RetailEye: Supervised Contrastive Learning with Compliance Matching for Retail Shelf Monitoring

Mamoun Alghaslan[1], Khaled Almutairy[1], El-Sayed M. El-Alfy[1,2,3,*], and Abdul-Jabbar Siddiqui[2,4,3]

[1] Information and Computer Science Department
[2] Computer Engineering Department
[3] Interdisciplinary Research Center of Intelligent Secure Systems
[4] SDAIA-KFUPM Joint Research Center for Artificial Intelligence
King Fahd University of Petroleum and Minerals, Saudi Arabia
∗ Corresponding author: alfy@kfupm.edu.sa

Abstract. By harnessing technological advancements in computer vision and artificial intelligence, retail entrepreneurs can not only meet their objectives but also position themselves for sustainable growth in a competitive marketplace. A critical area of focus is inventory management, particularly the monitoring of grocery products on shelves and the identification of misplaced or out-of-stock items. However, automatically detecting and recognizing products in real-time retail environments presents significant challenges, including several challenging factors such as varied visual representations, unpredictable poses, partial or full occlusions, and variations of lighting reflections on glossy packaging, and a lack of unified resources. In this paper, we propose and evaluate a two-stage approach, termed RetailEye, which employs supervised contrastive learning with compliance matching and leverages the latest developments in deep learning. After evaluating different models for object detection and recognition, we designed our system based on YOLOv8s in the first stage and EfficientNetV2-S and ResNet18 in the second stage. The proposed model outperformed the one-stage approach with high detection and recognition accuracies. Additionally, we unveil a custom dataset specifically curated for this research, aimed at advancing the field of inventory management.

Keywords: Contrastive Learning · Video Analytics · Deep Learning · Product Detection and Recognition · Retail Management · Shelf Monitoring

1 Introduction

In retail stores, maintaining optimal inventory levels and ensuring accurate product placement on store shelves are critical aspects that directly impact customer satisfaction and overall business efficiency. The traditional methods of monitoring stock levels and repositioning misplaced items involve regular manual shelf inspection by retail workers, which is time-consuming, prone to human errors, labor-intensive, inflexible, and costly. Subsequently, they may lead to non-optimized operations and revenue losses.

The main objectives of entrepreneurs in the retail industry are to boost profit margins and attract a larger customer base [6]. Embracing recent technological advancements has enhanced the customer experience while also reducing operational costs. Real-time inventory management has become increasingly feasible through computer vision technology, which is largely achieved through AI-powered distributed monitoring systems that automatically track shelf conditions. These systems can effectively detect out-of-stock situations and assess product organization on racks in accordance with store's layout, commonly referred to as planogram. Moreover, these advanced systems can provide valuable analytics that can help retailers reduce out-of-stock rates, enhance customer satisfaction, streamline operations and optimize business performance, and ultimately increase profitability.

There are two primary techniques for vision-based object detection and recognition, each with its own set of strengths and limitations. The traditional methods involve extracting manual features of the image based on domain knowledge, such as color histograms or texture patterns to classify grocery products. While this approach can be interpretable and computationally efficient, it does not generalize well due to the highly dynamic nature of product images. Manually extracted features struggle to adapt to the common challenges faced in real environments. On the other hand, deep learning models, specifically Convolutional Neural Networks (CNNs) and transformers, have recently demonstrated remarkable success in object detection and recognition. Deep learning can automatically learn representations and provide end-to-end solutions with different levels of abstraction, which is more efficient compared to manual feature engineering. Although deep learning methods are widely used nowadays, the performance of these methods is influenced by two factors. First, their effectiveness diminishes when working with small training datasets. Second, the previously learned knowledge from prior classes or tasks tends to be lost when the model is trained on newly introduced classes or tasks. Since supermarkets have numerous products that the model was trained on, retraining the model from scratch every time a new product appears in the market is impractical [5].

Even though deep learning methods outperformed traditional detection and recognition techniques, their performance is greatly influenced by the amount of training data available. Building a dataset for product recognition is an expensive task. The dataset should contain products captured from various perspectives and under different lighting conditions. It should also handle scenarios in which the products might be deformed or occluded by other items since they do not always maintain their intended form. Afterward, the captured images need to be manually annotated with all the necessary information to describe the products accurately, such as brand, flavor, type, size, and other relevant attributes. As a result, making a well-represented dataset becomes a challenge for supermarkets. Several studies have tried to mitigate these limitations. In [21, 22], generative adversarial networks (GANs) have been successfully used to create synthetic context. In [4, 11], a matching technique is described to classify products based on a reference database containing product images under a controlled environment with different viewpoints. If a new product is introduced into the model, it will be classified based on the most similar product on the database. While these solutions tackle the challenges of recognizing objects, detecting them given the complexity of the background and domain shift adds another layer of complexity to the problem. Therefore, an automatic shelf monitoring system is still very challenging to implement.

