Garbage Image Classification using Deep Learning

By

SHUBHAM VEDPRAKASH MISHRA

(Enrollment No. : 220280702010)

Guided by

PROF. PARTH R. DAVE

Assistant Professor, Department of Computer Engineering,

L.D. College of Engineering, Ahmedabad

A Thesis Submitted To

Gujarat Technological University in Partial Fulfilment of the Requirements for the Master of Engineering Degree in

COMPUTER ENGINEERING

Computer Engineering Department

L.D. College of Engineering, Ahmedabad

(Affiliated to Gujarat Technological University)

May, 2024

ACKNOWLEDGMENT

I am very grateful to the Principal of L. D. College of Engineering for providing facilities to achieve the desired milestone.

I also extend my thanks to Head of Department Dr. Chirag S. Thaker for his inspiration and continuous support.

I wish to warmly thank my guide, Prof. Parth R. Dave for his inspiration and continuous support. Without his valuable advice and assistance, it would not have been possible for me to attain this landmark. He has always been willingly present whenever I needed the slightest support from him. I would not like to miss a chance to say thank you for the time that he spared for me, from her extremely busy schedule.

I would like to thank all of them whose names are not mentioned here but have played a significant role in my way to accomplish the work.

Last but not least, I would thank almighty God and especially my parents for always being there believing, putting the faith in me and bringing about the strength to complete this entire process.

Shubham Mishra

(220280702010)

INDEX

List Of Figures

ABSTRACT

In this study focused on garbage image classification for recycling purposes, the fundamental task of identifying single garbage objects in images and classifying them into recycling categories is addressed. We have chosen ResNet-50 as our base model . Leveraging deep learning techniques, particularly CNN-based methods, proves effective in addressing the garbage classification challenge for the target database, highlighting the potential of deep learning in enhancing recycling efforts through efficient image classification.This study proposes a novel approach to address shortcomings in garbage image classification using deep learning, focusing on enhancing the ResNet-50 network architecture. The improvements are :(1).Attaching a CBAM module to better filter input features(2)Modifying Downsampling to decrease information loss (3).Attaching Multi-Scale Feature Fusion. . These modifications enable the model to achieve superior classification performance, particularly on small datasets with limited samples. Experimental results on the TrashNet dataset demonstrate a significant improvement of 7.62% over the original ResNet-50 model, showcasing increased robustness. Moreover, comparative analysis highlights the superior accuracy of the proposed model compared to other state-of-the-art methods in garbage image classification.

Keywords: ResNet-50,Convolutional Neural Network,CBAM,Garbage Classification

CHAPTER-1
INTRODUCTION

1.1 INTRODUCTION

Nowadays, the whole world is producing billions of tons of garbage or waste.Such huge production of the garbage damages the environment.Management of these garbages is must.Recycling is a way to manage the these waste.Sorting of these garbage manually is difficult and costly process.To make recycling speedy and cost-effective , we need to automate this task.

This thesis focuses on the application of deep learning techniques to automate the classification of garbage images, aiming to improve waste sorting processes and enhance recycling initiatives. Traditional methods of waste sorting are labor-intensive, error-prone, and often inefficient, leading to suboptimal recycling rates and environmental degradation. By harnessing the power of deep learning algorithms, we seek to develop efficient and accurate systems for automated garbage classification, thereby facilitating more sustainable waste management practices.

From the past couple of decades ,deep learning has proved to be effective for tasks like computer vision like image classification,image segmentation and object detection.Various models were created for Image Classification.Among various models, CNN Models proved to be more efficient and accurate than rest of the other models.CNNs are easy to train and can be trained using few parameters

Initially, no dataset was available for garbage classification, but in 2016 Mindy Yang and Gay Thug created the “TRASHNET DATASET” for garbage image classification.After this dataset,major work started happening in this domain.

Generally, Deeper Neural Networks are used for images because they better capture the features.But the size of the dataset is small,so we will try to modify the structure of the architecture of CNN model to achieve a better accuracy.In this thesis, we will explore this idea in detail.

1.2 MOTIVATION

The exponential increase in waste production globally has led to pressing concerns regarding environmental pollution, resource depletion, and public health hazards. As urban populations burgeon and consumption patterns evolve, traditional waste management systems struggle to cope with the sheer volume and diversity of waste generated. Manual waste sorting processes, reliant on human labor and prone to errors, are not only inefficient but also contribute to environmental degradation

In light of these challenges, there is a critical need for innovative solutions that can automate and optimize waste sorting processes. Deep learning, a subset of artificial intelligence, has emerged as a powerful tool for image classification tasks, with applications ranging from medical diagnosis to autonomous driving. Leveraging deep learning techniques for garbage image classification presents a compelling opportunity to revolutionize waste management practices.

So, to sum it up following are the main factors of the motivation:

1. Improve Efficiency and Accuracy for Garbage Classification
2. Reduce Human Labor
3. Environmental Conservation
4. Scalability

1.3 OBJECTIVE

The primary objective of the paper is to develop an Garbage classification model by modifying the structure of the ResNet-50 network. This modification aims to address the challenges posed by small datasets with limited samples, where conventional models may struggle to achieve satisfactory performance.

In this research, we are trying to attach an attention mechanism to residual blocks. Attention mechanisms allow the model to focus on important features while suppressing irrelevant or noisy information .We will also try to incorporate a multi-scale feature fusion to utilize information at different spatial resolutions, enhancing its capacity to capture diverse and complex features present in garbage images.

1.4 PROBLEM STATEMENT

In the classic ResNet-50 ,1*1 convolution kernel having step size of 2 is used for downsampling processes which ultimately results in the information loss.This thing can be fixed to avoid excess information loss.

Attention mechanisms like CBAM can be integrated in CNN models to filter the irrelevant features and emphasize focus on important details.It can help to improve generalization and robustness of the model.

The dataset is small, robustness and generality is still an issue.To mitigate this problem, multi-scale feature fusion can be used.

There are two sub modules in it:
(1).Channel Attention Module(CAM)
(2).Spatial Attention Module (SAM)

CHAPTER 2

LITERATURE SURVEY

2.1 Review of Papers

2.1.1.Research on Garbage Image Classification Based on Convolutional Neural Network

[1] Title	:	Research on Garbage Image Classification Based on Convolutional Neural Network
Author	:	Kangjian Tang, Zhan Wen ,Yahui Chen, Wenzao Li
Publication	:	International Journal of Computer Applications Technology and Research
Published Date	:	2019
Dataset	:	Huawei Garbage Dataset

Observation:

The paper addresses the issue of garbage classification and proposes the use of deep learning, specifically Convolutional Neural Networks (CNNs), to improve accuracy and efficiency in waste classification.

Two types of CNNs, Inception V3 and Inception V4, were trained on Huawei's public garbage dataset, and their performance was compared.

The results showed that both models achieved higher accuracy in garbage classification, with Inception V4 being more stable and accurate than Inception V3.

The accuracy rate of Inception V4 is

97.37%, and the accuracy rate of Inception V3 is **89.2%**

2.1.2 X-DenseNet -Deep Learning for Garbage Classification Based on Visual Images.

[2] Title	:	X-DenseNet -Deep Learning for Garbage Classification Based on Visual Images.
Author	:	Sha Meng, Ning Zhang and Yunwen Ren
Publisher	:	Journal of Physics: Conference Series
Publication Year	:	2020
Dataset	:	Self Prepared

Observation:

The paper proposes a garbage classification model called X-DenseNet, which is based on deep convolutional neural networks and incorporates the ideas of dense connections and multi-scale feature fusion from DenseNet.

The X-DenseNet model is constructed using the Xception network as the basic structure, which is an improvement of InceptionV3, and includes the ResNet residual network mode to improve convergence speed.

The paper conducts experiments by obtaining a dataset, preprocessing the data, building the X-DenseNet model, and training and testing the model. The accuracy of the model on the testing set is reported to be **94.1%**, surpassing some classic classification networks.

In the experiment, the paper compares X-DenseNet with other models such as AlexNet, ResNet50, InceptionV3, Vgg16, and Vgg19, and presents the classification accuracy results.

2.1.3 Image Recognition for Garbage Classification Based on Transfer Learning and Model Fusion

[3] Title	:	Image Recognition for Garbage Classification Based on Transfer Learning and Model Fusion
Author	:	Wei Liu , Hengjie Ouyang , Qu Liu , Sihan Cai , Chun Wang , Junjie Xie , and Wei Hu
Publisher	:	HINDAWI
Publication Year	:	2022
Dataset	:	Trash Net

Observation:

The paper proposes a novel garbage image recognition model called Garbage Classification Net (GCNet) that combines transfer learning and model fusion Pretrained models, including EfficientNetv2,Vision Transformer, and DenseNet, are used to extract garbage image features

Data augmentation is employed to expand the dataset, resulting in 41,650 garbage images The models are fused using a voting fusion technique to improve the generalization ability and learn the differences between garbage categories.

