Journal of Engineering Science and Technology
Vol. 20, No. 6 (2025) 1877 - 1894

DESIGN OF SMART WASTE SORTING SYSTEM USING DEEP LEARNING TECHNIQUES

MANICKAM RAMASAMY[1], S. DEIVASIGAMANI[1], MOHAMMAD ARIF BIN IIYAS[1], A.THANKARAJ[2], C. SENTHILPARI[3,*], B. SAKTHIVEL[4], K. GOWRISHANKAR[5]

[1]Faculty of Engineering, Technology & Build Environment, UCSI University, Kuala Lumpur, Malaysia
[2]Department of Electrical and Electronics Engineering, Rrase College of Engineering, Chennai, India
[3]Faculty of Engineering, Multimedia University, Cyberjaya, Selangor DE, Malaysia
[4]Department of Electronics and communication Engg, Pandian Saraswati Yadav College of Engineering, Sivagangai Dist.,Tamilnadu, India
[5]Dept.of Marine Engineering, AMET University, Tamilnadu, India
*Corresponding Author: c.senthilpari@mmu.edu.my

Abstract

Due to the exponential growth in garbage production worldwide, garbage management has emerged as one of the planet's most critical challenges. The effectiveness of waste classification and sorting systems, which are an integral part of waste management, is vital to recycling programs. However, designing an automated system for waste classification is a significant difficulty in waste management. The experiments were performed in five phases: dataset preparation, training the DCNN models for waste classification, evaluation of the trained framework, model performance comparison, and real-time performance analysis. This study assesses the efficacy of multiple Deep Convolutional Neural Network (DCNN) models in waste material classification. The goal will be identifying the optimal Deep Learning (DL) technique for solid waste classification. Following the training of three pre-trained DCNN models on a trash image dataset, the most accurate model was chosen for evaluation with our prototype garbage sorting device. The DCNN models were trained using a publicly available collection of waste image datasets, enabling automated waste classification into distinct categories such as plastic, glass, paper, and metal. Standard performance criteria, such as accuracy, precision, recall and F1-score, were employed to assess the trained DCNN models. Further, the trained model was tested in real-time using an in-house-developed prototype waste sorting system model. The experiential results show that the trained DCNN-based waste classification model scored an average of 98% waste classification accuracy for real-time evaluation.

Keywords: Convolutional neural network, Gaussian clustering, Image classification, Waste recycling.

1. Introduction

In the 21st century, there has been a substantial increase in waste production, with just 19% of the total 2.01 billion metric tons of municipal solid waste being successfully recycled globally. Recycling reduces waste production and decreases the need for landfills, which can cause problems owing to their spatial requirements and the possibility of contaminating soil or groundwater. Efficient recycling operations rely on the categorization of waste according to its composition. Waste can be classified and separated using manual, semi, and fully automated sorting, which employ different approaches. The first type depends entirely on the human capacity to sort, and the second type focuses on machine-assisted strategies for waste sorting.

In recent years, deep learning and computer vision technologies have demonstrated significant potential in automating waste sorting systems. Yin et al. [1] utilised deep learning methodologies for robotic table cleaning, whereas Meena Malik et al. [2] investigated DCNNs for garbage classification to promote sustainable development. Gómez et al. [3] introduced a self-reconfigurable robot for garbage detection and adaptive cleaning utilising deep learning, while Rahman et al. [4] developed an intelligent IoT-integrated waste management system. Uppala et al. [5] developed a bespoke 7-layer CNN for waste classification, whereas Arbeláez-Estradaet al. [6] performed an extensive review of automated waste identification systems. These findings indicate the increasing prevalence of deep learning in semi- and fully-automated waste sorting. Convolutional Neural Network (CNN) is the backbone of Deep Learning algorithms [7, 8]. It has a superior ability to learn features from images, delivering astonishing results in image classification tasks. It was widely studied in waste classification applications.

A computer-vision-based robot grasping system for automatic sorting waste was proposed by Zhihong et al. [9]. A Fast RCNN is used to recognize various objects in the image. There have been reports of customized CNN architecture for specific object-detection applications and debris categorization [10, 11] Customized CNN layers, however, significantly impact object-detection accuracy and are ineffective when the outcomes of two item classes are highly similar. A smartphone application was developed by Mittal et al. [12] to locate and identify outdoor waste. The debris data set was acquired using Bing image search, and the authors used an AlexNet CNN model that had been trained beforehand to recognize garbage in outdoor shots.

The model's classification accuracy for this application is 87%. However, their approach lacks specificity since it employs segmentation to identify locations rather than providing waste-type information. Another method in [12] is sorting waste into recycling categories using an SVM with SIFT characteristics. This approach does not identify the location of the waste; instead, it solely uses cropped images of solid waste for classification, which is satisfied SDG 9 and 11 . For the task at hand, the accuracy of the technique is 94%. Rad et al. [13] used the garbage identification method in outdoor areas to train the overfeat Google Net model. The authors trained CNN to detect solid detritus found in outdoor areas, such as leaves, newspapers, food packaging, cans, and other items, using 18,672 images of diverse litter and waste. In this case, the scheme's precision for debris detection was 68.27%. CNN architectures have also been applied to FLS image-based underwater debris detection [14].

