Spatial and Semantic Scene Graph Generation in Retail

Amaan Sheikh
aas2438
Deep learning for Computer Vision
12/4/2025

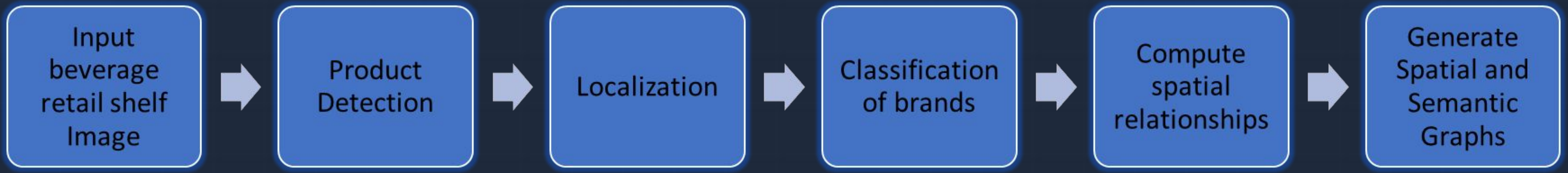
THE RETAIL PROBLEM

Planogram Compliance

- **Goal:** Automate auditing to verify SKU
- **Current Challenge:** Most CV models struggle on densely packed scenes with high occlusion
- **The Difficulty:** Extreme visual similarity (e.g., distinguishing Coke vs. Coke Zero) and high object counts
- **Solution:** A system that locates items and reasons about them semantically and spatially

Overall Pipeline

Targeted brands: Coca-cola, Fanta, Pepsi, Sprite, Mountain Dew, 7UP



Dataset Preparation

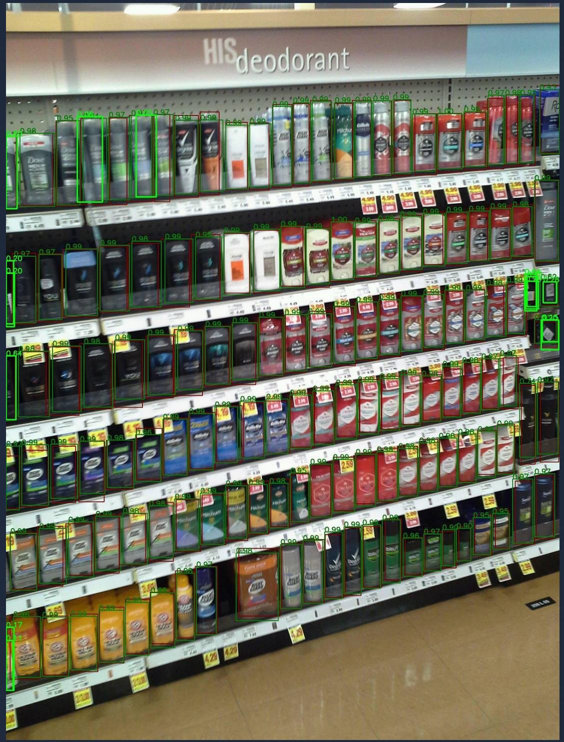
Multiple datasets for roughly 3 categories

A) Detection

Dataset: SKU-110K

Size: 10,000+ images

Classes (1) : Product



B) Classification

Dataset: Combination and augmentation of 3 datasets: Refrescos Dataset, Pepsi dataset, Cold drinks dataset

Size: 4,000+ images

Classes (6): Coca-cola, Fanta, Pepsi, Sprite, Mountain Dew, 7UP

Dataset Preparation (continued)

Multiple datasets were used

C) Overall pipeline testing

- Dataset: Cold Drinks Inventory Dataset and Beverage Detection Dataset
- Source: Kaggle
- Size: 200+ images
- Classes (6): Coca-cola, Fanta, Pepsi, Sprite, Mountain Dew, 7UP

Main Stages

Stage 1: Detection & Localization

Goal: Locate every object box. Classify as a generic "object"

Models: RT-DETR Large, YOLOV11n,YOLOV11L via Ultralytics and PyTorch

Why?
Comparing the accuracies of the 3 finetuned models based on hyerparameter search and then fine-tune the best one for more epochs

Stage 2: Recognition

Goal: Identify specific products within boxes.

Model: CLIP via Hugging Face and PyTorch

Input image → detected, localized, classified image

(Input)

(Detection)

(Classification)

Main Stages (continued)

Stage 3: Compute Relationships

Use geometry to build the relationships using the localized coordinates to build the knowledge graph

