Exploring the Effects of Contrastive Pretraining and Test-Time Post-training on Semantic Segmentation of Waste

James Cheng
Stanford University
jzcheng@stanford.edu

Andy Ouyang
Stanford University
andyou@stanford.edu

Nishikar Paruchuri
Stanford University
nishikar@stanford.edu

Abstract

We investigate the semantic segmentation of recyclable waste using both supervised CNNs and self-supervised contrastive pretraining, as well as various post-training techniques. We train and evaluate SOTA U-Net and DeepLabV3 models on the ZeroWaste dataset as starting baselines. We then explore the effects of SimCLR-style pretraining and test-time refinement via Conditional Random Field Post-Processing and Uncertainty-Aware Refinement Networks on our baseline models. Contrastive pretraining is shown to underperform without a sufficient batch size and displays an inability to extract finer-grained semantic details required for semantic segmentation, while alternatively improving high-level feature extractions. Our results show that Conditional Random Fields can improve precision, but at a significant cost to performance due to oversmoothing object boundaries. Uncertainty Aware Refinement networks are shown not to offer strict overall performance improvements, but do offer an equally performing variant with tradeoffs between higher accuracies for already high-performing materials classes and lower accuracies for underperforming classes. DeepLabV3 remains the strongest baseline, with UARNet offering modest improvements and tradeoffs.

1. Introduction

We are investigating the problem of using image recognition to identify and sort recyclable materials. Waste is often not sorted correctly in recycling facilities, which leads to both false negatives (recyclable materials thrown into landfills) and false positives (landfill waste mixed with recyclables), making efficient and automated sorting pipelines integral to proper waste disposal and environmental safety. Our project aims to prevent these errors from happening using computer vision on complex and cluttered data to ensure accurate, robust, and efficient recycling.

Our problem takes as input RGB images containing multiple types of recyclable waste, such as plastics, cardboard, metal, and soft plastics, arranged in cluttered, real-world scenes. We then use a segmentation model (i.e., DeepLab, U-Net) to output a dense per-pixel labeling (semantic segmentation) where each pixel is assigned to one of the target waste classes or background. This requires both global context understanding (to recognize material textures and shapes) and precise local boundary delineation (to separate adjacent or overlapping objects). Accurate segmentation enables downstream sorting mechanisms to identify and segregate different waste materials for recycling.

In addition to training various baseline CNN models on our dataset, we test the effects of contrastive pretraining and post-training augmentations on the waste segmentation problem to probe which techniques are best for each use case. We evaluate performance using standard segmentation metrics such as Intersection over Union (IoU) and pixel accuracy. Our experiments are conducted on the ZeroWaste dataset, a challenging benchmark featuring real-world, cluttered scenes with diverse recyclable materials.

2. Related Work

Semantic segmentation of waste is an emerging area with several recent datasets and approaches, with a variety of challenges that come along with it.

2.1. Datasets

For example, the TACO dataset [5] provides waste in the wild images with diverse environments like beaches and roads, each annotated with pixel-level classes (plastic, paper, etc.). Likewise, the ZeroWaste dataset [2] provides industrial conveyor-belt scenes with highly cluttered waste objects such as soft/rigid plastic, cardboard, and metal labeled at the pixel level. Other novel datasets include the MJU-Waste dataset [8], which adds depth data to RGB images, helping to distinguish between overlapping or occluded items, and the WasteMS dataset [12], which is the first multispectral dataset established for the semantic segmentation of lakeside waste. These datasets all highlight common challenges within the waste segmentation field,

which are deformable objects, severe occlusion, class imbalance, and background clutter. We tended towards the ZeroWaste dataset due to its inclusion of both labelled segmentation masks and unsupervised images.

2.2. CNN-based Segmentation Methods

Traditional CNNs and encoder-decoder models remain the state-of-the-art (SOTA) for waste segmentation. Common architectures such as FCN, U-Net, and DeepLabV3 have been applied as baselines. For instance, DeepLabV3 trained on ZeroWaste has only achieved $\sim$52% mean IoU on the test set, demonstrating the domain's difficulty. These architectures have been extended in various ways in hopes of improving their performance:

- **DeepLab-based baselines**: ZeroWaste's authors reported that DeepLabV3 with a ResNet101 backbone achieved $\sim$52% mean IoU [2]. While effective in a multi-scale context, the model struggled with rare classes such as rigid plastic and metal, and data augmentation only slightly improved the results.
- **U-Net variants**: U-Net architectures are popular for their simplicity on small datasets. For example, Wei et al. [10] developed an improved U-Net for underwater garbage, achieving $> 85\%$ IoU on each class by adding focal loss and deeper encoder backbones. Similarly, Qi et al.'s NUNI-Waste method [6] uses a U-Net with adaptive loss and novel data augmentation on ZeroWaste to raise mean IoU from $\sim$51% to 55.4% over the baseline by weighting classes and consistency regularization.
- **Boundary-aware CNNs**: New architectures modify CNNs specifically for waste. For example, COSNet introduces novel components such as feature-sharpening blocks and boundary enhancement modules to better capture irregular waste object edges [1]. COSNet was shown to yield a $\sim$1.8% mean IoU gain on ZeroWaste-f, showcasing how enhancing boundary features improves CNN segmentation in cluttered scenes.

