IOT Based Real-Time Wild Animal Detection And Alerting System

Soundari Rebekha A
Dept of Computer Science and Engineering
Ponjesly College of Engineering

Abstract- *Object recognition in unseen indoor environments remains a challenging problem for visual perception of mobile robots. In this letter, we propose the use of topologically persistent features, which rely on the objects' shape information, to address this challenge. In particular, we extract two kinds of features, namely, sparse persistence image (PI) and amplitude, by applying persistent homology to multi-directional height function-based filtrations of the cubical complexes representing the object segmentation maps. The features are then used to train a fully connected network for recognition. For performance evaluation, in addition to a widely used shape dataset and a benchmark indoor scenes dataset, we collect a new dataset, comprising scene images from two different environments, namely, a living room and a mock warehouse. The scenes are captured using varying camera poses under different illumination conditions and include up to five different objects from a given set of fourteen objects. On the benchmark indoor scenes dataset, sparse PI features show better recognition performance in unseen environments than the features learned using the widely used ResNetV2-56 and EfficientNet-B4 models. Further, they provide slightly higher recall and accuracy values than Faster R-CNN, an end-to- end object detection method, and its state-of-the-art variant, Domain Adaptive Faster R-CNN. The performance of our methods also remains relatively unchanged from the training environment (living room) to the unseen environment (mock warehouse) in the new dataset. In contrast, the performance of the object detection methods drops substantially. We also implement the proposed method on a real-world robot to demonstrate its usefulness.*

Keywords- Recognition, AI-Enabled Robotics, Object Detection, Segmentation and Categorization

I. INTRODUCTION

Perception is one of the core capabilities for autonomous mobile robots, and vision plays a key role in developing a semantic-level understanding of the robot's environment. Object recognition and localization, often referred together as detection, therefore, form an essential aspect of robot visual perception. Deep learning models for object recognition [1],

[2] and object detection [3]–[5] have achieved tremendous success in recognizing and detecting objects even in cluttered or crowded scenes. Such models, however, tend to require a large amount of training data. Therefore, a common practice is to use a model that is pre-trained on large databases with millions of images such as ImageNet, and then fine-tune the model based on scene images from the environments where the robots are expected to operate. This practice becomes cumbersome and runs into challenges for long-term robot autonomy, where the robots operate in complex and continually-changing environments for extended periods of time. Moreover, it renders the models sensitive to variations in environmental conditions such as illumination and object texture [6], [7]. Therefore, perception robustness becomes critical in such applications [8].

Different human interaction-based [9] and human supervision-based [10] methods have been developed recently to address this challenge. Additionally, various domain adaptation methods have also been proposed for cross- domain object detection, i.e., detection in environments that have distributions different from that of the original training environment [11]. For instance, within the Faster R-CNN framework, adversarial learning has been used for image and object instance level adaptation [12] and feature alignment [13]. Saito et al. [14] use adversarial learning for strong local and weak global alignment of features in an unsuper- vised setting. Hsu et al. [15] also use adversarial learning but perform progressive adaptation through an intermediate domain, which is synthesized from the source domain images to mimic the target domain. Inoue et al. [16] propose a two-step progressive adaptation technique in a weakly-supervised setting where the detector is fine-tuned on samples generated using CycleGAN and pseudo-labeling. Similarly, Kim et al. [17] use CycleGAN to perform domain diversification, followed by multidomain-invariant representation learning.

Alternatively, we can consider domain-invariant shape- based features in such continually-changing environments to make object recognition more robust to illumination, context, color, and texture variations (we still expect these features to be important though). Topological data analysis (TDA) focuses on extracting such shape

information from high-dimensional data using algebraic and computational topology. In particular, persistent homology has been widely used to extract topologically persistent features for machine learning tasks [18], especially, computer vision [19]–[21]. Reininghaus et al. [19] propose a new stable representation that suits learning tasks, known as persistence images (PIs), and demonstrate its use for 3D shape classification/retrieval and texture recognition. Guo et al. [20] use sparsely sampled PIs for human posture recognition and texture classification. Garin et al. [21] use persistent features from different filtra- tion functions and representations to classify hand-written digits. Generating approximate PIs using a deep neural network has also been explored for image classification [22]. In this work, we propose the use of topologically persis- tent features for object recognition in indoor environments. Particularly, the key contributions of our work are as follows:

- We propose two kinds of topologically persistent features, namely, sparse PI and amplitude, for object recognition.
- We present a new dataset, the UW Indoor Scenes dataset, to evaluate the robustness of object recognition methods on unseen environments.
- We show that recognition using topologically persistent features is more robust to changing environments than a state-of-the-art cross-domain object detection model.
- We demonstrate that sparse PI features have better recognition performance than features from deep learning-based recognition methods and lead to better performance than end-to-end object detection methods (in terms of accuracy and recall) in unseen environments.

We also successfully implement the proposed framework on a real-world robot.

II. MATHEMATICAL PRELIMINARIES

We begin by covering some of the mathematical preliminaries associated with TDA.

A. Cubical Complexes

For TDA, data is often represented by cubical or simplicial complexes depending on the type of data. Images can be considered as point clouds by treating every pixel as a point in R2. Such a point cloud is commonly represented using a simplicial complex. However, since images are made up of pixels, they have a natural grid-like structure to them. Therefore, they are more efficiently represented as cubical complexes in various ways [21], [23].

A cubical complex in Rn is a finite set of elementary cubes aligned on the grid Zn, where an elementary cube is a finite product of elementary intervals with dimension given by the number of its non-degenerate components [24]. An n-dimensional image is a map   : I   Zn   R. A voxel is an element v   I, and its value   (v) is the intensity. When n = 2, the voxel is known as a pixel, and the intensity is known as the grayscale value. While there are various ways of constructing a cubical complex from an n-dimensional image, Garin et al. [21] adopt a method in which an n-cube represents a voxel, and all the adjacent lower-dimensional cubes (faces of the n-cube) are included. The values of the voxels v are extended to all the cubes λ in the resulting cubical complex C as

$$I'(\lambda) := \min_{\lambda \ \text{is face of} \ \hat{v}} I(v). \qquad (1)$$

After the complexes are generated, a filtration is constructed, as described next.

