Waste Classification system using Transfer Learning and Image Segmentation

MSc Research Project
Data Analytics

Kalpesh Dhande
Student ID: 20185821

School of Computing
National College of Ireland

Supervisor: Mr.Jorge Basilio

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Kalpesh Dhande
Student ID:	20185821
Programme:	Data Analytics
Year:	2022
Module:	MSc Research Project
Supervisor:	Mr.Jorge Basilio
Submission Due Date:	15/08/2022
Project Title:	Waste Classification system using Transfer Learning and Image Segmentation
Word Count:	8652
Page Count:	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Kalpesh Dhande
Date:	16th September 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Contents

Waste Classification system using Transfer Learning and Image Segmentation

Kalpesh Dhande
20185821

Abstract

Waste is a big issue in many regions of the world. Many countries are dealing with the issue of waste management. Every day, tons of rubbish is created, and the majority of this waste ends up in landfills. In most places, the process of classification is carried out by people, exposing them to health problems due to an unhygienic working environment, especially in developing and under-developed countries. On top relying on humans makes this entire process prone to human errors as humans experience sentiments, and have their biases and assumptions. Utilizing image classification and deep learning algorithms to separate recyclable garbage from organic waste is one strategy to address this problem. Both the health hazards are decreased and the procedure is made more effective as a result. So, in this research study, I created a model that identified garbage as recyclable or organic by employing transfer learning models such as VGG16 and DenseNet-121. I utilized image segmentation to segment the image, which was then submitted to the model to categorize. I obtained an accuracy of 87 % percent for the VGG 16 model and a 90 % accuracy for the DenseNet-121 model. Multiclass segmentation may be used to improve the image segmentation model even more in future.

1 Introduction

1.1 Overview

If we look at the world population trend, we can notice a sudden rise since the beginning of the 19$^{\text{th}}$ century. And at this point of time, we are at an all-time high. Combining this with our constant effort to make our life more and more convenient we have ended up with much one-time use and throw products. Making things at a mass level and making it cheaper to buy a new one instead of changing or making it reusable has contributed to generating more waste. Add things like non-bio-degradable polymers to this and we have a perfect combination for creating a catastrophe. We can find examples of this all around us like replacing the glass milk bottle with plastic jugs and containers, using plastic bags instead of fabric, and using Styrofoam plates and plastic cutlery to save money and avoid the expense and time of washing metal cutlery. In such a case an effective system to process this generated waste is imperative.

Since we have established the importance of having a good waste management system in place, the very first step that is very important in waste management after collecting the waste is to segregate based on whether it is organic or inorganic to decide what should end in landfills and what should go the recycling plants. Segregating the water at the source would be a very ideal situation but the thing is we can't expect all the people to make the right decision all the time. Many times, things are very ambiguous. Hence waste is segregated again once it is collected and in most systems, this process is carried out by humans. This not only makes the process prone to human errors but also it is very dangerous for humans. We cannot expect what one might encounter while segregating waste. Especially, in developing countries like India where the humungous amount of waste is generated through a plethora of sources combined with an unconscious ignorance of hygiene can be dangerous.

So as mentioned earlier to reduce the human interaction with waste, make the process more efficient and add segregating the waste as a source feature to the process we can automate this process by using deep learning and image classification. A very ideal implementation of this would be making smart bins and installing them in public spaces like shopping malls, cinema theatres etc. Where a large amount of waste is generated and where individuals are expected to do the classification at the source. These smart bins will be able to classify the waste for us instead of relying on individual decision-making. This would make the entire process of waste management far more effective by reducing the time to process and the finance involved in it.

Many image classification and deep learning methods could be applied to solve this problem. Deep learning models have a showcase for learning from data and retrieving features that can generate consistent findings. Several deep learning models, including AlexNet, DenseNet, VGG, MobileNet, and Xception, have been used in earlier work on automatic trash categorization. In this article, I suggest a model for automatic waste classification systems that makes use of pre-trained models like DenseNet-121 and VGG-16 to classify waste and the UNET model to segment the object which can be further passed to the classification model for classification.

1.2 Motivation

We all have had those situations in malls, restaurants, parks, or other places where we are unable to decide whether the thing in our hands is organic waste or inorganic or whether it is recyclable or non-recyclable. Now one can casually throw it in any bin but that's where the problem begins if food-related waste is thrown in a recyclable bin the entire waste collected in that bag is now discarded and will end up in a landfill. Hence, we can't rely on an individual's decision-making regarding this critical classification. Now, this might sound like a very first-world country problem but imagine the situation in countries developing countries or even in very poor countries where the waste management system is already inadequate. On the top, most of the process of sorting is handled by people putting their health at risk. These two reasons are the main inspirations behind this idea of using transfer learning and image segmentation to develop a waste categorization program.

1.3 Research Question

How image segmentation and transfer learning models like DenseNet-121, and VGG-16 could help to solve the problem of waste Classification?

1.4 Research Objective

The study article's major goal is to use deep learning and image segmentation techniques to create an effective garbage sorting system. Furthermore, research tries to achieve the following goals.

- To improve the accuracy of the deep learning models that are used to classify garbage.
- Compare the waste classification performance of the DenseNet-121 and VGG-16 models.

1.5 Structure Of Document

This paper's structure is as follows: Section 2 discusses related studies, including earlier methodologies utilized for this research. Sections 3 and 4 detail the methodology and design specifications that I employed in this research. The methods described is implemented in section 5. In the sixth section, evaluation of applied methods is carried out.

2 Related Work

In this part, I will research and examine the machine learning algorithms that have previously been utilized to categorize garbage. I'm going to pay particular attention to how machine learning approaches are used to tackle the waste problem. In this study, we are particularly interested in two categories: organic and recyclable. We will be using photographs of the rubbish while doing our research. When the data type is the picture, we often employ the Convolution Neural Network. Convolutional neural networks are capable of extracting information from images. This method can learn from a huge number of photos, and the learning may be used to categorize the images.

2.1 Previously used classification model for garbage classification:

Ba Alawi et al. (2021) developed the deep learning model with the help of transfer learning to categorize solid waste into two categories which are recyclable and organic. In this research study, they used 3 different pre-trained models which were DenseNet121, AlexNet, and SqueezNet. They used the waste classification dataset which was available on Kaggle to train this model. This dataset contains approximately 22 thousand images. 55% of the images were organic, and 45% were recyclable. The author's model in this article was built on three basic elements. Images were first acquired, then pre-processed, and Finally, the images were sent to the CNN, which withdraws the feature and categorizes it based on the classes. The author used four values to evaluate the performance of the models: accuracy, recall score, f1 score, and precession. In this study, the author achieved 94% accuracy with a pre-trained DenseNet121 model, which was the best model among all of the pre-trained models that the authors have used. According to this author, the concept of image segmentation will aid in improving the model's accuracy.

Ramsurrun et al. (2021) created the model that will categorize and identify the garbage using deep learning and computer vision. They classified the garbage into five separate categories: paper, plastic, glass, cardboard, and metal. On pre-existing photos, they applied 12 distinct algorithms across three different classifiers: Sigmoid, Support Vector Machine, and SoftMax. With the aid of the generated model, the authors created a hardware solution in which a camera captured the image and passed it to the model, and the model's output was used to sort garbage into separate bins. They're working with a collection of around 2500 photos divided into six groups. The photos were resized programmatically by the researchers in order to maximize the training reaction time. The creators of the model employed data augmentation before training to add colours and positions to the dataset. The author employed 12 distinct transfer learning models for training. Using VGG19 with the SoftMax classifier, they attained the greatest accuracy of 88%. The main issue with this model is the dataset; the dataset used to train the model is quite tiny, and all of the photos in the dataset have a white backdrop, which affects the accuracy. Aral et al. (2018) faced a similar issue in this story. In this research work, the author also advised that picture segmentation be used to increase the model's accuracy.

Both of the above experiments employed the transfer learning technique to train the models, implying that including the concept of picture segmentation in this can improve the model's accuracy. Let's have a look at some other models that have previously been used to tackle the trash sorting problem. The models are then compared with and without transfer learning, and finally, the notion of picture segmentation is presented.

