LECTURE 13
OBJECT DETECTION

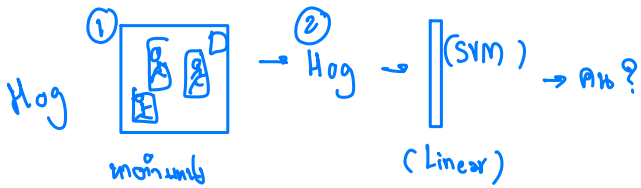
Punnarai Siricharoen, Ph.D.

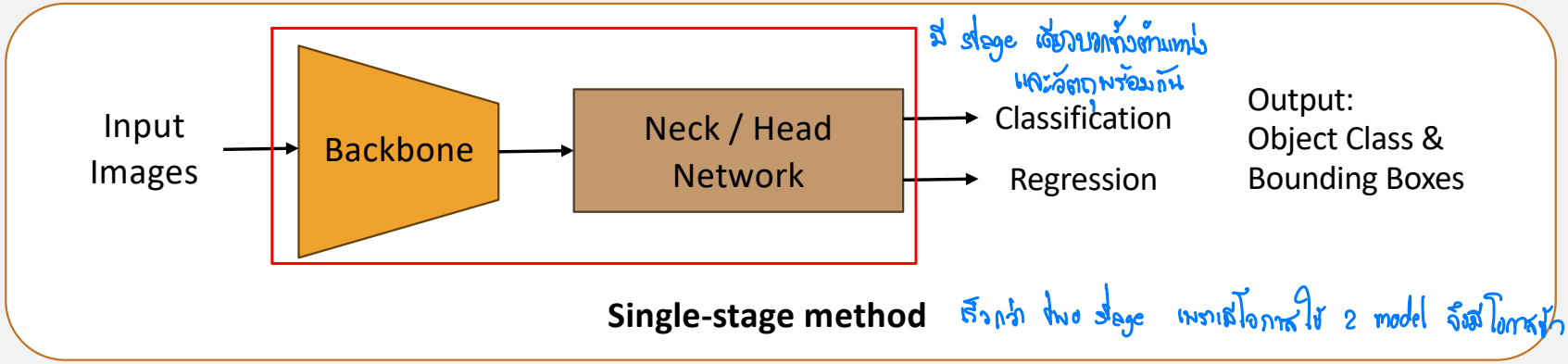
OBJECT DETECTION

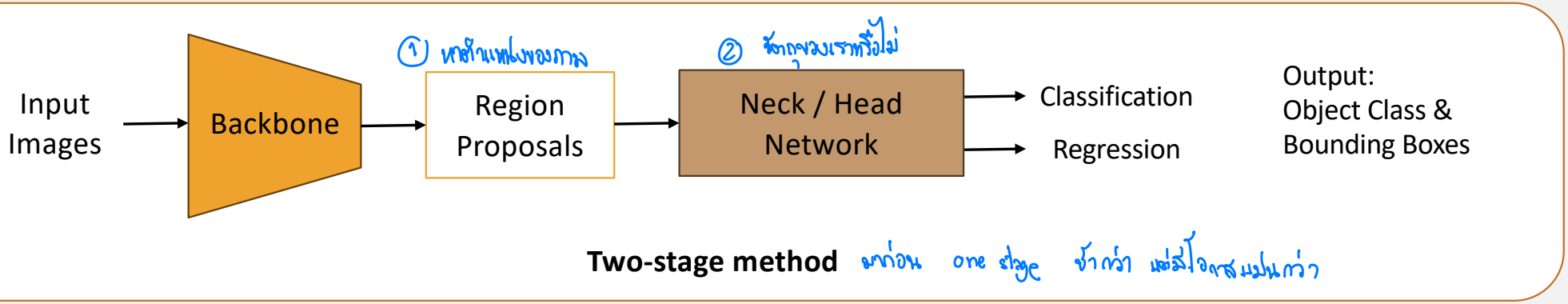
- **Object Detection** is a computer vision task in which the goal is to detect and locate objects of interest in an image or video. The task involves identifying the position and boundaries of objects in an image, and classifying the objects into different categories.
  1. One-stage methods prioritize inference speed, and example models include YOLO, SSD and RetinaNet.
     - perform object classification and bounding box regression in a single pass through the network.
  2. Two-stage methods prioritize detection accuracy, and example models include Faster R-CNN, Mask R-CNN and Cascade R-CNN.
     - First, generate region proposals, which are areas of the image that might contain objects.
     - Then, classify these regions and refine their boundaries to accurately detect objects.
- The most popular benchmark is the MSCOCO dataset. Models are typically evaluated according to a **Mean Average Precision (mAP)** metric.

https://paperswithcode.com/task/object-detection

OBJECT DETECTION







OBJECT DETECTION

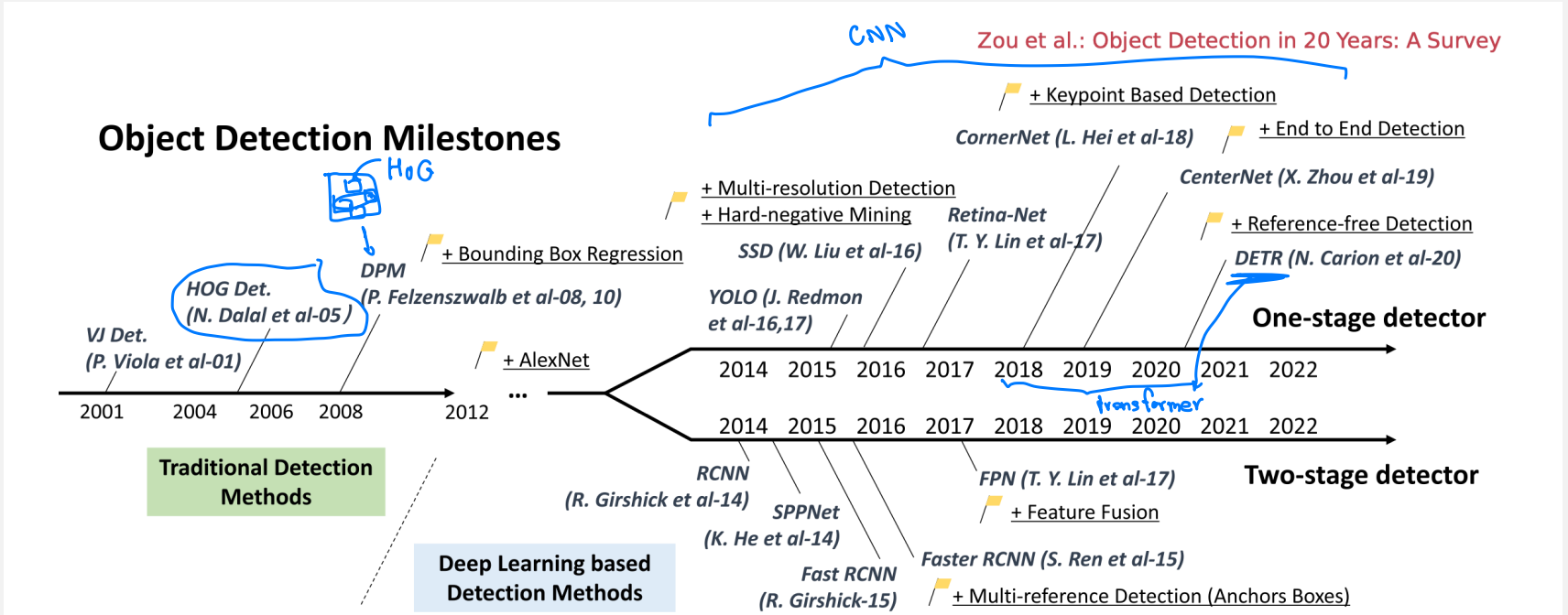
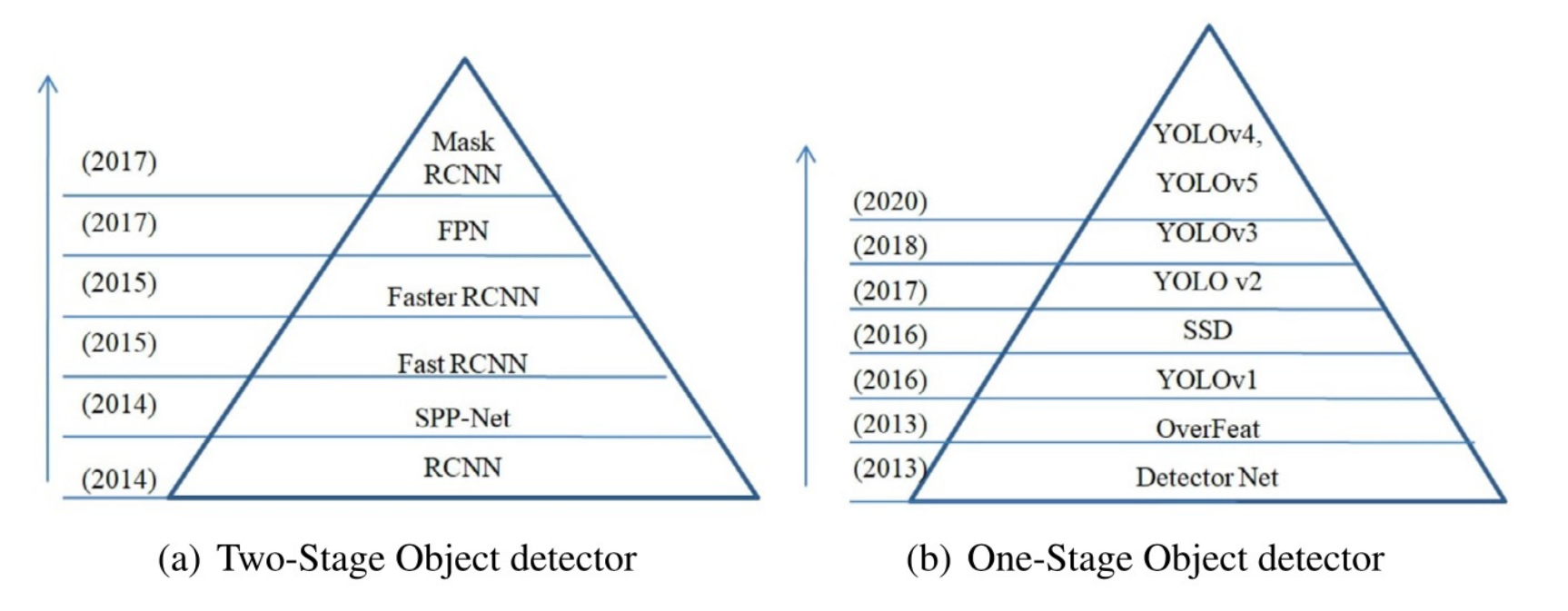


Fig. 2. ***Road map of object detection. Milestone detectors in this figure: VJ Det. [10], [11], HOG Det. [12], DPM [13], [14], [15], RCNN [16], SPPNet [17], Fast RCNN [18], Faster RCNN [19], YOLO [20], [21], [22], SSD [23], FPN [24], Retina-Net [25], CornerNet [26], CenterNet [27], and DETR [28].***

Z. Zou, K. Chen, Z. Shi, Y. Guo and J. Ye, "Object Detection in 20 Years: A Survey," in Proceedings of the IEEE, vol. 111, no. 3, pp. 257-276, March 2023, doi: 10.1109/JPROC.2023.3238524.

OBJECT DETECTION



(a) Two-Stage Object detector

(b) One-Stage Object detector

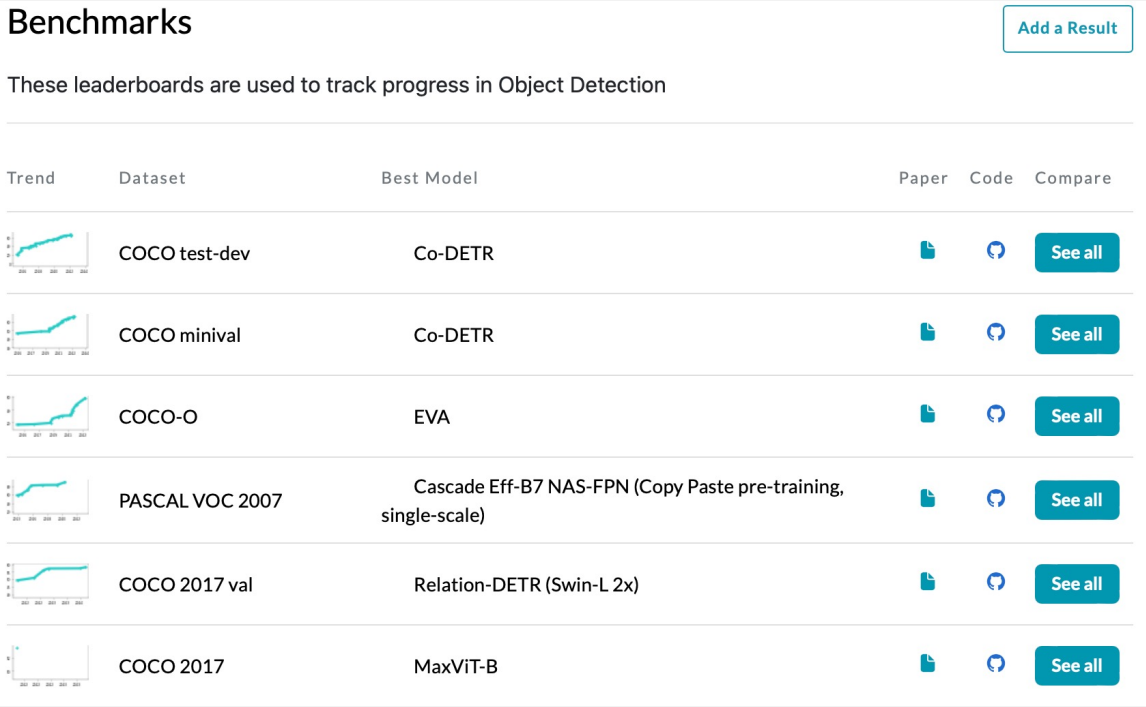
Kaur & Singh (2023), A comprehensive review of object detection with deep learning, Digital Signal Processing

Large Dataset
class → ImageNet

seg → ADE20k

OBJECT DETECTION

- **COCO** - over 330,000 images, with more than 2.5 million object instances labeled across 80 object categories 2.5m label
- **PASCAL VOC 2007** - 10k, 2010 - 15k images - 20 object classes, such as person, car, and dog.



