

Concept-based Anomaly Detection in Retail Stores for Automatic Correction using Mobile Robots

Aditya Kapoor¹, Vartika Sengar¹, Nijil George¹, Vighnesh Vatsal¹
Jayavardhana Gubbi¹, Balamuralidhar P¹ and Arpan Pal¹

Abstract—Tracking of inventory and rearrangement of misplaced items are some of the most labor-intensive tasks in a retail environment. While there have been attempts at using vision-based techniques for these tasks, they mostly use planogram compliance for detection of any anomalies, a technique that has been found lacking in robustness and scalability. Moreover, existing systems rely on human intervention to perform corrective actions after detection. In this paper, we present Co-AD, a Concept-based Anomaly Detection approach using a Vision Transformer (ViT) that is able to flag misplaced objects without using a prior knowledge base such as a planogram. It uses an auto-encoder architecture followed by outlier detection in the latent space. Co-AD has a peak success rate of 89.90% on anomaly detection image sets of retail objects drawn from the RP2K dataset, compared to 80.81% on the best-performing baseline of a standard ViT auto-encoder. To demonstrate its utility, we describe a robotic mobile manipulation pipeline to autonomously correct the anomalies flagged by Co-AD. This work is ultimately aimed towards developing autonomous mobile robot solutions that reduce the need for human intervention in retail store management.

I. INTRODUCTION

In recent years, there have been significant technological innovations in retail store management, focusing on supply chains, logistics, and inventory tracking. With demographic shifts in industrialized economies, there is a push towards automating repetitive, labor-intensive tasks in domains such as retail stores and supermarkets. Advancements in robotics have enabled the possibility of deploying mobile manipulator robots for performing these tasks efficiently and autonomously.

One of the key activities in the retail domain is planogram compliance. A planogram is a schematic diagram predefined by store operators, laying out the shelf-wise location of each product, aimed at maximizing visibility and sales potential. Compliance implies checking the current layout and ensuring that it adheres to the planogram by correcting any deviations from it. Anomalies in the context of this paper refer to such deviations from the planogram. Most planogram compliance solutions depend on recognition of individual product categories followed by comparison with the reference planogram to detect anomalies. These solutions call for at least one reference image per product and a broad layout of how things should be placed [1]. Such solutions are reliant on manual

¹The authors are with TCS Research, Tata Consultancy Services Ltd., Bengaluru, Karnataka - 560066, India. e-mails: {aditya.kapoor1, vartika.sengar, george.nijil, vighnesh.vatsal, j.gubbi, balamurali.p, arpan.pal} @tcs.com

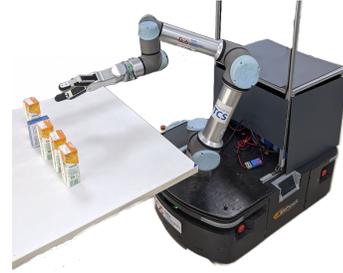


Fig. 1: Concept-based Anomaly Detection (Co-AD) can act as a key element in autonomous anomaly correction pipelines using mobile manipulator robots.

updates as new product varieties are constantly introduced and shop layouts are updated often. Also, there may be large variations in the kinds of planograms that are available at a given retail store. For instance, a planogram may simply contain representative shapes or gray scale images corresponding to each product. This negatively affects the performance of systems that rely solely on planogram matching to check for anomalies. To address this challenge, we propose Concept-based Anomaly Detection (Co-AD) which is a method for detecting out-of-context products in a retail environment. We also demonstrate how Co-AD can be a key component in developing mobile manipulation solutions that autonomously correct any flagged anomalies.

A. Related Work

Current state-of-the-art solutions for autonomous retail store shelf management through computer vision typically solve the problem of detecting anomalies in a two-stage manner. The anomalies are flagged by checking whether or not the observed image is in agreement with the pre-defined planogram [1]–[6]. In the first stage, each product placed on the shelf is localized and classified. The recognition is done using either traditional hand-crafted feature descriptors such as key-points, gradients, patterns, colors, or feature embeddings acquired from deep learning (DL) models [7]. In [2], the products are detected based on different methods such as sliding window-based HOG (Histogram of Oriented Gradients) and BOVW (Bag of Visual Words) features. Some solutions [1], [3], [4] use SURF, SIFT and Hough transform-based features for detection. The other solutions for planogram compliance use template matching based on morphological gradients [5] and recurrent pattern recognition [6]. In the second stage, matching is done between the

observations of the first stage and the actual layout given by the planogram. The methods used for matching include sub-graph isomorphism [1] and spectral graph matching [6].

However, the dependence on planograms limits the use of such systems in retail settings due to periodic layout updates and constant introduction of anomalies by human agents (store visitors). Also, there are challenges arising from accurate object detection and recognition in cluttered scenes, classification with extremely high number of object classes, and adaptability towards new classes [8]. These difficulties, along with class imbalances, large label spaces, and low inter-class variance limit the success of these methods.

