Deep Learning in the Wild for Industrial Scale Plastic Waste Sorting

Al Shafayet Haque Silvy[1,2], Kanishka Tyagi[2], Isha K. Maun[2], Bin Chen[1] and Nalin Kumar[2]

[1]Purdue University Fort Wayne
[2]UHV Technologies Inc.

Abstract

Efficient sorting of plastic waste remains a critical bottleneck in recycling systems, with current approaches relying on manual labor or semi-automated solutions that contribute to large amounts of plastics ending up in landfills. Despite the rapid growth of the global plastic recycling market, projected to reach $120 billion by 2030, existing sorting technologies struggle to meet demands for accuracy and throughput [14]. While recent deep learning breakthroughs show promise in waste sorting, a complete industrial-scale pipeline has been overlooked. We propose a novel, low-cost deep learning system that addresses real-world challenges in plastic sorting such as varying material types, inconsistent lighting conditions, and contaminated surfaces. Our key contributions include: (1) a scalable deep learning architecture featuring two adaptive pipelines - one for data collection and another for classification, optimized for industrial deployment, (2) curation of the world's first comprehensive industrial dataset of 40,000 plastic samples, and (3) an interpretable approach leveraging Grad-CAM and t-SNE visualizations to tackle challenging cases like dark and distorted plastics. The proposed sorting system demonstrates commercial viability by processing 200 samples per hour in five plastic types common in municipal solid waste (MSW), with a potential cost of $30 per ton.

1 Introduction

The growing crisis of plastic waste mismanagement directly impacts human health, ecosystems, and climate, driven by the rising levels of microplastics and nanoplastics [20, 4, 23, 11]. As recycling expands globally and stricter regulations limit plastic exports to developing countries [5, 6, 18], these limitations, along with sustainability demands, present an excellent opportunity for a deep-learning solution tailored to this unique situation. Current Material Recovery Facilities (MRFs) face significant challenges in accurately sorting mixed plastics, black plastic films, and various polymer types, studies indicating misclassification rates as high as 35% for colored plastics and a rejection rate of bales sorted 5% due to contamination [12]. Existing sorting approaches broadly fall into three major categories: manual, semi-automated systems, and automated. While manual sorting offers high accuracy for complex materials, it suffers from low throughput and poses health risks to workers. Semi-automated sorting combines manual labor with machine processing improves efficiency compared to manual methods but still needs workers to handle contaminated materials missed by machines. While combining multiple sensing technologies (near-infrared, mid-infrared, X-ray fluorescence, and vision-based systems) could enable efficient large-scale sorting, such an integrated system remains unrealized due to prohibitive costs and inherent difficulties in handling contaminated and deformed materials. The technical challenges of existing sorting approaches present several critical limitations, which our proposed solution addresses, as detailed in Table 1.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Table 1: Challenges in material classification and our proposed solution

Challenges	Proposed solution
High-intra-class variation due to deformation and contamination	Robust dataset curation that addresses these cases, and experiments on evaluation of the model in these cases
Complex feature extraction requirements for visually similar materials	Deep learning architecture tuned to effectively process irregular and deformed inputs
Need for robust performance under varying industrial conditions	Integration of high frame camera with YOLO-based segmentation to improve object isolation for feeding the classifier
Particular difficulty with dark plastics that absorb light and provide limited visual features	A systematic approach to dark plastic classification through targeted data augmentation and feature extraction

Building on our past successful computer vision applications in metal sorting [8, 10, 9], we present EcoSorter (Figure 1). Our system leverages the fundamental relationship between plastic's visual and chemical properties to achieve accurate classification through specialized deep learning architectures. Using a low-cost camera setup, our machine learning model identifies subtle visual markers while maintaining system simplicity with just a conveyor belt and moderate computing power. The system remains scalable for additional sensors or computational resources. Our evaluation methodology employs Grad-CAM and t-SNE analysis to provide interpretability into the model's decision-making process, crucial for industrial applications where understanding failure modes is as important as overall accuracy.

2 Related Work

The Trashnet, TACO, and ZeroWaste are publicly available datasets, each offering unique advantages [24, 22, 1]. Trashnet provides 2,527 images across 6 categories, TACO covers a broader spectrum with 1,500 images in 60 categories, and ZeroWaste has 10,715 images divided into 7 classes. However, none of these datasets classify plastics by type, a critical factor in recycling, as even trace contamination (50 ppm of PVC) can render a batch of PET unusable [3]. In technical implementations, NIR technology has become prominent, often relying on statistical approaches or shallow ML models that analyze hyperspectral data [26, 13]. Although NIR is trustworthy, it struggles with multi-layer, dark, and contaminated plastics, and has high implementation costs. Computer vision approaches are emerging to address these issues, employing architectures like YOLO [19] and N-BEATS [16], offering promising results of 0.67% recall for black plastics and 91.7% mAP [21, 2]. Yet, these models focus on specific plastics under controlled conditions, overlooking the complexity of real-world MSW streams, where diverse plastics, contamination, and variability pose an unresolved challenge.

3 Dataset Creation and Curation

Our dataset includes 40,000 high-resolution images (640x465 pixels) representing five classes of plastics commonly found in municipal solid waste (MSW): Type 1 (PETE), Type 2 (HDPE), Type 3 (PVC), Type 5 (PP), and Type 7 (Other). We emphasize on challenging cases, including deformed, contaminated, and dark plastics. The data collection process involved MSW bales sourced from the Solid Waste Authority in Palm Beach, Florida. These bales were initially sorted by hand, then refined by our automated system, and finally validated by our team of plastic sorting researchers. Ambiguous cases were evaluated using a hyperspectral NIR camera at Idaho National Lab. Each object was imaged three times from different angles using a custom dual-lane conveyor system equipped with a calibrated Basler camera (acA2040-120um) featuring the Sony IMX252 CMOS sensor. The camera, combined with 1200 lumen lighting, delivers 120 frames per second at 3.2 MP resolution. A basic segmentation method ensured well-centered objects within the image frame. We curated three dataset variants: normal (centered on uniform background), contextual (with 15-pixel margin), and isolated (segmented object only). Type 3 plastics are scarce, accounting for only 4.4% of the dataset and 100 distinct objects. We addressed class imbalance through targeted augmentation and class weighting during training. For robust evaluation, we created a test set from a separate bale using single-shot capture per object. Automated outlier detection and manual verification filtered approximately 20%

of initial captures, ensuring dataset quality while maintaining real-world applicability. This approach eliminates data leakage, simulates real-world deployment, and prepares the data for classifier training pipeline.

