INDUSTRIAL-GRADE AI SYSTEM FOR AUTOMATED PLASTIC SORTER

by

Al Shafayet Haque Silvy

A Thesis

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Master of Science in Engineering



Department of Electrical and Computer Engineering

Fort Wayne, Indiana

December 2024

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Bin Chen, Chair

Department of Electrical and Computer Engineering

Dr. Todor Cooklev

Department of Electrical and Computer Engineering

Dr. Chao Chen

Department of Electrical and Computer Engineering

Approved by:

Dr. Hosni Abu-Mulaweh

ACKNOWLEDGMENTS

I want to express my deepest gratitude to God and my parents, especially my mom and dad, for their unwavering support and patience, even when they don't fully understand what I’m saying. I am profoundly thankful to Dr. Bin Chen for his invaluable ideas and guidance throughout my master's journey. I also extend my heartfelt thanks to Dr. Guoping Wang, Dr. Todor Cooklev, and Dr. Chao Chen for their support. This experiment would not have been possible without Dr. Kanishka Tyagi's contributions and the funding provided by UHV and the Department of Energy.

TABLE OF CONTENTS

LIST OF TABLES 5
LIST OF FIGURES 6
ABSTRACT 8
1. INTRODUCTION 9
2. RELATED WORK 13
2.1 Existing Datasets 13
2.2 Existing Work 15
3. DATA 19
3.1 Dataset Collection 19
3.2 Dataset Overview 23
4. METHODOLOGY 27
4.1 CNN Models 28
4.1.1 ResNet 29
4.1.2 EfficientNet-V2 30
4.1.3 YOLO-V5 31
5. EXPERIMENTS 32
5.1 Training Classifiers 32
5.1.1 ResNet-18 32
5.1.2 ResNet-34 34
5.1.3 ResNet-50 36
5.1.4 EfficientNet-v2m 39
5.2 Training YOLO-v5 for Segmentation 40
5.3 Model Accuracy Evaluation on Deformed Objects 42
6. RESULTS 45
6.1 Model Analysis 46
6.2 Discussions 72
7. CONCLUSIONS 74
APPENDIX A. OPERATIONAL FRAMEWORK OF THE UHV BLUE SORTER 75
REFERENCES 77

LIST OF TABLES

Table 1**.** *Different datasets for plastic waste sorting purposes.*..... 15

Table 2. *Existing approaches to solving plastic waste sorting*..... 18

Table 3. *Accuracy, F1 score, and Recall for each trained classifier on plastic waste data*..... 47

LIST OF FIGURES

Figure 1. *Left- Images from Trashnet dataset, Right- Images from TACO dataset.* 14

Figure 2**.** *Images from ZeroWaste dataset.* 15

Figure 3**.** *Unsorted Bales of Municipal Solid Waste* 20

Figure 4**.** *Gaylord boxes with Plastics sorted into specific types* 21

Figure 5**.** *UHV Blue Sorter data collection system diagram* 21

Figure 6. *UHV Blue Sorter CPU head* 22

Figure 7. *Plastic material under the UHV Blue Sorter camera* 22

Figure 8. *Dataset distribution for 33,387 images of 5 types of plastics* 23

Figure 9**.** *Sample images of each plastic type* 24

Figure 10**.** *Sample Images from dataset-1 (D1)* 25

Figure 11**.** *Sample Images from dataset-2 (D2)* 26

Figure 12. *Sample Images from dataset-3 (D3)* 26

Figure 13. *Pipeline for Industrial Sorting* 28

Figure 14. *Loss/ Accuracy curve for ResNet-18* 34

Figure 15. *Loss/ Accuracy curve for ResNet-34* 36

Figure 16. *Loss/ Accuracy curve for ResNet-50 (R1)* 38

Figure 17. *Loss/ Accuracy curve for ResNet-50 (R2)* 38

Figure 18. *Loss/ Accuracy curve for EfficientNet-v2m* 40

Figure 19. *Samples of YOLO segmentation on unseen data* 41

Figure 20. *Samples of Ideal Type 7 Plastics* 43

Figure 21. *Samples of Crushed/deformed Type 7 Plastics* 44

Figure 22. *Test Dataset distribution for 3264 images of 5 types of plastics* 46

Figure 23. *Confusion matrix for ResNet-18* 49

Figure 24. *Confusion matrix for ResNet-34* 49

Figure 25**.** *Confusion matrix for ResNet-50(R1)* 50

Figure 26. *Confusion matrix for ResNet-50(R2)* 50

Figure 27. *Confusion matrix for EfficientNetv2-m* 51

Figure 28**.** *t-SNE features against ResNet-18 on Training Dataset* 54

Figure 29. *t-SNE features against ResNet-34 on Training Dataset* 54

Figure 30. *t-SNE features against ResNet-50 on Training Dataset* 55

Figure 31. *t-SNE features against EfficientNetv2-m on Training Dataset* 55

Figure 32**.** *Grad-CAM of Type 2 object against ResNet-18, rows represent type specific gradients and column represents layer based heatmap on based on the specific type* 58

Figure 33**.** *Grad-CAM of Type 2 object against ResNet-34, rows represent type specific gradients and column represents layer based heatmap on based on the specific type* 59

Figure 34. *Grad-CAM of Type 2 object against ResNet-50, rows represent type specific gradients and column represents layer based heatmap on based on the specific type* 60

Figure 35.*Grad-CAM of Type 2 object against EfficientNetv2-m, rows represent type specific gradients and column represents layer based heatmap on based on the specific type* 61

Figure 36**.** *Scatter plot of confidence for ResNet50 trained on 50% Type 7 Training data* 67

Figure 37. *Histogram of confidence for ResNet50 trained on 50% Type 7 Training data* 67

Figure 38. *Scatter plot for ResNet50 trained on 75% Type 7 Training data* 68

Figure 39. *Histogram of confidence for ResNet50 trained on 75% Type 7 Training data* 68

Figure 40**.** *Scatter plot for ResNet50 trained on 100% Type 7 Training data* 69

Figure 41. *Histogram of confidence for ResNet50 trained on 100% Type 7 Training data* 69

Figure 42**.** *Grad-CAM of misclassified object for model trained on 75% Training dataset, rows represent type specific gradients and column represents layer based heatmap on based on the specific type* 70

Figure 43. *Grad-CAM of misclassified object for model trained on full-dataset, rows represent type specific gradients and column represents layer based heatmap on based on the specific type* 71

ABSTRACT

The problem of plastic recycling, particularly within municipal solid waste, fundamentally hinges on efficient sorting. Sorting is the bottleneck that, if optimized, could make the recycling process easier and economically viable. Unfortunately, the current state of plastic sorting is labor-intensive, costly, and fraught with health risks, which results in large amounts of plastic ending up in landfills and oceans. This mismanagement contributes directly to the growing issue of microplastics, which have infiltrated our ecosystems and even human bodies, posing significant concerns for future generations. The global plastic recycling market is expanding, expected to grow from $69.4 billion in 2023 to $120 billion by 2030, with a Compound Annual Growth Rate (CAGR) of 8.1% (MarketsandMarkets, 2023). In the U.S. alone, the market was valued at $2.19 billion in 2022 and is projected to increase at a CAGR of 7.9% by 2030 (Fortune Business Insights, 2024). Yet, without significant advances in sorting, this growth risks being unsustainable. With the rise of Machine Learning(ML) and the increasing accessibility of advanced technologies, there is real potential to revolutionize plastic sorting. Current methods range from basic statistical models to more sophisticated ML algorithms. However, these solutions remain far from addressing the full complexity of sorting plastics in real-world conditions. We propose a novel, low-cost ML system that addresses real-world challenges in plastic sorting: 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 36k plastic image samples, and (3) an interpretable approach leveraging Grad-CAM and t-SNE visualizations to tackle challenging cases like dark and distorted plastics. Our vision is to design a system that accounts for the inherent challenges of plastic sorting and adapts dynamically to the nuances of real-life waste streams.

1. INTRODUCTION

The common use of plastics in modern society has led to a dramatic increase in environmental pollution. Most plastic waste ends up in landfills, oceans, and other natural environments, causing significant ecological harm. Of the 6300 million metric tons (Mt) of plastic waste anticipated in 2015, 79% was sent to landfills, 9% recycled, and the remaining 12% was burned (Geyer, Jambeck, & Law, 2017). Additionally, around 1.7 Mt of plastic waste, accounting for 0.5% of the world's plastic waste ends up in the oceans yearly (Ritchie, 2023). A particularly concerning aspect of this pollution is the breakdown of larger plastics into Microplastics(MP) (tiny particles less than 5 mm in size). These particles are not only ingested by marine life, leading to bioaccumulation and potential toxicity but also have been found in human consumables, posing direct risks to human health. MPs are ubiquitous; recent studies have found them in far-flung places and have shown that they can disseminate their effects through the atmosphere over great distances (Ghosh, et al., 2023).

As time passes, plastic waste's negative effects on the ecosystem worsen. It has an immediate effect on human health, ecosystems, and the climate. Low-density plastics are easily transported into Microplastics and Nanoplastics (NP) through wind erosion when they infiltrate water or soil (Rezaei, Riksen, Sirjani, Sameni, & Geissen, 2019). These MP and NP particles impact plant growth, development, photosynthesis, and protein functions by causing oxidative stress and photosynthetic injuries, even altering expressions of thousands of genes in crops like rice, maize, and tobacco, disrupting important metabolic pathways, which are crucial for plant growth (Santini, et al., 2023). On the other hand, MPs constantly pile up in maritime environments which are often caused by industrial discharge, waste degradation, and fishing gear. The most common source of MPs that endanger aquatic life is bottom trawl wear and tear, which can enter food chains and bioaccumulate in marine species including fish and filter feeders (Grimaldo, et al., 2023). Over 311 tons of MPs are produced every year worldwide from only Danish seine fishing, primarily from demersal seines, which each fishing season loses 20–30% of their plastic mass owing to wear and tear (Syversen, Lilleng, Vollstad, Hanssen, & Sønvisen, 2022). According to (Liang, et al., 2024) MPs build up in the hearts, intestines, and gills of fish, which causes oxidative stress and compromises cardiovascular health; plastic particles have an impact on the immune system and blood cells in mussels. They also induce inflammation, oxidative stress, and cardiac

damage in terrestrial species like mice. Humans are susceptible to cardiovascular diseases such as thrombosis and myocardial fibrosis due to the presence of MPs in the heart, brain, and lungs through diet and respiration.

Since plastic waste is almost impossible to eradicate without impacting the natural environment, recycling plastics and reducing their production has been the only practical solution. In recent years, growing environmental concerns and strict regulations have highlighted the urgent need for more efficient plastic waste sorting for recycling. Most wealthy countries, including the USA, dealt with plastic waste by exporting it to China until 2018. China had been the largest importer of plastic garbage. Due to their increment in plastic consumption and increasing environmental concerns, China banned the import of plastics with more than 0.05% contamination in 2018 as part of the National Sword Policy (Dunn, 2018). The illegal plastic waste trade continued despite the 2021 Basel Convention amendment, wealthier nations, such as the United States and portions of the European Union, continued to ship polluted plastic to developing countries like Malaysia, Thailand, Indonesia, and Turkey (Basel Action Network, 2022). In response, tougher regulations have been enacted, spearheaded by Basel Action Network (BAN), which has pushed for greater transparency and enforcement in the global plastic trade (Winters, 2022). As a result, initiatives to enhance domestic waste management and recycling systems are slowly gaining momentum, aiming to curb the environmental impact of plastic waste.

Plastic recycling at the industrial level struggles significantly due to the complexities of accurately sorting mixed plastics, black plastics, films, and various polymer types. Surveys of Material Recovery Facilities (MRFs) indicate that up to 35% of colored plastics may be incorrectly sorted, leading to wastage, and 5% of sorted bales are often rejected due to contamination (Lubongo & Alexandridis, 2022). This situation is exacerbated by the limited market value of sorted materials beyond PET and HDPE, even small contamination levels, such as 50 ppm of PVC, can render an entire batch of PET unmarketable (Dubanowitz, 2000). MRFs utilize manual, automated, and semi-automated sorting. While manual sorting has a lower throughput and poses greater risks to workers, it is labor-intensive, time-consuming, and accurate when dealing with complicated materials. Conversely, automated sorting systems, including Near-Infrared (NIR), X-Ray Fluorescence (XRF), vision, and AI sorters are very effective at processing huge volumes rapidly saving human expenses, however, they have a high initial cost and are less accurate when it comes to plastic films and complex materials. Semi-automated sorting, which combines manual

labor with machine processing, improves efficiency compared to manual methods but is less effective than fully automated systems, allowing workers to handle complex or contaminated materials missed by machines (Lubongo & Alexandridis, 2022).

In this work, we introduce a complete pipeline for an innovative ML system designed for industrial-level sorting tasks. Our approach begins with the development of a robust system to curate a dataset comprising five types of plastics commonly found in Municipal Solid Waste (MSW) streams. Through careful preprocessing, we generate three dataset variants, each crafted to enhance model performance during training. We then experiment with state-of-the-art deep learning models to classify these plastics, leveraging a combination of architectures to assess their effectiveness. To further improve classification accuracy, we incorporate a segmentation model using YOLOv5, refining the cropping process to better isolate the objects of interest. Addressing the challenge posed by deformed plastics, we also propose a novel solution that adapts the model's handling of these irregular inputs. This approach not only improves the model's performance on complex cases but also offers a scalable pathway for enhanced sorting in real-world industrial applications. We concentrated on two major enhancements to overcome the shortcomings of the existing industrial plastic sorting systems, especially regarding sorting efficiency and affordability. To improve accuracy, we apply and optimize a deep learning model to deal with actual MRF situations. After segmenting each plastic object using a YOLO, this object is fed into the trained model that extracts visible features enabling the model to make a more informed and precise classification of the plastic type. Our system focuses on the classification of dark plastics, addressing this challenging task by collecting a diverse range of samples and applying targeted data augmentation. This approach overcomes the limitations of conventional computer vision algorithms, which struggle with the low reflectivity and light-absorbing properties of dark plastics. This presents a unique difficulty in capturing adequate visual features. To address this, we collect and augment our dataset tailoring the improvement of the model's ability to distinguish these plastics, overcoming the limitations posed by their optical characteristics and ensuring accurate sorting performance. We kept our system simple, limiting it to a conveyor belt, a camera, and a moderately powerful computer to keep costs down. However, the system is capable of adding lasers or a strong computer for faster segmentation as well as increased throughput. Finally, we interpret the model through Grad-CAM and t-SNE to better understand its decision structure ensuring it aligns with our expected outcomes. We placed particular emphasis on understanding

the misclassifications generated by our model, seeking to approach these errors from an engineering perspective that balances accuracy with cost efficiency. By analyzing the confusion matrix and utilizing cross-validation techniques, we systematically evaluated our model's performance on unseen data. This process allowed us to pinpoint the key areas for improvement, driving a more targeted optimization of the system while maintaining a focus on practical, cost-effective solutions. The core hypothesis driving this project is that the visual characteristics defining each plastic type are intrinsically linked to its chemical structure. By leveraging a robust dataset and the capabilities of advanced machine learning models, we aim to uncover and classify the subtle visual features that correspond to specific chemical bonds, offering insights into what visually distinguishes one plastic type from another.

The upcoming sections of this thesis will examine several crucial elements. Firstly, the **Related Work** section will cover the current dataset, sorting technologies, and their shortcomings. Secondly, the **Data** section will outline the kinds of data we collected, as well as the methods of collection and post-processing for training. The **Methodology** section will concentrate on our proposed pipeline, the various kinds of advanced deep-learning models for classification and segmentation, and our experiment with deformed objects. The **Experiments** section explores the training strategies we implemented, and the tuning methods applied to optimize performance. In the **Results** section, we will evaluate the models' performance on the test dataset by analyzing various metrics to gauge their accuracy and robustness. To enhance interpretability, we will employ Grad-CAM, offering a visual understanding of the regions the model prioritizes during classification. Additionally, we will leverage t-SNE to provide a two-dimensional representation of the feature space, revealing the degree of separation between classes and further illuminating the system's efficacy. Lastly, the **Conclusion** will provide an overview of the major discoveries and suggest more research ideas that could be done in this field of sustainability for a better future.

2. RELATED WORK

As both Artificial Intelligence (AI) and Recycling are emerging fields for engineers, a good amount of work in this field has been done. From data collection to AI models full-fledged industrial-grade sorting systems consisting of different types of sensors exist as we speak. This part of the thesis is divided into two subsections: existing datasets and existing work.

2.1 Existing Datasets

The relevance and richness of the training data greatly influence the accuracy and dependability of predictions; a high-quality dataset is crucial to the operation of machine learning models. In the case of Convolutional Neural Networks (CNN) well-annotated and diverse datasets are crucial to effectively learn intricate patterns for good prediction/classification. "More data beats better algorithms," as Halevy, Norvig, and Pereira (2009) contend, highlighting the fact that a model's performance frequently depends more on the availability of large and high-quality datasets than on the method of choice (Halevy, Norvig, & Pereira, 2009). Despite the ease of access to large quantities of plastic waste, we are unable to locate a diversified image/NIR dataset on MSW. The primary cause is that acquiring representative samples is challenging due to the wide range of plastic types and their physical/chemical conditions. On top of that, creating image/NIR datasets requires advanced equipment like NIR cameras and detailed preprocessing techniques to accurately capture the material properties of plastics, which is resource-intensive (Neo, Low, Goodship, & Debattista, 2023).

According to our project goal, we will only discuss MSW datasets that contain disposed materials. Trashnet (Thung & Yang, 2017) is a rather small dataset that provides images of six waste categories: glass, paper, metal, plastic, cardboard, and trash, a total of 2527 images were collected by smartphone. The main purpose of this dataset is waste classification. The images in this dataset are well-labeled, but not annotated for segmentation tasks. However, there is no fine-grained distinction between types of plastic (e.g., PET, HDPE, PP), which is a limitation for a more detailed classification of plastic types. Also, the images in this dataset don’t quite represent the MRF environment. The second dataset we came across is the open-source dataset Trash Annotations in Context for Litter Detection (Proença & Simões, 2020) also known as TACO which

is a growing dataset by mostly crowdsourcing. This dataset contains 1500 images with 4786 annotations in 60 categories namely: plastic straws, broken glasses, glass bottles, plastic film, etc. Most of the images in this dataset are collected outdoors. These images are human-annotated to serve segmentation, and the author has also trained Mask R-CNN for a benchmark. In conclusion, TACO provides images of plastic waste in real-world, cluttered environments, with natural lighting and various backgrounds, which makes TACO versatile datasets for detecting and classifying plastic waste in uncontrolled, urban environments, but the plastic types are not labeled which lacks it to serve as a dataset for plastic sorter in MRF environment.

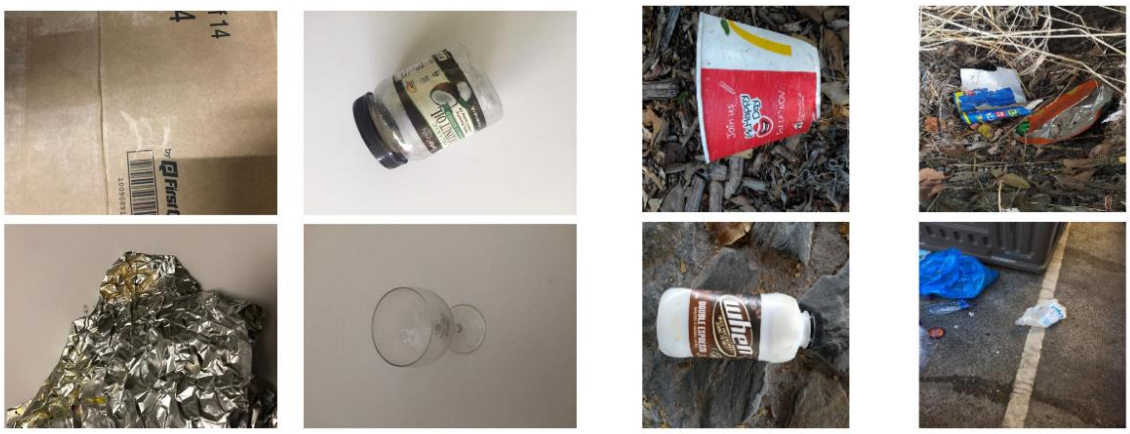


Figure 1. *Left- Images from Trashnet dataset, Right- Images from TACO dataset.*

ZeroWaste (Bashkirova, et al., 2021) is the most recent dataset that focuses on MSW. This dataset contains 10,715 images of waste items such as cardboard, rigid plastic, soft plastic, and metals, it was collected from an MRF institution in Massachusetts in regular operation. Their main goal was to curate a dataset that could be recreated in any MRF environment, addressing the challenges associated with deformable and translucent objects, which are common in MRFs and are harder to detect than rigid objects. They have also created three different subsets (Zerowaste-F, Zerowaste-S, Zerowaste-W) where F stands for focused, clean images with minimal background and distractions, S represents images taken in a semi-controlled environment where the background is more complex than F but not as cluttered as real-world, and finally W contains the wild-subset where the images reflect real MRF scenario. The authors utilize these subsets to evaluate their model performance, F being controlled settings to S and W being more realistic. The images in ZeroWaste are well-annotated so they perform well in tasks like segmentation and

classification, and they serve in sorting the materials the data was taken for. Although this dataset captures real-life MRF environments for waste sorting it can only be utilized for a general sorter which could only sort/segment a couple of materials in a cluttered scenario. Table 1 represents an overview of these datasets.