B. Filtration

For any cubical (or, simplicial) complex K, let Ki denote the i-th sublevel set of K. A filtration is then defined as a collection of complexes {Ks}s∈R such that Ks is a subcomplex of Kt, for each s ≤ t.

The pixel values in a grayscale image embed a natural filtration, which are used to obtain the sublevel sets of the corresponding cubical complex. The i-th sublevel set of the cubical complex C is then obtained as

$$C_i := \{\lambda \in C \mid I'(\lambda) \le i\} , \quad (2)$$

which defines the filtration as Ci i . For binary images, various functions known as descriptor functions (or, filtration functions) are used to construct such grayscale images [21]. Similarly, filtration functions are also defined for generating filtrations from point cloud data or mesh data [19].

C. Persistent Homology

Persistent homology is a common tool in TDA that studies the topological changes of the sublevel sets Ci as i increases from (,). During filtration, topological features, interpreted as h-dimensional holes, appear and dis- appear at different scales, referred to as their birth and death times, respectively. This information is summarized in an h-dimensional persistence diagram (PD). An h-dimensional PD is a countable multiset of points in R2. Each point (x, y) represents an h-dimensional hole born at a time x and filled at a time y. The diagonal of a PD is a multiset

$\Delta = (x, x) \quad R2 \, x \quad R$, where every point in Δ has infinite multiplicity.

Several other stable representations of persistence can be obtained from a PD [25], [26]. One such representation is the Persistence Image (PI), a stable and finite dimensional vector representation of persistent homology [26]. To obtain a PI, an equivalent diagram of birth-persistence points, i.e., (x, y x), is computed. The birth-persistence points are then regarded as a sum of Dirac delta functions, which are convolved with a Gaussian kernel over a rectangular grid of evenly sampled points to compute the PI.

III. METHODS

Given an RGB scene image, our goal is to recognize all the objects in the scene, based on their shape information that are captured using topologically persistent features. We first generate segmentation maps for the objects using a deep neural network, as explained in Section III-A. We then extract topologically persistent features from the object segmentation maps, as described in III-B. These features are then fed to a fully connected network for recognition. Fig. 1 illustrates the proposed framework.

A. Object Segmentation Map Generation

To generate the object segmentation maps for an input scene image, we follow a foreground segmentation method that is similar to the one proposed in [27]. In particular, we use the state-of-the-art DeepLabv3+ architecture [28], which is pre-trained on a large number of classes and is hypothesized to have a strong representation of 'objectness'. Subsequently, we train the network using pixel-level foreground annotations for a limited number of images from our datasets.
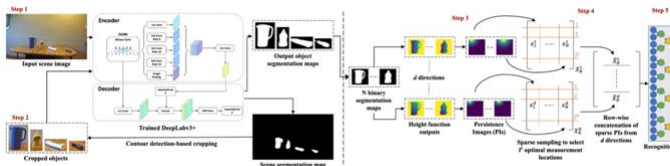

Fig. 1: Proposed framework for object recognition using topologically persistent features

A shape-based object recognition method relies on the objects' contours for distinguishing among multiple objects. However, the number of foreground pixels is very low as compared to the background when segmentation is performed on images taken from distances as large as two meters. Hence, it is difficult for a segmentation model to capture the

minor details in the objects' shapes in a single shot. Therefore, we employ a two-step segmentation frame- work to preserve the objects' contours details. In the first step, the segmentation model predicts a segmentation map for the input scene image. Contour detection is performed on this scene segmentation map to obtain the bounding boxes for the objects. These bounding boxes are used to divide the scene image into multiple sub-images, each of which contains a single object. In step two, these sub-images are fed to the same trained segmentation model for predicting the individual object segmentation maps.

B. Persistent Features Generation

Segmentation maps are essentially binary images com- prising only black and white voxels. A grayscale image, suitable for building a filtration of cubical complexes, can be generated from such binary images using various filtration functions. We select a commonly used filtration function, known as the height function, which computes a sufficient statistic to uniquely represent shapes in R2 and surfaces R3 in the form of the persistent homology transform [29].

For the cubical complexes in our case, we define the height function as stated in [21]. Consider a binary object segmentation map $B : I \subseteq Z^n \longrightarrow \{0, 1\}$, a grayscale image

$H : I \longrightarrow R$ for the segmentation map I, and a direction $p \in R^n$ of unit norm. The values of all the voxels of H are then reassigned as

$$H(v) := \begin{cases} \langle v, p \rangle & \text{if } B(v) = 1 \\ H_\infty & \text{if } B(v) = 0. \end{cases}$$

Here, $\langle v, p \rangle$ is the distance of voxel v from the hyperplane defined by p and H_∞ is the filtration value of the voxel that is farthest away from the hyperplane.

In step three, we obtain d such grayscale images for each object segmentation map by considering d directions that are evenly spaced on a unit 1-sphere S1. We construct cubical complexes from each grayscale image according to Eq. (1). The sublevel sets of the d complexes are then computed according to Eq. (2) to obtain d filtrations. Persistent homology is applied to these filtrations to obtain d persistence diagrams (PDs) for every object segmentation map. We only consider 0th order homology for generating the PDs. We investigate the performance of two types of persistent features from the generated PDs. The following subsections III-B.1 and III-B.2 describe their computation details.

1) Sparse persistence image features: Since the number of points in a PD varies from shape to shape, such a representation is not suitable for machine learning tasks. Instead, we use the persistence image (PI) representation to

generate suitable features for training the recognition network. However, only a few key pixel locations of the PIs, which contain nonzero entries, sparsely encode topo- logical information. Therefore, we adopt QR-pivoting based sparse sampling to obtain a Sparse PI, as proposed in [20]. For every object segmentation mask, d PIs are generated from their corresponding PDs. Fig. 2 shows sample PIs for two different objects using height functions in multiple directions, illustrating the (collective) presence of sufficient discriminative information. In step four, for every direction pk, k   1, . . . , d , the corresponding PIs for all the N training (binary) object segmentation maps are vectorized and arranged as columns of a large matrix Xk .

