Plastic Classification using Deep Learning

Avnish Patel
Northeastern University
Boston, USA
patel.avni@northeastern.edu

***Abstract*—The widespread use of plastic in grocery store packaging has led to a significant increase in plastic waste generation, which poses a severe threat to the environment. In this study, we aim to address this issue by developing an image classification model to identify the amount of plastic in grocery store products, which can be used for other plastic wastes also.To accomplish this, we will explore various deep learning techniques, including convolutional neural networks (CNNs) and transfer learning, to develop an accurate and robust model.Analysis and error analysis of this study will provide valuable insights regarding the amount of plastic generated through grocery store sales and enable us to identify products with excessive plastic packaging. This information can be used to develop policies and strategies to reduce plastic waste generation and promote sustainable packaging practices in the grocery store industry.**

***Index Terms*—EfficientNet, Adamax, Saliency Maps, Optuna, SubsetRandomSampler, Sensitivity Analysis**

I. Introduction

Plastic pollution is a major global problem, with millions of plastic bottles and bags used every minute and half of all plastic produced designed for single-use purposes. As a result, approximately 400 million tonnes of plastic waste is generated every year. To address this issue, various techniques have been developed to control plastic pollution, including the use of artificial intelligence (AI). In recent years, machine learning and deep learning techniques have been applied to tackle this problem, and this study aims to develop an image classification model to identify the amount of plastic in grocery store products.

At present, plastic waste detection methods are mainly manual investigation, aerial survey, and satellite monitoring. The manual investigation, Generally, plastic waste is visually inspected along the cross-section, time-consuming, labor intensive, and unsafe for operators.Traditional Machine Learning techniques were also implemented like Support Vector Machines (SVM), Logistic Regression etc. But these methods have been in vain if used solely as they manually extract features and select them which makes them error prone.To combat this issue and with the enhancement of the computing power of graphics processors and the increase of open training data sets, powerful Deep learning techinques have been trained and gave the birth to Convolutional Neural Network.The convolutional layers in this network use filters to extract features from the input image. These filters slide over the input image and perform a convolution operation to produce a feature map. The pooling layers reduce the dimensionality of the feature maps by downsampling them. Finally, the fully connected layers process the feature maps and produce the final output.This technique has shown to be highly successful in object recognition, image classification, and segmentation.Its application have been to most of the industries including manufacturing, food, medical etc.This inspired researchers to use advanced techniques and algorithms to solve environmental probelms, one of the most important is plastic waste classification.

In addition to identifying the amount of plastic generated through grocery sales, this research will provide valuable insights into identifying products with excessive plastic packaging. Retailers can then take steps to reduce plastic use and support environmental conservation efforts. This information can be used to develop policies and strategies that aim to reduce plastic waste generation and promote sustainable packaging practices in the grocery store industry.Overall, this study demonstrates the potential of AI and deep learning techniques in tackling plastic pollution, and its results have significant implications for environmental conservation and sustainability efforts.

II. Related Work

Ivana et al.(1) proposed a deep feature-based approach for marine debris classification, which utilizes a large annotated dataset of underwater trash, including Glass, Metal, Plastic, Rubber, Other Trash, and No Trash. They explored various deep convolutional networks, such as VGG19, InceptionV3, Resnet50, and Densenet 121, in comparison with traditional machine learning algorithms like Support Vector Machines (SVM), Random Forests, and k-Nearest Neighbors. The study found that deep convolutional networks outperformed traditional machine learning algorithms.

To perform feature extraction, the authors employed three different training methods: freezing all layers and training only the newly added classifier, fine-tuning all weights of the deep CNN, and first freezing the deep CNN and training the classifier before fine-tuning the deep CNN. Data augmentation was used during training to mitigate overfitting. The authors found that transfer learning significantly improved the model's performance compared to training from scratch. The fine-tuned Inception-ResNetV2 model architecture achieved the best results in terms of accuracy, F1-score, and Kappa coefficient.

