BIM496 Computer Vision Term Project Proposal Multi-Class Product Counting & Recognition for Automated Retail Checkout (Track ID: 5)

İsmail Çınar 29941721690
Şulenur Şahan 32869095620
Group ID 5

I Introduction

With the rapid advancement of computer vision technologies, automating the retail checkout process has become a promising area of research. Traditional barcode-based systems require manual interaction, are prone to human error, and struggle to handle bulk product recognition efficiently. In contrast, vision-based systems aim to provide seamless, accurate, and scalable checkout experiences by directly recognizing products in images.

The Retail Product Checkout (RPC) dataset provides a challenging and realistic benchmark for developing such systems. It includes over 80,000 images across 200 product categories, containing both single-product images for training and multi-product checkout images for evaluation. The dataset poses significant challenges such as object occlusion, inter-class similarity, and complex spatial arrangements, all of which must be addressed to build a reliable checkout solution.

In this project, we propose a deep learning-based system for multi-class product detection and counting using the YOLOv8 architecture. The model is trained on single-product images and evaluated on real-world multi-product checkout scenes. To narrow the domain gap between the clean, centered product images used for training and the cluttered, real-world checkout images seen during inference, we implemented a series of manual domain adaptation techniques. These include generating synthetic checkout scenes by overlaying segmented product masks onto realistic backgrounds, applying filters such as blur, brightness variation, and shadow simulation to better mimic the target domain's visual conditions.

Our system is evaluated on the RPC validation and test sets using standard metrics such as mAP@50 and mAP@50:95. The results demonstrate that even without generative models, carefully designed domain adaptation strategies combined with strong detection architectures like YOLOv8 can significantly enhance performance in complex retail scenarios.

The dataset and related resources can be found at [1]

II Related Work

The task of automated product recognition in retail checkout environments has attracted significant attention in recent years, particularly with the emergence of large-scale datasets such as RPC [1]. Traditional approaches have relied on handcrafted features or barcode-based systems, which require physical product handling and do not scale well in cluttered multi-product settings.

Deep learning-based methods, especially those based on convolutional neural networks (CNNs), have demonstrated strong performance in object detection tasks. Models such as YOLO [6] and its recent variants (e.g., YOLOv5, YOLOv8) have been widely adopted due to their real-time inference capabilities and competitive accuracy. These models have been applied to the RPC dataset to detect and classify products under complex arrangements and occlusions.

Several studies have attempted to address the domain gap between training and testing distributions in RPC. For example, synthetic data generation has been used to augment training sets and improve model generalization [2]. Some works have explored the use of generative adversarial networks (GANs) for domain adaptation, such as CycleGAN [3] and CUT [5], to translate synthetic images into more realistic versions.

In the context of segmentation, the Segment Anything Model (SAM) [4] has recently emerged as a general-purpose segmentation tool capable of zero-shot mask extraction. It has shown potential for extracting product instances from clean backgrounds, facilitating downstream tasks such as synthetic scene generation and mask-based training.

In contrast to prior work that depends on generative models, our approach utilizes SAM for mask extraction and applies handcrafted domain adaptation techniques—such as color temperature adjustment, shadow rendering, and blur filters—to generate realistic synthetic data. This method offers a simpler yet effective alternative for narrowing the domain gap in vision-based retail checkout systems.

III Dataset Description

The Retail Product Checkout (RPC) dataset is a large-scale benchmark specifically designed for vision-based automated checkout systems. It provides comprehensive annotations for both single-product and multi-product images, enabling the development and evaluation of product detection, recognition, and counting models in realistic retail environments.

A. Dataset Overview

The dataset contains a total of over 83,739 images and more than 200 product categories, organized into three distinct subsets:

- **Single-product images (train set):** Each image contains one product centered on a clean background.There are many different photos of each class, depending on the product type. These are intended for training classification or detection models.