Relationships computed using geometry formulas and opencv

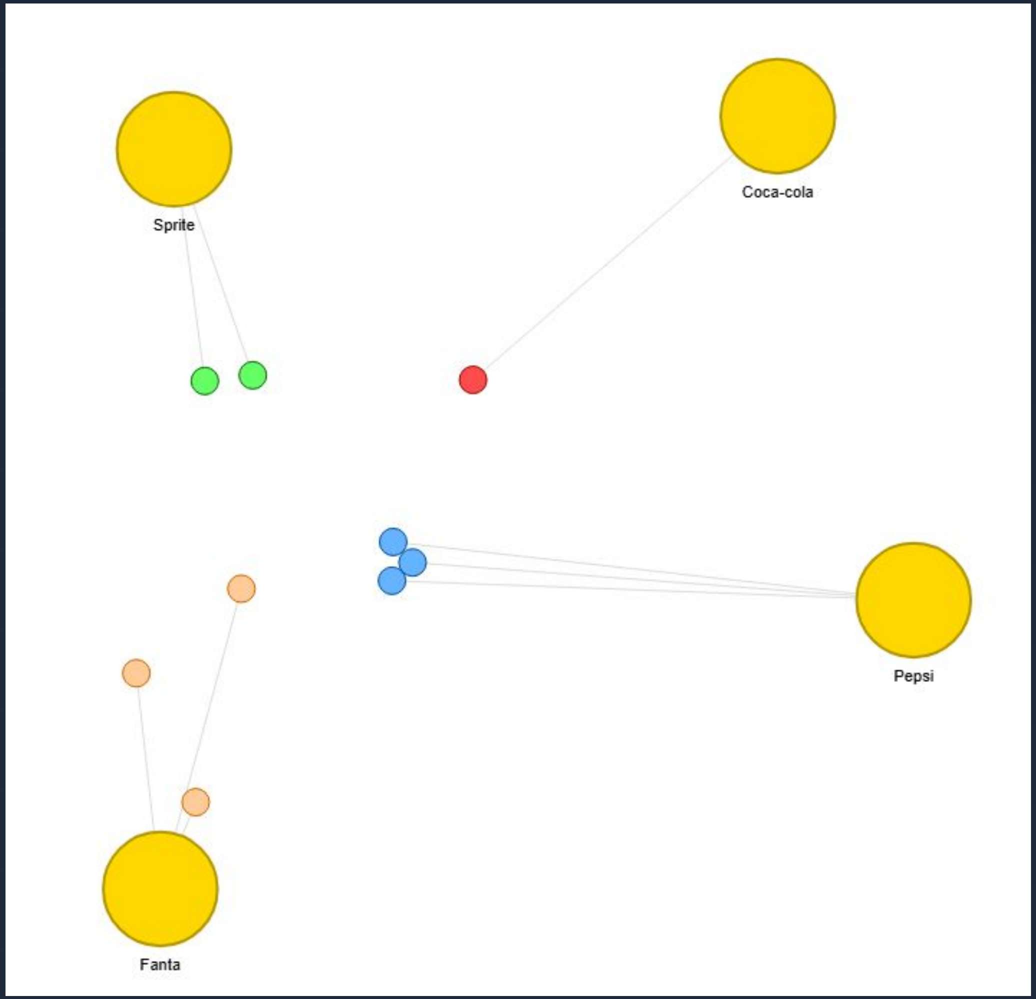
Stage 4: Graphs and Interface

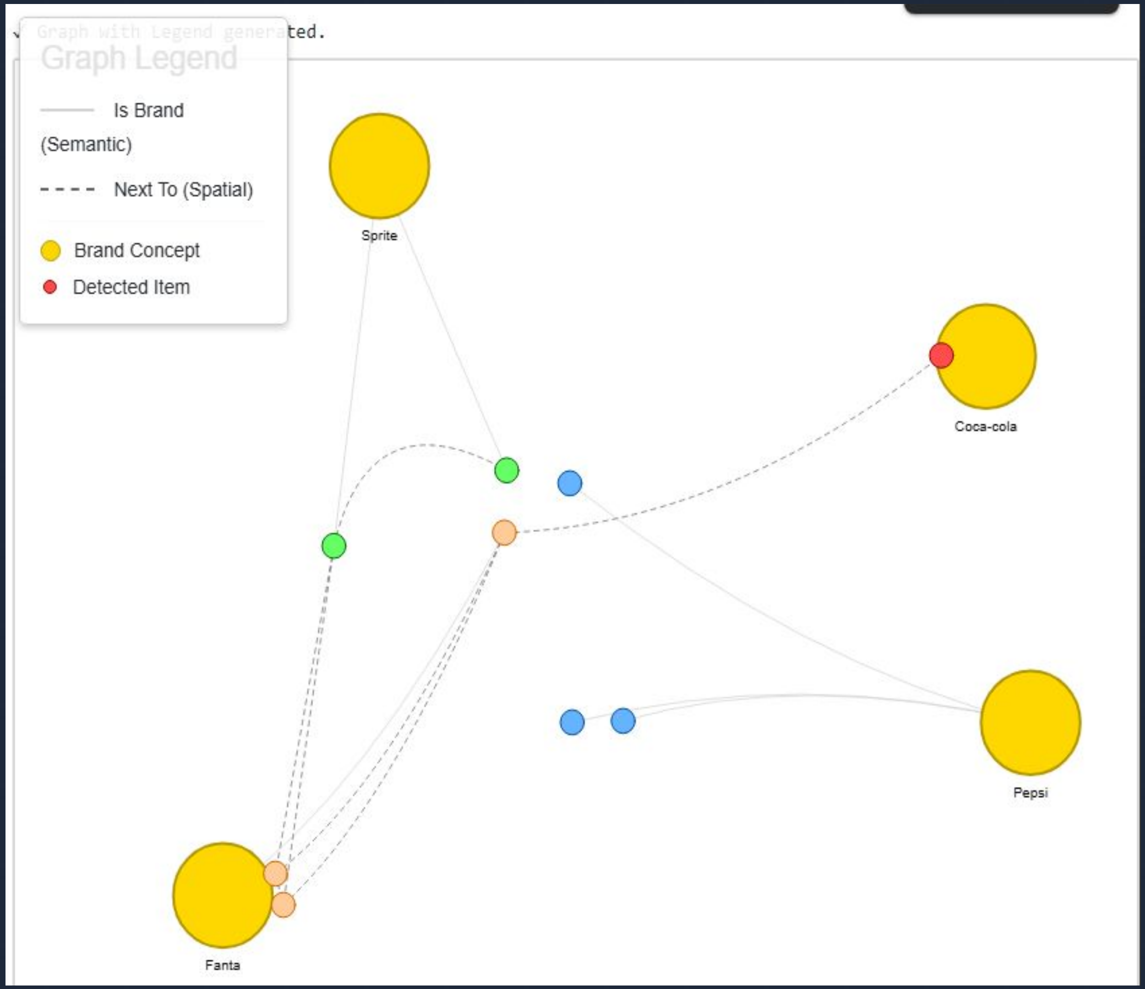
Build the Spatial and Semantic Knowledge graphs and display them on the User Interface