To address these challenges, we propose a novel end-to-end solution for the automatic shelf monitoring system following a two-stage approach that leverages advanced machine learning techniques for enhanced performance and accuracy. Additionally, it explores various methods and loss functions for model selection and tuning. In the first stage, our system employs a state-of-the-art object detection model specifically fine-tuned on a large-scale dataset of retail product images. This model can accurately localize products on the shelves by generating precise bounding boxes. In the second stage, these bounding boxes are processed by a robust recognition model, which utilizes a deep convolutional neural network (CNN) architecture. This CNN is fine-tuned using supervised contrastive learning to ensure it can effectively classify products even in the presence of varying lighting conditions, occlusions, and packaging variations. By optimizing each stage independently, our solution ensures high detection and classification accuracy, making it a reliable tool for real-time shelf monitoring and inventory management.

The remainder of the paper is organized as follows. Section 2 briefly reviews related work. Section 3 describes the proposed methodology. Section 4 describes the conducted experiments and results and finally Section 5 concludes the paper.

2 Related Work

Traditional image processing techniques have a long history in textural, morphological, statistical, and multi-scale descriptors for compact representation of images due to their low costs and reasonable performance. These algorithms include, but are not limited to, Scale Invariant Feature Transform (SIFT), Speeded Up Robust Features (SURF), Binary Robust Invariant Scalable Keypoints (BRISK), Robust Independent Elementary Features (BRIEF), Features from Accelerated Segment Test (FAST), Oriented FAST and Rotated BRIEF (ORB), Gray Level Co-occurrence Matrix (GLCM), Hu's and Zernike's Moments, HARRIS-Stephens, Maximally Stable External Regions (MSER), MinEigen, Local Binary Pattern (LPB), Multilevel Binary Morphological Analysis, and Wavelets. Three examples of traditional methods are discussed in the following paragraphs.

Moorthy et al. [15] proposed a solution for detecting misplaced or missing on-shelf products in retail stores. Their approach involves storing a list of product cropped images to serve as reference, capturing target shelf images using a video or camera device, matching reference images with target images in grayscale, and identifying if the product is misplaced or missing. The input and target image descriptors are extracted and matched using the SURF algorithm, then the resulting indices of the matched features are passed to the Estimate Geometric Transform function to determine the presence of the reference in the target, and lastly drawing a bounding box around it. This method has a limitation to front-facing products and cannot handle occlusion.

Out-of-stock (OOS) products can be detected by looking for empty spots on shelves. However, empty shelves do not always indicate an OOS product. The presence of labels is also an important factor in distinguishing whether an empty space is for an OOS product or simply an empty space. The proposed method by [18] consists of three parts: Stitching panoramic images using homography estimation and Fast Explicit Diffusion (FED), detecting labels using a cascade of weak classifiers measuring Local Binary Pattern (LBP), and detecting OOS by doing aisle segmentation, OOS segmentation, vertical separation of OOS candidates, filtering candidates by the CIE L*C*h* color space, and finally feature extraction and classification.

Another approach is based on RGBD (RGB images with Depth information) point cloud data captured using consumer-grade sensors has been implemented to estimate the availability of products and OOS, with no prior knowledge [14]. The proposed system was made to monitor perishable fresh products stored in countertops, baskets, or crates, although it can monitor any type of layout. Top-mounted and front-facing camera setups were experimented with a reference model that is first calibrated on empty shelves, regardless of the camera orientation or floor layout; meaning it could be flat, or inclined in any direction. Multiple images are taken of the reference shelf floor and averaged together to reduce the Maximum Likelihood Estimation (MLE) error. Once the products have been placed, another calibration is done and depth points sticking up of the shelf floor are used to estimate the On-Shelf-Availability (OSA). Segmentation is applied to distinguish points belonging to products from the original reference plane using a threshold that is automatically set from the sensor's Root Mean Square Error (RMSE).

Recent advances in object detection techniques have attained remarkable success. However, detecting densely packed objects, especially in scenes containing many identical objects closely packed together, such as products on shelves, remains a challenging task. To tackle this issue, an innovative deep learning-based method is proposed in [5], with two key components. The first component uses Soft-IoU score as a Jaccard index to estimate the similarity between the predicted bounding box and the ground truth box. It is computed by incorporating a fully-convolutional layer added as a third head on top of the Region Proposal Network (RPN). This Soft-IoU value helps to resolve the issue of distinguishing overlapping objects that are closely positioned and may have similar appearances. The second component is the EM-Merger unit, which transforms the predicted bounding boxes along with their corresponding Soft-IoU scores into a Mixture of Gaussian's representations. The Expectation-Maximization (EM) algorithm is then employed to cluster these representations into groups, effectively separating overlapping or adjacent detected bounding boxes that were initially identified as separate objects.

The most difficult part of any machine learning solution is collecting the dataset. Often, real-world datasets are unlabeled, which means that significant human effort is required for annotation. This is particularly the case in retail stores, where collecting and labeling the dataset is an expensive process. There are various techniques to tackle this problem with satisfactory performance, one of which is semi-supervised learning (SSL), where a large amount of unlabeled data is utilized in combination with a smaller set of labeled data to train a model. In [23], a proposed approach fine-tunes three pre-trained models (RetinaNet, YOLOv3, and YOLOv4) on the labeled set based on mAP (mean average precision), F1-score, and recall evaluation metrics. Subsequently, the unlabeled set is assigned pseudo-labels generated from the best-performing model's predictions. These pseudo-labels are treated as ground truth during the training process for the unlabeled data. The best model is then retrained using both the labeled data and the unlabeled data with the assigned pseudo-labels. The final trained model is employed to detect 'Product', 'Empty Shelf', and 'Almost Empty Shelf' areas on the shelves.

