Supplementary Material - Tailoring Visual Object Representations to Human Requirements: A Case Study of a Recycling Robot

Debasmita Ghose, Michal Adam Lewkowicz, Kaleb Gezahegn, Julian Lee,*
Timothy Adamson,* Marynel Vázquez, Brian Scassellati
Yale University
{first name}.{last name}@yale.edu

1 Additional Details of the Case Study with a Recycling Robot

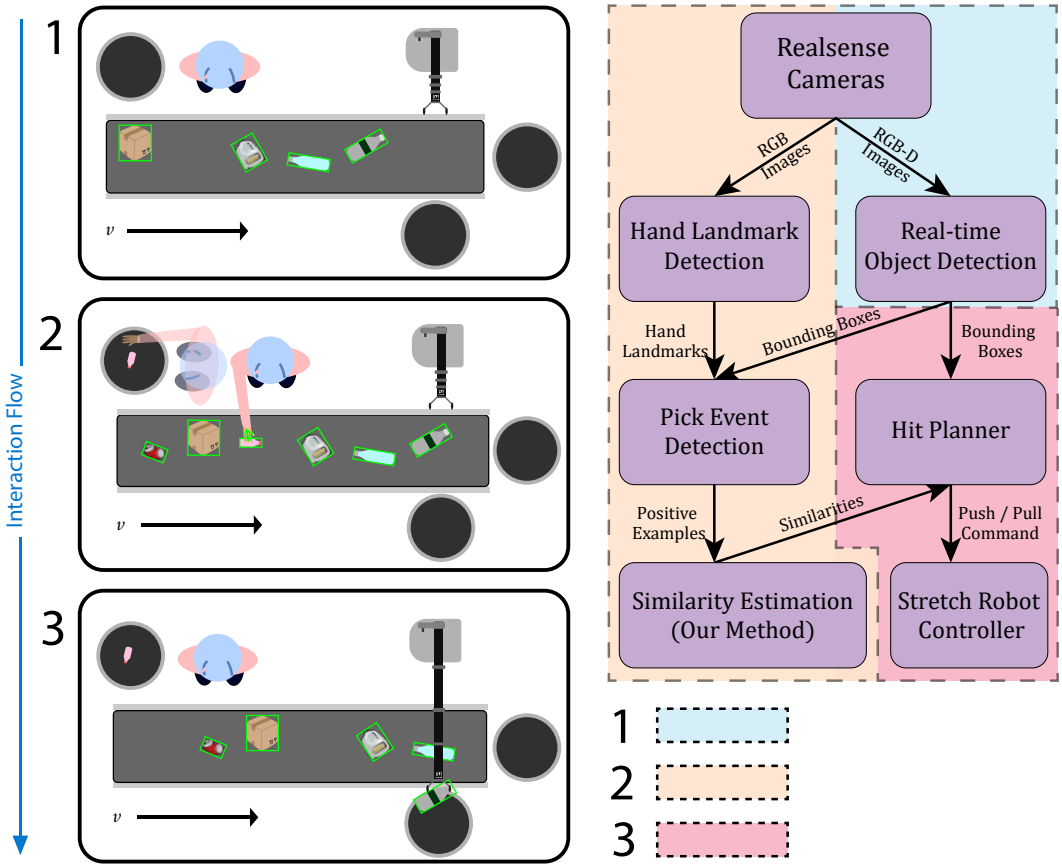


Figure 1: **Interaction Flow with System Architecture:** In our interaction flow, a human selects some items from a given category from a conveyor belt. The system estimates the similarity of each remaining item on the conveyor belt. Finally, the robot selects objects most similar to the human-selected objects and sends a push/pull command to the robot. Each colored zone (and its corresponding number code) shows the system components deployed during different phases of the interaction flow.

*Authors contributed equally

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.

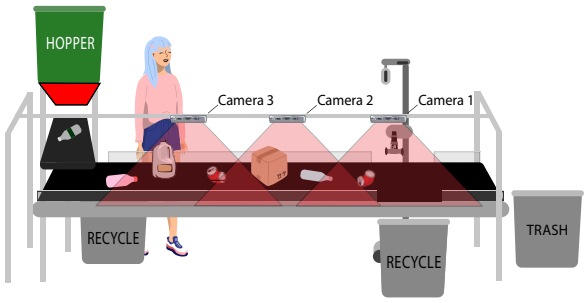


Figure 2: Experimental Setup with Field of View of each camera highlighted

This section explains in detail components of our system architecture. Figure 2 shows our experimental setup highlighting the field of view of our three cameras. Figure 1 depicts the interaction flow of our experimental setup in conjunction with the components of our system.

