Detecting and classifying bus stop trash cans using camera-equipped public transit vehicles

Tim Storm[1] and Christoph Mertz[2]

***Abstract*— Trash cans are a central tool in managing the disposal of trash in urban areas, but require human supervision to ensure regular emptying. It is difficult to manage a large number of waste bins spread across a whole city, which presents an opportunity for computer vision technology to identify cans that require attention without human intervention.**

Previous work has leveraged a camera-equipped bus to deploy a single deep learning based computer vision model to detect trash cans along the path of the bus and classify their fill level. We improve upon their work by presenting a multi-stage pipeline that combines their detection model with a separate, second model trained purely for classification. The detector will identify and cut out trash cans from an image, which will then be classified as either "Empty", "Full" or "having a garbage bag next to it". Our approach significantly increases the overall accuracy and precision for both tasks, as calculated by the commonly used COCO metrics. Additionally, we present a lightweight variant of our detection model, which can be run on the bus itself, where only limited computational resources available. This enables us to actually deploy our system in a near real-time setting.

***Index Terms*— Computer Vision for Transportation, Intelligent Transportation Systems, Environment Monitoring and Management, Object Detection, Segmentation and Categorization**

I. INTRODUCTION

The comprehensive report "What a Waste 2.0" published in 2018 estimates that by 2050 waste generation rates will outpace population growth by a factor of two, which will pose large challenges in the solid waste management sector, especially in low-income countries [1].

This is a considerable point of costs in many cities: According to that same report, solid waste management takes up to 19 percent of a city's budget, depending on the area. On average 60 to 70 percent of total operational costs can be accounted to the task of waste collection.

Even in high-income areas waste collection typically relies on designated garbage pickup schedules and human supervision, but these methods are highly inflexible in adjusting to short-term trends and scaling to large areas. These can be serious problems when demand for garbage collection grows, as cities and population increase in size.

Planning-based systems can help in determining effective routes to help garbage disposal companies attend to the trash cans that actually need to be emptied, but they require information about the state of trash cans in an area.

To address this problem, we implement a system, which uses camera-equipped public transport buses to continuously monitor trash cans along a bus route. This approach utilizes existing infrastructure and ensures reasonably accurate measurements through regular traversal of the same areas.

We aim to deploy this system on the BusEdge platform [2]. BusEdge controls a number of sensors and safety cameras on the bus itself and allows accessing and working with the resulting data without any additional hardware, by offloading the computational load to a nearby Cloudlet instead.

In particular we are presenting a three-stage detection-and-classification pipeline, which is able to detect trash cans along the side of the road and estimate their fill level based on their outward appearance, while respecting the resource restrictions imposed by BusEdge.

To do so, we apply multiple computer vision models based on the ResNet architecture [3] trained with Detectron2, an open source library that provides a number of state-of-the-art object detection algorithms [4].

This is a continuation of previous work [5], which proposed this pipeline and implemented parts of it. We provide an analysis of commonly accepted metrics like precision and recall, and show how our changes improve the previous results.

Once deployed our pipeline can assist an employee of the bus company by automatically providing an overview of all existing trash cans, monitoring their fill level over time and suggesting bus stops where trash cans are or will soon be too full.

II. RELATED WORK

A. IoT-based Trash level monitoring

In order to improve effective emptying of waste containers a number of Internet of Things (IoT) solutions have been proposed [6]–[9], that aim to build "smart" trash cans, equipped with sensors and a network connection, which allow for reliable remote monitoring. These systems can track trash levels much more accurately than a computer vision approach, but they have the distinct disadvantage of requiring specialized hardware.

This makes retrofitting a city with such devices a tedious endeavor and requires additional maintenance, which makes scaling up difficult.

Our approach uses existing public infrastructure, which makes deployment easier and more flexible.