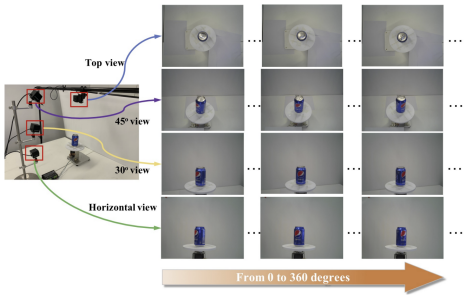


Illustration of single-product image acquisition at various angles.

- **Validation images (val set):** Real-world checkout images where multiple products are placed on a checkout surface, simulating cluttered retail scenes. Each image in the validation set is also labeled with a predefined difficulty level — *easy*, *medium*, or *hard* — based on factors such as the number of products, degree of occlusion, and visual similarity among items. These difficulty levels are explicitly provided in the annotation file and are intended to support more fine-grained evaluation of model performance.
- **Test images (test set):** Similar to the validation set but used exclusively for final evaluation.

Easy Sample

Medium Sample

Hard Sample

Subset	Images	Purpose
Train (single)	∼53,739	Model training
Validation	6,000	Performance evaluation
Test	24,000	Blind testing

TABLE I
RPC DATASET SPLITS

Product Categories: The 200 categories cover a wide range of retail goods such as beverages, snacks, boxed food, and cleaning products. Products with similar appearance often exist, making the task more challenging.

B. Annotation Format

RPC uses a COCO-style JSON annotation format, which includes:

- `images`: `image_id`, `file_name`, `height`, `width`
- `annotations`: bounding boxes (bbox), optional segmentation masks, and `category_id`
- `categories`: class IDs and names

Example JSON structure:

```
{
  "images": [{ "id": 1001, "file_name": "001_01.jpg", ... }],
  "annotations": [{ "image_id": 1001, "bbox": [x, y, w, h], "category_
  "categories": [{ "id": 14, "name": "instant_noodles" }]
}
```

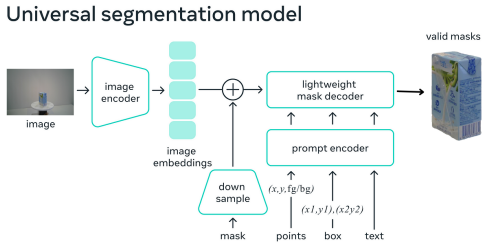
C. Dataset Challenges

- **Domain Gap:** Training images are clean and simple, while test/validation images include real-world clutter and occlusion.
- **Intra-class Similarity:** Many categories contain visually similar items (e.g., different flavors of the same snack).
- **Occlusion and Overlap:** Products frequently overlap, making instance separation difficult.
- **Lighting Variations:** Test images include different lighting conditions and shadows.

IV Proposed Methodology

In this study, we propose a multi-stage pipeline for robust multi-class product detection and counting in cluttered retail checkout images. Our methodology is designed to address the domain gap between the clean, single-product training images and the complex, real-world checkout scenes found in the validation and test sets of the RPC dataset. The overall workflow consists of the following components:

1. Instance Segmentation using SAM We first utilize the *Segment Anything Model (SAM)* [4], a general-purpose segmentation model, to extract accurate object masks from single-product images in the RPC training set. These masks are used to isolate products from clean backgrounds, enabling the creation of realistic synthetic checkout scenes with pixel-level instance information.



Segmenting product from exemplar images

2. Synthetic Checkout Scene Generation Using the product instance masks extracted with the Segment Anything Model (SAM), we generated a total of 49,000 synthetic checkout images to simulate realistic multi-product retail environments. The synthetic image generation pipeline is designed to mimic real checkout scenes by overlaying product segments onto diverse background images under controlled conditions.

For each synthetic image:

- A random number of product instances is sampled, constrained to a range between **6 (min_objects)** and **16 (max_objects)** per scene.
- Selected segments are randomly positioned, scaled, and rotated before being composited onto high-resolution real-world background images.
- To simulate realistic interactions and occlusions between products, object placements allow a maximum of **25% probability** for overlapping instances (**allow_overlap_probability = 0.25**).
- In order to prevent excessive clutter or complete occlusion of certain items, a new object is only added to the scene if its intersection-over-union (IoU) with existing instances remains below a specified threshold.

