An Approach of Classifying Waste Using Transfer Learning Method

Zian Md Afique Amin, Khan Nasik Sami, Raini Hassan

Department of Computer Science, Kulliyyah of ICT,
International Islamic University Malaysia, Kuala Lumpur, Malaysia.
afiquezian@gmail.com

Abstract— One of the most critical issues facing the world is waste management, regardless of whether the region is being established or becoming established. There is a waste partitioning process in waste management, and the main challenge is that the garbage space is flooded long before the beginning of the following cleaning process at clear spots. Only unskilled workers conduct waste separation, which is less accurate, time-consuming, and not utterly possible due to the enormous amount of waste. Using the Convolutional Neural Network, we propose an artificial waste classification problem to compile and organize a dataset into seven categories consisting of metal, plastic, glass, paper, cardboard, trash, and E-waste. We then distinguished between specific transfer learning algorithms for our project: Xception, DenseNet121, Resnet-50, MobilenetV2, and EffiecienNetB7. DenseNet121 achieved a high precision characterization of about 93.3% for our model, while Mobilenet also demonstrated an incredible conversion to different forms of waste of 93% and Resnet-50, Xception and EfffiecienNetB7 achieved 92%, 92.5%, and 87%, respectively. In the future, we would like to increase the accuracy by using some other hyperparameter tuning, and we would like to deploy the project on mobile devices. We will use dockers or Kubernetes to deploy and YOLO real-time object detection as a framework for the post.

Keywords— Waste Classification, Image Classification, Machine Learning, Deep Learning, Transfer Learning, MobileNet, Resnet-50, EffiicienNetB7, Convolutional Neural Network, Sustainable Development Goals

I. Introduction

Recently, the production of waste has risen significantly. If waste managing waste is not done correctly, it may have a destructive impact on the atmosphere. Therefore, sorting the waste to optimize the number of recyclable materials should be achieved at the initial level. And there would be less likely for other items to damage the environment. It's down to a lot of pollution. Every year, the planet produces around one and a half billion tons of heavy civilian waste [1].

The World Bank has estimated that the world emits about 1.5 billion tons of highly industrial waste per year, with that figure expected to rise to 2.2 thousand pounds by 2025. Diversion of plastics from waste to reuse would conceivably spare up to 20 per cent of what could be equivalent to 60 million barrels of oil per year and reduce landfill requirements. The U.S. Natural Protection Agency has recommended that the reduction, reuse, volume reduction, and landfilling of the source be extended to managing urban heavy waste in a particular order. As of late, they are reusing has become a significant segment of the executive's initiatives for city squandering. There is sufficient computerized engineering to recover MSW steel, aluminium, compostable food waste, and paper products. The heft of MSW is made up of these elements. In any event, the primary current technique for recycling MSW glass and plastic holders is to handpick them from the waste manually.

The United Nations declared 17 Sustainable Development Goals (SDGs), also known as the global goals of eliminating poverty, preserving the environment, and ensuring peace and security for all citizens by 2030. We have taken our project as a landmark, based on the SDGs' priorities, to address some of the problems facing almost all nations. Our research aims to achieve goal number 14 by destroying the remnants of plastic fishing nets, cigarette butts, and other sea debris forms. Big aquatic animals have perished because their stomachs are too full of plastic and are dead onshore. This finding suggests Goal 15 though life on land (SDG 15) can only be safe if waste is adequately controlled. Again, because there is no management, waste pollutes the environment. For the burning of plastics, open burning's health consequences are disastrous everywhere (SDG 3). The primary goal of SDG3 is to ensure safe lives and encourage well-being for all ages. Besides, up to a tenth of human-made greenhouse emissions would be caused by climate change, the risks of methane and CO2 releases from poorly managed material. As a result, goal 13 will be hampered, pointing to the climate effect (SDG 13) on our culture.

Consequently, if we want clean water and sanitary facilities (SDG 6) [3], we might have to think about waste being segregated appropriately. So, for sustainable development goals, our study carries high importance. Therefore, the implementation of A.I. and machine learning will provide a decent response to managing this significant problem and sustain our condition as an ideal place for everybody to live.

Here some of the research questions we have listed down:

A. What are the categories in this research for segregation: The primary purpose of this project is to determine whether the waste is paper, plastic, metal, glass products, cardboard, or e-waste? By comparing the properties of the given examples to our training data in the dataset, we attempted to determine the identified testing data.

B. What is the essential attributes for deciding the proper object: At that point, our code initially took an input picture and separated the characteristics. It compared the features with the previously trained data at that stage. Eventually, it decided if the data or how accurate the trained model is and whether the object provided is compatible with it or not. It is a more innovative option to do some pre-processing in the data to display signs of change inaccuracy so that the object can be analyzed from various angles and views. It is often easier to do reshaping in the pre-processing step to hold all instances of a similar scale.

Transfer learning and fine-tuning are two pretty similar concepts in many aspects, and the two terms are frequently used interchangeably. Although the two conditions do not relate to the same objective or motivation, they allude to the same concept. Fine-tuning entails taking a machine learning model that has already learned something on different data and training it on new data in most instances. Transfer learning is when a machine learning model's expertise is applied to a new task. We employ part or all a model that has been trained on something like a new model and train it on new data. Even though both Transfer Learning and fine-tuning models pertain to the concept of taking an existing, learned model and further training it, either as is or as part of a new model, they are not the same thing. However, in Transfer Learning, the model is trained with one dataset and then used to train another dataset with a different distribution of classes, or even with classes different from those in the training dataset. The model helps to see how the item will fluctuate based on various factors using classification algorithms and distinct transfer learning methods. Separating the images into tiny chunks, for example, and then inserting them into neural network layers. Each of the neural layers uses various filters and methods to help classify the object. We, therefore, need to resize it at that point while preserving the image's subtleties. 224*224 sized photos are accepted from ImageNet transfer learning models from Keras. Then we resized it while maintaining the image's details. The model estimates the precision of how likely an item is to fit the trained sample after considering all the variables. This research aims to explore the dataset, which includes analyzing each function's variable to see if the features are essential for the model's creation. Secondly, we visualize the dataset, find incorrect images, and develop models that categorise and label the images according to the groups. Then, we analysis, the accuracy based on performance evaluation. Finally, we will see the most effective algorithm for this operation.

II. Related Works

Many researchers have used several different methods for classifying wastes into different categories. The use of Machine Learning and Deep Learning Algorithms are the most popular approaches for it. For the deep learning algorithms, many researchers have used deep transfer learning methods with various layers of neural networks to get the best result for the segregation of images.

This section contains findings from the literature reviews conducted to get a sense of related works in this field. We related several works of literature in this paper.

Pearson Correlation Coefficient and Spearman Rank Order Correlation were used to calculating the performance of every neural network. VGG16, Alexnet, SVM, KNN, and Random Forest were used for their experiments. VGG16 transfer learning method was the most successful process in this research, with 93% accuracy [4].

In a research paper, the researchers have explored the percentage of waste of each category and its adverse impact on Grenada's economy. The researchers have calculated the amount of waste from different sources and prototyped the model for recycling the waste of different categories in the city.