4 Methodology

EcoSorter comprises of two integrated pipelines: data collection and classification as in Figure 1). Designed with modularity and scalability in mind, each component can be modified or enhanced independently, enabling continuous system improvement. The classification pipeline was developed after evaluating various deep learning architectures, including ResNet [7] variants (18, 34, 50) and EfficientNetV2M [15], optimizing for both accuracy and inference speed given our hardware constraints (single 1080Ti GPU in the EcoSorter system). To enhance processing efficiency for dark objects, we implemented a dual-model approach: a YOLOv5-based segmentation model for precise object isolation, followed by our classification model. This setup enables real-time processing while continuously expanding our dataset during operation. To ensure robust classification, we employed targeted data collection for underrepresented categories along with optimized data augmentation that preserves real-world characteristics. After experimentation, we selected ResNet-50 as our primary classifier, with customizations including two linear layers with batch normalization, LeakyReLU activation functions, and a 0.7 dropout rate. Selectively unfreezing the last two layers yielded 11,287,045 trainable parameters, enabling the capture of fine-grained material-specific features while maintaining computational efficiency. We conducted a systematic investigation into model robustness against object deformation, focusing particularly on Type 7 plastics (98% Arizona tea jugs in our dataset). This analysis addresses a critical challenge in industrial plastic sorting: the impact of physical deformation on classification accuracy, where traditional computer vision approaches often fail despite unchanged chemical composition. To quantify the relationship between deformation variability and model robustness, we implemented a progressive training regime using three datasets of increasing size: T_{50}, T_{75}, and T_{100}, representing training on 50%, 75%, and full data respectively. We analyzed predictions with confidence below 0.5 on our test set to create a hierarchical understanding of model behavior under increasing data exposure. The model's generalization limit on deformed plastics was quantified by measuring accuracy improvements as more diverse samples were added, with the confidence threshold empirically determined to capture significant deformation effects while minimizing false positives.

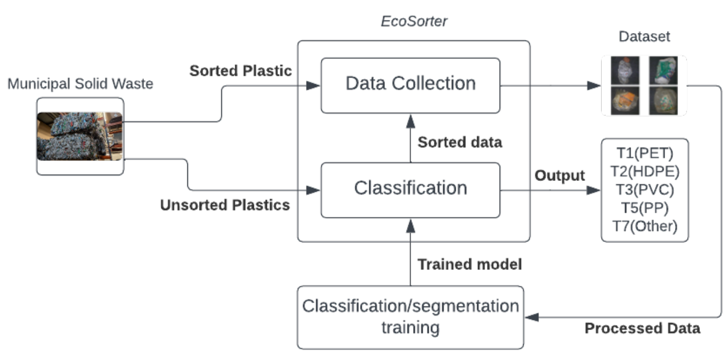


Figure 1: Complete Pipeline of the EcoSorter

5 Results and Discussion

To establish a reliable benchmark, we evaluated all models on a distinct test set of 6,613 images sourced from a completely separate bale. For model interpretability, we employed t-SNE [25] visualization for class separation analysis and Grad-CAM [17] to verify the model's attention to relevant features, critical for industrial deployment transparency. Table 1 shows the performance metrics across architectures. For industrial deployment analysis, we established key operational parameters: ResNet-50 achieved the highest accuracy of 87.19% with optimal dataset utilization at 75% (30,000 images) while maintaining real-time performance at 0.181 ms inference time. These findings were verified through 5-fold cross-validation ($p < 0.01$). The t-SNE analysis revealed ResNet-50's superior feature cohesion across deformation levels, particularly evident in our controlled study with Type 7 plastics (Arizona tea jugs). Using Grad-CAM visualizations, we observed that pristine objects generated sharp, localized activations on critical features, while deformed objects showed diffused activations predominantly focused on object edges. This analysis provided crucial insights into the model's feature recognition capabilities under varying deformation conditions. Our industrial optimization demonstrated significant cost-efficiency improvements through several key strategies. The adaptive processing pipeline, defined as $T_{total} = T_{base} + \sum_{i=1}^{N} p_i \cdot T_{specialist_i}$, enabled dynamic model loading and batch optimization for improved efficiency. Quantization and model pruning techniques were implemented to optimize throughput while maintaining accuracy. Our system's economic viability is demonstrated through its scalable performance across multiple GPUs. While the model achieves an inference time of 0.2061 ms, the overall system throughput is governed by the physical constraints of the sorting mechanism and data collection process. The return on investment, calculated as $ROI = \frac{(A_{ensemble} - A_{base}) \cdot V_{throughput} \cdot T_{operation}}{C_{hardware} + C_{maintenance}}$, helps establish deployment feasibility. The system's scaling efficiency followed $E(n) = \frac{S(n)}{n} = \frac{T_1}{n \cdot T_n}$, where $S(n)$ is speedup with n processors. Detailed error analysis revealed that while dark and glossy surfaces showed $3.1\times$ and $2.3\times$ higher error rates respectively, our proposed approach maintained robust performance across varying surface conditions through our adaptive processing pipeline.

Table 2: Performance metrics across architectures

Model	Accuracy	F1-Score	Parameters	Inference Time
ResNet-18	79.93%	0.80	4.8M	0.104 ms
ResNet-34	82.93%	0.83	9.5M	0.145 ms
ResNet-50	87.19%	0.87	11.2M	0.181 ms
EfficientNetV2-M	80.02%	0.66	0.66M	0.88 ms

6 Limitations

Our system's performance fundamentally depends on two critical aspects: data preprocessing quality and diversity of object representations. Our models may excel in controlled settings, but real-world deployment reveals challenges beyond algorithmic solutions. The class imbalance in our dataset, with Type 3 plastics at only 4.4% highlights a core challenge: bridging the gap between training data and real-world distribution. Deformation handling is a critical challenge, with model performance degrading non-linearly as deformation severity increases, especially for rare plastic types. This suggests prioritizing data collection and augmentation over model optimization for future improvements. Our work highlights the potential of continuous improvement in industrial settings, enabling dynamic model adaptation through simultaneous data collection. However, balancing adaptation with stability will be essential for consistent long-term performance.

7 Conclusion

This work presents significant advances in automated plastic sorting through a novel ML-driven approach for municipal solid waste management. Our comprehensive dataset of 40,000 industrial plastic samples, along with the dual-pipeline architecture combining data collection and classification, demonstrates the feasibility of deploying deep learning systems in challenging industrial environments. The systematic evaluation of architectures led to selecting ResNet-50, achieving 87.19% accuracy while maintaining real-time performance at 0.181 ms inference time. Our work bridges the gap between theoretical ML capabilities and industrial requirements through three key contributions: (1) a scalable system architecture that efficiently handles varying plastic types and lighting conditions, (2) an interpretable approach using Grad-CAM and t-SNE for robust classification, particularly demonstrated through our controlled study with Type 7 plastics, and (3) an industrially viable implementation of simultaneous data collection and classification through our adaptive processing pipeline. Looking forward, this work establishes a foundation for future improvements through synthetic data generation for underrepresented plastic types and continuous learning mechanisms for dynamic adaptation to new materials and deformation patterns.

Acknowledgments

This material is based upon work supported by U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) Bioenergy Technologies Office under the Award Number DE-EE0009266. The authors would like to acknowledge their collaborators at University of Illinois at Urbana-Champaign, West Palm Beach Solid Waste Authority, Idaho National Laboratories for their feedback and contribution.