Aral et al. (2018) created a waste categorization system that uses a deep learning approach to classify waste into six different categories. They used the 5 different deep learning architectures to categorize this waste which were Xception, Inception, DenseNet 169, DenseNet121, and InceptionResNetV2. The authors used adadelta and Adam optimiser to compile the model. The TrashNet dataset was used by the authors to solve this classification problem. Metal, glass, paper, trash, cardboard, and plastic are among the categories in the dataset. Because the dataset they used was so small, they used the data augmentation technique to increase the size of the data before training. They used the fine-tuning approach on a few of the models. While developing the models, the authors experimented with different learning rates and batch sizes. Among all the models developed by the authors, the Densenet 169 architecture with fine-tuning yielded the highest accuracy of around 95%. The dataset was the major drawback of this article. The dataset is too small, and all of the images have a white background.

In this research paper, the Wang and Wang (2021) created a deep learning model that improved the model's precision and performance. They used the Transfer learning and CNN models during the model's development. The dataset they used contains 1400 images divided into four categories: dry waste, recyclable waste, hazardous waste, and wet waste. To avoid image distortion, they used x width and h height for image scaling. The authors used the Adam optimiser with a learning rate of 0.002 and a batch size of 64 while training the model. The authors calculated the accuracy for each class in the dataset. The MobileNet V2 transfer learning model has the highest combined average accuracy of 93.5%.

Frost et al. (2019) created a classification model that can be used to create a simple mobile application for sorting waste after meals. They used the MobileNet architecture and CNN architecture, to develop this model. They used the TrashNet dataset for this. However, while using the TrashNet dataset, they modified it by including more images of landfill waste and food waste, and they created three categories from this: recyclable, compostable, and trash. Before moving on to model training, the authors scaled those images in two different resolutions in order to develop two different models. They created two models for both resolutions, one with a pre-trained mobile net architecture and one with a four-layer custom CNN model. The pretrained Mobile Net model achieved 77% accuracy, while the Custom CNN model achieved only 23% accuracy.

Singh (2021) created a deep learning model for categorizing different types of polythene bags. They used the 72-layered Xception model to categorize polythene into three different types. The PolythNet model is another name for this model. The model's architecture consists of five distinct layers, the first of which is the input layer, followed by the Xception layer, 2 Fc dense layers, and the last which is the output layer. The dataset utilized by the publishers was gathered from garbage dumps. This dataset was very small because of that the authors used the data augmentation technique to resize the images, then zoomed and flipped the images to increase the data. The authors achieved 96% accuracy with the validation dataset and 89% accuracy in training after feeding this data to the model for training. The main disadvantage of this article is the small dataset.

Niu et al. (2020) proposed a deep learning-based model for trash classification. The authors used four different types of approaches in this study. They used the traditional deep learning approach first, then the conventional deep learning method, then the transfer learning models, and finally, the authors created their modified models in the fourth approach. The dataset they used for this is the TreshNet dataset. DDC-AlexNet, DDC-ResNet, DeepCoral-AlexNet, and DeepCoral-Resnet were the names of the models developed in four different approaches. According to the author, the Deep Coral- Resnet model achieved the highest accuracy of around 96% in all papers that used the TrashNet dataset.

Using the deep learning technique, Sreelakshmi et al. (2019) created a model for classifying plastic as plastic or not plastic. These authors employed two distinct architectures: Capsule Neural Network (CNN) and Convolution Neural Network (CNN) (CNN). The capsule neural network is a convolution neural network improvement. For training the models, the authors used two types of datasets. One is made by taking photographs with a camera, while the other is made by downloading photographs of plastic from the internet. During training, the authors experimented with different batch sizes, learning rates, and epochs. They achieved the best results with a learning rate of 0.01 and a batch size of 32. The authors achieved 95.7% accuracy with the capsule neural network and 93.6% with the convolution neural network in the first dataset, 96.3% accuracy with the capsule neural network and 95.8% with the convolution neural network in the second dataset. We can clearly see that the capsule neural network performed better on both datasets.

Yang, Zeng, Wang, Zou and Xie (2021) created a deep learning-based classification model that categorizes waste into 43 different categories. This model is known as the GarbageNet model. They developed a model based on the concept of incremental learning, in which the model is trained on a new dataset while not forgetting previous learning. In The development of the model, the authors used the Huawei cloud garbage dataset, which contains 20000 images. Some elements comprise the garbage net model. The classifier and feature extractor, which are used to extract features from images, were the first elements in this model. After that, the extracted feature information is saved in a memory pool. The new internet categories are also saved in the memory pool. Using cosine distance, the test sample is compared with categorical memory, yielding the category that is closest comparable to the forecasted result. When compared to the other pre-trained model, this generated model showed an improvement in accuracy.

Using Deep Learning, Abdulrahman and Hewahi (2021b) created a model to classify waste as biodegradable or non-biodegradable. They used three different convolution neural network-based architectural approaches. The dataset they used has approximately 1.5 lakh images. The issue with the dataset is that the bio-degradable category contains approximately 82 percent of the images. To address the issue of class imbalance, the authors reduced the images from the biodegradable class. They have a computational power problem. Because the image sizes are uneven, they must resize the images. The authors of this article developed a model using three approaches: The first is the AlexNet mode and the second and third model is consist of fc Layers and the conv Layers. The third model achieved the highest accuracy in architectural design as the first and last layers are fc layers and all middle layers are conv layers.

Qiuhao (2021) created a models that classifies kitchen waste using deep residual network and transfer learning models. This waste information is divided into 20 categories. The authors used Alexnet, Resnet, and Vgg 16 transfer learning models to compare the results of the deep residual network. This kitchen dataset was divided into 20 categories, with each category containing approximately 300 images. After that, the images were resized to a single size. For this experiment, the learning rate is set to 0.001, and three batch sizes are tested. Various architectures such as Vgg 16, AlexNet, ResNet, and Deep residual network are trained in this environment. In this case, the Deep residual network achieves the highest accuracy of around 87% across all models.

Using the CNN architecture, Gan and Zhang (2020) created a model to sort garbage into four categories: cartoons, cans, batteries, and plastic bottles. They compared the BP neural network algorithm to the Migrated CNN based on AlexNet in this paper. Before training the model on the dataset, the authors need to change the image resolution. There are two methods for training the BP neural network. They have implemented it on the basis of texture and shape in the first approach, and they are able to achieve an accuracy of around 55%, while another approach is based on the HSC colour space algorithm, and they are able to achieve an accuracy of around 76%. When the model has been trained They achieved 100% accuracy with the Migrated CNN

Tammina (2019) used the VGG16 pre-trained model and deep convolutional network to create a model to classify dogs and cats from image data. This model was created using approximately 7000 images. They used the VGG16 pre-trained model as the feature extractor, which transfers low-level characteristics and adapts extra features at a higher level. They used an image augmentation technique to increase the data for training. They created the models using three different approaches: first, they created the basic CNN model, then they created CNN with data augmentation and fine-tuning, and finally, they created the VGG 16 pretrained model with fine-tuning. VGG 16 Pretrained model outperforms all other models, with an accuracy of around 95%.

The deep learning model was created by Adedeji and Wang (2019) using the 50 Layers ResNet architecture and support vector machine. The authors used the ResNet architecture as the feature extractor in this model, and the extracted feature is then passed to the support vector machine for classification. The authors used the TrashNet Dataset to solve this classification problem. The authors used data augmentation to increase the size of the dataset because it was too small. The model's architecture began with the ResNet pre-trained model, from which the top layer was deleted but the features were retained, and these features were then passed to the support vector machine for classification. They were able to achieve an accuracy of around 87% using this method.

Shaikh et al. (2020) created a deep learning model using the Inception-V3 architecture to classify waste into wet and dry waste. They created a web application that uses this developed model to determine whether the object in the image is wet waste or dry waste. For model development, they used the Inception V3 pre-trained model. They put their model to the test by uploading some of the images to their website, and they were able to classify the images most of the time. They were able to achieve an accuracy of approximately 84%.