Overall, fully-supervised CNNs achieve the highest accuracy given labels, with SOTA results in the mid-50% IoU range. The strengths of CNNs are that they are well-understood, fast at inference, and excel with sufficient annotations. Weaknesses include expensive pixel-level labels, overfitting to specific backgrounds, and struggling with rare classes or occlusion.

2.3. Transformer-based Methods

Vision Transformer (ViT) and hybrid models have been able to achieve general SOTA in segmentation; however, their application to waste images has not been fully explored. Models such as SegFormer, which uses transformer encoders with MLP decoders, reached higher mIoU than previous convnets on Cityscapes, demonstrating the power of global self-attention [11]. In principle, these architectures could benefit waste segmentation through reasoning across clutter, but would require large training data.

In practice, so far, there are a few studies that hint at the potential of transformers. NUNI-Waste briefly tests transformer backbones, with them reporting similar results to CNN baselines [6]. Another paper found Swin-Transformer to be effective on building material segmentation for construction, suggesting its potential for waste segmentation [9]. However, no fully dedicated waste segmentation paper fully exploits transformers.

Pros of transformers include their ability to model global context and flexible attention, which could capture variations in waste objects. The cons of transformers are their need for an abundance of data and compute, of which there are few annotated waste images. Although CNNs still dominate the field, future work will likely integrate transformers to push the frontier as more annotated data becomes available.

2.4. Weakly-Supervised Methods

To address expensive annotation, weak supervision (training with image-level or coarse labels) has been an emerging field. In waste segmentation, ZeroWaste authors collected ZeroWaste-w, where each frame is labeled "before" or "after" manual removal of target material. They then used Class Activation Mapping (CAM) to get pseudo-masks. However, CAM and other improved/simpler CAM variants failed to get above 34.6% IoU, which is far below the 52% IoU of the fully-supervised baseline [2].

Video is also utilized as another weak cue. Marelli et al. build saliency maps that exploit the temporal coherence between consecutive frames in a video, promoting consistency when objects appear in different frames [4]. On a small lab dataset, they were able to outperform a CAM method, highlighting how weak supervision can help.

Weakly-supervised segmentation in waste is an active but very early field with strengths being cheap annotation and the weakness of noisy and incomplete masks with IoU scores that fall significantly below fully-supervised baselines. We incorporate an unsupervised training stage into our training pipeline to explore how semi-supervised processes affect overall performance and class predictions.

3. Methods

3.1. Baseline: Predict-All-Background

Our simplest baseline method is to predict every pixel as a background class. This is a valid baseline because the majority of the pixels in our images are background, so this method establishes a lower bound for our more advanced

methods.

3.2. CNN Architectures

We trained two pre-initialized, SOTA CNN architectures (U-Net and DeepLabV3) on the ZeroWaste-f train dataset as standard baselines for semantic segmentation. We also apply contrastive pretraining and various post-training methods to the baselines in hopes of improving performance at test time.

3.2.1 U-Net + MobileNetV2

U-Net is a convolutional neural network architecture that has become a standard model for general semantic segmentation tasks [7]. The architecture is characterized by a symmetric "U" shape consisting of a contracting encoder path and an expanding decoder path. The encoder is a typical convolutional neural network that applies repeated convolution and max-pooling operations to capture high-level semantic features while reducing the spatial resolution of the image. The decoder performs upsampling, which is the process of increasing the spatial resolution of feature maps in order to recover the original input size for the output predictions. Some common techniques for upsampling include transposed convolutions, where sets of weights are learned to operate, bilinear or nearest neighbor interpolation, which resizes feature maps based on simple mathematical rules, and unpooling, which reverses the max-pooling operation by placing values back into their original location. A key feature of U-Net is the use of skip connections, in which at each level feature maps from the encoder are concatenated with corresponding decoder features. We utilize a pretrained MobileNetV2 as our encoded backbone. This configuration offers a strong balance between efficiency and accuracy: MobileNetV2's depthwise separable convolutions reduce parameter count and FLOPs, while U-Net's skip connections recover spatial detail and localization, making it well-suited for segmenting small, irregularly shaped, and occluded waste objects under limited compute budgets.