Benchmarks

Add a Result

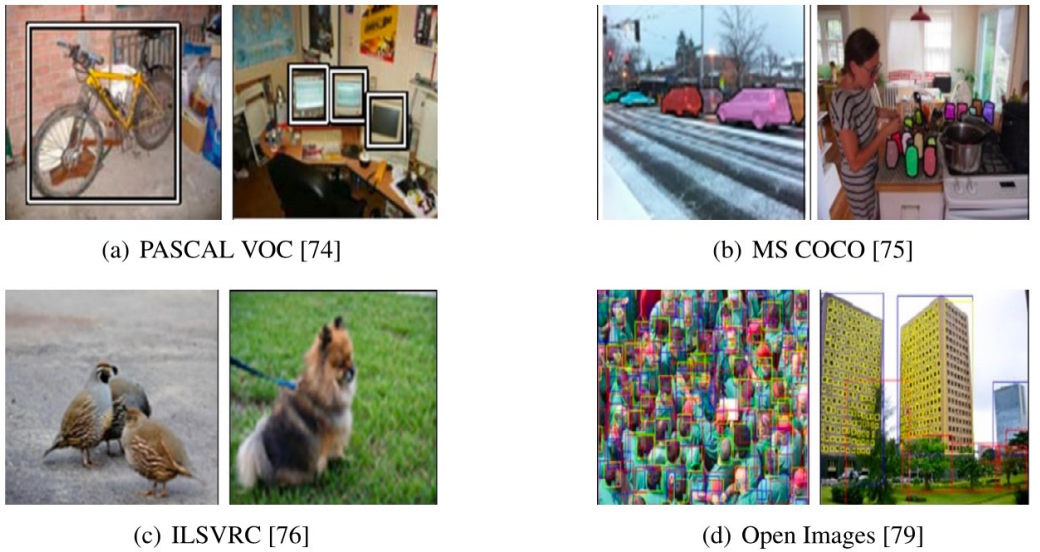
These leaderboards are used to track progress in Object Detection

Trend	Dataset	Best Model	Paper	Code	Compare
	COCO test-dev	Co-DETR			See all
	COCO minival	Co-DETR			See all
	COCO-O	EVA			See all
	PASCAL VOC 2007	Cascade Eff-B7 NAS-FPN (Copy Paste pre-training, single-scale)			See all
	COCO 2017 val	Relation-DETR (Swin-L 2x)			See all
	COCO 2017	MaxViT-B			See all

https://paperswithcode.com/task/object-detection

DATASET

- **Open Images**: The year 2018 sees the introduction of the open images detection (OID) challenge - 1910k images with 15 440k annotated bounding boxes on 600 object categories.
- **ILSVRC** is organized each year from 2010 to 2017. It contains a detection challenge using ImageNet images - contains 200 classes of visual objects (many single-object images and focuses on identifying specific object classes without much context, less popular compared with COCO)



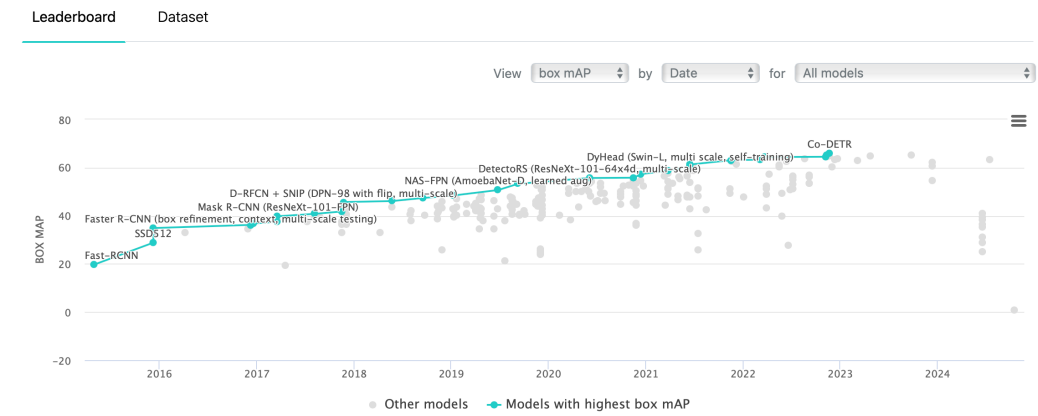
(a) PASCAL VOC [74] (b) MS COCO [75] (c) ILSVRC [76] (d) Open Images [79]

Fig. 6. Sample of annotated images taken from commonly used datasets.

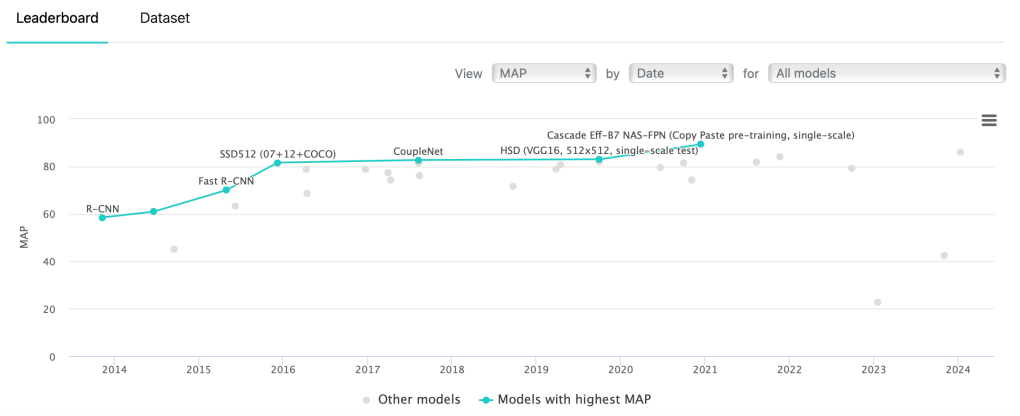
OBJECT DETECTION

- **COCO** - over 330,000 images, with more than 2.5 million object instances labeled across 80 object categories
- **PASCAL VOC 2007** - 10k, 2010 - 15k images - 20 object classes, such as person, car, and dog.

Object Detection on COCO test-dev

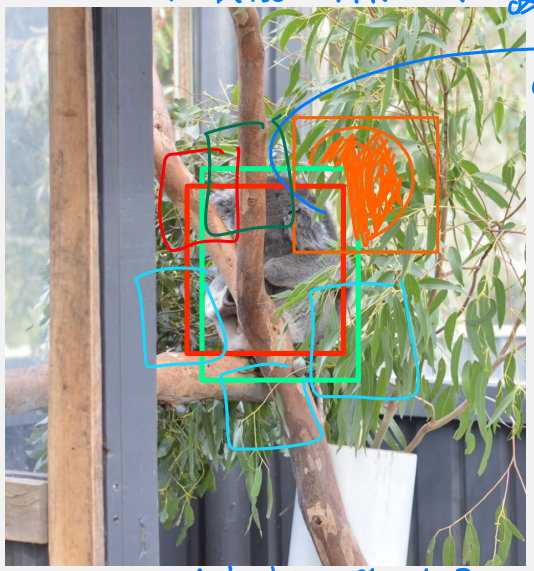


Object Detection on PASCAL VOC 2007



https://paperswithcode.com/task/object-detection

EVALUATION FOR OBJECT DETECTION



Ground-truth

Detected Object

Overlap

Union

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

mAP – Mean Average Precision

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

1) Rank the confidence score of the detected objects 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) = 1/4 = 0.25

Rank 3:

Precision = 2/3 = 0.67

Recall = 2/4 = 0.5

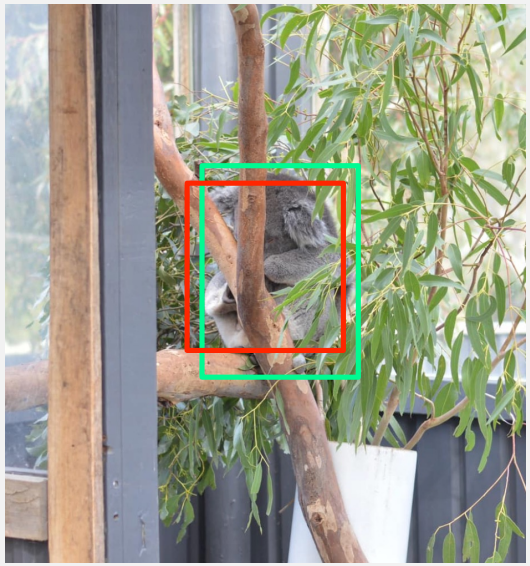
Rank 5:

Precision = 3/5 = 0.6

Recall = ¾ = 0.75

EVALUATION FOR OBJECT DETECTION

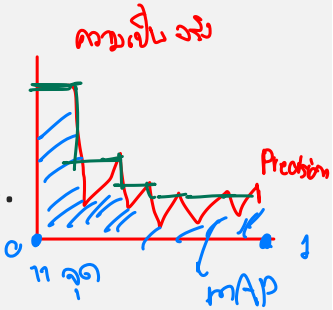
mAP = $\sum_{i \in C} AP_i$ mAP แต่ละ class แล้วเอามาเฉลี่ย



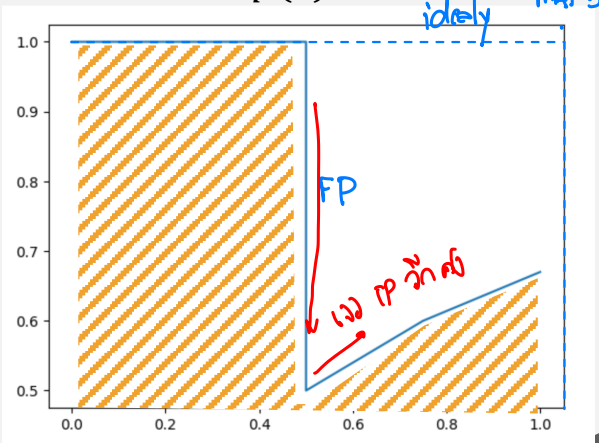
mAP – Mean Average Precision

mAP50 – The prediction is correct, when IoU ≥ 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
- mAP50:95 – average over 10 IoUs



$p(r)$



r

precision

mAP 75 = overlap 75% กรณีอื่นๆ 5 mAP 50, 55, 60

mAP โดยทั่วไปใช้ ~ mAP 50:95

mAP 50 > mAP เสมอ

ทำ model robust แต่ก็ยากขึ้น

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

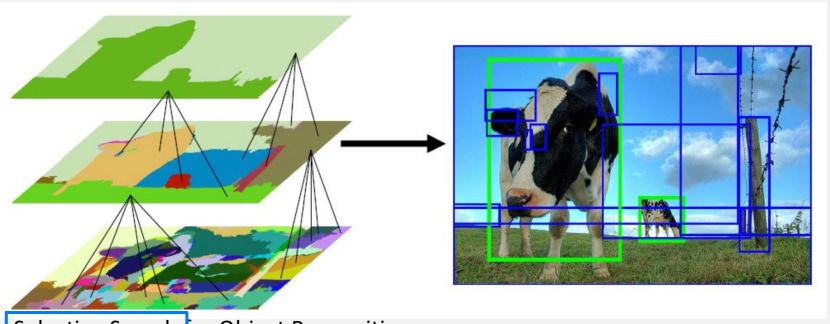
HOG – CNN

REGIONS WITH CNN FEATURES (RCNNS)

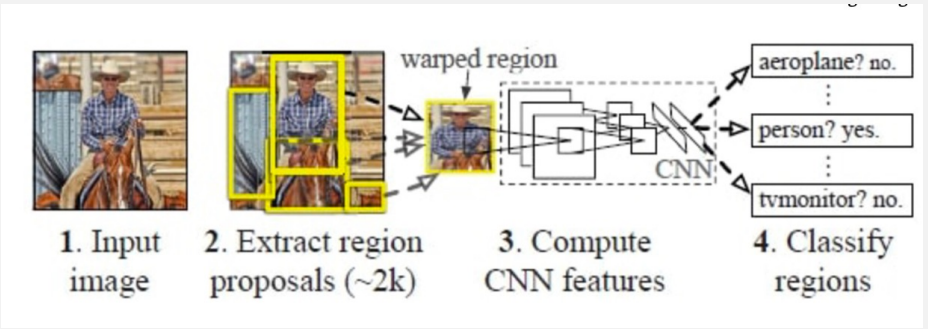
Regions with CNN features (RCNNs)

1. starts with the extraction of a set of object proposals (object candidate boxes) by selective search
2. Then, each proposal is rescaled to a fixed-size image and fed into a CNN model pretrained on ImageNet (e.g., AlexNet) to extract features
3. Finally, linear SVM classifiers are used to predict the presence of an object within each region and to recognize object categories.

- Although RCNN has made great progress, its drawbacks are obvious: the redundant feature computations on a large number of overlapped proposals (over 2000 boxes from one image) lead to an extremely slow detection speed (14 s per image with GPU)
- **mAP** VOC07 – 58.5%



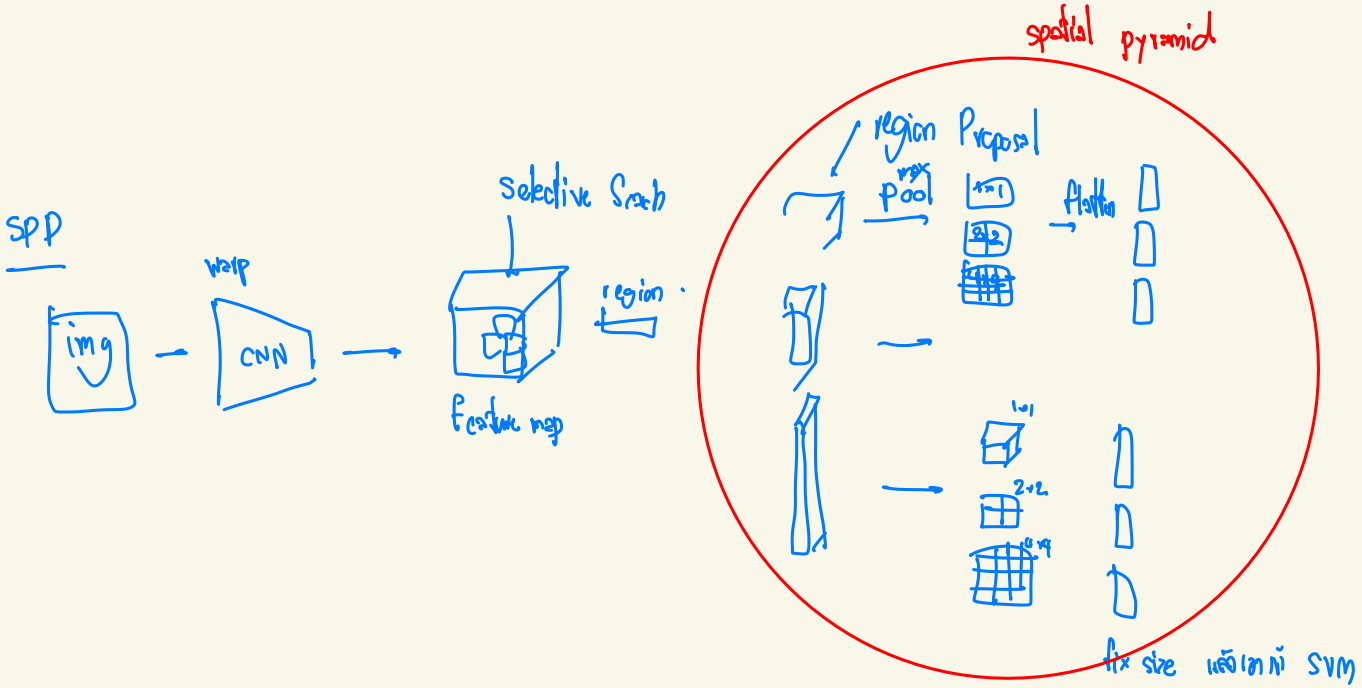
Selective Search for Object Recognition
J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders
In International Journal of Computer Vision 2013.