B. Computer vision for trash detection

In the past there have been many attempts to utilize computer vision methods to fight increased pollution, both

[1]Tim Storm is with University of Paderborn, Paderborn, Germany tistorm@mail.upb.de

[2]Christoph Mertz is with the NavLab, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA cmertz@andrew.cmu.edu

in terms of localizing and classifying it. [10], [11]
Machine learning techniques such as Support-vector machines (SVMs) [12] or Convolutional Neural Networks (CNNs) [13], [14] were used to detect littering in both urban, as well as aquatic enviroments [15]. The TACO [16] data set offers a good starting point for detecting waste left in nature. Other research directions include classifying trash to help with recycling [17], [18].
While such methods certainly help with improving municipal cleanliness, they do not account for waste, that has been properly disposed of and that is ready for collection.

C. Deep Learning in computer vision

In recent times convolutional neural networks have been out-performing more classical computer vision algorithms, which has shifted the field towards heavy use of deep learning based methods.
A number of frameworks [4], [19], [20] at varying degrees of abstraction have opened up the field by providing pre-trained models for different architectures and applications.
The task of object detection in particular allows for a wide range of combinations: We will be using the Faster R-CNN architecture of the R-CNN family [21]–[23], a two-stage detector with an emphasis on accuracy, readily available within our framework of choice - Detectron2 [4]. Depending on the application, many different backbones can be used for feature extraction. A common choice are entries from the ResNet family [3], but other choices such as Inception [24], [25] or combinations of the two [26] exist.
These architectures, while highly accurate, tend to be quite slow, which is unsuitable for mobile and embedded devices. This has created other approaches with an emphasis on speed and efficiency. These range from more efficient backbones like the MobileNets family [27] to a class of single-stage detectors like SSD [28], YOLO [29] or RetinaNet [30], that aim for (near) real-time application.

III. OVERVIEW

We propose a three-stage pipeline in order to detect trash cans in a real-time context: On-the-edge preselection, Trash can detection and Trash can classification. The pipeline is designed to be deployed on the BusEdge platform proposed by [2]. This section largely recaps the previous work by [5].

A. BusEdge

BusEdge [2] provides a framework for us to work with data collected by sensors on a common public transportation bus. This includes information such as images from exterior cameras, GPS, acceleration etc. Notably the bus itself is equipped with a computer not capable of (and not designed to) executing computationally intensive tasks. Instead a typical BusEdge pipeline uses only lightweight filters that run on the bus itself. Data that passes through the filter is then sent to a more capable ”cognitive engine” via wireless network, where a thorough analysis is possible.

B. On-the-edge preselection

The large amount of incoming data from the cameras makes it infeasible to consider every frame equally. Instead we want to concentrate our efforts on a promising subset of all incoming data.
The largest limitation in this context are the limited computational resources available: The limited bandwidth requires us to perform this preselection on the bus itself, since we cannot pass all images onto the cloudlet server.
We can address this issue by applying a lightweight detection model to roughly detect possible trash cans and then analyze the candidates more precisely later. Instead of being very precise, we are aiming for high recall at this stage. This leaves us with images that may contain a trash can, while discarding the ones that clearly do not. The whole step has to be performed reasonably fast, even with the lack of a dedicated graphics processing unit (GPU) typically available in a machine learning setting, so that the whole pipeline can run in near real-time.
For that reason, we want to look at efficient models specially designed for the deployment on mobile devices such as MobileNets [27] to account for the bus' hardware limitations.

C. Detection

Our main goal in this stage is to accurately detect and localize trash cans, regardless of their trash level, in the previously identified images.
This step will be performed on the cloudlet server, which is why we can apply more complex and computationally expensive models like RetinaNet [30] or ResNet [3] to achieve highest possible accuracy.
Previous experiments [5] have shown that, while architectures like Faster R-CNN are able to both localize and classify, the overall accuracy decreases heavily when relying on a combined model.
That is why we cut out the identified bounding boxes and feed them into a separate final stage instead.

