ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT851

Deep Learning for Sustainable E-Waste Management: Leveraging Transfer Learning for Classification

[1]Jyoti Kukade1[0009-0009-4633-0965]; [2]Ansh Kanungo2[0009-0001-8025-0389]; [3]Akshat Tamrakar3[0009-0004-7318-6288]
[1]Assistant Professor, [2,3]Student,
Medi-Caps University, Indore

Abstract:- The rapid rise of electronic waste (commonly referred to as "e-waste") has become a world's growing challenge which should be managed by creative approaches. The number of e-waste produced is estimated to be 53.6 million metric tons in 2019. From this we can see that the seriousness of the issue direly calls upon taking the measures to prevent the environmental and public health risks associated with this expanding crisis [1]. Since a lot of the e-waste may contain hazardous materials such as mercury, lead and cadmium, which can impact the health and the environment if not treated properly, the mismanagement of it increases the problem [2]. In the case of e-waste, there is wide assortment of the electronic devices and components hence, it becomes difficult to classify them into their product categories properly. Sorting processes can't keep up with the pace of production waste as a result of being tedious, error-prone, and slow. This research employs deep learning approaches to segregate E-waste items using images for automated category. Utilizing contemporary models like VGG16, DenseNet121, InceptionV3, MobileNetV3, and ResNet50, the research designs classification systems that have these great attributes. Dataset building (training and assessment) become easy when an extensive dataset of 3000 images from 10 different types of equipment is developed. This research study helps to offer useful implications for managing current methods of electronic waste disposal and developing sustainable circular economies with quantitative analyzing of model performance factors that include accuracy, precision, F1-score, mean squared error (MSE), and mean absolute error (MAE).

Keywords:- *E-Waste, Deep Learning, CNN, Pre-trained Models, Waste Classification, Machine Learning, Transfer Learning.*

I. INTRODUCTION

The situations in which the environment and the health of the public are greatly jeopardized by the "e-waste" which is the electronics junk produced by worldwide usage and demise of electronic equipment. As there tends to be a series of new technologies keeping on emerging at a rapid pace, there is always a risk of more electronics ending up in landfills, which is quite significant in terms of human health and the environment [4]. For instance, hazardous substances such as lead, mercury, cadmium, and brominated flame retardants may contain the essence of e-waste into water and soil sources and cause ecotoxicity and a risk to the health of populations exposed [5].

E-waste created nationwide from the twenty-one (21) forms of EEE specified under the E-Waste (Management) Rules, 2016, as of Financial Year (FY) 2017–18 is given below (Figure 1), based on statistics kept by CPCB, according to reports [6] from the India Press Information Bureau:

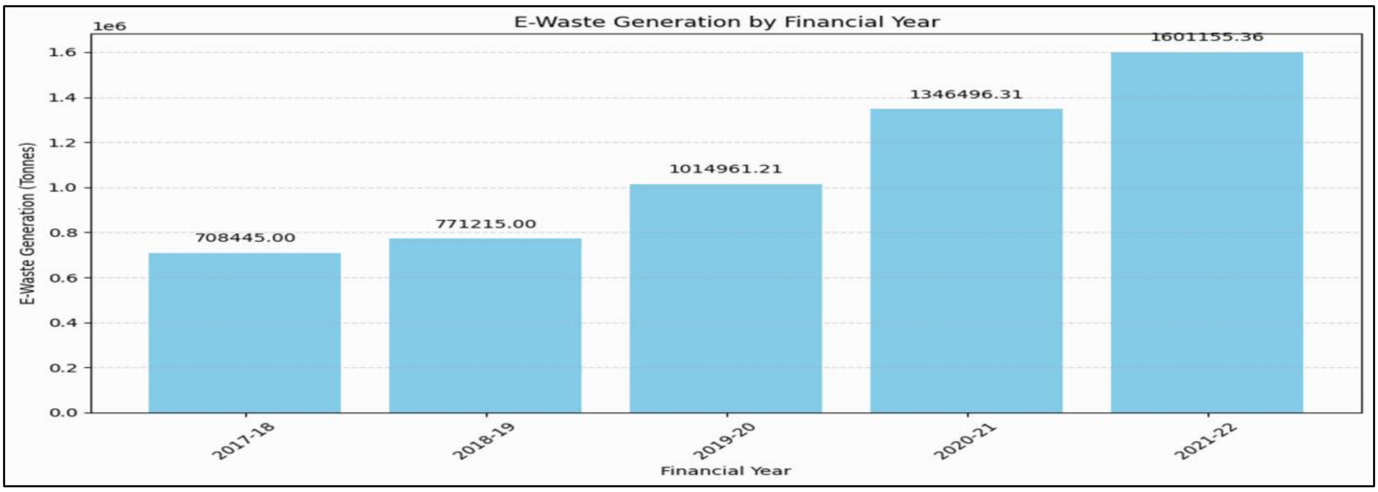

Fig 1: Yearly E-Waste Generation

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT851

Due to its varied composition and the inclusion of potentially harmful elements, managing e-waste poses difficult issues. Conventional techniques for classifying and disposing of e-waste sometimes entail labor-intensive, time consuming, and inaccurately prone manual sorting procedures. Furthermore, incorrect disposal techniques, such as the unofficial recycling procedures used in many areas, pollute the environment and endanger the health of those who handle e-waste [7].

In regard to these challenges, creative e-waste management techniques that can boost productivity, lower threats to the environment and public health, and speed up the classification process are becoming more and more important. Because deep learning techniques are capable of extracting characteristics from complex data, they have shown promise in solving e-waste classification difficulties [8]. Transfer learning, which is especially helpful for e-waste classification applications, uses pre-trained models on massive databases for adaptation to new tasks that require fewer annotated data [9].

The effectiveness of many well-known deep learning techniques, including VGG16, DenseNet121, InceptionV3, MobileNetV3, and ResNet50, for e-waste classification is investigated in this work. Evaluating these models' performance at two distinct image sizes (150x150 and 224x224 pixels) is the study's special objective. Choosing the appropriate image size is crucial since it significantly affects the performance of deep learning algorithms.

A dataset including pictures of 10 distinct electronic applications, developed in order to carry out this study. A total of 3000 photos makes up the dataset, with 300 images per application. The dataset's variety of electronic applications seeks to represent the heterogeneity found in actual e-waste situations, improving the findings' generalizability.

➢ *This Study Aims to Achieve Two Main Goals:*

- To evaluate, via transfer learning approaches, the classification performance of VGG16, DenseNet121, InceptionV3, MobileNetV3, and ResNet50 models on e-waste photos.
- To find the ideal picture size for e-waste classification tasks by comparing these models' performance across two distinct image sizes (150x150 and 224x224 pixels).

II. LITERATURE REVIEW

In the following literature review section, where we have meticulously compiled and analyzed the works of various authors who have published papers on the topic of e-waste, focusing specifically on algorithms and their corresponding results, providing a comprehensive analysis of how these works have contributed to our understanding and management of e-waste related challenges. Table 1 contains the work done in related domain.

Table 1: Related Work

S. No	Citation	Methodology	Results
1.	[10]	(CNN) Image recognition algorithm.	For 164 color images, the CNN model had a validation accuracy of 93.9% and a training accuracy of 96.9%.
2.	[11]	Faster region based convolutional neural network (RCNN) 210 Images	For classification, the CNN model had a 96.7% accuracy rate, and the RCNN model had a 90% accuracy rate, which was less than the best CNN.
3.	[12]	Neural networks with gradient boosting regression trees (GBRT, NN)	Machine learning computation increases the correctness of arrangements by 99.1% when employing the best approach.
4.	[13]	YOLOv5	After 500 epochs of training with 2400 e-waste pictures, the model generated a 0.352 mAP value.
5.	[14]	RESNET-18	The ability of the proposed research to combine IOT segregation with E-waste identification was demonstrated by the creation of a prototype. The average accuracy of this method is 93%.
6.	[15]	Convolution neural networks (CNN) and support vector machines (SVM)	The author proposed a non-standard method here using 600 images which yielded 82.2% and 79.4% accuracy for CNN and SVM respectively.
7.	[16]	YOLOv4, DSSD, and Faster-RCNN were merged into one network by MCCNN	30000 images from 52 various categories were used to build this model. The accuracy achieved using this strategy is 93% on average.
8.	[17]	FLANN-based object classification- IRD object recognition (DLSODC-GWM)	The model was intended to classify small garbage objects. The author used 2400 images of 6 different categories of Ewaste. DLSODC-GWM technique yielded 98.61% accuracy.
9.	[18]	YOLOv3	The dataset used contains close to 7800 images of 6 different categories of Organic and Inorganic waste. The accuracy of YOLOV3 reached 85.29%.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT851

10.	[19]	Resnet-50 serves as the foundation of the depth-wise separable convolution attention module (DSCAM) model.	After evaluating VGG16, Xception, MobileNet-v3, and GNet, the author found that 98.9% of DSCAM modules had the highest accuracy possible.

[20] Presented the YOLO-Green waste identification methodology. Seven of the most common kinds of solid waste were identified by the algorithm after it was trained on a dataset made up of real waste. After just 100 training epochs, YOLO-Green has an amazing map of 78.04%.

In [21], a method for identifying trash in aquatic habitats was proposed. The author used the YOLOV5 algorithm and the MobileNetV3 network for feature extraction. The Model was designed to distinguish between various trash cans in an aquatic environment.

In order to reduce monitoring costs and better match the information collected with the vital information needs of cities, [22] proposed a system for monitoring urban garbage. The scientists used a deep convolutional neural network model and vehicle-mounted cameras to assess the quantity of urban garbage that was collected along roadsides. Mask R-CNN emerged as the most successful algorithm, with 91% recall, 83% precision, and 77% accuracy, among the three trash recognition systems they tested.

The author's Retina-Net model, which was created in [23], is based on Resnet-50 and has an average precision of 0.814. Six different types of inorganic waste, including glass, paper, cardboard, plastic, metal, and trash, were represented by the author using 2527 images.

