Plastic Usage Classification

1st Yalala Mohit
Khoury College of Computer Sciences
Northeastern University
Boston, USA
mohit.y@northeastern.edu

2nd Dhruv Kamalesh Kumar
Khoury College of Computer Sciences
Northeastern University
Boston, USA
kamaleshkumar.d@northeastern.edu

***Abstract*—This research paper reports on a project led by Professor Amanda Welsh and her team, which aimed to quantify the amount of plastic generated through grocery store sales. The team developed a set of 6,000 product images that were classified into four categories: no-plastic, some-plastic, heavy-plastic, and no-image (not a product). In addition, they collected 350,000 unlabeled images of products in-situ and aimed to classify them according to the amount of plastic present. The primary objective of this project was to develop a model capable of accurately classifying the labeled product images according to the amount of plastic in them. The study evaluated various models using different metrics such as accuracy, precision, recall, and F1 score. The highest F1 score was 88.77 for the classification task. Additionally, the team provided a set of product description metadata for further experimentation. The paper highlights the challenges faced and the deep learning techniques utilized in model development. The study results demonstrated promising accuracy rates and identified potential opportunities for future research in the area of plastic waste reduction. The project can significantly contribute to ongoing efforts to mitigate plastic pollution. The findings of this study may inform future initiatives aimed at reducing the amount of plastic waste generated through grocery store sales, which is a critical issue in today's world.**

***Index Terms*—Plastic classification, Deep Learning, Transfer Learning, Fine-Tuning**

I. Introduction

The issue of plastic waste is a growing global problem, and one of the major contributors to this issue is the grocery industry. A significant portion of products sold in grocery stores are packaged in plastic, which eventually ends up in landfills, oceans, and other natural habitats, causing severe environmental damage. To address this issue, Professor Amanda Welsh and her team from Northeastern University initiated a project to quantify the amount of plastic generated through grocery store sales.

The team developed a set of 6,000 product images classified into four categories based on the amount of plastic present. The categories include no-plastic, some-plastic, heavy-plastic, and no-image (not a product). Additionally, they collected 350,000 unlabeled images of products in-situ, which they aimed to classify similarly. The primary objective of the project was to develop a model capable of accurately classifying labeled product images according to the amount of plastic in them.

To achieve this objective, our team evaluated various deep-learning models using different metrics, including accuracy, precision, recall, and F1 score. The highest F1 score achieved was 88.77, indicating a high level of precision and recall in the classification task. Furthermore, the team under Professor Amanda Welsh provided a set of product description metadata for further experimentation, which can inform future research on plastic waste reduction.

The paper highlights the challenges faced in developing the model, including handling unlabeled images and developing deep learning techniques to achieve accurate classification. The study results demonstrated promising accuracy rates, which can contribute significantly to ongoing efforts to mitigate plastic pollution. The findings of our work may inform future initiatives aimed at reducing plastic waste generated through grocery store sales.

Overall, our work highlights the potential of advanced technology and data analysis techniques to address pressing environmental challenges. By accurately quantifying the amount of plastic waste generated through grocery store sales, this study can help in identifying areas for improvement and developing strategies for reducing plastic waste. The results of this study can contribute to ongoing efforts towards sustainable practices and environmental protection

II. Related Works

The study by Wolf et al. (2020) proposes a machine learning system based on convolutional neural network (CNN) technology to detect, classify, and quantify plastic litter in aquatic environments. The study investigates the effectiveness of CNNs in identifying plastics in different surroundings, such as rivers, river carpets, and beaches. The article also highlights the potential applications of automated detection and quantification algorithms in complementing traditional monitoring strategies and policy decisions aimed at reducing plastic pollution in our oceans and waterways.

The paper by Bobulski and Kubanek (2021) provides an overview of the potential of automated sorting methods using deep learning for plastic waste classification. The authors highlight the challenges associated with manual sorting and provide a detailed explanation of how deep learning can be used to recognize different types of plastic waste based on their visual characteristics. They also discuss the potential benefits and limitations of using deep learning for plastic waste classification. This paper is a valuable resource for researchers and policymakers interested in promoting sustainable waste management practices.