In terms of robotic solutions, there have been mobile robots deployed in large retail stores for inventory tracking [9], using cameras mounted on a mobile base [10], [11]. These robots are designed to detect missing items on shelves, and alert store employees so that they may take corrective action. However, the reliance on human intervention along with the aforementioned limitations of planogram compliance-based approaches affect the scalability and profitability of these systems.

A possible solution to these shortcomings involves using scalable perception methods such as Co-AD, together with mobile manipulators to correct anomalies as they are detected, minimizing human intervention. In this paper, we focus mainly on perception—detecting the anomalies, while demonstrating a correction pipeline using existing techniques for task planning, motion planning and grasping. Another line of recent work in this field focuses on reactive action planning for mobile manipulators given a target object and goal state [12]. Our ongoing and future work is in this domain as well, using our custom mobile base [13] with a manipulator arm as a research platform (Fig. 1) for autonomous task and motion planning in retail scenarios.

B. Contributions and Overview

The key contributions of this paper are:

- A concept-based anomaly detection method (Co-AD) for out-of-context objects in a retail environment. This method does not rely on planogram compliance.
- A demonstration of how such an anomaly detection technique can drive a robotics pipeline involving mobile manipulation to correct the anomalies.

In Sec. II, we define the problem statement and the overall robotics pipeline. In Sec. III, we describe the Co-AD algorithm in detail. In Sec. IV, we find that Co-AD has a peak success rate of 89.9 % on a retail image dataset (RP2K [14]), and 95.83 % on simulated images (YCB [15]). Sec. V includes a demonstration of the application of Co-AD on mobile manipulation pipelines in simulated and real-world settings, followed by conclusions and future outlook in Sec. VI.

II. SYSTEM DESCRIPTION

A. Anomalies in the Retail Setting

Anomalies in retail settings are particularly different from the ones that have been tackled by current computer vision

approaches. In the computer vision literature, anomalies can be classified as spatial or temporal, based on the type of input data [16]. In case of temporal anomalies, the goal is to detect the anomalous activity in a video. In spatial anomalies, the goal is to find an image which deviates from the distribution of a normal class. For example, for the CIFAR10 [17], MNIST [18] and ImageNet [19] datasets, anomaly detection problems are posed as One-Class Novelty Detection where one class is considered as the normal class and the remaining classes are treated as anomalous [20]–[22]. Another example is defect detection [23], where the aim is to detect and localize spatial anomalies like cracks and structural changes in given images. The spatial anomalies discussed so far have a well defined distribution of normal class, however in our case the anomalies are contextual, making the problem of anomaly detection difficult. To ease some of these difficulties, we propose the following four types for categorizing anomalies in retail settings—

1) *Misplaced*: A *misplaced item anomaly* occurs when an object of type P has been incorrectly placed on a shelf row that has majority of type Q objects. The intervention would be to move the object P to either a region containing other type P objects, or to move it to an inventory buffer location such as a backroom or storage area.

2) *Out-of-stock*: An *out-of-stock or missing-item anomaly* occurs when several objects of type P are unavailable on a shelf row or the shelf row is empty. The intervention, in case of an empty shelf, would be to report that *shelf is vacant* else restock the objects P back to the shelf row from the inventory buffer location such as a backroom or storage area.

3) *Misalignment*: A *misalignment anomaly* occurs when one or several objects of type P are found in undesirable orientations. The intervention would be to re-align the objects P back to their original orientation on the shelf row by inferring the original orientation from the already existing group of objects of type P .

4) *Rearrangement*: A *rearrangement anomaly* occurs when one or several objects of type P and Q are shuffled on a shelf that is meant for them. The intervention would be to rearrange or regroup the objects P and Q back to their original positions on the shelf by inferring the original positions by gauging the existing object setup (based on count).

B. Problem Formulation

As discussed in Sec. I, any disruptions to a retail shelf arrangement that may require intervention from employees, are encapsulated under the umbrella term of “anomalies”. In a real large-scale store, it would not be feasible to enumerate every instance of the possible anomaly types and generate relevant plans for the robot to correct them. Based on feedback from retail store employees, particularly supermarkets, we restrict the scope of this work to one of the most commonly encountered anomalies— misplaced items.

The problem formulation for carrying out an intervention using a mobile manipulator therefore involves the following steps:

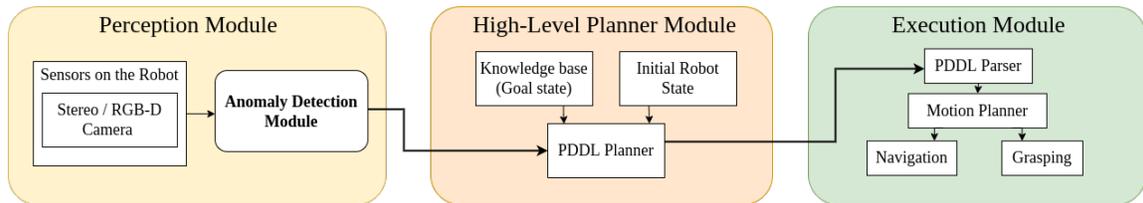


Fig. 2: The proposed solution pipeline for autonomous anomaly correction in a retail store.