Figure 2**.** *Images from ZeroWaste dataset.*

Table 1**.** *Different datasets for plastic waste sorting purposes.*

Dataset	Classes	Number of images
Trashnet (2017)	6	2,527
TACO (2020)	60	1,500
ZeroWaste (2021)	7	10,715

2.2 Existing Work

Due to the ability of NIR spectroscopy to detect and classify different types of plastics based on their chemical composition, researchers have been using the NIR for plastic sorting. Since plastics have distinct molecular structures that affect how they reflect and absorb NIR light, NIR

is a great technology for distinguishing between plastics that have similar outward appearances but differ chemically. In "A Discrimination Model in Waste Plastics Sorting Using NIR Hyperspectral Imaging System" (Zheng, Bai, Xu, Li, & Zhang, 2018), the authors implemented a plastic sorting method purely based on NIR images. The authors collected 94 standard samples of the 6 plastics (PS, ABS, PP, PE, PET, PVC) from municipal waste disposal, scanned them continuously on the conveyor belt with an NIR spectrometer, and captured both spatial and spectral information of each element. The collected spectral data was smoothed, and noise was reduced by wavelet analysis and the Savitzky-Golay method, based on error measurements such as Root Mean Square Error (RMSE) and signal-to-noise ratio (SNR), the data was refined. The authors then used Principal Component Analysis (PCA) on the hyperspectral data to identify the unique and important features that contributed to the differentiation of each plastic. After identifying the key wavelengths using PCA, the authors built a Fisher discrimination model to do the final classification. Fisher discrimination seeks to maximize the separation between classes by calculating the distance between their centroids (barycenter) in a new feature space. In this instance, the discriminant function was derived from the 40 most relevant wavelengths (selected through PCA), and Fisher discrimination was applied to minimize within-class scatter while maximizing between-class separation.

The previous method used PCA and Fisher discrimination which are not machine learning models, so to accompany the NIR and ML Romans Maliks et al. (Maliks, 2021) investigated the application of CNN and NIR spectroscopy in conjunction with ML models and NIR to categorize primarily plastic bottles. The authors collected plastics that were wet, dry, scraped, and covered in dirt in an effort to replicate their data under real-world circumstances. The dataset has 2118 spectra samples that fall into 4 plastic classes (PP, HDPE, Double PET, PET, etc.) and 1 opaque class that includes 4 subclasses of the same elements. Since their dataset did not fairly represent all the classes, they artificially generated extra data points by interpolating between the training dataset's nearest spectral values. This allowed them to build four augmented datasets, improve quality, and prevent overfitting. The most augmented dataset (1:9) gave out 99.2 percent accuracy on a CNN consisting of a total of 7 layers (First 1D Convolutional layer + Average Pooling layer + Second 1D Convolutional layer + Max Pooling layer + Flatten layer + First Dense Fully Connected (FC) layer + Second Dense FC layer) with 139.301 thousand parameters and they fed 512 absorption coefficients at specific wavelengths. The model was tested on both wet and dry plastics. After

benchmarking their model against t-SNE, PCA, and Support Vector Machine (SVM) techniques, the authors concluded that NIR spectroscopy is a quick and reliable way to obtain spectra for polymer-type classification in the sorting of plastic garbage.

Even though NIR is a trustworthy method for classifying plastics, there are still problems, such as contamination, multilayer plastics, black plastic films, and the most prevalent problem-higher cost. To focus on the issue of black plastic films and improve the classification with the help of new ML models Attilio Sbrana et al. (Sbrana, et al., 2023) came up with the Neural Basis Expansion Analysis Time Series (N-Beats) model which achieved a recall of 0.67 for black plastics. They collected a total of 1491 hyperspectral images of 4500 plastics and 161 billion sample pixels under a conveyor belt, to differentiate between the belt and plastics they employed Facebook AI Similarity Check (FAISS) which also reduces the computational load on the classifier by focusing on the plastic. N-BEATS is a deep learning architecture designed primarily for time series forecasting by decomposing waveforms into trends and seasonal components. (Oreshkin, Carpov, Chapados, & Bengio, 2019). The authors edited N-BEATS for classification by adding a sigmoid function to an output vector to handle multi-class output and incorporated Mish instead of Rectified Linear Unit (ReLU) as an activation function for better performance. They also trained LightGBM to benchmark N-BEATS, for black plastics, N-BEATS had a recall of 0.67, substantially better than LightGBM's 0.45. While this approach seems sound in theory, it only processes two images (2800*640 pixels) per minute, which is not optimal for industrial sorters when hand sorting is quicker and more accurate.

To deal with the high cost of NIR machines and complicated systems Choi et al. (Choi, Lim, & Yoo, 2023) developed an image-based classification that especially distinguishes similar plastics like PET and PET-G, which the authors aim to overcome by analyzing the shape and texture of plastics instead of just chemical composition. The authors took advantage of new robust deep-learning methods for computer vision. They incorporated three depth cameras to ensure each object was captured and collected from the visual data at 2m/s conveyor belt. Using YOLO-8 as a real-time classifier with a 91.7 percent Mean Average Precision (mAP) and air nozzles for physical sorting, their system maintained efficiency in real-world settings. The paper discusses the incorporation of NIR devices since it is challenging for a pure computer vision algorithm to accurately sort everything due to the vast array of patterns and forms found in plastics.

Table 2. *Existing approaches to solving plastic waste sorting*

Approach/Work	**Data**	**Technology Used**	**Details**	**Cons**
Discrimination Model with NIR (Zheng, Bai, Xu, Li, & Zhang, 2018)	Spectral (6 classes) 94 samples	NIR, PCA, Fisher Discrimination	Captured and processed NIR hyperspectral images with PCA & FD to classify 6 types	Hight cost, unreliable real-life accuracy, ineffective on mixed, dark plastics, limited data
Multispectral classification with deep CNN (Maliks, 2021)	Spectral (PET, HDPE) 2118 samples	NIR, Data augmentation, 7 layers CNN, t-SNE, PCA, SVM	Collected & augmented NIR spectral data, implemented CNN and benchmarked against t-SNE, PCA, SVM (accuracy 99.2%)	High Cost, unreliable real-life accuracy, ineffective on mixed, dark plastics, limited classification types
Classification with NIR and DL (Sbrana, et al., 2023)	Spectral (17 classes) 1491 samples	NIR, FAISS, N-BEATS, LightGBM	Adapted N-BEATS with FAISS to improve classification of black plastics (recall 0.67)	Slow processing, high computational load, scalability, unfeasible for real-time application
Integrating image sensors and DL (Choi, Lim, & Yoo, 2023)	Image (PET, PET-G) 2000 samples	Depth Camera,data augmentation, YOLOv8,	Built cost-effective image based classification with YOLO and depth cameras (91.7% mAP)	Doesn’t consider other types and varying shapes, not feasible in real-life scenario

3. DATA

One of the major goals of this project was collecting extensive data that can work on the current UHV blue sorter (United States Patent No. US20210346916A1, 2021) pipeline. As we wanted to mainly focus on sorting municipal plastic waste, most contributing plastics belonged to 4 classes: Type 1 (PETE), Type 2 (HDPE), Type 5 (PP), Type 7 (other), and representation of other 3 classes: Type 3 (PVC), Type 4 (LDPE), Type 6 (PS) were lower. We also discovered thin plastic bags, metal fragments, aluminum cans, and other items. Depending on their significance and quantity, we also collected data on aluminum cans and metal pieces that weren't added to this project. A thorough explanation of data collection and preprocessing will be provided in this section.

3.1 Dataset Collection

We bought numerous bales of MSW from Florida, each bale (Figure 3) consisting of different types of unsorted materials and weighing around 800 kg. We hand-sorted the bales extensively and put the plastics into different gaylord (Figure 4) 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 each day. 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 in order 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 or 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.

The data collection process is semi-automated, using a UHV blue sorter (Figure 6) for high-speed image acquisition. While automatic, human oversight ensures images are well-centered and visible. The sorter uses a dual conveyor system, with two Basler cameras and one sets of bright lights per lane. Images are captured automatically when objects are detected on the conveyor belt, except for dark plastics, which require manual adjustment of segmentation settings. Approximately 1,000 images are collected every 5 hours and 2 hours extra for manual outlier removal. A total of

33,387 images from 5 bales of plastic were captured, with each object imaged at least three times from different angles, resulting in a robust dataset of 5 types of plastic.

In Figure 5 we are presenting a simpler version of the UHV Blue Sorter (United States Patent No. US20210346916A1, 2021) for data collection. Plastic objects are placed on a conveyor belt, moving steadily toward the vision system positioned near the middle. As each object passes beneath the camera, an image is captured in real time, and the segmentation pipeline is activated to catch the object. Currently the system operates solely for data collection, with the conveyor, camera, and segmentation modules functioning in unison. Later, classifier will be introduced, the same system will be used for sorting, leveraging the captured data to assign plastics to their correct categories. This modular approach allows for a seamless transition from passive data acquisition to active classification and sorting, with minimal changes to the system architecture.



Figure 3**.** *Unsorted Bales of Municipal Solid Waste*



Figure 4**.** *Gaylord boxes with Plastics sorted into specific types*

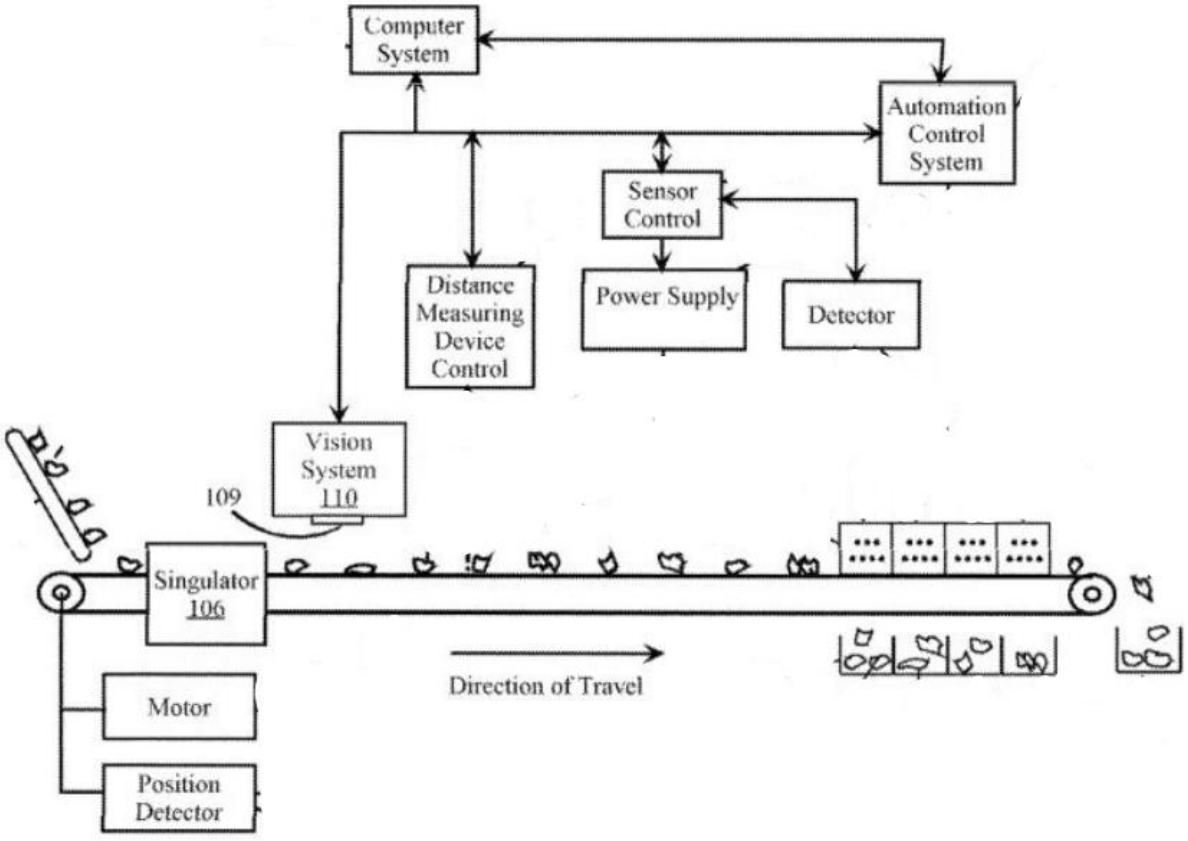


Figure 5**.** *UHV Blue Sorter data collection system diagram*



Figure 6. *UHV Blue Sorter CPU head*



Figure 7. *Plastic material under the UHV Blue Sorter camera*

3.2 Dataset Overview

The original dataset consists of 33,387 images, each at a resolution of 640x465 pixels. The 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 (PVC) plastics had the lowest representation, with only 55 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.

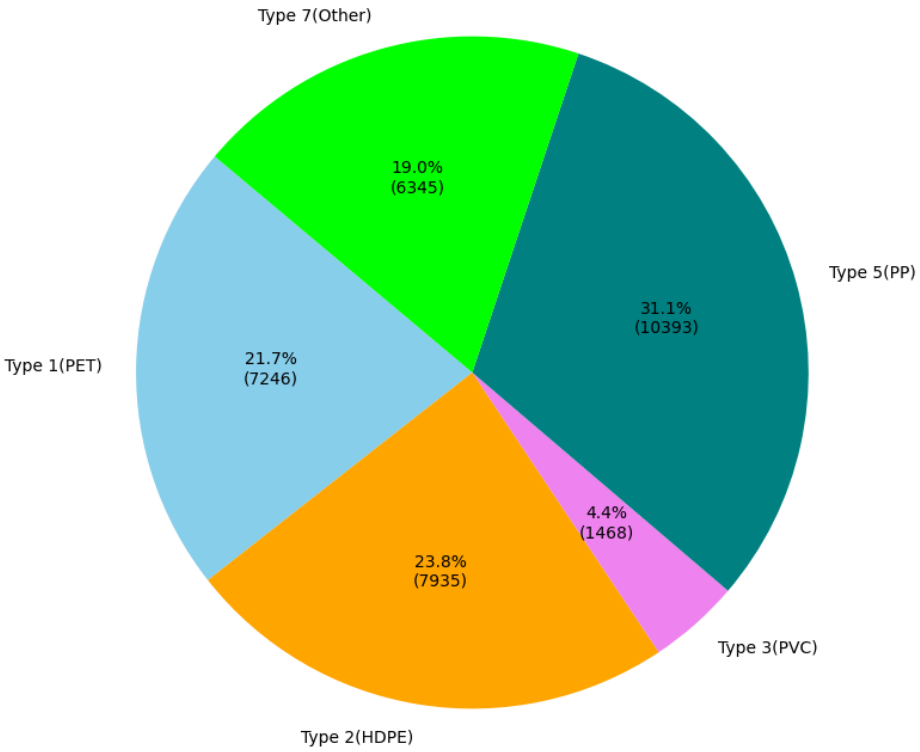


Figure 8. *Dataset distribution for 33,387 images of 5 types of plastics*



Figure 9**.** *Sample images of each plastic type*

We curated three datasets from the original images to evaluate model accuracy under conditions similar to the machine's view of the objects on the conveyor belt. The first dataset (Figure 10) 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 second dataset (Figure 11) was constructed by centering the cropped objects on a uniform, dark square background. Finally, the third dataset (Figure 12) contains the objects alone, with the conveyor belt entirely

removed. Conversely, the Test dataset 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 MSW, represents an unprecedented effort and holds the potential as a valuable resource for training different CNN 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. However, the challenge remains that manual sorting is required for accurate data collection, which is inherently labor-intensive and poses health risks. A more efficient approach could involve automating the sorting process through the sorting machine itself. 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 10**.** *Sample Images from dataset-1 (D1)*



Figure 11**.** *Sample Images from dataset-2 (D2)*



Figure 12. *Sample Images from dataset-3 (D3)*

4. METHODOLOGY

This section provides a comprehensive overview of the various CNN models trained for this project. Our focus was on training modern CNN architecture that could be integrated into the industrial sorting system (UHV blues sorter). The sorter is equipped with a single 1080Ti GPU, so we measured the final model's inference time in relation to the time required for the sorting application to segment and crop the image to the appropriate input size. Our objective is to develop a model that strikes a balance between accuracy and efficiency, enabling the system to classify and sort plastic materials with a reduced latency. Figure 13 provides an overview of the whole system, structured into two pipelines: data collection and classification. In data collection pipeline we manually sort MSW. The sorted plastics are then imaged using the UHV Blue Sorter, generating a comprehensive dataset, which concludes the data collection phase. In the classification pipeline, the dataset is preprocessed and fed into training classifiers and segmentation models, each fine-tuned through parameter optimization. Finally, these refined models are deployed on the UHV Blue Sorter, enabling real-time plastic classification.

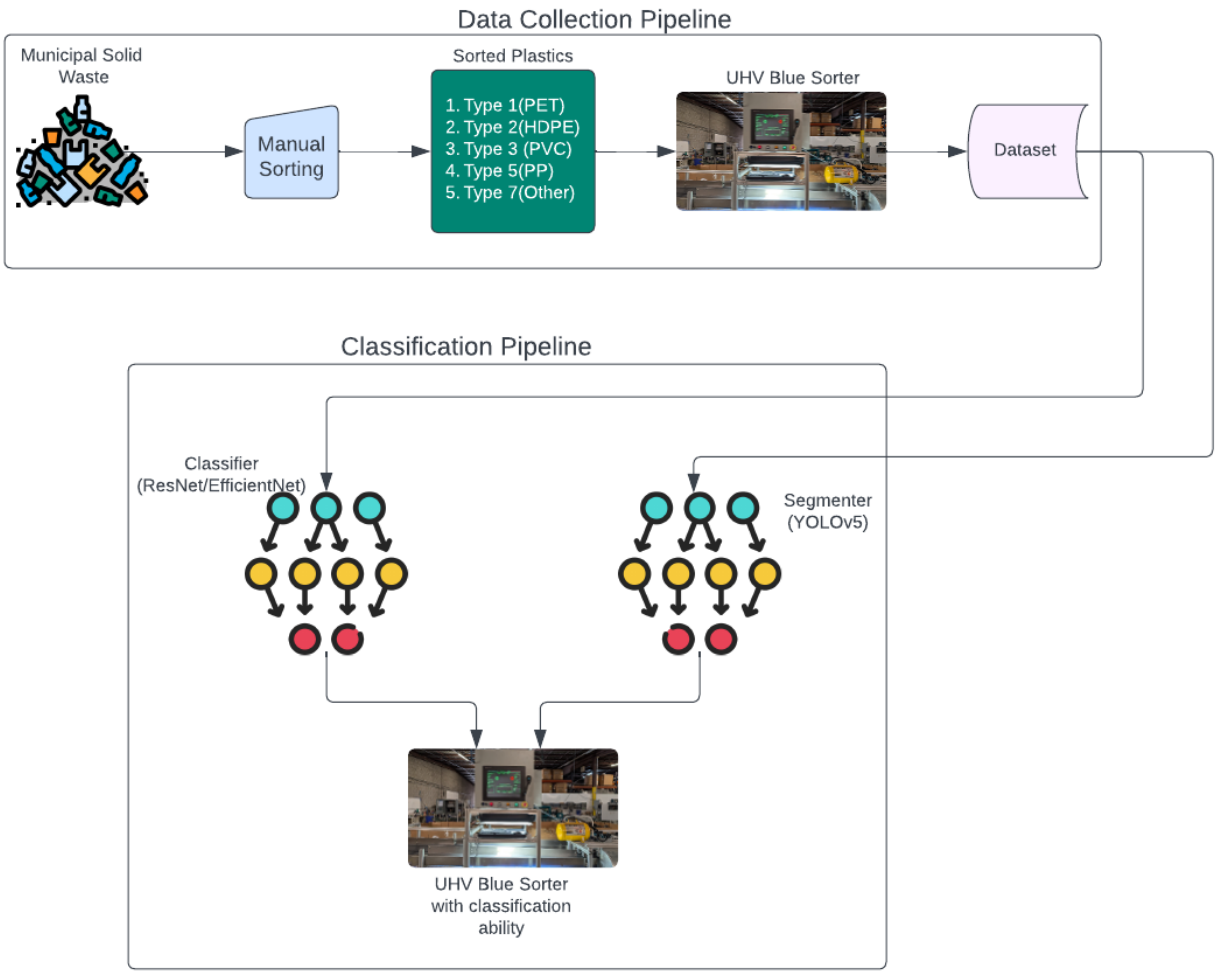


Figure 13. *Pipeline for Industrial Sorting*

4.1 CNN Models

We trained a total of five models across the three dataset variants we curated specifically for this task. 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.

