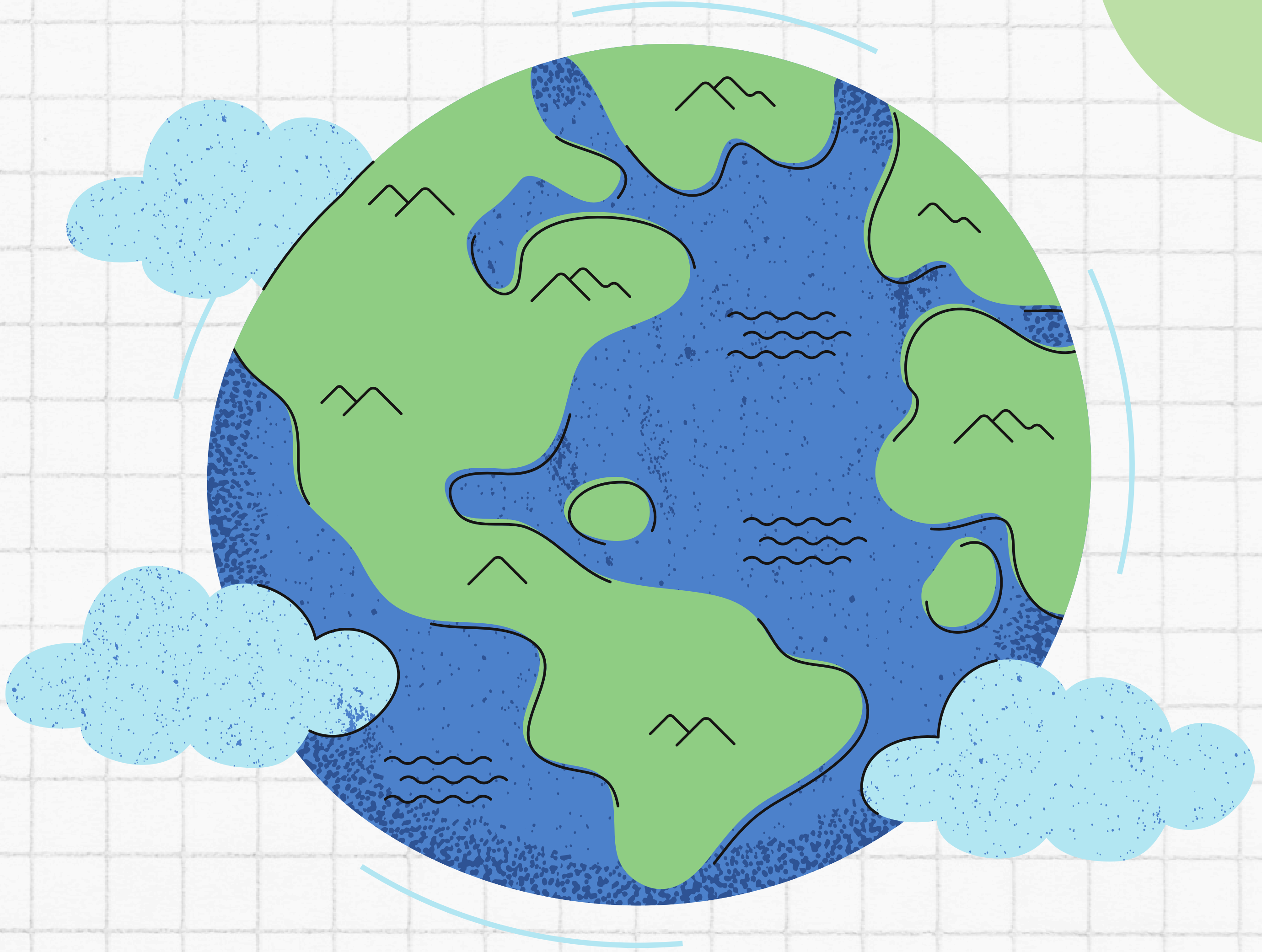
Final Report:

Waste Classifier for Degradable and Non-Biodegradable Wastes

Team Members: Abhishek IIITD (2021121), Vickey Kumar (2021299)



Abstract:

Inefficient waste sorting poses a significant challenge to environmental sustainability. Traditional methods, relying on manual sorting of degradable and non-biodegradable waste, are slow and prone to errors. We propose developing an automated waste classifier utilizing Convolutional Neural Networks (CNNs) to address this. This project aims to streamline and improve waste sorting processes by analyzing image data of waste objects, enhancing environmental sustainability.

