MARINE DEBRIS PLASTIC DETECTION

A Thesis

Submitted in fulfillment of the

Requirements for the award of the Degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

P.MOUNIKA 17191A0544

S.LAVANYA 17191A0549

M.JOGENDRA 18191A0504

Under the esteemed guidance of

Smt. P. VIJAYA KUMARI M.Tech

Assistant Professor (Adhoc)

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR COLLEGE OF ENGINEERING

(Autonomous)

PULIVENDULA-516 390

ANDHRA PRADESH-INDIA 2021

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

COLLEGE OF ENGINEERING (AUTONOMOUS) PULIVENDULA-516 390

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

STUDENT DECLARATION

We hereby declare that this submission is our own work and that to the best of our knowledge and belief, it contains no material previously published or written by another person or material which has been accepted for the award of any degree or diploma of any university or institute of higher learning.

P. MOUNIKA

17191A0544

S. LAVANYA

17191A0549

M.JOGENDRA

18191A0504

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTAPUR

COLLEGE OF ENGINEERING

(Autonomous)

PULIVENDULA-516 390

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "MARINE DEBRIS PLASTIC DETECTION MODEL" is being submitted by

P. MOUNIKA	**17191A0544**
S. LAVANYA	**17191A0549**
M. JOGENDRA	**18191A0504**

in partial fulfillment for the award of the Degree of Bachelor of Technology in Computer science & engineering to the **Jawaharlal Nehru Technological University, Anantapur college of Engineering (Autonomous) Pulivendula**, is a record of bonafide work carried out under my guidance and supervision. The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree or Diploma.

Head of the Department
Smt. Dr. Jessica Saritha
Assistant Professor & Head
Department of CSE JNTUACE, Pulivendula

Signature of the Guide
(Smt.P.Vijaya Kumari M.tech)

Signature of the External Examiner

Acknowledgement

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mention of people who made it possible, whose constant guidance and encouragement crowned our efforts with success. It is a pleasant aspect that I now have the opportunity to express my guidance for all of them.

I wish to thank Prof. **G. K. RangaJanardhan** Honorable Vice Chancellor, JNTUA. His motivation words that created burning desire in us and that eventually led to successfully completing this project work.

The man who has helped us a lot in times of trouble, and who helped us in recovering from intricate problems **Prof.G. Shankara Sekhar Raju** Garu, principal, JNTUA College of Engineering, Pulivendula. I am thankful to him.

It's my pleasure to say thanks to **Prof. G. V. SubbaReddy** Garu, Vice Principal, JNTUA College of Engineering, Pulivendula for motivating us to have vision and persistence to work in spite of many obstacles.

Foremost, we gratefully acknowledge my sincere gratitude to our respected **Smt. Dr. Jessica Saritha** Garu**,** Assistant Professor, Head of the Department, Department of Computer Science and Engineering, JNTUA College of Engineering, Pulivendula for her valuable guidance, timely suggestions, and lively intensity throughout the work. We are always thankful to her for providing me the opportunity to do work.

The real mentor and motivator of this project **Smt.Vijaya Kumari** mam. Her wide knowledge and logical way of thinking have made a deep impression on me. Her understanding, encouragement and personal guidance have provided the basis for this thesis. Her source of inspiration for innovative ideas and his kind support is well to all her students and colleagues.

Last but far from least, I also thank my family members, my friends, staff and faculty members of CSE Department for their moral support and constant encouragement. I am very much thankful to one and all that helped me for the successful completion of the project.

With gratitude

P. MOUNIKA	**17191A0544**
S. LAVANYA	**17191A0549**
M. JOGENDRA	**18191A0504**

ABSTRACT

Estimating plastics that dot the world’s oceans is one of the most pressing environmental concerns of our time. Prevailing methods for determining the amount of floating plastic debris, usually conducted manually, are time demanding and rather limited in coverage. With the aid of machine learning, herein, we propose a method for automatically identifying floating marine plastics.

Image Recognition Model to detect plastics, glass, paper, rubbish, metal, and cardboard. This is used to detect this pollution in the ocean to allow the eradication of these materials, helping marine life, fishermen, tourism, and making the world resilient to climate change.

A generic image detection program that uses Google's Machine Learning library, Tensor flow, and a pre-trained Deep Learning Convolutional Neural Network model called Resnet34.

Resnet34 is used to detect images of plastics, garbage, metal, paper, glass, and cardboard in the ocean to allow for the collection and reduction of these materials, particularly plastic in the ocean waters.

LIST OF CONTENTS

Chapter 1: Introduction

One of the biggest environmental challenges currently threatening our ecosystems is marine debris and especially pollution from plastics. Plastics are entering the marine environment through various channels, such as being carried by wind, shipping, coastal activities, or through effluent water discharge. Consequently, the world’s oceans contain an enormous amount of plastic debris. More precisely, the five subtropical gyres harbor the largest accumulations of plastics in comparison to other ocean zones.

The Great Pacific Gyre alone, also known as the Great Pacific Garbage Patch (GPGP), is believed to contain at least 79 thousand tons of marine plastics that are floating in an area of 1.6 million km2 roughly the size of Mongolia. Boat surveys constitute the primary means for retrieving information about plastics floating on (or near) the water surface, such as, estimating their type and density. Meanwhile, vessel surveys are time demanding, expensive, and limited in coverage because of the subsequent processing of the material and data that are collected, notwithstanding the risk of tainting observations with human errors.

Attempting to identify floating debris, other researchers have proposed a sensing platform fitted with an onboard camera. However, devising an automated way for detecting and classifying marine debris, grounded on a machine learning approach, offers some added advantages. Such a method holds the potential to save valuable search time, lower expedition costs, and increase the level of accurately identifying large-size plastic debris.