In [8], the authors focused on the quality of the data, starting from collection, cleaning, and annotation before modeling. They collected a dataset of 1000 images following well-defined guidelines and annotated the images. The main goal of this work is to propose an end-to-end, real-time, computationally efficient pipeline solution for empty-shelf detection. They compared two different versions of both EfficientDet and YOLOv5 models to balance accuracy and inference run-time trade-offs. Additionally, they conducted extensive latency and throughput analysis of the models, utilizing several quantization and inference run-time optimization techniques. The observations are as follows. First, the time spent on image decoding and preprocessing must be optimized using parallelization. Second, the choice of batch size affects the model's maximum throughput. Third, memory requirements for model deployment should be determined carefully as they significantly impact both latency and throughput. Lastly, runtime optimizations like OpenVINO can boost the model's performance.

3 Methodology

A high-level overview of the proposed system pipeline for retail store shelf monitoring is shown in Figure 1. It consists of two stages. In the first stage, we evaluated several state-of-the-art detection methods to localize and generate bounding boxes around products on the shelves. We evaluated YOLOv8s, RetinaNet, Soft-IoU and EM-Merger, and SAPD on the SKU-110K dataset [5], which includes tightly-packed images of various scales, orientations, lighting conditions and noise levels from thousands of stores.

Based on the pilot testing, YOLOv8s [9] demonstrated superior performance achieving an mAP_{50-90} score of 0.558. While it was comparable to SAPD [24] in this metric, YOLOv8s showcases strong performance, making it an optimal choice for accurately localizing products on shelves. Additionally, YOLOv8s has a lightweight architecture (7.9M parameters) and fast inference speed (30FPS). These characteristics are important for real-world applications where computational resources and efficiency are essential considerations. Therefore, we select YOLOv8s as the primary model for product detection in our proposed shelf monitoring system.

Subsequently, these generated boxes will be input into the second stage, which is a recognition model for classification. The recognition model is a pretrained CNN model that is fine-tuned using supervised contrastive learning on products cropped from shelves instead of products taken under a controlled environment. This approach enables fine-grained classification, overcoming challenges of limited and under-represented datasets and domain shifts when new products are

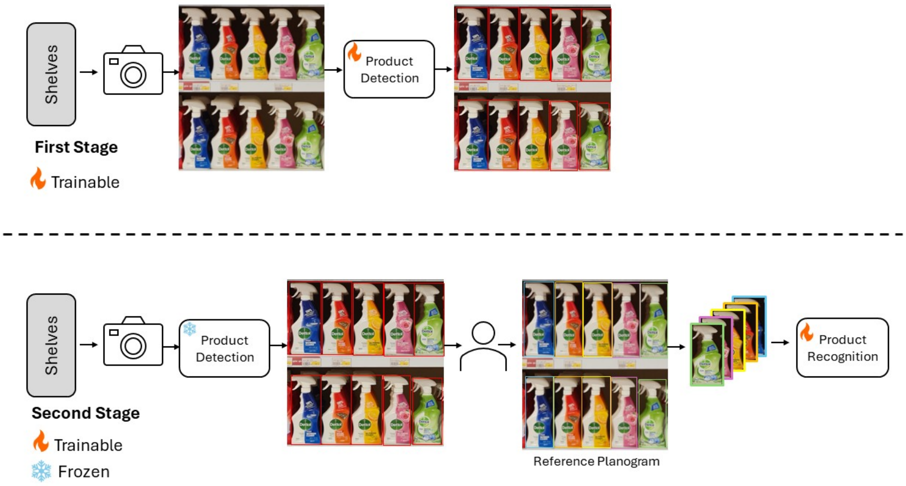


Fig. 1: High-level overview of the shelf monitoring system pipeline. The first stage involves training the detection model framework whereas the second stage is trained for product recognition utilizing detected products from the captured shelf images.

launched. This final output will be a comprehensive shelf image with all products detected and classified. This output will then be compared with a reference image of the shelf, adhering to the planogram layout of the rack. The rationale of the two-stage process allows for independent optimization of each component, ensuring the best performance of the detection and recognition models.

Recognition Stage Backbone: The backbone consists of a convolutional neural network serving as the feature extractor, responsible for capturing essential patterns and features from the input data. The choice of the backbone architecture can impact the performance and efficiency of the recognition model. More complex architectures may capture complex features but require more computational resources. On the other hand, simpler architectures may be more computationally efficient but could struggle to capture fine-grained features. We initially considered four options as potential backbone architectures: ResNet, EfficientNet, Vision Transformer (ViT), and MobileNet. ResNet is known for its strong generalization capabilities and its ability to capture features of varying complexities. EfficientNet strikes a balance between accuracy and computational efficiency, making it suitable for resource-constrained scenarios. ViT models excel in capturing global context but may require a large amount of training data. MobileNet is designed for resource-constrained devices and embedded applications, offering a good balance between accuracy and computational efficiency. The choice of backbone architecture should consider the specific requirements and available resources of the shelf monitoring system, such as computational resources, and available labeled data. Each mentioned backbone architecture has its advantages and trade-offs that should be carefully considered during the selection process.