All of the papers mentioned above take different approaches to develop deep learning models. However, after reviewing all of the papers, I can conclude that the majority of time transfer learning helps to improve model accuracy. Let us look at some research that compares models with transfer learning to models without transfer learning.

2.2 Model building with transfer learning vs Without transfer learning

Zhang et al. (2021) created a CNN-based Deep Learning model to classify waste into five categories. They have used the NWNU-Trash dataset, which is superior to the TrashNet Dataset. This dataset has higher-quality images, more evenly distributed categories, and more diverse background. They used the DenseNet-169 architecture both with and without transfer learning in this article. When the two models were compared, the model with transfer learning outperformed the model without transfer learning by 35%. This researcher also evaluated numerous different approaches, such as VGG 16, AlexNet, and Google Net V2, and found that the VGG 16 surpassed the other models by approximately 75%. When compared to DenseNet with transfer learning with VGG16, DenseNet with transfer learning gave better accuracy.

Yang, Zhang, Lv and Wang (2021) have developed a model which will detect wind turbine blade damage with the help of a deep learning model. In this article, the data pre-processing authors have implemented the otsu image segmentation technique which helps in removing the background of the image. To boost the performance of the model they have used transfer learning and ensemble learning. The proposed model in this study consists of 5 convolutional layers, 3 fully-connected layers, and 1 classification layer. The first 5 layers are based on the pre-trained AlexNet architecture. Three layers in between were randomly initialized and the random forest classifier is the last layer. The dataset used by the authors is from a wind turbine in western China. They compared this proposed model to four other models and were able to achieve a higher accuracy of 98% with this proposed model. They also compared AlexNet with and without transfer learning and found that AlexNet with transfer learning performed better.

Wu et al. (2018) have developed a deep learning model to classify the flowers. They have developed the models with the help of pre-trained models such as Inception-V3, VGG-19, ResNet50 and VGG16. The models which they developed were then they have compared with the previous works that developed models with the same architecture but without transfer learning. In this comparison, they found out that the models with transfer learning performed better than models without transfer learning. In this, they have used two different datasets for training the models. The dataset which containing the most photos and species providing the best accuracy. The ResNet50 model performed better than every other applied model when they were all compared. Models with transfer learning perform better than models without it in terms of accuracy, with gains of 15 to 20 percent in each case.

It is clear from some of the earlier articles that the model created using transfer learning offered increased accuracy. See a few more studies that illustrate image segmentation, which I'll be using to build the model in this study.

2.3 Image Segmentation

Using deep learning models, Alkassar et al. (2019) created a model to conduct image segmentation on magnetic resonance imaging (MRI) scans of brain tumors. They have employed the Fully connected network (FCN) and VGG16 with transfer learning to complete this challenge. The architecture of this model is divided into three sections, the first of which is the VGG-16 transfer learning-based encoder. The second section is the decoder, and the third section is the classification layer. They must adjust the image sizes to maintain the same resolution on which VGG 16 is trained. Authors can achieve an accuracy of about 98% with this design.

The classification of minerals using a deep learning network based on picture segmentation was proposed by Abdulrahman and Hewahi (2021a). The four crucial components of this deep learning architecture are the module design, segmentation model, loss function construction, and backbone selection. The Ore image segmentation system is divided into three parts: data set preparation (which includes data augmentation), mask preparation (which includes erosion processing and manual annotation), and model building (which includes designing the loss function and setting the backbone selection module). They created four alternative deep learning-based picture segmentation models, with the VGG16, Segnet, and Unet serving as the backbone. In this instance, fine tweaking is used. The author's findings indicate that image segmentation can be successfully used in models to increase classification accuracy.

3 Methodology

I have used the Cross-Industry Standard Process of Data Mining (CRISP-DM) technique to finish this study. This methodology aids in our comprehension of the steps that must be taken when dealing with any data-related research issues. The CRISP-DM approach consists of six phases in total. The first is an understanding of the business, followed by an understanding of the data, data preparation, model construction, evaluation, and deployment. Let's go through each stage in turn and how I used it in our research.

1. Business Understanding: The CRISP-DM approach starts with this phase. I have first comprehended the problem statement. In this study, the classification of garbage using deep learning algorithms is the key problem.

2. Data Understanding: This is the point at which I began to solve our problem. First, I need to find the dataset and examine the dataset that I utilized to answer the problem. Obtaining useful insights from it. In this phase, I realized that I needed to use the dataset containing the photographs.

3. Data Preparation: At this point, I've pre-processed the data so that it can be immediately put into the model for training. I prepared the data in a different way for each model in this paper. For VGG16 mode,l I initially performed data augmentation on the original dataset and then divided the training dataset in an 80:20 ratio to create the training and validation dataset. For DenseNet121 I have

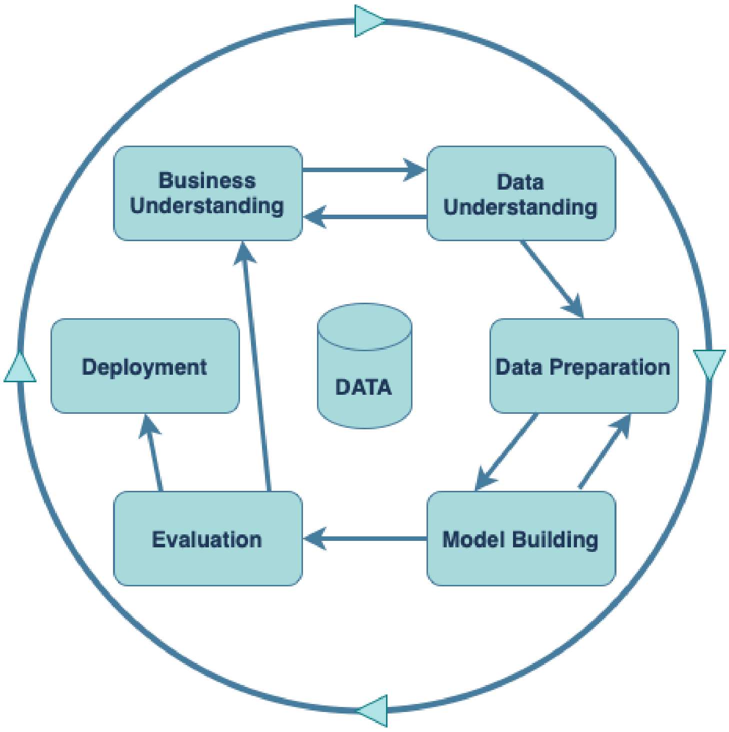


Figure 1: CRISP-DM Methodology

divided the original training dataset to training and validation in 80:20 ration. In order to do picture segmentation, I annotated the dataset and then divided tat dataset in Train, test and validation in 60:20:20 ratio.

4. Model Building - Based on the pre-trained models, I have built the architecture of our VGG16 and DenseNet121 models n this step. After creating these models, I have used pre-processed datasets to train them so they can give the desired results. I have developed an image segmentation model which is based on UNET architecture.

5. Evaluation: In this stage, I have checked how well our models worked. I have first examined the validation and training accuracy and loss while first examining the models. I have looked at the graphs of training versus validation accuracy and loss. The precision, recall, and f 1 score were determined using the confusion matrix. After this, the image segmentation model was examined on the basis of the Accuracy and loss of training, validation, and test data.

6. Deployment: If the models are constructed appropriately, then it can be deployed at the industry level in this final step.

In this study, I have used trash categorization data that I acquired from Kaggle. The data were then pre-processed using data augmentation. Then I divided the data into training, testing, and validation datasets. Following that, I used the DenseNet121 and VGG16 models, which were pre-trained on the Image Net dataset. Then I trained the model, and based on the validation results, I adjusted the weights and trained the model again. Finally, I computed the accuracy and generated the categorization result based on it. For the segmentation step, I annotated certain photos taken from the original dataset to create binary masks. After producing the data masks, I divided the dataset into the train, test, and validation groups, and then applied the image segmentation model to that data. Following that, I assessed the image segmentation model and tested it on the test data. Then this can be passed to the classification model in case of multi-class segmentation to classify each object.Figure 2 depicts the entire flow of this.