3.2.2 DeepLabV3 + ResNet50

The other model we used is PyTorch's DeepLabV3 segmentation model with a ResNet-50 backbone. DeepLabV3 is a state-of-the-art semantic segmentation model that combines powerful feature extraction with refined boundary localization, designed for solid performance across objects of varying sizes. This is useful because in the ZeroWaste dataset, there is much more clutter and variance in object size. The model uses a ResNet-50 encoder, which is pretrained on ImageNet; this helps extract strong image features without needing to train from scratch.

DeepLabV3 introduces convolution with upsampled filters or "atrous convolution," which allows explicit control of the resolution at which feature responses are computed [3]. The central innovation is the Atrous Spatial Pyramid Pooling (ASSP) module, which applies multiple parallel atrous convolutions with different sampling rates. This allows the model to capture contextual information at multiple scales without reducing the spatial resolution. ASSP enables the network to have a large receptive field and handle objects of various sizes more effectively than standard CNNs. Lastly, to address the challenge of segment boundary refinement, DeepLabV3 introduces a decoder module that upsamples the ASSP output and combines it with low-level features from earlier, helping the model to produce sharper object boundaries and more accurate segmentation.

We initialized the segmentation head randomly (without pretraining) while using a ResNet-50 backbone pretrained on ImageNet, so the model learns pixel-wise class predictions specific to the ZeroWaste dataset. Overall, we chose this model because of ASPP's ability to capture objects at multiple scales, which we believed would be useful in ZeroWaste's more complicated and cluttered images.

3.3. Self-Supervised Contrastive Pretraining

We used a SimCLR approach for contrastive pretraining to help our model's encoder extract general, transferable visual features from unlabeled data. These features are later fine-tuned for downstream tasks, specifically semantic segmentation. The aim of this self-supervised contrastive pretraining is to push representations of similar materials closer together and push further classes apart. This contrastive pretraining stage is only applied to our DeepLabV3 encoder due to computational limitations.

Our DeepLabV3 encoder is a ResNet-50 network backbone (excluding the final fully connected layer), which extracts high-level feature representations from input images. These were then passed through a lightweight projection head consisting of a Linear $\rightarrow$ ReLU $\rightarrow$ Linear architecture, mapping them to a lower-dimensional space where contrastive learning is performed to optimize feature extraction.

The contrastive loss used is the NT-Xent (Normalized Temperature-scaled Cross Entropy) loss, defined as:

$$l\ (i,j) = -\log \frac{\exp(\ \mathrm{sim}(z_i, z_j)\ /\ \tau)}{\sum_{k=1}^{2N} \mathbf{1}_{k \neq i} \exp(\ \mathrm{sim}(z_i, z_k)\ /\ \tau)}$$

where $\mathrm{sim}(z_i, z_j)$ denotes cosine similarity between embeddings and τ is a temperature parameter.

We initialize Resnet50 with ImageNet weights and train the encoder + projection head on the unsupervised ZeroWaste-S dataset. We then discard the projection head and reintegrate the contrastively pre-trained encoder as the backbone of a DeepLabV3 segmentation model. After the pretraining stage, we then fine-tune this new model on the

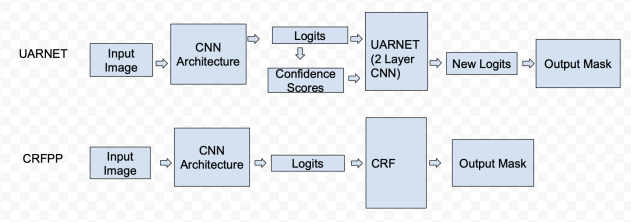

Figure 1. Post-training Architectures

labeled ZeroWaste-f segmentation dataset, as we did with the previously initialized U-Net and DeepLabV3 models.

3.4. Evaluation-time Post-training Methods

To improve the output quality of our segmentation models without retraining the backbone, we experimented with two post-training refinement techniques. These methods operate on the output logits of trained models (DeepLabV3, U-Net, and contrastively pre-trained DeepLabV3) and aim to enhance segmentation predictions without modifying the original model weights.

3.4.1 CRFPP (Conditional Random Field Post-Processing)

We apply a dense Conditional Random Field (CRF) as a post-processing step to refine raw segmentation outputs. CRFs model contextual dependencies by encouraging label consistency between nearby pixels with similar appearance. Specifically, we use a fully-connected CRF with Gaussian edge potentials and apply mean-field approximation for MAP inference. This method aims to sharpen object boundaries and reduce spurious noise by applying smoothing. A pixel surrounded by conflicting pixels should be relabeled, while areas of stark differences should be reinforced as boundaries.