The dominant PI patterns Ulk are obtained by computing the truncated singular value decomposition of Xk as

$$Xk \approx U \; \Sigma \; (V \;) \; , \qquad (4)$$

where lk is the optimal singular value threshold [30] for the pkth direction PIs. Ulk is then discretely sampled using the pivoted QR factorization as

$$UT \; (\Pi k)T \; = QkRk. \qquad (5)$$

 is farthest away from the hyperplane.

In step three, we obtain d such grayscale images for each object segmentation map by considering d directions that are sparsely sampled PIs. We finally perform row wise concate- nation of all the d sparsely sampled PI matrices to generate the overall set of features for recognition in step five.
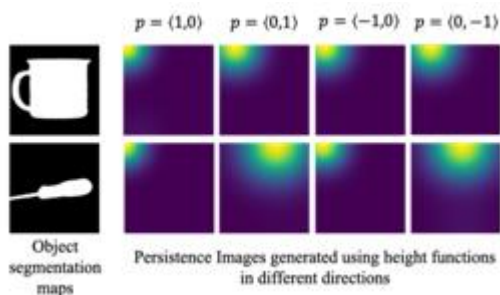

Fig. 2: Persistence images (PIs) for two sample objects. The collection of PIs, computed using height functions in different directions on object segmentation maps, have enough information to distinguish between the two objects.
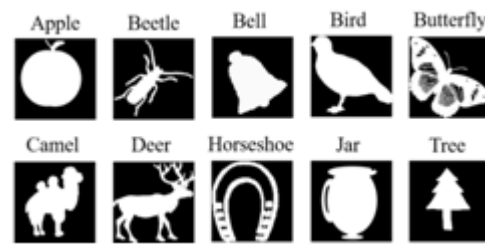

Fig. 3: Sample images from the MPEG-7 Shape Silhoutte Dataset

2) Amplitude features: An alternative method of generat- ing topologically persistent features is using the amplitude or the distance of a given PD from an empty diagram. For each of the d generated PDs corresponding to an object segmentation map, we compute the bottleneck amplitude [21], Ak, as

$$A_b^k = \frac{1}{-} \sup_b (y_b^k - x_b^k), \qquad (6)$$

where (x_b^k, y_b^k) are all the non-diagonal points in the ρ_k-th direction PD. These amplitudes are stacked to form a d-dimensional feature vector. Such d-dimensional feature vectors are generated for all the N object segmentation maps to train the recognition network.

IV. DATASETS

A. MPEG-7 Shape Silhouette Dataset

We first choose the widely used MPEG-7 Shape Silhouette dataset to solely characterize the shape recognition capability of the two persistent features-based networks for (almost) ideal object segmentation maps. In particular, we use a subset of this dataset, namely, the MPEG-7 CE Shape 1 Part B dataset, which is specifically designed to evaluate the performance of 2D shape descriptors for similarity-based image retrieval [31]. It includes the shapes of 70 different classes and 20 images for each class, for a total of 1,400 images. Fig. 3 shows sample images from the dataset.

B. RGB-D Scenes Dataset v1

The shapes in the MPEG-7 are detailed and fairly dis- tinguishable from each other. However, common objects in indoor environments are often less detailed and, therefore, more challenging for topological methods. Most deep learn- ing models work exceptionally well in recognizing such everyday objects in their training environments. However, they face challenges when used in new (previously unseen) environments without any retraining, even if the objects remain the same. Therefore, we choose the widely used benchmark, the RGB-D Scene Dataset v1 [32], to evaluate the performance of our proposed methods. The dataset consists of

eight scene setups with objects that belong to the RGB-D Object Dataset [32]. The scenes are shot in five different environments: desk, kitchen small, meeting small, table, and table small. In particular, we select the table small and desk environments for our evaluation. Among all the objects present in the scenes, we consider six object classes for recognition corresponding to the six objects types that appear in both the environments. For our current analysis, we only consider RGB images where the objects are not occluded.

C. UW Indoor Scenes Dataset

While the RGB-D Scenes Dataset consists of scenes from multiple environments, all of them are tabletop environments with limited variation in terms of lighting conditions and object types. On the other hand, the dataset in [33] provides images with several objects but only in a single environ- ment. Although there are quite a few other indoor scene datasets in the literature, none of them includes a large enough set of object types, poses, and arrangements with varying backgrounds and lighting conditions. Therefore, we introduce a new RGB dataset, which we call the UW Indoor performance in multiple, typical indoor environments. For this purpose, we select a fixed set of fourteen different objects from the benchmark Yale-CMU-Berkeley (YCB) object and model set [34].

The UW-IS dataset consists of indoor scenes taken in two completely different environments. The first environment is a living room scene where the objects are placed on a tabletop. The second environment is a mock warehouse setup where the objects are placed on a shelf. For the living room environment, we have a total of 347 scene images. The images are taken in four different illumination settings from three different camera perspectives and varying distances up to two meters. Sixteen out of the 347 scene images are with two different objects, 135 images are with three different objects, 156 images are with four different objects, and 40 images are with five different objects. For the mock warehouse environment, we have a total of 200 scene images taken from distances up to 1.6 meters. Sixty out of 200 images are images with three different objects, 68 images with four different objects, and 72 images are with five different objects. Fig. 4a shows some sample living room scene images, and Fig. 4b shows sample images from the mock warehouse environment. Fig. 4c shows all the fourteen objects used in our dataset. The dataset is

(a) Living room environment

(b) Mock warehouse environment

(c) Objects

Fig. 4: Representative images from the UW Indoor Scenes Dataset

publicly available at https://data.mendeley.com/datasets/dxzf29ttyh/.

