A

Major Project

On

AN AUTOMATIC GARBAGE CLASSIFICATION SYSTEM

(Submitted in partial fulfillment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

K.N.V Raja Sekhar	(177R1A0587)
B. Jayanth	(177R1A0570)
K.Santhosh	(177R1A0579)

Under the Guidance of

DR. M.VARAPRASAD RAO

(Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

2017-21

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the project entitled "**AN AUTOMATIC GARBAGE CLASSIFICATION SYSTEM**" being submitted by **K.N.V RAJASEKHAR(177R1A0587), B.JAYANTH(177R1A0570) &K.SANTOSH(177R1A0579)** in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, is a record of bonafide work carried out by him/her under our guidance and supervision during the year 2020-21.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

INTERNAL GUIDE
Dr. M. Varaprasad Rao
Professor

DIRECTOR
Dr.A. Raji Reddy

HOD
Dr. K. Srujan Raju

EXTERNAL EXAMINER

Submitted for viva voice Examination held on ________________

ACKNOWLEGDEMENT

Apart from the efforts of us, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We take this opportunity to express my profound gratitude and deep regard to my guide.

Dr. M. Varaprasad Rao, Professor for his exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by him shall carry us a long way in the journey of life on which we are about to embark. We also take this opportunity to express a deep sense of gratitude to Project Review Committee (PRC) Coordinators: **Mr. J. Narasimha Rao, Mr. B. P. Deepak Kumar, Mr. K. Murali, Dr. Suwarna Gothane and Mr. B. Ramji** for their cordial support, valuable information and guidance, which helped us in completing this task through various stages.

We are also thankful to **Dr. K. Srujan Raju,** Head, Department of Computer Science and Engineering for providing encouragement and support for completing this project successfully.

We are obliged to **Dr. A. Raji Reddy,** Director for being cooperative throughout the course of this project. We would like to express our sincere gratitude to Sri. **Ch. Gopal Reddy,** Chairman for providing excellent infrastructure and a nice atmosphere throughout the course of this project.

The guidance and support received from all the members of **CMR Technical Campus** who contributed to the completion of the project. We are grateful for their constant support and help.

Finally, we would like to take this opportunity to thank our family for their constant encouragement, without which this assignment would not be completed. We sincerely acknowledge and thank all those who gave support directly and indirectly in the completion of this project.

K.N.V Raja Sekhar (177R1A0587)

B. Jayanth (177R1A0570)

K.Santhosh (177R1A0579)

ABSTRACT

Trash arrangement has consistently been a significant issue in ecological security, As the economy is developing rapidly and with the improvement of people’s living standards, the amount of garbage is also increasing rapidly. To tackle the provincial living trash handling issues, we freely built up a one of a kind innovation of perfect and far reaching use of trash assets. Based on modern bio-degradation and automatic sorting, we built a complete industrial chain with mixed domestic waste as mineral raw material. The natural discharge would be siphoned to anaerobic maturation framework, while the buildup would be another puncturing through broken drum to fundamental gathering framework. By our advanced gadgetry, we secure inorganic material as sand and glass, mash and sundries, for example, bamboo materials, and unadulterated plastic finally. The items would be extra preparing to mechanical materials as expected of market. The sand and rock glass and other inorganic materials can deliver empty squares, clearing block and wipe block. At last, the prevalence of the proposed grouping calculation is checked with the built trash information. The order exactness of the proposed calculation is upgraded by 1.01%. The trial results show that the grouping precision is pretty much as high as 95%, the characterization pattern of the framework is pretty much as brisk as 0.95 s.

LIST OF FIGURES

LIST OF SCREENSHOTS

TABLE OF CONTENTS

1. INTRODUCTION

1.INTRODUCTION

1.1 PROJECT SCOPE

This project is titled as “An Automatic garbage classification system”. This software provides facility to upload the images and classify the garbage . This project uses deep-learning methods and computer vision to classify the garbage into different classes. First, we use a resnet-50 to classify garbage images into different classes like plastic, cardboard, glass etc We then compare a number of classification algorithms that use certain features to predict the garbage class shown in the image.

1.2 PROJECT PURPOSE

This has been developed to facilitate the identification, retrieval of the items and information. System is built with manually exclusive features. In all cases system will specify object which are physical or on performance characteristics. They are used to give optimal distraction and other information. Data are used for identifying, accessing, storing and matching records. The data ensures that only one value of the code with a single meaning is correctly applied to give entity or attribute as described in various ways.

1.3 PROJECT FEATURES

The main features of this project are that the designer now functions as a problem solver and tries to sort out the difficulties that the enterprise faces. The solutions are given as proposals. The proposal is then weighed with the existing system analytically and the best one is selected. The proposal is presented to the user for an endorsement by the user. The proposal is reviewed on user request and suitable changes are made. This is loop that ends as soon as the user is satisfied with proposal.

2.SYSTEM ANALYSIS

2. SYSTEM ANALYSIS

SYSTEM ANALYSIS

System Analysis is the important phase in the system development process. The System is studied to the minute details and analyzed. The system analyst plays an important role of an interrogator and dwells deep into the working of the present system. In analysis, a detailed study of these operations performed by the system and their relationships within and outside the system is done. A key question considered here is, “what must be done to solve the problem?” The system is viewed as a whole and the inputs to the system are identified. Once analysis is completed the analyst has a firm understanding of what is to be done.

2.1 PROBLEM DEFINITION

Trash arrangement has consistently been a significant issue in ecological security. As the people’s living standards are increasing , the amount of garbage is also increasing rapidly.To improve the efficiency of garbage collection, an automatic garbage classification system would be very beneficial and is proposed based on deep learning.The proposed garbage classification algorithm is based on “ResNet-50” algorithm.

2.2 EXISTING SYSTEM

Kano et al. proposed a group-control robot system, which provides a decentralized control scheme for indoor garbage collection. But it depends on the cooperation between robots and takes a long time.In countries like India still doesn’t know about the garbage classification system,we are still doing it manually.But it depends on the cooperation between robots and takes a long time. P. Wesley et al. proposed a low-cost intelligence garbage bin.

2.2.1 LIMITATIONS OF EXISTING SYSTEM

- Much time is being consumed
- Data acquisition
- Highly error-prone
- Includes high cost

2.3 PROPOSED SYSTEM

The proposed garbage classification algorithm is based on ResNet-50 algorithm, and its network structure is further optimized by three aspects, including the multi feature fusion of input images, the feature reuse of the residual unit, and the design of a new activation function.

2.3.1 ADVANTAGES OF THE PROPOSED SYSTEM

The system is very simple in design and to implement. The system requires very low system resources and the system will work in almost all configurations. It has got following features

- Performs the task consistently.
- Provides better accuracy.
- Simple and easy to use .

2.4 FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis are

- Economic Feasibility
- Technical Feasibility
- Social Feasibility

2.4.1 ECONOMIC FEASIBILITY

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. One of the factors, which affect the development of a new system, is the cost it would require.

The following are some of the important financial questions asked during preliminary investigation:

- The costs conduct a full system investigation.
- The cost of the hardware and software.
- The benefits in the form of reduced costs or fewer costly errors.

2.3.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.3.3 BEHAVIORAL FEASIBILITY

This includes the following questions:

- Is there sufficient support for the users?
- Will the proposed system cause harm?

The project would be beneficial because it satisfies the objectives when developed and installed. All behavioral aspects are considered carefully and conclude that the project is behaviorally feasible.

2.5 HARDWARE & SOFTWARE REQUIREMENTS

2.5.1 HARDWARE REQUIREMENTS:

Hardware interfaces specifies the logical characteristics of each interface between the software product and the hardware components of the system. The following aresome hardware requirements.

- Processor : Intel i3
- Ram : 4 gb
- Hard Disk : 250 gb

2.5.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and ofthe system. The following are some software requirements,

- Operating System : Windows 8 or above
- Programming Language : Python
- Library : Tensorflow , keras, matplotlib , numpy

3.ARCHITECTURE

3.ARCHITECTURE

3.1 PROJECT ARCITECTURE

This project architecture shows the procedure followed for garbage classification using resnet50, starting from input to final prediction.

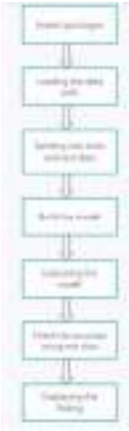


Figure 3.1: Project Architecture of Garbage Classification

3.2 DESCRIPTION

Image Acquisition: In this phase, images of plant leaves are gathered using digital media like camera, mobile phones etc. with desired resolution and size. The formation of database of images is completely dependent on the application system developer.

Image Segmentation: This phase aims at simplifying the classification of garbage such that it becomes more meaningful and easier to analyze.

Feature Extraction: In this step the features from this area of interest need to be extracted. These features are needed to determine the meaning of a sample image.Features can be based on colour, shape, and texture. Recently, most of the researchers are intending to use texture features for classification of garbage collection.

Training and test data: Training data is passed to train the model. Test data is used to test the trained model whether it is making correct predictions or not.

Classification : The classification phase implies to determine if the input image is classified correctly or not. If the image is found to be classified, some existing works have further classified it into a number of classes.

3.3 USE CASE DIAGRAM

In the use case diagram we have basically one actor who is the user. The user takes the picture and uploads it. Whereas the code does the data import and train and test model and gives accuracy.