The fused model GCNet achieved a higher accuracy of 97.54% on the test set compared to individual models.

The proposed model showed good convergence, high recall rate and accuracy, and short recognition time.

Transfer learning using pretrained models greatly improved results.

2.1.4 Depth-Wise Separable Convolution Attention Module for Garbage Image Classification

[4] Title	:	Depth-Wise Separable Convolution Attention Module for Garbage Image Classification
Author	:	Fucong Liu, Hui Xu ,Miao Qi, Di Liu, Jianzhong Wang and Jun Kong
Publisher	:	MDPI
Publication Year	:	2023
Dataset	:	These datasets are constructed by Huawei Cloud and Baidu AI Studio.

Observation:

The paper addresses the issue of garbage image classification and proposes a Depth-Wise Separable Convolution Attention Module (DSCAM) to overcome the limitations of existing methods.

DSCAM captures the inherent relationships of channels and spatial positions in garbage image features using attention modules with depth-wise separable convolutions. It focuses on important information and ignores interference, improving classification accuracy

The DSCAM method is evaluated on five garbage datasets, including Huawei-40, Baidu-214, Baidu-RC, and the newly generated BR-124 dataset.

Experimental results demonstrate that DSCAM effectively classifies garbage images and outperforms classical methods, including VGG-19, Xception, X-DenseNet, MobileNet-V3, and GNet

2.1.5 Garbage image recognition and classification based on hog feature and SVM-Boosting

[5] Title	:	Garbage image recognition and classification based on hog feature and SVM-Boosting
Author	:	WANG Weifeng,ZHANG Baobao,WANG ZhiQiang,ZHANG FangZhi , LIU Qiang
Publisher	:	Journal of Physics: Conference Series
Publication Year	:	2021
Dataset	:	Self Prepared using Smartphone camera

Observation:

The paper proposes a method that combines hog features and boosting algorithm to improve the recognition efficiency of SVM in garbage image recognition and classification.

The input image is preprocessed to enhance recognition, and the hog algorithm is used to extract image features

A SVM classification device is trained using the extracted features, and the classification situation is detected based on the trained device

The experimental results show that the proposed method achieves a classification efficiency of 95% or even higher, which is about 10% higher than that of the single SVM classification method; the accuracy of the experiment is compared using SVM classifier and SVM boosting classifier.

2.1.6 A Study of Garbage Classification with Convolutional Neural Networks

[6] Title	:	A Study of Garbage Classification with Convolutional Neural Networks
Author	:	Shanshan Meng ,Wei-ta chu
Publisher	:	IEEE
Publication Year	:	2020
Dataset	:	Garbage Classification Dataset

Observation:

Support vector machines (SVM) with HOG features were employed for garbage classification

Basic CNN which consisted of 2D convolutional layers with 3x3 filters, max pooling layers, a flatten layer, and fully connected layers for classification were used

CNN models with residual blocks, such as the ResNet50 architecture, were utilized.

These models included two conv. layers with a 1x1 filter before and after a 3x3 conv. layer, creating a bottleneck structure

The paper compares different techniques for garbage classification using deep learning models, including support vector machines (SVM) with HOG features, basic convolutional neural networks (CNN), and CNN with residual blocks

The SVM-based approach achieves a test accuracy of approximately 47.25% using the same training and test sets

The combination of CNN models, with or without residual blocks, performs well for garbage classification

2.1.7 Automatic Image-Based Waste Classification

[7] Title	:	Automatic Image-Based Waste Classification
Author	:	Victoria Ruiz 1, Angel S´ ´ anchez1(B) , Jos´e F. V´elez1, and Bogdan Raducanu 2
Publisher	:	Lecture Notes in Computer Science
Publication Year	:	2019
Dataset	:	TrashNet

Observation:

Many Convolutional Neural Network (CNN) architectures, including VGG, Inception, and ResNet, are compared .

The networks' weights are initialized randomly, and a batch size of 16 samples is used with Stochastic Gradient Descent (SGD) as the optimization algorithm .

Early stopping is implemented, and batch normalization layers are introduced at the end of each block of convolutional layers .The best classification results are achieved using a combined Inception-ResNet model, with an accuracy of 88.6%.

The ResNet model achieves the highest accuracy of 88.66% and is the most stable with the smallest standard deviation .The Inception-ResNet model produces similar results to the ResNet model .The ResNet model requires fewer epochs to be trained compared to other models

2.1.8 Deep Learning based Smart Garbage Classifier for Effective Waste Management

[8] Title	:	Deep Learning based Smart Garbage Classifier for Effective Waste Management
Author	:	Sidharth R, Rohit P, Vishagan S, Karthika R, Ganesan M
Publisher	:	Fifth International Conference on Communication and Electronics Systems
Publication Year	:	2020
Dataset	:	TrashNet

Observation:

A simple CNN on a labelled dataset is used.

Softmax activation function for output layer.

TrashNet dataset is used here.Achieved a testing accuracy of 76% using supervised learning.

Maximum accuracy achieved is 76% at 100 Epochs and image size of 50x50.

Theclassifier can be improved with a better dataset and more images.

The proposed model can be automated for practical applications in households

2.1.9 New Benchmark for Household Garbage Image Recognition

[9] Title	:	New Benchmark for Household Garbage Image Recognition
Author	:	Zhize Wu, Huanyi Li, Xiaofeng Wang* , Zijun Wu, Le Zou, Lixiang Xu, and Ming Tan
Publisher	:	ELSEVIER
Publication Year	:	2021
Dataset	:	HGI-30 Dataset

Observation:

The paper introduces the 30 Classes of Household Garbage Images (HGI-30) dataset having 18000 images.

CNN is applied on the HGI-30 and results were compared with SIFTBoVW, HOG SVM, MobileNet, VGG16, ResNet50, DenseNet, XCeption, and EfficientDet

The average mAP(Mean Average precision) of the six target detection algorithms in this dataset was 76%.

Overall,YOLOv4 achieved the best effect, having an average detection precision of **96%.**

2.1.10 CBAM: Convolutional Block Attention Module

[10] Title	:	CBAM: Convolutional Block Attention Module
Author	:	Sanghyun Woo,Jongchan Park ,Joon-Young Lee ,In So Kweon
Publisher	:	KAIST, Lunit, and Adobe Research.
Publication Year	:	2018
Dataset	:	ImageNet-1K, MS COCO, and VOC 2007

Observation:

The CBAM module greatly improves the performance of various networks on multiple benchmarks.

CBAM is a descendant of the Squeeze and Excitation Network

The CBAM-integrated network (ResNet50 CBAM) shows improved visualization results compared to the baseline and SE-integrated networks.

Object detection mAP is boosted by 0.9 for both baseline networks when CBAM is applied.

2.2 SUMMARY OF REVIEW PAPERS

Sr No	Paper Title	Method and Dataset	Results/Conclusion
[1]	Research on Garbage Image Classification Based on Convolutional Neural Network	Inception V3 and Inception V4 convolutional neural networks are used Huawei's public garbage dataset (Garbage Date) was used for training	The Inception V4 model is more stable and accurate than Inception V3. The accuracy rate of Inception V4 is 97.37%, and the accuracy rate of Inception V3 is 89.2%
[2]	X-DenseNet-Deep Learning for Garbage Classification Based on Visual Images	It combines Xception network with dense connections and multi-scale feature fusion The paper uses a dataset of 150 x 150 color garbage pictures.	X-DenseNet model achieves **94.1%** accuracy in garbage classification. Reduces manual investment and improves garbage recovery rate
[3]	Image Recognition for Garbage Classification Based on Transfer Learning and Model Fusion	Proposed model GCNet for garbage image recognition using transfer learning and model fusion. Combined EfficientNetv2, Vision Transformer, and DenseNet for neural network model. Dataset contains 41,650 garbage images with data augmentation and is obtained from Crawler[Self Created dataset]	The fused model GCNet has a higher accuracy **(97.54)** compared to single models DenseNet **(96.40)**, Vision Transformer **(96.75)**, and EfficientNetV2 **(96.12)** on the test set. The proposed model, GCNet, showed good convergence, high **recall rate** and accuracy, and short recognition time Improves generalization

