CHULA ENGINEERING
Innovation toward Sustainability | ACTNOW

CHULA COMPUTER ENGINEERING

OBJECT DETECTION :

AI VEHICLE TRACKER

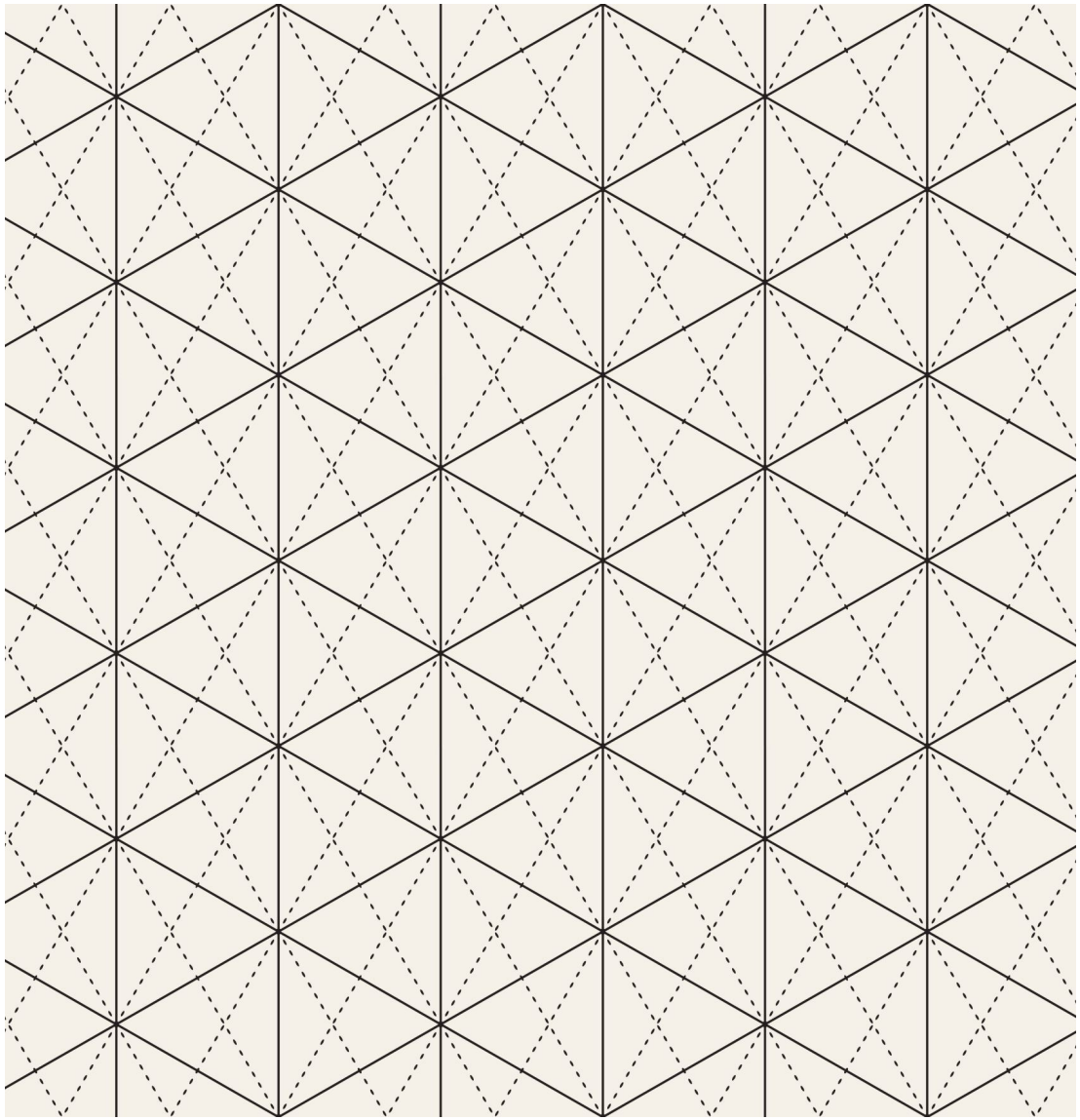
Asst. Prof. Punnarai Siricharoen, PhD

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Punnarai.s@chula.ac.th

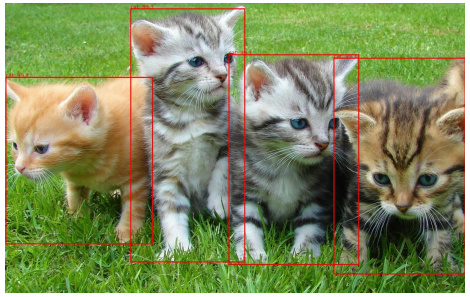


Outline

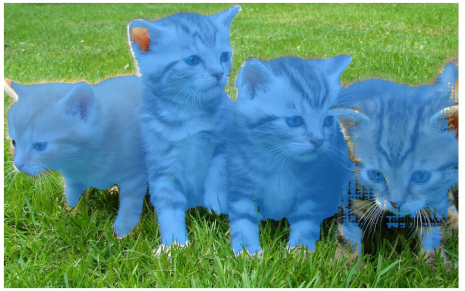
- Object Detection : Evolution
- Deep Learning Models
- Object Detection Models : YOLO
- Tracking Models : ByteTrack
- DEMOs

"Cat"

Classification



Object Detection



Semantic Segmentation

"four kittens on the grass"

Instance Segmentation

Image Captioning

Image Generation

“Cat”

Classification

Features?

Colour is powerful - Color Features: Color Histogram, Mean RGB, HSV, CIELAB, YCbCr color model

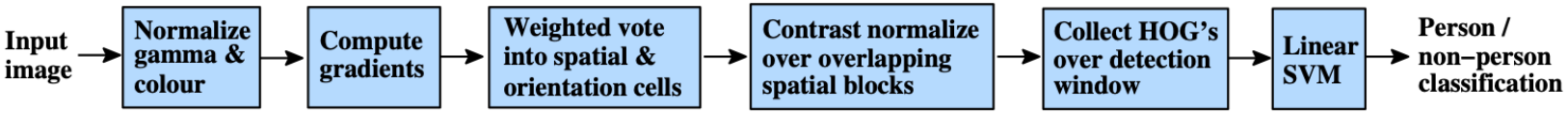
Texture Features: Local Binary Pattern, Co-occurance matrix, Tamura’s textural features, Wavelet Transform

Statistical Features: Mean, Standard Deviation and Variance

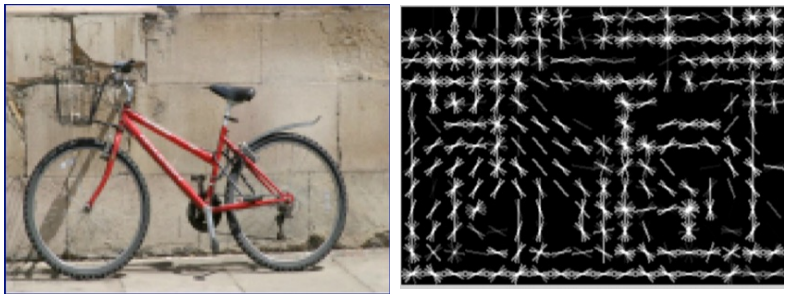
Structural Features: Edge, corner, shape (Region properties)

- Edge detection (human-perception) - First / second derivatives – sobel, prewitt, canny

Features – Histogram of Oriented Gradient (HOG)



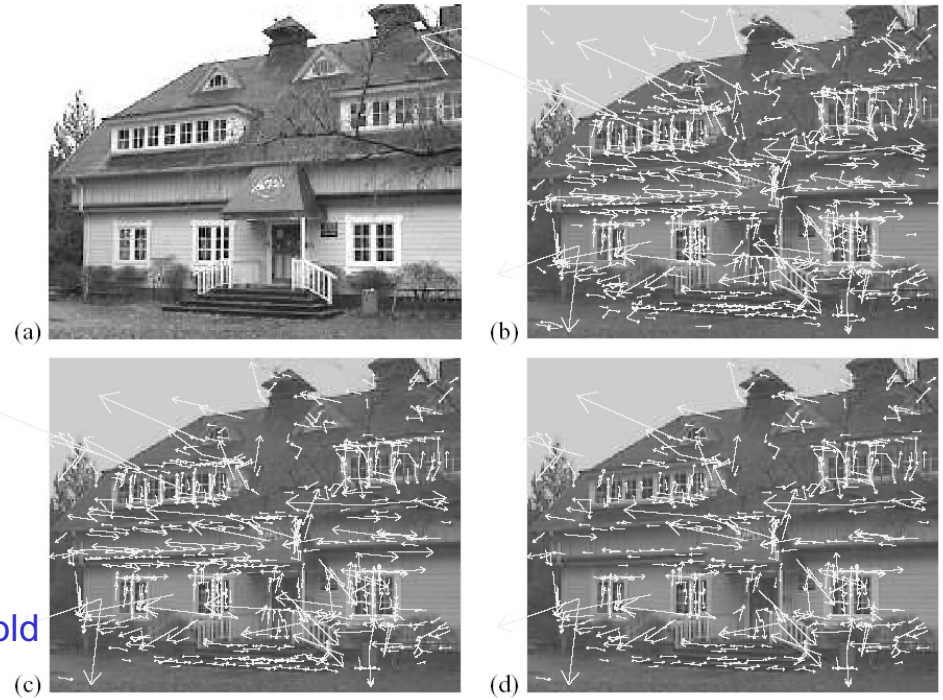
Overview of HOG feature extraction for person / non-person classificaiton (Dalal and Triggs, 2005)



Dalal and Triggs, 2005, Histograms of Oriented Gradients for Human Detection (CVPR'05)