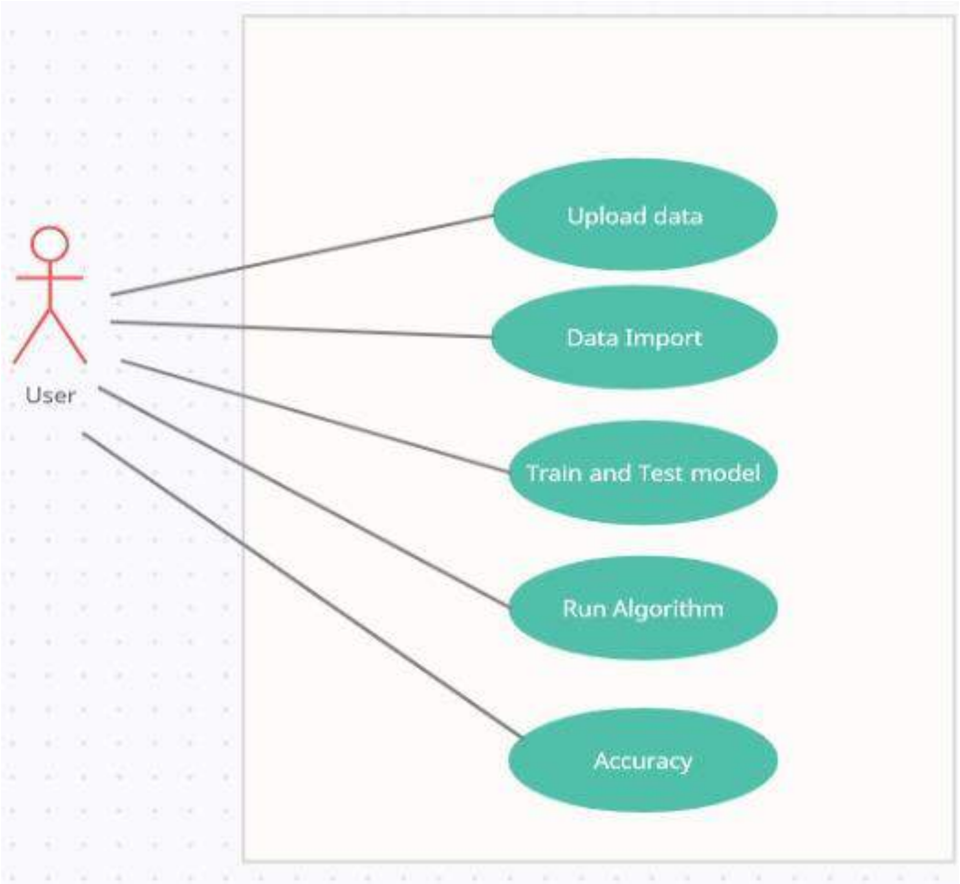


Figure 3.2: Use Case Diagram for Garbage Classification System

3.4 CLASS DIAGRAM

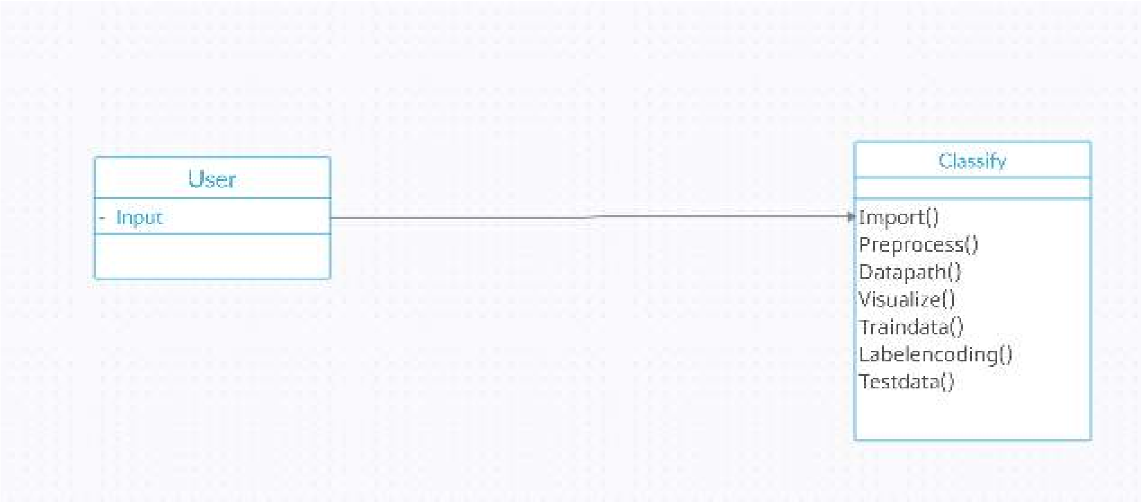


Figure 3.3: Class Diagram for Garbage Classfication System

3.4 ACTIVITY DIAGRAM

It describes about flow of activity states.

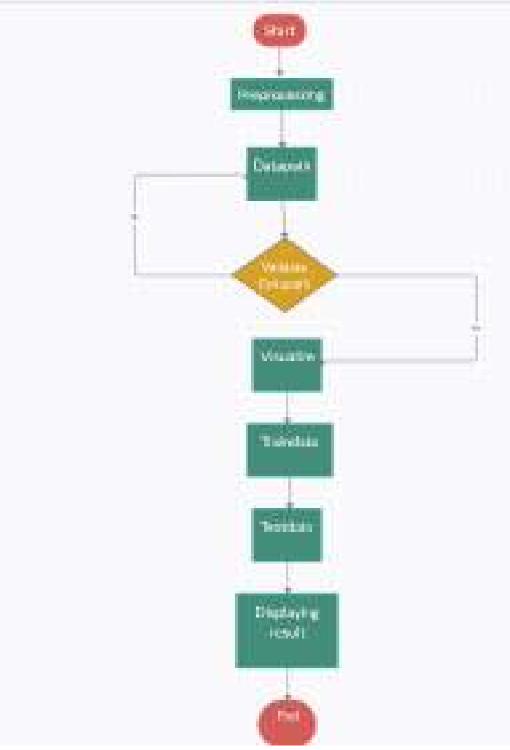


Figure 3.5: Activity Diagram for Garbage Classification System

4.IMPLEMENTATION

4.IMPLEMENTATION

4.1 RESNET-50 CODE

```
# dependencies
import matplotlib.pyplot as plt
import numpy as np
import os
import tensorflow as tf

# where the data is located
data_dir = '/content/drive/MyDrive/Garbage classification'

# review number of files and directories in the dataset
total_dir = len(os.listdir(data_dir))
total_files = 0
for dirname, _, filenames in os.walk(data_dir):
    print('counting:', dirname)
    files_counter = 0
    for file in filenames:
        files_counter += 1
    total_files += files_counter
    print('total files in dir:', files_counter)
print('--------')
print('total number of files',total_files)
print('total number of directories',total_dir)

# create datasets
train_ds = tf.keras.preprocessing.image_dataset_from_directory(
    data_dir,
    validation_split=0.2,
    subset='training',
    seed=100
)

validation_ds = tf.keras.preprocessing.image_dataset_from_directory(
    data_dir,
    validation_split=0.2,
    subset='validation',
    seed=100
)

# get class names
class_names = train_ds.class_names
print(class_names)
```

```
# view some images from the train_ds
plt.figure(figsize=(16, 16))
for images, labels in train_ds.take(1):
    for i in range(6):
        ax = plt.subplot(2, 3, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(class_names[labels[i]])
        plt.axis("off")


# view some images from the validation_ds
plt.figure(figsize=(16, 16))
for images, labels in validation_ds.take(1):
    for i in range(6):
        ax = plt.subplot(2, 3, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(class_names[labels[i]])
        plt.axis("off")


# inspect the train_ds
train_batch = train_ds.as_numpy_iterator().next()

print('total of batches:',len(train_ds))
print('images batch shape:',train_batch[0].shape)
print('labels batch shape:',train_batch[1].shape)


# inspect the train_ds
validation_batch = validation_ds.as_numpy_iterator().next()

print('total of batches:',len(validation_ds))
print('images batch shape:',validation_batch[0].shape)
print('labels batch shape:',validation_batch[1].shape)


# instantiate the base model
input_shape = (256,256,3)
base_model = tf.keras.applications.ResNet50V2(include_top=False, input_shape=input_shape)

# make the layers of the model trainable to fine-tunning
base_model.trainable = True
```

```
# find the tunning layer and its index
tuning_layer_name = 'conv5_block1_preact_bn'
tuning_layer = base_model.get_layer(tuning_layer_name)
tuning_index = base_model.layers.index(tuning_layer)

# freeze all the layers before the tuning layer
for layer in base_model.layers[:tuning_index]:
    layer.trainable =  False

# create a data augmentation stage with horizontal and vertical flipping, rotations and zooms
data_augmentation = tf.keras.Sequential([
    tf.keras.layers.experimental.preprocessing.Rescaling(1./127.5, offset= -1),
    tf.keras.layers.experimental.preprocessing.RandomFlip("horizontal_and_vertical"),
    tf.keras.layers.experimental.preprocessing.RandomRotation(0.2),
    tf.keras.layers.experimental.preprocessing.RandomZoom(0.2)
], name='data_augmentation')

# create the neural network architecture
model = tf.keras.Sequential([
    data_augmentation,
    base_model,
    tf.keras.layers.GlobalAveragePooling2D(),
    tf.keras.layers.Dense(6, activation='softmax')
])

learning_rate = 0.00001
model.compile(
    loss='sparse_categorical_crossentropy',
    optimizer=tf.keras.optimizers.Adam(lr=learning_rate),
    metrics=['accuracy']
)

# train the model
history = model.fit(train_ds,validation_data=validation_ds,epochs=20)

# visualize the training history
acc = history.history['accuracy']
val_acc = history.history['val_accuracy']

loss = history.history['loss']
val_loss = history.history['val_loss']
```

```
# plot accuracy
plt.figure(figsize=(8, 8))
plt.subplot(2, 1, 1)
plt.plot(acc, label='Training Accuracy')
plt.plot(val_acc, label='Validation Accuracy')
plt.legend(loc='lower right')
plt.ylabel('Accuracy')
plt.ylim([min(plt.ylim()),1])
plt.title('Training and Validation Accuracy')

plt.subplot(2, 1, 2)
plt.plot(loss, label='Training Loss')
plt.plot(val_loss, label='Validation Loss')
plt.legend(loc='upper right')
plt.ylabel('Cross Entropy')
plt.ylim([0,1.0])
plt.title('Training and Validation Loss')
plt.xlabel('epoch')
plt.show()

# verify the performance of the model
loss, accuracy = model.evaluate(train_ds)
print('Test accuracy :', accuracy)
print('Test loss:', loss)

# get a batch from validation_ds to do some inference
image_batch, label_batch = validation_ds.as_numpy_iterator().next()

# inference
inference = model.predict_on_batch(image_batch)

# show imgs and labels
plt.figure(figsize=(16, 16))
for i in range(12):
    ax = plt.subplot(4, 4, i + 1)
    plt.imshow(image_batch[i].astype("uint8"))
    plt.title('pred:{}, {:.2f}% Confidence\ntruth:{}'
            .format(class_names[np.argmax(inference[i])], 100 * np.max(inference[i]), class_names[label_batch[i]]))
    plt.axis("off")
```

4.1 SEQUENTIAL CODE

```
#dependencies
import numpy as np
import matplotlib.pyplot as plt
from keras.preprocessing.image import ImageDataGenerator, load_img, img_to_array, array_to_img
from keras.layers import Conv2D, Flatten, MaxPooling2D, Dense
from keras.models import Sequential
import glob, os, random

#datapath
base_path = '/content/drive/MyDrive/Garbage classification'
img_list = glob.glob(os.path.join(base_path, '*/*.jpg'))
print(len(img_list))

#displaying six random images
for i, img_path in enumerate(random.sample(img_list, 6)):
    img = load_img(img_path)
    img = img_to_array(img, dtype=np.uint8)
    plt.subplot(2, 3, i+1)
    plt.imshow(img.squeeze())

train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.1,
    zoom_range=0.1,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True,
    vertical_flip=True,
    validation_split=0.1
)

test_datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.1
)

train_generator = train_datagen.flow_from_directory(
    base_path,
     target_size=(300, 300),
    batch_size=16,
    class_mode='categorical',
    subset='training',
    seed=0
)
```

