PROJECT 2
PERSON TRACKING

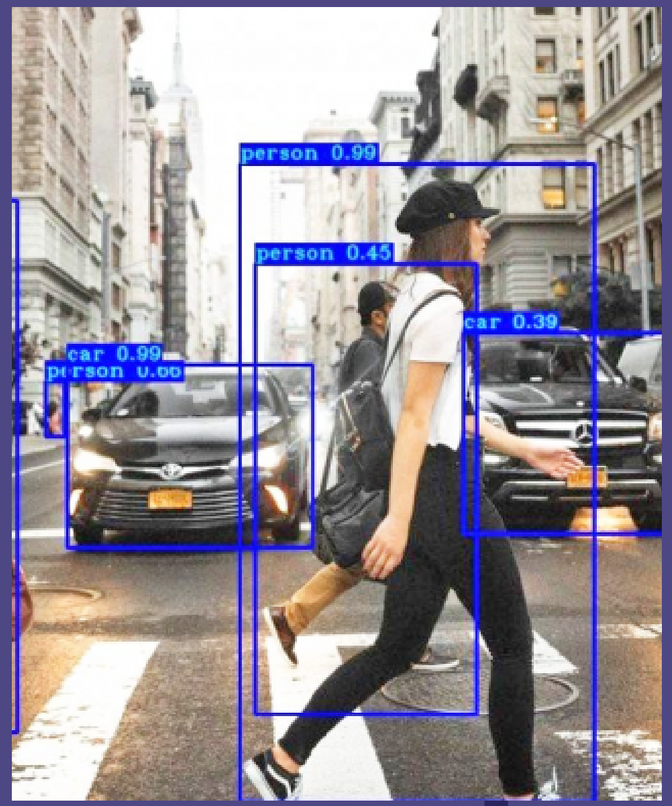
Faster RCNN & YOLO

Presented by CV B & CV D

LATAR BELAKANG

Dalam era perkembangan teknologi komputer dan kecerdasan buatan, **pemantauan objek dan deteksi manusia** menjadi **fokus utama dalam** berbagai aplikasi, termasuk **keamanan lingkungan, pemantauan lalu lintas**, sampai dengan **sektor industri dan perdagangan**.

Sistem pelacakan manusia yang efektif menjadi kunci untuk meningkatkan keamanan dan efisiensi di berbagai kondisi lingkungan.

