1. INTRODUCTION

1.1 Overview of Waste Management

Efficient waste management is a critical component of sustainable environmental practices. As global populations grow and urbanize, the volume of waste generated increases, posing significant challenges for effective waste disposal and recycling systems. Traditional methods of waste sorting, which predominantly rely on manual labor, are often inefficient, time-consuming, and prone to human error. This inefficiency can impede recycling efforts, leading to higher rates of landfill use and environmental degradation. The advent of computer vision technology offers a transformative potential for automating waste classification, thereby enhancing the efficiency and accuracy of recycling processes. By leveraging advanced image recognition capabilities, automated systems can categorize waste more effectively than manual sorting. This technological innovation not only reduces the burden on human workers but also optimizes the sorting process, facilitating higher recycling rates and contributing to environmental sustainability. In this project, we explore the application of deep learning models to the problem of automated waste classification. Utilizing the TrashNet dataset, which comprises images of waste items categorized into six distinct classes—Cardboard, Glass, Metal, Paper, Plastic, and Trash/Non-Recyclable—we implement and compare the performance of four state-of-the-art deep learning architectures: DenseNet121, VGG16, and MobileNetV3.

The comparative analysis of these models is crucial for determining the most effective architecture for waste classification. By assessing various performance metrics, including accuracy, precision, and recall, we aim to identify the strengths and limitations of each model in the context of waste sorting. This evaluation not only provides insights into the optimization of trash classification methods but also underscores the broader implications for sustainable waste management practices and environmental conservation. Our findings from this study are expected to contribute valuable knowledge to the field of waste management, demonstrating the feasibility and benefits of integrating computer vision technology into recycling systems. Ultimately, this research supports the development of more efficient, reliable, and scalable waste sorting solutions, which are essential for mitigating environmental impact and promoting sustainability in modern society.

1.2 Major Challenges

Improper and inefficient waste management poses several significant challenges, which can have wide-ranging environmental, health, social, and economic impacts. Here are the major challenges:

Environmental Challenges:

1. Pollution:

- **Air Pollution**: Burning waste releases harmful pollutants, including dioxins, furans, and particulate matter, which contribute to air quality degradation and respiratory issues.
- **Water Pollution**: Improper disposal of waste can lead to leachate formation, which can contaminate surface and groundwater with hazardous chemicals and pathogens.
- **Soil Pollution:** Hazardous substances from improperly managed waste can leach into the soil, affecting its quality and harming plant life.

2. Habitat Destruction:

- Landfills and illegal dumping sites can lead to the destruction of natural habitats, threatening biodiversity and disrupting ecosystems.

3. Marine Pollution:

- Improperly managed plastic waste often ends up in oceans, leading to marine pollution that harms marine life through ingestion and entanglement, and disrupts marine ecosystems.

Health Challenges:

1. Disease Spread:

- Accumulated waste can become a breeding ground for disease vectors such as rodents, insects, and other pests, increasing the risk of diseases like dengue, malaria, and cholera.

2. Toxic Exposure:

- Exposure to hazardous waste, including electronic waste (e-waste) and industrial chemicals, can lead to serious health issues, including cancers, respiratory diseases, and reproductive problems.

3. Food and Water Contamination:

- Contaminants from waste can enter the food chain through crops grown in polluted soil and water supplies, leading to foodborne illnesses and long-term health effects.

1.3 Deep Learning Techniques

Deep learning, a subset of artificial intelligence (AI), has the potential to significantly address and mitigate the challenges posed by improper and inefficient waste management. Deep learning offers numerous avenues to enhance various facets of waste management, from automated sorting and collection optimization to illegal dumping detection and environmental monitoring. By integrating deep learning technologies, waste management systems can become more efficient, accurate, and responsive, ultimately contributing to more sustainable and environmentally friendly waste handling practices. This not only helps in mitigating the adverse effects of waste on the environment but also enhances public health and resource efficiency.

Automated Sorting Systems:

- Image Recognition: Deep learning models such as Convolutional Neural Networks (CNNs) can be trained on datasets (like TrashNet) to accurately classify different types of waste (e.g., plastic, metal, glass, paper, cardboard, non-recyclable trash). These models can be integrated into automated sorting systems in recycling facilities, significantly improving the efficiency and accuracy of waste sorting.
- Robotic Sorting: Robotics equipped with deep learning algorithms can sort waste on conveyor belts in real-time, reducing the reliance on manual labor and increasing sorting speed.

Waste Collection Optimization:

- Predictive Analytics: Deep learning models can predict waste generation patterns based on historical data, helping to optimize collection routes and schedules. This reduces fuel consumption and operational costs.
- Dynamic Routing: Real-time data from sensors and cameras on waste bins can be fed into deep learning models to dynamically adjust collection routes, ensuring timely waste collection and preventing overflow.

2. RELATED WORK

The primary objective of enhancing waste management systems is to address critical issues such as inefficiency in sorting, collection, and recycling processes, as well as to mitigate environmental and health impacts. Consequently, numerous conventional waste management techniques and advanced deep learning algorithms are employed to optimize these processes. The diverse approaches include automated sorting systems, predictive analytics for waste collection, surveillance for illegal dumping detection, and recycling process improvements. This module encompasses a comprehensive literature survey of all pertinent references consulted to propose this enhanced waste management model. Additionally, it features a comparison table that juxtaposes the various methodologies, evaluation metrics, advantages, limitations, and gaps identified in the reviewed studies. This structured analysis provides a robust foundation for developing a more efficient, accurate, and sustainable waste management system.

2.1 Literature Survey

[1] Shi, C., Tan, C., Wang, T., & Wang, L. (2021). A waste classification method based on a multilayer hybrid convolution neural network. Applied Sciences, 11(18), 8572.

The objectives of the paper are focused on improving waste classification accuracy using a multilayer hybrid convolution neural network and evaluating the performance of the proposed method on the TrashNet dataset.The proposed method involves preprocessing waste images, extracting image features, normalizing the features, and using the Softmax classifier for classification.The MLH-CNN method provides good feature extraction ability, focusing on the main target and effectively extracting features, leading to improved classification performance.

[2] Melinte, D. O., Travediu, A. M., & Dumitriu, D. N. (2020). Deep convolutional neural networks object detector for real-time waste identification. Applied Sciences, 10(20), 7301.

The study aims to compare the performance of different CNN architectures trained on the TrashNet dataset and evaluate their precision, recall, and F1 score. The focus is on developing an accurate and fast CNN architecture for waste detection, which is crucial for applications like waste collection by autonomous robots.The paper also discusses the optimization of learning rate during training and the use of different loss optimization methods.

[3] Kang, B., & Jeong, C. S. (2023). ARTD-Net: Anchor-Free Based Recyclable Trash Detection Net Using Edgeless Module. Sensors, 23(6), 2907.

The paper discusses the need for automatic systems for separate waste collection using deep learning and computer vision techniques.It proposes two anchor-free-based recyclable trash detection networks (ARTD-Net1 and ARTD-Net2) that efficiently recognize overlapped multiple wastes of different types.The paper concludes that the proposed ARTD-Net1 and ARTD-Net2 methods achieve competitive performance in mean average precision and F1 score compared to other deep learning models..

[4] Yang, J., Zeng, Z., Wang, K., Zou, H., & Xie, L. (2021). GarbageNet: a unified learning framework for robust garbage classification. IEEE Transactions on Artificial Intelligence, 2(4), 372-380.

The paper presents a novel incremental learning framework called GarbageNet for garbage classification, addressing challenges such as lack of data, high cost of category increment, and noisy data quality**.** The paper contributes to the field of AI for the environment, promoting environmental ethics, rotation economy, and relieving the pressure of consumption doctrine in smart cities.The GarbageNet framework utilizes weakly-supervised transfer learning for feature extraction, embedding new categories as anchors for reference, and classifying test samples by finding their nearest neighbors in the latent space.

[5] Chen, Z., Yang, J., Chen, L., & Jiao, H. (2022). Garbage classification system based on improved ShuffleNet v2. Resources, Conservation and Recycling, 178, 106090..

The paper introduces the use of deep learning technology for garbage classification and mentions previous studies that have used deep learning algorithms for this purpose.The self-built garbage image dataset used in the study consists of four categories of household garbage: recyclable garbage, wet garbage, hazardous garbage, and dry garbage.The paper presents the improvements made to ShuffleNet v2, including the parallel mixed attention mechanism (PMAM), the use of FReLU activation function, and transfer learning.

[6] Wahyutama, Aria Bisma, and Mintae Hwang. "YOLO-based object detection for separate collection of recyclables and capacity monitoring of trash bins." Electronics 11.9 (2022): 1323

This paper researches on the trash and recycled material identification using Alexnet CNN and making the robot applicationThe research is tested on two different ways one is identifying the indoor images and other is detecting the outdoor imagesThe author Integrating this image processing-based classification into smart trash cans will be more suitable for cleaning garbageThe robot detect the outdoor images and classify it to take the object or not(two classes take or non take)This paper achieved the accuracy of 92% on the trash net dataset and 93.6% on the outdoor images by the Alexnet CNN.

[7] Sultana, Rumana, et al. "Trash and recycled material identification using convolutional neural networks (CNN)." 2020 SoutheastCon. IEEE, 2020.

The paper focus on a incremental learning framework called GarbageNet for garbage classification, addressing challenges such as lack of data, new categories, and noisy data quality. The author proposed the incremental learning method for future adding components, the model should learn and classify newly builted objects In this article the AFM(attentive feature mixup) is used to leverage the noisy garbage data So, it can classify the objects in the different classes The proposed method achieved state-of-the-art performance in terms of accuracy, robustness, and extendibility, winning the first place in the HUAWEI Cloud Garbage Classification Challenge in 2019.

[8] Yang, Jianfei, et al. "GarbageNet: a unified learning framework for robust garbage classification." IEEE Transactions on Artificial Intelligence 2.4 (2021): 372-380.

The paper describes the development of a smart trash bin that uses a webcam and YOLO real-time object detection to separate and collect recyclables into their correct categories. The YOLO model achieved an accuracy of 91% under optimal computing conditions and 75% when deployed on a Raspberry Pi .The performance of the YOLO model was evaluated using the mAP measurement method, which assesses the average accuracy of object classification, box drawing, and the model's confidence in generating predictions .The system also incorporates hardware such as ultrasonic sensors for measuring trash bin capacity and GPS for locating trash bin coordinates. This information is uploaded to Firebase Database via the ESP8266 Wi-Fi module and displayed on a mobile application in real-time .The study aims to solve the recyclable waste separation problem in rural areas.

[9] Teng, X., Fei, Y., He, K., & Lu, L. (2022, July). The Object Detection of Underwater Garbage with an Improved YOLOv5 Algorithm. In Proceedings of the 2022 International Conference on Pattern Recognition and Intelligent Systems (pp. 55-60).