Architecture of RCNN

Z. Zou, K. Chen, Z. Shi, Y. Guo and J. Ye, "Object Detection in 20 Years: A Survey," in Proceedings of the IEEE, vol. 111, no. 3, pp. 257-276, March 2023, doi: 10.1109/JPROC.2023.3238524.

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


SPP
img
warp
CNN
Selective Search
feature map
region
spatial pyramid
region Proposal
max pool
1×1
2×2
4×4
flatten
fix size เพื่อเข้า SVM

SPPNET – SPATIAL PYRAMID POOLING NETWORK

- R-CNN performs a ConvNet forward pass for each region proposal without sharing computation, R-CNN takes a long time on SVMs classification
- **SPPNet:**
  - A single convolutional pass is applied to the entire image to create feature maps.
  - Region proposals are generated using an external method like Selective Search.
  - The SPP layer extracts fixed-size features (e.g., 1x1, 2x2, 4x4 grids) from the feature maps for each proposal, making SPPNet faster and more efficient by reducing redundant computations.
  - **mAP** 59.2% for VOC07 0.382s per img.

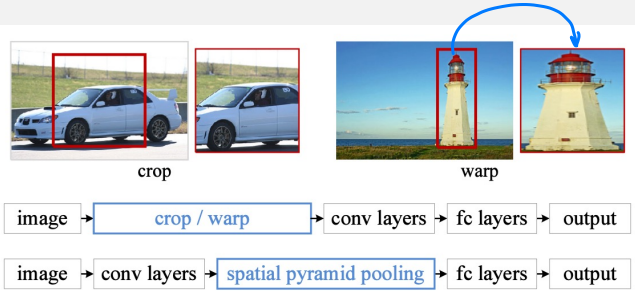


Figure 1: Top: cropping or warping to fit a fixed size. Middle: a conventional CNN. Bottom: our spatial pyramid pooling network structure.

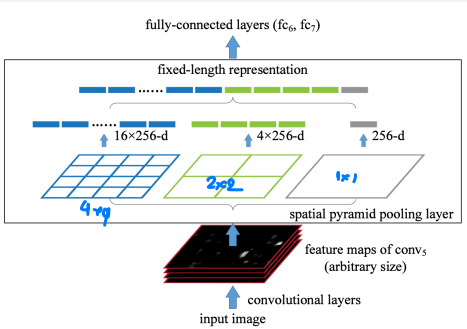


Figure 3: A network structure with a **spatial pyramid pooling layer**. Here 256 is the filter number of the $conv_5$ layer, and $conv_5$ is the last convolutional layer.

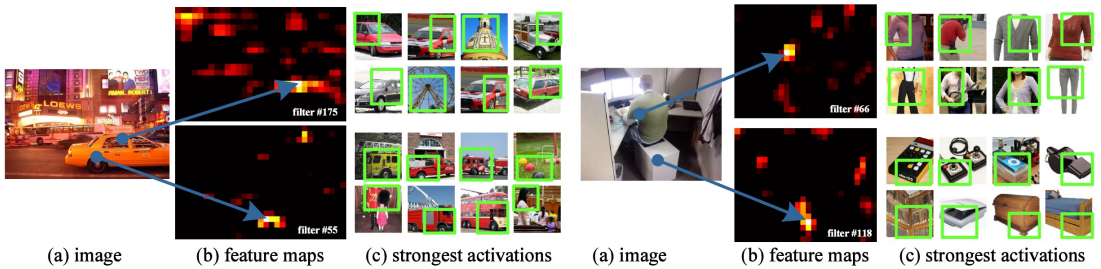
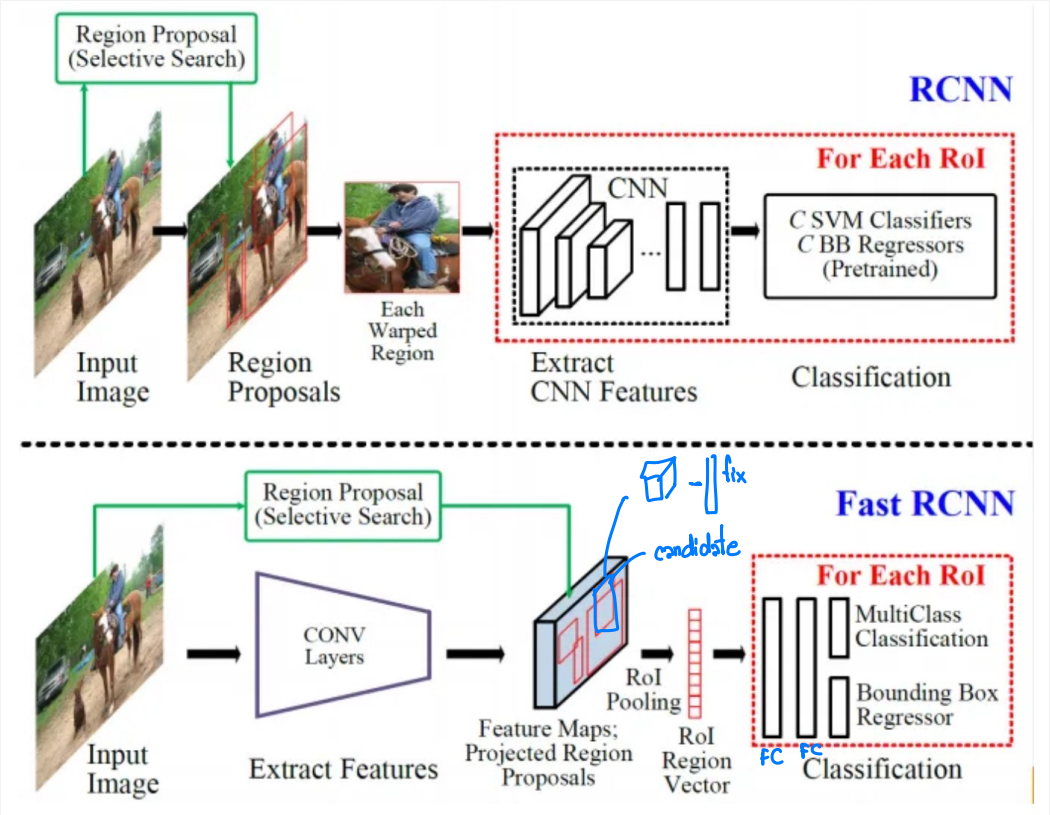


Figure 2: Visualization of the feature maps. (a) Two images in Pascal VOC 2007. (b) The feature maps of some $conv_5$ filters. The arrows indicate the strongest responses and their corresponding positions in the images. (c) The ImageNet images that have the strongest responses of the corresponding filters. The green rectangles mark the receptive fields of the strongest responses.

Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition
Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

FAST R-CNN

- SPPNet (1) depends on external methods Selective Search to generate region proposals. These methods can be computationally expensive and slow (2) fixed-length feature vector by pooling over multiple spatial scales (e.g., 1x1, 2x2, 4x4 grids) requires memory.
- **Fast R-CNN** extracts features from an entire input image and then passes the region of interest (RoI) pooling layer to get the fixed size features as the input of the following classification and bounding box regression fully connected layers
  - simultaneously train a detector and a bounding box regressor under the same network configurations.
  - 200 times faster than R-CNN
  - **mAP** VOC07 – 70.0%

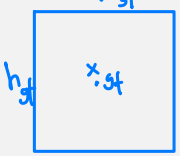


RCNN vs Fast-RCNN (source: Deep Learning for Generic Object Detection: A Survey)

FAST R-CNN

Bounding Box Regressor:

- A region proposal box : Center coordinate (x, y) and width (w) and height (h)
- Prediction output: $(\hat{t}_x, \hat{t}_y, \hat{t}_w, \hat{t}_h)$, represent the adjustments in Horizontal, Vertical, Width scale and Height scale, respectively. The target values are calculated as follows (aims to predict):



$$t_x = \frac{x_{gt} - x}{w}$$

$$t_y = \frac{y_{gt} - y}{h}$$

$$t_w = \log\left(\frac{w_{gt}}{w}\right)$$

$$t_h = \log\left(\frac{h_{gt}}{h}\right)$$

- **Smooth L1 Loss for the bounding box regression** :

$$L(x, y) = \begin{cases} 0.5(x - y)^2, & if \ |x - y| < 1 \\ |x - y| - 0.5, & otherwise \end{cases}$$

$$Total\ loss = L(\hat{t}_x, t_x) + L(\hat{t}_y, t_y) + L(\hat{t}_w, t_w) + L(\hat{t}_h, t_h)$$

Use Bounding Box Regressor at Inference:

- The regressor outputs $\hat{t}_x, \hat{t}_y, \hat{t}_w, \hat{t}_h$ for each proposal.
- These values are used to compute the refined bounding box coordinates

$$New\ center\ x' = x + \hat{t}_x \times width$$
$$New\ center\ y' = y + \hat{t}_h \times height$$
$$New\ width\ w' = exp(\hat{t}_w) \times width$$
$$New\ height\ h' = \exp(\hat{t}_h) \times height$$

FASTER R-CNN

- Selective Search is a slow and time-consuming process
- **Faster R-CNN** replaces it with a novel RPN (region proposal network)
  - A fully convolutional network to efficiently predict region proposals with a wide range of scales and aspect ratios.
  - Fast, integrated proposal generation, end-to-end training for efficiency, and shared convolutional layers that improve speed and accuracy.
- Region Proposal Network (RPN)
  - Shared Convolutional Feature Map
  - The RPN slides a small 3x3 convolutional window over the feature map, examining each spatial location to generate region proposals.
  - At each sliding window location, the RPN generates several anchor boxes - predefined bounding boxes that serve as reference boxes around which the network will propose regions.

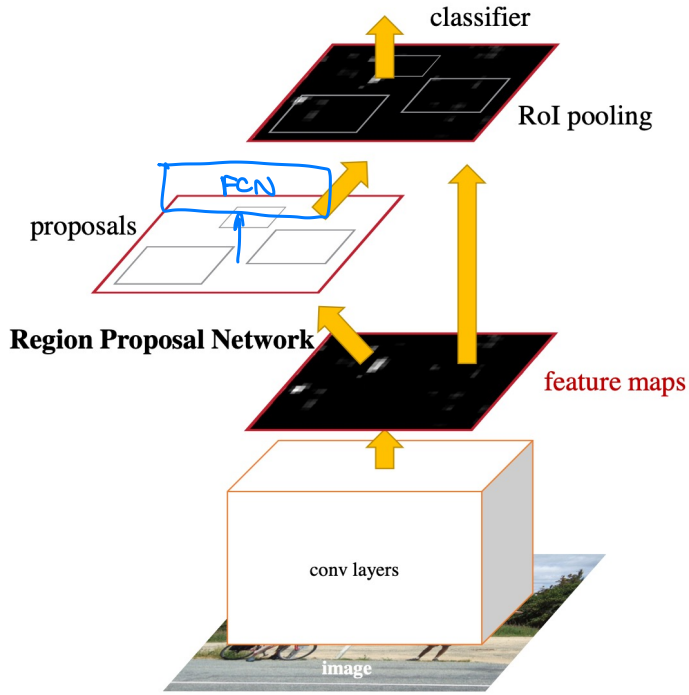
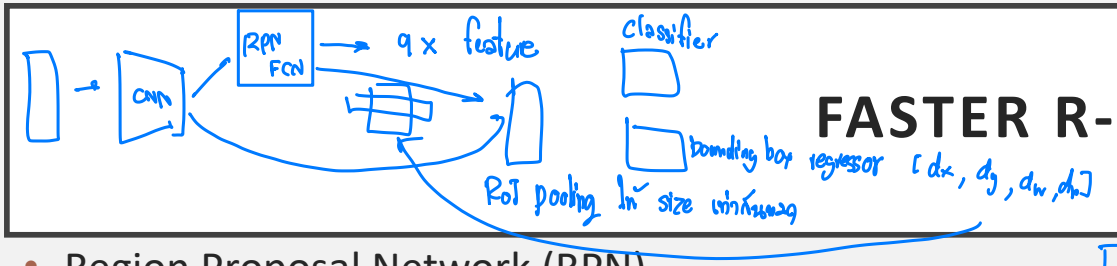
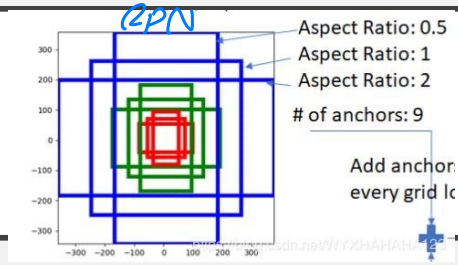


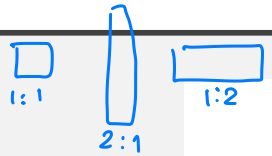
Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

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

FASTER R-CNN







3 scales and 3 aspect ratios

- Region Proposal Network (RPN)
  1. Use Shared Convolutional Feature Map
  2. The RPN slides a small 3x3 convolutional window over the feature map, examining each spatial location to generate region proposals.
  3. At each sliding window location, the RPN generates several anchor boxes - predefined bounding boxes that serve as reference boxes around which the network will propose regions.
  4. For each anchor, the RPN outputs :
     - objectness score (whether contain object or NOT)
     - Bounding box regression offsets: Adjustments to refine the anchor box coordinates, making the proposal fit the object more closely.
  5. Non-Maximum Suppression (NMS) to eliminate redundant proposals that have high overlap with each other, keeping only the highest-scoring proposals. คือ เอาเฉพาะ proposal ที่ทับกันมากๆ เก็บเฉพาะ score สูงๆ
  6. Final set of region proposals will be applied with ROI pooling for further classification and bounding box regression.