Sr No	Paper Title	Paper Title	Method and Dataset
[4]	Depth-Wise Separable Convolution Attention Module for Garbage Image Classification	Depth-Wise Separable Convolution Attention Module for Garbage Image Classification	➔ The paper proposes a DepthWise Separable Convolution Attention Module (DSCAM) ➔ It adopts a residual network as the backbone of DSCAM. ➔ Baidu garbage dataset (Baidu214),Huawei dataset are us
[5]	Garbage image recognition and classification based on hog feature and SVM-Boosting		The paper combines hog features and boosting algorithm to develop a SVM classification method for garbage image recognition and classification ➔ The experimental samples are divided into categories A, B, C, and D, and six SVM classifiers are designed ➔ The experiment uses SVM classifier and SVM boosting classifier for comparison and evaluation of accuracy ➔ Image Dataset is Self Prepared using Smartphone
[6]	Garbage Classification Using Deep Learning Techniques		➔ Support vector machines (SVM) with HOG features were employed for garbage classification ➔ Basic CNN which consisted of 2D convolutional layers with 3x3 filters, max pooling layers, a flatten layer, and fully connected layers for classification were used ➔ CNN models with residual blocks, such as the ResNet50 architecture, were utilized. These models included two conv. layers with a 1x1 filter before and after a 3x3 conv. layer, creating a bottleneck structure

SR No.	Paper Title	Method and Dataset	Method and Dataset
[7]	Automatic ImageBased Waste Classification		➔ Various architectures of CNN like VGG,Inception and ResNet is used. ➔ The best classification results were obtained using a combined Inception-ResNet model with an accuracy of 88.6%. ➔ Image analysis techniques such as statistical moments, Fourierbased descriptors, Gabor-based descriptors, and Histogram Oriented Gradients (HOG). ➔ TrashNet Dataset having 2567 images was used
[8].	Deep Learning based Smart Garbage Classifier for Effective Waste Management		➔ A simple CNN on labelled dataset is used. ➔ Softmax activation function for output layer. ➔ TrashNet dataset is used here.
[9].	New Benchmark for Household Garbage Image Recognition		➔ Dataset Household Garbage Images (HGI-30) is created by simulating different lightings, backgrounds, angles, and shapes and having 30 classes. ➔ CNN is applied on the HGI-30 and results were compared with SIFTBoVW, HOG SVM, MobileNet, VGG16, ResNet50, DenseNet, XCeption, and EfficientDet
[10].	Research on deep learning image recognition technology in garbage classification		➔ Model based on EfficientNet is used ➔ Group normalization (GN) is used instead of batch normalization (BN) to normalize the features within each group, as GN has stable accuracy under various batch sizes and reduces the error rate .
[11]	CBAM:Convolutional Block Attention Module		The paper introduces the CBAM module as a simple and effective attention mechanism for CNNs, enhancing network performance on various benchmarks The CBAM module combines both channel and spatial attention, with the sequential arrangement of two attention modules proving more effective than a parallel arrangement

			The authors conducted experiments to validate the effectiveness of the CBAM module on different datasets, showcasing its impact on improving accuracy and performance

2.3 COMPARATIVE ANALYSIS

Model	Accuracy(avg)	Advantages	Disadvantages
ResNet	85-90%	High accuracy, well-established, good for transfer learning	Computationally expensive, large memory footprint
MobileNet	75-80%	Lightweight, fast inference, suitable for mobile devices	Lower accuracy than deeper models, limited feature extraction capabilities
DenseNet	80-85%	Efficient feature reuse, strong gradient flow, improved accuracy	More complex training, higher memory requirements
VGGNet	70-80%	Simple architecture, easy to understand and interpret	Requires large datasets for good performance, can be computationally expensive
Inception	80-85%	Multi-scale processing, improved feature representation	Complex architecture, can be time-consuming to train

CHAPTER 3
PROPOSED METHODOLOGY

3.1 DATASET:

3.1.1 DATASET SELECTION:

To train our model, we are going to use the TrashNet Dataset.

Description of the dataset is as follows:

The TrashNet dataset is a collection of images focused on garbage objects commonly encountered in everyday life.

The dataset spans six classes: glass, paper, cardboard, plastic, metal, and trash. Currently, the dataset consists of 2527 images:

501 glass

594 paper

403 cardboard

482 plastic

410 metal

137 trash

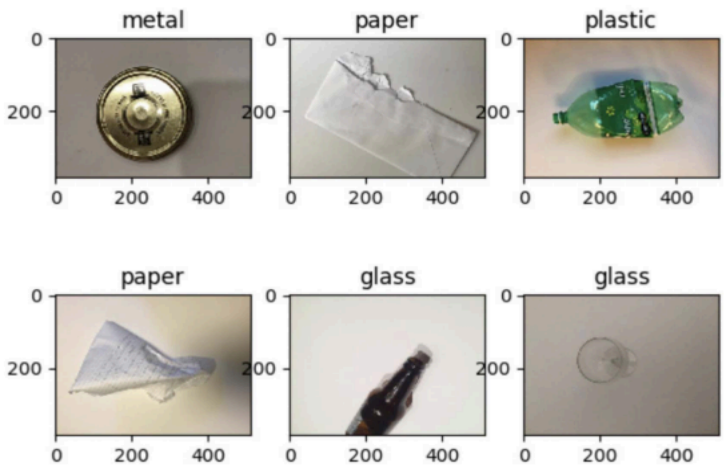


Fig 3.1.1 Dataset Details

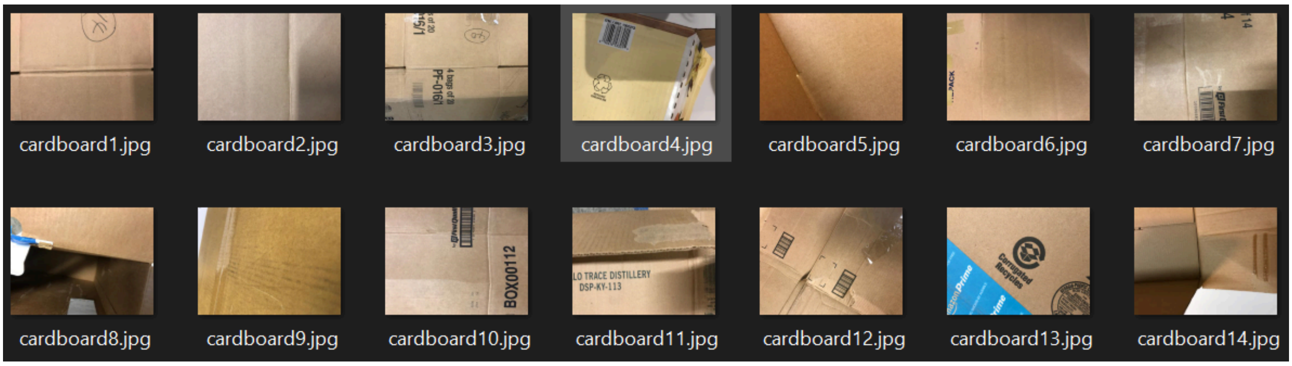


Fig 3.1.1 Cardboard

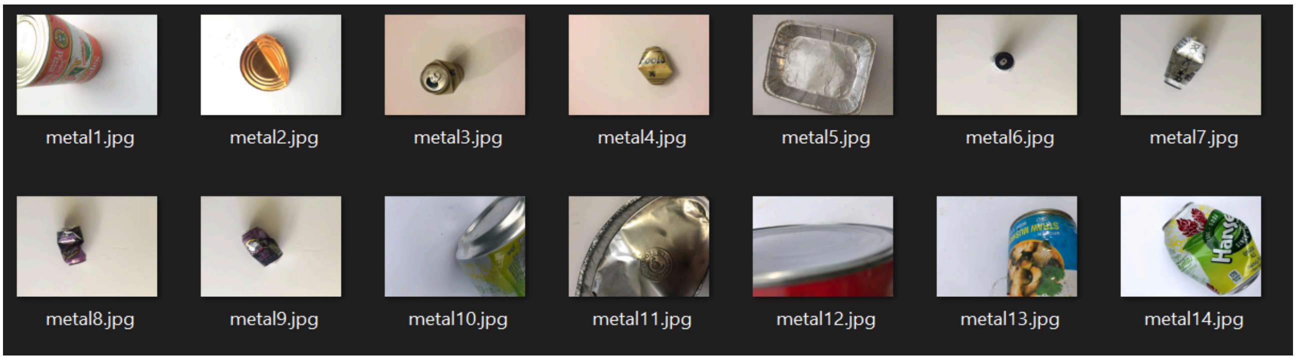


Fig 3.1.1 Metal

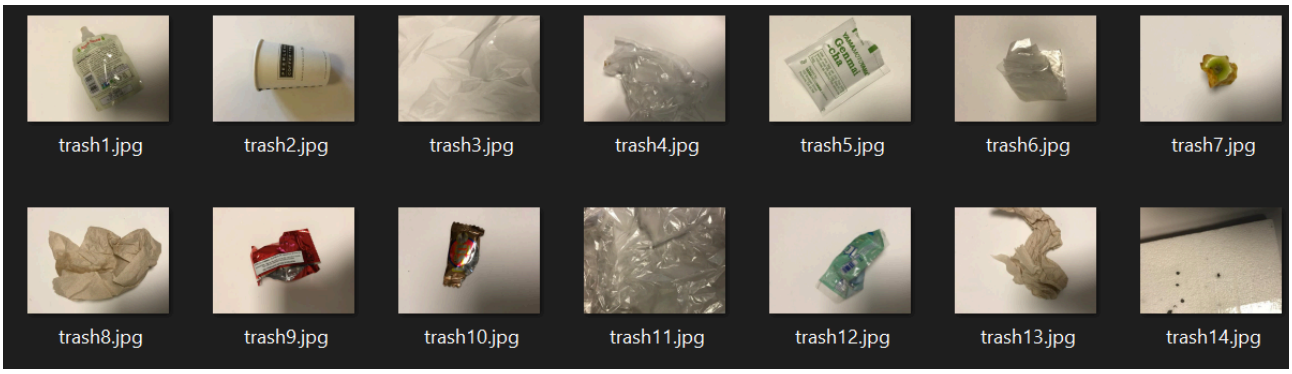


Fig 3.1.1 Trash

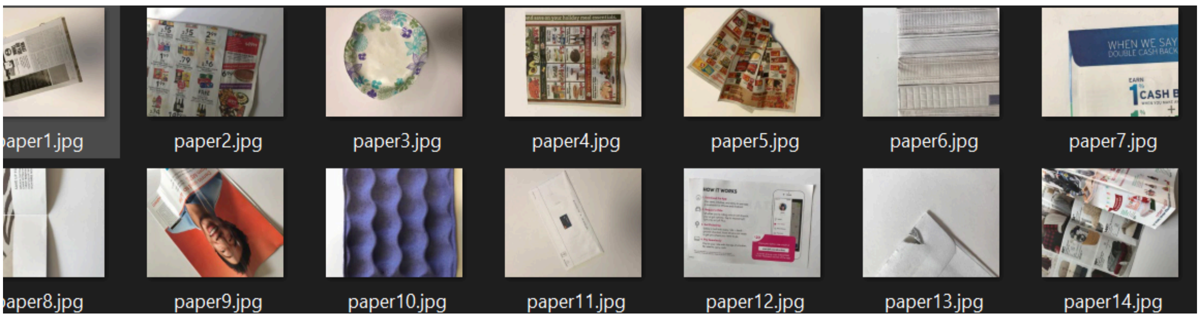


Fig 3.1.1 Paper

3.1.2 Data Preprocessing:

We know that the dataset is small.But to increase the generality, we use Data Augmentation.Various Data Augmentation techniques like **flipping**, **scaling**, **cropping**, **rotation**,etc are applied on the image to increase the size of the dataset.

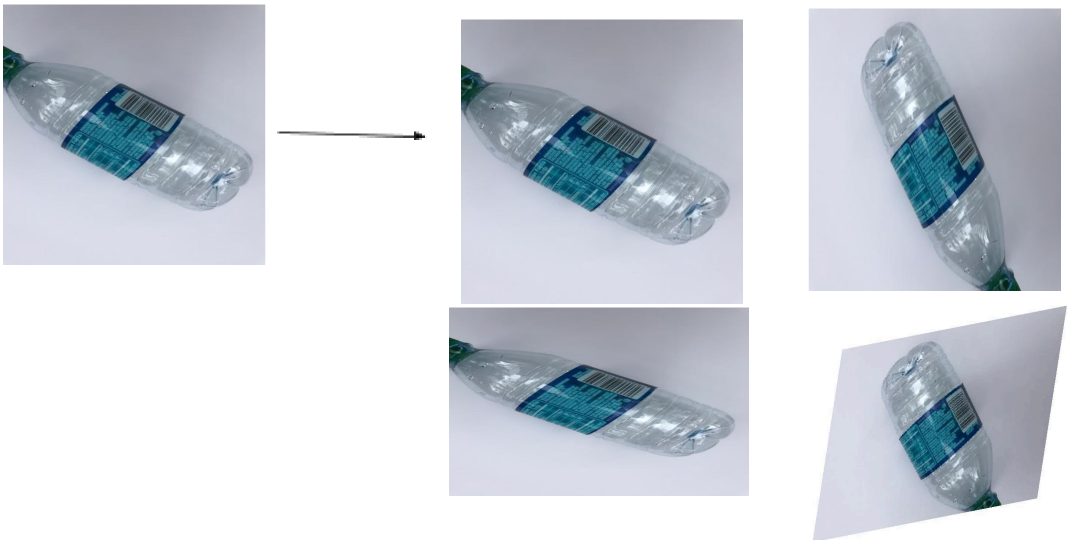


Fig 3.1.2 Data Augmentation Example

```
train_datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=45,
    width_shift_range=0.15,
    height_shift_range=0.15,
    zoom_range=0.15,
    horizontal_flip=True,
    vertical_flip=True,
    shear_range=0.05,
    brightness_range=[0.9, 1.1],
    channel_shift_range=10,
    fill_mode='nearest'
)

val_datagen =ImageDataGenerator(rescale=1./255)
```

Snapshot of the code for Data Augumentation

3.2 METHODOLOGY OVERVIEW

The proposed methodology comprises three primary modifications in the classic ResNet-50 model.

(1).Integration of attention module (CBAM)

(2).Modifying Downsampling Process

(3).Integration of Multi-Scale feature fusion

3.2.1 CBAM:

CBAM, or Convolutional Block Attention Module, is a attention mechanism which focuses on the important details and informations and neglects the irrelevant noises.There are two sub modules in it:

(1).Channel Attention Module(CAM)

(2).Spatial Attention Module (SAM)

1. Channel Attention: Channel attention aims to capture interdependencies among feature channels within a convolutional block. It computes channel-wise statistics (e.g., mean and max) and learns attention weights for each channel. By dynamically weighting the importance of each channel, CBAM can effectively emphasize informative features while suppressing less relevant ones.

2. Spatial Attention: Spatial attention focuses on capturing relationships among spatial locations within feature maps. It utilizes global pooling operations (e.g., average pooling and max pooling) to aggregate spatial information across feature maps. Then, it learns attention weights for each spatial location to highlight regions of interest.

By integrating both channel and spatial attention mechanisms, CBAM enables CNNs to adaptively attend to relevant features at both channel and spatial levels, leading to improved feature representation and ultimately enhanced performance in various computer vision tasks such as image classification, object detection, and semantic segmentation.

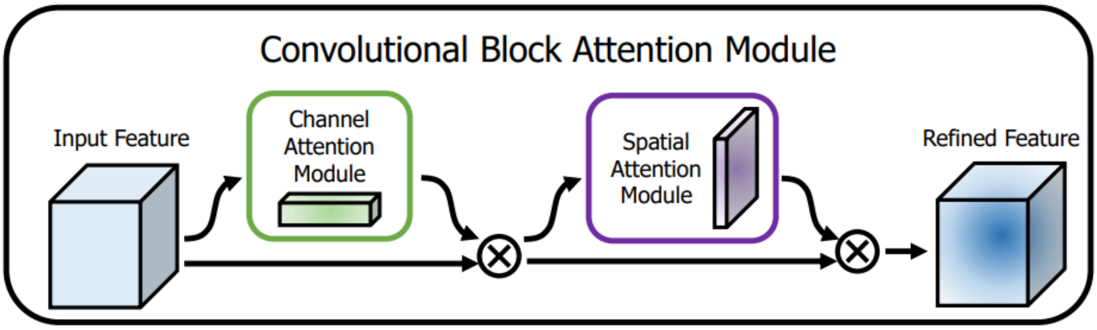


Fig 3.2.1 CBAM Architecture

3.2.2 Modification of Downsampling Process:

The ResNet-50 model has 1*1 convolution kernel with a step size of 2 which is a result for information loss. It reduces the feature map and decreases the number of parameters.