- Scan single rows in a shelf to detect anomalous object(s). Identification or categorization of the object is not necessary to detect a misplaced item.
- Identify the misplaced item. Determine its goal position in the store using prior knowledge (planogram or group locations or buffer zone in a store room).
- Autonomously grasp the object, plan a path, navigate and place the object at its goal location.

C. Solution Pipeline

We propose a modular framework that would allow the robot to achieve the aforementioned steps (Fig. 2). We broadly categorize the anomaly detection and correction problem into the following three modules, allowing for incremental future scaling of each step of the solution:

- Anomaly Detection using **Perception**
- Planning to correct the Anomaly using a **High-Level Task Planner**
- Executing the plan using **Motion Planning**

By doing so, it becomes easier to replace the algorithms in a module with another whereas an end-to-end approach does not offer such flexibility since it requires retraining or fine-tuning the the entire system. The anomaly detection module within the perception module and its inter-play with the other modules is the focus of this paper.

1) *Perception Module*: The *Perception Module* perceives the environment around the robot with the help of sensors and uses this information to realise the anomalous object. In our current approach, we take a snapshot of the entire shelf and process it row by row. At first, we localize all the objects in the same shelf row (in simulation we directly use the ground truth whereas one can rely on prior works like [24]). Then we localize each object in the shelf row using Yolo-v6 [25] and pass the cropped images of them to a feature extraction module which produces the latent concept embeddings corresponding to each object. These embeddings will be used to identify the anomalous object.

2) *High-Level Planner Module*: The *High-Level Planner Module* takes inputs from the *Perception Module* about the anomalous object and the existing *Knowledge Base* (eg: its target location) and synthesizes a plan that enables the robot to correct the anomalies. In our current approach, we use PDDL [26] to plan and schedule the sequence of subroutines that would allow the robot to move and manipulate the anomalous object for correction.

3) *Execution Module*: The *Execution* module comprises of lower-level task primitives such as autonomous navigation and grasping that enable the robot to interact with the environment and correct any anomalies. In our current approach we use the ROS MoveIt! package [27] for motion planning which uses state-of-the-art inverse kinematics solvers, path planning algorithms, and collision detection. In case of grasping, we plan an inverse-kinematic path for the robotic arm’s end-effector in joint space, placing it in front of the target object, followed by the execution of a two-fingered grasp.

All of these modules are integrated with one another using the ROS framework.

III. ANOMALY DETECTION MODULE

The Co-AD approach consists of a number of components, which we will describe in order: product localization, disentangled concept embedding extraction and finally the full algorithm that arrives at anomaly detection and localization decisions.

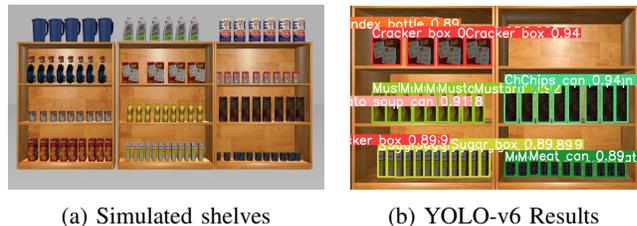


Fig. 3: Simulation setup in Gazebo: (a) Simulated shelves, (b) The localization results of YOLO-v6.

A. Product Localization Module

The first module uses a single-stage deep learning detector to localize retail products. We use a YOLO-v6 [25] detector to perform localization of different products in a given shelf image. The input to this module is pre-processed shelf images. The detector is trained on the SKU110k dataset [28] for 400 epochs and batch size of 16 on a Tesla V100 GPU machine.

Let us assume that I_s represents the shelf image which contains several products (e.g. Fig. 3a). The problem is to find the location of all the products in the given shelf image by rows. The localization module takes I_s as input and gives N_s bounding boxes, denoted by b_{ij} where j ranges from 1 to N_{sr} and i ranges from 1 to N_s . Here, N_{sr} is the number of shelf rows and N_s is the total number of objects. The