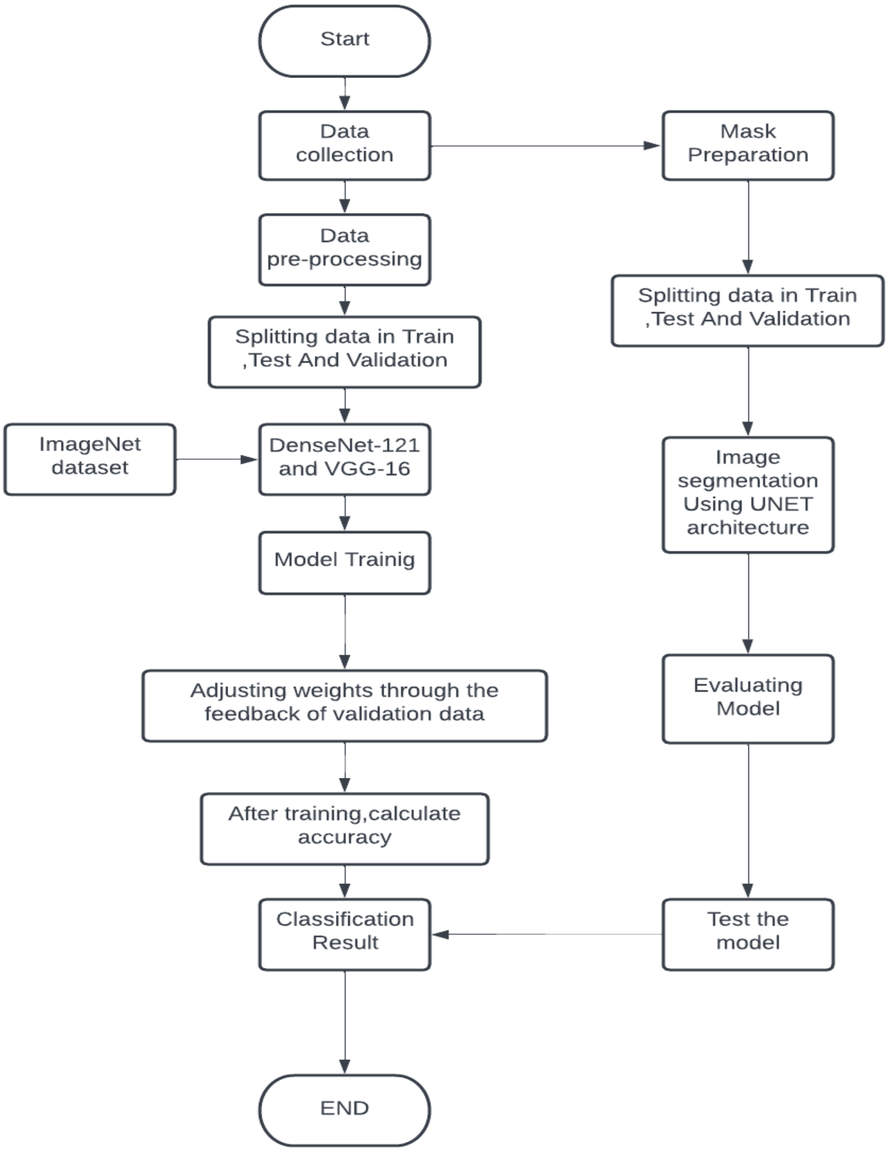


Figure 2: Flow of the proposed reserch

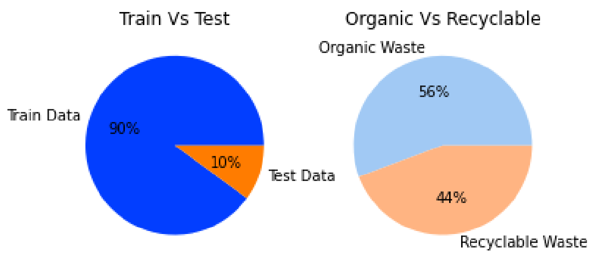
3.1 Data Collection and Data Understanding:

In this study, I have used a dataset called "Waste Classification Data" that is available on Kaggle. This dataset contains the image type data. This dataset is divided into two parts: the training dataset and the test dataset. The training dataset contains 22564 images, whereas the test dataset has 2513 images, as shown in figure 3a. There are two classes in each of these folders. One is organic, while the other is recyclable. The organic class training dataset has 12565 images, while the test dataset has 1401 images. The

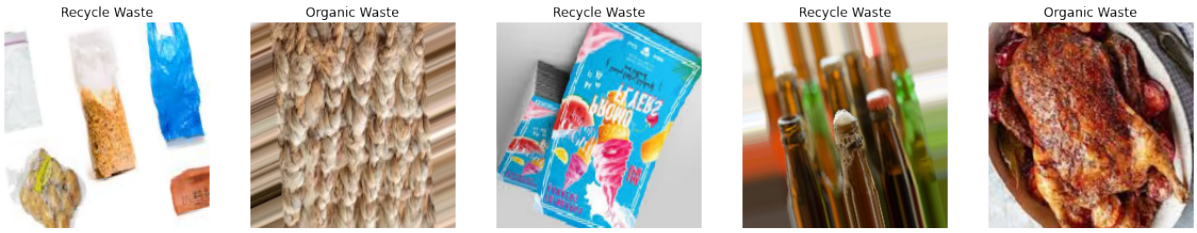
training data for the Recyclable class contains 9999 images, while the test data contains 1112 images, as shown in figure 3a. The training dataset comprises 90% of the images, whereas the test dataset contains 10% of the images. The organic dataset consists images of vegetables, beans, fruits, meat, and other organic items, whereas the recyclable class includes images of plastic, cardboard, papers, straws, and other recyclable items. Samples of these images are given in the figure 3c
Dateset Link: `https://www.kaggle.com/datasets/techsash/waste-classification-data`

```
Total Number of Training Images 22564
Total Number of Test Images 2513
Total Number of Organic Train Images:  12565
Total Number of Organic Test Images:  1401
Total Number of Recycle Train Images:  9999
Total Number of Recycle Test Images:  1112
```

(a) Basic Stats about Dataset



(b) Pie chart of Training vs Test and Organic vs Recyclable for Training Data



(c) Samples From Dataset

Figure 3: Basic Information Of Dataset

3.2 Data Pre-Processing

In this study, I used the 'Waste Classification Data' that I discovered on Kaggle. This dataset is divided into two training categories: organic and recyclable.I have a moderately big dataset in this. However, in order to expand the dataset, I did data augmentation which can help to improve the accuracy of the model. The photos were then magnified, scaled, and then flipped vertically and horizontally. Section 3.2.1 goes into detail on data augmentation. Following that, I divided the training dataset in an 8:2 ratio into training and validation datasets. I separated the dataset for testing purposes. I did not use the data augmentation technique for DenseNet121. I just split the training dataset in an 8:2 ratio. From training dataset I have taken 10 images of one class i.e banana and created the masks of them on Apeer.com by annotating them for the image segmentation.

3.2.1 Data Augmentation

A strategy for boosting the amount of a dataset is data augmentation. This method involves taking the original dataset and transforming it with various operations, such as rotating, rescaling, flipping, and so on. This enables more data to be collected for training and improved model accuracy. However, this can sometimes produce problems. When augmenting the training dataset, it generates some images that are similar to those in the validation dataset, resulting in model overfitting for particular deep learning architectures. When I tried to utilize the DenseNet121, this happened to me. DenseNet121 overfits when I utilize augmented data to train it. As a consequence, I only used the augmented data for the VGG16 architecture in my scenario. I rescaled the photos to 1/255, zoomed them to 0.6, and then flipped them vertically and horizontally. For DenseNet121, I used the original data.