This paper by Chazhoor et al. (2022) presents a benchmark study on using transfer learning for plastic waste classification. The authors investigate the effectiveness of six state-of-the-art models on the WaDaBa plastic dataset and demonstrate that incorporating transfer learning significantly reduces the required training time. Although the focus is on supervised learning, the research methodology is detailed, and the authors' contributions are described. The study suggests that further improvements in accuracy can be achieved with a larger dataset in the future, and it may serve as a baseline for future research in this area. The paper provides valuable insights for researchers, policymakers, and industry professionals interested in improving recycling efforts and reducing plastic waste.

This paper by He et al. (2015) introduces a residual learning framework called ResNet, which consists of residual blocks that enable the reuse of learned features, allowing for the training of substantially deeper neural networks. The authors conducted experiments on CIFAR-10 and ImageNet datasets and found that ResNets outperformed other state-of-the-art models on both datasets while using fewer parameters and computational resources. The authors' approach has won several competitions and has significant implications for the field of deep learning.

A recent paper by Dosovitskiy et al. (2021) introduced a new architecture called the Vision Transformer, which uses a self-attention mechanism instead of convolutions for feature extraction. The Vision Transformer (ViT) achieved state-of-the-art performance on several image classification benchmarks, demonstrating that the self-attention mechanism can be effective for computer vision tasks. In addition, ViT has the advantage of being more interpretable than CNNs, as the self-attention mechanism allows for the visualization of feature importance.

Another recent paper by Vasu et al. (2023) proposed a hybrid architecture that combines the strengths of both CNNs and ViT. The authors demonstrated that the hybrid architecture, called Vision Transformer Hybrid (ViTH), can achieve better performance than both CNNs and ViT alone on several image classification benchmarks. The ViTH architecture uses a CNN to extract local features from the input image, which are then fed into a ViT to extract global features. This allows ViTH to capture both local and global information, leading to improved performance. The authors also introduced a new training procedure for ViTH, which involves pre-training the CNN and ViT separately and then fine-tuning them together.

Overall, the ViT and ViTH architectures represent promising developments in the field of computer vision, demonstrating that self-attention mechanisms can be effective for image classification tasks and that combining different architectures can lead to improved performance. Furthermore, the importance of Transfer learning and learning global contexts have been discussed in the above papers. These developments may have important implications for various applications, such as object detection, semantic segmentation, and medical image analysis.

III. Methods

In this section, we set up the workflow for our approach and walk through till our proposed methodology.

A. Data

The data statistics presented in Table I reveal some interesting insights about the dataset. Firstly, there is a significant class imbalance in the data. The "Some Plastic" class has the highest number of instances, followed by "Heavy Plastic," "No Plastic," and "No Image" classes. This class imbalance can lead to biased models that perform well on the majority class but poorly on the minority classes. It can also cause difficulties in model training, where the model struggles to learn the patterns from the minority class due to insufficient data.

TABLE I
CLASS-WISE DATA AVAILABILITY

	Heavy Plastic	**No Image**	**No Plastic**	**Some Plastic**
Full	1965	601	1375	2772
Train	1570	487	1098	2215
Test	189	68	128	287
Val	206	46	149	270

Another issue highlighted in Table I is the low data availability for some classes, such as the "No Image" class, which has only 601 instances in the full dataset. This low data availability can lead to overfitting, where the model learns to fit the noise in the data rather than the underlying patterns, resulting in poor generalization performance on new data. It can also cause difficulties in model evaluation, where the model's performance on the minority classes may be unreliable due to the small number of instances available for testing.

To address the problems caused by class imbalance and low data availability, there are several approaches that can be taken. One common approach is to perform data augmentation, where new data is generated by applying various transformations to the existing data. This approach can increase the number of instances in the minority classes, thus improving the model's ability to learn from them. Another approach is to use techniques such as oversampling or undersampling to balance the class distribution in the data. Oversampling involves duplicating instances from the minority classes while undersampling involves removing instances from the majority class.

Furthermore, transfer learning can be utilized to overcome low data availability issues, where pre-trained models can be fine-tuned on the dataset to improve performance. In addition, active learning can be used to select the most informative instances to annotate, thus maximizing the benefit of limited labeling resources.

In conclusion, Table I reveals class imbalance and low data availability issues in the dataset, which can affect model performance. However, there are various approaches that can be taken to address these problems, including data augmentation, class balancing, transfer learning, and active learning.

