Safety Helmet Detection Neural Network Report

รายงานระบบตรวจจับหมวกนิรภัยในเขตก่อสร้างด้วย Deep Learning

(Deep Learning การเรียนรู้เชิงลึก)

รหัสวิชา 01204466

จัดทำโดย

นางสาวบุณยาพร บุญทับ 6610505454

นางสาวไข่มุกข์ ธานะกิจ 6610505292

คณะวิศวกรรมศาสตร์ สาขาวิศวกรรมคอมพิวเตอร์

อาจารย์ผู้สอน

อาจารย์ ภารุจ รัตนวรพันธุ์

มหาวิทยาลัยเกษตรศาสตร์

ภาคต้น ปีการศึกษา 2568

1. บทนำ (Introduction)

ในพื้นที่ก่อสร้างและโรงงานอุตสาหกรรม การสวมหมวกนิรภัยถือเป็นข้อกำหนดด้านความปลอดภัยที่สำคัญที่สุดอย่างหนึ่ง เนื่องจากหมวกนิรภัยสามารถลดความรุนแรงของการบาดเจ็บจากการกระแทกหรือวัตถุตกใส่ได้อย่างมีประสิทธิภาพ อย่างไรก็ตามในทางปฏิบัติกลับพบว่าแรงงานจำนวนไม่น้อยละเลยต่อการสวมหมวกนิรภัย โดยเฉพาะในพื้นที่ที่มีการทำงานต่อเนื่องเป็นเวลานานหรือในสภาวะแวดล้อมที่มีอุณหภูมิสูง ส่งผลให้เกิดอุบัติเหตุและการบาดเจ็บที่อาจหลีกเลี่ยงได้หากมีการปฏิบัติตามมาตรการด้านความปลอดภัยอย่างเคร่งครัด

ด้วยเหตุนี้การพัฒนาเทคโนโลยีที่สามารถตรวจจับการสวมหมวกนิรภัยโดยอัตโนมัติจากภาพกล้องจึงเป็นแนวทางที่ได้รับความสนใจอย่างมาก ในช่วงไม่กี่ปีที่ผ่านมาเทคโนโลยีการมองเห็นของคอมพิวเตอร์ (Computer Vision) ร่วมกับเทคนิคการเรียนรู้เชิงลึก (Deep Learning) ได้แสดงให้เห็นถึงศักยภาพในการประมวลผลภาพและจำแนกรูปแบบวัตถุได้อย่างแม่นยำ สามารถระบุได้ว่าในภาพมีบุคคลที่สวมหมวกนิรภัยหรือไม่ โดยไม่จำเป็นต้องพึ่งการตรวจสอบจากมนุษย์โดยตรง

โครงงานนี้มีวัตถุประสงค์เพื่อพัฒนา โมเดล Deep Learning สำหรับการตรวจจับหมวกนิรภัย (Safety Helmet Detection) ในภาพถ่ายหรือวิดีโอจากเขตก่อสร้างแบบอัตโนมัติ โดยใช้สถาปัตยกรรมโครงข่ายประสาทเทียมเชิงลึกที่สามารถตรวจจับวัตถุได้พร้อมกันหลายตำแหน่งในภาพ เช่น Faster R-CNN ซึ่งได้รับการยอมรับว่าให้ความแม่นยำสูงในการตรวจจับวัตถุหลากหลายประเภทผลลัพธ์จากระบบนี้สามารถนำไปประยุกต์ใช้ร่วมกับกล้องวงจรปิดในพื้นที่ก่อสร้าง เพื่อแจ้งเตือนกรณีที่ตรวจพบแรงงานไม่ได้สวมหมวกนิรภัย ช่วยลดภาระงานของเจ้าหน้าที่เพิ่มความปลอดภัยในสถานที่ทำงาน และสอดคล้องกับมาตรฐานความปลอดภัยสากล

ดังนั้นโครงงาน “ระบบตรวจจับหมวกนิรภัยในเขตก่อสร้างด้วย Deep Learning” จึงมุ่งเน้นการประยุกต์ใช้เทคโนโลยีการเรียนรู้เชิงลึกเพื่อเพิ่มประสิทธิภาพในการตรวจสอบแรงงานในพื้นที่ปฏิบัติงานให้มีความปลอดภัยยิ่งขึ้น และเป็นแนวทางหนึ่งในการสนับสนุนการนำระบบอัตโนมัติและปัญญาประดิษฐ์มาใช้ในงานด้านอุตสาหกรรมก่อสร้างและความปลอดภัยแรงงานในอนาคต

2. วัตถุประสงค์ (Objectives)

โครงงาน “ระบบตรวจจับหมวกนิรภัยในเขตก่อสร้างด้วย Deep Learning” มีวัตถุประสงค์ดังต่อไปนี้