V. EXPERIMENTS

A. Implementation Details

We perform five-fold training and testing on all images of the MPEG-7 Shape Dataset. For evaluation on the RGB- D Scenes dataset, we use the table small environment for both training and testing, and the desk environment only for testing. For the UW-IS dataset, we use the living room images for training and testing, and the mock warehouse images only for testing. All the training and testing is performed using GeForce GTX 1080 and 1080 Ti GPUs on workstations running Windows 10 and Ubuntu 18.04 LTS, respectively. The code for the proposed methods is available at https://github.com/smartslab/objectRecognitionTopologicalFeatures.

For the MPEG-7 Shape Dataset, we divide the 1,400 images into five sets of 280 images each, with four images of each class included in every set. We perform five-fold training and testing using these sets, such that each set is used once as a test set while the remaining four sets are used for training and validation. We use the giotto-tda [35] library to generate the PDs and the Persim package in Scikit-TDA Toolbox to

generate the PIs. The PDs are generated using height functions along 8 evenly spaced directions on S1. We choose a grid size equal to 50   50, a spread of 10, and a linear weighting function for generating the PIs. We use three-layered and five-layered fully connected networks for recognition using amplitude features and sparse PI features,

respectively. We use the ReLU activation (last layer uses softmax activation), Adam optimizer and categorical cross-entropy loss function for training. The learning rate is set to 0.01 for the first 500 epochs. It is decreased by a factor of 10 after every 100 epochs for the next 200 epochs, and by a factor of 100 for the last 100 epochs.

For the RGB-D Scenes dataset and the UW-IS dataset, we use the DeepLabv3+ segmentation model with Xception- 65 backbone following the implementation in [36] to obtain object segmentation maps. The network is initialized using a model pre-trained on the ImageNet and PASCAL VOC2012 datasets available from [36]. We use 147 images from the table small environment to train the network for the RGB- D Scenes dataset and 200 living room images to train the network for the UW-IS dataset. Horizontally flipped counterparts of the images are also used for training the models. We generate the corresponding foreground annotations using LabelMe [37]. We train the network for 20,000 steps using the categorical cross-entropy loss with 10% hard example mining after 2,500 steps in the case of the RGB-D Scenes dataset. For the UW-IS dataset, we use 1% hard example mining1.

We then perform five-fold training and testing of the proposed methods on both datasets by diving the scene images from the training environment into five folds. The resulting five models are also tested separately on all the im- ages of the test environment. First, the trained DeepLabv3+ segmentation models are used to obtain object segmenta- tion maps from all the images. To ensure the quality of segmentation maps remain consistent across training and test environments, we generate object segmentation maps for the test environments using segmentation models fine- tuned in these environments. We then appropriately pad all the segmentation maps with zeros and consistently resize them to obtain 125 125 binary images. We augment the training data by rotating every object segmentation map by 90o, 180o, and 270o. The PDs and PIs are generated in the same manner as for the MPEG-7 dataset, except that the spread value is chosen to be 20. We also use the same fully connected network architectures and hyperparameters as for the MPEG-7 dataset.

Comparison with deep learning-based object recogni-tion methods: To solely characterize the recognition perfor-

mance of the topologically persistent features, we compare their performance with features extracted using two other widely used object recognition methods, namely, ResNetV2-56 [1] and EfficientNet-B4 [2], on the RGB-D Scenes benchmark. We use the same sets of cropped objects obtained using DeepLabv3+ in Step 1 of our proposed method for five-fold training and testing of both the methods. The networks for both the methods are trained using the implementations available in Keras [38].

1Unlike the UW-IS dataset, the smaller training set for the RGB-D Scenes dataset leads to poor quality object segmentation maps that are unsuitable for extracting topological features. Therefore, we use a separate DeepLabv3+ model for Step 2, which is fine-tuned on images of individual objects obtained from the original scene images

Comparison with end-to-end object detection meth-ods: We also compare the overall performance of both the persistent features-based methods against end-to-end object detection methods on both the RGB-D Scenes dataset and the UW-IS dataset. Particularly, we compare performance against Faster R-CNN [3], a widely used object detection method, and its state-of-the-art variant for cross-domain object detection, Domain Adaptive Faster R-CNN [12]. For Faster R-CNN (referred to as FR-CNN), we use a pre- trained model with InceptionResNet-V2 feature extractor and hyperparameters available with the TensorFlow Object Detection API [39]. For Domain Adaptive Faster R-CNN, we use a modified implementation from [40], where the VGG backbone is replaced with ResNet-50, and the RoI-pooling layer is replaced with the more popular RoIAlign. The mod- ified implementation has been shown to outperform other state-of-the-art methods for cross-domain object detection [40]. We call this improved method DA-FR-CNN*. Similar to the proposed methods, we perform five-fold training and testing of these methods on both datasets. In the case of DA-FR-CNN* for the UW-IS dataset, all the training set images with artificial lighting (e.g., bottom row in Fig. 4a) are used as the source domain, while all the images with natural lighting (e.g., top row in Fig. 4a) are used as the target domain. Since such a split is not possible in the RGB- D Scenes dataset, we randomly divide the images into the source and target domains. The ground truth bounding boxes for both the datasets are generated using LabelImg [41].

B. Results

We first examine the recognition performance of both amplitude features and sparse PI features on the MPEG-7 dataset. We use the weighted F1 score, weighted precision, weighted recall, and accuracy for evaluating performance.

Table I shows the test-time performance of the trained recognition networks. We observe that sparse PI-based recognition performance is quite impressive (more than 0.85) with respect to all the four metrics. It is also consistently better than amplitude-based recognition, indicating the usefulness of sparse sampling in selecting the key features. It is worth noting that 100% recognition performance using 2D shape knowledge is not possible for this dataset, since some of the classes contain shapes that are significantly different from the others in the same class but are similar to certain shapes in other classes [31].