D. Classification

The final stage of our pipeline is responsible for classifying the cropped images into three separate categories:
Trash cans that do not visibly contain trash will be considered empty for our purposes, as they are not in need of immediate attention. Trash cans that do visibly contain trash are considered (soon-to-be) full for our purposes. Since the bus camera can only see the outside of the cans, visible trash is an indicator of reaching maximum capacity. That means they will be ready to be emptied in the near future and are therefore of interest to the bus company. Our third class covers trash cans with a garbage bag in their immediate vicinity, regardless of their trash level, as garbage bags are indicative of a can that needs immediate attention, since trash is piling up beyond the contents of the can. To account for this class the cropped out bounding boxes are extended into all directions to include the surrounding area.
Since we can rely on human assistance for ambiguous images, we will assign a fourth class for predictions that fall

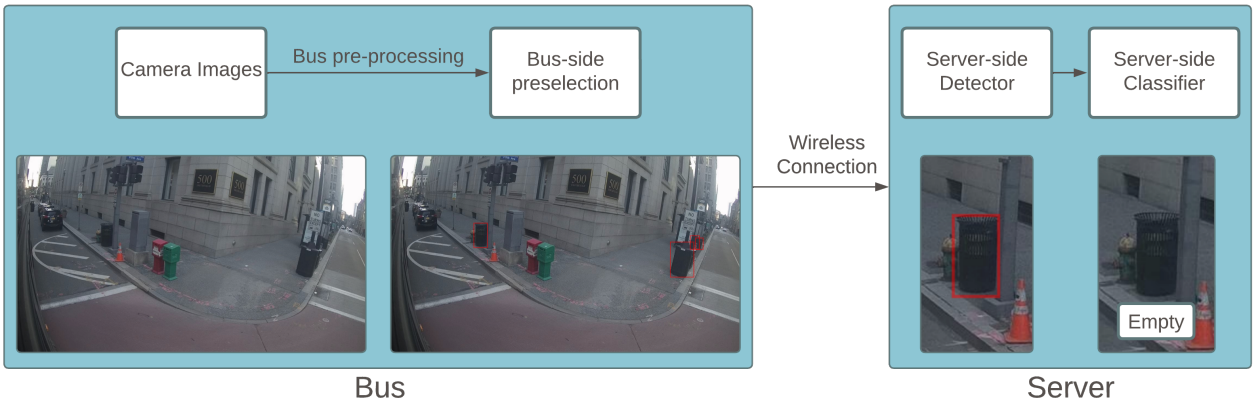


Fig. 1: Proposed detection and classification pipeline

below a certain confidence threshold. Under the (reasonable) assumption that a human operator will always be accurate, this option presents a trade-off between human effort and overall performance by relying on the operator's judgement for all predictions under the threshold.

E. Notifcation

In order for the proposed pipeline to be useful to the bus company, an additional stage is required: The responsible party has to be notified about the location of trash cans, which we deem to need attention. The BusEdge platform offers GPS data, which will be used for this purpose.
Our system provides all the necessary information for such an application to be built on top of it.

IV. METHODOLOGY

A. Data sets

1) Regarding existing data sets: We adapted existing data sets [31], [32] from previous work [5] for our purposes:
The data sets consist of 14,981 images of which 2682 images contain 3909 annotations. In the first case, the annotations assign a single label to every object of interest, whereas in the second case, they distinguish between "Full", "Empty" and "Garbage Bag".
It has to be noted, that the annotations used here do not distinguish between between domestic (movable) trash cans and permanently installed trash cans. We used the full provided data set for training purposes, but since we are mainly interested in permanent installations at bus stops, we disregard any failure cases related to other trash cans in our analysis (unless explicitly stated otherwise).

2) Detection data set: To train and test our detection models, we used the unaltered data set, as available at [31].

