmetAP)}{2}$$

$$\text{mAP} = \frac{(0.78 + 0.86)}{2}$$

$$\text{mAP} = 0.82$$

ตารางที่ 4.8 การวัดประสิทธิภาพด้วยค่าความถูกต้องของชุดข้อมูลทดสอบโปรแกรมประเภทรูปภาพ

วิธีการวัดประสิทธิภาพ	ค่าที่ได้
Accuracy	0.956
Precision	0.951
Recall	0.963
F1-score	0.957
mAP	0.820

จากตารางที่ 4.8 พบว่าการวัดประสิทธิภาพมีความถูกต้องที่สูงแสดงว่าโมเดลที่โปรแกรมใช้ทำงานได้ดี มีประสิทธิภาพแสดงดังภาพประกอบที่ 4.18 ภาพประกอบ 4.19 ภาพประกอบ 4.20 แต่ยังมีข้อผิดพลาดดังภาพประกอบที่ 4.21

ตัวอย่างการทดสอบโปรแกรมด้วยภาพนิ่ง โปรแกรมตรวจจับได้ถูกต้อง

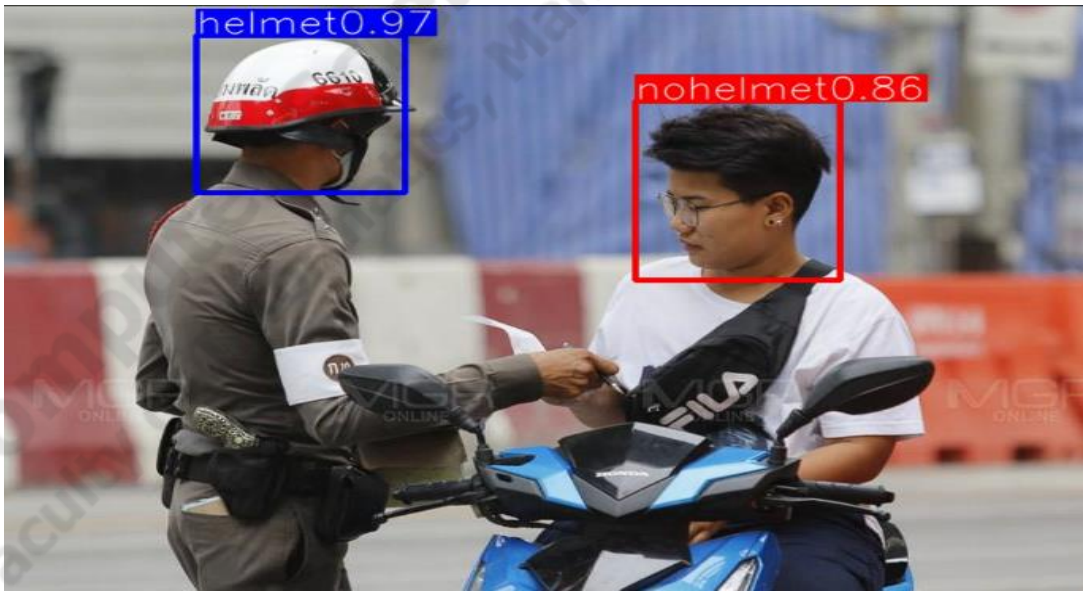
(ก)

(ข)

ภาพประกอบที่ 4.18 การทดสอบภาพนิ่ง (ก) ภาพทดสอบ (ข) ผลลัพธ์การค้นหา 1

จากภาพประกอบที่ 4.18 มีผู้ไม่สวมหมวกนิรภัย 2 คน สวมหมวกนิรภัย 2 คน พบว่าโปรแกรมสามารถตรวจจับได้ถูกต้องทั้งหมด

(ก)



(ข)

ภาพประกอบที่ 4.19 การทดสอบภาพนิ่ง (ก) ภาพทดสอบ (ข) ผลลัพธ์การค้นหา 2

จากภาพประกอบที่ 4.19 มีผู้ไม่สวมหมวกนิรภัย 1 คน สวมหมวกนิรภัย 1 คน พบว่าโปรแกรมสามารถตรวจจับได้ถูกต้องทั้งหมด

(ก)