TABLE I: Performance comparison of amplitude and sparse PI features (best in bold) on the MPEG Shape Silhouette dataset. (w) indicates weighted metric.

	Amplitude	Sparse PI
F1 score (w)	0.75±0.01	**0.87±0.01**
Precision (w)	0.77±0.01	**0.89±0.02**
Recall (w)	0.76±0.01	**0.87±0.01**
Accuracy	0.76±0.01	**0.87±0.01**

We then examine the performance[2] of both the persistent features-based methods along with other object recognition and detection methods on the RGB-D Scenes benchmark. Table II reports the performance of all the methods (trained in the table small environment) on the unseen desk en- vironment. We observe that sparse PI-based recognition, which achieves an overall accuracy of 0.71 0.01, has the highest performance among all the methods in terms of recall and accuracy. Particularly, for the same set of object images obtained using the DeepLabv3+ model, recognition using sparse PI features computed from object segmentation maps is better than recognition using features learned by ResNetV2-56 and EfficientNet-B4 from RGB inputs with respect to accuracy, recall, and F1-score. Additionally, amplitude features-based recognition is comparable to both ResNetV2-56 and EfficientNet-B4. Additionally, the difference between the recognition performance using sparse PI features and amplitude features is, however, much lower than that for the MPEG-7 dataset. We also note that the sparse PI features-based method outperforms Faster R-CNN (referred to as FR-CNN) substantially and DA-FR-CNN* by a small margin in terms of accuracy and recall.

We then assess the performance of both the persistent features-based methods, FR-CNN, and DA-FR-CNN* on the living room scene images from the UW-IS dataset reported in Table III. We observe that both the persistent features- based methods, which only use the information in object segmentation maps, perform reasonably well. Sparse PI- based recognition achieves an overall accuracy of 0.71 0.01 and is marginally better than amplitude-

based recognition, whose accuracy is 0.69 0.01. Moreover, both FR-CNN and DA-FR-CNN*, which are trained on this environment and use RGB images as inputs, outperform these methods with respect to all the metrics including class-wise F1 scores.

We believe that the quality of the object segmentation maps has substantial impact on the performance of the persistent features-based methods. Notably, performance is considerably worse for the fork as compared to the other objects. To assess this impact, we compare the recognition performance of both these methods with that of a human given an identical set of segmentation maps. Table IV sum- marizes the comparison results. For the persistent features based-methods, we only report the accuracy for those images where the human recognizes the object correctly. We observe that a human achieves an accuracy of 0.84 0.01, which is lower that FR-CNN performance, and finds it difficult to recognize the objects based on the generated segmentation maps, especially for the spoon and fork classes. We refer the reader to Section VI for further discussion on segmentation map quality.

Despite this challenge, we expect recognition performance to be relatively unaffected, when the environments vary con- siderably but the objects are identical, provided the quality

[2]We do not account for the false negatives and false positives resulting from errors of the segmentation model to ensure a fair judgement of our methods' effectiveness. Equivalently, for object detection methods, false positives arising from incorrect region proposals are also not considered.

TABLE II: Performance comparison of the proposed persistent features-based methods with ResNetV2-56, EfficientNet-B4, FR-CNN, and DA-FR-CNN* (best in bold) on the desk images from the RGB-D Scenes dataset.

Metric	Class	Amplitude	Sparse PI	ResNetV2-56	EfficientNet-B4	FR-CNN	DA-FR-CNN*
F1 score	Bowl Cap	0.71±0.01	0.69±0.01	0.95±0.01	0.90±0.03	0.83±0.02	0.88±0.01
	Cereal box Cup	0.80±0.01	0.88±0.01	0.76±0.03	0.83±0.03	0.65±0.01	0.79±0.01
	Soda can Stapler	0.76±0.01	0.66±0.01	0.30±0.05	0.29±0.04	0.62±0.01	0.51±0.03
		0.70±0.01	0.71±0.01	0.88±0.03	0.84±0.05	0.69±0.03	0.78±0.01
F1 score (w)	-	0.70±0.00	0.71±0.01	0.65±0.01	0.65±0.01	0.70±0.01	**0.73±0.01**
Precision (w)	-	0.75±0.01	0.74±0.00	0.79±0.01	0.77±0.01	0.82±0.01	**0.87±0.01**
Recall (w)	-	0.68±0.00	**0.71±0.01**	0.68±0.01	0.69±0.01	0.66±0.01	0.69±0.01
Accuracy	-	0.68±0.00	**0.71±0.01**	0.68±0.01	0.69±0.01	0.66±0.01	0.69±0.01

TABLE III: Performance comparison of the proposed persistent features-based methods with FR-CNN and DA-FR-CNN* (best in bold) on the living room images from the UW-IS dataset.

Metric	Class	Amplitude	Sparse PI	FR-CNN	DA-FR-CNN*
F1 score	Spoon	0.62±0.04	0.57±0.01	0.84±0.03	0.68±0.02
	Fork	0.15±0.05	0.81±0.07	0.81±0.02	0.32±0.06
	Plate	0.83±0.02	0.89±0.01	0.98±0.01	0.95±0.02
	Bowl	0.95±0.01	0.94±0.02	0.98±0.01	0.97±0.01
	Cup	0.67±0.02	0.78±0.04	0.91±0.01	1.00±0.00
	Pitcher base	0.81±0.03	0.87±0.02	0.92±0.02	0.98±0.01
	Bleach cleanser	0.73±0.02	0.71±0.02	0.87±0.03	0.97±0.01
	Mustard bottle	0.63±0.02	0.68±0.03	0.84±0.03	0.98±0.01
	Soup can	0.68±0.04	0.69±0.02	0.84±0.04	0.86±0.02
	Chips can	0.72±0.02	0.75±0.02	0.92±0.03	0.90±0.03
	Meat can	0.55±0.05	0.56±0.04	0.82±0.03	0.88±0.02
	Gelatin box	0.55±0.02	0.64±0.04	0.91±0.02	0.91±0.01
	Screwdriver	0.58±0.04	0.72±0.04	0.91±0.02	0.94±0.02
	Padlock	0.74±0.04	0.72±0.03	0.97±0.02	0.93±0.03
F1 score (w)	-	0.68±0.01	0.71±0.01	**0.89±0.01**	0.88±0.01
Precision (w)	-	0.69±0.01	0.71±0.02	**0.91±0.01**	0.90±0.01
Recall (w)	-	0.69±0.01	0.71±0.01	0.88±0.01	**0.88±0.00**
Accuracy	-	0.69±0.01	0.72±0.01	**0.88±0.01**	0.88±0.00

