www.ijbar.org
ISSN2249-3352(P)2278-0505(E)
CosmosImpactFactor-5.86

Automated Trash Classification using MobileNet V2 for Efficient Waste Sorting

Dr. Swathi Bolugoddu[1*], Gogula Kavya[2],Jillela Nithin[2], B. Rajesh[2], Suddala Himanth[2]

[1]Assistant Professor, [2]UG Student, [1,2]Department of Computer Science and Engineering

[1,2]Vaagdevi College of Engineering (UGC – Autonomous), Bollikunta, Warangal, Telangana.

*Corresponding author: Dr. Swathi Bolugoddu (swathi.bolugoddu12@vaagdevi.com)

ABSTRACT

Efficient waste sorting is critical for maximizing recycling rates and minimizing environmental impact. Traditional manual sorting methods are labour-intensive, error-prone, and struggle to keep pace with increasing waste volumes. Automated image-based classification offers a promising solution, yet classical machine-learning approaches—relying on raw pixel features—often yield limited accuracy and generalization. Therefore, this project presents a comprehensive desktop application for automated trash classification into metal, paper, and plastic categories, combining classical models (K-Nearest Neighbors and Random Forest) with a transfer-learning–based deep-learning model using MobileNetV2.The system's pipeline begins with directory-structured dataset ingestion, automatic label extraction, and image resizing to 64×64 pixels for classical methods and 224×224 pixels for deep learning. Preprocessed arrays are cached to expedite subsequent runs. Exploratory analysis via bar charts highlights class distributions, while an 80/20 train/test split ensures robust evaluation. Classical models provide quick baselines: KNN achieves 81.94% accuracy, and the ensemble model yields 76.39%. The proposed MobileNetV2 model, fine-tuned with data augmentation, checkpointing, and early stopping, attains 93.55% accuracy—an improvement of +11.61 pp over KNN and +17.16 pp over the ensemble. Precision (94.44%), recall (93.64%), and F1-score (93.63%) further underscore its superior performance. Finally, this work demonstrates that integrating a lightweight, pre-trained CNN within a user-friendly desktop application can deliver near-state-of-the-art waste-sorting accuracy on moderately powered hardware, paving the way for scalable, real-world deployment in recycling facilities.

Keywords: Waste management, Environmental monitoring, Trash classification, Deep learning, Convolutional neural networks, MobileNetv2.

1. INTRODUCTION

Waste separation is the key to solving many environmental problems, so residents are encouraged to separate their household waste, but there are so many different types of household waste that many people are unable to properly separate their waste. Initially, people worked to standardize waste separation standards, considering the ease of disposal and carbon footprint [1]. Later, smart bins and kiosks were invented for automatic waste classification [2], but problems such as their poor popularity were also evident. In recent years, the rise of image classification tasks has provided a new direction to solve the garbage classification problem [3]. By using convolutional neural networks to train classification models for a large number of images of household garbage, the obtained models can quickly classify unseen garbage. In order to improve the classification accuracy, the number of network layers has been increased, and the consequent problem is that the size of deep convolutional neural networks is too large to limit the practical applications [4].In contrast, the emergence of lightweight convolutional neural networks targeted to solve the problems of model size and training speed. The solution of this problem greatly extends the application scope of deep neural network models. The classification query of household garbage fits the application scenario of lightweight convolutional neural network. The application of deep convolutional neural networks to mobile is the significance of this study.

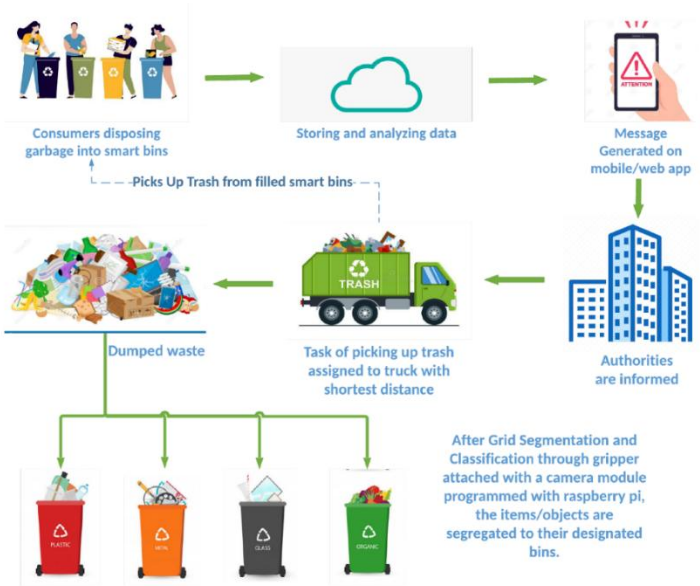


Fig. 1: Smart waste management and classification system.