1. เพื่อแก้ไขปัญหาที่เกิดขึ้นจริงในภาคอุตสาหกรรมก่อสร้าง เนื่องจากแรงงานบางส่วนไม่ปฏิบัติตามข้อกำหนดด้านความปลอดภัยในการสวมหมวกนิรภัย ซึ่งเป็นสาเหตุสำคัญของอุบัติเหตุในพื้นที่ก่อสร้าง
2. เพื่อพัฒนาระบบตรวจจับหมวกนิรภัยโดยอัตโนมัติจากภาพกล้องวงจรปิดหรือภาพถ่าย โดยใช้เทคโนโลยีปัญญาประดิษฐ์ (AI) และการเรียนรู้เชิงลึก (Deep Learning) ในการระบุว่าบุคคลในภาพมีการสวมหมวกนิรภัยหรือไม่
3. เพื่อลดข้อจำกัดของการตรวจสอบด้วยมนุษย์ เช่น ความผิดพลาดจากความเมื่อยล้าของผู้ตรวจสอบ การตรวจไม่ครบทุกมุมกล้อง หรือการเฝ้าระวังที่ไม่ต่อเนื่อง
4. เพื่อเพิ่มความปลอดภัยในสถานที่ทำงานและลดความสูญเสียที่เกิดจากอุบัติเหตุ โดยระบบสามารถนำไปประยุกต์ใช้ร่วมกับกล้องวงจรปิด เพื่อแจ้งเตือนเจ้าหน้าที่เมื่อพบแรงงานที่ไม่ได้สวมหมวกนิรภัย
5. เพื่อสร้างโมเดล Deep Learning ที่มีขั้นตอนครบถ้วนตามหลักการพัฒนาโมเดล AI ได้แก่ การเตรียมข้อมูล (Data Preparation) การสร้างโมเดล (Model Construction) การฝึกสอน (Training) และการประเมินผล (Evaluation)
6. เพื่อศึกษาการประยุกต์ใช้เทคโนโลยี Deep Learning ในงานจริง (Real-world Application) เนื่องจากหัวข้อนี้สามารถนำไปใช้งานได้จริงในภาคสนาม เช่น พื้นที่ก่อสร้าง โรงงาน หรือสถานประกอบการที่ต้องควบคุมมาตรการความปลอดภัย
7. เพื่อเป็นโครงงานตัวอย่างที่เหมาะสมกับการเรียนรู้ในระดับมหาวิทยาลัย เพราะมีความครบถ้วนทั้งด้านเทคนิคและประโยชน์เชิงสังคม รวมทั้งใช้เทคโนโลยีสมัยใหม่ที่กำลังได้รับความนิยมสูงในวงการอุตสาหกรรมและการวิจัย

3. เหตุผลที่ใช้ Deep learning (Reason of using Deep learning)

เหตุผลที่ใช้ Deep Learning (Faster R-CNN) เพราะ

1. ปัญหาเป็นการตรวจจับวัตถุ (Object Detection) ซึ่งต้องแยกทั้ง ตำแหน่ง และ ประเภทของวัตถุ (หมวกนิรภัย, คน) งานลักษณะนี้ซับซ้อนเกินกว่าจะใช้การประมวลผลภาพแบบดั้งเดิม (เช่น edge detection หรือ color threshold)
2. Deep Learning สามารถเรียนรู้ฟีเจอร์จากภาพได้โดยอัตโนมัติ ไม่ต้องเขียนกฎ (feature engineering) เอง ช่วยให้ตรวจจับหมวกในสภาพแสง มุมมอง หรือพื้นหลังที่หลากหลายได้ดี
3. Faster R-CNN มีสถาปัตยกรรมสองขั้นตอน (RPN + ROI Head) ที่ให้ความแม่นยำสูงกว่า one-stage detectors (เช่น YOLO) ในงานที่ต้องการความถูกต้องมากกว่าความเร็ว
4. สามารถปรับ fine-tune จากโมเดลที่ pretrain แล้ว (transfer learning) ได้ ทำให้ใช้ dataset จำนวนน้อยแต่ยังได้ผลลัพธ์ดี
5. รองรับการขยายไปยังคลาสหรือสถานการณ์ใหม่ เช่น ตรวจจับคนไม่สวมหมวก, ตรวจจับในโรงงานหรือไซต์ก่อสร้างอื่น ๆ ได้ง่ายในอนาคต

4. สถาปัตยกรรม Deep Learning ที่ใช้ (Model Architecture)

4.1 ประเภทของโมเดล

1. เป็น Convolutional Neural Network (CNN)
2. ทำหน้าที่ Object Detection → ทั้ง ตำแหน่ง (bounding box) และ ประเภทของวัตถุ (class)
3. มีโครงสร้างแบบ Two-stage detector
    a. Region Proposal Network (RPN) เสนอกรอบวัตถุที่น่าจะมีของ
    b. ROI Head จำแนกคลาส + ปรับกรอบ (Refine bounding box)

4.2 โครงสร้างที่ใช้จริง

โมเดลที่ใช้ในโครงงานนี้คือ Faster R-CNN (Faster Region-based Convolutional Neural Network) ซึ่งเป็นสถาปัตยกรรมแบบสองขั้นตอน (Two-Stage Detector) ที่ได้รับความนิยมสูงในงานตรวจจับวัตถุ เนื่องจากสามารถให้ผลลัพธ์ที่มีความแม่นยำสูงทั้งในด้านการระบุตำแหน่งและการจำแนกประเภทของวัตถุภายในภาพ

