REPORT

Human Motion Generation: A Survey

Presented by: Nguyen Duy Tan

Ho Chi Minh City – 2025

Table of Content

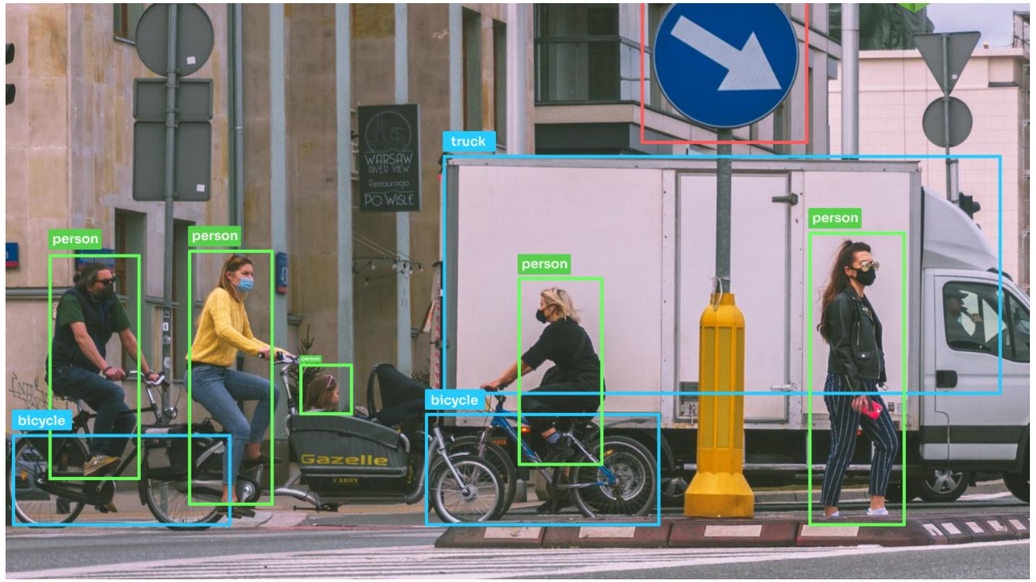
I. Overview

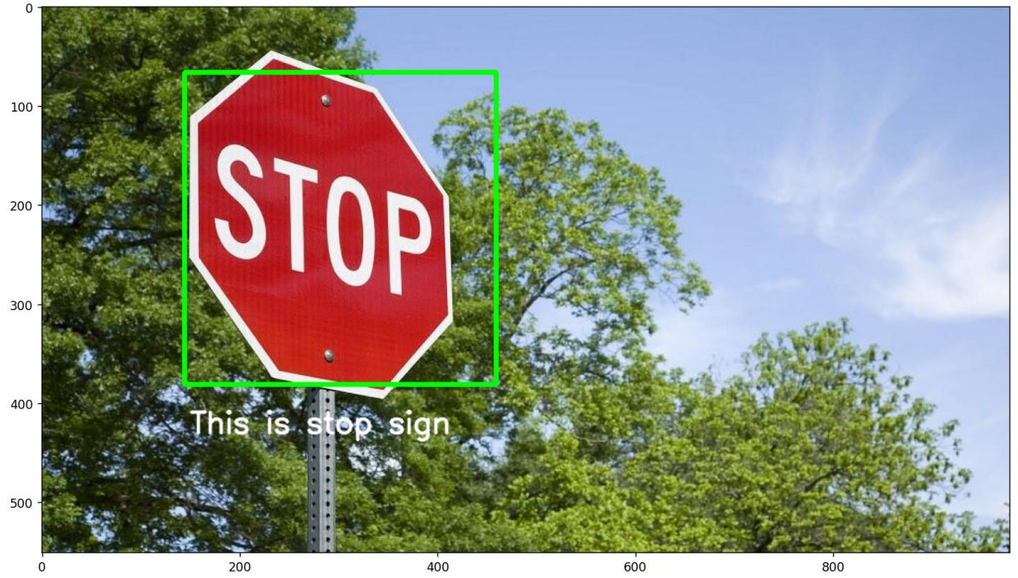
I. Overview

Object Detection and Bounding Boxes

Object Detection

- Object Detection (gọi tắt là OD) sẽ trả lời cho câu hỏi “đó là đối tượng nào và đối tượng đó nằm ở đâu” hay what? và where?
- OD có ứng dụng rộng rãi trong nhiều lĩnh vực an ninh, y tế, robotics, …





I. Overview

Object Detection and Bounding Boxes

Bounding Boxes

- Bounding Boxes dùng để mô tả vị trí không gian của một object.
- Có dạng hình chữ nhật, được xác định bởi cặp tọa độ (x, y) của góc trên bên trái hình chữ nhật và tọa độ tương ứng của góc dưới bên phải hình chữ nhật.
- Nhưng cách thường được sử dụng là cặp tọa độ (x, y) của tâm bounding boxes cùng với chiều dài và chiều rộng của bounding boxes.

I. Overview

Object Detection and Bounding Boxes

Bounding Boxes

```
# @save
def box_corner_to_center(boxes):
    """Chuyển từ (góc trên-trái, góc dưới-phải) sang (tâm, chiều rộng, chiều cao)."""
    x1, y1, x2, y2 = boxes[:, 0], boxes[:, 1], boxes[:, 2], boxes[:, 3]
    cx = (x1 + x2) / 2
    cy = (y1 + y2) / 2
    w = x2 - x1
    h = y2 - y1
    boxes = torch.stack((cx, cy, w, h), axis=-1)
    return boxes
```

I. Overview

Object Detection and Bounding Boxes

Bounding Boxes

```
# @save
def box_center_to_corner(boxes):
    """Chuyển từ (tâm, chiều rộng, chiều cao) sang (góc trên-trái, góc dưới-phải)."""
    cx, cy, w, h = boxes[:, 0], boxes[:, 1], boxes[:, 2], boxes[:, 3]
    x1 = cx - 0.5 * w
    y1 = cy - 0.5 * h
    x2 = cx + 0.5 * w
    y2 = cy + 0.5 * h
    boxes = torch.stack((x1, y1, x2, y2), axis=-1)
    return boxes
```

I. Overview

Anchor Boxes

Generating Multiple Anchor Boxes

- Nếu ảnh có kích thước $h \times w$, cho $scale\ s$ và $ratio\ r$ thì chiều rộng anchor là $ws\sqrt{r}$ và chiều dài anchor là $hs/\sqrt{r}$.
- Nếu có $n\ scales$ và $m\ ratios$, thường dung các kết hợp chứa s_1 hoặc r_1 để giảm số lượng.
- Số anchor trên cùng một pixel là $n + m - 1$.
- Số anchor trên một ảnh là $wh(n + m - 1)$.

I. Overview

Anchor Boxes

Intersection over Union (IoU)

Jaccard Index đo lường độ tương đồng giữa hai tập. Với hai tập $\mathcal{A}$ và $\mathcal{B}$, chỉ số Jaccard được định nghĩa bằng diện tích giao chia cho diện tích hợp, tức là

$$J(\mathcal{A},\mathcal{B}) = \frac{|\mathcal{A} \cap \mathcal{B}|}{|\mathcal{A} \cup \mathcal{B}|}$$

Ta có thể coi tập các pixel nằm trong một bounding box là một tập các điểm ảnh. Khi đó, độ tương đồng giữa hai bounding box được đo bởi chỉ số Jaccard của hai tập pixel tương ứng. Với hai bounding box, chỉ số Jaccard thường được gọi là **intersection over union (IoU)** — tỉ lệ diện tích giao trên diện tích hợp của hai bounding box.

I. Overview

Anchor Boxes

Labeling Anchor Boxes in Training Data

Assigning Ground-Truth Bounding Boxes to Anchor Boxes

❖ Tập huấn luyện cung cấp vị trí của các bounding box thực tế và lớp của các đối tượng bên trong. Để gán nhãn cho một anchor box đã sinh ra, ta tham chiếu tới bounding box thực tế **được gán (assigned)** cho anchor box đó — tức bounding box có IoU lớn nhất (gần nhất) với anchor box.

Labeling Classes and Offsets

❖ Giả sử anchor A được gán với ground-truth B. Khi đó:

Nhãn class của anchor A = nhãn class của B.

Nhãn offset của anchor A dựa trên vị trí tương đối giữa tâm của B và A, cùng với tỉ lệ kích thước tương đối giữa hai hộp.

I. Overview

Anchor Boxes

Predicting Bounding Boxes with Non-Maximum Suppression

- Mục tiêu: loại các bounding box dự đoán quá giống nhau (chồng lấp) và giữ những box mạnh nhất.
- Thuật toán:

  Sắp xếp các box theo **confidence** giảm dần.

  Lấy box mạnh nhất B_1, loại (suppress) mọi box có IoU với $B_1 > \varepsilon$.

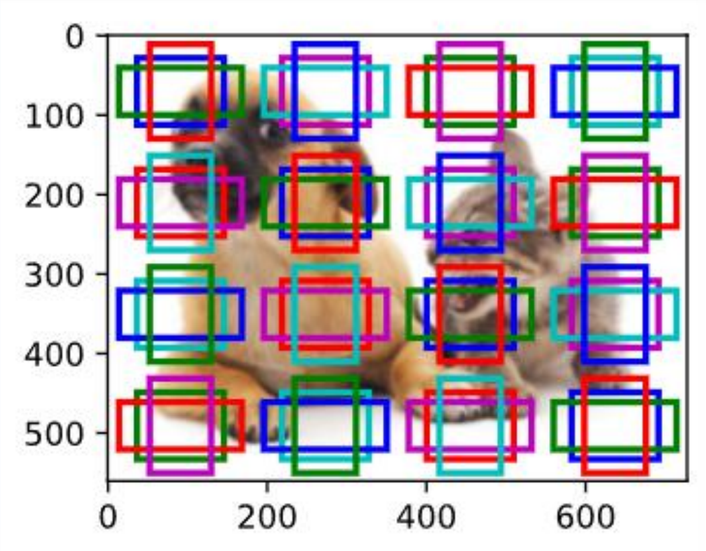
  Lặp với phần còn lại; kết quả là các box giữ lại có **pairwise** $\mathrm{IoU} \leq \varepsilon$.

I. Overview

Multiscale Object Detection

Multiscale Anchor Boxes

- Multiscale Anchor Boxes là kỹ thuật tạo ra nhiều hộp tham chiếu (anchor) với kích thước và tỉ lệ khác nhau, giúp mô hình phù hợp với nhiều loại đối tượng.
- Mỗi pixel trên feature map sẽ sinh ra nhiều anchor box dựa trên danh sách các **kích thước (sizes)** và **tỉ lệ (aspect ratios)** đã chọn.