3.3 Model Evaluation

The final phase in the study project was model evaluation. I worked on the classification problem in this study. I have started by looking at the area under the curve (AUC) and loss for training data. I have then verified the same information for the validation data. Both of these will be checked when the model is being trained. Then, after evaluating the model using test data, I checked the AUC and loss again. After that, I plot the confusion matrix. Since there were only two classes in our classification issue, the confusion matrix was a 2d matrix. In my scenario, the classification consists of four sorts of values: true positive (organic trash anticipated as organic), true negative (recyclable waste predicted as recyclable), false positive (recyclable waste predicted as organic), and false negative (organic waste predicted as recyclable). On this basis, I have computed the precision, recall, F1 score, and accuracy. For the segmentation model first I have computed the Accuracy and loss for the training and validation dataset. Then I calculated the same for the test data. After this, I verified the model by creating the predicted masks of the test images.

4 Design Specification

4.1 Model Building

As I discovered in my literature review, deep learning models perform exceptionally well when dealing with classification tasks. When it comes to deep learning image classification, I can see that CNN does a better job of obtaining knowledge from them and classification of images. Let's have a look at the CNN-based designs I employed in my study.

4.1.1 Transfer learning

In the real world, if I consider any type of object, I can create a variety of images on that object. If I take into account my problem statement, I would categorize the waste image as recyclable and organic. When this task is compared with a real-world scenario, multiple images of waste can be captured. However, in this case, I have a large number of images to process, which will necessitate a large amount of computational power. The concept of transfer learning is introduced in this case. Transfer learning is the process of transferring the knowledge that has been already generated after training on the larger datasets. The knowledge acquired through training on a huge dataset is utilized for training our selected dataset in this technique. In my case, the dataset that I am utilizing may be deemed moderate, but when compared to the real world, it is a very little dataset. As a result, employing this concept in my research is advantageous to the study. Building a model from scratch that is sufficiently intelligent to recognize garbage with extreme accuracy is challenging. Additionally, the model developed from scratch requires significant processing power, and the machine I am using lacks that capability. Furthermore, I observed through the literature review that when the model is developed using transfer learning, the accuracy increases. As a result, I have taken the knowledge acquired on a huge dataset, the ImageNet dataset, and then train it on my dataset. Now, have a look at some of the pre-trained models that I have used in my research project.

- **VGG16 pre-trained model:** Simonyan and Zisserman (2014) developed the VGG16 model, which is based on the Convolution Neural Network. The VGG16 model contains a total of 16 weighted layers, which is why it is called VGG16. The VGG16 is a well-known image classification architecture. 13 convolution layers, 5 max-pooling layers, and 3 dense layers make up the architecture of the VGG16, although only 16 of these layers are weighted. VGG16 accepts input in the form of (224,224,3). This VGG-16 model was trained on the enormous ImageNet dataset, which contains over 14000 million images divided into approximately 22000 categories. I used this pre-trained model information in my architecture. When designing the architecture for my classification problem, I initially loaded this pre-trained model with the include top argument set to false and with the input shape as (224,224,3) and with weights as image net. After that, I set the trainable layers to false to freeze the layers. Then I modified the following layers, adding a 30% dropout first, then a flatten layer and batchnormalization layer, then a denser layer with 1024 neurons and he uniform initializer, then batchnormalization, then activation as relu, then 20% dropout, then dense layer with 512 neurons, then batchnormalization, then activation as relu, and finally 10% dropout. Then, in the output layer, I chose activation as the Sigmoid in the final layer since it is better for binary classification tasks, and I maintained 1 neuron for the binary output. So I improved this design by adding three dropout layers, one flatten layer, three batch normalization layers, and three dense layers.. These all layers I have modified on the basis of the output of the validation layer.

- **DenseNet121 pre-trained model:** The DenseNet collection of models includes the DenseNet-121 model, which is made to conduct image classification. The DenseNet models work as the properties that each dense block in DenseNet extracts are used by the next dense blocks, indicating that the last dense block used the features of all earlier blocks. This paradigm has two benefits: the features may be reused, and the vanishing-gradient problem is no longer an issue. DenseNet's high accuracy is due to its detailed architecture and feature reusability. The architecture of the DenseNet121 contains a total of 120 convolution layers and 4 avg pool layers. This DenseNet121 model is also trained on the large ImageNet dataset. I loaded this pre-trained model with the include top parameter set to false , weights as ImageNet and pooling as average while creating the architecture for my classification challenge.Then I have kept the trainable layers as false to freeze the layers. Then I modified the following layers, adding a flatten layer and batchnormalization layer, then a denser layer with 128 neurons and he uniform initializer, then activation as relu, then batchnormalization, then dense layer with 64 neurons, then batchnormalization, then activation as relu. Then, in the output layer, I used the activation layer as the Sigmoid because it is superior for binary classification jobs. So I improved this design by, 1 flatten layer, 2 batch normalization, and 3 dense layers. I removed the dropout layer since I was getting an overfitted model.

Sometimes information is lost due to dropout layers, and when validating the model on validation data, it predicts data based on partial observations, resulting in overfitting.

4.2 Image Segmentation

Image segmentation is a well-known idea in the field of image processing. Medical imaging is one of the fields where image segmentation is widely employed. There are several forms of picture segmentation. I did semantic segmentation in this research article. Semantic segmentation is the classification of pictures of the same class into one class. I utilized the UNET architecture to do this picture segmentation. The UNET architecture is made up of the encoder, bridge, and decoder. In this, the architecture I have designed it 5 filters in which filter 1024 the last encoder block will act as the bridge. For the same filters, I have designed the decoder block. And the last I have kept the output layer in which I have used the activation as the sigmoid. This will be used in one of my organic image class, which is banana.

5 Implementation

I began by uploading my dataset to Google Drive because I was executing my code in the Google Collab notebook. Then I linked my Google Colab notebook and Google Drive. The dataset I utilized is made up of photos of organic and recyclable garbage. Then I read this image data and augmented those images using an image data generator, after which I separated the training dataset into training and validation in an 8:2 ratio. Following that, I loaded VGG16 and DenseNet121 pre-trained architectures. Then I made this architecture sequential and updated the model by adding some layers which i have defined it in detatiled in section 4.1.1. To compile this model, I utilized the Adam optimizer and loss as binary cross-entropy. Then, using the Training and Validation data, I trained both models for 20 epochs. To see if the model was developed appropriately, I first examined the model's loss and area under the curve after training. Then, using the confusion matrix, I determined the test accuracy as well as the estimated precision, recall, and f1 score. Then, for image segmentation, I first built the masks of the images. Then I established my model's architecture, which is based on the UNET architecture. I then imported and augment the data. Following that, I divided the data into train test and validation data. Following that, I compiled the model with binary crossentropy and the adam optimizer, and then trained it on the training dataset and validate on validation dataset while training. Following that, I evaluated the model by running it over the testing dataset and calculating the Accuracy and loss.

All of my code was written in Python. Python version 3.7.13 was used. I utilized the image data generator to load and process the data. TensorFlow's Keras library was used to load models. I used the sklearn package to create the confusion matrix and classification report. I used the matplotlib and seaborn libraries to create the visualization. My system's setup is as follows:

- Apple M1
- Ram 8 GB
- 256 GB SSD

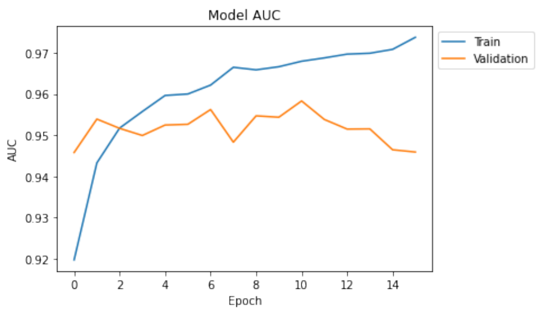
- macOS Monterey version 12.5

6 Evaluation

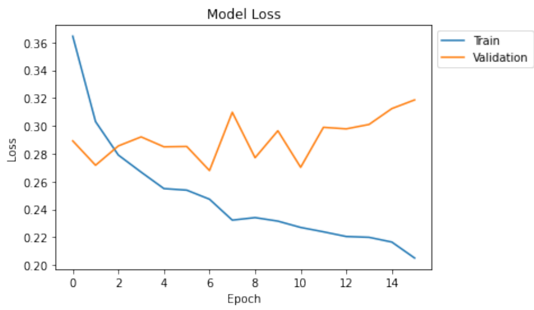
First, I checked to see if the created model was doing well or not using the area under the curve and loss for both training and validation data. Following that, the Test dataset will be used to confirm the same. However, I cannot declare that the developed model performed well based just on these two factors. So, following that, I generated the confusion matrix and used it to determine the accuracy, recall score, f1-score, and precision.For image segmentation i have checked the Accuracy and loss for the training data and validation data first. Then same i have checked for the test data. And finally i have checked the segmented image output.

