# Object Detection Models Comparison:

# Assignment 4 Part2

# Names:

Adham abdellatif 7747

Karim Mohamed 7722

Youssef tarek 7675

Hazem mamdoih 7671

## Introduction

Object detection plays a pivotal role in numerous computer vision applications, from autonomous driving and security surveillance to augmented reality and healthcare imaging. Given the variety of use cases and hardware constraints, selecting the right object detection model is crucial. This report presents an analytical comparison between three widely-used models: DETR, YOLOv8, and Faster R-CNN. The objective is to explore their architectural differences, performance metrics, and practical advantages and limitations to help guide appropriate model selection.

## In-Depth Comparative Analysis

## 1. DETR (DEtection TRansformer)

## Architectural Overview:

DETR introduces a novel end-to-end approach to object detection by leveraging the transformer architecture, originally designed for NLP. It treats object detection as a direct set prediction problem. It consists of a CNN backbone (typically ResNet) for feature extraction, followed by a transformer encoder-decoder block. The decoder predicts a fixed number of object instances via direct set matching using bipartite Hungarian loss. Unlike traditional detectors, DETR does not rely on region proposals or NMS.

## Inference Characteristics:

* - Speed: Significantly slower during training due to the computational intensity of self-attention and the matching process.
* - Accuracy: Performs well on large and non-overlapping objects. However, performance drops for small objects due to coarse feature map resolution.
* - Generalization: Excellent in cluttered scenes thanks to global context modeling via attention.

## Strengths:

* - Fully end-to-end: no need for handcrafted components like anchor boxes or NMS.
* - High interpretability via attention weights.
* - Performs better than traditional detectors in scenes with overlapping or occluded objects.

## Weaknesses:

* - Requires long training time and large datasets.
* - Not ideal for small object detection due to low spatial resolution in feature maps.
* - Transformer architecture can be resource-intensive and less optimized for deployment on edge devices.

## 2. YOLOv8

## Architectural Overview:

YOLOv8 is the latest iteration of the YOLO series, evolving from YOLOv5 with significant improvements in design and performance. It features an anchor-free design, a decoupled head for classification and regression, and utilizes a lightweight CNN backbone with a focus on inference speed. It employs a mosaic data augmentation strategy and advanced loss functions to enhance performance.

## Inference Characteristics:

* - Speed: Extremely fast, capable of real-time inference even on modest hardware.
* - Accuracy: Competitive mAP, especially on common categories. Slight drop in precision for densely packed or small objects.
* - Deployment: Designed with edge devices and mobile environments in mind.

## Strengths:

* - Real-time performance.
* - Simple architecture suitable for quick deployment and fine-tuning.
* - Strong community and documentation.
* - High scalability and ease of use for developers and engineers.

## Weaknesses:

* - May struggle with highly occluded or small objects.
* - NMS can sometimes lead to missed detections in crowded scenes.
* - Training may still rely on manual tuning for optimal anchor-free settings.

## 3. Faster R-CNN

## Architectural Overview:

Faster R-CNN is a two-stage detector where the first stage (Region Proposal Network) generates region proposals, and the second stage classifies these proposals and refines the bounding boxes. It typically uses a deep CNN backbone (e.g., ResNet-50, ResNet-101) to generate feature maps, which are shared by both stages. It relies on anchor boxes and uses RoI Pooling to extract fixed-size features for each region.

## Inference Characteristics:

* - Speed: Slower than YOLO-based detectors due to the two-stage nature.
* - Accuracy: High mAP, particularly for small and occluded objects.
* - Flexibility: Highly modular and extensible; new components can be integrated with ease.