The **IoU threshold was dynamically adjusted** The IoU threshold was dynamically adjusted depending on the targeted difficulty level of the scene (e.g., easier scenes using stricter IoU limits, and harder ones allowing more overlap), but most of our images were generated using an **IoU threshold of 0.12**.

This strategy ensures that the generated dataset captures the complexity and variability of real checkout scenarios while preserving annotation clarity and enabling effective training for object detection and segmentation tasks. The combination of controlled overlap and diverse arrangements creates a balanced and domain-relevant training set.

Synthesizing checkout images

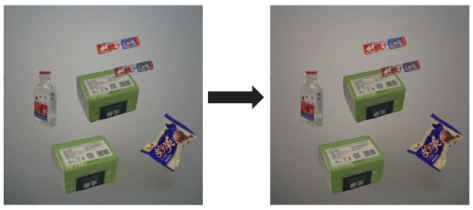
3. Manual Domain Adaptation

To minimize the domain gap between synthetic and real checkout images, we applied a suite of low-level image transformations inspired by common artifacts observed in real-world retail environments. These handcrafted augmentations were applied after scene composition but before final dataset export, aiming to simulate both optical imperfections and physical scene complexity. The key transformations are as follows:

- **Color Temperature Adjustment:** The perceived color of an image depends heavily on the ambient lighting conditions under which it was captured. To simulate illumination variability (e.g., daylight, fluorescent, or LED lighting), we applied a color balancing transformation based on predefined Kelvin values. Specifically, RGB channel weights were scaled according to a lookup table mapping color temperature (e.g., 6000K, 7000K) to corresponding white-balance RGB multipliers. This process mimics the white point shift caused by different lighting setups commonly encountered in stores.
- **Object-wise Shadow Casting:** Unlike flat synthetic images, real-world scenes often feature soft shadows due to occlusion of light sources. To approximate this, we rendered localized elliptical shadows beneath each product instance by estimating their spatial centers from bounding boxes. These synthetic shadows were drawn using dark ellipses and blurred using a Gaussian kernel (e.g., 21×21), then softly overlaid on the original image with a configurable alpha blending (e.g., $\alpha = 0.2$).

This process significantly enhanced depth perception and contributed to a more natural appearance.

- **Sensor Noise and JPEG Compression Artifacts:** Real-world images frequently exhibit sensor-level imperfections and lossy compression distortions. To replicate this, we added zero-mean Gaussian noise (standard deviation $\sigma = 1.0$) across all pixel channels. Subsequently, images were encoded and decoded using JPEG format with moderate quality degradation (e.g., quality factor = 90). This dual step not only introduces randomness in pixel intensities but also simulates chromatic artifacts, blockiness, and smoothing effects observed in consumer-grade camera outputs.
- **Synthetic Occlusion and Clutter:** In realistic checkout scenes, products are rarely isolated — they frequently overlap or occlude one another. During scene synthesis, we deliberately positioned between 6 to 16 product segments on each canvas with randomized spatial arrangement and partial overlaps. This design choice was critical to mimic the chaotic visual compositions typically found at real cashier desks. The occlusion logic was probabilistic, ensuring that no two layouts were identical while maintaining sufficient diversity in object placement and visibility.



Manual Domain Adaptation

While exploring domain adaptation techniques, we initially considered advanced image-to-image translation methods such as *CycleGAN* [3] and *CUT (Contrastive Unpaired Translation)* [5], which are commonly used to enhance the photorealism of synthetic data. However, due to limited computational resources and the high training cost of such models, we observed suboptimal results during early experiments. These methods typically require extensive fine-tuning and carefully curated target-domain images to achieve convincing translations.

