Classification of Recyclable Wastes with Deep Learning Applications

Mahmut Emir Arslan, N22131539, email:emirarslan.ceng@gmail.com

Hacettepe University, Institute of Informatics, Turkey

Summary

The concept of "recycling," which has gained increasing popularity year by year in response to environmental pollution caused by the rapid growth of the world population, has become a matter of utmost importance. Recycling involves the reuse of certain waste materials and raw resources. In this study, software has been developed to classify recyclable waste materials such as cardboard, glass, plastic, paper, and metal using a machine learning approach.

Keywords: Machine Learning, Recycling, Environmental Pollution.

1 Introduction

Industrialization, which fosters human development, also leads to environmental pollution. This environmental pollution poses an increasing threat to human life daily, resulting in the accumulation of vast amounts of waste and severely impacting marine life. This study aims to identify recyclable waste materials and categorize them for recycling purposes. The dataset required for developing the system using machine learning methods was sourced from the website "www.kaggle.com." This dataset comprises 393 cardboard images, 491 glass images, 400 metal images, 584 paper images, and 472 plastic images. Additionally, there are 127 images serving as examples of non-recyclable waste. To develop the system, the "TensorFlow" library, which supports machine learning software, will be utilized.

2 Related Studies

This article highlights the adverse impact on human health resulting from waste collection and sorting processes in India, emphasizing the associated challenges related to time, cost, and quality. In this study, CNN algorithms were applied to analyze four distinct datasets. Upon analyzing the datasets and their corresponding accuracy rates, it becomes evident that datasets with fewer classes, achieved through more extensive clustering, exhibit higher accuracy rates, as anticipated (Singh, Rana, Tiwari, & Mittal, 2020).

In this study, two distinct approaches were pursued: an image classification algorithm and an object detection algorithm. The results from these two algorithms were integrated to arrive at a final decision. This combination of approaches contributed to a 2% increase in the accuracy rate, resulting in an impressive recognition rate of 98%. For image classification, ResNet and MobileNetV2 technologies were employed, while the YOLOv5 family provided three algorithms for object detection (Yang, Bao, & Liu, 2022).

This study conducts classification utilizing pre-trained deep learning models, including EfficientNetB7, InceptionV3, NASNet-Large, and ResNet50-V2. Among these models, ResNet50-V2 and InceptionV3 achieved the highest accuracy rates. ResNet50-V2 attained a remarkable accuracy rate of 97.07%, while InceptionV3 outperformed the others with a coefficient of 0.8277 according to "Cohen's Kappa," a statistical measure (Sürücü & Ecemiş, 2022).

In this study, SVM and Softmax classifiers were employed to enhance the accuracy of deep learning models, which included AlexNet, GoogleNet, ResNet, VGG-16, and SqueezeNet. The dataset, consisting of 2527 images, was divided into two equal halves for training and testing. The results indicate that SVM consistently yielded higher accuracy rates. Specifically, the SVM classifier achieved the highest accuracy rate of 97.86% when used in conjunction with the GoogleNet model. Conversely, the lowest accuracy rate of 83.43% was observed in the combination of SqueezeNet and Softmax (Özkaya & Seyfi, 2018).

In this study, V3 and V4 versions of the YOLO (You Only Look Once) model were investigated using a dataset comprising a total of 22,000 images and 15 different types of garbage objects for training. The research indicates that within the range of 0-200 iterations, there is a significant decrease in loss values and a notable increase in accuracy rates. As the iterations progress from 700 to 1200, training performance steadily improves and stabilizes, ultimately reaching a MAP (mean average precision) value of 64%. Among the YOLO versions employed, YOLOV4 outperformed YOLOV3. When comparing the enhanced YOLOV4 with the standard YOLOV4, it was observed that the accuracy rate remained unchanged, but the number of parameters, FPS value, and loss rates decreased, making it a superior choice. Notably, the increased FPS value proved beneficial for distinguishing between more garbage categories, and the model can be integrated into embedded systems, enabling real-time applications (Chen & Xiong, 2020).

