King Fahd University of Petroleum & Minerals

ICS-504 Term 251 Project: Promoting Clean Environment with Smart Waste Identification

For: Dr. Muzammil Behzad

Student Name	ID
Ammar Alsafwani	201183690
Abdullah Alshammasi	202415500
Jafar Abu Qurayn	201687520

Abstract

Litter detection has become an important problem for environmental monitoring and smart cities. The TACO: Trash Annotations in Context for Litter Detection paper dataset has 1500 images with annotations for different types of trash. The paper used Mask R-CNN model and achieved on TACO-1 around 26% AP and TACO-10 around 19% AP because the dataset is small, the number of classes is very large and it has many tiny objects. In our term project for ICS-504 shows an enhanced model version of the TACO paper where we achieve better results and performance through modern deep learning techniques that we learned from ICS-504 course this semester. Using the original TACO-1 and TACO-10 baselines that were achieved in the paper we enhanced its model by applying some enhancement techniques and we got improved results. Our improvement to the original TACO model includes:

1. Shifting to Detectron2 and PyTorch 2.8
2. Combined the classes into 4 classes (Bottle, Can, Plastic Bag, Other litter) instead of the 10 classes used in the paper or the original 60 classes to handle imbalance so our final model is 4 classes litter detector
3. Using stratified train, validation and test splits (60% train, 20% validation and 20% test)
4. Using the paper base learning rate of 0.001
5. Using batch size of 4
6. Using 3000 iteration
7. Using evaluation period of 500 iterations
8. Using COCO pretrained Mask R-CNN R50-FPN model.
9. Evaluation on held out test set of 228 images

Our improvement shows improved performance as we achieved 35.69% bbox AP and segmentation 35.19% with AP50 around 44.6%. We achieved this because we picked and combined 18 classes from the original TACO classes into 4 categories so the classification space became simpler. This reduction led to higher AP when we compared it to the baseline reported in the TACO paper where our goal was not to reproduce the original multiclass evaluation but to build a more robust classifier for practical deployment.

1. Introduction

Litter pollution is one of the serious environmental problems and it affects also smart cities. Trash like bottles, cans, plastic bags and wrappers appears on streets, beaches, parks, rivers and a lot of other places which cause environmental pollution. Using manual monitoring is very slow, expensive and hard to scale. Computer vision models can help in detecting litter in images or videos which will support the city planning and cleaning up campaigns.

The TACO dataset was created to support this type of research and help solving litter pollution problem. It contains real world images of litter that are taken in different countries and environments and annotated with detailed masks. However, training deep learning models on TACO dataset is challenging as it is small, the classes are not normally distributed and many objects are very small or partially hidden.

The original TACO paper used Mask R-CNN with a ResNet-50 FPN backbone and showed that the overall Average Precision (AP) is low. In this project, our target was to enhance the litter detection results by using Detectron2 and by simplifying the classification problem into four categories and applying some enhancements we achieved better results.

1.1 Problem Statement

Training a robust instance segmentation model on TACO is difficult because:

- The dataset has only 1500 images and 4784 annotations
- There are originally 60 classes and many of which have very few instances
- Some objects are tiny or hidden which makes detection harder

So how can we implement and train a Detectron2-based model on TACO to achieve better test performance while keeping the task reasonable and practical for deployment.

1.2 Objectives

Our main objectives are:

1. Reimplement TACO litter detection using Detectron2 instead of the original Matterport Mask R-CNN
2. Making the label simple into a smaller set of meaningful classes: Bottle, Can, Plastic bag, Other litter
3. Create a proper train, validation and test splits and evaluate strictly on the test set.
4. Train and tune a Mask R-CNN R50-FPN model and measure detection and segmentation performance
5. Compare our results with the original TACO paper and discuss why the metrics differ

1.3 Scope and Contribution

The scope of this project is limited to:

- The TACO dataset only
- 4 classes litter segmentation task
- A single model: Mask R-CNN with ResNet-50 FPN backbone implemented in Detectron2
- Offline evaluation on static images and no real time video or deployment on devices

2. Literature Review

Researchers have tried different ways to automatically detect litter in images using traditional machine learning and deep learning. Older methods used simple features and classifiers which sometimes failed because of the high variability of outdoor environments and object appearances. Today deep learning has changed the game where modern models like Faster R-CNN, YOLO and Mask R-CNN are now widely used because they can learn patterns directly from images and work better for finding and identifying trash.