Supervised Contrastive Learning: It aims to enhance the discriminative power of representations by leveraging labeled data. This approach contrasts representations of positive pairs (instances belonging to the same class) while simultaneously pushing representations of negative pairs (instances belonging to different classes) apart in an embedding space. By optimizing a contrastive loss function, which encourages similar representations for positive pairs and dissimilar representations for negative pairs, the model learns to capture meaningful and semantically rich features. Unlike traditional supervised learning methods that rely solely on cross-entropy loss, supervised contrastive learning enhances robustness against label corruption and dataset biases. The most commonly used loss functions in contrastive learning is InfoNCE, which is defined as follows:

$$L_{\text{InfoNCE}} = -\log\left(\frac{\exp\left(z_i.z_p/\tau\right)}{\sum_{n\in\mathcal{N}}\exp(z_i.z_n/\tau)}\right)$$

where $s_i = z_i.z_p/\tau$ represents the similarity score between the representation of the original sample i and its positive sample p, and $z_i.z_n/\tau$ represents the similarity score between the representation of the original sample and each of its negative sample $n \in N$. To extend InfoNCE to a supervised setting with multiple positives, SupCon (Supervised Contrastive) loss is introduced [10]:

$$L_{\text{SupCon}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathcal{P}_i|}\sum_{p\in\mathcal{P}_i}\log\left(\frac{\exp(z_i.z_p/\tau)}{\sum_{a\in A_i}\exp(z_i.z_a/\tau)}\right)$$

where $\mathcal{P}_i$ and $\mathcal{A}_i$ are the positive examples and negative examples of x_i, respectively, $z_i = f(x_i)$ represents the embedding or feature vector of x_i, τ is a temperature parameter for normalization, and N is the number of samples in the batch. To enhance the control over positive-negative pair distances and mitigate biases, an enhanced version of InfoNCE, called ε-SupInfoNCE, is proposed in [1], as illustrated in Fig. 2. It is defined by:

$$L_{e\text{-SupInfoNCE}} = -\sum_{i}\log\left(\frac{\exp(s_i^+)}{\exp(s_i^+ - \varepsilon) + \sum_j \exp(s_j^-)}\right) + \lambda R_{\text{FairKL}}$$

where λ is a regularization parameter that controls the strength of the FairKL debiasing regularization term while ε is the margin between positive and negative samples. By preventing the use of biased features and improving control over positive-negative pair distances, ε-SupInfoNCE enables the model to learn more robust and unbiased representations. This refinement can lead to better generalization and performance, especially in product recognition. Once the recognition model is fine-tuned using the supervised contrastive learning loss function, we do not use the classification layer anymore. Instead, the encoder layer of the model will be used to extract features of the input image. The extracted features will then be used later for planogram compliance optimization, which is explained next.

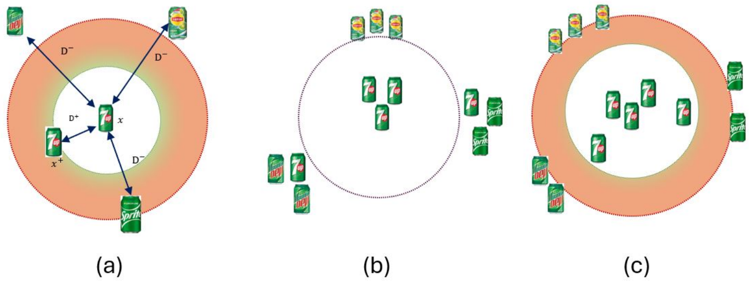


Fig. 2: Illustration of ε-SupInfoNCE that increases the distance between a positive sample and the nearest negative sample, creating a margin that better separates them. This margin prevents biased clusters from mixing positive and negative samples, as shown in scenario (c) compared to (b).

Planogram Compliance Control: Planograms are visual guides used in retail to arrange products on shelves effectively. They help promote products, improve customer navigation, and ultimately boost sales [2, 13, 19]. Traditionally, ensuring planogram compliance involves manual inspection, which is time-consuming and prone to errors. Various automated methods have been explored to address this, including inventory-based systems, RFID tags, and depth cameras. However, these approaches often fall short in terms of accuracy or cost-effectiveness. Our focus is on leveraging computer vision, a promising approach that has received less attention [12]. Our proposed solution involves installing fixed cameras in front of target shelving sections. Initially, a reference image is captured when products are correctly positioned. Each product in this image is annotated with a bounding box and labeled manually. These bounding boxes are then encoded into feature vectors and stored in a vector database for future comparison using the FAISS library [3]. When a test image is captured from the same camera, our detection model generates bounding boxes around products. To determine which products are correctly positioned, these boxes are compared with the reference boxes by calculating their Intersection over Union (IoU) values. The IoU value measures the overlap between the bounding boxes, indicating the degree of alignment between the detected and reference positions. Boxes with IoU values below a certain threshold (e.g., 15%) are considered misaligned and discarded. Subsequently, each correctly positioned product from the test image is cropped and encoded into a feature vector. These vectors are then compared with those in the reference database. If one of the top-5 matching vectors is found with the same desired label, the product is considered correctly placed; otherwise, it's deemed misplaced. Figure 3 illustrates the algorithm for compliance matching.