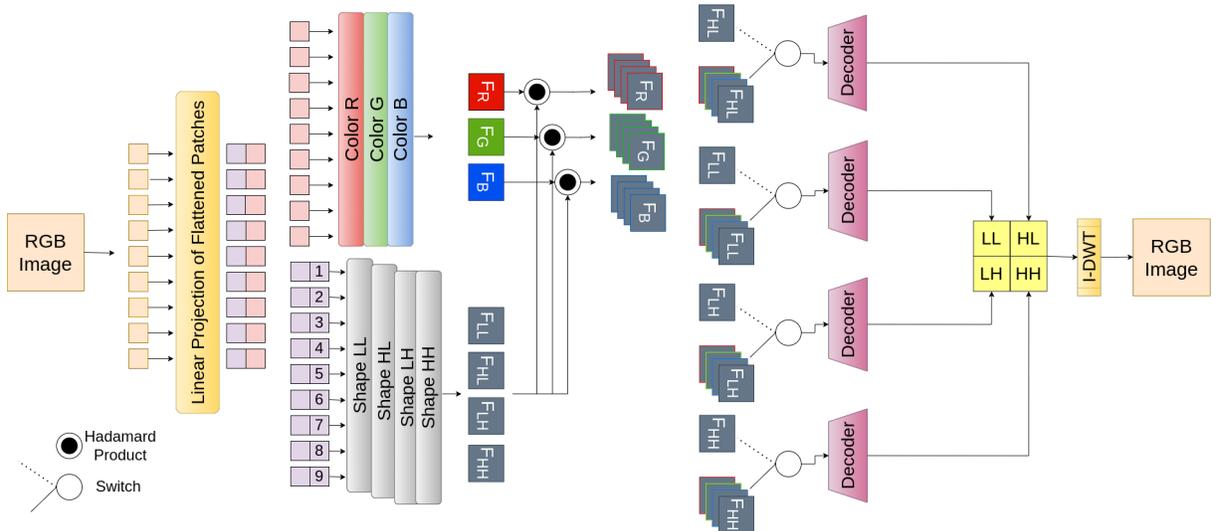


Fig. 4: The Concept Embedding Extraction Module Architecture.

localization results are shown in Fig. 3b. In simulation, we directly use the ground truth to uniquely identify the group of objects that are placed in different shelf rows after the localization step. Further, we crop the product images based on b_{ij} , which results in I_c which is a set of cropped images. We process all the cropped images row-wise i.e., all cropped images belonging to a row j are fed into a feature extraction module which outputs the concept latent embeddings for each object localized and is used for anomaly detection. This process is repeated for all the shelf rows.

B. Concept Embedding Extraction Module

Assume that a retail product dataset contains images of different products. Note that different products in the dataset have different colors, orientations, physical and pixel dimensions. We don't assume the presence of a classification label for each product. The goal of this module is to extract features corresponding to each product, that can be used to perform context-based anomaly detection.

In the retail domain, collecting labels for every product may not be feasible in all cases and because of constant change in product packaging, there is a high probability that a model might not be able to classify different products into their respective categories. Additionally, it has been observed that even within the same class, there exist variations in visual appearance based on different flavors (colors of packaging), sizes, etc. Therefore, it might not be the best strategy to train a classifier using class labels and then use this classifier to extract discriminative features corresponding to each product in order to perform anomaly detection. As a result of these factors, an unsupervised technique is required to get meaningful features that can be utilised to establish contexts for context-based anomaly identification. For instance, an anomaly can be color-based when a red colored packet is placed on a shelf among blue packets or it can be shape-based when a ball is placed among Rubik's cubes.

One possible solution which is extensively explored for the planogram compliance problem is to use handcrafted features like key-point-based (Scale-Invariant Feature Transform (SIFT) [29], Speeded-Up Robust Feature (SURF) [30]), gradient-based (Histogram of Oriented Gradients (HOG), Sobel, Prewitt operators), color-based (color histogram), pattern-based (Haar-like features). However, these techniques do not always display sufficient information and it becomes an engineering effort to choose the features and parameters that fit best. To deal with this, we can use deep learning (DL)-based solutions which have shown improved performance. Thus, we propose an auto-encoder architecture which can learn disentangled color and shape content latent embeddings as in [31].

We compare the performance of our method with other baselines and show that the disentangled latent embeddings are useful for identifying the anomalies in an explainable manner. Further, the advantage in having disentangled latent embeddings over distributed embeddings is that only important and necessary subsets of the embeddings can be utilized for a downstream task. For example, a robotic gripper requires information about the shape, size and texture of the surface of the object to infer a grasp but does not require color information. As a result, by providing relevant information to the gripper, one can increase the signal-to-noise ratio in the input space. This is not possible to do in conventional auto-encoder setups since they contain all the necessary information to accurately reconstruct the original image, but the information related to different concepts like shape, size, location, color, etc. are all in a tightly entangled representation. This entangled information is not discriminative enough to perform context based anomaly detection. Also, the planning module typically acts on local representations of the concepts but DL models give distributed representations where concepts are denoted by continuous-value vectors [32]. Thus, in order to bridge between the two paradigms the first step is to disentangle the representations

given by standard DL models without the use of external concept annotations.

In order to capture disentangled color and content information, we introduce two methods wherein we train a vision transformer (ViT) [33], [34] to encode images into latent embeddings. The two methods differ in how the shape and texture information (content information) of the object is captured in the latent representation. In the first method, ViT concept mining DWT (ViT-CM-DWT), we use Discrete Wavelet Transform (DWT) using Haar as the mother wavelet, with content latent embeddings specialized to reconstruct the DWT components of the gray scale image of the object (e.g., Level-1 DWT components: LL, HL, LH, and HH) similar to [31]. In the second method, ViT concept mining (ViT-CM), we extract content latent embeddings to reconstruct the gray scale image of the object without using DWT. On the other hand, the color latent embeddings are trained to capture the color information of an image.