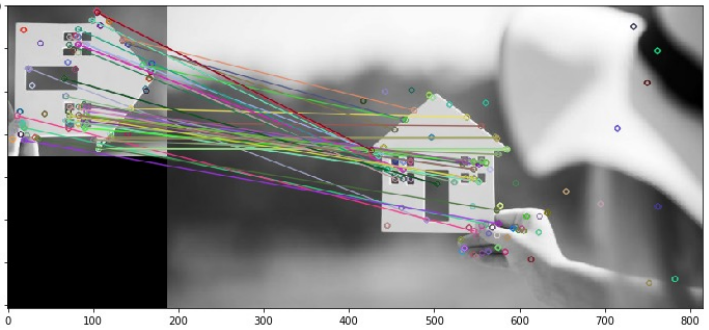
Features – SCALE INVARIANT FEATURE TRANSFORM (SIFT)

233x189



729

keypoints after gradient threshold



832

initial keypoints

536

keypoints after ratio threshold

David Lowe, IJCV 2004

Deep Learning Model

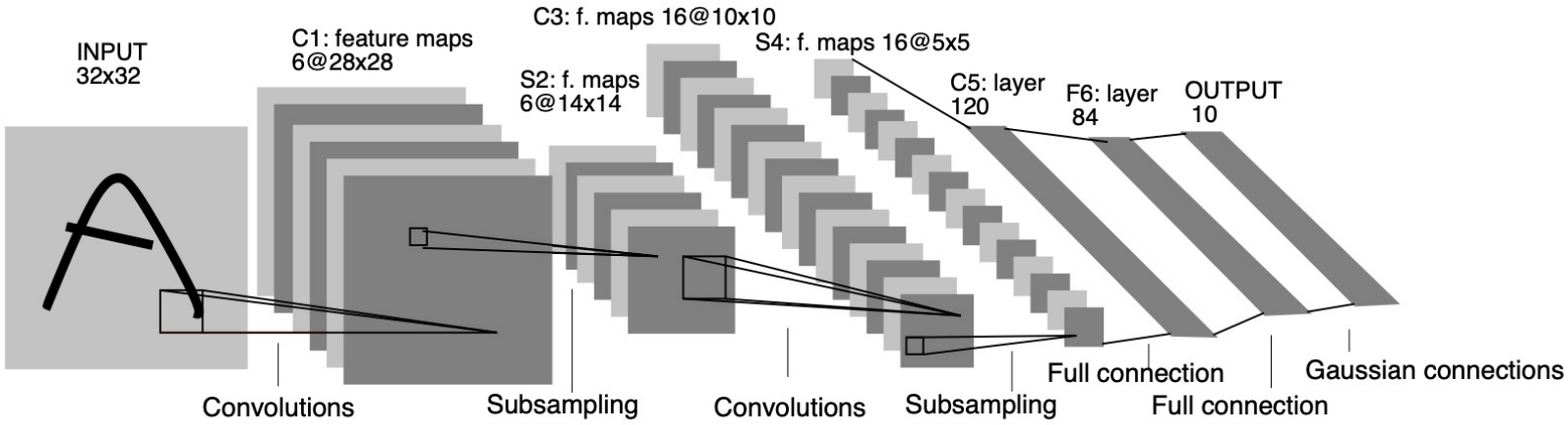


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Léon Bottou, Yoshua Bengio and Patrick Haffner. "Gradient-based learning applied to document recognition." Proc. IEEE 86 (1998): 2278-2324.

Convolutional Neural Networks

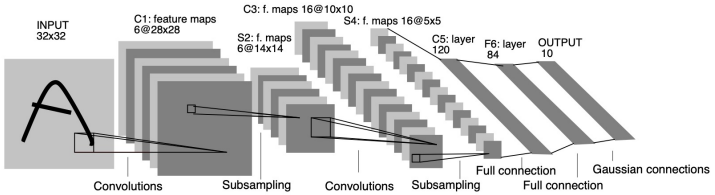
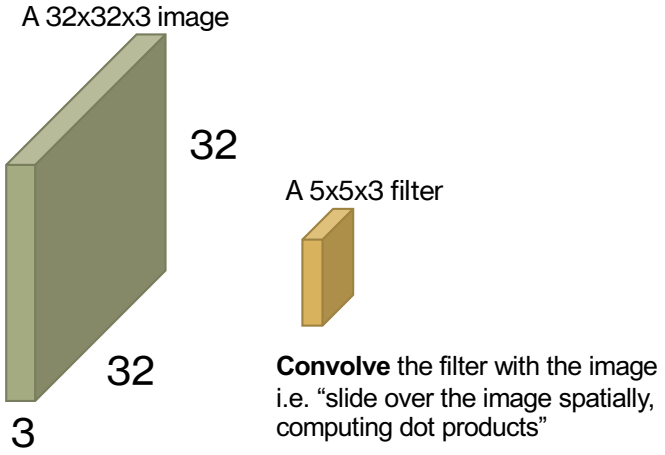


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

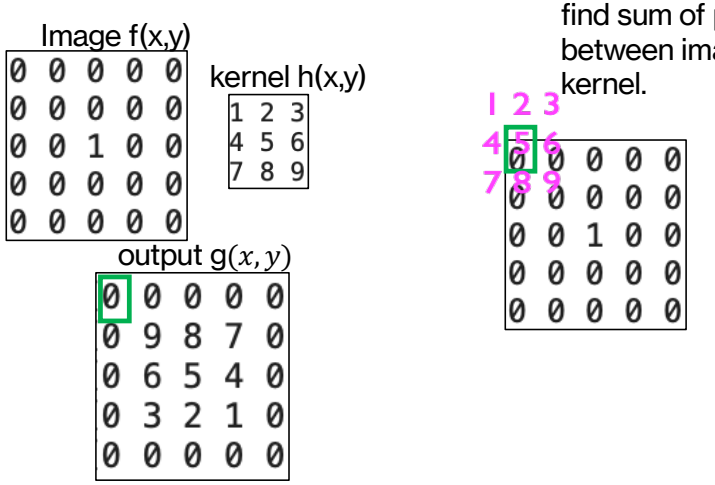
LeCun, Yann, Léon Bottou, Yoshua Bengio and Patrick Haffner. "Gradient-based learning applied to document recognition." Proc. IEEE 86 (1998): 2278-2324.

- **Convolution Layer**
  - To extract spatial features from data, such as edges, textures, or patterns, by applying filters across the input.



- **Filtering (Convolution)**

Sliding the kernel over the image and find sum of products between image and kernel.



Convolutional Neural Networks

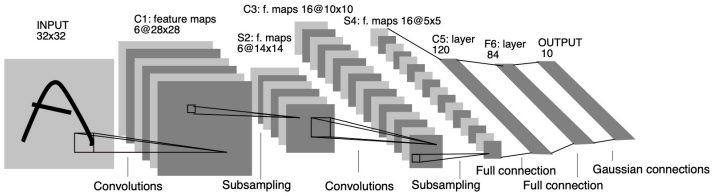
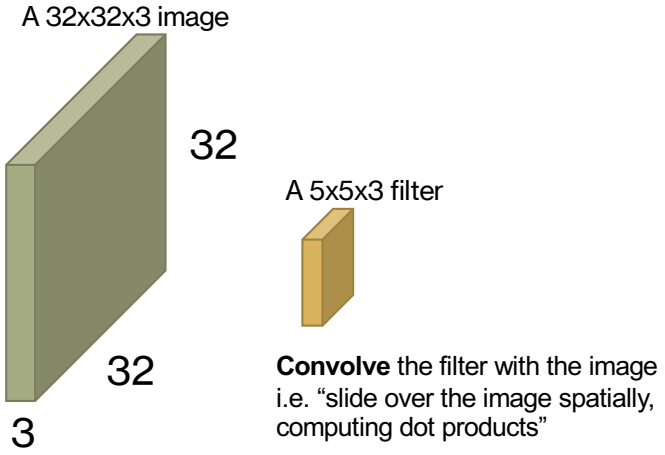
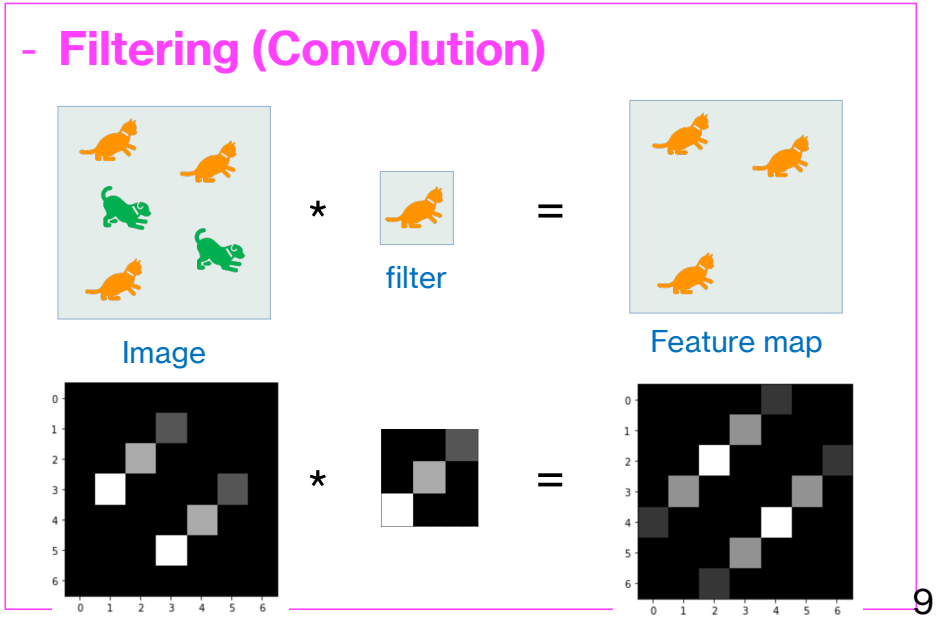


Fig. 1. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

LeCun, Yann, Léon Bottou, Yoshua Bengio and Patrick Haffner. "Gradient-based learning applied to document recognition." Proc. IEEE 86 (1998): 2278-2324.

- **Convolution Layer**
  - To extract spatial features from data, such as edges, textures, or patterns, by applying filters across the input.