To automate the process of macro-plastic detection, we employ a machine learning method, which constitutes a tool that powers many aspects of today’s society and finds applications ranging from content filtering in web searches to object identification and speech recognition. Domains of the scientific community, the business world, and governments are utilizing machine learning techniques mainly owing to their excellent performance and ability to conduct novel tasks.

1.1 Motivation and Scope:

The main motivation of the project is to reduce effort, time and cost while detecting plastic debris in water resources (sea’s, ocean’s) and increasing the marine creature’s life span and this reduction also helps to human livelihood.

Applying deep learning methods in the context of marine debris research can help create faster and more accurate tools for detecting and classifying floating litter.

1.2 Need for Study:

The oceans face a massive and growing threat from man-made works and littering the plastics. An estimated 17.6 billion pounds of plastic leaks into the marine environment from land-based sources every year this is roughly equivalent to dumping a garbage truck full of plastic into the oceans every minute.

Plastics never go away. Instead, they break down into smaller and smaller pieces, which act as magnets for harmful pollutants. When eaten by fish, some of those chemical-laden micro plastics can work their way up the food chain and into the fish we eat.

Recycling alone is not enough to solve the plastics crisis. To have an impact, we must reduce the amount of single-use plastic being produced at the source. Oceana's plastics campaign will urge companies to adopt alternatives for single-use plastic packaging.

Chapter 2: Problem Overview

2.1 Problem Statement

As plastics continue to flood into our oceans, the list of marine species affected by plastic debris expands.

Tens of thousands of individual marine organisms have been observed suffering from entanglement or ingestion of plastics permeating the marine environment from fish, to sea turtles, marine mammals and seabirds.

So to reduce this plastic debris and its effects on both sea creatures and humans, we are going to detect this debris in oceans using an image processing model in machine learning.

2.2 Objectives

The main objective of the project is to build an efficient solution for reducing plastic debris in water bodies and to reduce several problems like….

- *To reduce the effort, time and cost of identifying marine plastic debris.*
- *To use in Dataset Construction*
- *Reduction of problems related to Health and Hygiene*
- *Help fishermen and develops tourism*

2.3 Hypothesis

To develop the algorithm in Deep learning which detects marine debris from available dataset.

Chapter 3: Research Methodology

Plastic pollution is a ‘wicked environmental problem’ with annual estimates indicating global rivers discharging several million metric tons of plastic waste into the oceans
The research methodology in this project include:

- Understanding the problem
- Download and extract the images
- Organize the images into different folders
- Train the model
- Make and evaluate the test predictions
- Analyzing the result

3.1 Technologies

3.1.1 Machine Learning:

Machine learning algorithms are tasked with extracting primary information and making a prediction about new data samples. In contrast to other mathematical techniques, these algorithms construct and update their predictive model based on known data (training dataset). Machine learning methods can be applied to different tasks: spam filter, image processing, broad societal impact, image recognition, signal processing, etc.

Workflow is as follows:

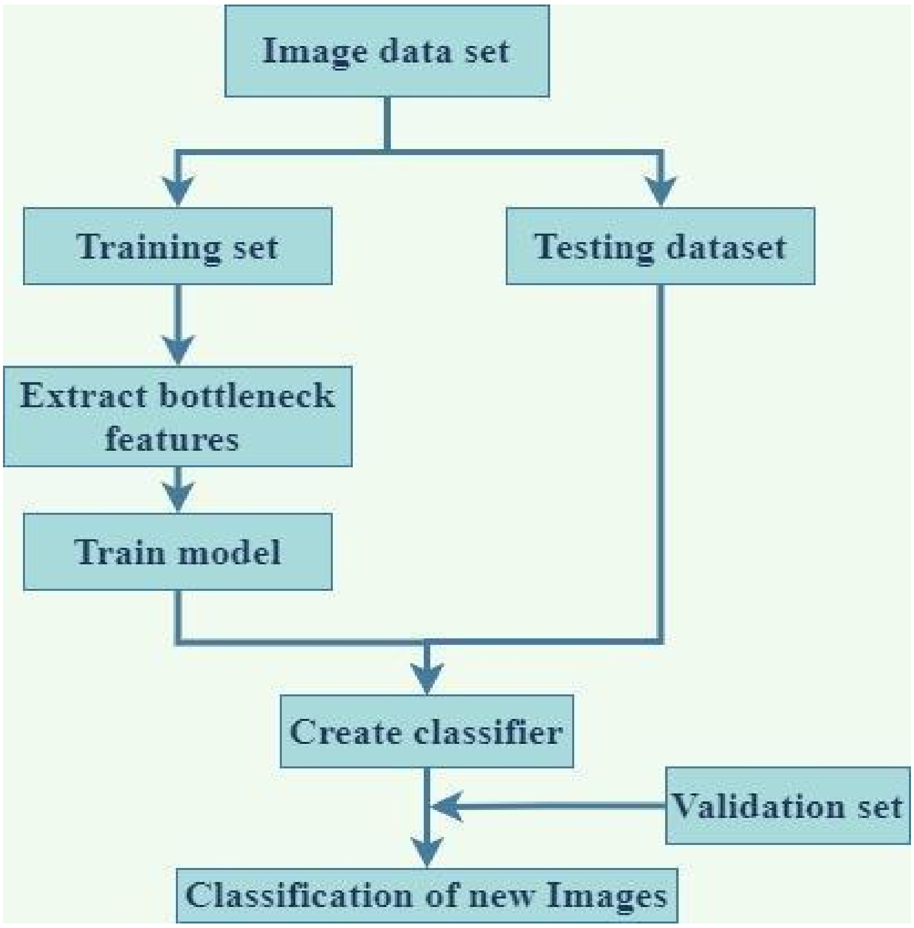

Fig.3.1.1 Machine learning model

3.1.2 **Image Processing Model**:

Image Processing is most commonly termed as 'Digital Image Processing' and the domain in which it is frequently used is 'Computer Vision'.