They observed that the models achieved higher precision than recall, with the Rubber class exhibiting the best performance

and the Other garbage class demonstrating the weakest performance. The study also showed promising results in applying the trained model to new images of waste items collected in the Adriatic Sea.
Furthermore, the authors attempted to combine deep feature classification with conventional machine learning classifiers, which resulted in excellent results in some cases. The MobileNetV2 feature extractor achieved the best classification result with an SVM, with an accuracy of 85.32%, compared to 82.60% accuracy of the neural network classifier.
In conclusion, the study found that Inception-based FT feature extractors, particularly Inception-ResNetV2 and InceptionV3, achieved the best performance, with overall accuracies of more than 90%. They also observed that traditional SVM classifiers were credible alternatives to the neural network classifier, often outperforming it. The results demonstrate the potential for further exploitation of deep-learning-based models for real-time marine debris identification and classification in natural aquatic environments.

Wenlong et al.(2) conducted a study to address practical challenges in monitoring floating plastic waste using Unmanned Aerial Vehicles (UAVs) and deep learning techniques. They proposed a Classification and Target Detection (C+D) model that combines the EfficientNet image classification algorithm with a modified version of the YoloV5 target detection algorithm, tailored for the specific features of plastic floating materials. The authors collected a custom dataset using UAVs, covering an aerial survey area of approximately $20km^2$, which included three types of plastic waste: plastic bottle, polyfoam, and plastic bag. Their method involved training the EfficientNet for garbage classification into two categories (with and without garbage), followed by training the three subcategories and using Yolov5 for target detection, focusing on the details of each type of floating waste.
The authors evaluated their model on 300 UAV images, comprising 225 no-junk images, 75 junk images, and 323 junk examples. Their modified Yolov5 algorithm outperformed the original Yolov5 and Faster-RCNN, achieving improved recognition accuracy by up to 7% and speed by 3x compared to Faster-RCNN. The EfficientNet+Modified Yolov5 model also achieved the highest mean Average Precision compared to EfficientNet+Yolov5 and EfficientNet+Faster-CNN. The authors concluded that the Modified Yolov5 with EfficientNet achieved promising results, and this algorithm could reduce the influence of subjective factors in manual identification and classification, improve the efficiency of waste cleaning, and make significant contributions to protecting the water surface environment.

ZHUANG et al.(3) developed an automated garbage classification system by integrating a solar-powered bin with a Raspberry Pi. They utilized Resnet-32 as the base model for deep learning, as it has good accuracy and low computational complexity. To enhance the model's performance, they performed multi-feature fusion to optimize the input of the network by extracting features in parallel and fusing them at the end. They also improved the residual unit structure by down-sampling the feature information from the output of one residual block to the input of another, changing the input scale accordingly. Additionally, they utilized the Swish activation function to address the problem of gradient exploding and vanishing, which cannot be handled by ReLU in cases with large eigenvalues.
For data collection, they used a combination of real-life and online sources, resulting in 4168 images. The modified Resnet-32 algorithm with multi-feature fusion, feature reuse, and a modified activation function was tested on a garbage dataset, resulting in a 1.37% increase in accuracy.
In conclusion, the authors successfully created an automatic garbage classification system by integrating the proposed algorithm and necessary hardware, achieving a high classification accuracy of 99.96% and a quick classification cycle of 0.95 seconds on average.

III. Dataset & Methods

A. Dataset

In a recent study on plastic waste classification at Northeastern University, a large dataset comprising approximately 6000 images was utilized. The dataset included images of varying plastic amounts, such as heavy-plastic, low-plastic, no-plastic, and no-image categories.
The dataset was divided into two subsets, namely the training and testing datasets, with an 80% and 20% distribution, respectively. To evaluate the performance of the model during the training process,a validation set comprising 20% of the training data was created.Table (1) shows the distribution.

Class	Images	Training Set	Test Set
Heavy-plastic	1966	1548	417
No-image	602	490	111
No-plastic	1376	1104	272
Some-plastic	2773	2230	542
Total	6717	5372	1342

TABLE I: Distribution of Images in the Plastic Classification Dataset for Training and Testing

The below figure shows the various classes represented in the dataset.