Graphs were built via NetworkX, Pyviz

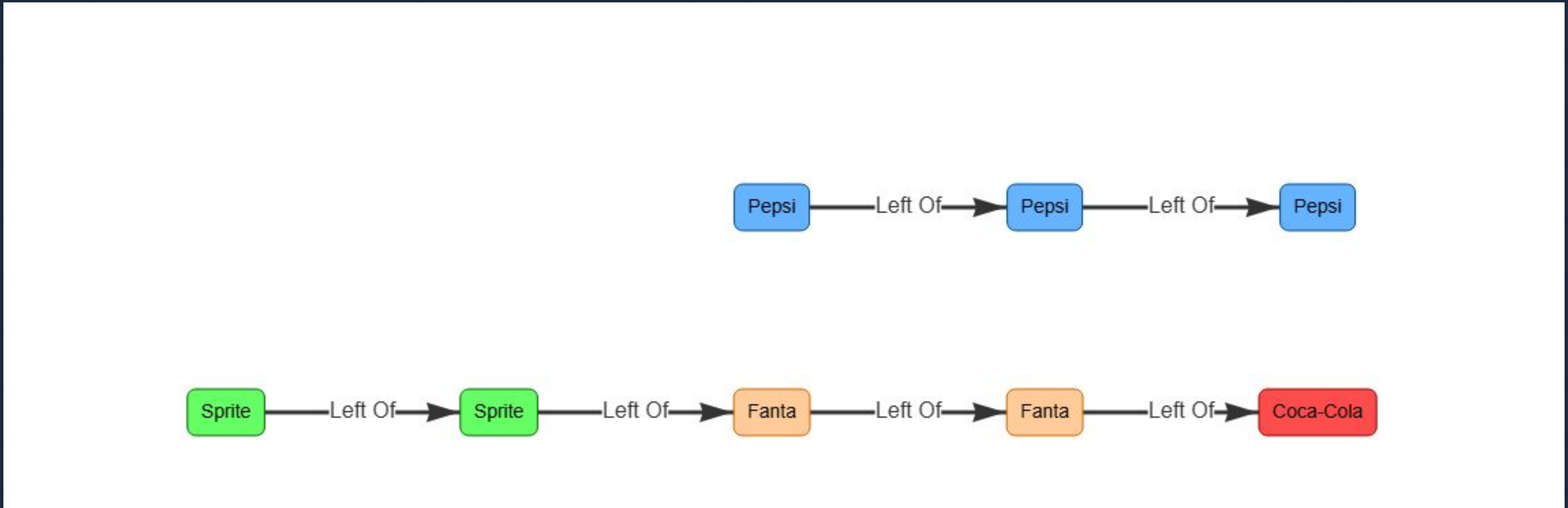
User Interaface was built using streamlit

Output scene / semantic graphs





Output scene / semantic graphs (continued)



Models Training: Hyperparameter Search

Objective: Maximize detection accuracy (mAP) for dense, overlapping objects along with precision and recall

Method: 3 to 5 iterations of fine tuning with different hyperparameters

Strategy: We defined distinct search spaces for CNN-based (YOLO) vs. Transformer-based (RT-DETR) architectures to respect their structural differences
Epochs varied from 10 to 15 depending on the model.

Hyperparameters-

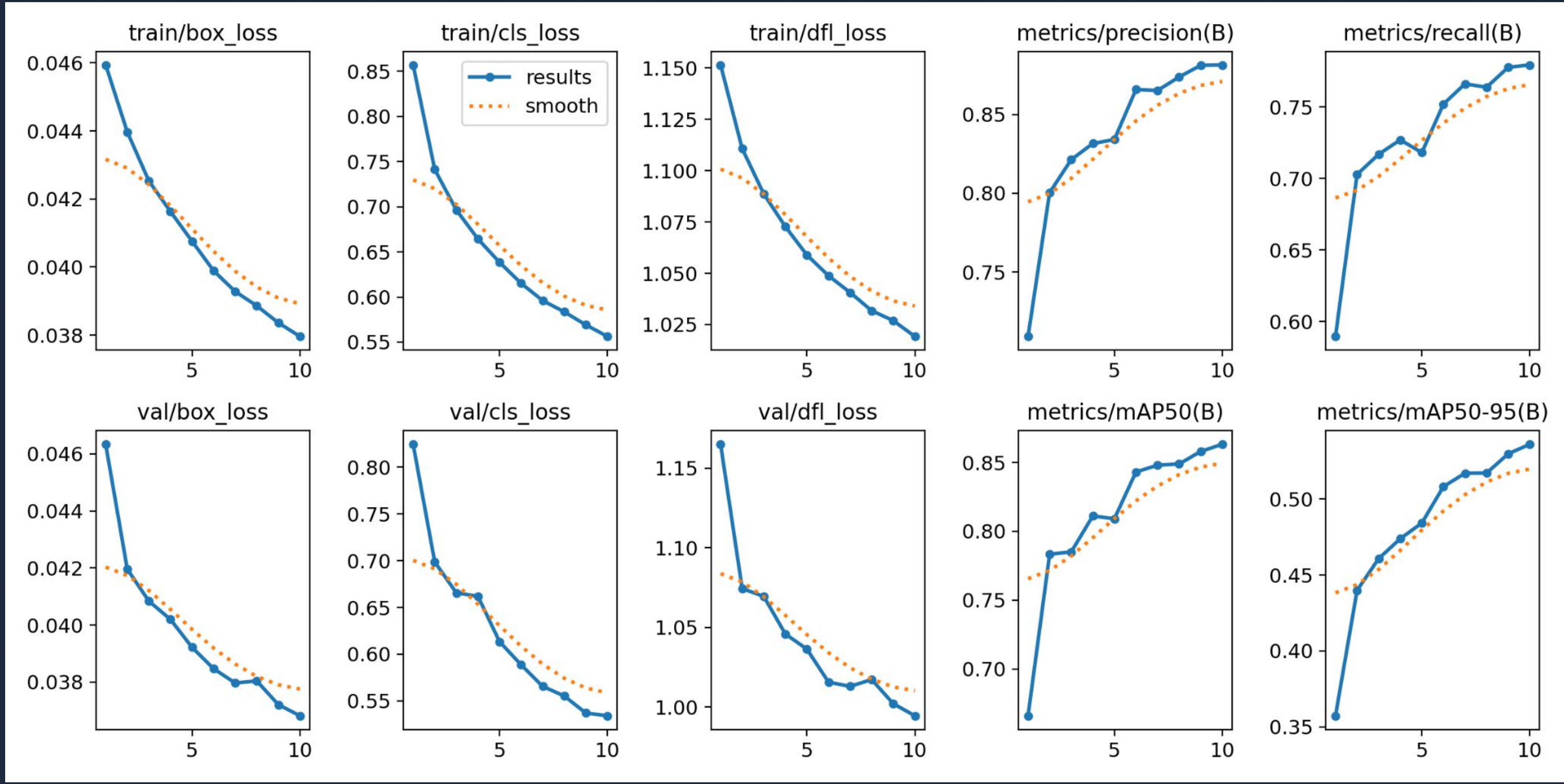
Hyperparameter	YOLOv11 (CNN) Space	RT-DETR (Transformer) Space	Reasoning
Learning Rate (lr0)	1e-4 to 1e-2	1e-5 to 1e-3	Transformers require lower, more stable learning rates to converge.
Final LR (lrf)	0.01 to 1.0	0.01 to 1.0	Controls how much the learning rate decays over time.
Momentum	0.7 to 0.98	0.9 to 0.95	RT-DETR benefits from stricter momentum control to stabilize attention weights.
Weight Decay	0.0 to 0.001	1e-4 to 5e-4	Regularization to prevent overfitting on the specific retail textures.
Box Loss Gain	0.02 to 0.2	0.02 to 0.2	Weight given to the bounding box regression loss.
Cls Loss Gain	0.2 to 4.0	0.5 to 4.0	Weight given to the classification loss (higher for RT-DETR to force class separation).
HSV-H Augment	0.0 to 0.05	N/A (Default)	Hue augmentation helps YOLO generalize to different store lighting conditions.
Mosaic Augment	0.0 to 1.0 (High)	0.0 to 0.5 (Low)	Transformers struggle with heavy spatial distortion (Mosaic) compared to CNNs.
MixUp Augment	0.0 to 1.0 (High)	0.0 to 0.2 (Low)	Excessive image mixing confuses the attention mechanism in RT-DETR.

