King Mongkut's University of Technology Thonburi

Real-Time Object Detection

Lecturer: Dr. Thittaporn Ganokratanaa

ตรวจจับวัตถุแบบrealtime
apply ใน image

❖ Problem Addressed: Object Detection

- ต้องแม่นยำในการระบุตำแหน่ง & classification
- เวลาที่ใช้ในการประมวลผล → ยอดเยี่ยม (runing time)

- Object detection is the problem of both locating AND classifying objects
- Goal of object detection algorithm is to do object detection both fast AND with high accuracy.

ทำให้คอมรู้ว่าในภาพมีอะไรบ้าง

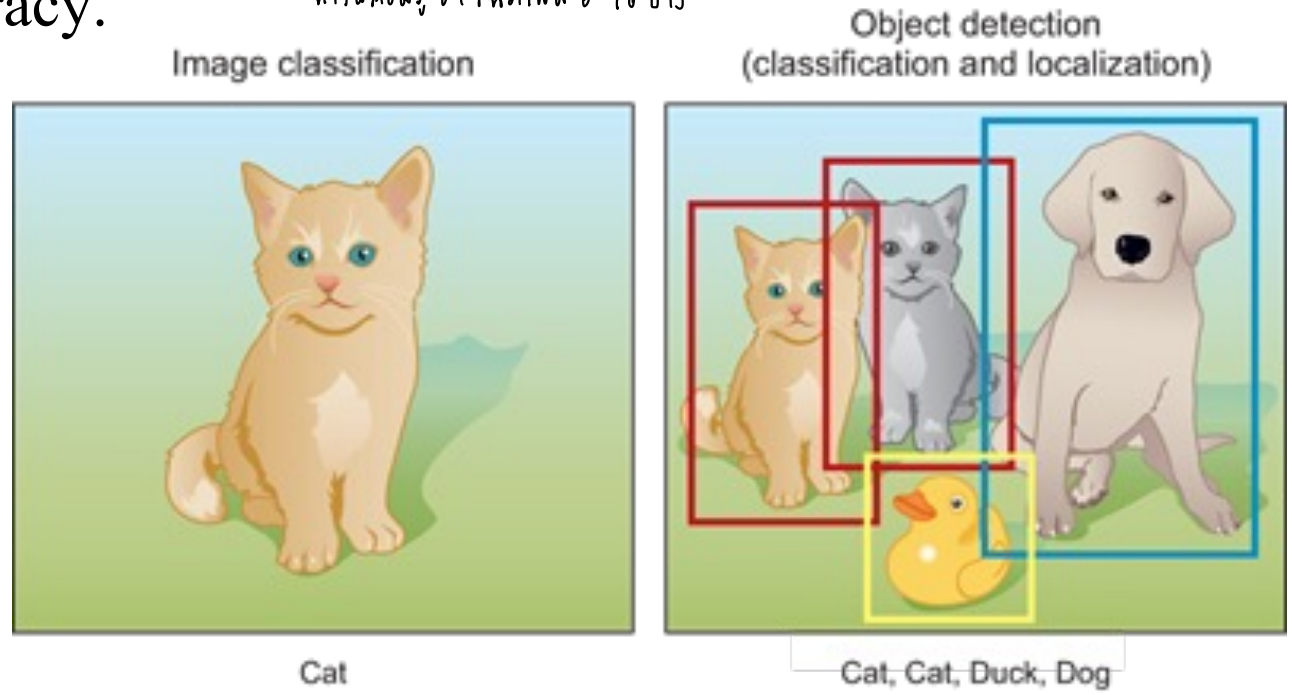


Image Source Object Detection

King Mongkut's University of Technology Thonburi

YOLO คือ input เป็นภาพ

❖ Importance of Object Detection

- Visual modality is very powerful
- Humans are able to detect objects and do perception using just this modality in real-time (not needing radar)
- If we want responsive robot systems that work real-time (without specialized sensors), almost real-time vision based object detection can help greatly.