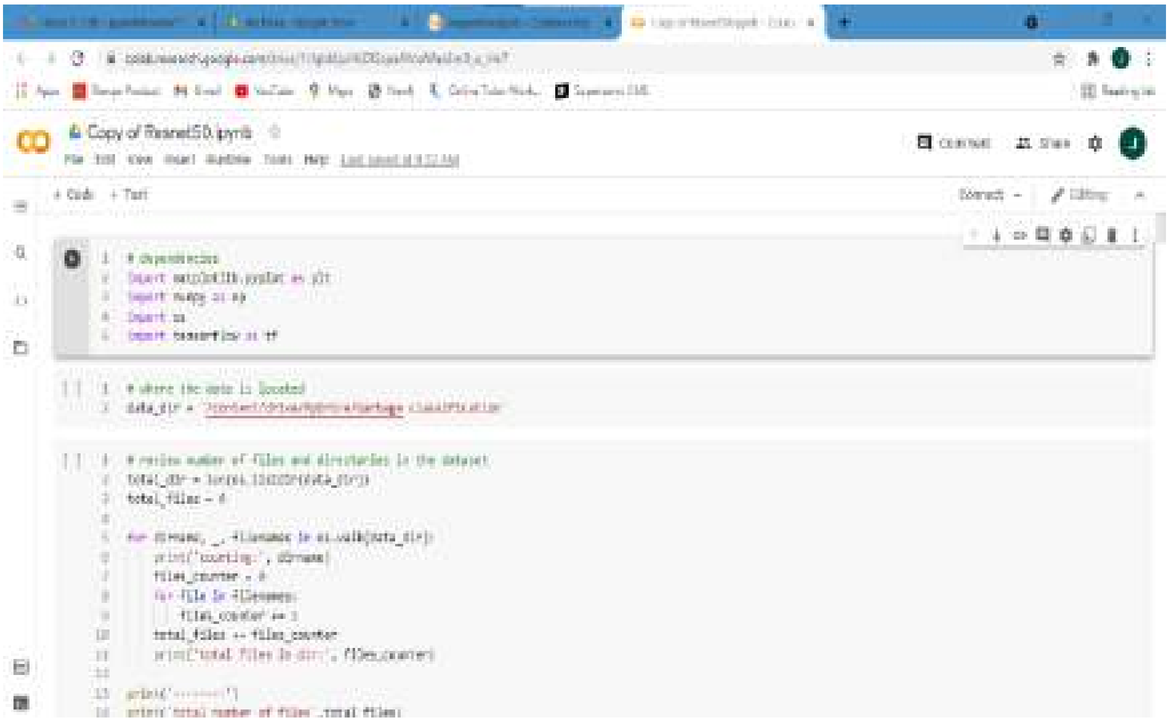
```
validation_generator = test_datagen.flow_from_directory(
    base_path,
    target_size=(300, 300),
    batch_size=16,
    class_mode='categorical',
    subset='validation',
    seed=0
)

#displaying labels of classes
labels = (train_generator.class_indices)
labels = dict((v,k) for k,v in labels.items())
print(labels)

#instantiate the model
model = Sequential([
    Conv2D(filters=32, kernel_size=3, padding='same', activation='relu', input_shape=(300, 300, 3)),
    MaxPooling2D(pool_size=2),

    Conv2D(filters=64, kernel_size=3, padding='same', activation='relu'),
    MaxPooling2D(pool_size=2),

    Conv2D(filters=32, kernel_size=3, padding='same', activation='relu'),
    MaxPooling2D(pool_size=2),

    Conv2D(filters=32, kernel_size=3, padding='same', activation='relu'),
    MaxPooling2D(pool_size=2),

    Flatten(),

    Dense(64, activation='relu'),

    Dense(6, activation='softmax')
])

model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['acc'])
model.summary()

#training the model
hist=model.fit_generator(train_generator, epochs=20, validation_data=validation_generator)

#display the accuracy and loss
loss,accuracy=model.evaluate(validation_generator)
print("Test accuracy :",accuracy)
print("Test loss :",loss)



# Plot the loss and accuracy curves for training and validation
```

```
fig, ax = plt.subplots(2,1)
ax[0].plot(hist.history['loss'], color='b', label="Training loss")
ax[0].plot(hist.history['val_loss'], color='r', label="validation loss",axes =ax[0])
legend = ax[0].legend(loc='best', shadow=True)

ax[1].plot(hist.history['acc'], color='b', label="Training accuracy")
ax[1].plot(hist.history['val_acc'], color='r',label="Validation accuracy")
legend = ax[1].legend(loc='best', shadow=True)

#display the output images with the labels
test_x, test_y = validation_generator.__getitem__(1)

preds = model.predict(test_x)

plt.figure(figsize=(16, 16))
for i in range(16):
    plt.subplot(4, 4, i+1)
    plt.title('pred:%s / truth:%s' % (labels[np.argmax(preds[i])], labels[np.argmax(test_y[i])]))
    plt.imshow(test_x[i])
```

5.SCREENSHOTS

5.1 IMPORTING PACKAGES

Import the packages which are required to implement the process and set the data path i.e., locate the dataset file path.



```
# dependencies
import matplotlib.pyplot as plt
import numpy as np
import os
import tensorflow as tf
```

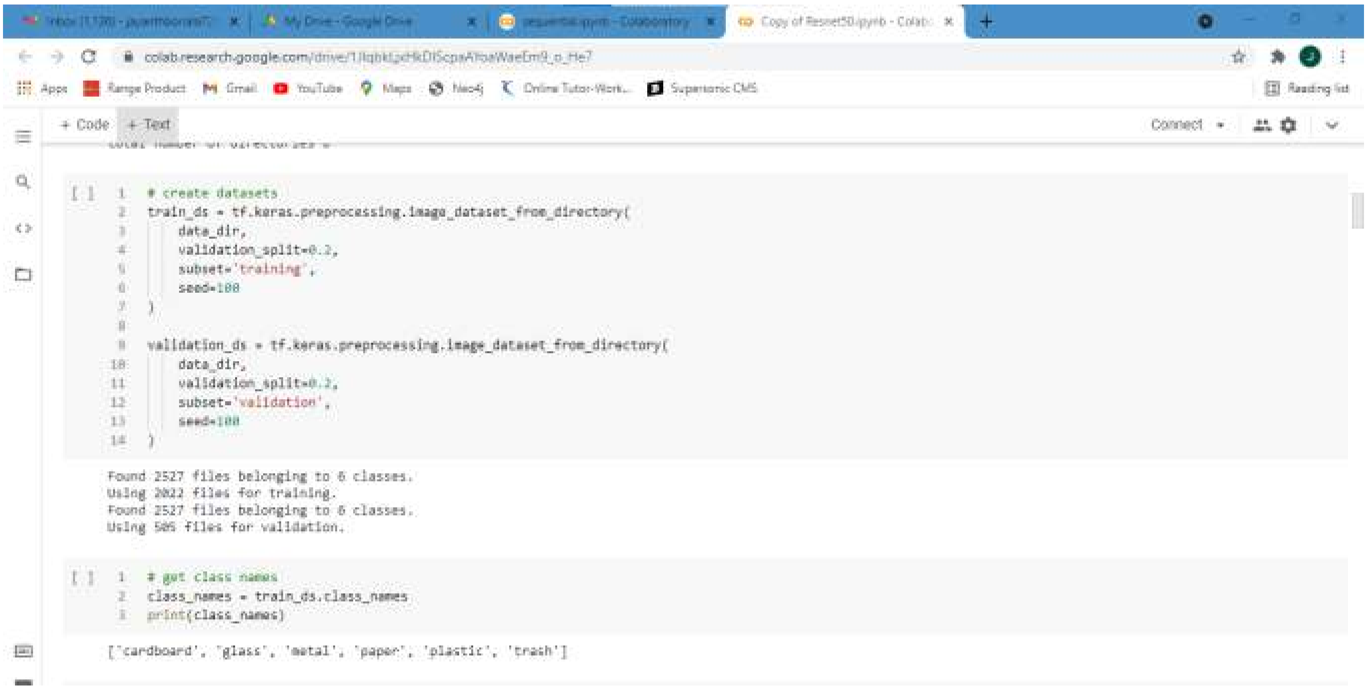
```
# where the data is located
data_dir = '/content/drive/MyDrive/garbage classification'
```

```
# review number of files and directories in the dataset
total_dir = len(os.listdir(data_dir))
total_files = 0

for dirname, _, filenames in os.walk(data_dir):
    print('counting:', dirname)
    files_counter = 0
    for file in filenames:
        files_counter += 1
    total_files += files_counter
    print('total files in dir:', files_counter)

print('--------')
print('total number of files',total_files)
```

Screenshot 5.1: Importing packages

5.2 TRAININGSET AND VALIDATIONSET

Divide the dataset into training and validation sets in the ratio of 80% and 20%, The total 2527 images are divided into 2022 as training images and 505 as validation images.



```
# create datasets
train_ds = tf.keras.preprocessing.image_dataset_from_directory(
    data_dir,
    validation_split=0.2,
    subset='training',
    seed=100
)

validation_ds = tf.keras.preprocessing.image_dataset_from_directory(
    data_dir,
    validation_split=0.2,
    subset='validation',
    seed=100
)
```

```
Found 2527 files belonging to 6 classes.
Using 2022 files for training.
Found 2527 files belonging to 6 classes.
Using 505 files for validation.
```

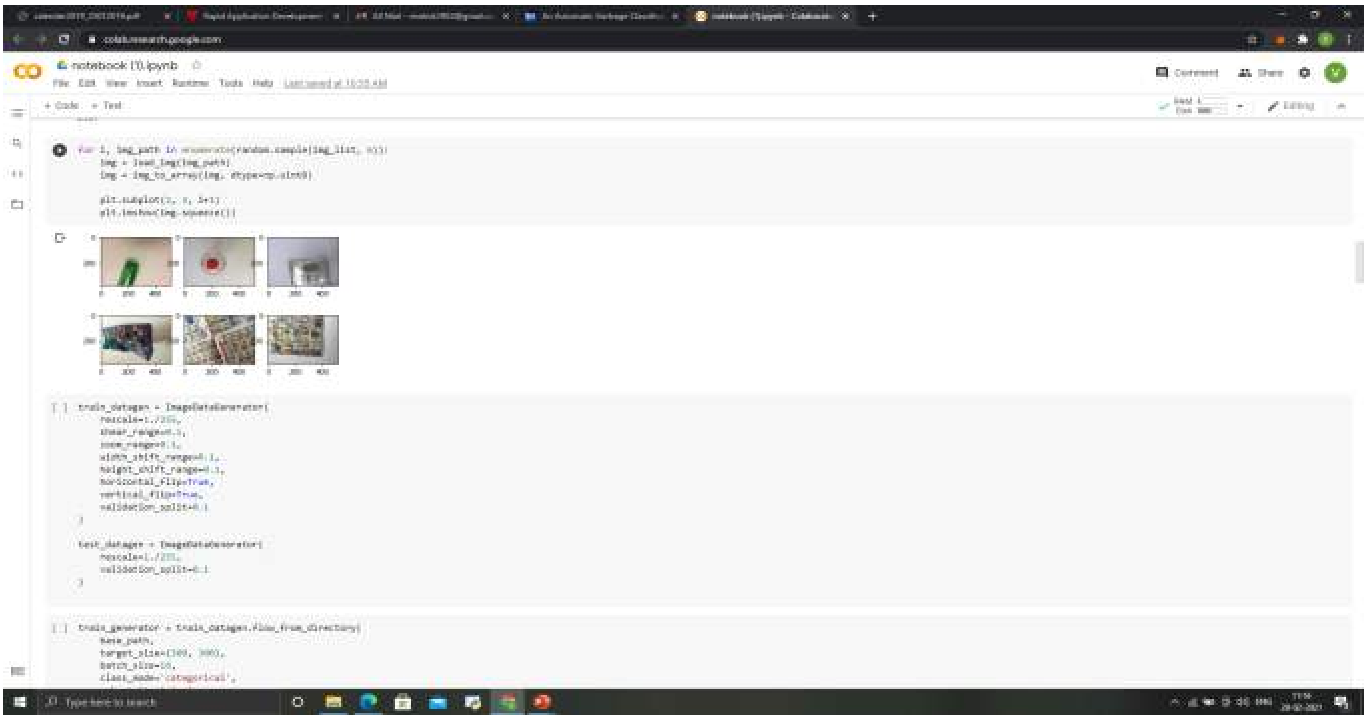
```
# get class names
class_names = train_ds.class_names
print(class_names)
```

```
['cardboard', 'glass', 'metal', 'paper', 'plastic', 'trash']
```

Screenshot 5.2: Training set and validation set

5.3 DISPLAYING IMAGES

Display randomly six images from the training and validation sets.



```
for i, img_path in enumerate(random.sample(img_list, 6)):
    img = load_img(img_path)
    img = img_to_array(img, dtype=np.uint8)

    plt.subplot(2, 3, i+1)
    plt.imshow(img.squeeze())
```