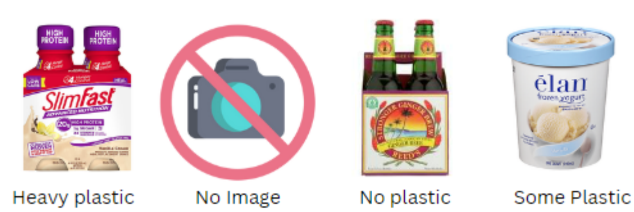

Fig. 1: Different Classes

B. Methods

1) Data Imbalance: The dataset used in this study exhibits a highly imbalanced distribution across its classes, with some-plastic and heavy-plastic images dominating the majority of the samples, while no-image and no-plastic images are relatively underrepresented. To address this issue, we propose using the SubsetRandomSampler technique, which randomly selects a subset of data for training and validation, ensuring a more balanced and representative sample of the dataset. This approach may improve the model's ability to generalize and perform well on both majority and minority classes.

Furthermore, we also employed the concept of class weighting to tackle the problem of data imbalance. Class weights are used to balance the contribution of each class to the loss function during training. By assigning higher weights to underrepresented classes, the model can learn to give equal importance to each class, regardless of their frequency in the dataset. We calculated class weights based on the relative frequency of each class in the dataset and used them in combination with the Cross-entropy loss function.

Overall, the use of SubsetRandomSampler and class weighting techniques can significantly enhance the performance of deep learning models when dealing with imbalanced datasets, which are prevalent in many real-world applications.

2) Deep Convolutional Architectures: In the domain of image classification tasks, Deep Convolutional Neural Networks have demonstrated remarkable performance when compared to traditional Computer Vision methods. To explore the applicability of these models in plastic classification, a comparative analysis was carried out on some of the most popular models including Resnet-50, VGG-16, Efficient-net-B0, and Efficient-net-B7. The selection of these models was primarily based on their state-of-the-art performance on the ImageNet Dataset in different years. Amongst these models, the present state-of-the-art model, EfficientNet, was selected for further analysis.

A) In regards to Resnet-50(4), it is composed of 50 layers, the majority of which are convolutional layers. The key innovation of ResNet is the utilization of "skip connections" or "identity mappings," which facilitate information flow around certain layers and directly to later layers. This allows for the preservation of gradients during backpropagation and enables the training of deep neural networks.

B) Regarding VGG16, it contains 16 layers, 13 of which are convolutional layers, and the remaining are fully connected layers. The convolutional layers are all 3x3 filters with max pooling layers between them. The fully connected layers are followed by a softmax activation function, which produces class probabilities.

(C) EfficientNet(4) is a recent deep convolutional neural network architecture that has gained popularity for its ability to achieve high accuracy while being computationally efficient. The architecture is based on a compound scaling method that involves scaling up the depth, width, and resolution of the network simultaneously. The backbone network of EfficientNet is composed of repeated blocks connected in a hierarchical fashion, each containing convolutional layers, activation functions, and pooling layers. The use of auxiliary classifiers during training further improves the robustness of the model.

In comparison to other popular models like ResNet-50 and VGG16, EfficientNet has been found to outperform them on various image classification tasks while requiring fewer parameters and computation. The model has been developed in different versions, denoted by a scaling coefficient, with higher coefficients indicating larger and more powerful networks.

In this study, we have used the EfficientNet-B0 model for plastic classification task. This decision was based on the fact that our dataset consisted of a relatively small number of images and choosing a larger model would not make much difference. Moreover, using a smaller model like EfficientNet-B0 was computationally and time efficient. To confirm our hypothesis, we trained both EfficientNet-B0 and EfficientNet-B7 models and found that the former achieved similar test accuracy while being computationally and memory-wise less expensive.

3) Hyperparameter Tuning: Hyperparameter tuning is an important step in optimizing the performance of a model. In this study, I used an open-source hyperparameter optimization framework called Optuna to find the optimal hyperparameters of the model. Optuna uses a Bayesian optimization algorithm, which is a probabilistic model-based approach that involves constructing a probability distribution over the function being optimized (i.e., the objective function). This distribution is updated as new observations are made, which allows the algorithm to explore the hyperparameter space more efficiently and converge to better hyperparameters in fewer iterations than other optimization algorithms.

By using Optuna, I was able to systematically explore the hyperparameter space of the model and find the combination of hyperparameters that resulted in the best performance. This approach saved time and resources compared to manually tuning hyperparameters, and resulted in a more robust and optimized model.

Overall, hyperparameter tuning is an essential step in developing machine learning models, and using a framework like Optuna can greatly improve the efficiency and effectiveness of the process.