Vision Transformer

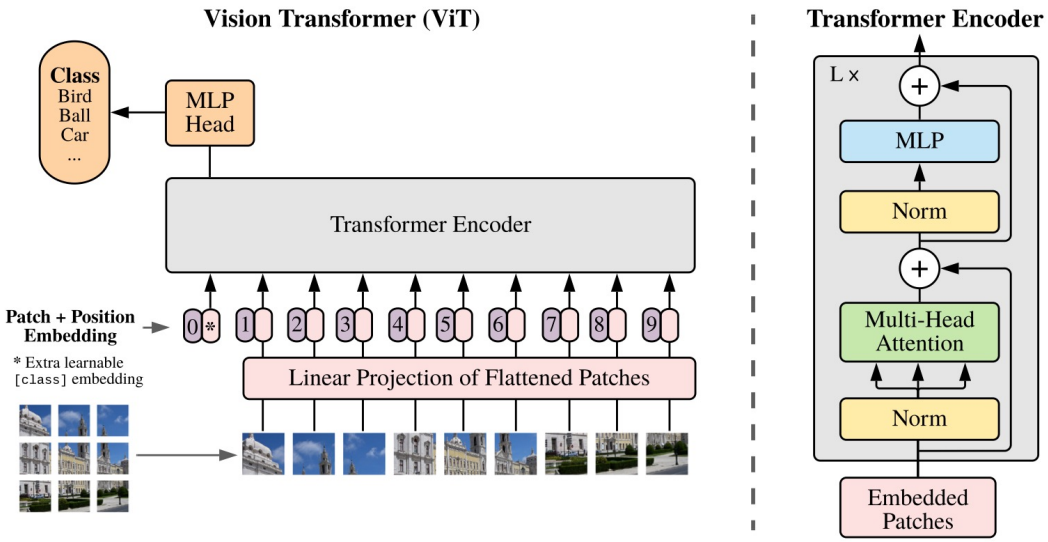


Figure 1: Model overview. We split an image into fixed-size patches, linearly embed each of them, add position embeddings, and feed the resulting sequence of vectors to a standard Transformer encoder. In order to perform classification, we use the standard approach of adding an extra learnable "classification token" to the sequence. The illustration of the Transformer encoder was inspired by Vaswani et al. (2017).

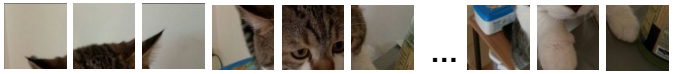
- For vision transformer, an input image is split into a set of image patches, called *visual tokens*.
- Position embeddings are added to the patch embeddings to retain positional information.

Attention is all you need, Vaswani A., et al. (2021)

Evolution of Deep learning: Vision Transformer (2021)

- Positional embedding – model knows where each patched would be placed.

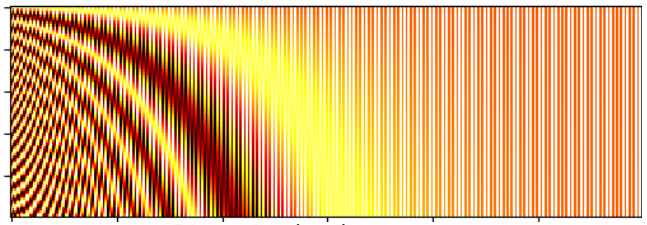
196, 768 (selected #feature)



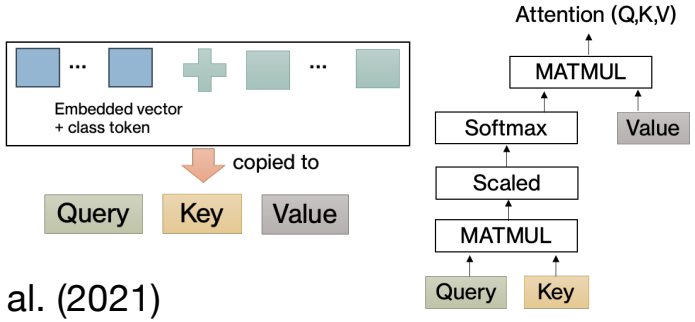
196, 768

Positional Encoding

$$PE_{pos,j} = \begin{cases} \sin(pos/10000^{2*j/dim}), j\ even \\ \cos(pos/10000^{2*j-1/dim}), j\ odd \end{cases}$$



- Transformer Encoder = Layer Norm + Multi-Head Self Attention
  - Multi-Head Self-Attention

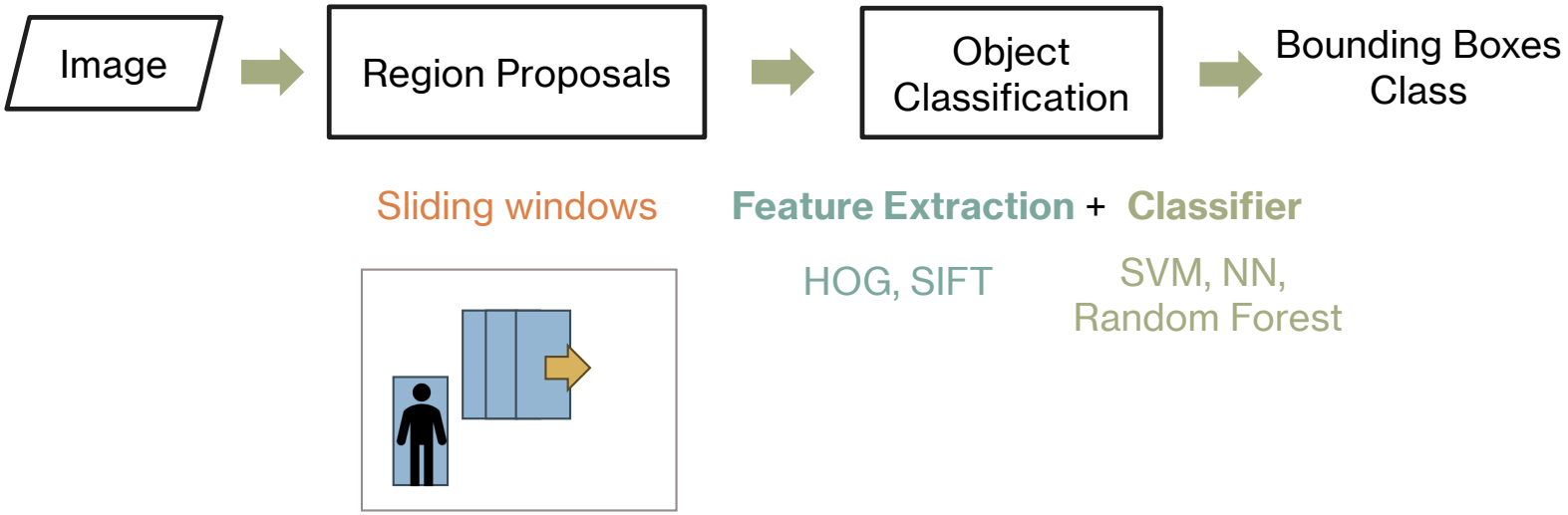


$$Attention\ (Q,K,V) = softmax(\frac{Q \cdot K^T}{\sqrt{dim}} \cdot V$$

Attention is all you need, Vaswani A., et al. (2021)

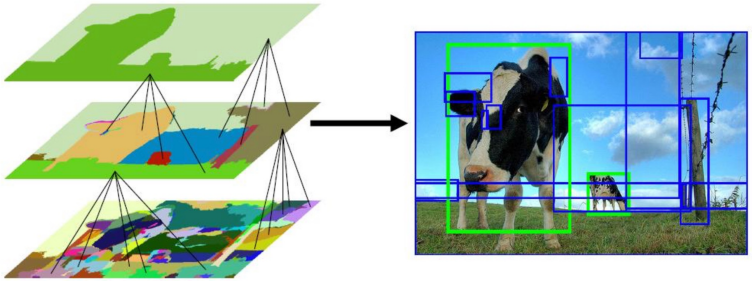
Object Detection Models

- Two-stage Detectors เช่น Faster-RCNN

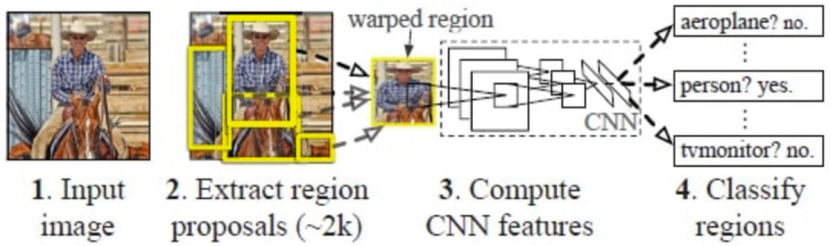


Object Detection Models

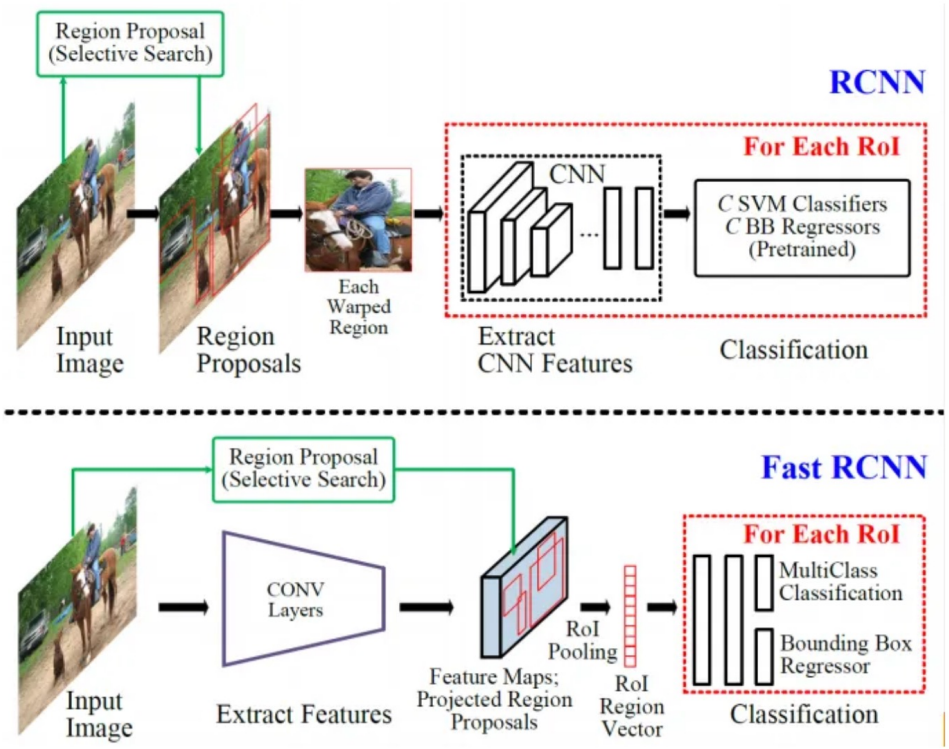
- Two-stage Detectors เช่น Faster-RCNN



Selective Search for Object Recognition
J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders
In International Journal of Computer Vision 2013.