B. Fine-Tuning Vs Transfer Learning

One of the key reasons for the success of deep learning is the availability of pre-trained models, which can be used as a starting point for developing new models.

Transfer learning is a technique in deep learning that involves leveraging pre-trained models to develop new models for different tasks. The basic idea behind transfer learning is to use the knowledge learned by a pre-trained model on a large dataset to improve the performance of a new model on a smaller dataset. In transfer learning, the pre-trained model is used as a feature extractor, and the new model is trained on top of the extracted features. This approach is particularly useful when the new dataset is small and insufficient for training a deep neural network from scratch.

Fine-tuning, on the other hand, is a technique in deep learning that involves adapting a pre-trained model to a new task by updating its weights using a new dataset. Fine-tuning is typically performed on the last few layers of the pre-trained model, which are responsible for task-specific predictions. The idea behind fine-tuning is to adjust the pre-trained model to the new task by fine-tuning the weights of the last few layers while keeping the weights of the earlier layers fixed.

The main difference between transfer learning and fine-tuning is the degree of adaptation of the pre-trained model to the new task. In transfer learning, the pre-trained model is used as a feature extractor, and only the weights of the new model are trained. In contrast, in fine-tuning, the weights of the pre-trained model are updated, and the new model is trained on top of the updated weights. Fine-tuning can lead to better performance than transfer learning when the new dataset is large and similar to the original dataset used for pre-training. However, fine-tuning requires more computational resources and training time than transfer learning.

In practice, transfer learning and fine-tuning are often used together in deep learning applications. For example, a pre-trained model may be used for transfer learning to extract features from a new dataset, and then the last few layers of the model may be fine-tuned to adapt the model to the new task. This approach can lead to better performance than using either transfer learning or fine-tuning alone. Overall, transfer learning and fine-tuning are powerful techniques in deep learning that can help to improve the performance of models on new tasks, while reducing the amount of data and training time required.

C. Resnet50

ResNet50 is a variant of the Residual Network (ResNet) architecture introduced by He et al. in 2015, and it is based on the concept of residual learning.

ResNet50 is a 50-layer deep neural network that has demonstrated excellent performance in various image classification tasks. The architecture includes a series of convolutional layers, pooling layers, and fully connected layers. The input image is initially fed through a convolutional layer that applies a set of filters to extract features from the image. The output of this layer is then passed through a series of additional convolutional layers, each followed by a batch normalization layer and a ReLU activation function. The batch normalization layer normalizes the output of the previous layer, reducing the internal covariate shift problem and stabilizing the learning process. ReLU activation function introduces non-linearity into the model, making it capable of learning complex features.

One of the key advantages of ResNet50 over traditional CNNs such as VGG19 is its ability to learn residual features. In traditional CNNs, each layer attempts to learn the input feature representation entirely from scratch, which can lead to the vanishing gradient problem, making it difficult to train deeper networks. ResNet50, on the other hand, uses residual blocks that skip one or more layers and directly pass the input to the output of the block. These residual connections enable the network to learn residual features, which are the difference between the input and the output of the block. By learning residual features, ResNet50 can more effectively train deeper networks, allowing it to achieve higher accuracy on image classification tasks.

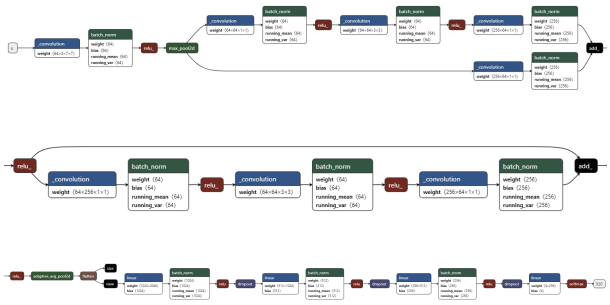

Fig. 1. The figure depicts the unique blocks in the architecture of a Resnet50 model. The first image corresponds to a skip layer block with Convolution downsampling, the second image corresponds to an identity skip layer block, and the third image corresponds to MLP head on top of the Resnet50 projections.

In conclusion, ResNet50 is a deep learning architecture that has proven to be highly effective in image classification tasks. The use of residual blocks allows it to effectively learn residual features, overcoming the limitations of traditional CNNs such as VGG19. The ResNet50 architecture has been widely adopted in various applications, and its success has inspired further research into deeper neural networks and improved feature learning.