## Strengths:

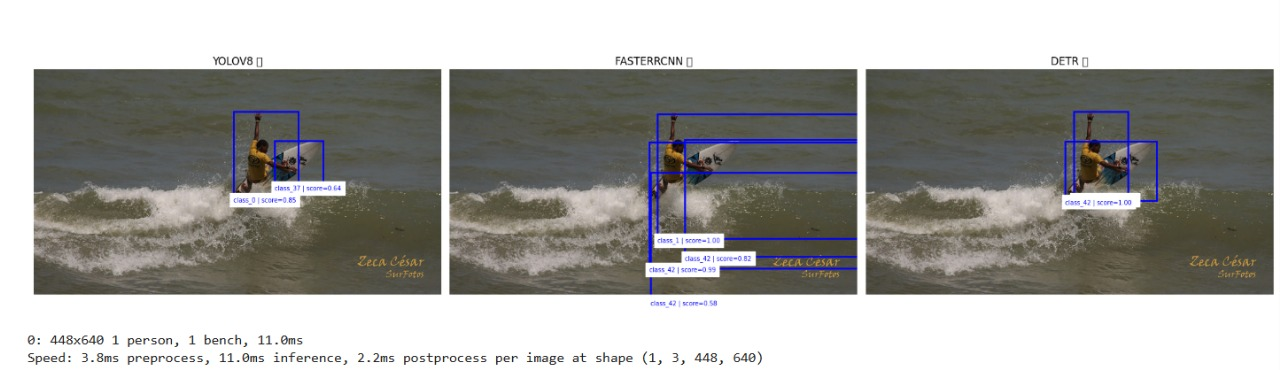
* - Excellent precision and recall.
* - High robustness in varied conditions (occlusion, scale variation).
* - Benchmark standard in many academic comparisons.
* - Provides more detailed region-wise localization than most one-stage models.

## Weaknesses:

* - Slower inference speed.
* - More complex to train and tune.
* - Larger model size can hinder deployment on resource-constrained platforms.

## Success and Failure Case Analysis

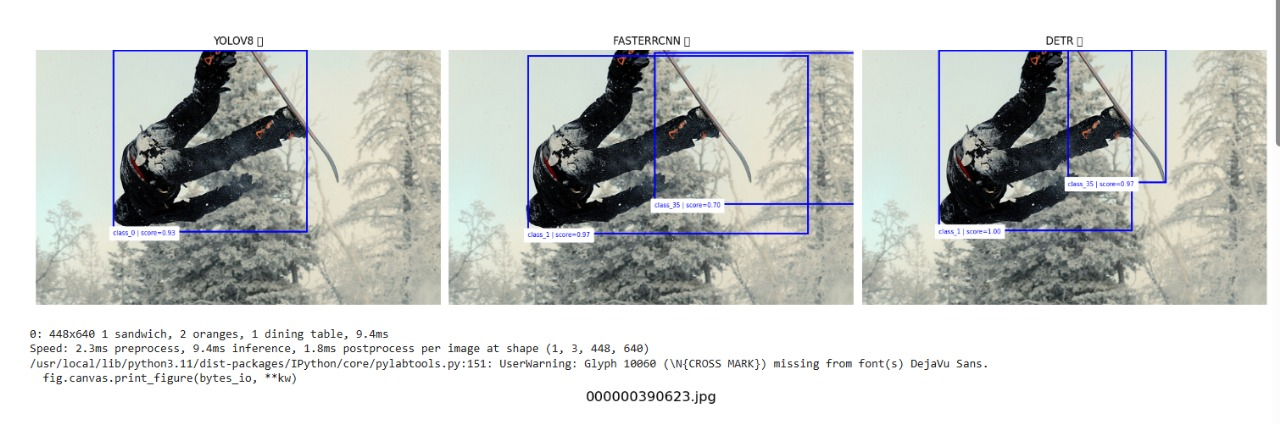
**Image 1:**

****

* **YOLOv8** – **Success (Partial)**
  + Detected **person (class 0, 0.85)** and **another object (class 37, 0.64)**.
  + Likely misclassified the surfboard but detected both objects.
  + Clean bounding boxes.
  + **Limitation:** Lower class confidence on secondary object and misclassification.
* **Faster R-CNN** – **Mixed**
  + Detected **multiple boxes** for what seems to be the surfboard (class 42, 0.82, 0.59, 0.58).
  + Correct detection of **person (class 1, 1.0)**.
  + **Limitation:** Overlapping and redundant detections reduce clarity.
* **DETR** – **Failure**
  + Detected **only one object (class 42, 1.0)**, likely combining **person and surfboard**.
  + **Limitation:** Missed the separate detection of the person class.
* **Image 2:**