The TACO dataset was proposed specifically for litter detection meaning that litter images are taken in its natural environment instead of in a clean background. This makes the task more realistic but also more difficult. The authors of the paper trained Mask R-CNN on two tasks:

- TACO-1 (classless litter vs background)
- TACO-10 (10 super-categories)

They reported best AP values around 26% for TACO-1 and around 19% for TACO-10, using 4 folds cross validation.

Other related works use variants of YOLO or custom CNNs but almost all of them face the same issues which are limited annotations, strong class imbalance and many tiny objects like cigarette.

2.1 Related Work

Mask R-CNN is a popular model for finding objects in images and drawing outlines around them. It is based on Faster R-CNN but adds an extra step to create a mask for each object. Detectron2 is a newer version of these models built with PyTorch. It makes training faster and easier by including ready to use tools and flexible settings.

The original TACO work used a Keras and TensorFlow version of Mask R-CNN. They tried improving results by adding data augmentations like rotation, blur, noise and brightness changes. They even created a tool to paste trash onto new backgrounds to make more training images. Still the model failed with small or hidden objects and performance stayed low.

2. 2 Limitation in Existing Approaches

Key limitations identified in previous work include:

- Large number of classes: 60 categories or even 10 super classes may be too many for such a small dataset
- Distribution is not normal and it is long tailed: some classes appear only once or a few times which hurts training stability
- Tiny objects: small trash like cigarettes are hard to detect after resizing images
- Complex training setups: custom frameworks and older code can be harder to maintain and improve

These issues motivated us to:

1. Simplify the task into 4 high level categories
2. Use Detectron2
3. Keep the pipeline clear so that other students can reproduce and improve it.

3. Proposed Methodology

Our methodology focuses on three aspects: task redesign, dataset preparation and Detectron2-based training.

3.1 Existing Model and Challenges

The original TACO tests were done with Mask R-CNN using the Matterport implementation. The model used a ResNet-50 FPN backbone using 1024×1024 input size and COCO pretrained weights. This method is strong but it had several challenges:

- Implementation based on older TensorFlow and Keras code
- Complex augmentation and transplantation tools
- 4 folds cross validation over many classes lead to moderate AP

These challenges make it harder to modify and extend the model, especially for new students.

3.2 Proposed Enhancements

We propose the following changes:

1. We picked 18 clear and common categories from the original 60 and grouped them into 4 main classes This approach let us avoid rare categories that have very few examples which would make training harder. By focusing on the most common types of trash our model works better than the original TACO-10 setup. The 4 classes are:
    - *Bottle* like clear plastic bottle, glass bottle and other plastic bottle

- *Can* like drink can, food can and aerosol
- *Plastic bag* like garbage bag, single use carrier bag, plastic film and wrappers
- *Other litter* like bottle caps, lids, plastic cups, cigarettes and small mixed trash

Categories that do not fall into these groups are removed from training and evaluation. This gives a simpler and more balanced task.

2. Stratified train, validation and test splits

We create one split:

- 60% training
- 20% validation
- 20% test

This split is done based on a primary class per image to keep class proportions similar in all splits. The final test set contains 228 images and 525 instances include 81 bottles, 45 cans, 147 plastic bags and 252 other litter objects.

3. Detectron2 implementation

We use Detectron2 with the configuration below instead of the original Mask R-CNN framework:

- Base model: mask_rcnn_R_50_FPN_3x (COCO pretrained)
- Number of classes: 4
- Batch size: 4 images
- Base learning rate: 0.001
- Max iterations: 3000
- LR steps: at 2000 and 2700 iterations
- Warmup iterations: 200
- Weight decay: 0.0001
- Evaluation period: every 500 iterations
- Random horizontal flip during training

4. Clear test set evaluation

After training we reload the final checkpoint (model_final.pth) and run evaluation only on the test set using the COCO evaluator. This ensures that the reported results reflect true generalization.

3.3 Algorithm and Implementation

The main steps of the algorithm are:

1. Load original TACO annotations from annotations.json
2. Apply class mapping from the 18 picked categories to 4 super classes
3. Filter annotations to keep only the selected categories

4. Compute a main class per image and perform stratified splitting into train, validation and test
5. Save three COCO style JSON files: train.json, val.json, test.json
6. Register the datasets in Detectron2 (taco_train, taco_val, taco_test)
7. Configure Mask R-CNN R50-FPN with 4 classes and the hyperparameters listed above
8. Train the model using the Detectron2 Default Trainer
9. Evaluate on the validation set during training every 500 iterations to monitor progress
10. Reload the final model and run test set evaluation plus visualization
11. Run a test using image taken by us

3.4 Loss Function and Optimization

The model uses the standard Mask R-CNN multitask loss:

- Classification loss using cross entropy for the ROI head
- Bounding box regression loss using smooth L1
- Mask loss using binary cross entropy for each instance
- RPN objectness and box regression losses

Training is done using stochastic gradient descent (SGD) with momentum as defined in the Detectron2 default config, base learning rate 0.001 and learning rate decay at 2000 and 2700 iterations. The total training took 3000 iterations on Google Colab with a single GPU. The training and evaluation completed without numerical errors according to the logs.