Structure of ResNet-50

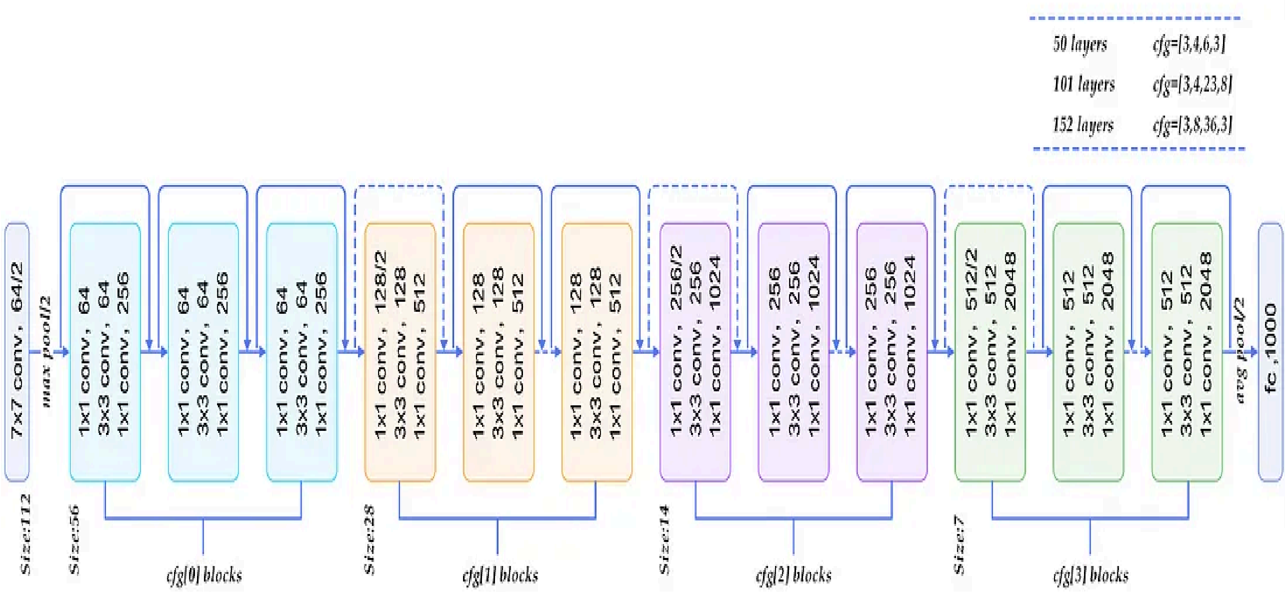
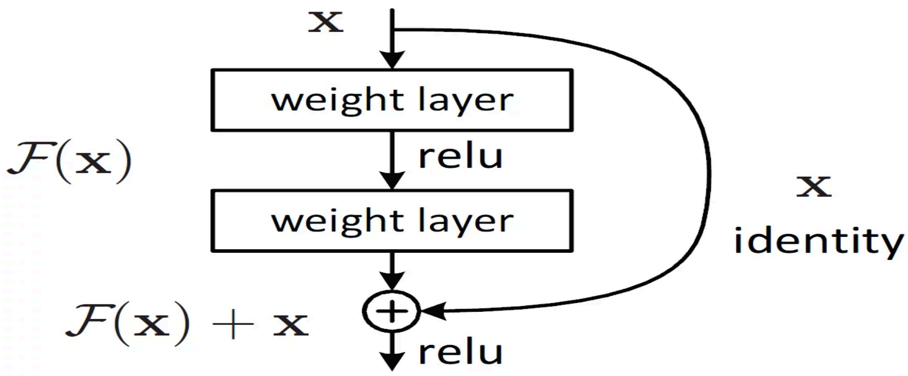
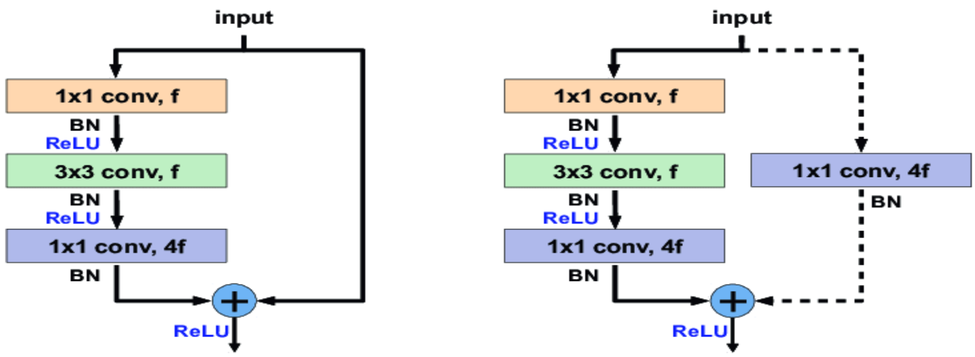


Fig 3.2.2 Structure of ResNet-50





Intially,When Downsampling is required in the first block of stage 2 onwards

3.2.3 INTEGRATION OF MULTI-SCALE FEATURE FUSION

In ResNet-50, multi-scale feature fusion refers to the integration of features from different layers of the network to capture information at multiple scales. ResNet-50 is a specific variant of the ResNet architecture, which is a type of deep convolutional neural network commonly used for image classification and other computer vision tasks.

There are various methods of how we can implement Multi-scale feature fusion in ResNet-50:
There are several methods to implement multi-scale feature fusion in ResNet-50. Here are a few commonly used techniques:

1. Pyramid Feature Fusion:
- This method involves creating a feature pyramid by downsampling the input image at different scales and extracting features independently at each scale.
- The features from different scales are then fused together using operations such as concatenation or addition.
- Each scale of the feature pyramid can correspond to a different layer in ResNet-50, and the fused features are combined before passing through subsequent layers of the network.

2. Dilated Convolutions:
- Dilated convolutions are a type of convolutional operation that increases the receptive field of the network without reducing spatial resolution.
- By using dilated convolutions in certain layers of ResNet-50, the network can capture information at multiple scales within the same layer.
- This approach allows the network to effectively fuse features from different scales without introducing additional layers or computational overhead.

3. Feature Concatenation:
- In this approach, features from different layers of ResNet-50 are concatenated together along the channel dimension.
- This concatenation operation combines features from multiple scales into a single tensor, which is then passed through subsequent layers of the network.

- Feature concatenation can be performed at multiple points in the network, allowing for multi-scale feature fusion at different depths.

4. Feature Addition:

- Similar to concatenation, feature addition involves combining features from different layers of ResNet-50.
- Instead of concatenating the features along the channel dimension, they are added element-wise.
- This operation allows the network to selectively combine features from different scales, emphasizing important information while suppressing irrelevant details.

5. Attention Mechanisms:

- Attention mechanisms can be incorporated into ResNet-50 to dynamically adjust the importance of features from different scales.
- These mechanisms use learnable weights to modulate the contribution of each scale to the fused feature representation.
- Attention mechanisms can help the network focus on relevant information while ignoring distractions, leading to improved performance in tasks such as object detection and segmentation.

These methods can be used individually or in combination to implement multi-scale feature fusion in ResNet-50, depending on the specific requirements of the task at hand.

Method we are going to use:

We will use Feature **"Concatenation Method"** because its suits our purpose

Trade off for this Method**:**

Computational Complexity

Architecture:

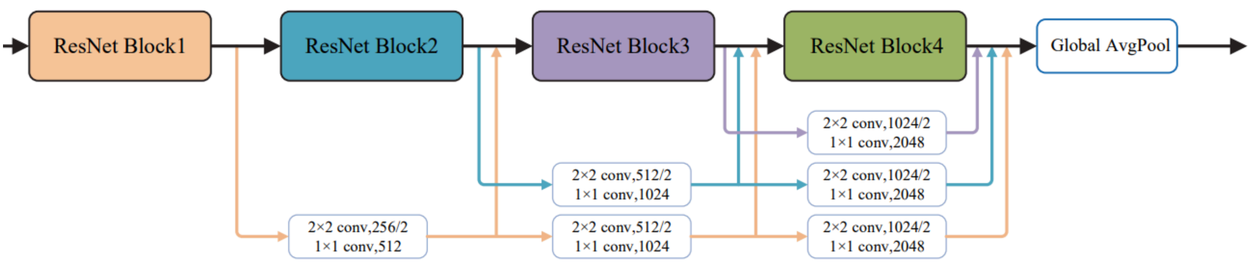


Fig 3.2.3 Multi-scale Feature Fusion

Workflow Steps:

Step1: Import dataset

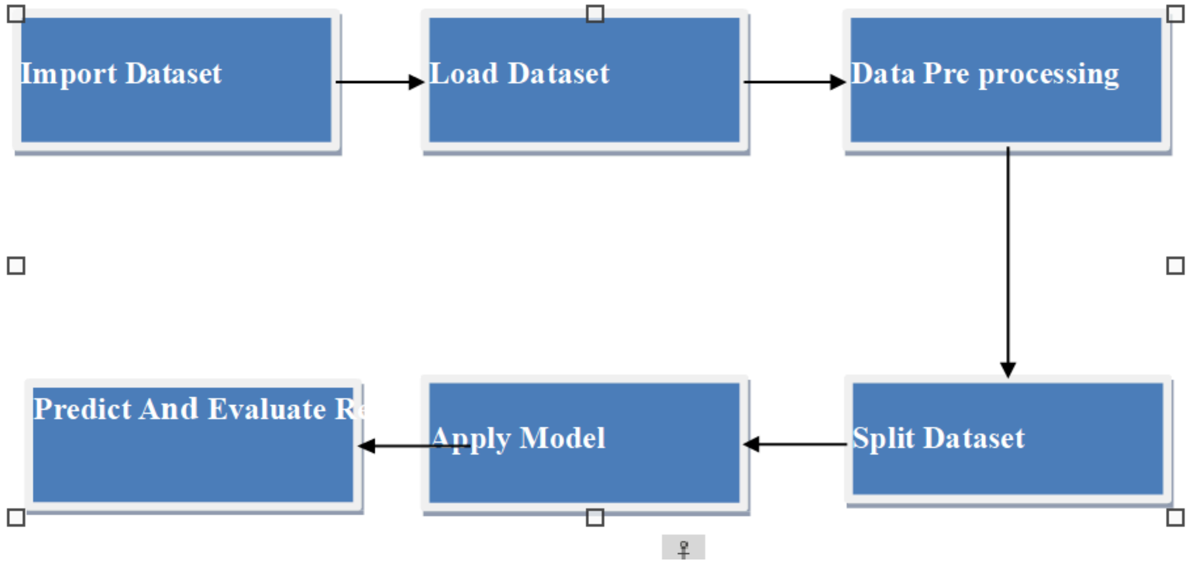
Step2: Load dataset

Step3: Apply data preprocessing techniques on dataset like Data Augmentation

Step4: Apply the Modified ResNet model on dataset

Step5: Predict result.

