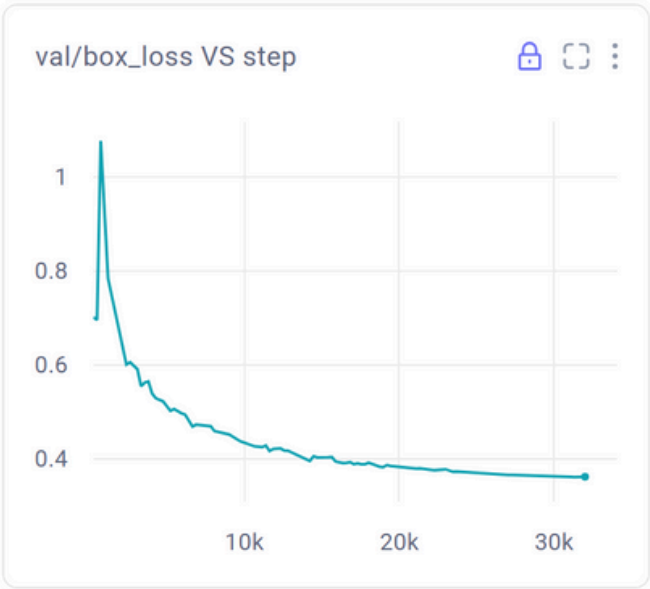


YOLO MODEL

DATA PREPARATION	MODEL DEVELOPMENT	INFERENCE IN DGX SERVER	LIVE VIDEO IN CLIENT
<p>KEEP SEG POINTS IN IMAGE BOUNDARIES</p> <ul style="list-style-type: none">• Ensure all x and y coordinates in the segmentations stay within image boundaries (width and height). <p>KEEP BBOX POINTS (X, Y, W, H) IN IMAGE & SEG BOUNDARIES</p> <ul style="list-style-type: none">• Constrain X and Y coordinates to the image dimensions.• Adjust width W and height H to ensure the bounding box fits within the image.• If segmentation data exists, ensure bounding boxes align with the segmentation boundaries. <p>CONVERT TO YOLO FORMAT</p> <pre>convert_coco(labels_dir, save_dir, use_segments=True, use_keypoints=False, cls91to80=False, lvis=False)</pre> <p>NORMALIZE (0, 1) AND REMOVE DUPLICATES</p>	<p>CREATE YAML FILE AND MANAGE DIRECTORIES</p> <p>EXPERIMENT TRACKING</p> <pre>comet_ml.login()</pre> <p>MODEL TRAINING</p> <pre>model = YOLO("yolo11n-seg.pt") results = model.train(data="dataset.yaml", project = "yolo-project-fin", epochs=200, imgsz=640, batch=32, patience=10, name="run01")</pre>	<p>SETUP INFERENCE SERVER</p> <pre>app = FastAPI()</pre> <p>LOAD MODEL</p> <pre>model = YOLO("models/run01.onnx")</pre> <p>PROCESS UPLOADED FRAME:</p> <ul style="list-style-type: none">• Receive an image frame from the client.• Convert it to a format suitable for the model. <p>RUN INFERENCE:</p> <pre>results = model(frame, conf=0.8) segmented_frame = result[0].plot()</pre> <p>SEND RESULT BACK:</p> <ul style="list-style-type: none">• Encode the segmented image as a JPEG.• Return the processed image to the client.	<p>SET UP WEBCAM</p> <pre>cap = cv2.VideoCapture(0)</pre> <p>SEND FRAME FOR PROCESSING</p> <ul style="list-style-type: none">• Encode each frame as a JPEG.• Send the frame to a remote server for segmentation via an HTTP POST request. <p>RECEIVE AND DISPLAY RESULTS</p> <ul style="list-style-type: none">• Decode the segmented frame returned by the server.• Calculate and overlay FPS (frames per second) on the segmented frame. <p>Can be done with gradio</p>

RESULTS



YOLO11n-seg summary (fused): 265 layers, 2,839,248 parameters, 0 gradients, 10.2 GFLOPs

Class	Images	Instances	Box(P	R	mAP50	mAP50-95)	Mask(P	R	mAP50	mAP50-95):
all	845	1863	0.968	0.935	0.953	0.899	0.968	0.931	0.946	0.866
bottled_soda	37	37	0.989	1	0.995	0.927	0.993	1	0.995	0.916
cheese	40	58	0.98	0.948	0.981	0.901	0.983	0.948	0.981	0.889
Kitkat	42	47	1	0.971	0.98	0.916	1	0.968	0.98	0.903
coffee	30	66	0.984	0.935	0.985	0.91	1	0.948	0.964	0.903
condensed_milk	37	52	0.968	1	0.989	0.963	0.97	1	0.989	0.952
Cooking_oil	40	40	0.993	1	0.995	0.955	0.995	1	0.995	0.932
corned_beef	41	247	0.972	0.986	0.987	0.935	0.972	0.978	0.987	0.843
garlic	33	33	0.968	0.939	0.985	0.905	0.976	0.939	0.985	0.888
instant_noodles	30	30	0.988	1	0.995	0.981	0.99	1	0.995	0.989
ketchup	35	35	0.99	1	0.995	0.995	0.991	1	0.995	0.991
lemon	37	46	0.982	1	0.995	0.982	0.985	1	0.995	0.988
Nestle All Purpose Cream	35	63	0.989	0.984	0.993	0.961	0.991	0.984	0.993	0.942
Mayonnaise	31	105	0.943	0.941	0.988	0.9	0.923	0.917	0.963	0.745
peanut_butter	26	158	0.908	0.519	0.609	0.577	0.891	0.506	0.581	0.511
pasta	37	39	0.972	0.974	0.983	0.913	0.978	0.974	0.983	0.922
pineapple_juice	35	116	0.941	0.948	0.972	0.942	0.946	0.948	0.972	0.921
crackers	32	48	0.922	0.979	0.967	0.929	0.929	0.979	0.967	0.937
sardines	45	61	0.978	0.967	0.974	0.91	0.98	0.967	0.986	0.88
pink_shampoo	36	51	0.993	1	0.995	0.993	0.994	1	0.995	0.99
soap	34	34	0.989	1	0.995	0.989	0.991	1	0.995	0.987
silverswan	34	95	0.907	0.821	0.911	0.743	0.868	0.763	0.855	0.581
toothpaste	25	42	0.991	1	0.995	0.944	0.993	1	0.995	0.925
canned_tuna	42	327	0.885	0.52	0.61	0.527	0.9	0.511	0.559	0.416
GreenCross_Ethyl_Alcohol	31	33	0.997	1	0.995	0.885	0.999	1	0.995	0.823

Speed: 0.1ms preprocess, 0.5ms inference, 0.0ms loss, 1.2ms postprocess per image

RESULTS

Confusion Matrix Normalized