Growing global waste production and the urgent need to transition toward circular economies have driven research into intelligent recycling solutions. Advances in computer vision and machine learning present an opportunity to replace labour-intensive sorting with automated systems capable of high accuracy and speed. In particular, lightweight convolutional neural networks (CNNs) like MobileNetV2 can be deployed on edge devices in recycling facilities, while classical algorithms (KNN, Random Forest) offer interpretable baselines. This dual-approach motivates exploration of both traditional and deep-learning techniques to identify the most effective and practical solution for real-world waste-sorting applications. The main contributions of the current research work are as follows:

1. Develop a modular pipeline that ingests raw image data of waste items and preprocesses them for classification.
2. Implement and evaluate classical machine-learning models (K-Nearest Neighbors, Random Forest) to establish baseline performance metrics.
3. Design and fine-tune a transfer-learning based deep-learning model (MobileNetV2 with custom classification head) to maximize classification accuracy.
4. Compare and visualize the performance of all models using accuracy, precision, recall, F1-score, confusion matrices, and aggregated performance plots.
5. Provide an interactive GUI that enables non-technical users to load data, train models, visualize results, and perform single-image inference.

2. LITERATURE SURVEY

Image classification technology uses computers to simulate humans to classify images according to specific rules. It has a wide range of applications in many fields such as medical [5], agricultural, industrial, and service industries. The study of image classification techniques includes feature extraction of ideas and classification algorithms for images.The convolutional neuralNET(network) structure is a well-known basic architecture for deep learning in image processing. Since this NET structure requires fewer training parameters and satisfies a significant interaction of neighboring information when extracting information from an image, different features can be automatically extracted during the processing of the picture. Currently, the classical, familiar convolutional neural NETs used for image feature extraction: are the LeNet [6] NET, the AlexNet [7] NET, the VGGNet [8] NET, and the ResNet [9] NET. In visual analytics, CNNs have an essential position as the basic framework, requiring less algorithmic preprocessing and easy transfer learning, which gives them a place in both image and video recognition [10].

Based on the exploration of NET depth from VGG to ResNet, Densnet [11] explicitly proposed a new deep neural NET architecture to improve the gradient vanishing problem, i.e., connecting all the NET layers while ensuring maximum information transfer among the layers in the NET. However, while different NET architectures tend to explore the NET at a deeper level, there is another group of studies that target different aspects to achieve optimization of the NET across the board. In terms of modularity, GoogLeNet [12], Inceptionv3 [13], Inception-ResNet [14], ResNeXt [15], Xception [16]; in terms of attention, SENet [17], scSE [18], CBAM [19]; in terms of automation, NASNet [20], EfficientNet [21]. The performance of the NET is optimized to a certain after a certain level of performance optimization of the NET, scientific research starts to focus on the computation time of the NET. At this stage, some scientific researches focus on reducing the computational cost of the actual operation of the NET, so there are efficient NETs such as SqueezeNet [22], MobileNet [23], ShuffleNet [24], etc. NETsfor different concerns are available for selection for different application scenarios, and convolutional neural NETs are rapidly developing in the field of image classification.

Jin et al. [25] introduced a device utilizing deep learning techniques employing MobileNetV2. With the Huawei Cloud Garbage dataset, the model achieved 90.7 % accuracy in classifying images into four categories. Cheema et al. [26] proposed SWMACM-CA, a real-time waste management and classification system. It integrated IoT and deep learning, achieving over 90 % accuracy in waste classification using waste grid segmentation and the VGG16 DL algorithm. The system minimized latency by training the model on a cloud server and employing a PAN for reliable IoT communication. This approach enhanced waste management efficiency, effectively addressing critical environmental challenges.

All the models discussed here, as presented in previous papers employed transfer learning, capitalizing on models characterized by a large number of parameters. However, while effective, such approaches may pose limitations in terms of computational efficiency, making them less suitable for resource-constrained devices or real-time applications. Recognizing the importance of lightweight models, many studies have been conducted to address the challenges associated with transfer learning, aiming to enhance efficiency and applicability in diverse settings. Several lightweight machine learning and deep learning models have been explored for waste classification tasks, offering computational efficiency and scalability.