1.1 Custom End-Effector Attachment Design for the Stretch RE-1 robot

The Stretch RE-1 robot [1] extracts items from the conveyor belt by either extending its telescopic arm to push objects off the conveyor belt or pulling objects towards itself. The default gripper of the robot was not appropriate for removing recyclables from the conveyor belt because of the slow nature of the servo motors used for engaging the grippers. Therefore, we designed and built a custom static attachment to the end of the telescoping arm of the Stretch RE-1 robot (as shown in Figure 3) that allowed the robot to push or pull items from the conveyor belt. Our custom end-effector attachment consists of a base plate with a divider across the middle of two concave halves to help guide targets to the pull or push action and is mounted onto the threaded shaft for the removed gripper. It is supported by a cover that slides onto the servo motor on the arm and also clips onto the edge of the telescoping bar for stability. We coat this attachment with sandpaper on all possible contact areas for enhanced grip.

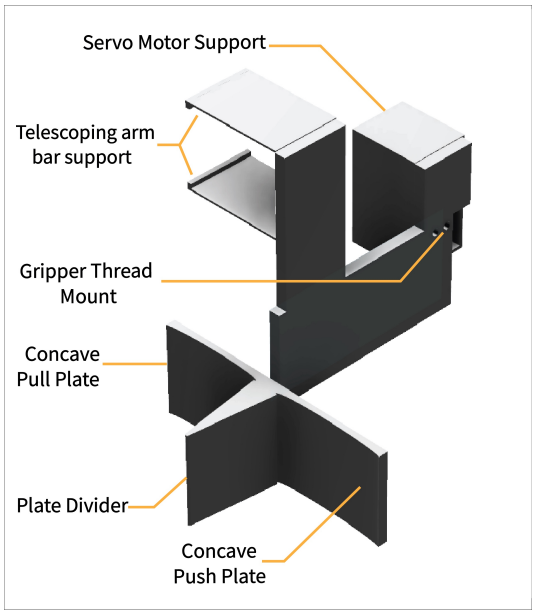


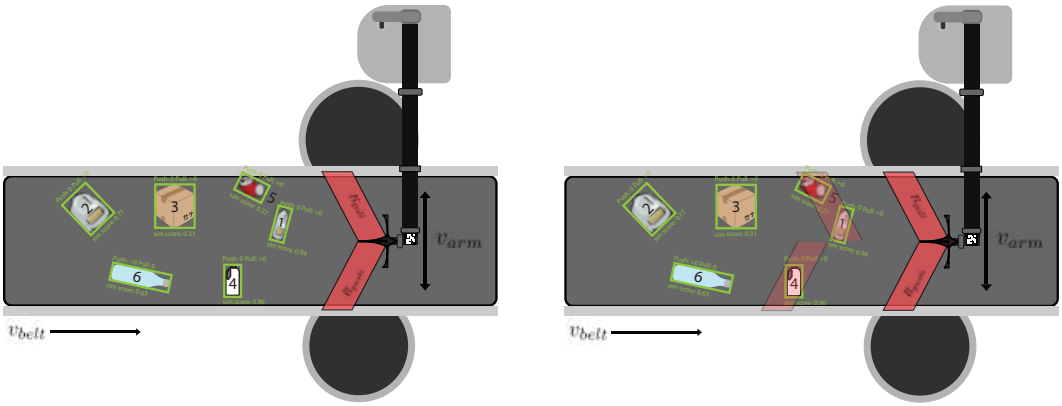
Figure 3: Design of our custom attachment for pushing and pulling objects off the conveyor belt

Empirically, we found this attachment to be highly effective at removing most kinds of objects. As shown in the supplementary video, the attachment would gently guide the object to the nearest edge if the target objects were primarily on either half of the belt. However, for highly deformed objects (like some heavily crushed cans or bottles) or objects with a lot of protrusions (like trays), the objects could sometimes get stuck on the attachment's divider. Occasionally, this caused the robot to fail in removing an object even when it attempted to remove the desired object.

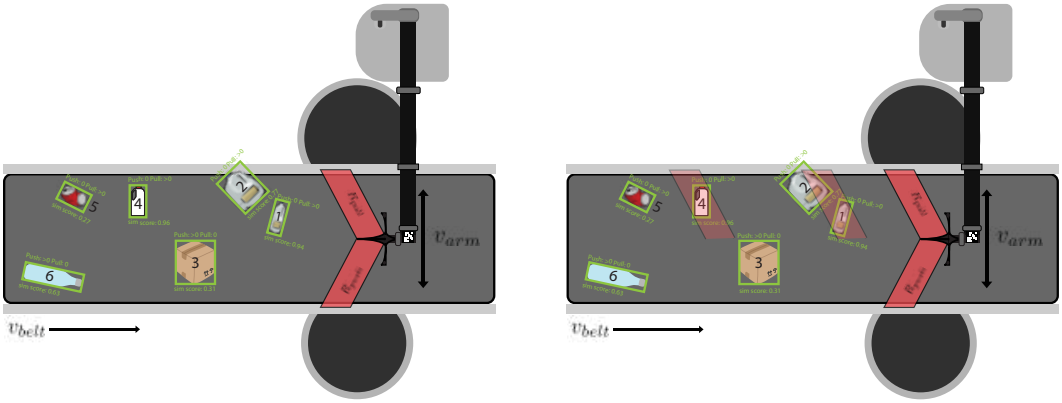
1.2 Detection of Picked Objects

This module determines which objects are picked by humans in real-time. This module has the following two parts:

1. **Pick Event Detection:** The system uses the MediaPipe real-time hand tracking API [2] to detect 21 2D landmarks on the user's palm. These landmarks were superimposed on the images. These images were then used to train a ResNet-18 [3] image classifier offline. The



(a) Scenario 1



(b) Scenario 2

Figure 4: **Visualization of Different Scenarios for the Hit Planner:** a) In Scenario 1, the removal of object 1 would inadvertently remove object 5, which has a relatively low similarity score. The hit planning algorithm would negate the score contribution of object 5 to object 1's hit score as explained in 2 and determine that object 4 by itself has the highest hit score, thereby initiating a push action sequence. b) In Scenario 2, the removal of object 1 would likely remove object 2, which has a relatively high similarity score and would only increase object 1's hit score as explained in 1. In this instance, the robot would target object 1 with a pull action sequence and target object 4 with a pull action sequence once again after resetting the telescopic arm to a central position.

images are classified as either a picking event, not a picking event, or none if there is no hand present in the image. To train this model, we collected videos of 13 people picking items off the moving conveyor belt to capture the different techniques people use to pick up different objects. We labeled each frame manually as either a pick event, not a pick event, or no hand present in the frame. The trained model predicts a pick-event onset with 96% accuracy on the leave-one-out cross-validation set.

2. **Identification of Picked Objects:** To detect which object was picked by the person, we first ran inference on the belt images using the trained instance segmentation model (YOLACT [4]) to identify each object's position. We then used the predictions of the trained pick onset detector described above to identify the precise time when the person picks up an object. We use the 21 landmarks predicted on the person's palm to draw a tightly fitted rotated bounding box on their palm. Finally, we find an overlap between the rotated bounding box over the person's palm and each of the predicted object bounding boxes in a few frames preceding the onset of *Pick* event to obtain an un-occluded image of the object that was picked.

1.3 Hit Planner

We used our method to obtain similarity scores for all objects on the conveyor belt within the camera's frame situated between the robot and the human (Camera 2 in Figure 2). These similarity scores were then passed to the object removal algorithm within the hit planner module to assign each object an associated "hit score" using a multi-variable function. This score was maximized across all objects in the frame, S, and one item was selected for removal by the robot.

For each object, our object removal action policy first determines the action sequence that would be most successful at removal given the object's positioning relative to the plate mounted to the end of the robot's telescopic arm, as seen in Figure 3. The action space of the robot is comprised of either a pushing, pulling or avoiding action sequence.

Given our removal actuation constraints, removing a target object may remove other unintended objects, referred to as "casualties." To account for the possibility of casualties, for every object, we approximated a region of collision using the speed of the belt, v_{belt}, and the robot's arm extension/retraction speed, v_{arm}. The regions of collision for every corresponding action sequence were two dimensional parallelograms extending from the push plate with inclination $tan^{-1}\left(\frac{v_{arm}}{v_{belt}}\right)$. All objects whose bounding boxes overlap with this region have a removal probability proportional to their percent overlap with the collision region.

Given a frame containing a set of objects S, we perform optimizations that yield the object with the highest push score, i_{push}, and similarly the object with the highest pull score, i_{pull}. If there are multiple objects in S it is possible for i_{push} and i_{pull} to be distinct. Therefore

$$i_r = \max\{i_{push}, i_{pull}\}$$

returns the single object i_r to be removed by the robot as well as the corresponding action sequence request that should be sent to the robot's onboard computer. For the push action sequence, we optimized for

$$i_{push} = \max_{i \in S} \mathbb{1}_{push}(i) * \left[\alpha_1 sim_i + \sum_{j \in S \ i \neq j} C(j) * \alpha_1 sim_j * \alpha_2 \left(\frac{|R_{push} \cap R_j|}{|R_j|} \right) \right] \tag{1}$$

Similarly for the pull action sequence, we optimized for

$$i_{pull} = \max_{i \in S} \mathbb{1}_{pull}(i) * \left[\alpha_1 sim_i + \sum_{j \in S \ i \neq j} C(j) * \alpha_1 sim_j * \alpha_2 \left(\frac{|R_{pull} \cap R_j|}{|R_j|} \right) \right] \tag{2}$$

where α_1 and α_2 are tuning constants, sim_i represents the similarity score for object i, and $\frac{|R_{pull} \cap R_i|}{|R_i|}$ represents the percent overlap of object's bounding box, R_i, with the collision region R_{pull}. $\mathbb{1}_{push}$ and $\mathbb{1}_{pull}$ are indicator functions for assessing whether a pull or push is necessary for given object i. If the position of an object i is above the push plate of the robot relative to the belt, $\mathbb{1}_{push} = 1$ and $\mathbb{1}_{pull} = 0$ and vice versa for when i is situated below the push plate. We include a mapping function $C(i)$ that either subtracts or adds the score contribution of a casualty i depending on whether or not the similarity score is above some predefined threshold T_{sim}.

$$C(i) \coloneqq \begin{cases} 1 & \text{if } sim_i \geq T_{sim} \\ -1 & \text{if } sim_i < T_{sim} \end{cases} \tag{3}$$

This is to ensure that we prioritize removal of objects that have a high probability of removing objects that are similar to the human-selected objects.