4) Adversarial Attacks: Adversarial attacks are carefully crafted perturbations added to input data, such as images, with the intention of causing a classification model to misclassify them. These perturbations are designed to cause the model to produce incorrect outputs with high confidence, and are used to test the robustness of models and identify weaknesses in their decision-making processes. The Fast Gradient Sign Method (FGSM) is a commonly used technique for generating adversarial examples, where the gradient of the loss function

with respect to the input image is calculated and used to create an adversarial perturbation. This perturbation is then added to the original image to create an adversarial example. The FGSM attack is formulated as
$adv_x = x + \epsilon * \text{sign}(\nabla_x J(\theta, x, y))$,
where adv_x represents the adversarial image, x represents the original input image, y represents the original input label, ϵ is a multiplier that ensures the perturbations are small, θ denotes the model parameters, and J represents the loss function.

5) Saliency Maps: A saliency map is a valuable tool that can be used to visualize the contribution of each pixel in an image towards the output of a convolutional neural network. By analyzing saliency maps, we can gain insight into how the network is making its decisions and which features are being prioritized by the model. In this study, we utilized Guided Backpropagation to generate saliency maps. Guided Backpropagation is a gradient-based approach that multiplies the gradient of the output of a neural network with respect to the input image by the ReLU function. This ensures that only the positively contributing pixels are highlighted in the saliency map. The generated saliency maps were then used to better understand the decision-making process of the network, and to identify important features that were being used to classify the input images. The mathematical representation of Guided Backpropagation is presented as follows:
$guided_backprop(x) = ReLU(gradient(y, x) * ReLU(x))$,
where x represents the input image, y is the output of the neural network, and $ReLU$ is the rectified linear unit activation function.

C. Experiments

1) Experimental Setup: For implementation of deep models, I used Python 3.9.16 programming language along with Pytorch 2.0.0 machine learning framework.Other libraries like the Seaborn package for visualizations, scikit-learn for t-SNE plots and captum.attr for Saliency maps.

Model	Test Accuracy	Saved Model size(Mb)
VGG16	70%	500
Resnet50	79%	90
EfficientNet-B0	80%	17
EfficientNet-B7	80%	240

TABLE II: CNN Architectures

2) Analysis of Deep Learning Networks: From Table(2), we can clearly see that VGG16 is the most computationally expensive taking 500 Mb of saved model and giving the least accuracy.Although Resnet50,EfficientNet-B0 and EfficientNet-B7 give nearly same test accuracy, but the memory compute is vastly different.Thus EfficientNet-B0 is the selected model for our plastic classification task.

3) Sensitivity Analysis: Sensitivity analysis is a method used to determine how sensitive a system is to changes in its parameters or inputs. It is a process of evaluating the behavior of a system or model by altering the values of its inputs or parameters and observing the resulting changes in its outputs or outcomes.

A. Hyerparameter Tuning

Hyperparameter optimization was conducted using Optuna, a Bayesian-based framework for hyperparameter optimization. We defined an objective function that takes a set of hyperparameters as input and returns a scalar value that represents the performance of the model. The hyperparameters included learning rate, weight decay, dropout, epochs, step size, gamma, and optimizer.

Hyperparameter	Range	type
Learning Rate	1×10^{-5} to 1×10^{-2}	log-uniform
Weight decay	1×10^{-5} to 1×10^{-2}	log-uniform
Dropout	0.0 to 0.5	uniform
Epochs	5 to 25	int
Step size	1 to 10	int
Gamma	1×10^{-5} to 1×10^{-2}	log-uniform
Optimizer	Adam,Adagrad,Adamax	–

TABLE III: Optuna Hyperparameters

In the hyperparameter tuning process using Optuna, a Trial object was utilized to represent a single execution of the objective function with a specific set of hyperparameters. To balance the computational resources and time required, 20 was chosen as a reasonable number of trials to conduct. The direction argument was set to ”maximize”, indicating that Optuna aimed to find the set of hyperparameters that maximized the objective function.
After various iterations with EfficientNet-B0, I was able to obtain the parameters as shown in Table(3)
Based on the optimized hyperparameters obtained through Optuna as shown in Table(4), our model was able to achieve an accuracy of 95% on both the training and validation sets. Furthermore, upon evaluating the model on the test set, we obtained an accuracy of 80%. This represents a substantial improvement over our previous results with the imbalanced dataset, where we only achieved 50% accuracy on the test set.