In another research, the researchers have used ImageNet classification with deep convolutional neural networks. ReLU was used to bring non-linearity, multiple GPUs were used, and Local Response Normalization and Overlapping Pooling were implemented. To reduce the overfitting, data augmentation and dropout layer were used. The findings show that using purely supervised learning and an extensive, deep convolutional neural network can achieve record-breaking results on a highly challenging dataset. It is noteworthy that removing a single convolutional layer degrades the output of our network [5].

Different machine learning models such as Decision Tree, Random Forest, and SVM were used along with Deep Neural Networks. It has been seen that CNN performed the best among all other methods with an accuracy rate of 90% [6].

CNN was used on Synthetic Aperture Radar Target Recognition Dataset. It was a multi-class classification problem with an image dataset of 10 categories. 4 different datasets were used, and six different algorithms were also used. The umbrella method had the most minimum loss rate.

In a separate experiment, Depth, wise Separable Convolutions along with Deep Learning was introduced. It is one of the most popular transfer learning models, which is Xception Net. It was compared with other transfer learning models, such as VGG16, Resnet152, and Inception V3. It has been seen that in both Top-1 Accuracy and Top-5 Accuracy, Xception Net was the most successful model among the other three. It was also the best for neural networks with no fully connected layers in the experiments [7].

Xception was also used in research. It has been seen that for the conducted experiments of benchmarking, Xception was again the most successful transfer learning model among the other models. However, the model size was a bit larger than the others [8].

Few research was for the automatic segregation of waste materials. Internet of things helped to count and monitor the type of waste and its quantity in mobile phone applications in this research. ESP8266 Wi-Fi module, 3 I.R. Sensor, three servo motors, metal sensor, the motor drive was used in this research [9].

Deep transfer learning model ResNet-50 was implemented in this research for classifying Malicious Software [10]. The experiment was done on the Malimg Dataset, which has 25 different categories of malware families. KNN classifier using the bottleneck features instead of the GIST features was used to perform the quantitative analysis in this research.

Fault diagnosis based on the ResNet-50 transfer learning model was used in other experiments. TCNN structures were proposed in this research [11]. Cross-Validation was used to evaluate the performance of the model. Tenfold CV was used on KAT Bearing Dataset for this experiment. Classification and regression trees (CART), extreme learning machine (ELM), Boosted Trees (B.T.), neural networks (N.N.), random forests (R.F.), Support Vector Machines with parameters optimally tuned using particle swarm optimization (SVM-PSO), k-nearest neighbours (KNN) and their ensemble algorithms using majority voting (Ensemble) were used to compare the performance [12]. The research shows that TCNN (ResNet50) has achieved the maximum prediction accuracy of 98.95% among all other different methods.

The classification of waste several Neural Network Algorithm has created various impacts, especially detecting glass objects and metal objects with ANN algorithms [13]. This technique has been used in the modern intelligent waste management and recycling system. Again, this method is obvious to detect and to obtain satisfactory results.

IoT-related devices are also used in classifying waste, but they have mainly focused on the ML model and come up to 75% accuracy. Automatic classifying waste and managing the garbage using vast modern technology, and implementing ML algorithms with a trash bin and monitoring system gives perfect waste classification [14].

Another research has shown that machine learning classification such as Support Vector Machine sometimes performed better than CNN [15]. The study compares the machine learning algorithm and deep learning algorithm to detect the best one and found that machine learning is better to choose. Also, the paper focused on the recycling of waste correctly.

On the other hand, garbage detection using a deep learning computer vision approach like CNN with Android smartphones and other modern devices showed approximately 89% accuracy [16]. This system deploys a mobile camera to show and detect the image of waste and try to classify and segregate them in good areas. A technique consolidates image features and other feature descriptions with multilayer perceptron to identify waste as recyclable or other waste. The model was deployed in a mobile app named SpotGarbage.

Multilayer hybrid-Deep learning method for waste classification and recycling is another advanced approach that has taken place and finds up to 99% accuracy for this hybrid system [17]. After reshaping the image 256*256 coloured PNG images, the researcher applies them to IoT devices and record the result for comparison where the hybrid system surpassed the usual technique.

Some of the transfer learning methods showed mixed results, and DenseNet121 scored highest among them by having 95% accuracy [18]. This research will continue to the CNN structure and expand the sample collection for the training and testing phase to compare the new results with the old one.

A few researchers researched and compared the epoch in the training set and figured out that the deep neural approach with ten epochs scored the max accuracy, and the accuracy level is nearly 99%. With lightweight network (transfer learning) such as mobileNetV2 and SVM figured out 98.4% accuracy [19].

Lightweight neural helps to train the model fast and smooth to run and detect images. Open-sourced library has been introduced in this research.

Researchers used malicious software classifications on using ImageNet and Resnet-50 networks to create the best comparison in transfer learning. This result showed very high accuracy, nearly 99%. [20]

In-Depth wise separable convolutions, the Xception transfer learning method showed more reasonable accuracy. The research used InceptionV3 and ImageNet images to detect and compare [21]. The researcher will look into the advanced research technique in this field.

The researcher and Resnet101 have introduced an autonomous traffic surveillance system with few Neural network transfer learning method showed more accuracy, but VGG16 and AlexNet also used for the classifications and localization [21].

Researchers have utilized various algorithms and transfer learning methods to compare performance metrics and get more than 90% accuracy. Vgg16, Resnet-50, inception V3 and Xception has been used mostly. Research has brought remarkable success in image classification problems by using CNN and transfer learning methods. These research knowledge has helped us to gain information about various model architecture and the functionlaties with the accuracy based on various situations which helped us to get our obejctives and choose the right methodology.

III. Methodology

We have implemented the standard data-science lifecycle for the research. After collecting the appropriate dataset, we applied our main transfer learning methods and tested the performance measurement results. The methodology cycle has shown in all the steps in fig 1.

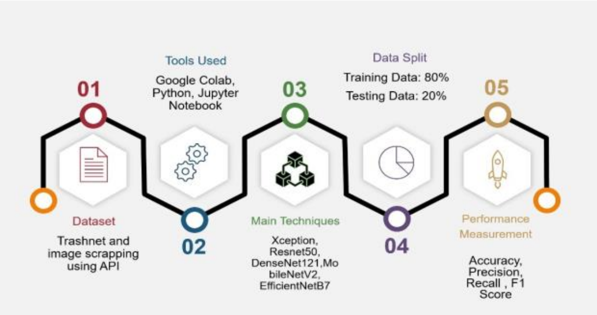

Tools Used
Google Colab, Python, Jupyter Notebook
Data Split
Training Data: 80%
Testing Data: 20%
01
03
05
Dataset
Trashnet and image scrapping using API
02
Main Techniques
Xception, Resnet50, DenseNet121,MobileNetV2, EfficientNetB7
04
Performance Measurement
Accuracy, Precision, Recall , F1 Score

Fig. 1 Methodology

A. *Data Collection and Categories*

In the dataset, we used the waste image dataset that Gary Thung and Mindy Yang created. The dataset was accessible as an open-source on the internet, and the dataset owner provided permission to use it for research purposes. Then we made another e-waste class and attempted to collect more photographs in that group. The dataset includes more than 2800 photographs of assorted cardboard, metal, plastic, paper, bottles, metals, and e-waste. E-waste is also very harmful to the atmosphere and should be deemed essential. It is also very much dangerous for the environment and should be given importance. We also categorized the dataset into sets for training and testing set. We have used google image API to scrap different e-waste images using Python and explored the BING image API to collect the data from the internet [3]. We have added 500 more pictures of this category for the experiment.