To identify and classify garbage and trash, [24] designed a semi-smart trash separation. Conductive metal is collected with the aid of magnetic separators, while non-conductive materials are then categorized based on their degree of hardness. By assigning each material a barcode or QR code, precycling processes made it possible to separate the materials according to the assigned code. 75% and 83% of the materials can be accurately detected, according to Alex-Net and Google-LeNet findings, respectively.

[25] has created and deployed an image-based detection system that is capable of differentiating between different trash cans for the purpose of classification.

[26] A smartphone software called SpotGarbage was created that leverages user-clicked, geo-tagged photographs to find trash in real time environments. Garbage-In-Images (GINI) was the dataset used, and it produced accuracy of 87.69%.

[27] employs a deep learning strategy to automatically detect waste. It was advised to use a data fusion and augmentation strategy together with the FastRCNN model to boost the method's precision. The investigations show that the approach has a high-precision detection function and good generalization properties.

In the study by [28], waste detection and recognition are carried out using an enhanced YOLOv3 network model. The dataset acquired for this purpose was used to fine-tune the network. The data suggest that the suggested strategy might significantly improve trash management in smart cities.

Using data from [29] research, a technique is suggested for locating trash that is clearly visible and floating on urban canals' water surfaces. The first of its type, the scientists also offer a sizable dataset with object-level annotations of waste in water channels. A brand-new attention layer is suggested to enhance the identification of tiny things.

Low image resolution prompted a study by [30] that resulted in the development of an algorithm that is an enhanced single-shot multibox detector (SSD): a brand-new, compact feature fusion module. This study switched the VGG16 backbone network to ResNet-101 in order to gain more accurate identification.

Using deep learning and narrowband IOT, [31] proposed an autonomous trash identification system. The system directly detects and recognizes decorative waste inside the embedded front-end monitoring module and manages thousands of monitoring front ends via the background server and narrow-band identification system should be developed. The Internet of Things. An improved YOLOV2 was used to identify the waste.

According to [32] research, a new, portable garbage Yolov5 algorithm is modified and used by the system. Additionally, the researchers developed two methods known as video backtracking and tracking object transmission. These strategies were provided by the researchers, along with a tracking system built on a kernelized correlation filter.

To visually recognize trash in realistic underwater situations, one study [33] investigated various deep-learning algorithms. It is intended to inspect, map, and remove garbage using autonomous underwater vehicles (AUVs). Using a sizable dataset of actual debris seen in open-water environments, the researchers train several convolutional neural network architectures for object detection. The dataset is made available to the public and annotated to ensure transparency and encourage more research in this field.

III. DATASET OVERVIEW

This research uses a dataset consisting of 3000 well selected images from various sources, categorized into ten distinct classes, some shown in Figure 2. These images are intended to support machine learning model development and evaluation for tasks involving object detection and classification, with a particular emphasis on consumer electronics and appliances. The dataset includes a broad range of electrical devices that are frequently utilized in daily life.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT851

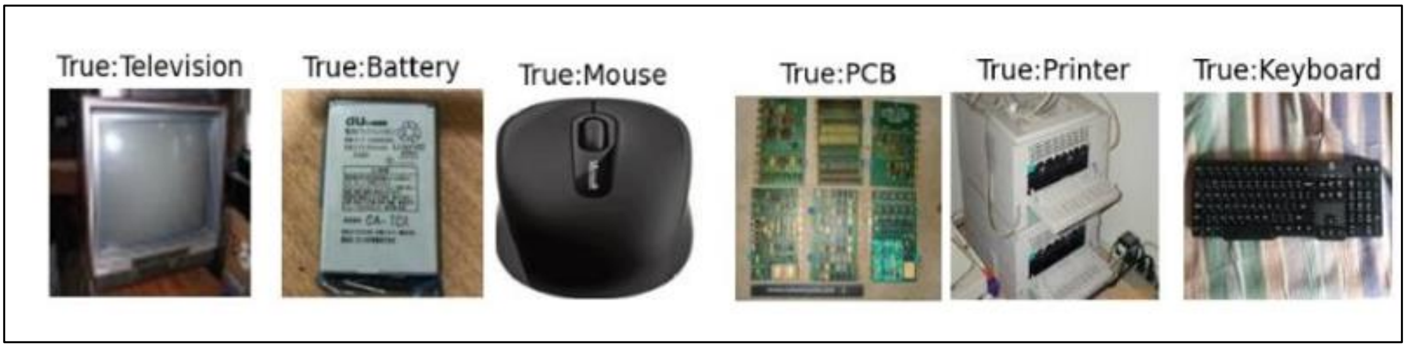

Fig 2: Dataset Images

A. Data Sources

The data was gathered from three sources (Images.cv, Kaggle, Web Scraping) to guarantee thorough coverage in all areas. Batteries, microwaves, televisions, washing machines, and certain printer pictures are among the photos that were only retrieved from the images.cv collection. For these particular categories, this provider offered a large selection of images. Images from the Keyboard, Mobile Phone, and Mouse classes were taken from the Kaggle dataset, which included a large number of imagery of gadgets. Furthermore, pictures for PCBs, Players, and Printers were carefully collected from the internet via web scraping methods, which made it possible to get a variety of images for these categories.

B. Data Subset

Three subsets of the dataset are separated out, each with a specific function in the creation and assessment of machine learning models. 2400 photos that have been evenly divided into the ten classes make up the training subset, which gives training models a starting point for identifying the characteristics and patterns connected to each class. With 300 images in the test subset, an objective evaluation of the model's performance is possible. In order to guarantee that the model is sensitive to fresh, testing data, samples from each of the ten classes are incorporated. Ultimately, the validation subset has 300 pictures that are evenly distributed over the 10 classes. Distribution of data is show in Figure 3. This allows for the optimization of hyper-parameters, which in turn prevents overfitting and improves the generalization of the model.

Fig 3: Distribution of Image Dataset

C. Data Augmentation

Through the data augmentation method, more training samples were generated to the end of modifying the images. Methods were made to enable rotation (up to 40°), shearing, shifting (up to 20% of image's size) and zooming (up to 20% zoom). To lift the patch size, the nearest pixel filling method was utilized. With these changes the data became more branched, followed the laws of nature, and that they avoided overfitting. Besides, the black-and-white image was normalized to [0, 1] white-black scale. In the training procedure batch were used with the image size of 128 and the shuffle was ensured to expose the model to different instances during each epoch. Resizing was done to adjust this to (224,224) and (150x150) pixels, to fit the input size expected by the pre-trained models. The generator for validation we built normalized the images in a way that was consistent with the training data, this normalization was done to make the model more consistent when training batch data, making it easy and just to evaluate model performance. The purpose of this careful preprocessing is to train the models on augmented and normalized data so that they will perform better and have higher accuracy in assigning the right waste electronic item classification to the input data.

Volume 9, Issue 10, October – 2024 International Journal of Innovative Science and Research Technology
ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT851

IV. EXPERIMENTAL SETUP

A. The Experimental Setup:

Cloud computing was used for the experiments that were executed in Google Colab environment to be able to run them properly and efficiently. The computing infrastructure here is made up of an Intel Xeon CPU with 2.00GHz clocks, consisting 2 physical cores, each with 2 threads, so totally there are 2 logical cores. In order to get all computationally challenging tasks carried out that include training deep learning models or processing large data in a short time the experimental setup made use of a NVIDIA Tesla T4 GPU of 15360 MB of dedicated GPU memory. This formidable GPU allowed to successfully execute efficient tasks in parallel, supporting arithmetic matrix operations as well as many other machine learning and data processing functions.

B. Experimental Design:

The historical impact of pre-trained CNN models used for image classification tasks were estimated by means of our experiment. The experimental design involved training and testing these models on two different image sizes: 150x150 and 224x224 images in size.

The dataset is imported into the program, which has images and target classes as input. In the preprocessing situation, images are processed by data augmentation methods. VGG16, Densenet121, Resnet50, InceptionV3 and MobilenetV3 are already pre-trained models. Their initial weights and the trained model are loaded. After that, a new global average pooling layer (CNN model) is created to train a new freeze of pre-trained layers with dataset. The CNN model is used by the classifier to obtain the prediction of the input images (YP). The training, validation (10%), and test (10%) data portions are 80%, 10% and 10%, respectively. The next stage is a dense layer which is fully-connected with 10 neurons (equal to the number of classes) and an activation function is added to the model. The weights of this output will be re-trained with the amount of data provided by our training and validation sets. The validation set is employed to gauge the model's performance and the training is stopped after 200 epochs. Eventually, the model performance is traced using the test set predicted labels (XS, YS). Figure 4 shows flow of steps followed for experiment. The specific details of the experimental setup are as follows:

- **Number of Epochs:** Two pre-trained CNN models were trained in twenty epochs for two hundred epochs to achieve a sufficient level of convergence. During the training, we observed how the model behaved over a period of time. Number of epochs was attained after trials and error and consulting some related works for image classification challenges.
- **Batch Sizes:** Concerning the training process, a batch size of 128 samples. This batch size was chosen to be appropriate in order to allow for the passage of full gradients during training while not wasting resources on oversized batches. This is the stage where validation and testing were done with the batch size set at 1 to teach us the specific output of a single sample and how accurate our statistics were.
- **Image Sizes:** We did some experiments with two image sizes scales, which aimed at the assessment of the model performance and the measuring of the impact of the input resolution on the classification accuracy. The two image sizes used were:
- **150x150 pixels:** We converted and retrained VGG16, DenseNet121, InceptionV3, and ResNet50 models on images each with dimension 150x150 using.
- **24x224 pixels:** In addition to the other models introduced here, we also used the MobileNetV3 model, which is a network that got trained on images with the resolution size of 224x224 pixels.