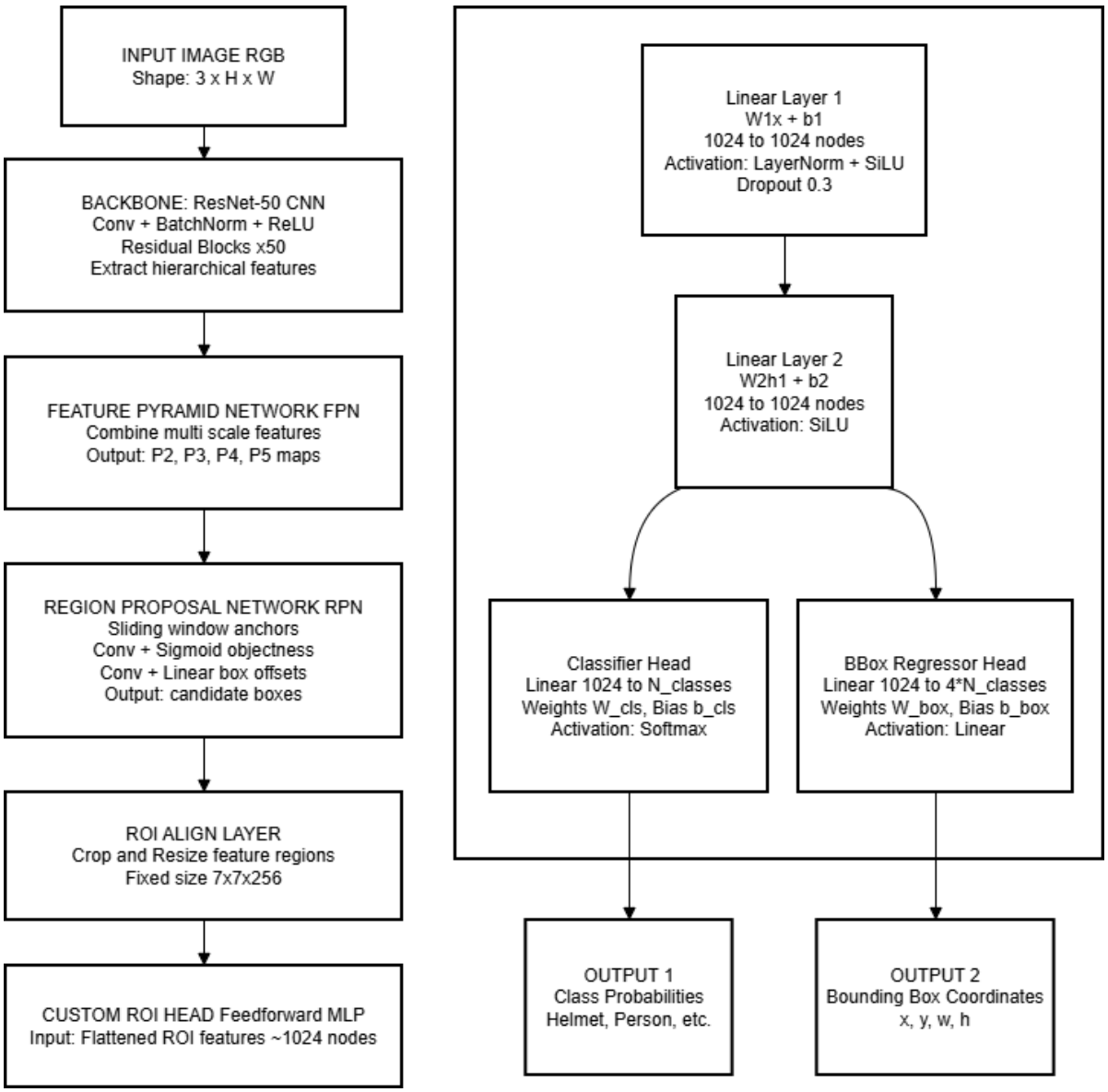
ในโครงงานนี้ โมเดลพื้นฐานถูกโหลดมาจากไลบรารี PyTorch Vision Model Zoo โดยใช้สถาปัตยกรรม Faster R-CNN ที่มี ResNet-50 เป็น Backbone และเพิ่มโครงข่าย Feature Pyramid Network (FPN) เพื่อรวมคุณลักษณะจากหลายระดับความละเอียดของภาพ (multi-scale features) ทำให้โมเดลสามารถตรวจจับวัตถุได้ทั้งขนาดเล็กและขนาดใหญ่ในภาพเดียวกันอย่างมีประสิทธิภาพ