I. Overview

Multiscale Object Detection

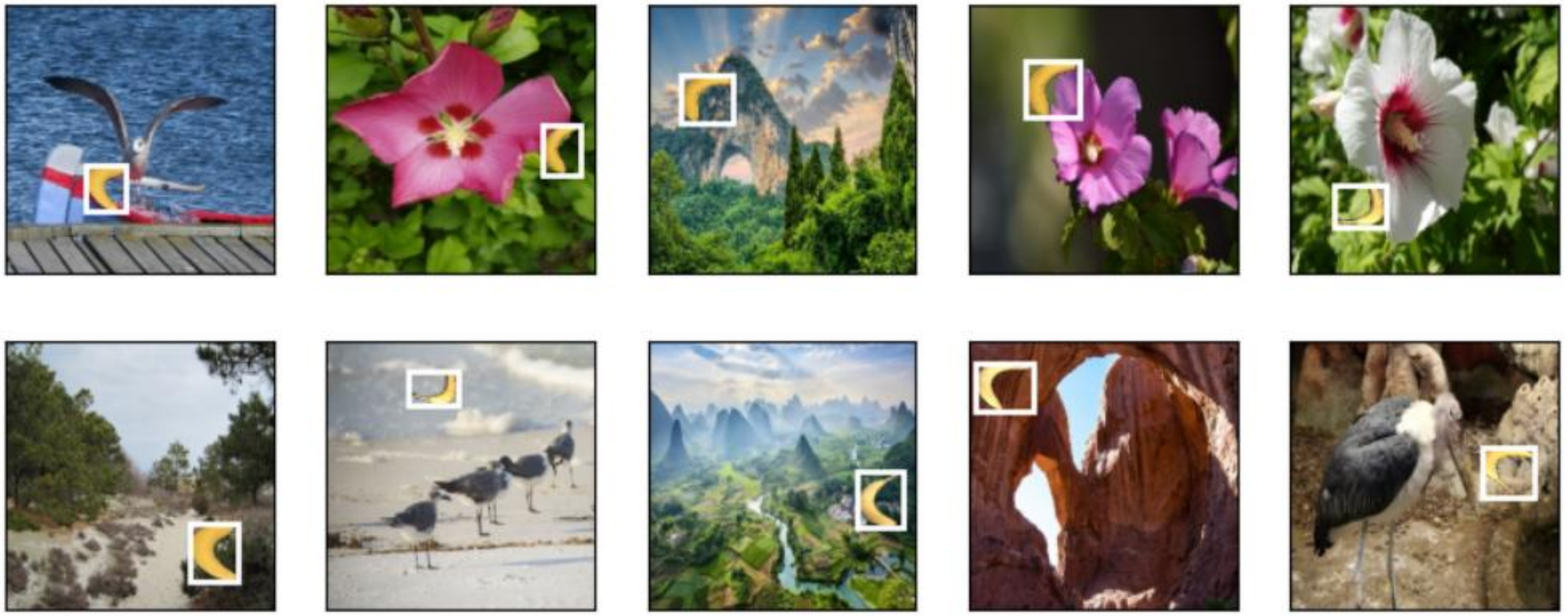
Multiscale Detection

- Phát hiện đa tỉ lệ nhằm mục tiêu nhận diện được đối tượng ở mọi kích thước.
- Thay vì chỉ dựa vào một feature map, mô hình sẽ sử dụng nhiều feature map ở các tầng khác nhau của mạng CNN.
- Các tầng nông tạo ra feature map lớn, phù hợp phát hiện đối tượng nhỏ; trong khi các tầng sâu tạo ra feature map nhỏ, phù hợp phát hiện đối tượng lớn.
- Mỗi tầng sẽ sinh anchor box với kích thước phù hợp, giúp cải thiện độ chính xác cho cả đối tượng rất nhỏ lẫn rất lớn.

I. Overview

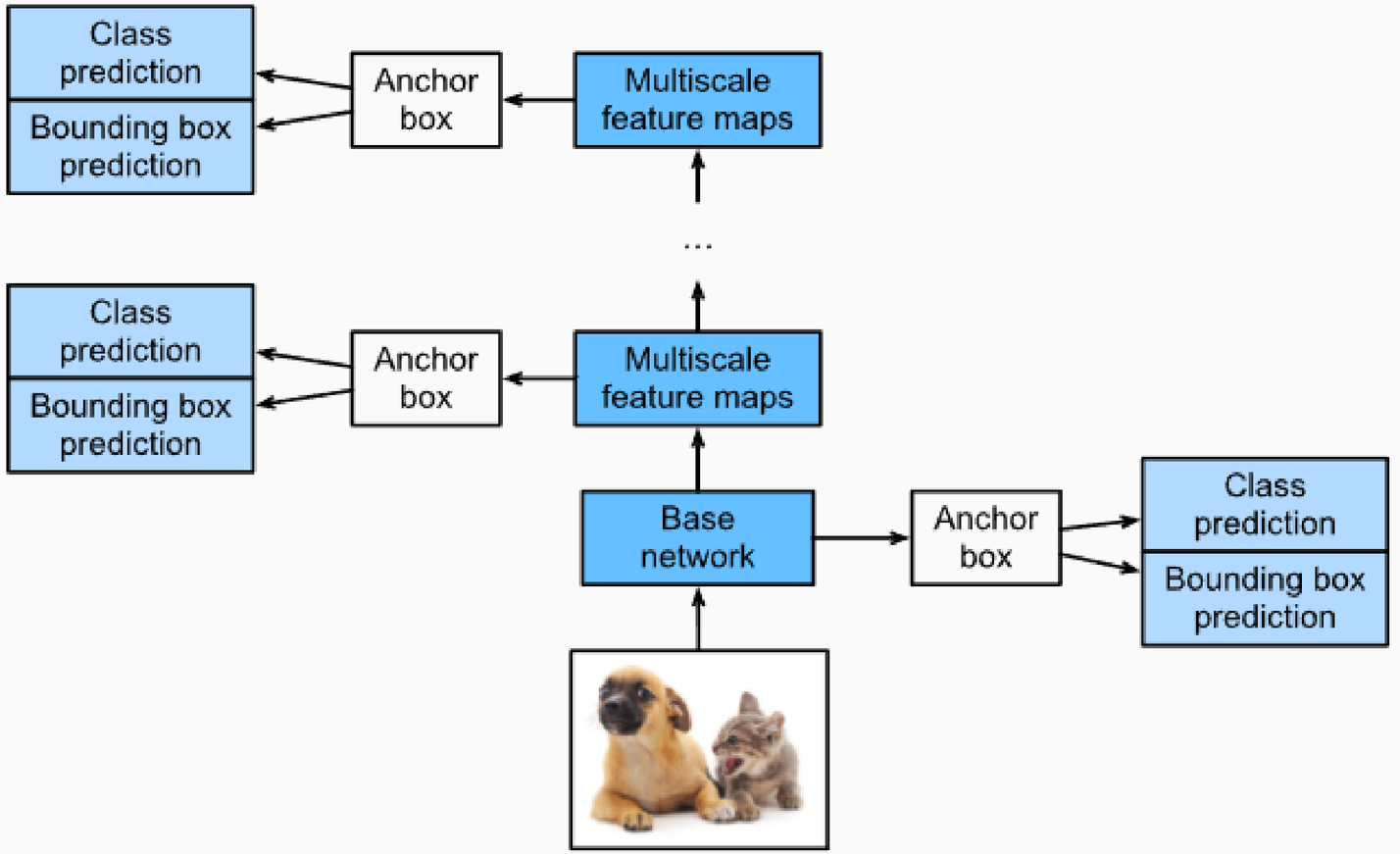
The Object Detection Dataset

Demonstration



I. Overview

Single Shot Multibox Detection - SSD



14.7. Single Shot Multibox Detection — Dive into Deep Learning 1.0.3 documentation

[1512.02325] SSD: Single Shot MultiBox Detector

II. Viola – Jones Algorithm

II. Viola – Jones Algorithm

Introduction

ACCEPTED CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION 2001

Rapid Object Detection using a Boosted Cascade of Simple Features

Paul Viola
viola@merl.com
Mitsubishi Electric Research Labs
201 Broadway, 8th FL
Cambridge, MA 02139

Michael Jones
mjones@crl.dec.com
Compaq CRL
One Cambridge Center
Cambridge, MA 02142

paper.dvi

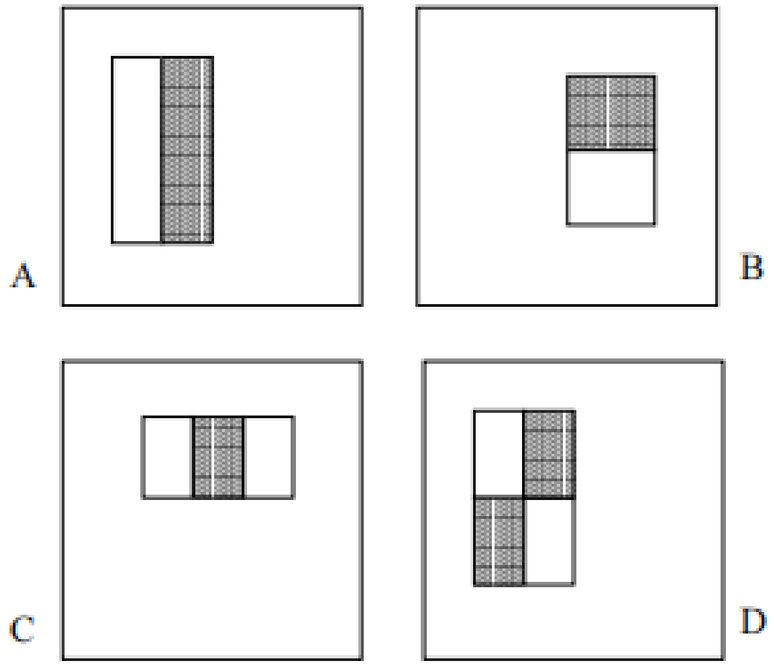
II. Viola – Jones Algorithm

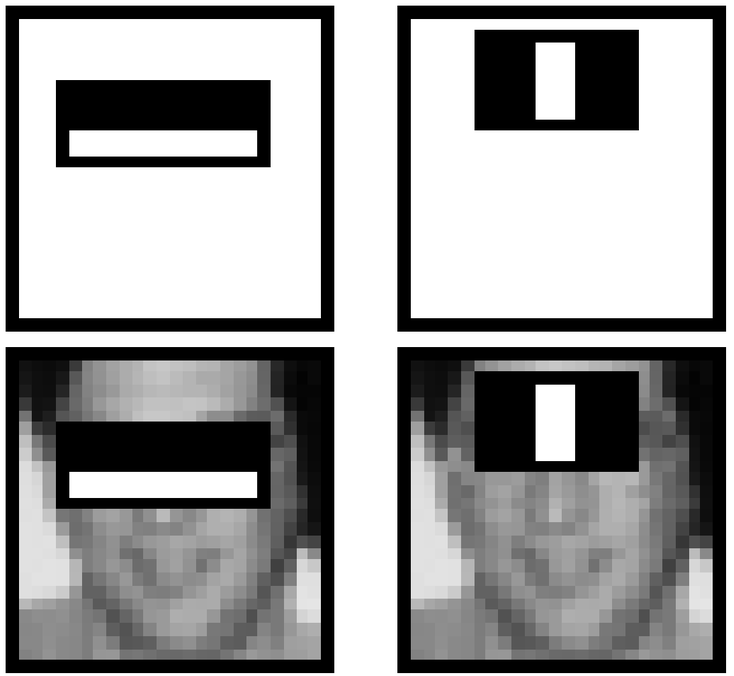
Introduction

Sử dụng cửa sổ trượt (sliding windows) với kích thước ban đầu thường là 24×24 pixels, sau đó quét qua hình ảnh ở các tỷ lệ và vị trí khác nhau, các cửa sổ được tạo động và kiểm tra bằng cascade classifiers dựa trên **Haar-like features**.