Hyperparameter	Range
Learning Rate	0.00045
Weight decay	0.0016
Dropout	0.2
Epochs	25
Step size	8
Gamma	0.1
Optimizer	Adamax

TABLE IV: Optimized Hyperparameters

We were able to reduce overfitting by utilizing the techniques of class weights and sampling. By using class weights, we were able to assign a higher weight to the minority classes, which helped to address the class imbalance problem. Additionally, we utilized sampling techniques using SubsetRandomSampler to generate a balanced dataset. This helped to ensure that the model did not become biased towards any particular class, thereby improving its overall performance on the test set.

B. Dataset Size Analysis and its Effect on Test Accuracy

In order to analyze the impact of dataset size on test accuracy, a plot was generated depicting the variation in test accuracy for different dataset sizes in the range of 100 to 1342.

It should be noted that the dataset sizes were randomly

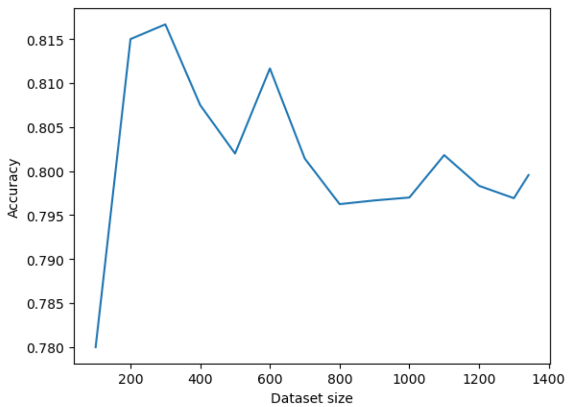

Fig. 2: Dataset sizes

sampled, and hence the test accuracy varied with each subset of images selected. This plot helps us understand the significance of dataset size and the quality of data. It was observed that a dataset with a smaller size of 300 images had higher accuracy than a larger dataset of around 500 images. This could be due to the presence of complex images in the larger subset, which the model may find difficult to discern. Overall, this analysis provides valuable insights into the importance of dataset size and the role it plays in model performance.

C. Saliency Maps

Saliency maps help us to provide insight into how the network is making its predictions and which features it is focusing on.

Fig(3) shows that in identifying the image, the model is also

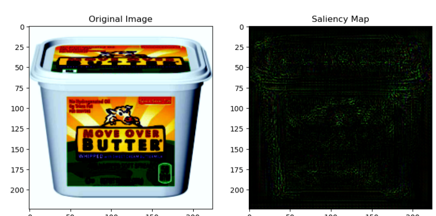

Fig. 3: Saliency Map 1

taking into account the unnecessary features like the cow and the colors present in the image

Fig(4) is another example that the model is focussing on the man's face on the box rather than the type of plastic.

Fig(5) shows that the model is correctly focussing on the plastic structure and maybe trying to learn its properties to identify the type of class it belongs to.

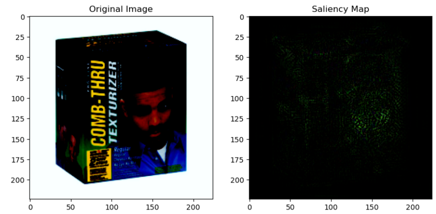

Fig. 4: Saliency Map 2

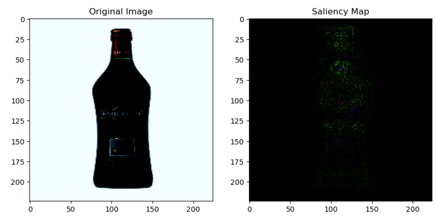

Fig. 5: Saliency Map 3

4) Visualizing Test results: Analysis of misclassified images in each class is an important step to improve the performance of a classification model. In this study, we analyzed misclassified images in each of the four classes, namely Heavy Plastic, No Image, No Plastic and Some Plastic.