The paper proposes the use of YOLOv5 as the object detection algorithm for detecting and clearing underwater garbage using Autonomous Underwater Vehicles (AUVs).The paper introduces improvements to the YOLOv5 algorithm, including re clustered anchor boxes using the improved KMeans++ algorithm and replacing the box loss function with CIoU .Evaluation metrics used in the research include precision (P), recall (R), and mAP, with precision and recall being basic indicators and mAP calculating the average AP value of different types of garbage The improved YOLOv5 algorithm in the paper achieves a detection accuracy of 88.7% and a mean average precision (mAP) of 90.6% on the trash_ICRA19 dataset, which is 9.6% higher than previous studies.

[10] Gondal, A. U., Sadiq, M. I., Ali, T., Irfan, M., Shaf, A., Aamir, M., ... & Kantoch, E. (2021). Real time multipurpose smart waste classification model for efficient recycling in smart cities using multilayer convolutional neural network and perceptron. Sensors, 21(14), 4916.

The authors discuss the challenges faced by cities in waste management due to rapid urbanization and the need for automatic waste classification and management systems

The paper presents a hybrid approach using a multilayer perceptron and a multilayer convolutional neural network (ML-CNN) for waste classification .The authors highlight the importance of better recycling of waste to reduce the amount of waste sent to landfills and the need for efficient waste classification techniques .The proposed model utilizes a camera placed in front of a waste conveyor belt to capture images of the waste for classificationThe model achieves high accuracy in waste classification, with a training, testing, and validation accuracy of 0.99% under different training batches and input features.

[11] Xiao, J. (2022, March). A waste image classification using convolutional neural networks and ensemble learning. In Proceedings of the 6th International Conference on Control Engineering and Artificial Intelligence (pp. 29-33).

The paper compares a single convolutional neural network (CNN) model and an ensemble model based on CNNs for garbage classification, finding that the ensemble model achieves higher accuracy Previous works have also explored garbage classification using CNN models, such as Public GarbageNet, which can identify multiple types of domestic garbage with high accuracy .The paper considers the problem of image-based garbage classification and compares different CNN models (Xception, VGG16, ResNet, DenseNet, Inception) and ensemble learning methods (random forest, AdaBoost, XGBoost, deep neural network), finding that ensemble learning generally outperforms single CNN models.Waste classification is seen as essential for sustainable development, and the paper's findings suggest that ensemble learning models have promising applications in garbage classification systems.

[12] Ma, Xiaoxuan, Zhiwen Li, and Lei Zhang. "An improved ResNet-50 for garbage image classification." Tehnički vjesnik 29.5 (2022): 1552-1559.

In This paper garbage picture categorization research can be done using many classification model but mainly using ResNet model.It uses CBAM(Convolutional Block Attention Module) consists of two components: the channel attention module (CAM) and the spatial attention module (SAM).Both global average pooling and max pooling are commonly used operations in convolutional neural networks (CNNs) for dimensionality reduction and feature extraction. They help in summarizing the information in the feature maps and providing a compact representation for further processing and classificationThe six types of wastes are mentioned are glass, cardboard, metal, paper, plastic, and trash.accuracy of ResNet-50 is 92.08%.

[13] Mittal, Ishika, et al. "Trash classification: classifying garbage using deep learning." Journal of engineering sciences 11.7 (2020).

The paper presents a deep learning algorithm that accurately classifies images of garbage, improving waste management and segregation processes CNN is a type of Deep Learning algorithm which accepts input in the form of images ,In this paper it collects the waste images using Stereo camera and finnaly detect the object to classify images into different categories like glass, paper, plastic, metal , cardboard . In CNN is trained under 7 layers due to size and resolution of images and gives better output. In this 4 types of dataset is used and combined it get different classes like Glass, paper, plastic, cardboard, metal, Organic and recyclable.

[14] Pandey, Ayush, et al. "Enhancing Waste Management: Automated Classification of Biodegradable and Non-biodegradable Waste using CNN.

The paper proposes an automated waste classification system using Convolutional Neural Networks (CNNs) to improve waste management and also to classify waste into biodegradable and non-biodegradable categoriesResearch has compared different CNN architectures such as VGG, Inception, ResNet, and others to determine the optimal model for waste classification tasks Input layer preprocesses waste image data, convolutional layers extract features, activation layers introduce nonlinearity, pooling layers reduce dimensionality, and dropout layers prevent overfitting. Fully connected layers map features to output nodes, and the output layer determines waste classification probabilities using SoftMax.accuracy of 99.23% in categorizing plastic garbage into 4 categories

[15] Ramsurrun, Nadish, et al. "Recyclable waste classification using computer vision and deep learning." 2021 zooming innovation in consumer technologies conference (ZINC). IEEE, 2021.

In this paper our work proposes a Deep Learning approach using computer vision to automatically identify the type of waste and classify it into five main categories: plastic, metal, paper, cardboard and glass.Also compare two Machine Learning techniques, Support Vector Machine (SVM) and Convolutional Neural Network (CNN) also known as AlexNet) for the classification of waste into five main classes (glass, paper, metal, plastic, cardboard).They reached a 92% accuracy and used Raspberry Pi to open the bin where images are sent using LoRaWan connectivity.

[16] Wang, Y., Zhao, W. J., Xu, J., & Hong, R. (2020). Recyclable waste identification using cnn image recognition and gaussian clustering. arXiv preprint arXiv:2011.01353.

The paper proposes a convolutional neural network (CNN) model for waste identification and classification, using transfer learning from a pretrained Resnet-50 CNN for feature extraction .The model utilizes a sliding-window process for image segmentation in the pre-classification stage and Gaussian Clustering to locate the objects in the post classification stage.The model achieves an overall detection rate of 48.4% in simulation and a final classification accuracy of 92.4% .The TrashNet dataset is augmented for training the model, which contains 2527 RGB waste images labeled with six categories: cardboard, glass, metal, paper, plastic, and trashPrevious works in waste object classification have focused on single object identification, while this study combines mass detection with high performance identification.

[17 Yang, M., & Thung, G. (2016). Classification of Trash for Recyclability Status; CS229 Project Report.

The paper proposes a computer vision approach to classify garbage into recycling categories using support vector machines (SVM) with scale-invariant feature transform (SIFT) features and a convolutional neural network (CNN) .The authors collected a dataset of around 400-500 images for each class, including glass,paper, metal, plastic, cardboard, and trash .Radial basis kernels were found to be the best for image datasets, and the SVM's C parameter was set to 1000, while gamma was set to 0.5 .Various image transformations were performed to account for different orientations of recycled material and maximize the dataset size .The SVM was chosen as the initial classification algorithm due to its simplicity and effectiveness.

[18] He, Y., Gu, Q., & Shi, M. (2020). Trash Classification Using Convolutional Neural Networks Project Category: Computer Vision.

The project aimed to provide an automated waste sorting tool using Convolutional Neural Networks (CNN) and explored several well-known architectures, including modified AlexNet, dropout, data augmentation, and learning rate decay.The project followed U.S. standards on sorting recyclables and developed a model that takes an image of waste and outputs a vector with probabilities of six categories: cardboard, glass, metal, paper, plastic, and trash. The model used linear activation function and categorical hinge loss function .The authors plan to explore other models like VGG and ResNet and utilize transfer learning for higher accuracy. They also aim to upgrade the tool to classify waste according to more detailed rules for different countries like Japan and China .The project observed that data augmentation could be helpful, and models with partial data augmentation achieved higher final test accuracy. However, adding dropout did not give obvious improvement to test accuracy The highest test accuracy achieved was 79.94% with partial data augmentation and a Support Vector Machine (SVM) classifier .

[19] Awe, O., Mengistu, R., & Sreedhar, V. Final Report Smart Trash Net Waste Localization and Classification. arXiv 2017. Preprint.

The paper focuses on waste localization and classification using Faster R-CNN, a region-based object detection model.The authors propose a fine-tuned Faster R-CNN architecture to categorize waste into landfill, recycling, and paper.The dataset used for training and evaluation is generated by composing images from the TrashNet dataset.The performance of the model is evaluated using precision-recall curves and the average precision (AP) metric.The paper discusses the challenges faced, such as bias issues and the use of a white background that may affect the classification of paper waste.

[20] Kulkarni, Hrushikesh N., and Nandini Kannamangalam Sundara Raman. "Waste object detection and classification." CS230 Stanford (2019).

The paper discusses the use of Hybrid Transfer Learning for waste object classification and Faster R-CNN for object detection.The authors propose an architecture that utilizes GANs for creating collages and a fine-tuned Faster R-CNN for object detection .The dataset used in the study consists of collaged images with different waste objects, including glass, plastic, paper, trash, metal, and cardboard.The authors mention the use of the TrashNet dataset as a baseline, which contains images of different waste objects divided into six labeled classes .Different classifiers were experimented with, including ResNet, and it was found that ResNet worked the best for robust classification

2.2 Comparison Table

Table.1 Comparison Table

	Title	**Year**	**Objectives**	**Limitations**	**Advantages**	**Performance metrics**	**Gaps**
Reference 1	A Waste Classification Method Based on a Multilayer Hybrid Convolution Neural Network	2021	The paper introduces an MLH-CNN-based waste classification method to improve accuracy, address existing model limitations, optimize parameters.	Lacks detailed discussions on implementation challenges, dataset limitations, real-world impact, computational resource requirements.	Simpler and efficient MLH-CNN method with higher classification accuracy compared to state-of-the-art methods, demonstrated feature extraction ability.	Accuracy of up to 92.6%	Include computational resource analysis, ethical considerations, and a need for broader evaluation on diverse datasets to gauge generalizability.

Reference 2	Deep Convolutional Neural Networks Object Detector for Real-Time Waste Identification	2020	The paper aims to enhance CNN object detectors for waste identification, generalization, and detection speed through fine-tuning SSD and RPN on the TrashNet database.	This include a narrow evaluation scope, reliance on specific datasets, lack of computational resource analysis, insufficient discussion on pre-trained model limitations	The paper presents advanced techniques, including fine-tuning SSD and RPN on the TrashNet database, resulting in superior waste detection performance compared to other methods.	High accuracy, with precision values ranging from 95.76% to 97.63%	The paper lacks exploration beyond municipal waste, limits generalizability due to dataset reliance, lacks detailed computational resource analysis.

Reference 3	ARTD-Net: Anchor-Free Based Recyclable Trash Detection Net Using Edgeless Module	2023	The paper aims to develop an automatic system for recyclable trash collection using anchor-free detection networks, improving accuracy through feature extraction and classification enhancements.	Lack of detailed model comparisons, dataset representation, discussion on computational requirements,	Efficiently detects overlapped wastes, offers flexibility with anchor-free models, improves accuracy via centralized feature extraction and multiscale feature maps, enhances classification.	ARTD-Net2 shows higher accuracy than ARTD-Net1.	Scalability concerns, interpretability insights, dataset representation limitations, and practical implementation challenges, hindering comprehensive assessment and real-world application.
Reference 4	GarbageNet: A Unified Learning Framework for	2021	Introduces an incremental learning framework addressing garbage classificatio	Include insufficient details on evaluation datasets, lack of specifics on GarbageNet	GarbageNet, an incremental learning framework addressing data scarcity,	The paper mentions that the proposed method	Computational resource insights, interpretability, and ethical considerations in AI-

	Robust Garbage Classification		n challenges, anchor-based classification, attentive mixup.	framework extendability, absence of computational resource discussion,	achieving state-of-the-art performance, utilizing weakly-supervised transfer learning.	achieved 94.6% accuracy.	based garbage classification.