B. *Research environment and setup*

We specifically used Google colab as an IDE and Python as the programming language with Tensorflow framework for our original experiment and review.

- Google Colab: It is a free application developed by Google and hosted on the cloud to do the coding experiments. One of the key advantages of the Colab is that the runtime can be modified quickly. We use the GPU runtime for our project because the dataset is quite large. It provided free GPU support for training our models. Python programming language is used in colab environment. It is easy to visualize the image outputs and graphs. For research, google colab is one of the best environments. Users can also use PyCharm, Visual Studio, or another high-level compiler or interpreter for production-based applications.
- Python: Python with dynamic semantics is an interpreted, object-oriented, high-level programming language. It is a high-level language with pre-built data structures. It can be combined with dynamic typing and dynamic linking, making it very attractive for Machine Learning research and software development.
- Numerical equations and machine learning algorithms were included (s). The main attributes for the models to fit the data in mathematical matrixes are finding the loss function and finding the accuracy and results. Machine Learning algorithm(s) were utilized, along with numerical equation(s). We have used different state of the art variants of CNN to transfer learning models for our research purpose.
- TensorFlow: Google launched Tensor Flow in 2015 as a free framework to make modeling, creating and training deep learning models smoother for developers and researchers. TensorFlow is only one of the first choices open to developers, and because of its versatile nature and ease of usage, we have used it in our research. TensorFlow uses the Python language at a high level to express arbitrary programming functionalities. The data in TensorFlow is interpreted as multidimensional arrays of tensors. TensorFlow is mainly used for deep learning in practice and study, but this tool for thinking about computation is helpful in several different fields. We have used TensorFlow version 2.x and Keras, a wrapper over TensorFlow, for our project.

C. *Performance Measurement*

Confusion matrix, accuracy, precision, recall, F1 score are some of the performance assessment criteria we used to examine the algorithms' performance. We have also tried with stratified k-fold cross-validation, but the result was not improving significantly. Therefore we focused more on data augmentation to make a robust model and improve the model performance by avoiding overfitting.

D. *Data Preparation & Pre-Processing*

Since the data used in this research has various waste-related images, some data pre-processing was performed to make the images suitable for fitting into the A.I. models. Images have conflicting sizes in the training dataset. As a result, pictures must be reformatted before being used as part of the model. As we have gathered the data from various sources and image scrapping, all the images did not have similar pixels, height, and width. ImageNet models require the training data to be reshaped into a fixed size to fit the image pixel matrices into the neural network layer. The classification models do not provide a biased and incorrect prediction result. The waste images in the training dataset were reformatted from the original shape to 224x224 pixels. We have constructed our models on this transformed data. We have also used the data augmentation method to avoid overfitting the model on the training set. Data augmentation does image flipping, rotation, scaling, cropping, translations, interpolation, and in some cases, Gaussian noise inclusion to add variation in learning the image from different image states. It is found that data augmentation can boost the model performance up to 21.6% if appropriately implemented. We have tried out experiments in different sets of data separation ratio. But 80:20 data split worked the best for me. It is also proved from the Pareto Principle and Scaling Law, which was introduced in 1997. It is found that the number of patterns reserved for the validation set should be inversely proportional to the square root of the number of free adjustable parameters.

E. *Data Visualization*

Here are some of the instances of the data, which we used to work on. We have scraped data online for our research. While doing the scrapping, multiple redundant data were stored. Many of them were not suitable for our study, as they contain objects which are not included in our research. In addition, many data extensions were not appropriate for the experiments, such as gif, animations, and poor-quality images. We have visualized the data and then removed the unwanted data to make a suitable training set to feed into our models. We have used an open-source application called Netron to explore the neural network layers' functionalities and attributes and transfer learning model architectures. Fig 2 and 3 demonstrates the data visualization. Netron helped us to see every single layer and the interconnections between them. Seeing the neural networks graphically helped us in our research. We could differentiate the distinct attributes for each model and see why a model performs in a certain way. It allowed us to understand why some models took a long time to run compared to other models. Even in the ImageNet model, sometimes trimming off some layers helps to make a robust model.

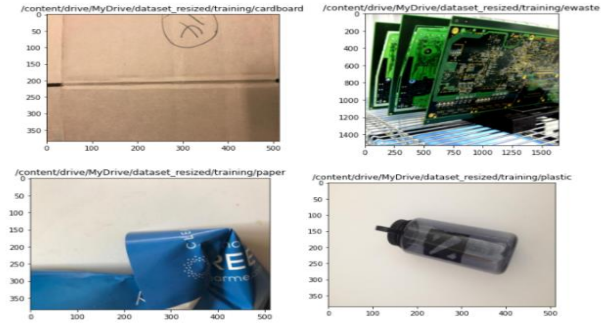
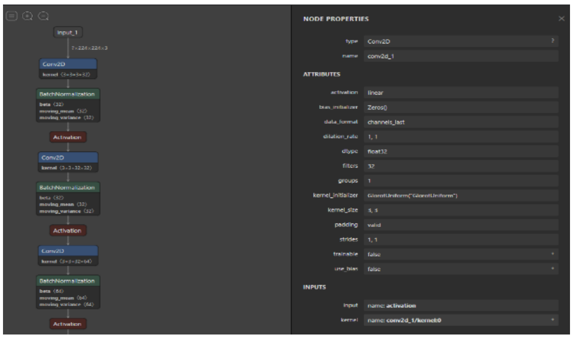

/content/drive/MyDrive/dataset_resized/training/cardboard
/content/drive/MyDrive/dataset_resized/training/ewaste
/content/drive/MyDrive/dataset_resized/training/paper
/content/drive/MyDrive/dataset_resized/training/plastic
0
50
100
150
200
250
300
350
400
500
600
800
1000
1200
1400
750
1250
1500

Fig. 2 Data Visualization



NODE PROPERTIES

type: Conv2D
name: conv2d_1

ATTRIBUTES

activation: linear
bias_initializer: Zeros()
data_format: channels_last
dilation_rate: 1, 1
dtype: float32
filters: 32
groups: 1
kernel_initializer: GlorotUniform("GlorotUniform")
kernel_size: 3, 3
padding: valid
strides: 1, 1
trainable: false
use_bias: false

INPUTS

input: name: activation
kernel: name: conv2d_1/kernel:0

Fig. 3 visualizing the layer in Netron App

IV. Modelling

A. *Xception*

The architecture of Xception is a linear stack of residual ties of strongly separable convolution layers. Making it very simple to describe and change the architecture; using a high-level library such as Keras or TensorFlow-Slim requires just 30 to 40 lines of code [6], unlike architectures like VGG-16, but unlike architectures like Inception V2 or V3, which are much more difficult to describe. The License agreement supports an open-source implementation of Xception using Keras and TensorFlow as part of Module2 of Keras Software. The visualization of cross-channel correlations and spatial correlations are included in the position maps of convolutional neural networks. Since this idea is a more robust version of the Inception model's hypothesis, we call our proposed architecture Xception, which stands for "Extreme Inception." The feature extraction base of the Xception architecture is made up of 36 convolutional layers [7]. We will exclusively examine picture classification in our experimental assessment, and therefore a logistic regression layer will obey our co-evolutionary basis.