For heavy class:

Fig. 6: Heavy class Confidence

For No image class:

Fig. 7: No Image Confidence

For No plastic class:

For Some plastic class:

Fig. 8: No Plastic Confidence

Fig. 9: Some Plastic Confidence

In the Heavy Plastic class, one of the images was almost black and contained only a partly visible dark line, which was difficult for the model to classify. The misclassification of the last image in this class was due to the contents or texture of the package. In the No Image class, the last image resembled a logo and was misclassified due to its unusual appearance. The misclassification of the last image in the No Plastic class was due to its bluish color and darker shades at the corner. This made it difficult for the model to predict its class, and the model had low confidence in its classification. In the Some Plastic class, the last image was misclassified as No Plastic due to its bluish color and darker shades at the corner. Overall, analyzing misclassified images helps in identifying the challenges faced by the model and provides valuable insights for improving the model's performance.

a) Confusion Matrix

It is a visualization which tells us the performance of the model on all the classes.

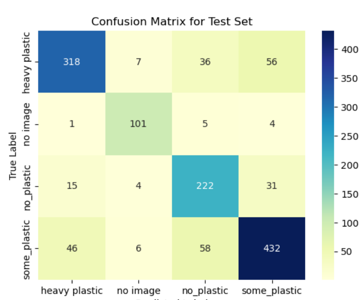

Fig. 10: Confusion Matrix

Fig(10) indicates that in Class 0 ("heavy plastic"), the model correctly predicted 318 instances as "heavy plastic" (true positives), but mistakenly predicted 1 instance as "no image" (false negative), 15 instances as "no_plastic" (false negative), and 46 instances as "some_plastic" (false negative). This provides a comprehensive breakdown of the classification results and highlights the areas where the model may need to be improved to increase its accuracy in predicting specific classes.

b) Classification Report

In the classification report as shown in Fig(11), various performance metrics such as accuracy, precision, recall and F1 score are presented.

	precision	recall	f1-score
heavy plastic	0.84	0.76	0.80
no image	0.86	0.91	0.88
no_plastic	0.69	0.82	0.75
some_plastic	0.83	0.80	0.81
accuracy			0.80

Fig. 11: Classification Report

Precision of 0.86 for the "no image" class indicates that 86% of the instances that the model predicted as "no image" were actually "no image". Similarly, a recall value of 0.82 for the "no_plastic" class shows that out of all instances that are truly "no_plastic", the model correctly classified 82% of them. The F1-score of 0.80 for the "heavy plastic" class represents the harmonic mean of precision and recall values, which were found to be 0.84 and 0.76 respectively. Lastly, the overall accuracy of the model was calculated as 0.80, indicating that the model correctly classified 80% of the instances in the dataset.

c) Violin Plot

A violin plot is a useful data visualization technique used to visualize the distribution of probabilities for the predicted class across all four classes as shown in Fig(12).

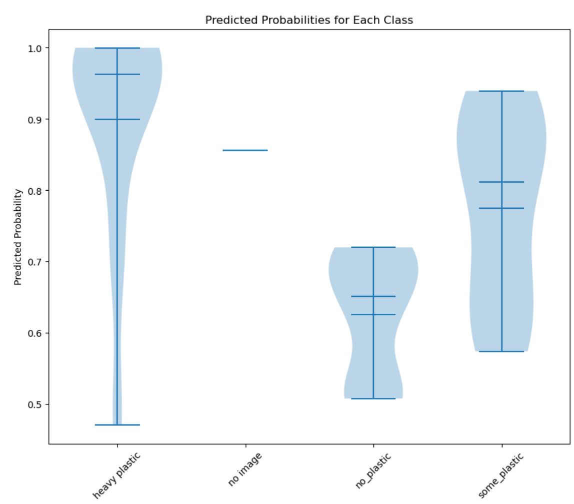

Fig. 12: Violin plot

The violin-shaped curve representing each category or group provides information on the density of data points at different probabilities, with thicker parts of the curve indicating regions with more data points and thinner parts indicating regions with fewer data points. Additionally, the horizontal line within each violin represents the median of the data. In this study, the heavy plastic class is displayed in the upper region of the plot, indicating higher probabilities and more certain regions. Conversely, the no plastic class lies in the lower region, representing lower probabilities and less certain regions. The no image class appears to have minimal data, as evidenced by the thin line, while the some plastic class exhibits a balanced distribution of probabilities, with equal width across the violin plot, ranging from the upper to the lower bound.

d) Bar plot

The presented bar plot depicts the mean predicted probability for each of the four classes in the dataset.