D. Multilayer Perceptron

Multilayer Perceptron (MLP) is a type of neural network architecture that consists of multiple layers of perceptrons (neurons) with nonlinear activation functions. MLPs are widely used for classification and regression tasks in deep learning. In the context of transfer learning and fine-tuning, MLP can be used as a classification head on top of a pre-trained convolutional neural network (CNN) like ResNet50.

As discussed, ResNet50 is a powerful CNN architecture that has achieved state-of-the-art performance on various computer vision tasks. However, ResNet50 is a general-purpose CNN, and its learned features may not be optimal for a specific task, such as image classification on a new dataset. To adapt

ResNet50 to the new task, we can use transfer learning, where we use the pre-trained weights of ResNet50 as a starting point and fine-tune the model on the new dataset.

To fine-tune ResNet50, we can replace the last fully connected layer (classification head) of ResNet50 with an MLP head that is specifically designed for the new task. The MLP head can have multiple hidden layers with nonlinear activation functions, which enables it to learn complex relationships between features and classes in the new dataset.

For instance, we can attach an MLP head with 3 hidden layers to ResNet50, and use it for image classification with 4 output classes. The MLP head can have a different number of nodes in each layer, depending on the complexity of the task and the size of the dataset. Once the MLP head is attached to ResNet50, we can freeze the weights of ResNet50 and train only the weights of the MLP head on the new dataset. This process is known as fine-tuning.

Fine-tuning the ResNet50 model with the MLP head allows us to leverage the pre-trained weights of ResNet50, which provides a good starting point for learning the new task. Moreover, the MLP head allows us to learn task-specific features that are not present in ResNet50, which can further improve the performance of the model on the new dataset. Fine-tuning with MLP head is a common technique in transfer learning, and it has been successfully applied in various computer vision tasks.

E. Data Augmentation

Data augmentation is the process of creating new training examples by applying random transformations to existing ones. This can help to increase the diversity of the training data and improve the model's ability to generalize to new data.

The dataTransform object is created using the Compose method, which allows multiple transformations to be applied sequentially to the input data. The following transformations are applied in order:

- **RandomResizedCrop**: This transformation crops the input image to a random size between 50% and 100% of the original size and then resizes it to a fixed size of 224x224. This helps the model to learn features that are invariant to the object's position and scale in the image.
- **RandomHorizontalFlip**: This transformation randomly flips the image horizontally with a probability of 0.5. This helps to increase the variety of the training data by presenting the model with both the original and mirrored versions of the same image.
- **RandomRotation**: This transformation randomly rotates the image by up to 10 degrees. This helps the model to learn features that are invariant to the object's orientation in the image.
- **ColorJitter**: This transformation randomly adjusts the brightness, contrast, saturation, and hue of the image. This helps to increase the variety of the training data and improve the model's ability to handle variations in lighting conditions.
- **ToTensor**: This transformation converts the image to a PyTorch tensor, which is a multi-dimensional array that can be processed by the model.
- **Normalize**: This transformation normalizes the pixel values of the image using the mean and standard deviation of the dataset. The mean and standard deviation values are obtained from the dataset, and do not use widely used values obtained from ImageNet dataset. This helps to ensure that the input data has a consistent range of values, which can improve the stability and performance of the model.

Overall, data augmentation is a powerful technique for improving the performance of deep learning models, especially when the training data is limited. By applying random transformations to the input data, data augmentation can help the model learn more robust and generalizable features that can improve its ability to classify new examples.

F. Our Workflow

Our work builds on top of the ideas discussed above. We use a pretrained Resnet50 model which was pre-trained on the ImageNet dataset, and added a MLP classification head to it's top.

In this specific MLP architecture, there are three hidden layers, with 512, 256, and 128 neurons in each layer, respectively. The input layer size and output layer size depends on the specific task.Starting from the input layer (the flattened embeddings from a Resnet50 model), a sequence of fully connected layers (nn.Linear) with the specified number of neurons for each hidden layer is created. After each hidden layer, a batch normalization layer (nn.BatchNorm1d) to normalize the outputs and improve training is added. A ReLU activation function (nn.ReLU) is then applied to the output of each hidden layer to introduce non-linearity. Finally, a dropout layer (nn.Dropout) is added to each hidden layer's output to prevent overfitting by randomly dropping out some neurons during training. The output layer consists of a softmax activation function (nn.Softmax), which produces the class probabilities for the given input data.

