Logo detection

Approach Document for Video Processing and Object Detection Pipeline

Table of Contents

1. Introduction
2. Objectives

Introduction

This document details the approach for developing a video processing brand logo detection pipeline aimed at detecting Pepsi and Coca-Cola in a video and recording their timestamps. The project consists of two main components: a Python script (`timestamp.py`) for finding timestamp by using custom trained model using Jupyter Notebook (`video.ipynb`).

Objectives

- Collecting databqse
    - first went to kaggle to check to get the data but no data found for coke and pepsi
    - then checked on roboflow as suggested in google.
    - Got the dataset for pepsi and coke but inn first time got the combined data and that was pepsi can and coke bottle.
    - then for more accurate downloaded pre-annotated dataset for pepsi and coke speractely.
    - as Yolo8 was seems good for more examples and python lib support not like yolo5 to cloning the repo to train.
    - yolo8 have a format for dataset like (<class_id> <x_center> <y_center> <width> <height>)

    - then merged the 2 datasets of coke and pepsi to single dataset.
    - as both dataset were downloaded seperatly both had class id 0.
    - so i changed the coke id to 1 to classify labels of coke and pepsi
    -
- Training the custom trained model in above dataset for coke and pepsi
    - now after combining both dataset to train.
    - installed ultralytics to import yolo for training
    - made changes in data.yaml
    - started training with yolo command with paramentes like epochs 200 for better accuracy as dataset had 1400 images in training data itself.
    - after waiting 1.5 hours i had got runs folder for stats of my model
    - used the best.pt for model with least loss.
- Using the trained model and detecting the logos in each frame and saving the timestamp: Uses a YOLO model to detect objects in the frames.
    - detecting the frame and checking whether frames consist coke or pepsi and updating the time stamp to json file.
    - After complete render we can save or return the json.