AUTOBRAINS

Home Assignment
Data Engineer

Version 1.1
Author: Eden A

Overview:
The following document contains the home assignment for candidates of the Data Engineer position at AutoBrains.

Important notes:

- Use coherent names in code.
- Always validate user input.
- All applications should be repeatable (Given the same input, the application)
- Should behave the same always, and produce the same output.
- Submissions must include working code only.

Background and terminology:

Basic Perception

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

Quality Metrics Terminology -

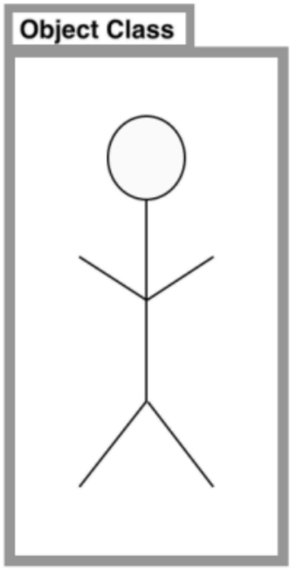
- IoU - Intersection over Union (IoU) - measures similarity between finite sample sets.
- TP - True Positive
  - if IoU ≥ 0.5, classify the object detection as True Positive(TP)
- FP - False Positive
  - if IoU <0.5, then it is a wrong detection and classify it as False Positive(FP)
- FN - False Negative
  - 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

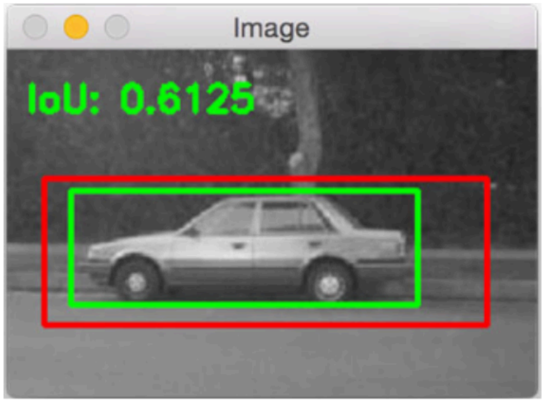
Most Object detection systems make predictions by output -

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

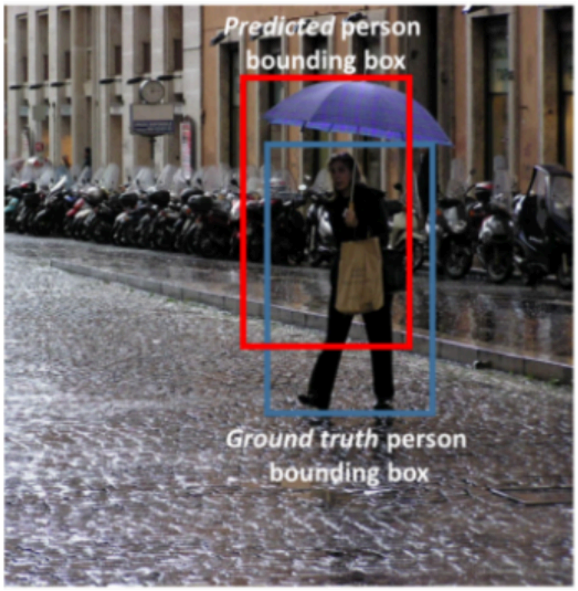
In practice, a bounding box predicted in the x_center, y_center, width, and height coordinates are sure to be off (even if only slightly) from the ground truth bounding box.

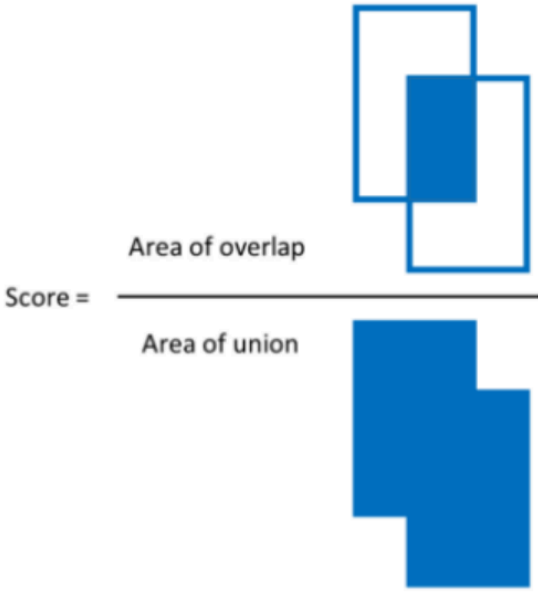


**A bounding box prediction is incorrect if it doesn’t detect the class correctly, but where should
we draw the line for the required bounding box positioning precision?**



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.