The goal was to leverage Transfer learning and fine-tuning to improve the performance of the Resnet50 model for this task. However, after a few experiments, which would be discussed in detail, in the Experiments section, it was identified that Freezing the weights of the Resnet50 model seemed ineffective, and the best way to improve performance was to fine-tune the entire model.

IV. Experiments & Results

Since this is a classification task, there are a wide variety of pre-trained models available, which we anticipated to provide good performance on our dataset after Fine-Tuning.

A. Models Explored

In this section, we will discuss the architecture and advantages of six popular deep learning models which were explored for our task.

- VGG19
- Resnet34
- Resnet50
- Resnet101
- Vision Transformers Hybrid
- Vision Transformers

VGG19: VGG19 is a convolutional neural network (CNN) architecture that was developed by the Visual Geometry Group at Oxford University. It consists of 19 layers, including 16 convolutional layers and three fully connected layers. The convolutional layers have a small receptive field of 3x3, which allows the network to learn local features effectively. VGG19 has achieved excellent results on several benchmark datasets, including ImageNet, CIFAR-10, and CIFAR-100. One of the main advantages of VGG19 is its simplicity and ease of implementation.

Resnet34, Resnet50, and Resnet101: Resnet (Residual Network) is a CNN architecture that was introduced by Microsoft Research in 2015. The Resnet architecture contains skip connections, which allow the network to learn residual functions instead of learning the direct mapping between the input and output. Resnet34, Resnet50, and Resnet101 are variants of the Resnet architecture, where the number after the model name indicates the number of layers. Resnet50 is the most widely used variant due to its excellent performance on several image classification benchmarks, including ImageNet. One of the main advantages of Resnet models is their ability to train deeper networks without vanishing gradients.

Vision Transformers Hybrid and Vision Transformers: Vision Transformers (ViT) is a recently proposed CNN architecture that uses transformers, a type of self-attention mechanism, to process image patches. The ViT architecture has achieved state-of-the-art results on several image classification benchmarks, including ImageNet. Vision Transformers Hybrid is a variant of ViT that combines convolutional layers with transformers to improve the model's performance further. One of the main advantages of ViT models is their ability to learn global features effectively, which is important for image classification tasks.

In summary, each deep learning model has its own unique architecture and advantages. VGG19 is a simple architecture that achieves excellent results on several image classification benchmarks. Resnet models have skip connections that allow them to train deeper networks without vanishing gradients. Vision Transformers use transformers to process image patches and have achieved state-of-the-art results on several image classification benchmarks. Vision Transformers Hybrid combines convolutional layers with transformers to further improve the model's performance.

B. Hyper-Parameter Tuning

Hyperparameter tuning is a critical step in optimizing deep learning models. Hyperparameters are the parameters that define the structure of the model and its training process, such as learning rate, batch size, number of hidden layers, and number of neurons in each layer. Tuning these hyperparameters is crucial to achieving the best performance for a given task.

There are several methods for hyperparameter tuning, including manual tuning, grid search, random search, and Bayesian optimization. Manual tuning involves iteratively adjusting the hyperparameters based on the model's performance on a validation set. This method can be time-consuming and may not always result in the best possible performance.

Grid search involves defining a grid of hyperparameters and testing all combinations of values on the validation set. While this method is exhaustive, it can be computationally expensive for high-dimensional search spaces.

The random search involves randomly sampling hyperparameters from a search space and evaluating them on the validation set. This method is more efficient than grid search for high-dimensional search spaces.

Bayesian optimization involves constructing a probabilistic model of the objective function (i.e., the model's performance on the validation set) and iteratively selecting hyperparameters to optimize the model's performance. This method can be computationally expensive but is more efficient than a random search for low-dimensional search spaces.

In this research, we used a combination of manual tuning and random search to optimize the hyperparameters of our deep learning models using the Optuna Framework. We started with a set of reasonable hyperparameter values based on prior knowledge and adjusted them manually based on the model's performance on a validation set. We then used Optuna to explore the search space and find better hyperparameter values.