Images of typical maritime debris were used to train this model. Fulton et al.'s recent evaluation of several DL frameworks for use in autonomous underwater vehicles' marine waste detectors was published in [15]. The authors acknowledge that, compared to the YOLOv2 and Tiny YOLO frame- works, CNN and SSD exhibit superior accuracy. SSD, how- ever, demands longer processing times. As mentioned earlier, the literature proves that DL-based frameworks play a vital role in waste detection and classification in various applications. Identifying the suitable DCNN model, training the model with an optimal waste image dataset, and fine tuning the model are the key challenges in designing the DL- based automated waste sorting field [16].

This work proposes the DCNN-based intelligent waste categorization system. The proposed system combines the two open-source waste image datasets to train DCNN models, which can automatically classify waste into different categories, such as plastic, glass, paper, and metal. After training three DCNN models, the most accurate model was selected for our suggested garbage sorting unit. The proposed system was evaluated using standard performance metrics and ensured that the proposed DCNN-based waste classification system has great potential for real-world recycling applications, helping improve waste management efficiency and effectiveness efforts. This work proposes the DL-based classification and optimized feature selection method and trained the data's [17, 18].This research presents a smart waste segregation system that combines capacitive sensing and inductive proximity sensors [19] with Machine Learning (ML) algorithms. The model addresses issues such as limited trash image categories and insufficient datasets, proposing strategies to improve machine learning model performance across a variety of waste classes [20]. This study addresses the creation of an intelligent garbage classification system that uses CNNs. By analysing photos of solid garbage, the system automatically categorises into glass, paper, metallic, and plastic, demonstrating the usefulness of deep neural networks in waste management applications [21].

Although numerous advancements in automated trash classification, a deficiency persists in dependable, high-accuracy systems capable of functioning efficiently in real-world environments with constrained, heterogeneous datasets. Numerous current models are not tailored for implementation in real-time embedded systems and frequently struggle to generalise effectively across various waste categories. This work aims to develop a reliable, precise, and efficient deep learning-based waste classification system suitable for integration into an economical smart sorting prototype.

This paper is arranged into the following sections. Section 1 involves the introduction, objective, and problem statement. Section 2 involves the related works. Section 3 has expressed the design idea methodology and simulation. Section 4 involves the simulation results and justification for the design.

2. Units

Figure 1 shows the overview of the proposed system. It comprises two functional modules: a waste classification module and an electronic-controlled waste sorting mechanism. The functionality of both modules is described as follows:

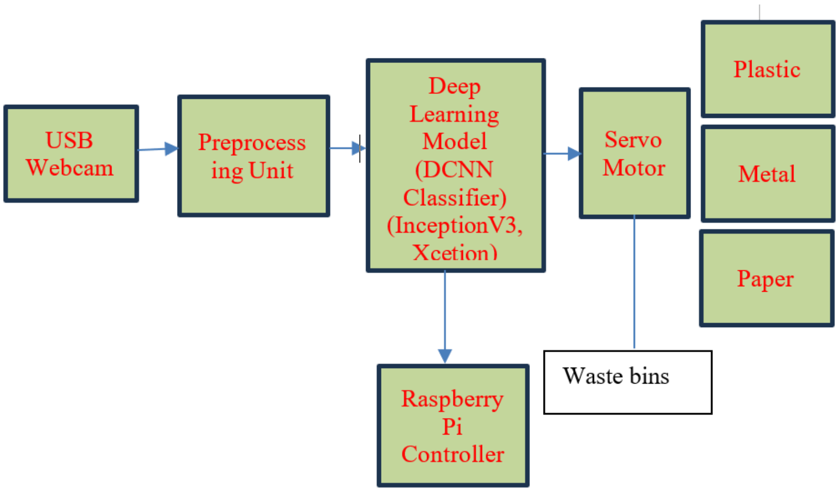

Fig. 1. Block diagram of the waste classification system.

2.1. Waste classification module

The waste classification module executes the waste classification task. It comprises two functional modules: an imagine unit and DCNN-based waste classification algorithms. The waste classification module takes the input from the USB webcam and performs the classification task. A USB webcam captures the image at 2.1 megapixels. Three DCNN algorithms, including Inception V3, Xception, and Ensemble models, were trained for the waste classification task, and their performance was analysed. The model with the best accuracy was chosen to be implemented in our proposed waste sorting unit. The details of the three DCNN frameworks are described below.

The waste sorting prototype utilised a Raspberry Pi 4B as the central controller, interfaced with a Logitech C270 USB camera for real-time image acquisition. The InceptionV3 classification model was deployed on the Raspberry Pi utilising TensorFlow Lite for enhanced inference efficiency. Following classification, the Raspberry Pi activates a standard SG90 servo motor linked to an arm mechanism that accurately directs the waste item into the designated bin according to the classification label. The prototype comprises five designated bins that align with the waste categories: plastic, metal, paper, cardboard, and mixed waste. The servo motor receives power from an external 5V power supply, enabling the system to function autonomously. The integration was accomplished utilising Python scripts executed on the Raspberry Pi, interfacing with the servo motor via GPIO pins. The prototype underwent testing in a controlled laboratory setting to assess its operational feasibility. Figures 1 and 5 depict the system architecture and prototype setup, respectively.

2.2. Inception V3

Google invented Inception V3. It is primarily designed for image classification tasks. Figure 2 illustrates the DCNN architecture of the Inception framework. The key innovation of Inception V3 lies in its use of “inception modules” which are small convolutional neural network modules that perform parallel processing at different scales. These modules allow the network to efficiently capture features at multiple resolutions, leading to better performance while minimizing computational costs. Inception modules comprise convolutional filters, including 1×1, 3×3, and 5×5 convolutions and max pooling operations. The network can effectively capture features at various scales by performing these operations in parallel and concatenating the results. Further, Inception V3 employs factorized convolutions, such as separable convolutions, which reduce the number of parameters in the network while maintaining expressive power. This helps reduce computational requirements and memory footprints.