The image sizes of these were chosen according to the system resources and network capacity that was required by the pre-trained models. Via the years of an exercise on various sizes of the images we were able to determine the compromises between the complexity of the model, input resolution, and its performance.

During the implementation of our experiments, we kept data preprocessing consistent with procedures, for instance, normalization and augmentation techniques so that fair comparisons would be obtainable by the models implementing in various sizes of images. The evaluation metrics, comprising accuracy, precision, recall, and F1-score, along with the Mean Squared Error (MSE), Mean Absolute Error (MAE), and Root Mean Squared Error (RMSE), are the metrics used to assess the level of precision of the model's performance.

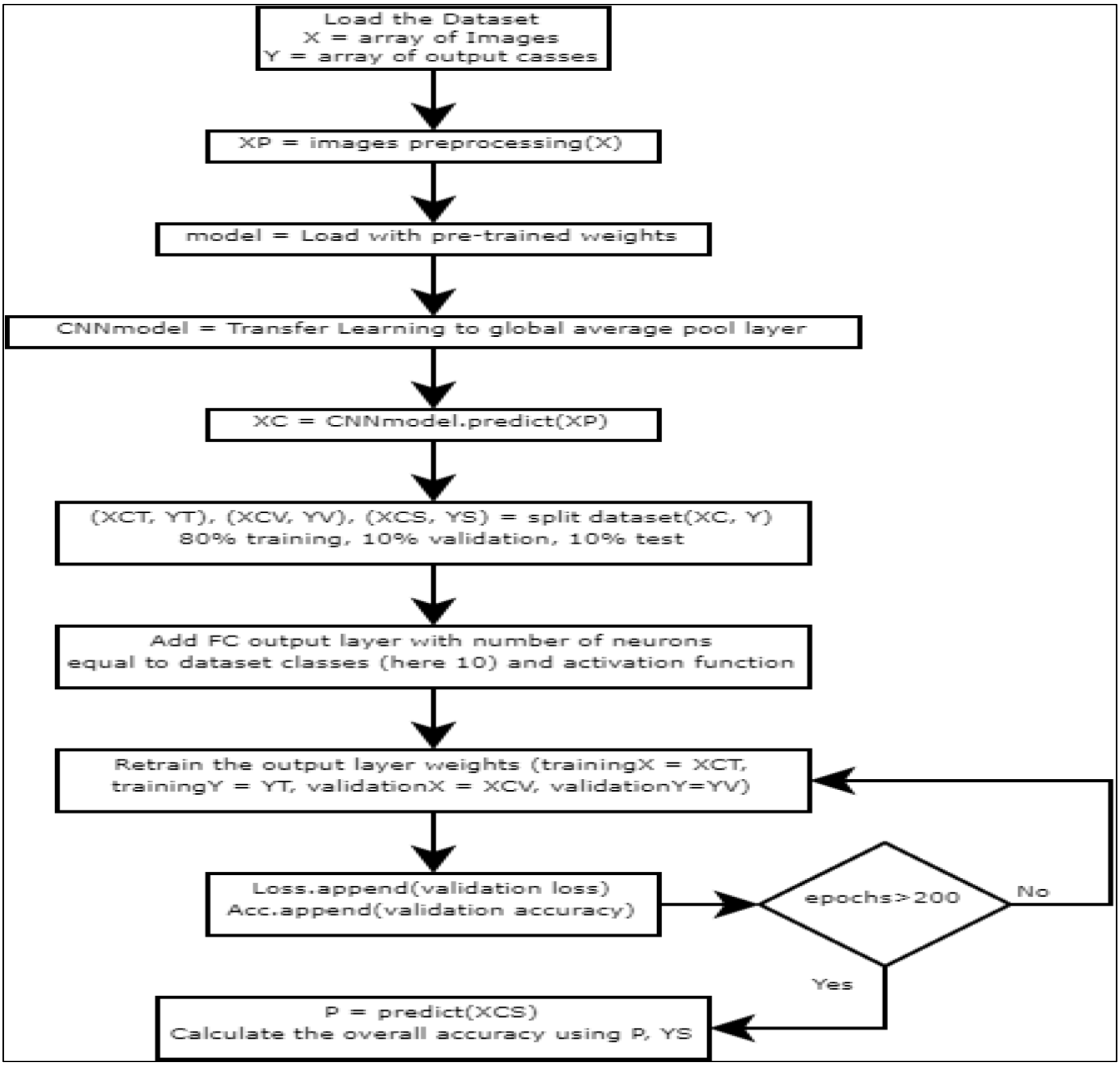

Fig 4: The Flow Chart of the Experimental Algorithms (Generalized)

V. THE METHODS EMPLOYED

In the following section, I go on to explore the models that are the subject of the study and explain their structures and characteristics. A brief snapshot of those classic networks which are used in this particular project will be showcased; the literature background for the same will be discussed in this context only. The last step is to elucidate the performance evaluation of the learning models, where the chosen methodology will be described.

A. The Convolution Neural Networks

The belief that Convolutional Neural Networks (CNNs) have become one of the most powerfully used classes of deep learning model for image analysis tasks, such as image classification, object detection, and semantic segmentation, is been strengthening day by day. Having a CNN, the main elements of its architecture as a combination of convolutional layers, pooling layers, and fully connected layers. The convolutional layers are the ones that extract local features out of the image by doing slide operation over the image with learnable filters, and performing element wise matrix multiplication and summations operation. It is set up to gain the particular characteristics or patterns that can be observed at different positions within the original image, for example, corners, lines, or textures. Figure 5 shows general structure of CNN.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT851

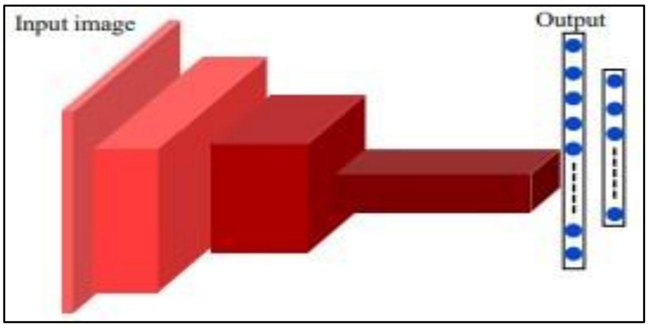

Fig 5: CNN Model Structure [34]

Mathematically, the convolution operation in a CNN can be expressed as:

$$(I \times K)(i, j) = \Sigma m\ \Sigma n\ I(i + m, j + n) \times K(m, n)$$

with where represents an input picture, K is the kernel or filter, i and j indicate the spatial positions that are considered as the grid, and m and n are dimensions or the size of the kernel. Convolution operation does an element - by - element multiplication between input image and kernel followed by summing up over all dimensions of the kernel. CNNs are especially suitable for image classification tasks because they can automatically build up complex and abstract features from low-level ones like edges and corners and eventually form the image hierarchical representation starting from the less complex to the more complicated details.

B. Vgg16

The convolutional neural networks (CNNs) have evoked a good impression on different visual tasks like image classification, and it is the VGG16 model that the two researchers (Simonyan and Zisserman, 2014) [35] designed which stands out for its profound structure and conceptual simplicity; and because of the features it captured from images, it has helped it perform well.

The structure of VGG16 comprises up to 16 weight layers where only 13 of those are convolutional and another 3 are fully-connected. Five blocks, each consisting of series of convolutional filters with 3×3 convolutional layers and max-pooling layers for spatial downsizing, are designed to make up the convolutional layers. The number of channels increases in every block by 2, starting with the first block which has 64 channels, and bumping up after the max-pooling layer until the last pooling layer which has 512 channels.

To execute VGG16 with our 10-class e-waste classification task, the fully connected finishing layer appended covering 128 units, followed by softmax output layer with 10 units conforming to our target classes is deployed (as shown in figure 6). While the VGG16 parameters were kept window-free, the layers newly introduced where for training. The noteworthy aspect of this method is the fact that it is capable of retaining the principal feature extraction abilities of the pre-trained model and at the same time using the later layers in order to perform the e-waste categorization task.

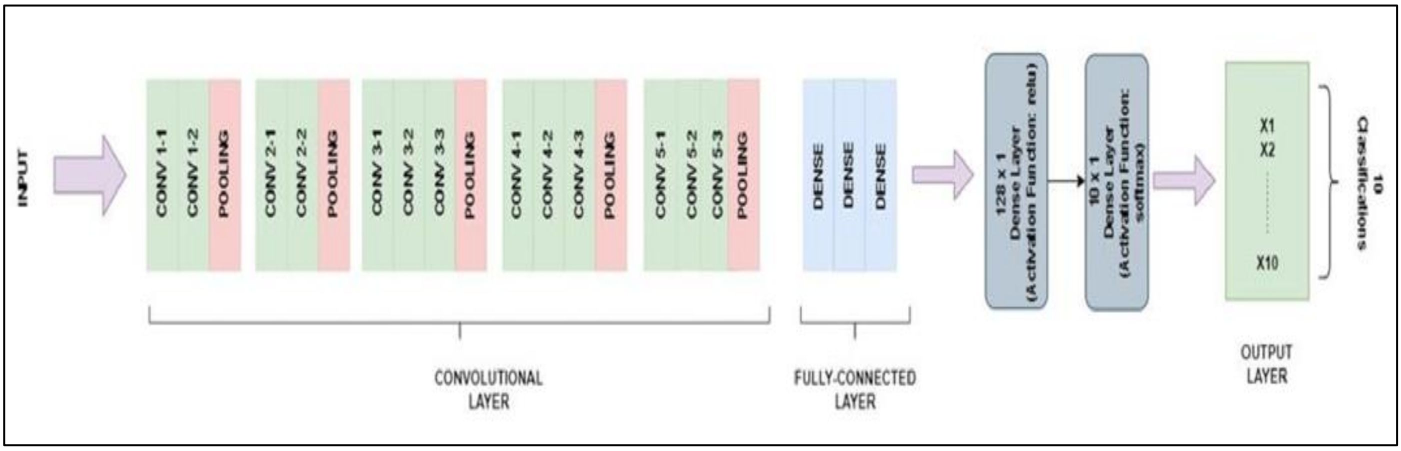

Fig 6: VGG16 Model Architecture

With the pre-trained layers being frozen, the model development spent on retrieving common generic image features (which include edges, texts and shapes) became easier as a result of the existing rich representation power of VGG16. At the same time, the trainable layers were learning the class-specific knowledge, which was in fact tuning the high-level representations to the feature of e-waste classification. This method so directly braids learning ability from a well-gained model with other methods and our specific model, and consequently, presents the efficient learning and improved generalization performance. Then the model trained and validated on same dataset for 200 epochs. Demonstrated training loss and accuracy for dimensions 150x150 and 224x224 in figures 7 and 8 respectively.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT851