4.1.1 ResNet

ResNet or Deep Residual Network (He, Zhang, Ren, & Sun, 2015) solved the vanishing gradient problem that traditional Deep Neural Networks (DNN) were facing. In ResNet, the core innovation lies in residual learning facilitated by skip connections. Traditionally, in DNN, each layer is tasked with learning a direct mapping, *H(x)*, where *x* is the input and *H(x)* is the output. However, in ResNet, rather than learning this full transformation, the network learns the residual function, *F(x)*, which represents the difference between the input and target output. In simple terms: the network only needs to learn how much the output should deviate from the input, rather than learning the output directly. This formulation makes it easier for the network to optimize, as learning small residuals is often more efficient than learning full transformations, particularly in very deep architectures. Skip connections play a critical role in this process by allowing the input (*x)* to bypass intermediate layers and be directly added to the output of the residual function. This direct pathway ensures that gradients can flow unimpeded through the network during backpropagation, which effectively mitigates the vanishing gradient problem that typically plagues deep networks. By reframing the learning process to focus on residuals and using skip connections to facilitate smoother gradient flow, ResNet enables the training of very deep models: sometimes with hundreds or even thousands of layers without the performance degradation that traditional architectures experience.

ResNet is well-suited for image classification, as it allows for the training of deep networks without suffering from vanishing gradients or performance degradation. Pretrained on large datasets, ResNet models offer state-of-the-art performance with minimal resources, making them ideal for transfer learning. Its' scalable architecture, with variants like ResNet-18, ResNet-34, and ResNet-50, enables flexible depth for different tasks. For our project, we trained ResNet-18, ResNet-34, and ResNet-50 to identify a model that balances well with our hardware constraints and avoids overfitting/underfitting on our dataset. We selected ResNet for its balance between performance and efficiency, particularly suited to the constraints of our project. Our dataset, while limited in size, requires a model capable of capturing fine-grained details without overwhelming the 1080ti we have at our disposal. ResNet's architecture, with its residual connections, allows us to scale effectively as data volume grows or compute resources improve, making it a future-proof choice that aligns well with both our current limitations and our long-term need for scalability.

4.1.2 EfficientNet-V2

To address the scalability challenges of traditional neural networks, EfficientNet (Tan & Le, 2019) introduced a new approach called compound scaling, which optimizes the way we scale a model's dimensions: depth, width, and input resolution by balancing them proportionally. This approach enables an efficient trade-off between accuracy and computational cost. Scaling each dimension independently leads to significant inefficiencies. For instance, increasing the depth of a network can result in overfitting and complex models, as well as difficulties in optimization due to vanishing gradients. Expanding the width of a network improves its capacity at each layer but significantly increases memory consumption during training. While increasing input resolution enables the model to learn finer details, it comes at the cost of slower inference and training time.

Compound scaling introduces a scaling coefficient (φ) that simultaneously adjusts all three dimensions (depth, width, and resolution) based on the available computational budget. The scaling is governed by the following relationships: depth $(d) = \alpha^{\wedge}\varphi$, width $(w) = \beta^{\wedge}\varphi$, resolution $(r) = \gamma^{\wedge}\varphi$. The constants α, β, and γ are empirically determined through grid search, optimizing for an effective balance. For example, if $\alpha = 1.2, \beta = 1.1, \gamma = 1.15$, and $\varphi = 2$, this means: depth increases by $1.2^2 \approx 1.44$, leading to 44% more layers, width increases by $1.1^2 \approx 1.21$, adding 21% more channels, resolution increases by $1.15^2 \approx 1.32$, meaning the input image resolution is scaled up by 32%. By scaling all dimensions simultaneously, compound scaling ensures that we achieve significant performance improvements without the inefficiencies introduced by traditional scaling methods.

The basic EfficientNet model begins with an optimized baseline architecture and scales up using a compound coefficient that ensures a balanced increase in FLOPS across network depth, width, and resolution. We employ the enhanced version, EfficientNetV2 (Mingxing Tan, EfficientNetV2: Smaller Models and Faster Training, 2021), which brings several key improvements: faster training, compact models, and better accuracy, making it highly suited for modern applications. While maintaining the core principle of compound scaling, EfficientNetV2 introduces more effective handling of higher input resolutions, a critical factor for tasks like object detection and image segmentation, where fine details are important. One of the standout features of EfficientNetV2 is its progressive learning strategy, where lower-resolution images are used in the earlier stages transitioning to higher-resolution images in late layers. This allows the model to reduce memory consumption and accelerate training without compromising performance. Given

that our project focuses on multi-class object detection, capturing intricate details of diverse plastic types is essential. So, we train EfficientNetV2 with higher-resolution images compared to ResNet, aiming to benchmark model performance and assess the implications of our architectural choices.

4.1.3 YOLO-V5

You Only Look Once (YOLO)-v5 is an object detection architecture built on the backbone of CSPDarknet53 to predict both object classes and bounding box coordinates in a single forward pass. YOLO-v5 (Jocher, 2020) is a successor of YOLO (Redmon, Divvala, Girshick, & Farhadi, 2016). The introduction of SiLU activation, along with the auto-optimized anchor boxes, contributes to smoother gradient flows and more precise bounding box predictions, while mosaic data augmentation helps the model generalize better to varied contexts and object scales. It can unify localization and classification within a single, efficient pass through the network. This streamlining, combined with PANet for multi-scale feature aggregation, ensures that the network is equally adept at detecting small and large objects in diverse settings. As a result, YOLOv5 represents a major step forward in the democratization of object detection offering a lightweight, scalable solution that performs with both rigor and speed, empowering a broader range of real-time applications.

YOLOv5 serves as a tool to precisely localize and extract bounding boxes for each object, allowing us to crop and feed these regions into a more specialized classification model such as ResNet. Currently, we rely on a basic segmentation approach, but it struggles with black plastics and often produces multiple segmentations for larger objects, leading to redundant classification tasks. This redundancy not only introduces computational overhead but also reduces the model's overall efficiency. By training YOLOv5m, we aim to explore whether it can effectively replace our segmentation pipeline. While it is true that our current method outpaces YOLO in terms of raw speed, YOLOv5 promises far superior performance in terms of precision and robustness, particularly when considering the computational resources at our disposal. The hope is that YOLOv5 will mitigate the segmentation pitfalls we're encountering, offering a more balanced trade-off between accuracy and computational cost, and ultimately streamlining the object detection and classification workflow.

5. EXPERIMENTS

In this section, we will explore three key experiments designed to tackle the classification task, and segmentation task with a particular focus on assessing model accuracy when dealing with deformed objects. We will first provide a detailed overview of the classifiers we have trained and integrated into the UHV blue sorter, highlighting their role in the broader system. Following that, we will delve into the challenges posed by deformed materials, an issue that complicates object recognition in real-world sorting environments. For these experiments, we selected the optimal dataset to present as the result. This choice was made after thoroughly training on all datasets and rigorously evaluating their accuracy, ensuring that the most reliable dataset (D-1) (Figure 10) was used for the outcomes.

5.1 Training Classifiers

The following hyperparameters guide our training processes for most of the training with different combinations till we get the best model:

Learning Rate=0.001, **Batch Size**=64, **Epochs**=2000, **Optimizer**= SGD/ Adam, **Patience**=20

5.1.1 ResNet-18

As the name suggests, ResNet-18 comprises 18 layers, structured around an initial convolutional layer and a final fully connected layer, with four intermediate stages in middle of them. Each stage contains two residual blocks and two convolutional layers, forming the core architecture of the network. 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:

```
model.fc=nn.Sequential(
        nn.Linear(512,256),
        nn.BatchNorm1d(256),
        nn.ReLU(inplace=True),
        nn.Linear(256,5))
```

We employ 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 plot 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 is overfitting or underfitting. To further refine the training process, we implement early stopping, a mechanism that halts training when there is no improvement over a predefined number of epochs. This not only prevents overfitting but also conserves computational resources, ensuring the model converges efficiently. We also apply class weighting, with a particular emphasis on Type 3 plastics, as this class is underrepresented in our dataset. By assigning 2.0 weights to Type 3 during training, we ensure that the model pays proportionally more attention to this class, compensating for its lower frequency. This strategy helps mitigate the risk of bias toward the more prevalent classes.

The training for ResNet-18 stopped at 1500 epochs as it stopped growing in accuracy. Figure 14 visualizes the validation/training curves of the most final epoch. The loss curve indicates a steady decline in both training and validation losses, with validation loss consistently lower than training loss, suggesting that the model generalizes well without significant overfitting. The decreasing trend in loss shows that the model is continuing to learn effectively over time. On the other hand, the accuracy curve reveals a positive upward trend for both training and validation accuracy. While training accuracy improves steadily, validation accuracy remains slightly higher, which may indicate that our validation set is less complex. However, both curves demonstrate strong convergence, with the model approaching high accuracy levels around 92%.

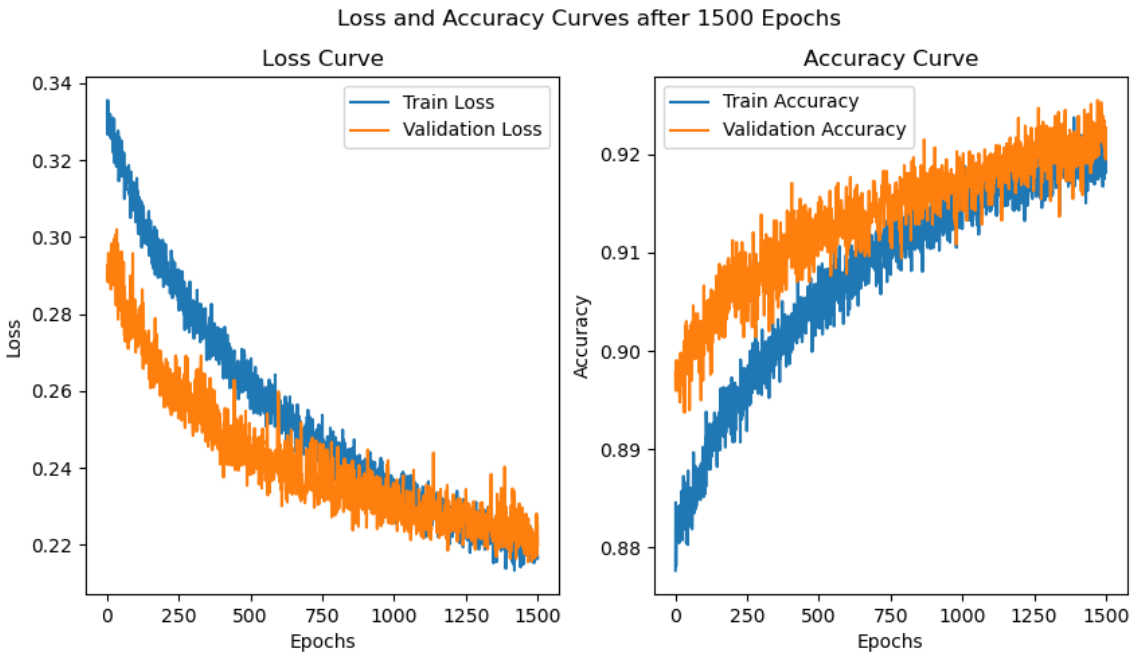


Figure 14. *Loss/ Accuracy curve for ResNet-18*

5.1.2 ResNet-34

With its 33 convolutional layers organized into residual blocks and capped with a fully connected layer, 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. By randomly dropping units during training, the network is forced to develop more resilient, generalized feature representations, preventing it from becoming too reliant on specific neurons. In doing so, it encourages the model to distribute learning across multiple pathways, leading to more robust performance on unseen data. Below is the structure of the final Fully Connected Layer for ResNet-34:

```
model.fc=nn.Sequential(
        nn.Linear(512, 256),
        nn.BatchNorm1d(256),
        nn.ReLU(inplace=True),
        nn.Droptout(0.3),
        nn.Linear(256,5))
```

Although the hyperparameters used to train this model are identical to those employed with ResNet-18, we experimented with Adam as an alternative optimizer, diverging from the traditional choice of SGD. The result was a noticeable improvement in both convergence speed and model performance. This highlights the importance of optimizer choice, even when other parameters remain constant.

Although ResNet-34 kept learning, we stopped it at 1995 epochs as we were satisfied with the accuracy. In Figure 15 loss curve, we observe a sharp initial drop in both training and validation loss within the first 200 epochs, followed by a gradual plateau. The validation loss remains consistently lower than the training loss, suggesting that the model generalizes well and avoids overfitting. The small gap between the two curves indicates a healthy balance between training and validation performance. In the accuracy curve, the model's accuracy improves dramatically during the early epochs, with both training and validation accuracy approaching saturation around 90%. The validation accuracy is consistently higher than the training accuracy, further suggesting that the model is not overfitting, but instead has a stronger performance on the validation set. Overall, this model could be trained with more epochs.

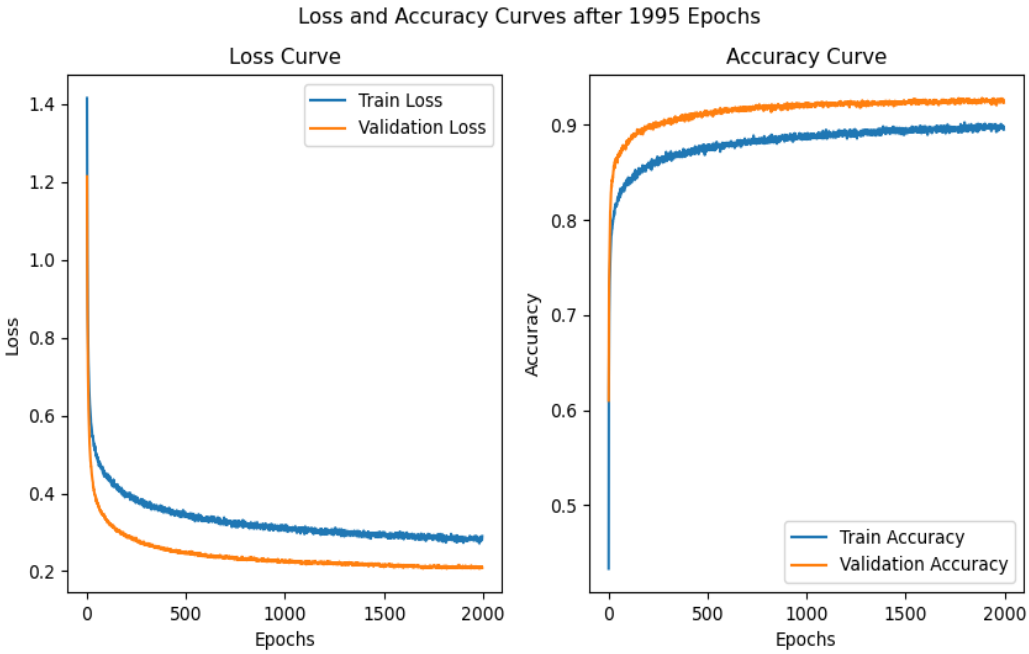


Figure 15. *Loss/ Accuracy curve for ResNet-34*

5.1.3 ResNet-50

Our final trained model in the ResNet group is ResNet-50 which is a fairly big architecture of the ResNet line. With 1 input layer and 1 output layer sandwiching 4 stages of residual blocks which contain 48 layers 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. At this level of depth for CNN architecture the models prone to start compromising accuracy as wanted, but despite having more layers, ResNet-50 is not prone to the vanishing gradient problem due to the use of skip connections in the residual blocks, which allows gradients to flow more easily through the network during backpropagation. 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 UHV blue sorter'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. The structure of two optimized models (R1, R2) is outlined below:

R1

```
model.fc=nn.Sequential(
        nn.Linear(2048,1024),
        nn.LeakyReLU(negative_slope=0.
01,inplace=True),
        nn.Droptout(0.7),
        nn.Linear(1024,5))
```

R2

```
model.fc=nn.Sequential(
        nn.Linear(2048, 512),
        nn.BatchNorm1d(512),
        nn.LeakyReLU(negative_slope=.02,
inplace=True),
        nn.Droptout(0.7),
        nn.Linear(512,5))
```

As ResNet-50 is a quite strong model compared to the earlier models, we can see a good performance in fewer epochs and the model starts overfitting on the validation dataset after a certain accuracy. In fact, the model starts to unlearn important features and have poor accuracy on test set. The loss curve for R1 (Figure 16) shows a sharp drop in both training and validation loss during the initial epochs, particularly within the first 20 epochs. After this initial phase, the training loss continues to decrease smoothly, while the validation loss fluctuates but generally stabilizes at a low level. The accuracy curve for R1 achieves rapid gains within the first 20 epochs. Both training and validation accuracy quickly approach near-perfect performance, with validation accuracy fluctuating slightly around the training accuracy. The model learns effectively and generalizes well, as both accuracies converge above 96%, with only minor variations in the validation accuracy as training progresses.

For R2 (Figure 17) there is a sharp decline in both training and validation loss during the first 20 epochs, indicating rapid learning. After this initial drop, the training loss continues to decrease steadily, while the validation loss fluctuates slightly but remains lower overall. The proximity of the two curves suggests that the model generalizes well, without significant overfitting. In the accuracy curve, we observe that both training and validation accuracy increase sharply within the first 20 epochs, with validation accuracy quickly surpassing the training accuracy. Beyond this point, both curves begin to stabilize above 94%, with validation accuracy fluctuating but consistently remaining slightly above training accuracy. The fluctuations in

validation accuracy suggest some variance in the validation set, but overall, both curves show strong convergence.