The data that we collect or generate is mostly raw data,i.e. it is not fit to be used in applications directly due to a number of possible reasons.Therefore,we need to analyse it first,perform the necessary pre-processing,then use it.

For instance,let's assume that we are trying to build a cat classifier.Our program would take an image as input and then tell us whether the image contains a cat or not.The first step for building this classifier would be to collect hundreds of cat pictures.One common issue is that all the pictures we have scraped would not be of the same size/dimensions , so before feeding them to the model for training, we would need to resize/pre-process them all to a standard size.

This is just one of many reasons why image processing is essential to any computer vision application.

3.1.3 **Convolutional Neural Network (CNN)**:

A **Convolutional Neural Network (ConvNet/CNN)** is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex.

Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. A collection of such fields overlap to cover the entire visual area.

An image is nothing but a matrix of pixel values. The role of the ConvNet is to reduce the images into a form which is easier to process, without losing features which are critical for getting a good prediction.

Here is an example of converting the 3*3 matrix pixel values

into a single column.

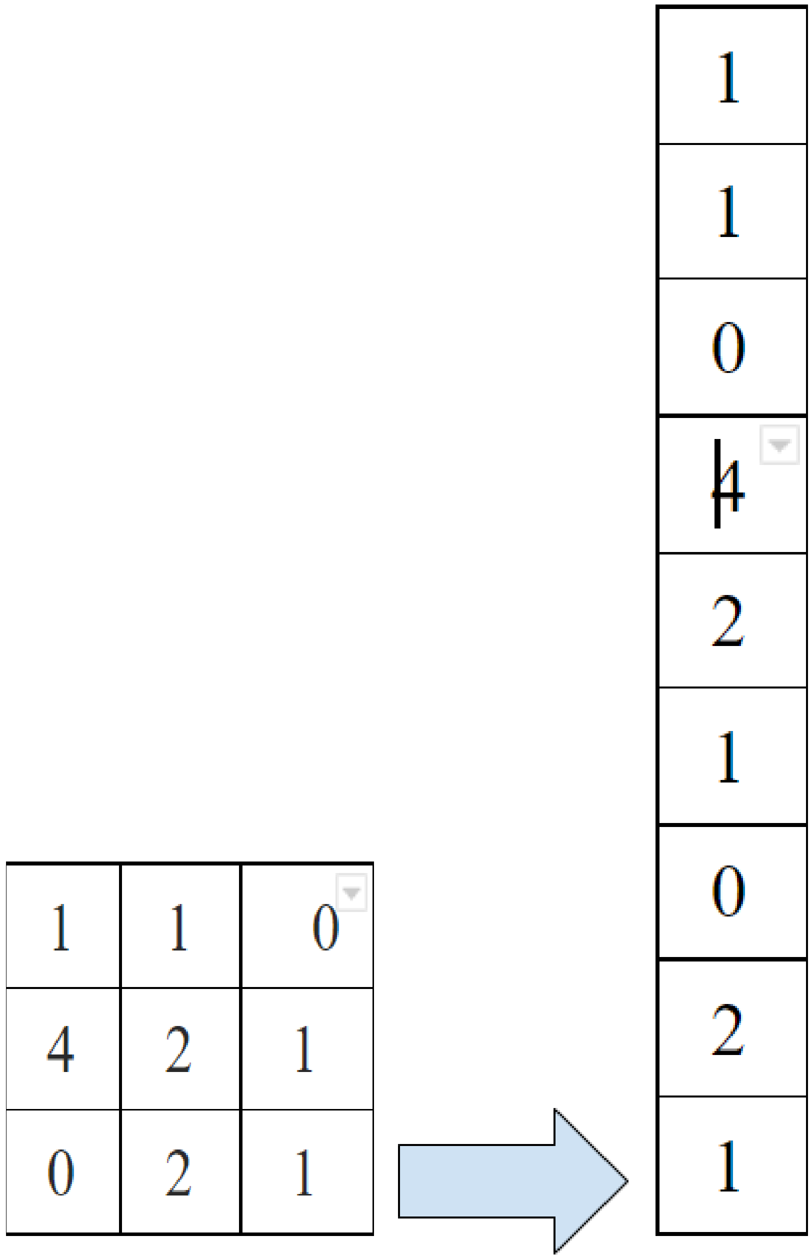

Fig.3.1.3 Flattening of a 3x3 image matrix into a 9x1 vector

3.1.4 ResNets:

One of the problems ResNets solve is the famous known vanishing gradient. This is because when the network is too deep, the gradients from where the loss function is calculated easily shrink to zero after several applications of the chain rule. This results in the weights never updating its values and therefore, no learning is being performed.

With ResNets, the gradients can flow directly through the identity function.

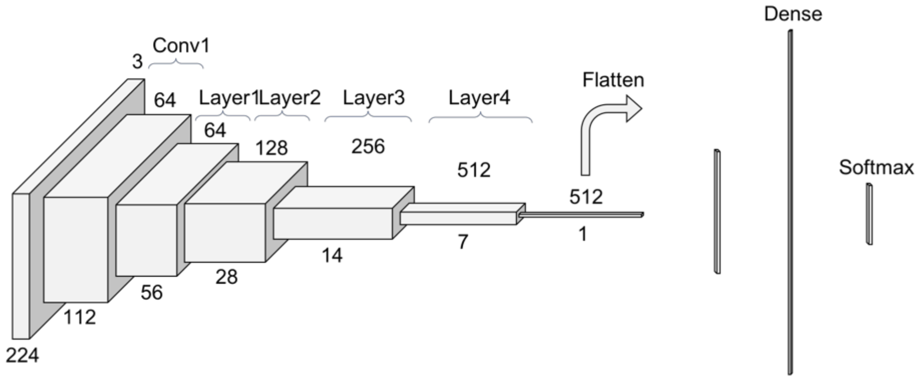

Fig 3.1.4 ResNet 34

3.1.5 Python:

Python is a general-purpose coding language—which means that, unlike HTML, CSS, and JavaScript, it can be used for other types of programming and software development besides web development.