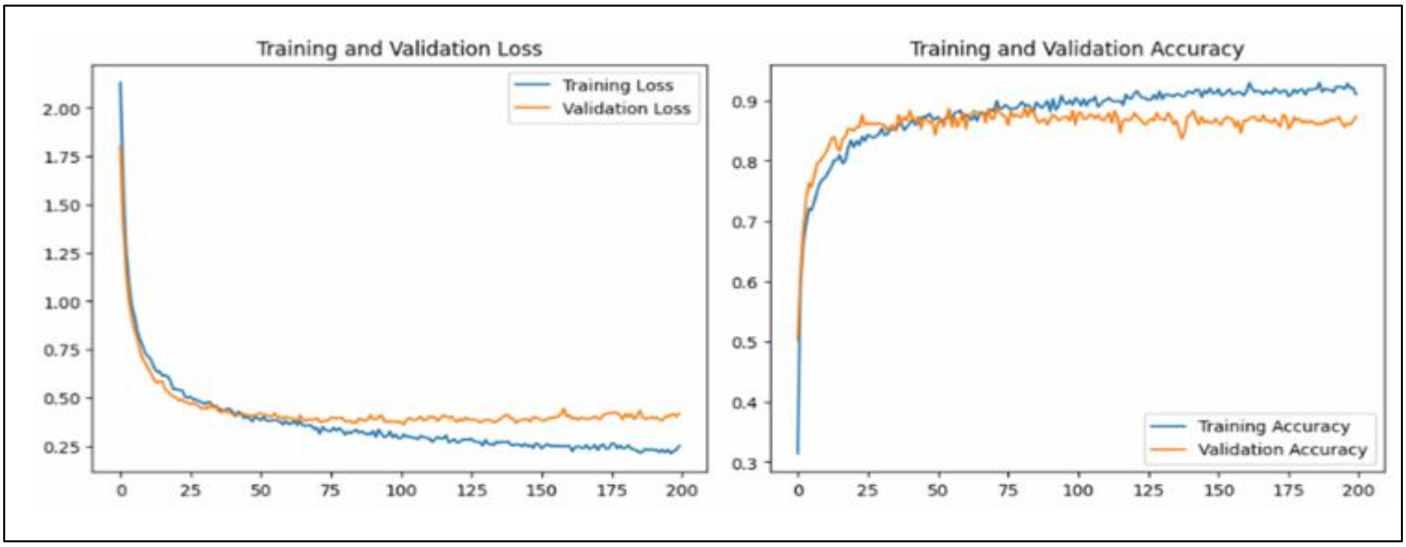

Fig 7: The Learning Curves Depicting Loss and Accuracy During the Training of the VGG16 Model on Image Size 150x150

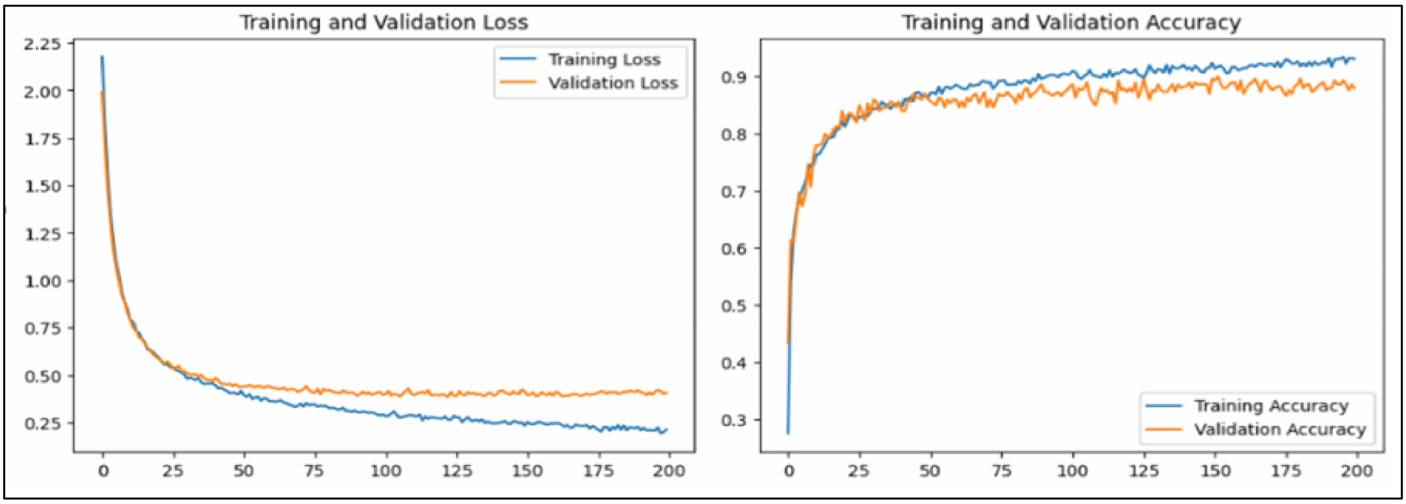

Fig 8: The Learning Curves Depicting Loss and Accuracy During the Training of the VGG16 Model on Image Size 224x224

C. Resnet50

Residual Networks (ResNets) has been an integral tool of deep learning technology to be applied to computer vision. ResNet50 architecture published by He et al. (2016) [36] is working amazingly fine in terms of capturing fine details from images. ResNet50 has 50 trainable weight layers comprising convolutional layers, batch normalization layers and residual sub-blocks as layers. Residual blocks are the cornerstone of architectures facilitating solutions to the vanishing gradient difficulty when deep neural networks are performed. These blocks apply the connectivity that lets the input flow through the convolutional layers, and this releases the gradients during backpropagation, so that the network can reap the maximum benefits.

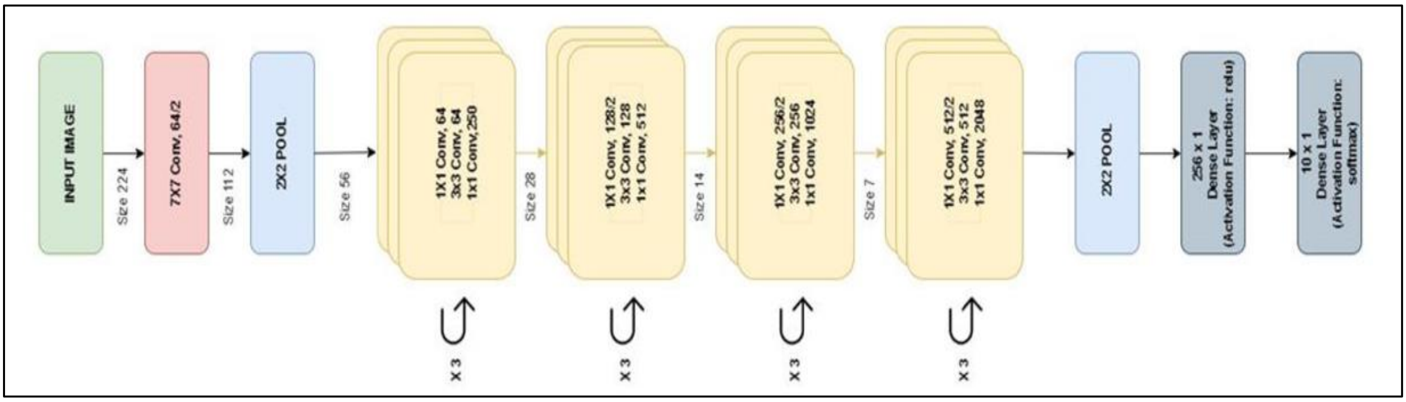

Fig 9: ResNet50 Model Architecture

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT851

We used ResNet50 in our work-flow since this network features a deep architectural structure and many powerful residual blocks, that can be trained for this particular purpose of 10-class e-waste classification (as shown in figure 9). To adopt ResNet50 for our task, we included a new fully connected layer with 256 neurons, the next that leads to a softmax output layer with 10 units, respectively assigned to our target categories. Fundamentally, the entire ResNet50 structure was optimized in a way that allowed fine-tuning and adaptation of the network to the structural and visual formats that of e-waste items. In making this decision the model has access to precise elements (of domain) and therefore, improve its performance and generalization ability. Then training with this configuration of model went for 200 epochs working with consistent datasets, taking care for 150x150 and 224x224 image sizes in figures 10 and 11. (The accuracy and loss training charts of the models are shown in figures 10 and 11 respectively).