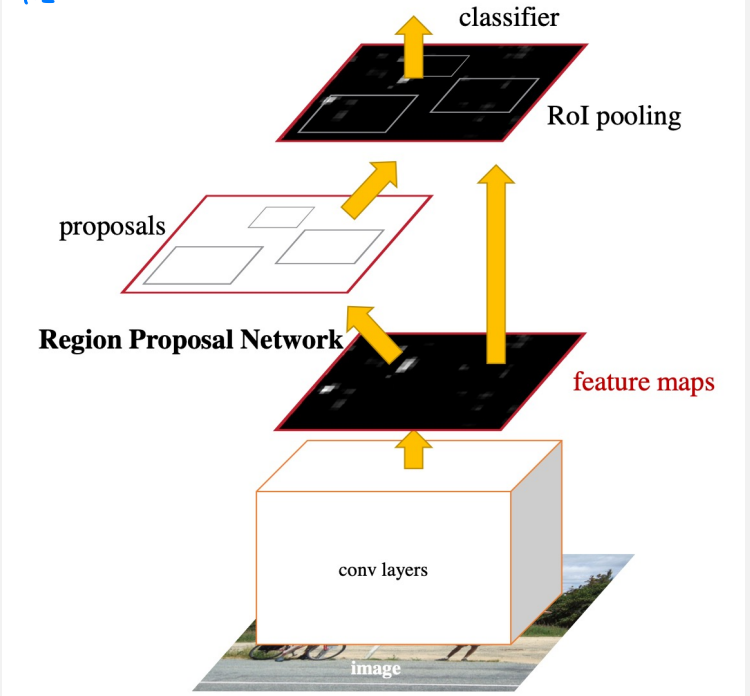


Figure 2: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

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

FEATURE PYRAMID NETWORKS

- The features in deeper layers of a CNN are beneficial for category recognition.
- A top- down architecture with lateral connections is developed in FPN for building high-level semantics at all scales.
- a top-down pathway where features from deeper layers (which have rich semantic information) are progressively upsampled and combined with features from earlier.
- CNN naturally forms a feature pyramid through its forward propagation, the FPN shows great advances for detecting objects with a wide variety of scales.
- FPN has now become a basic building block of many latest detectors.

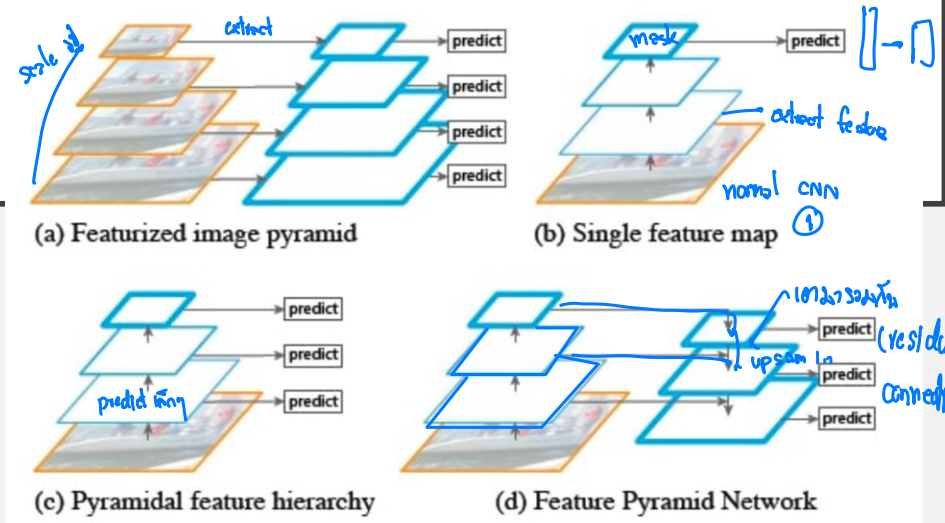


Fig. 5. Feature Pyramid Network [28]. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

1. This process begins from the highest level (deepest layer) in the backbone and moves upward, layer by layer.
2. Upsampling Method: Nearest Neighbor or Bilinear Interpolation
3. Lateral Connections with 1x1 Convolutions - to adjust the number of channels
4. Merging Features via Addition
5. Applying 3x3 Convolutions to Refine Feature

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

MASK R-CNN

- Mask R-CNN
  1. Pixel-level instance segmentation - additional Mask Branch – using a transposed convolution
     - Binary Mask Prediction - outputs a binary mask for each object class, predicting whether each pixel within the Region of Interest (RoI) belongs to the object or not.
     - A sigmoid activation function is applied to each pixel in the mask, producing a probability score that indicates the likelihood of the pixel being part of the object.
  2. RoIAlign layer to improve spatial precision - ensuring better alignment of features for accurate mask prediction.

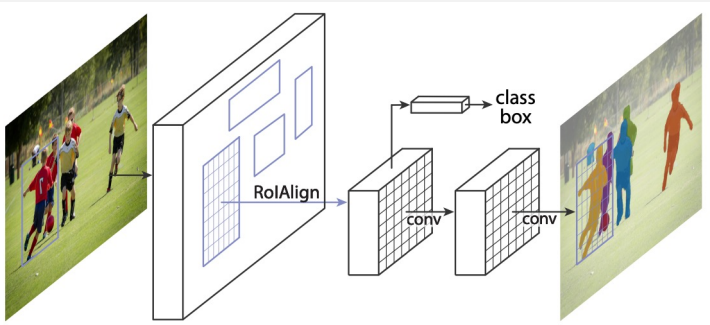


Figure 1. The **Mask R-CNN** framework for instance segmentation.

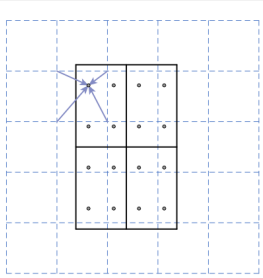


Figure 3. **RoIAlign:** The dashed grid represents a feature map, the solid lines an RoI (with 2×2 bins in this example), and the dots the 4 sampling points in each bin. RoIAlign computes the value of each sampling point by bilinear interpolation from the nearby grid points on the feature map. No quantization is performed on any coordinates involved in the RoI, its bins, or the sampling points.

K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 386–397, 2020, doi: 10.1109/TPAMI.2018.2844175.

MASK R-CNN

- Mask R-CNN
  1. RoIAlign layer to improve spatial precision - ensuring better alignment of features for accurate mask prediction.
     - Problem with **RoIPooling** extract object regions from feature maps by (max) pooling the features to fit a fixed grid ----- introduces some **inaccuracies** because rounding and selecting maximum values can misalign the features slightly.
     - RoIAlign keeps the ***exact floating-point coordinates*** of each region. It divides each region into a grid and samples exact points (without rounding) within each cell of the grid.
     - RoIAlign uses ***bilinear interpolation***, which means it smoothly blends nearby pixel values to get an accurate feature value.
     - RoIAlign captures features ***more precisely***, helping Mask R-CNN produce accurate masks and object boundaries.

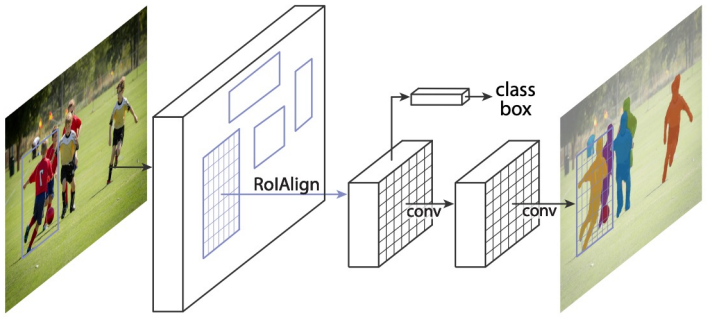


Figure 1. The **Mask R-CNN** framework for instance segmentation.

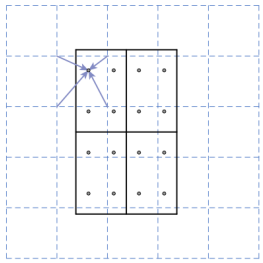
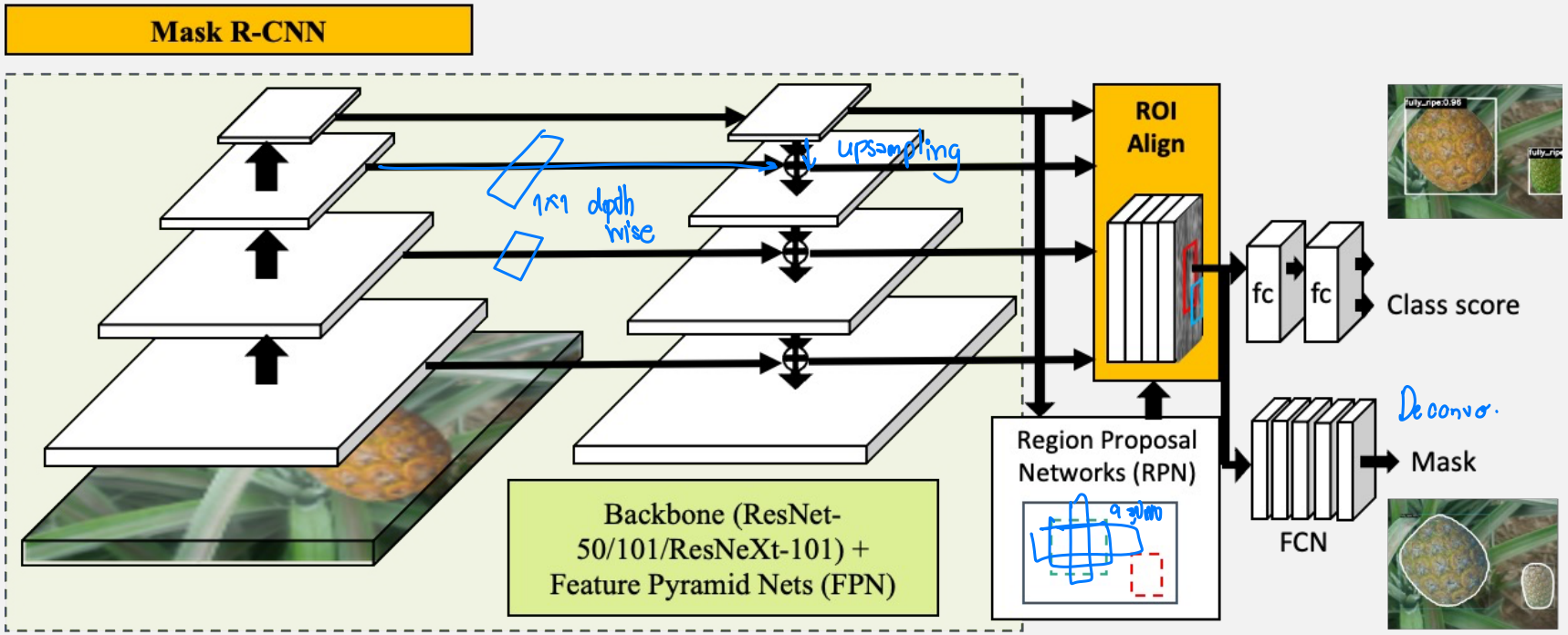


Figure 3. **RoIAlign:** The dashed grid represents a feature map, the solid lines an RoI (with 2×2 bins in this example), and the dots the 4 sampling points in each bin. RoIAlign computes the value of each sampling point by bilinear interpolation from the nearby grid points on the feature map. No quantization is performed on any coordinates involved in the RoI, its bins, or the sampling points.

K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 386–397, 2020, doi: 10.1109/TPAMI.2018.2844175.

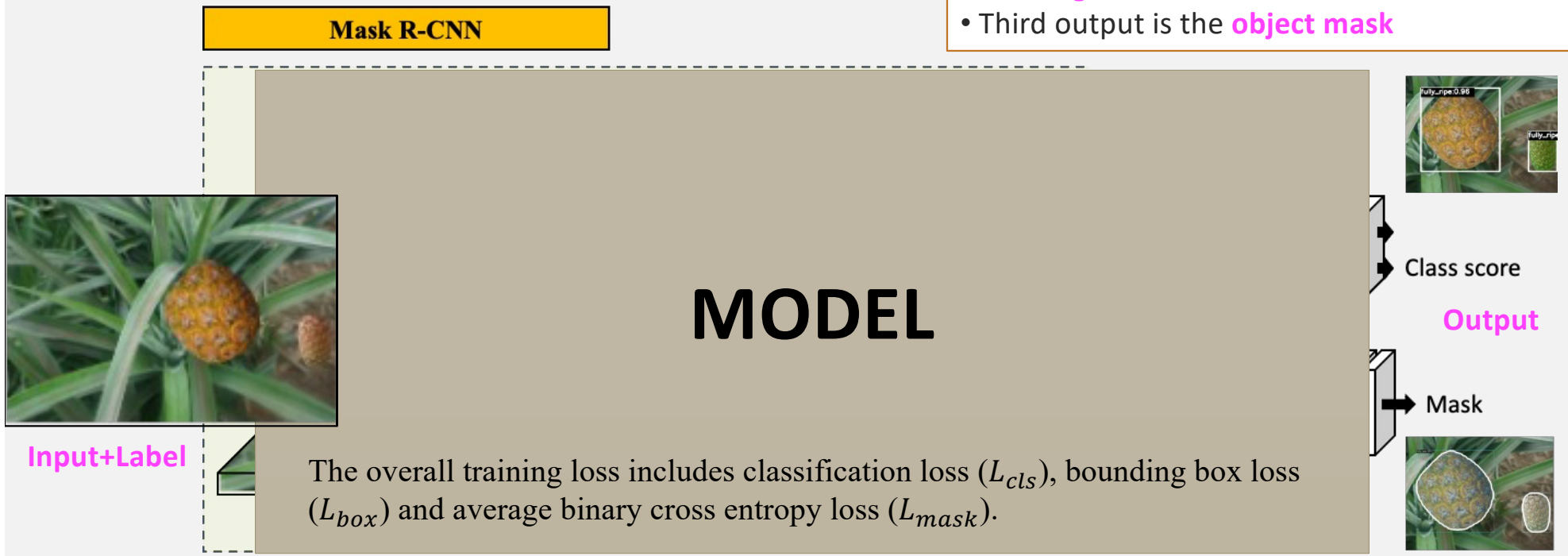
MASK R-CNN



K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 386–397, 2020, doi: 10.1109/TPAMI.2018.2844175.