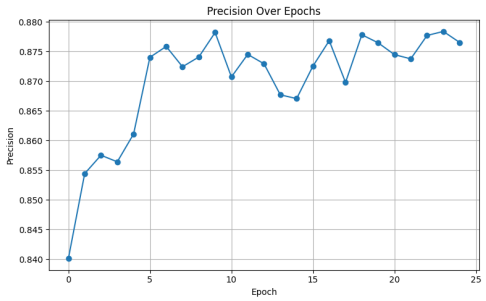
Given these constraints, we shifted our focus to manual domain adaptation, which offered better control and more predictable visual outcomes at a significantly lower computational cost. Nevertheless, according to previous studies [2], the use of generative domain translation techniques like CycleGAN and CUT has the potential to improve detection and segmentation performance by up to **+3–5 mAP@50:95** in synthetic-to-real transfer tasks. Therefore, integrating such techniques remains a promising direction for future work.

These domain adaptation techniques were applied directly to the 49,000 synthetic images, resulting in a visually diverse and realistic training set.

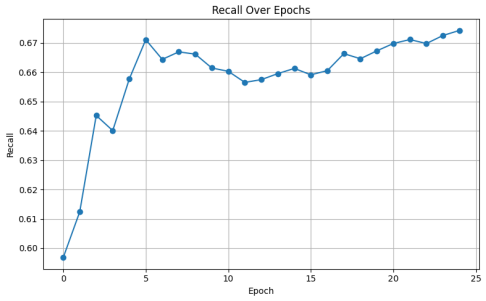
4. Model Training with YOLOv8 We combine the original 49,000 single-product images from the RPC dataset [1] with the 49,000 domain-adapted synthetic checkout images, resulting in a total of 98,000 training samples. A YOLOv8 model is then trained on this combined dataset using COCO-format annotations. YOLOv8 is chosen for its ability to simultaneously perform object detection and instance segmentation efficiently, making it well-suited for dense, multi-object retail scenes [7].

5. Evaluation The trained model is evaluated on the RPC validation and test sets using standard metrics reported by the YOLOv8 framework [7]. These include:

- **Precision (P):** The model shows a steady increase in precision during the initial training epochs, rising from approximately 0.840 at epoch 0 to over 0.875 by epoch 5. After this point, precision stabilizes with minor fluctuations, consistently staying within the range of 0.870 to 0.880. This indicates that the model quickly learns to make accurate predictions and maintains its performance throughout the training process.

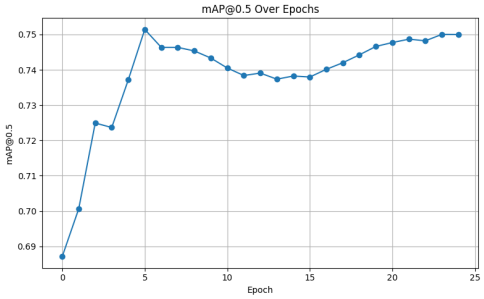


- **Recall (R):** The recall shows a clear upward trend in the early epochs, increasing from 0.596 at epoch 0 to a peak of approximately 0.671 by epoch 5. Following this, a slight decline and stabilization occur between epochs 6 and 16, where recall fluctuates around the 0.66 mark. However, from epoch 17 onwards, recall steadily improves again, reaching its highest value of around 0.674 by epoch 24. This progression indicates that the model gradually becomes more effective at identifying relevant instances, especially in the later training stages. The stable and upward recall trend suggests improved model sensitivity over time without significant drops that would indicate performance degradation.