Optionally, before the logistic regression layer, which is discussed in the experimental assessment portion, one may insert fully connected layers. Except for the first and last modules, the 36 convolutional layers are ordered into 14 components, all of which have linear residual connections between them [8]. Fig 4 shows the flow architecture of the Xception model.

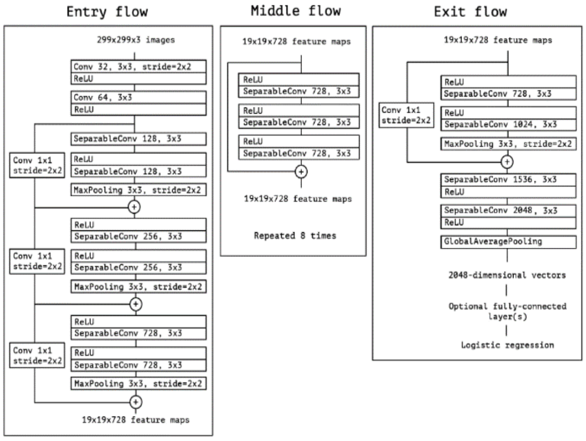


Fig. 4 Xception Architecture

B. Resnet-50

ResNet is a modular core architecture that is extensively used in many image classification tasks. The ResNet-50 model is made up of five-stage convolution and identity chains. Each convolution layer has three convolution layers, and each recognition block has three convolution layers [9]. Thus, ResNet-50 can be based on over 23 million variables. In addition, a skip connection is used by ResNet to move output from a previous layer to the last layer. This helps resolve the degradation problem. The transfer learning process for Resnet-50 is shown in Fig 5.

Residual Networks (Resets) are deep convolutional networks with simple bypassing blocks of convolutional layers using quick ties. The 'bottleneck' is a fundamental block that implements two relatively simple rules: I) for the same output feature map size, the layers have the same number of filters, and (ii) the number of filters is multiplied if the feature map size is halved [10]. The downsampling is transmitted simultaneously by convolutional layers with a step of 2, and the batch normalization is done right after each convolution and before the ReLU is triggered. When the input and output dimensions are the same, the recognition shortcut can be used. If the measurements get broader, the projection shortcut is used to balance them using 1 x 1 convolutions. They are done with a stage of 2 in both cases, although the shortcuts were in feature maps in two scales. Finally, the network is halted by a 1,000 connected (fc) layer with SoftMax activation [11]. There are 50 layers balanced in maximum, with 23,534,592 trainable parameters.

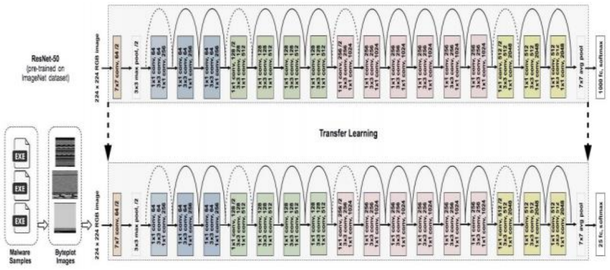
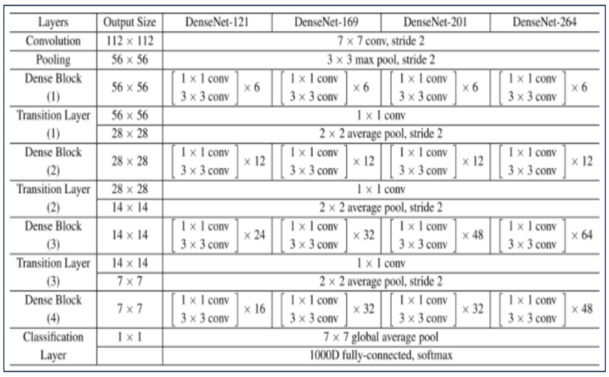


Fig. 5 Resnet-50 Architecture

C. DenseNet121

DenseNet (Dense Coevolutionary Network) is a structure that involves making deep learning networks go even deeper but still making them more active for testing by using shorter interactions between the layers. DenseNet is a neural co-evolutionary network in which each layer is attached to the layers underneath it, i.e., the first layer is linked to the 2nd, 3rd, 4th, and so on, the second layer is connected to the 3rd, 4th, 5th, and so on [12]. This is achieved to allow for the optimal transfer of information between the network layers. Each layer obtains inputs from all preceding layers to sustain the feed-forward nature and passes on its feature maps to all subsequent layers [13]. It does not combine functions by summation, unlike Resnets, but it incorporates the features by concatenating them. So, the ith 'layer has' I 'inputs and consists of all the preceding convolutional blocks function maps. Both the following 'I-i' layers are moved on with their feature charts. As in typical deep learning architectures, this introduces'(I(I+1))/2 'connections in the network, rather than just' I 'connections [14]. Therefore, it needs fewer parameters than conventional convolutional neural networks since unimportant feature maps do not need to be trained. Four DenseBlocks of varying layer numbers are part of each architecture. For example, DenseNet-121 has [6, 12, 24, 16] layers in the four dense blocks. Fig 6 shows the architectural overview of DenseNet-121, and it compares the attributes with other variations of the DenseNet model.



Layers	Output Size	DenseNet-121	DenseNet-169	DenseNet-201	DenseNet-264
Convolution	112 × 112	7 × 7 conv, stride 2			
Pooling	56 × 56	3 × 3 max pool, stride 2			
Dense Block (1)	56 × 56	[1 × 1 conv 3 × 3 conv] × 6	[1 × 1 conv 3 × 3 conv] × 6	[1 × 1 conv 3 × 3 conv] × 6	[1 × 1 conv 3 × 3 conv] × 6
Transition Layer (1)	56 × 56	1 × 1 conv			
	28 × 28	2 × 2 average pool, stride 2			
Dense Block (2)	28 × 28	[1 × 1 conv 3 × 3 conv] × 12	[1 × 1 conv 3 × 3 conv] × 12	[1 × 1 conv 3 × 3 conv] × 12	[1 × 1 conv 3 × 3 conv] × 12
Transition Layer (2)	28 × 28	1 × 1 conv			
	14 × 14	2 × 2 average pool, stride 2			
Dense Block (3)	14 × 14	[1 × 1 conv 3 × 3 conv] × 24	[1 × 1 conv 3 × 3 conv] × 32	[1 × 1 conv 3 × 3 conv] × 48	[1 × 1 conv 3 × 3 conv] × 64
Transition Layer (3)	14 × 14	1 × 1 conv			
	7 × 7	2 × 2 average pool, stride 2			
Dense Block (4)	7 × 7	[1 × 1 conv 3 × 3 conv] × 16	[1 × 1 conv 3 × 3 conv] × 32	[1 × 1 conv 3 × 3 conv] × 32	[1 × 1 conv 3 × 3 conv] × 48
Classification Layer	1 × 1	7 × 7 global average pool			
		1000D fully-connected, softmax			

Fig. 6 DenseNEt121 Architecture

D. MobileNetV2

MobileNetV2 is a convolutional neural network optimized to function well in smartphones. It is based on an inverted residual structure between the bottleneck layers where the residual connections are [15]. As a source of non-linearity, the intermediate expansion layer uses lightweight depth-wise convolutions to filter features. The MobileNetV2 architecture includes the initial complete convolution layer with 32 filters, followed by 19 residual bottleneck layers.