Reference 5	Garbage classification system based on improved ShuffleNet v2	2021	GCNet, a model with a parallel mixed attention mechanism and new activations, aiming for high accuracy and real-time performance in garbage classificati	Limitations include dataset specificity, single-object classification support, data collection challenges, lack of detailed real-time performance analysis	GCNet, a achieving high accuracy, with advantages including low computational requirements, PMAM-enhanced feature extraction, FReLU activation,	Accuracy = 97.9%	The paper's gaps include lack of dataset details, generalizability discussion, computational resource analysis, and ethical

			on, with potential environmental applications.		transfer learning.		considerations in deep learning-based garbage classification systems.
Reference 6	Trash and Recycled Material Identification using Convolutional Neural Networks (CNN)	2022	The paper Integrating this image processing -based classification into smart trash cans will be more suitable for cleaning garbage	-Incorrect detection of the indoor images from the dataset	-Detecting the Outdoor images and depositing into trash and recycled cans	Accuracy= 92%	

Reference 7	Garbage Net: A Unified Learning Framework for Robust Garbage Classification	2021	The paper discuss on lack of data and noisy data with multiple classes for object classification	-The paper does not focus on the object detection by camera	-It can explain about the detection of object with multiple classes	Accuracy= 94.9%	
Reference 8	YOLO-Based Object Detection for Separate Collection of Recyclables and Capacity Monitoring of Trash Bins	2022	Utilizes object detection, capacity monitoring, and GPS for waste management	-Users can only throw away one recyclable at a time	-YOLO achieves 155 FPS and twice the map of other models.	Accuracy= 91%	

Reference 9	Recyclable Waste Identification Using CNN Image Recognition and Gaussian Clustering	2021	-Develop CNN model for waste identification and classification. -Achieve 48.4% detection rate	-Complex waste classification due to suboptimal lighting and overlapping positions. -False-positive rate high for Glass objects, low false-negative rate.	-Waste identification accuracy of 92.4% achieved. -Utilizes transfer learning from Resnet-50 CNN for feature extraction.	Accuracy= 92.4%	
Reference 10	Classifying garbage using Deep Learning	2020	Enhance waste segregation system with wifi and proximity sensors to alert when bins are full.	Neglecting potential biases and errors in classification.	Accelerates waste segregation, improving waste management practices. Decreases contamination risks to land and water resources	Accuracy 98.2%	

Reference 11	Automated Classification of Biodegradable and Non-biodegradable Waste using CNN	2023	Develop an automated waste classification system using Convolutional Neural Networks (CNNs) to improve waste management. Collect and preprocess data to create a dataset of biodegradable and non-biodegradable waste images for training the CNN model	The small collection of garbage photos used to train the model may limit its overall performance and accuracy	Reduced human error: the potential expansion of waste classification systems to include other types of waste beyond biodegradable and non-biodegradable categories	Training Set Accuracy: 96.06% Test Set Accuracy: 91%	Developing the model to handle mixed and visually challenging waste enhances practical applicability.

Reference 12	Recyclable Waste Classification Using Computer Vision And Deep Learning	2021	Develop a system that utilizes computer vision and deep learning algorithms to automatically identify and classify different types of recyclable waste.	misdiagnosing a patient with heart disease when they do not have it.	Dataset size and diversity constraints may impact result generalizability. Incomplete waste classification excludes food waste, limiting scope.	SVM Accuracy: Around 63% 1 CNN (AlexNet) Accuracy: 22% 1 ResNet 50 + SVM Accuracy: Around 87% 1 VGG19 with SoftMax Accuracy: Around 88%	Inclusion of food waste for a more comprehensive waste classification.

Reference 13	An Improved ResNet-50 for Garbage Image Classification	2022	To implement multi-scale feature fusion for improved classification performance To improve the robustness of the classification model on small datasets	Complex network modifications increase computational demands, hindering deployment on resource-constrained devices.	Higher classification performance than existing models on small datasets with few samples.	ResNet-50: 0.8446 [T5] Inception-ResNet: 0.8834 [T5] DenseNet121: 0.89 [T5] ResNet-50-B (proposed): 0.9208 [T5]	Absence of detailed analysis on computational resources for training and deployment hinders practical implementation.

Reference 14	The Object Detection of Underwater Garbage with an Improved YOLOv5 Algorithm	2022	-Improve object detection for underwater garbage using YOLOv5 algorithm. -Enhance prediction side with reclustered anchor boxes and optimized loss function.	GIoU has limitations in non-overlapping cases for gradient updates.	-Improved YOLOv5 algorithm achieved 88.7% detection accuracy. -YOLOv5 enhances detection of small objects with FPN.	Accuracy= 88.7%	

Reference 15	Real Time Multipurpose Smart Waste Classification Model for Efficient Recycling.	2021	-Develop a real-time waste classification model using ML-CNN and perceptron. -Implement binary classification for metal and non-metal waste.	-Low computation power affects local system performance. -is 2 KGRobotic arm can pick items of 12 cm to 20 cm.[2] Arm weight limitation	-Real-time waste classification model with high accuracy. -Hybrid approach using multilayer perceptron and convolutional neural network	Accuracy= 89%	
Reference 16	A waste image classification using convolutional neural networks and ensemble learning.	2022	-Compare single CNN vs. ensemble model for garbage classification on accuracy. -Explore lightweight deep learning	-Overfitting in neural networks addressed with early stopping mechanism. -Need for lightweight deep learning models for	-Ensemble learning outperforms single neural network models. -Random forest achieves the highest accuracy	Accuracy= 80%	

			models suitable for mobile devices	mobile devices	among ensemble		
Reference 17	The research paper is titled "Classification of Trash for Recyclability Status.	2016	-Classify garbage into recycling categories using SVM and CNN. -Identify and classify multiple objects from a single image or video	-CNN underperformed due to trouble finding optimal hyperparameters. -SVM outperformed CNN due to its simplicity and ease of use.	-SVM outperformed CNN in trash classification. -CNN architecture similar to AlexNet used for trash classification	Accuracy= 63%	

Reference 18	Final Report: Smart Trash Net: Waste Localization and Classification	2018	-Categorize waste into landfill, recycling, and paper categories. -Utilize Faster R-CNN for object detection and classification.	-Bias issue with training examples, despite even representation. -White background similarity affecting 'paper' category performance. -	-Faster R-CNN provides cost-free region proposals. -Automated waste sorting enhances recycling rates.	Accuracy= 87%	

Reference 19	The research paper is about trash classification using Convolutional Neural Networks	2021	-Develop automated waste sorting tool using Convolutional Neural Networks. -Achieve highest test accuracy of 79.94% with partial data augmentation.	Limited computation power hindered full model convergence. -Dropout feature slowed down training loss and accuracy convergence.	Explored various model architectures and techniques for trash classification. -Focused on CNN models to classify recyclables and trash effectively	Accuracy= 79.4%	
Reference 20	The research paper is titled "Waste Object Detection and Classification."	2019	-Waste object detection and classification using Faster R-CNN. -Generating collages with minimal overlap for	-Hybrid training complexity due to two learning rates. -Avoiding GP-GANs due to blurred image features affecting	-Waste object detection and classification using Hybrid Transfer Learning. -Generating collages to train the	Accuracy= 88%	

			optimal image placement.	performanc e.	model from scratch		

3. REQUIREMENT SPECIFICATION

3.1 Software and Hardware Requirements

Software Requirements:

- **Python:**

Python is a general-purpose, high-level, interpreted programming language. Code readability is prioritized in its design philosophy, which makes heavy use of indentation. Python uses garbage collection and has dynamic typing. It supports a variety of programming paradigms, including procedural, object-oriented, and functional programming as well as structured programming (especially this). Due to its extensive standard library, it is frequently referred to as a "batteries included" language.

- **IDE (Integrated Development Environment):**

Visual Studio Code:

Visual Studio Code is a free and open-source code editor developed by Microsoft that provides an integrated development environment (IDE) for building and debugging applications. It is available on Windows, macOS, and Linux. Visual Studio Code supports a wide range of programming languages, including popular languages like JavaScript, Python, C++, and Java, as well as emerging languages like Rust and Go.

Jupyter NoteBook:

Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It provides an interactive computing environment that allows you to execute code in real-time, which is particularly useful for data analysis, scientific computing, and machine learning tasks. The code is executed in a kernel, which is a separate process that can be started and stopped independently of the notebook interface.

Hardware Requirements:

- Processing Power: Require a computer with sufficient processing power, including a multi-core CPU or GPU (Graphics Processing Unit), to handle complex image processing algorithms and machine learning tasks efficiently.

- Graphics Card (GPU): Utilize a dedicated GPU, especially NVIDIA CUDA-enabled GPUs, for accelerating deep learning computations and speeding up training of machine learning models.
- Memory (RAM): Ensure an adequate amount of RAM (Random Access Memory) to store and manipulate image data, especially when processing large images or datasets.

3.2 Functional and Non Functional Requirements

Functional Requirements:

They typically outline the actions, processes, or tasks that the system must be able to perform.

- Develop algorithms to enhance low-light images, including denoising, contrast adjustment, color correction, and sharpening.
- Implement edge enhancement techniques to improve image contrast and definition, particularly in low-light conditions.
- Integrate a Generative Adversarial Network (GAN) model to generate realistic and natural-looking enhancements, learning from the dataset's distribution.
- Create a preprocessing pipeline to prepare images for enhancement, including noise reduction and edge detection.
- Design a user-friendly interface for users to upload, process, and download enhanced images, providing options for parameter adjustments and visual feedback.

Non Functional Requirements:

They define the qualities or attributes that a system or software application must possess, beyond its basic functionality.

- Accuracy: The image enhancement algorithms should produce accurate and faithful representations of the original scene, preserving important details and minimizing artifacts.
- Scalability: The system should be scalable to accommodate a large volume of image processing requests, with the ability to scale resources dynamically based on demand.
- Robustness: The system should be robust to variations in input data, including different lighting conditions, image quality, and noise levels.
- Usability: The user interface should be intuitive and easy to use, with clear instructions and feedback to guide users through the image enhancement process.

- Compatibility: Ensure compatibility with a wide range of operating systems, web browsers, and devices commonly used by users, ensuring a seamless experience across different platforms.