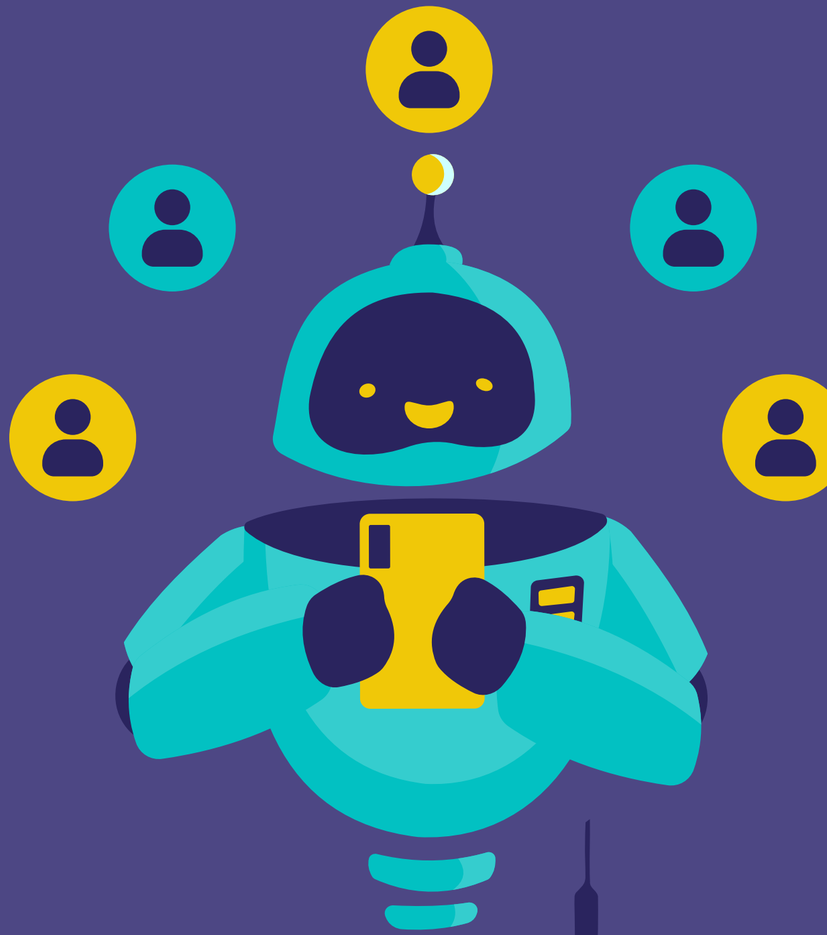


PROBLEM STATEMENT

Dalam deteksi objek khususnya person tracking, terdapat beberapa tantangan dikarenakan perubahan pose dan interaksi antar objek dalam satu frame.

Eksplorasi algoritma Faster RCNN dan YOLO diharapkan memberikan solusi terkait deteksi manusia yang cepat dan akurat.

Proyek ini bertujuan mengatasi kendala deteksi dalam kerumunan, perubahan pose, dan pemantauan real-time, mendukung pengembangan sistem pelacakan manusia yang lebih efisien.



Timeline

Activity	Week I							Week II							PIC
	18 NOV	19 NOV	20 NOV	21 NOV	22 NOV	23 NOV	24 NOV	25 NOV	26 NOV	27 NOV	28 NOV	29 NOV	30 NOV	01 DEC	
Data Collection & Data Analytics	■	■	■	■	■	■									ALL
Model Selection & Assignment (Meet)							■								ALL
Build Model (Faster RCNN MobileNet)							■	■	■	■	■	■	■		Faturahman, Fitrah
Build Model (Faster RCNN GoogLeNet)							■	■	■	■	■	■	■		Hendra, Dani
Build Model (Faster RCNN ResNet)							■	■	■	■	■	■	■		Harrison, Yogi
Build Model (YOLOv3)							■	■	■	■	■	■	■		Satriaji
Build Model (YOLOv5)							■	■	■	■	■	■	■		Wahyu, Yurixa
Build Model (YOLOv8)							■	■	■	■	■	■	■		Satriaji, Nadya
Evaluation (Meet)											■		■	■	ALL
Making PPT (Meet)													■	■	ALL