Problem being Addressed, Literature Review, Methodology

- Effective waste management is crucial for environmental sustainability.
- Manual sorting of waste is time-consuming and error-prone.
- Automated waste classifiers streamline sorting, ensuring accuracy and efficiency.
- Development of such a classifier aims to improve waste management and environmental conservation.

Literature Review :

- Reduction techniques such as PCA and LDA for feature extraction and classification.
- Techniques may not capture intricate image details as effectively as CNNs.
- CNNs are chosen for their proven effectiveness in image classification and their ability to learn complex features.
- The decision to use CNNs is based on their superior performance in handling image data and learning intricate visual characteristics.

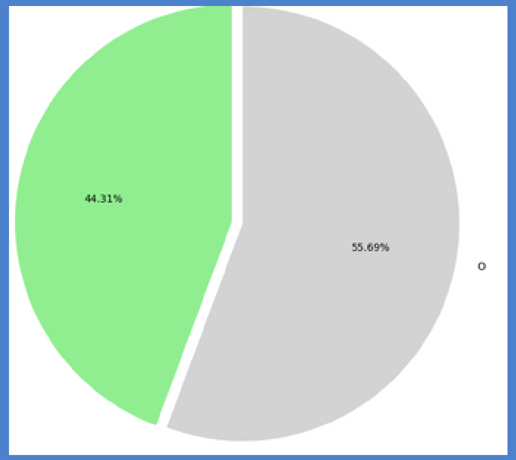
Data set

In Data set we have images of the object which are degradable or non degradeable

Dataset Link: <u>Kaggle Waste Classification Data</u>

• Source: Kaggle Waste Classification Dataset
• Features: Images of waste items

• Labels:
– Degradable (1)
– Non-Biodegradable (0)



• Train Folder Composition:
– Organic (O)
– Recyclable (R)
• Total Images: 22,564

Methodology

- Data Collection: Gather a diverse dataset of waste images.
- Data Preprocessing: Clean and standardize the images.
- Model Selection: Choose a pre-trained CNN model.
- Fine-tuning: Adapt the model to waste classification.
- Training: Train the model on the preprocessed dataset.
- Evaluation: Assess model performance using metrics.
- Optimization: Refine model parameters and architecture.
- Deployment: Implement the model for automated waste classification.
- Testing and Validation: Verify system performance with new data.
- Feedback Incorporation: Continuously improve the system based on user feedback

MODELS TRIED

LDA

Accuracy : 32%
class wise accuracy
class 0 : 18%
class 1 : 44%

Decision TREES

Accuracy : 44%
class wise accuracy
class 0 : 11%
class 1 : 70%

CNN

Accuracy : 88%
class wise accuracy
class 0 : 63%
class 1 : 37%

DATASETS USED

Fliping around y_axis

Fliping around x_axis

CREATED NEW DATASETS USING DATA AUGEMENTATION

Random rotation

Adding Noise

Experimental settings, Results, Comparisons

How We Reached the Proposed Architecture:

- Researched existing image classification projects, finding CNNs widely used and effective.
- Experimented with simpler architectures but found them inadequate for capturing complex waste features.

Dataset Preprocessing:

- Grayscale Conversion: Converted images to grayscale for simplicity and reduced computational complexity.
- Resizing: Standardized all images to 64x64 pixels to facilitate machine learning.
- Preprocessed images serve as input for further analysis and model training.