The ViT auto-encoder (ViT-CM-DWT) architecture for color and shape content feature specialisation is shown in Fig. 4. The RGB images used to train the ViT-AE are of size $3 \times 224 \times 224$ (CxHxW). The Linear Projection layer is made up of a single convolutional network (input channels=3, output channels=128, kernel size = 16 and patch size = 16) which takes in an object image and implicitly splits them into image patches of size $3 \times 16 \times 16$ to generate a feature vector of 128 dimension for each image patch. Further, the latent embeddings are split into two equal halves of dimension 64 each. One part of the embedding is passed through the 4 shape transformer encoder layers (one for each DWT component LL, HL, LH, HH) and the other half is passed to the 3 color transformer encoder layers (one for each Red channel, Green channel and Blue channel). Each transformer encoder comprises of 4 attention heads, a feed forward network with a hidden dimension of 2048 and uses GELU activation function. The intention is to specialize a part of the embedding to learn the different shapes and textures present in the object image and the other part to learn the colors. The position embeddings are only made available to the latent embeddings for content. The latent embedding $f \in R^{N \times N_C \times M}$ of an input image $I_c \in R^{H \times W \times 3}$ is produced by the encoder network (E).

$$f = E(I_c; \theta_E) \quad (1)$$

Here, latent embedding $f \in R^{N \times N_C \times M}$ comprises of disentangled color and content specialized features where N represents the total number of patches ($N=16$), N_C is the total number of specialized features or concepts ($N_C=7$ because there are 3 color channels and 4 DWT components) and M denotes the dimension of these features.

$$\begin{aligned} f &= \{f_{color}, f_{content}\} \\ f_{color} &= \{f_R, f_G, f_B\} \\ f_{content} &= \{f_{LL}, f_{HL}, f_{LH}, f_{HH}\} \end{aligned}$$

Similar to [31], the color embeddings f_R, f_G and f_B are used to modulate the content embeddings $f_{content}$, as described

below:

$$f_{content}^R = \{f_{LL} \odot f_R, f_{HL} \odot f_R, f_{LH} \odot f_R, f_{HH} \odot f_R\} \quad (2)$$

$$f_{content}^G = \{f_{LL} \odot f_G, f_{HL} \odot f_G, f_{LH} \odot f_G, f_{HH} \odot f_G\} \quad (3)$$

$$f_{content}^B = \{f_{LL} \odot f_B, f_{HL} \odot f_B, f_{LH} \odot f_B, f_{HH} \odot f_B\} \quad (4)$$

Here, \odot represents the Hadamard product and f_{LL}, f_{HL}, f_{LH} and f_{HH} are the specialized features for each of the DWT components. Following this, the modulated features are passed to the decoder bank to reconstruct the DWT components of the input image I_c .

$$\phi_{DWT}^x = D(f_{content}^x; \theta_D) \quad (5)$$

Where, $x \in \{R, G, B\}$, θ_D represent the parameters of decoder D . The image is reconstructed back using the IDWT module [35] through the standard process.

The auto-encoder is trained in an end-to-end fashion with squared L2 loss between the input image and the reconstructed image. Note that to disentangle the color and content information, we train the network alternatively where in one iteration the $f_{content}$ embeddings are modulated with f_{color} embeddings as described in equations (2)-(4) above. $f_{content}$ is detached from the computational graph so that it does not learn to contain information about the color of the object image. In this case, the L2 loss is calculated between the RGB image and the reconstructed image. In the next iteration, $f_{content}$ embeddings are passed as is, without modulation, enforcing the network to only learn the content features and no color information is passed to the decoder for reconstruction. In this case, the loss is computed between gray-scale input image and the reconstructed image.

In case of ViT-CM, we replace the 4 content encoder layers with a single content encoder layer to learn the shape and texture information of the object. The training process remains the same except, we have a single content latent embedding $f_{content}$ instead of 4.

To train our models and baselines we use the object RGB images and segmentation masks in the YCB dataset [15]. We train our models for 100 epochs with a learning rate of $1e-4$ using Adam optimizer and mean squared error loss.

C. Algorithm for Anomaly Detection

The features obtained from the concept embedding extraction modules are then used to perform anomaly detection using outlier detection algorithms. We implemented two such algorithms. In the first one, we applied agglomerative clustering on the concept embeddings corresponding to each cropped image. In the second one, we calculated pairwise distances between the embeddings corresponding to each cropped image and then computed a row-wise sum on the resulting distance matrix. On the aggregated distance we then detected outliers based on the inter-quartile range.