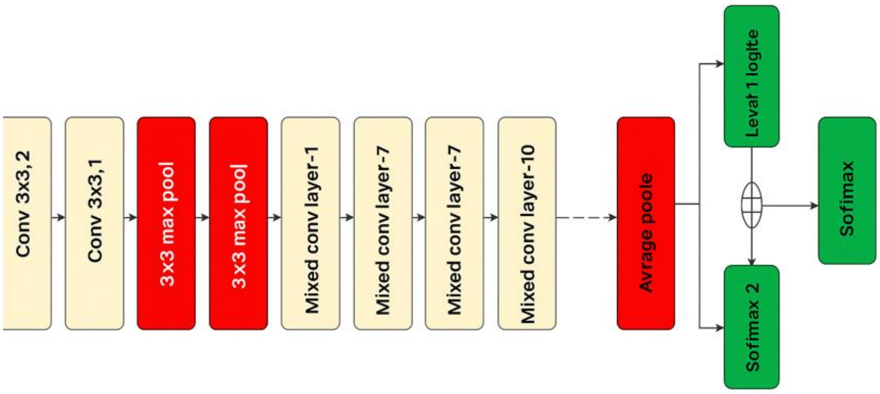

Fig. 2. Inception V3 architecture flow.

Auxiliary classifiers are intermediate layers of the network during training. These layers help combat the vanishing gradient problem and provide additional regularization, improving the network's performance. At the end, batch normalization layers are used throughout the network to stabilize and accelerate training by normalizing the inputs to each layer. Large datasets like ImageNet have been utilized to pre-train Inception V3, which can be customized for particular applications or utilized as a feature extractor in transfer learning situations. It has attained cutting-edge results on several image classification benchmarks and is extensively used in academic and real-world settings

2.3. Xception V3

Xception [22] is another DCNN architecture developed by researchers at Google. It stands for "Extreme Inception" and is inspired by the Inception architecture, particularly using multiple parallel convolutional pathways within a single module. However, Xception takes a different approach to the design of these modules as shown in Fig. 3. The key idea behind Xception is to decouple the cross-channel and spatial correlations using depth-wise separable convolutions.

Unlike traditional convolutional layers, where each filter operates on the entire input volume, depthwise separable convolutions split the process into pointwise and depthwise convolutions. A single filter is applied independently to each input channel via depthwise convolution. For every channel, it individually computes the spatial correlations. This operation essentially performs a lightweight spatial convolution on each channel individually. After the depthwise convolution, the output channels generated by the depth wise convolution are combined using a pointwise convolution (1×1 convolution). By learning cross- channel correlations, the network can efficiently blend information across channels at this stage. By decoupling the spatial and cross- channel correlations, Xception can capture more abstract features while reducing computational complexity and the number of parameters compared to traditional convolutional layers. Furthermore, residual connections akin to those found in ResNet are incorporated into Xception, aiding in solving the vanishing gradient issue and streamlining the training of profound networks. It has been demonstrated that Xception outperforms competing designs computationally while achieving competitive performance on various picture categorisation challenges.

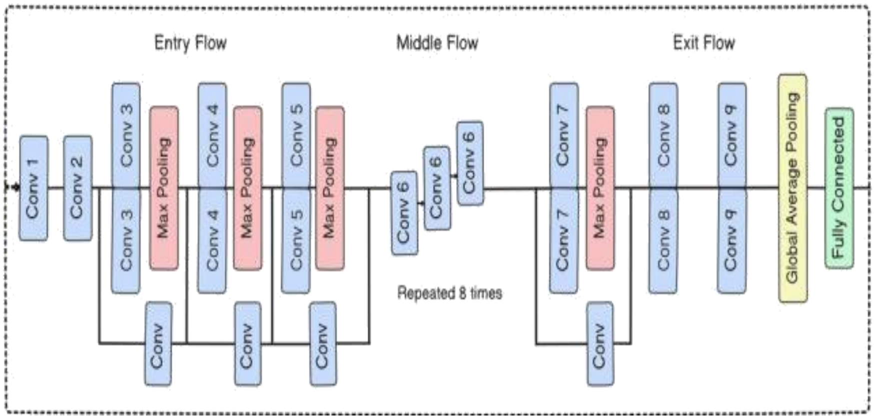

Fig. 3. Xception model architecture flow.

2.4. Ensemble models

Ensemble models are machine learning techniques combining multiple models to improve predictive performance. The idea behind ensemble learning is to leverage the diversity of different models to compensate for each other's weaknesses and enhance overall accuracy and robustness. Following training, each model may provide a prediction, which can then be pooled to increase the classification performance. The output layer employs a softmax activation function to address the multi-class classification issue involving more than two classes. Prior to calculating the class value using argmax, the sum of the probabilities for each predicted class maybe determined in this study, three ImageNet pre-trained models—ConvNeXt [23], DenseNet [24], and ResNet101 [25] - are tested individually for the waste categorisation task. The findings of each model are compared to select the most efficient architecture.