6.1 VGG16 pre-trained Model

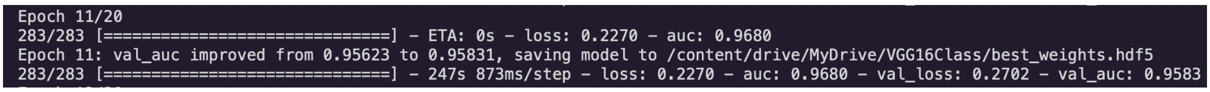
First, I validated the model using the AUC and Loss of validation and training data. After looking at the graph in figure 4a, I can see that the AUC for training data does not change much towards the end, and I can also see that in figure 4b the loss for training does not change much. Both of them almost becoming the same for each epochs in the end. Figure 4c shows that the Training AUC is 0.9680 and the Validation AUC is 0.9583, indicating that the model generated is a good model because the AUC is considerably higher. Figure 4c shows that the validation loss is bigger than the training loss, implying that the resulting model is not overfitted.



(a) Training vs Validation Area Under Curve for VGG16



(b) Training vs Validation Loss for VGG16



(c) Training vs Validation Loss and AUC for VGG16

Figure 4: AUC and Loss for VGG16 model

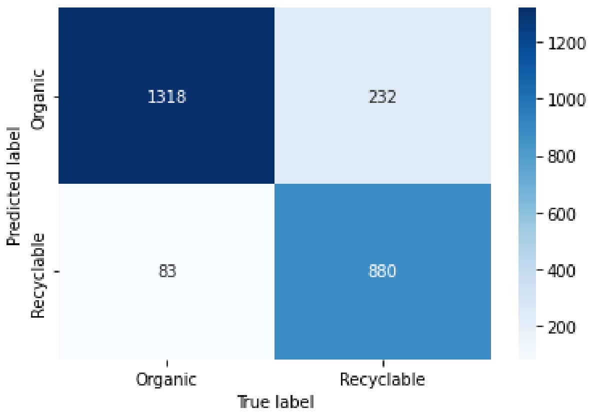
For the test dataset, I obtained an AUC of 0.9457, as shown in figure 5, indicating that the generated model performs remarkably well on the testing dataset. However, when I examine the loss, I notice that it is somewhat more than 0.34.So, let's check how well the model predicts the data in the confusion matrix.

Figure 6a shows that the developed model predicted 1318 photos as organic trash, which were truly organic waste, and 83 as recyclable rubbish, which were actually organic waste. The constructed model predicted 880 as recyclable trash, which is truly recyclable

```
79/79 [==============================] - 782s 10s/step - loss: 0.3457 - auc: 0.9457
Test dataset AUC: 0.945737 and Loss: 0.345722
```

Figure 5: VGG16 AUC and Loss for test data

waste, and 232 as organic garbage, which is actually recyclable waste. Figure 6b shows that for the organic waste class, the model gives accuracy, recall, and f1-score values of 0.94, 0.85, and 0.89, respectively, but for the recyclable class, the values are 0.79, 0.91.0.85. This demonstrates that our model is performing well in the organic class but not that accurate for the recyclable class. The overall accuracy is 87 percent, and the overall precision, recall, and f1-score are 0.87, 0.88, and 0.87, respectively. The all above observation shows that the build model is good model. If I compare this VGG16 model with the models developed by previous researcher for the waste categorization it show that the model which I have developed showed the significant improvement for waste classification model.



(a) Confusion Matrix For VGG16

	precision	recall	f1-score	support
Organic	0.94	0.85	0.89	1550
Recyclable	0.79	0.91	0.85	963
accuracy			0.87	2513
macro avg	0.87	0.88	0.87	2513
weighted avg	0.88	0.87	0.88	2513

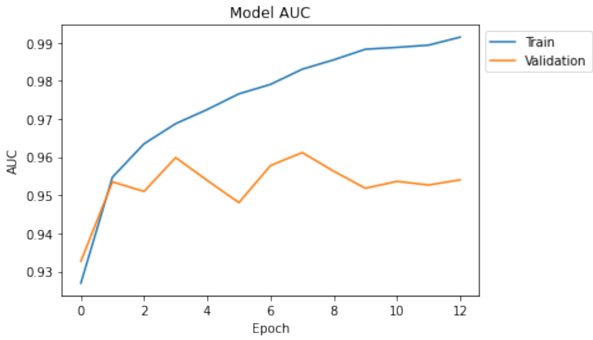
(b) Precision,Recall,F1-score and Accuracy for VGG16

Figure 6: Confusion Matrix and Obtained score from CM for VGG16 model

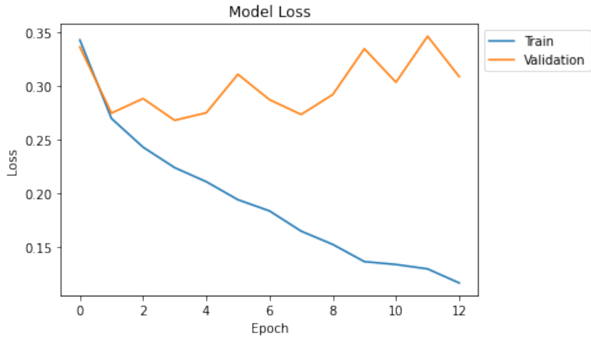
6.2 Densenet121 pre-trained Model

Initially, I used the AUC and Loss of the validation and training data to validate the model. The AUC for training data does not change much at the end of the graph in figure 7a, and the loss for training similarly does not significantly change, as can be seen in figure 7b. The Training AUC in Figure 7c is 0.9830, whereas the Validation AUC is 0.9612, demonstrating that the model created is a good model because the AUC is significantly higher. The validation loss is greater than the training loss, as seen in 7c, suggesting that the resultant model is not overfitted. If i compare this model with the VGG16 model which i have devloped i can say that on basis of observation till now the DenseNet121 performed better than the VGG16 model.

I achieved an AUC of 0.9619 for the test dataset, as shown in figure 8, suggesting that the constructed model works admirably on the testing dataset. When I study the loss, I discover that it is around 0.29 which shows that its little bit on higher side. So, let's see how well the model predicts the confusion matrix data.On the test data also the DenseNet121 model have preformed better than the VGG16.



