Inference script for detection (with Rajin's model)

Documentation & User Guide

📋 Table of Contents

Model Details

Purpose

The model is trained to detect and classify various Unilever Bangladesh Limited (UBL) products in retail environments, distinguishing them separately. My model (UBLModel_v2.pt) focuses on sachets, specifically Horlicks (CTSL 18ml), Clear Men's Shampoo (5ml), and Dove Conditioner (IRP DOLCE 7ml).

Classes

My model was trained on the following classes:

Sachet Products:

- **ClearMen5ml** - Specifically labeled and trained on this product
- **HorlicksCTSL18ml** - Specifically labeled and trained on this product
- **DoveConditionerIRPDOLCE7ml** - Specifically labeled and trained on this product

Model Versions

The best version of the model has been provided:

2. **UBLModel_v2.pt** - Trained for 500 epochs (recommended for better accuracy)

Installation

Prerequisites

- Python 3.8 or higher (my specific one was 3.13)
- Git (for cloning the repository)

Step 1: Clone the Repository

```
git clone https://github.com/rajin-khan/UBLRetailAnalysis.git
cd UBLRetailAnalysis/Pipeline
```

Step 2: Install Dependencies

```
pip install ultralytics opencv-python numpy tqdm pyyaml
```

or,

```
pip install -r requirements.txt
```

Running the Pipeline

Basic Usage

```
python pipeline.py --model ../Main/models/UBLModel_v2.pt --input ../test_images/ --output ../predictions/
```

Command Line Arguments

Argument	Description	Default
`--model`	Path to the trained YOLOv8 model (.pt file)	Required
`--input`	Path to input directory or image file(s)	Required
`--output`	Path to output directory	"output"
`--conf`	Confidence threshold for detections	0.25 (optional)
`--iou`	IoU threshold for non-maximum suppression	0.45 (optional)

Argument	Description	Default
`--img-size`	Image size for detection	640 (optional)
`--no-visualize`	Don't visualize detections	False (optional)
`--no-save-results`	Don't save detection results to text files	False (optional)

Examples

Process a Single Image

```
python pipeline.py --model ../Main/models/UBLModel_v2.pt --input /path/to/image.jpg --output ../predictions/
```

Process Multiple Images

```
python pipeline.py --model ../Main/models/UBLModel_v2.pt --input /path/to/image1.jpg /path/to/image2.jpg --output ../predictio
```

Process a Directory with Higher Confidence Threshold

```
python pipeline.py --model ../Main/models/UBLModel_v2.pt --input ../test_images/ --output ../predictions/ --conf 0.4
```

Understanding the Results

Output Structure

The pipeline generates several types of output:

1. **Visualized Images**:
   - Original images with bounding boxes around detected objects
   - Class labels and confidence scores displayed on the image
   - Detection quality and accuracy metrics shown in the top-left corner
   - Object count summary
2. **Per-Image Text Files**:
   - Lists of detected objects with their class names and counts
   - Estimated detection accuracy and quality assessment
   - Detailed information about each detection (coordinates, confidence)
3. **Summary File** (`detection_summary.txt`):
   - Overall detection accuracy and quality across all processed images
   - Total object counts by class
   - Per-image breakdown of detections

Accuracy Metrics

The pipeline provides two metrics to evaluate detection quality:

1. **Estimated Accuracy**: A percentage score derived from the average confidence of detections
   - 90-100%: Excellent detections (confidence > 0.8)
   - 80-90%: Good detections (confidence 0.6-0.8)
   - 70-80%: Moderate detections (confidence 0.4-0.6)
   - 50-70%: Low quality detections (confidence 0.25-0.4)
2. **Detection Quality**: A human-readable label (Excellent, Good, Moderate, or Low)

Dataset Information

Dataset Structure

Two folders have been provided:

/annotated: The original folder, with all the images together, and classes.txt and other files in /labels.

/dataset: The files after train/test/val split was done.

Labeling Process

The dataset was annotated using LabelImg, with specific focus on:

- Clear Men's Shampoo 5ml *(ClearMen5ml)*
- Horlicks Standard 18ml *(HorlicksCTSL18ml)*
- Dove Conditioner 7ml *(DoveConditionerIRPDOLCE7ml)*

Training Process

Data Preparation

1. **Collection**: Gathered images of Horlicks, Dove and Clear sachets
2. **Augmentation**: Applied techniques like rotation, cropping, brightness variations, and background noise using `albumentations` library to created augmented images.
3. **Annotation**: Used LabelImg to create bounding box annotations for each product class

Model Training

1. **Initial Training**:
   - 50 epochs locally using YOLOv8 architecture
   - Resulted in `UBLModel_v1.pt`
2. **Extended Training**:
   - 500 epochs on Google Colab with TPU/GPU acceleration
   - Resulted in `UBLModel_v2.pt` with improved generalization
3. **Hyperparameter Tuning**:
   - Adjusted batch size, learning rate, and augmentation strategies
   - Optimized for best performance across diverse retail environments

I have also provided the **dataset.yaml** used when training.

Getting Info

For additional assistance, please refer to my GitHub repository for the latest updates and issue tracking.