Fig 7 depicts the general architecture of a MobileNetV2 model. The theory is that the bottlenecks translate the model's initial inputs and outputs, while the inner layer captures the model's ability to convert from lower-level concepts like pixels to higher-level descriptors like picture types. Finally, as with traditional residual links [16], shortcuts make for quicker training and better precision.

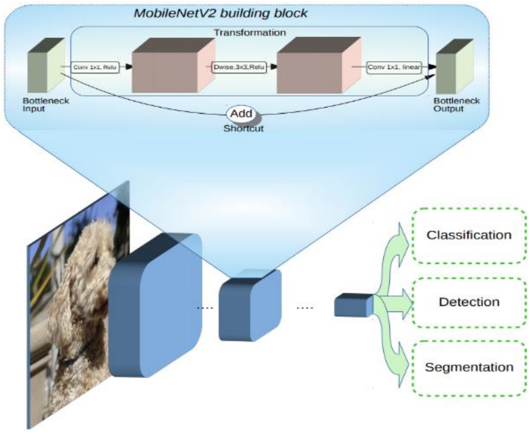


Fig. 7 MobileNetV2 Architecture

E. EffiecienNetB7

EfficientNet is a neural network architecture and scalable method based on convolutional layers that use a collective coefficient to scale all depth/width/resolution parameters uniformly. Unlike standard technique, which randomly scales network size, Depth, and resolution, the EfficientNet also including uniformly scales network distance, Depth, and resolution with a set of fixed scaling coefficients [17].

Combined scaling is that if the input image is larger, the network requires more structures to increase the activation function and more channels to capture the bigger picture with finer symbols.

EfficientNet-architecture B7's was created by the neural network itself rather than by researchers. They developed this model that enhances both precision and floating-point operations using a multi-objective neural networks scan [18]

Convolutional Neural Networks (ConvNets) are commonly created at a fixed resource budget and then scaled up if more resources are required for better accuracy. The researchers built an entire family of EfficientNets from B1 to B7, taking B0 as a baseline model, which achieved state-of-the-art accuracy on ImageNet while being very effective for its competitors. We used the B7 version variant of the model. Instead of ReLu, EfficientNet used the Swish Activation function. It also utilizes Squeeze and Block of Excitation. EfficientNet-B7 on ImageNet achieves state-of-the-art accuracy of 84.3 per cent top-1 while 8.4x smaller and 6.1x faster on inference than the best current ConvNet. Fig 8 illustrates Xception architecture with different scalings.

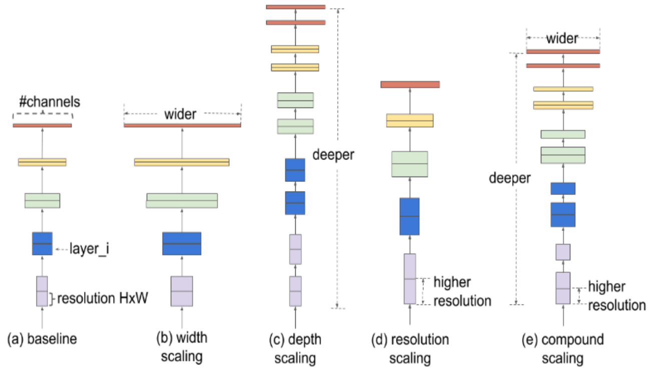


Fig. 8 Xception Architecture

V. IMPLEMENTATION

A. Xception

There are three primary sections of the Xception module. The flow of the entrance, the middle flow (which is repeated eight times), and the departure flow. There are two blocks of the convolutional layer in the entry phase, accompanied by a ReLU activation. The illustration also discusses the number of filters, the height of the filter (kernel size), and the strides in detail. There are also separate layers of Separable convolution. Max Pooling layers still occur. If the steps are different from one, the steps are listed as well. Skip connections are also open, where we use 'ADD' to combine the two tensors. In each flow, it also displays the form of the input tensor. We begin with an image size of 299x299x3. For instance, we get an image size of 19x19x728 after the input process. Similarly, this diagram explicitly illustrates the image size, different layers, number of filters, form of the filters, method of pooling, number of repetitions, and the probability of inserting a fully connected layer for the Middle Flow and Exit Flow.

B. Resnet-50

ResNet50's architecture has 5 phases. The network can take the input signal's height, width input as multiples of 32 and 3 as channel width, which is for RGB filters. We will assume the input layer's dimension to be 224 x 224 x 3 for the sake of clarification. Each ResNet architecture executes the original convolution and max-pooling using 7 x 7 and 3 x 3 kernel size. Afterwards, the network begins with Stage 1, and the network has three residual blocks, having three layers each.

The kernel size used to execute the convolution process is 64, 64, and 128, respectively, in all three layers of the stage 1 block. The curved arrows apply to the relation to the identification. The bottleneck architecture is used for deeper networks, such as ResNet50, ResNet152, etc. Three layers are stacked one over the other with each residual function F. 1x1, 3x3, 1x1 convolutions are the three layers. The 1/1 convolution layers must minimize and then recover the measurements. With narrower input/output lengths, the 3 x 3 layer is left as a bottleneck. In the end, the network consists of an average pooling layer, which is followed by a fully connected layer of 1000 neurons, which is ImageNet class output for multi-class classification [19].

C. DenseNet121

Other than the usual convolutional and pooling layers, DenseNet is made of two major blocks. They are the layers of the Thick Blocks and the Transformation. With a superficial pooling layer and convolution, DenseNet begins. There is then a dense block of neural network followed by a layer of transition, another dense block followed by a transition layer, another layer of transition that follows a dense block, and eventually, a dense block of network followed by a classification layer.

There are 64 7x7 filters and a phase of 2 in the first convolution block. A layer of MaxPooling accompanies this with 3x3 max pooling and a step of 2. Each of the 1x1 convolutions has four times the number of filters in a dense block. But we use 4* filters, but only once are there 3x3 filters. Even with the output tensor, we must concatenate the input. Each block is run using the 'for loop' for the 6,12,24,16 repeats, respectively [20].

D. MobileNetV2

We have used the second version of MobileNet, for our research project. It has additional performance improvement and optimization in comparison to the older version. We have used the pre-trained weights of ImageNet and made the parameters non-trainable as they are already trained. We used max-pooling and SoftMax activation function at the output layer as we solve a multi-class classification problem.

We used Sequential layers to train our model. After importing pre-trained weights, we have added our 7 class images at the last layer, rather than using 1000 trained classes of ImageNet. To avoid overfitting, we have used data augmentation on the training data. Also used Early Stopping to prevent the model overfitting. After training and saving the model, we have measured the accuracy and other performance measurements of the model discussed in the latter part of the paper.