2.5. Waste sorting mechanism

Waste sorting mechanism was developed in-house. According to the classification result, it automatically drops the waste into an appropriate bin. The waste sorting mechanism comprises an ultrasonic level sensor to monitor the waste level in the waste bin and a servo motor mechanism to dispense the waste into an appropriate waste bin. The Raspberry Pi SBC controlled both the sensor and servo motor mechanisms.

3. Experimental Analysis

This section elaborates on the experimental methods and results. The experiments were performed in five phases: dataset preparation, training the DCNN models for waste classification, evaluation of the trained framework, model performance comparison, and real-time performance analysis.

3.1. Dataset preparation

The waste classification models were trained using two datasets, wasteNet and Kaggle domestic waste/garbage datasets, containing a combined 3211 images. Figure 4 shows the sample images from our training image dataset, and Table 1 gives detailed information about the waste image dataset used for training. As a pre-processing step, all images were downsized to 227×227 or 224×224 to improve the speed. Data augmentation is applied to the gathered dataset. It increases the dataset's diversity and quantity by creating unique data instances by applying.

Table 1. Waste category in training dataset.

Metal	Paper	Plastic	Cardboard	Waste
410	625	474	1095	607

Fig. 4. Ensemble model framework for waste classification.

Data augmentation is applied to the gathered dataset. It increases the dataset's diversity and quantity by creating unique data instances by applying various alterations to the initial image. Rotation, scaling, resizing, and horizontal and vertical flipping are augmentation techniques that boost training efficacy and model generalization. The on-the-fly data augmentation technique increases the model's sensitivity to a wider variety of samples and its ability to handle variations and real-world occurrences by increasing the size of the training dataset while keeping the number of images constant. Figure 5 shows the sample images of the data augmentation process.

3.2. Model training

The model was trained via TensorFlow 2.15 within a Google Colab environment featuring an NVIDIA Tesla T4 GPU and 12.69 GB of RAM. The model was initialised with pre-learnt weights from the ImageNet dataset and subsequently trained utilising a transfer learning approach. Initially, the foundational layers of the network were immobilised, and solely the newly incorporated classifier layers underwent training. Following 10 epochs, all layers were unfrozen to provide comprehensive fine-tuning of the model.

To guarantee the reproducibility of the proposed study, the following configurations were uniformly applied across all experiments. The dataset was partitioned into 80% for training, 10% for validation, and 10% for testing purposes. Images were resized to 224x224 pixels and converted to RGB colour mode. Data augmentation techniques comprised random rotations (up to ±30 degrees), horizontal flipping, and a zoom range of 0.1–0.2. The models utilised a batch size of 32, employing the Adam optimiser with a learning rate of 0.0001 and a categorical cross-entropy loss function. A fixed random seed of 42 was established during data shuffling and model initialisation to maintain consistency across runs.

3.3. Parameter configuration

The DCNN models were trained using a transfer learning strategy, which enables the reuse of pre-learnt weights from large-scale datasets, such as ImageNet. Initially, each model's convolutional base (e.g., InceptionV3, Xception, ResNet101, etc.,) was frozen to retain the learnt general features. Only the newly added classifier layers, which were specifically designed for the trash classification task, were trained in the first phase. After about ten epochs, the basal layers were unfrozen, and the entire network was fine-tuned at a much lower learning rate. This two-stage training procedure guarantees both generalization and task-specific adaptation. The adam optimizer was employed with a 0.0001 learning rate, and categorical cross-entropy was chosen as the lose function. This practice improves model convergence while lowering the risk of overfitting, especially given the dataset's moderate size.

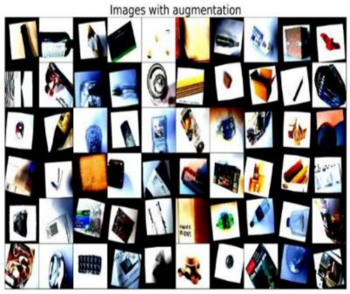

Fig. 5. Real time prototype setup.

3.4. Inception V3 training accuracy

The training commenced with an initial training accuracy of 0.65 and a validation accuracy of 0.85. Throughout 700 epochs, the training accuracy consistently increased, attaining 0.93. A notable enhancement in performance was detected throughout the initial

100 epochs, indicating effective feature acquisition. During epochs 300 to 700, the learning rate became stable, resulting in negligible variance and enhanced convergence. The validation accuracy remained consistent, signifying effective generalisation and negligible overfitting. Figure 6 depicts the advancement of accuracy throughout the training process. The graph indicates steady enhancement in both training and validation accuracy, reflecting the model's excellent learning and resilience.

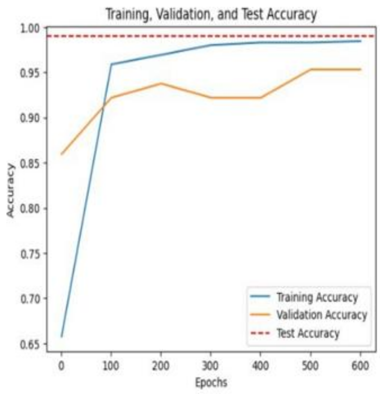

Fig. 6. Validation accuracy of Inception V3.

3.5. Xception training accuracy

The training of the Xception model commenced with an accuracy of 0.56 and a validation accuracy of 0.32. Accuracy consistently enhanced across the initial five epochs, with a notable increase in validation accuracy. A slight decline in validation accuracy was observed at epoch 6, indicating temporary overfitting, which was rectified in later epochs. Beginning with epoch 14, both training and validation accuracy exhibited steady improvement, signifying enhanced generalisation. Figure 7 illustrates the evolution of training and validation accuracy. Notwithstanding significant variability throughout the mid-training phases, the model ultimately attained robust convergence with commendable classification efficacy.