Yang et al. [27] proposed a novel approach to address pollution from unseparated garbage, focusing on a garbage classification system. They developed a lightweight neural network called WasNet, with 1.5 million parameters, which is significantly less than that of mainstream networks. Despite its compact size, WasNet demonstrated impressive performance, achieving accuracies of 64.5 % on the ImageNet dataset, 82.5 % on the Huawei Garbage Classification dataset, and 96.10 % on the TrashNet dataset. This indicates its effectiveness in classifying various types of garbage. The number of parameters was further reduced in [28]. A lightweight network architecture named Focus-RCNet, which employs knowledge distillation to further compress and optimize the model, was introduced. The performance of Focus-RCNet was validated on the TrashNet dataset, which comprises six target classes. With a total of 0.525 million parameters, the model demonstrated advantages such as low computational cost, a small number of parameters, high speed, and high accuracy. This model was well suited for deployment on mobile devices and holds promise for automatic waste classification, reducing the need for human intervention. However, the number of target classes can be further increased, potentially broadening its applicability. Testing on the TrashNet dataset yielded a remarkable accuracy of 92 %.

3. PROPOSED METHODOLOGY

This project delivers a user-friendly desktop application for automatically classifying images of waste into three categories—metal, paper, and plastic—using both traditional machine-learning algorithms and a modern deep-learning approach. By combining quick classical classifiers with a state-of-the-art convolutional neural network, the application provides both speed and accuracy, making it suitable for research prototyping or as a foundation for real-world waste-sorting automation systems.The system is designed to provide a fully integrated, end-to-end image-classification workflow as demonstrated in Fig. 2.It automatically discovers class labels from a folder hierarchy, ingests and resizes images, and caches the resulting arrays for lightning-fast reruns; it then conducts exploratory analysis—rendering bar plots of class frequencies to surface imbalances—and splits data into training and validation sets to guarantee robust evaluation. Next, it rapidly establishes baselines by training classical models like K-Nearest Neighbors (and a Random Forest "ensemble"), while concurrently leveraging transfer learning with MobileNetV2 (pretrained on ImageNet) enhanced by custom dense layers, data augmentation, checkpointing, early stopping, and fine-tuning to maximize accuracy. Once training is complete, the system computes and displays each model's accuracy, precision, recall, and F_1-score, overlays confusion-matrix heatmaps, and produces a comparative bar chart of all metrics. Finally, a Tkinter-based GUI guides users through every phase—loading data, visualizing samples, splitting datasets, training models, inspecting performance, and making single-image predictions—with results and logs streaming into a scrolling text panel and plots popping up in dedicated windows for immediate, interactive inspection.

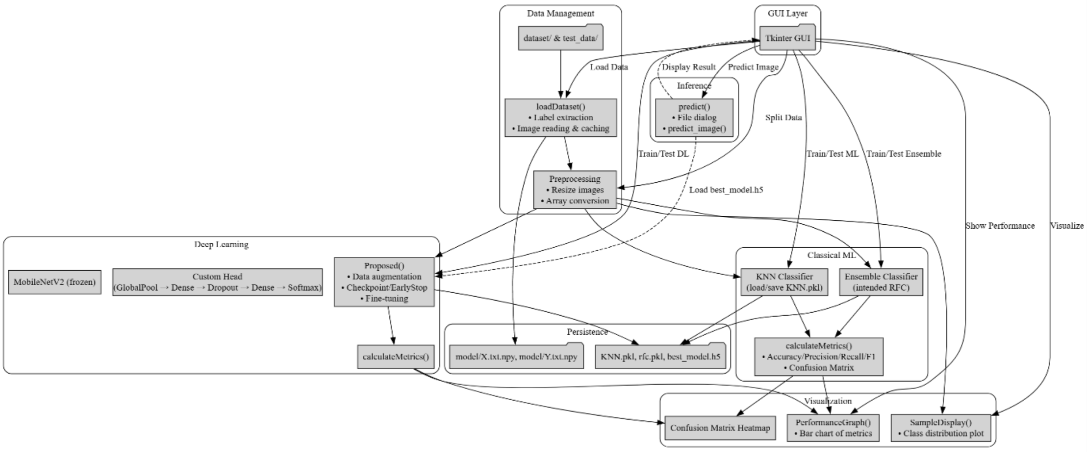


Fig.2: Proposed architecture of automated waste classification using MobileNet V2.