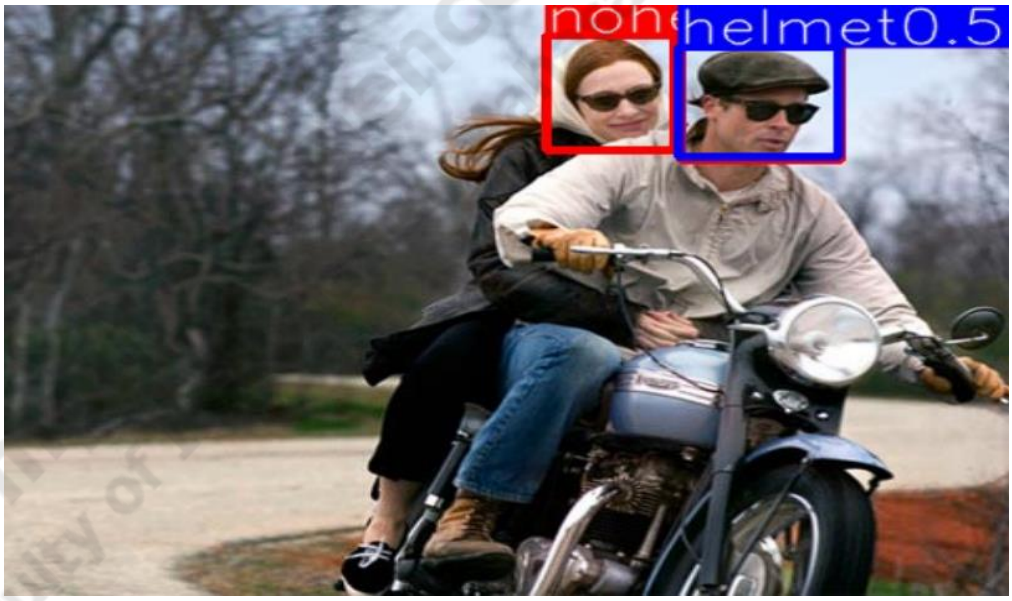
(ข)

ภาพประกอบที่ 4.20 การทดสอบภาพนิ่ง (ก) ภาพทดสอบ (ข) ผลลัพธ์การค้นหา 3

จากภาพประกอบที่ 4.20 มีผู้ไม่สวมหมวกนิรภัย 0 คน สวมหมวกนิรภัย 2 คน พบว่าโปรแกรมสามารถตรวจจับได้ถูกต้องทั้งหมด

ตัวอย่างการทดสอบโปรแกรมด้วยภาพนิ่ง โปรแกรมตรวจจับได้ผิดพลาด

(ก)



(ข)

ภาพประกอบที่ 4.21 การทดสอบภาพนิ่ง (ก) ภาพทดสอบ (ข) ผลลัพธ์การค้นหา 4

จากภาพประกอบที่ 4.21 มีผู้ไม่สวมหมวกนิรภัย 2 คน สวมหมวกนิรภัย 0 คน พบว่าโปรแกรมตรวจจับผิดพลาด

(ก)

(ข)

ภาพประกอบที่ 4.22 การทดสอบภาพนิ่ง (ก) ภาพทดสอบ (ข) ผลลัพธ์การค้นหา 5

จากภาพประกอบที่ 4.21 และภาพประกอบที่ 4.22 พบว่า โปรแกรมไม่สามารถตรวจจับได้ถูกต้องทั้งหมด เนื่องจาก ตัวบุคคลได้มีการสวมหมวกที่ไม่ใช่หมวกนิรภัยไว้ที่ศีรษะจะเห็นได้ว่า มีข้อผิดพลาดในการตรวจจับ ซึ่งเป็นเพราะ หลักการทำงานของ Yolo คือการแบ่งรูปภาพออกเป็นตารางหลายๆตาราง และในแต่ละตารางจะมี Anchor อยู่และนำแต่ละ Anchor ไปทำการสกัด Features ด้วย CNN และเข้าโมเดล เพื่อทำการหาว่าวัตถุที่อยู่ในกรอบเป็นคลาสใด และ มีค่า Precision เท่าใด จากภาพประกอบที่ 4.21 และ ภาพประกอบที่ 4.22 โมเดล ตอบออกมาว่าเป็นคลาส helmet และเมื่อนำวัตถุที่อยู่ในกรอบ ไปเปรียบเทียบกับผลเฉย แล้ว มีค่า IOU ที่สูง โมเดลจึง

ตอบออกมาว่าเป็นคลาส helmet เมื่อตรวจจับคนที่สวมวัตถุอื่นไว้ที่หัวโอกาสที่จะตรวจจับผิดพลาดมีมากถึง 50%