Predicted person bounding box
Ground truth person bounding box



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

Part 1: Calculate the average IOU metric:

Use Case:
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'*).

<u>File Format - Sample Line:</u>

```
'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 (No change to the main function) 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.

Part 2: Choose frames that will be used for training

Use-case:
As part of our continuous effort to refine our AI models, particularly in addressing niche scenarios, we aim to enhance the training dataset effectively. To achieve this, we must carefully select images for our training sets, ensuring that they are tagged appropriately and utilized for model training. Our goal is to choose images strategically, minimizing the quantity while maximizing the impact on improving the model's performance on edge cases. These edge cases may include unique instances such as specialized vehicles like tractors or three-wheelers.

Resources:

- Highend_results.tsv - the high-end results are the detections of a high-end and heavy object detection network, that is being used as an “auto-labeling” network. You can consider the detections as a baseline for GT.
- Detection_results.tsv- the model object detection results file.
- Images for this set are available as “Q2_Images” folder

The task:
Utilizing the two results files you have as resources and using the matching methodology used in part 1, create a list of images that should be sent for our tagging pipeline, and used as training data to improve our model.
Please share your code, and explain your way of thinking, why you chose those images, based on what rules, and what the logic behind it is.

Part 3: Data Anomalies Report

Use Case:
You want to update GT for images in Autobrains Data Center so other users can use it by need, before that- you need to validate the data

Sources:
GT.tsv
Q3_Imaged directory with the corresponding frames to the GT file.

GT Format Explanation:
Each row represents a bounding box in a frame with its coordinates and further details.
TSV file with the columns below:
Name - Name of the image, generally represents the timestamp of frame recorded
x_center - x-axis value of the center of the bounding box - should have only positive non-zero values.
y_center - y-axis value of the center of the bounding box - should have only positive non-zero values.
width - width value of the bounding box - should have only positive non-zero values.
height - height value of the bounding box - should have only positive non-zero values.
label - object/label/class of the bounding box
is_rider_on_2_wheels - relates only for classes of BIKE/MOTOR or any object with 2 wheels - value 0 or 1. Value 0 is assigned when there is no rider on top of the object. Value 1 is assigned when there is a rider on top of the object.
d3_separation - relates only for classes of 4W or MOTOR such as CAR, TRUCK, VAN, etc. This value represents the x value (vertical line) that splits the object into 2 different parts e.g. - the SIDE and FRONT of a car.

The task:
Please search for anomalies in the data, and document, and illustrate the anomalies you found comprehensively. Mention per each anomaly how you found it and how you will suggest addressing those anomalies.

Part 4: Pipeline designing and presentation

Use-case:
Let’s delve into the pivotal use-case of tagging training data, a routine and indispensable task in our operational workflow. This process involves the identification of images for tagging, executed in a semi-automated manner, followed by tagging and subsequent uploading to our company's data center. These tagged images are then made accessible to our deep-learning developers. Beyond tagging objects within the images, we also include metadata such as time of day, weather conditions, and location. This additional metadata serves to augment the usability of the data for our DL developers. To streamline our operations, we utilize Jira for managing open tasks, encompassing data collection, tagging, validation, and uploading to the data center.

Task:

Your task is to design a comprehensive pipeline for 'Train Data Tagging' based on the provided use-case and the preceding sections of this assignment. This pipeline should encompass data selection, tagging, validation, and monitoring of the entire pipeline's status. Your presentation should effectively encapsulate this pipeline on a slide for a comprehensive understanding.

Submission:

You should submit a zip file with the following files:

1. Part 1 + 2: A folder with the Python files and the input files that were used to create the results.
2. Part 1: A PDF with your Python part answers:
   a. What is the average IOU?
   b. Histogram plot
   c. What are the minimum height and the maximum height of all the boxes that pass iou_threshold
3. Part 2: a text file including the list of images that were chosen
4. Part 2: A PDF with explanation and examples for part 3 solution.
5. Part 3: PDF/ Jupyter notebook describing the anomalies analysis
6. Part 4: PDF with the pipeline slide