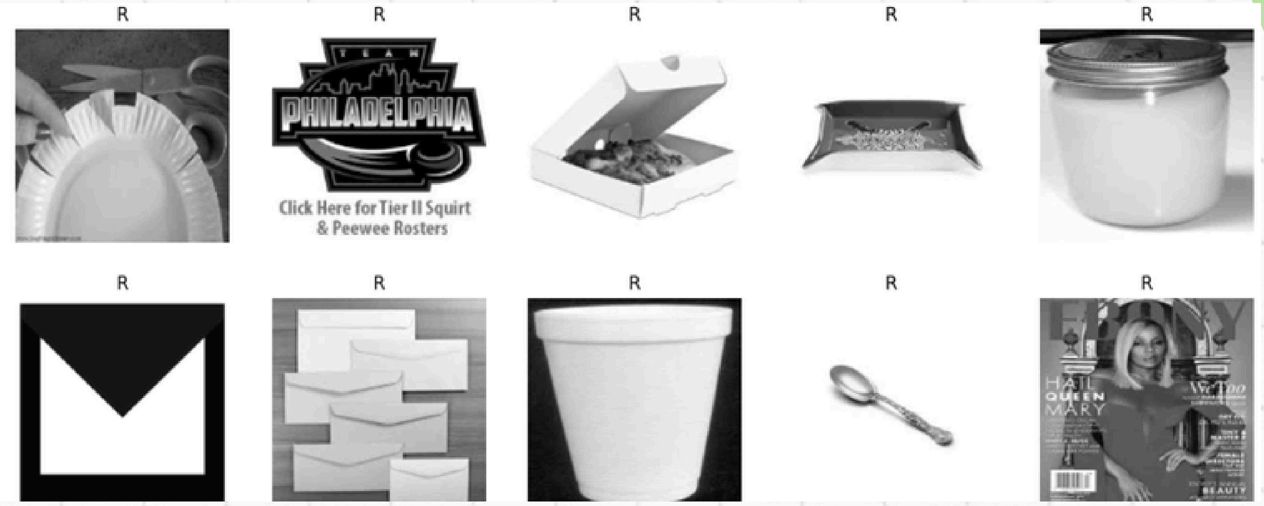
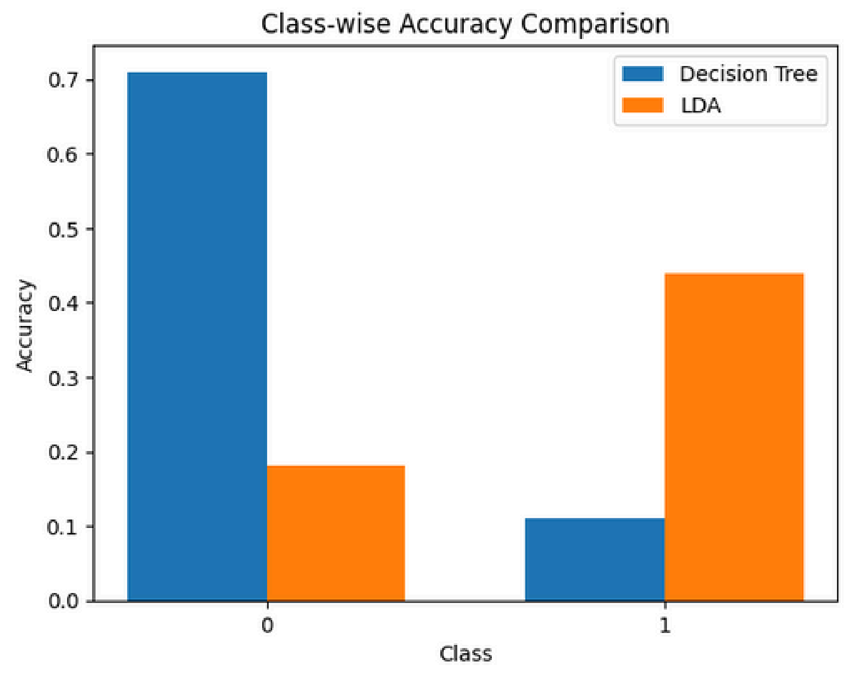