Technically, machine learning falls under data science. Python for machine learning is pretty cool. Machine learning includes things like speech recognition, financial services, even the recommendations and it is used for machine learning via specific machine learning libraries and frameworks including scikit-learn and TensorFlow.

3.1.6 Seaborn:

What is data visualization: Data visualization provides insight into the distribution and relationships between variables in a dataset.

This insight can be helpful in selecting data preparation techniques to apply prior to modelling and the types of algorithms that may be most suited to the data.

Seaborn is a data visualization library for Python that runs on top of the popular Matplotlib data visualization library, although it provides a simple interface and aesthetically better-looking plots.

The primary plotting library for Python is called Matplotlib.
Seaborn is a plotting library that offers a simpler interface, sensible defaults for plots needed for machine learning, and most importantly, the plots are aesthetically better looking than those in Matplotlib.
Seaborn requires that Matplotlib is installed first. You can install Matplotlib directly using pip, as follows:

```
1 “sudo pip install Matplotlib”
```

Once installed, you can confirm that the library can be loaded and used by printing the version number, as follows:

```
# matplotlib
import matplotlib
print ('matplotlib: %s' % matplotlib. __version__)
```

3.1.7. fastai:

Fastai is a deep learning library which provides practitioners with high-level components that can quickly and easily provide state-of-the-art results in standard deep learning domains, and provides researchers with low-level components that can be mixed and matched to build new approaches.

3.2. Analyzing result:

A residual neural network (ResNet) is an artificial neural network (ANN) of a kind that builds on constructs known from pyramidal cells in the cerebral cortex. Residual neural networks do this by utilizing *skip connections*, or *shortcuts* to jump over some layers.

Typical *ResNet* models are implemented with double- or triple- layer skips that contain nonlinearities and batch normalization in between. An additional weight matrix may be used to learn the skip weights; these models are known as *Highway Nets*. Models with several parallel skips are referred to as *Dense Nets*. In the context of residual neural networks, a non-residual network may be described as a *plain network*.

Chapter 4: Technical Requirements

4.1 Software Requirements

- Operating System - Windows 10
- IDE - Online IDE (Google Colab notebooks)
- Languages - Python
- Technologies - Artificial Intelligence, Machine Learning, CNN, ResNets

4.1.1 IDE:

Online IDE built for developing the project at a time by the team of members. The IDE can be accessed by using the browser.

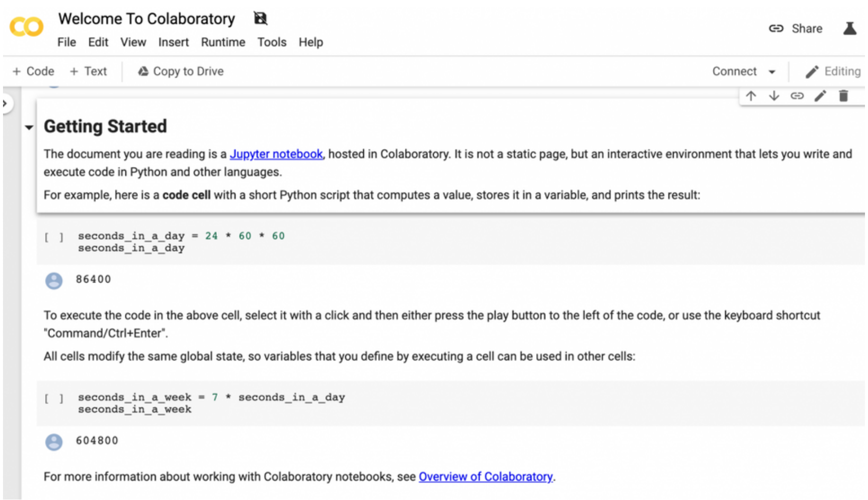

Fig 4.1 Online IDE

4.1.2 Python:

Python is free and simple to learn. Its primary features are that it is high-level, dynamically typed and interpreted. This makes debugging of errors easy and encourages the rapid development of application prototypes, marking itself as the language to code with.

Python is the best language for machine learning and artificial intelligence models. In this python is used to process the user utterances and inputs and understand the intent of the user using robust Image Processing models trained.

4.2 Hardware Requirements

Following are the hardware requirements for this project:

- Processor - Intel Core i3
- RAM - 4 GB
- Hard Disk - 128 GB (Not necessary)
- Monitor - 15.6” color display (Not necessary)
- Mouse - Three button optical mouse
- Keyboard - Standard qwerty keyboard

Chapter 5: Implementation

The goal is to implement a Marine debris detecting model that is pre trained and how accurately it can detect debris with the dataset given by the user.

- **Improved Image processing accuracy**: The Resnet34 model understands the image and gets trained according to the label.
- **Detects images that are wrongly predicted**: The model is capable of finding images that are incorrectly visualized.

5.1. Architecture

A residual neural network is a convolutional neural network (CNN) with lots of layers. In particular, resnet34 is a CNN with 34 layers that’s been pretrained on the ImageNet database. A pre-trained CNN will perform better on new image classification tasks because it has already learned some visual features and can transfer that knowledge over (hence transfer learning).
Since they’re capable of describing more complexity, deep neural networks will theoretically perform better than shallow networks on training data.