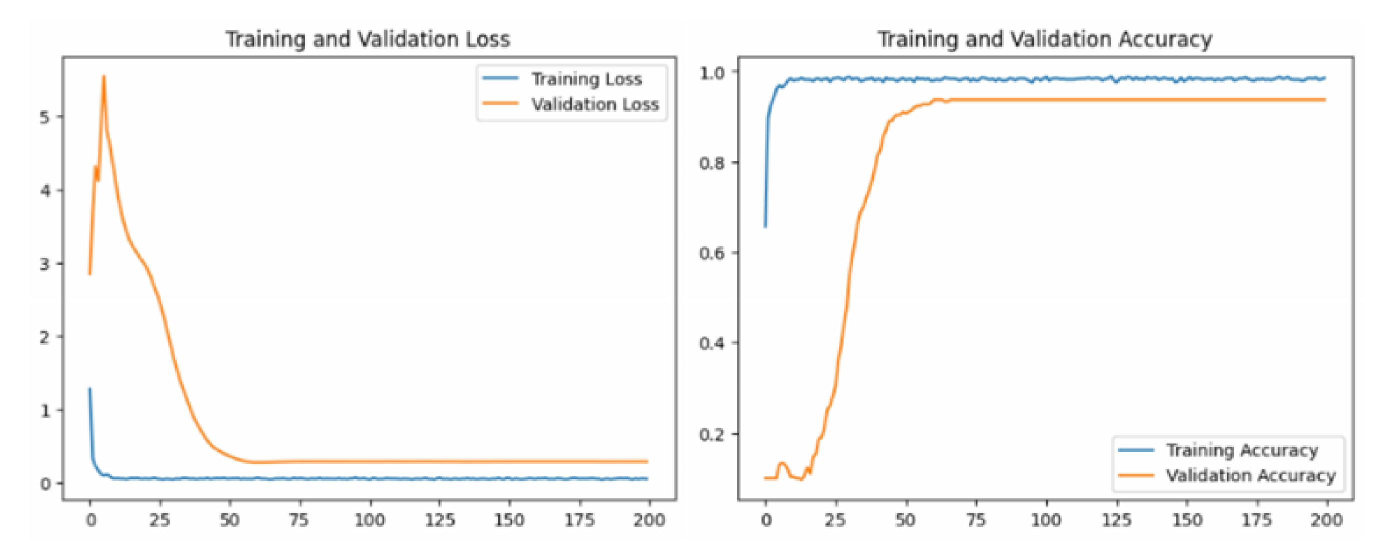

Fig 10: The Learning Curves Depicting Loss and Accuracy During the Training of the ResNet50 Model on Image Size 150x150

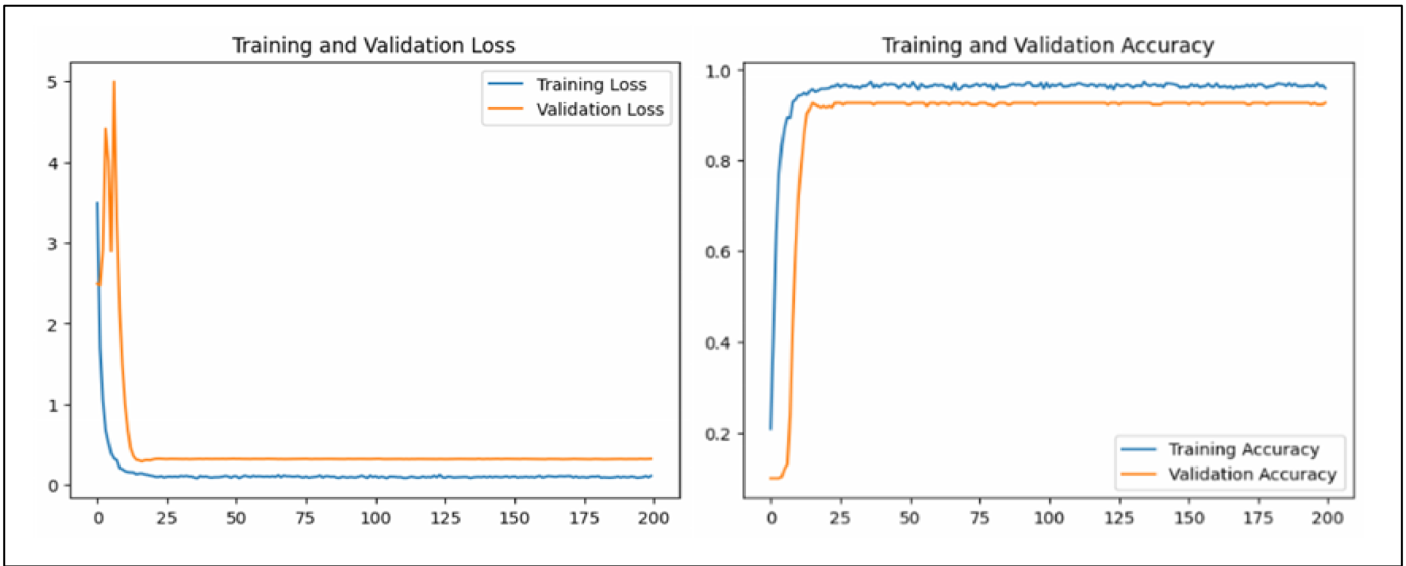

Fig 11: The Learning Curves Depicting Loss and Accuracy During the Training of the ResNet50 Model on Image Size 224x224

D. Densenet121

DenseNet-121, which is an architecture of the convolutional neural network introduced by Huang et al., (2017) [37], utilizes a certain patter which is known as dense connectivity in which, every layer is connected to every other layer in the feedforward fashion. By virtue of this connection with in-network pattern, the process of reusing the network input features becomes more reliable and thus the network is able to generate more accurate and informative representations of the most intricate features. The architecture of DenseNet121 possesses of these four dense blocks of which are made of the stacks of multiple convolutional layers and a positive activation function (batch normalization + ReLU). For all the follow-up layers, the feature maps that were created in previous segments are concatenated into a single input and extracted by the following layers (as shown in figure 12).

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT851

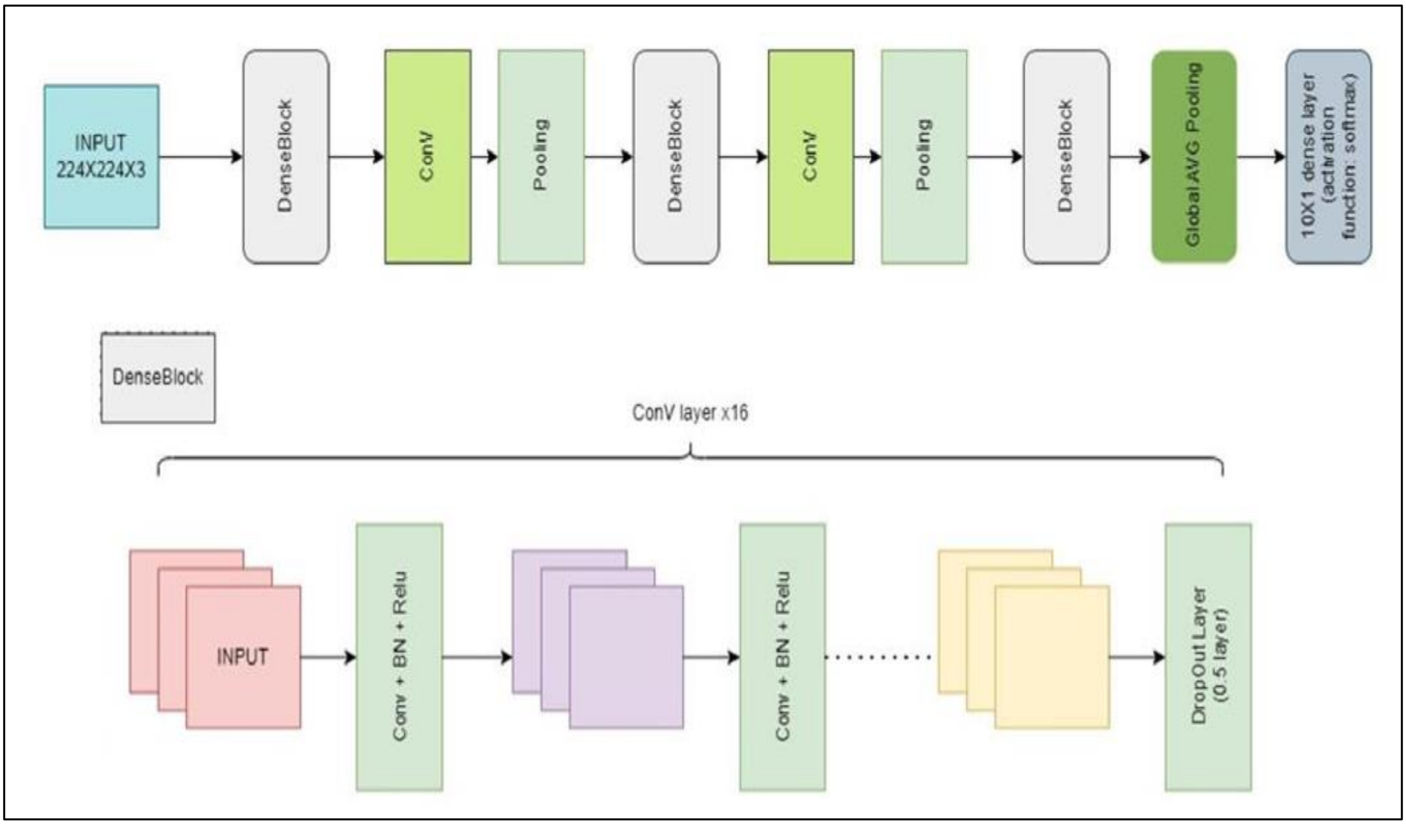

Fig 12: DenseNet121 Model Architecture

Our research took the advantage of high-efficiency DenseNet121 that was created to meet the challenges faced in classification and discrete form of e-waste. The structure utilizes normalization and dropout layers being used for model regularization and lowering the baseline of the curve. Moreover, we have grabbed a fully connected layer of 256 units which is reinforced with a softmax output layer with 10 units that matching our 10 sorting ewaste categories. Through its status quo, the model grasps the significant characteristics of e-waste objects alongside keeping a compact representation which makes it easy to interpret and leads in good classification results. The model was trained for 200 epochs using consistent dataset for image dimensions of 150x150 and 224x224, see accuracy and loss images in Figures 13 and 14.