Architecture of RCNN
Rich feature hierarchies for accurate object detection and semantic segmentation
Ross Girshick, Jeff Donahue, Trevor Darrell, Jitendra Malik



RCNN vs Fast-RCNN
(source: Deep Learning for Generic Object Detection: A Survey)

Object Detection Models

- Two-stage Detectors เช่น Faster-RCNN

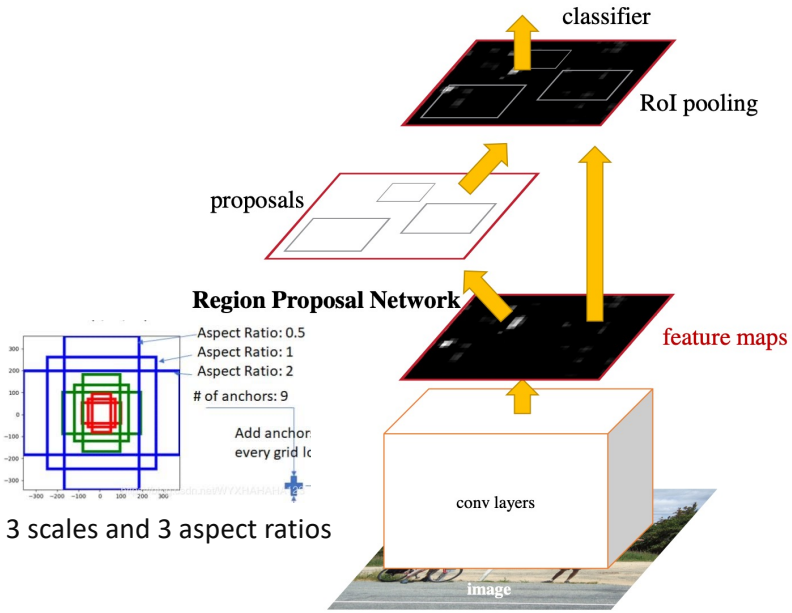


Figure 2: Faster R-CNN is a single, unified network

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks
Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun

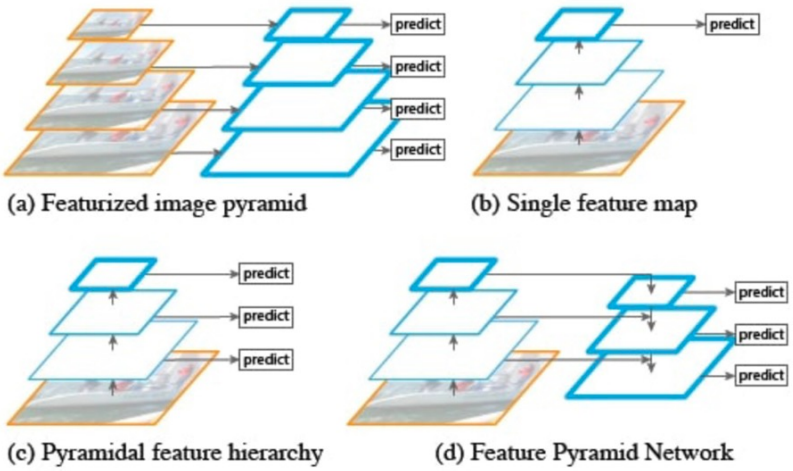


Fig. 5. Feature Pyramid Network [28]. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Feature Pyramid Networks for Object Detection
Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, Serge Belongie

Object Detection Models

- Two-stage Detectors เช่น Faster-RCNN

  - แม่นยำสูง แต่ค่อนข้างช้า
- One-stage Detectors เช่น YOLO, DETR

  - เร็วมาก เหมาะกับการใช้งานแบบเรียลไทม์ เช่นในมือถือหรือกล้อง

Object Detection

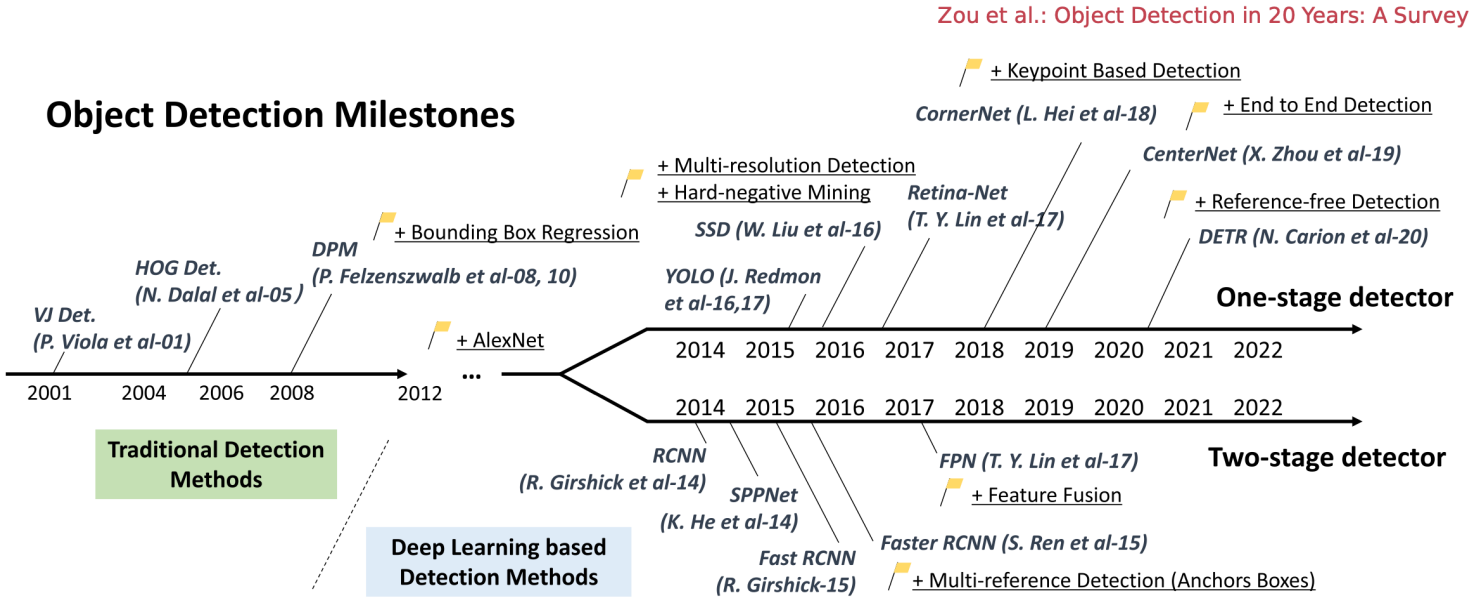
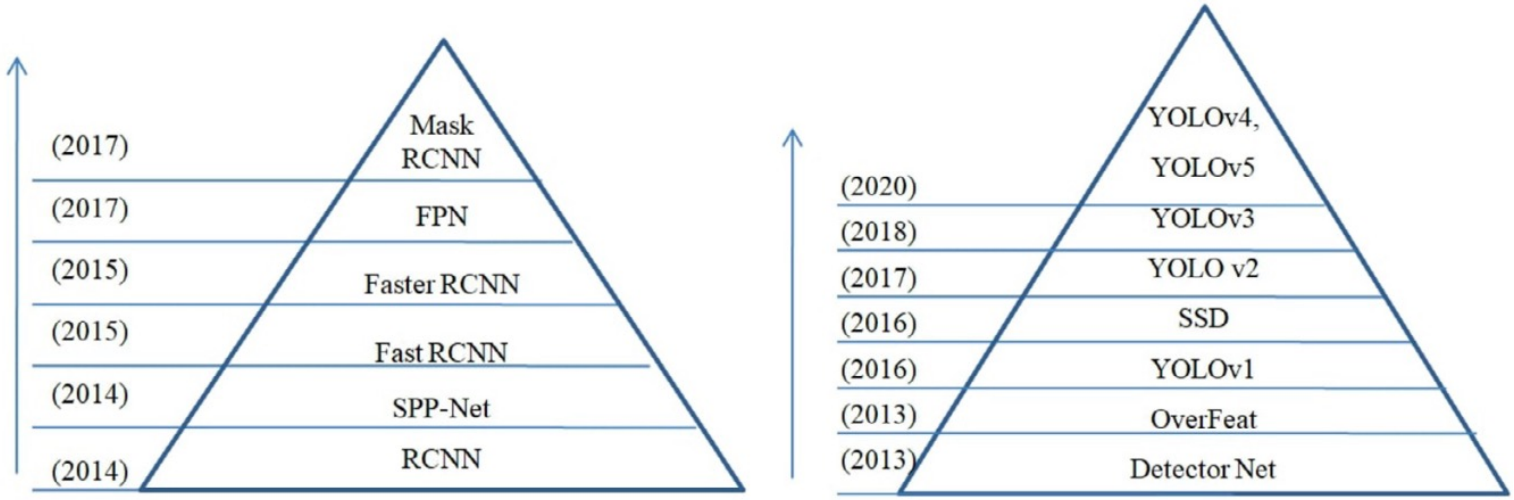