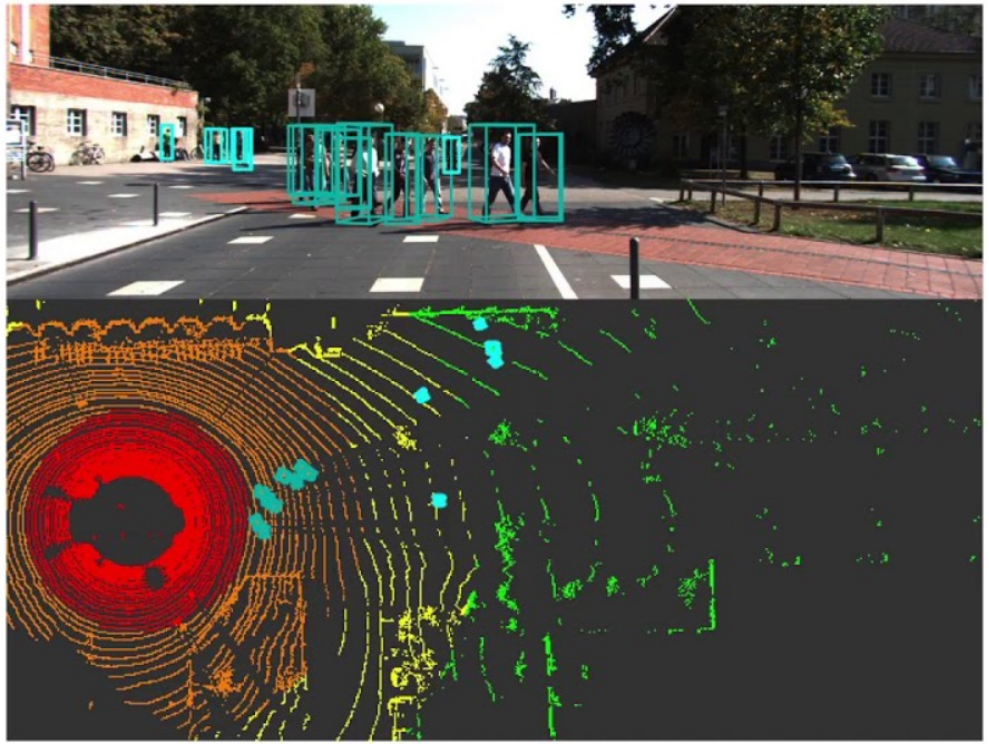
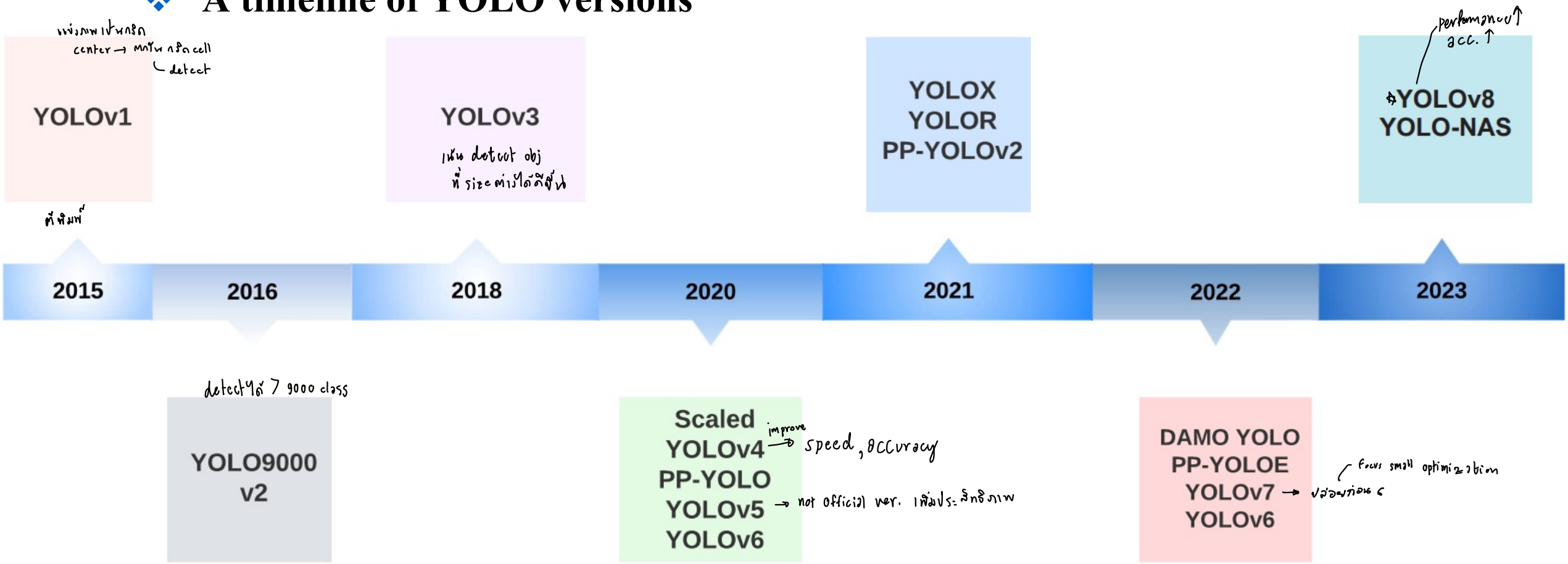