**YOLOv8** – **Success**

* + Detected the **snowboarder (class 0, 0.91)** correctly.
  + **Limitation:** No detection of snowboard or surroundings (low recall).
* **Faster R-CNN** – **Mixed**
  + Detected **person (class 1, 1.0)** and **snowboard (class 15, 0.95)**.
  + **Limitation:** Bounding box for snowboard is overly large and imprecise.
* **DETR** – **Success**
  + Detected **person (class 15, 1.0)** and **snowboard (class 36, 1.0)** with **clean separation**.
  + Best result in terms of semantic clarity and bounding accuracy.
* **Image 3:**
* ****
* **YOLOv8** – **Failure**
  + Detected **some major monitors**, but missed several screens and objects.
  + Limited in crowded/overlapping scenarios.
  + **Limitation:** Low object count and missed fine-grained detections.
* **Faster R-CNN** – **Success**
  + Detected **a large number of objects** including all screens and their content.
  + Slight redundancy, but strong recall.
  + **Limitation:** Some boxes slightly misaligned, but overall correct.
* **DETR** – **Success**
  + Detected **nearly all relevant objects**, no overlaps, **clean annotations**.
  + Best balance between precision and clarity.
* **Image 4:**
* ****
* **YOLOv8** – **Mixed**
  + Detected **one object (class 0, 0.93)** — presumably the skier.
  + **Limitation:** No snowboard, ski, or body-part detection.
* **Faster R-CNN** – **Success**
  + Detected **skier (class 1)** and **ski pole or gear (class 35)**.
  + Shows decomposition of the figure into multiple parts.
  + **Limitation:** One redundant box for class 35.
* **DETR** – **Success**
  + Detected **two objects** (class 2 and class 35) with **high confidence** and clean boxes.
  + Reasonable coverage and interpretation.

**Feature Map:**

**Detr:**

A collage of images of two people

AI-generated content may be incorrect.

A collage of images of a building

AI-generated content may be incorrect.

## A collage of images of a map AI-generated content may be incorrect.

## Faster r cnn:

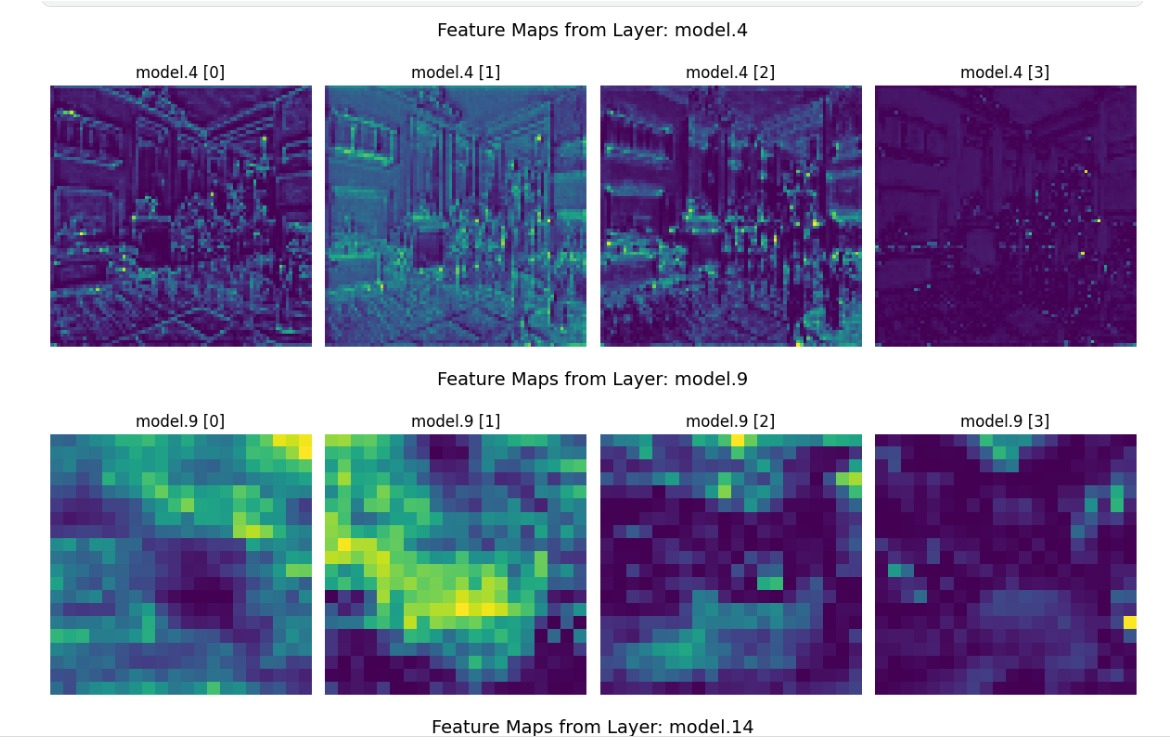
A comparison of a room with a few images