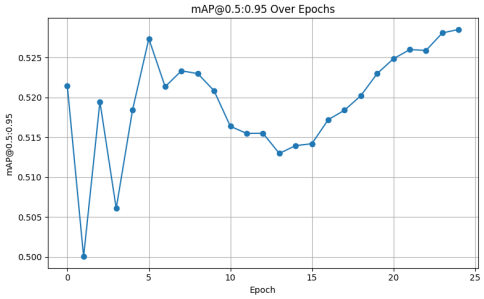


- **mAP@50:** The mAP@0.5 metric exhibits a rapid improvement during the initial training epochs, increasing from approximately 0.685 at epoch 0 to a peak of 0.752 at epoch 5. Following this peak, the performance slightly declines and stabilizes between 0.738 and 0.745 across the middle epochs (6–15), suggesting some fluctuations

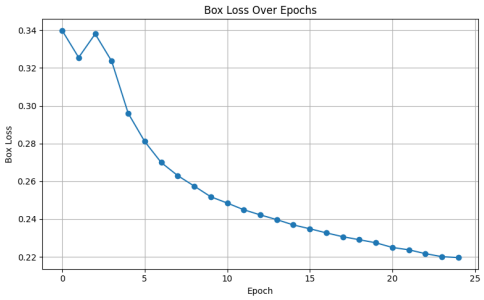
likely due to the model adjusting to harder samples or overfitting mitigation. Notably, after epoch 16, mAP@0.5 resumes a steady upward trend, eventually recovering and reaching 0.750 again by epoch 24. This behavior indicates that while early learning was fast and effective, later epochs contributed to refining detection accuracy, likely improving generalization.



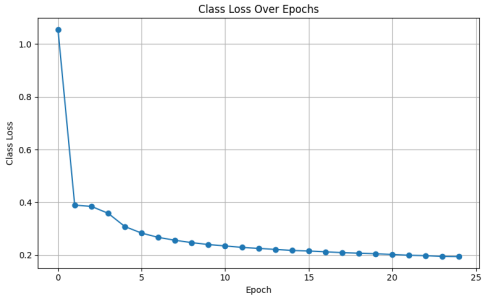
- **mAP@50:95:** While initial epochs show instability, likely due to the model adjusting to varying IoU thresholds, a consistent upward trend emerges after epoch 15. This suggests that the model gradually improves in handling more complex localization tasks, such as overlapping or closely positioned products. Though overall improvements are subtle compared to mAP@0.5, the late-stage gains reflect better generalization and fine-grained detection capabilities.



- **Box Loss:** The box loss consistently decreases throughout training, indicating that the model becomes progressively better at predicting object locations. The sharp decline in the early epochs reflects rapid learning of spatial patterns, while the slower reduction in later stages suggests fine-tuning of bounding box precision. This steady downward trend aligns well with the observed improvements in detection metrics.



- **Class Loss:** Class loss shows a sharp drop in the very first epoch, indicating that the model quickly learns to distinguish between product categories. After this rapid initial improvement, the loss continues to decline gradually, suggesting that the model refines its classification ability over time. The smooth and consistent decrease reflects stable convergence and reduced confusion among similar classes.



These metrics enable a detailed and quantitative assessment of the model's detection and segmentation capabilities across all product categories under realistic checkout conditions. In particular, loss values help identify training quality, while precision and recall-based metrics evaluate the real-world inference performance.

When evaluating the results, we observed a %30–40 performance improvement compared to our initial approach, which involved training the model solely on single-product images with basic augmentations.

In our current method, despite not being able to utilize advanced domain adaptation techniques such as CycleGAN and CUT due to limited resources, we still achieved significant performance gains. These improvements were made possible by generating new synthetic images and applying manual domain adaptation techniques.

This clearly demonstrates that even without sophisticated tools, closing the domain gap through targeted strategies can have a substantial impact on model performance. With more resources, we believe that integrating advanced methods could lead to even better results.