MASK R-CNN

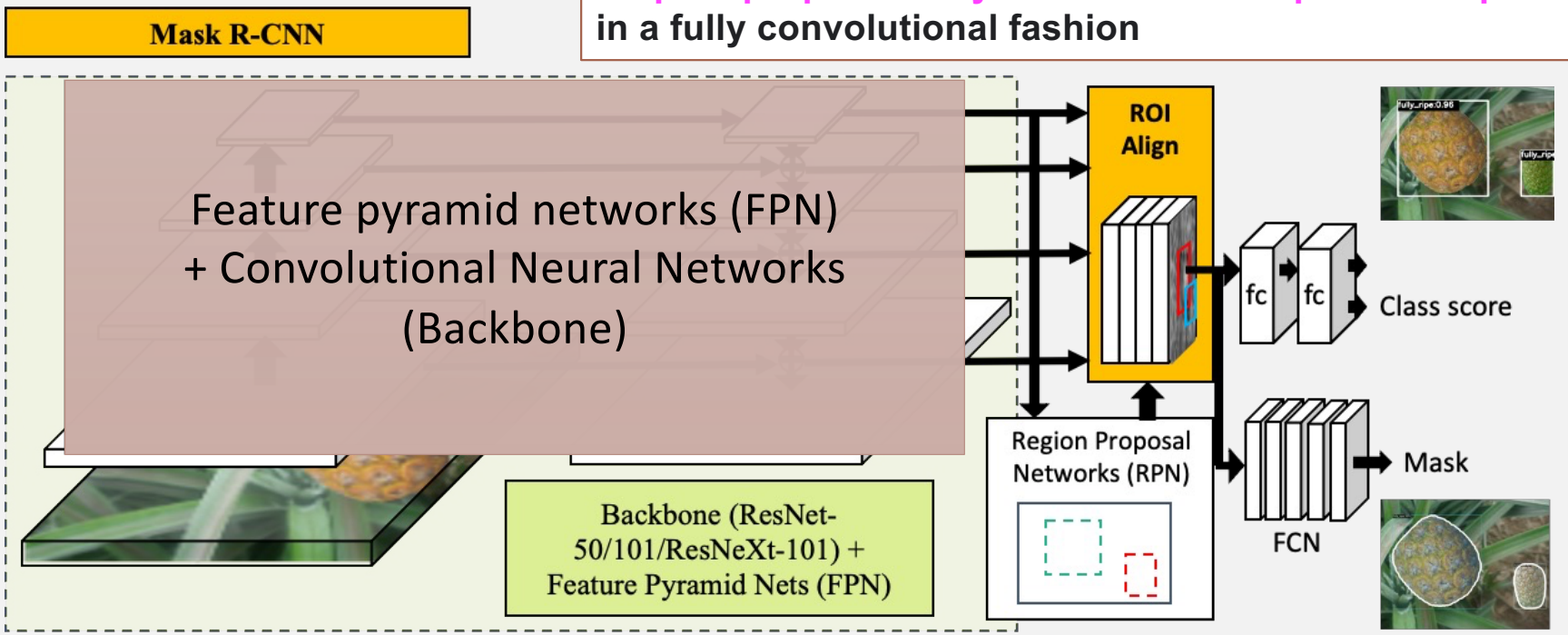
Mask R-CNN has three outputs
- For each candidate object, a **class label** and a **bounding-box** offset;
- Third output is the **object mask**



K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 386–397, 2020, doi: 10.1109/TPAMI.2018.2844175.

MASK R-CNN

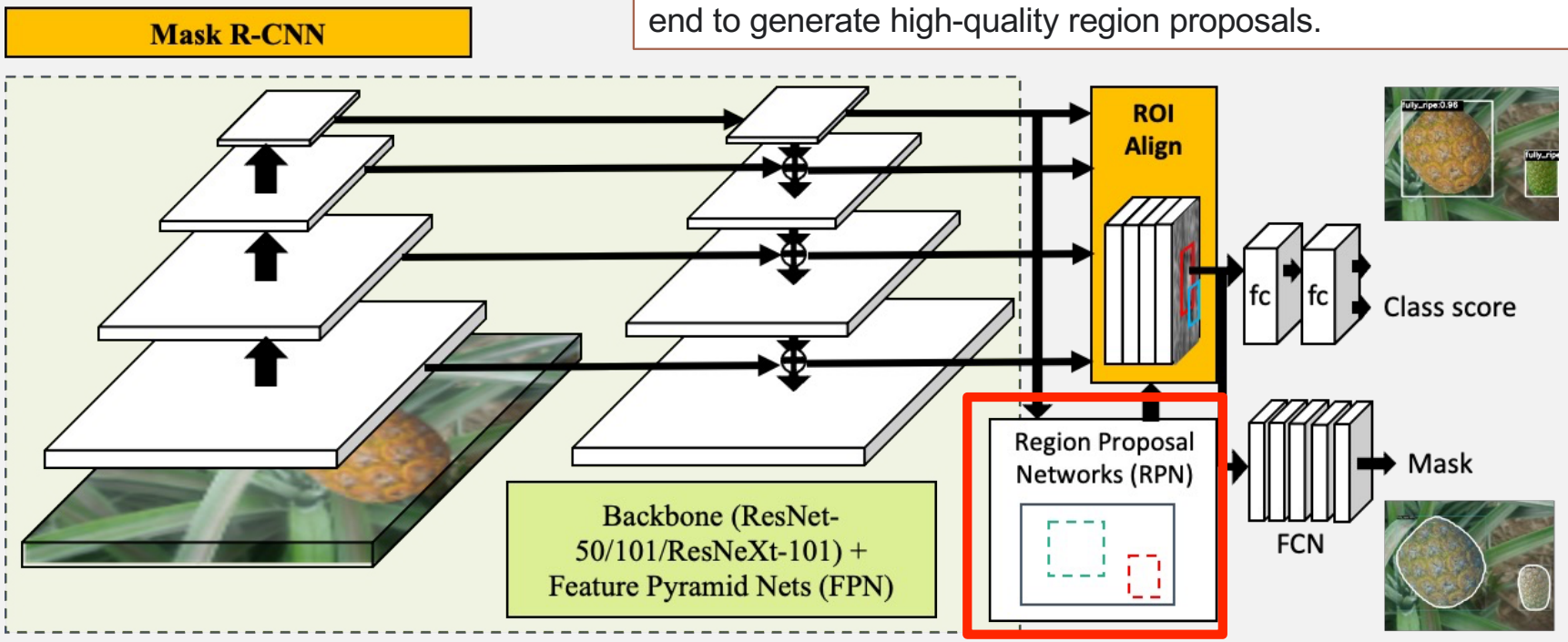
A Feature Pyramid Network, or FPN, is **a feature extractor that takes a single-scale image of an arbitrary size as input, and outputs proportionally sized feature maps at multiple levels, in a fully convolutional fashion**



K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 386–397, 2020, doi: 10.1109/TPAMI.2018.2844175.

MASK R-CNN

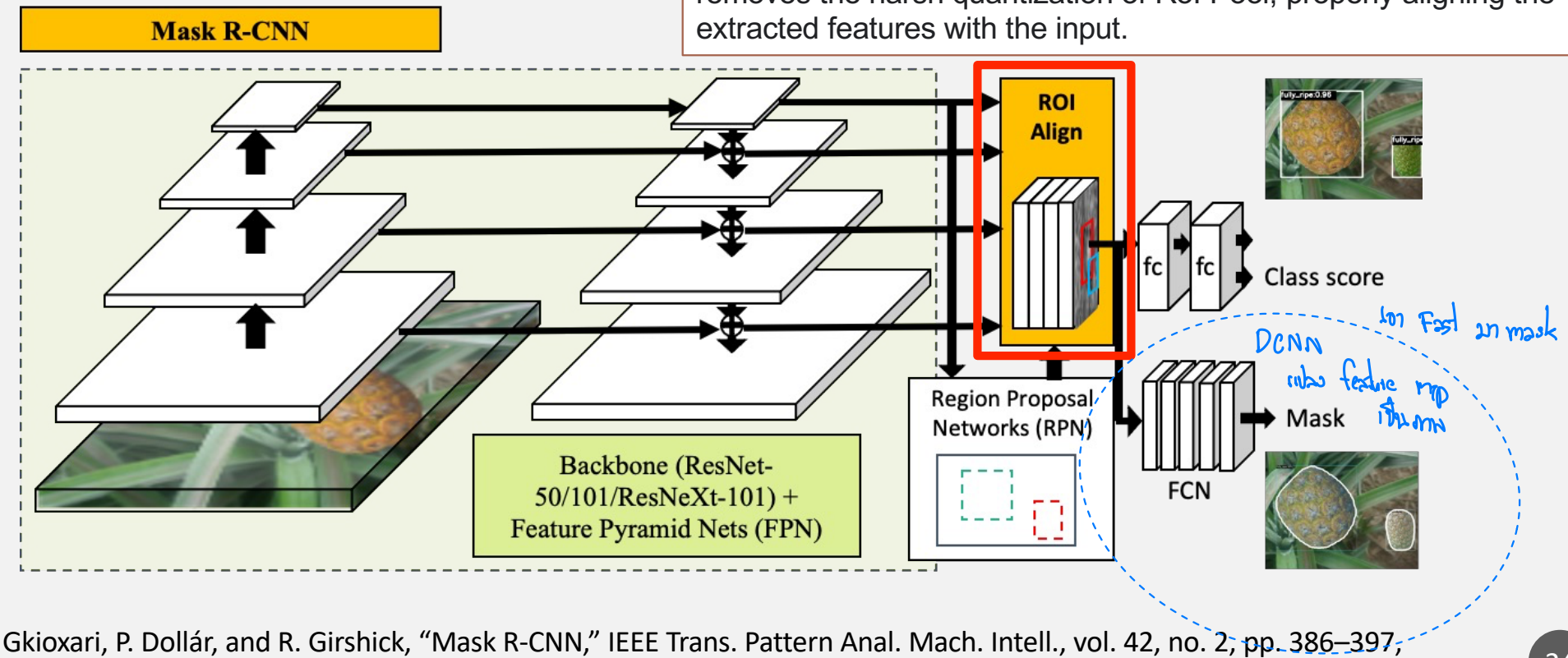
A Region Proposal Network, or RPN, is **a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position**. The RPN is trained end-to-end to generate high-quality region proposals.



K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 386–397, 2020, doi: 10.1109/TPAMI.2018.2844175.

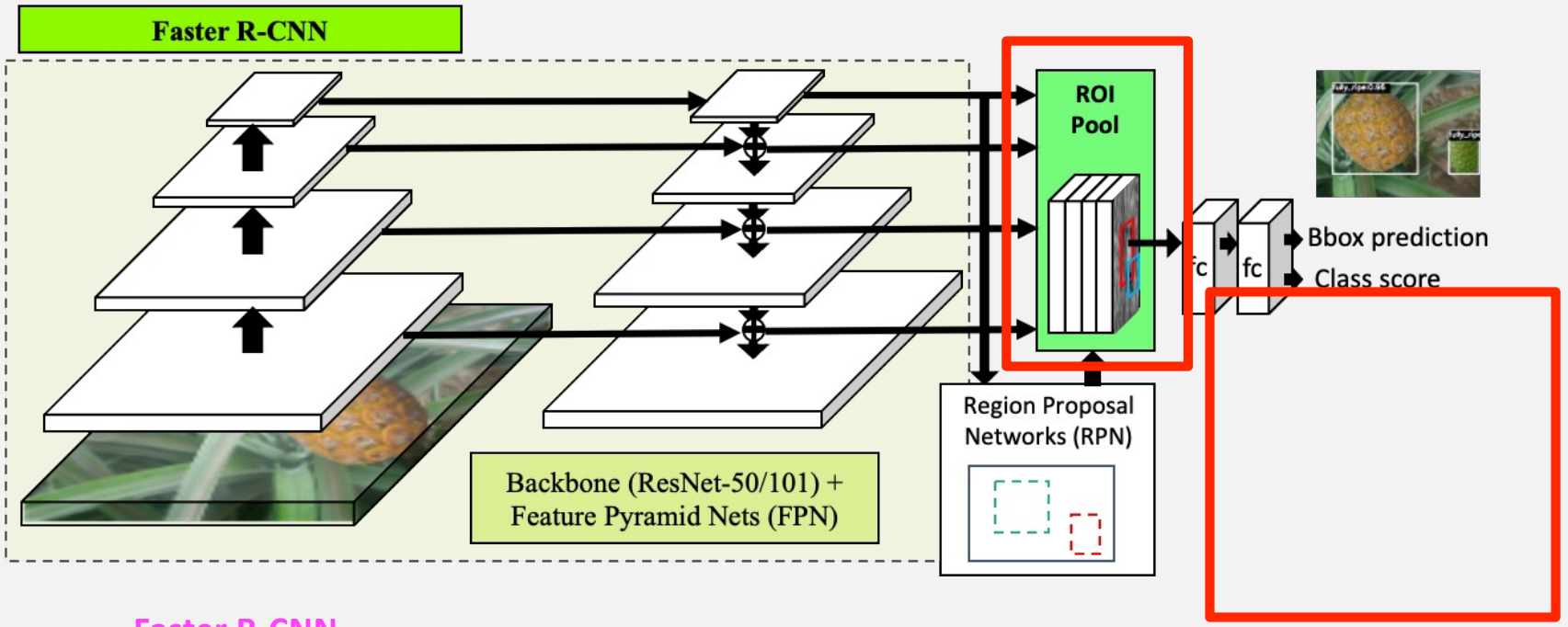
MASK R-CNN

Region of Interest Align, or RoIAlign, is **an operation for extracting a small feature map from each RoI in detection and segmentation based tasks (bilinear interpolation)**. It removes the harsh quantization of RoI Pool, properly aligning the extracted features with the input.



K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 2, pp. 386–397, 2020, doi: 10.1109/TPAMI.2018.2844175.

FASTER R-CNN



Faster R-CNN

For each candidate object, a class label and a bounding-box offset;

S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.

LOSS FUNCTION

Model	Classification Loss	Bounding Box Regression Loss	Mask Loss	Training Type
R-CNN	Softmax Cross-Entropy	L2 Loss	None	Separate (not end-to-end)
SPPNet	Softmax Cross-Entropy	L2 Loss	None	Separate (not end-to-end)
Fast R-CNN	Softmax Cross-Entropy	Smooth L1 Loss	None	End-to-End
Faster R-CNN	Binary Cross-Entropy (RPN), Softmax (Fast R-CNN)	Smooth L1 Loss for RPN and Fast R-CNN	None	End-to-End
Mask R-CNN	Binary Cross-Entropy (RPN), Softmax (Fast R-CNN)	Smooth L1 Loss for RPN and Fast R-CNN	Binary Cross-Entropy for Mask Prediction	End-to-End

Moreover, IoU is utilized to determine if the prediction box corresponds to the actual ground-truth box in evaluation.