II. Viola – Jones Algorithm

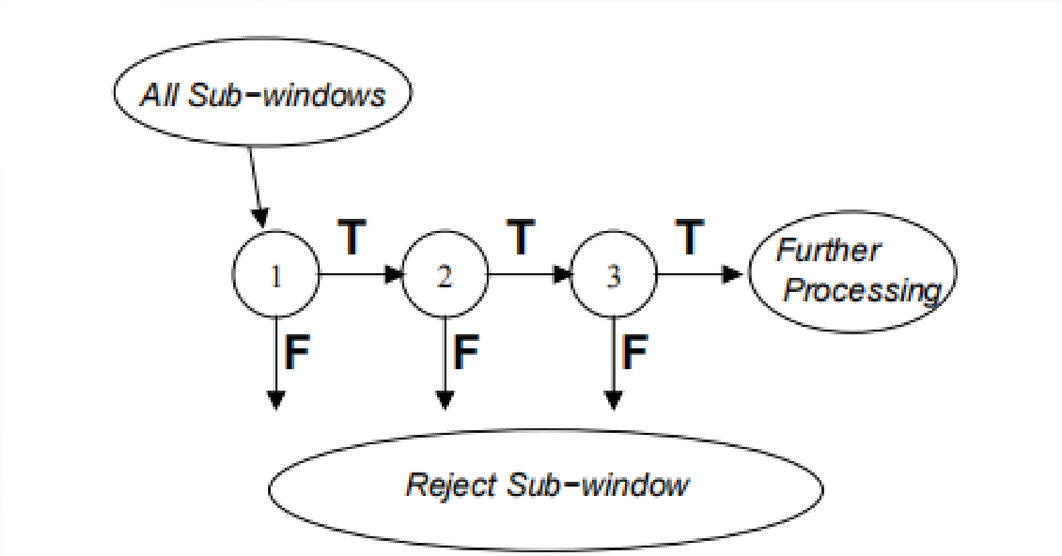
Haar-like Features





II. Viola – Jones Algorithm

Cascade

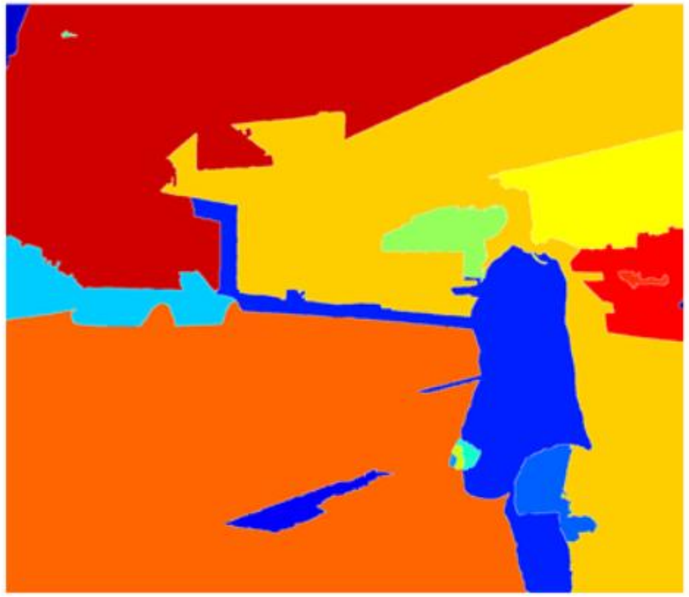


Hình ảnh minh họa về **detection cascade**. Một chuỗi các bộ phân loại được áp dụng cho mỗi cửa sổ con. Bộ phân loại ban đầu loại bỏ một số lượng lớn các ví dụ tiêu cực với rất ít xử lý. Các lớp tiếp theo loại bỏ thêm các ví dụ tiêu cực nhưng yêu cầu tính toán thêm. Sau vài giai đoạn xử lý, số lượng cửa sổ con đã **giảm đáng kể**. Việc xử lý thêm có thể dưới bất kỳ hình thức nào như các giai đoạn bổ sung của chuỗi (như trong hệ thống phát hiện của chúng tôi) hoặc một hệ thống phát hiện thay thế.

II. R-CNN Series

II. R-CNN Series

Region Proposal



Một số "*phương pháp sinh vùng đề xuất độc lập với loại đối tượng, ví dụ: objectness [1], selective search [32], category-independent object proposals [11], constrained parametric min-cuts (CPMC) [5], multi-scale combinatorial grouping [3], và Cireşan et al. [6] (dùng CNN để dò tế bào phân bào từ các crop vuông đều nhau, cũng là một dạng region proposal).*"

Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation

II. R-CNN Series

Selective Search

Algorithm 1: Hierarchical Grouping Algorithm

Input: (colour) image
Output: Set of object location hypotheses L

Obtain initial regions $R = \{r_1, \cdots, r_n\}$ using [13]
Initialise similarity set $S = \emptyset$
foreach *Neighbouring region pair* (r_i, r_j) **do**
    Calculate similarity $s(r_i, r_j)$
    $S = S \cup s(r_i, r_j)$

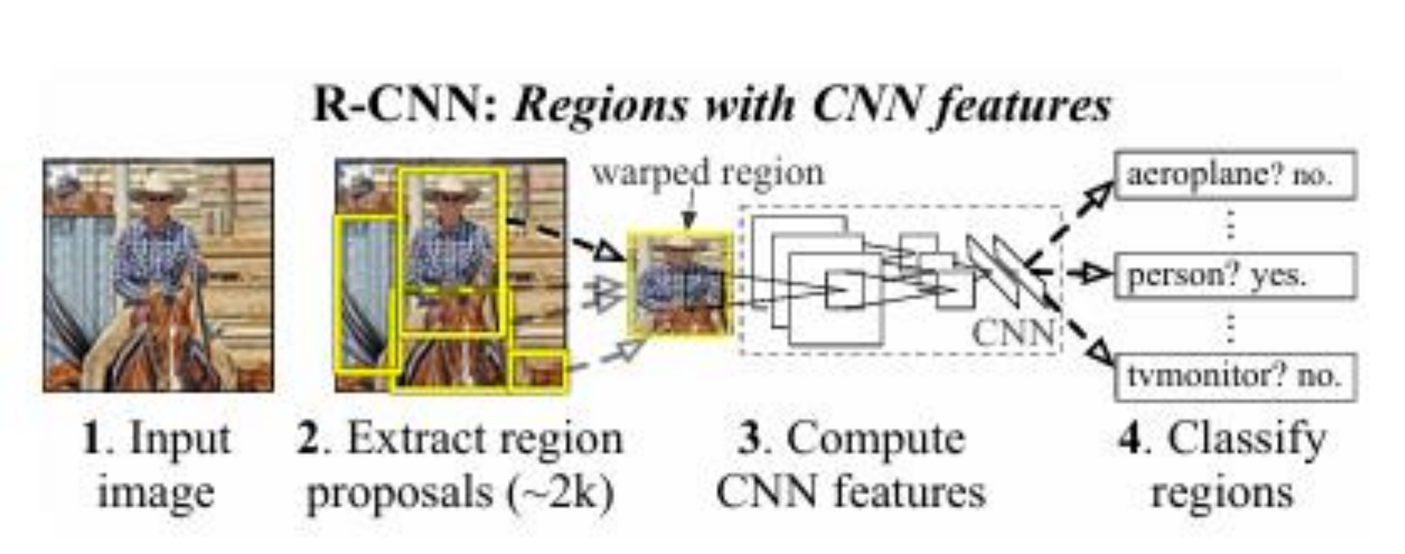
while $S \neq \emptyset$ **do**
    Get highest similarity $s(r_i, r_j) = \max(S)$
    Merge corresponding regions $r_t = r_i \cup r_j$
    Remove similarities regarding $r_i : S = S \setminus s(r_i, r_*)$
    Remove similarities regarding $r_j : S = S \setminus s(r_*, r_j)$
    Calculate similarity set S_t between r_t and its neighbours
    $S = S \cup S_t$
    $R = R \cup r_t$

Extract object location boxes L from all regions in R

Selective Search for Object Detection | R-CNN - GeeksforGeeks

II. R-CNN Series

Region CNN – R-CNN

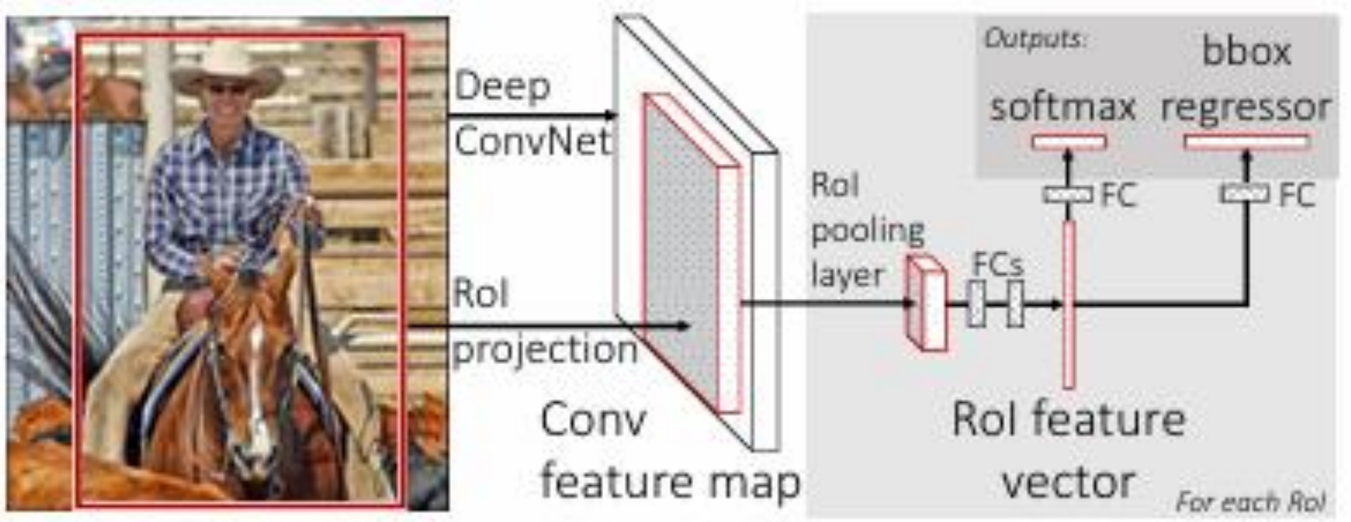


Hệ thống của chúng ta (1) nhận một hình ảnh đầu vào, (2) trích xuất khoảng 2000 đề xuất vùng từ dưới lên, (3) tính toán các đặc trưng cho mỗi đề xuất bằng cách sử dụng một mạng nơ-ron tích chập lớn (CNN), và sau đó (4) phân loại mỗi vùng bằng cách sử dụng SVM tuyến tính cụ thể cho lớp.

Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation

II. R-CNN Series

Fast R-CNN



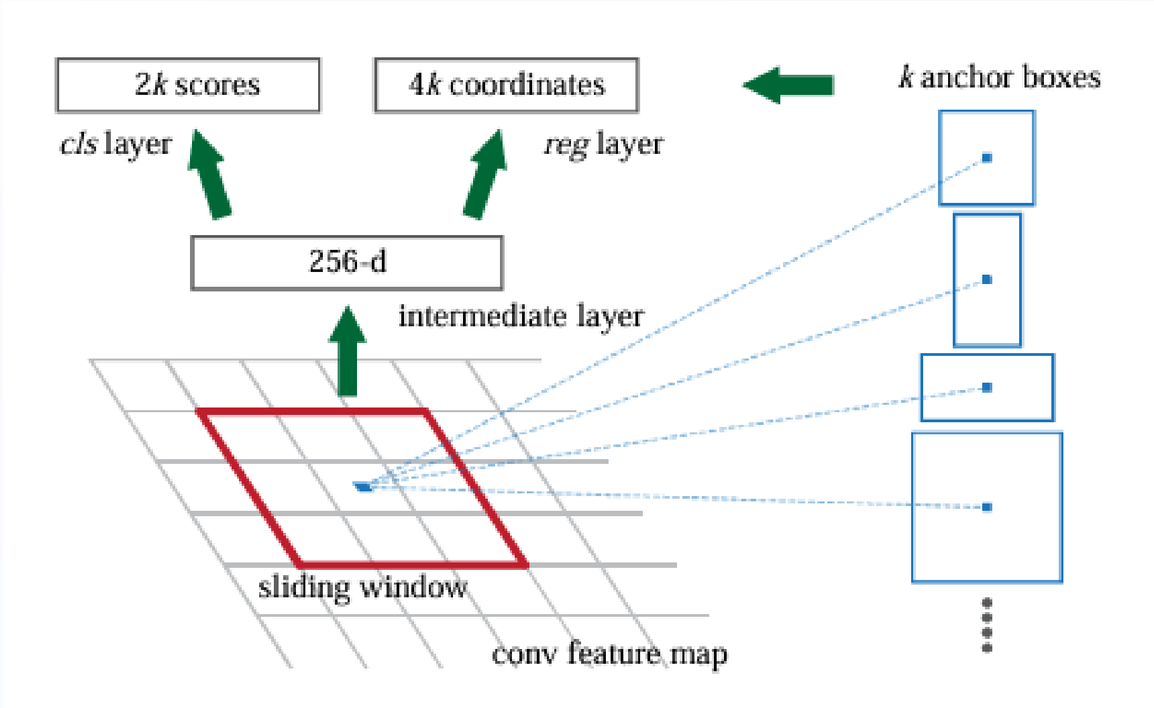
Một hình ảnh đầu vào và nhiều khu vực quan tâm (RoIs) được đưa vào một CNN hoàn toàn. Mỗi RoI được lấy mẫu thành một feature map có kích thước cố định và sau đó được ánh xạ tới một vector đặc trưng bởi các lớp liên kết hoàn toàn (FCs). Mạng có hai vector đầu ra cho mỗi RoI: xác suất softmax và độ dịch hồi quy hộp giới hạn theo lớp. Kiến trúc này được đào tạo theo cách end-to-end với một hàm mất mát đa nhiệm.