1

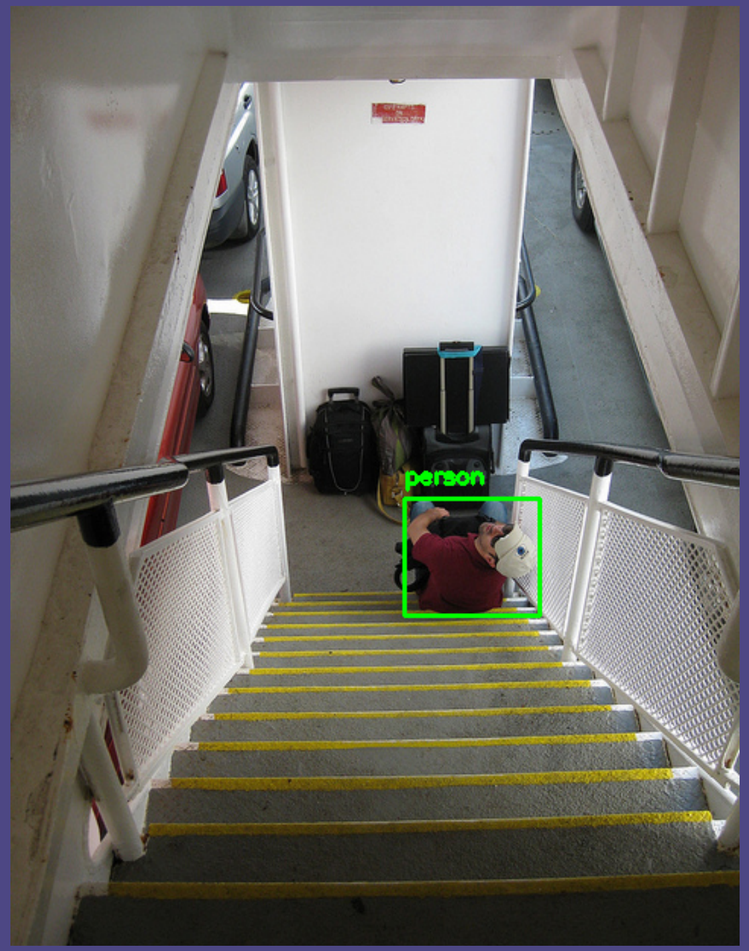
Data Collection & Preparation

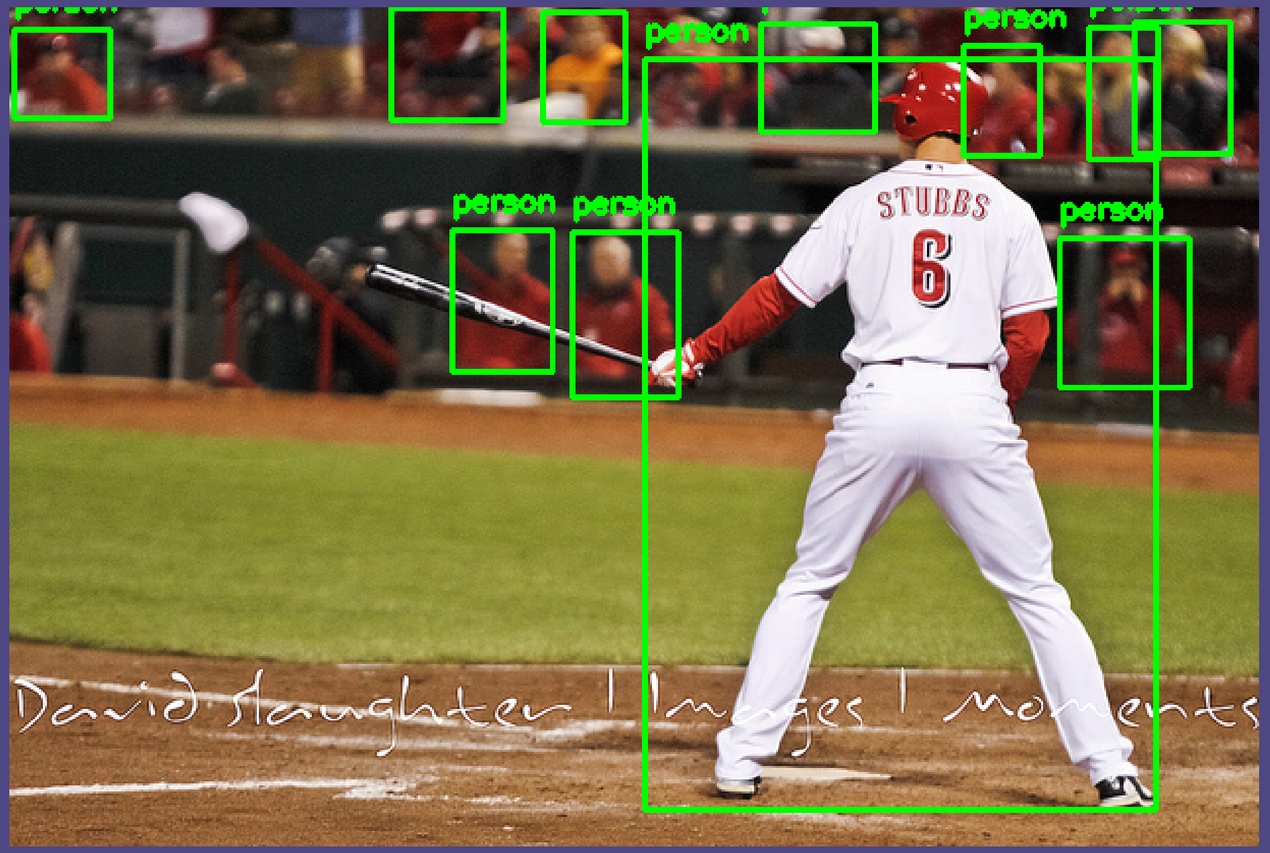
COCO 2017-FiftyOne Dataset

- 4000 train
- 500 validasi
- 500 test

Note : seluruh data difilter untuk diambil satu class person saja

Images Example With Bounding Box from Dataset



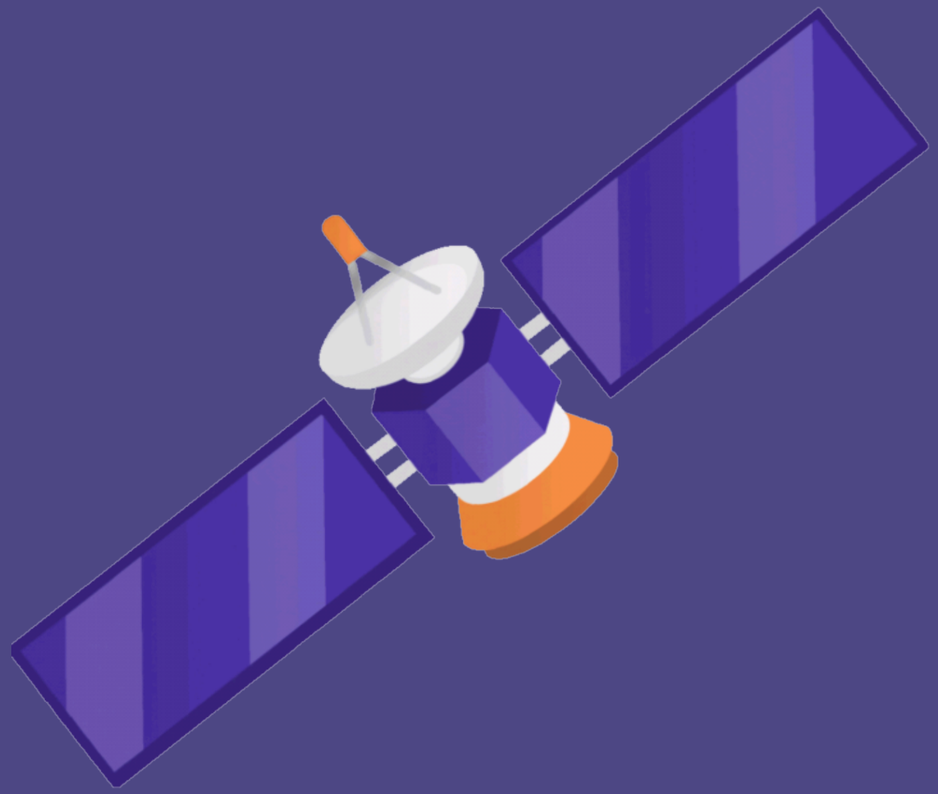


MODEL DEVELOPMENT

FASTER RCNN

AND

YOLO



LIBRARY AND TOOLS

FiftyOne

FASTER R-CNN

Backbone

- GoogLeNet
- MobileNet
- ResNet50

GOOGLENET

- GoogleNet (Inception V1) & InceptionV3
- Normalisasi
- Optimizer
  - SGD
    - lr = 0.0001
    - momentum = 0.9
    - weight decay = 0.0001
  - Adam
    - lr = 0.001
- LR scheduler (gamma=0.1)