SINGLE SHOT MULTIBOX DETECTOR (SSD)

- Multiple feature maps of different scales (from different layers of the network) to detect objects at various sizes.
- SSD introduces **default boxes** (or anchor boxes) of varying aspect ratios for each cell in the feature maps, allowing it to handle objects of different shapes and sizes within each layer.
- **VGG-16** as the backbone, but newer versions and adaptations use more advanced backbones like **MobileNet** (for SSD MobileNet) to make the model even faster and more lightweight,

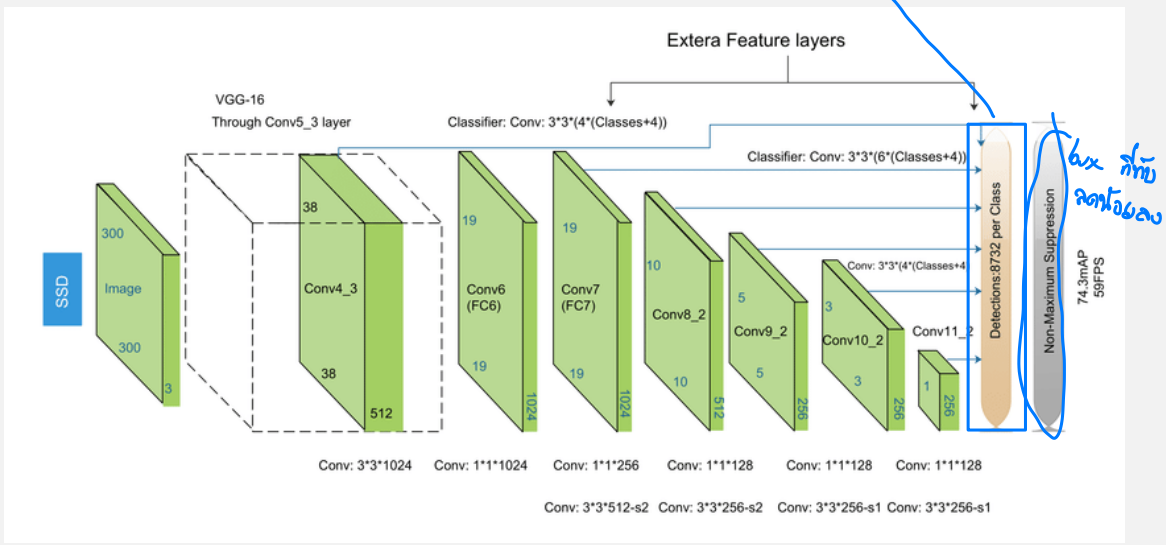


Figure 4. Single Shot Multi-Box Detector (SSD) architecture

Bahaghighat, Mahdi & Xin, Qin & Motamedi, Seyed & Mohammadi Zanjireh, Morteza & Vacavant, Antoine. (2020). Estimation of Wind Turbine Angular Velocity Remotely Found on Video Mining and Convolutional Neural Network. Applied Sciences. 10.3390/app10103544.

ONE-STAGE METHOD: YOU ONLY LOOK ONCE (YOLO)

- To apply **a single neural network** to the full image. This network divides the image into regions and predicts bounding boxes and probabilities for each region simultaneously.
- Great improvement in detection speed, YOLO suffers from a drop in localization accuracy - runs at 155 fps with VOC07 **mAP** = 52.7% and enhanced version VOC07 with mAP = 63.4%

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [38]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

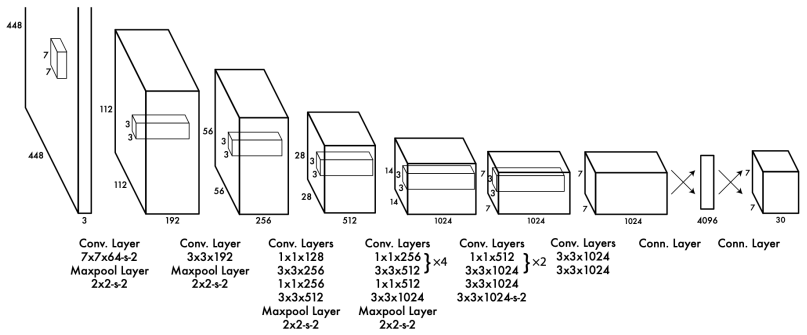


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

YOLOv1

backbone

- Network, inspired by GoogLeNet, has 24 convolutional layers followed by 2 fully connected layers (1x1 and 3x3).

ONE-STAGE METHOD: YOU ONLY LOOK ONCE (YOLO)

- The input image into an S × S grids.
- Each grid cell predicts B bounding boxes and confidence scores for those boxes.
- Confidence score reflect the box containing an object.
- Each bounding box consists of 5 predictions: x, y, w, h, and confidence,
  - x,y is center of the box, w and h - width and height
  - confidence prediction represents the IOU between the predicted box and any ground truth box.
  - Predict one set of class probabilities per grid cell, regardless of the number of boxes B.
- Loss fns : (1) loss for the coordinates of the object's center, (2) the dimensions of the bbox, the class of the object, (3) the class if the object is absent, and (4) the probabilities of finding some object in the bbox.

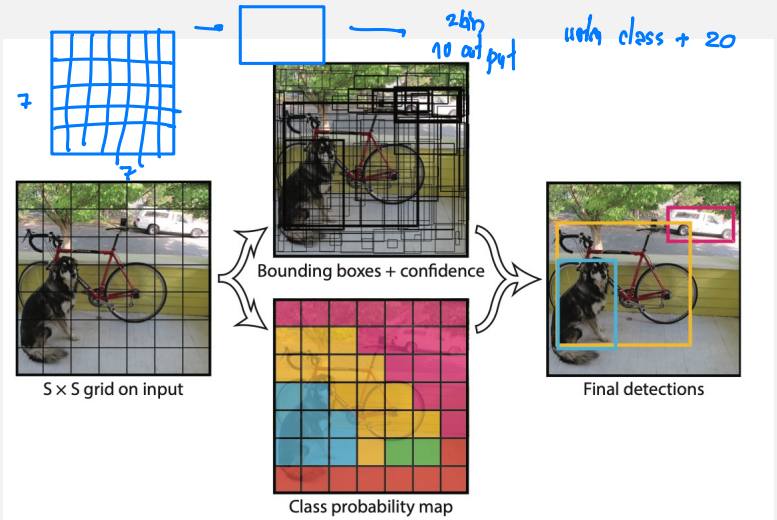


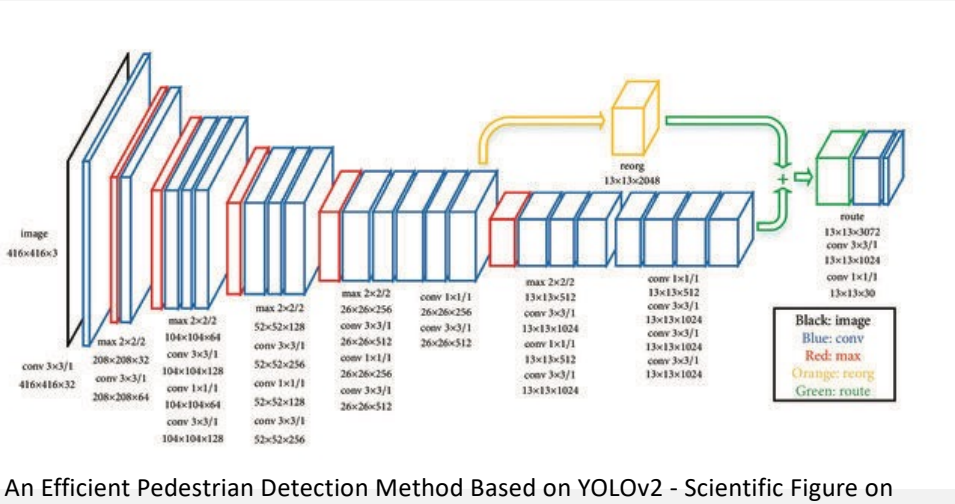
Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

For evaluating YOLO on PASCAL VOC, we use $S = 7$, $B = 2$. PASCAL VOC has 20 labelled classes so $C = 20$. Our final prediction is a $7 \times 7 \times 30$ tensor.

YOLO VARIANTS

YOLOv2

- Anchor Boxes: Introduced anchor boxes, allowing detection of objects with varying shapes and sizes.
  - YOLOv2 is configured with 5 anchor boxes, it will find 5 representative bounding box shapes using k-means clustering to determine the best set of anchor box dimensions.
- Batch Normalization: Added batch normalization layers to stabilize and speed up training, improving performance.
- Improved detection for objects of different sizes but **still struggled with overlapping** small objects.



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

The network architecture of YOLOv2.

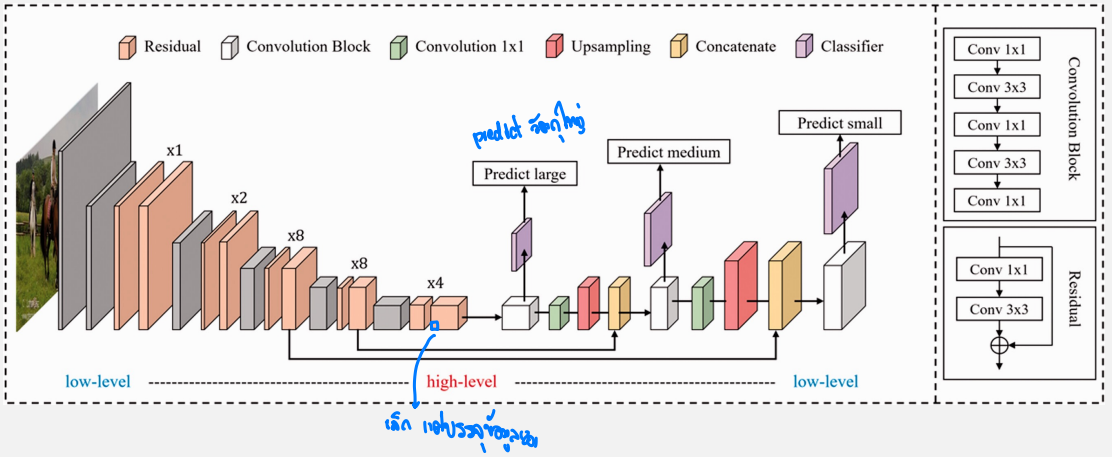
REORG layer

- Reshapes high-resolution feature maps by reducing spatial dimensions and increasing depth.
- Enables the integration of fine-grained details from early layers with high-level semantic information.

YOLO VARIANTS

YOLOv3

- **Multi-Scale Predictions**: Used three prediction scales (small, medium, large), significantly improving performance on small objects.
- **Darknet-53 Backbone**: Replaced the original backbone with Darknet-53, a deeper network that improved feature extraction. (designed to be lightweight and fast, making it suitable for real-time applications, and it uses mostly **3x3 convolutions** with some **1x1 convolutions** for channel reduction)

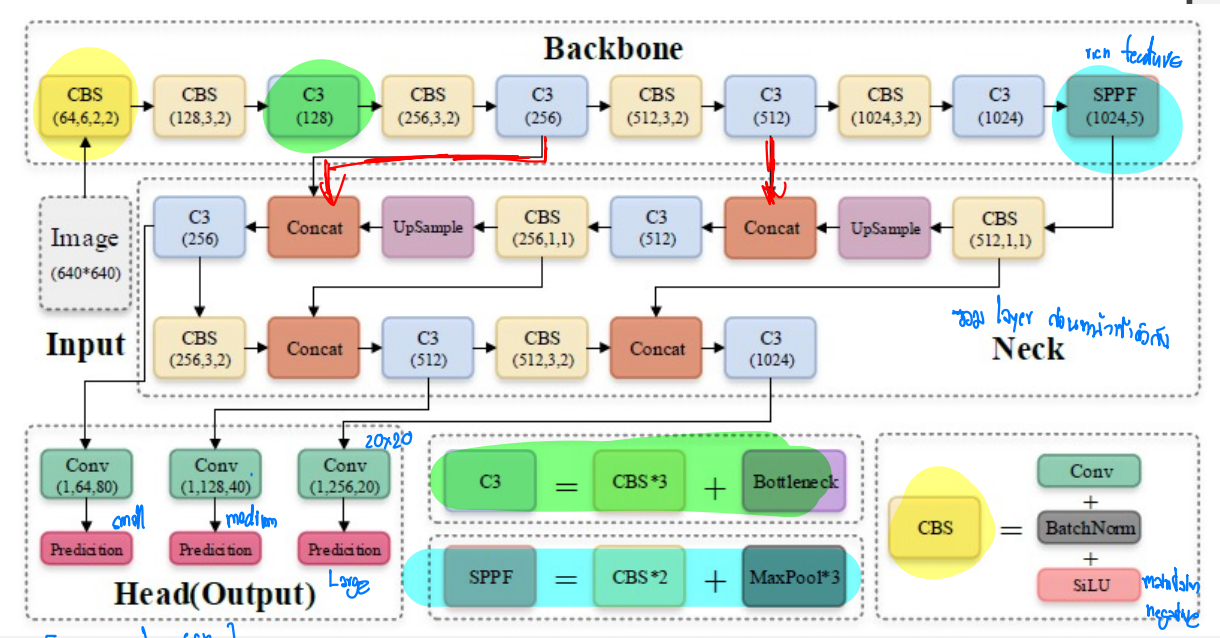


- **Binary Cross-Entropy for Classification**: Replaced softmax with binary cross-entropy for multi-label classification, making it more flexible.
- Improved Speed and Accuracy: Balanced speed and accuracy better than previous versions, making it highly popular for **real-time applications.**

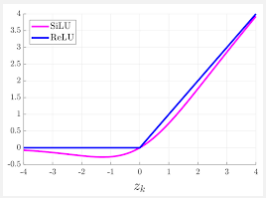
YOLO VARIANTS

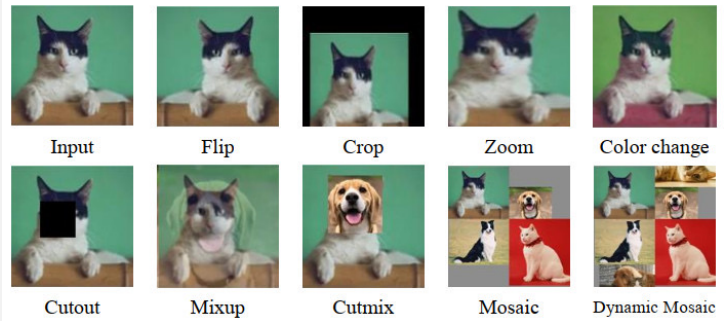
YOLOv5

- Simplified and Scalable: YOLOv5 was engineered to be lightweight and scalable, available in different model sizes (small to extra-large).
- Enhanced Data Augmentation: e.g., Mosaic and auto-learning anchor boxes, enhancing performance across diverse datasets.
- backbone is CSPDarknet53 - stacking of multiple CBS (Conv + BatchNorm + SiLU) modules and C3 modules, and finally one SPPF module is connected
  - SiLU allows a small negative output for small negative inputs, which can help the network retain some information from negative values.
- Easy Deployment / PyTorch Implementation