IV. EVALUATION

We have evaluated our proposed approach for anomaly detection on a simulated dataset as well as on publicly

Algorithm 1 Anomaly Detection

```

1: for  $evaluation\_set = 1, 2, \dots, K$  do
2:   for  $object\_images = 1, 2, \dots, N$  do
3:     Compute feature vectors  $\hat{f}_1, \dots, \hat{f}_N$  using Eq 1
4:   end for
5:   Compute distance between feature vectors using a
   similarity metric
6:    $anomaly[K] = argmax(distance)$ 
7: end for

```

available retail product dataset, RP2K. To evaluate our anomaly detection approach, we have created evaluation sets by randomly choosing majority samples from one class and anomalous sample from another randomly chosen class. The Co-AD approaches tested here include ViT-CM and ViT-CM-DWT architectures with selections of content features, color features, or both. These are compared with a similar architecture that uses convolutional neural networks (CNN) for concept mining to generate latent embeddings (CNN-CM-DWT) with the same feature selection choices [31]. We have also compared our Co-AD approach with standard ViT auto-encoders which do not need product labels, as well as with a deep residual network (ResNet50) [36] that requires labelled data. Co-AD was found to generate anomaly detection outputs from a given simulated or real scene at ~ 5 fps on the onboard computer of the mobile robot (NVIDIA Jetson TX2).

A. Image Dataset - RP2K

The RP2K dataset [14] contains two components: the original shelf images and the individual object images cropped from the shelf images. All images are captured in physical retail stores with natural lightings, matching the scenario of real applications. In this dataset each individual object is at least 80 by 80 pixels. To prepare the evaluation set, we considered cropped images from the dataset. There are a total of 198 test cases in the evaluation set. The anomaly detection accuracy is listed in Table I. ViT-CM-DWT with color features and a pairwise boxplot distance metric had the highest accuracy for this dataset (89.9%).

B. Simulated Images

We also evaluate Co-AD on a set of images containing objects arranged on shelves in a ROS Gazebo simulation environment. As shown in Fig. 3, the test set contains images of 12 different object classes drawn from the YCB dataset [15]. We have created 72 test cases containing different anomalous objects. Table II lists the accuracy on simulated data. ViT-CM with color features and a pairwise boxplot distance metric had the highest accuracy for this dataset (95.83%).

C. Evaluation Sets - Failure Cases

To demonstrate the limitations of Co-AD approaches, we present two examples of evaluation sets from the simulated images and RP2K images where ViT-CM fails to identify an anomaly. The failure cases are identified after using

Model	Agglomerative clustering on features	Boxplot on pairwise distance
ResNet	51.52%	69.7%
ViT-AE	74.75%	80.81%
CNN-CM-DWT (Color features)	29.8%	43.43%
CNN-CM-DWT (Content features)	46.97%	56.06%
CNN-CM-DWT (Content & Color features)	48.48%	59.6%
ViT-CM-DWT (Color features)	84.34%	89.9%
ViT-CM-DWT (Content features)	41.41%	45.45%
ViT-CM-DWT (Content & Color features)	83.84%	86.87%
ViT-CM (Color features)	76.77%	84.85%
ViT-CM (Content features)	57.58%	65.66%
ViT-CM (Content & Color features)	80.3%	89.39%

TABLE I: Success rate on RP2K dataset

Model	Agglomerative clustering on features	Boxplot on pairwise distance
ResNet	75.0%	88.89%
ViT-AE	72.22%	70.83%
CNN-CM-DWT (Color features)	50.0%	62.5%
CNN-CM-DWT (Content features)	47.22%	50.0%
CNN-CM-DWT (Content & Color features)	55.56%	69.44%
ViT-CM-DWT (Color features)	86.11%	90.28%
ViT-CM-DWT (Content features)	40.28%	43.06%
ViT-CM-DWT (Content & Color features)	84.72%	88.89%
ViT-CM (Color features)	80.56%	95.83%
ViT-CM (Content features)	45.83%	47.22%
ViT-CM (Content & Color features)	93.06%	91.67%

TABLE II: Success rate on Simulated dataset

agglomerative clustering on both color and content features. In the first row of both Fig. 6 and Fig. 7, the ViT-CM approach fails because the color feature vectors of each of the object cannot be told apart. On the other hand, the second row of both Fig. 6 and Fig. 7 we observe that the content features of all the object images are close to each other that leads to inaccurate anomaly detection.

The Concept-based Anomaly Detection (Co-AD) approach presented in this paper consists of various design choices— inclusion of DWT, choice of color and content features and choice of outlier detection algorithm. As seen in the evaluation results, the anomaly detection performance on different test sets varies with these choices. Co-AD can therefore be suitably modified and adapted to a particular application area through these design choices with minimal



Fig. 5: Representative images of retail objects present in the RP2K dataset [14]



Fig. 6: Examples of images from the RP2K [14] evaluation set where anomaly detection using ViT-CM fails due to color features (top row) and content features (bottom row).

changes in the underlying architecture.

V. AUTOMATIC ANOMALY CORRECTION DEMOS

While the concept-based anomaly detection methods outperformed baselines on the evaluation image datasets, their real utility is as part of an autonomous robotics pipeline, obviating the need for human intervention.

We demonstrate the pipeline on a mobile manipulation platform (Fig. 1) consisting of a custom mobile base [13], a Universal Robots UR5e arm, an OnRobot RG6 gripper (Robotiq 2F-85 in simulation), and an Intel Realsense D415 depth camera.