TABLE IV: Comparison of recognition accuracy of the proposed persistent features-based methods with a human given the object segmentation maps for living room scene images from the UW-IS dataset

Class	Human performance	Amplitude	Sparse PI
Spoon	0.37±0.07	0.44±0.05	0.47±0.09
Fork	0.40±0.11	0.16±0.08	0.12±0.07
Plate	0.96±0.02	0.90±0.03	0.90±0.04
Bowl	0.97±0.01	0.96±0.01	0.97±0.01
Cup	0.92±0.03	0.65±0.05	0.81±0.03
Pitcher base	0.97±0.01	0.87±0.04	0.92±0.02
Bleach cleanser	0.91±0.01	0.86±0.02	0.77±0.02
Mustard bottle	0.91±0.03	0.63±0.04	0.73±0.04
Soup can	0.76±0.05	0.83±0.06	0.81±0.06
Chips can	0.89±0.04	0.81±0.04	0.74±0.07
Meat can	0.86±0.03	0.55±0.07	0.61±0.03
Gelatin box	0.88±0.04	0.59±0.03	0.56±0.07
Screwdriver	0.83±0.03	0.63±0.06	0.74±0.06
Padlock	0.85±0.04	0.81±0.05	0.82±0.03
Accuracy	0.84±0.01	0.74±0.01	0.76±0.01

of the segmentation maps remains consistent. Accordingly, we test the performance of the persistent features-based methods on all the warehouse scene images of the UW- IS dataset without retraining the recognition networks. We also test the performance of FR-CNN and DA-FR-CNN* on all the warehouse images without any retraining. Table V summarizes the performances of all the four methods on the mock warehouse test environment. We observe that the performance of sparse PI-based recognition is unchanged from the living room scenario, and is substantially better than that of amplitude-based recognition, whose accuracy reduces by 4%. However, the performance of FR-CNN degrades a lot without fine-tuning (accuracy drops by 25%). The performance of DA-FR-CNN* also degrades considerably (accuracy drops by 18%). Similar to the RGB-D Scenes benchmark case, the sparse PI features-based method outperforms FR-CNN substantially and DA-FR-CNN* by a small margin with respect to accuracy and recall.

C. Robot Implementation

We also implement our proposed framework on a LoCoBot platform built on a Yujin Robot Kobuki Base (YMR-K01-W1) and powered by an Intel NUC NUC7i5BNH Mini PC. We mount a ZED2 camera with stereo vision on top of the LoCoBot and control the robot using the PyRobot interface [42]. The camera images are fed to the trained segmentation model and recognition networks, which are run on an on- board NVIDIA Jetson AGX Xavier processor, equipped with a 512-core Volta GPU with Tensor Cores and 8-core ARM v8.2 64-bit CPU. We use TensorRT [43] for optimizing the trained segmentation model. Fig. 5 shows a screenshot of the platform. The sparse PI features-based recognition runs at a speed of 1.1s per frame on this platform. A video demonstration of object recognition on this platform is included in the Supplementary Materials.

VI. DISCUSSION

We note that segmentation map quality has a considerable impact on the performance of the persistent features-based methods. For example, the performance is particularly bad for the fork and spoon classes, as observed from Tables III and V. The inherent similarity between forks and spoons, along with low segmentation quality, often makes it hard to distinguish between them, as shown in Fig. 6a. Table

TABLE V: Performance comparisons of the proposed persistent features-based methods with FR-CNN and DA-FR-CNN* (best in bold) on mock warehouse scene images from the UW-IS dataset.

Metric	Class	Amplitude	Sparse PI	FR-CNN	DA-FR-CNN*
F1 score	Spoon	0.44±0.04	0.52±0.02	0.39±0.03	0.32±0.05
	Fork	0.28±0.04	0.20±0.04	0.24±0.03	0.00±0.00
	Plate	0.16±0.07	0.84±0.02	0.27±0.16	0.81±0.01
	Bowl	0.94±0.00	0.91±0.01	0.97±0.01	0.97±0.00
	Cup	0.51±0.02	0.90±0.01	0.94±0.01	0.92±0.00
	Pitcher base	0.86±0.01	0.94±0.00	0.93±0.03	1.00±0.00
	Bleach cleanser	0.79±0.02	0.78±0.01	0.84±0.04	0.96±0.01
	Mustard bottle	0.76±0.02	0.74±0.01	0.36±0.05	1.00±0.00
	Soup can	0.73±0.02	0.66±0.02	0.85±0.02	0.81±0.01
	Chips can	0.76±0.01	0.74±0.01	0.71±0.03	0.87±0.02
	Meat can	0.58±0.09	0.71±0.01	0.75±0.05	0.74±0.04
	Gelatin box	0.23±0.04	0.25±0.02	0.70±0.03	0.70±0.03
	Screwdriver	0.78±0.02	0.72±0.01	0.42±0.04	0.76±0.03
	Padlock	0.61±0.03	0.64±0.01	0.89±0.03	0.60±0.04
F1 score (w)	-	0.65±0.01	0.70±0.00	0.65±0.01	**0.75±0.01**
Precision (w)	-	0.67±0.01	0.70±0.01	0.77±0.02	**0.87±0.01**
Recall (w)	-	0.66±0.01	**0.71±0.01**	0.63±0.01	0.70±0.01
Accuracy	-	0.66±0.01	0.73±0.01	0.63±0.01	0.70±0.01

Fig. 5: Screenshot of the LoCoBot operating in a mock warehouse setup

IV shows that even humans have difficulty distinguishing between them from the generated segmentation maps. Both Faster R-CNN (referred to as FR-CNN) and DA-FR-CNN* also have difficulties with these classes, especially in the unseen warehouse environment. On a related note, incom- plete segmentation maps also affect performance of the proposed methods. For example, the shape in the incomplete segmentation map of a padlock shown in Fig. 6b is quite different from that of a padlock, making it difficult for a topology-based method to recognize the object. Moreover, we observe from Table IV that our methods only achieve 76% and 74% of human performance using sparse PI and amplitude features, respectively. We believe this difference is primarily due to humans' additional cognitive capability to complete (partially visible) shapes.