(a) Training vs Validation Area Under Curve for Densenet121



(b) Training vs Validation Loss for Densenet121

```
Epoch 8/20
283/283 [==============================] - ETA: 0s - loss: 0.1643 - auc: 0.9830
Epoch 8: val_auc improved from 0.95987 to 0.96118, saving model to /content/drive/MyDrive/VGG16Class/best_weights.hdf5
283/283 [==============================] - 71s 249ms/step - loss: 0.1643 - auc: 0.9830 - val_loss: 0.2733 - val_auc: 0.9612
```

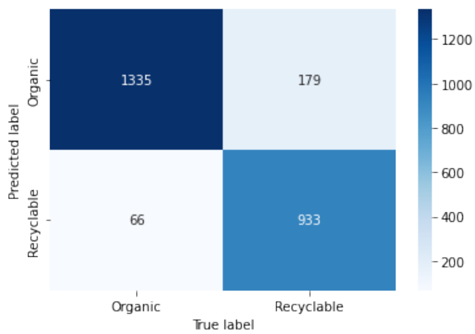
(c) Training vs Validation Loss and AUC for Densenet121

Figure 7: AUC and Loss for DenseNet121 model

```
79/79 [==============================] - 11s 107ms/step - loss: 0.2969 - auc: 0.9619
Test dataset AUC: 0.961853 and Loss: 0.296869
```

Figure 8: DenseNet121 AUC and Loss for test data

The generated model correctly identified 1335 photographs as organic trash, which were genuinely organic waste, and 179 photos as recyclable trash, which were also organic waste, as shown in Figure 9a. The built model projected 933 as recyclable trash, which is indeed recyclable waste, and 66 as organic rubbish, which is truly recyclable waste. According to Figure 9b, the model's accuracy, recall, and f1-score values are 0.95, 0.88, and 0.92 for the organic waste class, but 0.84, 0.93, and 0.88 for the recyclable garbage class. This shows that while our model does well for the organic class, it is less accurate for the recyclable class. The overall accuracy is 90%, while the overall precision, recall, and f1-score are, respectively, 0.90, 0.91, and 0.90. When I compare the overall outcome of the DenseNet121 model to the VGG16 model that I constructed, the DenseNet121 model outperformed the VGG16 model in every aspect.



(a) Confusion Matrix For DenseNet121

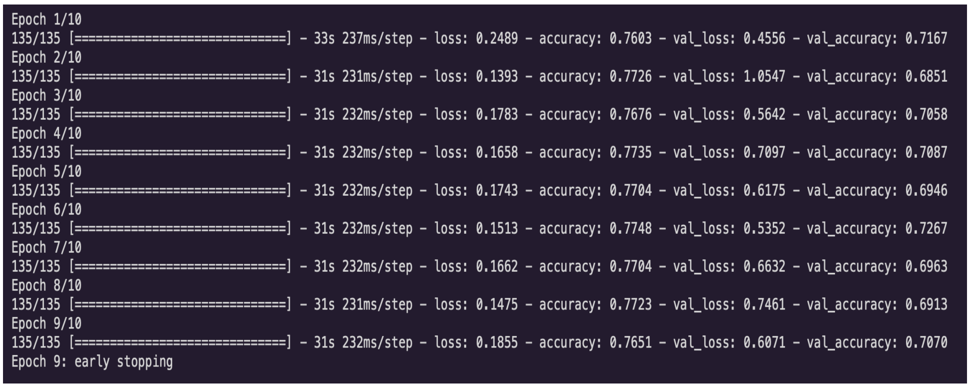
	precision	recall	f1-score	support
Organic	0.95	0.88	0.92	1514
Recyclable	0.84	0.93	0.88	999
accuracy			0.90	2513
macro avg	0.90	0.91	0.90	2513
weighted avg	0.91	0.90	0.90	2513

(b) Precision,Recall,F1-score and Accuracy for DenseNet121

Figure 9: Confusion Matrix and Obtained score from CM for DenseNet121 model

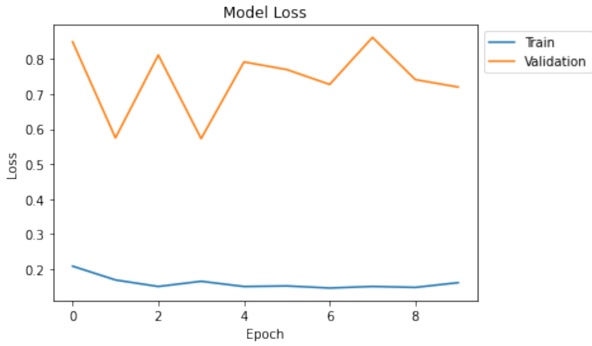
6.3 Image Segmentation

For my image segmentation model, I have got a loss of about 0.1855 and an accuracy of about 0.7651 while training the model. For the validation dataset, I have got a loss of around 0.6071 and an accuracy of roughly 0.7071, as seen in figure 10 . In light of this, it can be seen that the created model does not do well on the validation data. This is due to the fact that I don't have a large dataset to train the segmentation model.



```
Epoch 1/10
135/135 [==============================] - 33s 237ms/step - loss: 0.2489 - accuracy: 0.7603 - val_loss: 0.4556 - val_accuracy: 0.7167
Epoch 2/10
135/135 [==============================] - 31s 231ms/step - loss: 0.1393 - accuracy: 0.7726 - val_loss: 1.0547 - val_accuracy: 0.6851
Epoch 3/10
135/135 [==============================] - 31s 232ms/step - loss: 0.1783 - accuracy: 0.7676 - val_loss: 0.5642 - val_accuracy: 0.7058
Epoch 4/10
135/135 [==============================] - 31s 232ms/step - loss: 0.1658 - accuracy: 0.7735 - val_loss: 0.7097 - val_accuracy: 0.7087
Epoch 5/10
135/135 [==============================] - 31s 232ms/step - loss: 0.1743 - accuracy: 0.7704 - val_loss: 0.6175 - val_accuracy: 0.6946
Epoch 6/10
135/135 [==============================] - 31s 232ms/step - loss: 0.1513 - accuracy: 0.7748 - val_loss: 0.5352 - val_accuracy: 0.7267
Epoch 7/10
135/135 [==============================] - 31s 232ms/step - loss: 0.1662 - accuracy: 0.7704 - val_loss: 0.6632 - val_accuracy: 0.6963
Epoch 8/10
135/135 [==============================] - 31s 231ms/step - loss: 0.1475 - accuracy: 0.7723 - val_loss: 0.7461 - val_accuracy: 0.6913
Epoch 9/10
135/135 [==============================] - 31s 232ms/step - loss: 0.1855 - accuracy: 0.7651 - val_loss: 0.6071 - val_accuracy: 0.7070
Epoch 9: early stopping
```

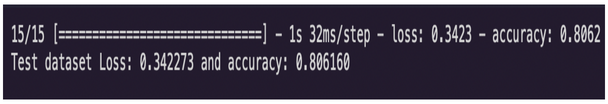
(a) Training vs validation Accuracy and loss for UNET



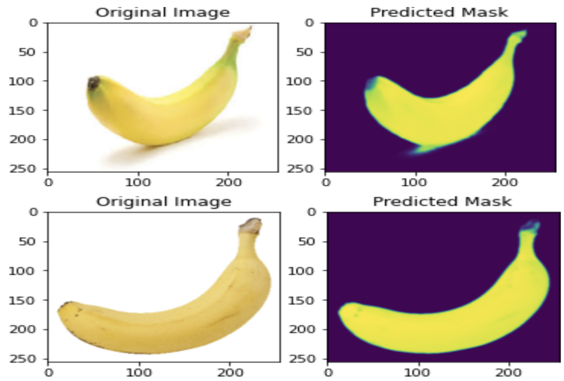
(b) Training loss vs validation loss for UNET

Figure 10: Accuracy and loss for UNET

The model did well on the test dataset, as evidenced by the loss of 0.34 and accuracy of 0.80 for the test dataset as shown in figure 11a. Figure 11b displays the segmentation's output.On which i can see that i am able to achieve the binary image segmentation with limited dataset.This model needs to improve in future by adding the more dataset and also this model can be improve by introducing the multi class image segmentation.



