

# OBJECT DETECTION AND YOLO...

- What's the difference between object detection and object recognition?

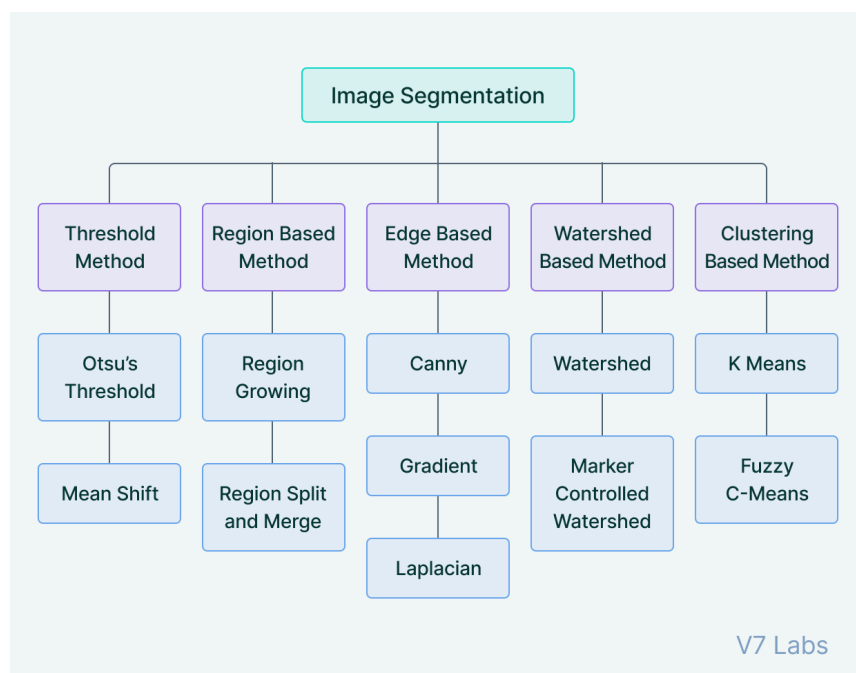
Object detection is the field of computer vision that deals with the localization and classification of objects contained in an image or video. Simply– Drawing bounding boxes around objects in an image allows us to detect them.

Image classification sends a whole image through a classifier (such as a deep neural network) for it to spit out a tag. Classifiers take into consideration the whole image but don't tell you *where* the tag appears in the image. Object detection is slightly more advanced, as it creates a bounding box around the classified object.

So, Detection: ability to distinguish an object from the background. Recognition: ability to classify the object class (animal, human, vehicle, boat ...) Identification: ability to describe the object in details (a man with a hat, a deer, a Jeep ...)

- What is Instance Segmentation?

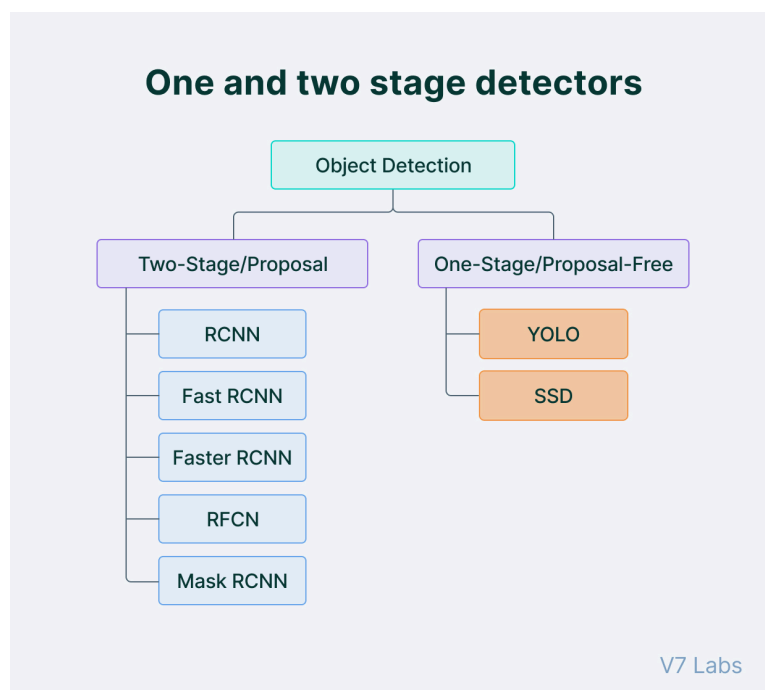
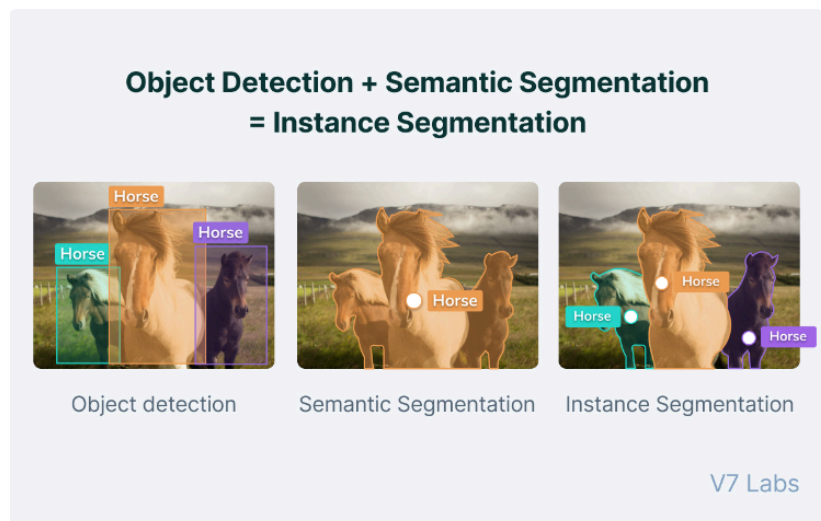
Image segmentation is the process of defining *which pixels* of an object class are found in an image.