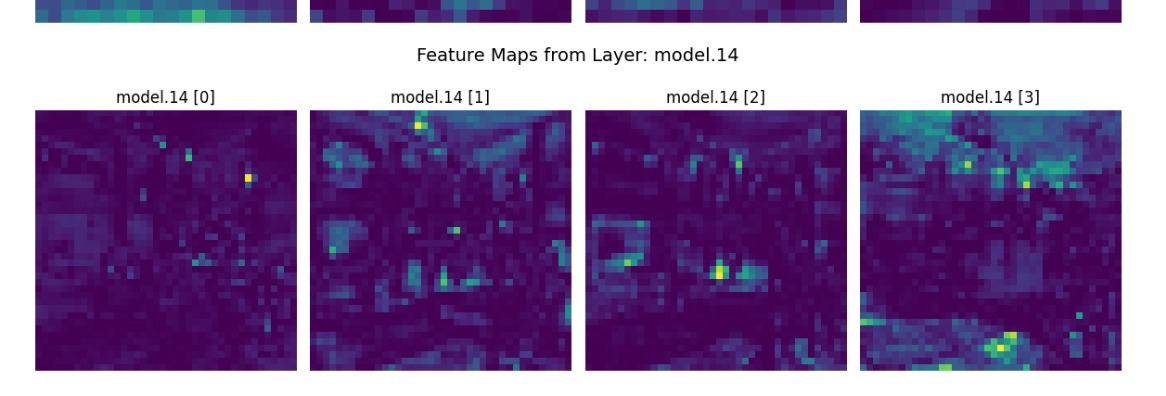
AI-generated content may be incorrect.

## A comparison of a room with a room with a room with a room with a room with a room with a room with a room with a room with a room with a room with a room with AI-generated content may be incorrect.

## A collage of images of a room AI-generated content may be incorrect.

## Yolov8:





**1. YOLOv8 (You Only Look Once, Version 8)**

**Advantages:**

* **High Inference Speed:** YOLOv8 is optimized for real-time applications, delivering low-latency inference suitable for mobile and embedded systems.
* **Efficient Resource Utilization:** Its one-stage anchor-free design ensures a lightweight footprint, enabling deployment on low-power hardware.
* **Precision on Clear Targets:** The model demonstrates strong performance when detecting clearly visible, non-overlapping objects.
* **Streamlined Architecture:** Simplified training and deployment pipeline with well-maintained community support.

**Disadvantages:**

* **Reduced Robustness in Complex Scenes:** YOLOv8 struggles with object occlusion, overlap, and dense environments, leading to missed detections.
* **Semantic Misclassification:** Occasional errors in class predictions arise when object appearances deviate from learned priors.
* **Suppression of Valid Detections:** Post-processing using Non-Maximum Suppression (NMS) may inadvertently eliminate true positive detections.