Fig. 2. ***Road map of object detection. Milestone detectors in this figure: VJ Det. [10], [11], HOG Det. [12], DPM [13], [14], [15], RCNN [16], SPPNet [17], Fast RCNN [18], Faster RCNN [19], YOLO [20], [21], [22], SSD [23], FPN [24], Retina-Net [25], CornerNet [26], CenterNet [27], and DETR [28].***

Object Detection



(a) Two-Stage Object detector

(b) One-Stage Object detector

Kaur & Singh (2023), A comprehensive review of object detection with deep learning, Digital Signal Processing

COCO Datasets

- https://cocodataset.org

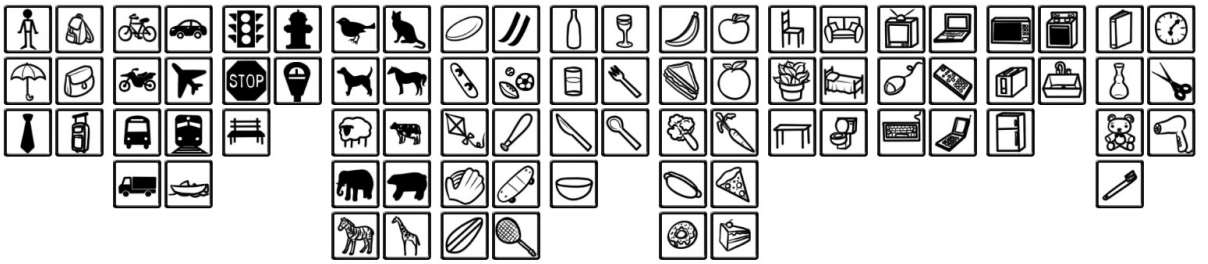
What is COCO?

COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- **Object segmentation**
- **Recognition in context**
- **Superpixel stuff segmentation**
- **330K images (>200K labeled)**
- **1.5 million object instances**
- **80 object categories**
- **91 stuff categories**
- **5 captions per image**
- **250,000 people with keypoints**

COCO Explorer

COCO 2017 train/val browser (123,287 images, 886,284 instances). Crowd labels not shown.

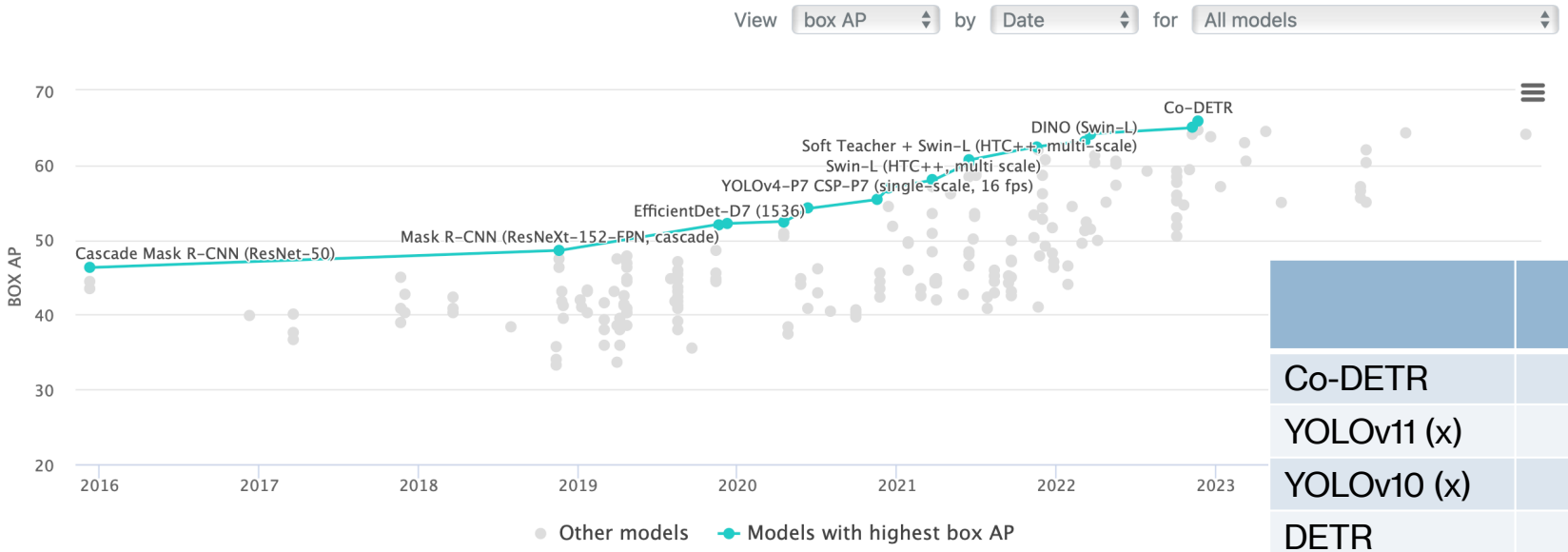


search

DETR / CO-DETR

Object Detection on COCO minival

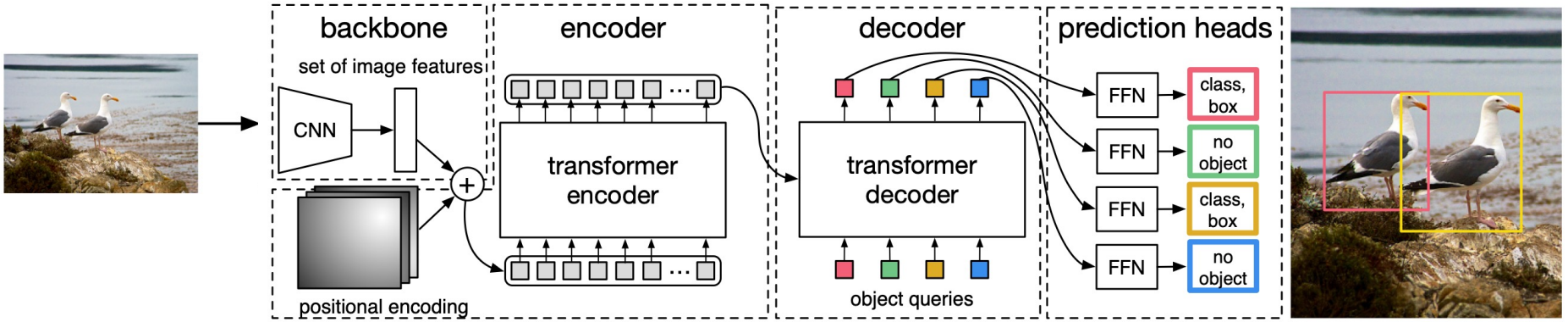
Leaderboard Dataset



	Box mAP (COCO Val)
Co-DETR	60.7
YOLOv11 (x)	54.7
YOLOv10 (x)	54.4
DETR	42.0

https://paperswithcode.com

Object Detection : DEtection TRanformer (DETR)



DEtection TRanformer (DETR)

N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, “End-to-End Object Detection with Transformers,” May 2020, [Online]. Available: http://arxiv.org/abs/2005.12872

DETR / CO-DETR

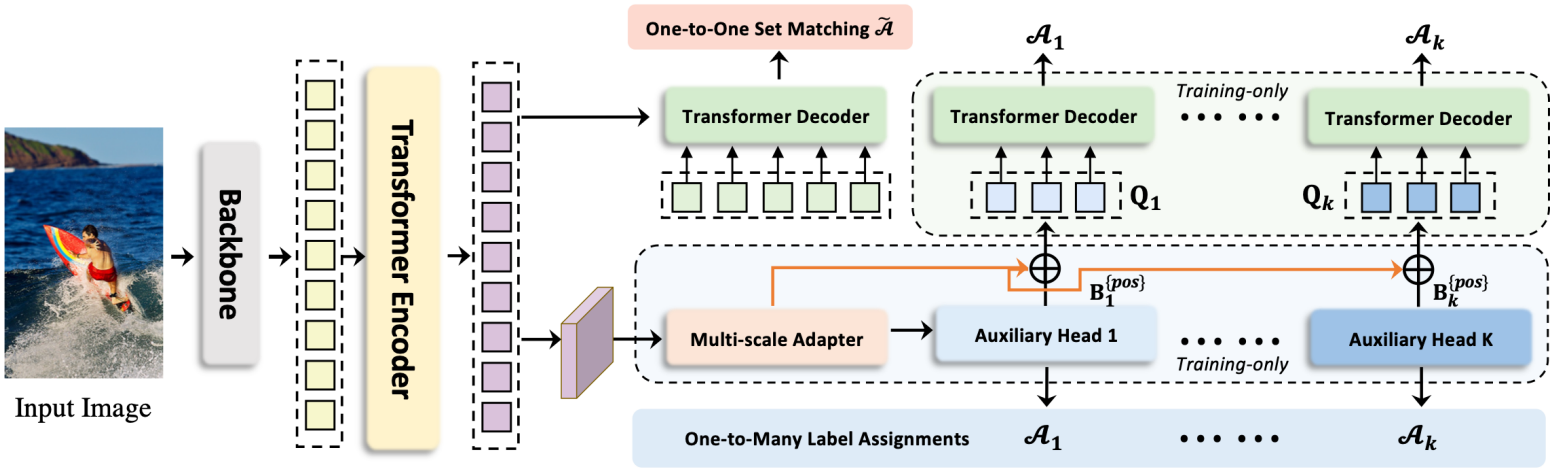
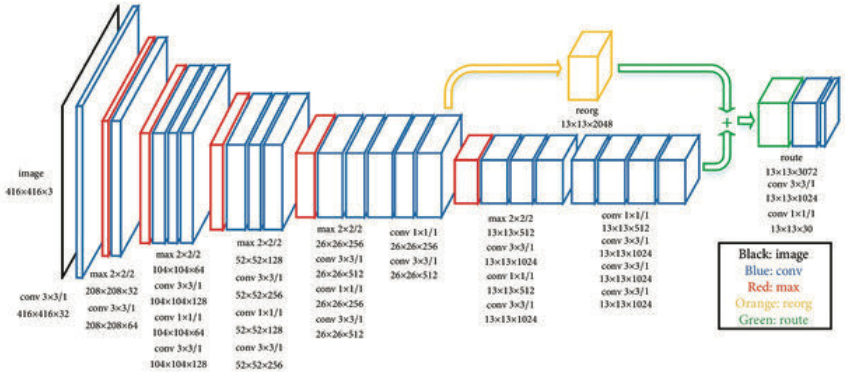