Semantic image segmentation will mark all pixels belonging to that tag, but won't define the boundaries of each object.

Object detection instead will not segment the object, but will clearly define the location of each individual object instance with a box.

Combining semantic segmentation with object detection leads to instance segmentation, which first detects the object instances, and then segments each within the detected boxes (known in this case as regions of interest).



- What are bounding boxes?

Bounding boxes are used to label data for computer vision tasks, including: Object Detection: Bounding boxes identify and localize objects within an image, such as detecting pedestrians, cars, and animals. They represent object locations and are compatible with many machine-learning algorithms.

- Which computer vision technique should I use?

Computer Vision is a subfield of Deep Learning and Artificial Intelligence where humans teach computers to see and interpret the world around them.

While humans and animals naturally solve vision as a problem from a very young age, helping machines interpret and perceive their surroundings via vision remains a largely unsolved problem.

Limited perception of the human vision along with the infinitely varying scenery of our dynamic world is what makes Machine Vision complex at its core.

1. Computer Vision is a subfield of Deep Learning and Artificial Intelligence that enables computers to see and interpret the world around them.
2. Applying computer vision technology isn't new—it dates back to the 1950s.
3. In its most basic form, computer vision is about acquiring, processing, and understanding an image.
4. Some of the common computer vision problems include image classification, object localization and detection, and image segmentation.
5. Computer vision applications include fields like: facial recognition technology, medical image analysis, self-driving cars, and intelligent video analytics.
6. Nowadays, a computer vision system can surpass a human vision system.

- How should I build an accurate object detection model?

To determine and compare the predictive performance of different object detection models, we need standard quantitative metrics.

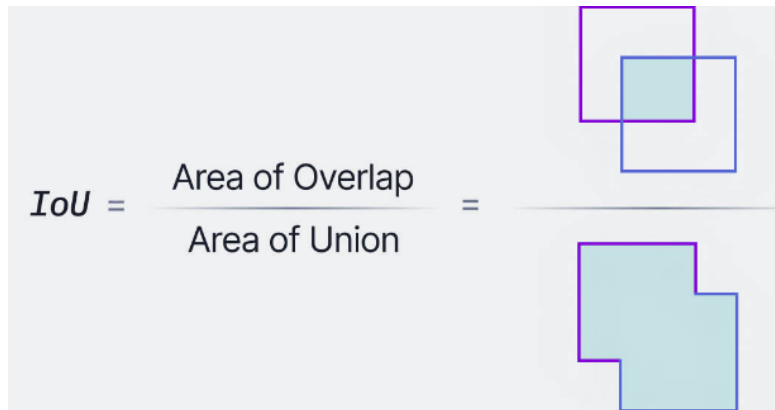
The two most common evaluation metrics are Intersection over Union (IoU) and the Average Precision (AP) metrics.

### Intersection over Union (IoU)

Intersection over Union is a popular metric to measure localization accuracy and calculate localization errors in object detection models.

To calculate the IoU between the predicted and the ground truth bounding boxes, we first take the intersecting area between the two corresponding bounding boxes for the same object. Following this, we calculate the total area covered by the two bounding boxes— also known as the “Union” and the area of overlap between them called the “Intersection.”

The intersection divided by the Union gives us the ratio of the overlap to the total area, providing a good estimate of how close the prediction bounding box is to the original bounding box.


$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} =$$

### Average Precision (AP)

Average Precision (AP) is calculated as the area under a precision vs. recall curve for a set of predictions.