4. Experimental Design and Evaluation

4.1 Dataset and Preprocessing

We use the original TACO images and all images are high resolution taken in real outdoor environments. Preprocessing steps are:

- Conversion of the original annotation file to 4 classes COCO format
- Removal of unneeded categories
- Splitting into train, validation and test as described earlier

During inference Detectron2 applies resize shortest edge length to (800, 800) and max size is 1333. This rescales images so that the shorter side is 800 pixels up to a max size of 1333.

The test set contains 228 images with the class distribution printed in the logs as below:

- Bottle: 81 instances
- Can: 45 instances
- Plastic bag: 147 instances
- Other litter: 252 instances
- Total: 525 instances

4.2 Performance Metrics

We use the standard COCO metrics for detection and segmentation:

- AP: Average Precision over IoU thresholds from 0.50 to 0.95 and the step is 0.05
- AP50: AP at IoU = 0.50
- AP75: AP at IoU = 0.75
- APs, APm, APl: AP for small, medium, and large objects
- AR: Average Recall at different maximum detections

Detectron2 prints both the global metrics and per class APs for each annotation type bbox and segmentation.

4.3 Experiment Setup

We use Google Colab with a single GPU and the key training configurations as printed in the notebook are:

- Base learning rate: 0.001
- Max iterations: 3000
- Batch size: 4 images per iteration
- Evaluation period: every 500 iterations
- Number of classes: 4

The final model checkpoint model_final.pth was then used for test set evaluation.

4.4 Results and Comparative Analysis

4.4.1 The Final Evaluation Logs on the Test Set Report:

Bounding box detection on test set:

AP (0.50:0.95):	35.686%
AP50:	44.558%
AP75:	41.281%
APs:	3.30% (small objects)
APm:	18.83% (medium objects)
APl:	42.25% (large objects)

Per class bbox AP:

Bottle:	57.43%
Can:	41.65%
Plastic bag:	31.58%
Other litter:	12.08%

Segmentation mask on test set:

AP (0.50:0.95):	35.194%
AP50:	44.671%
AP75:	38.310%
APs:	2.01%
APm:	18.94%
APl:	42.40%

Per class mask AP:

Bottle:	57.82%
Can:	40.78%
Plastic bag:	30.79%
Other litter:	11.38%

These results show that the model performs very well on bottles and cans, reasonably on plastic bags and less strongly on the other litter class which is expected because that class is very diverse and contains many small or unclear items.

4.4.2 Visual Comparison

We generated visualizations of model predictions on several TACO test images. Some examples:

- In one image a plastic bag lying on the pavement is correctly detected and segmented while small distant pieces of trash are missed
- In another image multiple bottles and cans scattered on the ground are all detected with high confidence and the masks align closely with the object boundaries

- Some difficult images with transparent plastic or very small litter items produce zero detections which explains the low AP for small objects

Image 819: 1 detections

Image 1333: 5 detections

Image 466: 0 detections

Image 148: 1 detections

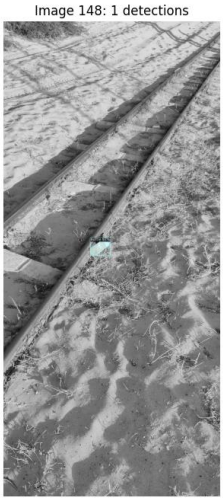
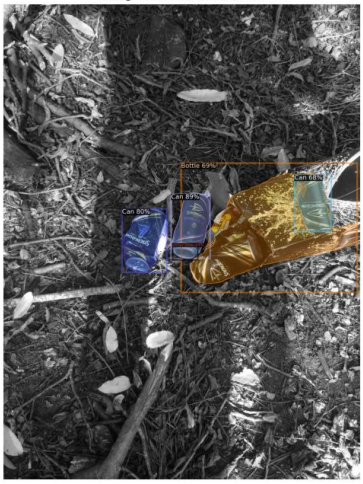


Image 1431: 0 detections

Image 1029: 5 detections



We also tested the trained model on a photo taken by us not from TACO. The image shows two drink cans and a bottle on a textured surface. The model:

- Detected both cans with high confidence around 96 to 97
- Detected the bottle correctly
- Detected the top area of the bottle as other litter due to label appearance

This qualitative result shows that the model generalizes beyond the dataset and works on new images.