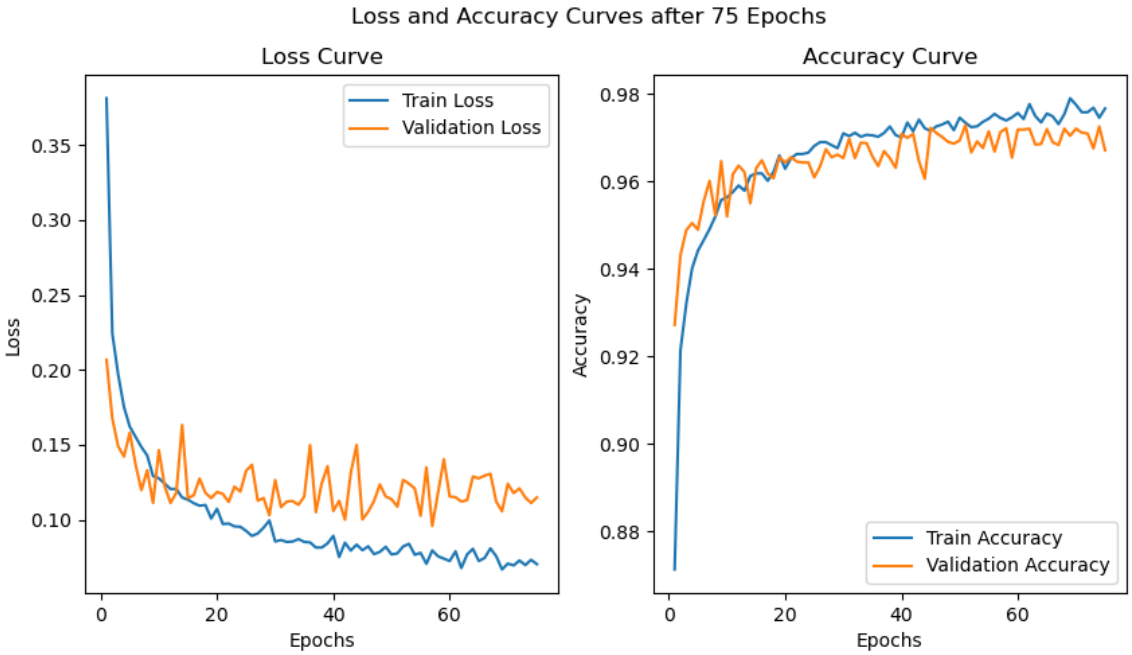


Figure 16. *Loss/ Accuracy curve for ResNet-50 (R1)*

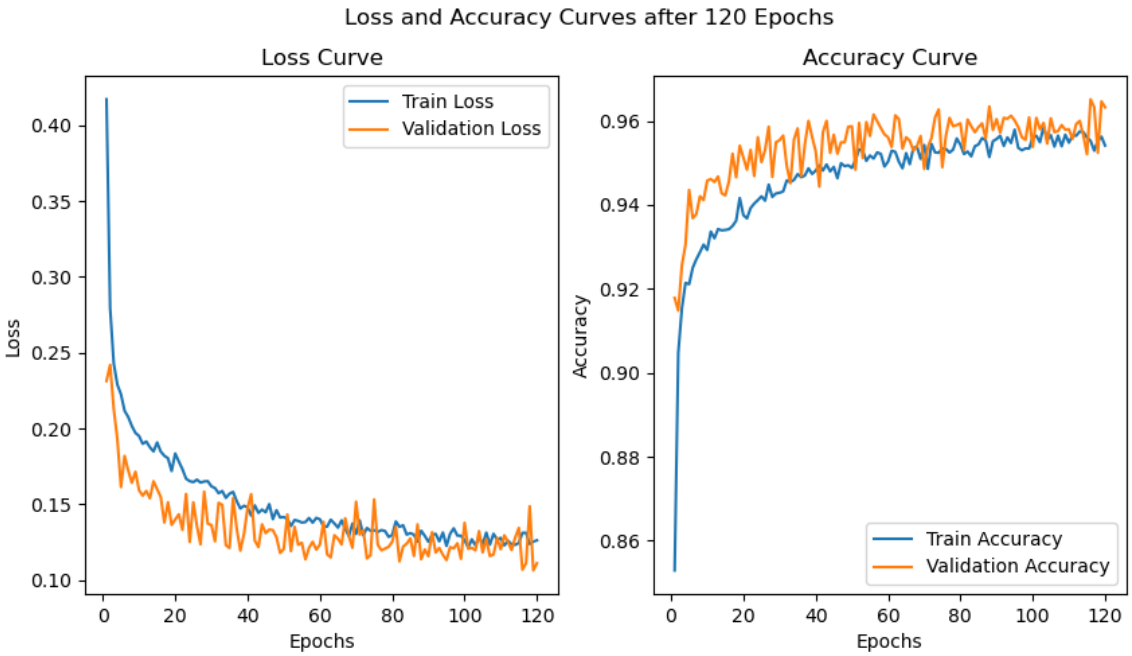


Figure 17. *Loss/ Accuracy curve for ResNet-50 (R2)*

5.1.4 EfficientNet-v2m

The EfficientNetV2 family represents a breakthrough in computational efficiency and scalability, built on the principles of compound scaling. This family includes three primary variants: EfficientNetV2-S (small), EfficientNetV2-M (medium), and EfficientNetV2-L (large). While each model shares the same core architectural foundation, they differ in layer depth and the number of trainable parameters, scaling progressively from S to L. EfficientNetV2-S, with 55 layers, is the lightest variant, optimized for speed and efficiency. EfficientNetV2-M, with 74 layers, balances performance, and computational demand, containing 54 million parameters and 24.3 billion floating-point operations (FLOPs). EfficientNetV2-L, the deepest model at 118 layers, achieves the highest accuracy but at the cost of increased computation. For our purposes, 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 the final fully connected layer, leaving us only 3,064 trainable parameters. The model was trained with hyperparameters closely matching previous experiments, with each training epoch taking 45 seconds and validation epoch taking 13 seconds. We set early stopping, after 60 epochs it ended since it was no longer improving.

The accompanying loss and accuracy curves (Figure 18) tell an important story: the model is learning; but at a sluggish pace. More critically, the validation loss and validation accuracy deviate significantly from the training loss and accuracy, signaling a classic case of overfitting. The validation loss is consistently much lower than the training loss, and the accuracy curves fail to converge, implying that the model generalizes poorly despite good performance on the training set. This divergence between training and validation metrics suggests that either the dataset or the model's configuration needs adjustment. To bring the curves closer together, we have several paths to explore: we can extend training to more epochs, modify the dataset for better diversity, or fine-tune other layers to allow more flexible learning. Without intervention, the model's current trajectory indicates that it will continue to overfit the training data while over-relying on validation patterns it has memorized.

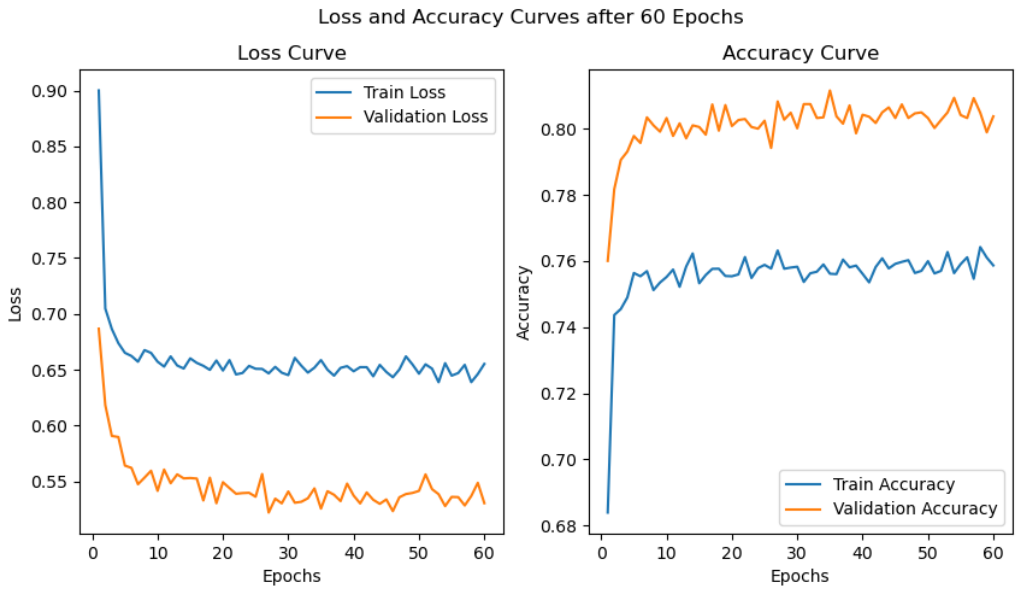


Figure 18. *Loss/ Accuracy curve for EfficientNet-v2m*

5.2 Training YOLO-v5 for Segmentation

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 the labels of those bounding boxes.

Once the dataset is transformed, YOLO's training procedure is simple. A YAML configuration file defines the class names and paths to the training data. Given the small scale of the dataset and the single-class nature of the problem, we opted for a minimalist set of hyperparameters fine-tuned to prevent overfitting. Training around 500 epochs ensures the model has ample opportunity to converge and adapt to the nuances of the data. After training the YOLO we segment the rest of our data and calculate the time it takes. Figure 19 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, which is 5.2 times slower than the basic segmentation approach. However, this additional processing time is a trade-off worth making, as it delivers more precise and fully automated segmentation, reducing the need for manual intervention.

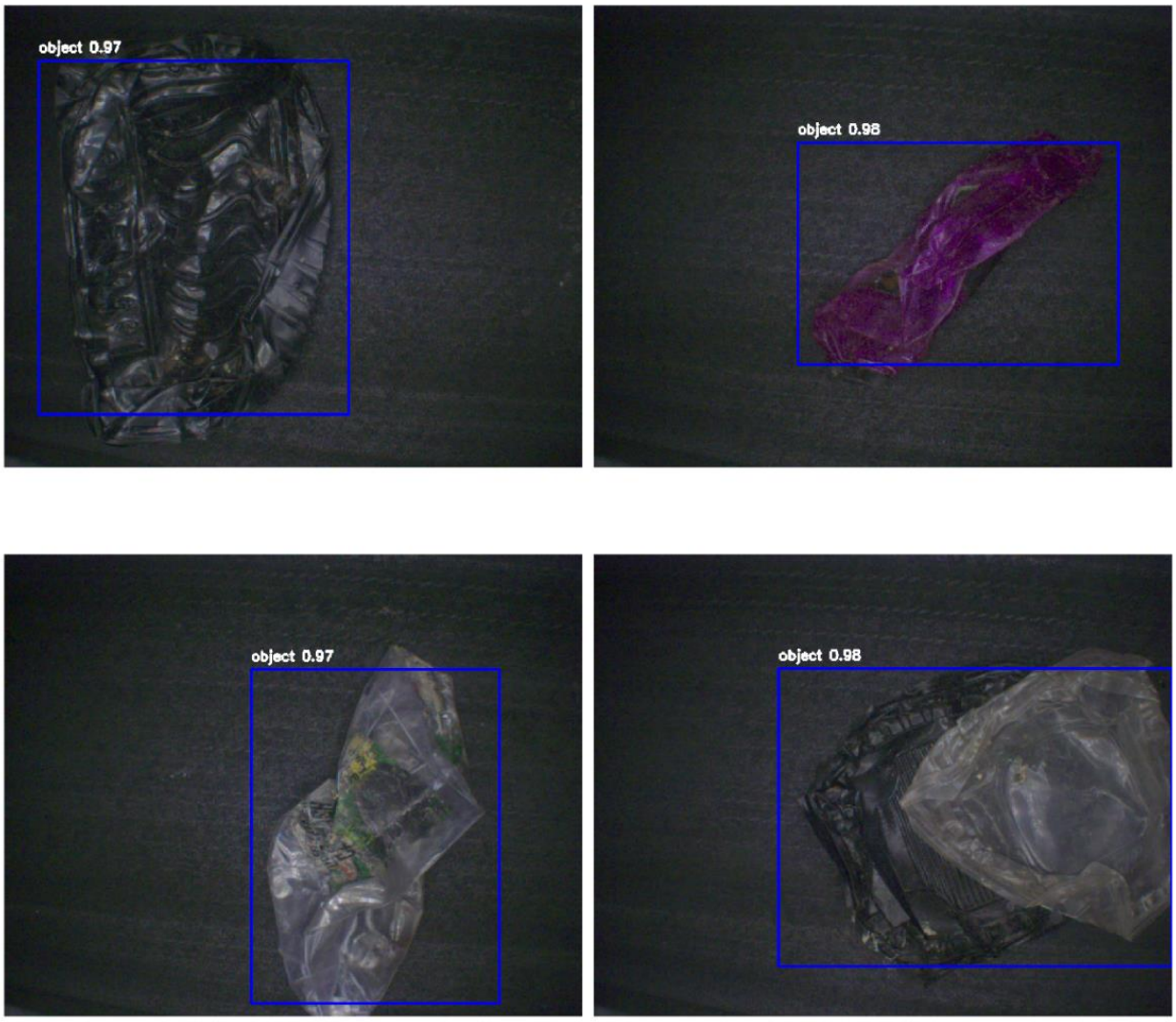


Figure 19. *Samples of YOLO segmentation on unseen data*

5.3 Model Accuracy Evaluation on Deformed Objects

Although our primary focus is on plastic sorting, one of the more experimental aspects of our work has been the classification of deformed objects. This line of inquiry emerged from dealing with Type 7 plastics, a category dominated by Arizona tea jugs (98% of total Type 7). The central question is how deformation affects the model's ability to classify these objects accurately. While a human can visually identify a deformed jug to some extent, as the deformation becomes more pronounced, even though the material remains the same, the task of classification becomes significantly more challenging. Our goal is to explore at what point the model's predictions start to deteriorate as a result of surface deformation.

To investigate this, we begin by examining a standard, uncrushed Arizona tea jug (Figure 20) lying flat and comparing it to a visibly crushed ones (Figure 21). The crumpling ability of plastics makes such deformations a natural occurrence, and understanding how the model responds to these variations is crucial. While Near-Infrared (NIR) data could address this problem directly, we aim to understand the limits of machine learning models when faced with such distortions.

Our approach starts with looking at model accuracy trained on half data that contains both pristine and deformed plastics of type 7. We then plot the performance degradation, which serves as an anchor point, signaling where the model's capacity to generalize falters. We test this model on the test dataset and separate the images that have less than 0.5 confidence. Then we implement two more training considering 75% of total deformed+pristine plastics and finally 100% of the dataset, and go through the same testing process. So, we have three models: trained on 50%, 75%, and 100% of deformed+pristine Type 7 plastics and three folders of images that contain objects with lower confidence levels for each of these models. To diagnose the root cause, we turn to Grad-CAM, generating heatmaps for both fine and deformed objects. These visualizations offer a window into the model's internal workings, revealing where it focuses its attention. For fine objects, the model displays sharp, localized activations focusing on critical features with precision. However, when confronted with deformed objects, these activations become diffused or misaligned, mostly focusing on the edges of the objects, as the model struggles to maintain its focus on relevant patterns.

Our hypothesis posits that the model's performance, particularly its accuracy, should progressively improve as we introduce greater variability in the dataset including more deformed objects. The underlying premise is that increased variability provides the model with a richer

representation of the problem space, enabling it to generalize better to unseen examples. By measuring the percentage increase in accuracy as more diverse samples are added, we can quantify this effect. Crucially, this hypothesis extends beyond only Type 7 plastics, as it is particularly well-suited to all types of plastics, which are inherently deformable. Whether a plastic object is pristine or deformed, its fundamental characteristics remain consistent, making it an ideal domain to test this approach. In conclusion, this experiment represents a critical step toward enhancing the model's ability to handle real-world scenarios, strengthening its overall capacity to generalize beyond controlled conditions.



Figure 20. *Samples of Ideal Type 7 Plastics*



Figure 21. *Samples of Crushed/deformed Type 7 Plastics*

6. RESULTS

In this section, we present the overall results for the trained CNN models and engage in a constructive discussion on the implications of our findings. This section is divided into two subsections: Model Analysis, where the performance of the trained models is benchmarked for plastic sorting, and Discussion, which explores the broader possibilities and potential future work on the overall pipeline arising from the results.

To acquire a distinct and reliable measure of model performance, we approached the testing phase with meticulous planning, simulating real-world conditions as closely as possible. While most of the plastics were visually similar, different plastics emerged in consecutive bales. In response, we designed our Test set (Figure 22) by isolating a completely independent bale that we manually sorted, ensuring no overlap with the training set. The model had no prior exposure to this data, allowing us to test its true generalization capabilities. One challenge we faced was the uneven distribution of types within this bale. Nonetheless, we decided to include one representative image per object, ensuring a diverse yet distinct Test dataset. As we move forward, we will continue to augment this dataset by following the same systematic procedure, providing a continually expanding and unbiased benchmark. The pie chart (Figure 22) below illustrates the current data ratio for each plastic type in this Test set.

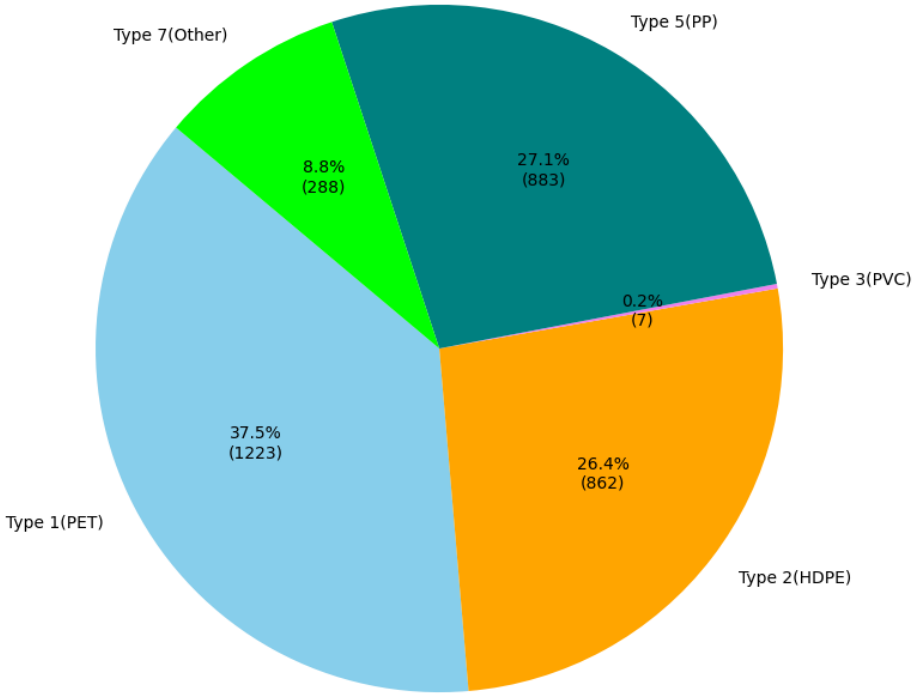


Figure 22. *Test Dataset distribution for 3264 images of 5 types of plastics*

6.1 Model Analysis

This section serves as a comprehensive benchmark for all the trained models. We will explore their Grad-CAM visualizations and evaluate their performance on a specially curated Test dataset.

We evaluate our models on three measurements: Accuracy, Recall, and F1 score. Accuracy is the ratio of correct observation to total observation. Accuracy is a good measurement for all types except for type 3 as it is lower than the other types. Recall measures the percentage of true positives over the sum of true positives and false negatives. Recall helps us identify the criticality of false positives. And, finally, the F1-score is a single evaluation point that suggests the harmony between precision and recall to understand the effect of our imbalance of the dataset.

Table 3. *Accuracy, F1 score, and Recall for each trained classifier on plastic waste data*

Model Name	Accuracy	F1 Score (weighted)	Recall
ResNet-18	79.93 %	80	85.07
ResNet-34	82.93%	83	82.94
ResNet-50 (R1)	83.79%	84	83.9
ResNet-50 (R2)	87.19%	87	87.19
EfficientNetv2m	73.19	59	59.65

Accuracy *= (Correct Predictions/Total Predictions)×100*

Recall *= True Positives/ (False Negatives+ True Positives)*

F1-Score *= 2× (Precision×Recall/ Precision+Recall)*

While the final ResNet-50 model demonstrates satisfactory accuracy and metrics on the test data, this alone doesn't provide a complete picture. It's equally important to gain deeper insights into the features within the training dataset and those extracted by the different models. Metrics like accuracy offer only a surface-level understanding of the model's performance, but analyzing the learned features allows us to understand what the model is capturing and how it distinguishes between classes. By visualizing feature maps and embeddings, we can examine the intermediate representations the network builds layer by layer. This helps ensure the model isn't overfitting, or worse, learning spurious correlations. Exploring the data in this way often reveals patterns, outliers, or biases that might not be apparent from high-level metrics. Understanding the underlying feature space is key to building more robust, generalizable models.

To better understand model performance, we analyze confusion matrices for each classifier. These matrices provide a detailed view of which classes are being misclassified and, importantly, to which classes. Misclassifications primarily arise due to two factors: the absence of specific plastic in the training set, or severe deformation of the plastic that alters its recognizable features. We investigated these misclassified images through Grad-CAMS. This granular insight allows us to identify patterns in misclassification and offers clues on how to fine-tune our models further. One of the most striking observations is the frequent misclassification of all types into Type 5. This pattern likely stems from two factors: the large dataset size for Type 5, which biases the model toward overfitting this class, and the visual similarities between Type 5 and other types, making

differentiation more challenging. Following closely in misclassification frequency is Type 1, likely for similar reasons. This points toward an important realization: focusing solely on increasing dataset size for particular types might not yield optimal results. Instead, we must emphasize data variability among all classes. The diversity of the dataset, in terms of texture, lighting conditions, and object presentation, could play a far more crucial role in eliminating these errors than sheer volume alone. By enriching the variability within each class, particularly for underrepresented or visually ambiguous types, we could significantly improve the model's capacity to generalize across unseen examples.

An intriguing potential avenue for addressing these misclassifications is incorporating a Siamese network within the existing architecture. By leveraging the power of contrastive learning, this addition could assist the model in distinguishing between classes that are visually similar, such as Type 5 and others. While this remains speculative, it could be a valuable experiment to further explore, especially given the potential for reducing misclassifications. However, even with such architectural innovations, the issue of data volume remains pertinent. Industrial-level deployment will require not only greater data diversity but also larger datasets to ensure robustness in edge cases. Data collection, particularly for the most misclassified types, will be essential as we continue to push toward a fully reliable industrial implementation.

Below, confusion matrices from each of the trained models illustrate these trends in misclassification.