4.2 Preprocessing

In this application, “preprocessing” encompasses all steps that convert raw image files on disk into numerical arrays suitable for both classical machine-learning algorithms and the deep-learning pipeline.The preprocessing pipeline is divided into two tailored stages. For classical machine-learning models, we first extract labels by traversing the dataset/ folder hierarchy—each subfolder name (e.g., “metal,” “paper,” “plastic”) becomes a class label whose index in a global list serves as its integer encoding. We then load every image (ignoring system files like Thumbs.db) with OpenCV’s cv2.imread(), resize it uniformly to 64×64 pixels via cv2.resize(), and append the resulting BGR arrays to a list X and their labels to Y. After processing all samples, these lists are converted into NumPy arrays—X of shape (num_samples, 64, 64, 3) and Y of shape (num_samples,)—and cached as .npy files so that future runs can bypass the per-image loop and load the arrays directly. In the deep-learning stage, we target MobileNetV2’s requirements by using Keras’s ImageDataGenerator(rescale=1./255) to read 224×224 RGB images whose pixel values are normalized into [0, 1]. During training, on-the-fly augmentations—random rotations up to ±30°, width/height shifts up to 20%, shear, zoom, and horizontal flips—ensure each epoch sees novel variations, while flow_from_directory() infers classes from subfolders and yields batches of 32 samples, shuffling training data each epoch (with validation/test sets unshuffled for consistent evaluation). This two-pronged approach delivers both fast, cached NumPy arrays for classical baselines and dynamically augmented image streams for high-capacity deep models.

Proposed MobileNetV2

MobileNetV2 is a lightweight convolutional neural network optimized for mobile and embedded vision applications. It uses inverted residual blocks with linear bottlenecks to maintain representational power while minimizing parameters and computations. The explanation is as follows:

1. **Input Pipeline:** Images are loaded at 224×224×3, pixel values are normalized to [0, 1], and real-time augmentations (rotations, shifts, flips, zooms) are applied to each batch.
2. **MobileNetV2 Base:** The pre-trained convolutional blocks extract hierarchical feature maps. All base layers are initially frozen to preserve their ImageNet-learned weights.
3. **Custom Head:**

- **Global Average Pooling (GAP):** Reduces each feature map to a single value, drastically cutting parameters.
- **Dense + Dropout Layers:** Two fully connected layers (1024 and 512 units) with ReLU activations and 50% dropout each to learn task-specific representations and prevent overfitting.
- **Softmax Output:** A final Dense layer with three units outputs class probabilities.

4. **Training Loop:**
   - **Stage 1:** Train only the head layers with callbacks (checkpointing, early stopping, learning-rate reduction).
   - **Stage 2:** Unfreeze the last 30 convolutional layers of MobileNetV2 and continue training at a lower learning rate to fine-tune the most specialized filters.