References

[1] Dina Bashkirova, Ziliang Zhu, James Akl, Fadi M. Alladkani, Ping Hu, Vitaly Ablavsky, Berk Çalli, Sarah Adel Bargal, and Kate Saenko. Zerowaste dataset: Towards automated waste recycling. *CoRR*, abs/2106.02740, 2021. URL https://arxiv.org/abs/2106.02740.

[2] Janghee Choi, Byeongju Lim, and Youngjun Yoo. Advancing plastic waste classification and recycling efficiency: Integrating image sensors and deep learning algorithms. *Applied Sciences*, 13(18), 2023. ISSN 2076-3417. doi: 10.3390/app131810224. URL https://www.mdpi.com/2076-3417/13/18/10224.

[3] Alexander Dubanowitz. An analysis of municipal solid waste management in new york city, 2000. URL http://www.columbia.edu/cu/seas/earth/wtert/newwtert/Research/sofos/dubanowitz_thesis.pdf. Master's thesis, Columbia University, Accessed: 2024-10-31.

[4] Eduardo Grimaldo, Christian Karl, Anja Alvestad, Anna-Maria Persson, Stephan Kubowicz, Kjell Olafsen, Hanne Hatlebrekke, Grethe Lilleng, and Ilmar Brinkhof. Reducing plastic pollution caused by demersal fisheries. *Marine Pollution Bulletin*, 196:115634, 10 2023. doi: 10.1016/j.marpolbul.2023.115634.

[5] InTheBlack. China's waste ban leaves australia in a mess, 2023. URL https://intheblack.cpaaustralia.com.au/environment-and-sustainability/china-waste-ban-australia-mess. Accessed: 2024-10-31.

[6] InvestigateWest. Rich countries are illegally exporting plastic trash to poor countries, data suggests, 2023. URL https://www.investigatewest.org/news/rich-countries-are-illegally-exporting-plastic-trash-to-poor-countries-data-suggests-176918 Accessed: 2024-10-31.

[7] Shaoqing Ren Jian Sun Kaiming He, Xiangyu Zhang. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. doi: 10.48550/arXiv.1512.03385. URL https://doi.org/10.48550/arXiv.1512.03385.

[8] Nalin Kumar, Manuel Gerardo Garcia, Kanishka Tyagi, et al. Material sorting using a vision system, July 14 2020. US Patent 10,710,119.

[9] Nalin Kumar, Manuel Gerardo Garcia, Kanishka Tyagi, et al. Material handling using machine learning system, January 27 2022. US Patent App. 17/495,291.

[10] Nalin Kumar, Manuel Gerardo Garcia, Kanishka Tyagi, et al. Computer program product for classifying materials, May 7 2024. US Patent 11,975,365.

[11] J. Liang, F. Ji, A. L. B. Abdullah, W. Qin, T. Zhu, Y. J. Tay, Y. Li, and M. Han. Micro/nanoplastics impacts in cardiovascular systems across species. *The Science of the Total Environment*, 942:173770, 2024. doi: 10.1016/j.scitotenv.2024.173770. URL https://doi.org/10.1016/j.scitotenv.2024.173770.

[12] Cesar Lubongo and Paschalis Alexandridis. Assessment of performance and challenges in use of commercial automated sorting technology for plastic waste. *Recycling*, 7(2), 2022. ISSN 2313-4321. doi: 10.3390/recycling7020011. URL https://www.mdpi.com/2313-4321/7/2/11.

[13] Romans Maliks and Roberts Kadikis. Multispectral data classification with deep cnn for plastic bottle sorting. In *2021 6th International Conference on Mechanical Engineering and Robotics Research (ICMERR)*, pages 58–65, 2021. doi: 10.1109/ICMERR54363.2021.9680850.

[14] MarketsandMarkets. Recycled plastic market by source (bottles, films, fibers, foams), type (pet, pe, pp, pvc, ps), end-use industry (packaging, textile, building construction, automotive, electrical electronics) region - global forecast to 2026, 2024. URL `https://www.marketsandmarkets.com/Market-Reports/recycled-plastic-market-115486722.html`. Accessed: 2024-10-31.

[15] Quoc V. Le Mingxing Tan. Efficientnetv2: Smaller models and faster training. *arXiv preprint arXiv:2104.00298*, 2021. doi: 10.48550/arXiv.2104.00298. URL `https://doi.org/10.48550/arXiv.2104.00298`.

[16] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

[17] Abhishek Das Ramakrishna Vedantam Devi Parikh Dhruv Batra Ramprasaath R. Selvaraju, Michael Cogswell. Grad-cam: Visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1610.02391*, 2019. doi: arXiv:1610.02391. Related DOI: https://doi.org/10.48550/arXiv.1610.02391.

[18] Recycling Magazine. Us plastic waste exports in violation of new un rules, 2021. URL `https://www.recycling-magazine.com/2021/03/11/us-plastic-waste-exports-in-violation-of-new-un-rules/`. Accessed: 2024-10-31.

[19] J Redmon. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[20] M. Rezaei, M. J. P. M. Riksen, E. Sirjani, A. Sameni, and V. Geissen. Wind erosion as a driver for transport of light density microplastics. *The Science of the Total Environment*, 669: 273–281, 2019. doi: 10.1016/j.scitotenv.2019.02.382. URL `https://doi.org/10.1016/j.scitotenv.2019.02.382`.

[21] Attilio Sbrana, Aline Gabriel de Almeida, André Mobaier de Oliveira, Henrique Seschin Neto, João Paulo Cesar Rimes, and Maria Cristina Belli. Plastic classification with nir hyperspectral images and deep learning. *IEEE Sensors Letters*, 7(1):1–4, 2023. doi: 10.1109/LSENS.2023.3234401.

[22] Pedro F. P. A. Simões. Taco: Trash annotations in context for litter detection. *arXiv preprint arXiv:2003.06975*, 2020. URL `https://doi.org/10.48550/arXiv.2003.06975`. arXiv:2003.06975 [cs.CV].

[23] T. Syversen, G. Lilleng, J. Vollstad, B. J. Hanssen, and S. A. Sønvisen. Oceanic plastic pollution caused by danish seine fishing in norway. *Marine Pollution Bulletin*, 179:113711, 2022. doi: 10.1016/j.marpolbul.2022.113711. URL `https://doi.org/10.1016/j.marpolbul.2022.113711`.

[24] Gary Thung and Mindy Yang. Trashnet: A dataset for garbage classification. `https://github.com/garythung/trashnet`, 2017. Accessed: 2024-10-31.

[25] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL `http://jmlr.org/papers/v9/vandermaaten08a.html`.

[26] Y. Zheng, J. Bai, J. Xu, X. Li, and Y. Zhang. A discrimination model in waste plastics sorting using nir hyperspectral imaging system. *Waste Management*, 72:87–98, Feb 2018. doi: 10.1016/j.wasman.2017.10.015. Epub 2017 Nov 10.