Fast R-CNN

II. R-CNN Series

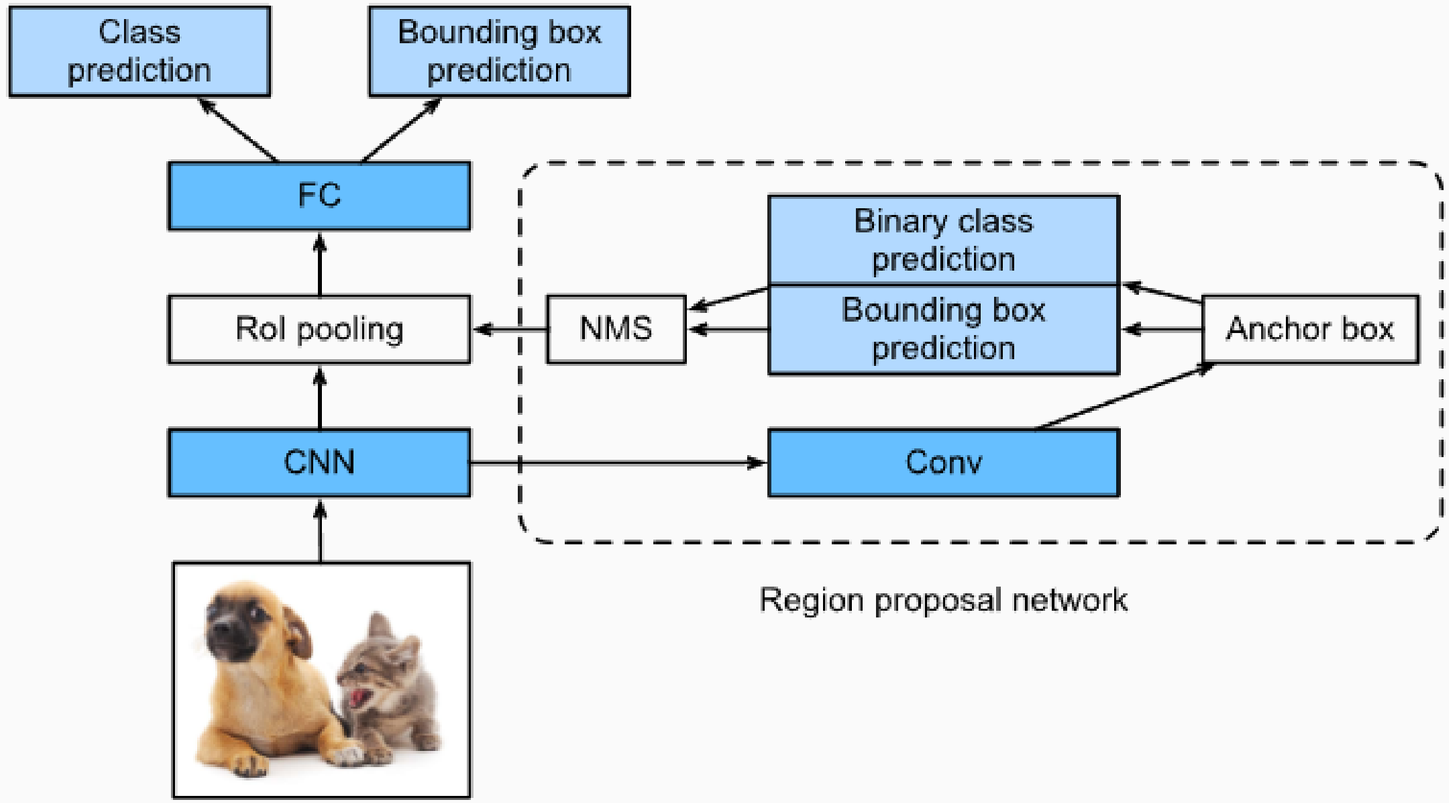
Faster R-CNN

Region Proposal Network



II. R-CNN Series

Faster R-CNN

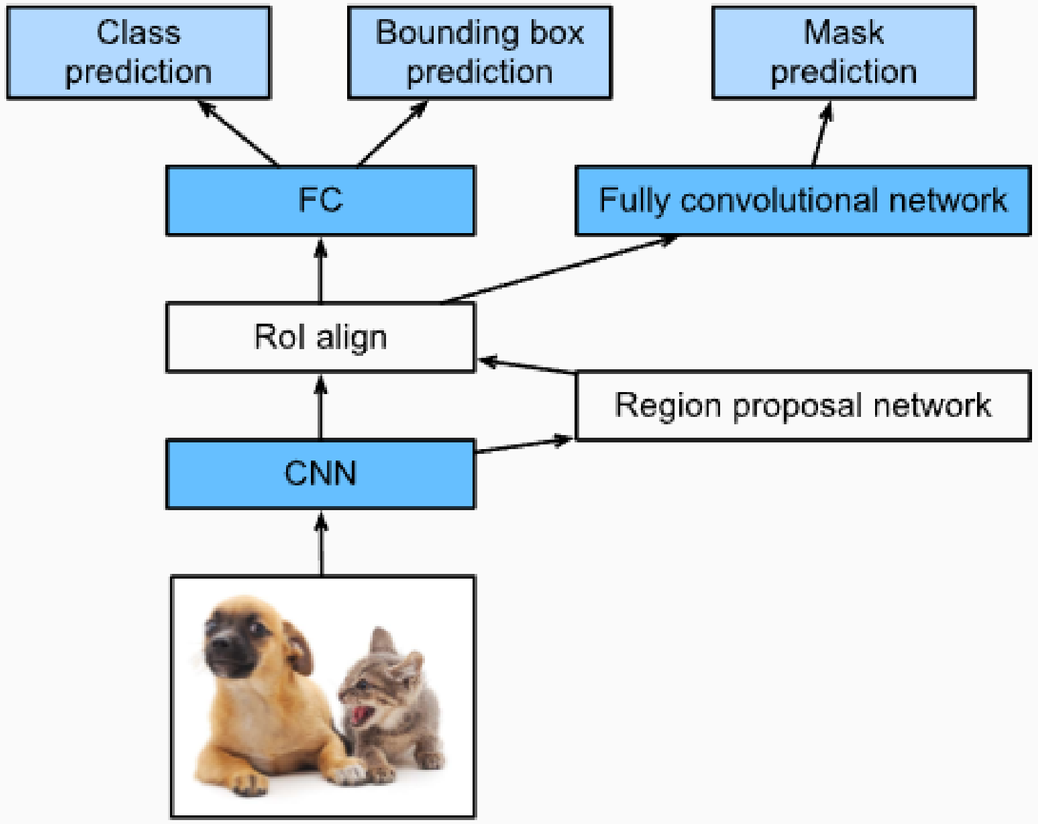


[1506.01497] Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

14.8. Region-based CNNs (R-CNNs) — Dive into Deep Learning 1.0.3 documentation

II. R-CNN Series

Mask R-CNN



Mask R-CNN

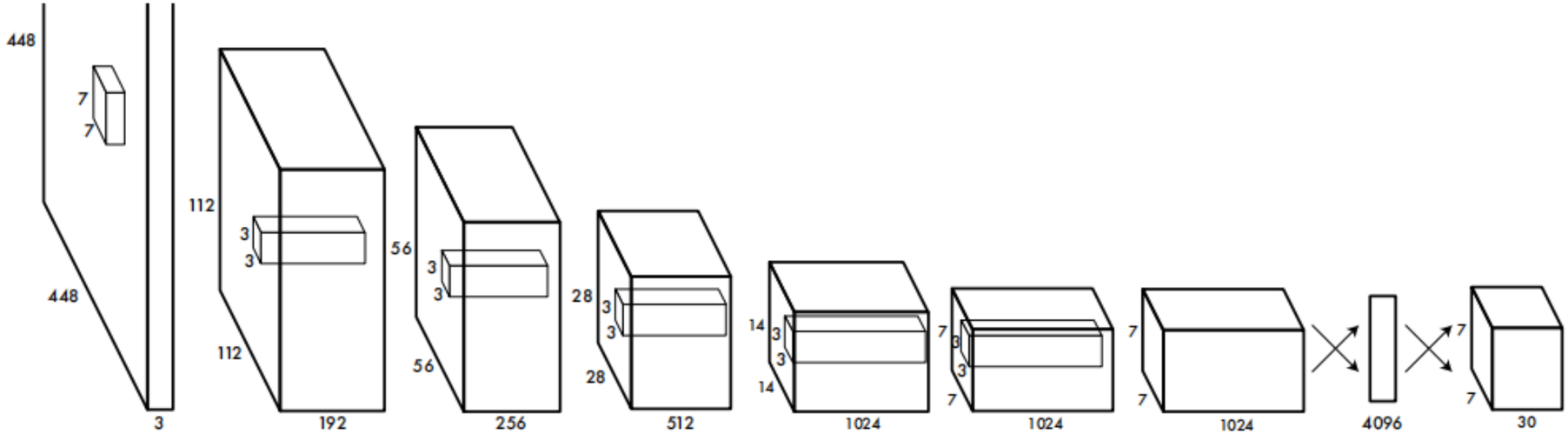
14.8. Region-based CNNs (R-CNNs) — Dive into Deep Learning 1.0.3 documentation

III. YOLO Series

III. YOLO Series

YOLOv1

The Architecture



III. YOLO Series

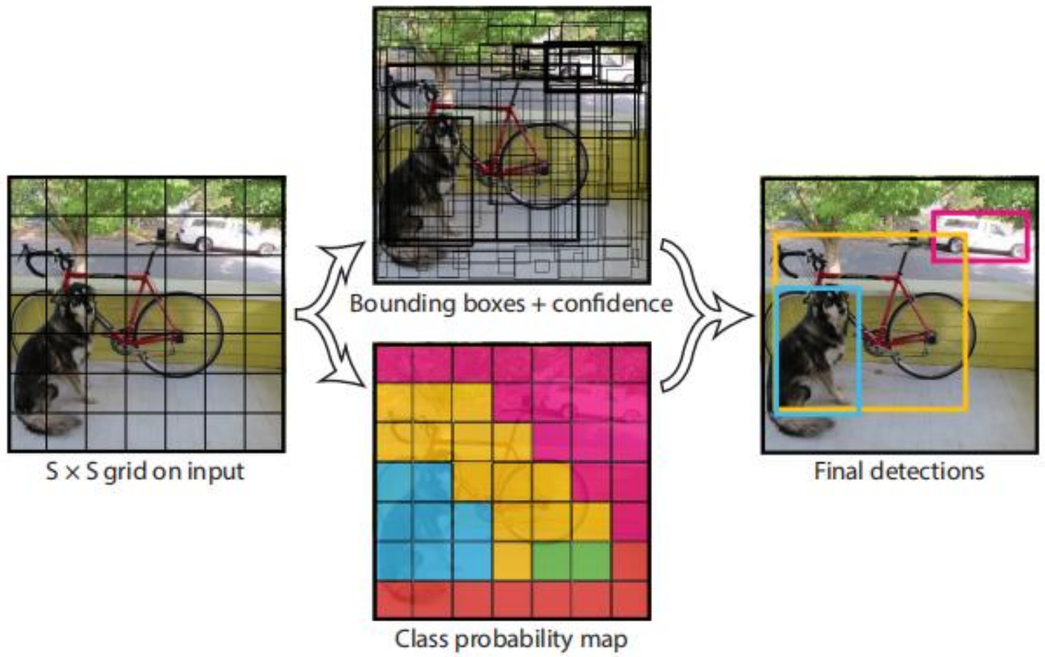
YOLOv1

The Architecture

- Input ảnh $448 \times 448 \times 3$ ($224 \times 224 \times 3$ khi pretrain trên ImageNet).
- Chia ảnh thành $S \times S$ grid (YOLOv1: $S = 7$).
- Mỗi cell dự đoán:
  - B bounding boxes (YOLOv1: $B = 2$).
  - Confidence cho mỗi box.
  - Một bộ class probabilities (C classes, VOC: $C = 20$).
- Output cuối: tensor $S \times S \times (B \times 5 + C) \rightarrow 7 \times 7 \times 30$.
- Conv layers trích đặc trưng → Fully Connected layers hồi quy ra (x, y, w, h, c) và $class\ probs$.