4.4.7 วัดประสิทธิภาพข้อมูลประเภทวิดีโอ

ทดลองชุดข้อมูลภาพวิดีโอทดสอบโปรแกรมจำนวน 10 วิดีโอ วิดีโอละ 1-2 นาที โดยมีผู้สวมหมวกนิรภัยทั้งหมด 292 คน ไม่สวมหมวกนิรภัยทั้งหมด 633 คน สามารถแสดงการค้นหาผู้สวมหมวกและผู้ที่ไม่สวมหมวกนิรภัยดังตารางที่ 4.9 ใน Confusion matrix videos

ตารางที่ 4.9 Confusion matrix videos

	หมวกนิรภัย	**ไม่ใส่หมวกนิรภัย**
หมวกกันนิรภัย	200	92
ไม่ใส่หมวกนิรภัย	44	589

จากตารางที่ 4.9 พบว่าผู้ใส่หมวกนิรภัยทั้งหมด 292 คน โปรแกรมถูกต้อง 200 ไม่สวมหมวกนิรภัยทั้งหมด 633 โปรแกรมถูกต้อง 589 จากตาราง Confusion matrix videos ที่ 4.9 นำมาคำนวณหาค่าความถูกต้องของ Accuracy ,Precision,Recall,F1-Score แทนค่า TP= 200 TN = 589 FP = 92 FN = 44 และคำนวณได้ดังนี้

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

$$\text{Accuracy} = \frac{598 + 200}{(200 + 598 + 92 + 44)}$$

$$\text{Accuracy} = 0.854$$

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

$$\text{Precision} = \frac{200}{(200 + 94)}$$

$$\text{Precision} = 0.684$$

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

$$\text{Recall} = \frac{200 + 44}{(200 + 44)}$$

Computer Science Department
Faculty of Informatics, Mahasarakham University

$$\text{Recall} = 0.819$$

$$\text{F1} - \text{Score} = \frac{2(Precision\ x\ Recall)}{(Precision + Recall)}$$

$$\text{F1} - \text{Score} = \frac{2(0.684x\ 0.819)}{(0.684 + 0.819)}$$

$$\text{F1} - \text{Score} \approx 0.745$$

วิธีการคำนวณหาค่า mAP (Mean Average Precision)

สมการ mAP

$$mAP = \frac{1}{N}\sum_{i=1}^{N} AP$$

$$\text{helmetAP} = \frac{(0.34 + 0.76 + 0.45 + 0.74 + 0.73 + \cdots + 0.86)}{(292)}$$

$$\text{helmetAP} \approx 0.700$$

$$\text{nohelmetAP} = \frac{(0.83 + 0.27 + 0.83 + 0.88 + 0.73 + \cdots + 0.46)}{(633)}$$

$$\text{nohelmetAP} \approx 0.700$$

$$\text{mAP} = \frac{(\text{helmetAP} + \text{nohelmetAP})}{2}$$

$$\text{mAP} = \frac{(0.70 + 0.70)}{2}$$

$$\text{mAP} \approx 0.700$$

ตารางที่ 4.10 การวัดประสิทธิภาพด้วยค่าความถูกต้องของชุดข้อมูลทดสอบโปรแกรมประเภทวิดีโอ

วิธีการวัดประสิทธิภาพ	ค่าที่ได้
Accuracy	0.854
precision	0.684
Recall	0.819
F1-score	0.745
mAP	0.700

Computer Science Department
Faculty of Informatics, Mahasarakham University

จากตาราง 4.10 พบว่า การวัดประสิทธิภาพมีความถูกต้องที่พอใช้ได้แสดงว่าโมเดลที่โปรแกรมใช้ทำงานได้ ชุดข้อมูลทดสอบประเภทวิดีโอมีประสิทธิภาพพอใช้แสดงดัง ภาพประกอบที่ 4.23 ภาพประกอบที่ 4.24 ภาพประกอบที่ 4.25 และยังมีข้อผิดพลาดดังภาพประกอบที่ 4.26

(ก)

(ข)