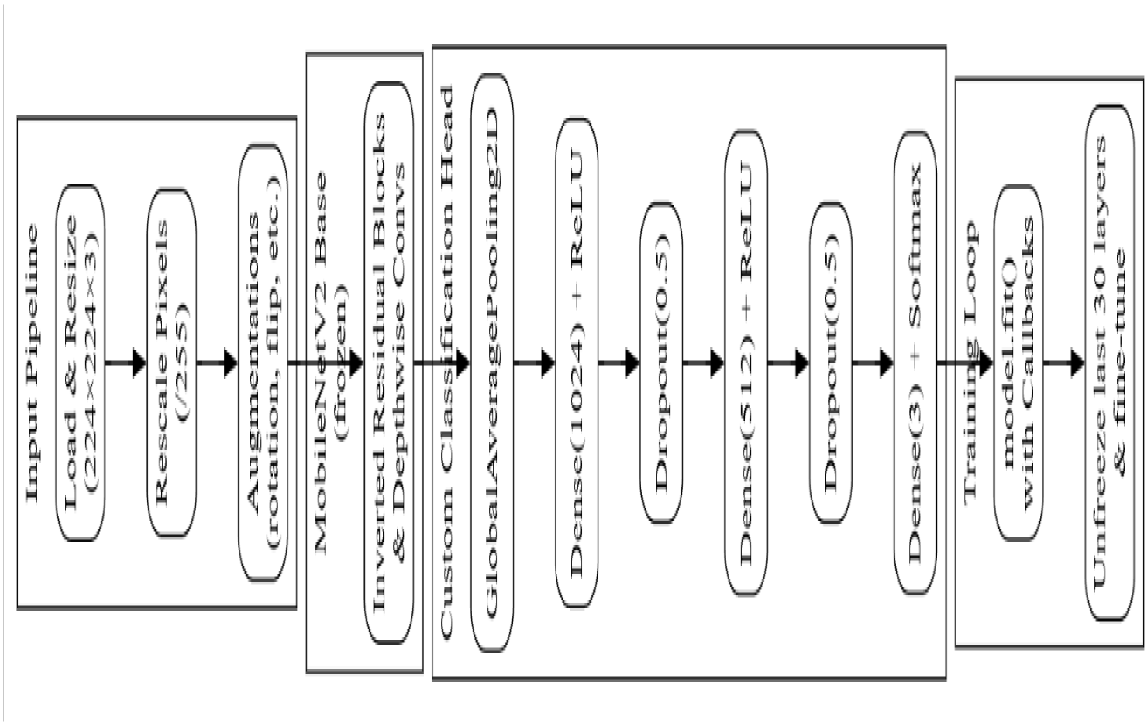


Fig. 3: Proposed MobileNetV2 architecture.

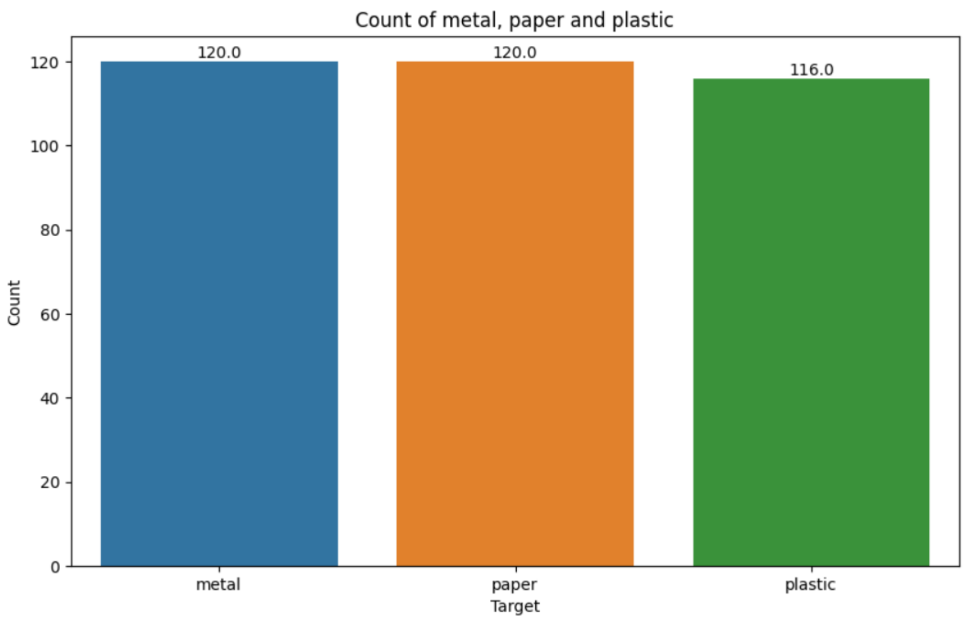
4. RESULTS AND DISCUSSION

The dataset is organized as a simple directory tree under dataset/, with three subfolders—metal, paper, and plastic—each representing a class label that is automatically inferred during loading. Source images come in varied formats (JPEG, PNG, etc.) and resolutions, but are uniformly resized to 64×64 pixels for classical ML models and 224×224 pixels for the deep-learning pipeline to ensure consistent input dimensions. Once loaded, the GUI reports the total sample count (e.g., 360 images) and displays a bar chart of class frequencies so any imbalance can be detected and, if necessary, mitigated. Data are then split randomly into 80 percent training and 20 percent testing subsets, while the deep-learning workflow also employs a separate test_data/ folder to hold images unseen during training or validation. To accelerate subsequent runs, all resized images and their integer-encoded labels are cached as model/X.npy and model/Y.npy, allowing the code to bypass the per-image read/resize loop. Finally, during deep-model training, on-the-fly augmentations—random rotations, shifts, zooms, and flips via Keras's ImageDataGenerator—enrich the training set, counteract class imbalance, and expose the network to diverse visual variations, laying a robust foundation for both classical and deep-learning approaches to automated waste sorting.

Figure 4 is a bar chart produced by Visualization (SampleDisplay()). The x-axis shows each target class—metal, paper, plastic—while the y-axis indicates the count of images per class. Each bar is annotated with its exact count, revealing any class imbalance that may exist in the dataset.