2 Properties of the Dataset

It is challenging to develop a dataset that includes the enormous variety of recyclables that a real Materials Recovery Facility (MRF) processes. For our research, we toured such facilities and watched

numerous videos documenting the composition of streams at MRFs. Then, we worked to create a collection of objects representing a characteristic sampling of recyclables typically seen in these facilities. More specifically, we selected our set of over 500 unique objects that were diverse in category, material, size, color, shape, deformability, reflectiveness, dirtiness, opacity, and density. We deliberately excluded paper-based and glass-based recyclables because air pumps and optical sorters typically remove these items reasonably well in a MRF. For our experiments, we captured images of the selected objects under different lighting conditions, orientations, and levels of motion blur on a moving conveyor belt, adding to the diversity of our dataset.

The dataset was collected using three Intel RealSense D435i RGB-D cameras mounted over the conveyor belt. We used our trained instance segmentation model to run inference on the overhead camera stream at 1fps. The predicted instance segmentation masks were converted to rotated bounding boxes, then cropped and saved offline. The dataset was manually curated to obtain 1502 images of objects across the ten categories whose distribution is shown in Figure 6. Figure 5 shows some sample images from each of our categories in the dataset.

Figure 5: **Sample Images from our Offline Dataset:** Ten sample images from each of our ten object categories.

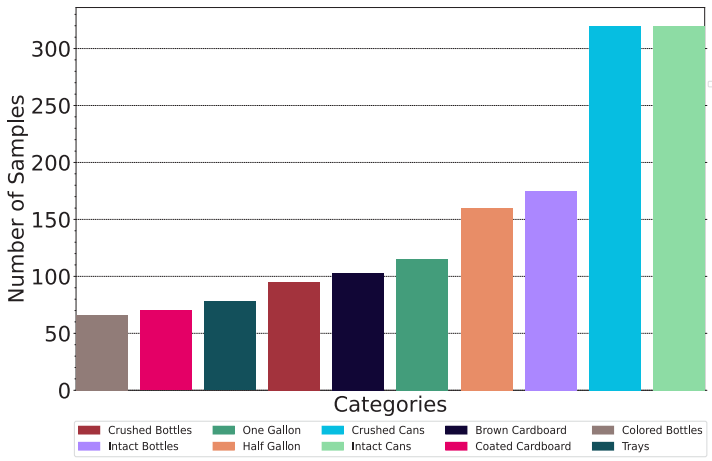


Figure 6: Data distribution in our offline dataset

3 Implementation Details for Our Method

To deal with memory and computational constraints, we resize each image to 128 x 128. We then stochastically apply three augmentations in the following sequence: *random cropping* followed by resizing to original size, *random color distortions*, and *random gaussian blur*. Finally, we train and evaluate our approach end-to-end on a single NVIDIA GTX-3080ti GPU with the hyperparameters shown in Table 1. We use the same set of hyperparameters for the ablation studies as we did not notice any significant variation in the results with changes in the learning rate, projection dimension, instance temperature, or cluster temperature. We also studied the importance of loss weighting of the instance and the cluster losses (λ_{ins} and λ_{clu}). Similar to Li et.al.[5], we obtained optimal performance when the losses were weighted equally. We ran a hyperparameter search between 0.1 and 1 to determine the optimal weighting for the human-supervised loss (λ_{human}). We obtained the best performance across all categories when λ_{human} was 1.

Parameters	**Values**
Learning Rate	0.01
Batch Size	64
Feature Extractor	ResNet-18
Number of Epochs	100
Projection Dimension	128
Number of Clusters	10
Instance Temperature (τ_{inst})	0.5
Cluster Temperature (τ_{clu})	1.0
Instance Loss Weight (λ_{ins})	1.0
Cluster Loss Weight (λ_{clu})	1.0
Human Supervised Loss Weight (λ_{human})	1.0

Table 1: Hyperparameters for Training Our Method

4 Design Considerations of Our Method

We wanted to design a system that enables a human to teach a robot partner the kinds of objects that they are sorting with a few examples and that could be demonstrated in the context of recycling in Materials Recovery Facilities (MRFs). In such facilities, the human workers typically sort out one type of recyclables while adapting to variations in task specifications and seasonal changes in the stream composition. Because of how objects are handled in a MRF and the economic reality of the recycling industry, one of our primary problem constraints was that the robot could quickly learn

K	Average F1 scores across all categories
2	0.820 ± 0.016
8	0.876 ± 0.02
9	0.882 ± 0.02
10	$\mathbf{0.886 \pm 0.017}$
11	0.883 ± 0.019
12	0.877 ± 0.02
15	0.867 ± 0.021
20	0.847 ± 0.018
50	0.828 ± 0.022

Table 2: Impact of the number of clusters (K) on overall model performance

from a limited number of examples of an object category provided by a human. The next paragraphs explain each of our design considerations in detail:

1. **Large object variety:** In the recycling setting, the objects are not guaranteed to look constant through time because of the way they are handled, their deformable properties, and the frequent introduction of new objects. For a standard supervised learning method to perform well at classifying these objects, it would require training models with large datasets to keep up with the performance requirements. Therefore, we chose to approach this problem from a self-supervised contrastive learning perspective, which could adapt to the changing stream composition by constantly re-training in a more active learning fashion with just a few new human labels.