III. YOLO Series

YOLOv1



- $(\boldsymbol{x}, \boldsymbol{y})$: tâm box (chuẩn hóa theo cell).
- $(\boldsymbol{w}, \boldsymbol{h})$: kích thước box (chuẩn hóa theo ảnh).
- **Confidence**:

$$\hat{C} = P(Object) \times IOU_{pred}^{truth}$$

- **Class probabilities**:

$$P(\, Class_i \mid Object \,)$$

- Mỗi object được detect bởi cell chứa tâm object.
- Trong B box predictors của cell đó, chọn box có IOU cao nhất với ground truth làm **responsible predictor**:
  - Predictor này chịu loss localization + confidence (object).
  - Các predictor còn lại học confidence = 0 (no-object loss).

III. YOLO Series

YOLOv1

Loss Function

$$L_{YOLOv1} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbf{1}_{ij}^{obj} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + \left(\sqrt{w_i} - \sqrt{\hat{w}_i}\right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i}\right)^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbf{1}_{ij}^{obj} \left(C_i - \hat{C}_i\right)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbf{1}_{ij}^{noobj} \left(C_i - \hat{C}_i\right)^2$$

$$+ \sum_{i=0}^{S^2} \mathbf{1}_{ij}^{obj} \sum_{c \in classes} \left(p_i(c) - \hat{p}_i(c)\right)^2$$

III. YOLO Series

YOLOv1

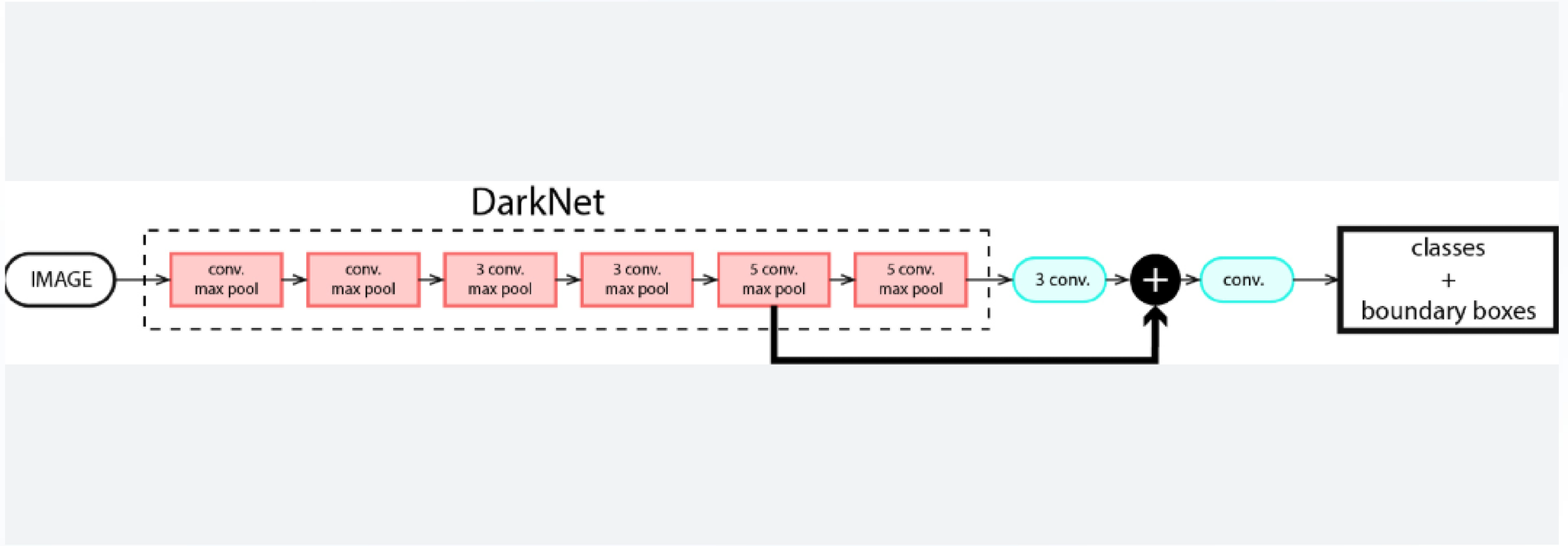
What are YOLO Models Used For?

- Xác định những kẻ xâm nhập trong nhà máy.
- Giám sát chuyển động của xe trên công trường.
- Hiểu các mô hình giao thông trên đường (tức là tìm thời gian mà một con đường được sử dụng nhiều nhất và ít nhất).
- Xác định khói từ các đám cháy trong tự nhiên.
- Giám sát để đảm bảo người lao động mặc trang bị bảo hộ cá nhân phù hợp trong một số môi trường nhất định (tức là khi sử dụng dụng cụ hoặc làm việc với các hóa chất thải ra khói độc hại).

What is YOLO? The Ultimate Guide [2025]

III. YOLO Series

YOLOv2

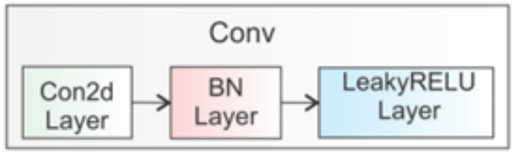


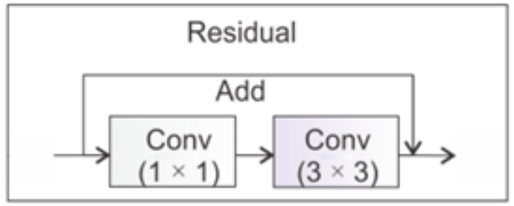
[What is YOLO? The Ultimate Guide [2025]]
[YOLO v2 - Object Detection - GeeksforGeeks]

III. YOLO Series

YOLOv3

Layer	Filters	size	Repeat		Output size
Image					416 × 416
Conv	32	3 × 3/1	1		416 × 416
Conv	64	3 × 3/2	1		208 × 208
Conv	32	1 × 1/1	Conv		208 × 208
Conv	64	3 × 3/1	Conv	×1	208 × 208
Residual			Residual		208 × 208
Conv	128	3 × 3/2	1		104 × 104
Conv	64	1 × 1/1	Conv		104 × 104
Conv	128	3 × 3/1	Conv	×2	104 × 104
Residual			Residual		104 × 104
Conv	256	3 × 3/2	1		52 × 52
Conv	128	1 × 1/1	Conv		52 × 52
Conv	256	3 × 3/1	Conv	×8	52 × 52
Residual			Residual		52 × 52
Conv	512	3 × 3/2	1		26 × 26
Conv	256	1 × 1/1	Conv		26 × 26
Conv	512	3 × 3/1	Conv	×8	26 × 26
Residual			Residual		26 × 26
Conv	1024	3 × 3/2	1		13 × 13
Conv	512	1 × 1/1	Conv		13 × 13
Conv	1024	3 × 3/1	Conv	×4	13 × 13
Residual			Residual		13 × 13