Fig 13: The Learning Curves Depicting Loss and Accuracy during the Training of the DenseNet121 Model on Image Size 150x150

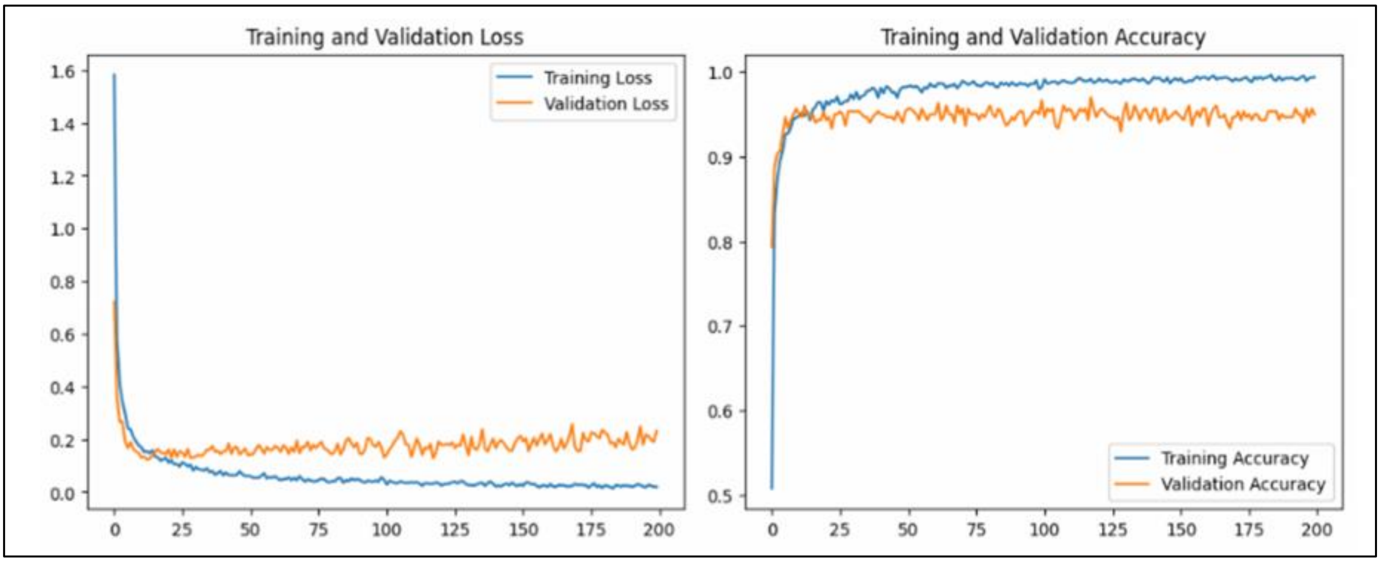

Fig 14: The Learning Curves Depicting Loss and Accuracy During the Training of the DenseNet121 Model on Image Size 224x224

E. Inceptionv3

The InceptionV3 architecture [38], which was put into use by Szegedy et al. (2016), is basically a convolutional neural network for image classification jobs that is not only efficient but also highly effective. It utilizes a newcomer building block called "the Inception module" which is a layer translating a convolution operation in parallel with different size of filter, hence capturing multi-scale features. Consequently, the concept of diversification of representations gets realized within the learning neural network from the input data in this type of structure. The InceptionV3's layered of inception modules is then interleaved by Pools layers or concatenation operations which forms a deep and complex structure.

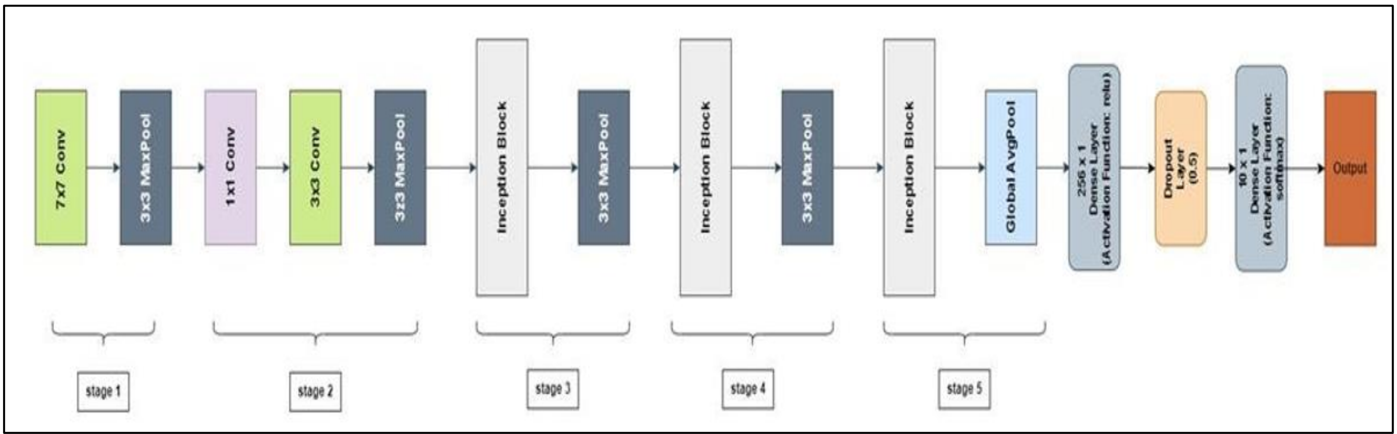

Fig 15: InceptionV3 Model Architecture

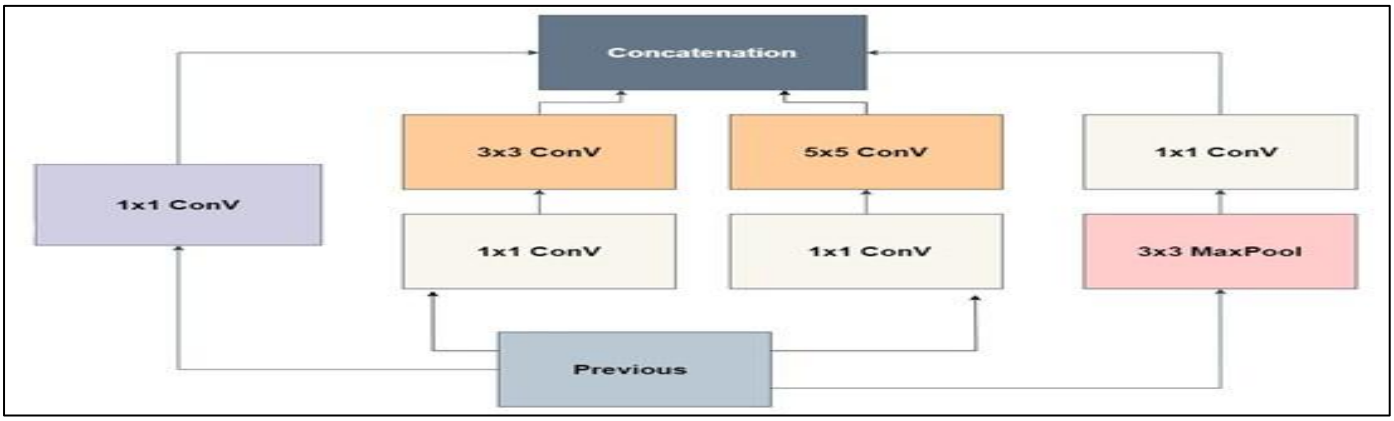

Fig 16: Inception Block Layers

In this study, the inception module of InceptionV3 is used and its capability to extract the features as well as the inception module is used to the classification of e - waste as a complex task. To modernize InceptionV3 for our task, we included a fully connected layer of 1024 units, and an output layer with 10 units that is generalized as softmax along with a 10-unit softmax output layer pertaining to different classes of e-waste. In its case, this modification made InceptionV3 distinguishable by providing exact classifications with around 10 e-waste categories (as shown in Figure 15 and Figure 16). The usage of a method where we froze an initial portion of layers and let perform training on later layers was another part of our strategy. Hence it is a middle way in making use of a transfer learning from a pre-trained model for the task of e-waste classification with the possibility of adapting the approach to the specific needs of e-waste classification. Choosing the right optimizer and setting the learning rate to maximize InceptionV3 recognition in the electronics waste classifying system, the model would optimize and enhance its performance. Then this model underwent training and validation over 200 epochs, utilizing data, while the 150x150 and 224x224 image sizes training are shown in figures 17 and 18, respectively; the accuracy and loss training graphs are depicted on the corresponding figures.