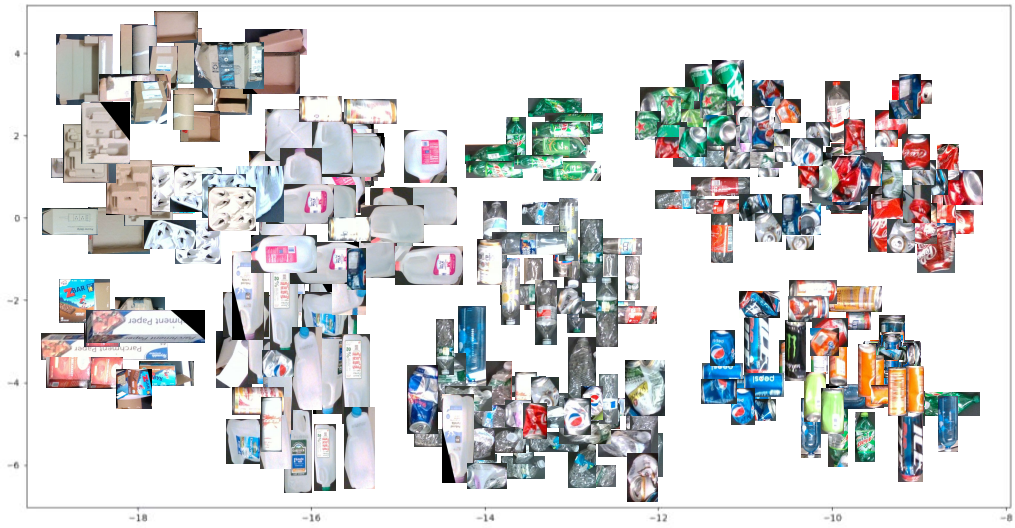
   We explored various state-of-the-art contrastive learning techniques for our work and found that these techniques typically pass two stochastically augmented views of the same image through an image classification network and a multi-layer perceptron (MLP) to bring them close to each other in the embedding space. These methods have been shown to learn very strong representations for initializing large neural networks that perform a wide variety of downstream supervised learning tasks like image classification, semantic segmentation, and object detection [6]. However, since we wanted to learn a representation of objects such that similar objects were grouped in the embedding space, we needed to take into account inter-instance similarity. We found contrastive clustering [5] to be the best self-supervised method for learning object representations.
2. **Humans are unable to select all examples of a given category:** In a real-world recycling setting, where the conveyor belts are typically very crowded, the human sorters cannot pick out all the objects that belong to a given category. So, there could be objects left on the conveyor belt that belong to the category of interest. This means that the robot would have to gain an understanding of the properties of the objects of interest to their human partner with a limited number of positive examples but no negative examples. Therefore, we designed the proposed human-supervised loss to force the features of the human-selected examples to be close to each other in the embedding space. When the human-supervised loss is jointly trained with the contrastive clustering objective function, the human-supervised loss guides the formation of clusters in the embedding space to be better aligned with human choices.
3. **Real time training considerations:** Theoretically, any image classification network could have been used as a backbone for the contrastive learner. However, since we intended to use this model in real-time, we opted for ResNet-18 [3], which is lightweight and has been shown in the literature to be powerful enough to learn good visual representations [7, 8].