8 Appendix / supplemental material

8.1 Dataset Collection and Overview

We bought numerous bales of municipal waste from Florida, each bale 2 consisting of different types of unsorted materials and weighing around 800 kg. We started the process by first hand-sorting the bales extensively and putting the plastics into Gaylord 3 boxes according to their types and named them B id, Type number where B id represents the Bin id. For protection, we wore face masks and hand gloves. Each bale required around two weeks to get sorted as we put in around 3 hours daily. Most of the time the plastics were sorted by their type engraved under the bottom of the material, in certain instances the plastics were too crushed or damaged, so we sent them to an NIR facility (Idaho National Lab) to recover the types. Each bale contributed to a different number of plastics for each type. Based on the external appearance of the bale, it was expected that some of them were Type 1 heavy, and some were Type 2 heavy. Since Type 1, Type 2, and Type 5 plastics made up the majority of the dataset, we attempted to open the bales with greater variability.

After completing the initial data sorting phase, we deployed the EcoSorter with a well-trained model to reduce manual labor. In its first trial, the machine achieved a notable sorting accuracy for Types 1, 2, and 5 plastics, reflecting the higher representation of these classes in the training dataset. However, manual intervention remained necessary to maintain overall accuracy across all types. Through iterative trials and refinements, we reached a consistent 85% accuracy in overall sorting, significantly reducing processing time from 15 hours per bale to just 4 hours.

The data collection process also started as semi-automated, utilizing the EcoSorter (Figure 6) for high-speed image acquisition. While automatic, human oversight ensured images are well-centered and visible. The sorter uses a dual conveyor system, with two Basler cameras and bright lights per lane. Images are captured automatically when objects are detected on the belt, except for dark plastics, which required manual adjustment of segmentation settings. Approximately 1,000 images were collected every 5 hours and 2 hours took to manually remove outliers. A total of 15,294 images from 5 bales of plastic were captured. Utilizing these data the YOLO segmentation model was integrated into the EcoSorter, enabling it to capture images in real-time as they were being sorted. Much like the sorting process, we conducted iterative trials to refine the model, enhancing its accuracy and performance over time.

Figure 2: Unsorted Bales of Municipal Solid Waste

Figure 3: Gaylord boxes with Plastics sorted into specific types

8.1.1 Dataset Overview

The original dataset consists of 40,000 images (Training/validation 33,387 + Testing 6,613), each at a resolution of 640x465 pixels. This dataset was curated to reflect real-world conditions by capturing multiple images of the same materials from different angles, simulating how they would appear on a conveyor belt. Type 3 plastics had the lowest representation, with only 35 unique objects found across all five bales. To compensate for this, we augmented the dataset by capturing these objects multiple times, altering their shapes to increase the number of samples and enhance model training. While the dataset included other plastic types, such as Type 4 and Type 6, we chose not to incorporate them into the current project due to their minimal representation. The small sample sizes for these types would not provide meaningful insights or robust model training for this specific task. Figure 8 represents the overview of the whole dataset.

We curated three dataset variations from the original images to evaluate model accuracy under conditions similar to the machine's view of the objects on the conveyor belt. The contexual dataset consists of objects cropped to their dimensions with an additional 15-pixel margin on all sides. This was achieved using a standard segmentation technique, where we determined the object's center, width, and height, then cropped the image with a 15-pixel margin. This approach worked well for most plastics, but dark plastics posed a challenge due to the dark conveyor background, requiring manual threshold adjustments for segmentation. The normal dataset was constructed by centering the cropped objects on a uniform, dark square background. Finally, the isolated dataset contains the objects alone, with the conveyor belt entirely removed. The test set for model evaluation was constructed using a completely distinct bale. We curated objects from a new bale, capturing a single image per object to ensure a more robust assessment of the model's accuracy.

Notably, this type of comprehensive dataset, derived from municipal solid waste, represents an unprecedented effort and holds the potential as a valuable resource for training different convolutional neural network models. Since our bales are predominantly sourced from Florida, the dataset is likely to reflect the composition of Florida's materials. To enhance the dataset's diversity and ensure broader applicability, it would be necessary to gather data from different regions. If segmentation algorithms within the machine were optimized for speed, this could allow for a faster conveyor belt, enabling the collection of a higher volume of images per second.

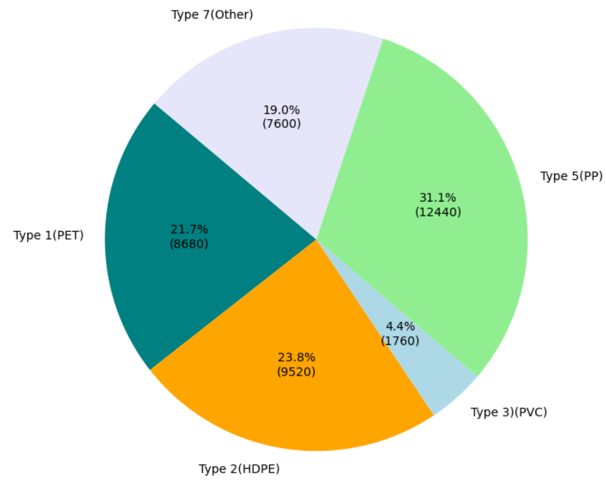


Figure 4: Dataset distribution for 40,000 images of 5 types of plastics

(a) (b) (c)

Figure 5: **(a)** Sample Image from Contexual Dataset **(b)** Sample Image from Normal Dataset **(c)** Sample Image from Isolated Dataset

8.2 Model Training & Results

We trained a total of four models across the three datasets we curated. The optimal dataset which did really well on the test data was the contexual dataset, so all the results presented here are based on that dataset. This choice was made after thoroughly training on all datasets and rigorously evaluating their accuracy. The following subsections will provide an overview of the architectures employed, detailing both their design principles and the rationale behind their selection. Additionally, we will explore the performance characteristics of each model in the context of our application, discussing their strengths and limitations with respect to processing speed, accuracy, and suitability for real-time sorting tasks. This will include an analysis of how these models scale with increasing data complexity and how they can be optimized for deployment in resource-constrained environments.

We employed our 33k dataset to train the model, partitioning it into a 70% training set and a 30% validation set randomly. For every 15 epochs, we plotted the training and validation accuracy and loss curves, offering a visual guide to the model's learning trajectory and helping us understand if the model was overfitting or underfitting. We also applied class weighting, with a particular emphasis on Type 3 plastics, as this class is underrepresented in our dataset. The following hyperparameters guide our training processes for most of the training with different combinations till we get the best model:

$$\begin{aligned}
&\text{Learning Rate} = 0.001,\\
&\text{Batch Size} = 64,\\
&\text{Epochs} = 2000,\\
&\text{Optimizer} = \text{SGD / Adam},\\
&\text{Patience} = 35
\end{aligned}$$

(Note: The batch size was reduced when training EfficientNetv2-m due to the 480x480 pixel input size.)

For training ResNets our data augmentation includes resizing images to 224x224 pixels, random affine transformations with 1.5 shear and scale, horizontal and vertical flips, color jittering (brightness, contrast, saturation, hue adjustments), and random rotations up to 15 degrees, followed by conversion to a tensor and normalization. After testing various parameters, we found this configuration to be the most effective. For training EfficientNetv2-m, we maintained all parameters, only adjusting the input size to 480x480.