4. SYSTEM ANALYSIS AND DESIGN

4.1 Existing Methodology

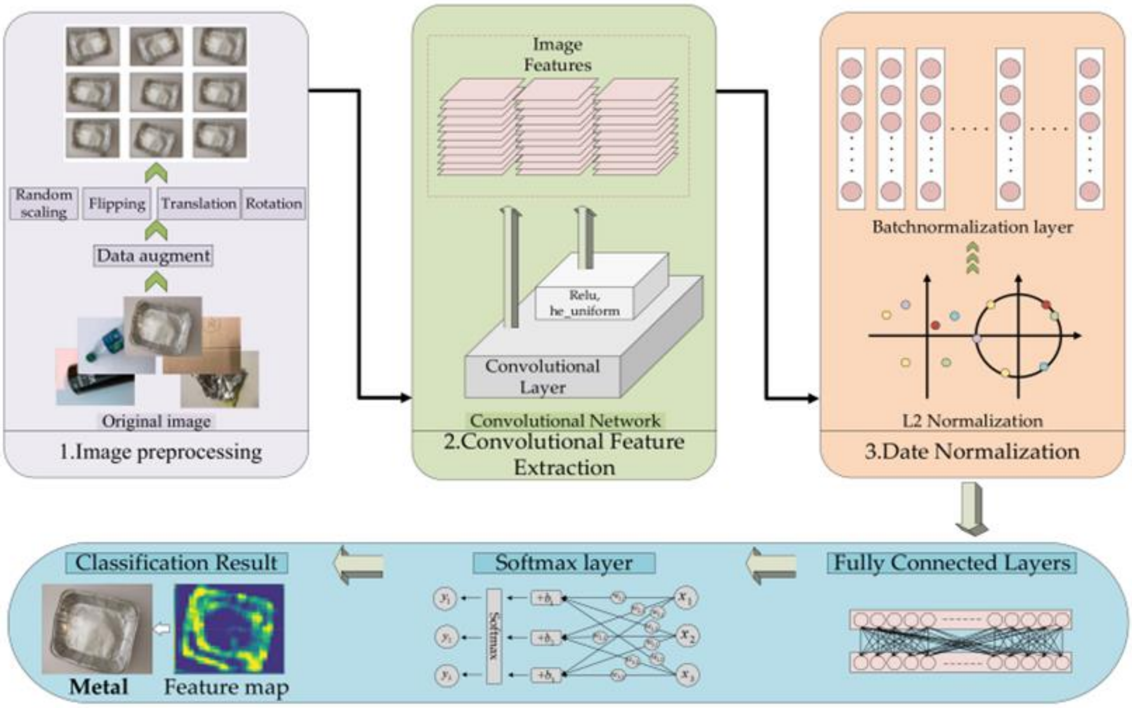

Fig.1 Existing Model

The diagram illustrates a comprehensive process of image classification using convolutional neural networks (CNN). The process begins with image preprocessing, where the original images are augmented through various techniques such as random scaling, flipping, translation, and rotation. These augmentations enhance the dataset by creating multiple variations of the images, which helps in improving the robustness of the model.Next, the augmented images are passed through a convolutional network for feature extraction. In this stage, convolutional layers are employed to detect and extract relevant features from the images. The convolutional layers apply filters to the images, capturing essential patterns and details such as edges, textures, and shapes, which are crucial for classification tasks.Following feature extraction, the data undergoes normalization. This involves passing the extracted features through batch normalization and L2 normalization layers. Batch normalization standardizes the inputs to a layer for each mini-batch, stabilizing the learning process and reducing the number of training epochs required. L2 normalization ensures that the data is scaled to a uniform range, which helps in improving the convergence of the neural network.Finally, the normalized features are fed into the classification section of the model, which consists of fully connected layers and a softmax layer. The fully connected layers combine the extracted features to form high-level representations, while the softmax layer produces probability distributions over the possible classes. The model then outputs the classification result, indicating the predicted category of the input image. This process enables the model to accurately classify images, such as identifying a metal object in the given example.

4.2 Methodology

4.2.1 Dataset

We are using the TrashNet dataset, it is a collection of images used for garbage classification, specifically designed for training machine learning models in waste management applications. It consists of images of various types of trash, such as paper, cardboard, plastic, metal, glass, and trash bags. Each image is labelled according to its respective category, enabling supervised learning approaches for garbage classification tasks. The dataset aims to facilitate research and development in the field of waste management, promoting the use of technology to automate and optimize waste sorting processes. It has been utilized in numerous studies and projects focused on image classification, object detection, and environmental sustainability.

SI.no	Attributes	Number of Images
1	Carboard	403
2	Plastic	482
3	Metal	410
4	Glass	501
5	Paper	594
6	Plastic	482
Total		2,872

Table 2: Dataset (Trashnet) description

Fig-2.1

Fig-2.2

Fig-2.3

Fig-2.4

Fig-2.5

Fig-2.6

Fig-2: Images of different classes in TrashNet Dataset

4.2.2 Preprocessing

The preprocessing stage involves a series of steps aimed at cleaning, enhancing, and standardizing the images in the TrashNet dataset. This process ensures that the input data is of high quality, which is crucial for training robust deep learning models such as DenseNet121 and MobileNetV3. By addressing common issues like blurriness, low lighting, poor contrast, and color integrity, we prepare the dataset for optimal performance in automated waste classification tasks. The preprocessing stage is a critical component of our waste management model, as it ensures that the input data is clean, standardized, and suitable for training deep learning algorithms. In this project, we leverage the TrashNet dataset and employ several preprocessing techniques to address issues such as blurriness, low lighting, poor contrast, and color integrity. The following steps outline the preprocessing pipeline used for preparing the dataset:

1. Data Acquisition and Loading

Dataset Description: The TrashNet dataset includes images of waste items categorized into six classes: Cardboard, Glass, Metal, Paper, Plastic, and Trash/Non-Recyclable.

Loading Data: Images are loaded into the system, ensuring they are correctly labeled according to their respective categories.

2. Image Resizing

Standardization: All images are resized to a fixed dimension of 224x224 pixels to ensure uniform input size across different deep learning models (DenseNet121, MobileNetV3).

Aspect Ratio: Care is taken to maintain the aspect ratio, using padding if necessary, to avoid distortion of images.

3. Image Enhancement

Blurriness Reduction: Applying filters such as Gaussian blur to smooth images and mitigate noise, followed by sharpening techniques to enhance edges and important features.

Lighting Correction: Adjusting the brightness and contrast of images to ensure that all features are visible under different lighting conditions. Histogram equalization can be used to improve the contrast.

Color Integrity: Enhancing color fidelity through color balancing techniques to ensure that the colors in the images are as true to life as possible.

4. Normalization

Pixel Value Scaling: Normalizing pixel values to a range of [0, 1] or [-1, 1] depending on the requirements of the deep learning model. This is typically done by dividing pixel values by 255.

Mean Subtraction and Standard Deviation Scaling: Further normalization can be performed by subtracting the mean pixel value and scaling to the standard deviation of the dataset, which helps in faster convergence during training.

5. Data Augmentation

Transformation Techniques: Applying random transformations such as rotations, translations, flips, and zooms to increase the diversity of the training dataset and improve model generalization.

Noise Injection: Adding slight noise to images to make the model more robust to variations and prevent overfitting.

6. Splitting Data

Training, Validation, and Test Sets: Splitting the dataset into training, validation, and test sets to evaluate the performance of the models. Typically, a split of 70% training, 20% validation, and 10% test is used.

Stratification: Ensuring that each subset of data has a balanced representation of all waste categories.

7. Data Cleaning

Removing Outliers: Identifying and removing images that do not belong to any category or are too ambiguous to classify.

Handling Duplicates: Checking for and removing duplicate images to ensure that the model does not get biased by redundant data.

8. Annotation Verification

Label Accuracy: Verifying that all images are correctly labeled and making corrections if necessary to ensure the integrity of the dataset.

4.2.3 VGG 16

VGG16 is a deep convolutional neural network architecture consisting of 16 layers, including convolutional and max-pooling layers, followed by fully connected layers. Its design emphasizes depth and simplicity, with small 3x3 convolutional filters used throughout the network.

VGG16, named for its 16 weight layers, is composed of:

1.Convolutional Layers:

- The network includes 13 convolutional layers, each using small 3x3 filters. This choice of filter size ensures that the network captures fine details while keeping the computational complexity manageable.
- Convolutional layers are grouped into blocks. Each block consists of 2-3 convolutional layers followed by a max-pooling layer.
- The number of filters increases progressively across the layers: 64 filters in the first two layers, 128 in the next two, 256 in the following three, and finally 512 in the last six convolutional layers (three per block).

2. Activation Function:

- Each convolutional layer is followed by a ReLU (Rectified Linear Unit) activation function, which introduces non-linearity to the model, helping it learn complex patterns.

3. Max-Pooling Layers:

- Five max-pooling layers, each with a 2x2 filter and a stride of 2, are used to downsample the spatial dimensions of the feature maps. This helps in reducing the computational load and also in extracting dominant features that are invariant to small translations.

4. Fully Connected Layers:

- After the series of convolutional and max-pooling layers, the network includes three fully connected (dense) layers: two layers with 4096 neurons each and one final layer with 1000 neurons (for the 1000-class classification in ImageNet).
- The fully connected layers also use ReLU activation functions, with the final layer using a softmax activation to produce probability distributions over the classes.

5. Softmax Layer:

- The final layer is a softmax layer that outputs probabilities for each class, summing up to 1. This is crucial for multi-class classification problems like ImageNet and TrashNet.
-

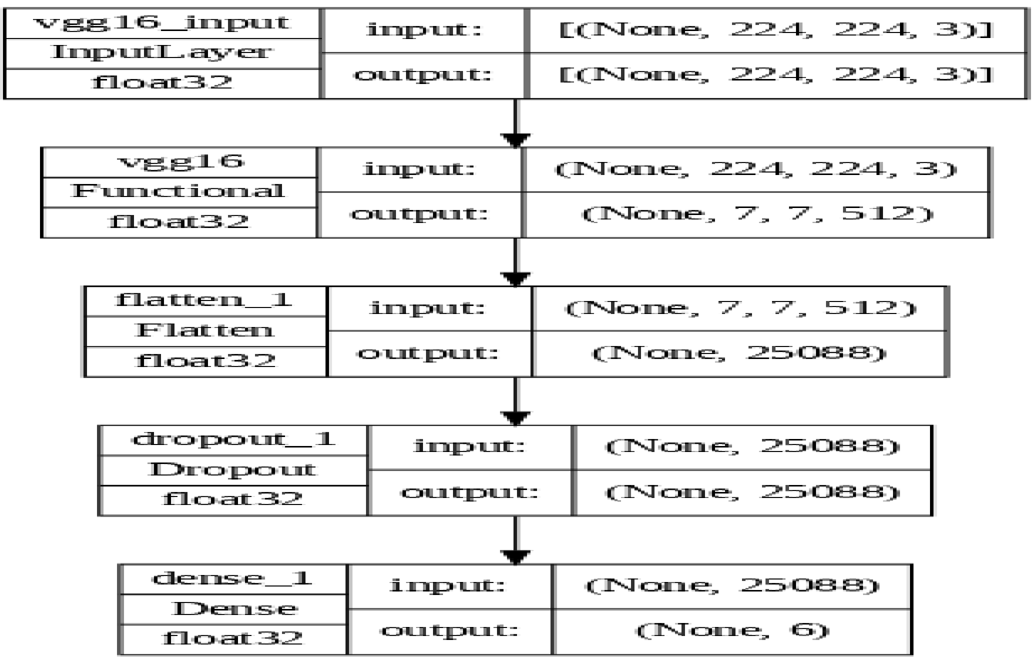

Fig.3: Architecture of VGG16

Working of VGG16 on TrashNet Dataset:

1. Input Layer:

The input to VGG16 is a fixed-size 224x224 RGB image. Images from the TrashNet dataset are resized to fit this dimension.