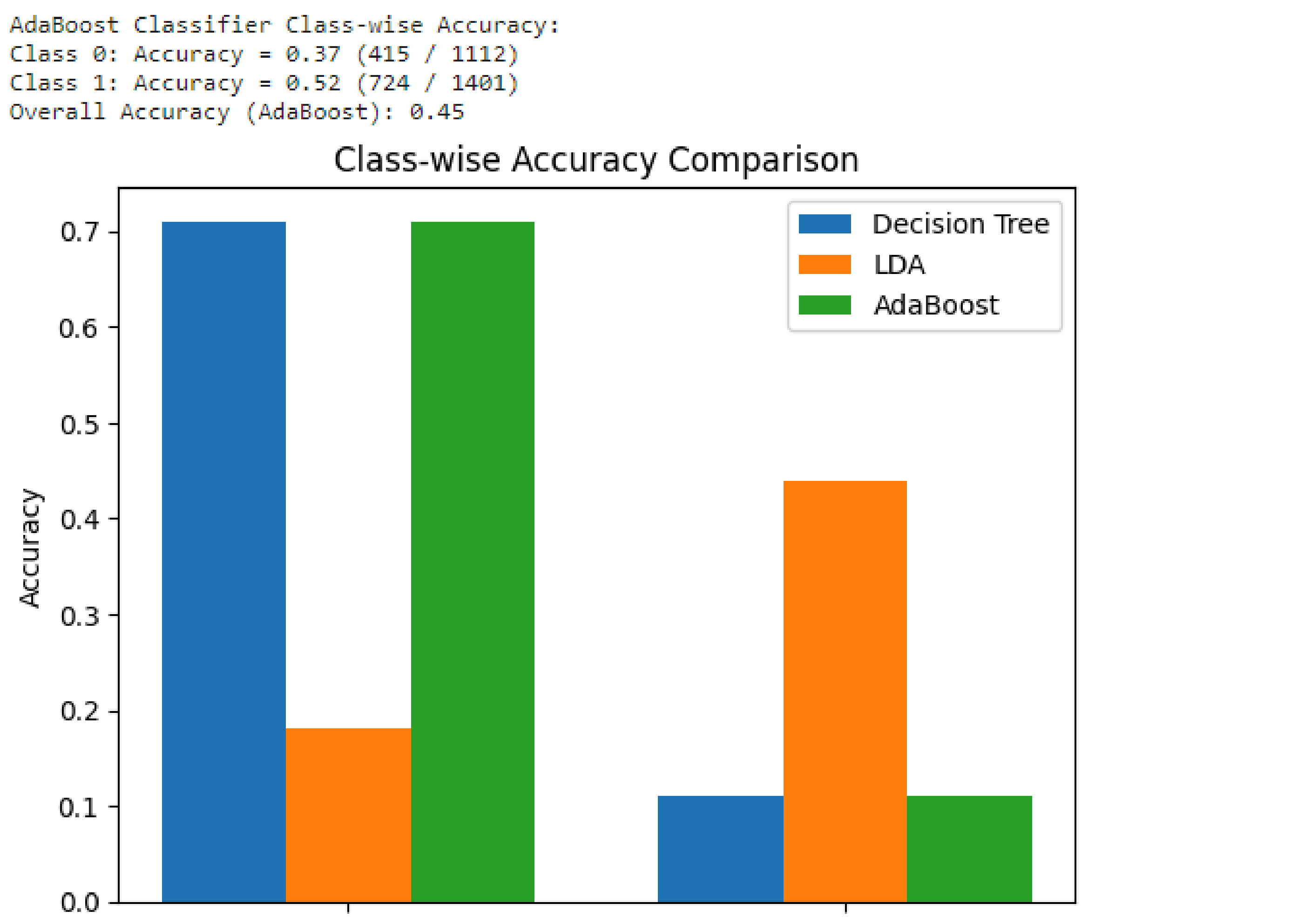
Figure 1: Sample grayscale image of Non-Biodegradable (Recyclable) waste

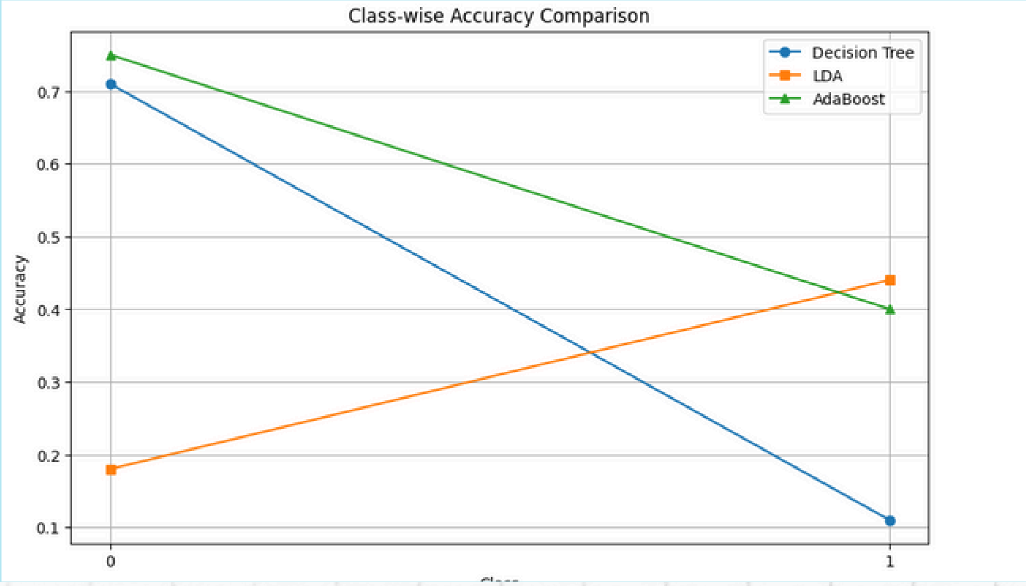
Figure 2: Sample grayscale image of Biodegradable (Organic) waste