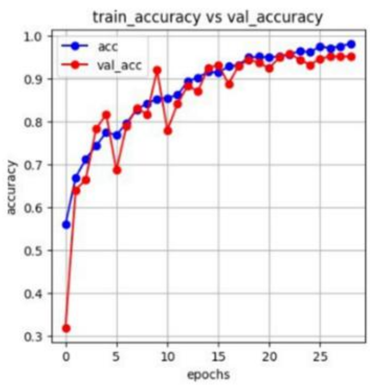

Fig. 7. Validation accuracy of Xception.

3.6. Ensemble model of (pre-trained convnext, densenet121_0, resnet101)

The ensemble model exhibited consistent and incremental enhancement in accuracy. Training commenced with limited precision because to the frozen backbone layers; however, substantial improvements were noted post the 10th epoch when all layers were unfrozen. The ensemble leveraged the synergistic benefits of the three architectures, leading to enhanced robustness and superior classification performance across categories. The validation accuracy consistently improved with negligible indications of overfitting during training. Figure 8 illustrates the training accuracy curve of the ensemble model. The graph depicts the enhancement trend upon unfreezing, validating the efficacy of the fine-tuning phase and ensemble learning approach.

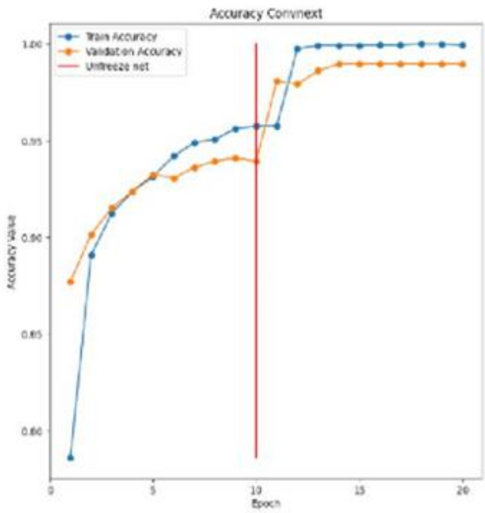

Fig. 8. Validation accuracy of ensemble model.

3.7. Trained model evaluation

Equations for accuracy Eq. (1), precision Eq. (2), recall Eq. (3) and F-measure Eq. (4) were used to assess a trained model's performance.

$$Accuracy = \frac{tp+tn}{tp+fp+tn+fn} \quad (1)$$

$$Precision(Prec) = \frac{tp}{tp+fp} \quad (2)$$

$$Recall = \frac{tp}{tp+fn} \quad (3)$$

$$F_{measure}\ (F1) = \frac{2\ X\ PrecX\ recall}{Prec+recall} \quad (4)$$

According to the conventional confusion matrix, tp, fp, tn, fn stands for the true positives, false positives, true negatives, and false negatives, respectively. The training results of models 2 and 3 are shown in Figs. 9 and 10.

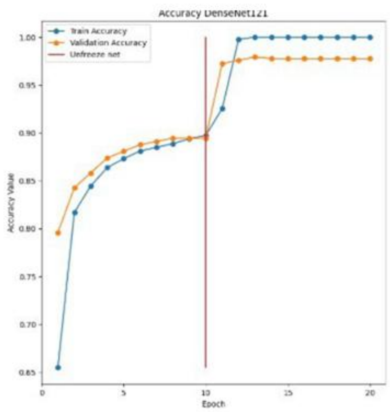

Fig. 9. Training result of ensemble model 2.

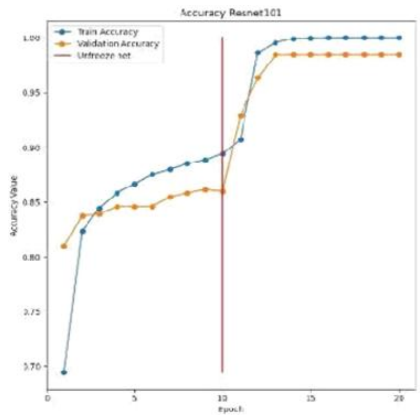

Fig. 10. Training result of ensemble model 3.

3.8. Inception model performance analysis

Figure 11 depicts the result of the confusion matrix analysis for the Inception model. The diagonal elements of the matrix show how many instances in each class were correctly identified (true positives). In our analysis, the model correctly predicted 305 occurrences of cardboard, 68 instances of metal, 111 instances of plastic, 72 instances of paper, and 80 instances of waste. The off-diagonal components, how- ever, indicate misclassifications. Notably, the model predicted one false positive for plastic, one false positive for paper, and one false negative for each of the two waste categories: cardboard and paper. The remaining elements in the matrix (200, 386, 342, and 381) represent samples that were successfully classified into their respective classes. Table 2 presents the statistical measure analysis for the inception model. The table findings show that the model's precision values range from 0.9967 for cardboard to 0.9855 for metal, indicating high accuracy in identifying positive samples. Recall scores range from 0.9863 for paper to a perfect 0000 for metal, which indicates high sensitivity in detecting true positives. The F1-score, which balances recall and precision, displays excellent performance in all subjects. With a total accuracy of 0.9938, the model can accurately categorize objects into cardboard, metal, plastic, paper, and rubbish. Given that the model's overall accuracy is 0.9938, the results demonstrate its dependability and effectiveness in classifying waste into cardboard, metal, plastic, and paper.