3.4.2 UARNet (Uncertainty-Aware Refinement Network)

UARNet is a lightweight convolutional refinement network trained to adjust model predictions using uncertainty cues in model output. For each input pixel, the UARNet takes the per-class logits of that pixel from the already trained CNN model and appends a confidence heuristic, defined below.

$$\text{conf}_p = \max_c \text{softmax}(y_p)_c$$

The softmax returns a probability distribution over classes for each pixel, and taking the maximum gives us a heuristic on how confident the model is in predicting that pixel correctly. This per-pixel confidence value is concatenated to the logits output from the trained CNN architecture. This confidence map highlights uncertain regions where the model is less confident in its predictions. This turns into $H * W * (\text{num\_classes} + 1)$ vector. This concatenated vector with num_classes + 1 channels is then fed to a 2-layer, 2D convolutional network, which again outputs a new set of per-pixel logits with shape $H * W * (\text{num\_classes})$. By concatenating this confidence channel with the logits and image, UARNet learns to identify both the correct and ambiguous predictions. It is trained on validation data using standard cross-entropy loss.

4. Dataset

Our project uses the ZeroWaste-f dataset, which has 4,503 fully annotated images for semantic segmentation tasks, split into 3002 training images, 572 validation images, and 929 test images, all from real, cluttered recycling factory floors. ZeroWaste has each labeled pixel belonging to one of 5 classes: Background image, rigid plastic, cardboard, metal, and soft plastic. To increase our training dataset and improve robustness, we perform both offline and online augmentations. For offline augmentation, we apply torchvision transforms: random horizontal flips (p=0.5), RandomAffine with ±5° rotation, up to 4% translation, 0.9–1.0 scaling, 5° shear, bilinear interpolation and zero padding, followed by ColorJitter (brightness/contrast/saturation ±0.2, hue ±0.1) and a 3×3 Gaussian blur. Corresponding mask transforms use nearest-neighbor interpolation and zero fill for consistency. During training, we use Albumentations for online augmentation: resizing to 256×256, horizontal flips, random Affine (scale 0.9–1.1, translate ±5%, rotation ±10°, shear ±5°), ColorJitter (same parameters), Gaussian blur (kernel 3–5, p=0.3), and ImageNet-style normalization. This two-stage pipeline ensures both training dataset size expansion to 6004 images and on-the-fly variability to improve generalization.

For our contrastive pretraining stage before supervised fine-tuning, we leverage unlabeled ZeroWaste-s frames, containing 6212 unlabeled images from a recycling facility's conveyor belt that is subject to varying lighting and occlusion. During contrastive training, we additionally applied a series of random data augmentations to generate two distinct views of each image. These included random resized cropping, horizontal flipping, color jittering, grayscale conversion, and Gaussian blur.

5. Experiments/Results/Discussion

5.1. Hyperparameters

For all our models, we trained for 10 epochs due to empirically approaching or achieving convergence, and also due to computational resource limitations.

For DeepLabV3, we used a learning rate of `1e-4` because it is a decently low value that results in stable training and convergence after we empirically tried different learn-

ing rates. We used a batch size of 64 because it balanced stable training and GPU memory limits. For our optimizer, we used the Adam optimizer because it is an adaptive learning rate method that results in stable training. U-Net used the same hyperparameters for the same reasons.

For contrastive pretraining, we used a learning rate of `3e-4` to allow our model to more quickly optimize for contrastive loss within 10 epochs. We also used a batch size of 64 because, for SimCLR, larger batch sizes are better due to there being more negative samples to compare with. We also used the Adam optimizer for the same reasons above. We used regular validation to tune our hyperparameters.

5.2. Evaluation Metrics

We evaluate our model using three standard metrics for semantic segmentation: **Intersection over Union (IoU)**, **Precision**, and **Pixel Accuracy**. All metrics are computed per class and averaged to obtain the mean score.

- **Intersection over Union (IoU)** measures the overlap between the predicted segmentation and the ground truth, normalized by their union. For class c, IoU is defined as:

$$\text{IoU}_c = \frac{TP_c}{TP_c + FP_c + FN_c}$$

  where TP_c is the number of true positive pixels for class c, FP_c is the number of false positives, and FN_c is the number of false negatives.

- **Precision** quantifies how many of the predicted pixels for class c are actually correct. This metric helps assess the model's tendency to overpredict a given class.

- **Pixel Accuracy** calculates the overall proportion of correctly classified pixels.

5.3. Experiments

We evaluated our models on the test set in the Zerowaste-f dataset. The baseline models include the predict-all-background, U-Net, and DeepLabV3 models finetuned on the Zerowaste-f training set. Additionally, we train a DeepLabV3 contrastively pre-trained on Zerowaste-s and fine-tuned on Zerowaste-f. Finally, we explore 2 additional variants of post-training (CRFPP, UARNet) for all 3 previously trained models.