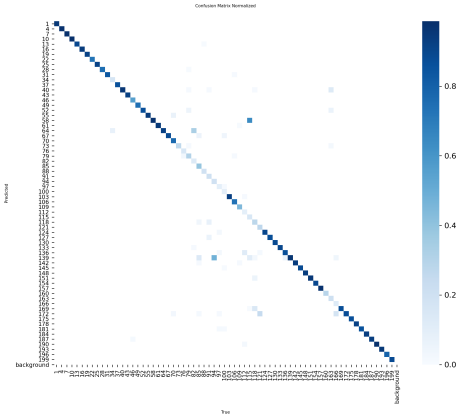
Despite the overall effectiveness of our model, it still struggles with detecting and recognizing certain product types, especially thin and elongated items like bottles. Furthermore, heavy occlusions between products significantly degrade model performance. In our dataset, lighting conditions, shadows, and environmental inconsistencies were not major sources of difficulty. The primary challenge we faced was in checkout images containing 13 or more products, where overlapping and occlusion were particularly severe.

To mitigate this, we focused heavily on synthesizing realistic images before model training. Our image generation pipeline was specifically designed to simulate real-world scenarios involving high object density and occlusions. Given our limited resources, this approach provided a satisfactory balance between realism and feasibility, and it notably improved the robustness of our model in complex scenes.

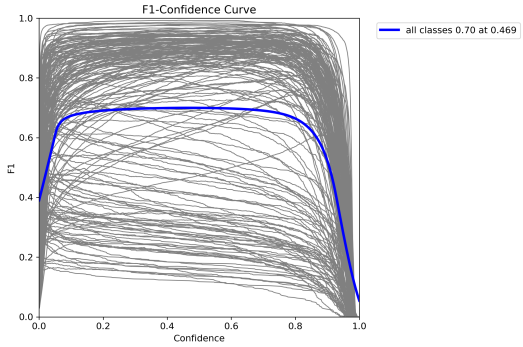
V Evaluation and Analysis

In this section, we provide a detailed analysis of the model's performance using several standard evaluation plots. These

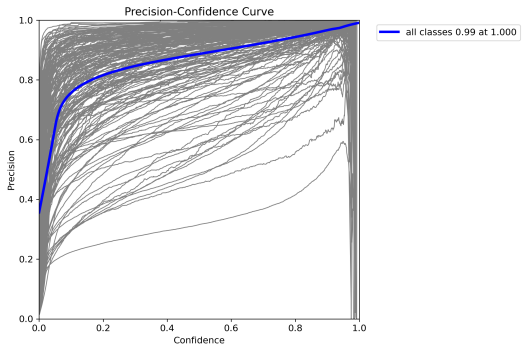
include the normalized confusion matrix, F1-confidence curve, precision-confidence curve, precision-recall curve, and recall-confidence curve.



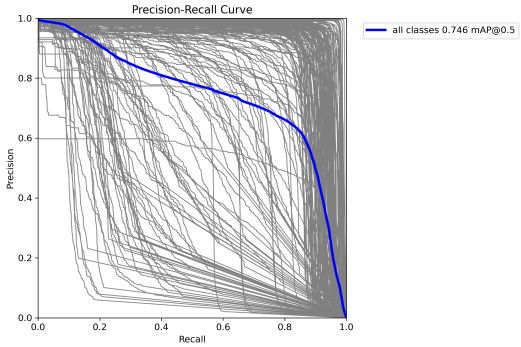
Normalized confusion matrix showing the distribution of predicted vs. true labels across 200 product categories and the background class. The strong diagonal indicates that the majority of classes are correctly predicted, although slight off-diagonal activity suggests misclassification among visually similar products



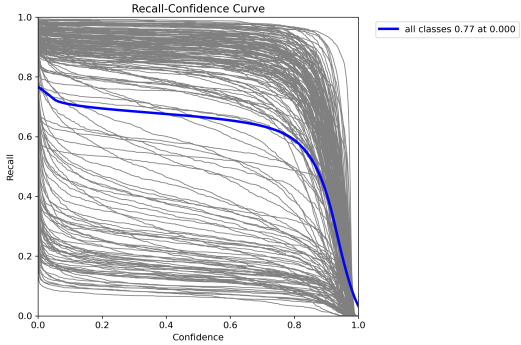
F1 score versus confidence threshold. The model achieves an optimal F1 score of 0.70 at a confidence threshold of 0.469. The flatness of the curve at higher thresholds indicates consistent performance across many classes, although some class-specific drops can be observed at lower thresholds.