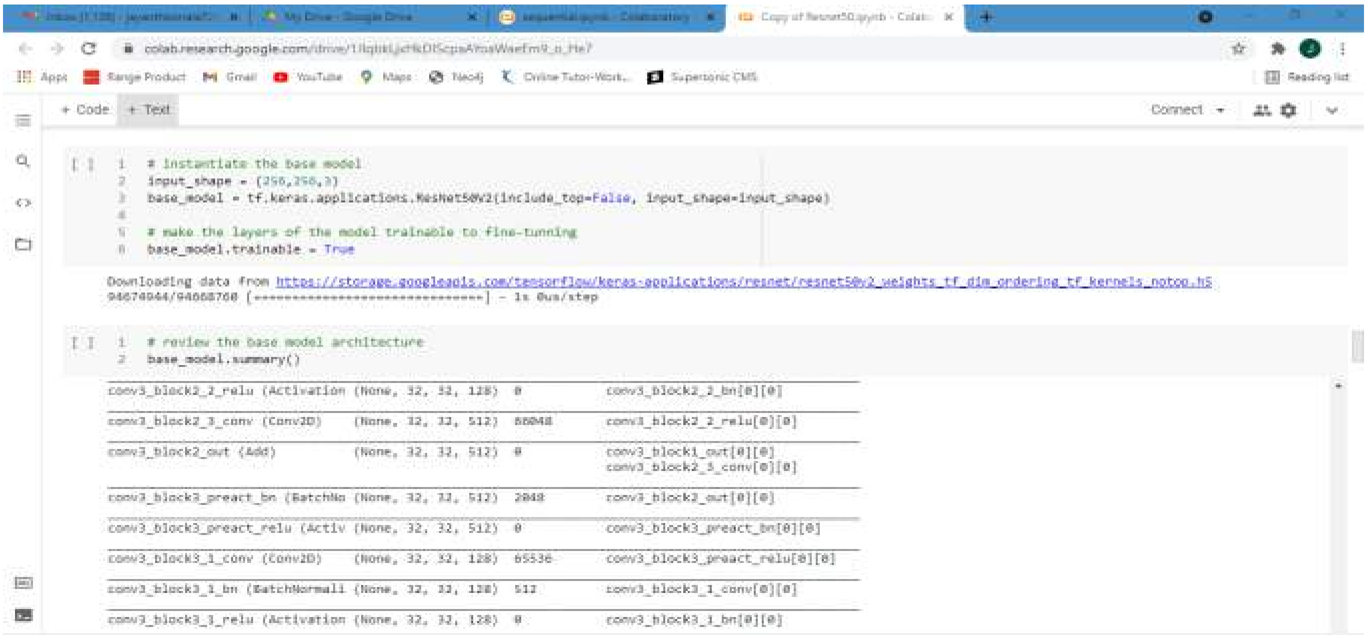
```
train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.1,
    zoom_range=0.1,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True,
    vertical_flip=True,
    validation_split=0.1
)

test_datagen = ImageDataGenerator(
    rescale=1./255,
    validation_split=0.1
)
```

```
train_generator = train_datagen.flow_from_directory(
    base_path,
    target_size=(300, 300),
    batch_size=16,
    class_mode='categorical',
```

Screenshot 5.3 Displaying images

5.4 INITIATING RESNET-50 MODEL

Initiate the resnet-50 model.



```
# instantiate the base model
input_shape = (256,256,3)
base_model = tf.keras.applications.ResNet50V2(include_top=False, input_shape=input_shape)

# make the layers of the model trainable to fine-tunning
base_model.trainable = True
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50v2_weights_tf_dim_ordering_tf_kernels_notop.h5
94674944/94668760 [==============================] - 1s 0us/step
```

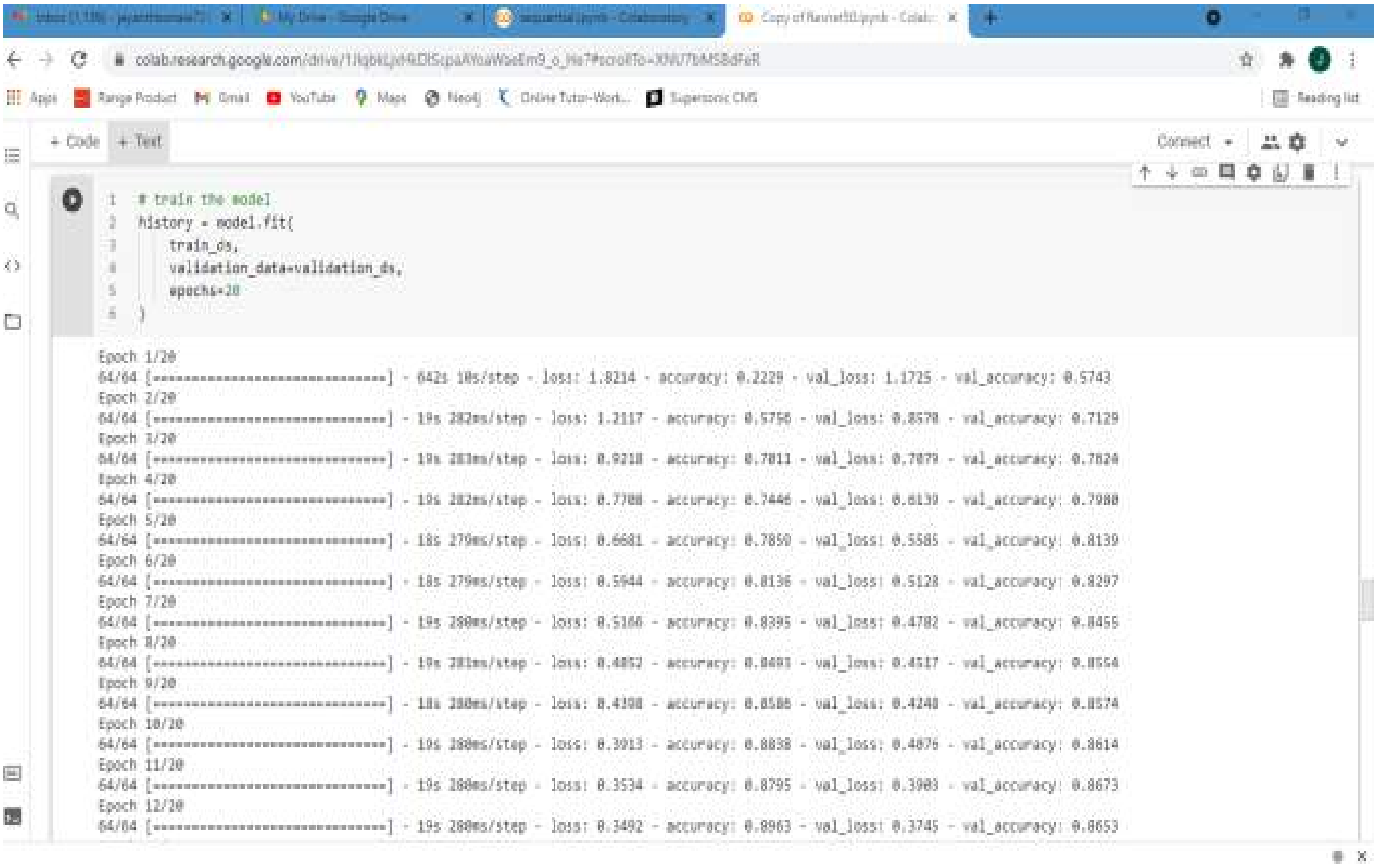
```
# review the base model architecture
base_model.summary()
```

```
____________________________________________________________________________________________
conv3_block2_2_relu (Activation (None, 32, 32, 128)  0           conv3_block2_2_bn[0][0]
____________________________________________________________________________________________
conv3_block2_3_conv (Conv2D)    (None, 32, 32, 512)  66048       conv3_block2_2_relu[0][0]
____________________________________________________________________________________________
conv3_block2_out (Add)          (None, 32, 32, 512)  0           conv3_block1_out[0][0]
                                                                 conv3_block2_3_conv[0][0]
____________________________________________________________________________________________
conv3_block3_preact_bn (BatchNo (None, 32, 32, 512)  2048        conv3_block2_out[0][0]
____________________________________________________________________________________________
conv3_block3_preact_relu (Activ (None, 32, 32, 512)  0           conv3_block3_preact_bn[0][0]
____________________________________________________________________________________________
conv3_block3_1_conv (Conv2D)    (None, 32, 32, 128)  65536       conv3_block3_preact_relu[0][0]
____________________________________________________________________________________________
conv3_block3_1_bn (BatchNormali (None, 32, 32, 128)  512         conv3_block3_1_conv[0][0]
____________________________________________________________________________________________
conv3_block3_1_relu (Activation (None, 32, 32, 128)  0           conv3_block3_1_bn[0][0]
```

Screenshot 5.4: Initiating resnet-50 model

5.5 TRAINING THE MODEL

Train the model by giving the epochs size as 20, it means it trains the model twenty times.



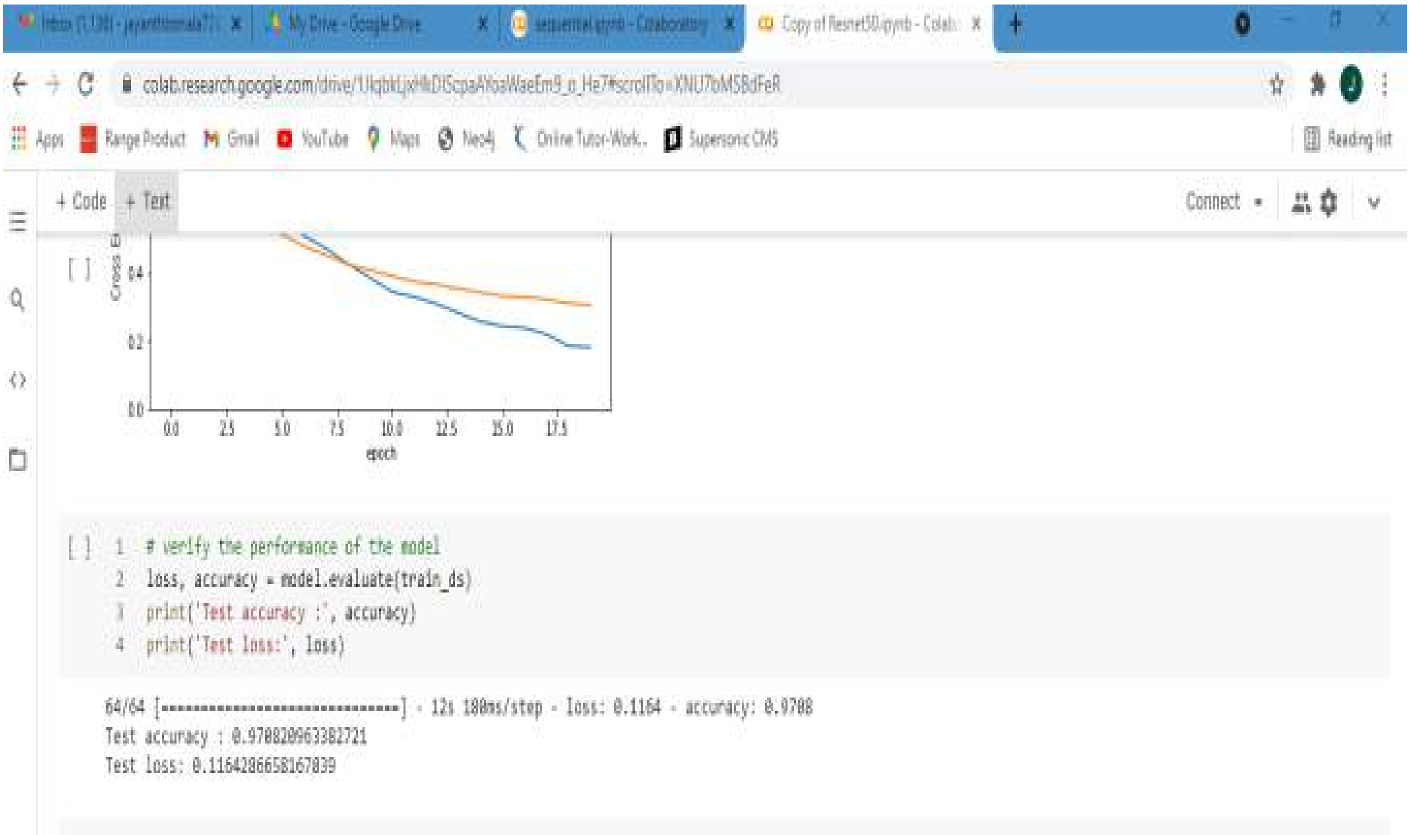
```
# train the model
history = model.fit(
    train_ds,
    validation_data=validation_ds,
    epochs=20
)
```

```
Epoch 1/20
64/64 [==============================] - 642s 10s/step - loss: 1.8214 - accuracy: 0.2229 - val_loss: 1.1725 - val_accuracy: 0.5743
Epoch 2/20
64/64 [==============================] - 19s 282ms/step - loss: 1.2117 - accuracy: 0.5756 - val_loss: 0.8570 - val_accuracy: 0.7129
Epoch 3/20
64/64 [==============================] - 19s 283ms/step - loss: 0.9218 - accuracy: 0.7011 - val_loss: 0.7079 - val_accuracy: 0.7624
Epoch 4/20
64/64 [==============================] - 19s 282ms/step - loss: 0.7708 - accuracy: 0.7446 - val_loss: 0.6139 - val_accuracy: 0.7980
Epoch 5/20
64/64 [==============================] - 18s 279ms/step - loss: 0.6681 - accuracy: 0.7850 - val_loss: 0.5585 - val_accuracy: 0.8139
Epoch 6/20
64/64 [==============================] - 18s 279ms/step - loss: 0.5944 - accuracy: 0.8136 - val_loss: 0.5128 - val_accuracy: 0.8297
Epoch 7/20
64/64 [==============================] - 19s 280ms/step - loss: 0.5166 - accuracy: 0.8395 - val_loss: 0.4782 - val_accuracy: 0.8455
Epoch 8/20
64/64 [==============================] - 19s 281ms/step - loss: 0.4852 - accuracy: 0.8493 - val_loss: 0.4517 - val_accuracy: 0.8554
Epoch 9/20
64/64 [==============================] - 18s 280ms/step - loss: 0.4398 - accuracy: 0.8586 - val_loss: 0.4240 - val_accuracy: 0.8574
Epoch 10/20
64/64 [==============================] - 19s 280ms/step - loss: 0.3913 - accuracy: 0.8838 - val_loss: 0.4076 - val_accuracy: 0.8614
Epoch 11/20
64/64 [==============================] - 19s 280ms/step - loss: 0.3534 - accuracy: 0.8795 - val_loss: 0.3903 - val_accuracy: 0.8673
Epoch 12/20
64/64 [==============================] - 19s 280ms/step - loss: 0.3492 - accuracy: 0.8963 - val_loss: 0.3745 - val_accuracy: 0.8653
```