5 Additional Results

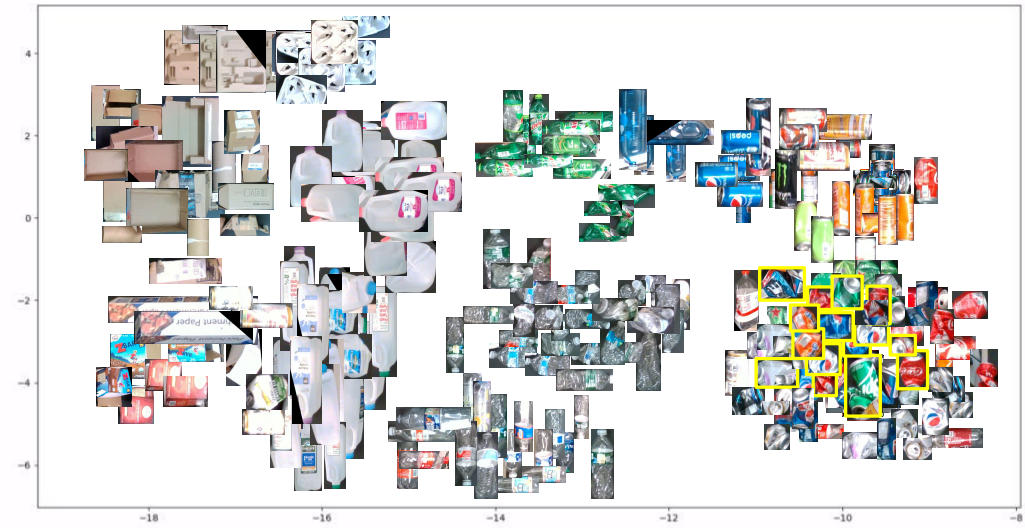
5.1 Offline Evaluation Results

5.1.1 Impact of the Number of Clusters (K) in the Cluster Projection Head

We base our work on a self-supervised clustering algorithm, Contrastive Clustering [5]. Thus, we are required to provide the number of clusters K as a hyperparameter of the model. In the results



(a) Contrastive Clustering



(b) Our Approach

Figure 7: **Comparison of t-SNE Visualization of the learned representations on the offline dataset:** a) t-SNE plot for the self-supervised contrastive clustering [5], b) t-SNE plot for our approach. Images with yellow border denote the objects selected by the human. In our approach, clusters for each category show better separation than the contrastive clustering approach, and clusters are overlapping when an increasing number of features are common.

presented in the paper, we defined the number of clusters K to be the number of categories C to keep our evaluation consistent with the original Contrastive Clustering paper [5]. However, our method does not need to know precisely how many pre-defined object categories are being observed; instead, a ballpark amount suffices. To demonstrate this idea, we conducted an experiment showing that the number of clusters (K) need not be precisely equal to the number of categories (C). As indicated in the table 2, if the number of clusters is significantly lower or higher than the number of categories in the downstream task, there is a significant decrease in performance. Otherwise, the performance of our approach is robust to the value of K.

Experiment	**1**	**2**	**3**	**4**	**5**	**6**	**7**
Crushed Bottles	0.12 ±0.00	0.65 ±0.02	0.50 ±0.06	0.56 ±0.01	0.59 ±0.03	0.70 ±0.03	0.82 ±0.02
Intact Bottles	0.21 ±0.00	0.58 ±0.03	0.51 ±0.02	0.64 ±0.01	0.67 ±0.01	0.68 ±0.02	0.80 ±0.03
One Gallon	0.14 ±0.00	0.76 ±0.01	0.64 ±0.03	0.73 ±0.01	0.68 ±0.01	0.81 ±0.01	0.84 ±0.02
Half Gallon	0.19 ±0.00	0.72 ±0.02	0.69 ±0.01	0.68 ±0.01	0.69 ±0.01	0.79 ±0.02	0.86 ±0.02
Crushed Cans	0.35 ±0.00	0.73 ±0.04	0.65 ±0.01	0.78 ±0.03	0.73 ±0.03	0.82 ±0.01	0.84 ±0.02
Intact Cans	0.40 ±0.00	0.81 ±0.00	0.73 ±0.02	0.82 ±0.01	0.81 ±0.01	0.85 ±0.02	0.86 ±0.02
Brown Cardboard	0.13 ±0.00	0.74 ±0.02	0.71 ±0.01	0.75 ±0.02	0.84 ±0.02	0.89 ±0.02	0.94 ±0.01
Coated Cardboard	0.05 ±0.00	0.87 ±0.05	0.37 ±0.03	0.52 ±0.02	0.80 ±0.01	0.91 ±0.04	0.98 ±0.02
Colored Bottles	0.08 ±0.00	0.91 ±0.01	0.71 ±0.01	0.77 ±0.00	0.64 ±0.00	0.93 ±0.02	0.93 ±0.00
Trays	0.06 ±0.00	0.68 ±0.01	0.56 ±0.01	0.55 ±0.01	0.66 ±0.01	0.98 ±0.01	1.00 ±0.00

[1] Human Supervision Only
[2] Instance Loss Only (SimCLR)
[3] Cluster Loss Only
[4] Instance Loss + Human Supervision
[5] Cluster Loss + Human Supervision
[6] Contrastive Clustering (Instance Loss + Cluster Loss)
[7] Human Supervised Contrastive Clustering (Our Method)

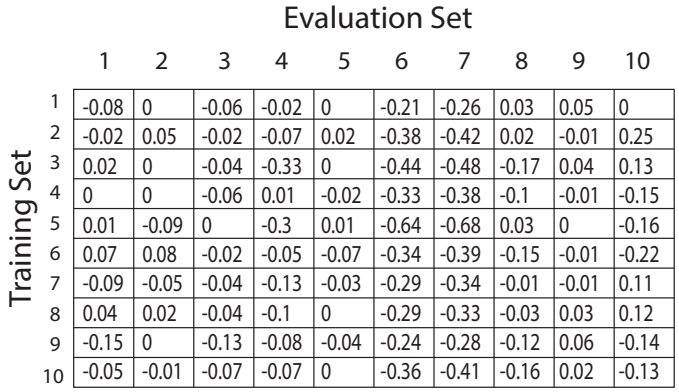
Table 3: **Results of our Offline Evaluation (Accessible Version):** Comparison of F1 Scores with Baselines and Ablation Study aggregated over 3 human-selected pools per category.

