**Group Members**

**Amish Faldu (af557)**

**Sneh Vora (sv992)**

**Palak Pabani (pp872)**

**Introduction**

Waste management is a critical environmental concern worldwide. Effective waste classification plays a significant role in efficient recycling and disposal processes. Traditional methods of waste sorting are often labor-intensive and time-consuming. This project aims to develop a deep learning model to automate waste classification tasks, specifically focusing on classifying images of different types of waste.

**Dataset**

The dataset we’ve worked on is Real Waste Dataset from UCI. This dataset contains 4752 image instances with 9 different waste categories. The image dimensions are 524 x 524 x 3 (524 Height, 524 Width and 3 RGB channels). The label distribution for the dataset looks like below.

* Cardboard: 461 images
* Food Organics: 411 images
* Glass: 420 images
* Metal: 790 images
* Miscellaneous Trash: 495 images
* Paper: 500 images
* Plastic: 921 images
* Textile Trash: 318 images
* Vegetation: 436 images

A graph of blue rectangular bars with white text

Description automatically generated

The dataset was standard scaled before feeding to the neural network for faster convergence.

**Evaluation Metrics**

Model performance was evaluated using two primary metrics: loss and F1 score. The loss function used was categorical cross-entropy loss, which measures the dissimilarity between predicted and actual class distributions. The F1 score, a harmonic mean of precision and recall, provides a balanced assessment of model classification performance across all classes.

**Experimentation & Contributions**

We trained 4 different CNN models. The dataset was split into 2 halves: 80% training and 20% validation set. We then trained the models using Adamax optimizer, CrossEntropy loss function and Learning Rate Scheduler.

For the transfer learning, we first removed the top layer of the pre-trained model and added new dense layer. Initially we have trained the only top layer for 10 epochs, then fine-tuned the model by unfreezing some of the pre-trained model’s layers with reduced learning rate for another 10 epochs.

The CNN architectures trained are:

1. Custom architecture (Our own model) – We built and trained our own custom model.
2. VGG 19 architecture – This was not previously explored in Kaggle notebooks.
3. Inception V3 – Transfer learning by top layer training and further finetuning.
4. MobileNets V1.0 – This was not previously explored in Kaggle notebooks.

We also added **Step** **Learning Rate Scheduling** that is different from Kaggle notebook and added **Model Checkpoints based on validation loss**, so when the validation loss improves the model’s weights are stored in a file.

**Results**

A chart of different types of waste

Description automatically generated

Custom CNN Model Confusion Matrix

**A chart of different types of objects

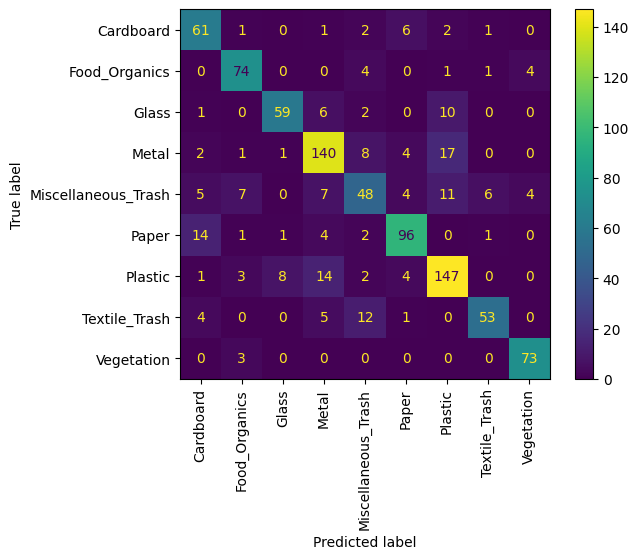
Description automatically generated**

Inception V3 Confusion Matrix

**A chart of different types of objects

Description automatically generated**

MobileNets Confusion Matrix



VGG 19 Confusion Matrix

**Comparison Table**

|  |  |
| --- | --- |
| **Model** | **Validation F1-Score** |
| Custom CNN architecture (30 epochs) | 0.6598 |
| VGG 19 (20 epochs) | 0.7905 |
| Inception V3 (20 epochs) | **0.8947** |
| MobileNets V1.0 (20 epochs) | 0.8464 |

**Conclusion**

In conclusion, the Inception V3 was highest scoring CNN architecture, but also slowest compared to MobileNets V1.0 because MobileNets uses Depthwise Separable Convolution Layers compared to Standard Convolution layers in other models.

**References**

**Dataset Link -** <https://archive.ics.uci.edu/dataset/908/realwaste>

**Code Link –** <https://drive.google.com/drive/folders/1UnGzeVwuwI4ap-v0p4m4ro1qXug3mnD_?usp=share_link>

**Video Link –**