Screenshot 5.5: Training the model

5.6 DISPLAYING ACCURACY

Display the accuracy of the model after training the model.



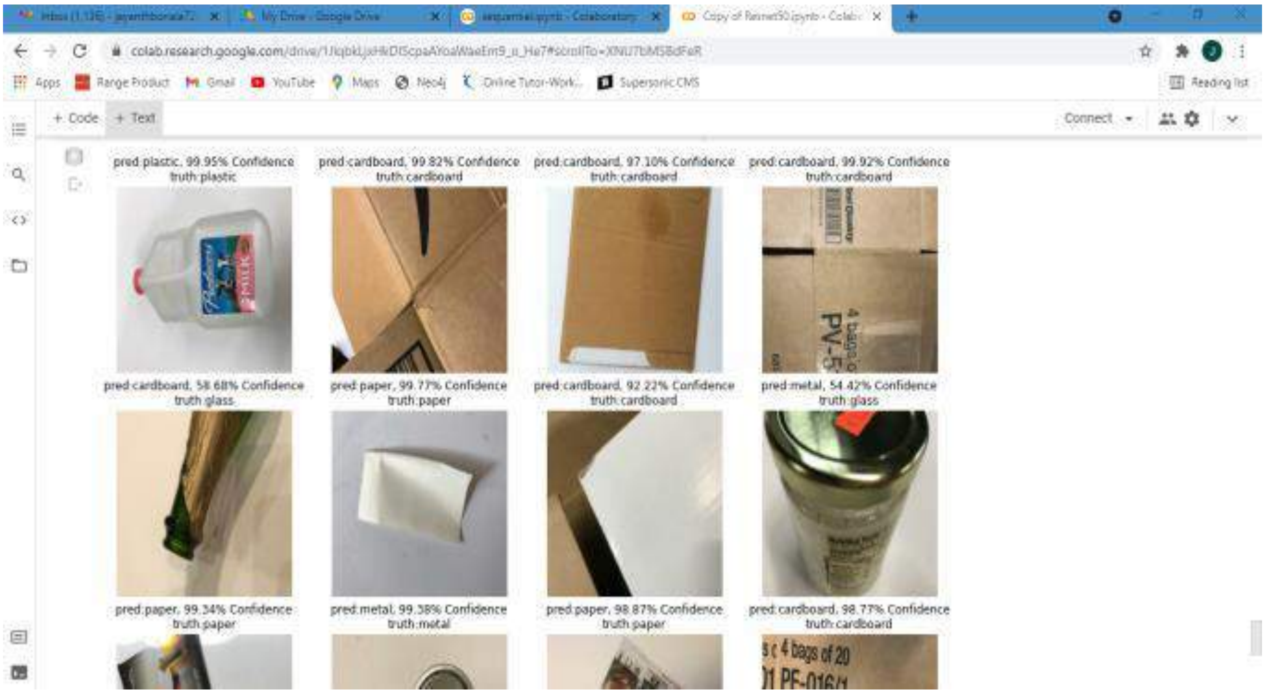
```
# verify the performance of the model
loss, accuracy = model.evaluate(train_ds)
print('Test accuracy :', accuracy)
print('Test loss:', loss)
```

```
64/64 [==============================] - 12s 180ms/step - loss: 0.1164 - accuracy: 0.9708
Test accuracy : 0.9708209633382721
Test loss: 0.1164286658167839
```

Screenshot 5.6: Displaying Accuracy

5.7 DISPLAYING TRUTH, PREDICTION AND CONFIDENCE

Display the images from the set with the labels of truth, prediction and confidence.

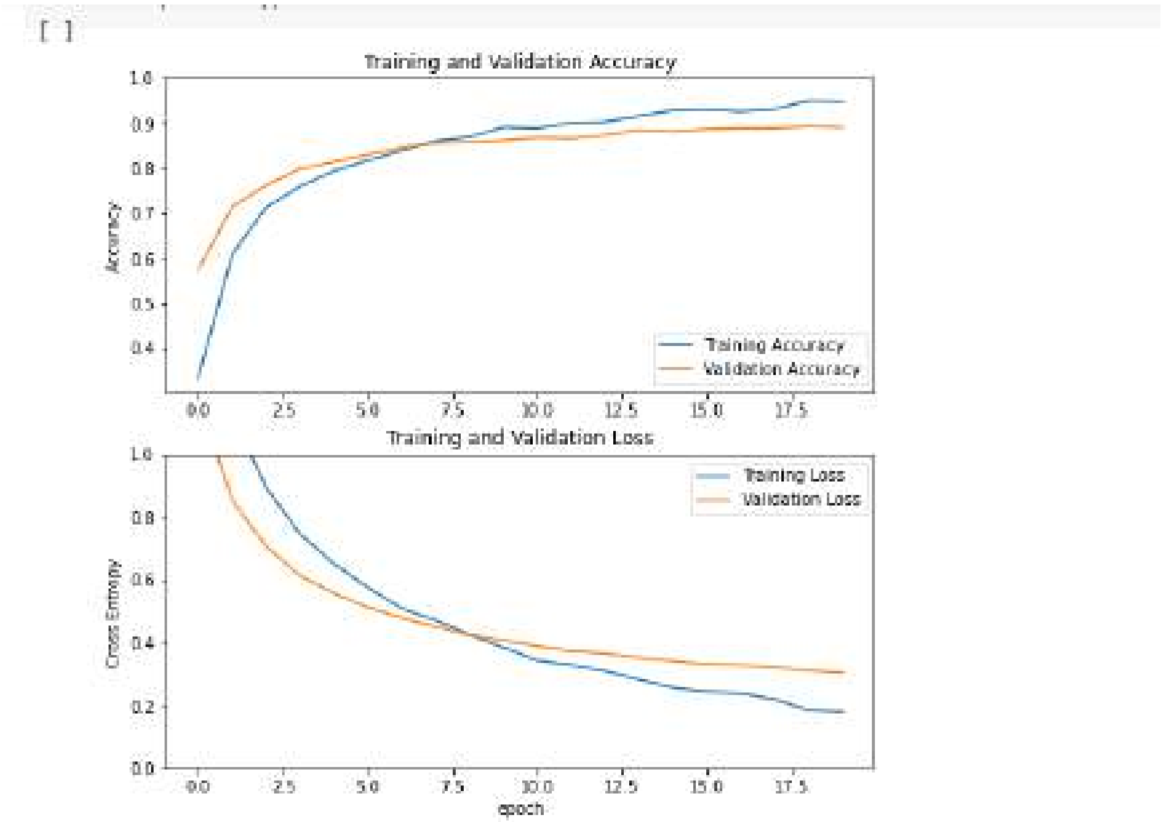


```
pred:plastic, 99.95% Confidence
truth:plastic

pred:cardboard, 99.82% Confidence
truth:cardboard

pred:cardboard, 97.10% Confidence
truth:cardboard

pred:cardboard, 99.92% Confidence
truth:cardboard

pred:cardboard, 58.68% Confidence
truth:glass

pred:paper, 99.77% Confidence
truth:paper

pred:cardboard, 92.22% Confidence
truth:cardboard

pred:metal, 54.42% Confidence
truth:glass

pred:paper, 99.34% Confidence
truth:paper

pred:metal, 99.38% Confidence
truth:metal

pred:paper, 98.87% Confidence
truth:paper

pred:cardboard, 98.77% Confidence
truth:cardboard
```

Screenshot 5.7: Displaying Truth, Prediction and Confidence

5.8 LOSS AND ACCURACY GRAPH

Display the plotted loss and accuracy graphs based on the fluctuations it got during the training.



Screenshot 5.8: Loss and Accuracy Graph

6.TESTING

6.TESTING

6.1 INTRODUCTION TO TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

6.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct & consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

6.2.3 FUNCTIONAL TESTING

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input	: identified classes of valid input must be accepted.
Invalid Input	: identified classes of invalid input must be rejected.
Functions	: identified functions must be exercised.
Output	: identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes.

6.3 TEST CASES

6.3.1 UPLOADING IMAGES

Test case ID	Test case name	Purpose	Test Case	Output
1	User uploads image	Use it for classification	The user uploads garbage image.	Uploaded successfully
2	User uploads 2nd image	Use it for classification	The user uploads the a garbage image.	Uploaded successfully

6.3.2 CLASSIFICATION

Test case ID	Test case name	Purpose	Input	Output
1	Classification test 1	To check if the classifier performs its task	An image with plastic is given.	Plastic is predicted
2	Classification test 2	To check if the classifier performs its task	An image with glass is given.	Glass is predicted.
3	Classification test 3	To check if the classifier performs its task	An image with trash is given	Trash is predicted.