ส่วนหัวของโมเดล (ROI Head) ได้รับการปรับปรุงให้เป็นแบบ Custom Feedforward Head ซึ่งออกแบบใหม่ให้มีชั้นเชิงลึก (deep layers) หลายชั้นที่ประกอบด้วย Linear Layer, Layer Normalization, ฟังก์ชันกระตุ้นแบบ SiLU (Sigmoid Linear Unit) และ Dropout เพื่อเพิ่มความสามารถในการเรียนรู้เชิงไม่เชิงเส้น (non-linearity) และลดปัญหา overfitting เมื่อฝึกด้วยข้อมูลจำกัด ส่วนหัวนี้จะแยกออกเป็นสองส่วนย่อย คือ

1. Classifier Head สำหรับจำแนกประเภทของวัตถุ เช่น หมวกนิรภัยหรือบุคคล
2. Bounding Box Regressor Head สำหรับปรับพิกัดกรอบวัตถุให้แม่นยำยิ่งขึ้น

นอกจากนี้ โมเดลยังได้รับการปรับแต่งเพิ่มเติมเพื่อเพิ่มประสิทธิภาพด้านความเร็ว โดยลดจำนวนกรอบผู้สมัคร (Region Proposals) ที่สร้างจากเครือข่าย RPN ลงในระหว่างการฝึกและการทดสอบ รวมถึงลดจำนวนตัวอย่างต่อภาพในขั้นตอน ROI Sampling เพื่อให้การทำงานเบาลงโดยไม่กระทบต่อความแม่นยำของผลลัพธ์

4.3 แผนภาพสถาปัตยกรรม



ภาพที่ 1 แสดงสถาปัตยกรรมของโมเดล Faster R-CNN ที่ใช้ในโครงงาน ซึ่งมีการปรับส่วน ROI Head ให้เป็นแบบ Custom Feedforward MLP พร้อม LayerNorm และ SiLU Activation เพื่อเพิ่มความยืดหยุ่นในการเรียนรู้ลักษณะของหมวกนิรภัย

5. ขั้นตอนการทำงานของโค้ด (Implementation)

5.1 การเตรียมข้อมูล

1. อ่านภาพและไฟล์ annotation ด้วย xml.etree.ElementTree เพื่อดึงค่าพิกัด bounding box และชื่อคลาส
2. แปลงข้อมูล annotation ให้เป็น list ของ bounding boxes (x_min, y_min, x_max, y_max) และ label id
3. ใช้ torchvision.transforms สำหรับการแปลงภาพ ได้แก่
    - Resize((640, 640)) → ปรับขนาดภาพให้เท่ากัน
    - ToTensor() → แปลงภาพเป็น Tensor เพื่อป้อนเข้าโมเดล
4. สร้างคลาส Custom Dataset (HelmetDataset)
5. เมธอด _getitem_ → ส่งคืนภาพ + target (boxes, labels, image_id, area, iscrowd)
6. เมธอด _len_ → ส่งคืนจำนวนภาพทั้งหมดใน dataset
7. สร้าง DataLoader สำหรับชุด Train, Validation, และ Test โดยใช้ torch.utils.data.DataLoader() พร้อมตั้งค่า batch_size, shuffle, และ collate_fn

5.2 การสร้างโมเดล

โมเดลพื้นฐานใช้สถาปัตยกรรม Faster R-CNN ที่มี ResNet-50 เป็น Backbone และ Feature Pyramid Network (FPN) สำหรับรวมคุณลักษณะหลายสเกล โดยมีการปรับส่วนหัว ROI Head ให้เป็นแบบ Custom Feedforward Network ที่ประกอบด้วย Linear Layer, Layer Normalization, SiLU Activation และ Dropout เพื่อเพิ่มประสิทธิภาพการเรียนรู้และลดการเกิด overfitting

ในการฝึกสอน ใช้ Optimizer แบบ AdamW พร้อม Learning Rate 7×10^{-4} และ Weight Decay 1×10^{-2} พร้อมเปิดใช้งาน Mixed Precision Training (AMP) เพื่อเพิ่มความเร็ว และลดการใช้หน่วยความจำของ GPU

5.3 การเทรน

1. การกำหนดตัวปรับค่าน้ำหนัก (Optimizer)
    - ใช้ AdamW ซึ่งเป็นตัวปรับค่าน้ำหนักที่มีการควบคุมค่า Weight Decay เพื่อป้องกัน overfitting ได้ดีกว่า SGD
    - ตั้งค่า Learning Rate = 7×10^{-4} และ Weight Decay = 1×10^{-2}
2. เทคนิคการเร่งการฝึก (Acceleration Technique)
    - ใช้ Automatic Mixed Precision (AMP) เพื่อลดการใช้หน่วยความจำและเพิ่มความเร็วในการคำนวณ
    - ทำให้สามารถฝึกโมเดลขนาดใหญ่บน GPU ได้โดยไม่สูญเสียความแม่นยำของผลลัพธ์
3. จำนวนรอบการฝึก (Epochs)
    - กำหนดการฝึกทั้งหมด 12 Epochs โดยในแต่ละรอบจะคำนวณค่าความสูญเสีย (Loss) เพื่อประเมินแนวโน้มการเรียนรู้
4. องค์ประกอบของ Loss Function