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



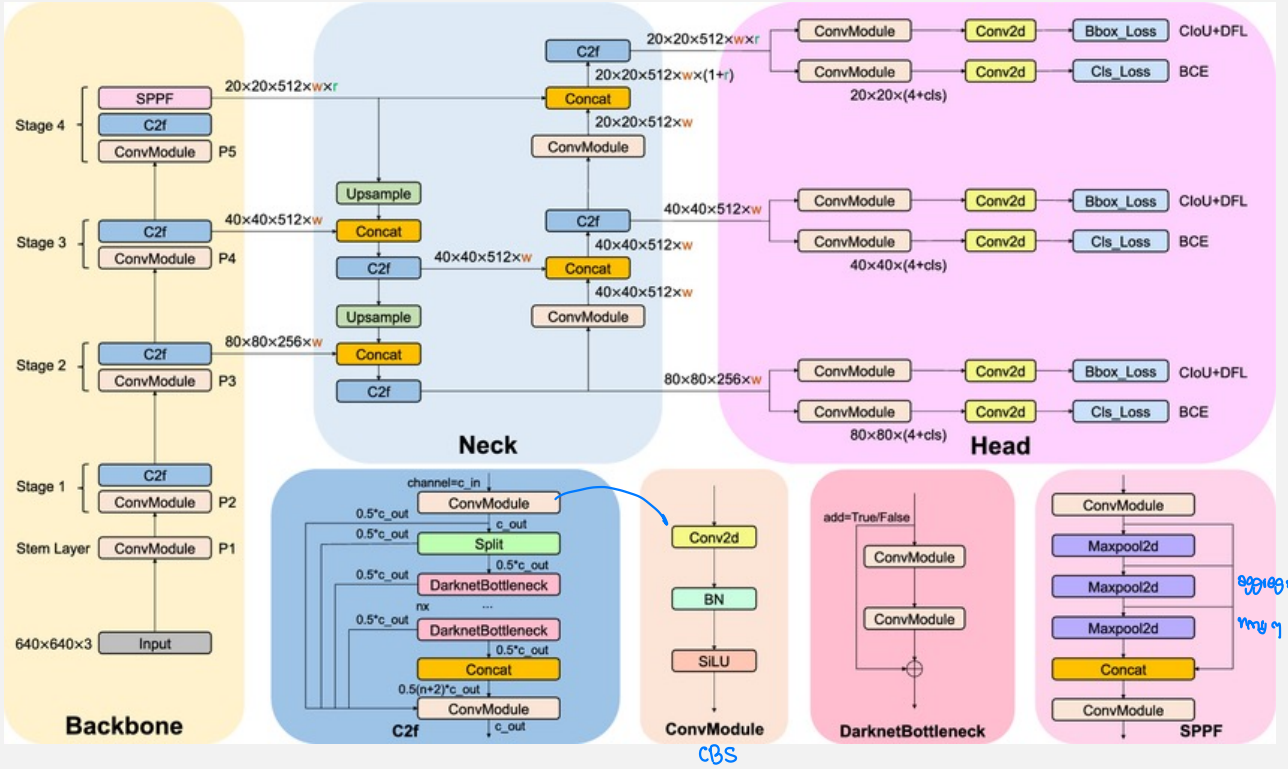


https://www.aimspress.com/article/doi/10.3934/mbe.2023311?viewType=HTML

YOLOV8

- **YOLOv8**

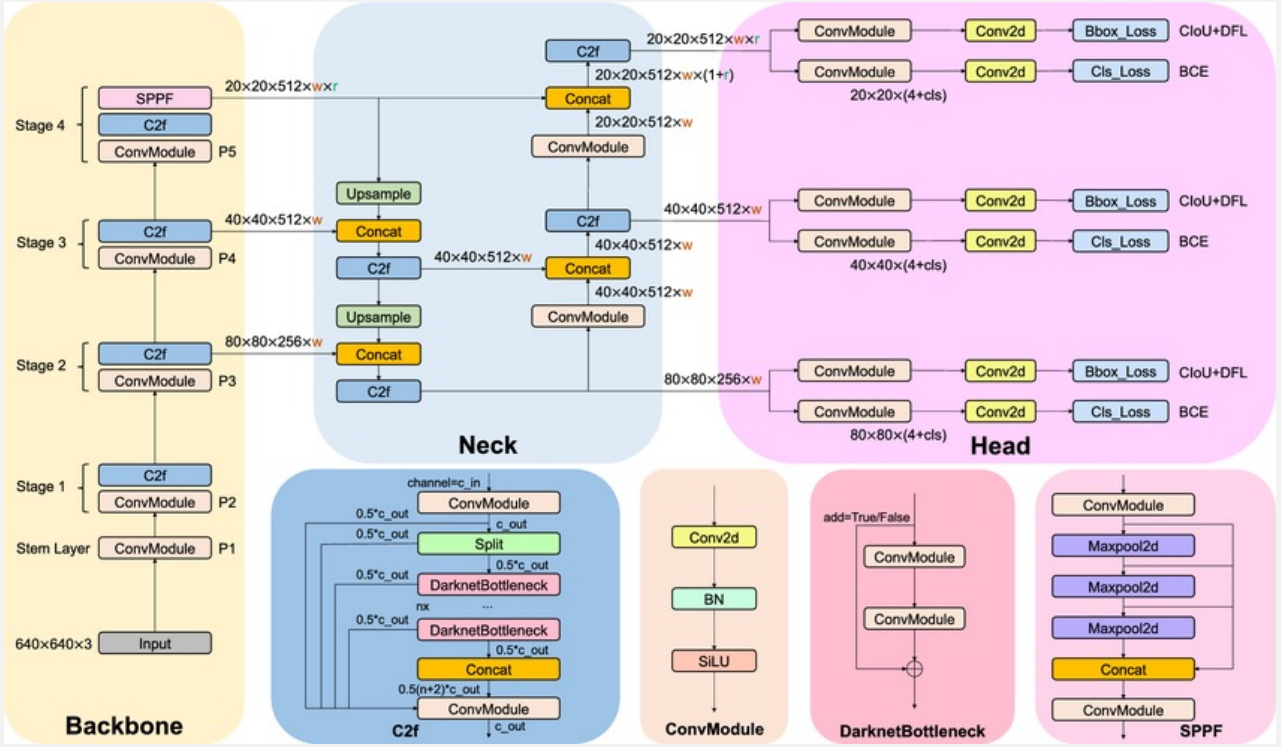
1. Input — an input layer to which an input image is fed
2. Backbone — a part in which the input image is encoded in the form of features.
3. Neck — here are additional parts of the model that process images encoded by features
4. Head(s) — one or more output layers that produce model predictions.



Split / Concat : the 'split' layer in YOLOv8 takes a feature map and splits it into two halves along the channel dimension. The split layer allows the network to create two separate paths for feature extraction, which are then recombined later.

YOLOV8

- **YOLOv8 Highlights**
  - **Anchor-free detection** approach ไม่ต้องเลือกว่าใช้ ขนาด
  - New Backbone and Head Architecture, Improved Training and Inference Speed, Enhanced Post-Processing with Non-Maximum Suppression (NMS)



Split / Concat : the 'split' layer in YOLOv8 takes a feature map and splits it into two halves along the channel dimension. The split layer allows the network to create two separate paths for feature extraction, which are then recombined later.

YOLOV11

- YOLOV11 - C3k2 block - Cross Stage Partial (CSP) Bottleneck - two smaller convolutions instead of one large - "k2" in C3k2 indicates a smaller kernel size
- Fast (SPPF) block from previous versions but introduces a new Cross Stage Partial with Spatial Attention (C2PSA) block after it

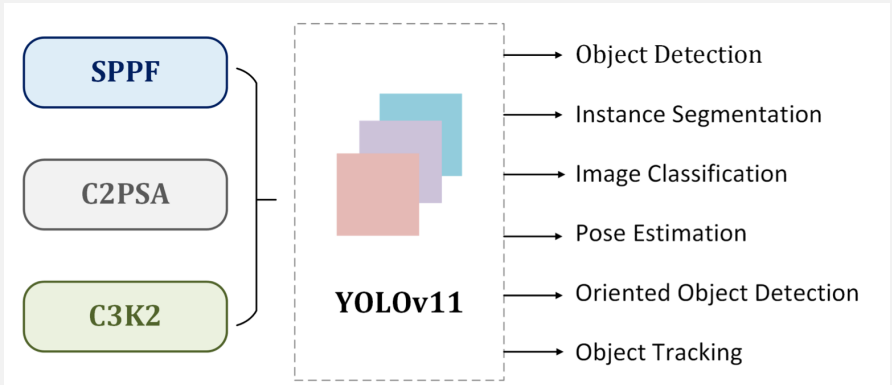
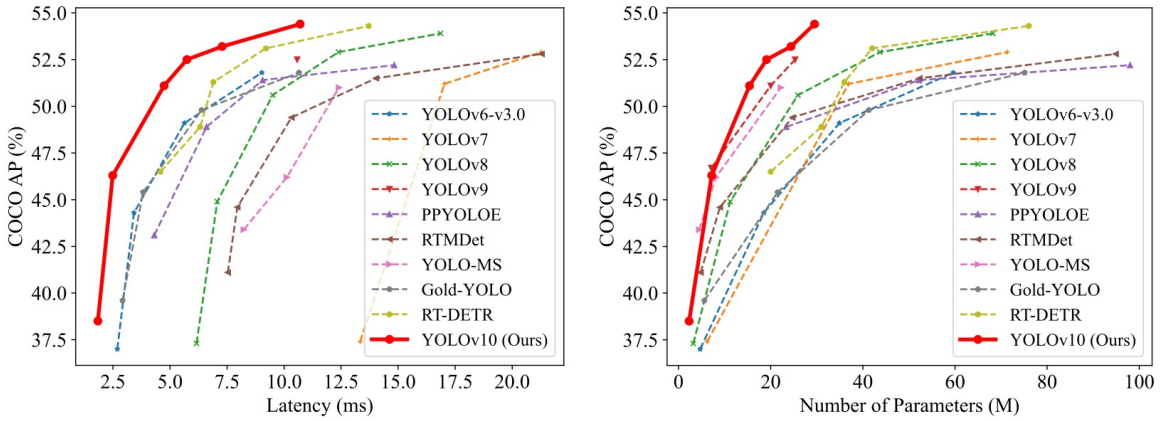
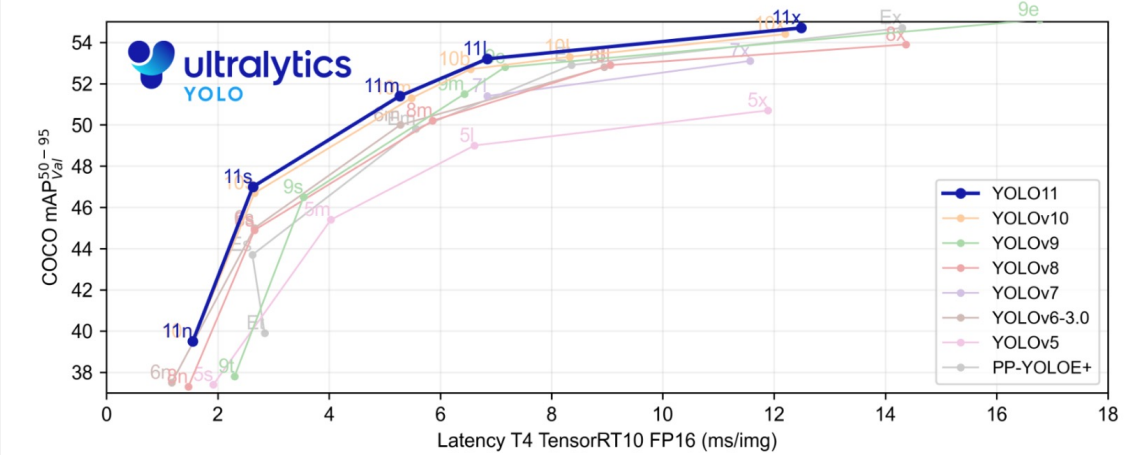


Figure 1: Key architectural modules in YOLO11





A. Wang *et al.*, "YOLOv10: Real-Time End-to-End Object Detection," *arXiv preprint arXiv:2405.14458*, 2024.

R. Khanam and M. Hussain, "YOLOv11: An Overview of the Key Architectural Enhancements," Oct. 2024, [Online]. Available: http://arxiv.org/abs/2410.17725

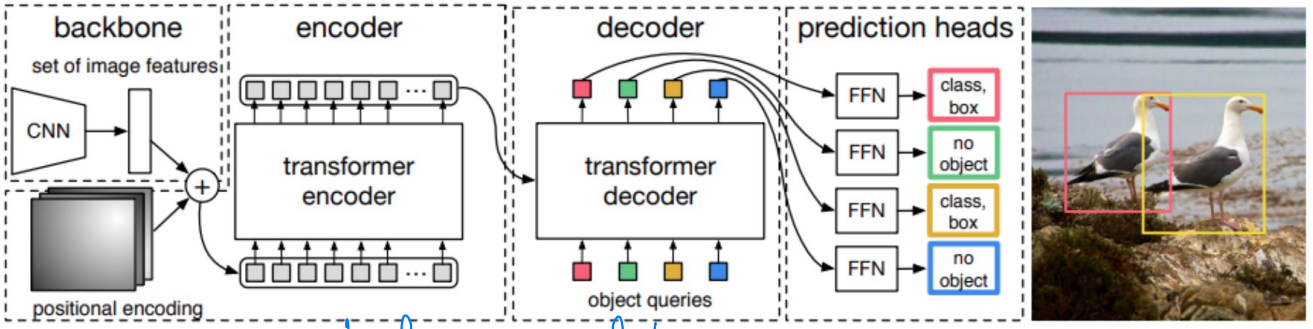
DETR / CO-DETR

- DETR – Detection Transformer
  - Facebook AI Research introduced Detection Transformers (DETR) (2020)
  - integrates Transformers as a central building block within the object detection pipeline.

Components:

1. CNN backbone - for feature extraction
2. Transformer encoder-decoder - reduces the channel dimension and applies multi-head self-attention
3. Feed-forward network (FFN) for final prediction

DETR reasons about all objects together using pair-wise relations, benefiting from the whole image context.



DETR architecture, featuring a backbone, a transformer encoder, a transformer decoder, and four prediction heads. Source.