Compare between differnt model we use :



comparison between the Decision tree and LDA





classwise accuracy of the LDA and Decision Tree

class wise accuracy of the Decision tree , LDA and Adaboost

Motiation

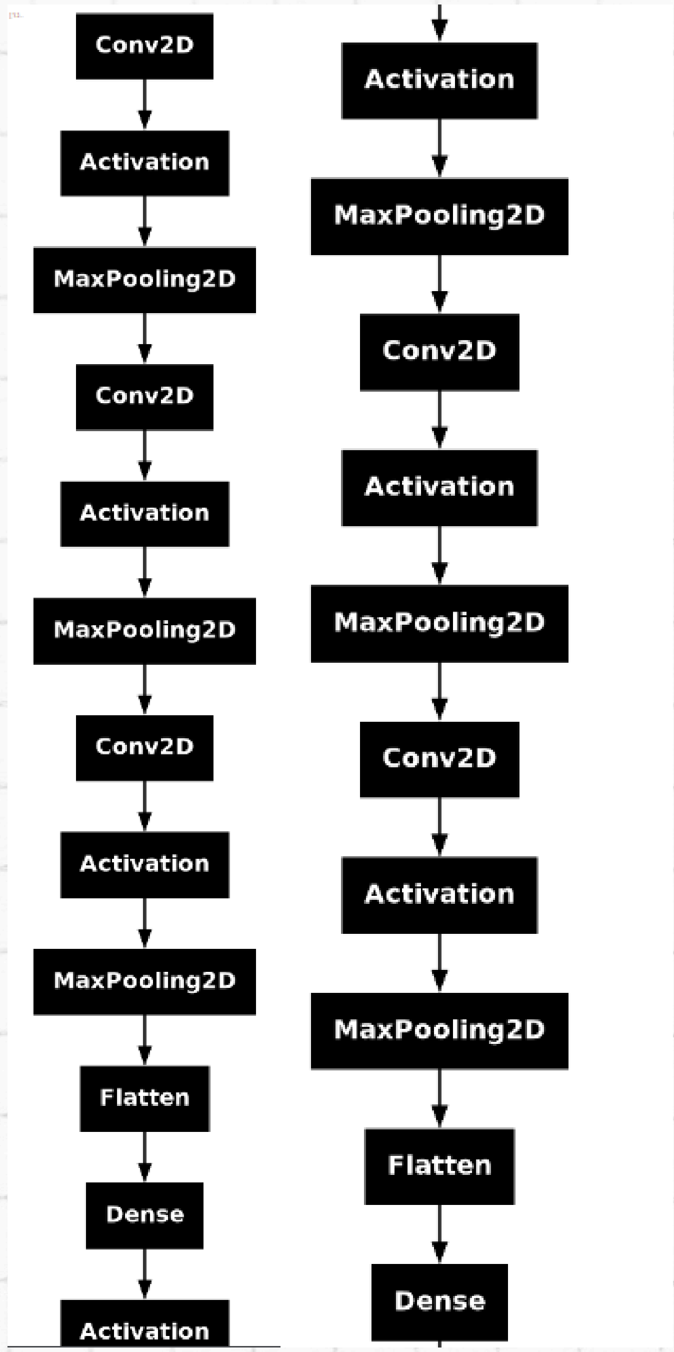
Why we need to use the CNN :

- CNNs excel in image processing due to:
  - Automatic hierarchical feature learning.
  - Translation invariance.
  - Parameter efficiency.
  - Handling complexity of image data.
- They perform well on large datasets.
- CNNs preserve spatial relationships in images.
- This makes them superior to LDA and Decision Trees for image processing tasks.

Advance version of CNN by useing VGG16 model :

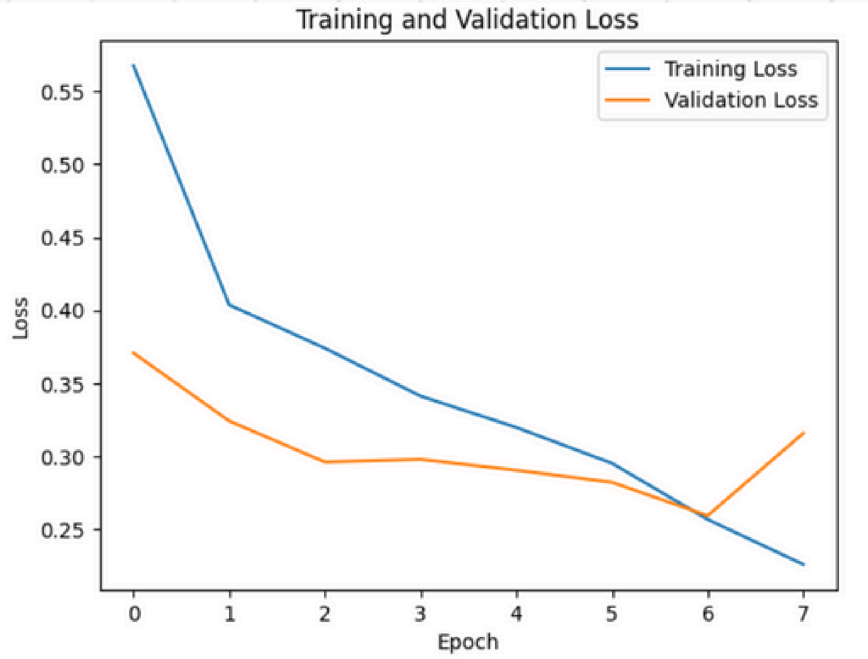
- VGG16 Architecture
- VGG16, as its name suggests, is a 16-layer deep neural network. VGG16 is thus a relatively extensive network with a total of 138 million parameters—it's huge even by today's standards. However, the simplicity of the VGGNet16 architecture is its main attraction.
- The VGGNet architecture incorporates the most important convolution neural network features.