E. EffiecienNetB7

We have separated the data into three sets for the experiment. We have used the pre-trained weights of ImageNet and made the parameters non-trainable as they are already trained. We then made the base model, which is EfficientNetB7 non-trainable. Instead of using the 1000 qualified imagery classes, we only applied our 7 class images to the last layer. We have used functional layers for this experiment. In total, there were 64,975,774 parameters, and we had to train 878,087 of them. We also used data augmentation on the training data to prevent overfitting. The model stopped training after 18 epochs, as the validation was not improving after a particular stage. Using EfficientNet was because of its lite weight, output model nature, which is compatible with mobile devices [21].

VI. Result Analysis

A. Accuracy

Accuracy is the ratio between the total number of accurate predictions and the total number of predictions.

Using accuracy as a determining criterion for the model makes good sense, but it is often advised to use Precision and Recall. This is because there may be other cases in which our precision is relatively high, but our recall or accuracy is comparatively poor. Ideally, it is easier to eliminate such circumstances for models. Accuracy is how close a measured value is to the actual value.

Fig 9 shows that every method's accuracy can speak the story that DenseNet121 is the more accurate for this research, where XceptionNet and MobileNetV2 are slightly less than DenseNet121 and EfficienNetB7 has shown the lowest accuracy.

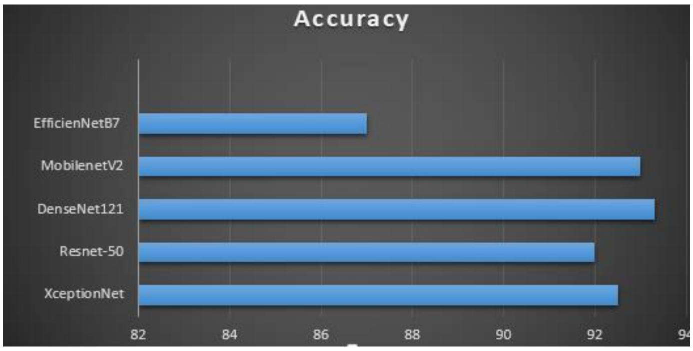


Fig. 9 Accuracy Comparison

B. Precision, Recall & F1-Score

True negatives and true positives show the recognition that is accurately predicted. The false-positive and false-negative are the opposite. So we want to restrict false positives and false negatives. The evaluation matrix formula is shown in Fig 10. These terms are somewhat confounding. So how about we take each term individually and comprehend it thoroughly.

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Fig. 10 Evaluation Matrix

- True Positive (T.P.):
  True positives indicate that the outcome was correctly predicted. Before we started modelling, the actual class's value was yes, and after we finished modelling, the class was expected to be yes. For example, if the actual class predicts that the picture belongs to the paper class, and we discover that the class is paper after prediction, we can consider it a true positive.

- True Negative (T.N.):
  True negative also denotes a correct prediction, but this time on the opposing side. That is to say, the actual value is no, and our forecasting model also predicts a negative value for the class. Then it is called True Negative. So, for example, if the actual class predicts that the picture is not a metal class and we discover that the class is not metal after prediction, it is known as a true negative.

- False Positives (F.P.):
  If the actual class is no, but the model predicts a yes, it is called a false positive. So, for example, if the actual class value indicates that the picture does not belong to the paper class, the predicted model indicates it.
- False Negative (F.N.):
  If the actual class is yes, but the model predicts a no, it is called False Negative. So, for example, if the image's actual class value indicates that it belongs to the paper class, our predicted mode means that it does not, the image does not belong to that class.

C. *Precision*

Precision is the percentage of all the positives to the true positives. Thus, precision gives a measure of the relevant data points. For example, we must not start handling e-waste that does not have glass waste attributes, but our model predicted it to have it. The comparisons of precision found in our research experiment are stated in Fig.11. The precision scores for various algorithms are listed in Table 1 below.

- Precision = actual positive / (true positive + false positive)

TABLE I
PRECISION TABLE

	cardboard	**E-waste**	**glass**	**metal**	**paper**	**plastic**	**trash**
Xception	0.99	1.00	0.92	0.93	0.82	0.94	0.87
Resnet-50	0.99	0.95	0.92	0.92	0.93	0.85	0.85
DenseNet121	0.95	0.98	0.91	0.91	0.92	0.98	0.81
MobileNetV2	0.91	0.94	0.94	0.95	0.90	0.83	0.68
EfficienNetB7	0.94	0.94	0.84	0.92	0.91	0.76	0.65

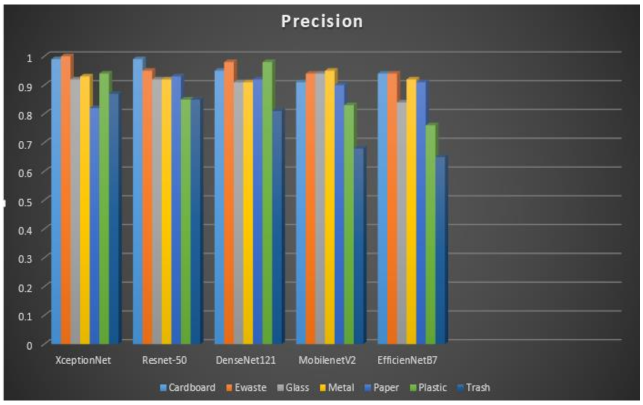


Fig. 11 Precision Comparison

D. *Recall*

- Recall = True positive (true positive + false negative)

The recall comparisons discovered in our study are stated in Fig.12. The recall scores for various algorithms are listed in Table 2 below.

TABLE II
RECALL TABLE

	cardboard	E-waste	glass	metal	paper	plastic	trash
Xception	0.90	0.98	0.95	0.90	0.98	0.85	0.71
Resnet-50	0.88	1.00	0.94	0.94	0.94	0.85	0.79
DenseNet121	0.93	1.00	0.96	0.98	0.92	0.81	0.93
MobileNetV2	0.89	1.00	0.87	0.89	0.93	0.89	0.61
EfficienNetB7	0.90	0.93	0.93	0.82	0.85	0.81	0.71

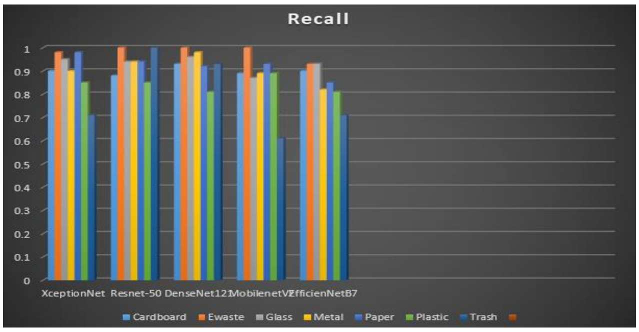


Fig. 12 Recall Comparison

E. F1-Score

F1-Score = (2* Precision * Recall) / (Precision +Recall)

There is a trade-off between accuracy and recall. This is easier to deal with as we can only strive for a good F1 score instead of balancing accuracy and recall and, it will also represent a good Precision and a good Recall value.