3) Classification data set: To train and evaluate our classification model, we had to make a few changes to the provided data set [32]. Since this task does not expect bounding boxes, we cut out the annotated bounding boxes from the existing images to use instead. To account for the "Garbage Bag" label, we extended the given bounding box by 25% horizontally and 10% vertically in each direction. If this extended bounding box contained a garbage bag annotation we assigned the appropriate "garbage bag" label, falling back to the original annotation otherwise.
This method produced a number of images not even classifiable by humans, which is why we then filtered out all images smaller than 32 pixels in either dimension (considered "small" by COCOeval [33]) For the reasons outlined in paragraph IV-A.1, we mainly look at a variant of our test set, which only contains bus stop trash cans.

B. Model training

We used our data sets as described in the previous section to finetune models from the ResNet family [3], as well as the MobileNet family [27]. All used models have been pre-trained on the ImageNet [34] data set.

1) Detection: Our server-side detector was trained using the Resnet101+FPN backbone for Faster R-CNN as proposed in [35], available within the Detectron2 model zoo [4], which we trained for approximately 30,000 iterations, while evaluating the validation set every 1,000 iterations.
The bus-side detector relies on the same architecture, but we compare different, simpler backbones capable of running on mobile devices, including Resnet18+FPN and MobilenetV2.

2) Classification: Our classifier is based on Resnet101 again: We only replaced the final fully-connected layer with a smaller fully-connected layer to account for our three classes. We also applied the Softmax function, so that the output becomes a probability vector.
We trained this model for around 10 epochs.

C. Evaluation

1) Detector: We report COCO-style metrics [33], as well as a precision-recall curve to evaluate the performance of our detection models.
Our use case needs only a very rough localization within an image, which is why we give special attention to the metrics

at an Intersection over Union (IoU) value of 50%. For the bus case we additionally benchmark the inference time in a CPU-only setting averaged across multiple runs.

2) Classifier: To evaluate classification performance, we provide overall precision and recall values for our three desired classifications on the bus stop-only test data. These are be partially visualized in a confusion matrix to highlight possible failure cases.
Additionally we compared these metrics given different confidence thresholds for human intervention as described in section III-D.
Calculating the CLASS-BALANCED-ACCURACY [36] allows us to a give a single, comparable value here, while accounting for the class imbalance of the data set.

V. RESULTS

A. Server-side Detection

When applied to our test set, we achieved an overall average precision of 78.3% at an overall average recall of 85.0%. The precision increased to 89.2% when looking at IoU 0.50.
We also saw a significant increase of over twenty percentage points across all categories, compared to the Retinanet model presented by [5] (see Table I).

The precision-recall curve shows promising results across all threshholds (see blue curve in Figure 2). This becomes even more apparent, if we limit ourselves to large (greater than 96 by 96 pixels) bounding boxes (see orange curve in Figure 2). These are especially interesting to us, since we expect to get close-up images of all bus stop trash cans when driving past them.

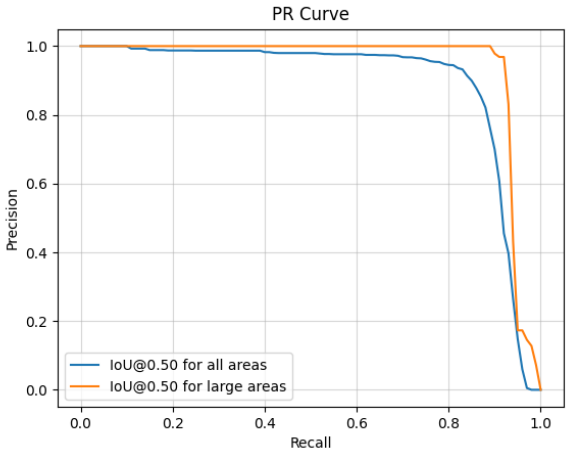


Fig. 2: Better model performance in the especially valuable ”large” case

However there is a small amount of trash cans that we never manage to detect. By looking at these images, we managed to identify a failure cases: Occlusion.
We expect each trash can to appear in full view at least once, but as Figure 3 shows this may not always the case.

Fig. 3: Occluded trash can (green bounding box) is only visible for a single frame. In the next recorded frame (bottom), the bus has already passed the trash can.