7.CONCLUSION

7. CONCLUSION & FUTURE SCOPE

7.1 PROJECT CONCLUSION

Aiming at the problem of garbage classification, this paper proposes an improved algorithm based on ResNet-50.Firstly, Experiments on common data sets will be implemented to determine the basic model and ResNet-50 shows the best performance and is chosen.The model will be tested on the garbage dataset with 6 types of garbage.Finally, an automatic garbage classification system will be integrated with the proposed algorithm.

7.2 FUTURE SCOPE

E-waste Collection System is future to reduce the electronic waste and reusability of the electronic material in future. □ The E-waste Collection System that develop the information and awareness about the recycling and problem affect by them collection. Decrease the Electronic & Electric Waste from the World and Less Development of Electronic and more recycling of the material.

8. GITHUB LINK

https://github.com/jayanth772/Anautomaticgarbageclassificationsustem.git

9.BIBILOGRAPHY

9. BIBILOGRAPHY

9.1 REFERENCES

- [1] J.-J. Wang, N.-N. Zhao, and J.-H. Li, ``Current situation of marine micro plastics pollution and prevention proposals in China," China Environ. Sci., vol. 39, no. 7, pp. 3056_3063, Jul. 2019, doi: 10. 19674/j.cnki.issn1000-6923.2019.0360.
- [2] W.-B. Li, G. Ma, E.-Q. Yang, Y.-M. Cai, Z. Chen, R.-F. Gao, J.-H. Yan, X.-F. Cao, and E.-J. Pan, ``Study on characteristics of electric dust removal _y ash and bag _y ash in circulating _uidized bed waste incineration system," Proc. CSEE, vol. 39, no. 5, pp. 1397_1405, Mar. 2019, doi: 10. 13334/j.0258-8013.pcsee.181110.
- [3] D.Porshnov, V. Ozols, and M. Klavins, ``Thermogravimetric analysis asexpress tool for quality assessment of refuse derived fuels used for pyrogasi_cation," Environ. Technol., vol. 41, no. 1, pp. 35_39, Mar. 2020.
- [4] P. Kellow, R. J. P. C. Joel, D. Ousmane, D. A. Kumar, D.-A. C. V. Hugo, and K. A. Sergei, ``A smart waste management solution geared towards citizens," Sensors, vol. 20, no. 8, pp. 1_15, Apr. 2020, doi: 10. 3390/s20082380.
- [5] Y.-G. Cheng, N. Chen, and H. Zhang, ``Coal _y ash as an inducer to• study its application in the production of methane gas from domestic waste," Fresenius Environ. Bull., vol. 29, no. 2, pp. 1082_1089, 2020
- [6] Y. Ren, J.-J. Yang, H.-J. Cao, Q.-Y. Zhang, and Q. Liu, ``All components resourcing system of rural garbage based on post-gather automatic sorting and disposal technology and its application," Trans. Chin. Soc. Agricult. Eng., vol. 35, no. 4, pp. 248_254, Feb. 2019, doi: 10.11975/j.issn.1002- 6819.2019.04.031.
- [7] D.-E. Zhao, R. Wu, B.-G. Zhao, and Y.-Y. Chen, ``Research on garbage classification and recognition on hyperspectral imaging technology," Spectrosc. Spectral Anal., vol. 39, no. 3, pp. 921_926, 2019.

An Automatic Garbage Classification System

Dr. M. Varaprasad Rao[1], KNV Raja Sekhar[2], B. Jayanth[3], K. Santosh[4]
[1]*Professor, CSE, CMR Technical Campus, Hyderabad, India*
[2]*Student, CSE, CMR Technical Campus, Hyderabad, India*
[3]*Student, CSE, CMR Technical Campus, Hyderabad, India*
[4]*Student, CSE, CMR Technical Campus, Hyderabad, India*

Abstract: Trash arrangement has consistently been a significant issue in ecological security, As the economy is developing rapidly and with the improvement of people's living standards, the amount of garbage is also increasing rapidly. To tackle the provincial living trash handling issues, we freely built up a one-of-a-kind innovation of perfect and far-reaching use of trash assets. Based on modern biodegradation and automatic sorting, we built a complete industrial chain with mixed domestic waste as mineral raw material. The natural discharge would be siphoned to an anaerobic maturation framework, while the build-up would be another puncturing through the broken drum to the fundamental gathering framework. With our advanced gadgetry, we secure inorganic materials like sand and glass, mash and sundries, for example, bamboo materials, and unadulterated plastic finally. The items would be extra preparing to mechanical materials as expected of the market. The sand and rock glass and other inorganic materials can deliver empty squares, clearing blocks, and wipe blocks. At last, the prevalence of the proposed grouping calculation is checked with the built trash information.

The trial results show that the grouping precision is pretty much as high as 95%.

Keywords: Deep learning, CNN, resnet-34, resnet-50, ANN, Sequential model, Tensorflow.

1.Introduction

With the fast advancement of economy and the improvement of individualsexpectations for everyday comforts, the measure of trash is increasing quickly. As per the most recent report of Worldwide Lianhe Zaobao, the worldwide trash volume will increment by 70% by 2050, and the errand of trash grouping will be significantly more laborious.

Researchers at home and abroad have done a ton of investigates on trash arrangement, however the majority of the proposed plans are advancements of terminal reusing technique. In 2019, China began to require private trash grouping, in which case the front-end assortment is exceptionally subject to individuals' mindfulness. Along these lines, the effectiveness of the trash order actually should be improved.

It is of extraordinary scholastic worth and pragmatic importance to examine a compelling programmed trash characterization technique. The exploration of trash characterization framework is developed, yet the exactness and speed of trash order actually should be improved. What's more, there are not many investigates on trash arrangement dependent on profound learning. As of now, profound learning innovation is generally utilized in picture order, and has some exceptional accomplishments.

Image classification algorithms are gradually diversified. The mainstream algorithms are AlexNet, VGG, Inception, and ResNet. Scholars implemented experiments using these methods and made attempts to modify them to gain better results. These are helpful for the improvement of network structure in this paper. Thanks to the nonlinear characteristics of activation function, a neural network with improved activation function have shown good results. To learn the distribution characteristics of the nonlinear data better, some improvements of the network are mainly focused on the network's depth.

2. Proposed System

The present way of separating waste/garbage is the hand-picking method, whereby someone is employed to separate the different objects/materials. The person, who separates waste, is prone to diseases due to the harmful substances in the garbage. With this in mind, it motivated us to develop an automated system that can sort waste. and this system can take short time to sort the waste, and it will be more accurate in sorting than the manual way. With the system in place, the beneficial separated waste can still be recycled and converted to energy and fuel for the growth of the economy. The system that is developed for the separation of the accumulated waste is based on the combination of Convolutional Neural Network and Artificial neural networks which helps in image classification. Because the trash image dataset is small, we used a pre-trained ResNet-50 model which is a type of Convolutional Neural Network architecture. When the depth is increased, the recognition accuracy of the convolutional neural network can be increased, but due to the increase in depth, the signal that is supposed to modify the weight is reduced at the earlier layer of the CNN. This will make learning at the earlier layers inconsequentially and this is called vanishing gradient. Adding more and more layers to the network always leads to training errors. Residual Network(ResNet-50) is different from the normal convolutional Neural networks can go around this problem of vanishing gradient by designing the Convolutional neural network.using modules which are called residual models, the ResNet model, and the basic block.

3.Methodology

A. System Architecture

In CNN, a few layers makes-up the network. The layers in CNN carry out certain steps, which permit it to classify input images. The convolutional layer convolves the image that is inputted utilizing a sequence of kernels window sizes of 3 x3, this was utilized because what makes difference of the objects are small and local features. The fundamental features are extricated from the input images. The primitive features are extracated with the help of the initial few layers. As the training moves down the layers more complex and detailed features are extracted, with the help of the loss function probability, that is, Softmax work and furthermore there are other enactment capacities like Relu, tanh and so forth however we use essentially relu and softmax. Our model was created dependent on the ResNet-50 pre-prepared model, this model was pre-prepared on ImageNet pictures with a size of 256 x 256 and characterized into 1000 classes

Our model was created dependent on the ResNet-50 pre-prepared model, this model was pre-prepared on ImageNet pictures with a size of 256 x 256 and characterized into 1000 classes.

The ResNet-50 model consists of 5 stages. In first stage there will be four blocks namely, convolution, batch normalization, Relu, Max pooling and remaining four stages each with a convolution and Identity block. Each convolution block has 3 convolution layers and each identity block also has 3 convolution layers. The ResNet-50 has over 23 million trainable parameters.

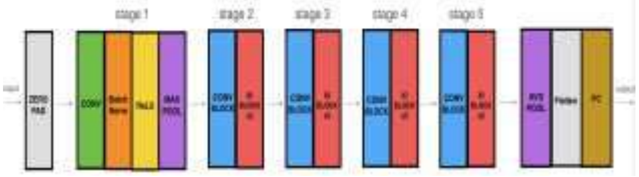


Fig. 1: Resnet-50 model

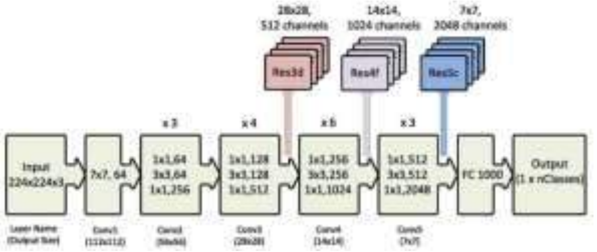
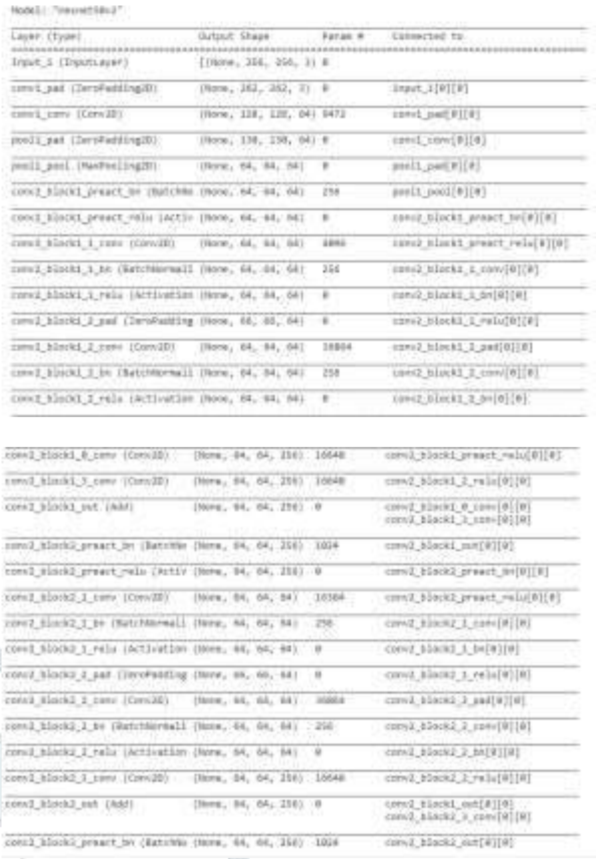