ภาพประกอบที่ 4.23 การนำวีดีโอมาทดสอบ (ก) ภาพทดสอบ (ข) ผลลัพธ์การค้นหา 1

จากภาพประกอบที่ 4.23 มีผู้ไม่สวมหมวกนิรภัย 3 คน สวมหมวกนิรภัย 0 คน พบว่าโปรแกรมไม่สามารถตรวจจับได้ถูกต้องทั้งหมด ยกตัวอย่างบุคคลในวงกลมสีเหลืองไม่มีการตรวจจับเนื่องจาก กรณีที่หนึ่ง การทำงานของ YOLO แบ่งรูปภาพออกเป็นกริด แล้ว ทำการกำหนด Anchor ในแต่ละกริด จากนั้นนำไปสกัด Features ด้วย CNN และเข้าโมเดล ซึ่งวัตถุที่อยู่ในกริดอาจไม่ได้อยู่

ในกรอบ Anchor จึงไม่ถูกนำไปคำนวณ กรณีที่สอง วัตถุอยู่ในกรอบ Anchor และนำแต่ละกรอบไปทำการสกัด Features ด้วย CNN แล้วเข้าโมเดล เพื่อทำการหาว่าวัตถุที่อยู่ในกรอบเป็นคลาสใด หลังจากนั้น ทำการหาค่า IOU โดยเปรียบเทียบวัตถุที่อยู่ในกรอบกับผลเฉลย เมื่อค่า IOU ต่ำกว่า 0.50 จะไม่ทำการคำนวณต่อ แสดงว่าวัตถุที่อยู่ในวงกลมสีเหลือง มีค่า Precision น้อยกว่า 0.50

(ก)

(ข)

ภาพประกอบที่ 4.24 การนำวีดีโอมาทดสอบ (ก) ภาพทดสอบ (ข) ผลลัพธ์การค้นหา 2

จากภาพประกอบที่ 4.24 มีผู้ไม่สวมหมวกนิรภัย 4 คน สวมหมวกนิรภัย 1 คน พบว่าโปรแกรมสามารถตรวจจับได้ถูกต้องทั้งหมด

(ก)

(ข)

ภาพประกอบที่ 4.25 การนำวีดีโอมาทดสอบ (ก) วีดีโอทดสอบ (ข) ผลลัพธ์การค้นหา 3

จากภาพประกอบที่ 4.25 มีผู้ไม่สวมหมวกนิรภัย 2 คน สวมหมวกนิรภัย 0 คน พบว่าโปรแกรมสามารถตรวจจับได้ถูกต้องทั้งหมด

ตัวอย่างการทดสอบโปรแกรมด้วยภาพวิดีโอ โปรแกรมตรวจจับได้ผิดพลาด

(ก)

(ข)

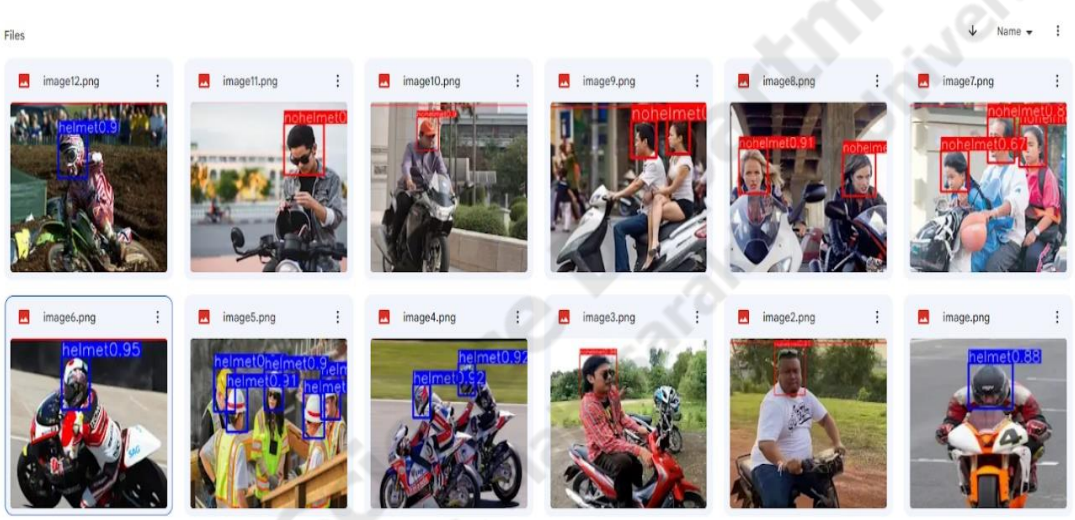
ภาพประกอบที่ 4.26 การนำวีดีโอมาทดสอบ (ก) วีดีโอทดสอบ (ข) ผลลัพธ์การค้นหา 4