training ± 2.5 jam 5 epochs
mAP 0.133

MOBILENET

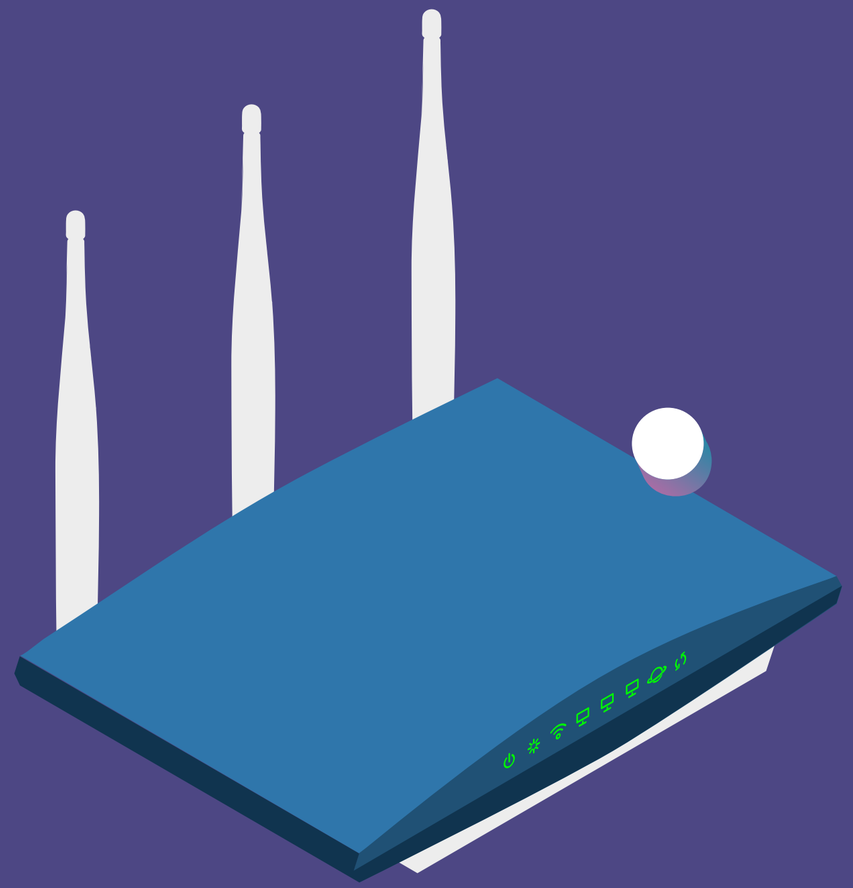
- mobilenet_v3_large_fpn & mobilenet_v3_large_320_fpn
- Normalisasi
- Optimizer
  - SGD
    - lr = 0.005 and 0.1
    - momentum = 0.9
    - weight decay = 0.0005 and 0.0001
- LR scheduler (gamma=0.1)

RESNET50

- Resnet50 (pretrained = True)
- Normalisasi
- Optimizer
  - SGD
    - lr = 0.005
    - momentum = 0.9
    - weight decay = 0.0005
- LR scheduler (gamma=0.1)