Detection Results (YOLOV11n)

Best iteration model-

Test Results:

- mAP@50: 0.8794
- mAP@50-95: 0.546
- Precision: 0.885
- Recall: 0.7940

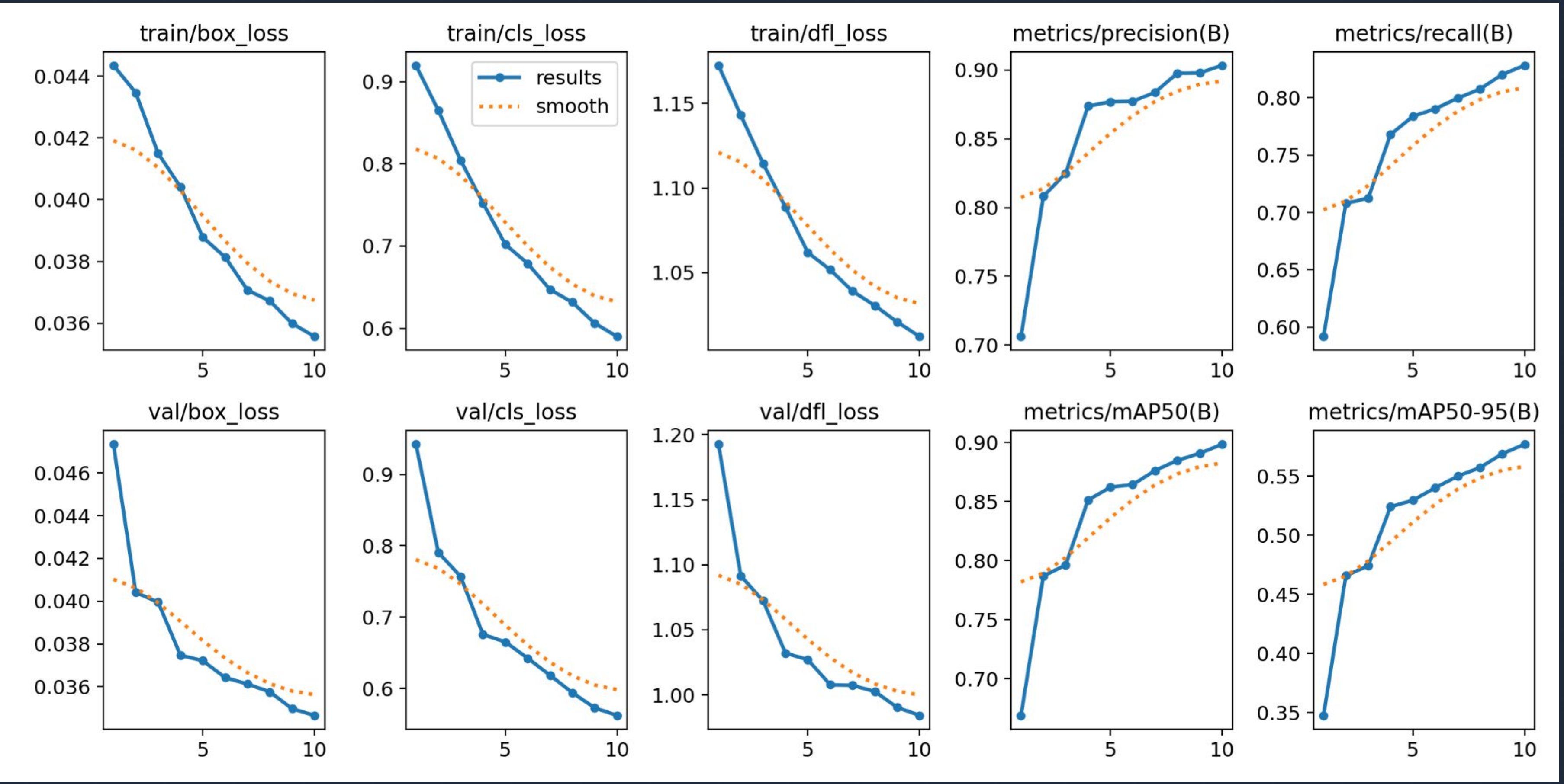


Detection Results (YOLOV11L)