Recall is calculated as the ratio of the total predictions made by the model under a class with a total of existing labels for the class. Precision refers to the ratio of true positives with respect to the total predictions made by the model.

Recall and precision offer a trade-off that is graphically represented into a curve by varying the classification threshold. The area under this precision vs. recall curve gives us the Average Precision per class for the model. The average of this value, taken over all classes, is called mean Average Precision (mAP).

In object detection, precision and recall aren't used for class predictions. Instead, they serve as predictions of boundary boxes for measuring the decision performance. An IoU value  $> 0.5$  is taken as a positive prediction, while an IoU value  $< 0.5$  is a negative prediction.

- What is YOLO?

‘You Only Look Once’ uses an end-to-end neural network that makes predictions of bounding boxes and class probabilities all at once. It differs from the approach taken by previous object detection algorithms, which repurposed classifiers to perform detection. YOLO performs all of its predictions with the help of a single fully connected layer.

One of the main advantages of YOLO is its fast inference speed, which allows it to process images in real time. It's well-suited for applications such as video surveillance, self-driving cars, and augmented reality.

Additionally, YOLO has a simple architecture and requires minimal training data, making it easy to implement and adapt to new tasks.

Despite limitations such as struggling with small objects and the inability to perform fine-grained object classification, YOLO has proven to be a valuable tool for object detection and has opened up many new possibilities for researchers and practitioners. As the field of Computer Vision continues to advance, it will be interesting to see how YOLO and other object detection algorithms evolve and improve.

WHAT IS COCO 8?

## What is COCO?



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

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

## Collaborators

Tsung-Yi Lin Google Brain  
Genevieve Patterson MSR, Trash TV  
Matteo R. Ronchi Caltech  
Yin Cui Google  
Michael Maire TTI-Chicago  
Serge Belongie Cornell Tech  
Lubomir Bourdev WaveOne, Inc.  
Ross Girshick FAIR  
James Hays Georgia Tech  
Pietro Perona Caltech  
Deva Ramanan CMU  
Larry Zitnick FAIR  
Piotr Dollár FAIR

## Sponsors



CODE...

Here's a breakdown of each part:

### 1. Importing YOLO from Ultralytics:

```
from ultralytics import YOLO
```

This line imports the YOLO class from the Ultralytics library, allowing you to use it to create, train, and evaluate YOLO models.

### 2. Creating or Loading YOLO Model:

```
# Create a new YOLO model from scratch  
model = YOLO("yolov8n.yaml")
```

```
# Load a pretrained YOLO model (recommended for training)
model = YOLO("yolov8n.pt")
```

This part shows two options for initializing the YOLO model.

- The first option creates a new YOLO model from scratch, using the configuration specified in the "yolov8n.yaml" file.
- The second option loads a pre-trained YOLO model from the "yolov8n.pt" file. This is recommended when you want to train the model further.

### **3. Training the Model:**

```
# Train the model using the 'coco8.yaml' dataset for 3 epochs
results = model.train(data="coco8.yaml", epochs=3)
```

This line trains the YOLO model using the COCO dataset with configurations specified in the "coco8.yaml" file for 3 epochs. The training results are stored in the results variable.

### **4. Evaluating the Model:**

```
# Evaluate the model's performance on the validation set
results = model.val()
```

This line evaluates the trained model's performance on the validation set. It calculates metrics such as precision, recall, and mAP (mean Average Precision). The evaluation results are stored in the results variable.

### **5. Performing Object Detection:**

```
# Perform object detection on an image using the model
results = model("https://ultralytics.com/images/bus.jpg")
```

This line performs object detection on the image specified by the URL using the trained YOLO model. The detection results are stored in the results variable.

### **6. Exporting the Model:**

```
# Export the model to ONNX format
success = model.export(format="onnx")
```

This line exports the trained YOLO model to the ONNX format.

The success of the export operation is stored in the success variable.

Each part of this code serves a specific purpose, from initializing the model to training it, evaluating its performance, performing inference, and exporting it for deployment.

#### CITATIONS:

Redmon, J. (no date) *You only look once: Unified, real-time object detection, You Only Look Once: Unified, Real-Time Object Detection.* Available at: [https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2016/papers/Redmon\\_You\\_Only\\_Look\\_CVPR\\_2016\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf) (Accessed: 24 May 2024).

Metrics	Definition
Accuracy	The percentage of correctly detected objects compared to the total number of objects in the video.
Precision	The proportion of true positive detections out of all positive detections, indicating the model's ability to identify objects accurately.
Recall	The proportion of true positive detections out of the actual number of objects present in the video, representing the model's ability to detect all objects.
Mean Average Precision (mAP)	An overall performance measure that combines accuracy, precision, and recall to evaluate the model's object detection capabilities across multiple frames.