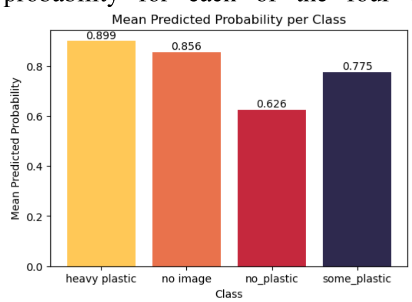

Fig 13 : Bar plot

Fig(13) highlights that the heavy plastic class, despite having a smaller number of instances than the no image class, has a higher mean predicted probability, indicating that the model performs well on this class. Furthermore, the some plastic class, which is larger than the no plastic class, also has a high mean predicted probability, indicating that the model can accurately classify some plastic instances. However, the mean predicted probability for the no plastic class is comparatively lower, suggesting that the model faces difficulties in correctly identifying instances of this class. Overall, the bar plot provides insight into the model's performance for each of the four classes and can be used to identify areas for improvement in future iterations of the model.

5) t-Distributed Stochastic Neighbor Embedding(t-SNE) Visualizations: In order to visualize the separation and clustering of the different classes in the dataset, t-SNE (t-distributed stochastic neighbor embedding) was utilized. The t-SNE plot for the training dataset in Fig(14) shows that all four classes have been clearly separated into distinct clusters in different parts of the plot.

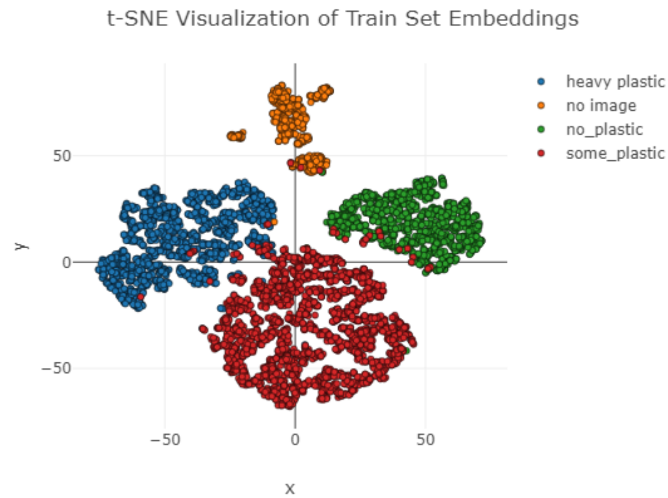

Fig 14: Trainset t-SNE

However, there are a few instances of the red dots (some_plastic) being merged with the green dots (no_plastic).

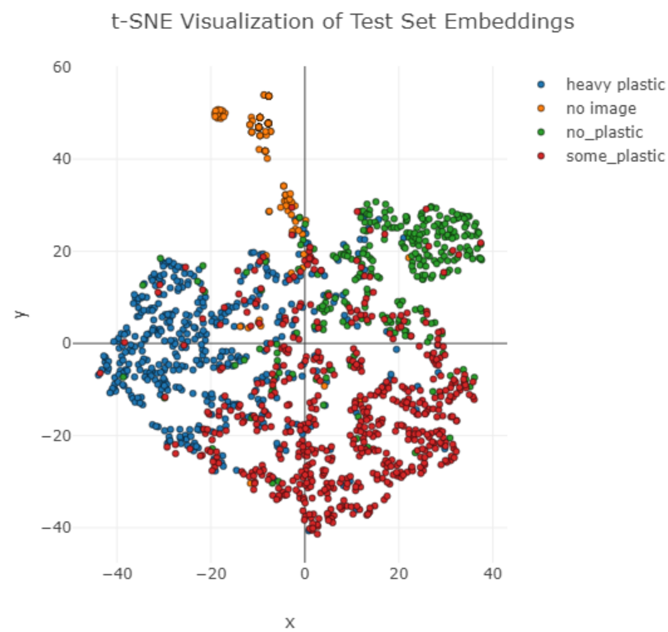

Fig 15: Testset t-SNE

The t-SNE plot for the test dataset in Fig(15) shows that there is still clear clustering happening in different parts of the plot, although it is not as separated as the training set. In the section where orange dots (no image), green dots (no_plastic), red dots (some_plastic), and blue dots (heavy plastic) meet, there is an overlap, indicating uncertainty and confusion by the model in correctly classifying these instances. The overlap between the red dots (some_plastic) and blue dots (heavy plastic) suggests that the model is facing difficulties in distinguishing between heavy and low amounts of plastic, which can also be difficult for humans to discern in some images.