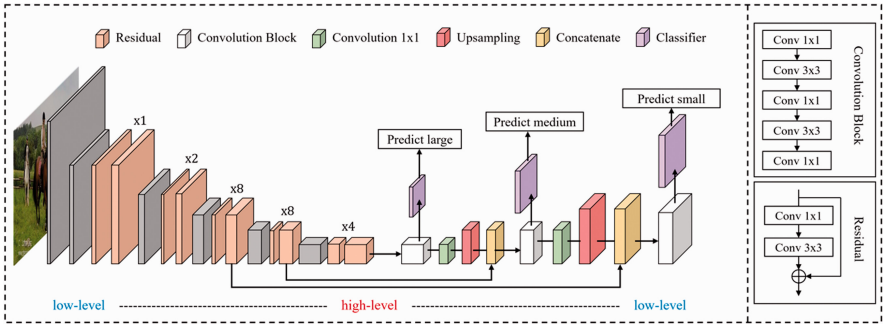
Figure 4. **Framework of our Collaborative Hybrid Assignment Training.** The auxiliary branches are discarded during evaluation.

DETRs with Collaborative Hybrid Assignments Training
Zhuofan Zong Guanglu Song Yu Liu* SenseTime Research

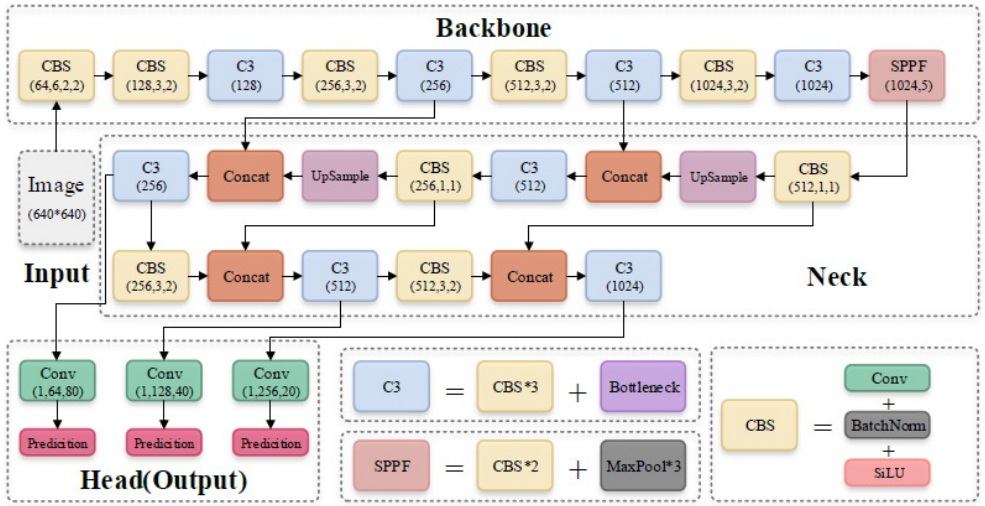
You Only Look Once (YOLO)



An Efficient Pedestrian Detection Method Based on YOLOv2 - Scientific Figu on ResearchGate. Available from: https://www.researchgate.net/figure/The-network-architecture-of-YOLOv2_fig2_330029907 [accessed 8 Nov 2024]

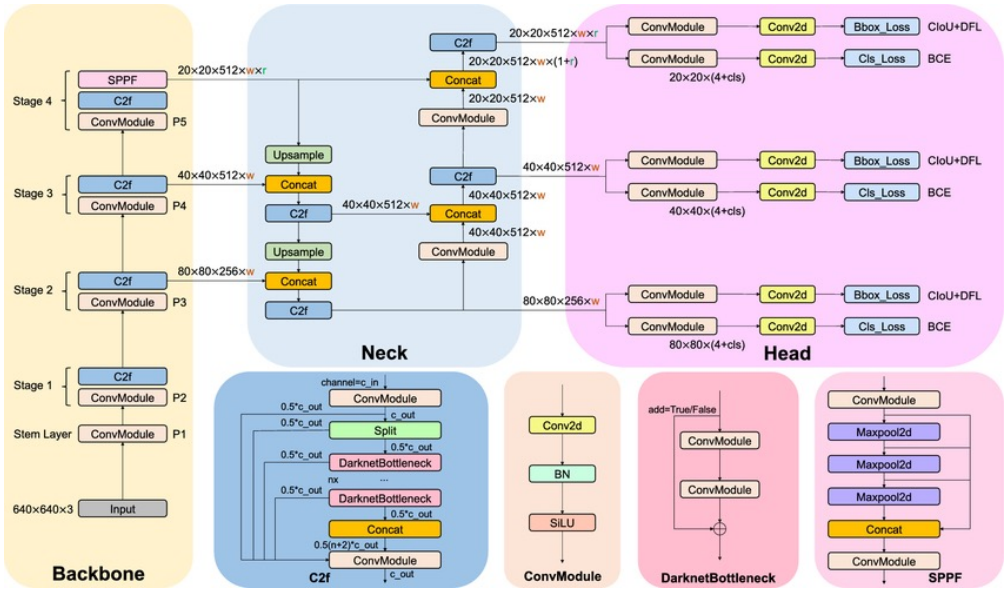


YOLOv3



https://sh-tsang.medium.com/brief-review-yolov5-for-object-detection-84cc6c6a0e3a

You Only Look Once (YOLO)

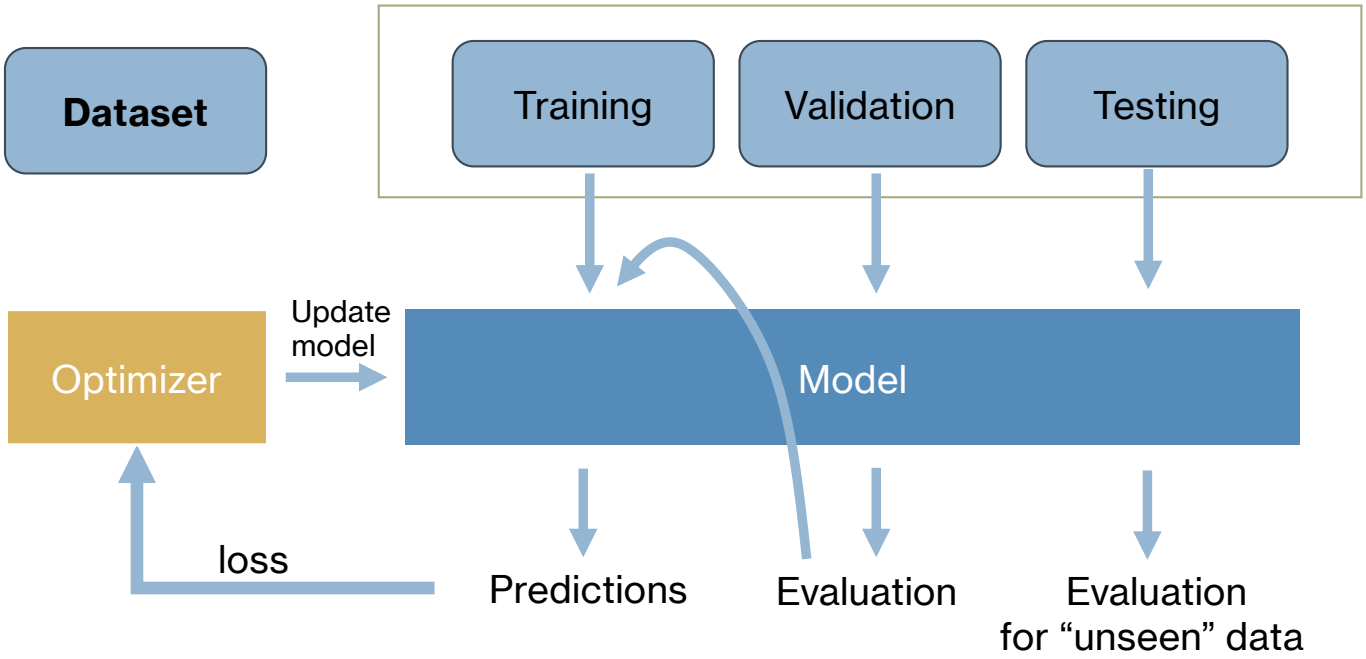


Anchor-free detection approach

Implementation of Slicing Aided Hyper Inference (SAHI) in YOLOv8 to Counting Oil Palm Trees Using High-Resolution Aerial Imagery Data - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/YOLOv8-model-architecture-20_fig2_382742087

- YOLOv8-11
  - Detection, Instance Segmentation, Keypoint Detection
- YOLOv11
  - Better Handling Small Objects
- YOLOv12
  - Attention-Centric Architecture (maintain speed)

Training Your Model



Loss functions

	Two-stage	One-stage
Stage-wise loss separation	✓	
Focus on high precision	✓	focus on speed
Has proposal + refinement	✓	
Detailed loss components	✓	✓ (single pass)

- Two-stage model loss

  -> Region proposal network Loss (Classification Loss + Regression Loss)

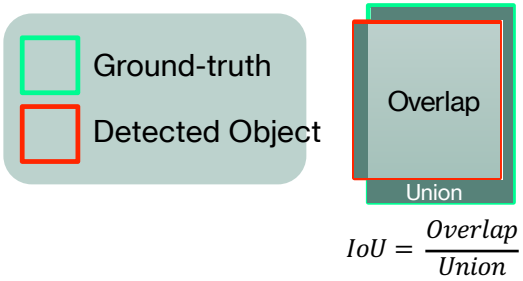
  -> RoI Head Loss (Classification Loss + Regression Loss)

- One-stage model loss for a single pass

  -> Classification Loss + Objectness Loss + Localization Loss

  -> YOLOv10-11 – No Objectness Loss, Classfication Loss + Localization Loss (Bbox + DFL)

Evaluation for Object Detection



$$IoU = \frac{Overlap}{Union}$$

mAP – Mean Average Precision

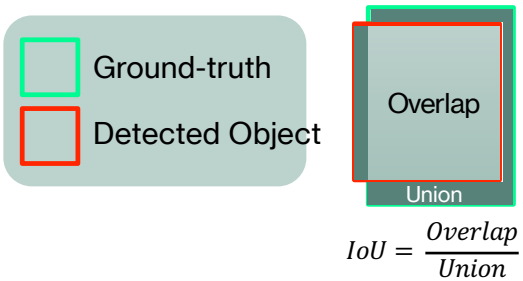
mAP50 – The prediction is correct, when IoU ≥ 0.5.