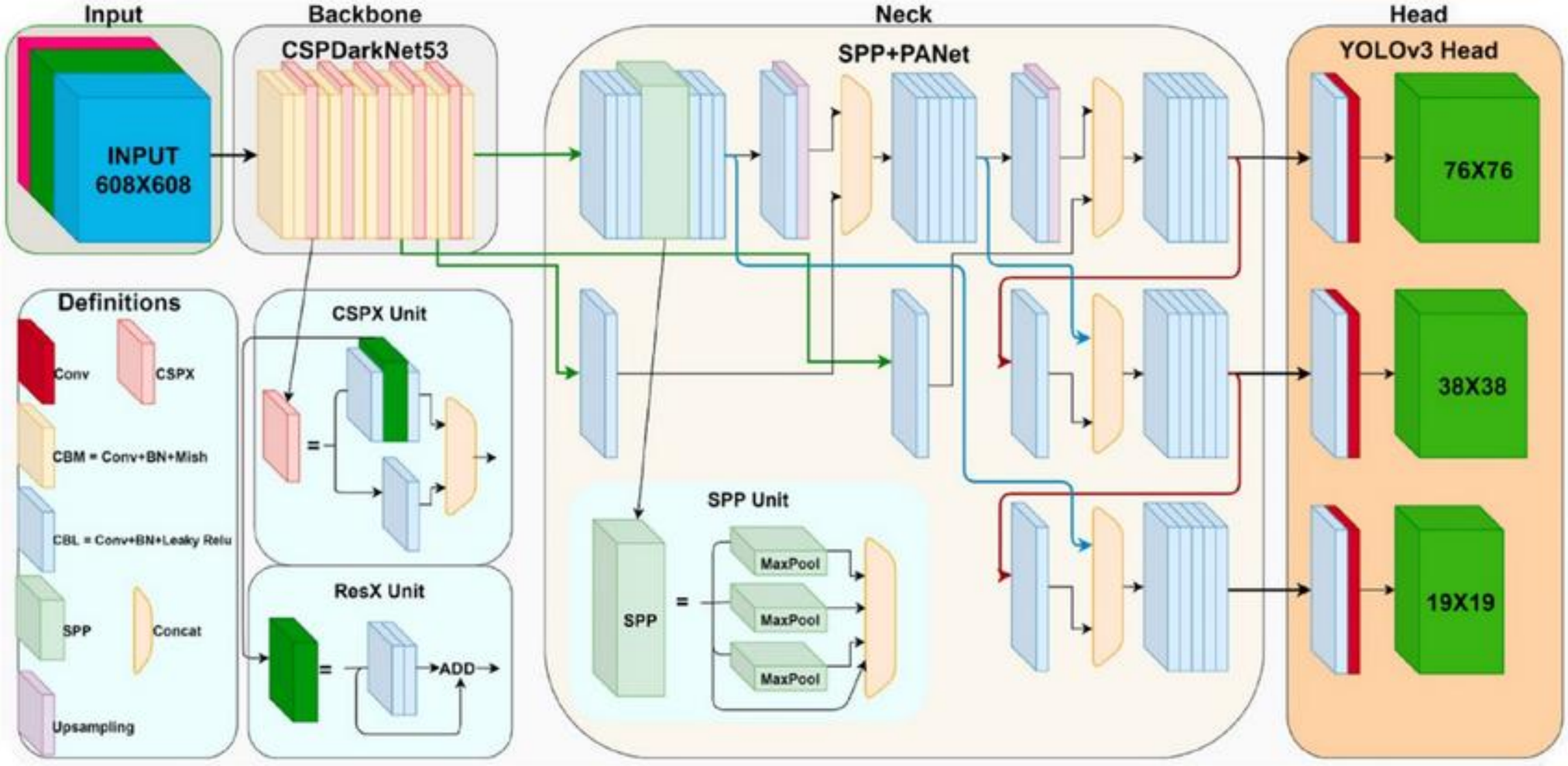




A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS

III. YOLO Series

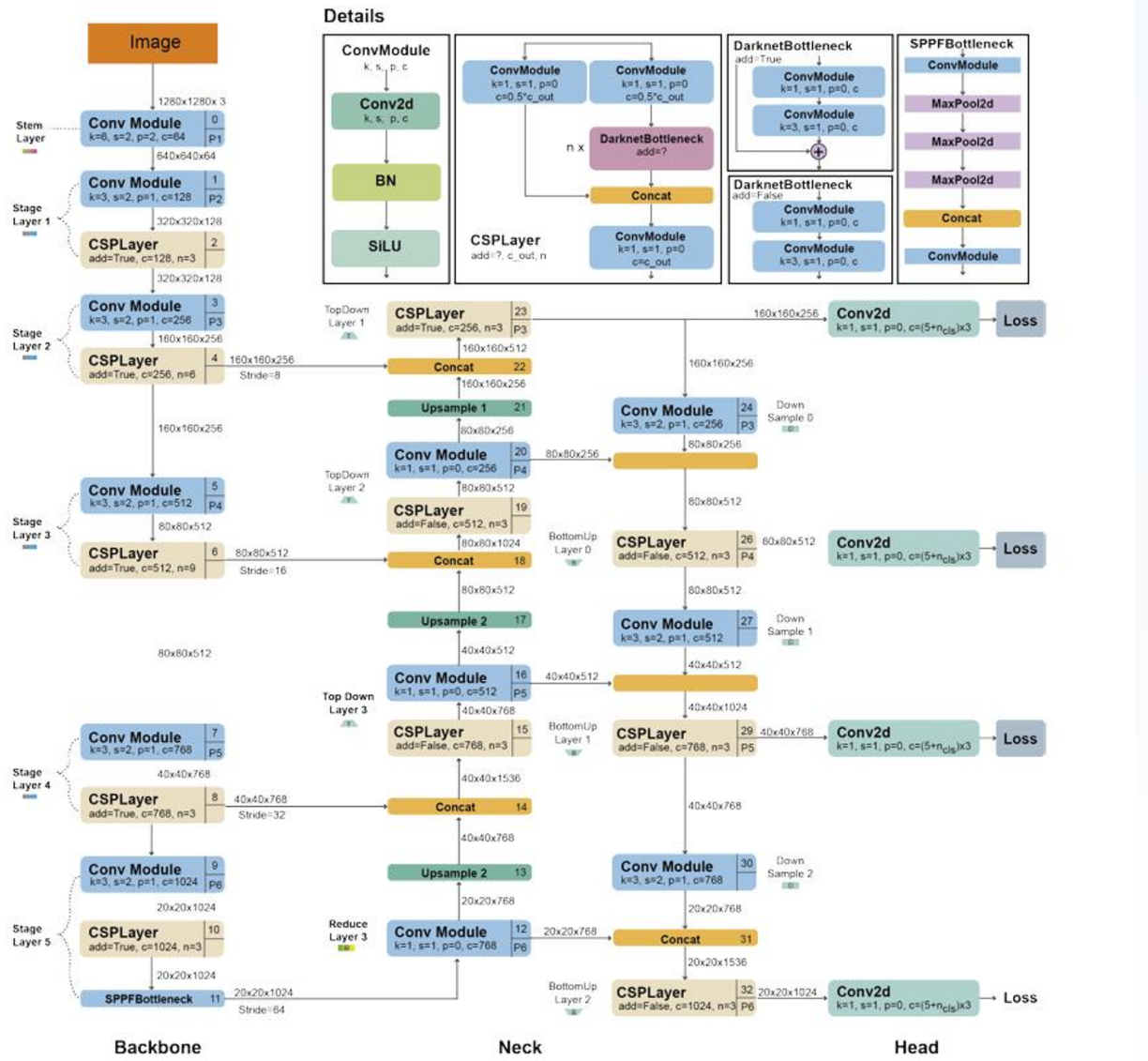
YOLOv4



A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS

III. YOLO Series

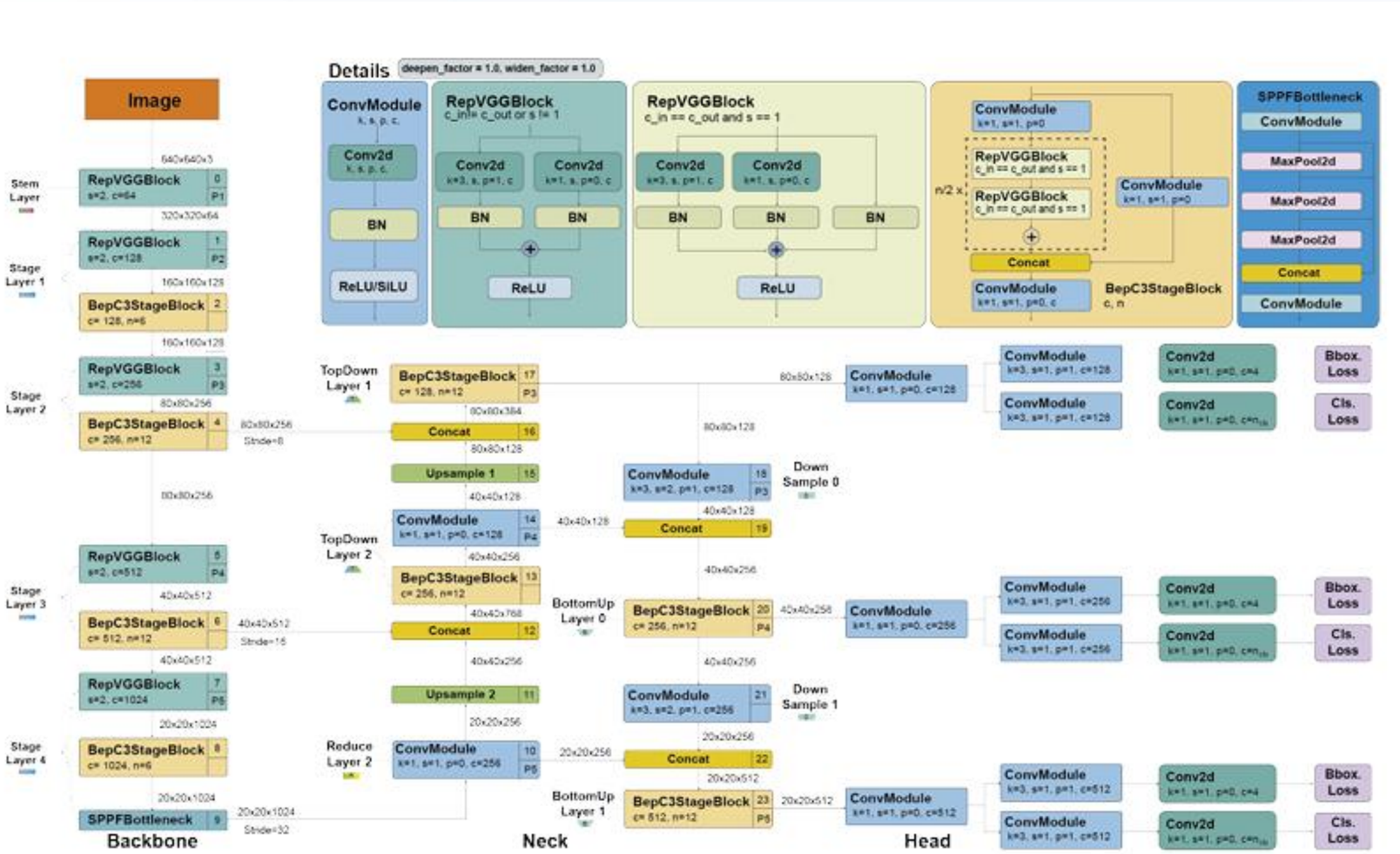
YOLOv5



[A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS]

III. YOLO Series

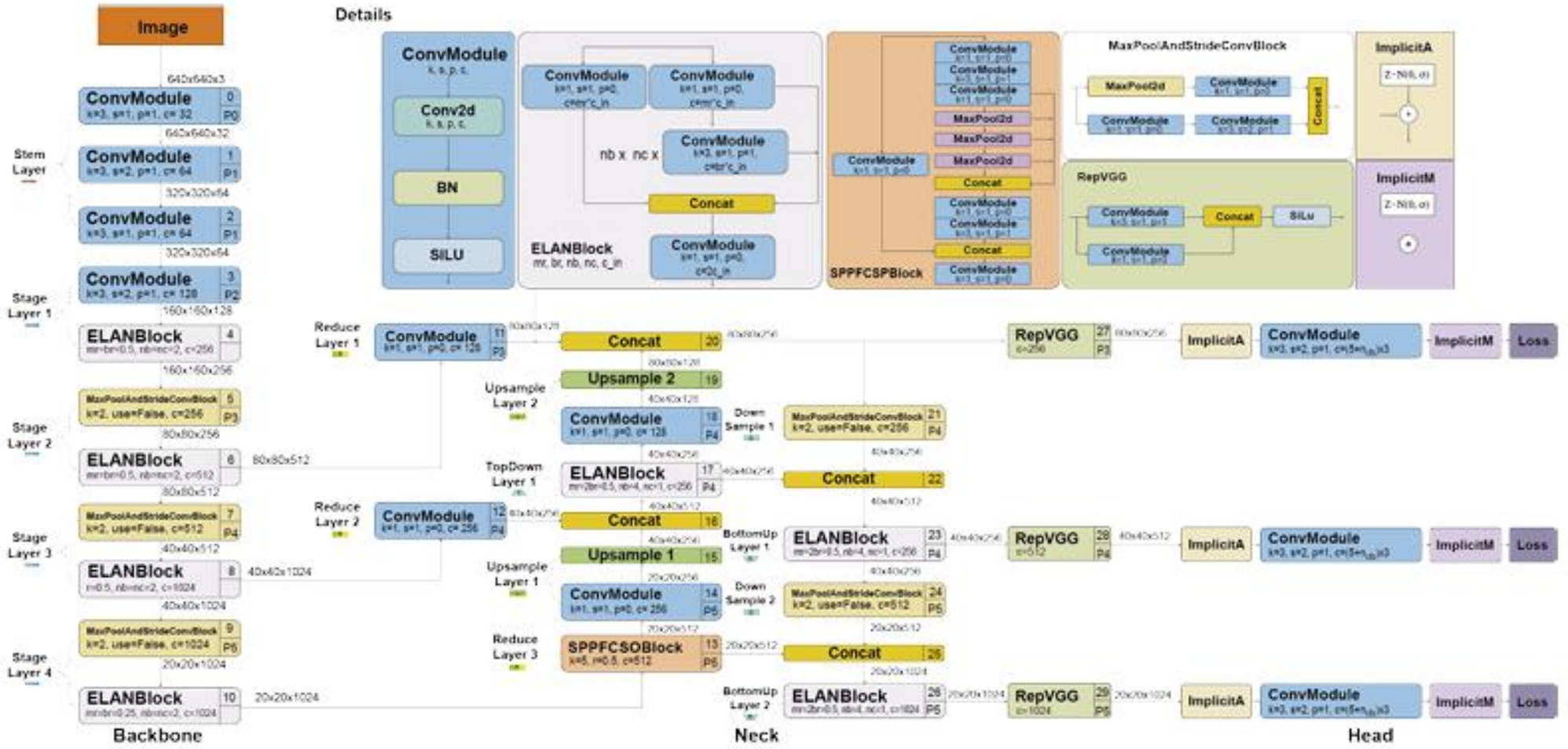
YOLOv6



[A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS]