5.4. Results and Analysis

Table 1 shows the per-class and mean segmentation metrics using the predict-all-background baseline. Since the majority of the pixels are background class, the IoU, precision, and pixel accuracy are all relatively high for background pixels, but everything else is zero, as expected. This is the bare minimum that future models should improve on.

Class	IoU	Precision	Pixel Acc.
Background	0.83	0.83	1.00
Rigid Plastic	0.00	0.00	0.00
Cardboard	0.00	0.00	0.00
Metal	0.00	0.00	0.00
Soft Plastic	0.00	0.00	0.00
Mean	**0.17**	**0.17**	**0.20**

Table 1. Predict-all-background Metrics

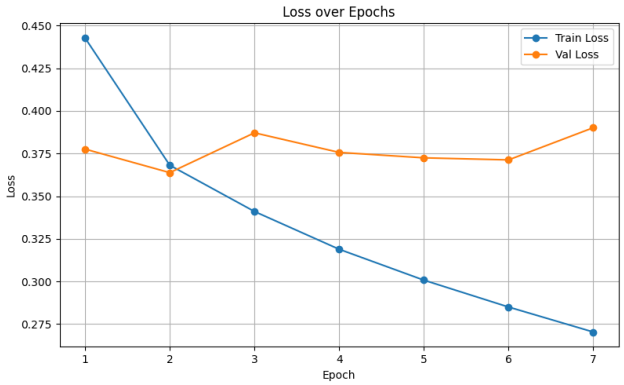

Figure 2. Training and validation loss over time for DeepLabV3.

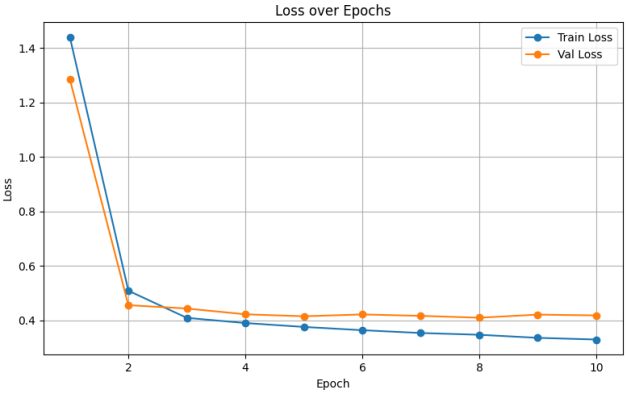

Figure 3. Training and validation loss over time for DeepLabV3 with contrastive pretraining.

Figures 2 and 3 show the loss patterns during fine-tuning for both DeepLabV3 and contrastively pre-trained DeepLabV3, respectively. DeepLabV3 without contrastive pretraining already has a very low loss after the first epoch, which demonstrates ResNet50's ImageNet encoder strength. Since we learn a new encoder with contrastive pre-training, we see the loss start high but converge after several epochs.

Table 2, 3, and 4 show the impacts of the two post-training methods on three different models: U-Net, DeepLabV3, and DeepLabV3 with contrastive pretraining, respectively. Across all three models, CRFPP drastically decreased the IoU scores for non-background classes because it tried to predict more pixels as background due to

U-Net	**Original**			**CRFPP**			**UARNet**		
	IoU	Precision	Pix. Acc.	IoU	Precision	Pix. Acc.	IoU	Precision	Pix. Acc.
Background	87.35	90.42	96.25	85.62	87.11	98.04	87.59	91.26	95.61
Rigid Plastic	05.16	46.69	05.49	00.14	30.37	00.14	00.59	59.80	00.60
Cardboard	43.41	67.01	55.21	38.03	73.45	44.10	46.32	65.54	61.24
Metal	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00	00.00
Soft Plastic	39.92	79.52	44.49	13.59	93.96	13.72	40.56	77.20	46.08
mean	35.17	56.73	40.29	27.48	56.98	31.20	35.01	58.76	40.70

Table 2. Per-class and mean segmentation metrics for U-Net that is post-trained with different models: Original, CRFPP, and UARNet.

DeepLabV3	**Original**			**CRFPP**			**UARNet**		
	IoU	Precision	Pix. Acc.	IoU	Precision	Pix. Acc.	IoU	Precision	Pix. Acc.
Background	88.51	90.96	97.05	86.43	87.49	98.61	88.64	92.30	95.71
Rigid Plastic	15.29	48.73	18.23	03.22	60.58	03.28	11.64	55.43	12.84
Cardboard	44.47	73.87	52.77	37.89	80.31	41.77	49.35	68.49	63.86
Metal	09.35	35.52	11.25	00.60	76.68	00.60	00.00	00.00	00.00
Soft Plastic	48.10	78.67	55.32	24.34	90.66	24.96	48.77	78.10	56.50
mean	41.14	65.55	46.92	30.50	79.14	33.85	39.67	58.87	45.78

Table 3. Per-class and mean segmentation metrics for DeepLabV3 that is post-trained with different models: Original, CRFPP, and UARNet.