(a) Test Accuracy and loss for UNET



(b) Segmentation output UNET

Figure 11: Segmentation reuslts with test data

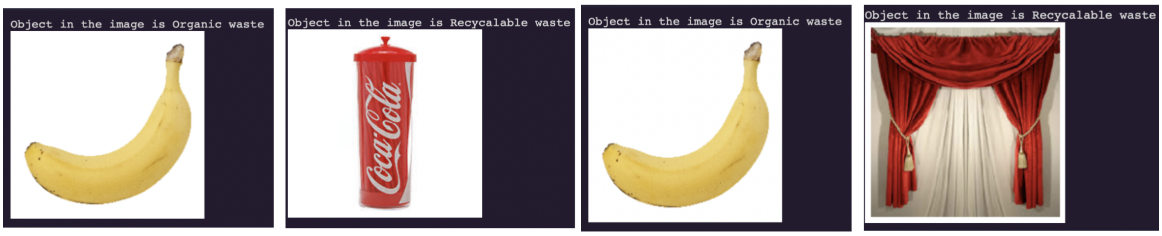
Table 1: Comparison of VGG16 and DenseNet121

	VGG-16	**DenseNet-121**
Training AUC	0.9680	0.9830
Validation AUC	0.9583	0.9612
Training Loss	0.2270	0.1643
Validation Loss	0.2782	0.2733
Test AUC	0.9457	0.9619
Test Loss	0.3457	0.2969
Accuracy	0.87	0.90
Precision	0.87	0.90
Recall	0.88	0.91
F1-score	0.87	0.90

7 Discussion

Both Ramsurrun et al. (2021) and Zhang et al. (2021) developed a categorization model for recyclable waste categories. Both employed the strategy to enhance the size of the dataset. In this article, Ramsurrun et al. (2021) employed a data augmentation technique, while Zhang et al. (2021) used a web crawler and manual photography to enhance the size of the dataset. The accuracy they have obtained with the VGG16 model is around 83% and 76% respectively. When we compare these models to the model I developed using the same technique, I got an accuracy of 87% on the testing dataset. This might be due to the use of a big dataset, as well as changes to the final layers of the VGG16 model.

When I compared this developed VGG16 model to the developed DenseNet121 model, I discovered that the DenseNet121 model outperformed the VGG16 model, as indicated in table 1. Furthermore, the datasets provided for the VGG16 are augmented, while the dataset provided for the DenseNet121 is not. If I extend the dataset and lower the learning rate, the accuracy of the DenseNet121 will improve. This, however, will need a large amount of computing power. However, based on the model training outcomes, DenseNet121 is the best model for this dataset. Figure 12 shows some predictions of the developed VGG16 and DenseNet121 models.



(a) VGG16 Prediction (b) DenseNet121 Prediction

Figure 12: Predictions using classification Model

Abdulrahman and Hewahi (2021a) in their research paper They segmented the ore images in their study article. They have several segmentation model architectures in this. In this study article, they obtained an accuracy of 92% and a loss of 0.14 with UNET model. The accuracy of the model I created is 0.80%, while the loss is 0.34%. When compared to the prior results, the model I constructed provided less accuracy. This due to a lack of data. The dataset that I utilized has fewer images.After building the model i have tested this on the on image which i have taken from internet which is open source image. The output of this shown in figure 11b.Then this image can be passed to the classification model which classify the segmented image.This model can be expand to the multi image segmentation in future to solve the classification problem of waste classification.

8 Conclusion and Future Work

The primary goal of this research is to create a waste categorization system using transfer learning models and an image segmentation model. I utilized the VGG 16 model and the DenseNet121 model to conduct this investigation. I extended the dataset to train the VGG16 model, whereas Densenet121 was trained on the original dataset. Following that, I created a binary image segmentation model utilizing the UNET architecture. I constructed the binary mask by annotating the dataset in order to train this model.

The proposed models were validated using the confusion matrix, Accuracy, AUC, and loss. The suggested VGG16 design performed better than previous studies in this area. When the two developed models were compared, the DensNet121 outperformed the VGG16. The image segmentation model was tested on accuracy and loss. To test the model, one image was imported from outside the dataset, and the results revealed that the model segmented the image correctly. I was able to develop the segmentation model despite the minimal dataset.

The dataset for image segmentation is the main limitation of the proposed methodology. By adding new data, this image segmentation model may be enhanced even more. Multi class image segmentation, which requires a large quantity of annotated data and is a time expensive procedure, may be used to develop the image segmentation model further. In the future, this proposed approach might be used in a real-world situation in which the first item is segmented and then submitted to a categorization model.

Acknowledgement

First and foremost, I want to thank my supervisor, Mr. Jorge Basilio, for his consistent support and guidance throughout this project. I'd also like to thank my parents for their constant support. Finally, I'd want to thank my housemates, colleagues, and friends for their unwavering support.

References

Abdulrahman, H. and Hewahi, N. (2021a). Efficient image segmentation based on deep learning for mineral image classification, *Advanced Powder Technology* **32**(10): 3885–3903.

Abdulrahman, H. and Hewahi, N. (2021b). Waste classifications using convolutional neural network, *IEEE Transactions on Artificial Intelligence* pp. 260–265.

Adedeji, O. and Wang, Z. (2019). Intelligent waste classification system using deep learning convolutional neural network, *Procedia Manufacturing* **35**: 607–612.

Alkassar, S., Abdullah, M. A. M. and Jebur, B. A. (2019). Automatic brain tumour segmentation using fully convolution network and transfer learning, *2019 2nd International Conference on Electrical, Communication, Computer, Power and Control Engineering (ICECCPCE)* pp. 188–192.

Aral, R. A., Keskin, R., Kaya, M. and Hacıömeroğlu, M. (2018). Classification of trashnet dataset based on deep learning models, *2018 IEEE International Conference on Big Data (Big Data)* pp. 2058–2062.

Ba Alawi, A. E., Saeed, A. Y. A., Almashhor, F., Al-Shathely, R. and Hassan, A. N. (2021). Solid waste classification using deep learning techniques, *2021 International Congress of Advanced Technology and Engineering (ICOTEN)* pp. 1–5.

Frost, S., Tor, B., Agrawal, R. and Forbes, A. G. (2019). Compostnet: An image classifier for meal waste, *2019 IEEE Global Humanitarian Technology Conference (GHTC)* pp. 1–4.

Gan, B. and Zhang, C. (2020). Research on the algorithm of urban waste classification and recycling based on deep learning technology, *2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL)* pp. 232–236.

Niu, S., Wang, J., Liu, Y. and Song, H. (2020). Transfer learning based data-efficient machine learning enabled classification, *2020 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)* pp. 620–626.

Qiuhao, Z. (2021). Kitchen waste classification based on deep residual network and transfer learning, *2021 6th International Symposium on Computer and Information Processing Technology (ISCIPT)* pp. 625–629.

Ramsurrun, N., Suddul, G., Armoogum, S. and Foogooa, R. (2021). Recyclable waste classification using computer vision and deep learning, *2021 Zooming Innovation in Consumer Technologies Conference (ZINC)* pp. 11–15.

Shaikh, F., Kazi, N., Khan, F. and Thakur, Z. (2020). Waste profiling and analysis using machine learning, *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)* pp. 488–492.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556* .

Singh, D. (2021). Polyth-net: Classification of polythene bags for garbage segregation using deep learning, *2021 International Conference on Sustainable Energy and Future Electric Transportation (SEFET)* pp. 1–4.

Sreelakshmi, K., Akarsh, S., Vinayakumar, R. and Soman, K. (2019). Capsule neural networks and visualization for segregation of plastic and non-plastic wastes, *2019 5th International Conference on Advanced Computing Communication Systems (ICACCS)* pp. 631–636.

Tammina, S. (2019). Transfer learning using vgg-16 with deep convolutional neural network for classifying images, *International Journal of Scientific and Research Publications (IJSRP)* **9**: 143–150.

Wang, D. and Wang, Z. (2021). Improved method of garbage classification based on deep learning, *2021 International Conference on Computer Network, Electronic and Automation (ICCNEA)* pp. 126–132.

Wu, Y., Qin, X., Pan, Y. and Yuan, C. (2018). Convolution neural network based transfer learning for classification of flowers, *2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP)* pp. 562–566.

Yang, J., Zeng, Z., Wang, K., Zou, H. and Xie, L. (2021). Garbagenet: A unified learning framework for robust garbage classification, *IEEE Transactions on Artificial Intelligence* **2**(4): 372–380.

Yang, X., Zhang, Y., Lv, W. and Wang, D. (2021). Image recognition of wind turbine blade damage based on a deep learning model with transfer learning and an ensemble learning classifier, *Renewable Energy* **163**: 386–397.

Zhang, Q., Yang, Q., Zhang, X., Bao, Q., Su, J. and Liu, X. (2021). Waste image classification based on transfer learning and convolutional neural network, *Waste Management* **135**: 150–157.