Best iteration model-

Test Results:

- mAP@50: 0.9158
- mAP@50-95: 0.588
- Precision: 0.9030
- Recall: 0.8436

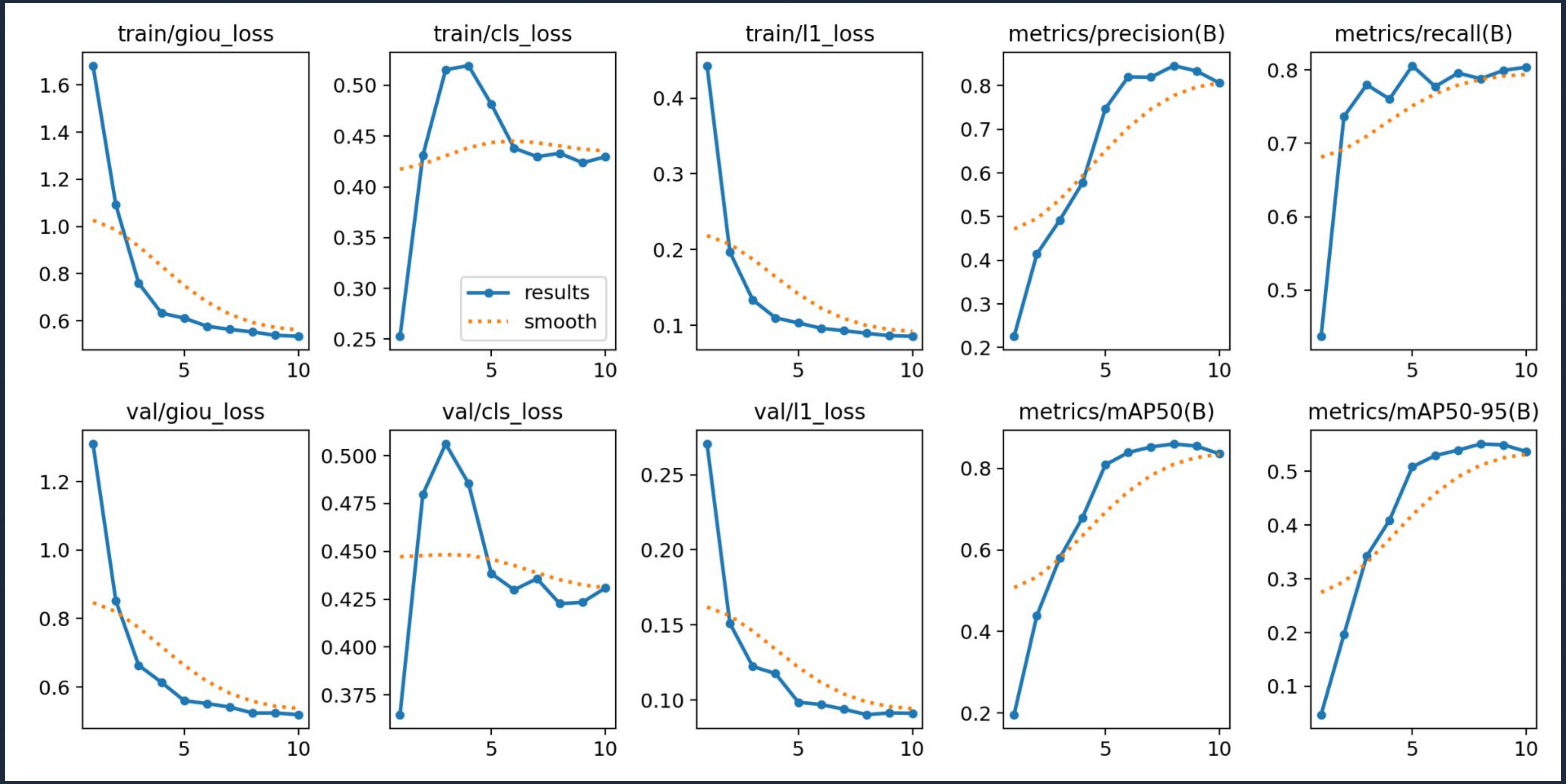


Detection Results (RT-DETR)

Best iteration model-

Test Results:

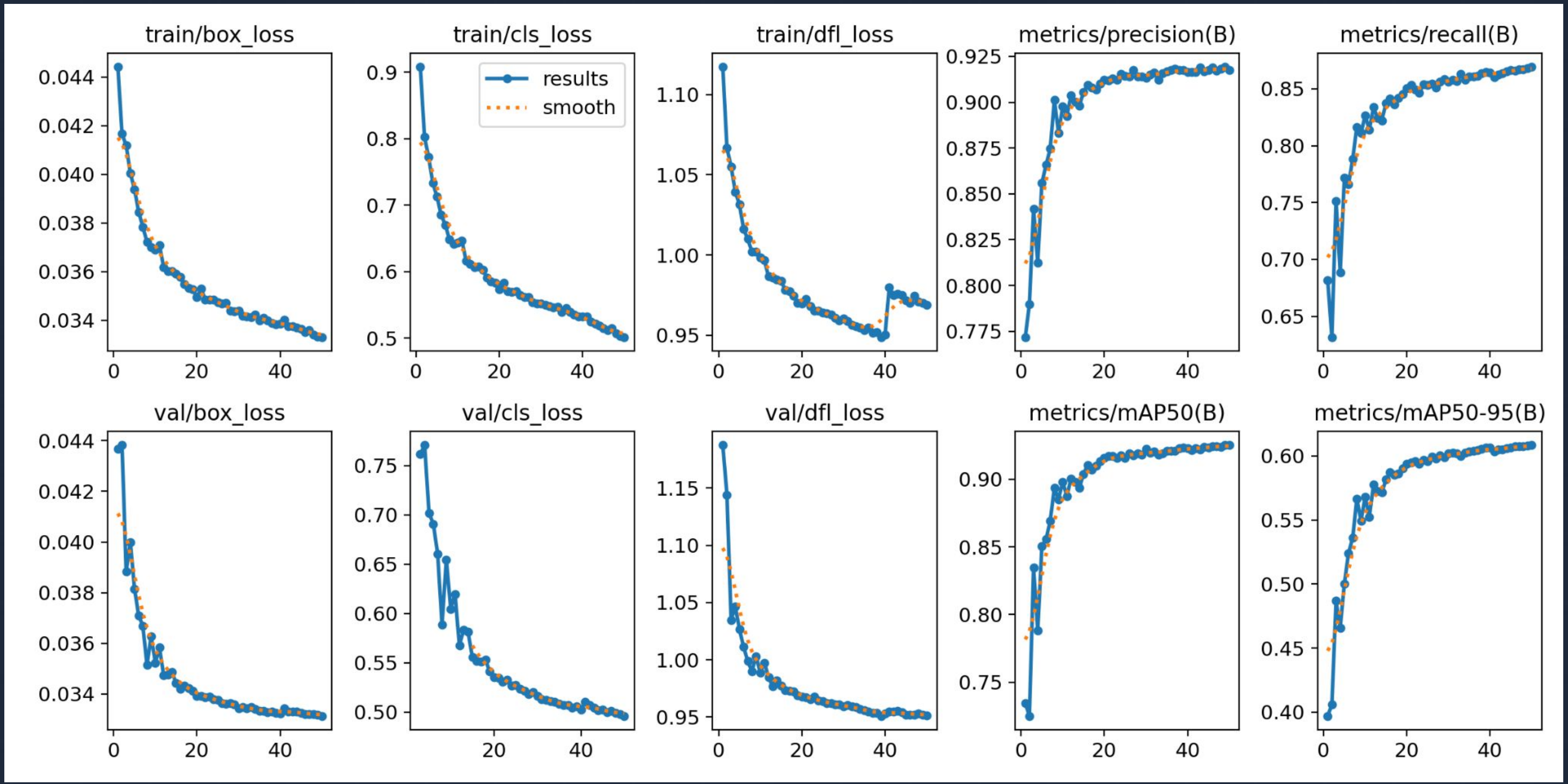
- mAP@50: 0.8918
- mAP@50-95: 0.572
- Precision: 0.8590
- Recall: 0.8478



Final Detection Results (YOLOV11L)

Final parameters list -

- 'lr0': 0.01
- 'lrf': 0.01,
- 'momentum': 0.937
- 'weight_decay': 0.0005
- 'box': 0.2
- 'cls': 0.5
- 'hsv_h': 0.015
- 'mosaic': 1.0
- 'mixup': 0.0



Best model-

Test Results:

Epochs = 50

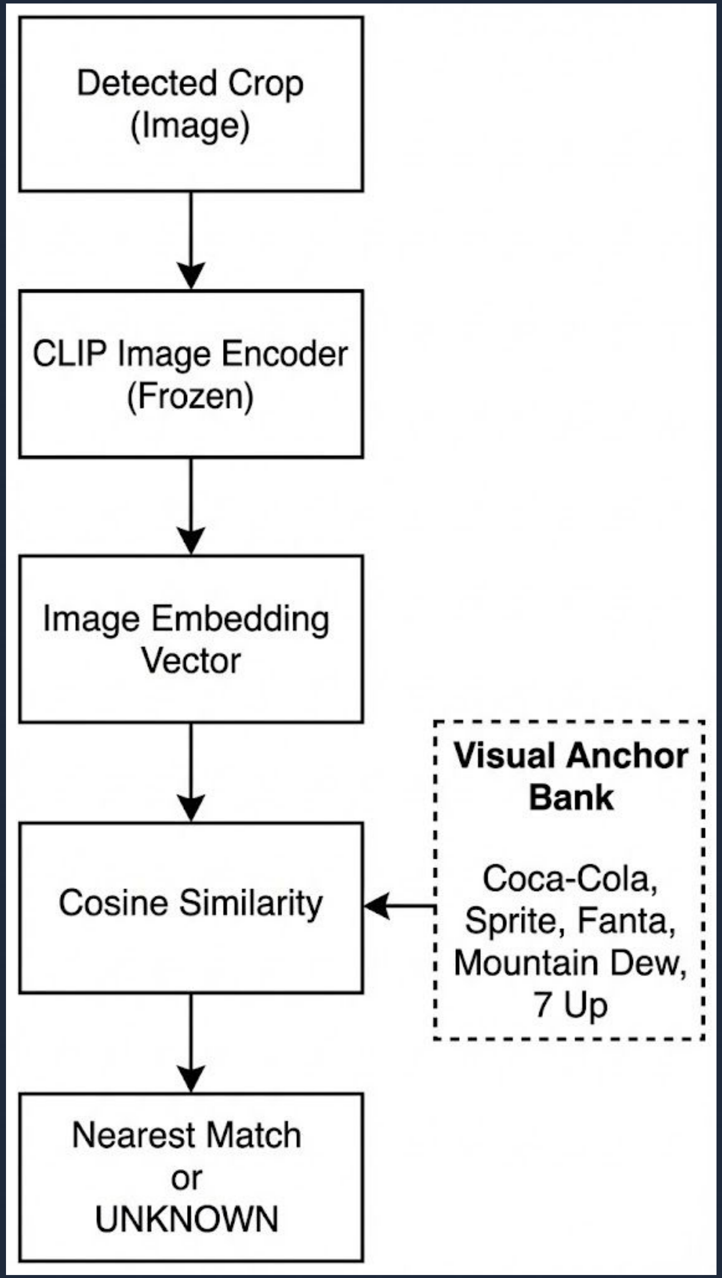
- mAP@50: 0.9395
- mAP@50-95: 0.6191
- Precision: 0.9171
- Recall: 0.8802

Few shot recognition using CLIP

Challenge: Base dataset is class-agnostic. We needed to identify brands

Methodology:

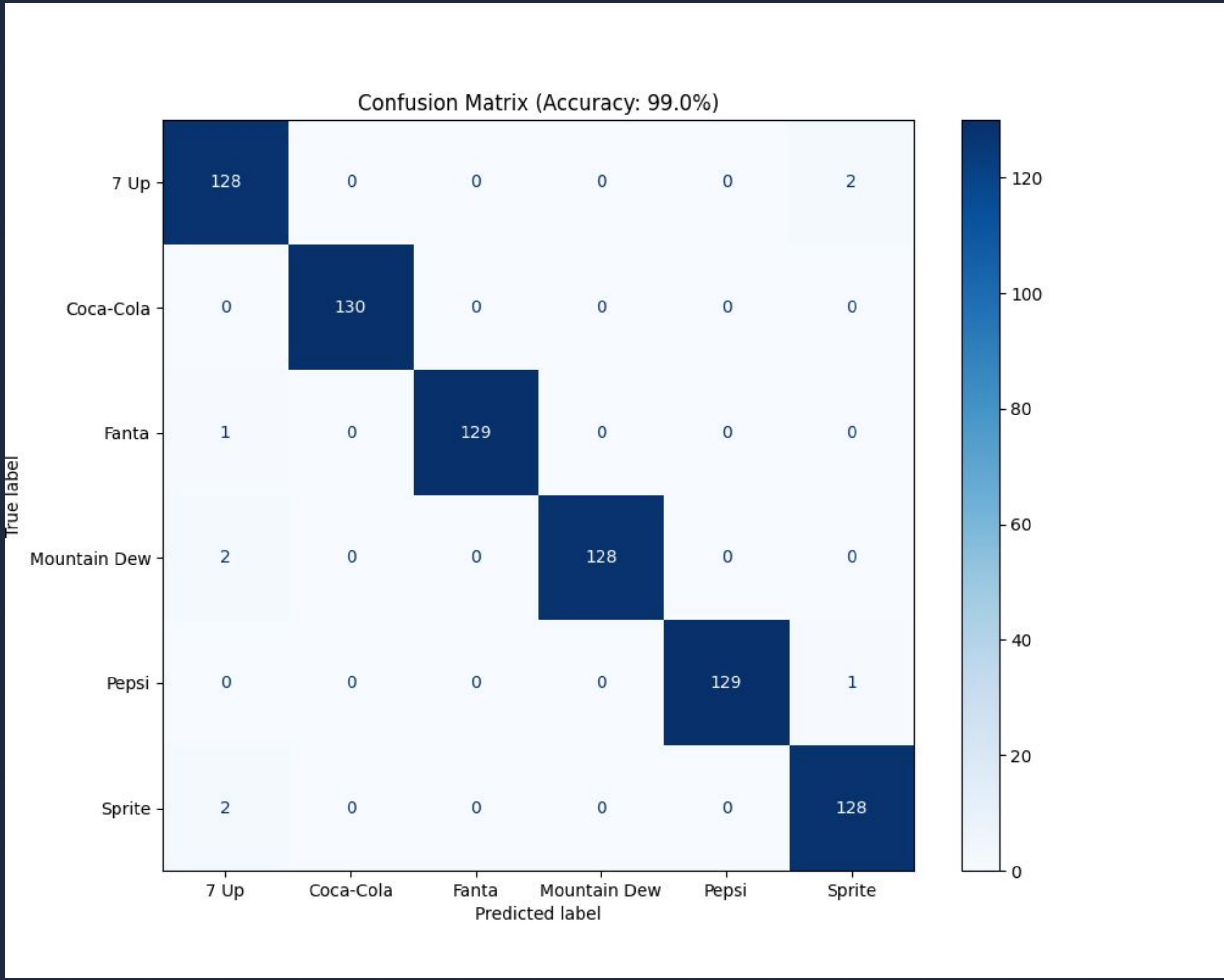
- Few shot recognition via Visual Anchors.
- By comparing shelf crops directly against real product features, we avoid confusion between visually similar brands
- Each detected crop is embedded with frozen CLIP and matched by cosine similarity to a bank of labeled anchor embeddings.
- Low similarity crops are labeled UNKNOWN