Fig. 2: Resnet-50 Architecture



```
Model: "resnet50v2"
________________________________________________________________________________
Layer (type)                    Output Shape          Param #     Connected to
================================================================================
input_1 (InputLayer)            [(None, 256, 256, 3)] 0
________________________________________________________________________________
conv1_pad (ZeroPadding2D)       (None, 262, 262, 3)   0           input_1[0][0]
________________________________________________________________________________
conv1_conv (Conv2D)             (None, 128, 128, 64)  9472        conv1_pad[0][0]
________________________________________________________________________________
pool1_pad (ZeroPadding2D)       (None, 130, 130, 64)  0           conv1_conv[0][0]
________________________________________________________________________________
pool1_pool (MaxPooling2D)       (None, 64, 64, 64)    0           pool1_pad[0][0]
________________________________________________________________________________
conv2_block1_preact_bn (BatchNo (None, 64, 64, 64)    256         pool1_pool[0][0]
________________________________________________________________________________
conv2_block1_preact_relu (Activ (None, 64, 64, 64)    0           conv2_block1_preact_bn[0][0]
________________________________________________________________________________
conv2_block1_1_conv (Conv2D)    (None, 64, 64, 64)    4096        conv2_block1_preact_relu[0][0]
________________________________________________________________________________
conv2_block1_1_bn (BatchNormali (None, 64, 64, 64)    256         conv2_block1_1_conv[0][0]
________________________________________________________________________________
conv2_block1_1_relu (Activation (None, 64, 64, 64)    0           conv2_block1_1_bn[0][0]
________________________________________________________________________________
conv2_block1_2_pad (ZeroPadding (None, 66, 66, 64)    0           conv2_block1_1_relu[0][0]
________________________________________________________________________________
conv2_block1_2_conv (Conv2D)    (None, 64, 64, 64)    36864       conv2_block1_2_pad[0][0]
________________________________________________________________________________
conv2_block1_2_bn (BatchNormali (None, 64, 64, 64)    256         conv2_block1_2_conv[0][0]
________________________________________________________________________________
conv2_block1_2_relu (Activation (None, 64, 64, 64)    0           conv2_block1_2_bn[0][0]
________________________________________________________________________________

________________________________________________________________________________
conv2_block1_0_conv (Conv2D)    (None, 64, 64, 256)   16640       conv2_block1_preact_relu[0][0]
________________________________________________________________________________
conv2_block1_3_conv (Conv2D)    (None, 64, 64, 256)   16640       conv2_block1_2_relu[0][0]
________________________________________________________________________________
conv2_block1_out (Add)          (None, 64, 64, 256)   0           conv2_block1_0_conv[0][0]
                                                                  conv2_block1_3_conv[0][0]
________________________________________________________________________________
conv2_block2_preact_bn (BatchNo (None, 64, 64, 256)   1024        conv2_block1_out[0][0]
________________________________________________________________________________
conv2_block2_preact_relu (Activ (None, 64, 64, 256)   0           conv2_block2_preact_bn[0][0]
________________________________________________________________________________
conv2_block2_1_conv (Conv2D)    (None, 64, 64, 64)    16384       conv2_block2_preact_relu[0][0]
________________________________________________________________________________
conv2_block2_1_bn (BatchNormali (None, 64, 64, 64)    256         conv2_block2_1_conv[0][0]
________________________________________________________________________________
conv2_block2_1_relu (Activation (None, 64, 64, 64)    0           conv2_block2_1_bn[0][0]
________________________________________________________________________________
conv2_block2_2_pad (ZeroPadding (None, 66, 66, 64)    0           conv2_block2_1_relu[0][0]
________________________________________________________________________________
conv2_block2_2_conv (Conv2D)    (None, 64, 64, 64)    36864       conv2_block2_2_pad[0][0]
________________________________________________________________________________
conv2_block2_2_bn (BatchNormali (None, 64, 64, 64)    256         conv2_block2_2_conv[0][0]
________________________________________________________________________________
conv2_block2_2_relu (Activation (None, 64, 64, 64)    0           conv2_block2_2_bn[0][0]
________________________________________________________________________________
conv2_block2_3_conv (Conv2D)    (None, 64, 64, 256)   16640       conv2_block2_2_relu[0][0]
________________________________________________________________________________
conv2_block2_out (Add)          (None, 64, 64, 256)   0           conv2_block1_out[0][0]
                                                                  conv2_block2_3_conv[0][0]
________________________________________________________________________________
conv2_block3_preact_bn (BatchNo (None, 64, 64, 256)   1024        conv2_block2_out[0][0]
```

Fig. 3: Layers of Resnet-50

B. Dataset

The performance of the proposed garbage classification was assessed on dataset, specifically the Garbage classification dataset.

Each image in each dataset contains just one object. Consequently, the aim of the two datasets is garbage material classification,rather than the detection of waste materials. The Garbage classification dataset contains 2527 images of garbage isolated into six distinct classes, specifically glass, paper, plastic, metal, cardboard, and trash.

Here, we divide the garbage classification dataset into 80% for training and 20% for validation. We divide 2527 images into 2 categories they are training set and validation set , 2022 for training purpose and 505 for validation.

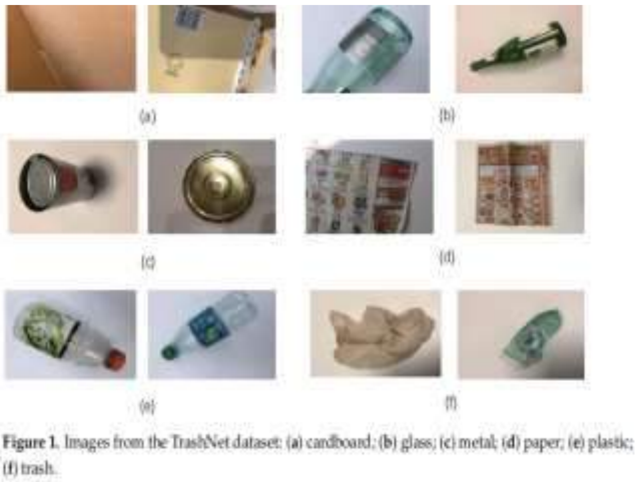


Figure 1. Images from the TrashNet dataset: **(a)** cardboard; **(b)** glass; **(c)** metal; **(d)** paper; **(e)** plastic; **(f)** trash.

Fig. 4: Garbage Classification dataset images

C. Algorithms Used

Numpy : It is a general purpose array processing package. It provides a high-performance multi-dimensional array object, and tools for working with these arrays.

Matplotlib : Matplotlib is an amazing visualization library in python for 2D plots of arrays.

Keras : Keras is an open source Neural network library written in python that runs on top of theano and tensorflow. It is designed to be modular, fast and easy to use.

ResNet :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.

D. Modules

1. **Data Collection** : Collect sufficient data samples and legitimate software samples.

2. **Data Preprocessing**: Data Augmented techniques will be used for better performance.

3. **Train and Test Modeling**: Split the data into train and test data Train will be used for training the model and Test data to check the performance.

E. Design

In this resnet-50 we have 5 Stages:

1. Stage-1
2. Stage-2
3. Stage-3
4. Stage-4
5. Stage-5

In stage 1 we have 4 blocks namely

i. Convolution layer
ii. Batch normalization
iii. RELU(Activation function)
iv. Max Pooling

But, whereas coming to remaining 4 layers we have only two blocks namely:

1. Convolution block
2. Identity block

And in this convolution block we have 3 convolution layers and also in in identity block we have 3 convolution layers.

i. Convolution Layer

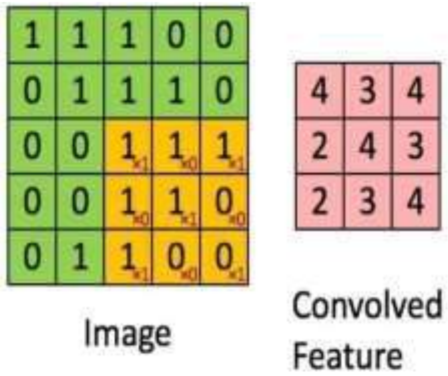


Fig. 5: Convoluting 5*5*1 image with 3*3*1 kernel

Image Dimensions = 5 (Height) x 5 (Breadth) x 1 (Number of channels, eg. RGB)

In the above show, the green segment resembles 5x5x1 input picture, I. The element involved in the convolution activity in the first part of a Convolutional Layer is known as the Kernel/Filter, K, addressed in the colour yellow. We have chosen K as a 3x3x1 grid. The Kernel shifts 9 times on account of Stride Length = 1 (Non-Strided), each time performing multiplication operation between K and the part P of the picture over which the kernel is sliding. The channel moves to right side with a specific Stride Value till it parses the complete width. Proceeding onward, it jumps down to the start (left) of the picture with similar Stride Value and rehashes the cycle until the whole picture is traversed.

ii. Batch Normalization

Batch normalization is a technique for training very deep neural networks that standardizes the inputs to a layer for each mini-batch. This has the effect of stabilizing the learning process and dramatically reducing the number of training epochs required to train deep networks. Importantly, batch normalization works differently during training and during inference. During training (i.e. when using fit() or when calling the layer/model with the argument training=True), the layer normalizes its output using the mean and standard deviation of the current batch of inputs. That is to say, for each channel being normalized, the layer returns gamma * (batch - mean(batch)) / sqrt(var(batch) + epsilon) + beta, where: epsilon is small constant (configurable as part of the constructor arguments)gamma is a learned scaling factor (initialized as 1), which can be disabled by passing scale=False to the constructor. beta is a learned offset factor (initialized as 0), which can be disabled by passing center=False to the constructor.

During inference (i.e. when using evaluate() or predict() or when calling the layer/model with the argument training=False (which is the default), the layer normalizes its output using a moving average of the mean and standard deviation of the batches it has seen during training. That is to say, it returns gamma * (batch - self.moving_mean) / sqrt(self.moving_var + epsilon) + beta.

iii. RELU

Relu is an activation function , it gives 0 when the value is negative and gives the same number or value if the value is positive.

It's formula is : max(0,z)

It avoids and rectifies vanishing gradient problem.

ReLu is less computationally expensive than tanh and sigmoid because it involves simpler mathematical operations.

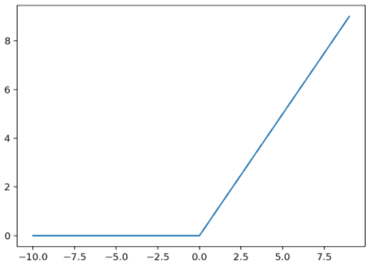
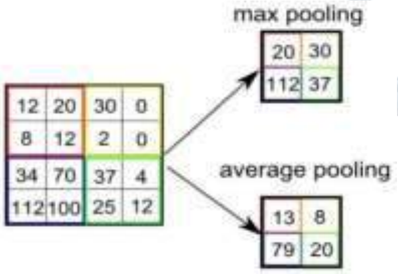


Fig. 6: Plot of Rectified Linear Activation for Negative and Positive Inputs

iv. Max Pooling

There are two types of Pooling: Max Pooling and Average Pooling. Max

Pooling returns the maximum value from the portion of the image covered by the Kernel. On the other hand, Average Pooling returns the average of all the values from the portion of the image covered by the Kernel.



12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

20	30
112	37

13	8
79	20

Fig. 7: Max pooling and Average pooling

Max pooling also performs as a Noise suppressant. It discards the noisy activations altogether and also performs de-noising along with dimensionality reduction. On the other hand, Average pooling simply performs dimensionality reduction as a noise supressing mechanism. Hence, we can say that Max pooling performs a lot better than Average pooling.

F. A Residual Neural Network

ResNet is a short name for a residual network, but what's residual learning?

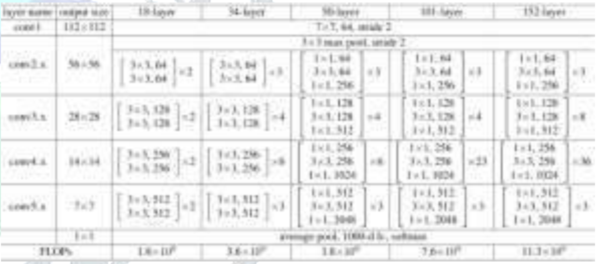
Deep convolutional neural networks have achieved the human level image classification result. Deep networks extract low, middle and high-level features and classifiers in an end-to-end multi-layer fashion, and the number of stacked layers can enrich the "levels" of features. When the deeper network starts to converge, a degradation problem has been exposed: with the network depth increasing, accuracy gets saturated (which might be unsurprising) and then degrades rapidly. Such degradation is not caused by overfitting or by adding more layers to a deep network leads to higher training error. The deterioration of training accuracy shows that not all systems are easy to optimize.