Additionally, we observe relatively small variations in the class-wise performance of the proposed methods between the living room and mock warehouse environments except for the plate and gelatin box classes. This observation can be attributed to changes in camera locations and viewing angles that lead to unseen object poses and naturally occur-ring variations in object placements. Fig. 6c illustrates this
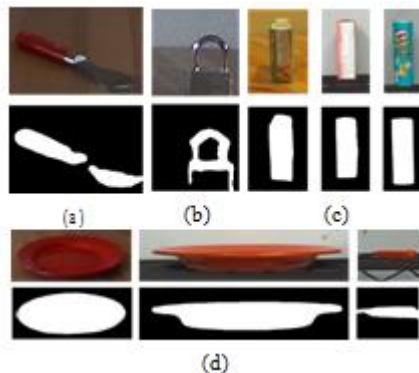

Fig. 6: Sample failure cases. Fig. 6a shows a case where a poorly segmented fork is confused with a spoon. Fig. 6b shows a case where, unlike the proposed topological methods, a human succeeds by being able to complete the shape. Fig. 6c and 6d illustrate that camera pose changes cause some of the objects to appear similar to other objects.

problem for the gelatin box. The leftmost image is from the living room. The middle image shows the same object pose captured from a different camera viewing angle in the warehouse. The 2D shape in the middle image becomes very similar to that of the chips can in the rightmost image. Similarly, Fig. 6d shows how such a change results in a completely different 2D shape for the plate, which is similar to a half-visible spoon. The performance of FR- CNN is also affected by such variations in camera pose, as observed for the plate class in Table V. On the other hand, the performance of

DA-FR-CNN* drops due to changes in object appearance. For example, the cups in the desk and table small environments look considerably different, leading to poor recognition, as reported in Table II.

VII. CONCLUSIONS

In this letter, we propose the use of topologically persis- tent features for object recognition in indoor environments. We construct cubical complexes from binary segmentation maps of the objects. For every cubical complex, we ob- tain multiple filtrations using height functions in multiple directions. Persistent homology is applied to these filtrations to obtain topologically persistent features that capture the shape information of the objects. We present two different kinds of persistent features, namely, sparse PI and amplitude features, to train a fully connected recognition network. Sparse PI features achieve better recognition performance in unseen environments than features learned from ResNetV2- 56 and EfficientNet-B4. Unlike end-to-end object detection methods Faster R-CNN and its state-of-the-art variant DA- FR-CNN*, the overall performance of our methods remains relatively unaffected on a different test environment without retraining, provided the quality of the object segmentation maps is maintained. Moreover, the sparse PI features-based method slightly outperforms DA-FR-CNN* in terms of recall and accuracy, making it a promising first step in achieving robust object recognition. In the future, we plan to use depth information to deal with the challenges associated with camera pose variations and to perform class-agnostic instance segmentation. Depth information, along with shape completion, might also help address the issues of incom- plete segmentation maps and partial occlusion of objects in cluttered environments. We also plan to explore the use of topologically persistent features in estimating 6D object poses through few-shot deep learning.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in European Conf. Comput. Vis., 2016, pp. 630– 645.

[2] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Int. Conf. Mach. Learn., 2019, pp. 6105–6114.

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real- time object detection with region proposal networks," in Adv. Neural Inform. Process. Syst., 2015, pp. 91–99.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, andA. C. Berg, "SSD: Single shot multibox

detector," in European Conf. Comput. Vis., 2016, pp. 21–37.

[5] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 779–788.

[6] B. Cheng, Y. Wei, H. Shi, R. Feris, J. Xiong, and T. Huang, "Revisiting R-CNN: On awakening the classification power of Faster R-CNN," in European Conf. Comput. Vis., 2018, pp. 453–468.

[7] S. Thys, W. Van Ranst, and T. Goedeme´, "Fooling automated surveil- lance cameras: adversarial patches to attack person detection," in IEEE Conf. Comput. Vis. Pattern Recognit. Workshop, 2019, pp. 0–0.

[8] L. Kunze, N. Hawes, T. Duckett, M. Hanheide, and T. Krajn´ık, "Artificial intelligence for long-term robot autonomy: A survey," IEEE Rob. Autom. Lett., vol. 3, no. 4, pp. 4023–4030, 2018.

[9] S. H. Kasaei, M. Oliveira, G. H. Lim, L. S. Lopes, and A. M. Tome´, "Towards lifelong assistive robotics: A tight coupling between object perception and manipulation," Neurocomputing, vol. 291, pp. 151–166, 2018.

[10] C. Eriksen, A. Nicolai, and W. Smart, "Learning object classifiers with limited human supervision on a physical robot," in IEEE Int. Conf. Robotic Comput., 2018, pp. 282–287.

[11] W. Li, F. Li, Y. Luo, and P. Wang, "Deep domain adaptive object detection: a survey," arXiv preprint arXiv:2002.06797, 2020.

[12] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain Adaptive Faster R-CNN for object detection in the wild," in IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 3339–3348.

[13] Z. He and L. Zhang, "Multi-Adversarial Faster-RCNN for unrestricted object detection," in IEEE Int. Conf. Comput. Vis., 2019, pp. 6668– 6677.