8.2.1 ResNet-18

ResNet-18 consists of 18 layers, beginning with an initial convolutional layer, ending with a fully connected layer, and containing four intermediate stages in between. To adapt ResNet-18 to our task, where we classify across 5 categories, we modify the final fully connected layer to accommodate these classes. Initially, by freezing all layers except for the fully connected layer, we reduce the trainable parameter count to a mere 132,613. After experimentation, we determined that a more effective strategy is to freeze only the last two blocks of the fourth layer. This adjustment significantly increases the model's capacity, raising the number of trainable parameters to 4,853,253, allowing for more fine-tuned feature extraction without overwhelming our computational resources. Below is the structure of our final Fully Connected layer for ResNet-18:

$$\begin{aligned}
\text{model.fc} = \mathrm{nn.Sequential}\Big(&\\
&\mathrm{nn.Linear}(512, 256),\\
&\mathrm{nn.BatchNorm1d}(256),\\
&\mathrm{nn.ReLU}(\text{inplace=True}),\\
&\mathrm{nn.Linear}(256, 5)\Big)
\end{aligned}$$

The figures below show the output of ResNet-18.

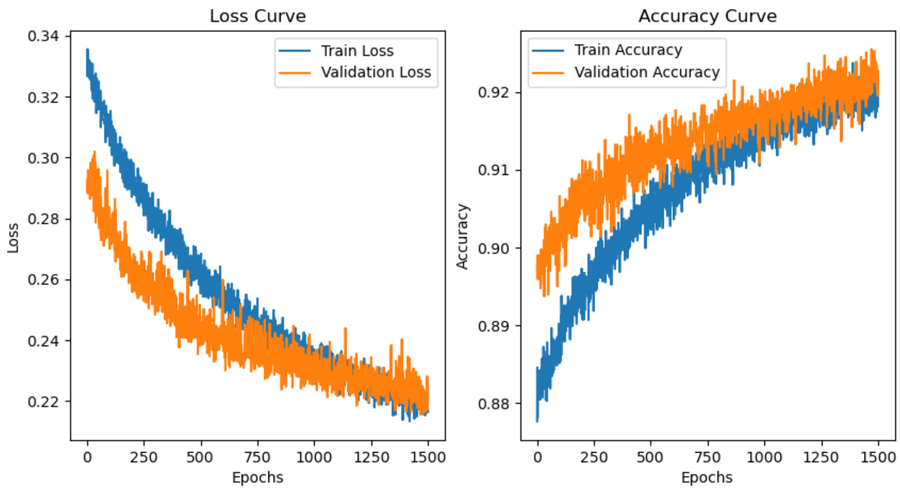
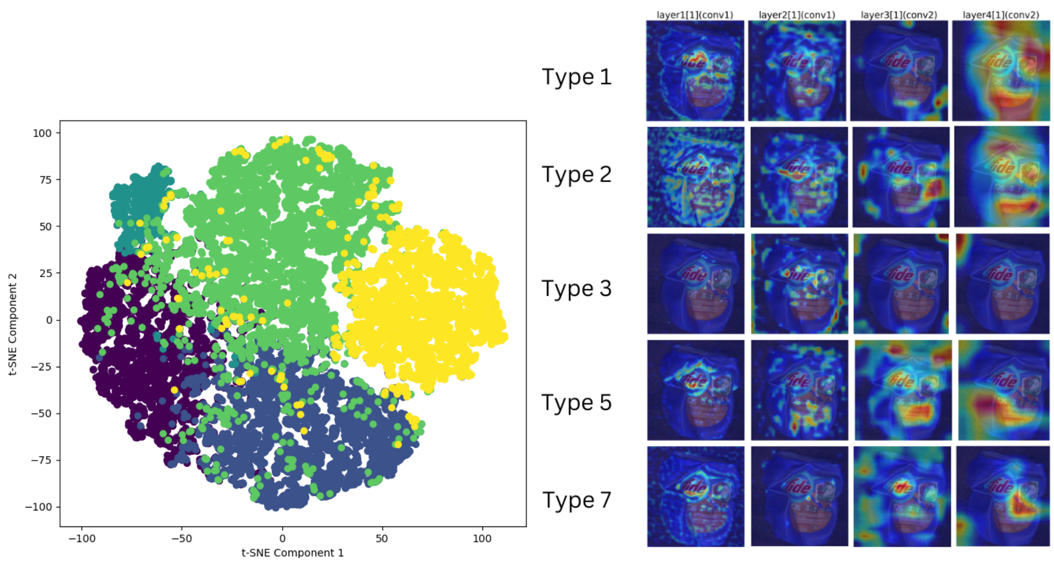


Figure 6: Loss and Accuracy curves for ResNet-18 after 1500 epochs



(a) (b)

Figure 7: **(a)** t-SNE features of ResNet-18 on training set, as indicated by the color: Dark Purple(Type 1), Blue(Type 2), Green(Type 3), Light Green(Type 5), Yellow(Type 7), **(b)** Grad-CAM heatmap of Type 2 plastic on ResNet-18. Columns represent layers 'layer1[1](conv1)', 'layer2[1](conv1)', 'layer3[1](conv2)', and 'layer4[1](conv2)', with rows showing features extracted from each class.

8.2.2 ResNet-34

ResNet-34 offers a well-balanced trade-off between depth and computational efficiency, making it an effective medium-capacity classifier. In this configuration, we freeze all layers except for the final two layers in the last stage, reducing the number of trainable parameters to 9,573,893. We introduce an additional 4,720,640 trainable parameters more than ResNet-18. Given the model's depth and complexity, it can easily overfit our dataset, so we incorporate dropout rate = 0.3 in the final layer. This approach slows down the training process, but serves as an essential regularization technique, reducing the risk of overfitting. Below is the structure of the final Fully Connected Layer for ResNet-34:

$$
\begin{aligned}
\text{model.fc} = \mathrm{nn.Sequential}\Big(\\
&\mathrm{nn.Linear}(512, 256),\\
&\mathrm{nn.BatchNorm1d}(256),\\
&\mathrm{nn.ReLU}(\text{inplace=True}),\\
&\mathrm{nn.Dropout}(0.3),\\
&\mathrm{nn.Linear}(256, 5)\Big)
\end{aligned}
$$

The figures below show the output of ResNet-34:

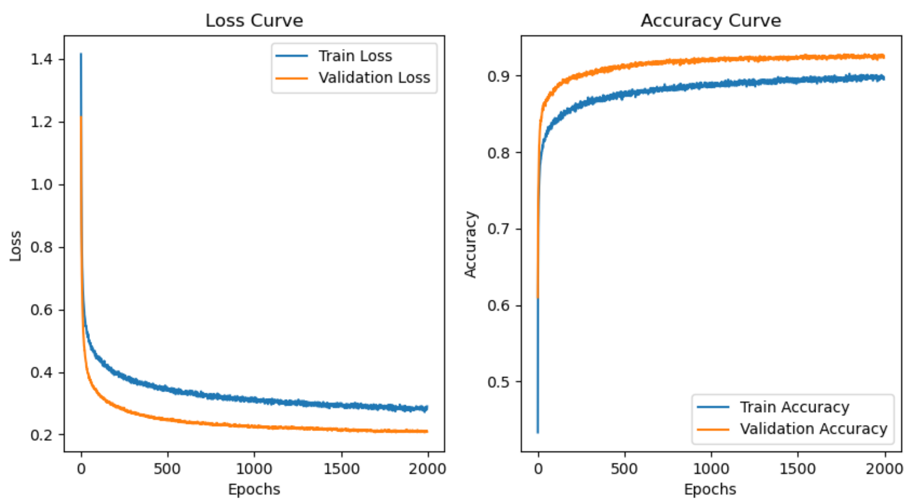
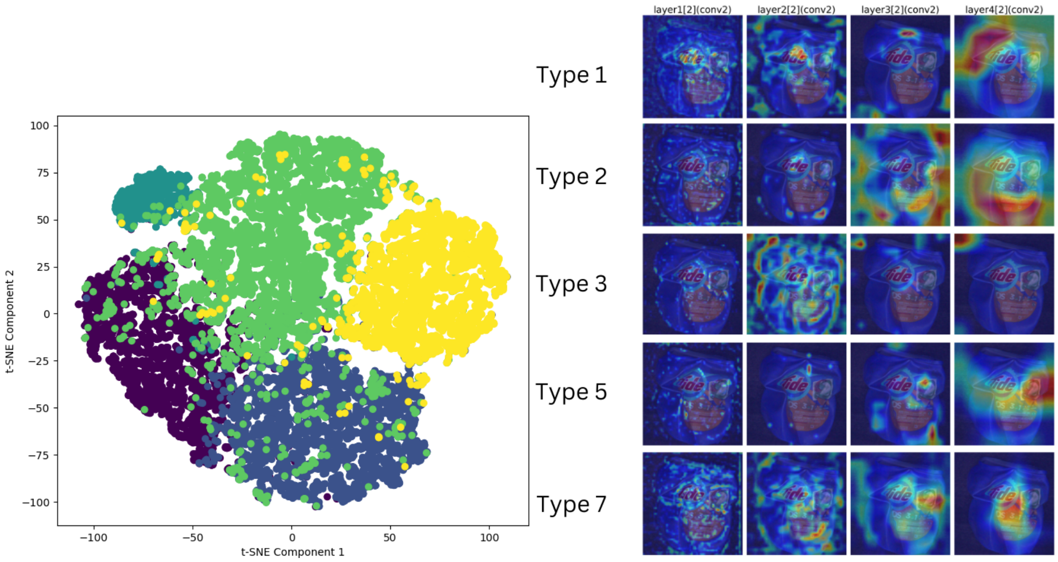


Figure 8: Loss and Accuracy curves for ResNet-34 after 1995 epochs



(a) (b)

Figure 9: **(a)** t-SNE features of ResNet-34 on training set, as indicated by the color: Dark Purple(Type 1), Blue(Type 2), Green(Type 3), Light Green(Type 5), Yellow(Type 7), **(b)** Grad-CAM heatmap of Type 2 plastic on ResNet-34. Columns represent layers 'layer1[2](conv2)', 'layer2[2](conv2)', 'layer3[2](conv2)', and 'layer4[2](conv2)', with rows showing features extracted from each class.

8.2.3 ResNet-50

With 1 input layer and 1 output layer sandwiching 4 stages of residual blocks which contain 48 layers ResNet-50 is designed in bottleneck architecture. The major difference between ResNet-50 with the prior ones is the architectural depth it contains which makes it complex as a model allowing it to learn more details of an image. After freezing all layers except the last two blocks of the last layer we have 11,287,045 numbers of parameters which is 1,713,152 more parameters than ResNet-34. We experimented with various configurations to fine-tune the model, as it plays a pivotal role in the EcoSorter's performance on the testing data. Our focus was on refining the architecture by introducing two additional layers in the final Fully Connected section, each accompanied by distinct dropout rates. We explored a range of activation functions, alternating between ReLU and LeakyReLU, combined with different dropout values to improve regularization and avoid overfitting. After multiple iterations, we converged on a final architecture that provided the best balance between generalization and performance. Below is the structure of the final Fully Connected Layer for ResNet-34:

$$\begin{aligned}\text{model.fc} = \text{nn.Sequential}\Bigg(\\ &\text{nn.Linear}(2048, 512),\\ &\text{nn.BatchNorm1d}(512),\\ &\text{nn.LeakyReLU}(\text{negative\_slope} = 0.02, \text{inplace=True}),\\ &\text{nn.Dropout}(0.7),\\ &\text{nn.Linear}(512, 5)\Bigg)\end{aligned}$$

The figures below show the output of ResNet-50:

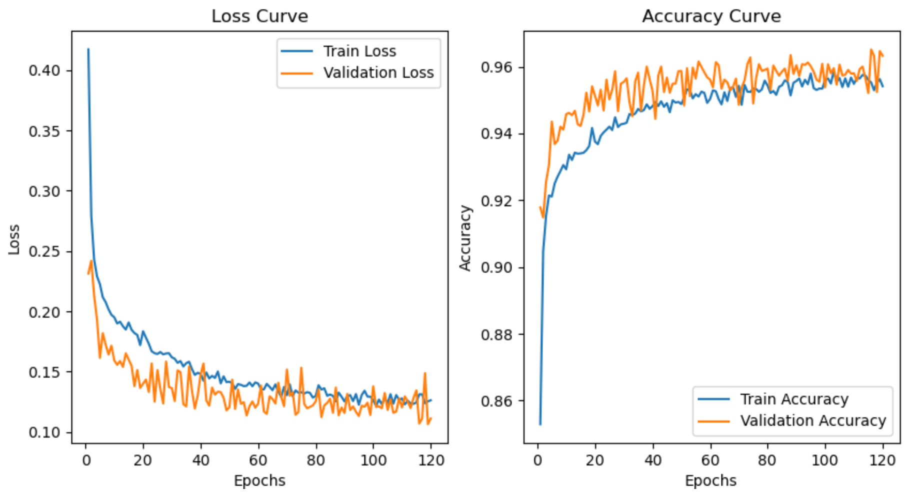
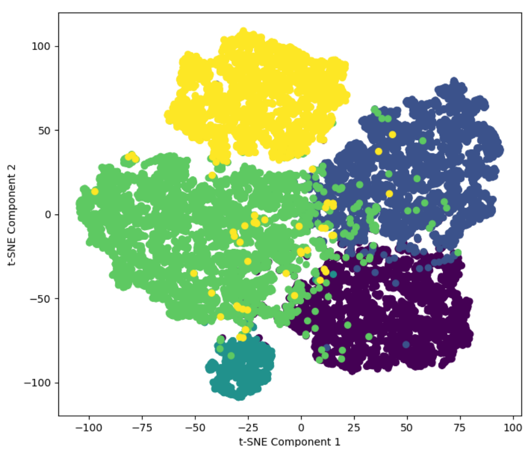
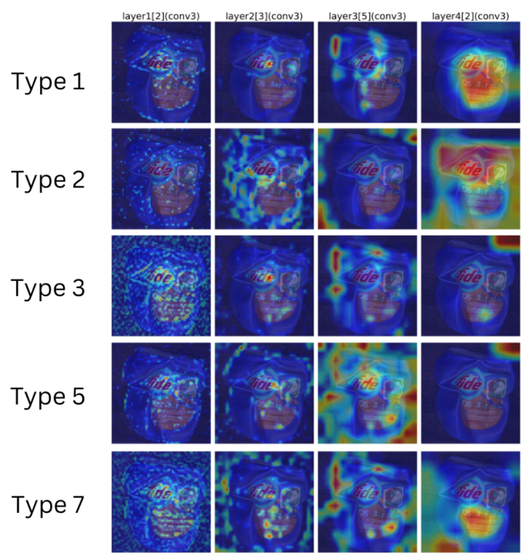