5.1.2 Qualitative Results

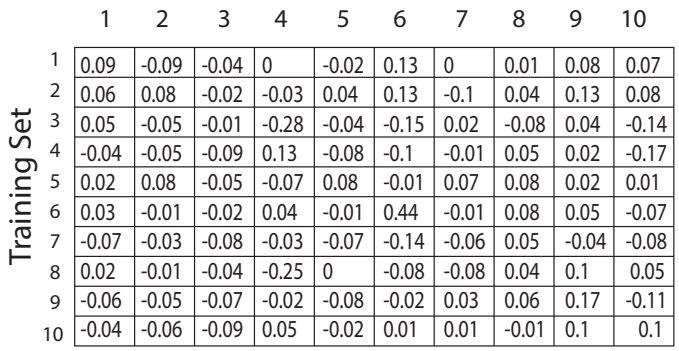
Figure 7 shows plots of 10 clusters obtained by projecting the embedding vectors for the points (128-dimensions) onto 2-dimensions using t-SNE for contrastive clustering (Figure 7a) and for our approach (Figure 7b). In Figure 7b, the images with a yellow border show the human-selected objects. This shows that the clusters formed as a result of our approach are more homogeneous and show good separation across classes, overlapping only when an increasing number of features are common. For example, in contrastive clustering (Figure 7a), many crushed cans lie in the same cluster as un-crushed cans if they are of the same color or with crushed bottles. However, in our approach, most crushed cans get pulled into the same cluster when the human provides examples of crushed cans.

5.2 Real Robot Evaluation

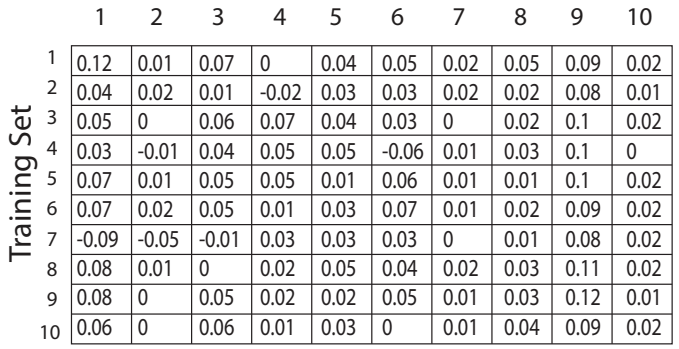
Figure 9 shows some qualitative examples of similarity scores for every object on the conveyor belt when the human trained our approach in real-time with 30 examples of different types of objects.



a) Impact of Human Supervision on Instance Loss



b) Impact of Human Supervision on Cluster Loss



c) Impact of Human Supervision on Constrastive Clustering

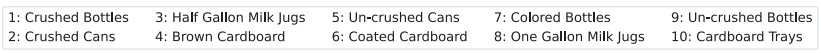
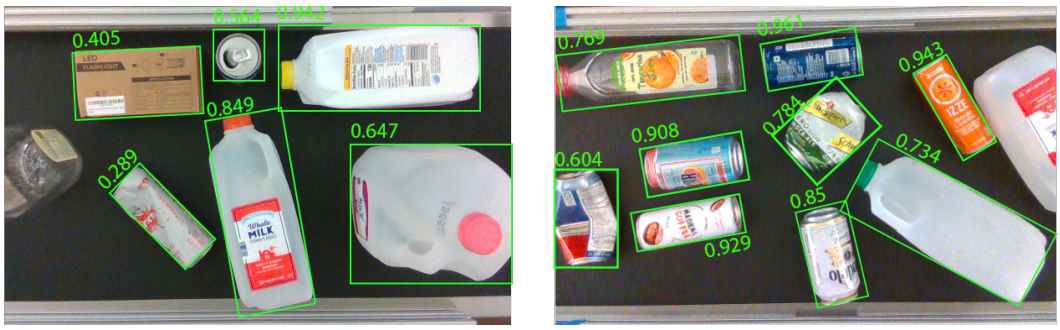


Figure 8: **Impact of augmenting human supervision to contrastive learning (Accessible Version)**: Difference in performance between self-supervised learning augmented with human supervision and the self-supervised learning model trained independently for instance loss, cluster loss, and contrastive clustering, respectively across categories. Specifically, w.r.t. the experiment numbers in Figure 3 in the main paper, a) represents Experiment 4 - Experiment 2, b) represents Experiment 5 - Experiment 3, c) represents Experiment 7 - Experiment 6. (Best viewed on a PDF processor with zoom)



a) b)

Figure 9: **Sample Qualitative Results from our Robot System**: Similarity scores of objects on the conveyor belt obtained from our method, when the human trained the system with 30 objects of type a) half-gallon milk jugs b) un-crushed cans