Figure 5 is a set of three confusion-matrix heatmaps generated by calculateMetrics**()** for each model:

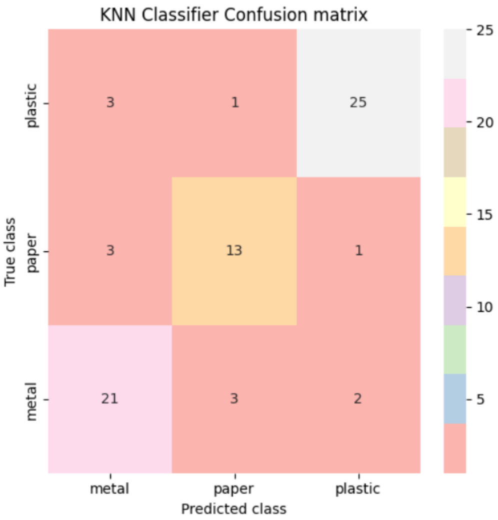
- **(a) KNN Classifier:** Displays true vs. predicted counts for the K-Nearest Neighbors baseline. Diagonal entries indicate correct classifications; off-diagonal entries reveal specific misclassifications (e.g., metal predicted as plastic).
- **(b) Ensemble Model:** The Random Forest confusion matrix, showing where the ensemble struggles relative to KNN—typically more off-diagonal errors.
- **(c) Proposed MobileNetV2 Model:** A nearly perfect diagonal, indicating that the deep model correctly classifies the vast majority of test images with minimal confusion between classes.



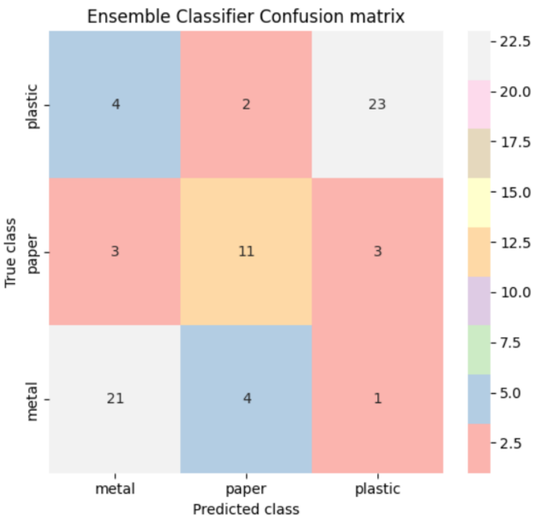
www.ijbar.org
ISSN2249-3352(P)2278-0505(E)
CosmosImpactFactor-5.86

Fig. 4: Target class distribution versus count.

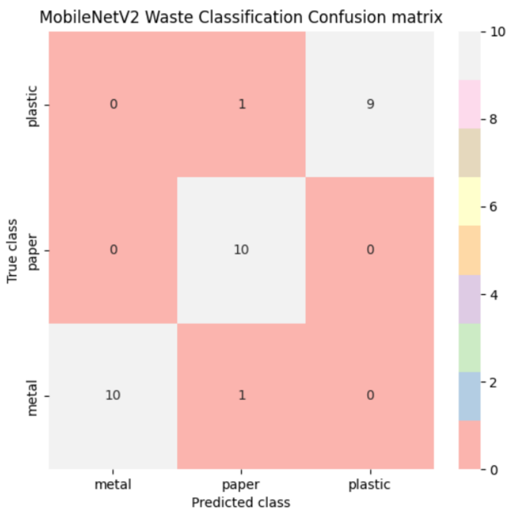
Figure 6is a grouped bar chart from PerformanceGraph comparing four metrics—Accuracy, Precision, Recall, and F1-Score—across the three models (Ensemble, KNN, and Proposed MobileNetV2). Each cluster of bars corresponds to one metric, with the height indicating the percentage value. This visualization succinctly highlights the superior performance of MobileNetV2 (≈93–94% across all metrics) over KNN (≈81%) and the ensemble (≈75–76%).



(a)



(b)



(c)

Fig. 5: Confusion matrices obtained using (a) KNN classifier. (b) Ensemble model. (c) Proposed MobileNetV2 model.