Figure 12 depicts the confusion matrix analysis result, while Table 3 provides statistical analysis for the Xception model. The confusion matrix depicts true positives and shown classes, with 199 examples correctly classified as cardboard and 90 and 120 cases of metal and paper, respectively. However, off-diagonal component results reveal misclassifications, such as misidentifying plastic as paper and rubbish as paper. These misclassifications could be due to issues distinguishing classes or similarities in their attributes. Further optimization may be required to reduce misclassifications between closely related classes. Overall, the model performs well across most classes, including cardboard, metal, and paper.

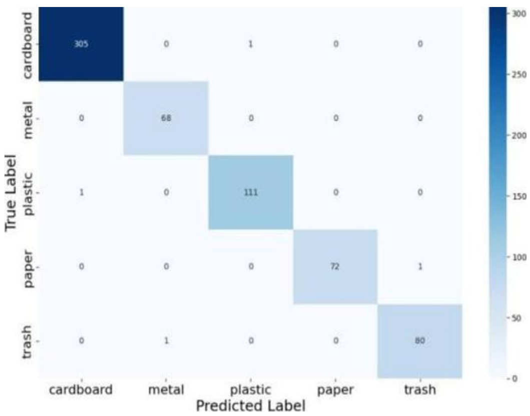

Fig. 11. Confusion Matrix of InceptionV3

Table 2. Statistical analysis for the inception model.

Class	Precision	Recall	F1-Score)
Cardboard	0.99	0.98	0.96
Metal	0.98	0.96	0.97
Plastic	0.99	0.99	0.98
Paper	0.97	0.98	0.99
Waste	0.98	0.99	0.97

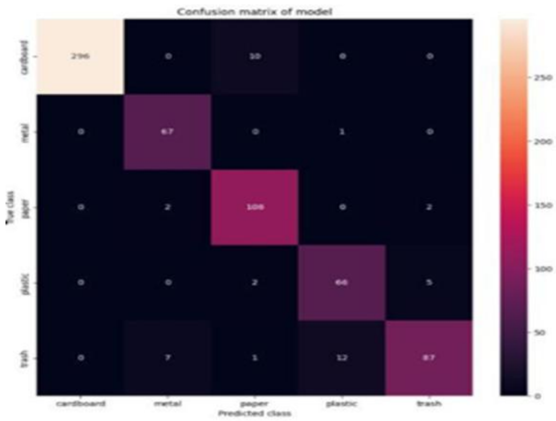

Fig. 12. Confusion matrix of Xception.

The statistical measure Table 3 demonstrates that the model accurately distinguishes cardboard occurrences with high precision and recall while minimizing false positives. However, it has lesser precision for metal, implying that certain cases may be wrongly identified as such. The model is capable of identifying the majority of genuine metal cases with better recall and precision. Paper and plastics have good precision and memory, respectively. The model has an over- all accuracy of 0.94, categorizing about 94% of the dataset's cases correctly. These metrics reflect the model's capacity to discriminate examples across all classes, which is critical to the classification task's success.

Table 3. Classification report of Xception.

Class	Precision	Recall	F1-Score
Cardboard	0.96	0.95	0.94
Metal	0.88	0.98	0.92
Plastic	0.92	0.94	0.93
Paper	0.94	0.88	0.91
Waste	0.89	0.89	0.89

3.9. Ensemble model performance analysis

Table 4 illustrates the classification result of the ensemble model. The model obtains an optimal precision score of 1.000 for cardboard, paper, and rubbish and 0.968 for metal. Every single class score 1. 000, with plastic having a recall of 0.971. The F1-scores vary from 0.981 to 1.000, illustrating that the model accurately optimizes precision and recall

Table 4. Classification report of ensemble.

Class	Precision	Recall	F1-Score
Cardboard	0.99	0.98	0.98
Metal	0.88	0.97	0.92
Plastic	0.92	0.94	0.93
Paper	0.94	0.88	0.91
waste	0.89	0.89	0.89

This section compares models' performance based on test accuracy; time taken to train and model size. The results are given in Table 5. The analysis results reveal that the InceptionV3 model performs well when sorting recyclable materials into numerous categories. The model was trained at a learning rate of 0.0001 across 700 epochs, and it shown a considerable in- crease in training accuracy. Figure 13 illustrates the model's accuracy in the confusion matrix, with 305 occurrences of cardboard, 68 instances of metal, 111 instances of plastic, 72 instances of paper, and 80 instances of waste accurately predicted. The classification result and confusion matrix reveal high precision, recall, and F1 scores for each class. The model can accurately. Categorise samples for each class, with metal having a perfect recall of 1.0000 and cardboard and plastic having high values of 0.9967 and 0.9911, respectively. F1 scores, which balance recall and precision, perform well across all classes, with values ranging from 0.9911 for plastic to 0.9967 for cardboard.

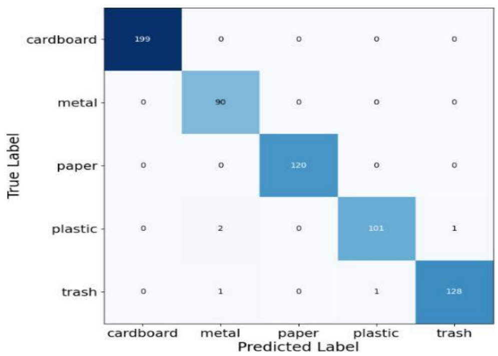

Fig. 13. Confusion matrix of ensemble model.