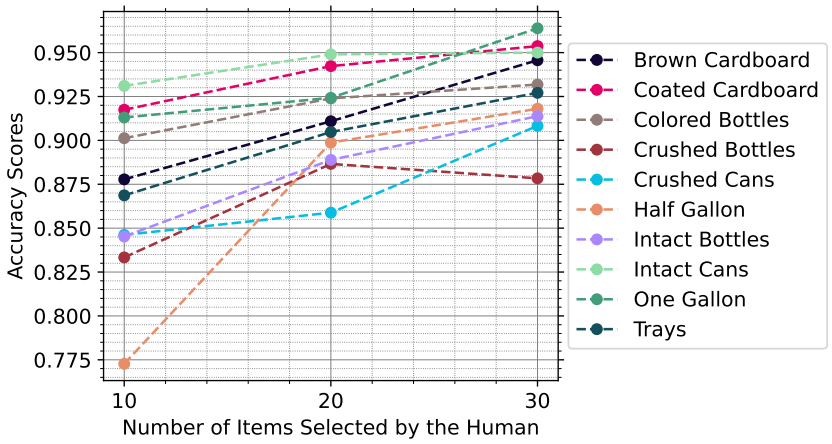


Figure 10: **Accuracy of Our Method in Real Time**

These results show that the similarity scores are closer to 1 for the objects belonging to the category of interest, and the similarity scores are lower for those not belonging to the category of interest.

Table 4 shows the average number of True Positives, False Positives, False Negatives, and True Negatives across all categories as our approach was trained with 10, 20, and 30 human-selected examples. These results show that the average number of True Positives and True Negatives increases with more human-provided examples. In contrast, the average number of False Positives and False Negatives decreases as the number of human-provided examples increases.

The main paper demonstrates the performance of the real robot by showing the F1-scores of the robot system as the researcher picks 30 items in increment of 10. Figure 10 shows the accuracy of the robot system when evaluated in a similar fashion. The robot's accuracy improved by 3 points on average across categories for every ten human-provided examples.

6 Discussion

6.1 Additional Intuitive Explanation of Our Method

It is important to note that inductive bias is key for a machine learning model to learn from just a few samples. In our paper, this bias is provided through the cluster loss, which partitions the embedding space into a pre-specified number of clusters that we hope represent object categories – though our approach does not enforce this. Adding the human-supervised loss to the representation learning process helps guide the formation of the clusters according to the human's requirements. Worth

Number of Training Items	10	20	30
True Positives	13.4 ± 4.94	15.2 ± 4.31	16.8 ± 4.73
False Positives	4.7 ± 1.88	3.5 ± 1.71	2.3 ± 1.33
False Negatives	5.8 ± 2.93	4.9 ± 2.23	4.2 ± 2.20
True Negatives	60.6 ± 16.61	69 ± 13.07	69.4 ± 14.30
Precision	0.739 ± 0.07	0.814 ± 0.074	0.885 ± 0.058
Recall	0.706 ± 0.10	0.761 ± 0.074	0.806 ± 0.08
F1-score	0.715 ± 0.05	0.782 ± 0.04	0.838 ± 0.03
Accuracy	0.87 ± 0.04	0.908 ± 0.02	0.929 ± 0.02

Table 4: Performance Statistics of our Method on the Real Robot System

noting, that knowing the exact number of object categories that the robot needs to handle in practice is not a requirement of our approach.

6.2 Assumption about the Human Selecting One Category of Objects

The assumption that a human would manipulate objects from a single object category holds across many real-world Materials Recovery Facilities (MRF), which our setup models. When we started this project, we visited a recycling facility to understand how each category of recyclables is sorted sequentially using a combination of automated sorters and humans. At every stage in the sorting process, a particular category of recyclable was extracted from the stream using an automatic sorter. The items of a given category missed by the automatic sorter were sorted by human sorters who tried manually extracting the remaining objects on a fast-moving and crowded conveyor belt. The human sorters are trained to extract only one category of interest at a given time because some objects in the recycling stream are more valuable than others. For example, after an automatic sorter has extracted most of the paper from the stream, the human sorters would be tasked with extracting all the remaining paper-based products. This would make the stream more uniform for subsequent automatic sorters to remove a certain kind of plastic. In practice, though, the human sorters may not be able to extract all the objects of the desired category, which could lead to quality issues in the target end-product. Therefore, if a robot were to assist a human sorter in sorting through a particular kind of object, the robot would have to learn the properties of the objects that its human collaborator is interested in. Our work proposes a solution to this problem whereby the robot observes the human picking up a limited number of objects from a single category and quickly learns to distinguish these types of objects.

References

[1] C. C. Kemp, A. Edsinger, H. M. Clever, and B. Matulevich. The design of stretch: A compact, lightweight mobile manipulator for indoor human environments. *arXiv preprint arXiv:2109.10892*, 2021.

[2] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann. Mediapipe hands: On-device real-time hand tracking. *arXiv preprint arXiv:2006.10214*, 2020.

[3] 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*, pages 770–778, 2016.

[4] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee. Yolact: Real-time instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9157–9166, 2019.

[5] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng. Contrastive clustering. In *2021 AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[6] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

[7] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[8] X. Chen and K. He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.