Our hyperparameter tuning process involved adjusting the learning rate, batch size, number of hidden layers, number of neurons in each layer, the dropout rates, the weight decay parameter, and much more. We evaluated the models on the same validation set and selected the best hyperparameters based on their performance on this set.

C. Metrics

In the evaluation of the performance of the deep learning models, we use several metrics to measure the accuracy, precision, recall, and F1 score. These metrics help in quantifying the performance of the models in classifying the input images correctly.

Accuracy: The accuracy metric measures the proportion of correctly classified samples among all the samples. It is calculated using the following formula:

$$Accuracy = \frac{\sum_{i=1}^{n} I(y_i = \hat{y_i})}{n} \tag{1}$$

where n is the total number of samples, y_i is the true label for the i-th sample, and $\hat{y_i}$ is the predicted label for the i-th sample.

Precision: The precision metric measures the proportion of true positives among all the predicted positives. It is calculated using the following formula:

$$Precision = \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} (TP_i + FP_i)} \tag{2}$$

where C is the number of classes, TP_i is the number of true positive predictions for class i, and FP_i is the number of false positive predictions for class i.

Recall: The recall metric measures the proportion of true positives among all the actual positives. It is calculated using the following formula:

$$Recall = \frac{\sum_{i=1}^{C} TP_i}{\sum_{i=1}^{C} (TP_i + FN_i)} \tag{3}$$

where FN_i is the number of false negative predictions for class i.

F1-score: The F1 score metric is the harmonic mean of the precision and recall metrics. It is a measure of the balance between precision and recall. It is calculated using the following formula:

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

Confusion Matrix: A confusion matrix is a table that is commonly used to evaluate the performance of a classification model. It is a matrix of actual versus predicted class labels, where each row represents the instances in an actual class and each column represents the instances in a predicted class.

For example, consider a binary classification problem where we are trying to predict whether an email is spam or not. The confusion matrix for this problem would look like, Table II.

TABLE II
CONFUSION MATRIX FOR BINARY CLASSIFICATION

	Gold Positive	**Gold Negative**
Predicted Positive	TP	FP
Predicted Negative	FN	TN

Here, TP represents true positives, which are the number of spam emails that are correctly classified as spam. FP represents false positives, which are the number of non-spam emails that are incorrectly classified as spam. FN represents false negatives, which are the number of spam emails that are incorrectly classified as non-spam. TN represents true negatives, which are the number of non-spam emails that are correctly classified as non-spam.

D. Results

In this project, we explored different deep learning models to classify images into different plastic content categories. We compared the performance of VGG19, Resnet34, Resnet50, Resnet101, Vision Transformers Hybrid, and Vision Transformers models on our dataset.

The baseline results without any Hyperparameter tuning, Data Augmentation, and Regularization techniques on our dataset resulted in a best F1 score of 67.9% while fine-tuning the entire Resnet50 pre-trained model.

The performance of the model was analyzed as to which features are being incorporated in the model to aid the classification. Figure 2 shows one such visualization with the filters learned from the first convolutional layer of a Resnet50 model and its effects on the images of each class. It can be seen that the effects of the filter on the Heavy plastic images looks completely different, while the representations from No Plastic and Some Plastic are very similar which can be attributed to the confusion of the model, as seen in Figure 3.