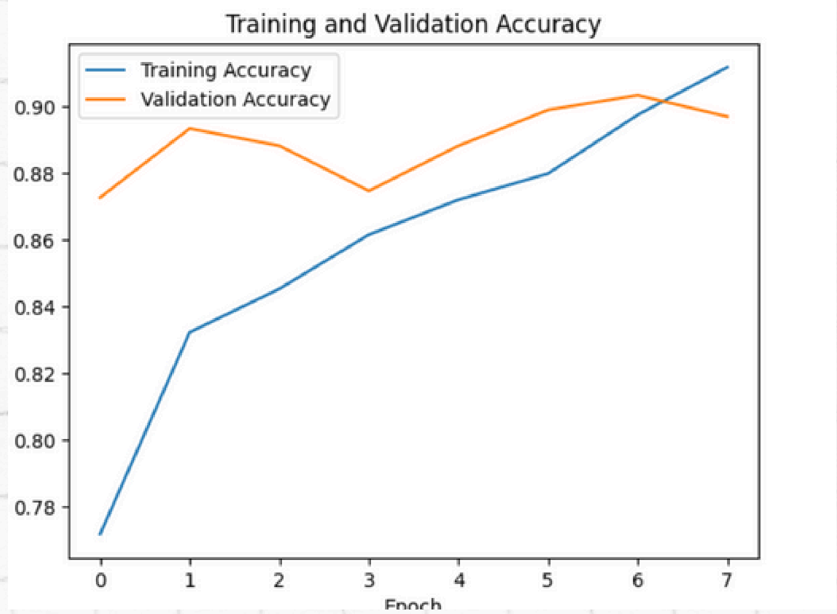
OUTPUT: BY USING THE CNN MODEL:



These step is used in to get the CNN Model to get work

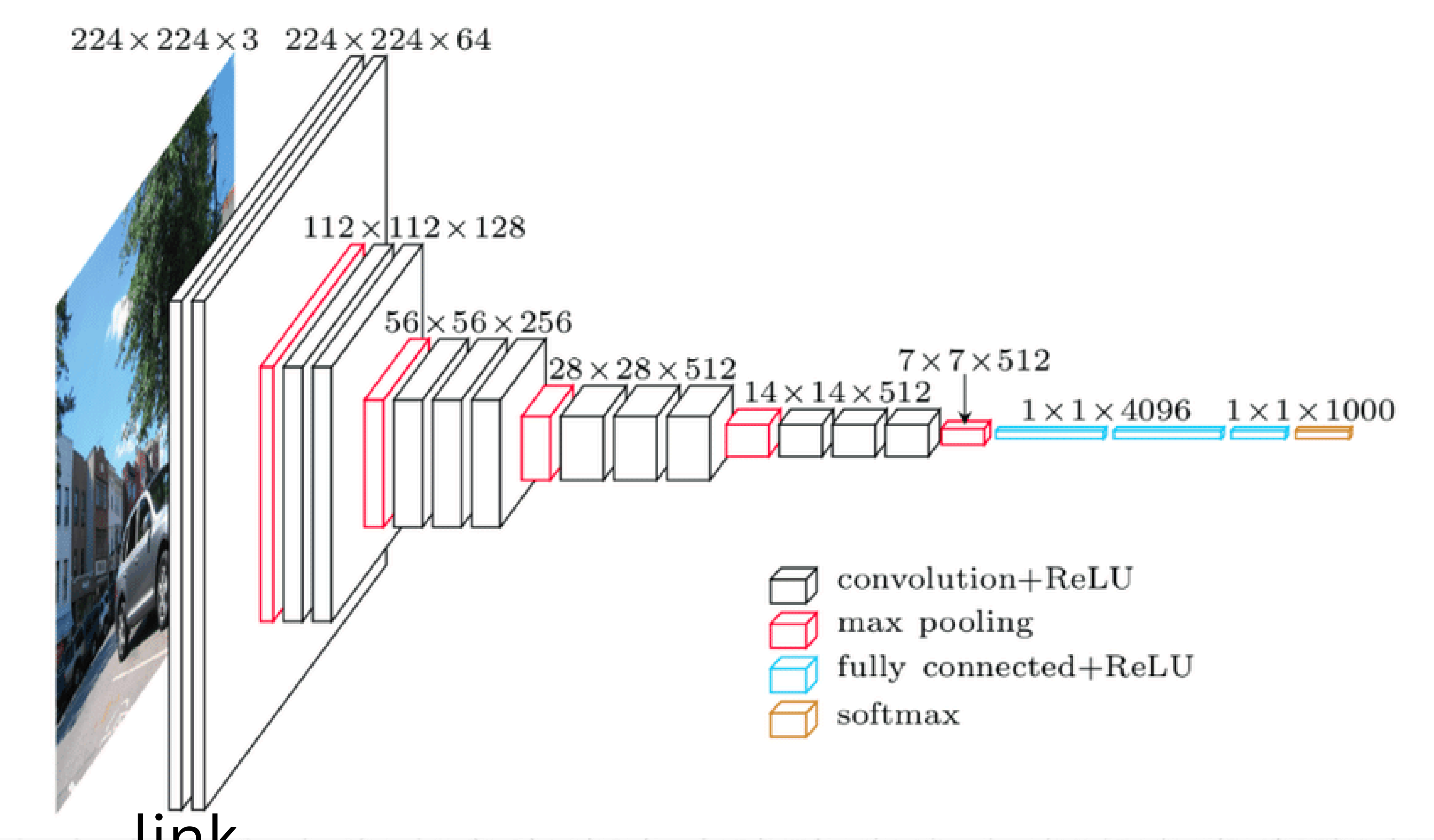
```
Num GPUs Available:  2
Found 22564 images belonging to 2 classes.
Found 2513 images belonging to 2 classes.
Epoch 1/8
W0000 00:00:1715545121.264930     112 graph_launch.cc:671] Fallback to op-by-op mode because memset node breaks graph update
89/89 ━━━━━━━━━━━━━━━━━━━━ 0s 698ms/step - accuracy: 0.6979 - loss: 0.8573
W0000 00:00:1715545185.717537     111 graph_launch.cc:671] Fallback to op-by-op mode because memset node breaks graph update
89/89 ━━━━━━━━━━━━━━━━━━━━ 143s 995ms/step - accuracy: 0.6988 - loss: 0.8541 - val_accuracy: 0.8727 - val_loss: 0.3708
Epoch 2/8
W0000 00:00:1715545209.116563     110 graph_launch.cc:671] Fallback to op-by-op mode because memset node breaks graph update
89/89 ━━━━━━━━━━━━━━━━━━━━ 61s 634ms/step - accuracy: 0.8280 - loss: 0.4116 - val_accuracy: 0.8934 - val_loss: 0.3243
Epoch 3/8
89/89 ━━━━━━━━━━━━━━━━━━━━ 60s 631ms/step - accuracy: 0.8491 - loss: 0.3746 - val_accuracy: 0.8882 - val_loss: 0.2963
Epoch 4/8
89/89 ━━━━━━━━━━━━━━━━━━━━ 61s 642ms/step - accuracy: 0.8623 - loss: 0.3416 - val_accuracy: 0.8747 - val_loss: 0.2981
Epoch 5/8
89/89 ━━━━━━━━━━━━━━━━━━━━ 61s 637ms/step - accuracy: 0.8692 - loss: 0.3251 - val_accuracy: 0.8882 - val_loss: 0.2907
Epoch 6/8
89/89 ━━━━━━━━━━━━━━━━━━━━ 61s 633ms/step - accuracy: 0.8843 - loss: 0.2876 - val_accuracy: 0.8989 - val_loss: 0.2824
Epoch 7/8
89/89 ━━━━━━━━━━━━━━━━━━━━ 61s 634ms/step - accuracy: 0.8944 - loss: 0.2603 - val_accuracy: 0.9033 - val_loss: 0.2596
Epoch 8/8
89/89 ━━━━━━━━━━━━━━━━━━━━ 60s 630ms/step - accuracy: 0.9143 - loss: 0.2246 - val_accuracy: 0.8969 - val_loss: 0.3160
```






```
Epoch
10/10 ━━━━━━━━━━━━━━━━━━━━ 6s 584ms/step - accuracy: 0.8948 - loss: 0.3202
Overall Accuracy: 0.8969359397888184
 1/10 ━━━━━━━━━━━━━━━━━━━━ 9s 1s/step
W0000 00:00:1715545644.162327     109 graph_launch.cc:671] Fallback to op-by-op mode because memset node breaks graph update
10/10 ━━━━━━━━━━━━━━━━━━━━ 6s 571ms/step
Class O: Accuracy = 0.63
Class R: Accuracy = 0.37
W0000 00:00:1715545649.296604     109 graph_launch.cc:671] Fallback to op-by-op mode because memset node breaks graph update
```