Pre-trained DeepLabV3	**Original**			**CRFPP**			**UARNet**		
	IoU	Precision	Pix. Acc.	IoU	Precision	Pix. Acc.	IoU	Precision	Pix. Acc.
Background	86.41	90.94	94.55	86.12	87.68	97.98	87.25	90.97	95.53
Cardboard	2.33	62.93	2.36	0.22	88.23	0.22	0.00	0.00	0.00
Soft plastic	41.18	61.37	55.59	38.84	74.82	44.69	44.69	66.38	57.76
Metal	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Rigid plastic	43.48	70.09	53.39	21.78	84.56	22.68	42.44	72.99	50.55
mean	34.68	57.07	41.18	29.39	67.06	33.11	34.88	46.07	40.73

Table 4. Per-class and mean segmentation metrics for Contrastive Pre-trained DeepLabV3 that is post-trained with different models: Original, CRFPP, and UARNet.

its smoothing properties. However, this improved the precision because the pixels that are still classified as non-background even after smoothing are more likely to be correct and reduce the false positive rate. For U-Net, this trend with CRFPP was not as apparent in terms of the increases in precision; this is likely due to U-Net having a shallow encoder, so its predictions are already more fuzzy and uncertain near boundaries. CRFPP may oversmooth and degrade precision in this case. For all three models, UARNet increased the scores for classes that it was already doing well on, and pushed down the scores for classes that it was previously not doing well on. Notably, we notice that IoU scores increase for background, cardboard, and soft plastic, while drastically decreasing for rigid plastic and metal. This makes sense because UARNet uses confidence levels in its predictions. Since it was already receiving a low reward for these classes, and had low confidence in them, the final refinement CNN learned to stop predicting them, or predict them less in favor of the classes it was already highly confident in. Thus, we see that UARNet training relatively maintained IoU and pixel accuracy, while dropping overall precision. Although UARNet does not strictly improve upon these performance metrics, it presents itself as a viable alternative to the baselines for use cases where identifying the less common materials is not as important as identifying the high-quality materials.

Looking at Figure 4, we consolidate our results from Table 2, Table 3, and Table 4 to compare the mean IoU across each architecture over our post-training techniques. We find that for both DeepLabV3 with contrastive pretraining and U-Net that both the original architecture and post-training with UARNet boast very similar results, with one having UARNet barely beat out the original and vice versa. In addition, we only see a slight drop in IoU for DeepLabV3 for UARNet, suggesting that UARNet, with more tuning, has the potential to be able to make slight improvements

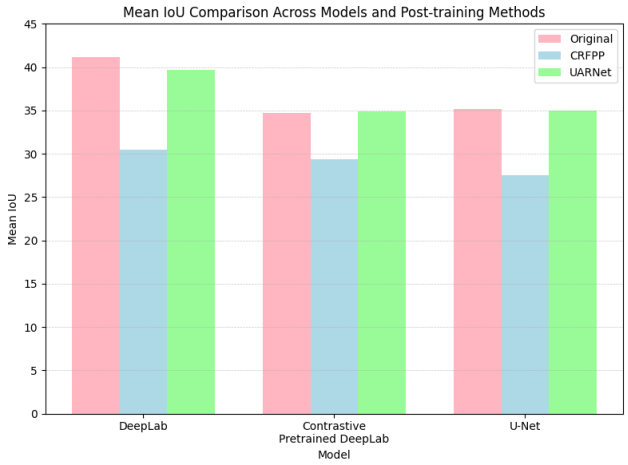

Figure 4. Comparisons on mean IoU for DeepLabV3, Contrastive Pretraining DeepLabV3 and U-Net across post-training techniques

to the original architecture. An interesting observation is how DeepLabV3 with contrastive pretraining consistently underperforms in comparison to the original DeepLabV3 model. We hypothesize that this is likely due to two factors. First, the DeepLabV3 encoder is already highly effective, leaving limited room for improvement. Second, contrastive pretraining is highly sensitive to the batch size of negative samples. Without a sufficiently large batch, it may fail to provide meaningful benefits. Additionally, negative examples in contrastive objectives may inadvertently push visually similar materials apart, negatively impacting the class segmentation performance.

Inside Figure 5, when we inspect the DeepLabV3 predictions and contrastively pre-trained DeepLabV3 predictions, we see that the pre-trained model predicts much smoother object boundaries. This makes sense as contrastive pretraining optimized the encoder to extract high-level features from the image to match images together. This is likely biased against low-level granular details, such as pixel-wise locations of each material, and instead focuses on capturing general locations and the presence of each object. Thus, the pre-trained DeepLabV3 misses out on the low-level details required for accurate semantic segmentation. However, the contrastively pre-trained DeepLabV3 is also the only model able to notice the presence of a sizeable red object class in the picture, highlighting that contrastive pre-training helped the model pick up on the high-level presence of objects better than before.

