Objective

Setup an ML pipeline which

1. takes input as a video file
2. `av` Python library to extract video frames and their timestamps
3. Prepare a yolo object detection model to detect Pepsi or CocaCola logos and a Python script which detects these logos in the exemplary video

Input: `video file`

Output: `json file`

```
{
        "Pepsi_pts"    : [10.1,10.2,10.3,...],
        "CocaCols_pts" : [20.3,31.8,40.12,...],
}
```

- `label` : List[time_stamps]
- (optional) size and distance from the centre of the frame

Research

YOLO (You Only Look Once) is a real-time object detection algorithm.It is a single-stage object detector that uses a convolutional neural network (CNN) to predict the bounding boxes and class probabilities of objects in input images.

To make it easier to work with I am using a library called `Ultralytics` which provides a YOLO8 model which is fast and performant with an easily usable APIs to train from dataset to model in minutes along with testing and prediction and benchmarking builtin.

Datasets

The required datasets were easily found online from `RoboFlow` .

Pepsi:

1. https://universe.roboflow.com/detectionanas/pepsi-logo-detection

This dataset had around 1500 images (pre annotated), making it the perfect choice.

CocaCola:

1. https://universe.roboflow.com/test01-fr735/brands-qv6fs

   This dataset only had 650 images (pre annotated), but included a wide variety.

2. https://universe.roboflow.com/tung-le-ssyfl/coca-cola-sam4d

   This dataset was added to increase CocaCola images by 500+ (pre annotated).

Merged:

1. RoboFlow provided an easy way to merge them, all i had to do was download both unzip them and then upload them in a new project and the rest worked like magic:

   https://universe.roboflow.com/name-y67ms

This shows the 3 variations of the merged datasets done so to increase CocaCola accuracy.

Testing

First I tested both the datasets individually by downloading them in code with the RoboFlow api and trained a small (5 epoch) model and tested it on a simple 30 sec clip and adjusted the confidence threshold as required.

Training

For training the merged model i used Google Colab Notebook (as i was not able to get torch to recognize my laptops 4050 GPU 😔). I trained a 200 and 50 epoch model to be used in the final Pipeline.

https://colab.research.google.com/drive/1MtmCI5rDIDGe1zvlUbXw-Slv2u9NjWKu?usp=sharing

Problems

1. When i used it on the example video i observed some false positives, because the background matched the colour schemes of the logos, so to tackle this increased the confidence threshold.

2. This then led to another problem. Sometimes the CocaCola can is missed as it is round in shape and shows only the middle and it got worse when I increased the confidence threshold to deal with the first problem. This had to be fixed by finding the sweet spot to not let both of them arise but it was far from perfect (accuracy 85%).

3. I then thought if the CocaCola dataset was lacking pictures so i made another merged dataset this time using one more dataset for CocaCola:

   https://universe.roboflow.com/tung-le-ssyfl/coca-cola-sam4d

4. This gave the model more CocaCola Context, yet since the dataset images were in good conditions and they did not cover cans with the logo/name written horizontally and the example video had that it was able to detect with about 50% threshold but i still got 2-3 false positives in the 11 min video from the background.

5. This had me really worried about the model but to test on other condition i downloaded an ad clip which had both logos in them and to my surprise the model performed very well and had 100% accuracy and even detected smaller logos when up close and i had no false positives and for the original example the same solution of finding the sweet spot of threshold worked in avoiding the false positives.

So to conclude the issue was with the logo on CocaCola cans being horizontal and no data with annotations for that (there were other datasets but they detected the cans themselves and not the logos).

The fix was to play around with the confidence threshold to find a value that maximised detections while keeping the false positives as low as possible (or more training data to fit the needs for real world scenarios which in hindsight is how the models can be scaled to face any test).

Models

Format - epoch_model_dataset

1. `200_best_small (best):`
   This model was trained on the combination of the 2 original datasets.
   It has the best accuracy

2. `200_best_large (don't use?):`
   This model had a dataset with increased CocaCola images.
   It still performed worse somehow than the first.

3. `50_best_final (try it?):`
   This model added some more CocaCola cans with horizontal logos.
   I have not tested it yet.

PipeLine

The final pipeline takes a video and model as input then outputs the video with annotations and json results to the same folder as input.

Optionally: you can pass the confidence threshold which defaults at 50.

Also the inTimestamps boolean is used to determine whether to output as av timestamps or frame counted strings.