Figure 10: Loss and Accuracy curves for ResNet-50 after 120 epochs



(a)



(b)

Figure 11: **(a)** t-SNE features of ResNet-50 on training set, as indicated by the color: Dark Purple(Type 1), Blue(Type 2), Green(Type 3), Light Green(Type 5), Yellow(Type 7), **(b)** Grad-CAM heatmap of Type 2 plastic on ResNet-50. Columns represent layers 'layer1[2](conv3)', 'layer2[3](conv3)', 'layer3[5](conv3)', and 'layer4[2](conv3)', with rows showing features extracted from each class.

8.2.4 EfficientNetv2-m

EfficientNetV2-M, with 74 layers, balances performance, and computational demand, containing 54 million parameters and 24.3 billion floating-point operations (FLOPs). EfficientNetV2-M provides an ideal middle ground as its' combination of moderate depth, parameter count, and inference speed positions it as a strong contender against our existing classifiers, making it well-suited for rigorous evaluation and potential deployment in our system. Training models with pre-trained weights in PyTorch offers both flexibility and efficiency. In our case, we leveraged EfficientNetV2-M with transfer learning, freezing all layers except for last block leaving us only 0.66 million trainable parameters. We retained the final layer unchanged due to limited computational resources.

The figures below show the output of EfficientNetv2-m:

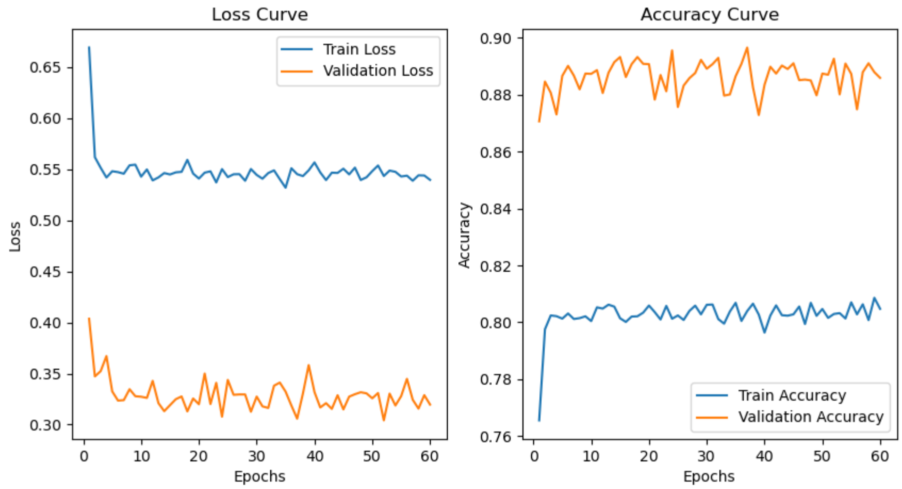
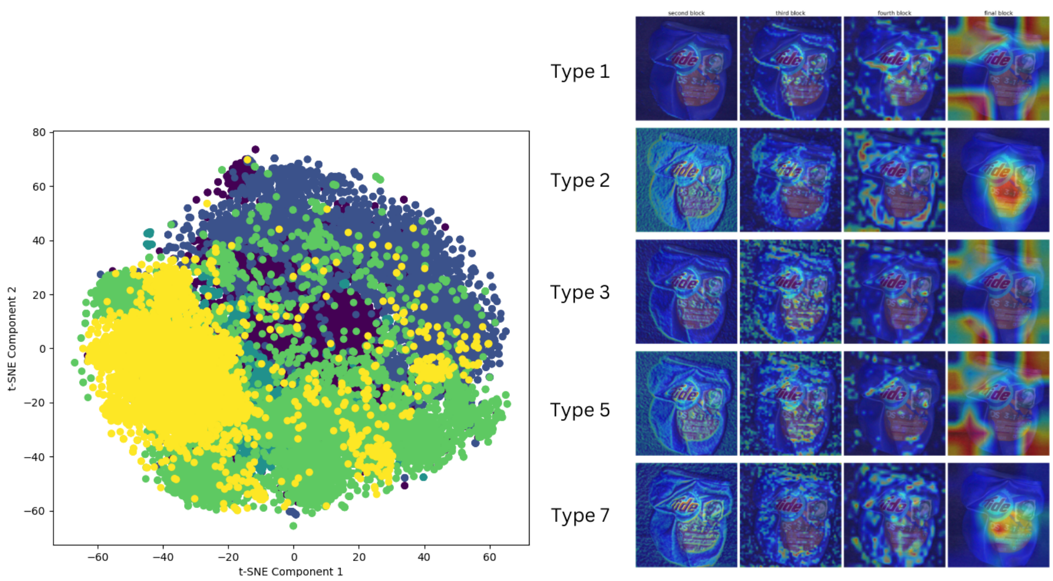


Figure 12: Loss and Accuracy curves for EfficientNetv2-m after 60 epochs



(a) (b)

Figure 13: **(a)** t-SNE features of EfficientNetv2-m on training set, as indicated by the color: Dark Purple(Type 1), Blue(Type 2), Green(Type 3), Light Green(Type 5), Yellow(Type 7), **(b)** Grad-CAM heatmap of Type 2 plastic on EfficientNetv2-m. Columns represent blocks 'second block', 'third block', 'fourth block', and 'final block', with rows showing features extracted from each class.

8.2.5 YOLO-v5

In constructing a detection model, we transitioned from a segmentation-centric approach to a bounding box-based one using YOLO-v5m. The choice of the medium-sized model, with 25.1 million parameters, balances computational efficiency with the precision required for our specific task. This decision stems from the balance of dataset size, the complexity of the problem, and desired performance outcomes. The core shift from classifiers to YOLO lies in the dataset preparation. Each image now carries labels in the format:

$$classID\ center_x\ center_y\ width\ height$$

With coordinates normalized according to the image dimensions, allowing the network to generalize better across different input sizes. We created such a dataset from our basic segmentation, we get the bounding boxes of each object from the image and get the labels of those bounding boxes. After training the YOLO we segment the rest of our data and calculate the time it takes. illustrates how well YOLO segments dark plastics, removing the need for human threshold modifications. To fully leverage this model, we extract the bounding boxes of the objects and pad them by approximately 25 pixels to ensure no part of the object is cropped. We then crop the padded region and pass it to the classifier. The model processes each image in approximately 0.0251 seconds.