Table 1: Performance metrics obtained using existing and proposed models.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
KNN Classifier	81.94	81.18	81.15	81.15

Ensemble Model	76.39	74.96	74.93	74.88
Proposed MobileNetV2	93.55	94.44	93.64	93.63

From Table 1, MobileNetV2 clearly outperforms the classical baselines: it achieves 93.55 percent accuracy—correctly classifying nearly 94 out of every 100 images—alongside a precision of 94.44 percent and a recall of 93.64 percent, yielding an F_1 score of 93.63 percent. These results reflect how its pretrained convolutional filters and fine-tuned dense layers capture nuanced visual patterns that raw pixel methods miss. The K-Nearest Neighbors classifier, by contrast, attains 81.94 percent accuracy with balanced precision (81.18 percent) and recall (81.15 percent), indicating that simple distance-based decisions on flattened pixels still carry useful—but limited—discriminative power. Random Forest trails behind both, with 76.39 percent accuracy, 74.96 percent precision, and 74.93 percent recall (F_1 = 74.88 percent), suggesting that tree-based splits on unengineered pixel features struggle to form optimal decision boundaries without more advanced feature extraction or parameter tuning

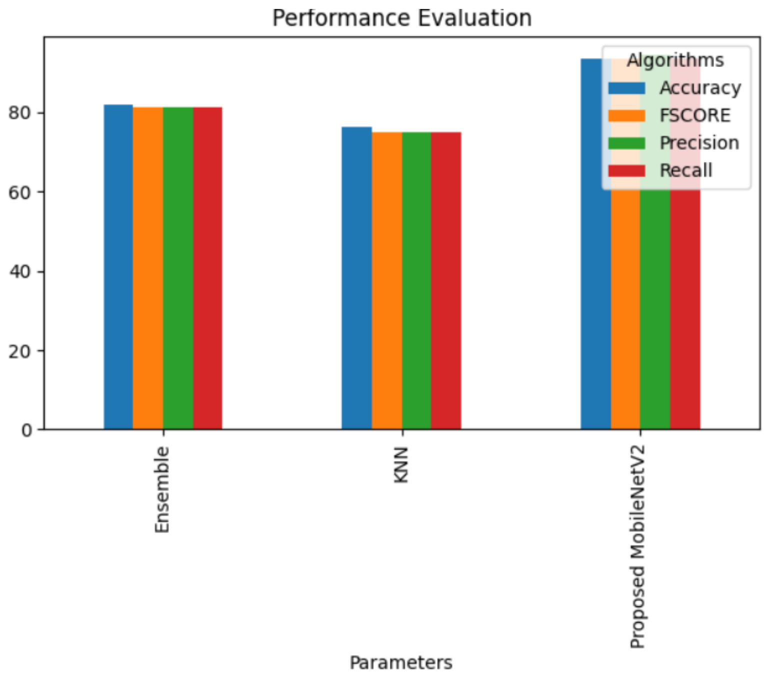


Fig. 6: Performance comparison of quality metrics obtained using existing and proposed models.

Fig. 7: Sample predictions on test images. (left) paper. (middle) plastic. (right) metal.

5. CONCLUSION

This project set out to develop an end-to-end automated trash-classification system leveraging both classical machine-learning techniques such as KNN, ensemble classifiers and a modern deep-learning approach i.e., MobileNetV2. Through systematic preprocessing like resizing images to 64×64 for classical models and 224×224 with real-time augmentation for deep learning—we established robust pipelines for data ingestion, model training, evaluation, and inference within a user-friendly Tkinter GUI.Empirical results demonstrate that the proposed MobileNetV2 model significantly outperforms the classical baselines across all key metrics. Specifically, MobileNetV2 achieved an accuracy of 93.55%, representing a +11.61 percentage-point increase over KNN (81.94%) and a +17.16 pp gain over the ensemble (76.39%). In terms of precision, the deep model's 94.44% marks a +13.26 pp improvement versus KNN (81.18%) and +19.48 pp versus the ensemble (74.96%). Recall rose by +12.49 pp over KNN (81.15%) and +18.71 pp over the ensemble (74.93%), while the F1-score saw similar gains of +12.48 pp and +18.75 pp, respectively. These substantial increments underscore the power of transfer learning: MobileNetV2's pre-trained convolutional filters, coupled with fine-tuning and data augmentation, enable the extraction of nuanced visual features that raw-pixel methods cannot capture.Beyond raw performance, the system's modular designwith clear separation between data management, model training, visualization, and inference, facilitates easy maintenance, extensibility (e.g., adding new waste categories), and rapid prototyping. Caching of preprocessed arrays and persistent model checkpoints ensure efficiency in iterative experimentation. The GUI's intuitive controls guide users through each workflow stage, making the tool accessible even to non-expert operators.