2. Feature Extraction:

The input image passes through the series of convolutional layers where the filters extract features like edges, textures, shapes, and other detailed patterns. Each convolutional block is followed by max-pooling, which reduces the spatial dimensions and highlights the most significant features.

3. Flattening:

After the convolutional layers and max-pooling, the resulting feature map is flattened into a one-dimensional vector.

4. Classification:

This flattened vector is passed through the fully connected layers, which act as a classifier. The layers learn to map the high-level abstract features extracted by the convolutional layers to the output classes (Cardboard, Glass, Metal, Paper, Plastic, Trash/Non-Recyclable).

5. Output Layer:

The final output layer uses the softmax function to provide a probability distribution over the six classes of waste in the TrashNet dataset. The class with the highest probability is chosen as the predicted category for the input image.

Advantages of VGG16:

Simplicity: The use of small 3x3 convolution filters simplifies the design and ensures that the network is deep while still manageable.

Performance: VGG16 has demonstrated high accuracy in various image classification tasks, making it a strong candidate for applications like automated waste sorting.

Transfer Learning: VGG16 pre-trained on large datasets like ImageNet can be fine-tuned for specific tasks, significantly improving performance with less training data.

By leveraging VGG16, our project aims to develop a robust and efficient system for automated waste classification, facilitating improved recycling practices and contributing to sustainable waste management solutions.

4.2.4 MobileNetV3:

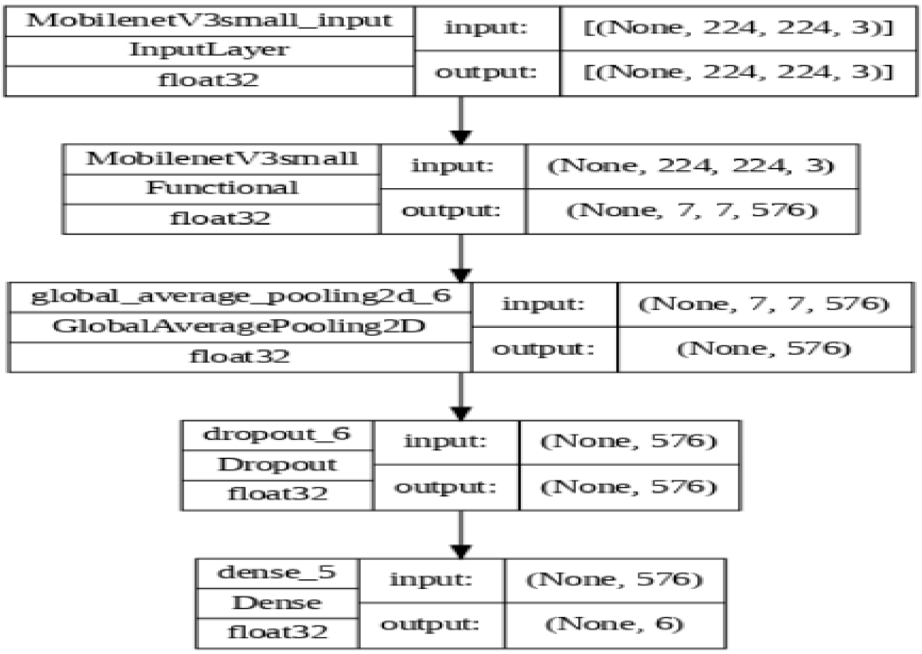

Fig-5: Architecture of MobileNetV3

In MobileNetV3 for image classification, input images are preprocessed by resizing them to a fixed size (e.g., 224x224) and normalizing pixel values.

The network consists of lightweight depthwise separable convolution layers, applying depthwise convolutions independently to each channel followed by pointwise convolutions to adjust channel dimensions.

Activation functions like Hard Swish and Swish-6 introduce non-linearity for learning complex patterns.

Squeeze-and-Excitation (SE) blocks are used to selectively emphasize informative features by recalibrating channel-wise dependencies. After feature extraction, a global average pooling layer aggregates spatial information into a fixed-size representation.

A linear classifier (e.g., fully connected layer) then predicts class probabilities. MobileNetV3's design optimizes for efficiency and accuracy, making it suitable for image classification on resource-constrained devices.

4.2.5 DenseNet121:

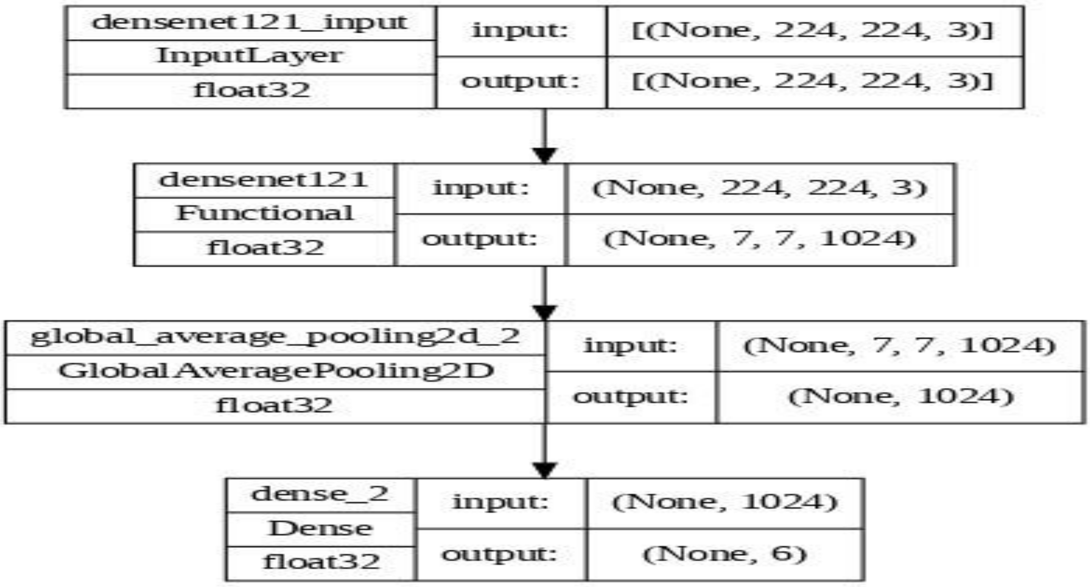

Fig 3: Architecture of Densenet121

DenseNet is a convolutional neural network architecture renowned for its dense connectivity pattern, characterized by direct connections from each layer to every other layer in a feed-forward fashion.

In our approach utilizing the TrashNet dataset, we initially resize the images to dimensions of 224x224 pixels, aligning them with the input requirements of the DenseNet121 model. We leverage a pre-trained DenseNet121 architecture to extract features from the input image, generating a feature map of size 7x7x1024. Subsequently, a global average pooling layer is applied to reduce the dimensionality of the feature map by averaging the values of each feature channel across its width and height, resulting in a vector of size 1024. Finally, we incorporate a dense layer, constituting a fully connected layer with 6 neurons, to classify the image into six categories. The output of this dense layer is a vector of size 6, where each element represents the probability of the image belonging to a specific category. Overall, our methodology employs the DenseNet121 model to extract features, followed by dimensionality reduction via global average pooling, culminating in image classification using a dense layer.

4.3 Proposed Model

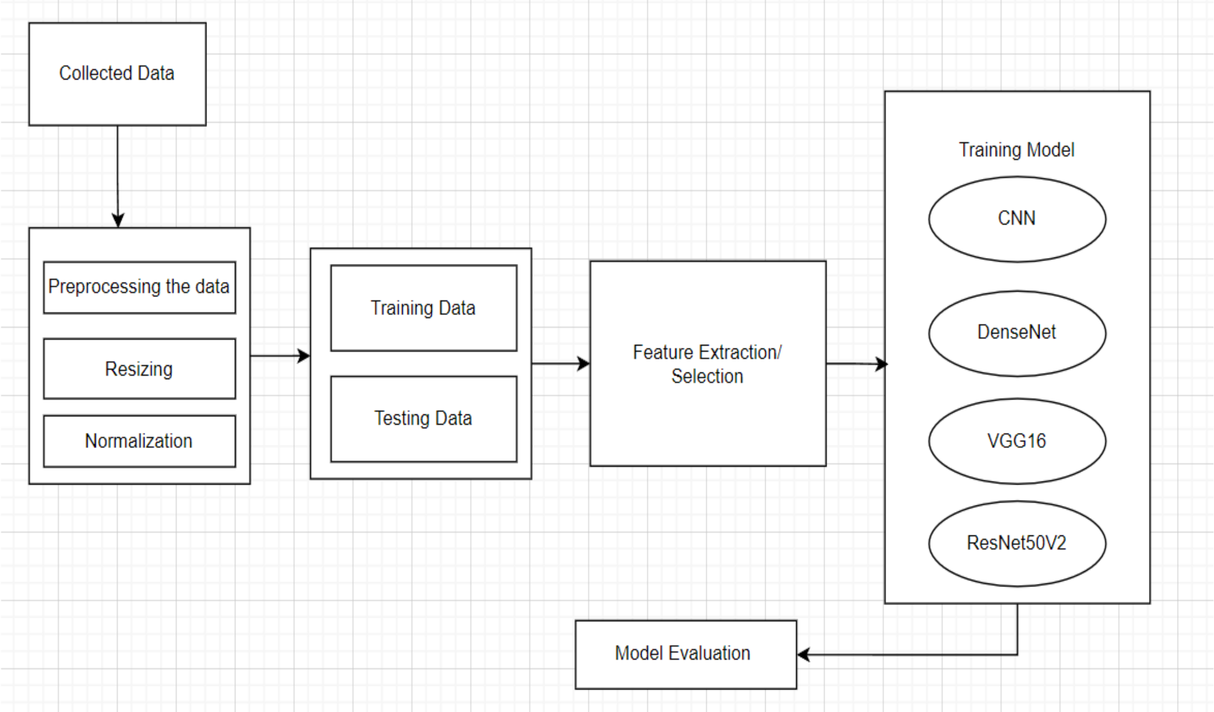

Fig-6: Proposed Model

1. Collected Data

Description: This is the initial step where raw data is gathered. The data can come from various sources, depending on the specific application (e.g., images, text, sensor data).