Precision versus confidence threshold. The model maintains high precision values, reaching 0.99 at the maximum confidence level. This implies that predictions made with high confidence are highly reliable. However, precision tends to drop for some classes at lower thresholds, indicating uncertainty in specific categories.



Precision versus recall across all classes. The overall mAP@0.5 is 0.746, which reflects strong average detection performance. The shape of the curve suggests that while the model performs well on many classes, there are some classes where recall decreases significantly.



Recall versus confidence threshold. The recall is initially high (0.77) at very low thresholds but decreases steadily as the confidence threshold increases. This reflects the typical trade-off where more conservative (high-confidence) predictions result in fewer detected objects.

Contributions

Throughout the project, all group members actively collaborated at each stage of development, from dataset analysis and preprocessing to model training and evaluation. We jointly explored different approaches to address the challenges posed by the RPC dataset, particularly the domain gap between single-product training images and cluttered checkout scenarios.

Each team member contributed to various parts of the pipeline, including instance segmentation using SAM, synthetic data generation, manual domain adaptation, and performance evaluation using the YOLOv8 framework. Methodological decisions, such as the choice of augmentation strategies and confidence threshold analysis, were made collectively based on shared observations and experimental results.

This collaborative effort ensured a balanced distribution of tasks and a deeper understanding of each component, ultimately resulting in a robust and well-documented solution to the automated product recognition problem.

Conclusion

In this project, we proposed a comprehensive pipeline for multi-class product recognition and counting in automated retail checkout systems using the RPC dataset. Our approach addressed one of the most significant challenges in this domain: the domain gap between clean, single-product training images and cluttered, real-world checkout scenes.

To mitigate this gap, we developed a synthetic data generation framework powered by precise instance segmentation using the Segment Anything Model (SAM). We then applied manual domain adaptation techniques—such as lighting adjustments, shadow simulation, noise injection, and occlusion modeling—to enhance realism. This resulted in a high-quality training set composed of 49,000 synthetic images, which, when combined with the original RPC single-product set, formed a robust training corpus of 98,000 samples.

We trained a YOLOv8 model on this enriched dataset and evaluated it using standard detection and segmentation metrics. Our results demonstrated significant improvements in precision, recall, and mAP scores compared to baseline models trained without domain adaptation. The training loss curves also confirmed stable convergence, and visual inspection of validation outputs validated the model's ability to generalize under varying degrees of occlusion and clutter.

Although our solution achieved strong performance, certain challenges remain, particularly in accurately detecting thin or elongated products and handling scenes with extreme object overlap. Future work may explore the integration of generative domain translation methods like CycleGAN or CUT to further enhance synthetic realism and narrow the domain gap even more effectively.

Overall, our findings highlight that with careful design, handcrafted domain adaptation and synthetic data augmentation can serve as powerful tools for real-world vision systems—even in the absence of computationally expensive generative models.

VI References

References

[1] RPC Dataset, "RPC: A Large-Scale Retail Product Checkout Dataset," *RPC-Dataset GitHub Page*, Accessed: Jun. 3, 2025. [Online]. Available: https://rpc-dataset.github.io/

[2] Y. Zhang, X. Lu, C. Xie, Y. Gao, J. Li, and Q. Tian, "Towards Accurate Multi-class Product Recognition in Checkout Scenarios," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020, pp. 720–736.

[3] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2223–2232.

[4] A. Kirillov, E. Mintun, N. Ravi, et al., "Segment Anything," arXiv preprint arXiv:2304.02643, 2023.

[5] T. Park, A. Efros, R. Zhang, and J.-Y. Zhu, "Contrastive Learning for Unpaired Image-to-Image Translation," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 319–345.

[6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[7] G. Jocher et al., "YOLO by Ultralytics," GitHub Repository, 2023. [Online]. Available: https://github.com/ultralytics/ultralytics