Out-of-stock Evaluation: We evaluate out-of-stock situations by measuring the alignment between the detected boxes in the test image and the reference boxes using a predefined IoU threshold value. if the IoU value falls below 15%, indicating poor alignment, we classify the corresponding reference box of the product as empty (out-of-stock). This algorithm works well most of the time as long as the reference and the test images are captured from the same camera position.

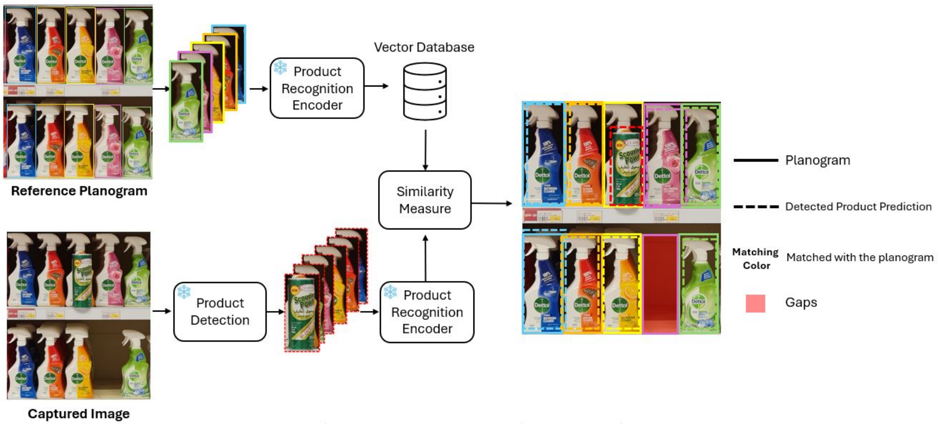


Fig. 3: Illustration of the compliance matching algorithm.

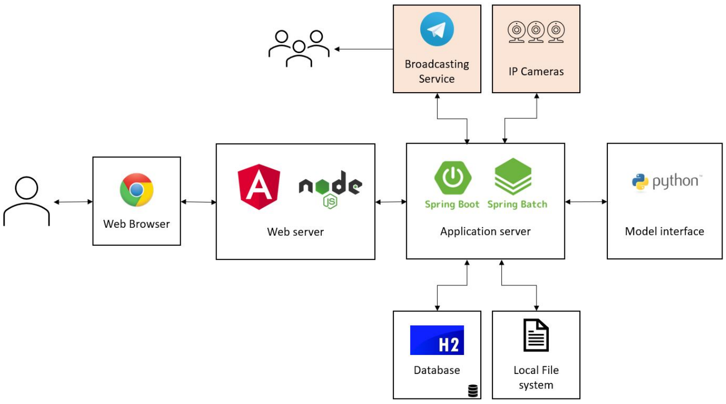


Fig. 4: High-level structure of the developed web-based application.

Web-based Application: After training and evaluating the model, we developed a full-stack web-based application as outlined in Figure 4. Through a web browser, the user interact with the web server, developed in Angular and Node.js, which in turn connects to an application server built using Spring Boot. The architecture is designed to easily integrate with Broadcasting services or IP cameras, enhancing the system's applicability in real-time monitoring and response scenarios. This flexibility is key in adapting to various operational environments and requirements. The front-end is developed using Angular, a platform and framework for building single-page client applications using HTML and TypeScript. Angular's modular nature allows developers to build large-scale, well-structured applications that are easy to maintain over time as well as providing intuitive navigation and interaction mechanisms for system users. This allows users to efficiently manage and interact with the system, from monitoring real-time data to configuring system settings and parameters.

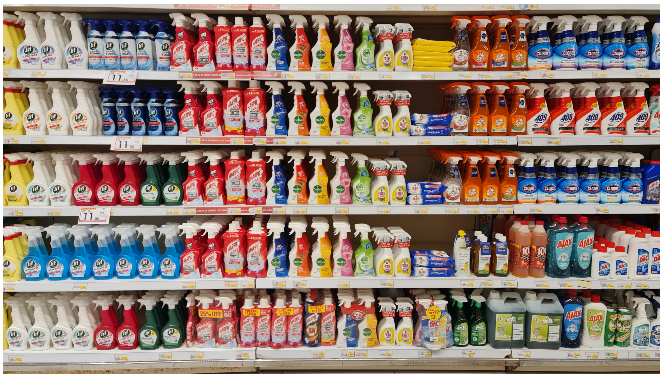


Fig. 5: An example image of the curated SOMAM dataset

4 Experimental Work

4.1 Data Collection and Preparation

SOMAM Dataset: This dataset is collected by us from Danube, which is a local supermarket in Saudi Arabia during October and November 2023, to address the lack of publicly available datasets that fit our vision-based shelf monitoring goals. SOMAM stands for Shelf Out-of-stock And Misplaced Monitoring. The images were captured using a mobile camera mounted on a tripod stand, positioned parallel in front of a shelf for a flat perspective. The dataset focuses on five shelves, containing a total of approximately 150 products. Each product is removed once as if OOS (Out-Of-Stock), then an image is captured, and then placed back in the original position, resulting in 150 images, each with a single OOS product. Next, 32 products were randomly selected, and each product was intentionally misplaced in five different places, each with a captured image, resulting in 160 images. A total of 310 images were collected for quick experimentation and evaluation. The dataset was annotated using Roboflow[5]. During the annotation process, we considered products that share similar visual appearance but differ by size, flavor, or quantity to have unique labels. Thus, each unique product has a unique class label. Figure 5 shows an example image of the dataset with all products on the correct place.