    โมเดล Faster R-CNN จะรวมค่าความสูญเสียจาก 4 ส่วนหลัก ดังนี้
    - loss_classifier — วัดความผิดพลาดในการจำแนกประเภทของวัตถุ
    - loss_box_reg — วัดความคลาดเคลื่อนของกรอบวัตถุที่ทำนาย
    - loss_objectness — ประเมินความมั่นใจของ RPN ว่าบริเวณที่ตรวจพบมีวัตถุจริงหรือไม่
    - loss_rpn_box_reg — วัดความถูกต้องของกรอบที่สร้างจากเครือข่าย RPN
5. การบันทึกและประเมินระหว่างการฝึก
    - บันทึกค่าความสูญเสีย (Loss) ของชุด Train และ Validation ทุก Epoch
    - แสดงผลเป็นกราฟ Loss Curve เพื่อวิเคราะห์การเรียนรู้ของโมเดล
    - เก็บเฉพาะโมเดลที่ให้ค่าความสูญเสียต่ำที่สุด (Best Model) สำหรับนำไปประเมินผลในขั้นตอนถัดไป

6. การปรับค่าพารามิเตอร์เพื่อเพิ่มความเร็ว
    - ลดจำนวน Region Proposal ที่สร้างต่อภาพลง เพื่อให้ฝึกได้เร็วขึ้น
    - ปรับ roi_heads.batch_size_per_image จากค่าเริ่มต้น 128 เหลือ 64 โดยไม่กระทบต่อความแม่นยำ

5.4 การประเมินผล

ตั้งค่า threshold = 0.5 สำหรับคัดกรองผลทำนาย (กล่องที่มีความเชื่อมั่นต่ำกว่าถูกละทิ้ง)

คำนวณ Metric หลัก ได้แก่

- Precision (ความแม่นยำ)
- Recall (การครอบคลุม)
- F1-score (ค่ากลางของ Precision และ Recall)
- mAP (mean Average Precision) หากมีการวัดผลแบบหลายคลาส

แสดงผลภาพทดสอบพร้อมกล่องทำนาย (Bounding Boxes) และชื่อคลาสเหนือกรอบ เปรียบเทียบผลระหว่างภาพจริง (Ground Truth) และภาพที่โมเดลตรวจจับได้

5.5 ลิงก์โค้ด (Public)

GitHub: https://github.com/svkii1610/final-helmet-detection.git

6. การฝึกโมเดล Deep Learning (Model Training)

6.1 โมเดลที่ใช้

โมเดลหลักของโครงงานคือ Faster R-CNN ที่มี ResNet-50 เป็น Backbone และใช้ Feature Pyramid Network (FPN) สำหรับรวมคุณลักษณะจากหลายระดับความละเอียด โดยมีการปรับส่วน ROI Head ให้เป็นแบบ Custom Feedforward เพื่อให้เหมาะสมกับการตรวจจับหมวกนิรภัยในไซต์ก่อสร้าง

6.2 ข้อมูลที่ใช้ในการฝึก (Dataset)

1. ชื่อชุดข้อมูล: Construction Site Safety Image Dataset (Roboflow)
2. แหล่งที่มา: ดาวน์โหลดจาก Kaggle

   https://www.kaggle.com/datasets/snehilsanyal/construction-site-safety-image-dataset-roboflow

3. จำนวนข้อมูลรวมประมาณ 5,000 ภาพ แบ่งเป็น 3 ชุดข้อมูล:
   - Train: ใช้สำหรับฝึกโมเดล
   - Validation (Val): ใช้สำหรับตรวจสอบระหว่างการฝึก
   - Test: ใช้สำหรับประเมินผลสุดท้าย

   สัดส่วนโดยประมาณ: Train 80% / Val 10% / Test 10%

4. รูปแบบข้อมูล: Pascal VOC (.xml)
   - 1 ภาพ จะมีไฟล์ annotation .xml คู่กัน (ชื่อไฟล์เหมือนกัน)
   - ภายในไฟล์ .xml จะระบุชื่อคลาสและพิกัดของ bounding box
5. Class:
   - Helmet (หมวกนิรภัย)
   - No Helmet / Head (หัวคนที่ไม่สวมหมวกนิรภัย)
   - Person (คน)
6. รูปแบบโฟลเดอร์:

   Dataset/

   ├── train/

   │ ├── images/

   │ └── annotations/

   ├── val/

   │ ├── images/

   │ └── annotations/

└── test/
│ ├── images/
│ └── annotations/

6.3 ขั้นตอนการฝึก

1. การเตรียมข้อมูลสำหรับการฝึก
    - ปรับขนาดภาพทั้งหมดให้อยู่ในขนาด 640×640 พิกเซล
    - แปลงภาพและ annotation ให้อยู่ในรูปแบบ Tensor
    - ใช้ DataLoader ของ PyTorch เพื่อสุ่ม (shuffle) ลำดับข้อมูลในแต่ละ Epoch
    - ใช้ collate_fn สำหรับรวมข้อมูลภาพและกล่อง annotation เข้าด้วยกันอย่างถูกต้อง
2. กระบวนการฝึกในแต่ละ Epoch
    - โหลดภาพทีละชุด (Batch) และส่งเข้าโมเดลในโหมด train()
    - โมเดลคำนวณ Loss รวมจากทุกส่วน และปรับพารามิเตอร์ด้วย Optimizer
    - ใช้เทคนิค Mixed Precision เพื่อให้การคำนวณรวดเร็วและประหยัดหน่วยความจำ
3. การตรวจสอบผลระหว่างการฝึก (Validation)
    - หลังจากฝึกในแต่ละ Epoch จะมีการประเมินด้วยชุด Validation
    - บันทึกค่า Loss ของ Validation เพื่อใช้เปรียบเทียบกับ Train Loss
    - หากพบว่า Validation Loss ต่ำที่สุดในรอบนั้น จะบันทึกโมเดลเป็น “Best Model”
4. การติดตามผลการเรียนรู้ของโมเดล
    - แสดงกราฟ Loss Curve เพื่อดูแนวโน้มการลดลงของค่าความสูญเสีย
    - วิเคราะห์ว่ามีการเกิด Overfitting หรือไม่
    - กราฟที่มีแนวโน้มลดลงอย่างต่อเนื่องแสดงถึงการเรียนรู้ที่เสถียรและมีประสิทธิภาพ
5. สรุปผลหลังการฝึก
    - โมเดลที่ได้สามารถตรวจจับหมวกนิรภัยและบุคคลได้อย่างถูกต้องในระดับสูง
    - มีความเสถียรในการเรียนรู้ และสามารถนำไปประเมินผลต่อบนชุดทดสอบได้โดยไม่เกิดการ overfitting

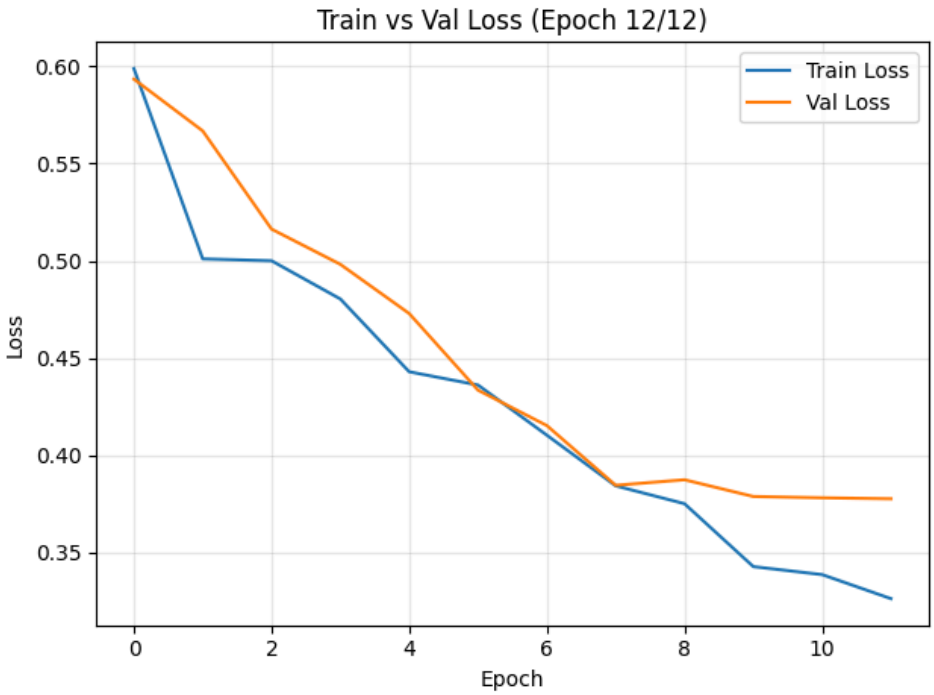
7. การประเมินผลของโมเดล (Model Evaluation)

7.1 ขั้นตอนการประเมิน

1. โหลดโมเดลที่ผ่านการเทรนเสร็จแล้ว (Best Model)
2. ทดสอบกับชุดข้อมูล Validation และ Test
3. ตั้งค่า Threshold = 0.5 สำหรับคัดเลือกกล่องที่มีความมั่นใจสูง
4. แสดงภาพผลลัพธ์พร้อมกรอบ Bounding Box และชื่อคลาส

7.2 ค่าความสูญเสีย (Loss Visualization)

- ค่าความสูญเสียหลักมาจากการรวมของ Loss 4 ส่วน (Classifier, Box Regression, Objectness, RPN Box Reg)
- จากกราฟการเรียนรู้ พบว่า Loss มีแนวโน้มลดลงต่อเนื่องตามจำนวน epoch แสดงให้เห็นว่าโมเดลสามารถเรียนรู้ลักษณะของวัตถุได้ดีและไม่เกิด overfitting อย่างชัดเจน