1) Rank the confidence score of the detected objects in the dataset
2) Calculate Precision / Recall for each of the object

Rank	IoU	Correct?	Precision	Recall
1	0.7	True	1.0	0.25
2	0.6	True	1.0	0.50 ↑
3	0.4	False	0.67 ↓	0.5
4	0.4	False	0.5 ↓	0.5
5	0.5	True	0.6 ↑	0.75↑
6	0.6	True	0.67 ↑	1.0↑

Rank 1:
precision = TP/(TP+FP) = 1/(1+0)
Recall = TP/(ALL GTs) = ¼ = 0.25
Rank 3:
Precision = 2/3 = 0.67
Recall = 2/4 = 0.5
Rank 5:
Precision = 3/5 = 0.6
Recall = ¾ = 0.75

Evaluation for Object Detection

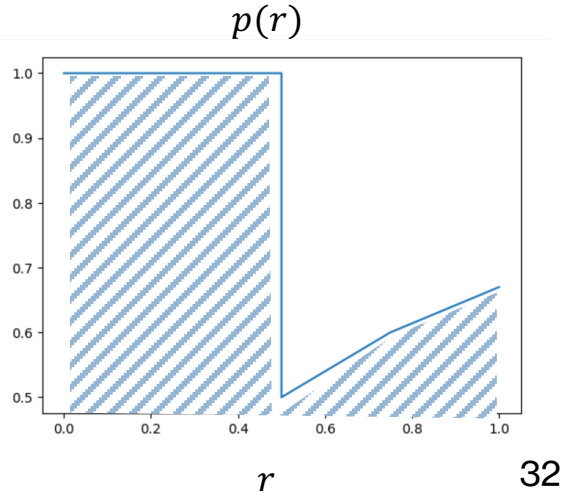


$$IoU = \frac{Overlap}{Union}$$

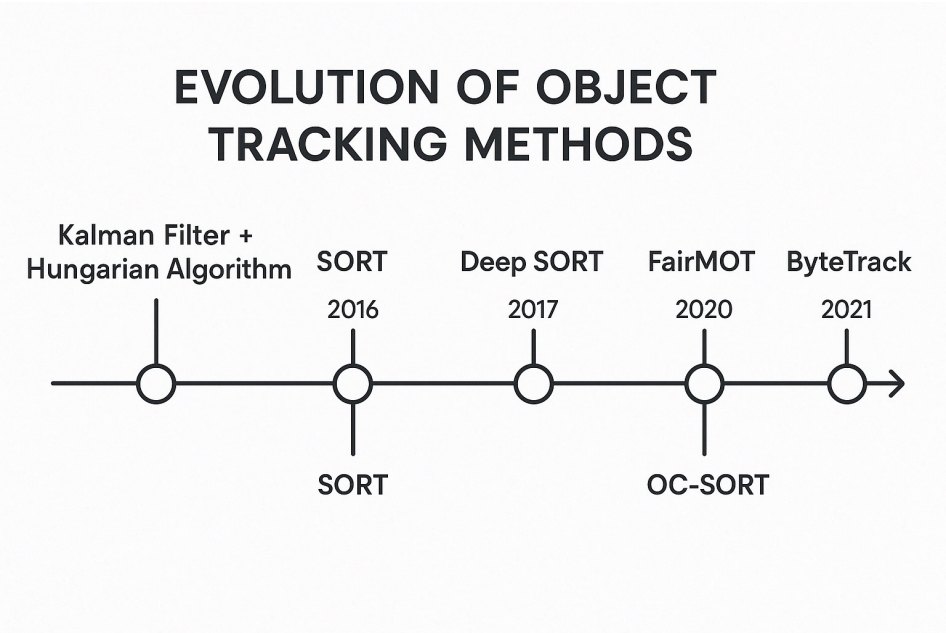
mAP – Mean Average Precision

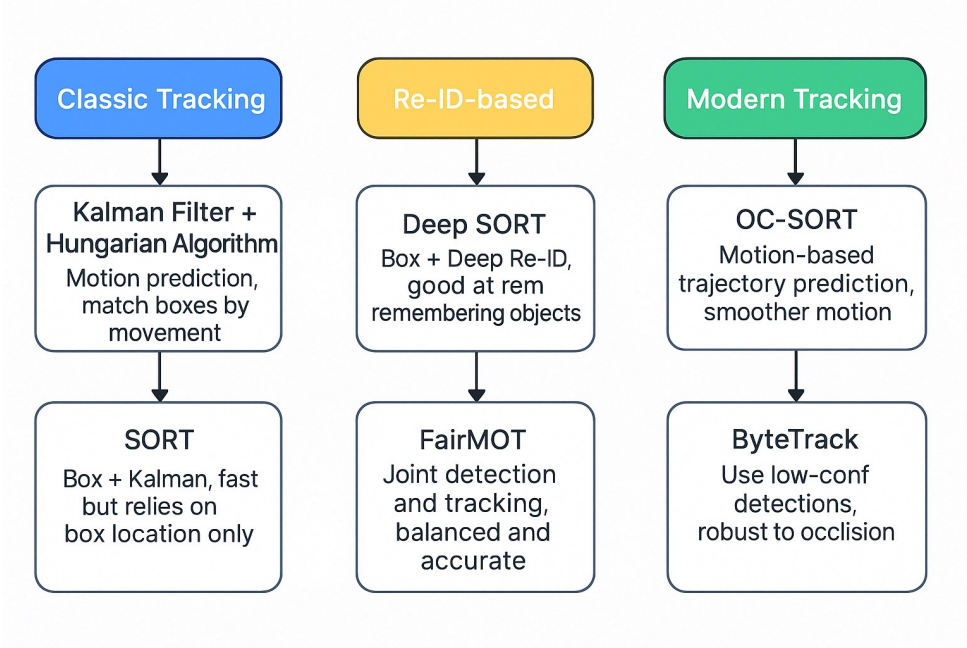
mAP50 – The prediction is correct, when IoU $\geq$ 0.5.

1) Rank the confidence score of the detected objects.
2) Calculate Precision / Recall for each of the object
3) Plot PR curve $p(r)$
4) AP for each class = $\int_0^1 p(r)dr$
   - 11-point interpolation
   - mAP - average over classes
- mAP50:90 – average over 10 IoUs



Object Tracking

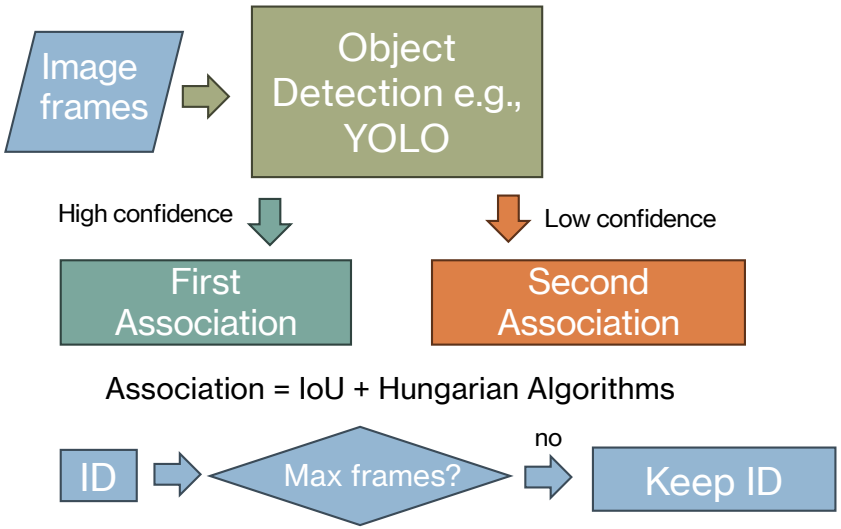


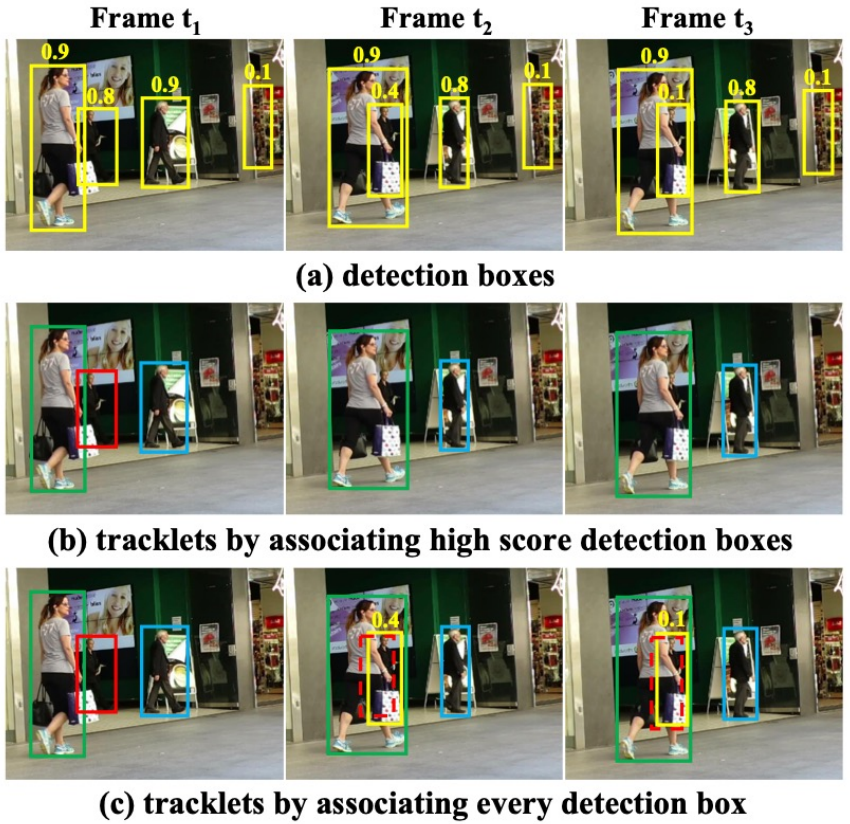


MOTA (Multiple Object Tracking Accuracy)

$$\mathrm{MOTA} = 1 - \frac{\mathrm{FP} + \mathrm{NP} + \mathrm{IDSW}}{\mathrm{GT}}$$