Table 5. Performance comparison of classes.

Model	Accuracy	Inference Time (ms)	Model Size (MB)
InceptionV3	99%	280	110
Ensemble	98%	728	303
Xception	93%	110	224
ResNet50	95%	380	210
MobileNetV2	90%	120	265
DenseNet121	94%	460	320
VGG16	92%	580	250
EfficientNetB0	94%	250	295

Table 6 presents a comprehensive evaluation of the deep learning models incorporated into the proposed trash classification system. It encompasses realistic deployment metrics, including Floating Point Operations (FLOPs), model parameter count, inference duration, model size, and throughput. These metrics evaluate the models' computing efficiency and practicality for real-time, embedded applications.

Table 6. Model performance and computational metrics for waste classification.

Model	Accuracy	Inference Time (ms)	FLOPs (GFLOPs)	Params (M)	Model Size (MB)	Throughput (img/s)
Inception V3	99%	280	5.72	23.8	110	3.57
Xception	93%	110	8.40	22.9	224	9.09
Ensemble	98%	728	18.30	60.5	303	1.37

FLOPs (GFLOPs) denote the quantity of floating-point operations necessary to process one input image. These originate from official model documentation and profiling instruments. Parameter Count (in Millions) indicates the total number of trainable weights in the model (e.g., obtained using model.summary() in TensorFlow/Keras). Throughput (images/sec) was computed as:

$$\text{Throughput} = \text{Inference}/\ \text{Time (ms)}1000 \tag{5}$$

where assuming batch size = 1 for real-time inference.

These measures elucidate the practical strengths and limitations of each model beyond conventional accuracy, facilitating appropriate model selection for implementation. Figure 14 depicts the waste classification prototype system. It consists of a Raspberry Pi Single Board Computer that serves as an edge computing source to execute the DL-based waste classification algorithm, a webcam for image capture, and servo motors automatically transferring waste into appropriate bins.

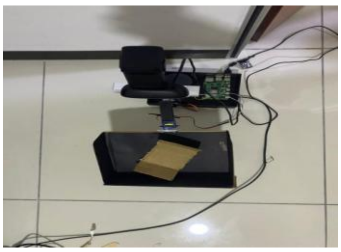

Fig. 14. Real time system performance evaluation output.

The trained Inception v3 model was installed on a Raspberry Pi SBC for real-time testing. The Raspberry Pi uses a webcam to capture picture data and perform the classification task. Once the classification is complete, the Raspberry Pi unit directs the servo motors to place the waste into the corresponding bin automatically. Figure 15 depicts real-time experimental results for waste classification. Experimental results show that the model classifies waste objects with 98% accuracy and takes an average of 650 milliseconds to infer one waste image.

Fig. 15. Prototype sorting outcome snapshots for each waste category.

4. Limitation and Future Work

The suggested system illustrates the viability of implementing deep learning models for real-time waste classification on an economical embedded platform; nonetheless, this study mainly concentrates on model assessment and prototype demonstration in controlled environments. A comprehensive investigation of the overall system cycle time, encompassing picture acquisition, preprocessing, inference, and actuation delays, beyond the current scope. Moreover, system performance under many environmental conditions (e.g., illumination, crowded backgrounds) necessitates additional examination. Future endeavours will focus on performing a comprehensive real-time performance assessment, investigating optimisation strategies to minimise inference and actuation delay, and evaluating the system's scalability and resilience across various operational contexts. Moreover, subsequent research will focus on executing a quantitative benchmark comparison of the proposed system with current state-of-the-art techniques in trash classification, utilising standardised datasets and evaluation methodologies to enhance the validity of the comparative analysis.

5. Conclusions

This paper proposed a DCNN-based garbage classification model for waste sorting applications. A waste image dataset was used to train three DCNN models: InceptionV3, Ensemble Model, and Xception. Throughout the study, their training and classification accuracy were tested and compared. Accuracy, precision, recall, and F1 -score were standard performance measures used to assess the trained model's performance. The Inception-V3 model, which successfully sorted garbage into several categories with high precision, recall, and F1-score values in all classes, achieved the highest level of accuracy in our research. The model's test accuracy revealed its capacity to generalise to novel and previously untested data. Despite some misclassifications, the InceptionV3 model's overall performance revealed its utility in automating waste classification processes. The empirical findings show that the trained DCNN-based waste classification system has a high potential for use in real-world recycling settings, helping improve waste management systems efficacy and efficiency.

Nomenclatures

tp	True positive
fp	False positive
tn	True negative
fn	False negative
prec	Precision

Abbreviations

DCNN	Deep Convolution Neural Network
DL	Deep Learning
CNN	Convolutional Neural Network
RCNN	Recurrent Convolutional Neural Network
SVM	Support Vector Machine
SIFT	Scale Invariant Feature Transform
FLS	Forward-Looking Sonar
YOLO	You Only Look Once
SSD	Single Shor multibox Detector
USB	Universal Serial Bus
CPU	Central Processing Unit
RAM	Random Access Memory
GPU	Graphic Processing Unit

References

1. Yin, J. et al. (2020). Table cleaning task by human support robot using deep learning technique. *Sensors*, 20(6), 1698.
2. Malik, M. et al. (2022). Waste classification for sustainable development using image recognition with deep learning neural network models. *Sustainability*, 14 (12), 7222.
3. Gómez, B.F. et al. (2022). Deep learning-based litter identification and adaptive cleaning using self-reconfigurable pavement sweeping robot. *Proceedings of IEEE* 18*th International Conference on Automation Science and Engineering* (*CASE*, 2022), Mexico, 2301-2306.
4. Rahman, M.W. et al. (2022). Intelligent waste management system using deep learning with IoT. *Journal of King Saud University - Computer and Information Sciences*, 34(5), 2072-2087.
5. Uppala, R.; Ramya, V.; and Murugan, V.S. (2023). Develop a 7 layers convolution neural network and IoT-based garbage classification system.