B. Bus-side detection

Finetuning both backbones showed that the Resnet18 backbone surpassed the MobileNets-V2 variant in all our core categories (see table I): While there was a slight difference in inference time between the two, the difference is not large enough to make up for the additional accuracy.

A look at the precision recall curve (Figure 4) shows excellent performance of our lightweight model, when considering ”large” trash cans, rivaling the recall of the server model with only slightly lower precision.

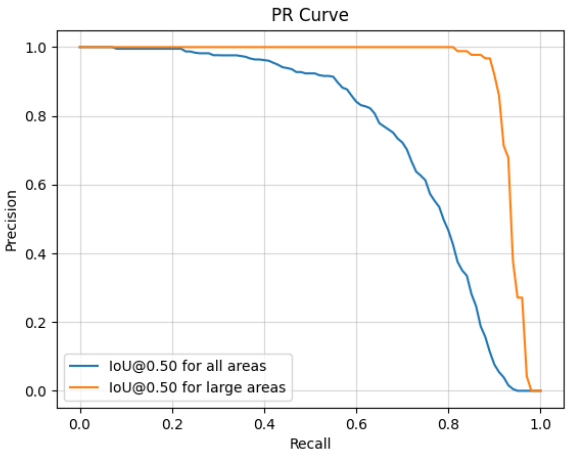


Fig. 4: Lightweight model performs exceptionally well in ”large” case, even compared with the server-side detector

TABLE I: Resnet18 achieves higher results across the board

Backbone	AP	AP50	APm	APl	Inference (on CPU)	Inference (on GPU)
Resnet101-FPN	78.3%	89.2%	84.16%	88.55%	3111ms	152ms
RetinaNet	44.2%	67.4%	57.3%	79.2%	N/A	N/A
Resnet 18-FPN	50.13%	**74.41%**	61.28%	77.09%	604ms	34ms
MobileNetsV2	35.09%	55.36%	49.68%	62.58 %	556ms	37ms

By applying the model as a pre-selector at a threshold of 0.5, our combined data set of 14981 individual frames was cut down to a set of 2191 candidate frames (approximately 14.6%), while retaining close to 95% of all large instances.

C. Classification

Evaluating the classifier on our reduced test set (only including bus stops), we see great results in the Garbage Bag and Empty category, with the weakest performance in the Full category (see table II).

This equates to an BALANCED-ACCURACY of 0.904.

TABLE II: Performance of our classifier on the reduced test set

Label	Precision	Recall	Support
Empty	98.52%	98.52%	270
Full	86.96%	90.91%	22
Garbage Bag	98.90%	97.83%	92

The confusion matrix (Fig. 5) highlights this: Full trash cans are mistakenly identified as empty around ten percent of the time. While these are acceptable results, the bus company is especially interested in these trash can.

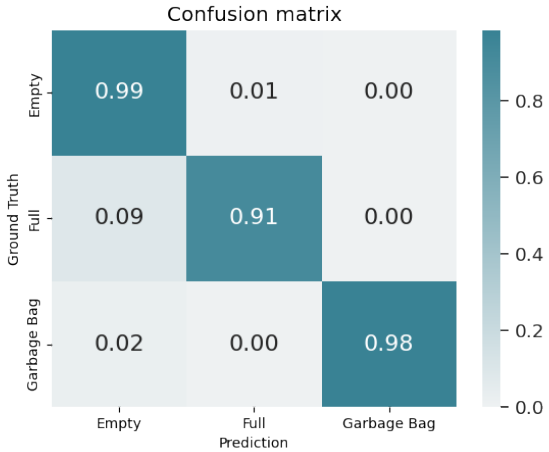


Fig. 5: One tenth of full trash cans are falsely predicted to be empty.