Step6: Evaluate results.



CHAPTER - 4

TOOLS AND TECHNOLOGY

4.1 Programming Language & Tools

Python

Hardware Requirements:

Operating System: Microsoft® Windows® or Linux. o Memory: At least 8 GB of RAM is recommended..

Storage: 150 GB of extra Space is required. o Graphics Processing Unit (GPU): 15 GB of Graphics Card Required

Software and Libraries:

Python Libraries:

Various Python libraries are available for crop disease detection, such as:

NumPy

Pandas

Scikit-learn.

TensorFlow

OpenCV

Matplotlib

Software

Jupyter Notebook or Google Colab or kaggle or Visual

CHAPTER 5

IMPLEMENTATION

5.1 Importing Libraries:

DP-2 Garbage Image Classification Codi... Draft saved

File Edit View Run Add-ons Help

Run All Code Draft Session off (run a cell to start)

```
[ ]:
import warnings
warnings.filterwarnings('ignore')

import os
import numpy as np
import pandas as pd
import seaborn as sns
import tensorflow as tf
import matplotlib.pyplot as plt
from tensorflow import keras
from tensorflow.keras import layers, regularizers
from matplotlib.colors import LinearSegmentedColormap
from PIL import Image
from sklearn.model_selection import train_test_split
from sklearn.utils.class_weight import compute_class_weight
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from keras.layers import Input, Activation, Add, Dense, Conv2D, GlobalAveragePool
from keras.layers import BatchNormalization, Dropout
from tensorflow.keras.applications import ResNet50
```

Fig 5.1 Importing Libraries

5.2 Splitting the Dataset

```
# SPlit into different train and test dataset
train_df, val_df = train_test_split(df, test_size=0.2, random_state=42, stratify=

# Print the number of images in each set
print(f"Number of images in the training set: {len(train_df)}")
print(f"Number of images in the validation set: {len(val_df)}")
```

Fig 5.2 Splitting Dataset

5.3 Data Augmentation

```
train_datagen = ImageDataGenerator(
      rescale=1./255,
      rotation_range=45,
      width_shift_range=0.15,
      height_shift_range=0.15,
      zoom_range=0.15,
      horizontal_flip=True,
      vertical_flip=True,
      shear_range=0.05,
      brightness_range=[0.9, 1.1],
      channel_shift_range=10,
      fill_mode='nearest'
)

val_datagen =ImageDataGenerator(rescale=1./255)
```

5.4 Creating Batches of the Dataset

```
# For Training Data
train_generator = train_datagen.flow_from_dataframe(
    dataframe=train_df,
    x_col="filepath",
    y_col="label",
    target_size=(384, 384),
    batch_size=32,
    class_mode='categorical',
    seed=42,
    shuffle=False
)

# For Testing Data
val_generator = val_datagen.flow_from_dataframe(
    dataframe=val_df,
    x_col="filepath",
    y_col="label",
    target_size=(384, 384),
    batch_size=32,
    class_mode='categorical',
    seed=42,
    shuffle=False
)
```

5.5 CBAM IMPLEMENTATION

```
import tensorflow as tf
from tensorflow.keras.layers import GlobalAveragePooling2D, GlobalMaxPooling2D,
Reshape, Dense, Multiply, Conv2D, Add, Activation, BatchNormalization

class ChannelAttention(tf.keras.layers.Layer):
    def __init__(self, ratio=8):
        super(ChannelAttention, self).__init__()
        self.avg_pool = GlobalAveragePooling2D()
        self.max_pool = GlobalMaxPooling2D()
        self.shared_mlp = Dense(units=1, activation='relu', use_bias=True,
kernel_initializer='he_normal')
         self.ratio = ratio

    def call(self, x):
        avg_pool = self.avg_pool(x)
        max_pool = self.max_pool(x)
        channel_attentions = self.shared_mlp(tf.concat([avg_pool, max_pool], axis=1))
        return Multiply()([x, Activation('sigmoid')(channel_attentions)])

class SpatialAttention(tf.keras.layers.Layer):
    def __init__(self, kernel_size=7):
         super(SpatialAttention, self).__init__()
        self.conv = Conv2D(filters=1, kernel_size=kernel_size, padding='same',
activation='sigmoid', kernel_initializer='he_normal')

    def call(self, x):
        avg_pool = tf.reduce_mean(x, axis=-1, keepdims=True)
        max_pool = tf.reduce_max(x, axis=-1, keepdims=True)
        combined = tf.concat([avg_pool, max_pool], axis=-1)
        return Multiply()([x, self.conv(combined)])

class CBAM(tf.keras.layers.Layer):
```

```
    def __init__(self, ratio=8, kernel_size=7):
        super(CBAM, self).__init__()
        self.channel_attention = ChannelAttention(ratio)
        self.spatial_attention = SpatialAttention(kernel_size)

    def call(self, x):
        x_out = self.channel_attention(x)
        return self.spatial_attention(x_out)
```

5.6 MODIFIED RESNET-50

```
def Modified_ResNet50(input_shape, classes):


    # Define the input as a tensor with shape input_shape

    X_input = Input(input_shape)
    # Stage 1
    X = Conv2D(64, (7, 7), strides=(2, 2), kernel_initializer='he_normal')(X_input)
    X = BatchNormalization(axis=3)(X)
    X = Activation('relu')(X)
    X = MaxPooling2D((3, 3), strides=(2, 2))(X)

    # Stage 2
    X = residual_block(X, 3, [64, 64, 256],reduce=True, stride=1)
    X = residual_block(X, 3, [64, 64, 256],reduce=False, stride=1)
    X = residual_block(X, 3, [64, 64, 256],reduce=False, stride=1)


    # Stage 3
    X = residual_block(X, 3, [128, 128, 512], reduce=True, stride=2)
    X = residual_block(X, 3, [128, 128, 512],reduce=False, stride=1)
    X = residual_block(X, 3, [128, 128, 512],reduce=False, stride=1)
    X = residual_block(X, 3, [128, 128, 512],reduce=False, stride=1)
    # Stage 4
```

```
X = residual_block(X, 3, [256, 256, 1024], reduce=True, stride=2)
X = residual_block(X, 3, [256, 256, 1024],reduce=False, stride=1)
X = residual_block(X, 3, [256, 256, 1024],reduce=False, stride=1)
X = residual_block(X, 3, [256, 256, 1024],reduce=False, stride=1)
X = residual_block(X, 3, [256, 256, 1024],reduce=False, stride=1)
X = residual_block(X, 3, [256, 256, 1024],reduce=False, stride=1)

# Stage 5
X = residual_block(X, 3, [512, 512, 2048], reduce=True, stride=2)
X = residual_block(X, 3, [512, 512, 2048],reduce=False, stride=1)
X = residual_block(X, 3, [512, 512, 2048],reduce=False, stride=1)

  # Global Average Pooling to reduce spatial dimensions
X = GlobalAveragePooling2D()(X)

# Add Dropout to prevent overfitting
X = Dropout(0.5)(X)

# Fully Connected Layer for classification
X = Dense(classes, activation='softmax')(X)

# Create the model
 model = Model(inputs = X_input, outputs = X, name='Modified_ResNet50')

return model
```

CHAPTER-6

EXPERIMENT,PARAMETER AND RESULT ANALYSIS

Here we have used 5 parameters for model evaluation. Accuracy, precision, recall and F-1 score and confusion matrix .

Here are for parameters:

Accuracy: Accuracy measures the proportion of correct predictions made by the model. It is calculated by dividing the number of correct predictions by the total number of predictions made.