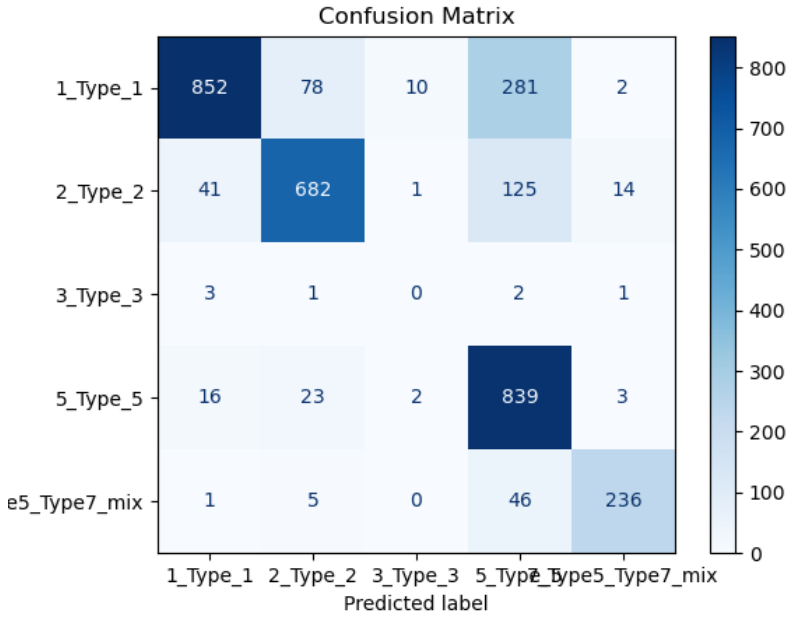


Figure 23. *Confusion matrix for ResNet-18*

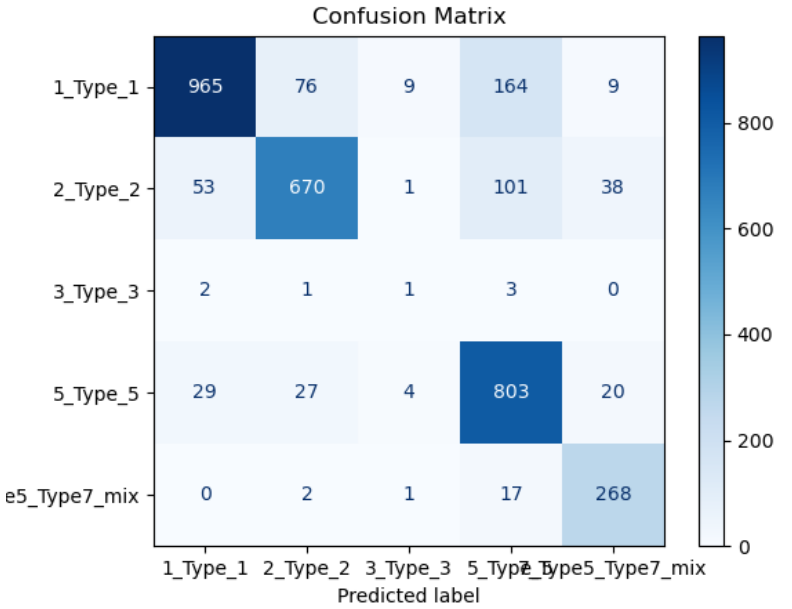


Figure 24. *Confusion matrix for ResNet-34*

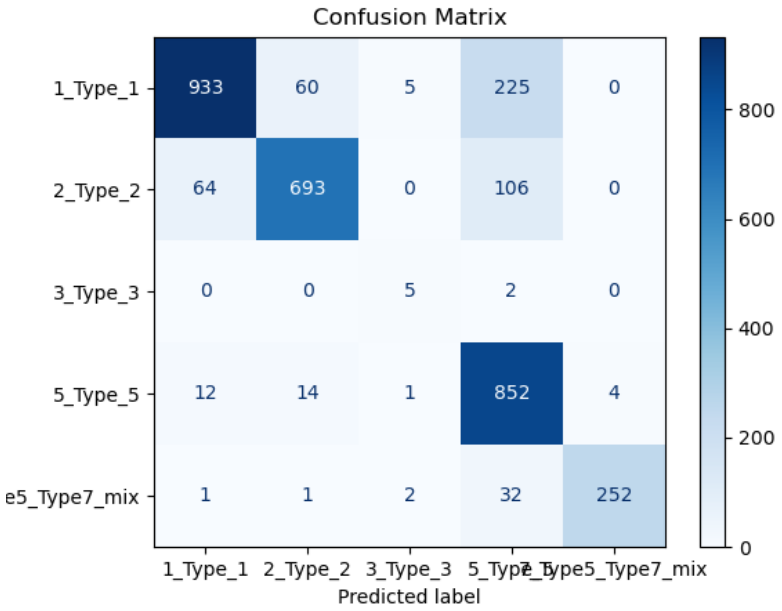


Figure 25**.** *Confusion matrix for ResNet-50(R1)*

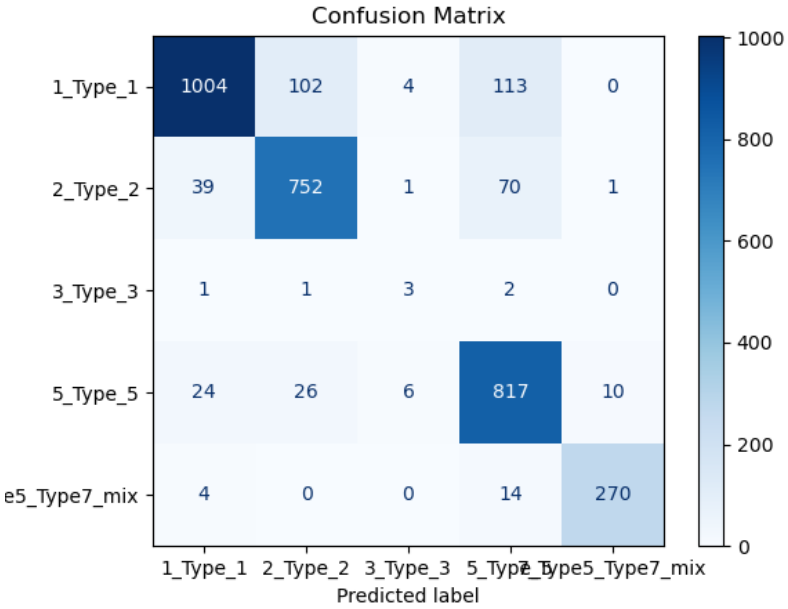


Figure 26. *Confusion matrix for ResNet-50(R2)*

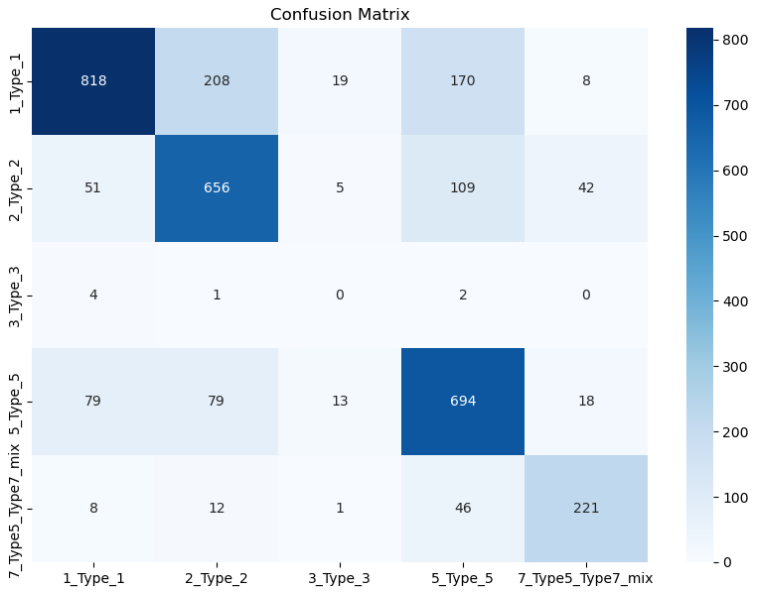


Figure 27. *Confusion matrix for EfficientNetv2-m*

We have chosen t-SNE (t- Distributed Stochastic Neighbor) (Maaten & Hinton, 2008) to extract feature maps and visualize the separability of different plastic classes, t-SNE is a powerful dimension reduction technique with which we can visualize high-dimensional datasets in lower dimensions while preserving data structure. It converts Euclidean distances into conditional probabilities in the high-dimensional space, where adjacent points have more similarities. Next, using Kullback-Leibler divergence, the algorithm produces a similar distribution of probabilities in the low-dimensional space and minimizes the difference between the two. This reduces the clustering of distant points and guarantees that all points within high dimensions stay close after projection. t-SNE is a popular tool for the visualization of huge datasets. It is particularly good at visualizing neural network features and word embeddings and exposing the underlying structure of complex data, such as clusters and manifolds. Its' power resides in revealing patterns that high-dimensional space might otherwise obscure.

Each point in the t-SNE graph corresponds to a sample, colored based on its class, purple for Type 1, dark blue for Type 2, teal for Type 3, green for Type 5, and yellow for Type 7. At first glance at Figure 28, we can observe that the model has reasonably separated Type 5 and Type 7. Though there is some slight overlap between the green and yellow clusters, most points are well-delineated, suggesting that the model has effectively captured the distinguishing features of these two types, which indicates that the model can correctly identify and discriminate between Type 5

and Type 7 with minimal confusion which reflects the accuracy type 5 having 95.2% and Type 7 81.94%. On the other hand, Types 1, 2, and 3 show significant overlap in the t-SNE space. The purple, dark blue, and teal points are interspersed across a common region, signaling that the features extracted for these classes are not sufficiently distinct. This overlap is a likely source of misclassifications, as the model struggles to separate the inherent features of these types, possibly due to insufficient or non-discriminative patterns in the dataset.

If we compare the t-SNE projections of ResNet-18 (Figure 28) and ResNet-34 (Figure 29), the differences in feature space separability become evident. Like ResNet-18, ResNet-34 successfully distinguishes between Type 5 and Type 7, maintaining a clear separation in these classes, which suggests both models have learned strong, distinguishing features for these types. However, while ResNet-18 struggles with a significant overlap among Types 1, 2, and 3, ResNet-34 offers a slight improvement in separating these classes. In particular, the separation of Type 3 in the feature space is more pronounced in ResNet-34, indicating that the model has learned more discriminative features for this type than ResNet-18. This improvement likely reflects the deeper architecture of ResNet-34, which provides greater capacity for feature extraction and generalization. Nonetheless, the feature representations for these classes remain entangled, preventing the model from achieving perfect class separability. This corresponds with the small gain in accuracy, ResNet-34 achieves just a 3% improvement over ResNet-18, reinforcing that while deeper architectures can offer better feature extraction, their impact on overall performance may be marginal. Ultimately, the t-SNE graph provides an intuitive visual explanation for why ResNet-34 slightly outperforms ResNet-18 in accuracy: better feature separation, particularly for Type 3, but not enough to drastically improve the performance across all classes.

The t-SNE projection for ResNet-50(R2) (Figure 30) reveals a striking clarity in how the model has organized the data points, each class is distinctly clustered, far more pronounced than in the earlier t-SNE visualizations of ResNet-18 and ResNet-34. The separation between the different types is now much more visible, showcasing ResNet-50's ability to learn and differentiate the fine-grained features that define each plastic type. In comparison, while ResNet-18 and ResNet-34 both demonstrated some level of class separation, their clusters were not as clearly delineated. ResNet-34, while improving the separation of certain classes like Type 3, still exhibited overlap, suggesting that the models were not fully exploiting the feature space. ResNet-50, with its deeper architecture and increased capacity, organizes the feature space more effectively,

forming tighter, more isolated clusters. This indicates that ResNet-50 is capable of capturing more subtle nuances in the current dataset, leading to better class distinction. However, the t-SNE visualization highlights areas of improvement, classes that are not yet perfectly separated can benefit from further curation. This could mean adding more representative samples, ensuring a better distribution of features, or refining the pre-processing pipeline to allow the model to extract more meaningful patterns.

Ultimately, this graph underscores that ResNet-50 is likely the optimal model for our task with its clear separation of feature clusters, it has demonstrated superior accuracy not only in theory but in real data. However, we need to conclude that by improving data curation and increasing the volume of training samples, this model's performance could be pushed even further, yielding an even more robust classification.

In examining the t-SNE visualization for EfficientNetV2-M (Figure 31), we observe an unexpected outcome: despite achieving an accuracy of 73.19%, the model fails to separate the classes. This highlights an important limitation of t-SNE. While it can be a useful tool for visualizing high-dimensional data, it doesn't always capture the nuances of model predictions. By compressing high-dimensional features into two dimensions, t-SNE can obscure the true separability of classes in the original feature space. As seen in the graph, class clusters are scattered with no clear distinction, suggesting that EfficientNetV2-M struggled to differentiate between them. This discrepancy between the model's accuracy and the t-SNE plot reminds us that high-dimensional relationships can be lost in such projections. As a result, we are led to explore other methods for visualizing and understanding feature representations more effectively.

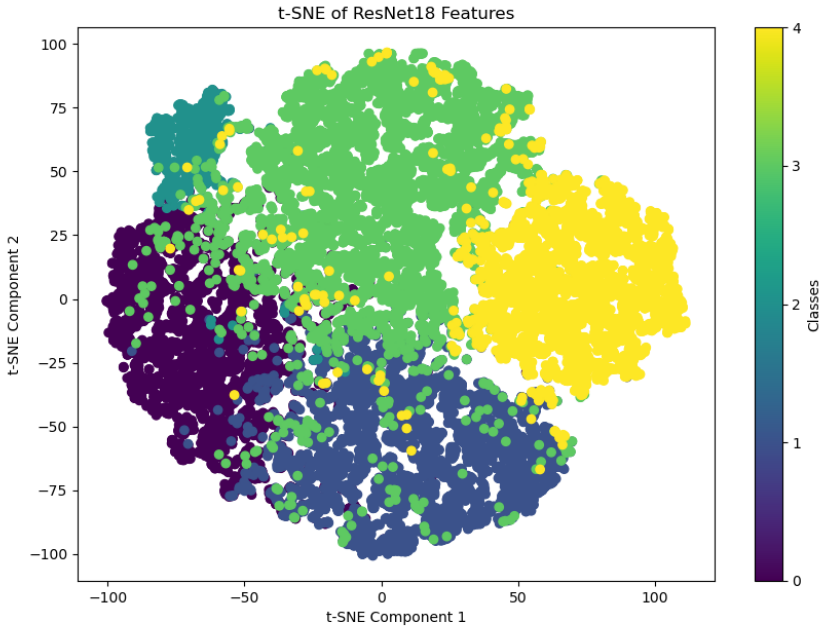


Figure 28**.** *t-SNE features against ResNet-18 on Training Dataset*

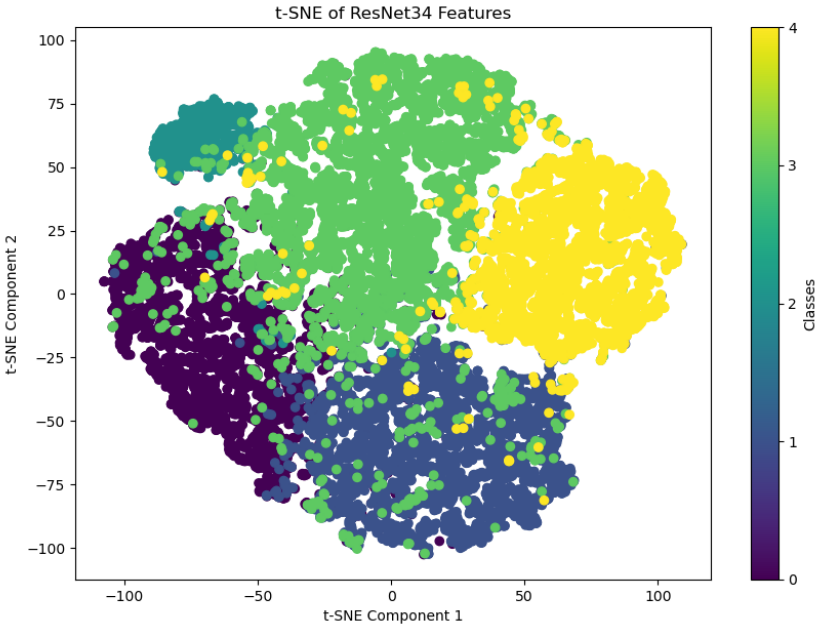


Figure 29. *t-SNE features against ResNet-34 on Training Dataset*

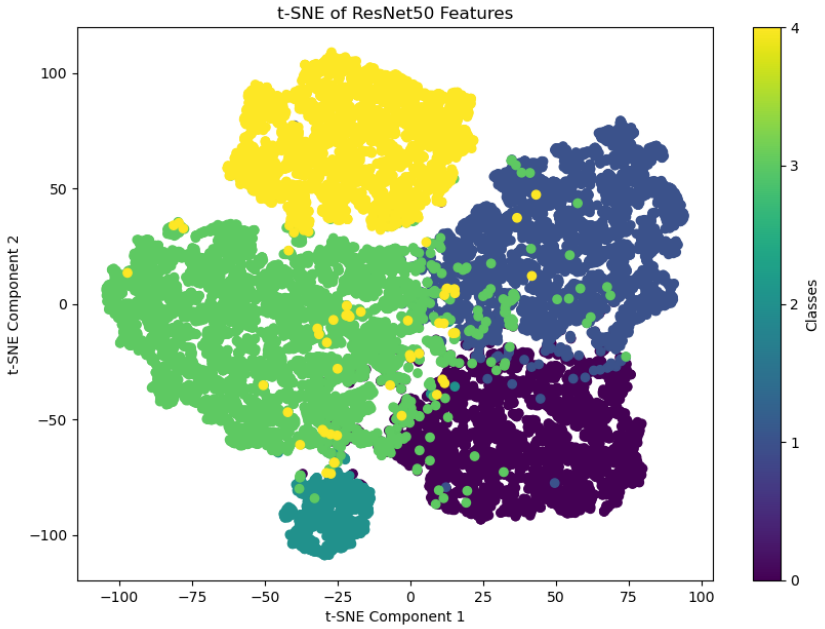


Figure 30. *t-SNE features against ResNet-50 on Training Dataset*

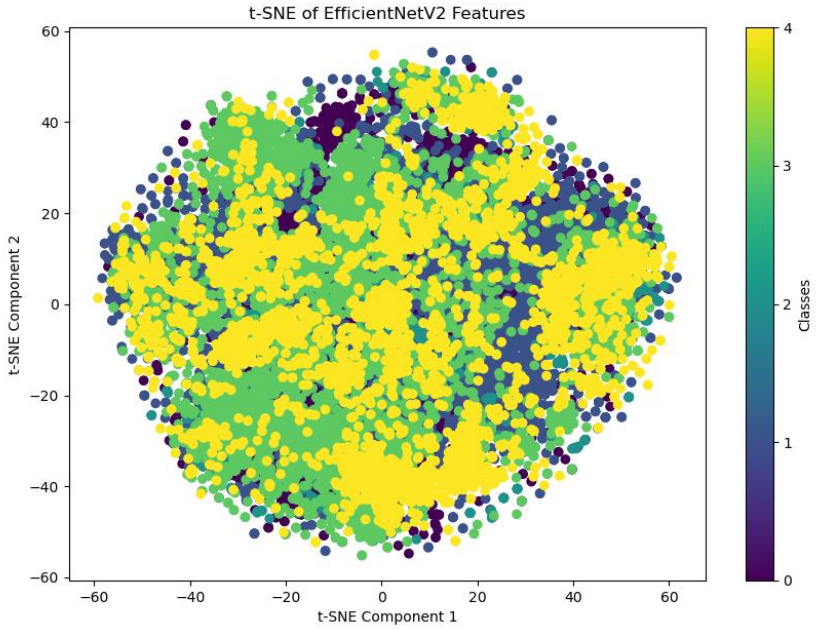


Figure 31. *t-SNE features against EfficientNetv2-m on Training Dataset*

Gradient weight Class Activation Map (Grad-CAM) (Selvaraju, et al., 2016) is a visual explanation technique for CNNs that utilizes the gradient weights of a certain class. First, it highlights the regions in the input image that are most important for the model's decision. Then it starts by performing a forward pass to make predictions and then computes the gradients of the class score concerning the feature maps of a convolutional layer. These gradients are pooled to determine the importance of each feature map, and then a weighted sum of the feature maps generates a heatmap. This heatmap overlaid on the original image shows where the model is focusing to make its prediction, providing a visual explanation of the decision-making process. As Grad-CAM only provides the heatmap for one class we can only understand the perspective of that class's weights on our desired image. But even then, it's convenient for us to understand why the model is focusing on the parts of the image and why it misclassifies. Also, Grad-CAM can help us understand which convolutional layer has more effect on the input and help us determine if we need to modify our input images.