Image Source: Object Detection and Lidar

King Mongkut's University of Technology Thonburi

A timeline of YOLO versions

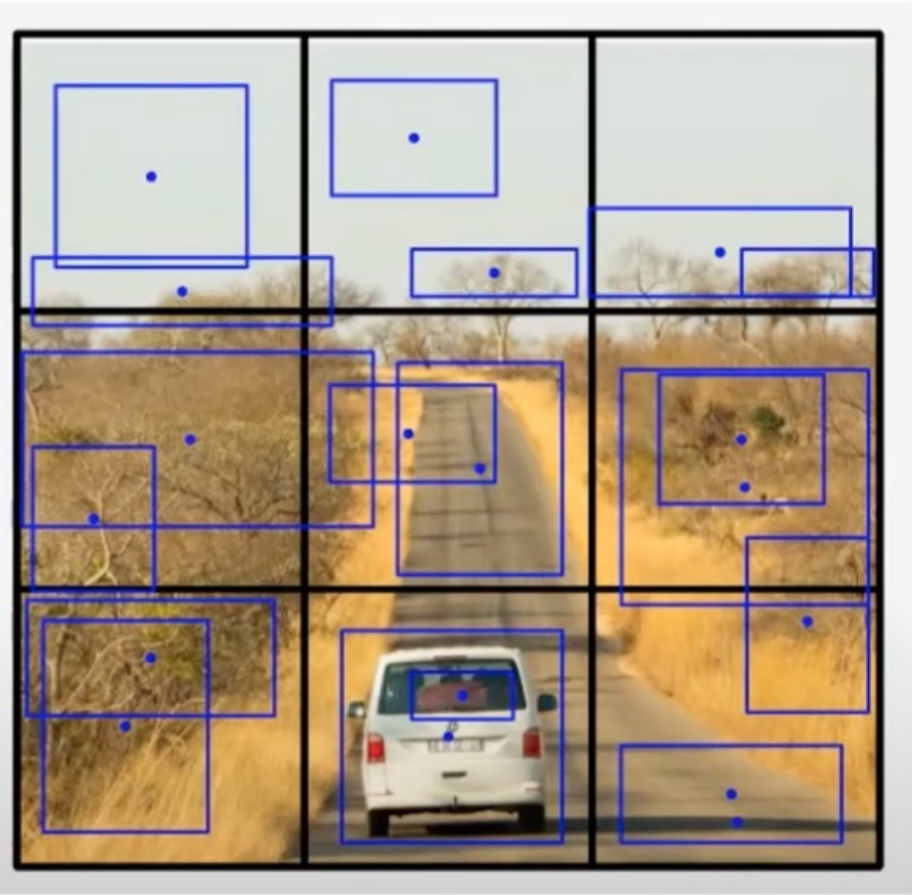


Source: YOLO

YOLO Overview

- First, image is split into a S×S grid
- For each grid square, generate B bounding boxes
- For each bounding box, there are 5 predictions:

x, y, w, h, confidence



S = 3, B = 2

Source: YOLO

King Mongkut's University of Technology Thonburi

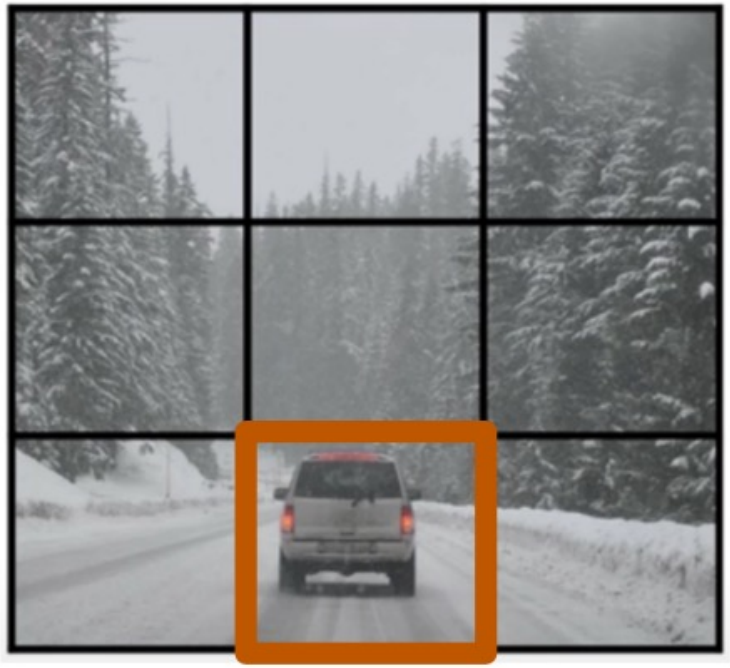
YOLO Training

- YOLO is a regression algorithm. What is X? What is Y?
- X is simple, just an image width (in pixels) * height (in pixels) * RGB values
- Y is a tensor of size S * S * (B * 5 + C)
- B*5 + C term represents the predictions + class predicted distribution for a grid block

predict S value

For each grid block, we have a vector like this. For this example B is 2 and C is 2

ρ_1
b_x_1
b_y_1
b_h_1
b_w_1
ρ_2
b_x_2
b_y_2
b_h_2
b_w_2
c_1
c_2



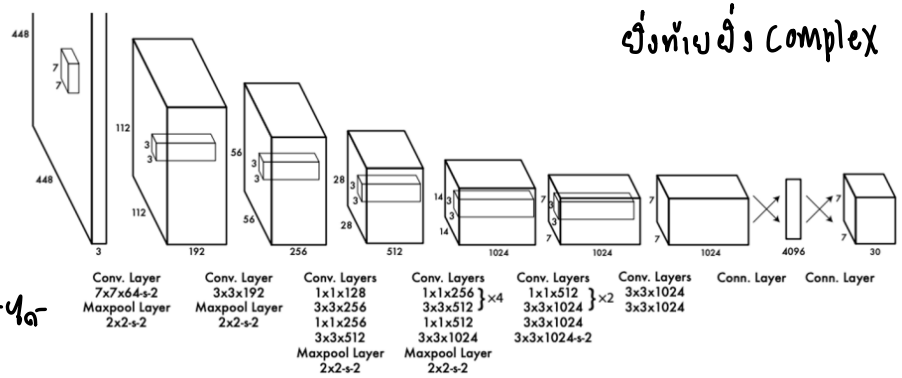
GT label example:

1
b_x_1
b_y_1
b_h_1
b_w_1
0
?
?
?
?
c_1 = 1
c_2 = 0

Source: YOLO

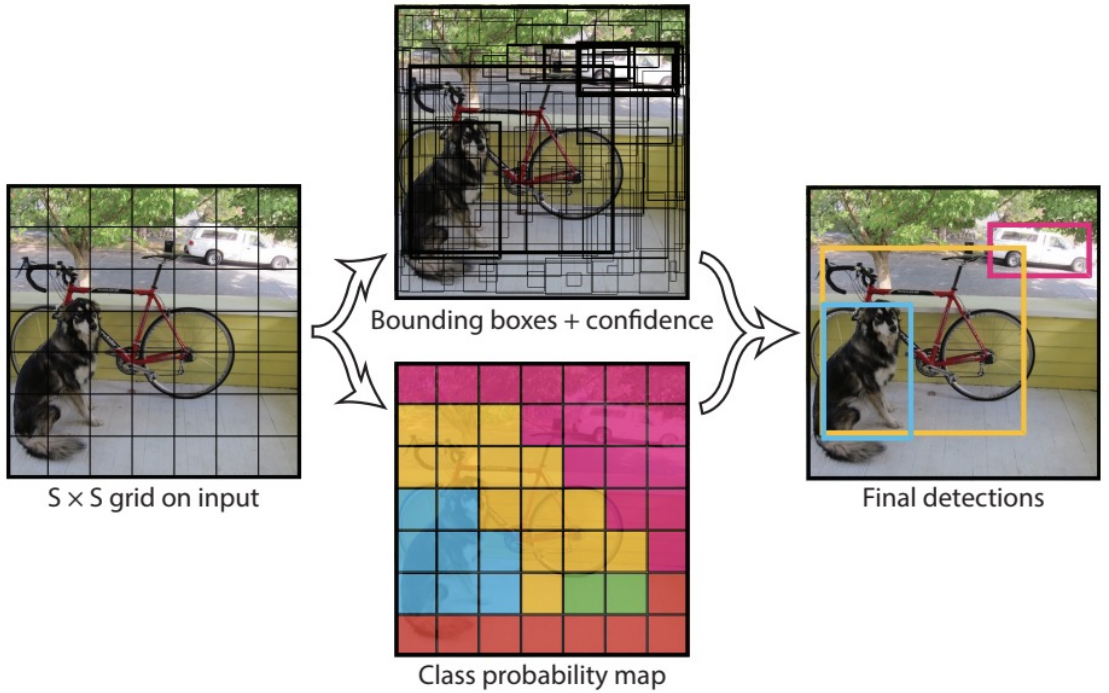
YOLO Architecture

- Now that we know the input and output, we can discuss the model
- We are given 448 by 448 by 3 as our input.
- Implementation uses 7 convolution layers
- Paper parameters: S = 7, B = 2, C = 20
- Output is S*S*(5B+C) = 7*7*(5*2+20) = 7*7*30



❖ Non-maximal suppression

- We then use the output to make final detections
- Use a threshold to filter out bounding boxes with low P(Object)
- In order to know the class for the bounding box compute score take argmax over the distribution Pr(Class|Object) for the grid the bounding box's center is in

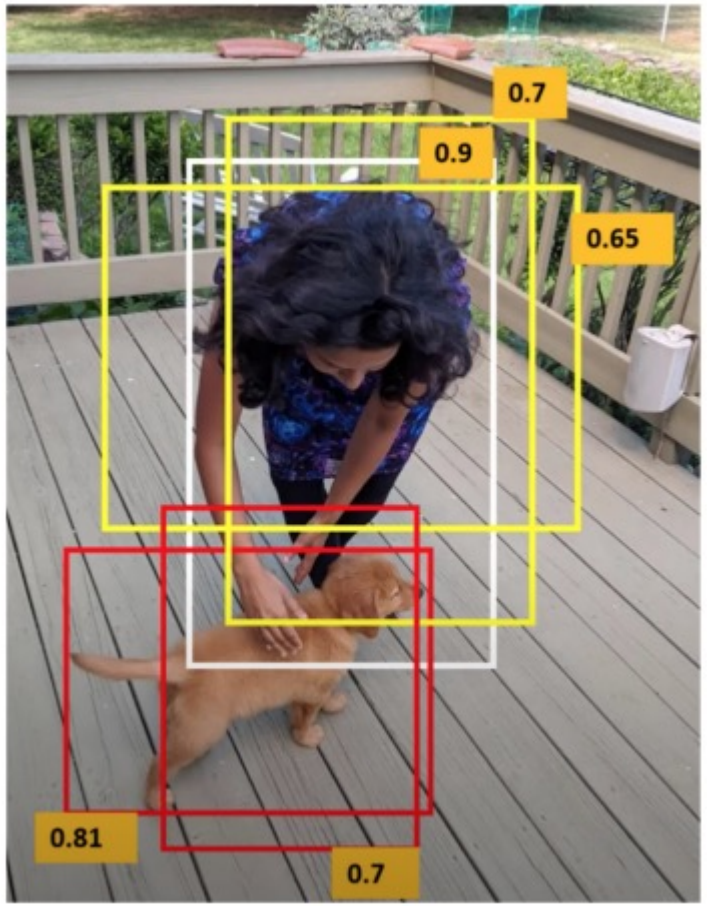


$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}^{\text{truth}}_{\text{pred}} = \Pr(\text{Class}_i) * \text{IOU}^{\text{truth}}_{\text{pred}}$$

Source: YOLO

❖ YOLO Prediction

- Most of the time objects fall in one grid, however it is still possible to get redundant boxes (rare case as object must be close to multiple grid cells for this to happen)
- Discard bounding box with high overlap (keeping the bounding box with highest confidence)
- Adds 2-3% on final mAP score