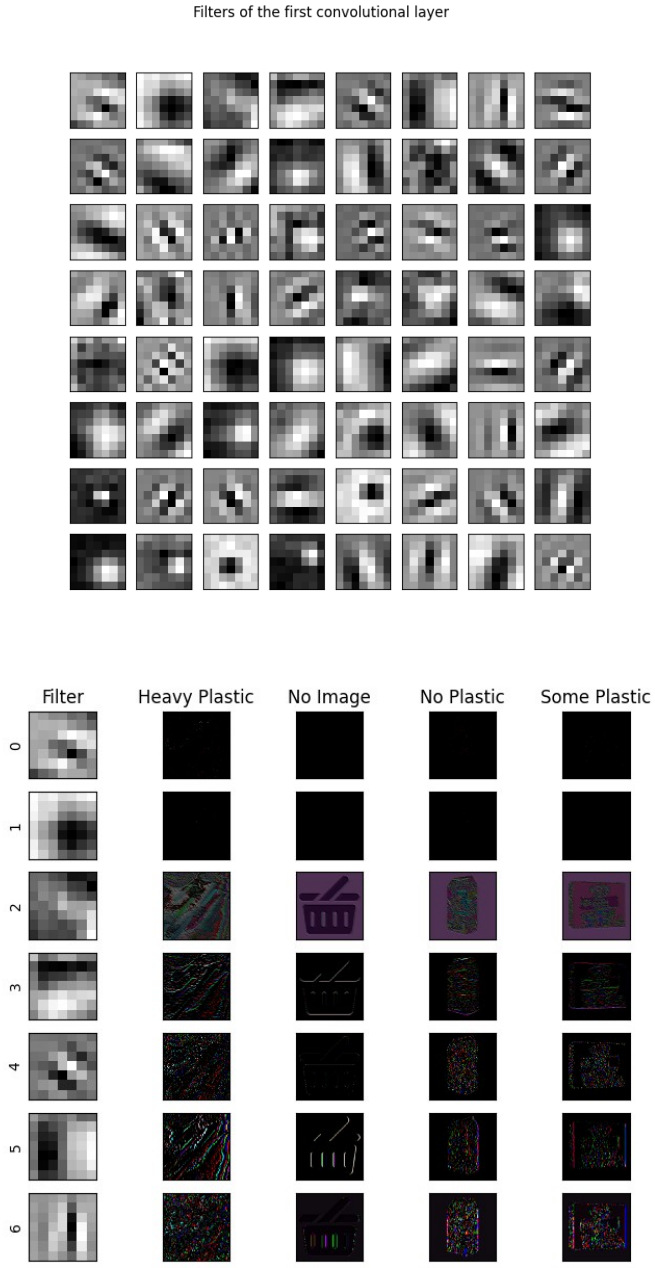

Fig. 2. The figure depicts the filters learned by the first convolutional layer of a resnet50 model, and the effects of the first 6 filters applied to an image from each class.

However, after various experiments, we found that the Resnet50 model achieved the best performance with an accuracy of 88.7% and an F1-score of 88.7%. We also performed data augmentation to improve the model's performance, and it helped further improve the model's performance.

As seen in Table III, Resnet50 (Final) performs the best

TABLE III
RESULTS OF VARIOUS MODELS

Model	Accuracy	Precision	Recall	F1 score
VGG19	65.3	65.1	65.3	65.2
Resnet34	62.8	62.4	62.6	62.5
Resnet50(Baseline)	67.9	67.9	67.9	67.9
Resnet101	59.4	59.2	59.4	59.3
ViT	49.1	47.5	46.28	46.88
ViT Hybrid	70.1	70	70.1	70.1
Resnet50 (Final)	**88.7**	**88.9**	**88.7**	**88.7**

among all the models, with the highest accuracy, precision, recall, and F1 score. ViT Hybrid also performs relatively well, with consistent precision, recall, and F1 score. The other models have varying levels of performance, with Resnet101 being the worst-performing model in the table.

The optimal values for the hyperparameters of Resnet50 (Final) were determined through tuning, and the following values were found to be the best:

- **BATCH_SIZE** of **32**
- **EPOCH** of **50**
- **LEARNING_RATE** of **0.001**
- **MOMENTUM** of **0.99**
- **WEIGHT_DECAY** of **0.001**
- **TRAIN_SPLIT** of **0.8**
- **MLP_HIDDEN_SIZE** of **[512, 256, 128]**

It can be seen in Figure 3 that previously, the baseline model was making numerous incorrect predictions, especially while classifying, the No plastic class. This showcased that the model, didn't generalize to unsee data well enough, and was confused.

However, further experimentations and tuning, allowed us to eradicate this confusion and the model seems to generalize to new data relatively well.