Formula: (True positives + True negatives) / (True positives + False positives + True negatives + False negatives)

Precision: Precision measures the proportion of true positive predictions out of all the positive predictions made by the model. It is calculated by dividing the number of true positive predictions by the total number of positive predictions made.

Formula: True positives / (True positives + False positives)

Recall: Recall measures the proportion of true positive predictions out of all the actual positive instances in the dataset. It is calculated by dividing the number of true positive predictions by the total number of positive instances in the dataset.

Formula: True positives / (True positives + False negatives)

F1 score: F1 score is a harmonic mean of precision and recall. It is a useful metric when both precision and recall are important. It is calculated by taking the harmonic mean of precision and recall.

Formula: 2 * ((Precision * Recall) / (Precision + Recall))

Terms used in evaluation parameters:

True Positive (TP): It refers to the cases where the model correctly predicts the positive class (condition present) when the actual ground truth is indeed positive. In other words, the model identifies a positive instance correctly.

False Positive (FP): It occurs when the model incorrectly predicts the positive class (condition present) when the actual ground truth is negative (condition absent). It represents a case where the model indicates the presence of a condition when it's not actually present.

False Negative (FN): It happens when the model incorrectly predicts the negative class (condition absent) when the actual ground truth is positive (condition present). It represents a case where the model fails to identify the presence of a condition.

True Negative (TN): It refers to the cases where the model correctly predicts the negative class (condition absent) when the actual ground truth is indeed negative. In other words, the model identifies a negative instance correctly.

Confusion matrix: Confusion matrix is a performance evaluation technique that is widely used to analyze the performance of a classification model. It is a square matrix that summarizes the results of the model's predictions on a set of test data. The confusion matrix provides a detailed breakdown of the predicted and actual classes, allowing us to assess the model's performance in terms of true positives, true negatives, false positives, and false negatives.

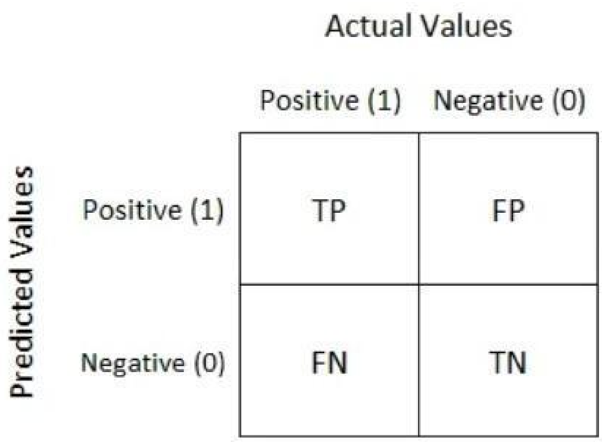


Figure 5.1 Confusion matrix

```
Confusion Matrix:
[[114   1]
 [  0 205]]
```

Figure 5.2 Output of model

6.2 Hyper Parameter

Hyper Parameter	Value
Optimizer	AdamW
Learning Rate	1e -3
batch size	10
epochs	100

6.3 Result Analysis

Confusion Matrix

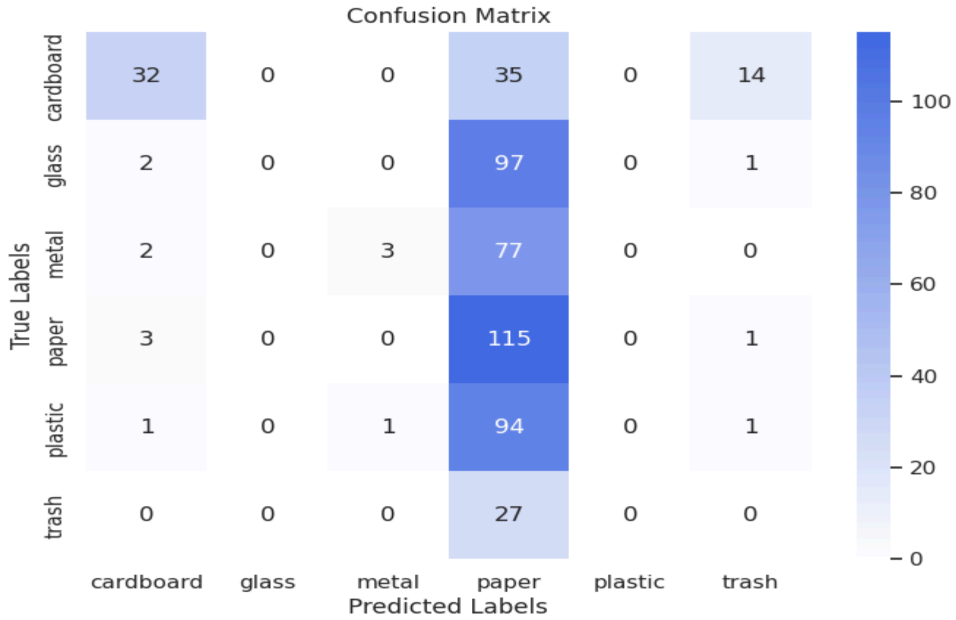


Fig 6.3.1 Confusion Matrix

	precision	recall	f1-score	support
cardboard	0.80	0.40	0.53	81
glass	0.00	0.00	0.00	100
metal	0.75	0.04	0.07	82
paper	0.26	0.97	0.41	119
plastic	0.00	0.00	0.00	97
trash	0.00	0.00	0.00	27
accuracy			0.30	506
macro avg	0.30	0.23	0.17	506
weighted avg	0.31	0.30	0.19	506