TABLE III

F1-SCORE TABLE

	cardboard	E-waste	glass	metal	paper	plastic	Trash
Xception	0.94	0.99	0.94	0.91	0.90	0.89	0.78
Resnet-50	0.93	0.98	0.93	0.93	0.93	0.85	0.81
DenseNet121	0.94	0.99	0.93	0.94	0.92	0.89	0.87
MobileNetV2	0.90	0.97	0.90	0.92	0.92	0.89	0.64
EfficienNetB7	0.92	0.93	0.88	0.79	0.86	0.79	0.68

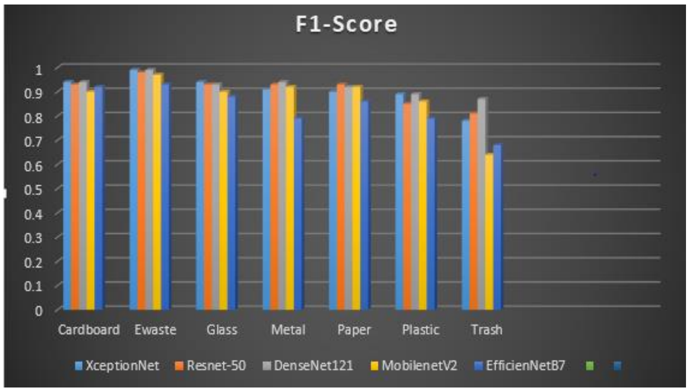
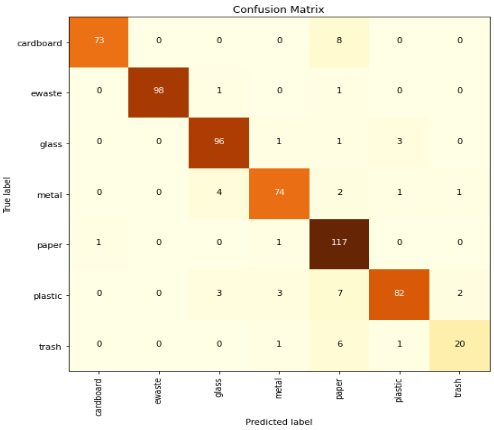


Fig. 13 F1-Score Comparison

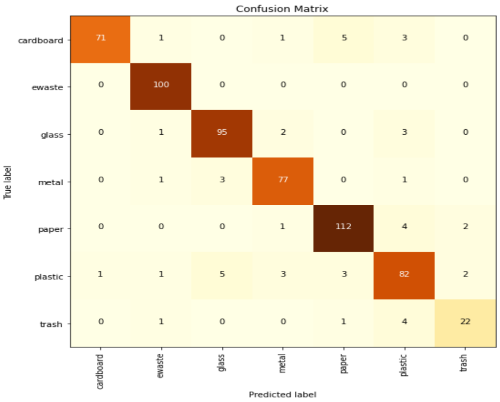
F. Confusion Matrix

The results of four different algorithms used to find the confusion matrix in our experiment are listed sequentially in Fig.14, Fig.15, Fig.16, Fig.17, and Fig.18.



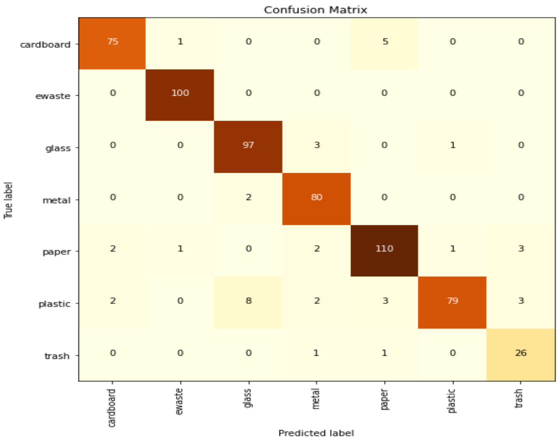
	cardboard	ewaste	glass	metal	paper	plastic	trash
cardboard	73	0	0	0	8	0	0
ewaste	0	98	1	0	1	0	0
glass	0	0	96	1	1	3	0
metal	0	0	4	74	2	1	1
paper	1	0	0	1	117	0	0
plastic	0	0	3	3	7	82	2
trash	0	0	0	1	6	1	20

Fig. 14 Confusion Matrix for Xception



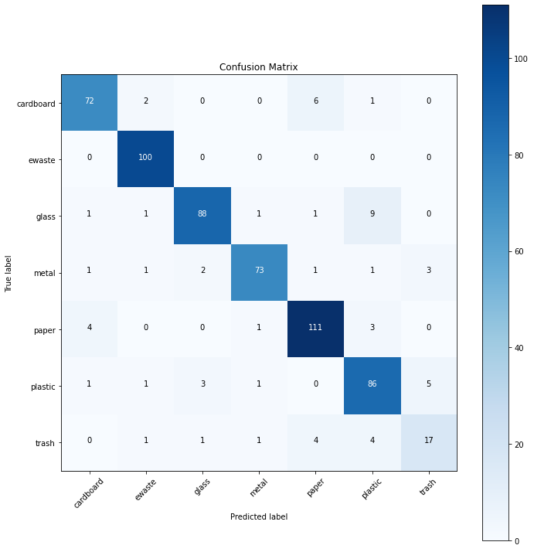
	cardboard	ewaste	glass	metal	paper	plastic	trash
cardboard	71	1	0	1	5	3	0
ewaste	0	100	0	0	0	0	0
glass	0	1	95	2	0	3	0
metal	0	1	3	77	0	1	0
paper	0	0	0	1	112	4	2
plastic	1	1	5	3	3	82	2
trash	0	1	0	0	1	4	22

Fig. 15 Confusion Matrix for Resnet-50



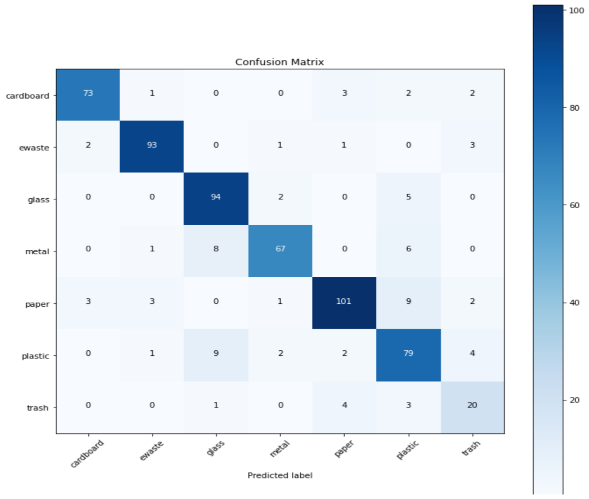
	cardboard	ewaste	glass	metal	paper	plastic	trash
cardboard	75	1	0	0	5	0	0
ewaste	0	100	0	0	0	0	0
glass	0	0	97	3	0	1	0
metal	0	0	2	80	0	0	0
paper	2	1	0	2	110	1	3
plastic	2	0	8	2	3	79	3
trash	0	0	0	1	1	0	26