SHAPE Dataset: While preparing the camera-ready version of this paper, we found another dataset published online in June 2024, called is SHAPE (SHelf mAnagement Product datasEt)[6]. It provides a rich resource for fine-grained SKU classification tasks, offering 50,000 images across 17,000 distinct SKUs from 62 product categories [16]. Each image is carefully cropped to the boundaries of the product package and labeled with its corresponding European Article Number (EAN). These images were collected from 2,000 supermarkets in Italy using various smartphone cameras, ensuring diversity in image capture conditions. The dataset utilizes the Global Product Classification (GPC) system, which organizes

[5] `https://roboflow.com/annotate`

[6] `https://figshare.com/articles/dataset/SHAPE_-_SHelf_mAnagement_Product_datasEt/24100704`

products hierarchically into categories based on shared properties. For the purposes of our work, we leveraged the SHAPE dataset to train and evaluate the product recognition models. While it is a robust dataset for training and evaluating recognition models, it does not include shelf images with products positioned and labeled on shelves, which is required for our shelf monitoring algorithm, with each unique product assigned a distinct label. This limitation is addressed in our collected dataset.

4.2 Implementation Details and Results

We developed our models using Python and PyTorch library. Whenever possible, we utilized existing models and supplemented them with our custom code as needed. All experiments were run on a single RTX-4090 GPU.

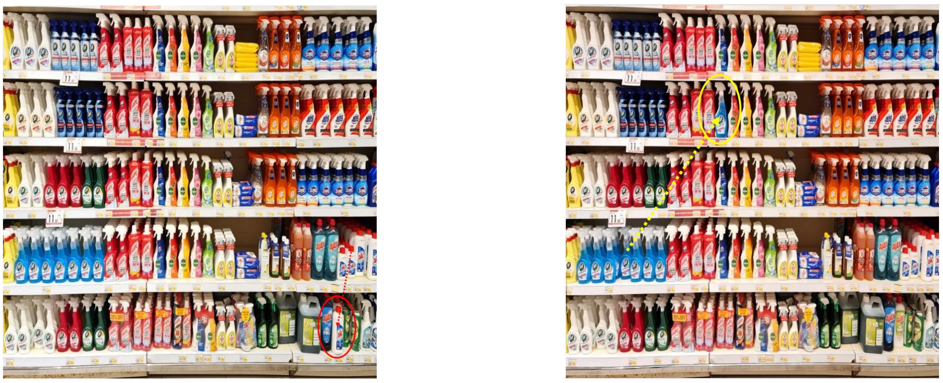
Baseline (One-stage model): Initially, we built a baseline model to establish a benchmark for comparison with more complex models. This baseline serves as a reference point to assess the performance of other proposed models. We created a version of SOMAM dataset with three classes only: out-of-stock (OOS), misplaced (MIS), and product, to focus on the target classes (MIS and OOS) and disregard all products that are correctly positioned on the shelves. In this way, the products can be localized and classified in one shot, which can also be compared to our proposed algorithm as described in the methodology section. We fine-tuned YOLOv5, YOLOv7, and YOLOv8s for 200 epochs. The model training performance is collected using a Tensorboard. Table 1 shows the comparison results of the three YOLO versions. The YOLOv8s model exhibits strong overall baseline performance on the test set. For misplaced products (MIS), the model achieves perfect precision but with a lower recall of 0.727, suggesting potential improvements in localization. For OOS, the model demonstrates a good performance with a precision of 0.962, perfect recall, and high mAP values of 0.995 and 0.78 for mAP_{50} and mAP, respectively. The ‘Product’ class is not our primary focus, as we are specifically interested in detecting OOS and MIS. However, including this class is necessary to enable the model to learn the correct location of each product. In conclusion, YOLOv8s shows reliable performance, especially in detecting Out-of-stock items. While it achieves a high precision for misplaced products, there is room for further improvement in recall scores.

Additionally, we identified instances where the baseline model failed to correctly detect the classes. Notably, our investigation revealed a specific challenge related to the detection of misplaced products, particularly those situated in front of items with similar shapes or colors. The model struggled in scenarios where the misplaced products shared visual similarities with neighboring items. A visual representation of these challenges is illustrated in Fig. 6. We believe that it’s necessary to augment the dataset with a variety of samples, particularly images with different product arrangements. Although the current model demonstrates a good performance for this task, a larger and more diverse dataset would provide it with a richer set of cases to learn from.