International Journal of Intelligent Systems and Applications in Engineering, 11(3), 268-275.

6. Arbeláez-Estrada, J.C. et al. (2023). A systematic literature review of waste identification in automatic separation systems. *Recycling*, 8(6), 86.
7. Nnamoko, N.; Barrowclough, J.; and Procter, J. (2022). Solid waste image classification using deep convolutional neural network. *Infrastructures*, 7(4), 47.
8. Yang, Z.; Xia, Z.; Yang, G.; and Lv, Y. (2022). A garbage classification method based on a small convolution neural network. *Sustainability*, 14(22), 14735.
9. Zhihong, C.; Hebin, Z.; Yanbo, W.; Binyan, L.; and Yu, L. (2017). A vision-based robotic grasping system using deep learning for garbage sorting. *Proceedings of* 36*th Chinese Control Conference* (*CCC*, 2017), Dalian, China, 11223-11226.
10. Wang, W.-C.; Chen, L.-B.; and Chang, W.-J. (2018). Development and experimental evaluation of machine-learning techniques for an intelligent hairy scalp detection system. *Applied Sciences*, 8(6), 853.
11. Elangovan, K. et al. (2017). Fault diagnosis of a re-configurable crawling–rolling robot based on support vector machines. *Applied Sciences*, 7(10), 1025.
12. Mittal, G.; Yagnik, K.B.; Garg, M.; and Krishnan, N.C. (2016). Spotgarbage: Smartphone app to detect garbage using deep learning. *Proceedings of the* 2016 *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, Heidelberg Germany, 940-945.
13. Rad, M.S. et al. (2017). A computer vision system to localise and classify wastes on the streets. *Proceedings of* 11th *International Conference on Computer Vision Systems* (ICVS 2017), Shenzhen, China, 195-204.
14. Valdenegro-Toro, M. (2016). Submerged marine debris detection with autonomous underwater vehicles. *Proceedings of* 2016 *International Conference on Robotics and Automation for Humanitarian Applications* (*RAHA*), Amritapuri, India, 1-7.
15. Fulton, M.; Hong, J.; Islam, M.J.; and Sattar, J. (2018). Robotic detection of marine litter using deep visual detection models. *Proceedings of* 2019 *International Conference on Robotics and Automation* (*ICRA*), Montreal, QC, Canada, 5752-5758.
16. Gondal, A.U. et al. (2021). Real-time multi-purpose smart waste classification model for efficient recycling in smart cities using multilayer convolutional neural network and perceptron. *Sensors*, 21(14), 4916.
17. Gowrishankar, K.; Muthukumar, V.; Pandian, S.R.; Ang, C.K. (2023). A novel SVM and K-NN classifier based machine learning technique for epileptic seizure detection. *International Journal of Online and Biomedical Engineering*, 19(07), 99-124.
18. Narmadha, G. et al. (2023). Detection of human stress using optimized feature selection and classification in ECG signals. *Mathematical Problems in Engineering*, 2023.
19. Abhishek, J.N.; Kulal, A.; Dhanapal, K.; Chouter, D.F.; and Seema, B.S. (2025). AI powered waste sorting: Real time cost effective smart waste detection and sorting system. *International Journal of Creative Research Thought* (*IJCRT*), 13(1), c117-c124.

20. Sayem, F.R. et al. (2025). Enhancing waste sorting and recycling efficiency: Robust deep learning-based approach for classification and detection. *Neural Computing and Applications*, 37, 4567-4583.
21. Adedeji, O.; and Wang, Z. (2019). Intelligent waste classification system using deep learning convolutional neural network. *Procedia Manufacturing*, 35, 607-612.
22. Chollet, F. (2016). Xception: Deep learning with depthwise separable convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1251-1258.
23. Liu, Z. et al. (2022). A ConvNet for the 2020s. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11966-11976.
24. Huang, G.; Liu, Z.; van der Maaten, L.; and Weinberger, K.Q. (2017). Densely connected convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700-4708.
25. He, K.; Zhang, X.; Ren, S.; and Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 770-778.

Appendix A

Computer Programme

A.1. Introduction

The waste classification module executes the waste classification task. It comprises two functional modules: an imagine unit and DCNN-based waste classification algorithms. The waste classification module takes the input from the USB webcam and performs the classification task. A USB webcam captures the image at 2.1 megapixels. Three DCNN algorithms, including Inception V3, Xception, and Ensemble models, were trained for the waste classification task, and their performance was analysed.

A.2. Experiment Structure and Description of framework

The experiments were performed in five phases: dataset preparation, training the DCNN models for waste classification, evaluation of the trained framework, model performance comparison, and real-time performance analysis.

The open-source machine learning framework TensorFlow 2.15 was used to construct the waste classification system. Using an Intel Xeon CPU, 12.69 GB of RAM, and an NVIDIA Tesla T4 Server GPU (16 GB of VRAM), the model was trained and evaluated using Google Colab.