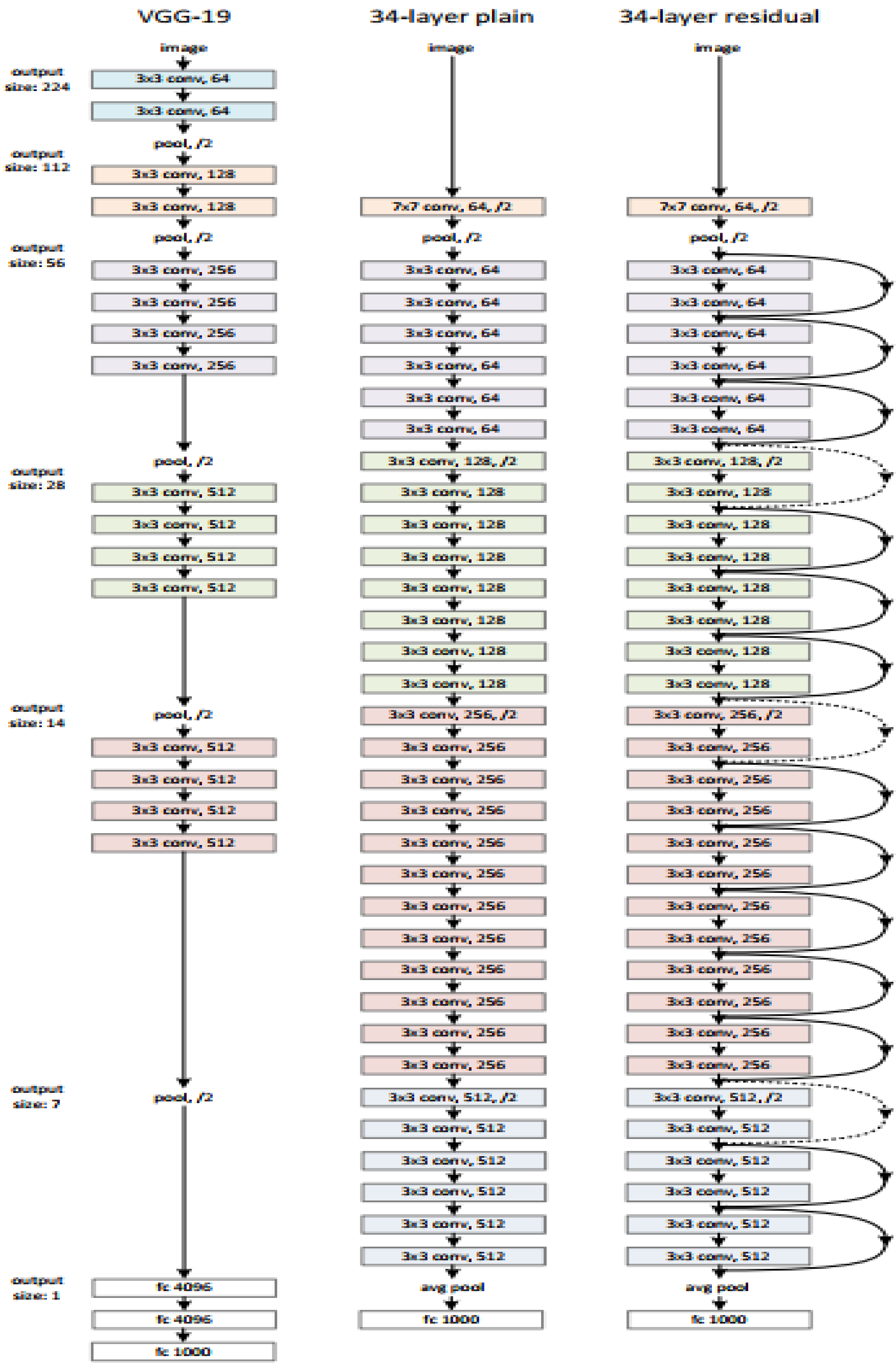

Fig 5.1Resnet Architecture

5.2 ResNet

5.2.1 Use case diagram

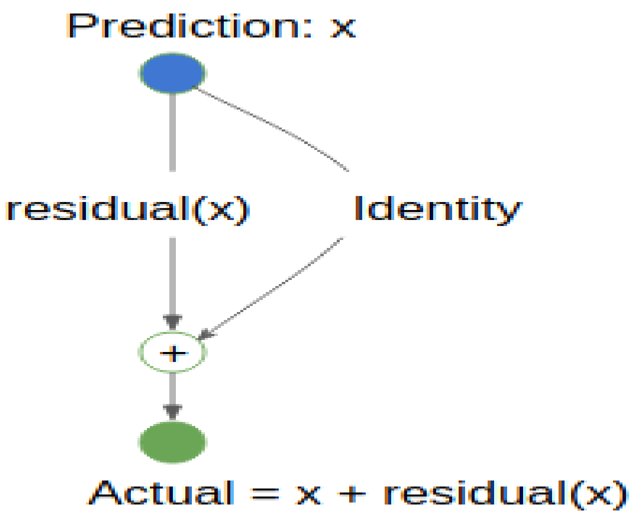

Fig 5.2.1 ResNet use case diagram

5.2.2 Implementation

- User selects data as the input contains folders named **cardboard, glass, metal, paper, plastic,** and **trash**.
- The data given as input is going to create a bunch of destination folders according to the ImageNet directory convention.
- This means it will have an outer folder which is named as **data** with three subfolders: **train**, **validation**, and **test**.
- Within each of those folders, there is a folder named **cardboard, glass, metal, paper, plastic,** and **trash**.
- And finally we are going to find a learning rate for gradient descent to make sure that my neural network converges reasonably quickly without missing the optimal error.
- **Training:** We ran our model for 20 epochs.
- This fitting method is that the learning rate decreases with each epoch, allowing us to get closer and closer to the optimum.
- At 8.6%, the validation error looks super good.