III. YOLO Series

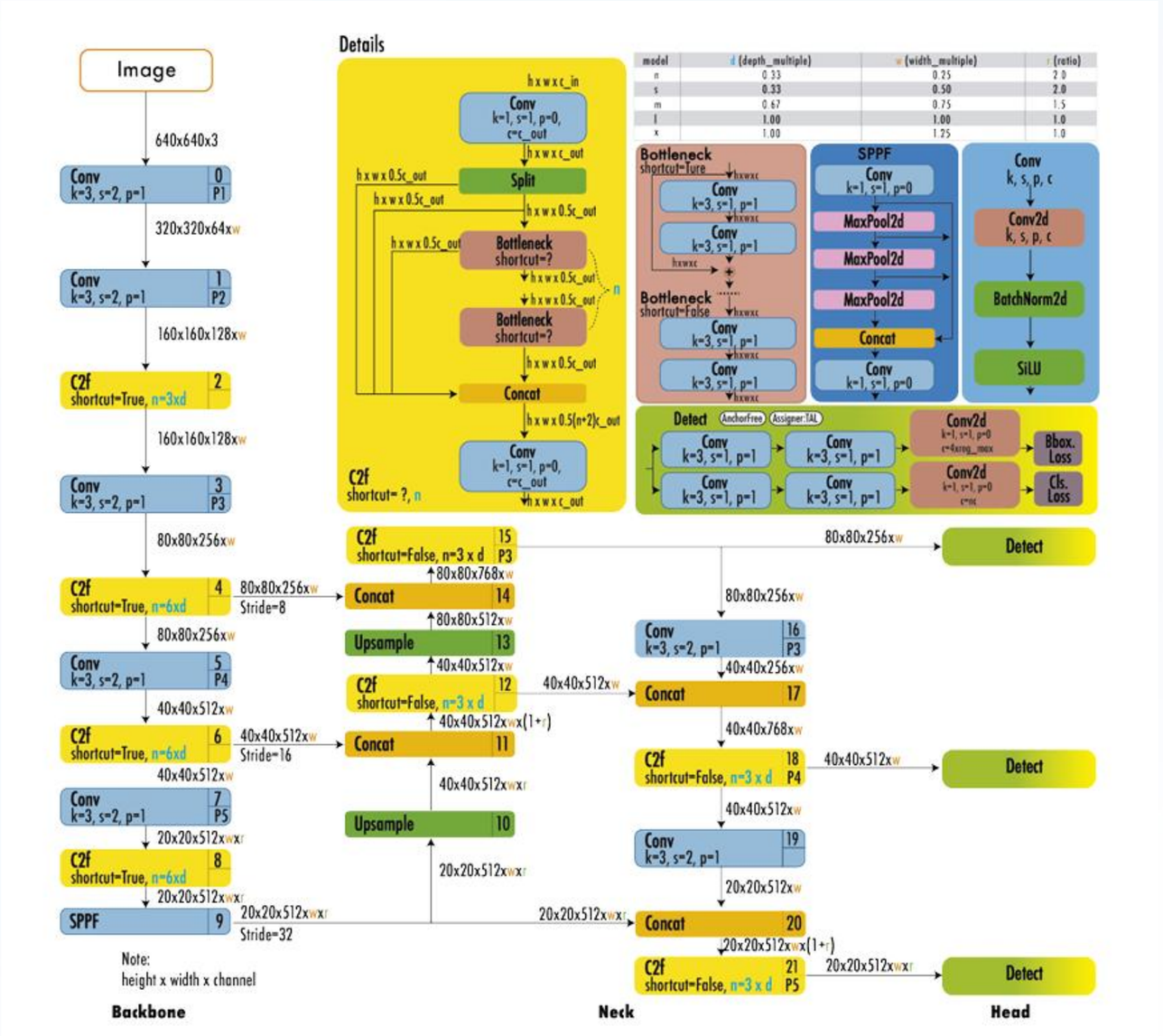
YOLOv7



[A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS]

III. YOLO Series

YOLOv8

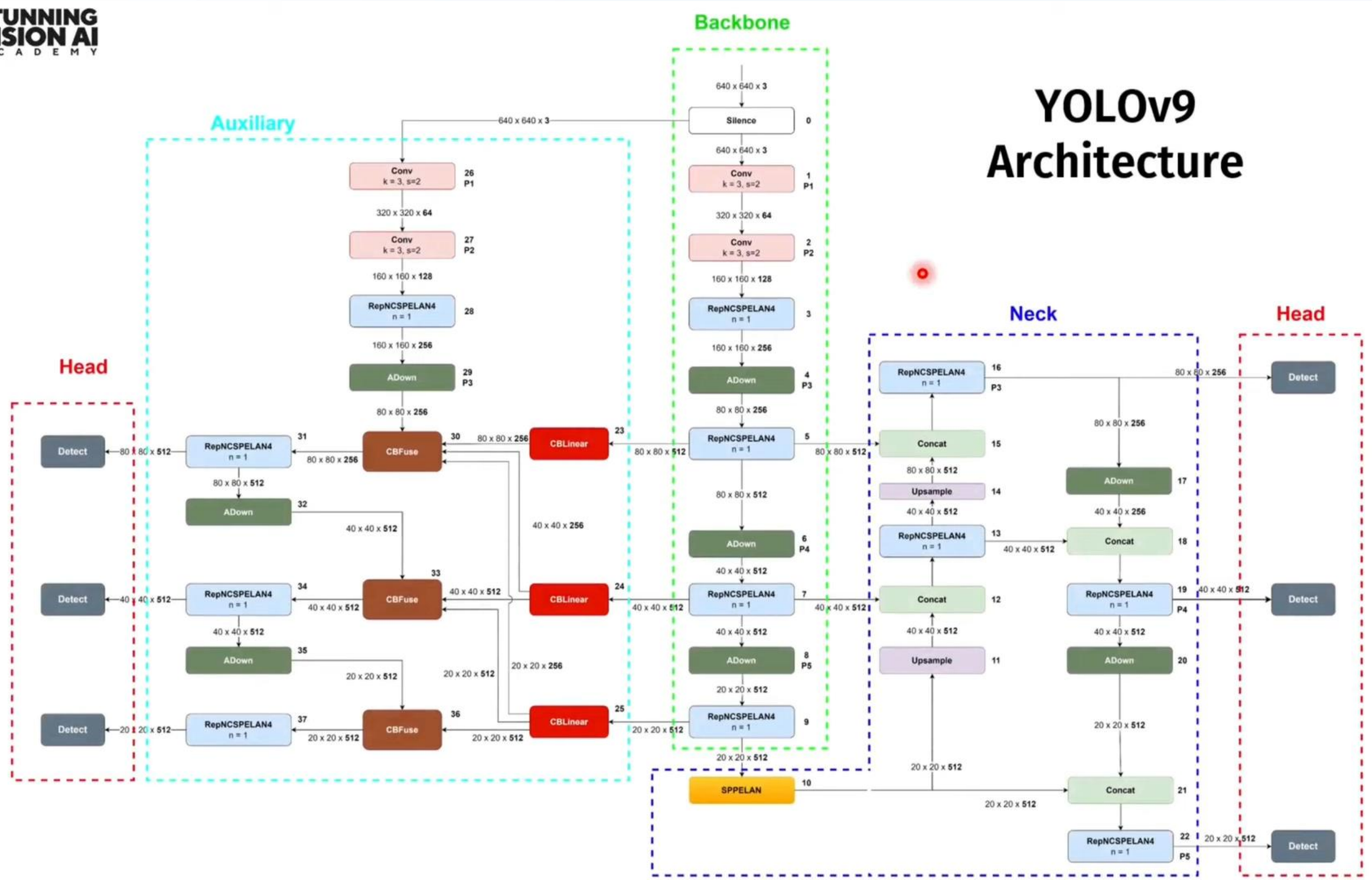


[A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS]

[2501.13400] YOLOv8 to YOLO11: A Comprehensive Architecture In-depth Comparative Review

III. YOLO Series

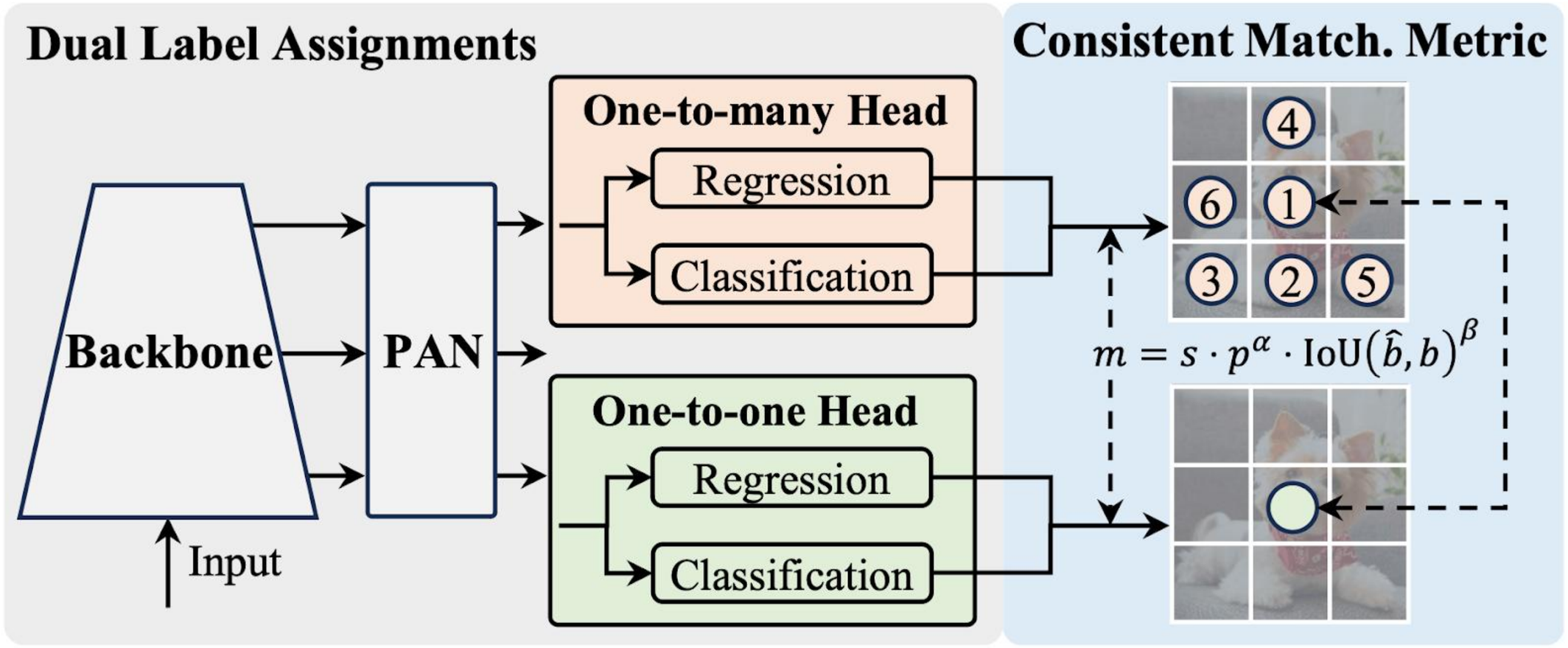
YOLOv9



[YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information]