How the VGG16 is work : Advance version of CNN

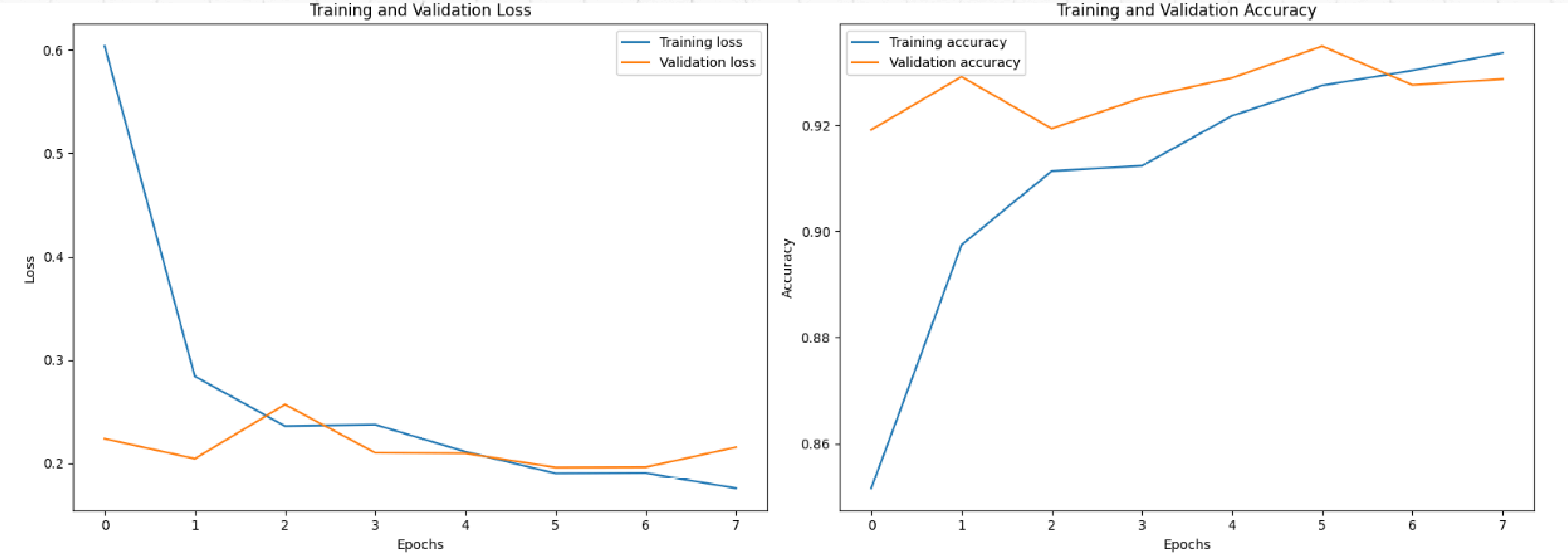


link

THE VGGNET ARCHITECTURE INCORPORATES THE MOST IMPORTANT CONVOLUTION NEURAL NETWORK FEATURES.

OutPUT: VGG16 is used to get overcome with CNN accuracy

Desirable output for loss and Accuracy for validation data and train data as show in the graph



[12]:

```
evaluation = model.evaluate(test_generator)
print(f'test Loss: {evaluation[0]:.4f}')
print(f'test Accuracy: {evaluation[1] * 100:.2f}%')
```

```
20/20 ━━━━━━━━━━━━━━━━━━━━ 52s 3s/step - accuracy: 0.8894 - loss: 0.3687
test Loss: 0.3607
test Accuracy: 89.22%
W0000 00:00:1715542509.721553     112 graph_launch.cc:671] Fallback to op-by-op mode because mem
breaks graph update
```

Preminariy Accuracy

by using the decision Tree and applies adaBoost

45%

Accuracy

Final Accuracy

by Using the VGG16

89%

Accuracy

Conclusion:

In summary, CNNs emerge as the preferred choice for automated waste classification due to their inherent capabilities in feature learning, robustness to image variations, parameter efficiency, and performance on large datasets. Their ability to handle image complexity and preserve spatial relationships makes them indispensable for accurate and efficient waste management practices.

Thank you very much!