We visualize the Grad-CAM heatmaps (Figure 32, Figure 33, Figure 34, Figure 35) for the last blocks of each layer across different models on an object that is classified as Type 2, aiming to provide a clear interpretation of how attention shifts within the network. For each block, we examine the final layer of that architecture, revealing how the model's attention develops as we move deeper. Each row corresponds to a specific plastic type, while each column represents the last layer of the respective blocks. Starting with ResNet-18 (Figure 32), which contains 4 blocks, we observe the model's focus evolving through the network. In the early convolutional layers (block 1, block 2), the attention is scattered and diffuse, primarily focused on low-level features: edges, textures, and broad patterns. The highlighted regions are distributed across the image but in relatively small amounts, indicating that the model is still capturing basic visual cues. As we move into the deeper layers (block 3, block 4), a marked transition occurs. The attention becomes concentrated on semantically important regions, specifically, key features like the logo or distinct details of the plastic that a human eye would instinctively recognize. In the case of this Type 2 plastic, we observe strong, focused activations in the final layer for Type 2 Grad-CAM, while Type 1 still receives some attention due to overlapping features or uncertainties in classification.

When comparing across architectures from ResNet-18 to ResNet-34 (Figure 33) and ResNet-50(R2) (Figure 34) there is a significant shift in both the depth and specificity of the attention maps. While the early layers (blocks 1 and 2) remain relatively similar across models,

the deeper layers (blocks 3 and 4) show substantial improvements in ResNet-50, which sharpens the model's focus on critical regions. ResNet-50 demonstrates a clearer, more precise localization of important features, allowing it to better capture the exact regions that matter for classification.

In contrast, the Grad-CAM (Figure 35) visualization for EfficientNetv2-m presents a fundamentally different heatmap in its final layer, which aligns with the distinct architectural differences from the ResNet family. EfficientNet's attention patterns reflect its unique capacity to interpret and localize features in a different manner, showcasing how architecture choice profoundly influences what the model learns to prioritize.

To weed out misclassification we focused primarily on the Grad-CAM visualizations for misclassified images, aiming to uncover the reasons behind these misclassifications. Through careful analysis, we identified that misclassification often occurred due to the presence of portions of conveyor belt, and reflected lights in the images, which introduced unnecessary background noise into the model's decision process. Based on these insights, we adjusted our preprocessing pipeline by cropping the images more effectively leaving a 15-pixel margin of conveyor around the object while reducing the extraneous background. This new cropping strategy resulted in a notable improvement across the dataset, reducing noise and helping the model focus on the essential features of the objects. Additionally, we removed outliers that were contributing to misclassifications, further refining the dataset. For our project, where the images differ significantly from conventional datasets used in training, Grad-CAM proved to be an invaluable tool. It not only highlighted the model's blind spots but also guided us in making precise adjustments to the data. By leveraging Grad-CAM, we were able to mitigate substantial misclassification issues and significantly enhance the data preparation process, leading to better overall performance in our training pipeline.

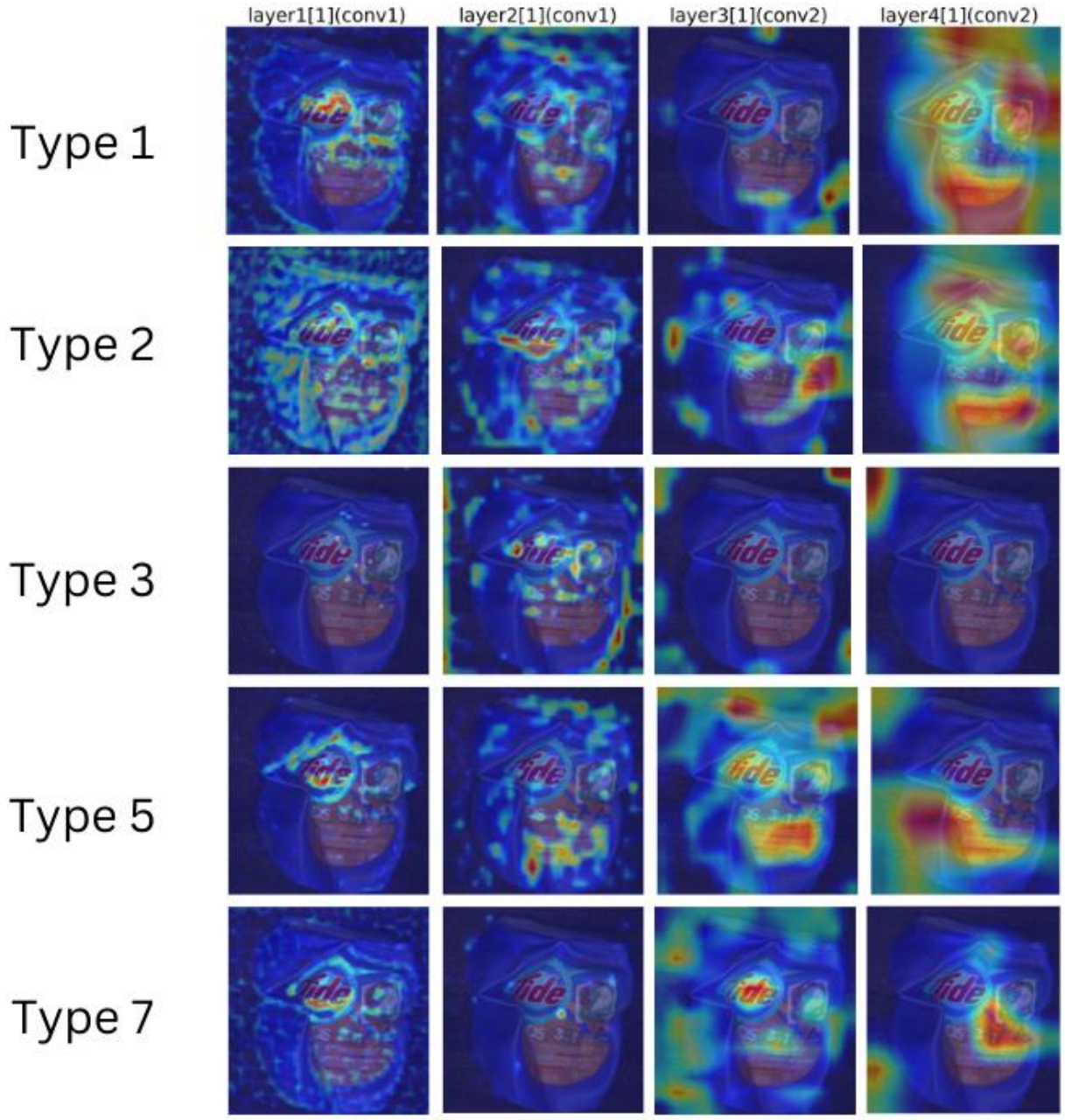


Figure 32**.** *Grad-CAM of Type 2 object against ResNet-18, rows represent type specific gradients and column represents layer based heatmap on based on the specific type*

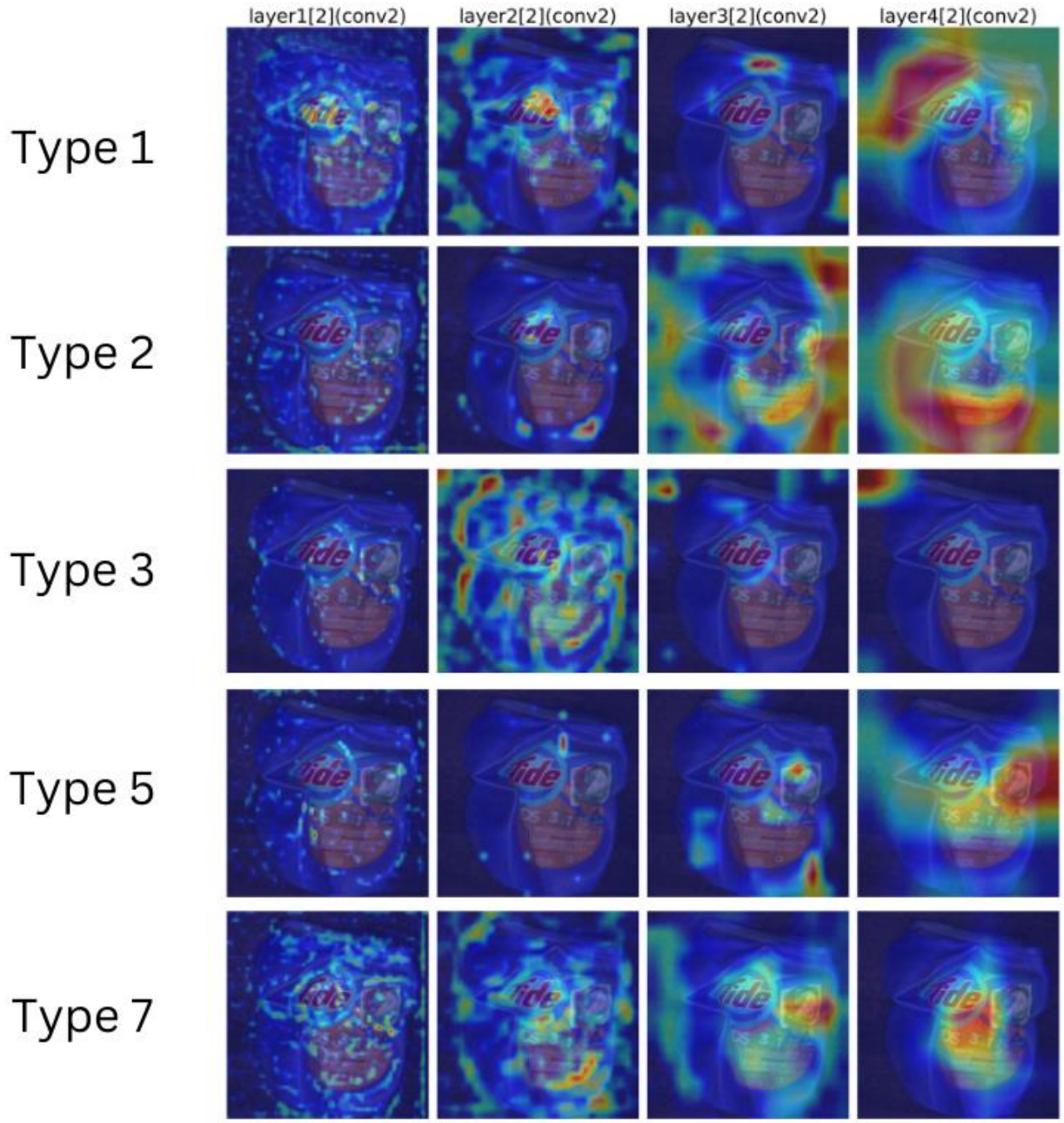


Figure 33**.** *Grad-CAM of Type 2 object against ResNet-34, rows represent type specific gradients and column represents layer based heatmap on based on the specific type*

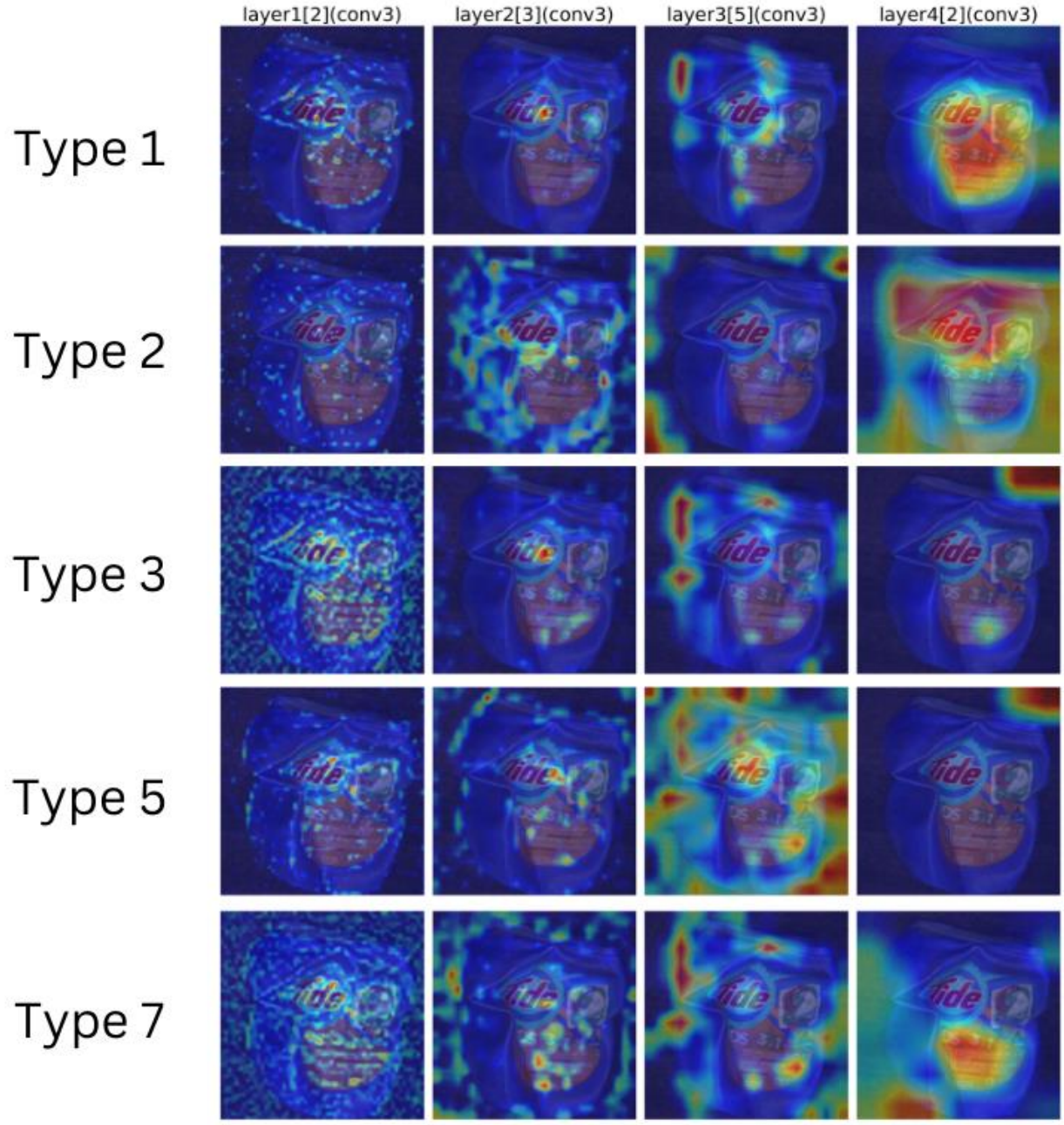


Figure 34. *Grad-CAM of Type 2 object against ResNet-50, rows represent type specific gradients and column represents layer based heatmap on based on the specific type*

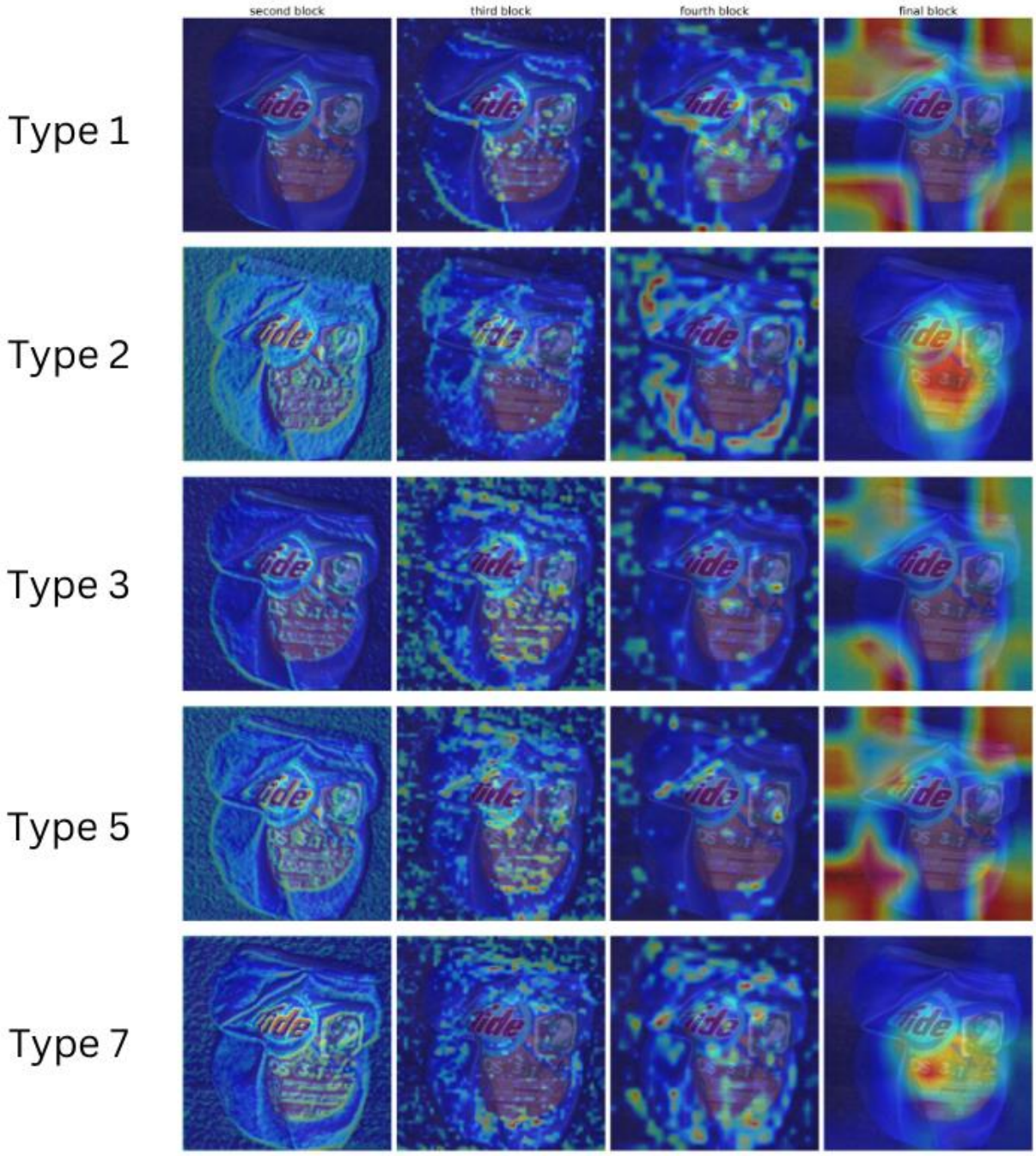


Figure 35.*Grad-CAM of Type 2 object against EfficientNetv2-m, rows represent type specific gradients and column represents layer based heatmap on based on the specific type*

As outlined in the last subsection of methodology, we trained three models on progressively larger subsets of our Training dataset to understand how data size and variability influence accuracy, particularly for deformed objects. The first model, trained on 50% of the available data, achieved an accuracy of 51% for Type 7 plastics from the Test dataset which is 288 images. At this stage, we observed that 161 images were predicted with a confidence level below 50%, highlighting substantial uncertainty in the model's predictions. We then scaled the training to 75% of the dataset, where the model's performance saw a notable improvement. Accuracy rose to 89.93%, and the number of images predicted with confidence below 50% dropped significantly to just 34. This suggests that with additional data, the model was able to learn more reliable features for Type 7, improving both its precision and confidence. Finally, when training the model with the entire dataset, accuracy increased further to 93.75%. The number of low-confidence predictions (below 50%) fell to just 23 images. To better understand these edge cases, we isolated the images predicted with low confidence and compared how the model's confidence levels shifted between the full dataset and the 75% dataset. This analysis provides insights into how data augmentation and variability affect model robustness, particularly for challenging or deformed samples.

The model trained on 50% dataset presents a markedly different behavior compared to the models trained on 75% and 100% of the data. In the scatter plot of the 50%-trained model (Figure 36), we observe a broad distribution of points across the confidence spectrum, with a roughly equal number of predictions falling below the 50% confidence threshold. This scattered arrangement is a strong indicator of a model struggling to generalize, particularly in the distinct test set we've reserved for evaluation. Simply put, this model lacks the certainty required to make reliable predictions. A similar trend appears in the histogram (Figure 37) too with confidence on the X-axis and frequency on the Y-axis, we see a nearly uniform distribution, where the count of poorly predicted objects, those with confidence below 50% is alarmingly high. This reinforces the idea that the limited dataset hampers the model's capacity to develop robust representations, leading to weak predictions. However, a marked improvement is seen when we examine the model trained on 75% of the dataset. Both the scatter plot (Figure 38) and the histogram (Figure 39) reflect this shift. In the scatter plot, predictions cluster near the upper end of the confidence spectrum. The number of low-confidence predictions drops drastically, suggesting that the model has learned far better from the additional data and has a better understanding of the objects beyond the deformations. The histogram further confirms this, with a significant reduction in the number of

predictions falling below 50% confidence. The error rate, as measured by the count of predictions below this threshold, drops from 161 to 34, a reduction of nearly 80%. This improvement, while largely attributable to the larger dataset, underscores the impact of more comprehensive training data on both confidence and accuracy.