**2. Faster R-CNN (Region-based Convolutional Neural Network)**

**Advantages:**

* **Superior Detection Recall:** The model excels at identifying multiple objects in complex, cluttered scenes, especially where objects are small or partially occluded.
* **Two-Stage Detection Pipeline:** The separation between region proposal and classification/refinement enhances detection precision.
* **Modular and Extensible:** Its architecture is well-suited for research and customization, supporting integration of advanced feature extractors or loss functions.

**Disadvantages:**

* **High Computational Cost:** Due to its two-stage pipeline, Faster R-CNN is significantly slower and more resource-intensive than one-stage counterparts.
* **Redundant Predictions:** The model often produces overlapping bounding boxes that rely heavily on NMS to resolve, which can reduce interpretability.
* **Training Complexity:** Requires careful tuning of multiple subcomponents (e.g., RPN, classifier, regression heads) for optimal performance.

**3. DETR (DEtection TRansformer)**

**Advantages:**

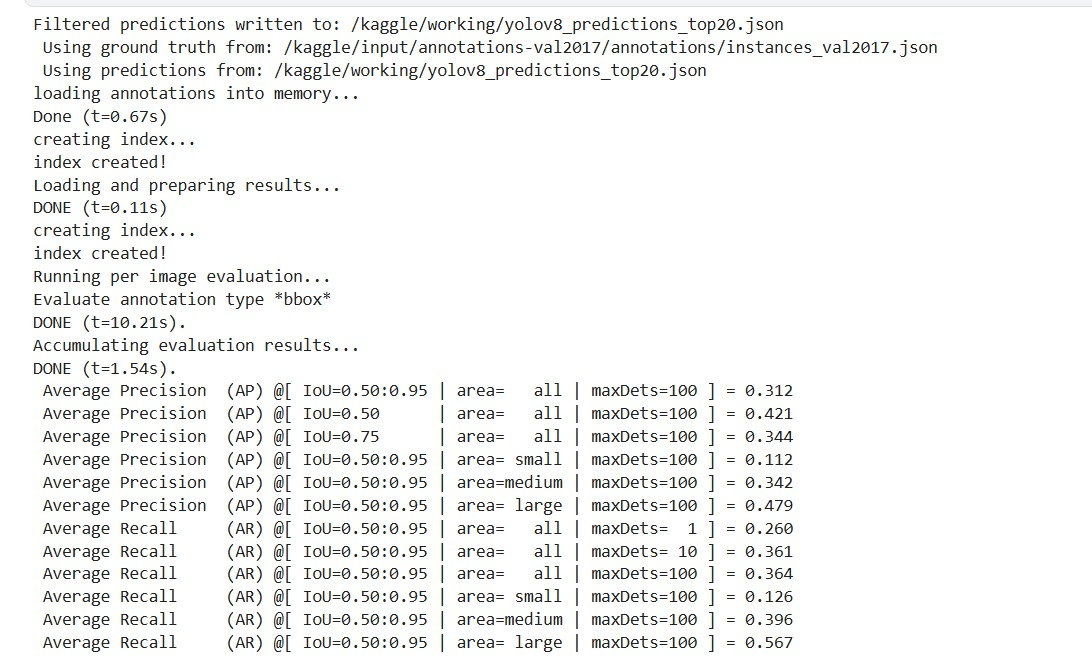
* **End-to-End Learning:** DETR eliminates the need for traditional heuristics (e.g., anchor boxes, NMS), offering a fully trainable object detection pipeline.
* **Global Scene Understanding:** Leveraging transformer-based self-attention, DETR captures contextual relationships across the entire image.
* **Non-Redundant Predictions:** Through bipartite matching, DETR ensures each output corresponds to a unique object, avoiding detection overlaps.
* **Strong Semantic Awareness:** Excels in scenarios where object semantics and spatial relationships are complex and context-dependent.