Source: YOLO

☺ลดerror optimize model

❖ YOLO Objective Function

ensure bounding box ของ obj match กับ box ground truth(GT) ที่มี error น้อย

Localization loss

คำนวณ position(x,y) : MSE loss

Set to 5 to increase the loss of bounding box predictions

GT bbox x-coordinate in the jth cell

GT bbox y-coordinate in the jth cell

Predicted bbox x-coordinate in the ith cell

Predicted bbox y-coordinate in the ith cell

Sum-squared error

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

For each grid cell

For each grid box

'1' if object appears in the ith cell and the jth box detect it, '0' otherwise

error ต่ำ

อยากให้ loss ต่ำ ค่าใกล้กับ GT

size(w,h) : MSRE loss

GT bbox height in the jth cell

$$+\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

Square root to reduce the range of the values

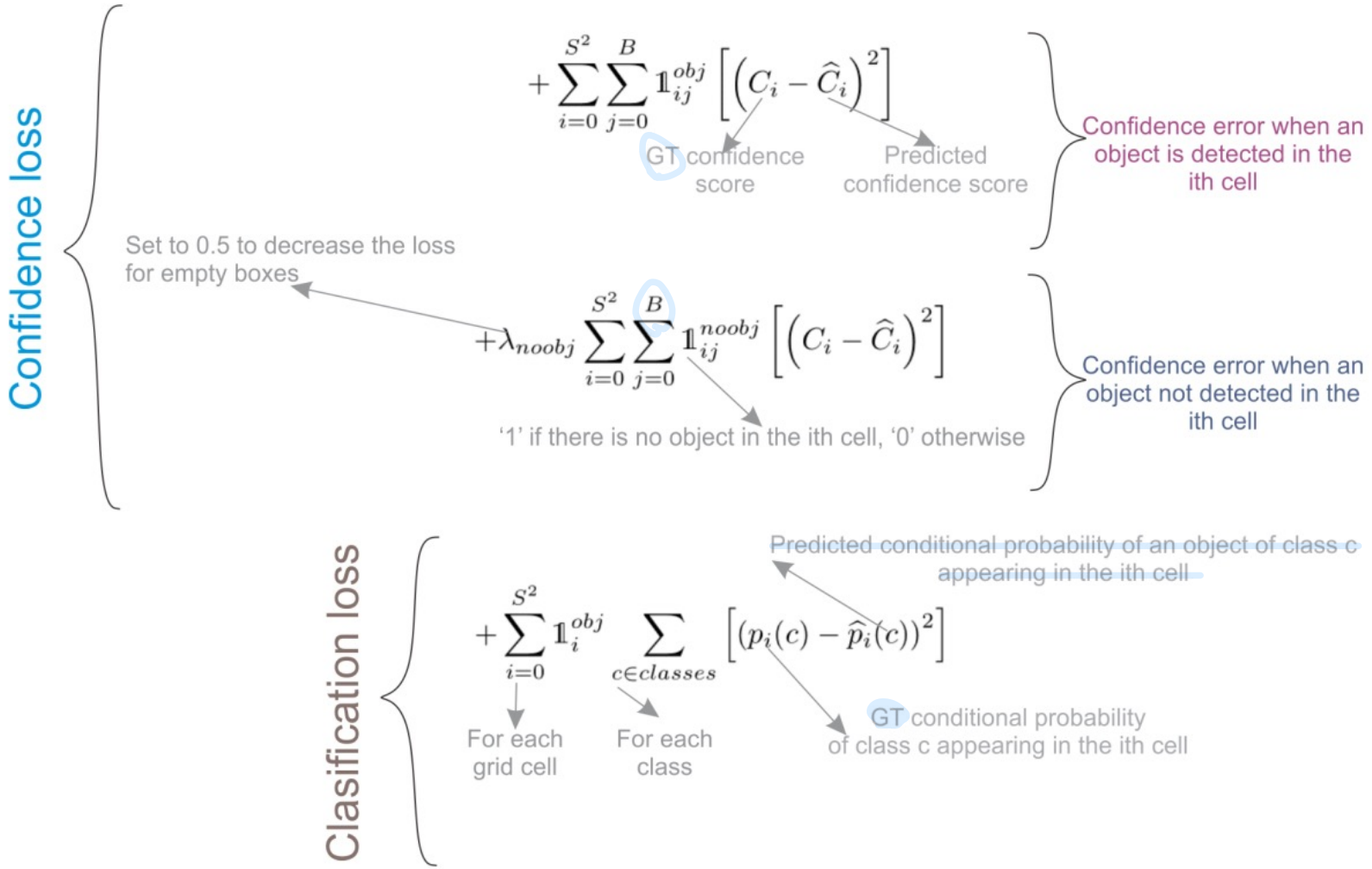
GT bbox width in the ith cell

Predicted bbox width in the ith cell

Predicted bbox height in the ith cell

Source: YOLO

YOLO Objective Function (Cont.)