50% dataset model Error Rate = 161/288=0.559,
75% dataset model Error Rate = 34/288=0.118
***Error Rate Reduction** = ((50% dataset model Error Rate - 75% dataset model Error Rate) / 50% dataset model Error Rate) * 100 = ((0.55-0.11)/0.55)*100 = 80%*

With an increase from 2221 to 3331 images we can calculate the misclassification rate per sample,
*50% dataset model Misclassification Rate= (50% dataset model misclassifications/50% dataset size) * 100= (161/2221) * 100 = 7.25%*
*75% dataset model Misclassification Rate= (75% dataset model Misclassifications/75% dataset size) * 100= (34/3331) * 100 = 1.02%*

We also can calculate the effectiveness of the dataset increase from 50% to 75%,
***Misclassification Improvement per Additional data** = (50% dataset model misclassification-75% dataset model misclassification)/(50% dataset size-75% dataset size)= (161-34)/(3331-2221)=0.114*

*This concludes that for every additional sample in the dataset, roughly **0.114** misclassifications occurred.*

As we turn our attention to the model trained on the full dataset, the gains, though present, are less dramatic compared to the leap from the 50% to 75% dataset. The scatter plot (Figure 40) reveals that while overall confidence levels have risen, the improvements are more subtle. Objects that were previously predicted with low confidence, around 10% now see their predictions rise to 37%, showing measurable progress, however, this increase is not universal. For some objects, confidence remains unchanged, pointing to patterns that are not yet fully captured by the model.

This suggests that while the additional data has improved the model's performance, particularly for certain difficult cases, we may be approaching the limits of what raw data can provide in terms of accuracy gains. The subtle shifts in confidence lead us to investigate these objects further through Grad-CAMs, where we can explore how the model focuses its attention and whether it is learning the correct visual features for classification. The ratio calculation comparing the performance of the model trained on 75% of the dataset versus the full dataset is shown below. This helps quantify the incremental gains and provides a clearer picture of where the model is excelling and where further refinement may be necessary.

75% dataset model Error Rate = 0.118
Full-dataset model Error Rate= 23/288=0.0798
Error Rate Reduction*= ((75% dataset model Error Rate – Full-dataset model Error Rate) / 75% dataset model Error Rate) *100= ((0.11-0.079)/0.11)*100= 28%*

With an increase from 3331 to 4441 images we can calculate the misclassification rate per sample,
75% dataset model Misclassification Rate= 1.02%
*Full-dataset model Misclassification Rate= Full-dataset model misclassifications/Full-dataset size= (23/4441) * 100 = 0.518%*

We also can calculate the effectiveness of the dataset increase from 75% to 100%,
Misclassification Improvement per Additional data *= (75% dataset model misclassification - Full-dataset model misclassification)/(75% dataset size - Full-dataset size)= (34-23)/(4441-3331)=* ***0.0099***

From the misclassification improvements we observe- 0.114 for the 50% to 75% dataset and 0.0099 for the 75% to 100% dataset we can draw several key conclusions:

I. **Diminishing Return-** The sharp drop in misclassification improvement from 0.114 to 0.0099 signals the onset of diminishing returns as we increase the dataset size. At first, moving from 50% to 75% of the data, we observe a significant reduction in errors—additional data provides the model with valuable diversity, allowing it to generalize better and make more confident predictions. But when we extend the dataset further, from 75% to

100%, the incremental improvement flattens for deformed objects. The model's capacity to benefit from more data diminishes.

II. **Data quantity vs quality**- As we've discussed before, the importance of the dataset often surpasses the choice of model architecture, and this principle is particularly evident in our case. The model's limitations aren't rooted in its design but rather in the nature of the data it has been exposed to. This brings us to a critical consideration: the balance between data quantity and data quality. At this stage, more data of the same kind is unlikely to yield meaningful improvements as observed by the calculations. Instead, the focus should shift toward data diversity, increasing the variability in the dataset, especially by incorporating new types of deformations and edge cases, we can have a far greater impact than simply scaling up the existing dataset. For example, deliberately creating deformed data by physically crushing objects after capturing their pristine images could simulate real-world scenarios more effectively. This would allow the model to learn robust representations of deformations without requiring any changes to its underlying parameters.

III. **Performance Bound-** While it is true that machine learning models often exhibit diminishing returns as data volumes increase at a certain point, our project still benefits from a significant learning curve. Despite the typical saturation seen in many models, we observe that each additional increment of data continues to refine performance; albeit more gradually. In this case, we can confidently assert that adding a certain volume of this specific type of dataset would allow us to establish a predictable boundary. This boundary ensures that the model consistently performs within a reliable range of accuracy, even as data diversity and complexity increase. By reaching this bound, we move toward a system where predictions are not only stable but maintain a level of confidence that meets our threshold.

IV. **Cost-effectiveness of data collection-** If we consider the resources required for collecting and processing additional data, the marginal improvement of 0.0099 from 75% to 100% of the dataset suggests a diminishing return on investment. Beyond this point, the cost and effort involved in gathering more data may not justify the limited gains in performance. In machine learning, we often encounter a trade-off between the quantity of data and the resources available. As we approach this saturation threshold, it becomes clear that adding more of the same kind of data is no longer the most efficient path forward. The benefit is

outweighed by the time, energy, and financial cost of data collection. This is especially true in our case, where the process of collecting plastics from municipal solid waste, followed by sorting, is both time-consuming and poses health risks. The effort required to expand the dataset is not trivial, it involves manual labor, potential exposure to hazardous materials, and significant logistical challenges.

When examining the Grad-CAM visualizations of misclassified deformed plastics, a clear pattern emerges, signaling both the strengths and limitations of the models trained with different dataset sizes. Two specific examples of misclassified objects, which consistently is misclassified by all models, reveal crucial insights into the model's capacity to generalize across deformed objects. The model trained with 75% of the dataset demonstrates strikingly low confidence (10%) on Type 7, while the full-dataset model improves this to 37%, though both remain below 50% confidence. Figure 42 shows that much of the heatmap's focus is on Type 1, with some spillover into Type 3 and eventually Type 7. Critically, this attention is concentrated in a small region rather than distributed across the full object, indicating that the model has not generalized well to this deformed plastic. The activation is sparse and narrow, pointing to an incomplete understanding of the features that define Type 7 plastics. In contrast, Figure 43 presents a more nuanced case. Here, the full-dataset model, despite achieving better generalization, misclassifies the object as Type 5. Interestingly, the heatmap spans the entire object, signifying that the model has learned a broader, more holistic representation of the object's features compared to the 75% dataset model. Yet, this broader activation also reveals areas where the model struggles: there are extraneous patches of activation outside the object itself, indicating that the model is picking up on irrelevant details in the background. This suggests that while generalization has improved, the model is still sensitive to noise and distractions that interfere with its ability to focus solely on the object.

These Grad-CAM visualizations provide a window into the models' behavior, but they are just one piece of the puzzle. The presence of these scattered heatmap patches signals the need for cleaner data and more focused attention mechanisms to ensure the model does not rely on spurious correlations. The key takeaway is that while the model has made strides in learning from the full dataset, its misclassifications, particularly for deformed objects suggest it has reached a boundary where architectural adjustments may be necessary. We need to explore more robust methods for guiding the model's attention to the relevant features of deformed objects, mitigating misclassifications by refining both the data and model architecture.

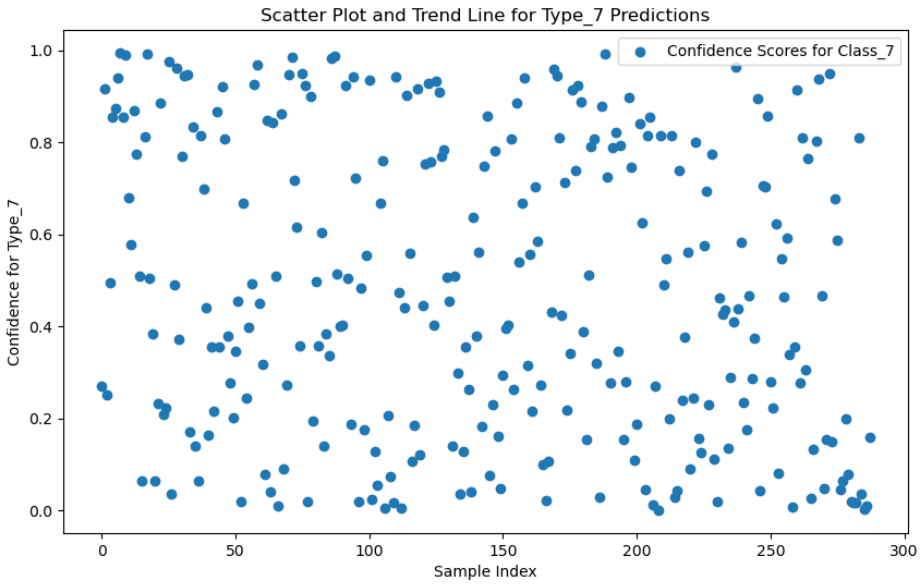


Figure 36**.** *Scatter plot of confidence for ResNet50 trained on 50% Type 7 Training data*

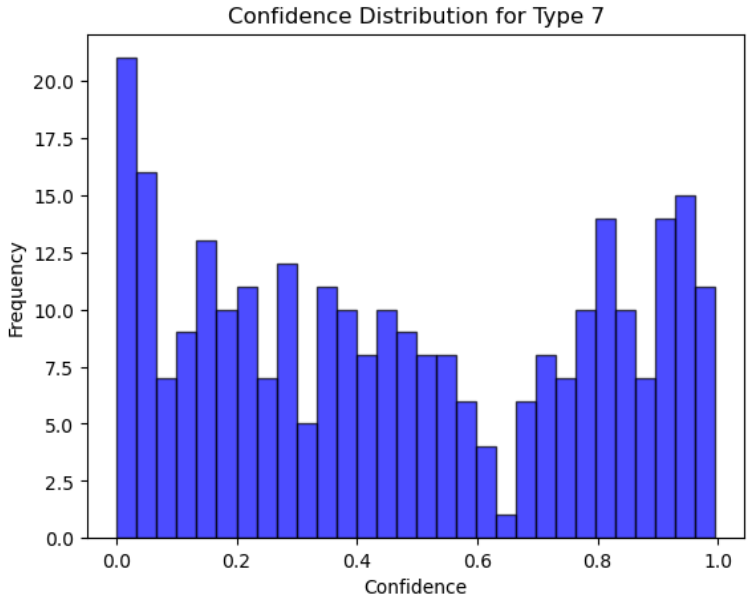


Figure 37. *Histogram of confidence for ResNet50 trained on 50% Type 7 Training data*

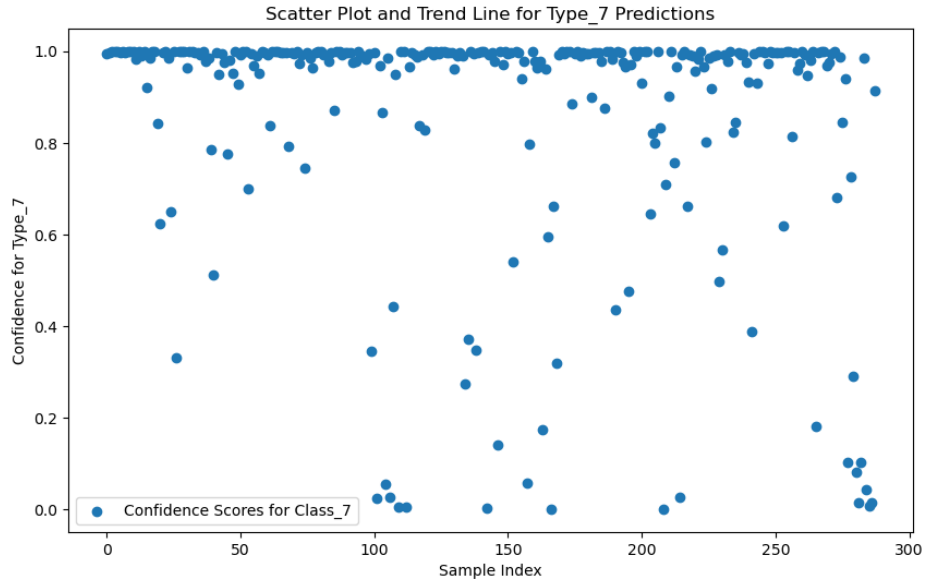


Figure 38. *Scatter plot for ResNet50 trained on 75% Type 7 Training data*

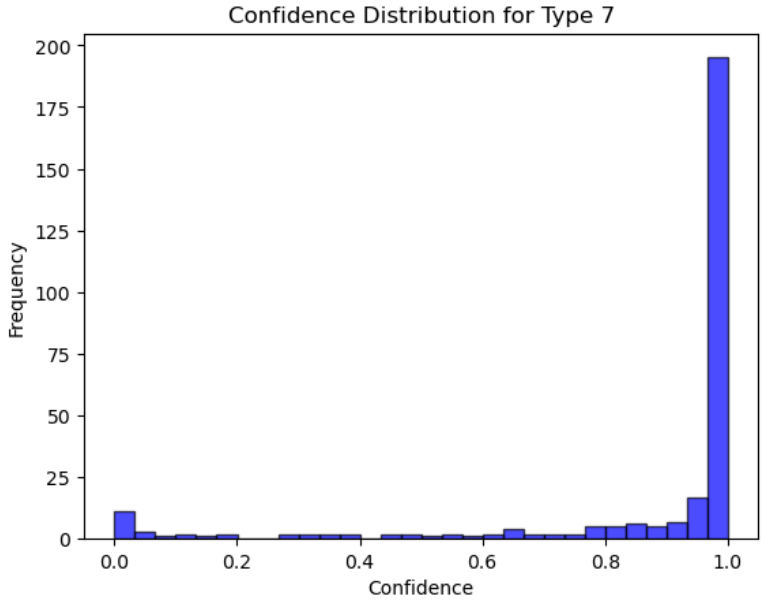


Figure 39. *Histogram of confidence for ResNet50 trained on 75% Type 7 Training data*

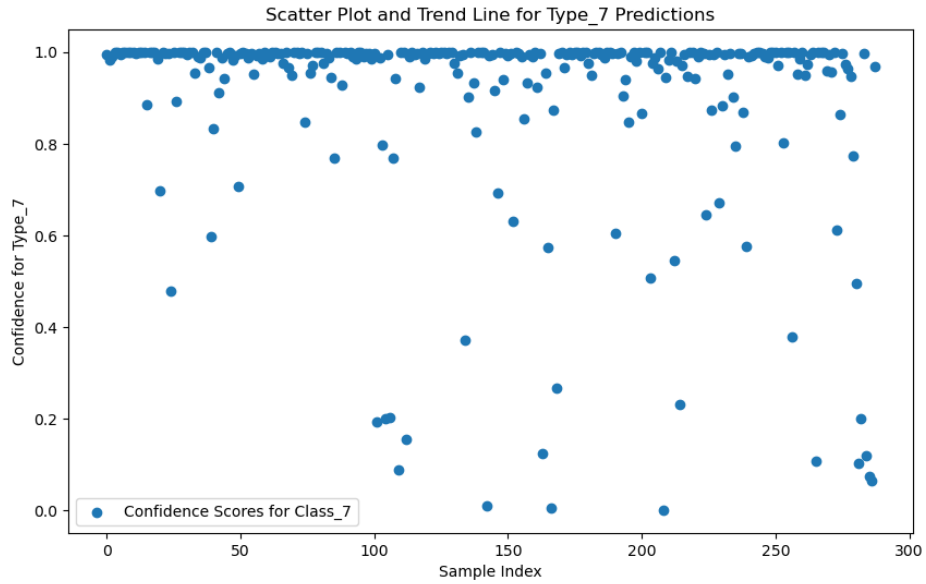


Figure 40**.** *Scatter plot for ResNet50 trained on 100% Type 7 Training data*

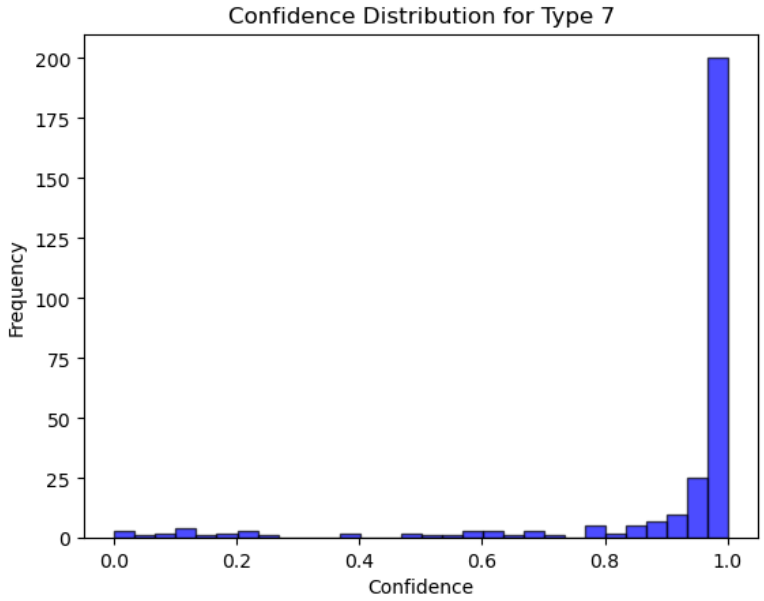


Figure 41. *Histogram of confidence for ResNet50 trained on 100% Type 7 Training data*

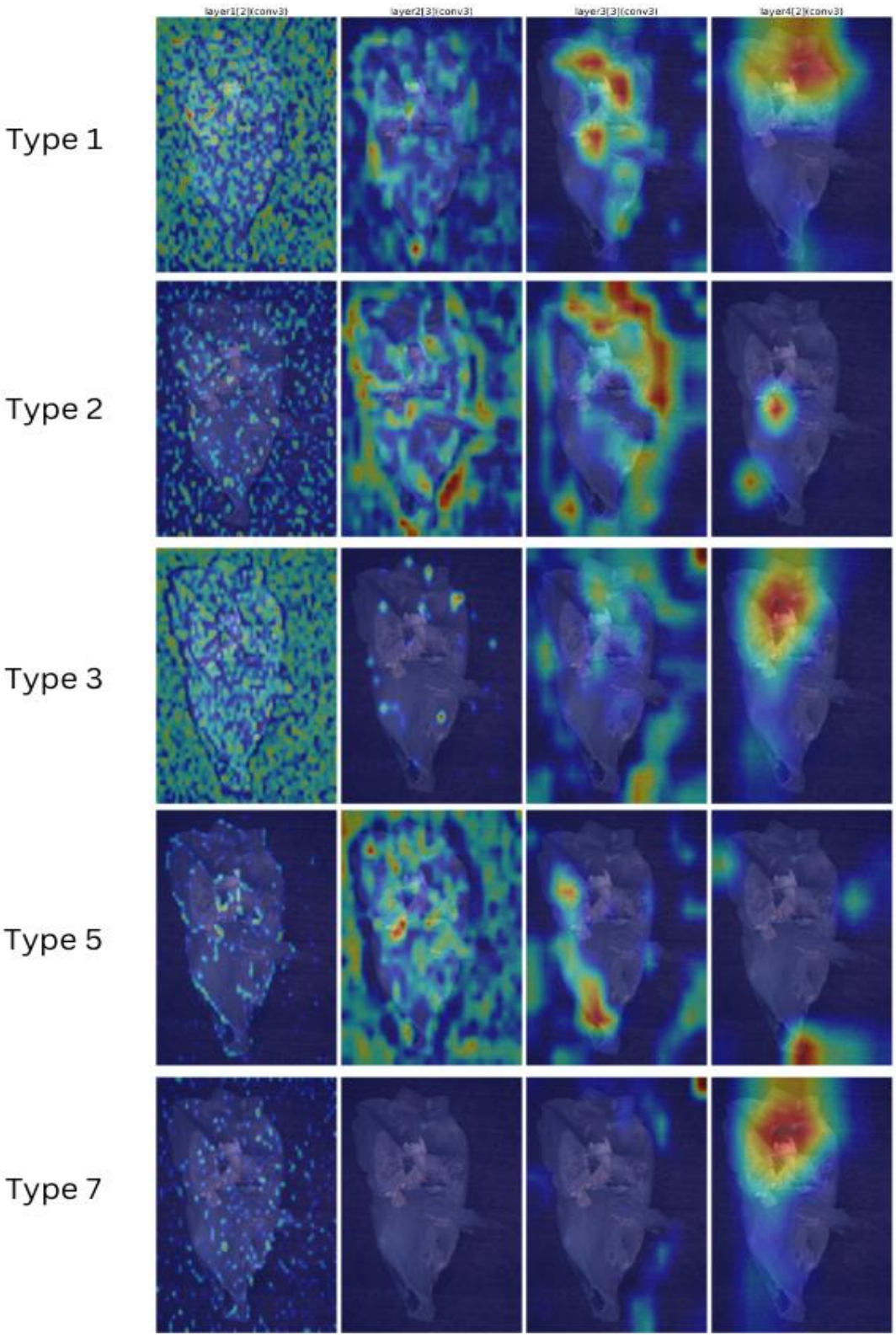


Figure 42**.** *Grad-CAM of misclassified object for model trained on 75% Training dataset, rows represent type specific gradients and column represents layer based heatmap on based on the specific type*