จากภาพประกอบที่ 4.26 มีผู้ไม่สวมหมวกนิรภัย 0 คน สวมหมวกนิรภัย 1 คน พบว่าโปรแกรมไม่สามารถตรวจจับได้ถูกต้อง เนื่องจาก การทำงานของ YOLO คือการแบ่งรูปภาพออกเป็นตารางหลายๆตาราง และในแต่ละตารางจะมี Anchor อยู่จากนั้นนำแต่ละ Anchor ไปทำการสกัด Features ด้วย CNN และเข้าโมเดล เพื่อทำการหาว่าวัตถุที่อยู่ในกรอบเป็นคลาสใด และ มีค่า

Precision เท่าใด จากภาพประกอบที่ 4.26 โมเดล ตอบออกมาว่าเป็นคลาส nohelmet และ เมื่อนำวัตถุที่อยู่ในกรอบ ไปเปรียบเทียบกับผลเฉย แล้ว มีค่า IOU ที่สูง โมเดลจึงตอบออกมาว่าเป็นคลาส nohelmet

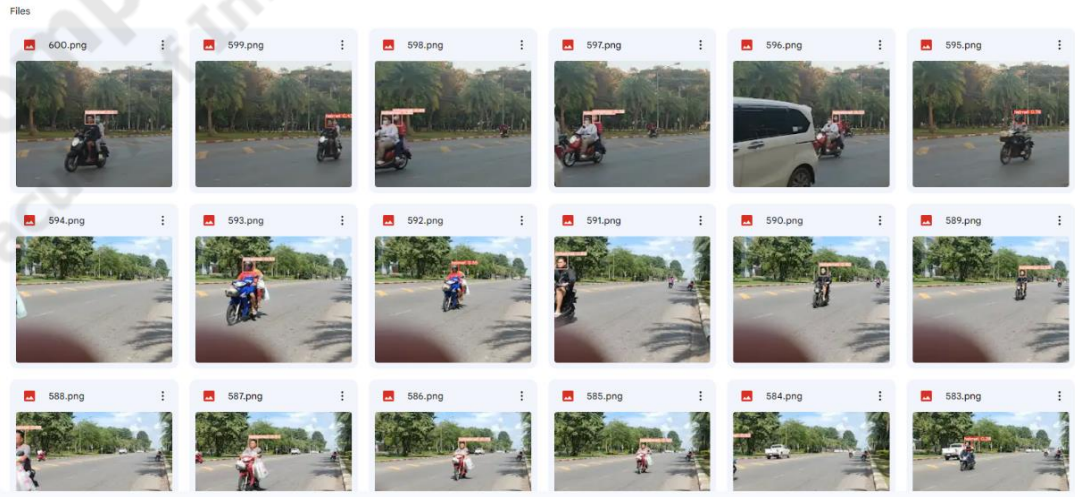
4.4.8 สรุปผลการทดสอบโปรแกรมด้วยภาพนิ่งจำนวน 100 ภาพ และ วิดีโอจำนวน 10 วิดีโอ

ผลการทดสอบโปรแกรมด้วยภาพนิ่งจำนวน 100 ภาพ



ภาพประกอบที่ 4.27 ผลการทดสอบโปรแกรมภาพนิ่ง

ผลการทดสอบโปรแกรมด้วยภาพวิดีโอจำนวน 10 วิดีโอ



ภาพประกอบที่ 4.28 ผลการทดสอบโปรแกรมภาพวิดีโอ

ตารางที่ 4.11 ผลสรุปการวัดประสิทธิภาพด้วยค่าความถูกต้องโปรแกรม

ประเภทข้อมูลที่ทดสอบ	Accuracy	Precision	Recall	mAP	F1-Score
ภาพ	95.6%	95.1%	96.3%	82.0%	95.7%
วีดีโอ	85.4%	68.4%	81.9%	70.0%	74.5%

สังเกตุจากค่า Accuracy,Precision,Recall,mAP,F1-Score ของการทดสอบโปรแกรมด้วยชุดข้อมูลประเภทรูปภาพ มีค่าที่สูงกว่า การทดสอบโปรแกรมด้วยชุดข้อมูลประเภทวิดีโอ หมายความว่าโปรแกรมตรวจจับข้อมูลประเภทรูปภาพได้ดีกว่าประเภทวิดีโ

Computer Science Department
Faculty of Informatics, Mahasarakham University