We found that the results can be improved by applying a minimum confidence threshold (as described in section III-D). We found the best results at a threshold of 87% (see Figure 6): With human intervention, we are able to correctly identify all full trash cans, while barely affecting the results

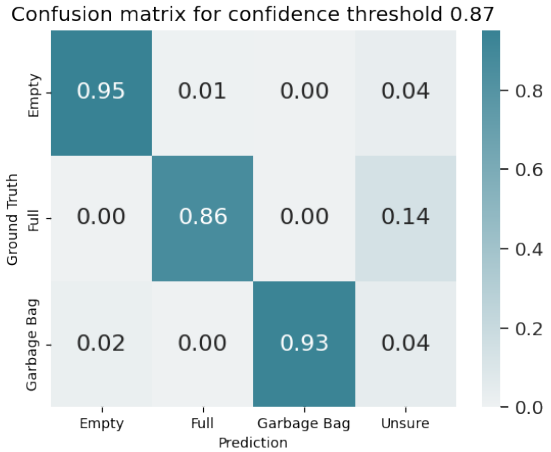


Fig. 6: A threshold of 87% allows us to find all full trash cans.

in the other categories. This translates to an improved BALANCED-ACCURACY score of 0.953. As a trade-off we required the operator's judgement for 15 images, which accounts for around 4 percent of all 384 images in the data set.

We found a similar trend when applying the classifier to the full test set (including every kind of trash can), but saw a significant increase in the fraction of misidentified full cans, where the confidence threshold proved to be less effective (see Appendix Figure 7). While there is a definite area of improvement, it serves as evidence, that our approach can be adapted for other kinds of waste containers.

VI. CONCLUSIONS AND FUTURE WORK

We presented a three stage pipeline for detecting and classifying trash cans, designed for deployment in an edge-computing context. Our model significantly outperforms previous results presented on this problem and the results indicate satisfying results in a real-world application. We also provide evidence, that our results may generalize to a wider variety of waste containers.

So far we have not tested our system in actual deployment. While we have tried to account for the limited resources, further experimentation is still required. Depending on the results we may want to consider other architectures such as SSD [28] or YOLO [29] in the pre-selection step.

ACKNOWLEDGMENT

Tim thanks Anurag Ghosh for his many helpful suggestions. Tim thanks Christoph for his mentorship and general guidance. Tim thanks Rachel and John for being able to participate in the RISS program, as well as the DAAD for its generous financial support, which made the experience possible in the first place. Data and background software were provided by projects sponsored in part by NSF under Award No 2038612 and Carnegie Mellon University's Mobility21 National University Transportation Center, which is sponsored by the US Department of Transportation.

REFERENCES

[1] S. Kaza, L. C. Yao, P. Bhada-Tata, and F. Van Woerden, *What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050*. Washington, DC: World Bank, Sept. 2018, accepted: 2018-08-31T16:25:22Z. [Online]. Available: https://openknowledge.worldbank.org/handle/10986/30317

[2] C. Ye, "BusEdge: Efficient Live Video Analytics for Transit Buses via Edge Computing," *The Robotics Institute Carnegie Mellon University*, July 2021. [Online]. Available: https://www.ri.cmu.edu/publications/busedge-efficient-live-video-analytics-for-transit-buses-via-edge-computing/

[3] "Deep Residual Learning for Image Recognition," Dec. 2015, arXiv:1512.03385 [cs]. [Online]. Available: http://arxiv.org/abs/1512.03385

[4] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," 2019. [Online]. Available: https://github.com/facebookresearch/detectron2

[5] E. Rotondo and C. Mertz, "Detecting and Classifying Waste Bin Garbage Levels Along Transit Bus Routes," *RISS Working Journal*, 2021.

[6] Y. Zhu, G. Jia, G. Han, Z. Zhou, and M. Guizani, "An NB-IoT-based smart trash can system for improved health in smart cities," in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, June 2019, pp. 763–768, iSSN: 2376-6506.

[7] S. R. J. Ramson, S. Vishnu, A. A. Kirubaraj, T. Anagnostopoulos, and A. M. Abu-Mahfouz, "A LoRaWAN IoT-Enabled Trash Bin Level Monitoring System," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 786–795, Feb. 2022, conference Name: IEEE Transactions on Industrial Informatics.