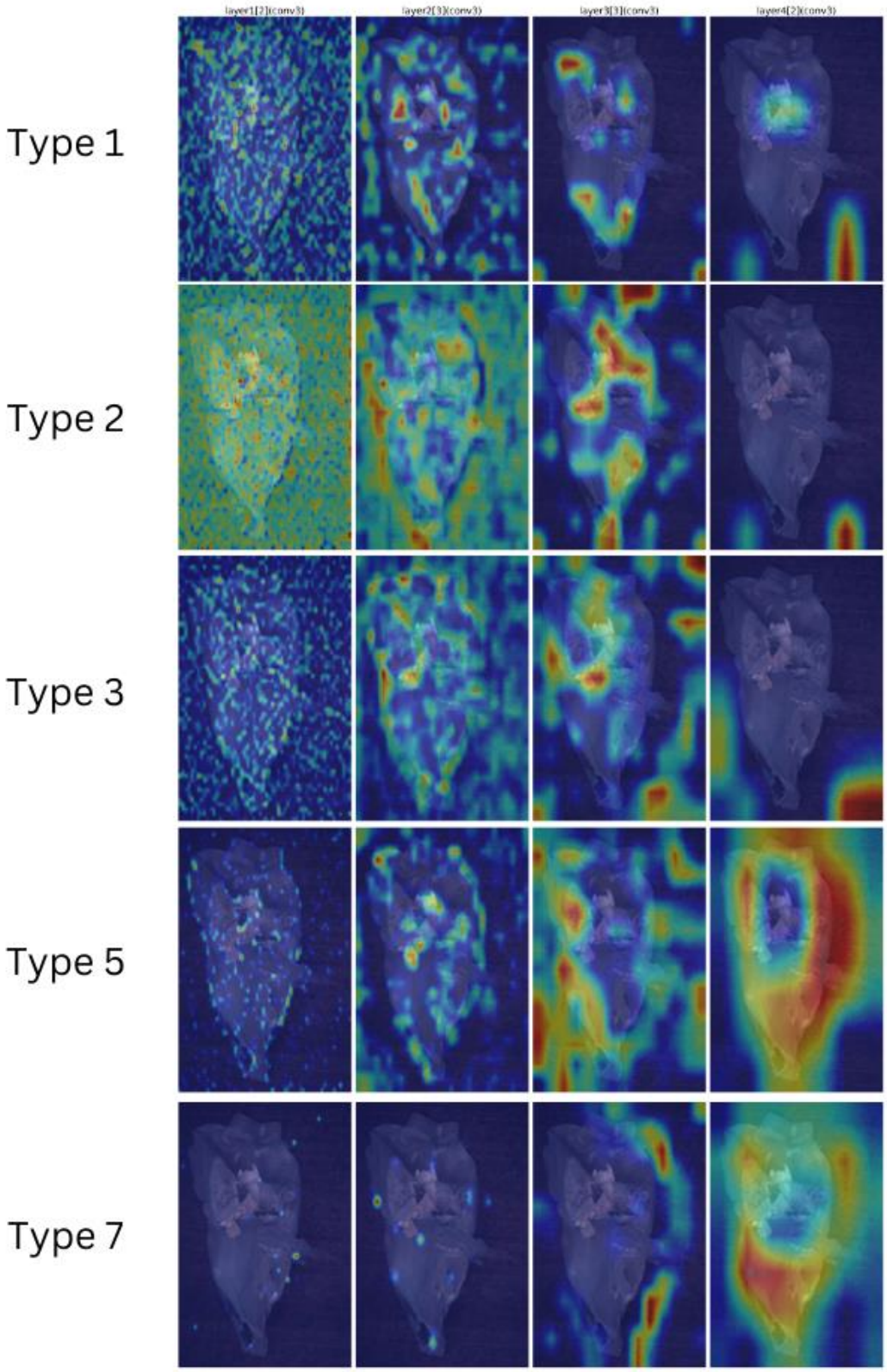


Figure 43. *Grad-CAM of misclassified object for model trained on full-dataset, rows represent type specific gradients and column represents layer based heatmap on based on the specific type*

6.2 Discussions

As with building a neural network, every part of an industrial sorting pipeline for the UHV blue sorter needs to be adjusted and optimized to provide dependable and scalable performance. The foundation of this system relies on two critical pillars: the quality of preprocessed data and the diversity of object representations. Without clean data and sufficient variability, even the most sophisticated model becomes susceptible to failure in real-world applications. For example, Type 3 plastics constitute only 4.4% of our current dataset and present a statistical imbalance that cannot be rectified merely by adjusting model weights. While such adjustments might temporarily boost accuracy, they do not address the core issue: the lack of sufficient representation of this class in our dataset. For an industrial-grade system, we need a data distribution that reflects reality, only then can the model achieve the precision required for an automated system that boasts a 90% sorting accuracy. Yet even this threshold of 90% accuracy is not the final frontier. We've explored generating synthetic images for rare plastic types using advanced image synthesis techniques. Additionally, we've considered leveraging methods to reconstruct complete images from plastic outliers, which would enhance data representation. However, implementing these strategies will require significant time and resources, making them viable for future development.

Misclassification is an inherent risk, especially in deformed plastic cases. Every misclassified object serves as a signal, a piece of feedback revealing the blind spots in the model. Rather than accepting these errors as a necessary compromise, we must engage in iterative analysis, studying the specific features of these objects, understanding the model's perception, and addressing the root causes through data augmentation, model refinement, or novel features. Ultimately, the goal is to reduce human intervention to a minimum. A fully automated sorting system must go beyond general accuracy. It must integrate continuous learning mechanisms, capable of refining its sorting capabilities as new data flows in, allowing for real-time adaptability. This will form the brain of the sorter, a dynamic, self-optimizing system that achieves industrial-grade performance through precision, feedback, and scalability.

The process of manually sorting plastics from bales during dataset collection was not just tedious; it was an inefficiency that highlighted the gap between human capability and scalable automation. Instead, leveraging the UHV blue sorter for pre-sorting tasks offers a clear path forward. This approach also opens the door to real-time data collection. The sorter, while functioning, could continuously gather data on the types of objects it processes, iteratively refining

its accuracy over time. Each sorting cycle becomes a training loop, allowing the model to learn from its mistakes and adjust predictions dynamically. This self-reinforcing feedback loop could be the key to a system that not only operates autonomously but also grows smarter with use.

Regarding object detection, the current segmentation threshold for dark objects introduces a manual oversight requirement, an inefficiency we should consider. Transitioning to YOLOv5 offers a more robust and precise approach to object localization, particularly for darker plastics, where traditional algorithms struggle. YOLOv5's capability to define tighter, more accurate bounding boxes eliminates the need for constant manual adjustment of thresholds, creating a more hands-off process. Lastly, it’s important to consider the hardware limitations that arise when scaling this operation. On the UHV blue sorter the overheating issue with the CPU, causing the cameras to fail, is a reminder that machine learning is more than just algorithms; it's also a systems challenge. The automated cooling solution we implemented is a perfect example of how hardware and software must co-evolve. As our models become more complex, and the demands on the system increase, managing this balance between computational load and operational stability will be vital for ensuring long-term, uninterrupted performance.

At this point, real-world testing remains uncharted territory for our project. Accuracy and efficiency benchmarks are yet to be established in practice, but the groundwork we’ve laid especially through machine-assisted data collection places us on the right trajectory. The convenience and speed with which we were able to collect large volumes of data underscores the power of automation. This efficient data acquisition is not merely a side benefit; it’s foundational to the project’s scalability. If the machine can capture and process data with such precision and volume at this stage, it points toward a future where real-life trials can be conducted with similar levels of efficiency. Once the system is fully refined, the leap from data collection to operational deployment will be critical. Each trial in real-life settings will provide feedback not just on accuracy but on the robustness and adaptability of the sorter in dynamic environments. If everything aligns as anticipated, these trials will offer the final proof that the machine is ready for its real-world debut.

7. CONCLUSIONS

This project tackles the pressing issue of municipal solid waste sorting, a challenge that has far-reaching implications for sustainability and recycling. By developing a novel system capable of efficiently separating recyclables, we aim to mitigate one of the most critical bottlenecks in the waste management ecosystem. The dataset used for training and validation consists of over 33k images, accompanied by a test set of 3k images is an unprecedented effort in the field. This robust dataset serves as a foundation not only for this project but also for future research into segmentation, classification, and object detection in waste sorting. A comprehensive exploration of various deep learning models was conducted to identify the most accurate architecture for implementation in the UHV blue sorter pipeline. The integration of these models into the pipeline was more than a matter of fitting data to a neural network; it involved rigorous evaluation of model performance, interpretability, and the capacity to visualize the decision-making processes within the models themselves. This approach underscores the importance of demystifying deep learning, moving beyond black-box solutions toward systems that offer actionable insights. The primary goal of this project was to build a scalable, industrial-grade plastic sorting pipeline, and while we have laid the groundwork, these first steps are only the beginning. The real value of this pipeline will emerge in future iterations, refined through real-world testing and engineering.

APPENDIX A. OPERATIONAL FRAMEWORK OF THE UHV BLUE SORTER

This section explores the operational framework of the UHV Blue Sorter, a system designed for object classification, which we've employed primarily for data collection but will likely serve as the final sorter, albeit with potential hardware modifications to enhance efficiency. The core architecture of the system can be segmented into three key components: the CPU head, the mechanical conveyor, and the air blower. 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 segmenter and the classifier, ensuring precise control of signals that govern the downstream processes. Built around a mid-range CPU 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 and four sets of LED lighting, arranged to provide optimal illumination during object capturing. The mechanical conveyor forms the backbone of the system's material handling. 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 (Figure A.2) completes the process by physically sorting the objects. Positioned at the end of the conveyor, the UHV Blue Sorter 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 UHV Blue Sorter as a versatile solution for industrial-grade object classification.



Figure A.1. UHV Blue sorter CPU head, conveyor belt

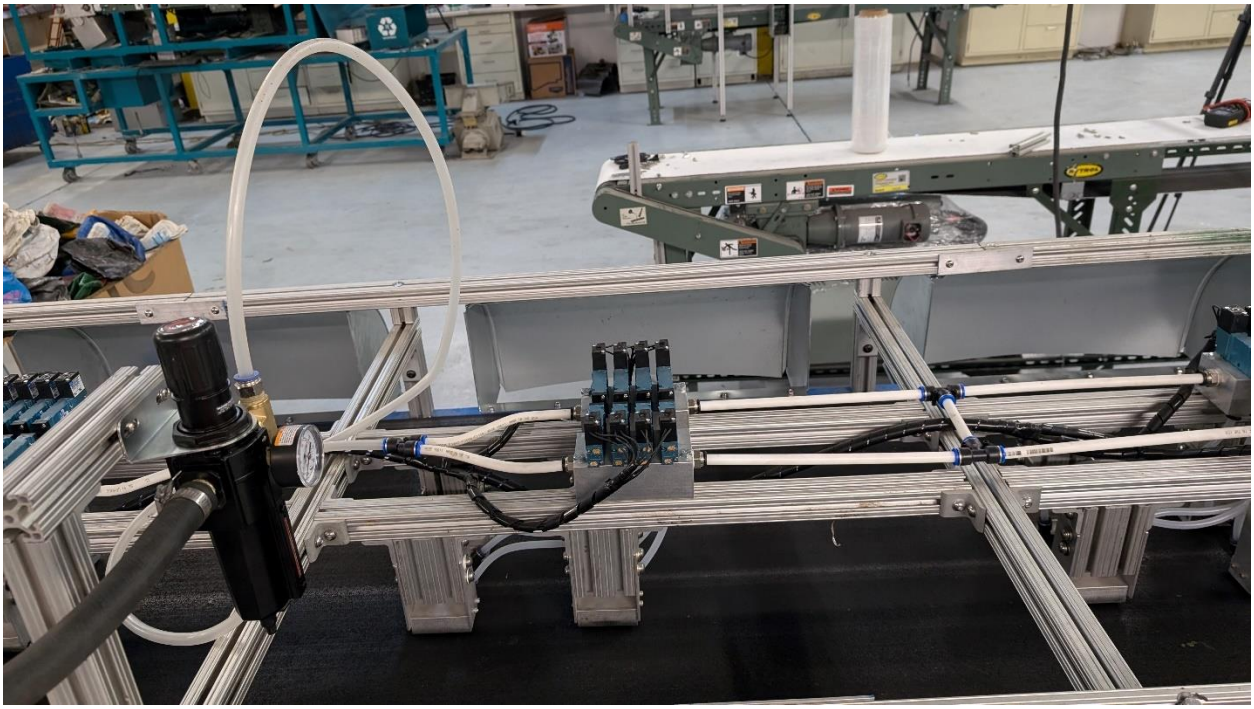


Figure A.2. UHV Blue sorter Air nozzles

REFERENCES

Basel Action Network. (2022). *Basel Plastic Waste Trade Violations Rampant One Year After Amendments Entry into Force.* Basel Action Network. Retrieved from https://myemail.constantcontact.com/Plastic-Waste-Trade-Violations.html?soid=1114999858498&aid=TjuIS3s34Ao

Bashkirova, D., Zhu, Z., Akl, J., Alladkani, F. M., Hu, P., Ablavsky, V., . . . Saenko, K. (2021). ZeroWaste Dataset: Towards Deformable Object Segmentation in Cluttered Scenes. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. doi:arXiv:2106.02740

Choi, J., Lim, B., & Yoo, Y. (2023). Advancing Plastic Waste Classification and Recycling Efficiency: Integrating Image Sensors and Deep Learning Algorithms. *Applied Sciences, 13*. doi:10.3390/app131810224

Dubanowitz, A. J. (2000). *Design of a Materials Recovery Facility (MRF) For Processing the Recyclable Materials of New York City's Municipal Solid Waste.* Masters Thesis, Columbia University, Department of Earth and Environmental Engineering . Retrieved October 2024, from https://www.columbia.edu/cu/seas/earth/wtert/newwtert/Research/sofos/dubanowitz_thesis.pdf

Dunn, J. (2018, June 1). *Intheblack*. Retrieved October 2024, from Intheblack: https://intheblack.cpaaustralia.com.au/environment-and-sustainability/china-waste-ban-australia-mess

Fortune Business Insights. (2024, November 18). *U.S. Recycled Plastics Market Size, Share, and COVID-19 Impact Analysis, 2023-2030.* Retrieved October 2024, from https://www.fortunebusinessinsights.com/u-s-recycled-plastics-market-108753

Geyer, R., Jambeck, J. R., & Law, K. L. (2017). Production, use, and fate of all plastics ever made,. *Science Advances, 3*, e1700782. doi:10.1126/sciadv.1700782

Ghosh, S., Sinha, J. K., Ghosh, S., Vashisth, K., Han, S., & Bhaskar, R. (2023). Microplastics as an Emerging Threat to the Global Environment and Human Health. *Sustainability, 15*. doi:10.3390/su151410821

Grimaldo, E., Karl, C. W., Alvestad, A., Persson, A.-M., Kubowicz, S., Olafsen, K., . . . Brinkhof, I. (2023). Reducing plastic pollution caused by demersal fisheries. *Marine Pollution Bulletin, 196*, 115634. doi:https://doi.org/10.1016/j.marpolbul.2023.115634

Halevy, A., Norvig, P., & Pereira, F. (2009). The Unreasonable Effectiveness of Data. *IEEE Intelligent Systems, 24*, 8-12. doi:10.1109/MIS.2009.36

He, K., Zhang, X., Ren, S., & Sun, J. (2015, December 10). Deep Residual Learning for Image Recognition. *ArXiv*. doi: arXiv:1512.03385v1

Jocher, G. (2020). *YOLOv5 by Ultralytics.* doi:10.5281/zenodo.3908559

Kumar, N., Garcia, M. G., & Tyagi, J. (2021). *United States Patent No. US20210346916A1*. Retrieved from https://patents.google.com/patent/US20210346916A1/en?oq=US20210346916A1

Liang, J., Ji, F., Abdullah, A. L., Qin, W., Zhu, T., Tay, Y. J., . . . Han, M. (2024). Micro/nano-plastics impacts in cardiovascular systems across species. *Science of the Total Environment, 942*, 173770. doi: https://doi.org/10.1016/j.scitotenv.2024.173770

Lubongo, C., & Alexandridis, P. (2022). Assessment of performance and challenges in use of commercial automated sorting technology for plastic waste. *Recycling, 7*. doi:10.3390/recycling7020011

Maaten, L. v., & Hinton, G. (2008). Visualizing Data using t-SNE. *Journal of Machine Learning Research, 9*, 2579--2605. Retrieved from http://jmlr.org/papers/v9/vandermaaten08a.html

Maliks, R. a. (2021). Multispectral data classification with deep CNN for plastic bottle sorting. *2021 6th International Conference on Mechanical Engineering and Robotics Research (ICMERR)*, 58-65. doi:10.1109/ICMERR54363.2021.9680850

MarketsandMarkets. (2023). *Markers and Markets*. Retrieved October 2024, from https://www.marketsandmarkets.com/Market-Reports/recycled-plastic-market-115486722.html

Mingxing Tan, Q. V. (2021, April 1). EfficientNetV2: Smaller Models and Faster Training. *ArXiv*. doi:arXiv:2104.00298

Neo, E. R., Low, J. S., Goodship, V., & Debattista, K. (2023). Deep learning for chemometric analysis of plastic spectral data from infrared and Raman databases. *Resources, Conservation and Recycling, 106718*, 188. Retrieved from https://www.sciencedirect.com/science/article/pii/S0921344922005511

Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2019). N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *ArXiv*. doi:arXiv:1905.10437

Proença, P. F., & Simões, P. (2020). TACO: Trash Annotations in Context for Litter Detection. *arXiv*. doi:arXiv:2003.06975

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *ArXiv*. doi:arXiv:1506.02640

Rezaei, M., Riksen, M. J., Sirjani, E., Sameni, A., & Geissen, V. (2019). Wind erosion as a driver for transport of light density microplastics. *Science of The Total Environment, 669*, 273-281. doi:https://doi.org/10.1016/j.scitotenv.2019.02.382

Ritchie, H. (2023). How much plastic waste ends up in the ocean? *Our World In Data*. Retrieved from https://ourworldindata.org/how-much-plastic-waste-ends-up-in-the-ocean

Santini, G., Castiglia, D., Perrotta, M. M., Landi, S., Maisto, G., & Esposito, S. (2023). Plastic in the Environment: A Modern Type of Abiotic Stress for Plant Physiology. *Plants, 12*. doi:10.3390/plants12213717

Sbrana, A., de Almeida, A. G., de Oliveira, A. M., Neto, H. S., Rimes, J. P., & Belli, M. C. (2023). Plastic Classification With NIR Hyperspectral Images and Deep Learning. *IEEE Sensors Letters, 7*, 1-4. doi:10.1109/LSENS.2023.3234401

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2016). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *ArXiv*. doi:arXiv:1610.02391

Syversen, T., Lilleng, G., Vollstad, J., Hanssen, B. J., & Sønvisen, S. A. (2022). Oceanic plastic pollution caused by Danish seine fishing in Norway. *Marine Pollution Bulletin, 179*, 113711. doi:https://doi.org/10.1016/j.marpolbul.2022.113711

Tan, M., & Le, Q. V. (2019, May 28). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *ArXiv*. doi:arXiv:1905.11946

Thung, G., & Yang, M. (2017). *TrashNet Dataset*. Retrieved 2024, from https://github.com/garythung/trashnet

Winters, J. (2022, April 18). *invw.org*. Retrieved October 2024, from InvestigateWest: https://www.invw.org/2022/04/18/rich-countries-are-illegally-exporting-plastic-trash-to-poor-countries-data-suggests/

Zheng, Y., Bai, J., Xu, J., Li, X., & Zhang, Y. (2018). A discrimination model in waste plastics sorting using NIR hyperspectral imaging system. *Waste Management, 72*, 87-98. doi:https://doi.org/10.1016/j.wasman.2017.10.015