End-to-End Object Detection with Transformers
Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, Sergey Zagoruyko

DETR / CO-DETR

DETRs with Collaborative Hybrid Assignments Training (CO-DETR) (2023)

- Too few positive queries in DETR during one-to-one matching leads to sparse supervision for the encoder, which negatively impacts the model's ability to learn discriminative features.
- Multiple **parallel auxiliary heads** are introduced during training, supervised by one-to-many label assignments, improving the encoder's discriminative feature learning.
  - provide additional positive samples that boost the training efficiency for the decoder.
  - extract positive coordinates from auxiliary heads to improve positive sample training in the decoder
- No hand-crafted non-maximum suppression (NMS)

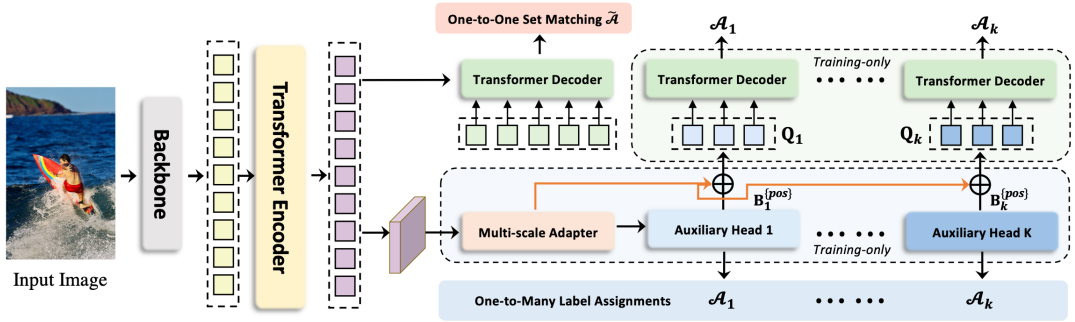
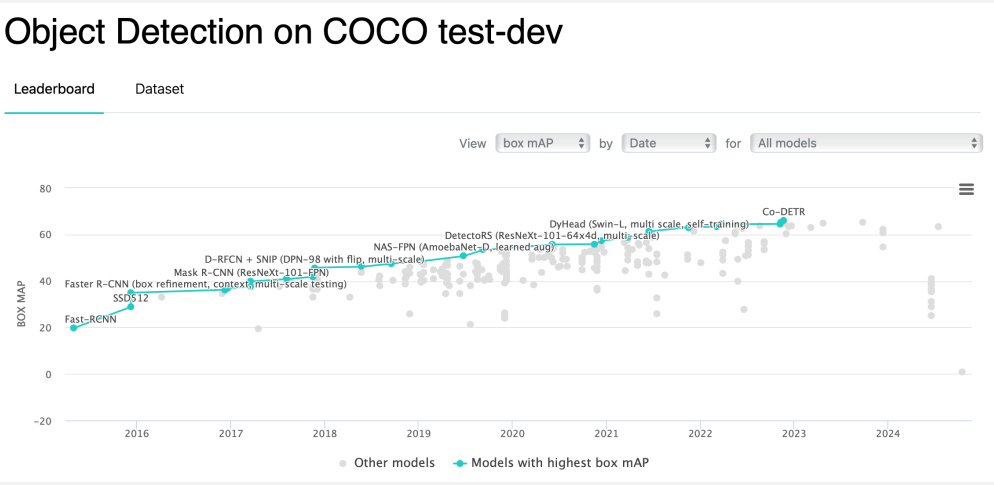


Figure 4. **Framework of our Collaborative Hybrid Assignment Training.** The auxiliary branches are discarded during evaluation.



DETRs with Collaborative Hybrid Assignments Training

Zhuofan Zong Guanglu Song Yu Liu* SenseTime Research

OBJECT DETECTION

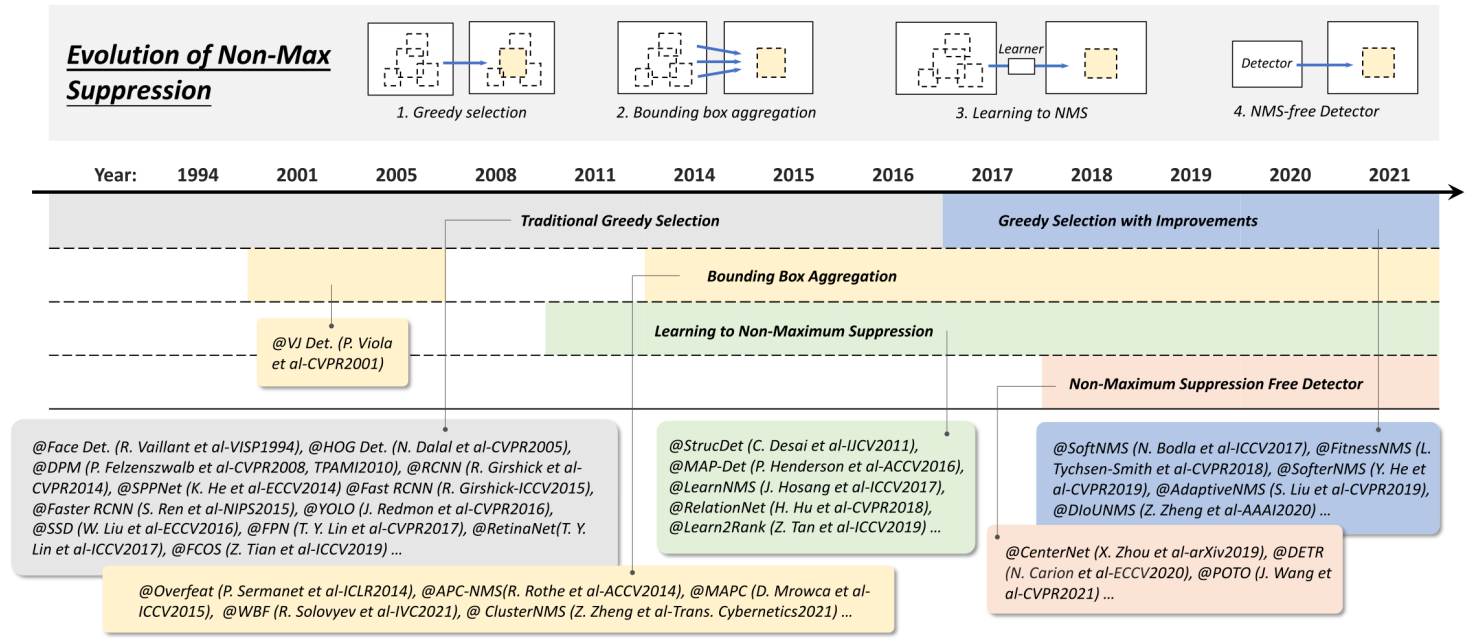


Fig. 8. ***Evolution of nonmax suppression (NMS) techniques in object detection from 1994 to 2021: 1) greedy selection; 2) bounding box aggregation; 3) learning to NMS; and 4) NMS-free detection. Detectors in this figure: Face Det. [108], HOG Det. [12], DPM [13], [15], RCNN [16], SPPNet [17], Fast RCNN [18], Faster RCNN [19], YOLO [20], SSD [23], FPN [24], RetinaNet [25], FCOS [41], StrucDet [85], MAP-Det [109], LearnNMS [110], RelationNet [93], Learn2Rank [111], SoftNMS [112], FitnessNMS [113], SofterNMS [114], AdaptiveNMS [115], DIoUNMS [107], Overfeat [65], APC-NMS [116], MAPC [117], WBF [118], ClusterNMS [119], CenterNet [40], DETR [28], and POTO [120].***

Z. Zou, K. Chen, Z. Shi, Y. Guo and J. Ye, "Object Detection in 20 Years: A Survey," in Proceedings of the IEEE, vol. 111, no. 3, pp. 257-276, March 2023, doi: 10.1109/JPROC.2023.3238524.

OBJECT DETECTION

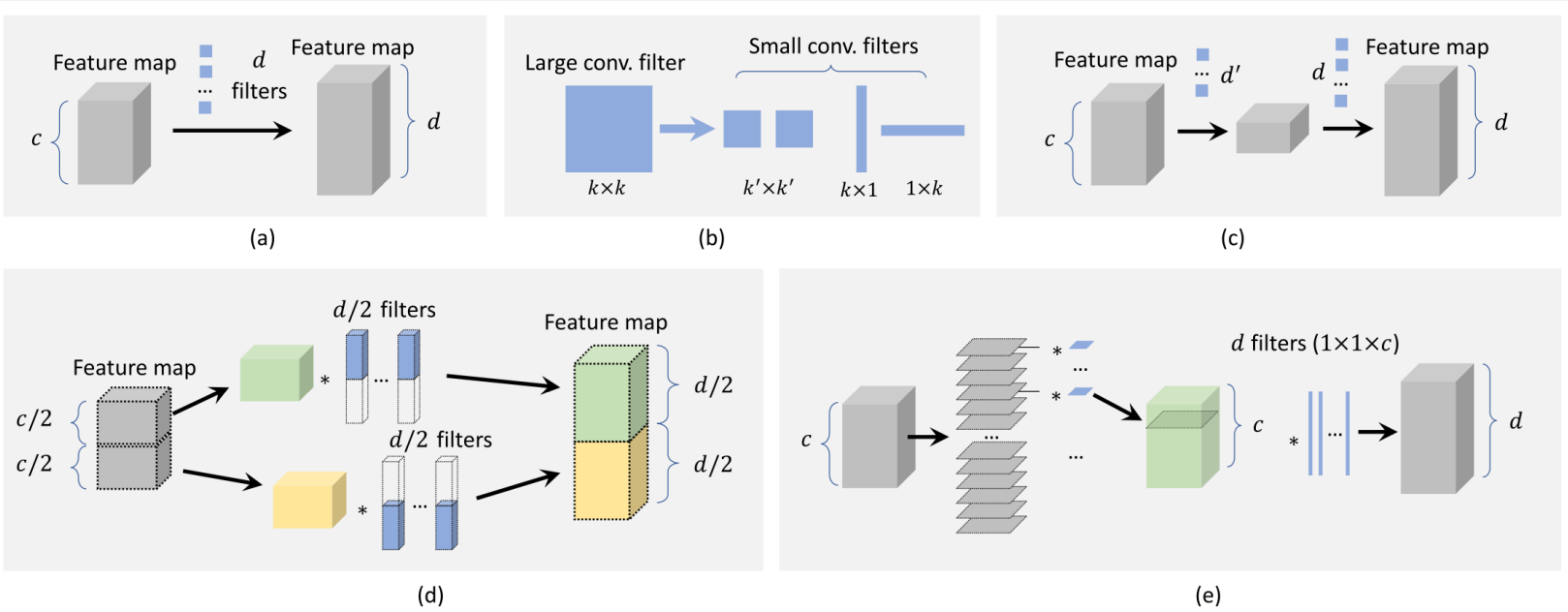


Fig. 10. ***Overview of speedup methods of a CNN's convolutional layer and the comparison of their computational complexity. (a) Standard convolution: $\mathcal{O}(dk^2c)$. (b) Factoring convolutional filters ($k \times k \rightarrow (k' \times k')^2$ or $1 \times k$, $k \times 1$): $\mathcal{O}(dk'^2c)$ or $\mathcal{O}(dkc)$. (c) Factoring convolutional channels: $\mathcal{O}(d'k^2c) + \mathcal{O}(dk^2d')$. (d) Group convolution (#groups = m): $\mathcal{O}(dk^2c/m)$. (e) Depthwise separable convolution: $\mathcal{O}(ck^2) + \mathcal{O}(dc)$.***

Z. Zou, K. Chen, Z. Shi, Y. Guo and J. Ye, "Object Detection in 20 Years: A Survey," in Proceedings of the IEEE, vol. 111, no. 3, pp. 257-276, March 2023, doi: 10.1109/JPROC.2023.3238524.

OBJECT DETECTION

Backbone

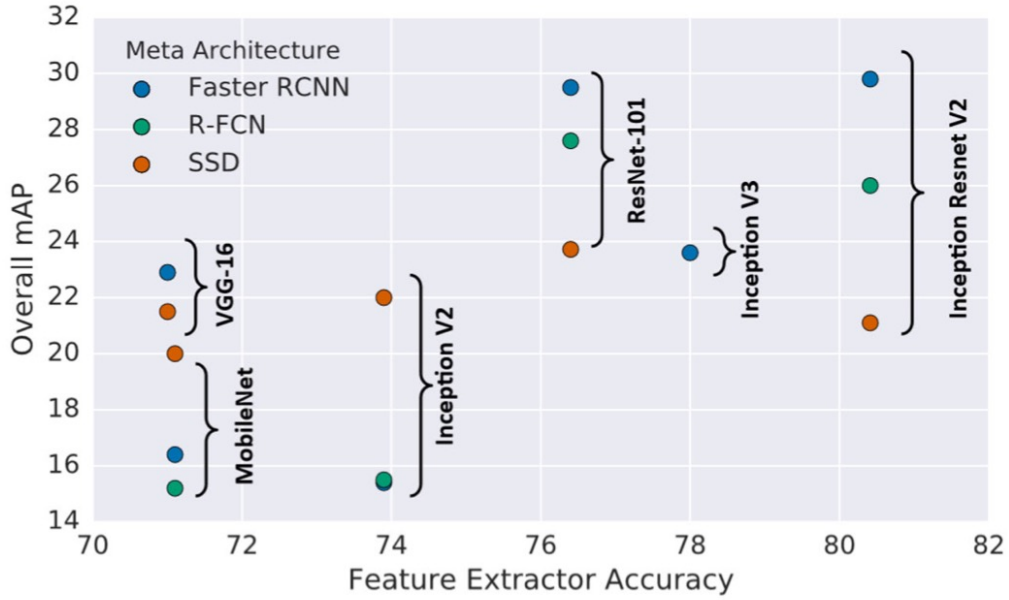


Fig. 13. ***Comparison of detection accuracy of three detectors: Faster RCNN [19], R-FCN [49], and SSD [23] on the MS-COCO dataset with different detection backbones. Image from CVPR 2017 [186].***

FUTURE DIRECTIONS

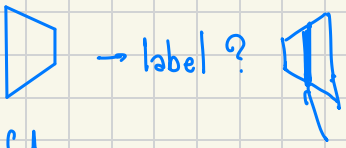
- Lightweight Object Detection - run on low-power edge devices
- Small Object Detection - people in crowd or animals in the open air / detecting military targets from satellite images
- 3-D Object Detection, e.g., autonomous driving
- Detection in Videos - real-time object detection/tracking in HD video surveillance and autonomous driving
- Cross-Modality Detection
- Toward Open-World Detection - Out-of-domain generalization, zero-shot detection, and incremental detection are emerging topics in object detection.

Z. Zou, K. Chen, Z. Shi, Y. Guo and J. Ye, "Object Detection in 20 Years: A Survey," in Proceedings of the IEEE, vol. 111, no. 3, pp. 257-276, March 2023, doi: 10.1109/JPROC.2023.3238524.

CODE

- https://colab.research.google.com/drive/1wHPJtW41NIVpF0msITOAym_kMJkasumA?usp=sharing

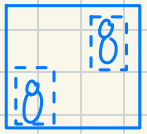
CNN , resnet, efficient Net , visual transformer



de - CNN = upsampling image

skip connection ระหว่าง low level กับ high level

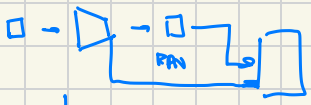
"Segmentation" UNET



object detection

- 2 - stage → selective search (RCNN, SPPNet) + RPN (หลักแสน)

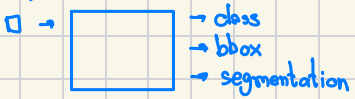
region pyramid network



segment เรียกใช้ใน mask R-CNN

- 1 - stage

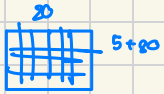
SSD, "YOLO"



coco = 5 + 80 = 85

coco category

20×20 คือ overall - Grid output



Focal loss ใช้กับ imbalance data

CIoU = คิด center + Aspect Ratio

SIoU คิด skewness (เอียงหรือเปล่า)

EIoU ใช้ width penalty ไม่ได้คิด skewness

keypoint loss คือการหาตำแหน่ง ไม่ได้เป็นกลุ่ม px