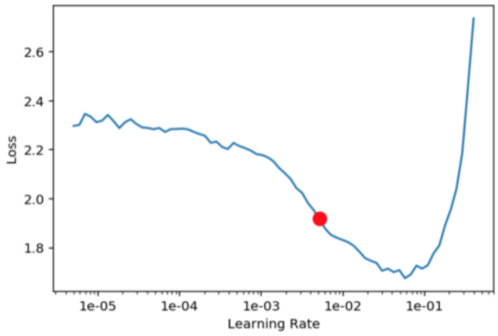

Fig 5.2.2 Figure depicting learning rate

5.2.3 Source Code

1.Importing necessary libraries:

```
from fastai.vision import *
from fastai.metrics import error_rate
from pathlib import Path
from glob2 import glob
from sklearn.metrics import confusion_matrix

import pandas as pd
import numpy as np
import os
import zipfile as zf
import shutil
import re
import seaborn as sns
```

2.Extracting data:

```
files = zf.ZipFile("dataset-resized.zip",'r')
files.extractall()
files.close()
```

3.Organizing images into different folders:

```
def split_indices(folder,seed1,seed2):
    n = len(os.listdir(folder))
    full_set = list(range(1,n+1))

    ## train indices
    random.seed(seed1)
    train = random.sample(list(range(1,n+1)),int(.5*n))
```

```
    ## temp
    remain = list(set(full_set)-set(train))

    ## separate remaining into validation and test
    random.seed(seed2)
    valid = random.sample(remain,int(.5*len(remain)))
    test = list(set(remain)-set(valid))

    return(train,valid,test)

## gets file names for a particular type of trash, given indices
    ## input: waste category and indices
    ## output: file names
def get_names(waste_type,indices):
    file_names = [waste_type+str(i)+".jpg" for i in indices]
    return(file_names)

## moves group of source files to another folder
    ## input: list of source files and destination folder
    ## no output
def move_files(source_files,destination_folder):
    for file in source_files:
        shutil.move(file,destination_folder)

## paths will be train/cardboard, train/glass, etc...
subsets = ['train','valid']
waste_types = ['cardboard','glass','metal','paper','plastic','trash']

## create destination folders for data subset and waste type
for subset in subsets:
    for waste_type in waste_types:
        folder = os.path.join('data',subset,waste_type)
        if not os.path.exists(folder):
            os.makedirs(folder)

if not os.path.exists(os.path.join('data','test')):
    os.makedirs(os.path.join('data','test'))

## move files to destination folders for each waste type
for waste_type in waste_types:
    source_folder = os.path.join('dataset-resized',waste_type)
    train_ind, valid_ind, test_ind = split_indices(source_folder,1,1)

    ## move source files to train
    train_names = get_names(waste_type,train_ind)
    train_source_files = [os.path.join(source_folder,name) for name in
train_names]
```

```
    train_dest = "data/train/"+waste_type
    move_files(train_source_files,train_dest)

    ## move source files to valid
    valid_names = get_names(waste_type,valid_ind)
    valid_source_files = [os.path.join(source_folder,name) for name in
valid_names]
    valid_dest = "data/valid/"+waste_type
    move_files(valid_source_files,valid_dest)

    ## move source files to test
    test_names = get_names(waste_type,test_ind)
    test_source_files = [os.path.join(source_folder,name) for name in
test_names]
    ## I use data/test here because the images can be mixed up
    move_files(test_source_files,"data/test")


tfms = get_transforms(do_flip=True,flip_vert=True)
data = ImageDataBunch.from_folder(path,test="test",ds_tfms=tfms,bs=16)
```

4.Model training:

```
learn = cnn_learner(data,models.resnet34,metrics=error_rate)
learn.model
```

5.Finding the learning rate:

```
learn.lr_find(start_lr=1e-6,end_lr=1e1)
learn.recorder.plot()
```

6.Training:

```
learn.fit_one_cycle(20,max_lr=5.13e-03)
```

7.Visualizing incorrect images:

```
interp = ClassificationInterpretation.from_learner(learn)
losses,idxs = interp.top_losses()

interp.plot_top_losses(9, figsize=(15,11))
```

8:Confusion matrix for visualizing predicted data:

```
doc(interp.plot_top_losses)
interp.plot_confusion_matrix(figsize=(12,12), dpi=60)
```

9:Making new predictions on test data:

```
preds = learn.get_preds(ds_type=DatasetType.Test)

print(preds[0].shape)
preds[0]

## saves the index (0 to 5) of most likely (max) predicted class for
each image
max_idxs = np.asarray(np.argmax(preds[0],axis=1))

yhat = []
for max_idx in max_idxs:
    yhat.append(data.classes[max_idx])
```

10:Test Confusion matrix:

```
cm = confusion_matrix(y,yhat)
print(cm)



df_cm = pd.DataFrame(cm,waste_types,waste_types)

plt.figure(figsize=(10,8))
sns.heatmap(df_cm,annot=True,fmt="d",cmap="YlGnBu")
```

11:Calculating Accuracy score:

```
correct = 0

for r in range(len(cm)):
    for c in range(len(cm)):
        if (r==c):
            correct += cm[r,c]

accuracy = correct/sum(sum(cm))
accuracy
```