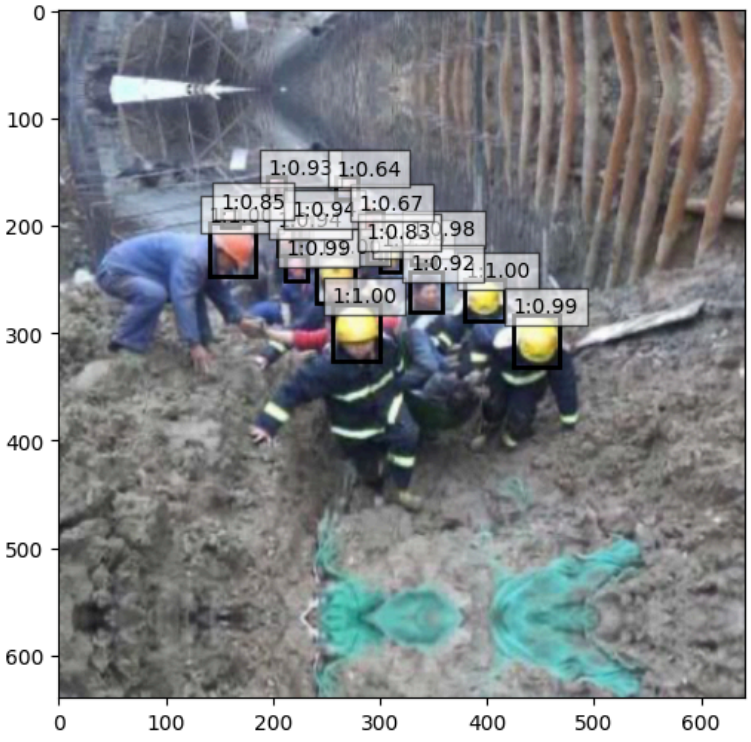
ภาพที่ 2 กราฟแสดงค่า Train Loss และ Validation Loss ตลอดการฝึกสอน 12 Epoch แสดงแนวโน้มการเรียนรู้ของโมเดล

7.3 ค่าชี้วัด (Evaluation Metrics)

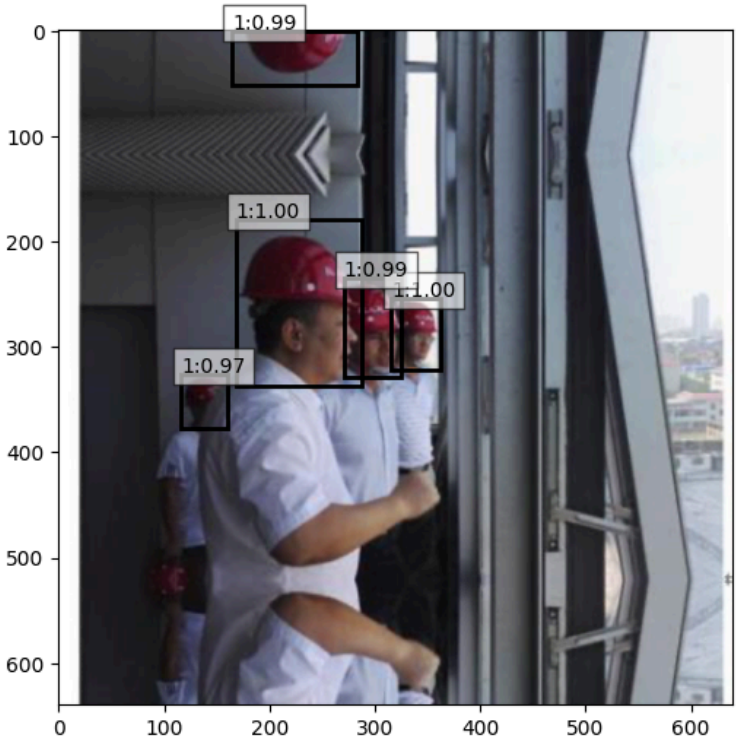
โมเดลถูกประเมินด้วยค่าชี้วัดมาตรฐานที่ใช้ในงานตรวจจับวัตถุ (Object Detection) เพื่อวัดประสิทธิภาพทั้งด้านความถูกต้อง ความแม่นยำ และความครอบคลุมของการตรวจจับ ดังนี้