To overcome this problem, Microsoft introduced a deep residual learning framework. Instead of hoping every few stacked layers directly fit a desired underlying mapping, they explicitly let these layers fit a residual mapping. The formulation of F(x)+x can be realized by feedforward neural networks with shortcut connections.The shortcut connections perform identity mapping, and their outputs are added to the outputs of the stacked layers. By using the residual network, there are many problems which can be solved such as:

ResNets are easy to optimize, but the "plain" networks (that simply stack layers) shows higher training error when the depth increases.

ResNets can easily gain accuracy from greatly increased depth, producingresults which are better than previous networks.

G. ResNet-50



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2.x	56×56	[3×3, 64 3×3, 64] ×2	[3×3, 64 3×3, 64] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3
conv3.x	28×28	[3×3, 128 3×3, 128] ×2	[3×3, 128 3×3, 128] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×8
conv4.x	14×14	[3×3, 256 3×3, 256] ×2	[3×3, 256 3×3, 256] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×23	[1×1, 256 3×3, 256 1×1, 1024] ×36
conv5.x	7×7	[3×3, 512 3×3, 512] ×2	[3×3, 512 3×3, 512] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig. 8: ResNet 50 Architecture

So, as we can see the Fig. 6 the ResNet-50 contains the following elements:

A convolution with a kernel size of 7*7 and 64 different kernels all with a stride size 2 gives us 1 layer.

We can also see Max pooling also with stride size of 2

In the next convolution there is a 1 * 1,64 kernel following this a 3 * 3,64 kernel and at last a 1 * 1,256 kernel, These three layers are repeated in total 3 time so giving us 9 layers in this step.

Next we see kernel of 1 * 1,128 after that a kernel of 3 * 3,128 and at last a kernel of 1 * 1,512 this step was repeated 4 time so giving us 12 layers in this step.

After that there is a kernel of 1 * 1,256 and two more kernels with 3 * 3,256 and 1 * 1,1024 and this is repeated 6 times giving us a total of 18 layers.

And then again a 1 * 1,512 kernel with two more of 3 * 3,512 and 1 * 1,2048 and this was repeated 3 times giving us a total of 9 layers.

After that we do average pool and end it with a fully connected layer containing 1000 nodes and at the end a Softmax function so this gives us 1 layer.

We don't actually count the activation functions and the max/ average pooling layers.

So,totalling this it gives us a 1 + 9 + 12 + 18 + 9 + 1 = 50 layers Deep Convolutional network.

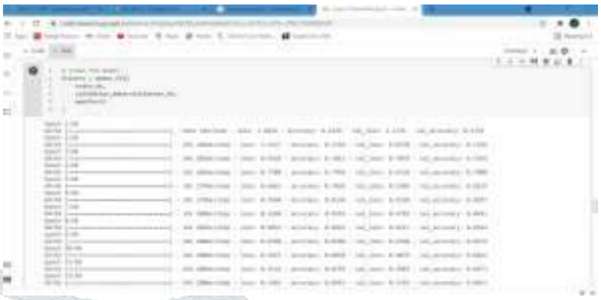
4.Result and discussion

Import the packages which are required to implement the process and set the datapath i.e.,locate the dataset filepath.

Divide the dataset into training and validation sets in the ratio of 80% and 20% ,The total 2527 images are divided into 2022 as training images and 505 as validation images.

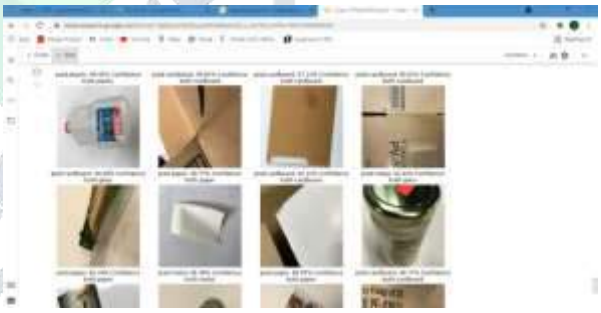
Display randomly six images from the training and validation sets.

Intiate the resnet-50 model.

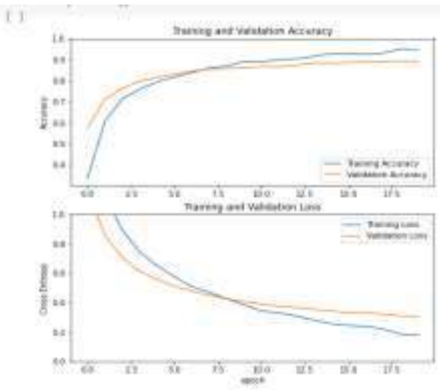


Train the model by giving the epochs size as 20,it means it trains the model twenty times.

Display the accuracy of the model after training the model.



Display the images from the set with the labels of truth,prediction and confidence.



Display the plotted loss and accuracy graphs based on the fluctuations it got during the training.

5. Conclusion

Aiming at the problem of garbage classification, this paper proposes an improved algorithm based on ResNet-50.

Firstly, Experiments on common data sets will be implemented to determine the basic model and ResNet-50 shows the best performance and is chosen.

The model will be tested on the garbage dataset with 6 types of garbage.

After giving 20 epochs we got nearly, 95% accuracy, which is best.

Finally, an automatic garbage classification system will be integrated with the proposed algorithm.

6. References

[1] J.-J. Wang, N.-N. Zhao, and J.-H. Li, ``Current situation of marine microplastics pollution and prevention proposals in China," China Environ. Sci., vol. 39, no. 7, pp. 3056_3063, Jul. 2019, doi: 10. 19674/j.cnki.issn1000-6923.2019.0360.

[2] W.-B. Li, G. Ma, E.-Q. Yang, Y.-M. Cai, Z. Chen, R.-F. Gao, J.-H. Yan, X.-F. Cao, and E.-J. Pan, ``Study on characteristics of electric dust removal _y ash and bag _y ash in circulating _uidized bed waste incineration system," Proc. CSEE, vol. 39, no. 5, pp. 1397_1405, Mar. 2019, doi: 10. 13334/j.0258-8013.pcsee.181110.

[3] D.Porshnov, V. Ozols, and M. Klavins, ``Thermogravimetric analysis asexpress tool for quality assessment of refuse derived fuels used for pyrogasi_cation," Environ. Technol., vol. 41, no. 1, pp. 35_39, Mar. 2020.

[4] P. Kellow, R. J. P. C. Joel, D. Ousmane, D. A. Kumar, D.-A. C. V. Hugo, and K. A. Sergei, ``A smart waste management solution geared towards citizens," Sensors, vol. 20, no. 8, pp. 1_15, Apr. 2020, doi: 10. 3390/s20082380.

[5] Y.-G. Cheng, N. Chen, and H. Zhang, ``Coal _y ash as an inducer to• study its application in the production of methane gas from domestic waste," Fresenius Environ. Bull., vol. 29, no. 2, pp. 1082_1089, 2020.

[6] Y. Ren, J.-J. Yang, H.-J. Cao, Q.-Y. Zhang, and Q. Liu, ``All components resourcing system of rural garbage based on post-gather automatic sorting and disposal technology and its application," Trans. Chin. Soc. Agricult. Eng., vol. 35, no. 4, pp. 248_254, Feb. 2019, doi: 10.11975/j.issn.1002- 6819.2019.04.031.

[7] D.-E. Zhao, R. Wu, B.-G. Zhao, and Y.-Y. Chen, ``Research on garbage• classi_cation and recognition on hyperspectral imaging technology," Spectrosc. Spectral Anal., vol. 39, no. 3, pp. 921_926, 2019.

[8] D.-E. Zhao, R. Wu, B.-G. Zhao, and Y.-Y. Chen, ``Research on garbage classi_cation and recognition on hyperspectral imaging technology," Spectrosc. Spectral Anal., vol. 39, no. 3, pp. 921_926, 2019.

[9] T. Kano,E. Naito, and T. Aoshima, ``Decentralized control for swarm52• robots that can effectively execute spatially distributed tasks," Artif. Life, vol. 26, pp. 243_260, Apr. 2020, doi: 10.1162/artl_a_00317

[10] W. Pereira, S. Parulekar, S. Phaltankar, and V. Kamble, ``Smart bin (waste segregation and optimisation)," in Proc. Amity Int. Conf. Artif. Intell. (AICAI), Dubai, United Arab Emirates, Feb. 2019, pp. 274_279.

[11] K. He, X. Zhang, S. Ren, and J. Sun, ``Deep residual learning for image• recognition," in Proc. CVPR, Dec. 2015, pp. 770_778. [Online].Available: http://arxiv.org/abs/1512.03385

[12]https://www.investopedia.com/terms/d/deeplearning.asp#:~:text=Deep%20learning%20is%20an%20AI,is%20both%20unstructured%20and%20unlabeled.

[13]https://www.analyticssteps.com/blogs/deep-learning-overview-practicalexamples-popular-algorithms

[14]https://neurohive.io/en/popular-networks/resnet/

[15]https://towardsdatascience.com/a-comprehensive-guide-to-convolutionalneural-networks-the-eli5-way-3bd2b1164a53 [16]https://www.geeksforgeeks.org/python-introduction-matplotlib/

[17]https://www.geeksforgeeks.org/numpy-in-python-set-1-introduction

JETIR

Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

KNV Raja Sekhar

In recognition of the publication of the paper entitled

An Automatic Garbage Classification System

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 8 Issue 6 , June-2021 | Date of Publication: 2021-06-10

EDITOR

EDITOR IN CHIEF

ISSN 2349-5162

JETIR2106148 **Research Paper Weblink http://www.jetir.org/view?paper=JETIR2106148** **Registration ID : 310325**

Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

B. Jayanth

In recognition of the publication of the paper entitled

An Automatic Garbage Classification System

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 8 Issue 6 , June-2021 | Date of Publication: 2021-06-10

Parin P

EDITOR

EDITOR IN CHIEF

ISSN 2349-5162

JETIR2106148 **Research Paper Weblink http://www.jetir.org/view?paper=JETIR2106148** **Registration ID : 310325**

JETIR

Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

K. Santhosh

In recognition of the publication of the paper entitled

An Automatic Garbage Classification System

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 8 Issue 6 , June-2021 | Date of Publication: 2021-06-10

Parin P

EDITOR

EDITOR IN CHIEF

ISSN
2349-5162

JETIR2106148 **Research Paper Weblink http://www.jetir.org/view?paper=JETIR2106148** **Registration ID : 310325**

JETIR

Journal of Emerging Technologies and Innovative Research

An International Open Access Journal Peer-reviewed, Refereed Journal

www.jetir.org | editor@jetir.org An International Scholarly Indexed Journal

Certificate of Publication

The Board of

Journal of Emerging Technologies and Innovative Research (ISSN : 2349-5162)

Is hereby awarding this certificate to

Dr. M. Varaprasad Rao

In recognition of the publication of the paper entitled

An Automatic Garbage Classification System

Published In JETIR (www.jetir.org) ISSN UGC Approved (Journal No: 63975) & 7.95 Impact Factor

Published in Volume 8 Issue 6 , June-2021 | Date of Publication: 2021-06-10

EDITOR

EDITOR IN CHIEF

ISSN 2349-5162

JETIR2106148 **Research Paper Weblink http://www.jetir.org/view?paper=JETIR2106148** **Registration ID : 310325**