5.2.4 Confusion matrix and Accuracy score(OUTPUT):

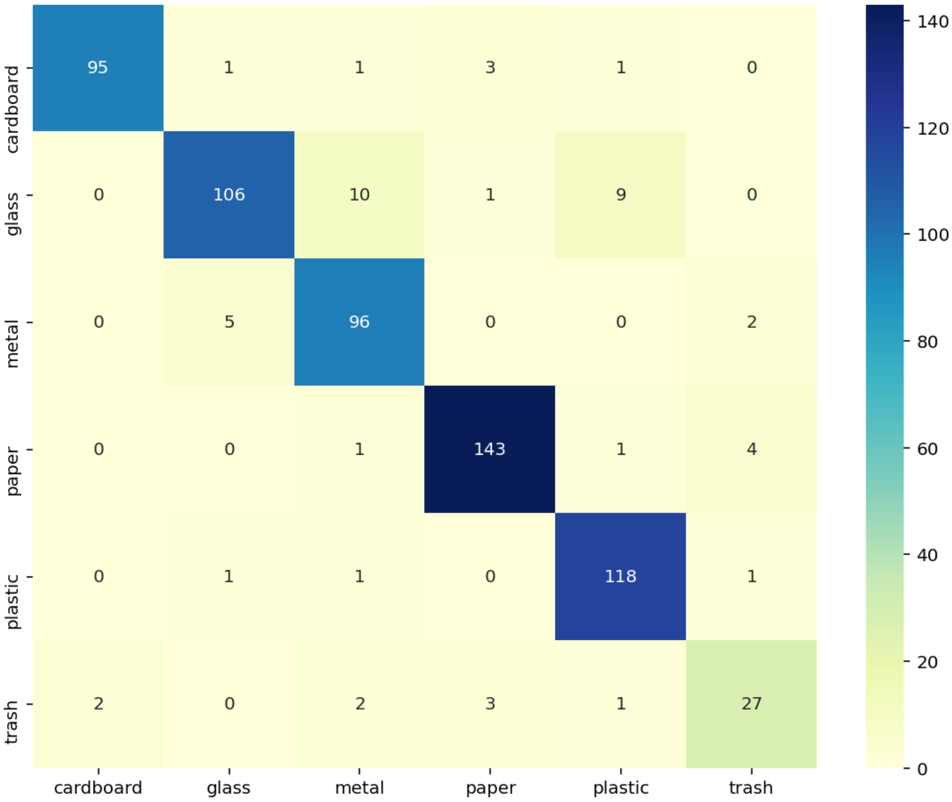

fig.5.2.4.1.Confusion matrix

```
accuracy = correct/sum(sum(cm))
accuracy
```

```
0.9212598425196851
```

fig.5.2.4.2.accuracy score

Chapter 6: Result Analysis

In this chapter we will analyze the results of the Resnet34 model and how it has performed.