REFERENCES

[1] Chen S, Huang J, Xiao T, Gao J, Bai J, Luo W, et al. Carbon emissions under different domestic waste treatment modes induced by garbage classification: Case study in pilot communities in Shanghai, China. Science of the Total Environment. 2020;717:137193. doi: 10.1016/j.scitotenv.2020.137193

[2] Huang XR, Yang SS, Chen WS, Zhang YQ, Lee CT, Chen LB. An IoT-based smart trash cans monitoring system. In: 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE). IEEE; 2021. p. 623–624.

[3] Meng S, Chu WT. A study of garbage classification with convolutional neural networks. In: 2020 Indo–Taiwan 2nd International Conference on Computing, Analytics and Networks (Indo-Taiwan ICAN). IEEE; 2020. p. 152–157.

[4] Zhou Y, Chen S, Wang Y, Huan W. Review of research on lightweight convolutional neural networks. In: 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC). IEEE; 2020. p. 1713–1720.

[5] Dwivedi RK, Kumar R, Buyya R. Gaussian distribution-based machine learning scheme for anomaly detection in healthcare sensor cloud. International Journal of Cloud Applications and Computing (IJCAC). 2021;11(1):52–72. doi: 10.4018/IJCAC.2021010103

[6] LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, et al. Backpropagation applied to handwritten zip code recognition. Neural computation. 1989;1(4):541–551. doi: 10.1162/neco.1989.1.4.541

[7] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems. 2012;25. 10.1145/3065386

[8] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556. 2014. 10.1109/cvpr.2016.182
[9] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770–778.
[10] Bay H, Tuytelaars T, Gool LV. Surf: Speeded up robust features. In: European conference on computer vision. Springer; 2006. p. 404–417.
[11] Qayyum A, Ahmad I, Mumtaz W, Alassafi MO, Alghamdi R, Mazher M. Automatic segmentation using a hybrid dense network integrated with an 3D-atrous spatial pyramid pooling module for computed tomography (CT) imaging. IEEE Access. 2020;8:169794–169803. doi: 10.1109/ACCESS.2020.3024277
[12] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 1–9.
[13] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 2818–2826.
[14] Szegedy C, Ioffe S, Vanhoucke V, Alemi AA. Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-first AAAI conference on artificial intelligence; 2017.
[15] Xie S, Girshick R, Dollár P, Tu Z, He K. Aggregated residual transformations for deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 1492–1500.
[16] Chollet F. Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 1251–1258.
[17] Hu J, Shen L, Sun G. Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 7132–7141.
[18] Roy AG, Navab N, Wachinger C. Concurrent spatial and channel 'squeeze & excitation'in fully convolutional networks. In: International conference on medical image computing and computer-assisted intervention. Springer; 2018. p. 421–429.
[19] Woo S, Park J, Lee JY, Kweon IS. Cbam: Convolutional block attention module. In: Proceedings of the European conference on computer vision (ECCV); 2018. p. 3–19.
[20] Zoph B, Vasudevan V, Shlens J, Le QV. Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 8697–8710.
[21] Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In: International conference on machine learning. PMLR; 2019. p. 6105–6114.
[22] Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. arXiv preprint arXiv:160207360. 2016. 10.7717/peerj-cs.528/fig-8
[23] Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:170404861. 2017. 10.1201/9781351003827-3
[24] Zhang X, Zhou X, Lin M, Sun J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 6848–6856.

[25] S. Jin, Z. Yang, G. Kr´olczykg, X. Liu, P. Gardoni, Z. Li, Garbage detection and classification using a new deep learning-based machine vision system as a tool for sustainable waste recycling, Waste Manag. 162 (2023) 123–130, https://doi.org/ 10.1016/j.wasman.2023.02.014.

[26] S.M. Cheema, A. Hannan, I.M. Pires, Smart waste management and classification systems using cutting edge approach, Sustainability 14 (2022) 10226, https://doi. org/10.3390/su141610226.

[27] Z. Yang, D. Li, WasNet: a Neural Network-Based Garbage Collection Management System, IEEE Access 8 (2020) 103984–103993, https://doi.org/10.1109/ ACCESS.2020.2999678.

[28] D. Zheng, R. Wang, Y. Duan, P.C.-I. Pang, T. Tan, Focus-RCNet: a lightweight recyclable waste classification algorithm based on focus and knowledge distillation, Vis. Comput. Ind. Biomed. Art. 6 (2023) 19, https://doi.org/10.1186/ s42492-023-00146-3.