[14] K. Saito, Y. Ushiku, T. Harada, and K. Saenko, "Strong-weak distribu- tion alignment for adaptive object detection," in IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 6956–6965.

[15] H.-K. Hsu, C.-H. Yao, Y.-H. Tsai, W.-C. Hung, H.-Y. Tseng, M. Singh, and M.-H. Yang, "Progressive domain adaptation for object detection," in IEEE Winter Conf. Appl. Comput. Vis., 2020, pp. 749–757.

[16] N. Inoue, R. Furuta, T. Yamasaki, and K. Aizawa, "Cross-domain weakly-supervised object detection through progressive domain adap- tation," in IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 5001– 5009.

[17] T. Kim, M. Jeong, S. Kim, S. Choi, and C. Kim, "Diversify and match: A domain adaptive representation learning paradigm for object detection," in IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 12 456–12 465.

[18] D. Pachauri, C. Hinrichs, M. K. Chung, S. C. Johnson, and V. Singh, "Topology-based kernels with application to inference problems in Alzheimer's disease," IEEE Trans. Med. Imaging, vol. 30, no. 10, pp. 1760–1770, 2011.

[19] J. Reininghaus, S. Huber, U. Bauer, and R. Kwitt, "A stable multi-scale kernel for topological machine learning," in IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 4741–4748.

[20] W. Guo, K. Manohar, S. L. Brunton, and A. G. Banerjee, "Sparse- TDA: Sparse realization of topological data analysis for multi-way classification," IEEE Trans. Knowl. Data Eng., vol. 30, no. 7, pp. 1403–1408, 2018.

[21] A. Garin and G. Tauzin, "A topological "reading" lesson: Classification of MNIST using TDA," in IEEE Int. Conf. Mach. Learn. Applicat., 2019, pp. 1551–1556.

[22] A. Som, H. Choi, K. Natesan Ramamurthy, M. P. Buman, and P. Turaga, "PI-Net: A deep learning approach to extract topological persistence images," in IEEE Conf. Comput. Vis. Pattern Recognit. Workshop, 2020, pp. 834–835.

[23] V. Robins, P. J. Wood, and A. P. Sheppard, "Theory and algorithms for constructing discrete Morse complexes from grayscale digital images," IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 8, pp. 1646–1658, 2011.

[24] T. Kaczynski, K. Mischaikow, and M. Mrozek, Computational homol- ogy. Springer Science & Business Media, 2006, vol. 157.

[25] P. Bubenik, "Statistical topological data analysis using persistence landscapes," J. Mach. Learn. Res., vol. 16, no. 1, pp. 77–102, 2015.

[26] H. Adams, T. Emerson, M. Kirby, R. Neville, C. Peterson, P. Shipman, S. Chepushtanova, E. Hanson, F. Motta, and L. Ziegelmeier, "Persis- tence images: A stable vector representation of persistent homology," J. Mach. Learn. Res., vol. 18, no. 1, pp. 218–252, 2017.

[27] B. Xiong, S. D. Jain, and K. Grauman, "Pixel objectness: learning to segment generic objects automatically in images and videos," IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 11, pp. 2677–2692, 2018.

[28] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in European Conf. Comput. Vis., 2018.

[29] K. Turner, S. Mukherjee, and D. M. Boyer, "Persistent homology trans- form for modeling shapes and surfaces," Information and Inference: A Journal of the IMA, vol. 3, no. 4, pp. 310–344, 2014.

[30] M. Gavish and D. L. Donoho, "The optimal hard threshold for singular values is 4/ 3," IEEE Trans. Inf. Theory, vol. 60, no. 8, pp. 5040– 5053, 2014.

[31] L. J. Latecki, R. Lakamper, and T. Eckhardt, "Shape descriptors for non-rigid shapes with a single closed contour," in IEEE Conf. Comput. Vis. Pattern Recognit., 2000, pp. 424–429.

[32] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi- view RGB-D object dataset," in IEEE Int. Conf. Rob. Autom., 2011, pp. 1817–1824.

[33] C. Rennie, R. Shome, K. E. Bekris, and A. F. De Souza, "A dataset for improved RGBD-based object detection and pose estimation for warehouse pick-and-place," IEEE Rob. Autom. Lett., vol. 1, no. 2, pp. 1179–1185, 2016.

[34] B. Calli, A. Walsman, A. Singh, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Benchmarking in manipulation research: Using the yale- CMU-Berkeley object and model set," IEEE Rob. Autom. Mag., vol. 22, no. 3, pp. 36–52, 2015.

[35] G. Tauzin, U. Lupo, L. Tunstall, J. B. Pe´rez, M. Caorsi, A. Medina- Mardones, A. Dassatti, and K. Hess, "giotto-tda: A topological data analysis toolkit for machine learning and data exploration," arXiv preprint arXiv:2004.02551, 2020.

[36] Tensorflow, "Tensorflow models." [Online]. Available: https://github. com/tensorflow/models/tree/master/research/deeplab

[37] Wkentaro, "Github - wkentaro/labelme," 2019. [Online]. Available: https://github.com/wkentaro/labelme

[38] Keras. [Online]. Available: https://keras.io/

[39] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, Fischer, Z. Wojna, Y. Song, S. Guadarrama, et al., "Speed/accuracy trade-offs for modern convolutional object detectors," in IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 7310–7311.

[40] Krumo, "Domain adaptive Faster R-CNN." [Online]. Available: https://github.com/krumo/Domain-Adaptive-Faster-RCNN-PyTorch

[41] Tzutalin, "LabelImg," 2015. [Online]. Available: https://github.com/ tzutalin/labelImg

[42] A. Murali, T. Chen, K. V. Alwala, D. Gandhi, L. Pinto, S. Gupta, andA. Gupta, "PyRobot: An open-source robotics framework for research and benchmarking," arXiv preprint arXiv:1906.08236, 2019.

[43] Nvidia, "TensorRT." [Online]. Available: https://github.com/NVIDIA/ TensorRT