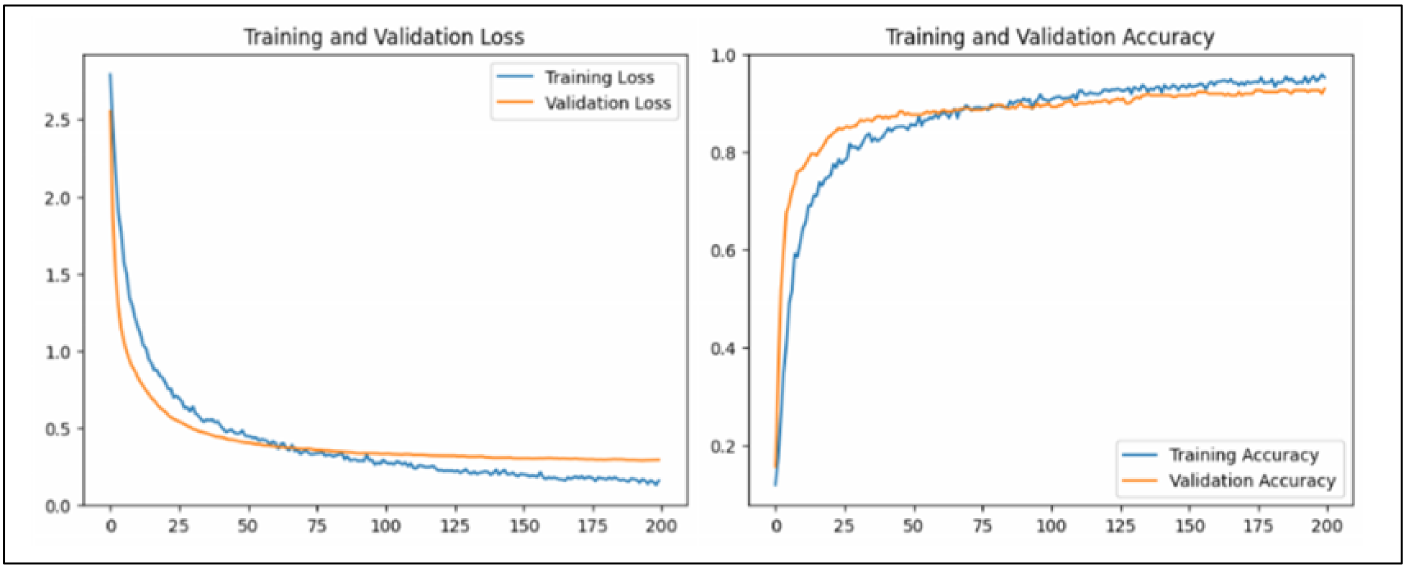

Fig 17: The learning curves Depicting Loss and Accuracy During the Training of the Inceptionv3 Model on Image Size 150x150

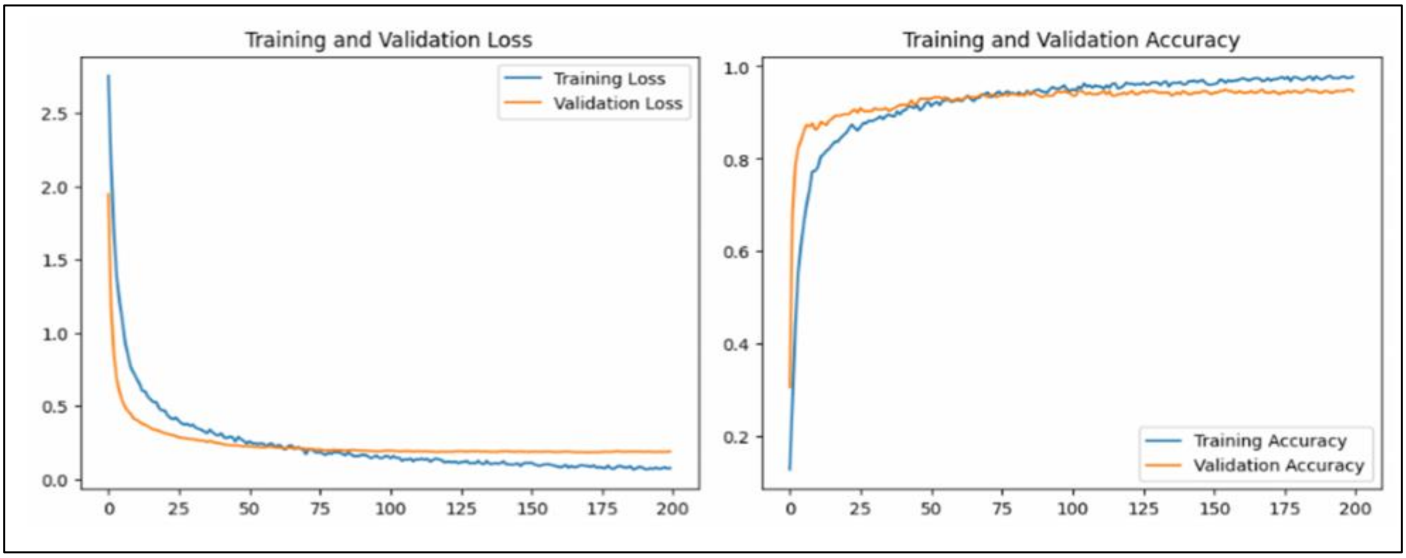

Fig 18: The Learning Curves Depicting Loss and Accuracy During the Training of the InceptionV3 Model on Image Size 224x224

F. Mobilenetv3

MobileNet presented by Howard et al., (2017) [39], is an architecture of convolutional cloud neural works built to perform efficiently on mobile and embedded devices. The architecture uses depth wise separable convolutions which perform depthwise convolution and pointwise convolution in place of single convection step. The reduced number of parameters together with the lesser computational complexity makes up a stock worth for MobileNet adoption by real-world applications who choose efficiency with inference.

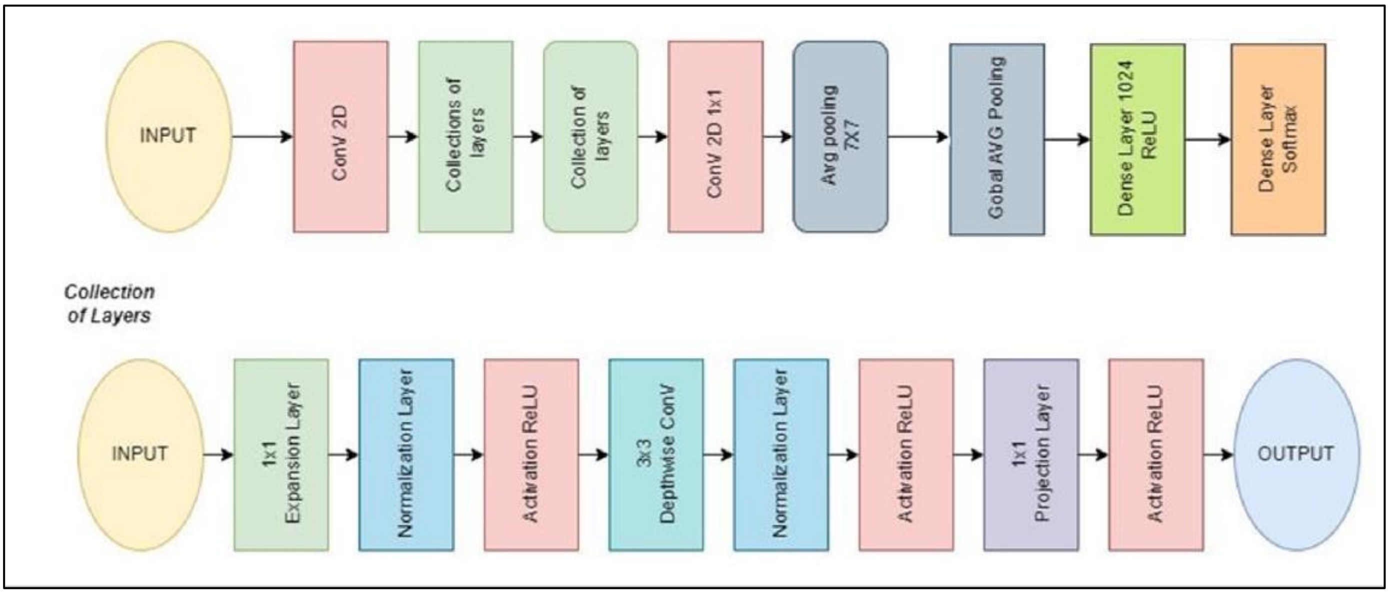

Fig 19: MobileNetV3 Model Architecture

In a study we exploited MobileNet's lighter architecture and the way it was using global average pooling, which are two of the best features of the model for real time applications like e-waste classification. In order to adopt Mobilenet for our purpose, we added dense layers with a final layer of 10 units that enabled the model to perform the function of multiple feature extraction and prediction of the probabilities of the 10 e-waste inventory. The 2 parts in the model that were omitted during the training adaptation process will be the pre-trained layers. This not only will allow the model to extract higher-order observations, but will also adequately adapt into the e-waste classification task. We also made the learning the rate and optimizer choice to serve as the contributors for better adaption to our particular problem domain. During 200 epochs of training together with the validation phase, which was aided by the use of datasets of 224x224 pixel dimensions, Figure 20, show the accuracy and loss training graphs.

Fig 20: The Learning Curves Depicting Loss and Accuracy During the Training of the MobileNetV3 Model on Image Size 224x224

G. Optimizer Employed

We used the Adam optimizer for VGG16, DenseNet121 and MobileNetV3, while the SGD optimizer was utilized for ResNet50, and InceptionV3. Each model was fine-tuned to adapt them to the specific classification task.

During training, the model parameters were updated using the Adam optimizer and SGD optimizer. According on the estimated first and second-order moments of the gradients, the Adam optimizer uses adaptive learning rates for each parameter. Adam's parameter update may be stated as:

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon}\hat{m}_t$$

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT851

Where θ_t represents the parameters at time step t, η is the learning rate, $\hat{m_t}$ is the estimate of the first-order moment (mean) of the gradients, $\hat{v_t}$ is the estimate of the second-order moment (uncentered variance) of the gradients, and ε is a small constant for numerical stability.

For the models trained with SGD, the parameter update is given by:

$$\theta_{t+1} = \theta_t - \eta \nabla L_i$$

Where ∇L_i represents the gradients of the loss function with respect to the model parameters.

H. Transfer Learning Paradigm: Adapting Pre-Trained Models for E-Waste Classification

Transfer learning is the practice of using knowledge gained from one activity to improve performance on another that is related but different. In our study, we used pre-trained models (M_{source}) that were trained on a source task, which may have been a totally unrelated categorization problem. High-level features and representations have been learned by these models from the original challenge. However, we aim to make use of this information to enhance the model's performance on a new objective job, namely the classification of e-waste products.

Let $f_{sourc}(x)$ be the output feature vector of the model M_{source} for an input image x, and let M_{source} represent a pre-trained model on a source task. The high-level representations that the model has learnt are captured in this feature vector. Thus, the following is a definition of the transfer learning process:

$$ftraget(x) = g(fsource(x))$$

Where $g(\cdot)$ is the adaptation function, which may be a shallow neural network or a linear transformation, and $f_{traget}(x)$ is the adapted feature representation for the target task.