A. ROS Gazebo Simulator

The ROS Gazebo simulation of a retail anomaly detection task consists of a shelf with objects placed on two rows. The objects are drawn from the YCB dataset [15] of everyday objects. The depth camera is simulated via a Gazebo plugin and placed on the robot’s body at roughly the same position as the real camera. The robot’s objective is to scan the shelf, flag the anomalous object using Co-AD, pick it up, and place it on the table behind the shelf that serves as an inventory buffer area (Fig. 8a) whose location is known *a priori* in the world frame.

Following the procedure described in Sec. III, the soup can is identified as the anomaly (Fig. 8b). This triggers the pipeline shown in Fig. 2, calling a PDDL planner that generates a task plan with the primitive actions *move*, *pick-up* and *put-down*, for taking the can from the shelf to the table. The task plan is parsed into sequence of motion plans using Moveit [27] for planning the robotic arm’s trajectory and ROS Navigation for the mobile base.



Fig. 7: Examples from the simulated images [15] evaluation set where anomaly detection using ViT-CM fails due to color features (top row) and content features (bottom row).



(a) Simulation environment

(b) RGB image

Fig. 8: An illustrative task simulated in a ROS Gazebo environment: (a) mock retail setup, (b) image captured by the robot’s on-body camera with anomalous object shown.

B. Physical Demonstration

Another demonstration was conducted with the physical robot, with one object of a different class placed among four other objects (Fig. 9). The objects were picked from a local grocery store. The anomalous object was detected using Co-AD, and picked up with the robotic arm. Fig. 9c shows boxplots for Co-AD concept embeddings for the five objects in the image. The similarity metric is normalized pairwise distances in the color features and shape content features. The dissimilar object can be seen as an outlier.

VI. CONCLUSION

In this paper, we presented a concept-based anomaly detection method (Co-AD) that allows for the detection of misplaced items in a retail store without relying on planograms or labeled object databases. While this approach performed well on real and simulated image datasets, investigation and improvement of its real-world performance in physical retail stores remains ongoing.

As part of a mobile manipulation platform that can autonomously correct anomalies in retail stores, Co-AD is useful due to its scalability and low computational burden. While the current implementation uses well-established techniques for task and motion planning, a real deployment requires more reactivity and adaptability to changing environments in terms of navigation and manipulation. These challenges have been enumerated in [37] and the development of solution strategies constitutes ongoing and future work.

REFERENCES

- [1] A. Tonioni and L. D. Stefano, “Product recognition in store shelves as a sub-graph isomorphism problem,” in *International Conference on Image Analysis and Processing*. Springer, 2017, pp. 682–693.

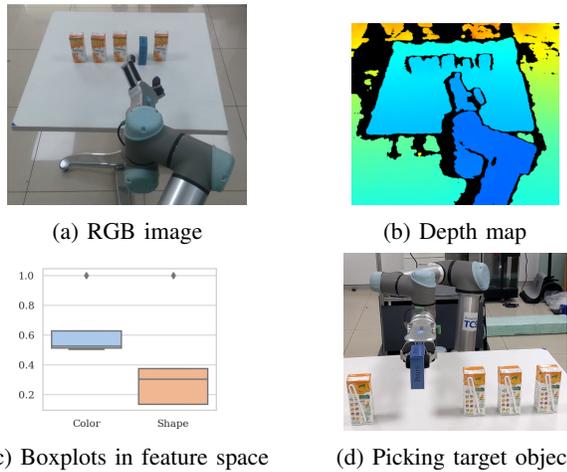


Fig. 9: A simplified task executed on the robot with anomaly detection: (a) image from robot's on-body camera, (b) corresponding depth image, (c) boxplots of normalized pairwise distance in shape content and color features with the outlier shown, (d) anomalous object picked up.

