Background and terminology:

Basic Perception Terminology -

- BBox - bounding box
- GT - Ground truth
- Detections - the output of the system

Quality Metrics Terminology -

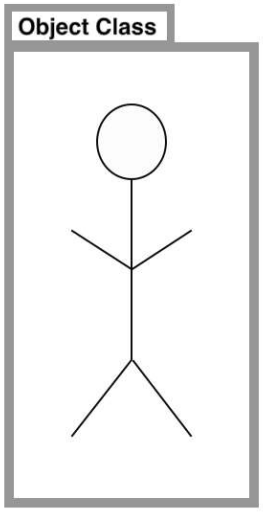
- IoU - Intersection over Union (IoU) - measures similarity between finite sample sets.
- TP - True Positive
    - if IoU ≥ 0.5, classify the object detection as True Positive(TP)
- FP - False Positive
    - if Iou <0.5, then it is a wrong detection and classify it as False Positive(FP)
- FN - False Negative
    - When ground truth is present in the image and the model fails to detect the object, classify it as False Negative(FN).
- Precision = TP / (TP + FP)
- Recall = TP/(TP+FN)
- FPPI (False Positives Per Image) = FP/(total amount of images)

Measuring Correctness via Intersection over Union

Most Object detection systems make predictions by outputting -

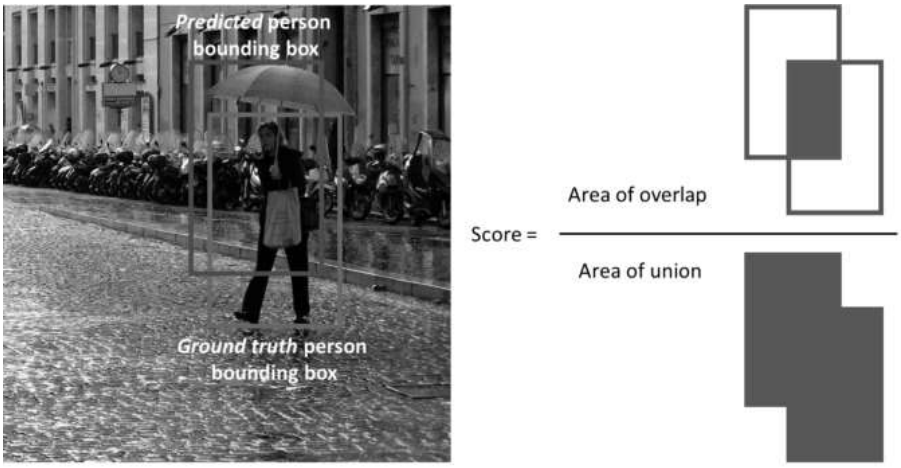
1. A bounding box - representing the position in the frame.
2. An Object class label - usually with the confidence level of the system.

In practice, a bounding box predicted in the **x_center, y_center, width, height** coordinates is sure to be off (even if only slightly) from the ground truth bounding box.



A bounding box prediction is incorrect if it doesn't detect the class correctly, but where should we draw the line for the required bounding box positioning precision?

The **Intersection over Union (IoU)** provides a metric to set this boundary. Measured as the area of the predicted bounding box that overlaps with the ground truth bounding box divided by the area of the union of both bounding boxes.



Picking the **reasonable single threshold for the IoU (iou_threshold)** metric should do the work.

1. Presentation Task

Given the attached video, **prepare a google-slides presentation** in English in which you analyze the systems' performance in the clip.

The analysis should be **qualitative**, and not **quantitative** - that this - describe your impressions of the system, its strengths, and weaknesses in the visualized results. The analysis should focus-on and include the terminologies defined above.

Add visual findings, examples, and suggestions for improvements guiding an R&D team to work on the most relevant issues.

The presentation should consist of no more than 5 slides - proposed structure:
1 slide - Background and terminology - presenting the metrics that will be used.
2-3 slides - Strengths and Weaknesses analysis with examples.
1 slide - Conclusions, pointers for the developers.

2. Python Question 1

Scenario:

We want to analyze the output of two detection files (containing bounding boxes) -

1. A GT(ground truth) file, which contains boxes labeled by humans. These describe the real location of the relevant objects in a frame(*'Q1_gt.tsv'*).

2. A Detection file, which contains the system's output - predicted boxes (*'Q1_system_output.tsv'*).

File Format - Sample Line:

```
'image_name.png' [
   {
      "x_center":,
      "y_center":,
      "width":,
      "height":,
   },]
```

Every line contains the image name and the coordinates of a detected object's bounding box.

instructions:

Your task is to implement **only** the missing functions in the python script - *Q1_DA_test.py*

```
def calculate_iou_for_2_boxes(box1, box2):
```

- Gets 2 boxes as input and returns the calculation of the intersection over union between the 2 Boxes (objects defined in the script).

For the below functions, assume box object has an already calculated iou attribute with the matched GT box, saved in the field (*box.iou*):

```
def calculate_average_iou(boxes, iou):
```

- The function gets a dictionary, mapping a frame to its boxes *{frame_name: list_of_boxes}*, and an iou_threshold, and returns the average IoU of all the boxes that pass the iou_threshold.

```
def create_historgram(boxes, iou_threshold):
```

- The function gets a dictionary mapping a frame to its boxes *{frame_name: list_of_boxes}*, and an iou_threshold, and outputs an IOU values histogram for the boxes that pass iou_threshold.
  x_axis: iou, y_axis: occurrences

```
def get_minimum_and_maximum_height(boxes, iou):
```

- Gets a dictionary mapping a frame to its boxes *{frame_name: list_of_boxes}*, and an iou_threshold, and outputs the minimum height and the maximum height of all the boxes that pass iou_threshold.

3. Python Question 2

Information:

- Two files are attached as resources for the following questions.
- Each file contains detected bounding-boxes and their properties, as a csv file, over a sequence of frames.
- The interval between the rows in the files (the frames timestamps in the sequence) is 60 milliseconds (0.06 seconds).
- Example - The first row represents the bounding box detected at time == 0 seconds. The second row represents the bounding box detected at time == 0.06 seconds, and so on.

instructions:

Write a program in Python that -

1. Loads the detection file '*Q2_detction_file_2.csv*'

2. Plots the distance of the detected object as a function of time, using the columns -
   - 'rw_kinematic_point_z' - Distances are in meters
   - 'time_sec' - Time is in seconds

3. Detects and removes the invalid distance measurements.
   Please plot the same graph as in the previous question, after the removal of the bad readings.

4. Calculates the velocity of the object (approximate constant velocity is sufficient).

5. Bonus Question - in file '*Q2_detction_file_2_bonus.csv*' the problematic distance readings now have valid values.
   Write a function that detects and removes outliers in the measurements, and re-plot the graph from question 2.