6) Adversarial Attacks: Small changes are done via the Fast Gradient Sign Method and different values of epsilon are passed to study the changes observed in the models.
Epsilon value=0.1
For heavy plastic:

Fig. 13: Adversarial Heavy on epsilon=0.1

In Fig(18), only a single image i.e bottle is being correctly classified.All others are misclassified.Heavy plastic images are being classified as some plastic and no image. For No Image:

Fig. 14: Adversarial No Image on epsilon=0.1

In Fig(19), as expected, the result is the same as the original image.
For No Plastic:

Fig. 15: Adversarial No Plastic on epsilon=0.1

In Fig(20), the first two items which are somewhat similar, are being classified as some plastic.The bottle is being identified correctly.For the last 3 examples, which are exactly the same but with different lighting, the first two are being classified as some plastic and the last one as no image.
For Some Plastic:

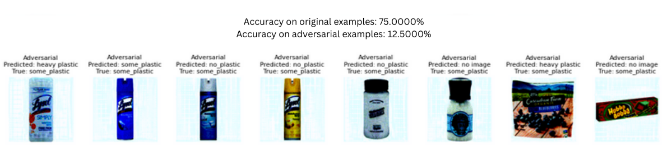

Fig. 16: Adversarial Some Plastic on epsilon=0.1

In Fig(21), this is a particularly challenging one as all bottles look the same.Only the second one is being correctly classified although the second and the third one are similar.There is huge disparity between the true class and the predicted class by adversarial network.

D. Discussion & Summary

The present study aimed to analyze the performance of several well-known CNN architectures in plastic classification. The results showed that EfficientNet-B0 outperformed Resnet50 and VGG16 in terms of both memory and time efficiency. The use of data balancing techniques, combining Class Weights and Subset Random Sampler, proved to be crucial in achieving an accuracy of 80%. Hyperparameter tuning was carried out using Optuna, which resulted in selecting Adamax as the most efficient optimizer with a learning rate of 0.00045 and weight decay of 0.0016. A dropout rate of 0.2 and a number of epochs of 25 were found to be effective in controlling the model overfitting and improving performance.
Our analysis showed that the model had high certainty in predicting the heavy plastic class, while no plastic showed the lowest confidence. However, the model found it difficult to differentiate between images of high plastic and some plastic. Visualizing the convolutional filters and saliency maps provided insights into what the feature was detecting, which could guide further improvement of the model. By performing Adversarial attacks, we also assessed the vulnerability of the data and its sensitivity to even small values of perturbation.
This analysis-based approach to plastic classification, with a greater focus on error and data analysis, can help in debugging the model and identifying areas for improvement. Future research could be done on improving the way the model is used, such as incorporating metadata like brand name or product name. Ensemble methods, such as combining the model with another model performing data enrichment, could also be explored to improve classification accuracy. Overall, the subject of performing deep learning models on plastic classification is essential for tackling the world's problems related to plastic pollution and dumping.

References

[1] Marin, I.; Mladenović, S.; Gotovac, S.; Zaharija, G. Deep-Feature-Based Approach to Marine Debris Classification. Appl. Sci. 2021, 11, 5644. https://doi.org/10.3390/app11125644

[2] Jakovljevic, G.; Govedarica, M.; Alvarez-Taboada, F. A Deep Learning Model for Automatic Plastic Mapping Using Unmanned Aerial Vehicle (UAV) Data. Remote Sens. 2020, 12, 1515. https://doi.org/10.3390/rs12091515

[3] Z. Kang, J. Yang, G. Li and Z. Zhang, "An Automatic Garbage Classification System Based on Deep Learning," in IEEE Access, vol. 8, pp. 140019-140029, 2020, doi: 10.1109/ACCESS.2020.3010496.

[4] [A ResNet-50-Based Convolutional Neural Network Model for Language ID Identification from Speech Recordings](https://aclanthology.org/2021.sigtyp-1.13) (Celano, SIGTYP 2021)

[5] Simonyan, K. & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR, abs/1409.1556.

[6] Tan, M. and Le, Q.V. (2019) EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, 9-15 June 2019, 6105-6114. http://proceedings.mlr.press/v97/tan19a.html