Fig. 16 Confusion Matrix for DenseNEt121



	cardboard	ewaste	glass	metal	paper	plastic	trash
cardboard	72	2	0	0	6	1	0
ewaste	0	100	0	0	0	0	0
glass	1	1	88	1	1	9	0
metal	1	1	2	73	1	1	3
paper	4	0	0	1	111	3	0
plastic	1	1	3	1	0	86	5
trash	0	1	1	1	4	4	17

Fig. 17 Confusion Matrix for MobileNEtV2



	cardboard	ewaste	glass	metal	paper	plastic	trash
cardboard	73	1	0	0	3	2	2
ewaste	2	93	0	1	1	0	3
glass	0	0	94	2	0	5	0
metal	0	1	8	67	0	6	0
paper	3	3	0	1	101	9	2
plastic	0	1	9	2	2	79	4
trash	0	0	1	0	4	3	20

Fig. 18 Confusion Matrix for EfficienNetB7

From the overall results and analysis, we have found that all the transfer learning methods are working perfectly in the models and have given more than 90% accuracy except mobileNetv2. The precision and recall tables show the results, and the maximum results are more than 90 per cent. As a result, the accuracy is well suited for the models. DenseNet121 has given the best accuracy because of its architecture.

The runtime/execution time for the models is nearly 35 to 45 minutes, but the execution time depends on the architecture parameter. As the efficientNetB7 has 66658687 parameters, which is more than any other architecture, the runtime of the training part has taken much time. On the other hand, the testing runtime remains between 600-800ms/step. The mobilenetv2 with fewer parameters(3538984) has taken less time for the testing.

VII. Conclusion And Future Work

After the research and analysis, the findings, almost every model used in this study, DenseNet121 of transfer learning models outperform the performance and accuracy of all the experiments. After seeing the transfer learning system, it is observed they have done correctly to classify the waste images accurately. From the overall results and analysis, we have found that all the transfer learning methods are working perfectly in the models and have given more than 90% accuracy except mobileNetv2 because Mobilenetv2 is faster on mobile devices is slightly slower on desktop GPU.

Many significant experiments have been done in the waste management field. Still, the use of transfer learning with higher accuracy than normal deep learning algorithm is the main contribution from the authors, which creates a base platform for other researchers to continue their research with waste management field. Therefore, this research is very effective towards maintaining sustainable development goals.

This research can be extended in future with these experiments:

- Enhance the accuracy by using some other hyperparameter tuning for the transfer learning methods for future work.
- Mobile application development for the deployment of this project on mobile devices.
- By implementing real-time object detection methods such as using YOLO or similar approaches using microprocessor devices or cameras.
- Implementing dockers and Kubernetes will help the model deployment and scalability on other platforms.
- The research results can also be tested on an industrial level with relevant hardware to see if the models are working correctly or not.

- A Dashboarding system can also be made, and an automatic model retraining using re-enforcement learning can also be tried with this research output.

ACKNOWLEDGEMENT

This paper's authors are genuinely grateful to Kulliyyah of Information and Communication Technology, International Islamic University Malaysia, for their support and guidelines.

REFERENCES

[1] A. Silva and E. Soares, “Artificial intelligence in automated sorting in trash recycling,” XV Encontro Nac. Inteligência Artif. e Comput., 2018.

[2] Z. Lenkiewicz, "Waste and the Sustainable Development Goals." Retrieved from Wasteaid: https://wasteaid. org. uk/waste-sustainable …, 2018.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Commun. ACM, vol. 60, no. 6, pp. 84–90, 2017. [4] A. Silva and E. Soares, “Artificial intelligence in automated sorting in trash recycling,” *XV Encontro Nac. Inteligência Artif. e Comput.*, 2018.

[4] K. N. Sami, Z. M. A. Amin, and R. Hassan, "Waste Management Using Machine Learning and Deep Learning Algorithms," Int. J. Perceptive Cogn. Comput., vol. 6, no. 2, pp. 97–106, 2020.

[5] J. Zhao, W. Guo, S. Cui, Z. Zhang, and W. Yu, "Convolutional neural network for SAR image classification at patch level," in 2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), 2016, pp. 945–948.

[6] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 1251–1258.

[7] J. Carreira, H. Madeira, and J. G. Silva, "Xception: A technique for the experimental evaluation of dependability in modern computers," IEEE Trans. Softw. Eng., vol. 24, no. 2, pp. 125–136, 1998.

[8] N. S. Gupta, V. Deepthi, M. Kunnath, P. S. Rejeth, T. S. Badsha, and B. C. Nikhil, "Automatic Waste Segregation," in 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018, pp. 1688–1692.

[9] E. Rezende, G. Ruppert, T. Carvalho, F. Ramos, and P. De Geus, "Malicious software classification using transfer learning of resnet-50 deep neural network," in 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA), 2017, pp. 1011–1014.

[10] L. Wen, X. Li, and L. Gao, "A transfer convolutional neural network for fault diagnosis based on ResNet-50," Neural Comput. Appl., pp.

[11] G. E. Sakr, M. Mokbel, A. Darwich, M. N. Khneisser, and A. Hadi, "Comparing deep learning and support vector machines for autonomous waste sorting," in 2016 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET), 2016, pp. 207–212.

[12] D. Ezzat and H. A. Ella, "GSA-DenseNet121-COVID-19: a hybrid deep learning architecture for the diagnosis of COVID-19 disease based on gravitational search optimization algorithm," arXiv Prepr. arXiv2004.05084, 2020.

[13] D. Ezzat, A. E. Hassanien, and H. A. Ella, "An optimized deep learning architecture for the diagnosis of COVID-19 disease based on gravitational search optimization," Appl. Soft Comput., p. 106742, 2020.

[14] M. Yang and G. Thung, "Classification of trash for recyclability status," CS229 Proj. Rep., vol. 2016, 2016.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4510–4520.

[16] H. NGUYEN, "FAST OBJECT DETECTION FRAMEWORK BASED ON MOBILENETV2 ARCHITECTURE AND ENHANCED FEATURE PYRAMID," J. Theor. Appl. Inf. Technol., vol. 98, no. 05, 2020.

[17] S. Kalvankar, H. Pandit, and P. Parwate, "Galaxy Morphology Classification using EfficientNet Architectures," arXiv Prepr. arXiv2008.13611, 2020.

[18] I. Kagirov, D. Ryumin, and M. Železný, "Gesture-Based Intelligent User Interface for Control of an Assistive Mobile Information Robot," in International Conference on Interactive Collaborative Robotics, 2020, pp. 126–134.

[19] Y. Chu, C. Huang, X. Xie, B. Tan, S. Kamal, and X. Xiong, "Multilayer hybrid deep-learning method for waste classification and recycling," Comput. Intell. Neurosci., vol. 2018, 2018.7, no. 7, pp. 8489–8515, 2015.

[20] R. S. S. Devi, V. R. Vijaykumar, and M. Muthumeena, "Waste Segregation using Deep Learning Algorithm."

[21] X. Xu, X. Qi, and X. Diao, "Reach on Waste Classification and Identification by Transfer Learning and Lightweight Neural Network," 2020.