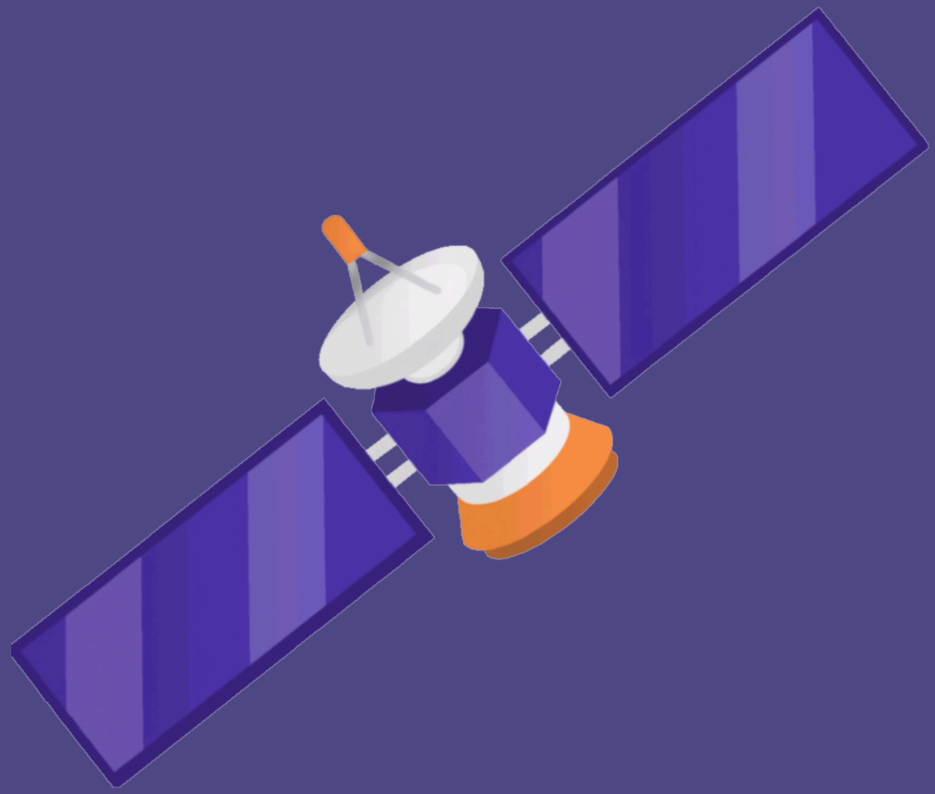
YOLO MODEL

- YOLO V3u
- YOLO V5
  - YOLO V5s
  - YOLO V5x
- YOLO V8
  - YOLOV8m
  - YOLOV8l
  - YOLOV8x



YOLO HYPERPARAMETER

1. Epochs: 50 | 100
2. Batch: 8 | 16
3. Image Size: 640x640
4. Learning Rate :
    a. lr0 = 0.01
    b. lrf = 0.0001