VI. RESULTS

The study compared the performance of various deep learning models, including VGG, ResNet50, InceptionV3, DenseNet121, and MobileNetV3, on image classification tasks using two different input image sizes: images with sizes: 150x150 and 224x224. Alongside accuracy, precision, F1-score, mean-squared error (MSE), and mean absolute error (MAE) were used as the evaluation metrics.

Table 2: The Right (Appropriate) Image Size Selection Can Substantially Shape Model Output

Metric	Model	Image Size 150	Image Size 224
Accuracy	VGG16	0.877	0.853
	ResNet50	0.917	0.873
	InceptionV3	0.910	0.927
	DenseNet121	0.913	0.920
	MobileNetV3	N/A	0.897
Precision	VGG16	0.881	0.859
	ResNet50	0.919	0.878
	InceptionV3	0.913	0.928
	DenseNet121	0.915	0.924
	MobileNetV3	N/A	0.910
F1 Score	VGG16	0.878	0.853
	ResNet50	0.917	0.873
	InceptionV3	0.910	0.927
	DenseNet121	0.913	0.920
	MobileNetV3	N/A	0.895
MSE	VGG16	2.437	3.373
	ResNet50	1.760	3.067
	InceptionV3	1.760	1.673
	DenseNet121	1.810	1.873
	MobileNetV3	N/A	2.777
MAE	VGG16	0.490	0.620
	ResNet50	0.340	0.560
	InceptionV3	0.347	0.307
	DenseNet121	0.350	0.340
	MobileNetV3	N/A	0.477

The results show (in Table 2) that the right (appropriate) image size selection can substantially shape model output. While some models including Inception and DenseNet showed a little more accuracy as well as F scores for 224 x 224 big images, the others including VGG and ResNet achieved this much on the smaller 150 x 150 images. Thus, this result shows that the suited input image size can be

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT851

different from model architecture and task and it may vary from each other

It is important to note that this model, MobileNet was the only one evaluated on the image size of 224x224 was also demonstrated competitiveness compared other when it comes to size that was the same. It is therefore clear that a similar tendency is evident in a parallel case of light-weight architectures similar to the MobileNet which are used in the efficient image classification tasks. Especially in the resource challenged environments.

Regarding the error metric (MSE and MAE) a consistent trend was observed across all models: in case the large image size such as 224x224 was considered, the rates of errors were generally the higher ones while they were lower in case of the small image size such as 150x150. It is thus implied that while larger input images could have more contextual characteristics of the image, they may as well introduce additional noise or low-level information in the image and as a result increase the prediction errors.

VII. CONCLUSION

Through using pre-trained CNN models along with assessing their effectiveness at different images sizes, this study offers practical knowledge that may improve the way that electronic waste management practices are executed. The findings emphasize the adoption of transfer learning in conjunction with suitable model architectures and image resolutions in the process of e-waste classification; these strategies notably contribute to improved classification efficiency and accuracy in waste management processes and initiate the roadmaps to more sustainable circular economy initiatives.

REFERENCES

[1]. V. Forti, C. P. Balde, R. Kuehr, and G. Bel, "The Global E-waste Monitor 2020: Quantities, flows and the circular economy potential," 2020.

[2]. H. Robinson, "E-waste: an assessment of global production and environmental impacts," *Science of the total environment*, vol. 408, no. 2, pp. 183–191, 2009.

[3]. P. Baldé, F. Wang, R. Kuehr, and J. Huisman, "The global e-waste monitor–2014, United Nations University, IAS–SCYCLE, Bonn, Germany," *ISBN Print*, pp. 978–992, 2015.

[4]. B. H. Robinson, "E-waste: an assessment of global production and environmental impacts," *Science of the total environment*, vol. 408, no. 2, pp. 183–191, 2009.

[5]. K. Grant *et al.*, "Health consequences of exposure to e-waste: a systematic review," *Lancet Glob Health*, vol. 1, no. 6, pp. e350–e361, 2013.

[6]. "Generation of e-waste." May 2023. [Online]. Available: https://pib.gov.in/PressReleasePage.aspx?PRID=1943201

[7]. Huang, S. Wen, J. Li, Y. Zhong, Y. Zhao, and Y. Wu, "The human body burden of polybrominated diphenyl ethers and their relationships with thyroid hormones in the general population in Northern China," *Science of the total environment*, vol. 466, pp. 609–615, 2014.

[8]. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[9]. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?," *Adv Neural Inf Process Syst*, vol. 27, 2014.

[10]. P. Zhou, "Machine Learning For The Classification And Separation Of E-Waste," in *2022 IEEE MIT Undergraduate Research Technology Conference (URTC)*, 2022, pp. 1–5.

[11]. P. Nowakowski and T. Pamuła, "Application of deep learning object classifier to improve e-waste collection planning," *Waste Management*, vol. 109, pp. 1–9, 2020.

[12]. Li, Z. Jin, and S. Krishnamoorthy, "E-waste management using machine learning," in *Proceedings of the 6th International Conference on Big Data and Computing*, 2021, pp. 30–35.

[13]. R. Dassi, S. Kamal, and B. S. Babu, "E-waste detection and collection assistance system using yolov5," in *2021 IEEE International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*, 2021, pp. 1–6.

[14]. S. Elangovan, S. Sasikala, S. A. Kumar, M. Bharathi, E. N. Sangath, and T. Subashini, "A deep learning based multiclass segregation of e-waste using hardware software co-simulation," in *Journal of Physics: Conference Series*, 2021, p. 12039.

[15]. S. Bhandari, "Automatic waste sorting in industrial environments via machine learning approaches," 2020.

[16]. Li *et al.*, "Automatic detection and classification system of domestic waste via multimodel cascaded convolutional neural network," *IEEE Trans Industr Inform*, vol. 18, no. 1, pp. 163–173, 2021.

[17]. F. S. Alsubaei, F. N. Al-Wesabi, and A. M. Hilal, "Deep learning-based small object detection and classification model for garbage waste management in smart cities and iot environment," *Applied Sciences*, vol. 12, no. 5, p. 2281, 2022.

[18]. S. Kumar, D. Yadav, H. Gupta, O. P. Verma, I. A. Ansari, and C. W. Ahn, "A Novel YOLOv3 Algorithm-Based Deep Learning Approach for Waste Segregation: Towards Smart Waste Management. Electronics 2021, 10, 14." s Note: MDPI stays neu-tral with regard to jurisdictional claims in~…, 2020.

[19]. F. Liu, H. Xu, M. Qi, D. Liu, J. Wang, and J. Kong, "Depth-wise separable convolution attention module for garbage image classification," *Sustainability*, vol. 14, no. 5, p. 3099, 2022.

[20]. W. Lin, "Yolo-green: A real-time classification and object detection model optimized for waste management," in *2021 IEEE International Conference on Big Data (Big Data)*, 2021, pp. 51–57.

ISSN No:-2456-2165 https://doi.org/10.38124/ijisrt/IJISRT24OCT851

[21]. Wu, Y. Sun, T. Wang, and Y. Liu, "Underwater trash detection algorithm based on improved YOLOv5s," *J Real Time Image Process*, vol. 19, no. 5, pp. 911–920, 2022.

[22]. Conley, S. C. Zinn, T. Hanson, K. McDonald, N. Beck, and H. Wen, "Using a deep learning model to quantify trash accumulation for cleaner urban stormwater," *Comput Environ Urban Syst*, vol. 93, p. 101752, 2022.

[23]. Panwar *et al.*, "AquaVision: Automating the detection of waste in water bodies using deep transfer learning," *Case Studies in Chemical and Environmental Engineering*, vol. 2, p. 100026, 2020.

[24]. Alzyoud, W. Maqableh, and F. Al Shrouf, "A semi smart adaptive approach for trash classification," *INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL*, vol. 16, no. 4, 2021.

[25]. Y. Yu, "A computer vision based detection system for trash bins identification during trash classification," in *Journal of Physics: Conference Series*, 2020, p. 12015.

[26]. Mittal, K. B. Yagnik, M. Garg, and N. C. Krishnan, "Spotgarbage: smartphone app to detect garbage using deep learning," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2016, pp. 940–945.

[27]. Y. Wang and X. Zhang, "Autonomous garbage detection for intelligent urban management," in *MATEC Web of Conferences*, 2018, p. 1056.

[28]. B. De Carolis, F. Ladogana, and N. Macchiarulo, "Yolo trashnet: Garbage detection in video streams," in *2020 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS)*, 2020, pp. 1–7.

[29]. M. Tharani, A. W. Amin, M. Maaz, and M. Taj, "Attention neural network for trash detection on water channels," *arXiv preprint arXiv:2007.04639*, 2020.

[30]. W. Ma, X. Wang, and J. Yu, "A lightweight feature fusion single shot multibox detector for garbage detection," *IEEE Access*, vol. 8, pp. 188577–188586, 2020.

[31]. Y. Liu, Z. Ge, G. Lv, and S. Wang, "Research on automatic garbage detection system based on deep learning and narrowband internet of things," in *Journal of Physics: Conference Series*, 2018, p. 12032.

[32]. Z. Yu, J. Liu, and X. Li, "LTDTS: A lightweight trash detecting and tracking system," in *International Conference on Adaptive and Intelligent Systems*, 2022, pp. 240–250.

[33]. M. Fulton, J. Hong, M. J. Islam, and J. Sattar, "Robotic detection of marine litter using deep visual detection models," in *2019 international conference on robotics and automation (ICRA)*, 2019, pp. 5752– 5758.

[34]. Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, "Deep feature extraction and classification of hyperspectral images based on convolutional neural networks," *IEEE transactions on geoscience and remote sensing*, vol. 54, no. 10, pp. 6232–6251, 2016.

[35]. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[36]. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[37]. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[38]. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.

[39]. G. Howard *et al.*, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.