Inspecting the UARNet output image, it is able to pick up on more fine-grained details, resulting in more complex object shapes than the DeepLabV3 baseline, and it also picks up on fine-grained presence of the red object class, although it is more scattered. However, we observe that the confidence map of UARNet displays that it is significantly less confident of its background predictions than all other models, as seen by the overwhelming blue regions. UARNet's CNN learned to override its "highest" confidence input logit class in place of classes it knows it is more accurate at predicting (i.e. the background class), Thus, the UARNet learns to output the predictions it has much less confidence in, explaining why much more of the confidence heatmap is blue. Inspecting the CRFPP prediction, we notice that many of the original baseline predictions got reduced to background predictions, and the detected yellow object is more granular. This makes sense as the CRF performed smoothing, and as many objects are surrounded by background (especially at narrow protruding sections), these pixels got smoothed out to become background pixels. The heatmap also displays that CRFPP is highly confident in all of its pixels, as any pixels that were not smoothed were surrounded by enough neighboring pixels that they must have been correct. This ultimately indicates that the CRFPP model did not end up reinforcing object boundaries as expected, and instead ended up converting many pixels on an object boundary into the background class.

6. Conclusion and Future Work

In conclusion, our highest-performing model was DeepLabV3 with no contrastive pretraining or post-training. Contrastive pretraining did not improve our results, which may be attributed to high intra-class variability in the dataset and the strength of the original ImageNet-pretrained ResNet50 encoder. However, the consistently smoother object boundaries suggest contrastive pretraining is not well-suited for fine-grained semantic segmentation tasks, but may serve significant benefits towards higher-level general object detection and bounding box placement tasks. CRFPP led to smoother predictions, which increased precision but often suppressed minority classes, lowering overall IoU. CRFPP ultimately proved to be a poor choice for waste segmentation tasks due to overactivated smoothing and inability to recognize object boundaries (in favor of background pixels). UARNet demonstrated a more selective effect, improving class predictions that the model was already strong in, while degrading low-confidence classes. Despite these tradeoffs, it remained competitive across models but did not surpass the performance of the original DeepLabV3 baseline. UARNet proves itself as a viable post-training architecture in waste classification tasks that prioritize the classification of a few, more important and abundant classes.

If we had more computational resources, we would revisit contrastive pretraining with significantly larger batch sizes, as SimCLR benefits from a greater number of negative samples. Alternatively, we could explore memory-efficient contrastive learning methods such as MoCo or BYOL, which mitigate the need for large batches through momentum encoders or asymmetric architectures. We are

