The goal of this project is to develop an instance segmentation model capable of detecting and recognizing Coca-Cola products in images. The model should be able to separate individual products from the background while ensuring accurate segmentation.

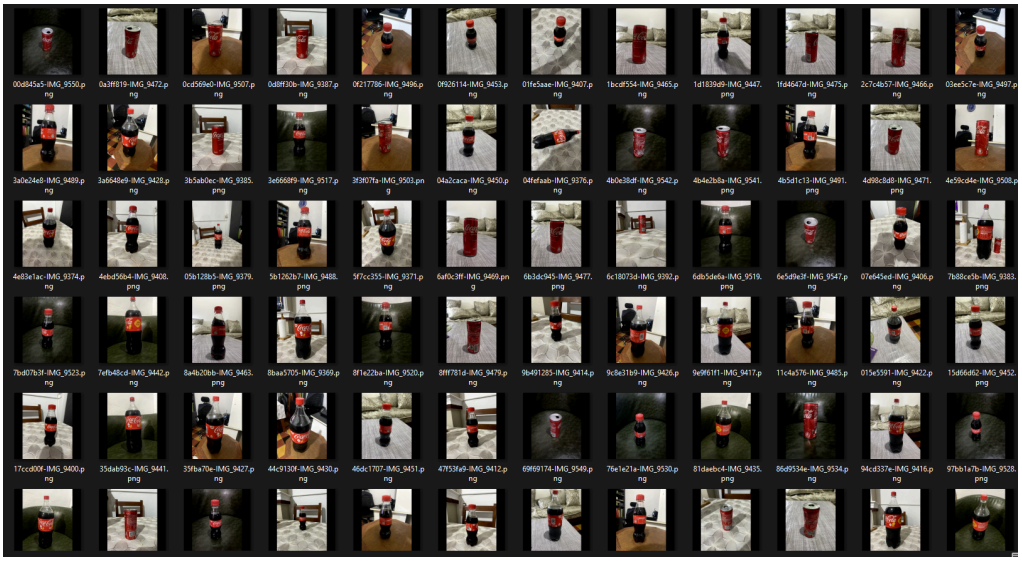
To achieve this, I opted for YOLOv11, a state-of-the-art object detection and segmentation model that provides high-speed and high-accuracy results, and the dataset is processed using various augmentation techniques to enhance the model's robustness.

2. Data Acquisition

Two approaches were considered for obtaining the dataset:

- Publicly available datasets (COCO, Open Images, Kaggle datasets)
- Custom dataset consisting of Coca-Cola product images collected manually

For this project, I chose to create my own dataset instead of using images from the internet. I personally took pictures of Coca-Cola bottles and cans available in Algeria. This decision ensures that the model is specifically trained to recognize Algerian Coca-Cola products, which may have different packaging or branding compared to products from other countries. The dataset contains around **180 images** before augmentation.



Data Labeling

To annotate the dataset, I used **Label Studio**, a widely used open-source annotation tool. The labeling process involved:

Drawing precise segmentation masks around each product

Exporting the labeled data in YOLO-compatible format (.txt and corresponding images)

3. Data Preprocessing & Augmentation

Before training the model, I applied several preprocessing and augmentation techniques to improve performance.

3.1 Preprocessing

Resizing: All images were resized to 640x640 pixels for uniformity.

Normalization: YOLO automatically normalizes the data, so no additional normalization was needed.

3.2 Data Augmentation

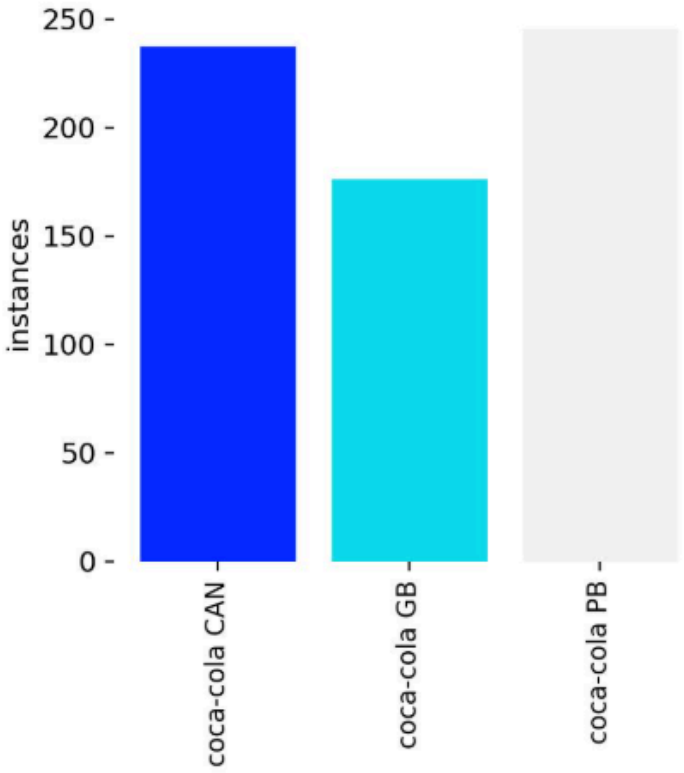
To increase the dataset's diversity and help the model generalize better, the following transformations were applied:

Rotation & Flipping: Random rotation and horizontal flipping to introduce viewpoint variations.