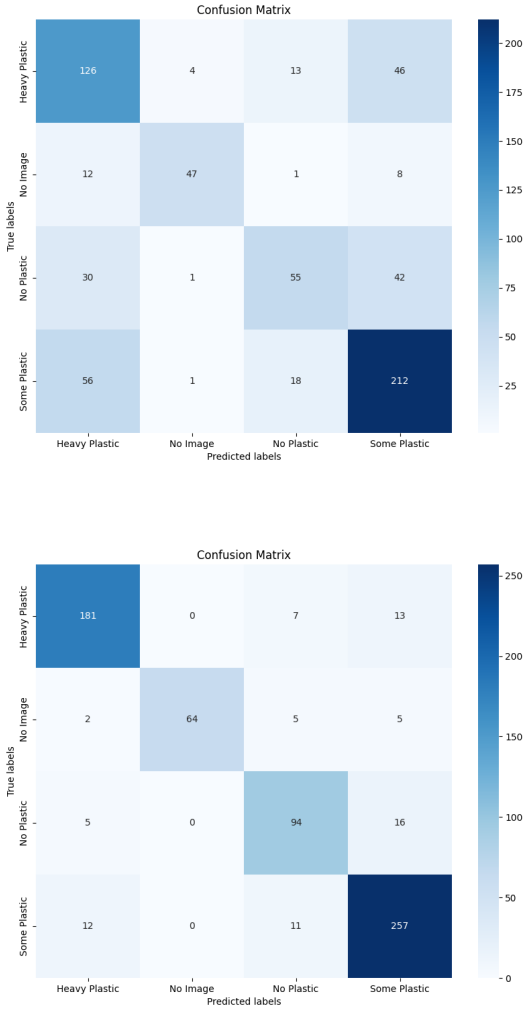

Fig. 3. The figure depicts the Confusion Matrix for both the baseline Resnet50 model, and Final Resnet50 model, respectively.

V. CONCLUSION & FUTURE WORK

In conclusion, this project aimed to address the issue of plastic waste generated through grocery store sales. Through the development of a set of labeled product images and the collection of unlabeled product images, the team led by Professor Amanda Welsh required a model capable of accurately classifying products based on the amount of plastic present. Our study evaluated various deep learning models using different metrics and achieved the highest F1 score of 88.77, indicating a high level of precision and recall in the classification task.

Our work identified potential opportunities for further research, including the exploration of additional product categories, like using product description metadata to convert the unimodal classification task into a multimodal problem. Moreover, our work demonstrated the potential of technology, data analysis, and deep learning techniques to address pressing environmental challenges.

As future work, this project can be extended in several directions. First, the study can be expanded to include additional product categories (text modality) and more extensive data collection could be done to further improve the accuracy of the model. Second, alternative deep learning techniques can be explored to handle the challenges of unlabeled data. Finally, the study's results can be used to inform initiatives aimed at reducing plastic waste in the grocery industry, such as the development of eco-friendly packaging or the promotion of reusable containers.

Furthermore, the project's findings can be used as a basis for further research on plastic waste reduction and sustainability, which is a critical issue in today's world. The study demonstrated the potential of advanced technology and deep learning techniques to address pressing environmental challenges and highlighted the need for collaborative efforts to promote sustainable practices and environmental protection.

VI. Acknowledgements

We are deeply grateful to Professor Bruce Maxwell for offering us the chance to be a part of such an exciting research project. His guidance and support throughout this project have been invaluable, and we would like to thank him for his patience and encouragement. We also extend our thanks to Professor Amanda Welsh and her team for providing us with all the necessary data resources, which made this research possible. Their hard work and dedication to this project have been inspiring, and we feel fortunate to have had the opportunity to work alongside them. Finally, we would like to express our appreciation to Ms. Shefali Khatri for her constant communication and support throughout this project. Her assistance has been instrumental in helping us navigate through the project, and we could not have done it without her.

References

Bobulski, J. and Kubanek, M. (2021). Deep learning for plastic waste classification system. *Applied Computational Intelligence and Soft Computing*, 2021.

Chazhoor, A. A. P., Ho, E. S. L., Gao, B., and Woo, W. L. (2022). Deep transfer learning benchmark for plastic waste classification. *Intelligence & Robotics*, 2(1):1–19.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition.

Vasu, P. K. A., Gabriel, J., Zhu, J., Tuzel, O., and Ranjan, A. (2023). Fastvit: A fast hybrid vision transformer using structural reparameterization.

Wolf, M., van den Berg, K., Garaba, S. P., Gnann, N., Sattler, K., Stahl, F., and Zielinski, O. (2020). Machine learning for aquatic plastic litter detection, classification and quantification (aplastic-q). *Environmental Research Letters*, 15(11):114042.