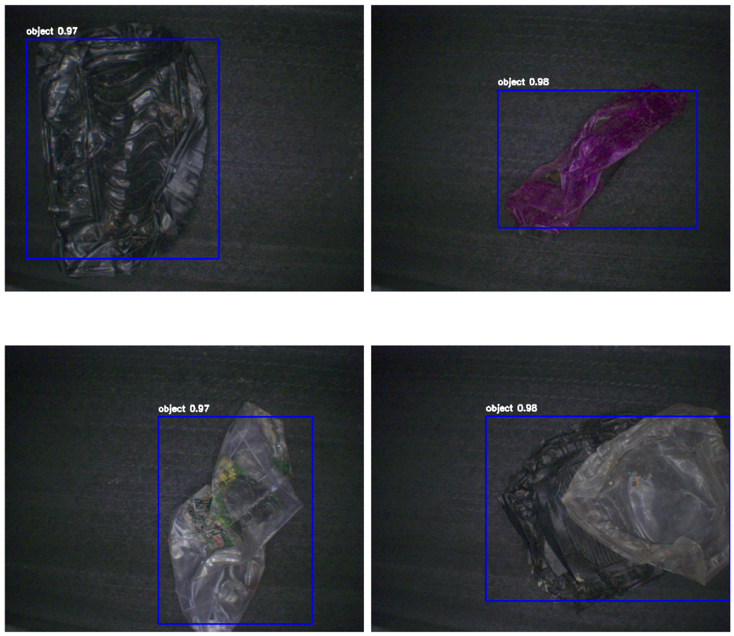


Figure 14: Samples of YOLO segmentation on unseen dark plastics

8.3 EcoSorter

This section explores the operational framework of the EcoSorter, the system is designed for object sorting and data collection simultaneously. The core architecture of the system can be segmented into three key components: the CPU head, the mechanical conveyor, and the air nozzles. At the heart of the system, the CPU head orchestrates all mechanical operations, functioning as the brain of the sorter. This module houses both the segmentation and the classification models, ensuring precise control of signals that govern the downstream processes. Built around a mid-range CPU AMD Ryzen 5 with a GPU 1080-Ti, the system has enough computational power for real-time processing. An integrated automated cooling system ensures that the temperature remains within operational bounds, maintaining system stability. Additionally, the CPU head is equipped with two Basler cameras (acA2040-120um) featuring the Sony IMX252 CMOS sensorc, 1200 lumen lighting arranged to provide optimal illumination during object capturing. The mechanical conveyor serves as the backbone of the system's material handling, operating at a speed of 120 ft/min. This belt moves objects from the input to the output stage, with its speed modifiable via the CPU head to adapt to different operational needs. The velocity of the conveyor is a critical factor, as it directly influences the timing of object classification and subsequent sorting. Finally, the air nozzle system completes the process by physically blowing the objects to the right bins. Positioned at the end of the CPU head, the EcoSorter utilizes ten air nozzles, each firing in precise intervals to direct classified objects into their corresponding bins. The CPU head calculates the exact distance and timing for the nozzles to activate, ensuring that objects are sorted accurately based on their classification. Though the current iteration is compact, its modular design allows for scalability. The system can be expanded with a larger conveyor and more powerful air nozzles for high-throughput sorting, or, conversely, reduced in size for smaller-scale operations. This flexibility, coupled with the robust computational and mechanical foundation, positions the EcoSorter as a versatile solution for industrial-grade object classification.

Figure 15: EcoSorter

8.4 Result analysis

8.4.1 Computational Resource Analysis

Our computational resource analysis revealed significant variations across architectures. ResNet-18 required minimal computational resources (1.8B FLOPs), ResNet-34 moderate (3.6B FLOPs), and ResNet-50 showed balanced efficiency with 4.1B FLOPs, while EfficientNetV2-M required substantially higher computations (24.3B FLOPs). With a batch size of 32, ResNet-50 achieved our target inference time while maintaining the reported 87.19% accuracy. In comparison, while ResNet-18 and ResNet-34 required fewer computational resources, they showed lower accuracy (79.93% and 82.93% respectively). EfficientNetV2-M, despite its theoretical efficiency, required more computational resources while achieving lower accuracy (80.02%) in our specific use case. Memory analysis during training showed peak usage following $M_{peak} = M_{base} + B \times (F_{maps} + G_{maps})$, where M_{base} is the model's base memory footprint, B is batch size, F_{maps} and G_{maps} are feature and gradient maps respectively.

8.4.2 Deformation Analysis

We focused on Type 7 plastics, specifically Arizona tea jugs (98% of our Type 7 samples), providing a controlled setting to study model behavior with deformed objects. We conducted a systematic analysis by training three progressive models using 50%, 75%, and 100% of our dataset containing both pristine and deformed samples.

For each model, we analyzed samples with classification confidence below 0.5 using Grad-CAM visualizations. The analysis revealed distinct patterns: pristine objects generated sharp, localized activations on critical features, while deformed objects showed diffused or misaligned activations, predominantly focused on object edges. This visualization approach provided insights into how deformation affects the model's feature recognition capabilities.

Our hypothesis that increased dataset variability through inclusion of more deformed samples would improve model generalization was tested through this progressive training regime. The model trained on a more diverse dataset demonstrated better handling of deformed samples, suggesting that exposure to varied deformation patterns during training enhances the model's robustness for real-world applications.

(a) (b)

Figure 16: **(a)** Pristine Arizona Tea Jug (Type 7) **(b)** Deformed Arizona Tea Jug(Type 7)

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims and contribution are reflected in both Abstract and Introduction.

   Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in the Limitation section.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification:Section 3 on Dataset Creation and Curation and section 4 on Methodology gives the required details on the theoretical assumptions that have been made in the paper.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.

- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The dataset will be publicly available as soon as the project is over and the details about training models, and benchmarking is given in the appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: The data or the code isn't provided open-access as they are property of UHV Technologies Inc, but sufficient instructions are given in both the Appendix and Methodology to reproduce.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.

- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The core methods are defined in Methodology but the specific details on data split, hyperparameters and everything related to training the models are presented in Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: The result section goes in depth to discuss the statistical significance of the experiments that have been discussed in the paper.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.

- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The details on compute resources to reproduce the experiments are given in Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in this paper conform with the NeurIPS Code if Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss potential positive societal and economical impacts. Our work doesn't have any negative societal impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper doesn't pose any kinds of risk in this manner as all the techniques are already available.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All assets which are utilized in this paper is credited with references.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

Justification: This paper isn't releasing any new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: Our research did not involve crowd sourcing or experimentation with human.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper did not require any crowdsourcing or research with human.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.