5. Momentum:0.937
6. Weight Decay:0.0005
7. Optimizer : AdamW



YOLO AUGMENTATION

- Mosaic : 1.0
- Translate: 0.1
- Scale: 0.5
- fliplr: 0.5
- hsv_h (hue): 0.015
- hsv_s (saturation): 0.7
- hsv_v (value): 0.4

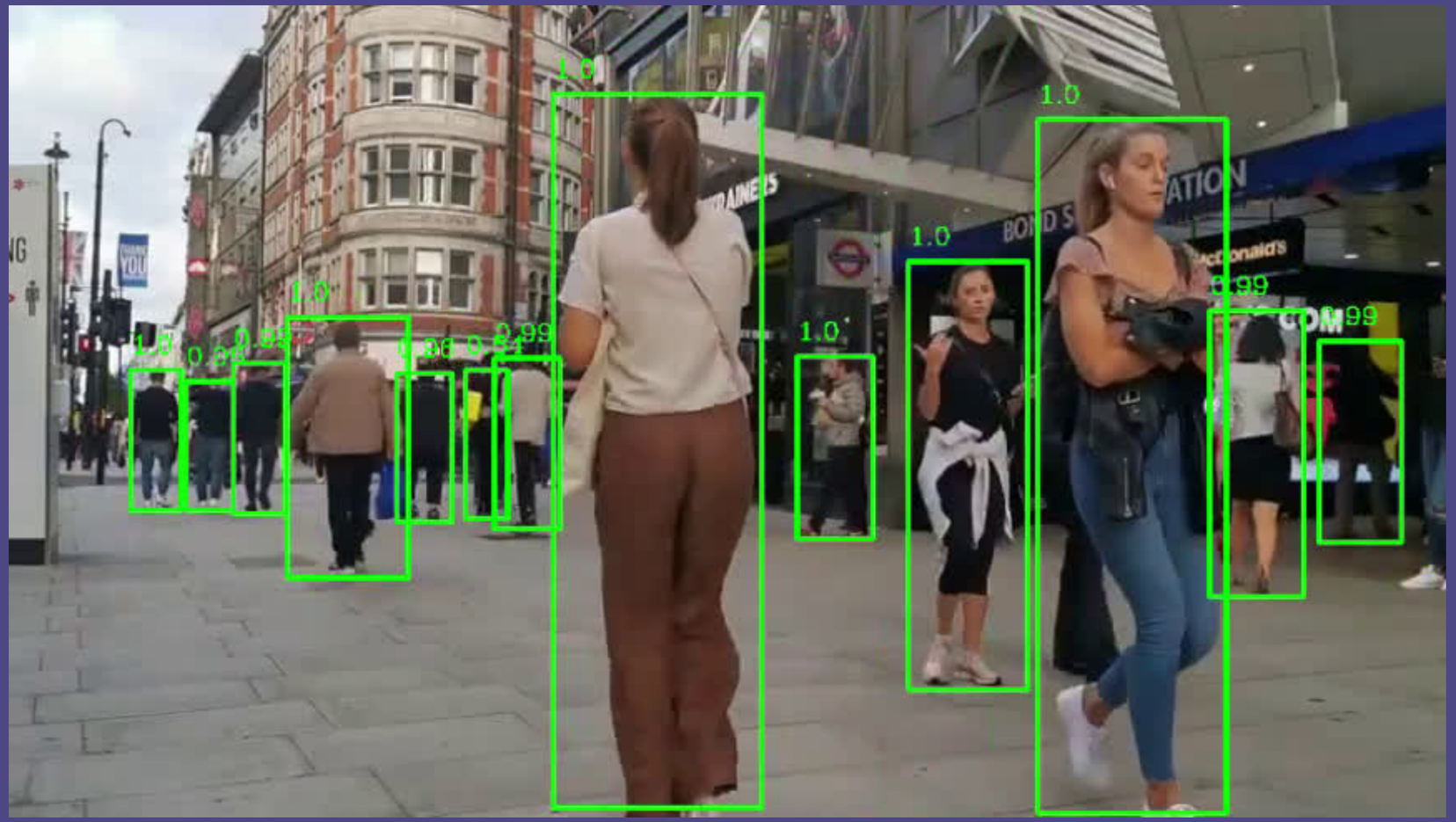
Training & Evaluation (YOLO)