III. YOLO Series

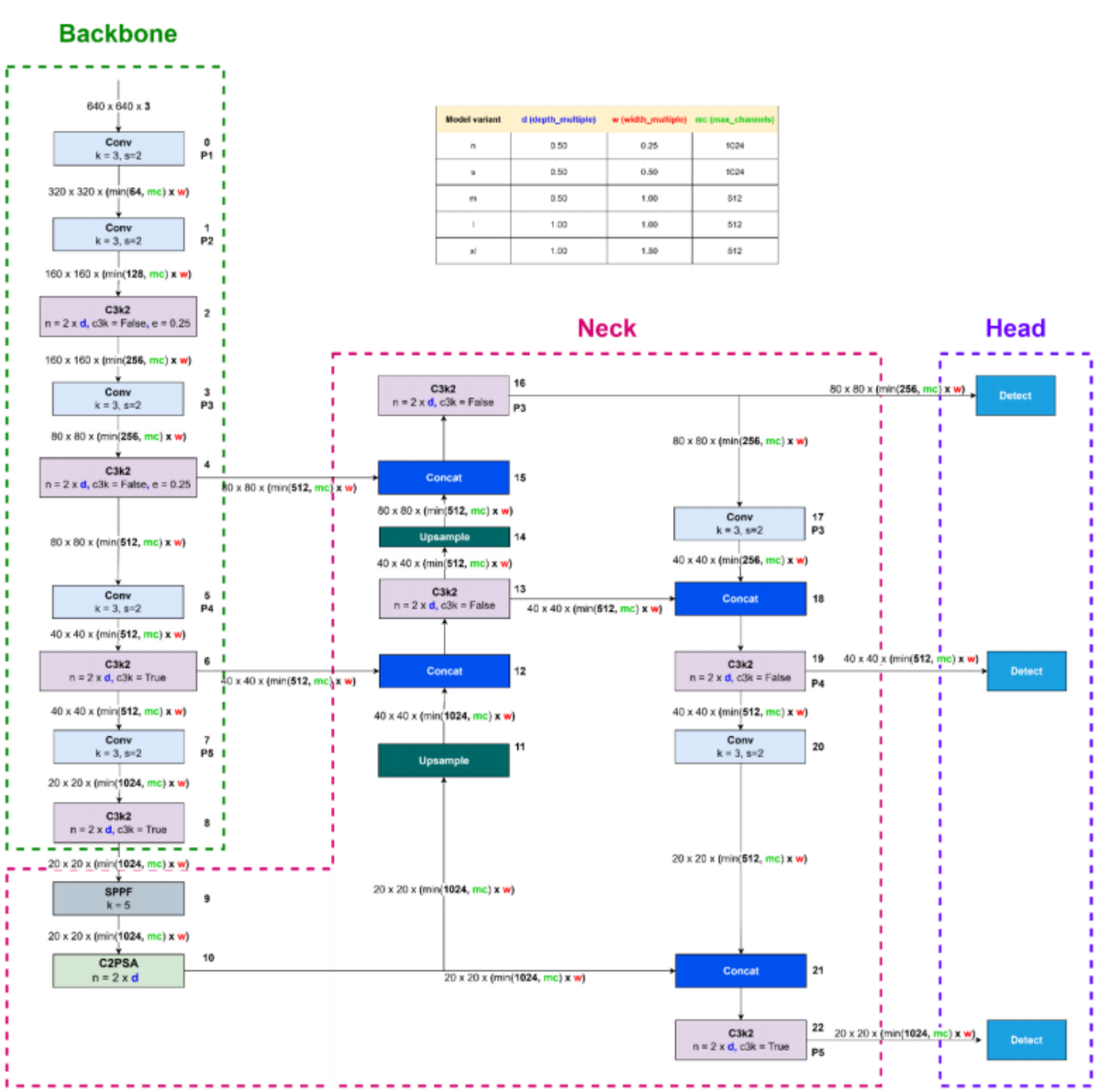
YOLOv10



[YOLOv10: Paper Explanation and Inference Results]
[What is YOLOv10? An Architecture Deep Dive.]

III. YOLO Series

YOLOv11

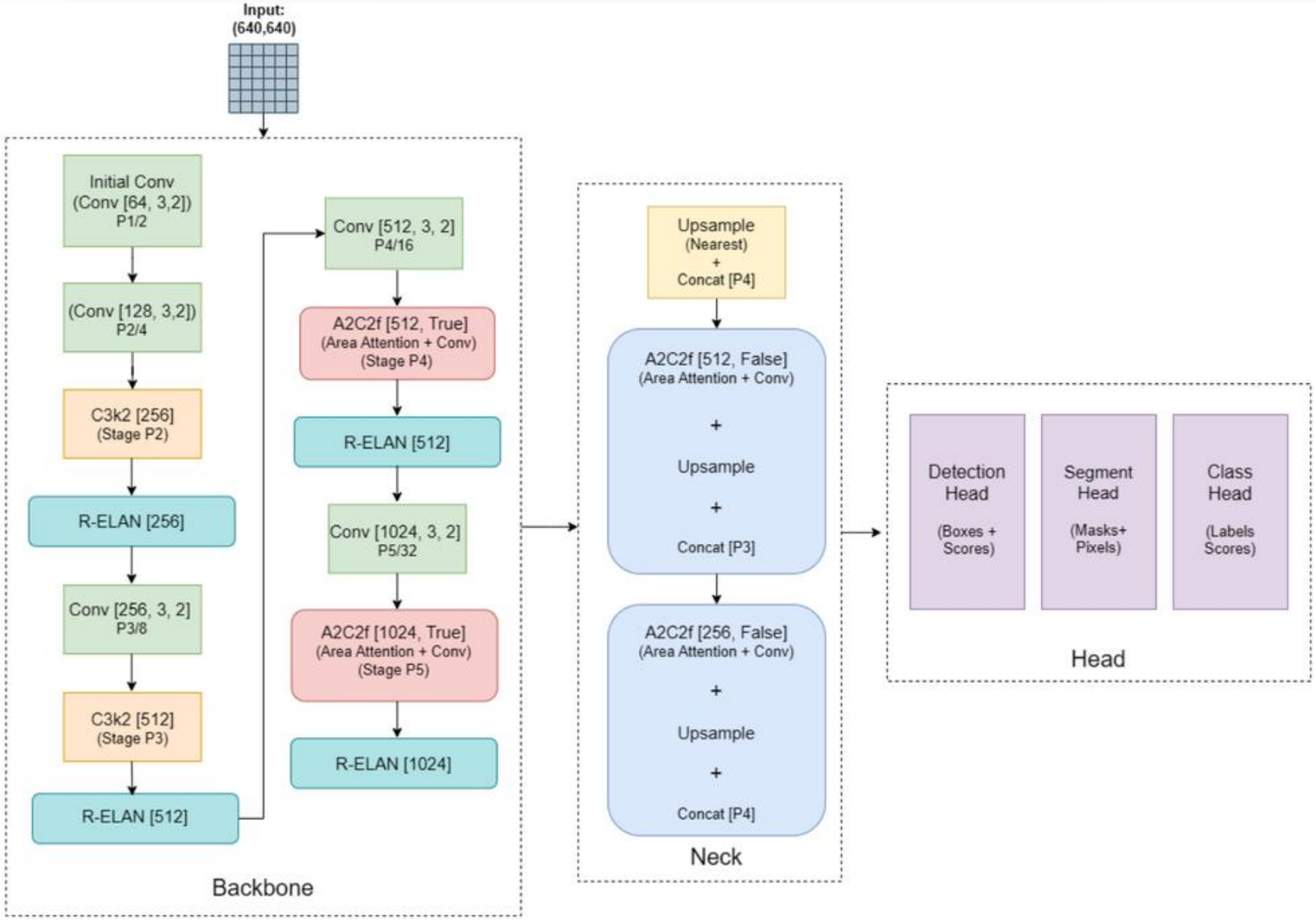


Model variant	d (depth_multiple)	w (width_multiplo)	mc (max_channels)
n	0.50	0.25	1024
s	0.50	0.50	1024
m	0.50	1.00	512
l	1.00	1.00	512
xl	1.00	1.50	512

[YOLOv11: An Overview of the Key Architectural Enhancements]

III. YOLO Series

YOLOv12

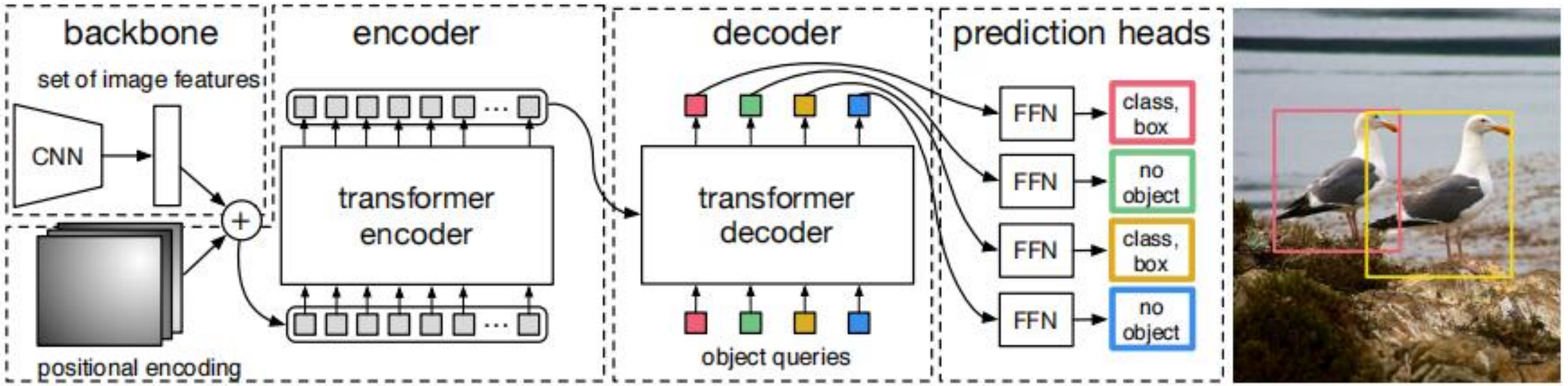


[YOLOv12: A Breakdown of the Key Architectural Features]
[2502.12524] YOLOv12: Attention-Centric Real-Time Object Detectors
YOLO Object Detection Explained: Real-Time Vision Tasks

IV. Detection Transformer - DETR

IV. Detection Transformer - DETR

DETR architecture

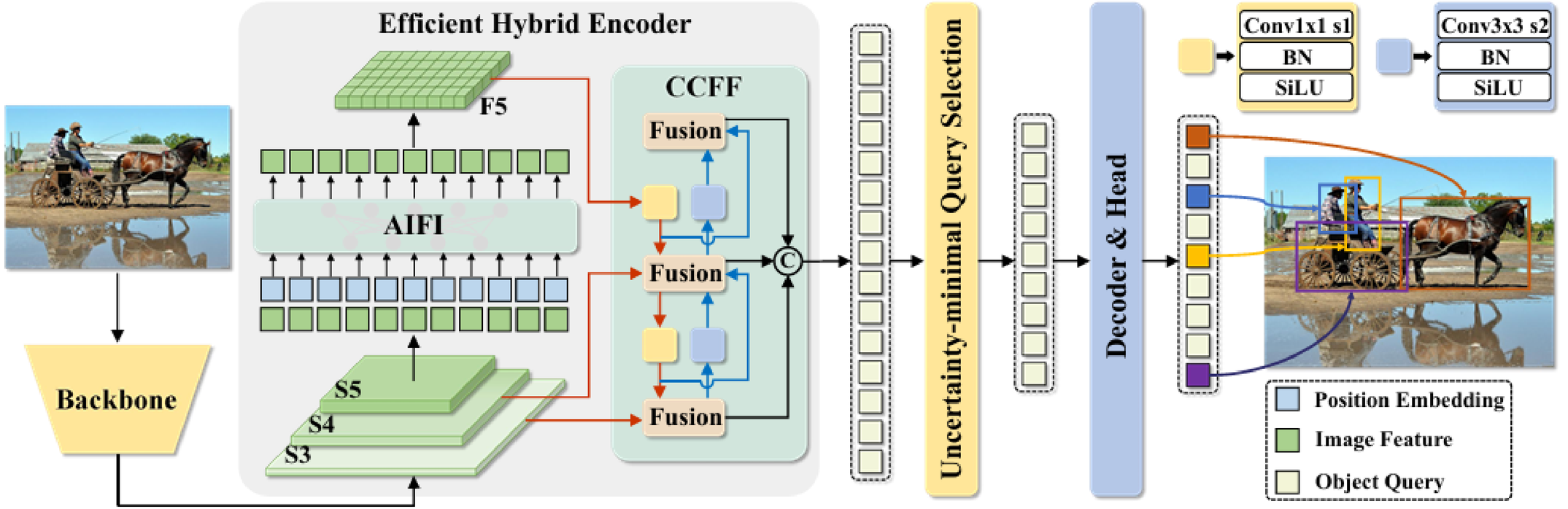


DETR sử dụng một **backbone CNN thông thường** để học biểu diễn 2D của ảnh đầu vào. Mô hình **làm phẳng** biểu diễn này và thêm **mã hoá vị trí (positional encoding)** trước khi đưa vào **transformer encoder**. Transformer decoder nhận một số lượng nhỏ vector vị trí học được (object queries) và đồng thời chú ý đến đầu ra của encoder. Mỗi vector đầu ra từ **decoder** được đưa qua một **mạng FFN** chung để dự đoán hoặc là một detection (nhãn + bounding box) hoặc là lớp “no object”.

DETRs Beat YOLOs on Real-time Object Detection

IV. Detection Transformer - DETR

RT-DETR architecture



DETRs Beat YOLOs on Real-time Object Detection

ML IoT
HCMUT EE MACHINE LEARNING & IOT LAB

THANK YOU

CONTACT US

403.1 H6, BKHCM Campus 2

mlandiotlab@gmail.com

mliotlab.github.io

facebook.com/hcmut.ml.iot.lab

youtube.com/@mliotlab