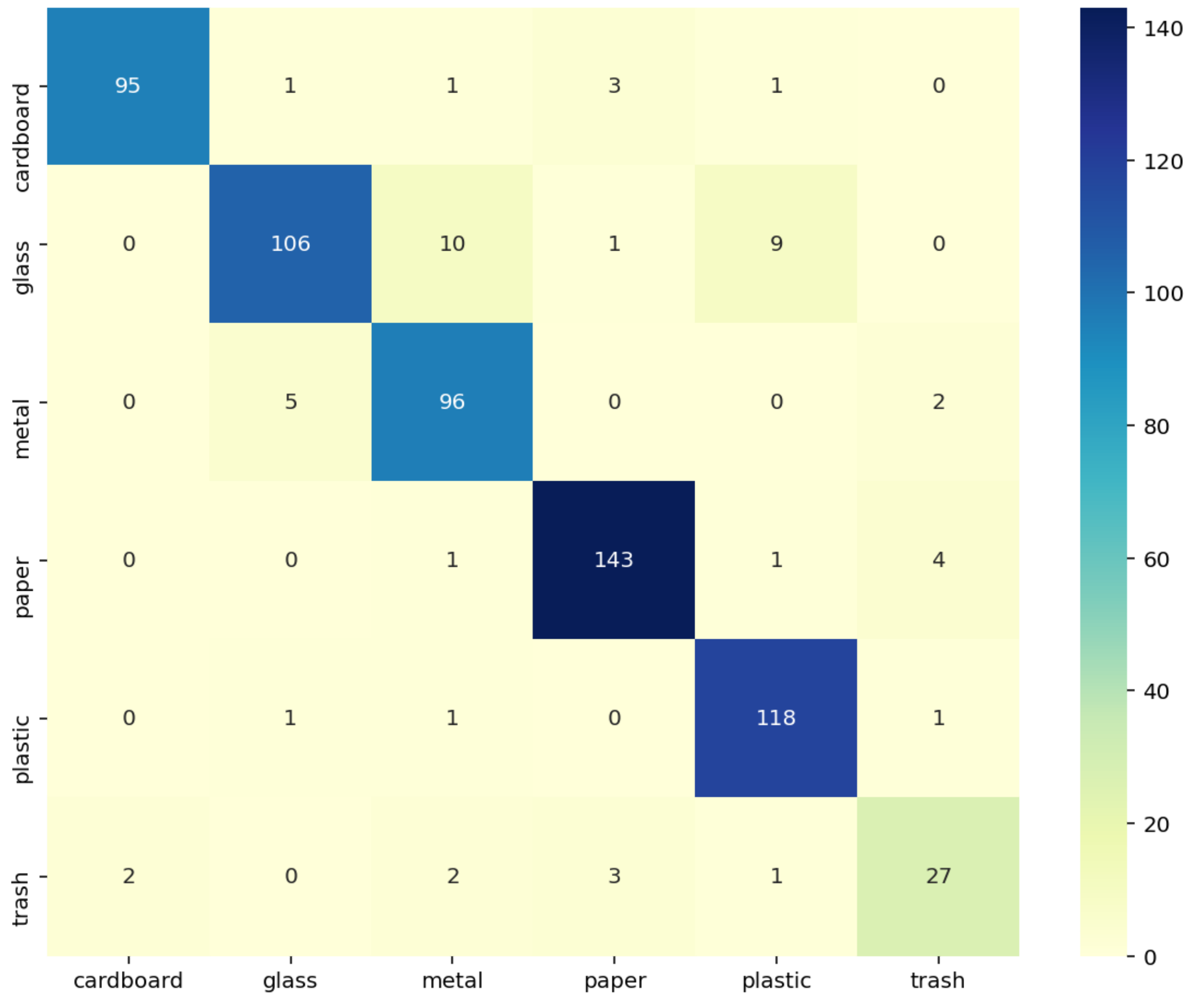

fig6.1 confusion matrix of the model

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

fig6.2 Confusion matrix

Accuracy score is calculated as follows:

Accuracy score =(TP+TN)/(TP+TN+FP+FN)

Which gave us **92%** accuracy.

Based on the above analytics and data

- This model is able to predict the debris in images with 92% accuracy.
- This model can be utilized as a state-of-the-art image classification model.
- ResNet has 25.5 million parameters and because of this it's faster.
- This model can be implemented without any attached GPU to the system unlike other models.
- Hence, **ResNet** is faster than many other models like VGG.

Chapter 7: Conclusion

7.1 Conclusion

- Marine debris & Plastic Detection using Resnet34 model is used to detect images of plastics, garbage, metal, paper, glass and cardboard in the ocean to allow for the collection and reduction of these materials, particularly plastic, in the ocean waters.
- The system can continuously improve in accuracy as it is used in the field while reducing the time and human labor costs associated with beach debris surveys to reduce different types of plastic waste on sea.
- This system monitors the spread of plastic pollution and other marine debris.
- The synthesized data will support innovative multi-disciplinary research and serve a diverse community of users.
- Based on the result analysis, this project has achieved the results more than the expected.

Solutions for addressing plastic pollution are available but will require coordinated action internationally and across a number of sectors/stakeholders.

Governments and policy change have a pivotal role to play in creating the critical legislative framework to stimulate mitigation actions that contribute to a reduction in plastic waste at source, as well as encouraging cleaning up of plastic pollution on coastlines. Education and awareness of the problems produced by plastic debris are important as a step toward changing people's behavior with regard to plastic consumption, but knowledge alone is unlikely to be sucient to change behavior.

In our view, this is an environmental problem that is largely avoidable. In short, the benefits of plastic can be realized without the need of end-of-life plastics to accumulate in the environment. Estimates of emissions to the environment indicate the severity of the issue.

It is also clear that once in the environment, plastics are highly persistent and challenging to remove. The need for action is pressing and the scale of the problem indicates no single action will be sucient. We need to simultaneously apply multiple actions including, reduction, re-use, and recyclability as a matter of urgency.

7.2 Objectives Justification

To reduce the effort, time and cost of identifying marine plastic debris.

- Model predicts the marine debris with no extra time required.It is a cost effective, effortless process to find debris in the pictures.

Data Preprocessing:

- Some images that we collected consist of plastic that is in water and is less illuminated. So, we segregated the pictures with less brightness by converting it into HSV format. We have set a threshold value to find the images with lower brightness to illuminate the image. To illuminate the images, we may use some Algorithms.

Reduction of problems related to Health and Hygiene

- The Marine Debris Plastic Detection helps to reduce plastic on sea, This makes marine life to live better.Since it is used in marine debris detection and removal which in turn result in healthy marine life sustainability.

Help fishermen and develops tourism

- Marine debris detection and removal helps in healthy marine life which gives livelihood for fishermen and helps developing national tourism revenue.

7.3 Future Enhancement

Lack of standardized protocols for plastic litter detection, sampling, and extraction creates issues for the comparability of data. Methods need to be improved in order to categorize different size classes (including Nano plastics), sampling procedures, analytical methods, and reference materials. Hartmannetal. recently proposed a definition and categorization framework.

This framework went beyond size classes to include physicochemical properties (polymer composition, solid state, solubility) as defining criteria with size, shape, color, and origin as classifiers for categorization.

There is also uncertainty about the specific extent and magnitude of the harm of plastic pollution in the marine environment. However, most agree that there is too much litter in the environment. The challenge is to take the most appropriate actions and to fit these interventions to particular causes of marine litter.

Chapter 8: References / Bibliography

1. Authors:Lizz Allen ,
   How Satellites And Machine Learning Are Being Used To Detect Plastic In The Ocean,
   forbes
2. Authors:N Maximenko, P Corradi, KL Law,
   Toward the Integrated Marine Debris Observing System,
   Google scholar
3. Authors:Andrew kokoulin,Alexander I.Tur,Alexander A.Yuzakhov,
   Convolutional neural networks application in plastic waste recognition and sorting,
   IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)
4. Authors:Thung Yang,
   Classification of Trash for Recyclability Status ,
   Leland stanford junior university
5. Authors:Imogan Ellen Epaper,
   Plastic Debris in the Marine Environment: History and Future Challenges,
   Research gate
6. Authors:Mattis Wolf, Katelijn van den Berg,Shungudzemwoyo P Garaba1,2, Nina Gnann1, Klaus Sattler3 , Frederic Stahl1,4 and Oliver Zielinski,
   Machine learning for aquatic plastic litter detection, classification and quantification (APLASTIC-Q)
7. Resnet34
   https://models.roboflow.com/classification/resnet34Roboflow
8. K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *CVPR*, 2016.
   Understanding and visualizing ResNets
   Towards data science