**Disadvantages:**

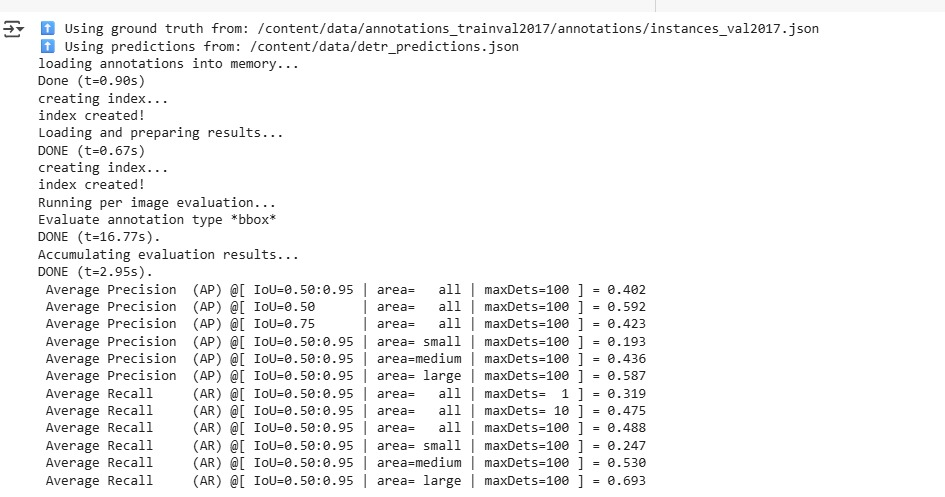
* **Slow Convergence:** DETR requires extensive training time and large datasets to reach competitive performance levels.
* **Limited Detection of Small or Low-Priority Objects:** The fixed-size output can lead to omission of less salient objects in dense scenes.
* **Inadequate for Real-Time Use:** The computational overhead of self-attention makes DETR unsuitable for time-sensitive or edge-based applications.

## Precision:

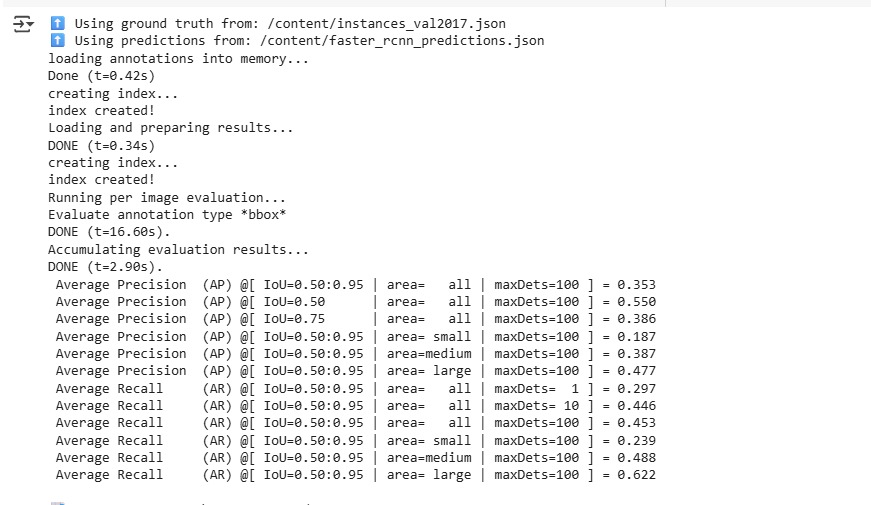
Yolov8:



Detr:



Faster r cnn



## Conclusion

In conclusion, the diversity of object detection models reflects the wide variety of application requirements in computer vision. DETR offers a research-oriented architecture that leverages transformers for global context modeling, making it ideal for scenes with high complexity, albeit at the cost of speed and training demands. YOLOv8 balances speed and accuracy, emerging as the preferred solution for real-time detection tasks on low-power devices. Faster R-CNN maintains its position as a gold standard for high-precision tasks, especially where small object detection and occlusion handling are crucial. The selection between these models must consider trade-offs in computational cost, inference time, and accuracy based on the specific use case.