Performance Evaluation (Two-stage model with contrastive learning): Our proposed shelf compliance algorithm follows a deterministic process and depends heavily on the performance of both detection and recognition models, where

Table 1: Comparison of three YOLO versions on the test set as a baseline (MIS: MISplaced, OOS: Out-Of-Stock, Prc: Precision, Rec: Recall)

Method	Class	Prc	Rec	mAP_{50}	mAP_{50-90}
YOLOv5	Overall	0.362	0.408	0.550	0.416
	MIS	0.356	0.160	0.246	0.139
	OOS	0.089	0.067	0.409	0.255
	Product	0.640	0.999	0.994	0.853
YOLOv7	Overall	0.916	0.893	0.918	0.672
	MIS	1	0.680	0.759	0.420
	OOS	0.753	1	0.995	0.752
	Product	0.996	0.999	0.999	0.844
YOLOv8s	Overall	0.987	0.909	0.957	0.767
	MIS	1	0.727	0.882	0.648
	OOS	0.962	1	0.995	0.78
	Product	0.998	1	0.995	0.873



(a) (b)

Fig. 6: Example illustrating the model's difficulty in detecting a misplaced product in (a) visually complex scene, (b) when surrounded with visually similar items.

errors in the earlier stages can negatively impact the algorithm's overall accuracy. For the detection stage, we employed the YOLOv8s model to generate proposals and experimented with variants of recognition models, namely MobileNet [17], ResNet [7], and EfficientNet [20]. When comparing our two-stage compliance algorithm to the baseline method, it demonstrated superior recall while maintaining high precision as shown in Table 2. Notably, it achieved perfect precision and recall in identifying out-of-stock products, whereas the baseline method produced slightly lower precision at 96.2% for the same task, which could result from different reasons including increased model complexity or overlap of extracted features for misplaced products in the two-stage method. These findings indicate that our two-stage method, which utilizes contrastive learning to compare feature vectors, is more effective than relying solely on detection models for identifying compli-

Table 2: Performance comparison of two-stage contrastive learning models with a baseline one-stage model in terms of precision (Prc), recall (Rec) and F1

Method	Misplaced product			Out-of-stock		
	Prc	Rec	F1	Prc	Rec	F1
Baseline (YOLOv8s)	1.0	0.7270	0.8419	0.9620	1.0	0.9806
YOLOv8s + MobileNetV3-Small	0.9233	0.9331	0.9135	1.0	1.0	1.0
YOLOv8s + MobileNetV3-Large	0.9315	0.9234	0.9099	1.0	1.0	1.0
YOLOv8s + ResNet18	0.9395	0.9895	0.9549	1.0	1.0	1.0
YOLOv8s + EfficientNetV2-S	0.9687	0.9135	0.9308	1.0	1.0	1.0
YOLOv8s + EfficientNetV2-B0	0.8568	0.9226	0.8685	1.0	1.0	1.0

Table 3: Comparison of recognition models on SHAPE dataset using different models with different loss functions

			Accuracy		
Ref	Model	Loss	Top-1	Top-5	Top-10
[16]	MobileNetV3-Large	Triplet	0.93	0.96	0.97
	MobileNetV3-Small	Triplet	0.86	0.95	0.97
	EfficientNetV2-B0	Triplet	0.91	0.96	0.97
Ours	MobileNetV3-Large	ε-SupInfoNCE	0.9323	0.9877	0.9926
	MobileNetV3-Small	ε-SupInfoNCE	0.8979	0.9582	0.9680
	EfficientNetV2-B0	ε-SupInfoNCE	0.9410	0.9803	0.9926

ance issues. By incorporating reference images and comparing feature vectors, our method improves recall without sacrificing precision.

We also used the SHAPE dataset to compare the performance of the recognition models using the ε-SupInfoNCE supervised contrastive learning loss. Specifically, we compared three models: two versions of the MobileNetV3 network and one EfficientNetV2 network. These models were evaluated against the results reported by the authors of the SHAPE dataset [16]. Our models were trained using the AdamW optimizer with a learning rate of 0.0075 and a cosine decay schedule over 30 epochs, with a batch size of 64. As shown in Table 3, the use of the ε-SupInfoNCE loss function led to superior performance across all models compared to the Triplet loss. This improvement is likely due to ε-SupInfoNCE's more effective handling of sample distances, allowing for better differentiation between classes.

5 Conclusion and Future Work

In conclusion, the retail industry is witnessing a transformative shift driven by technological advancements aimed at enhancing the customer experience and operational efficiency. Real-time inventory management facilitated by computer vision technology offers substantial benefits by automating tasks such as shelf monitoring and product recognition. However, the complexity of real-world environments poses challenges such as varied visual representations and lighting conditions, necessitating robust solutions to ensure accurate performance. Our proposed solution for automatic shelf monitoring employs a two-stage deep learning approach to enhance performance and accuracy. The first stage utilizes a state-of-the-art object detection model, fine-tuned on a comprehensive dataset of retail product images, to accurately localize products on store shelves by generating precise bounding boxes. In the second stage, these bounding boxes are processed by a robust recognition model using a deep convolutional neural network (CNN) architecture fine-tuned with supervised contrastive learning. This two-stage process allows for the independent optimization of detection and recognition, ensuring high accuracy in real-time shelf monitoring and inventory management, thereby reducing out-of-stock rates and enhancing customer satisfaction. Moreover, a custom dataset, SOMAM, has been curated for further related work advancing the field. Future research can enhance the robustness of automatic shelf monitoring by focusing on several key areas, e.g. improving image quality through advanced preprocessing techniques to mitigate blurring and perspective distortion, and using incremental learning or one-shot learning to update the model without requiring complete retraining.