[8] S. Kristanto, T. Yashiro, N. Koshizuka, and K. Sakamura, "Dynamic Polling Algorithm for Low Energy Garbage Level Measurement in Smart Trash Bin," in *Proceedings of the Second International Conference on IoT in Urban Space*, ser. Urb-IoT '16. New York, NY, USA: Association for Computing Machinery, 2016, pp. 92–94. [Online]. Available: https://doi.org/10.1145/2962735.2962748

[9] H. Sohail, S. Ullah, A. Khan, O. B. Samin, and M. Omar, "Intelligent Trash Bin (ITB) with Trash Collection Efficiency Optimization using IoT Sensing," in *2019 8th International Conference on Information and Communication Technologies (ICICT)*, Nov. 2019, pp. 48–53.

[10] R. J. Firby, R. E. Kahn, P. N. Prokopowicz, and M. J. Swain, "Collecting Trash: A Test of Purposive Vision," p. 20.

[11] Y. Wang and X. Zhang, "Autonomous garbage detection for intelligent urban management," *MATEC Web of Conferences*, vol. 232, p. 01056, 2018, publisher: EDP Sciences.

[12] I. Salimi, B. S. Bayu Dewantara, and I. K. Wibowo, "Visual-based trash detection and classification system for smart trash bin robot," in *2018 International Electronics Symposium on Knowledge Creation and Intelligent Computing (IES-KCIC)*, Oct. 2018, pp. 378–383.

[13] M. S. Rad, A. von Kaenel, A. Droux, F. Tieche, N. Ouerhani, H. K. Ekenel, and J.-P. Thiran, "A Computer Vision System to Localize and Classify Wastes on the Streets," in *Computer Vision Systems*, ser. Lecture Notes in Computer Science, M. Liu, H. Chen, and M. Vincze, Eds. Cham: Springer International Publishing, 2017, pp. 195–204.

[14] N. Jayawickrama, "Detecting trash and valuables with machine vision in passenger vehicles," Aug. 2020, accepted: 2020-08-23T17:20:37Z. [Online]. Available: https://aaltodoc.aalto.fi:443/handle/123456789/46148

[15] M. Fulton, J. Hong, M. J. Islam, and J. Sattar, "Robotic Detection of Marine Litter Using Deep Visual Detection Models," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 5752–5758, iSSN: 2577-087X.

[16] "TACO: Trash Annotations in Context for Litter Detection," Mar. 2020, arXiv:2003.06975 [cs]. [Online]. Available: http://arxiv.org/abs/2003.06975

[17] S. Zhang, Y. Chen, Z. Yang, and H. Gong, "Computer Vision Based Two-stage Waste Recognition-Retrieval Algorithm for Waste Classification," *Resources, Conservation and Recycling*, vol. 169, p. 105543, June 2021. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0921344921001506

[18] M. Yang and G. Thung, "Classification of Trash for Recyclability Status," p. 6.

[19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[20] F. Chollet and others, "Keras," 2015. [Online]. Available: https://keras.io

[21] "Rich feature hierarchies for accurate object detection and semantic segmentation," Oct. 2014, arXiv:1311.2524 [cs].

[22] "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," Jan. 2016, arXiv:1506.01497 [cs]. [Online]. Available: http://arxiv.org/abs/1506.01497

[23] "Mask R-CNN," Jan. 2018, arXiv:1703.06870 [cs]. [Online]. Available: http://arxiv.org/abs/1703.06870

[24] "Going Deeper with Convolutions," Sept. 2014, arXiv:1409.4842 [cs]. [Online]. Available: http://arxiv.org/abs/1409.4842

[25] "Rethinking the Inception Architecture for Computer Vision," Dec. 2015, arXiv:1512.00567 [cs]. [Online]. Available: http://arxiv.org/abs/1512.00567

[26] "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," Aug. 2016, arXiv:1602.07261 [cs]. [Online]. Available: http://arxiv.org/abs/1602.07261