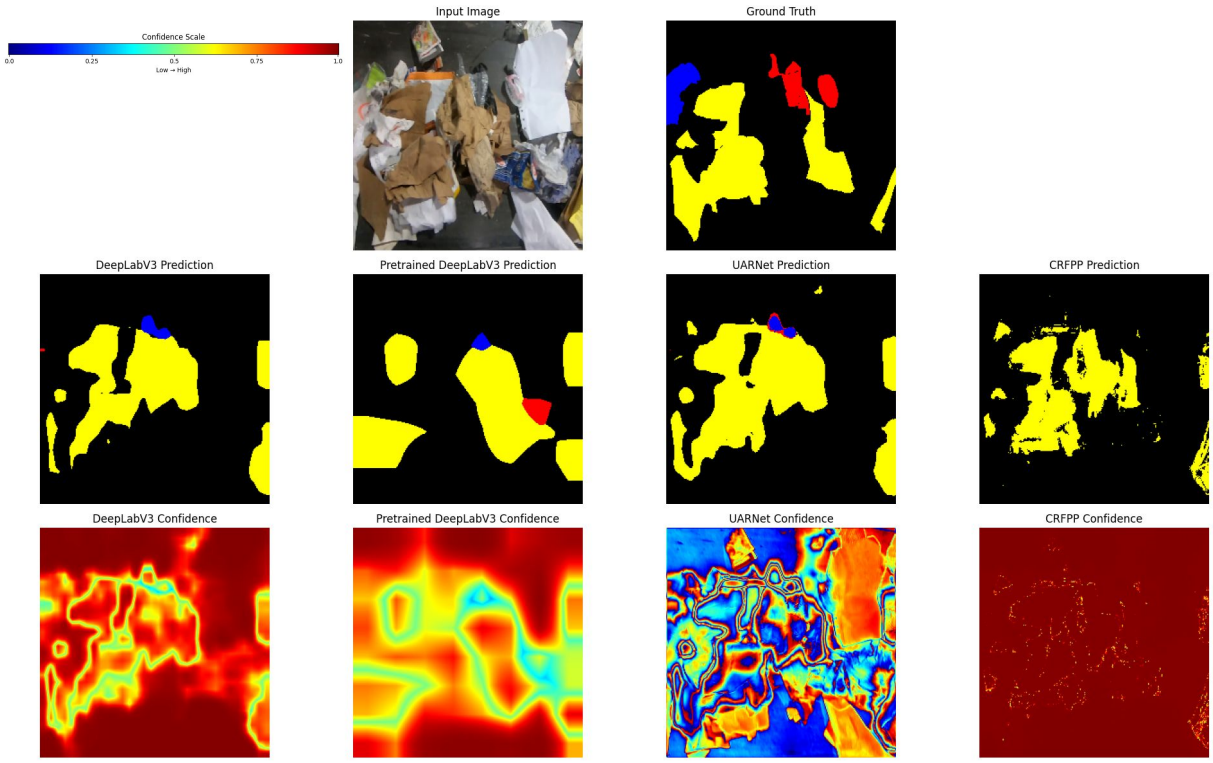

Figure 5. Qualitative example showcasing prediction maps and confidence heatmaps of DeepLabV3, Contrastive Pretraining DeepLabV3, DeepLabV3 + UARNet, DeepLabV3 + CRFPP

also interested in augmenting contrastive learning with multi-task objectives, such as edge detection or jigsaw reconstruction, to help the encoder learn richer semantic and structural features for segmentation.

Beyond pretraining, we see opportunities to improve post-processing methods. For example, class-specific CRFPP could help prevent over-smoothing of minority classes by adjusting refinement strength per class. Additionally, we can experiment with different metrics besides confidence for refinement. The confidence heuristic in UARNet proved useful for improving model performance, suggesting that exploring more complex confidence heuristics and other per-pixel heuristics may be beneficial for reweighting model outputs at test time.

A. Appendix

B. Contributions

- James Cheng: Developed Predict-All-Background, DeepLabV3, and contrastive pretraining methods. Also created evaluation metrics.
- Andy Ouyang: Conducted literature review, implemented qualitative analysis code
- Nishikar Paruchuri: Trained U-Net, UARNet, and CRFF post-processing methods.

C. Python Libaries

Package	Version
numpy	1.24.0
matplotlib	3.6.3
pandas	1.5.3
torch	2.0.1
torchvision	0.15.2
tqdm	4.64.0
PyDenseCRF	1.0
segmentation-models-pytorch	0.3.3
albumentations	2.0.8
torchmetrics	1.7.2

Table 5. Library versions used in our Colab environment.

References

[1] M. Ali, M. Javaid, M. Noman, M. Fiaz, and S. Khan. Cosnet: A novel semantic segmentation network using enhanced boundaries in cluttered scenes, 2024. 2

[2] D. Bashkirova, M. Abdelfattah, Z. Zhu, J. Akl, F. Alladkani, P. Hu, V. Ablavsky, B. Calli, S. A. Bargal, and K. Saenko. Zerowaste dataset: Towards deformable object segmentation in cluttered scenes, 2022. 1, 2

[3] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation, 2017. 3

[4] A. Marelli, L. Magri, F. Arrigoni, and G. Boracchi. Temporal-consistent cams for weakly supervised video segmentation in waste sorting. *arXiv preprint arXiv:2502.01455*, 2025. 2

[5] P. F. Proença and P. Simões. Taco: Trash annotations in context for litter detection. *arXiv preprint arXiv:2003.06975*, 2020. 1

[6] J. Qi, M. Nguyen, and W. Q. Yan. Nuni-waste: novel semi-supervised semantic segmentation waste classification with non-uniform data augmentation. *Multimedia Tools and Applications*, 83(26):68651–68669, 2024. 2

[7] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 3

[8] T. Wang, Y. Cai, L. Liang, and D. Ye. A multi-level approach to waste object segmentation. *Sensors*, 20(14), 2020. 1

[9] X. Wang, W. Han, S. Mo, T. Cai, Y. Gong, Y. Li, and Z. Zhu. Transformer-based automated segmentation of recycling materials for semantic understanding in construction. *Automation in Construction*, 154:104983, 2023. 2

[10] L. Wei, S. Kong, Y. Wu, and J. Yu. Image semantic segmentation of underwater garbage with modified u-net architecture model. *Sensors*, 22(17), 2022. 2

[11] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo. Segformer: Simple and efficient design for semantic segmentation with transformers, 2021. 2

[12] Q. Zhu, N. Weng, L. Fan, and Y. Cai. Enhancing environmental monitoring through multispectral imaging: The wastems dataset for semantic segmentation of lakeside waste. In *International Conference on Multimedia Modeling*, pages 362–372. Springer, 2025. 1