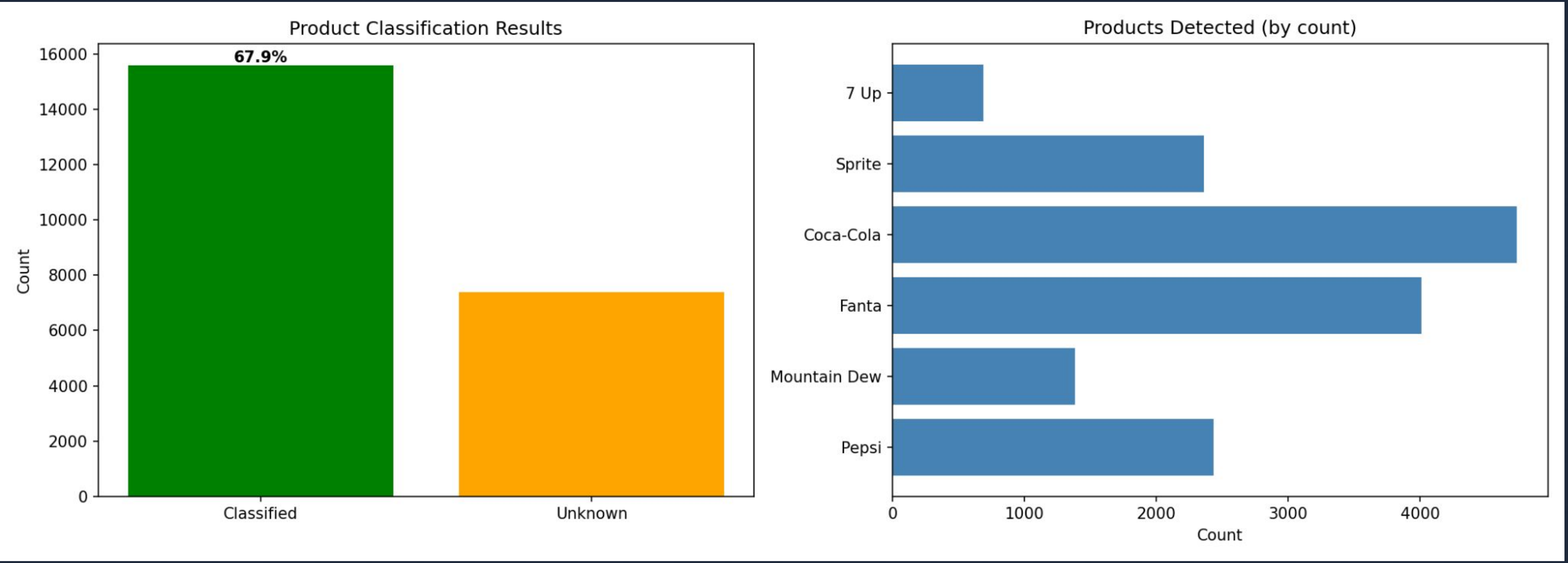
CLIP CLASSIFICATION PERFORMANCE

Average confidence: 80.4 %

Accuracy on test set: 98.97%



Overall pipeline testing results



OUTPUT: Spatial and Semantic Knowledge Graphs

Turning Detections into Relationships

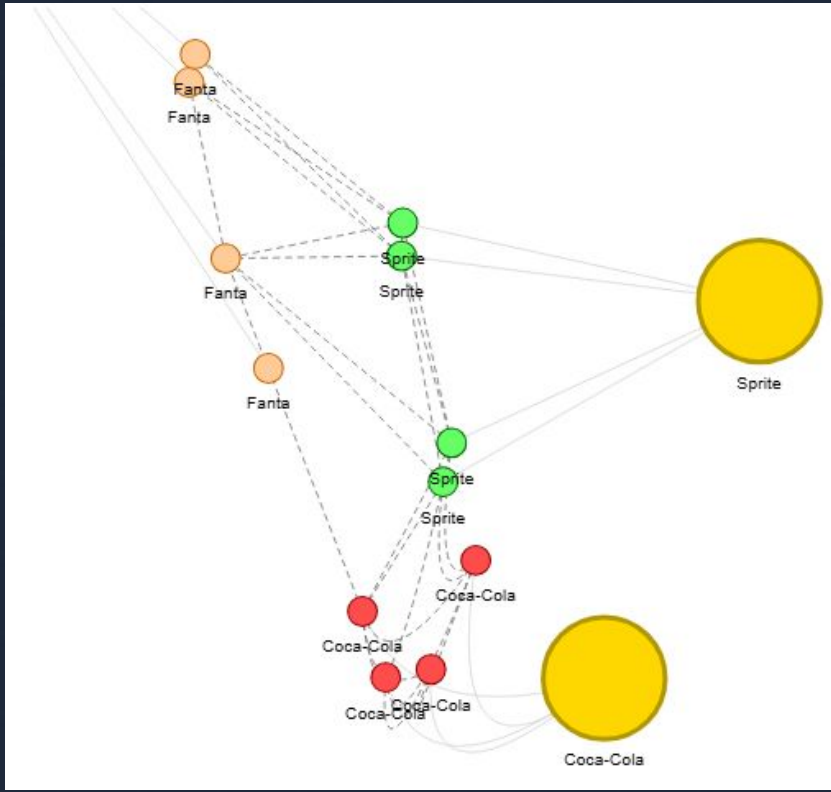
We map raw coordinates to graph using NetworkX

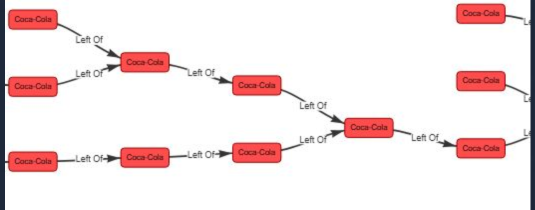
Nodes (Entities)

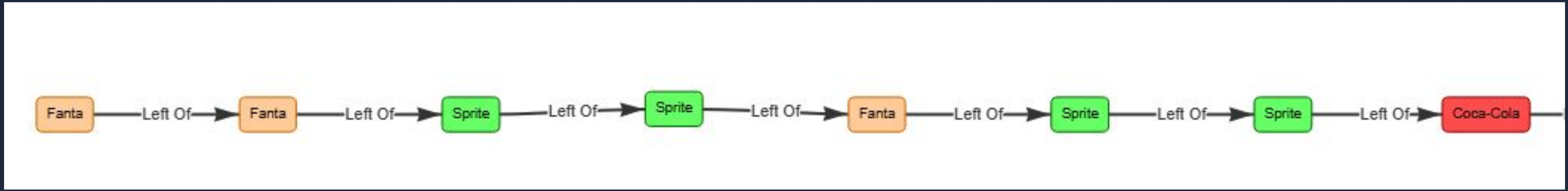
1. Brand Concept (e.g., "Coca-Cola")
2. Physical Item (e.g., "Item_42")

Edges (Relationships)

1. (Item_42) --[is_brand]⟶ (Coke)
2. (Item_42) --[next_to]⟶ (Item_43)
3. (Item_42) --[left_of]⟶ (Item_44)







CONCLUSION & FUTURE WORK

Achievements

Adapted YOLOv11n, YOLOv11l and RT-DETR for dense retail

Data-efficient few-shot recognition using CLIP visual embeddings

Computing Spatial relationships using custom geometry engine

Scene graph generation successful

Possible Enhancements

SKU enhancement: A lot more SKUs can be added

Knowledge graph enhancement: More relationships and IDs can be inferred

Detection model: Finetune Fast R-CNN, RetinaNet and compare detection results

Custom YOLO: Change architecture

Segmentation: Train Mask R-CNN to get higher mAP@90

Adaptation

Integrate unlabelled real-world images to mitigate Domain Shift

THANK YOU