No.	Model	Epoch	Batch	Precision (%)	Recall (%)	mAP@50 (%)
1	YOLO V3u	50	8	78.1	61.5	71.3
2	YOLO V5s	100	16	76.2	59.7	66.4
3	YOLO V5x	50	8	67	53.4	59
4	YOLO V8n	50	16	74.2	59.5	68.3
5	YOLO V8m	50	16	77.3	63.5	73.4
6	YOLO V8l	50	8	76.6	63.6	72.7
7	YOLO V8x	50	8	80.4	62.1	73.2

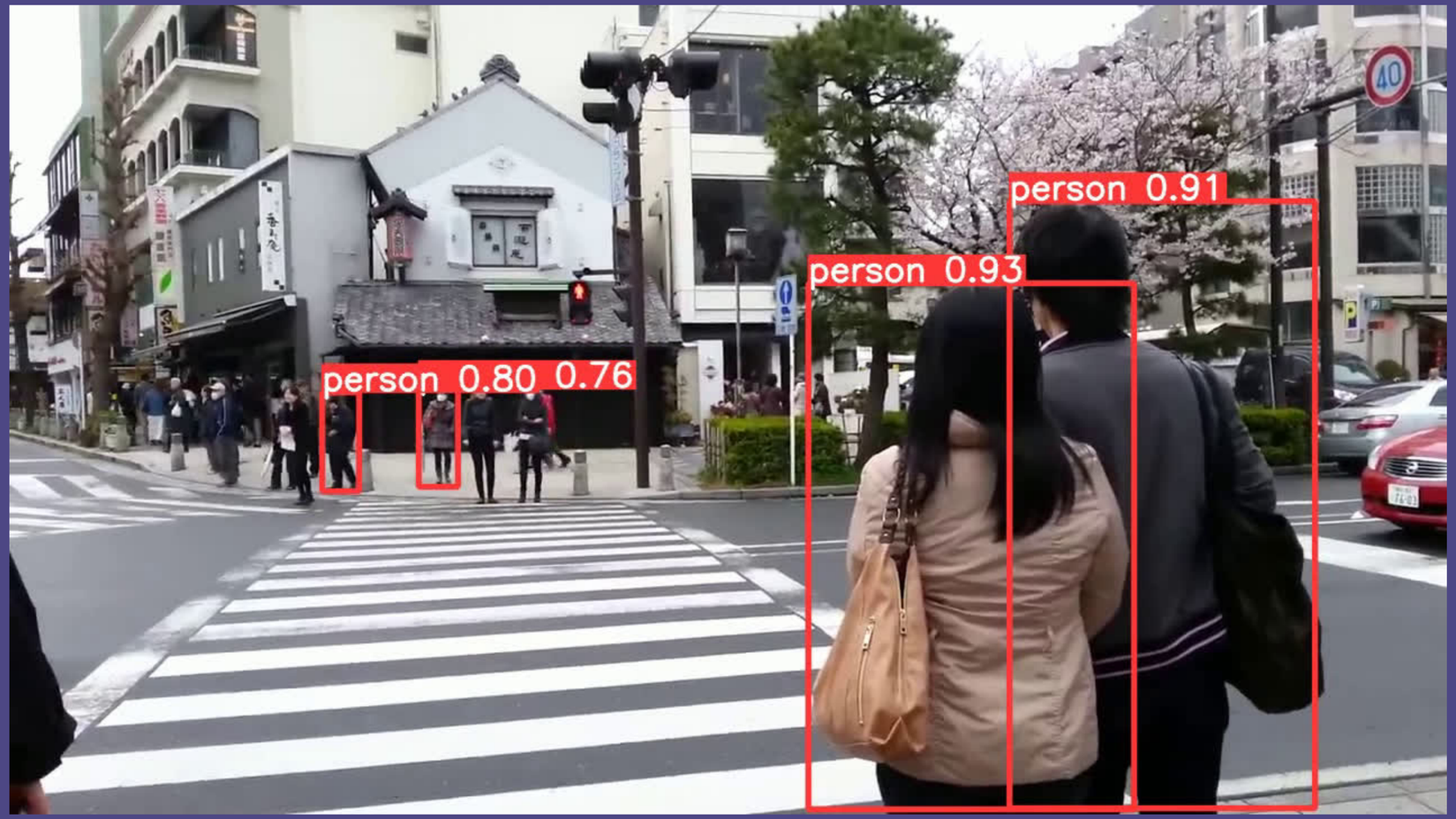
COMPARISON

Model	Precision	Recall	mAP (50)	mAP (50-95)	Inference Time (ms)
Faster R-CNN GoogLeNet	0.21	0.84	0.133	0.133	1652
Faster R-CNN MobileNet	0.97	0.42	0.442	0.225	512.4
Faster R-CNN ResNet50	0.89	0.71	0.756	0.240	143.43
YOLO v3	0.782	0.617	0.714	0.472	1.5
YOLO v5	0.76	0.534	0.67	0.391	1.7
YOLO v8	0.8	0.621	0.732	0.485	1.3

Inference Example
(Faster RCNN-ResNet50)



Inference Example (YOLOV8)



FUTURE IMPROVEMENT

Untuk perbaikan di masa depan, ada beberapa langkah yang dapat diambil:

Penambahan Proses Augmentasi: Dalam upaya untuk terus meningkatkan kemampuan model deteksi objek, salah satu langkah utama yang perlu diambil adalah memperluas data dan memperdalam proses augmentasi data.

Eksplorasi Algoritma Lain: Salah satu pendekatan adalah mencoba variasi algoritma lain yang memiliki kemampuan untuk mendeteksi objek dalam berbagai skala dengan lebih baik.

Menambahkan Algoritma Tracking: Algoritma ini memungkinkan model untuk lebih efisien dalam melacak pergerakan objek dan memberikan informasi yang lebih akurat tentang lokasi dan perilaku objek.

CONCLUSIONS

Berdasarkan komparasi dari model yang kami kerjakan, Faster R-CNN-ResNet50 menonjol sebagai model terbaik dengan keseimbangan optimal antara waktu inferensi yang relatif cepat (143.43 ms) dan kinerja deteksi yang tinggi, ditunjukkan oleh nilai Precision yang baik (0.89), Recall yang memadai (0.71), dan mAP yang cukup tinggi (0.756).

Namun apabila mempertimbangkan kebutuhan deployment secara real-time, YOLOv8 memiliki waktu inferensi yang paling cepat (1.3 ms), performa deteksinya yang baik dengan nilai Precision (0.8), Recall (0.62) dan mAP (0.732) membuatnya menjadi alternatif yang kuat terutama jika prioritas utama adalah responsivitas waktu inferensi yang cepat.

THANK YOU