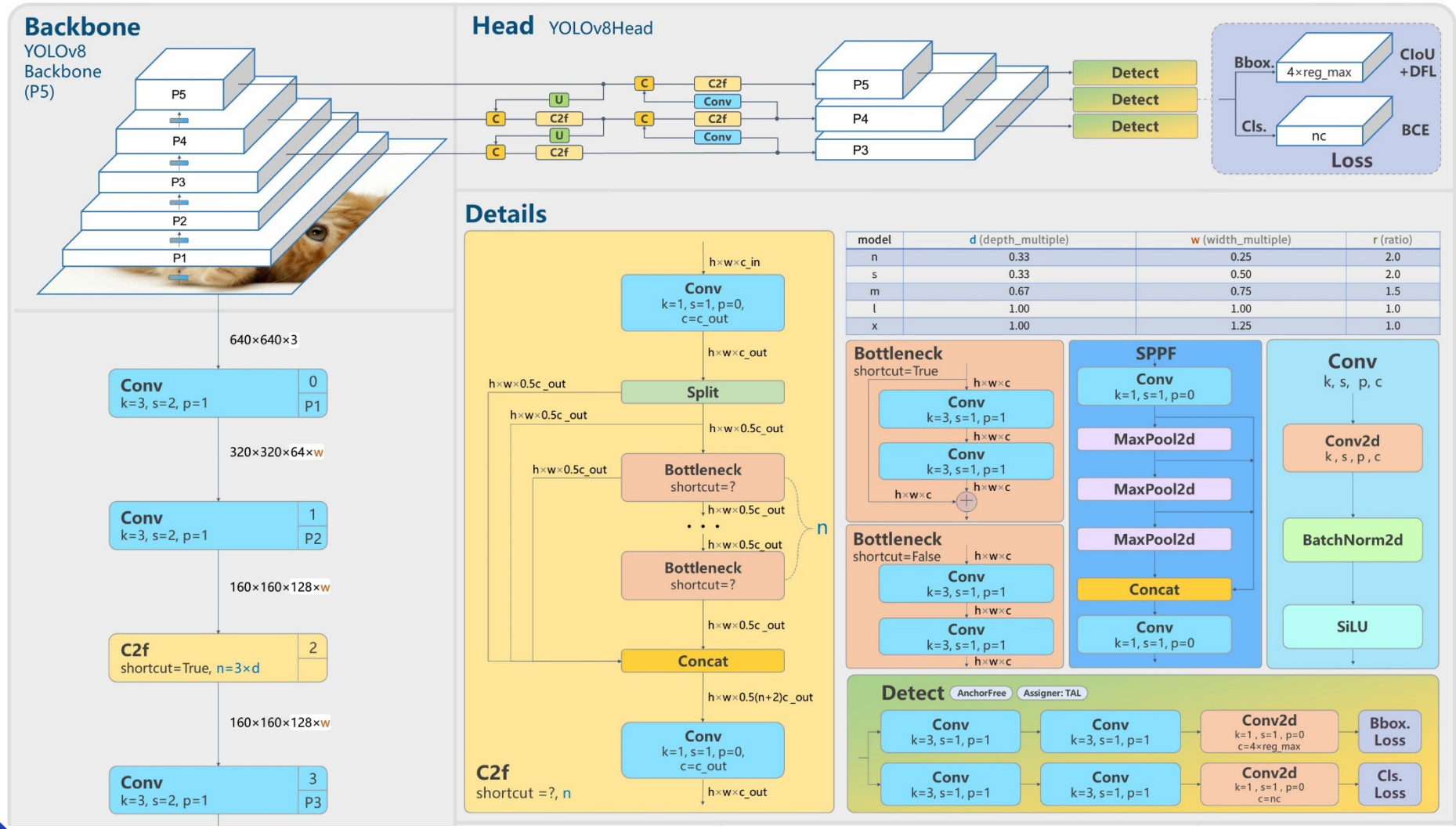
Source: YOLO

❖ YOLO V8

- YOLOv8 uses a similar backbone as YOLOv5 with some changes on the CSPLayer, now called the C2f module.
- The C2f module (cross-stage partial bottleneck with two convolutions) combines high-level features with contextual information to improve detection accuracy