Scaling & Cropping: To simulate zoomed-in and cropped images.

Color Jittering: Adjusting brightness, contrast, and saturation to handle lighting variations.

Gaussian Blur & Noise: Adding slight noise to make the model robust against image distortions.



4. Model Development

Model: YOLOv11 was used for instance segmentation, initialized with pre-trained weights (yolo11m-seg.pt).

Training: The model was trained for 86 NVIDIA RTX 3070 GPU.

```
model.train(
    data='dataset.yaml',
    imgsz=640,          # Image size
    device=0,           # GPU device
    batch=16,           # Batch size
    epochs=100,         # Number of epochs
    workers=0,          # Data loading workers
    optimizer='SGD',    # Optimizer
    lr0=0.01,           # Initial learning rate
    momentum=0.9,       # Momentum
    weight_decay=0.0005, # Weight decay
    patience=50         # Early stopping patience
)
```

5. Results

The model was evaluated on the test set, and the following metrics were obtained:

Class	Images	Instances	Box(P	R	mAP50	mAP50-95)	Mask(P			
all	37	38	0.995	1	0.995	0.98	0.995	1	0.995	0.982
coca-cola CAN	17	17	0.997	1	0.995	0.992	0.997	1	0.995	0.989
coca-cola GB	5	5	0.988	1	0.995	0.995	0.988	1	0.995	0.995
coca-cola PB	16	16	1	1	0.995	0.954	1	1	0.995	0.961

6. Limitations

- The dataset size was relatively small, which may limit the model's ability to generalize to more diverse scenarios.
-
- The model struggles to detect objects in images with a high density of products (e.g., many Coca-Cola bottles or cans in the same picture).
-
- Collecting additional images with dense product arrangements is challenging within the given time frame.

7. Repository Hierarchy

The project repository is organized as follows:

Copy

```
project/
│
├── dataset/
│   ├── images/              # Original images
│   ├── labels/              # YOLO-format labels
│   ├── augmented_images/    # Augmented images
│   ├── augmented_labels/    # Augmented labels
│   └── dataset.yaml         # YAML file for dataset configuration
│
├── runs/
│   ├── segment/             # Training results and model weights
│   └── val/                 # Validation results
│
├── scripts/
│   ├── train.py             # Script for training the model
│   ├── val.py               # Script for validation
│   └── live.py              # Script for live webcam predictions
```

Model path : **runs\segment\train3\weights\best.pt**
Val curve path : **runs\segment\val5**

Results :


130
Coca

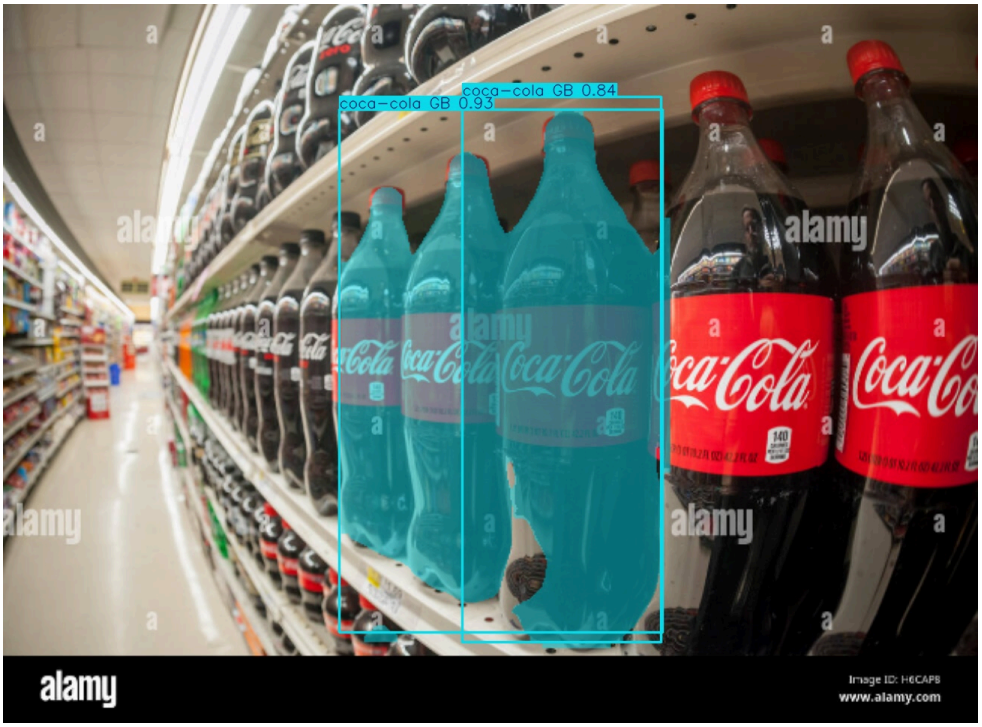

Coca-Cola
Coca-Cola
24 cl


coca-cola PB 0.79
coca-cola CAN 0.89


coca-cola CAN 0.89


coca-cola GB 0.86
coca-cola GB 0.83


coca-cola GB 0.89


coca-cola GB 0.84
coca-cola GB 0.93
alamy
Image ID: H6CAP8
www.alamy.com