Object Tracking : ByteTrack

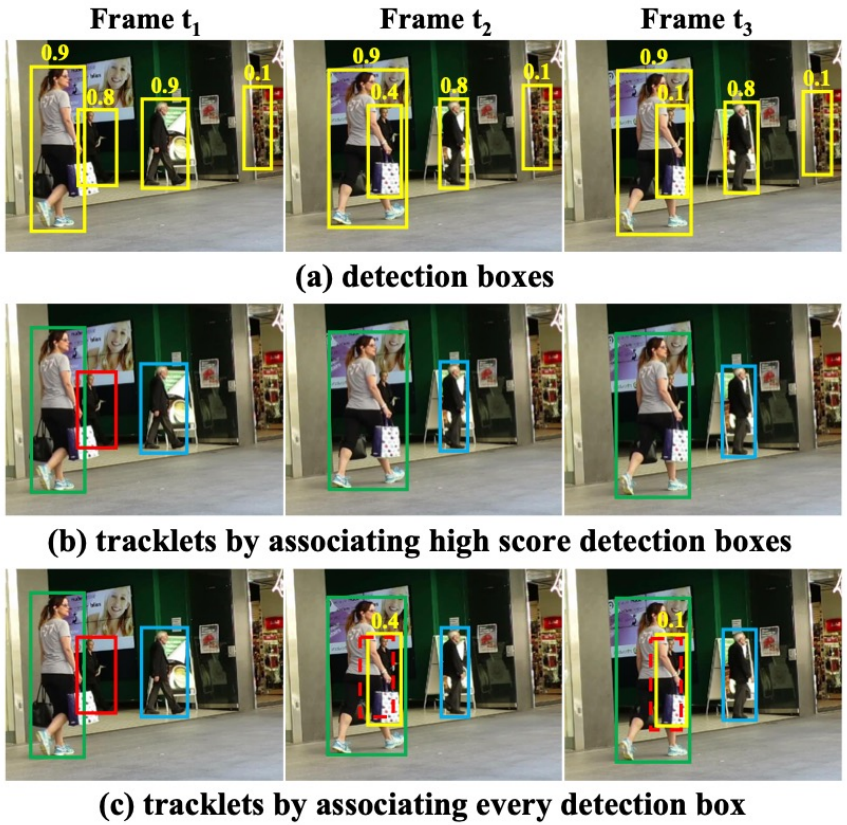


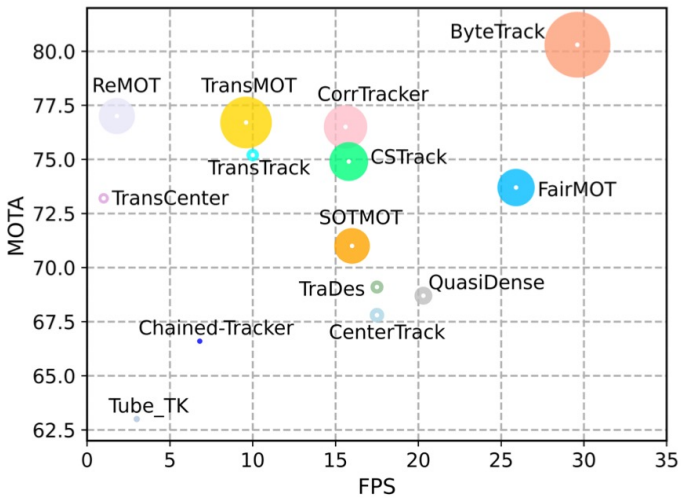


- track_high_thresh: Minimum score for high-confidence detection (first matching)
- track_low_thresh: Minimum score for low-confidence detection (second matching)
- match_thresh: Minimum IoU to associate detection with a track
- track_buffer: Max frames to keep unmatched tracks before deletion
- min_box_area: Ignore boxes smaller than this area
- mot20: Use settings optimized for dense scenes
- fuse_score: Final score = objectness × class confidence

Y. Zhang *et al.*, "ByteTrack: Multi-object Tracking by Associating Every Detection Box," in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, Berlin, Heidelberg: Springer-Verlag, 2022, pp. 1–21. doi: 10.1007/978-3-031-20047-2_1.

Object Tracking : ByteTrack





MOTA (Multiple Object Tracking Accuracy)

$$\mathrm{MOTA} = 1 - \frac{\mathrm{FP} + \mathrm{NP} + \mathrm{IDSW}}{\mathrm{GT}}$$

Y. Zhang *et al.*, "ByteTrack: Multi-object Tracking by Associating Every Detection Box," in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, Berlin, Heidelberg: Springer-Verlag, 2022, pp. 1–21. doi: 10.1007/978-3-031-20047-2_1.

DEMOs

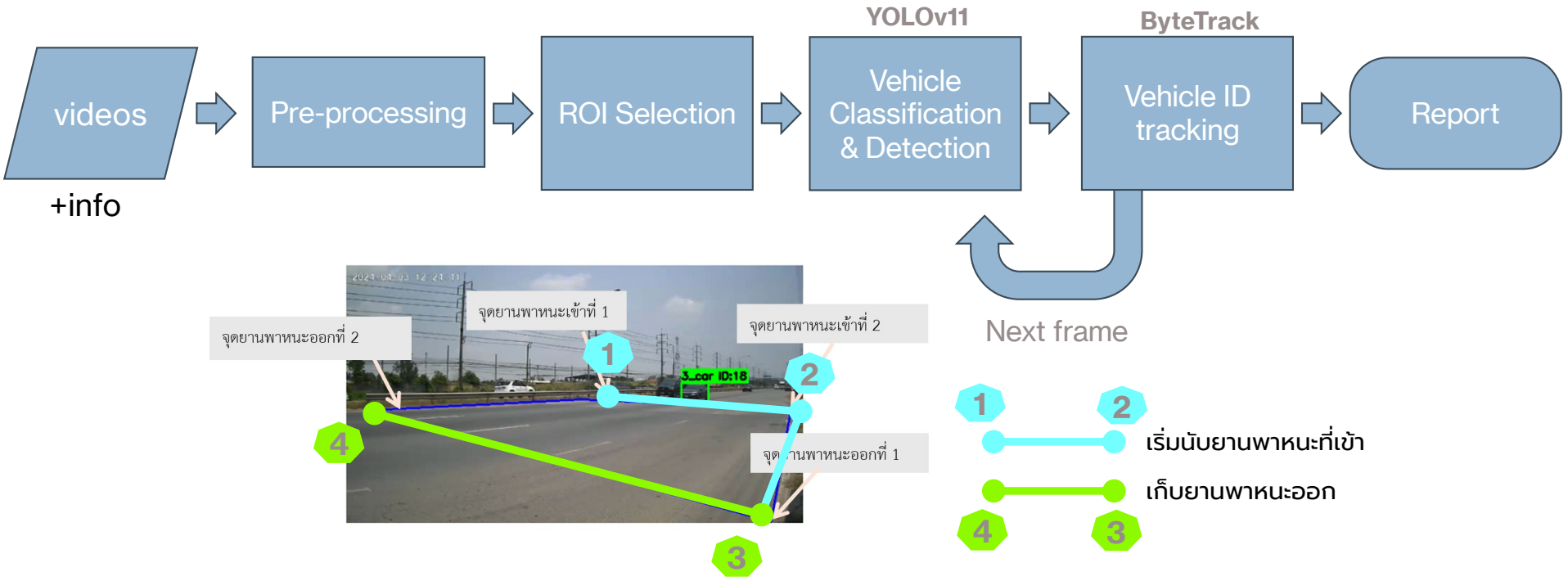
- **Demo 1 : YOLO - Cats**
- https://colab.research.google.com/drive/11YoWKLkZUNHmJllAPfVajyJ6o6R6znXf?usp=sharing

- **Demo 2 : YOLO - ByteTrack**
- https://colab.research.google.com/drive/1o5fnniqV1FVKkGvLD2u0zkWK8EHcfNUg?usp=sharing

- **Demo 3 : YOLO - ByteTrack - Vehicle AI Tracker**
- https://colab.research.google.com/drive/1ndn1gPaggyG4TToluezKBDiU0tYZGkf3?usp=sharing

- **Demo 4 : Vehicle AI Tracker : CapCapCar Year 2**
- https://capcapcar.piney.me
- Video ตัวอย่าง เลือกสั้น ๆ : https://youtu.be/Y1jTEyb3wiI?si=7u5cbXyLqOjRA9BQ

VEHICLE AI TRACKER



DEMO : VEHICLE AI TRACKER