- [2] M. Marder, S. Harary, A. Ribak, Y. Tzur, S. Alpert, and A. Tzadok, "Using image analytics to monitor retail store shelves," *IBM Journal of Research and Development*, vol. 59, no. 2/3, pp. 3–1, 2015.
- [3] A. Saran, E. Hassan, and A. K. Maurya, "Robust visual analysis for planogram compliance problem," in *2015 14th IAPR International Conference on Machine Vision Applications (MVA)*. IEEE, 2015, pp. 576–579.
- [4] A. Ray, N. Kumar, A. Shaw, and D. P. Mukherjee, "U-pc: unsupervised planogram compliance," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 586–600.
- [5] E. Frontoni, M. Contigiani, and G. Ribighini, "A heuristic approach to evaluate occurrences of products for the planogram maintenance," in *2014 IEEE/ASME 10th International Conference on Mechatronics and Embedded Systems and Applications (MESA)*. IEEE, 2014, pp. 1–6.
- [6] S. Liu, W. Li, S. Davis, C. Ritz, and H. Tian, "Planogram compliance checking based on detection of recurring patterns," *IEEE MultiMedia*, vol. 23, no. 2, pp. 54–63, 2016.
- [7] B. Santra and D. P. Mukherjee, "A comprehensive survey on computer vision based approaches for automatic identification of products in retail store," *Image and Vision Computing*, vol. 86, pp. 45–63, 2019.
- [8] Y. Wei, S. Tran, S. Xu, B. Kang, and M. Springer, "Deep learning for retail product recognition: Challenges and techniques," *Computational intelligence and neuroscience*, vol. 2020, 2020.
- [9] J. Gee, B. Bogolea, and M. A. Shah, "Inventory-tracking robotic system," U.S. Patent USD819712S1, May 05, 2018.
- [10] W. Knight, "Robot makes sure stores don't run out of doritos," *Technology Review*, 2015. [Online]. Available: <http://www.technologyreview.com/news/543281/robot-makes-sure-stores-dont-run-out-of-doritos>
- [11] M. McCarthy, "Chilean robot-as-a-service company zippedi raises 6.9 million to digitize stores," *Bloomberg Linea*, 2021. [Online]. Available: <https://www.bloomberglia.com/2021/09/07/chilean-robot-as-a-service-company-zippedi-raises-69-million-to-digitize-stores/>
- [12] C. Pezzato, C. Hernandez, S. Bonhof, and M. Wisse, "Active inference and behavior trees for reactive action planning and execution in robotics," *arXiv preprint arXiv:2011.09756*, 2020.
- [13] V. P. B. Srinivas, V. R. C. Patta, and S. Kumar, "Industrial robot," U.S. Patent USD924291S, July 06, 2021.
- [14] J. Peng, C. Xiao, and Y. Li, "Rp2k: A large-scale retail product dataset for fine-grained image classification," *arXiv preprint arXiv:2006.12634*, 2020.
- [15] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Yale-cmu-berkeley dataset for robotic manipulation research," *The International Journal of Robotics Research*, vol. 36, no. 3, pp. 261–268, 2017. [Online]. Available: <https://doi.org/10.1177/0278364917700714>
- [16] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.
- [17] A. Krizhevsky, G. Hinton, et al., "Learning multiple layers of features from tiny images," 2009.
- [18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [20] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture on IE*, vol. 2, no. 1, pp. 1–18, 2015.
- [21] R. Chalapathy, A. K. Menon, and S. Chawla, "Anomaly detection using one-class neural networks," *arXiv preprint arXiv:1802.06360*, 2018.
- [22] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International conference on machine learning*. PMLR, 2018, pp. 4393–4402.
- [23] P. Bergmann, M. Fauser, D. Sattlegger, and C. Steger, "Mvtec ad—a comprehensive real-world dataset for unsupervised anomaly detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 9592–9600.
- [24] J. Fan and T. Zhang, "Shelf detection via vanishing point and radial projection," in *2014 IEEE International Conference on Image Processing (ICIP)*, 2014, pp. 1575–1578.
- [25] C. Li, L. Li, H. Jiang, K. Weng, Y. Geng, L. Li, Z. Ke, Q. Li, M. Cheng, W. Nie, et al., "Yolov6: A single-stage object detection framework for industrial applications," *arXiv preprint arXiv:2209.02976*, 2022.
- [26] M. Ghallab, C. Knoblock, D. Wilkins, A. Barrett, D. Christianson, M. Friedman, C. Kwok, K. Golden, S. Penberthy, D. Smith, Y. Sun, and D. Weld, "PDDL - The Planning Domain Definition Language," *Technical Report CVC TR-98003/DCS TR-1165*, Yale Center for Computer Vision and Control, 1998.
- [27] I. A. Sucas and S. Chitta, "MoveIt," [Online] Available at <https://moveit.ros.org>.
- [28] E. Goldman, R. Herzig, A. Eisenschat, J. Goldberger, and T. Hassner, "Precise detection in densely packed scenes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5227–5236.
- [29] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *International journal of computer vision*, vol. 60, no. 1, pp. 63–86, 2004.
- [30] H. Bay, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.
- [31] V. Sengar, B. Vivek, G. Bhattacharya, J. Gubbi, A. Pal, and P. Balamuralidhar, "Low-level bias discovery and mitigation for image classification," in *2022 IEEE International Conference on Signal Processing and Communications (SPCOM)*. IEEE, 2022, pp. 1–5.
- [32] A. d. Garcez and L. C. Lamb, "Neurosymbolic ai: the 3rd wave," *arXiv preprint arXiv:2012.05876*, 2020.
- [33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," 2017. [Online]. Available: <https://arxiv.org/abs/1706.03762>
- [34] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," 2020. [Online]. Available: <https://arxiv.org/abs/2010.11929>
- [35] F. Cotter, "Uses of complex wavelets in deep convolutional neural networks," Ph.D. dissertation, University of Cambridge, 2020.
- [36] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [37] V. Sengar, A. Kapoor, N. George, V. Vatsal, J. Gubbi, A. Pal, et al., "Challenges in applying robotics to retail store management," *arXiv preprint arXiv:2208.09020*, 2022.