[27] "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," Apr. 2017, arXiv:1704.04861 [cs]. [Online]. Available: http://arxiv.org/abs/1704.04861

[28] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," 2016, vol. 9905, pp. 21–37, arXiv:1512.02325 [cs]. [Online]. Available: http://arxiv.org/abs/1512.02325

[29] "You Only Look Once: Unified, Real-Time Object Detection," May 2016, arXiv:1506.02640 [cs]. [Online]. Available: http://arxiv.org/abs/1506.02640

[30] "Focal Loss for Dense Object Detection," Feb. 2018, arXiv:1708.02002 [cs]. [Online]. Available: http://arxiv.org/abs/1708.02002

[31] "Waste Bin Detection Dataset (RISS2021)." [Online]. Available: https://www.kaggle.com/datasets/elirotondo/waste-bin-detection-dataset-riss-2021

[32] "Waste Bin Multi-Class Detection Dataset (RISS2021)." [Online]. Available: https://www.kaggle.com/datasets/elirotondo/waste-bin-multiclass-detection-dataset-riss2021

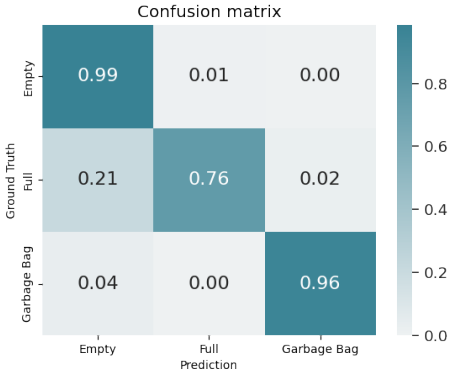
[33] "Microsoft COCO: Common Objects in Context," Feb. 2015, arXiv:1405.0312 [cs]. [Online]. Available: http://arxiv.org/abs/1405.0312

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp. 248–255, iSSN: 1063-6919.

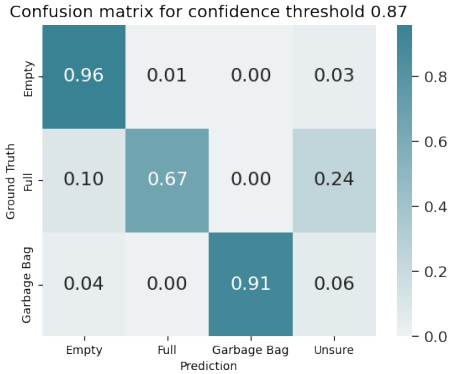
[35] "Feature Pyramid Networks for Object Detection," Apr. 2017, arXiv:1612.03144 [cs]. [Online]. Available: http://arxiv.org/abs/1612.03144

[36] L. Mosley, "A balanced approach to the multi-class imbalance problem," Doctor of Philosophy, Iowa State University, Digital Repository, Ames, 2013, pages: 5050377. [Online]. Available: https://lib.dr.iastate.edu/etd/13537/

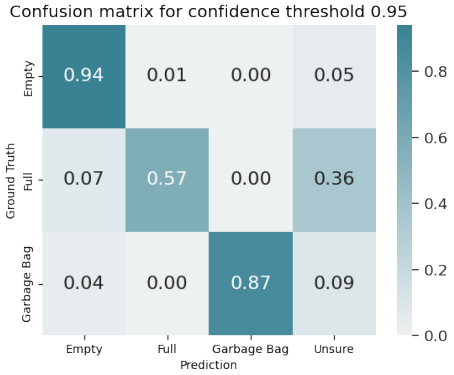
VII. APPENDIX



(a) The proportion of full trash cans falsely identified as empty is more than doubled when applied to the full data set, while the other categories perform similarly.



(b) Applying the same threshold as 6 does not yield nearly as strong of an improvement.



(c) Increasing the threshold even more only marginally improves the rate of misidentifcations but heavily increases the number of ”Unsure” classifications

Fig. 7: Confusion matrices for classification on full test set