4.4.3 Comparison with the Original TACO Paper

The TACO paper reports best AP values of:

- Around 26% AP for the classless task in TACO-1
- Around 19% AP for the 10-class task in TACO-10

Our model achieves 35.2-35.7% AP on 4 classes task. The difference in metrics should be interpreted carefully:

- We simplified the label space to four categories which makes the problem easier and naturally increases AP.
- We evaluated on a single stratified train, validation and test splits not on 4 folds cross validation.

So, we do not claim a direct drop in improvement over TACO-10. Instead we show that for 4 classes practical scenario the Detectron2 with our training setup can reach high performance that is higher than the numbers reported for the more complex tasks in the original work.

5. Extended Contributions

This project provides beyond the raw AP numbers:

1. A clean Detectron2 pipeline for TACO with code for class mapping, stratified splitting and dataset registration
2. A practical 4 classes formulation (Bottle, Can, Plastic bag, Other litter) that aligns well with real monitoring tasks and is easier for city operators to understand
3. Evidence that even with a relatively small dataset, a carefully designed model and label space can give strong detection and segmentation performance on real world litter
4. Example visualizations and an external test image which demonstrate that the model can process new images beyond the training set

6. Conclusion and Future Work

In this project, we enhanced litter detection on the TACO dataset using Detectron2 and a simplified 4 classes label. We designed a stratified train, validation and test splits, trained a COCO pretrained Mask R-CNN R50-FPN model and evaluated it strictly on the test set. The model achieved around 35% mAP for both bounding boxes and masks with especially strong performance on bottles and cans.

These results are higher than the AP values reported in the original TACO paper mainly because our task is simpler and uses fewer classes. Still, the project shows that Detectron2 is an effective framework for this problem and that a smaller, well chosen label set can be a good option for practical litter monitoring systems.

For future work we suggest the following:

- Extending the pipeline to TACO-10 while keeping the same Detectron2 setup to allow a direct comparison with the original paper.
- Using stronger data augmentation like Albumentations to improve detection of small objects like cigarette butts.
- Trying larger backbones or transformer based detectors like Swin Transformer, YOLOv8-Seg or DETR-style models.
- Exploring weakly supervised or semi supervised learning to handle the data with unlabeled data.

- Deploying the trained model in a simple web app or mobile and drone prototypes.

7. References

1. Proença, P. F., & Simões, P. (2020). *TACO: Trash annotations in context for litter detection*. arXiv preprint arXiv:2003.06975.
2. He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, 2961-2969.
3. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 91-99.
4. Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. *Proceedings of the IEEE International Conference on Computer Vision*, 2980-2988.
5. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, 248-255.
6. Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., & Girshick, R. (2019). *Detectron2*. https://github.com/facebookresearch/detectron2
7. Singh, B., & Davis, L. S. (2018). An analysis of scale invariance in object detection: SNIP. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3578-3587.
8. Girshick, R. (2015). Fast R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, 1440-1448.
9. Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779-788.
10. Fulton, M., Hong, J., Islam, M. J., & Sattar, J. (2019). Robotic detection of marine litter using deep visual detection models. *2019 International Conference on Robotics and Automation (ICRA)*, 5752-5758.
11. Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The PASCAL visual object classes (VOC) challenge. International Journal of Computer Vision, 88(2), 303-338.
12. Huang, Z., Huang, L., Gong, Y., Huang, C., & Wang, X. (2019). Mask scoring R-CNN. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 6409-6418.
13. Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. Journal of Big Data, 6(1), 1-48.

14. Zhang, H., Cissé, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). Mixup: Beyond empirical risk minimization. International Conference on Learning Representations (ICLR).
15. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). SSD: Single-shot multibox detector. European Conference on Computer Vision, 21-37.
16. Tan, M., & Le, Q. (2020). EfficientDet: Scalable and efficient object detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 10781-10790.
17. Islam, M. J., Xia, Y., Sattar, J., & Li, H. (2020). Detecting marine debris in the deep sea with YOLOv3 and transfer learning. IEEE International Conference on Robotics and Automation (ICRA), 7960-7966.