Fig 6.3.2 Model Performance Evaluation

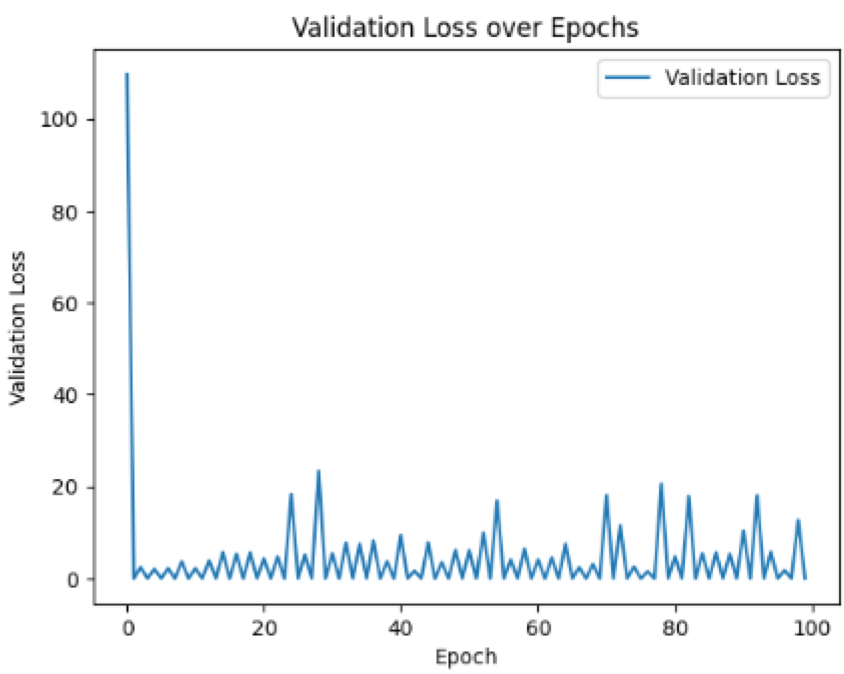


Fig 6.3.3 Validation Loss Over Epochs

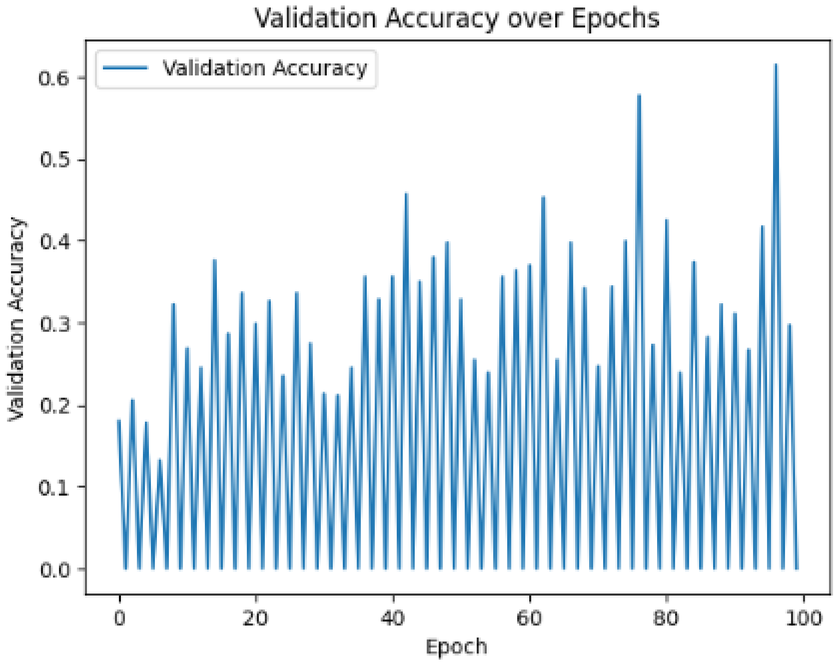


Fig 6.3.4 Validation Accuracy over epochs

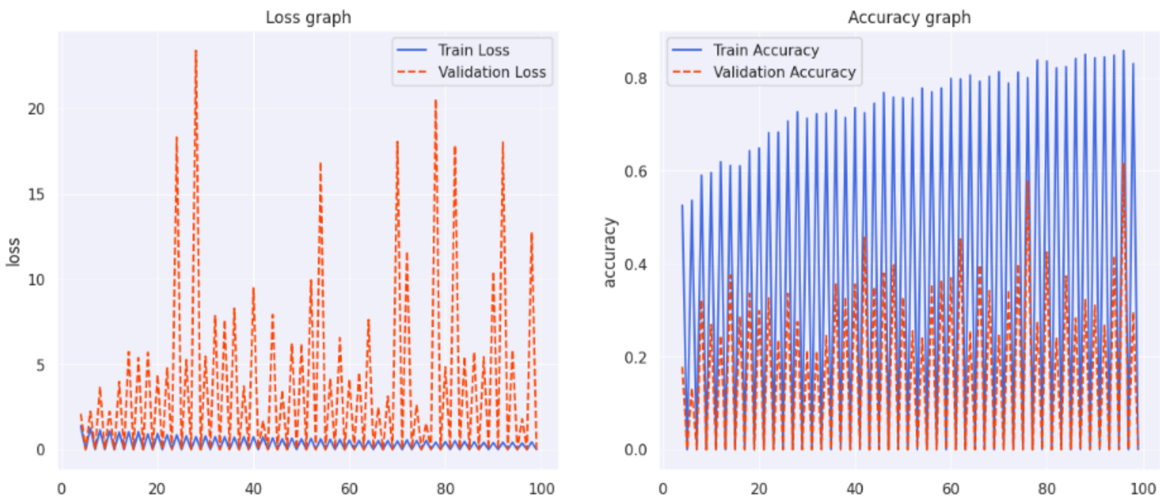


Fig 6.3.5 Loss Graph and Accuracy Graph

Without Multi-Scale Feature fusion trained from scratch

```
Epoch 48/100
64/64 ━━━━━━━━━━━━━━━━━━━━ 0s 322us/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accur
y: 0.0000e+00 - val_loss: 0.0000e+00 - learning_rate: 1.2500e-04
Epoch 49/100
64/64 ━━━━━━━━━━━━━━━━━━━━ 74s 1s/step - accuracy: 0.8502 - loss: 0.3816 - val_accuracy: 0.808
- val_loss: 0.5587 - learning_rate: 1.2500e-04
Epoch 50/100
64/64 ━━━━━━━━━━━━━━━━━━━━ 0s 305us/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accur
y: 0.0000e+00 - val_loss: 0.0000e+00 - learning_rate: 1.2500e-04
Epoch 51/100
64/64 ━━━━━━━━━━━━━━━━━━━━ 73s 1s/step - accuracy: 0.8676 - loss: 0.3420 - val_accuracy: 0.800
- val_loss: 0.5829 - learning_rate: 1.2500e-04
Epoch 52/100
64/64 ━━━━━━━━━━━━━━━━━━━━ 0s 311us/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accur
y: 0.0000e+00 - val_loss: 0.0000e+00 - learning_rate: 1.2500e-04
Epoch 52: early stopping
Restoring model weights from the end of the best epoch: 2.
```

Fig 6.3.6 Performance Without Multi-Scale Feature Fusion

When Multi-Scale Feature Fusion is enabled

```
Epoch 91/100
64/64 ━━━━━━━━━━━━━━━━━━━━ 79s 1s/step - accuracy: 0.8441 - loss: 0.3798 - val_accuracy: 0.3103 - val_loss: 10.4278
Epoch 92/100
64/64 ━━━━━━━━━━━━━━━━━━━━ 0s 384us/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accuracy: 0.0000e+00 - val_loss: 0.0000e+00
Epoch 93/100
64/64 ━━━━━━━━━━━━━━━━━━━━ 79s 1s/step - accuracy: 0.8581 - loss: 0.3749 - val_accuracy: 0.2668 - val_loss: 18.0242
Epoch 94/100
64/64 ━━━━━━━━━━━━━━━━━━━━ 0s 392us/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accuracy: 0.0000e+00 - val_loss: 0.0000e+00
Epoch 95/100
64/64 ━━━━━━━━━━━━━━━━━━━━ 79s 1s/step - accuracy: 0.8459 - loss: 0.4521 - val_accuracy: 0.4170 - val_loss: 5.8358
Epoch 96/100
64/64 ━━━━━━━━━━━━━━━━━━━━ 0s 388us/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accuracy: 0.0000e+00 - val_loss: 0.0000e+00
Epoch 97/100
64/64 ━━━━━━━━━━━━━━━━━━━━ 79s 1s/step - accuracy: 0.8596 - loss: 0.3810 - val_accuracy: 0.6146 - val_loss: 1.7904
Epoch 98/100
64/64 ━━━━━━━━━━━━━━━━━━━━ 0s 406us/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accuracy: 0.0000e+00 - val_loss: 0.0000e+00
Epoch 99/100
64/64 ━━━━━━━━━━━━━━━━━━━━ 80s 1s/step - accuracy: 0.8436 - loss: 0.3865 - val_accuracy: 0.2964 - val_loss: 12.7367
Epoch 100/100
64/64 ━━━━━━━━━━━━━━━━━━━━ 0s 395us/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accuracy: 0.0000e+00 - val_loss: 0.0000e+00
```

CHAPTER 7

CONCLUSION

From the results we can see that the modified ResNet-50 which is built from the scratch and not fine-tuned on the pre-trained model performs well when combined with the attention module called “CBAM” and modified downsampling process.But when Multi-Scale feature fusion is added to the Network, the complexity of the network increases which leads to the learning rate being zero after every alternate epoch.The Multi-Scale feature fusion still need to be modified to perfectly fit with the network.

.

REFERENCES

[1] Li, K. T. Z. W. Y. C. W. (2020). Research on Garbage Image Classification Method Based on Convolutional Neural Network. https://ijcat.com/archieve/volume9/issue1/ijcatr09011007

[2] Meng, S., Zhang, N., & Ren, Y. (2020). X-DenseNet: Deep learning for garbage classification based on visual images. Journal of Physics, 1575(1), 012139. https://doi.org/10.1088/1742-6596/1575/1/012139

[3] Liu, W., Ouyang, H., Liu, Q., Cai, S., Wang, C., Xie, J., & Hu, W. (2022). Image recognition for garbage classification based on transfer learning and model fusion. Mathematical Problems in Engineering, 2022, 1–12. https://doi.org/10.1155/2022/4793555

[4] Liu, F., Xu, H., Qi, M., Liu, D., Wang, J., & Kong, J. (2022). Depth-Wise separable convolution attention module for garbage image classification. Sustainability, 14(5), 3099. https://doi.org/10.3390/su14053099

[5] Wang, W., Baobao, Z., Wang, Z., FangZhi, Z., & Liu, Q. (2021). Garbage image recognition and classification based on hog feature and SVM-Boosting. Journal of Physics, 1966(1), 012002. https://doi.org/10.1088/1742-6596/1966/1/012002

[6] S. Meng and W. -T. Chu, "A Study of Garbage Classification with Convolutional Neural Networks," 2020 Indo – Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN), Rajpura, India, 2020, pp. 152-157, doi: 10.1109/Indo-TaiwanICAN48429.2020.9181311

[7] Ruíz, V., Sánchez, Á. M., Vélez, J. F., & Raducanu, B. (2019). Automatic Image-Based Waste Classification. In *Lecture Notes in Computer Science* (pp. 422–431). https://doi.org/10.1007/978-3-030-19651-6_41

[8] S. R., R. P., V. S., K. R. and G. M., "Deep Learning based Smart Garbage Classifier for Effective Waste Management," 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 1086-1089, doi: 10.1109/ICCES48766.2020.9137938.

[9] Wu, Z., Li, H., Wang, X., Wu, Z., Zou, L., Xu, L., & Tan, M. (2022). New benchmark for household garbage image recognition. Tsinghua Science & Technology, 27(5), 793–803. https://doi.org/10.26599/tst.2021.9010072