2. Preprocessing the Data

Description: The collected data undergoes several preprocessing steps to ensure it is in a suitable format for model training.

Resizing: Adjust the dimensions of the data, such as images, to a consistent size required by the model.

Normalization: Scale the data values to a standard range (e.g., 0 to 1) to facilitate better convergence during training.

3. Splitting Data

Description: The preprocessed data is divided into two subsets:

Training Data: Used to train the machine learning models.

Testing Data: Used to evaluate the performance of the trained models.

4. Feature Extraction/Selection

Description: Extracting or selecting significant features from the training data that will be used by the machine learning models. This step helps in improving the model's efficiency and performance.

5. Training Model

Description: Different machine learning models are trained using the extracted features from the training data. The models depicted are:

CNN (Convolutional Neural Network): A deep learning model particularly effective for image data.

DenseNet (Densely Connected Convolutional Networks): A type of CNN that connects each layer to every other layer in a feed-forward fashion.

VGG16: A popular CNN model known for its simplicity and depth with 16 weight layers.

ResNet50V2: A residual neural network with 50 layers, known for its skip connections which help mitigate the vanishing gradient problem.

6. Model Evaluation

Description: After training, the models are evaluated using the testing data to determine their performance. This step involves measuring various metrics such as accuracy, precision, recall, and F1-score to assess how well the models generalize to new, unseen data.

5. IMPLEMENTATION

The model implements image classification for waste management using VGG16 and MobileNetV3 architectures with TensorFlow and Keras. The script begins by importing essential libraries such as PIL for image processing, sklearn for data splitting, and TensorFlow for model building. Key parameters such as batch size, image size, and dataset split ratios are defined. The code then sets up directories for loading images categorized into six waste types: Cardboard, Glass, Metal, Paper, Plastic, and Trash/Non-Recyclable. Images are loaded, resized to 224x224 pixels, normalized, and stored in numpy arrays for training and testing.

The data is split into training and testing sets using `train_test_split`. The datasets are then converted into TensorFlow `tf.data.Dataset` objects and batched for efficient training. The VGG16 model is initialized with pre-trained weights from ImageNet, excluding the top fully connected layers, to leverage transfer learning. Custom fully connected layers are added to match the six waste categories. Similarly, MobileNetV3Large is initialized with pre-trained weights, and custom classification layers are added.

Both models are compiled using the Adam optimizer and categorical cross-entropy loss, and they are trained on the prepared datasets. During training, the models are evaluated using metrics such as accuracy, precision, recall, and F1-score. After training, the models are tested on the test set, and the results are displayed, showcasing the classification performance on waste images.

This process illustrates the workflow of using VGG16 and MobileNetV3 for automated waste classification, from data preprocessing and model setup to training and evaluation, highlighting the potential for improved efficiency in waste management through deep learning.

MODEL:

```
import numpy as np # linear algebra

import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)


# Input data files are available in the read-only "../input/" directory

# For example, running this (by clicking run or pressing Shift+Enter) will list all files under
the input directory
```

```
import os
for dirname, _, filenames in os.walk(r'/content/drive/MyDrive/dataset-resized'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
import numpy as np
import pandas as pd
import os
import cv2
import matplotlib.pyplot as plt
import seaborn as sns
import tensorflow as tf
import keras
from tqdm import tqdm
from keras.callbacks import EarlyStopping,ModelCheckpoint
from sklearn.metrics import confusion_matrix , accuracy_score
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
import warnings
warnings.filterwarnings('ignore')
#Create Files_Name
image_data="/content/drive/MyDrive/dataset-resized"
pd.DataFrame(os.listdir(image_data),columns=['Files_Name'])
import tensorflow as tf
```

```
# Define parameters
train_data_dir = image_data
batch_size = 32
target_size = (224, 224)
validation_split = 0.1
test_split = 0.2

# Load the entire dataset
full_dataset = tf.keras.preprocessing.image_dataset_from_directory(
    train_data_dir,
    validation_split=validation_split,
    subset="training",
    seed=100,
    image_size=target_size,
    batch_size=batch_size,
)

# Split the dataset into training and validation subsets
num_examples = full_dataset.cardinality().numpy()
train_size = int((1 - validation_split - test_split) * num_examples)
val_size = int(validation_split * num_examples)

train_dataset = full_dataset.take(train_size)
validation_dataset = full_dataset.skip(train_size).take(val_size)
```

```
# Load the test dataset
test_dataset = tf.keras.preprocessing.image_dataset_from_directory(
    train_data_dir,
    validation_split=validation_split,
    subset="validation",
    seed=200,
    image_size=target_size,
    batch_size=batch_size,
)
plt.figure(figsize=(15, 20))
for images, labels in train_dataset.take(1):
    for i in range(8):
        ax = plt.subplot(8, 4, i + 1)
        plt.imshow(images[i].numpy().astype("uint8"))
        plt.title(class_names[labels[i]])
        plt.axis("off")
import tensorflow as tf
from tensorflow import keras

# Define VGG16 with pre-trained weights
base_model = tf.keras.applications.VGG16(input_shape=(224, 224, 3), include_top=False,
weights='imagenet')
base_model.trainable = False

# Build the sequential model
```

```
keras_model = keras.models.Sequential([
    base_model,
    keras.layers.Flatten(),
    keras.layers.Dropout(0.5),
    keras.layers.Dense(6, activation=tf.nn.softmax)
])

# Provide a sample input
sample_input = tf.random.normal((1, 224, 224, 3))

# Call the model with the sample input to initialize the shapes
_ = keras_model(sample_input)

# Print model summary
keras_model.summary()

checkpoint = ModelCheckpoint(filepath='model.keras', monitor='val_accuracy',
save_best_only=True)
early_stopping =EarlyStopping(patience=5, restore_best_weights=True)
keras_model.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
hist = keras_model.fit(train_dataset, epochs=10, validation_data=validation_dataset,
callbacks=[checkpoint, early_stopping])
# Unfreeze some layers of the base model
base_model.trainable = True
```

```
# Fine-tune from this layer onwards
fine_tune_at = 15

# Freeze all layers before the `fine_tune_at` layer
for layer in base_model.layers[:fine_tune_at]:
    layer.trainable = False

# Compile the model
keras_model.compile(optimizer=keras.optimizers.Adam(learning_rate=1e-5),
                    loss='sparse_categorical_crossentropy',
                    metrics=['accuracy'])

# Train the model
history = keras_model.fit(train_dataset,
                          epochs=10,
                          validation_data=validation_dataset)

hist_=pd.DataFrame(history.history)
hist_
plt.figure(figsize=(15,5))
plt.subplot(1,2,1)
plt.plot(hist_['loss'],label='Train_Loss')
plt.plot(hist_['val_loss'],label='Validation_Loss')
plt.title('Train_Loss & Validation_Loss',fontsize=20)
plt.legend()
```

```
plt.subplot(1,2,2)
plt.plot(hist_['accuracy'],label='Train_Accuracy')
plt.plot(hist_['val_accuracy'],label='Validation_Accuracy')
plt.title('Train_Accuracy & Validation_Accuracy',fontsize=20)
plt.legend()

X_val,y_val,y_pred=[],[],[]
for images, labels in validation_dataset:
    y_val.extend(labels.numpy())
    X_val.extend(images.numpy())
predictions=keras_model.predict(np.array(X_val))
for i in predictions:
    y_pred.append(np.argmax(i))
df=pd.DataFrame()
df['Actual'],df['Prediction']=y_val,y_pred
df

plt.figure(figsize=(25,25))
for i in range(32):
    ax = plt.subplot(8, 4, i + 1)
    plt.imshow(X_val[i].astype("uint8"))
    plt.title(f'{class_names[y_val[i]]} :: {class_names[y_pred[i]]}')
    plt.axis("off")

ax= plt.subplot()
```

```
CM = confusion_matrix(y_val,y_pred)
sns.heatmap(CM, annot=True, fmt='g', ax=ax,cbar=False,cmap='RdBu')
ax.set_xlabel('Predicted labels')
ax.set_ylabel('True labels')
ax.set_title('Confusion Matrix')
plt.show()
CM
plt.figure(figsize=(25, 25))
plot_count = 1  # Counter to keep track of the number of plotted images
for i in range(len(X_val)):
    if class_names[y_val[i]] != class_names[y_pred[i]]:
        ax = plt.subplot(8, 4, plot_count)
        plt.imshow(X_val[i].astype("uint8"))
        ax.title.set_text(f"Actual: {class_names[y_val[i]]}\nPredicted:
{class_names[y_pred[i]]}")
        plot_count += 1
        if plot_count > 32:  # Plot up to 32 mismatched images
            break
plt.show()


# prompt: print the index and class label of each


# Create a dictionary to store the class names and their corresponding indices
class_names = full_dataset.class_names
class_names_to_indices = {class_name: i for i, class_name in enumerate(class_names)}
```

```
# Print the index and class label for each class

for class_name, index in class_names_to_indices.items():

  print(f"Index: {index}, Class Label: {class_name}")
```

output:

```
Index: 0, Class Label: cardboard

Index: 1, Class Label: glass

Index: 2, Class Label: metal

Index: 3, Class Label: paper

Index: 4, Class Label: plastic

Index: 5, Class Label: trash
```

```
from sklearn.metrics import accuracy_score
import numpy as np

# Convert y_val and y_pred to NumPy arrays if they are not already
y_val = np.array(y_val)
y_pred = np.array(y_pred)

# Calculate the accuracy for each class
class_accuracies = {}
unique_labels = np.unique(y_val)
for label in unique_labels:
    indices = np.where(y_val == label)[0]  # Get indices where y_val equals the current label
    class_accuracies[label] = accuracy_score(y_val[indices], y_pred[indices])

# Print the accuracy for each class
for label, accuracy in class_accuracies.items():
    print(f"Accuracy for class {label}: {accuracy:.3f}")
```

output:

Accuracy for class 0: 0.967

Accuracy for class 1: 0.923

Accuracy for class 2: 0.944

Accuracy for class 3: 0.982

Accuracy for class 4: 0.878

Accuracy for class 5: 1.000

```
img = cv2.imread(r'/content/drive/MyDrive/dataset-resized/trash/trash45.jpg')
resized_image = cv2.resize(img, (224, 224))
resized_image.shape
import numpy as np
resized_image = np.expand_dims(resized_image, axis=0)
predicted=keras_model.predict(resized_image)
predicted_class=np.argmax(predicted)
class_label=class_names[predicted_class]
print(class_label,end="-")
if class_label in recycle:
   print("recyclable object")
else:
   print("Non Recyclable object")
```

Output:

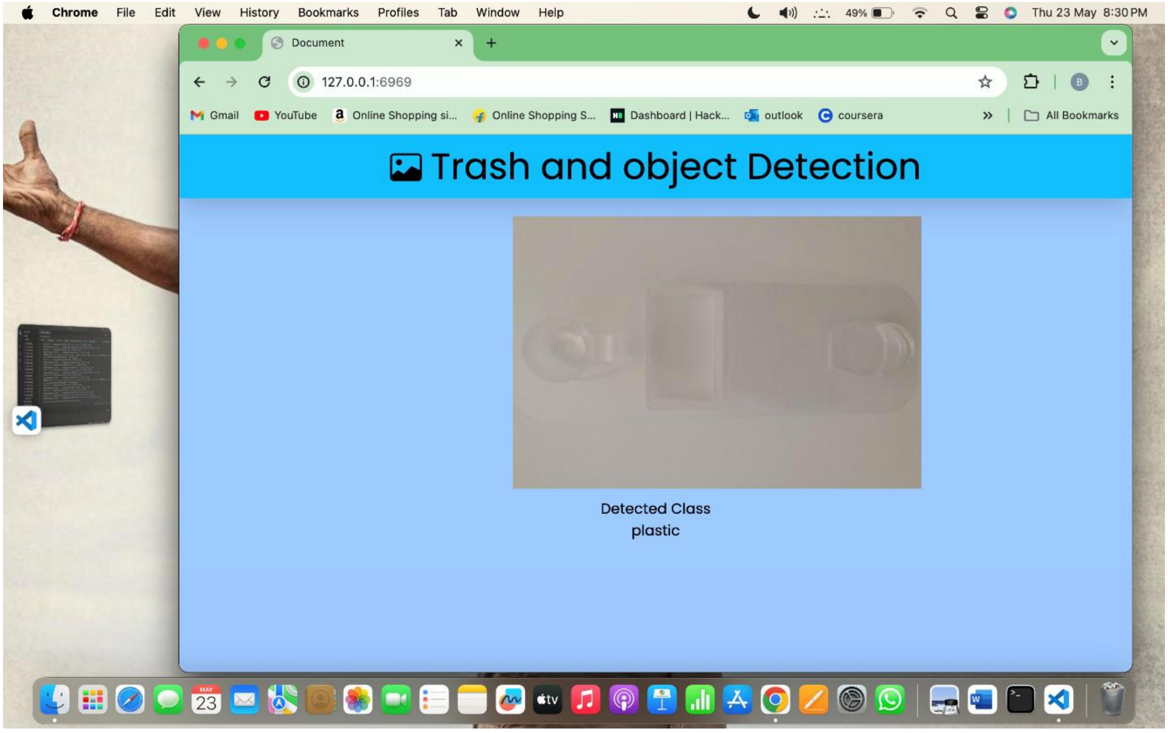

Fig.7: Detecting plastic

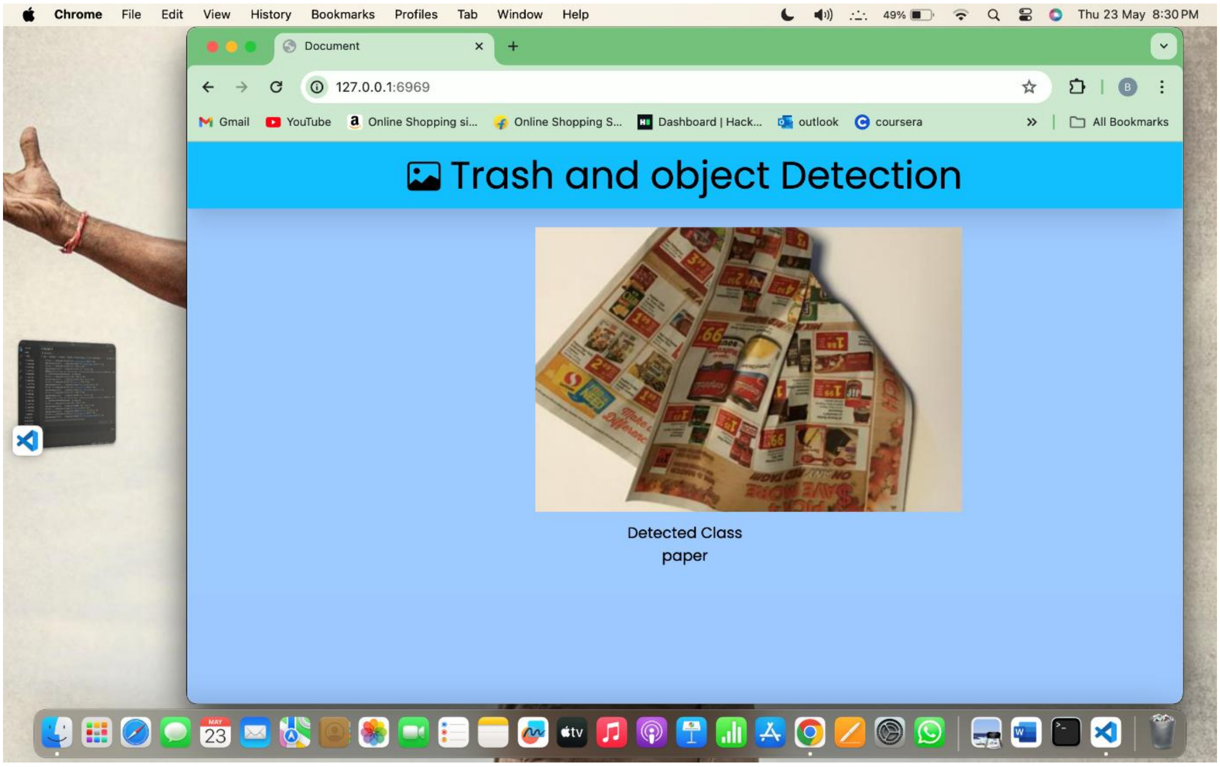

Fig-8: detecting paper

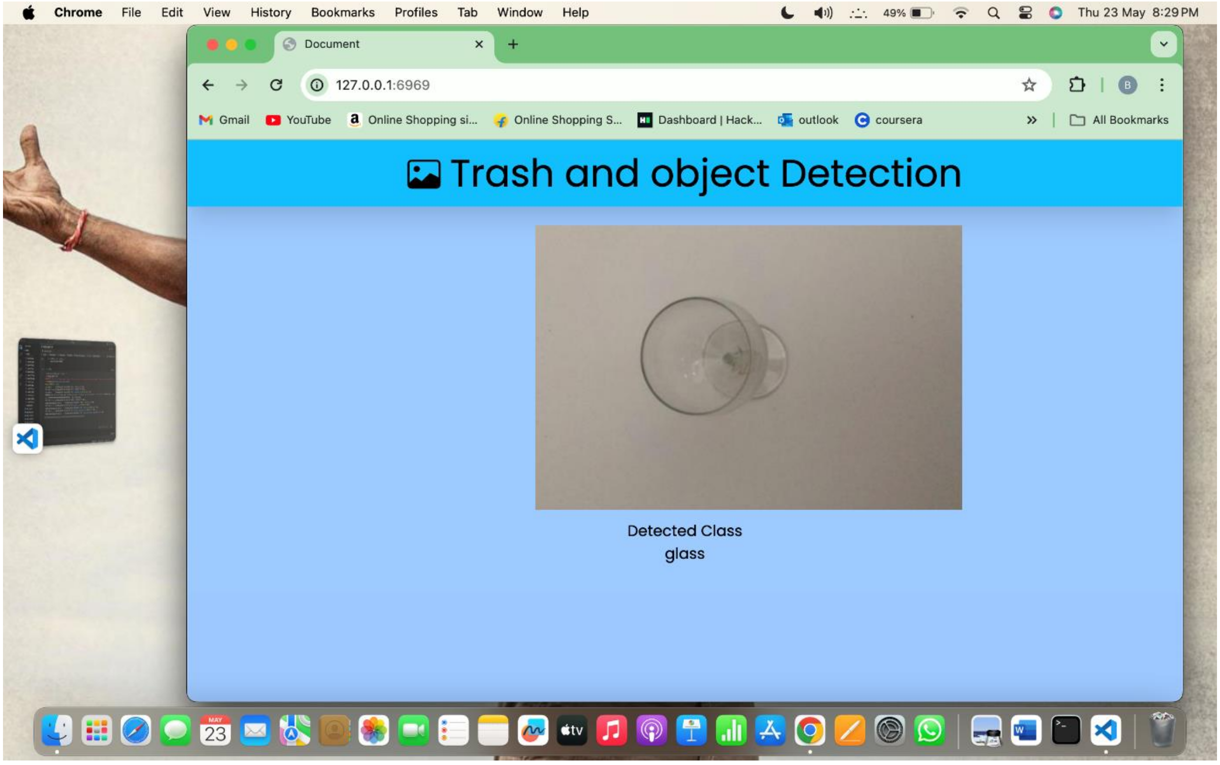

Fig-9: detecting glass

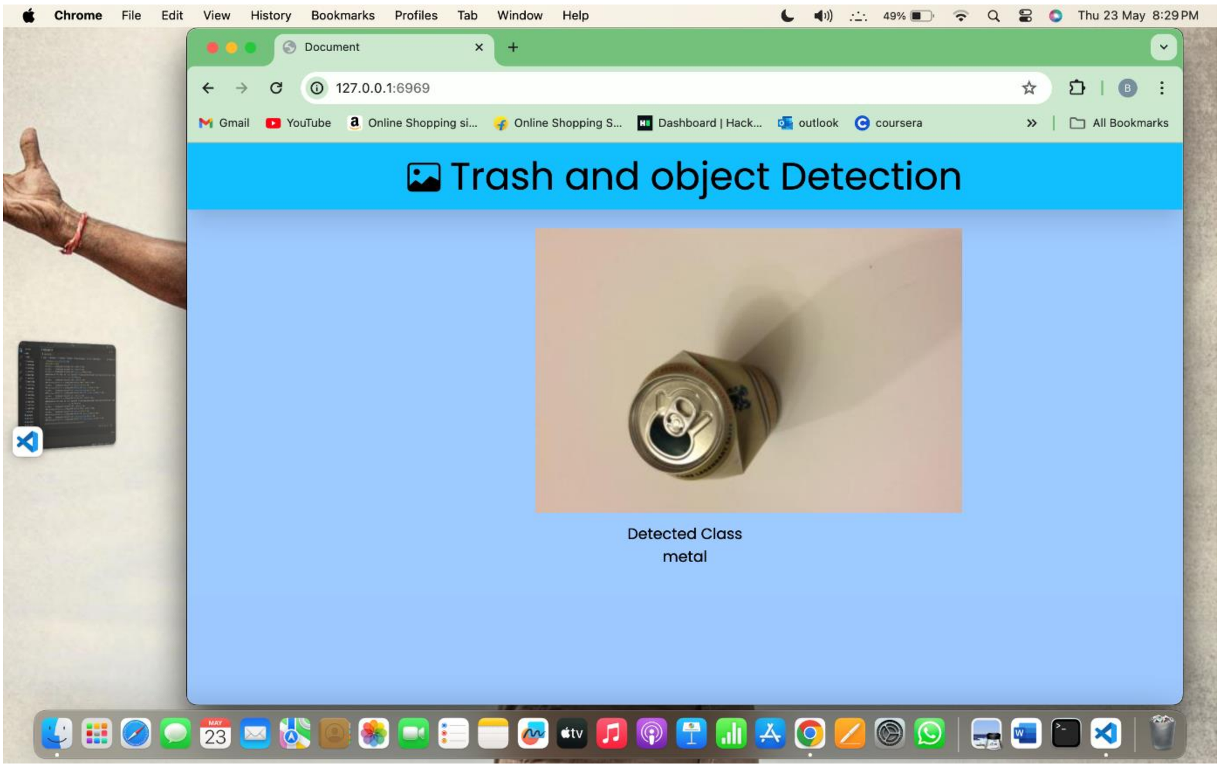

Fig-10: detecting metal

6. RESULTS AND DISCUSSION

Performance metrics:

	Precision	Recall	F1-score	Support	Accuracy
Cardboard	0.94	0.97	0.95	30	0.96
Glass	0.95	0.92	0.94	39	0.92
Metal	1.00	0.94	0.97	36	0.94
Paper	0.95	0.98	0.96	56	0.98
Plastic	0.91	0.88	0.90	49	0.87
Trash	0.88	1.00	0.93	14	1.00
Accuracy					0.94

Table 3: Classification report for Vgg16

We had calculated the performance metrics for each class of the trashnet dataset and mentioned average accuracy, The table shows model performs very well on most classes, achieving a precision, recall, and F1-score close to 1.00 for Cardboard, Metal, Paper, and Plastic. This indicates the model can accurately identify these materials with high confidence. and shows a slight drop in performance for glass

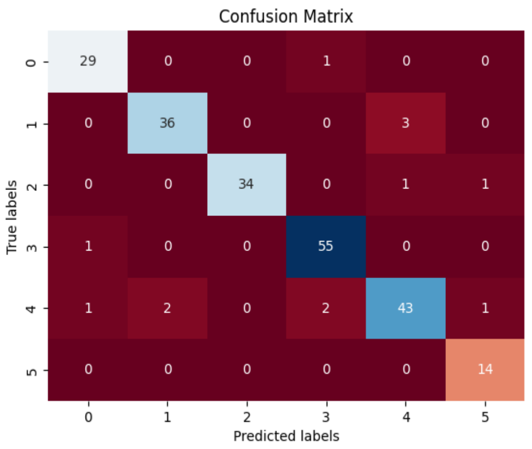

Fig 11.1: Confusion metrics

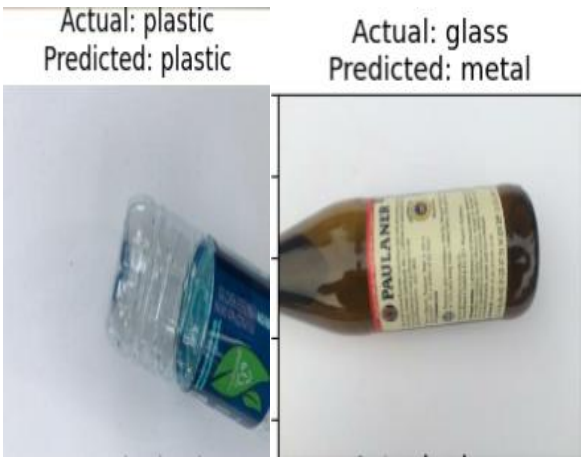

Fig 11.2: Actual and predicted images

The confusion matrix serves as a comprehensive evaluation tool for assessing the performance of a speech emotion classification model. It provides insights into the model's predictive accuracy by displaying the number of correct and incorrect classifications across different emotions. For instance, the values along the diagonal represent the instances where the model correctly classified an emotion, while off-diagonal values indicate misclassifications.

Model	Train Accuracy	Test Accuracy
VGG 16	98%	94.19%
Densenet121	93%	84.82%
MobilenetV3	82.9%	87.05%

Table 4: Experiment results

The table presents the performance metrics of three different models: VGG16, MobilenetV3, and DenseNet121, in terms of their train accuracy and test accuracy.

For VGG16, the train accuracy is 98% and the test accuracy is 94.19%. This indicates that when trained on the dataset, the model correctly classifies 94.19% of the training data and achieves a similar level of accuracy (94.19%) on unseen test data.

In contrast, DenseNet121 achieves a train accuracy of 93% and a test accuracy of 84%. While DenseNet's accuracy is slightly lower compared to the VGG models, it still maintains a respectable level of performance on both training and test datasets.

For MobilenetV3, the train accuracy is slightly higher at 83%, and the test accuracy significantly surpasses at 87%..

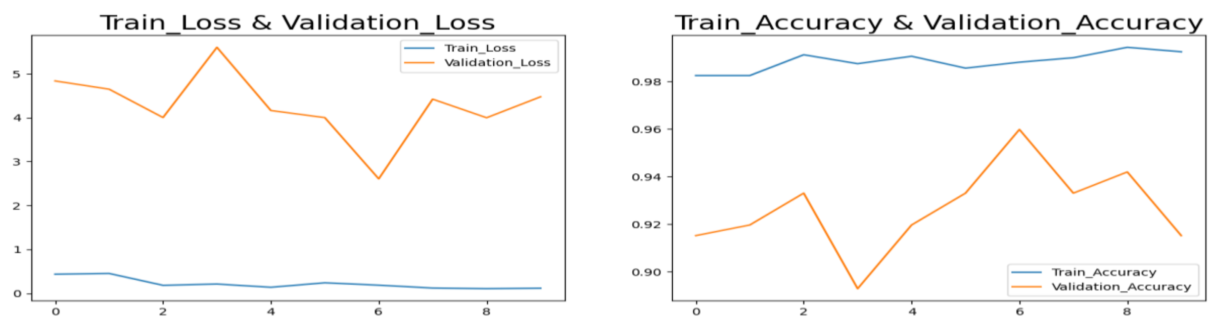

Fig-11.3: plots of the loss and accuracy during training and testing

7. CONCLUSION AND FUTURE SCOPE

In this study, the implementation of the VGG16 model yielded remarkable results, achieving an impressive accuracy of 94% on the test data, outperforming both DenseNet121 (84% accuracy) and MobileNetV3 (87% accuracy). This highlights the model's high efficiency in classifying garbage images, underscoring its effectiveness in waste management applications. The success of VGG16 underscores its potential to contribute significantly to reducing garbage in our surroundings. With its robust classification capabilities, VGG16 holds promise for enhancing waste sorting processes and ultimately promoting cleaner environments.

Moving forward, future work could involve leveraging feature maps to further enhance garbage detection and classification systems. By utilizing tools like OpenCV, a versatile system for detecting various images, including cars, could be developed. With advancements in image detection, this system could be expanded to effectively identify a wide variety of trash, including ocean debris. Additionally, the development of a website or mobile app where users can upload images for classification could be explored. Handheld or portable machines for image detection could also be designed to aid in tasks such as beach garbage collection. Ultimately, these advancements could lead to the creation of robots specifically designed for garbage cleaning, further automating waste management processes and contributing to cleaner surroundings and reduced effort in waste management.

8. REFERENCES

[1] Shi, C., Tan, C., Wang, T., & Wang, L. (2021). A waste classification method based on a multilayer hybrid convolution neural network. Applied Sciences, 11(18), 8572.

[2] Melinte, D. O., Travediu, A. M., & Dumitriu, D. N. (2020). Deep convolutional neural networks object detector for real-time waste identification. Applied Sciences, 10(20), 7301.

[3] Kang, B., & Jeong, C. S. (2023). ARTD-Net: Anchor-Free Based Recyclable Trash Detection Net Using Edgeless Module. Sensors, 23(6), 2907.

[4] Yang, J., Zeng, Z., Wang, K., Zou, H., & Xie, L. (2021). GarbageNet: a unified learning framework for robust garbage classification. IEEE Transactions on Artificial Intelligence, 2(4), 372-380.

[5] Chen, Z., Yang, J., Chen, L., & Jiao, H. (2022). Garbage classification system based on improved ShuffleNet v2. Resources, Conservation and Recycling, 178, 106090..

[6] Wahyutama, Aria Bisma, and Mintae Hwang. "YOLO-based object detection for separate collection of recyclables and capacity monitoring of trash bins." Electronics 11.9 (2022): 1323

[7] Sultana, Rumana, et al. "Trash and recycled material identification using convolutional neural networks (CNN)." 2020 SoutheastCon. IEEE, 2020.

[8] Yang, Jianfei, et al. "GarbageNet: a unified learning framework for robust garbage classification." IEEE Transactions on Artificial Intelligence 2.4 (2021): 372-380.

[9] Teng, X., Fei, Y., He, K., & Lu, L. (2022, July). The Object Detection of Underwater Garbage with an Improved YOLOv5 Algorithm. In Proceedings of the 2022 International Conference on Pattern Recognition and Intelligent Systems (pp. 55-60).

[10] Gondal, A. U., Sadiq, M. I., Ali, T., Irfan, M., Shaf, A., Aamir, M., ... & Kantoch, E. (2021). Real time multipurpose smart waste classification model for efficient recycling in smart cities using multilayer convolutional neural network and perceptron. Sensors, 21(14), 4916.

[11] Xiao, J. (2022, March). A waste image classification using convolutional neural networks and ensemble learning. In Proceedings of the 6th International Conference on Control Engineering and Artificial Intelligence (pp. 29-33).

[12] Ma, Xiaoxuan, Zhiwen Li, and Lei Zhang. "An improved ResNet-50 for garbage image classification." Tehnički vjesnik 29.5 (2022): 1552-1559.

[13] Mittal, Ishika, et al. "Trash classification: classifying garbage using deep learning." Journal of engineering sciences 11.7 (2020).

[14] Pandey, Ayush, et al. "Enhancing Waste Management: Automated Classification of Biodegradable and Non-biodegradable Waste using CNN.

[15] Ramsurrun, Nadish, et al. "Recyclable waste classification using computer vision and deep learning." 2021 zooming innovation in consumer technologies conference (ZINC). IEEE, 2021.

[16] Wang, Y., Zhao, W. J., Xu, J., & Hong, R. (2020). Recyclable waste identification using cnn image recognition and gaussian clustering. arXiv preprint arXiv:2011.01353.

[17 Yang, M., & Thung, G. (2016). Classification of Trash for Recyclability Status; CS229 Project Report.

[18] He, Y., Gu, Q., & Shi, M. (2020). Trash Classification Using Convolutional Neural Networks Project Category: Computer Vision.

[19] Awe, O., Mengistu, R., & Sreedhar, V. Final Report Smart Trash Net Waste Localization and Classification. arXiv 2017. Preprint.

[20] Kulkarni, Hrushikesh N., and Nandini Kannamangalam Sundara Raman. "Waste object detection and classification." CS230 Stanford (2019).