Acknowledgment

The authors would like to acknowledge the support during this work provided by King Fahd University of Petroleum and Minerals, and the Interdisciplinary Research Center for Intelligent Secure Systems (IRC-ISS), Saudi Arabia, under Grant No. INSS2204.

References

1. Barbano, C.A., Dufumier, B., Tartaglione, E., Grangetto, M., Gori, P.: Unbiased supervised contrastive learning. arXiv preprint arXiv:2211.05568 (2022)
2. Battiato, S., Gallo, G., Schettini, R., Stanco, F.: Image Analysis and Processing-ICIAP 2017: 19th International Conference, Catania, Italy, September 11-15, 2017, Proceedings, Part I, vol. 10484. Springer (2017)
3. Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.E., Lomeli, M., Hosseini, L., Jégou, H.: The faiss library (2024)
4. Georgiadis, K., Kordopatis-Zilos, G., Kalaganis, F., Migkotzidis, P., Chatzilari, E., Panakidou, V., Pantouvakis, K., Tortopidis, S., Papadopoulos, S., Nikolopoulos, S., et al.: Products-6k: a large-scale groceries product recognition dataset. In: 14th Pervasive Technologies Related to Assistive Environments Conference. pp. 1–7 (2021)
5. Goldman, E., Herzig, R., Eisenschtat, A., Goldberger, J., Hassner, T.: Precise detection in densely packed scenes. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5227–5236 (2019)
6. Guimarães, V., Nascimento, J., Viana, P., Carvalho, P.: A review of recent advances and challenges in grocery label detection and recognition. Applied Sciences **13**(5), 2871 (2023)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proc. IEEE conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
8. Jha, D., Mahjoubfar, A., Joshi, A.: Designing an efficient end-to-end machine learning pipeline for real-time empty-shelf detection. arXiv preprint arXiv:2205.13060 (2022)

9. Jocher, G., Chaurasia, A., Qiu, J.: Yolo by ultralytics (2023), `https://github.com/ultralytics/ultralytics`, accessed: February 30, 2023
10. Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., Krishnan, D.: Supervised contrastive learning. Advances in neural information processing systems **33**, 18661–18673 (2020)
11. Klasson, M., Zhang, C., Kjellström, H.: Using variational multi-view learning for classification of grocery items. Patterns **1**(8) (2020)
12. Laitala, J., Ruotsalainen, L.: Computer vision based planogram compliance evaluation. Applied Sciences **13**(18), 10145 (2023)
13. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14. pp. 21–37. Springer (2016)
14. Milella, A., Petitti, A., Marani, R., Cicirelli, G., D'orazio, T.: Towards intelligent retail: Automated on-shelf availability estimation using a depth camera. IEEE Access **8**, 19353–19363 (2020)
15. Moorthy, R., Behera, S., Verma, S., Bhargave, S., Ramanathan, P.: Applying image processing for detecting on-shelf availability and product positioning in retail stores. In: Proc. 3rd International Symposium on Women in Computing and Informatics. pp. 451–457 (2015)
16. Pietrini, R., Paolanti, M., Mancini, A., Frontoni, E., Zingaretti, P.: Shelf management: A deep learning-based system for shelf visual monitoring. Expert Systems with Applications **255**, 124635 (2024)
17. Qian, S., Ning, C., Hu, Y.: Mobilenetv3 for image classification. In: IEEE 2nd International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE). pp. 490–497 (2021)
18. Rosado, L., Gonçalves, J., Costa, J., Ribeiro, D., Soares, F.: Supervised learning for out-of-stock detection in panoramas of retail shelves. In: 2016 IEEE International Conference on Imaging Systems and Techniques (IST). pp. 406–411. IEEE (2016)
19. Saran, A., Hassan, E., Maurya, A.K.: Robust visual analysis for planogram compliance problem. In: 14th IAPR International Conference on Machine Vision Applications (MVA). pp. 576–579. IEEE (2015)
20. Tan, M., Le, Q.: Efficientnetv2: Smaller models and faster training. In: International conference on machine learning. pp. 10096–10106. PMLR (2021)
21. Tonioni, A., Serra, E., Di Stefano, L.: A deep learning pipeline for product recognition on store shelves. In: IEEE International Conference on Image Processing, Applications and Systems (IPAS). pp. 25–31 (2018)
22. Wei, X.S., Cui, Q., Yang, L., Wang, P., Liu, L.: Rpc: A large-scale retail product checkout dataset. arXiv preprint arXiv:1901.07249 (2019)
23. Yilmazer, R., Birant, D.: Shelf auditing based on image classification using semi-supervised deep learning to increase on-shelf availability in grocery stores. Sensors **21**(2), 327 (2021)
24. Zhu, C., Chen, F., Shen, Z., Savvides, M.: Soft anchor-point object detection. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16. pp. 91–107. Springer (2020)