1. Accuracy (ความถูกต้องโดยรวม)
   เป็นค่าที่แสดงสัดส่วนของการทำนายที่ถูกต้องทั้งหมดเมื่อเทียบกับจำนวนข้อมูลทั้งหมดในชุดทดสอบ (ความสามารถของโมเดลในการจำแนกภาพได้ถูกต้องในภาพรวม)
2. Precision (ความแม่นยำ)
   เป็นค่าที่แสดงสัดส่วนของกรอบที่โมเดลทำนายว่าเป็น “หมวกนิรภัย” แล้วถูกต้องจริงจากจำนวนกรอบที่ทำนายว่าเป็นหมวกทั้งหมด (ความแม่นยำในการระบุว่าวัตถุที่ตรวจพบเป็นหมวกนิรภัยจริง ไม่ใช่การตรวจจับผิดพลาด)
3. Recall (ความครอบคลุม)
   เป็นค่าที่แสดงสัดส่วนของวัตถุจริงทั้งหมดในภาพที่โมเดลสามารถตรวจจับได้ถูกต้อง (ความสามารถของโมเดลในการค้นหาวัตถุจริงทั้งหมดโดยไม่พลาด)
4. F1-Score (ค่ากลางระหว่าง Precision และ Recall)
   เป็นค่าที่คำนวณจากค่า Precision และ Recall เพื่อประเมินความสมดุลระหว่างความแม่นยำและความครอบคลุม
   ค่า F1-Score สูงแสดงว่าโมเดลมีความแม่นยำและสามารถตรวจจับวัตถุได้ครบถ้วนในระดับที่เหมาะสม
5. mAP (mean Average Precision)
   เป็นค่าความแม่นยำเฉลี่ยในหลายระดับของค่า Threshold สำหรับการตรวจจับทุกคลาส
   เป็นตัวชี้วัดที่นิยมใช้ในงาน Object Detection เพื่อวัดประสิทธิภาพโดยรวมของโมเดล ทั้งในด้านความถูกต้องของการจัดตำแหน่งกรอบ (bounding box) และความแม่นยำของการจำแนกประเภทวัตถุ

7.4 ตัวอย่างผลการทำนาย



ภาพที่ 3 ผลการตรวจจับหมวกนิรภัยของเจ้าหน้าที่กู้ภัยในไซต์ก่อสร้างที่มีสภาพแวดล้อมซับซ้อน



ภาพที่ 4 ผลการตรวจจับหมวกนิรภัยในภาพถ่ายภายในอาคาร ซึ่งมีการสะท้อนแสงและมุมมองเฉียง

7.5 สรุปผลการประเมิน

จากการประเมินผล โมเดลได้ค่า Precision เฉลี่ย ~ 0.916 และ Recall ~ 0.828 ซึ่งถือว่าอยู่ในระดับสูง สามารถตรวจจับหมวกนิรภัยได้อย่างมีประสิทธิภาพทั้งในกลางแจ้งและในอาคาร

8. งานที่เกี่ยวข้อง (Related Work)

งานที่เกี่ยวข้องส่วนใหญ่ เช่น YOLOv5 (IEEE, 2021) เน้นประสิทธิภาพแบบ Real-time แต่มีความแม่นยำต่ำกว่าในสภาพแสงซับซ้อน ขณะที่งาน Faster R-CNN (Elsevier, 2020) ให้ความแม่นยำสูงและเหมาะกับการตรวจจับวัตถุในสภาพแวดล้อมจริง เช่น พื้นที่ก่อสร้าง ซึ่งเป็นเหตุผลหลักที่เลือกใช้สถาปัตยกรรมนี้ในโครงงาน

9. การแบ่งหน้าที่ภายในกลุ่ม (Team Contribution)

1. นางสาวบุณยาพร บุญทับ — รับผิดชอบงานดังนี้
    - ออกแบบและพัฒนาโครงสร้างโมเดล Deep Learning (Faster R-CNN)
    - ค้นคว้าและรวบรวมข้อมูลพื้นฐานเกี่ยวกับ Faster R-CNN
    - เขียนโค้ดฝึกโมเดล (Training) และปรับพารามิเตอร์ให้เหมาะสม
    - ประเมินผลของโมเดล ทดลองทำนายภาพ (Prediction & Visualization)
    - วิเคราะห์ผลลัพธ์และจัดทำกราฟแสดงค่า Loss และ Metric
    - เขียนรายงานส่วน Model Architecture, Training และ Evaluation
2. นางสาวไข่มุกข์ ธานะกิจ — รับผิดชอบงานดังนี้
    - ร่วมออกแบบโครงสร้างโมเดล Deep Learning (Faster R-CNN)
    - เตรียมและจัดการชุดข้อมูล รวมถึงปรับโครงสร้างโฟลเดอร์และ Annotation
    - ค้นคว้าและรวบรวมข้อมูลพื้นฐานเกี่ยวกับ Object Detection
    - ตรวจสอบคุณภาพของ Dataset และช่วยเตรียมข้อมูลสำหรับการทดสอบ
    - เขียนรายงานส่วน Dataset, Implementation และสรุปผลการทดลอง
    - จัดรูปแบบรายงานและตรวจทานความถูกต้องก่อนส่ง

10. สรุปผลและข้อเสนอแนะ (Conclusion & Future Work)

1. ระบบสามารถตรวจจับการสวมหมวกนิรภัยได้อย่างมีประสิทธิภาพ
2. สามารถนำไปติดตั้งร่วมกับกล้องวงจรปิดในไซต์งานได้
3. ข้อจำกัด: ภาพที่แสงน้อยหรือหมวกสีใกล้เคียงฉากหลังทำให้ตรวจพลาด
4. งานต่อยอด:
    - ทำ real-time detection ด้วย YOLO
    - เพิ่มคลาส “vest” (เสื้อสะท้อนแสง)
    - สร้าง Dashboard สำหรับระบบแจ้งเตือน