Source: YOLO

YOLO V8 Architecture

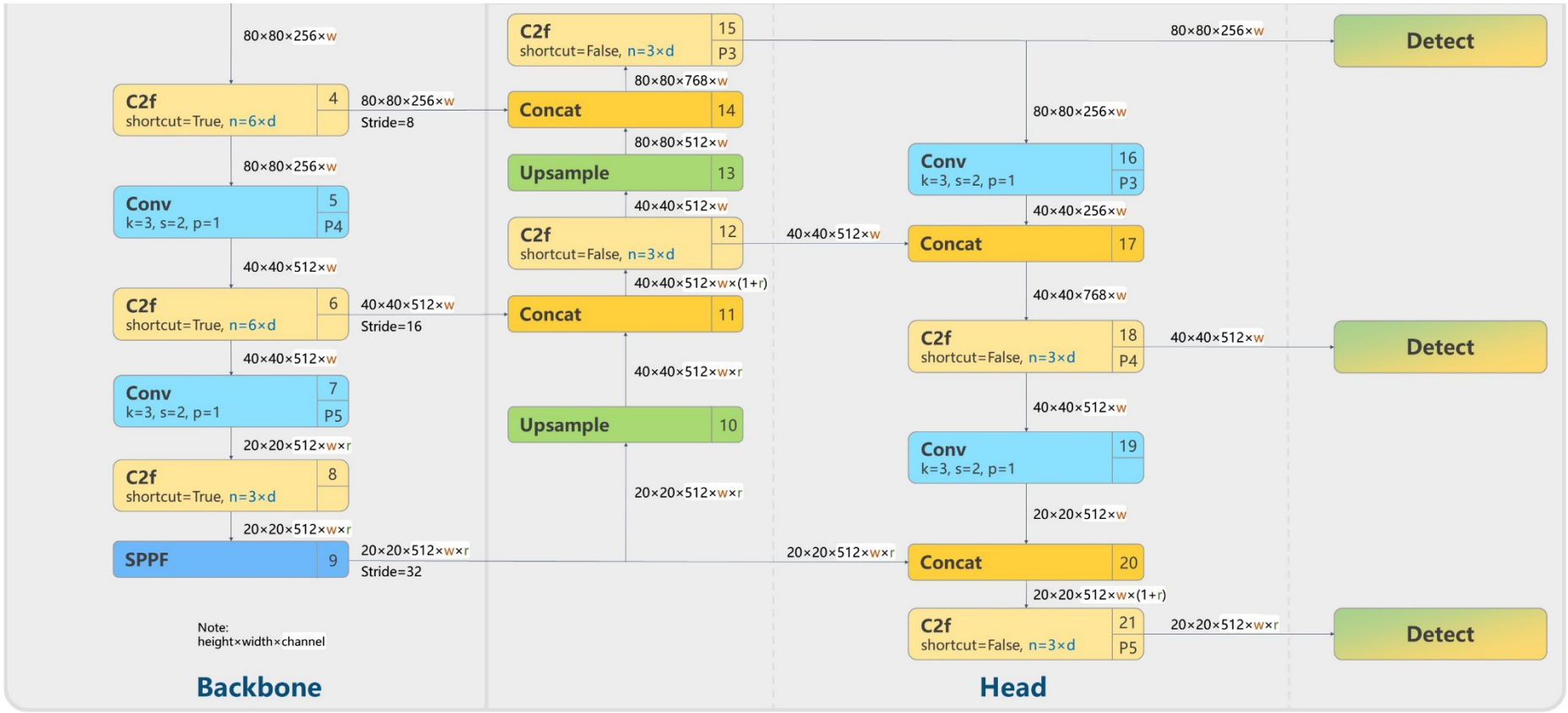


Source: YOLO

YOLO V8 Architecture (Cont.)

v5 → v8

replace C3 ด้วย C2f



Source: YOLO

เพิ่มประสิทธิภาพ ค.แม่นยำ

❖ YOLO V8 Experiment

➢ Using this Google Colab:

https://colab.research.google.com/drive/14x7_B44tBvAe8RzuETDVJ14cYWstnT2D?usp=sharing

King Mongkut's University of Technology Thonburi

Exercise

Extract this video into frame and label it into four classes (bus, taxi, car, and pedestrian), then generate the model to classify those four classes using yolov8

King Mongkut's University of Technology Thonburi

Conclusion

- The research focused on utilizing AI technology to augment police efficiency in Thailand.
- We aimed to enhance law enforcement capabilities and bolster public trust in crime prevention measures.
- By employing AI in crime data analysis, leveraging intelligent CCTV technology for crime monitoring, and integrating real-time alerts for suspicious activities to police.

King Mongkut's University of Technology Thonburi

Q&A