In this study, waste classification was conducted across three groups: Organic, Inorganic, and Recyclable. A distinctive aspect of this research is the manual collection of the dataset from the internet. The deep learning model utilized in the system is ResNet-50, which achieved first place in the 2015 ILSVRC and COCO competitions. The dataset comprises a total of 1500 images, with 500 images allocated to each group. During model training, the training set achieved a high accuracy value of 0.966 by the 20th epoch. Test results indicated that confidence values exceeding 99% were consistently attained in each group classification. The average classification time for each image was calculated to be 147 milliseconds (Huy, Tung, Linh, & Minh, 2023).

The dataset obtained from Kaggle comprises 12 different garbage groups, totaling 12,539 images. In this study, 80% of the dataset was allocated for training data, 10% for validation data, and the remaining 10% for testing data. Furthermore, data augmentation techniques, such as random flip, random rotation, and random zoom, were employed to further expand the dataset. The InceptionV3 model served as the deep learning model to extract image features, incorporating "Batch Normalization" layers to mitigate the risk of overfitting. The model attained an impressive accuracy rate of 93.1% in the test phase. To address overfitting, the early stopping method was employed, terminating training at the 25th Epoch instead of the initially set 200th Epoch. The model began to stabilize around the 16th Epoch but showed a slight loss of stability by the 23rd Epoch. Combining 5 fully connected layers, 2 batch normalization layers, and transfer learning, the study achieved an accuracy of 95.5% on the training dataset, 90.77% on the validation dataset, and 93.1% on the test dataset (Chen, et al., 2022).

This paper delves into the detection and classification of garbage objects using a sophisticated system. This system autonomously operates a household robot to determine the location and class of garbage objects. The object detection method employed in the study is YOLOV5m. Within this method, the "YOLOV5m-Attention detection" algorithm stands out, as it amalgamates incoming image information with a priori databases, culminating in the "YOLOV5m-Attention-KG" model. The dataset utilized encompasses a total of 15,000 domestic rubbish images, primarily drawn from the garbage classification competition dataset organized by Ali Yun Tianchi. Additionally, some images were manually collected by the author. The dataset is categorized into four groups: recyclable waste, food waste, hazardous waste, and other waste. Of these groups, 10% was designated as the validation dataset, while the remaining 13,500 images constituted the training dataset. The evaluation criteria for the experimental results encompassed accuracy, sensitivity, average sensitivity, and detection times. After the 100th epoch, the loss attained its minimum value, signifying stability. The original "YOLOV5m-Attention-KG" model exhibited a 0.4% increase in accuracy compared to YOLOV5 at the same detection time (Wu, Shen, Liu, Xiao, & Li, 2021).

During the process of selecting a deep learning model for this study, various versions of Le-Net, AlexNet, VGG, Inception, and ResNet models, which were popular in 2020, were assessed. They were compared through 200-step iteration training using the cifar-10 dataset. Ultimately, the ResNet model series was chosen as the base model due to its suitability for the study's objectives. Although Inception models had more complex network structures compared to the ResNet models, potentially leading to higher parameter calculations, ResNet was favored. Specifically, ResNet-34 and ResNet-152 models were evaluated, and ResNet-34 was selected as the base model after a depth comparison between the two. For the testing phase involving 14 different objects, a dataset comprising over 200 images for each item, totaling 4,168 images, was prepared as new training data. The model underwent retraining with this dataset. When the system was integrated into a camera and tested, accuracy rates exceeding 95% were achieved for 12 out of the 14 items. Further enhancements were implemented on the model, including multiple feature fusion, the utilization of residual units, and activation functions. Subsequent to these refinements, the model demonstrated effective performance with an average accuracy rate of 99% and an average classification time of 0.95 seconds (Kang, Yang, Li, & Zhang, 2020).

3 Method

In this study, Convolutional Neural Network (CNN) and Multilayer Perceptron (MLP) deep learning models were employed. Various regression methods and optimization algorithms were applied to these models, and multiple training and testing experiments were conducted.

3.1 Dataset

The dataset used in the study was obtained from the Kaggle website (cchangcs, 2018). It comprises images categorized into 6 different classes, including 5 recyclable waste classes and 1 non-recyclable waste class. The dataset consists of 393 cardboard images, 491 glass images, 400 metal images, 584 paper images, 472 plastic images, and 127 rubbish images, resulting in a total of 2467 images. The original dimensions of these images are 384x512, but due to the extensive time required for model training at these dimensions, the image dimensions were resized to 128x128. Figure 1 displays 7 images from each class in the dataset.

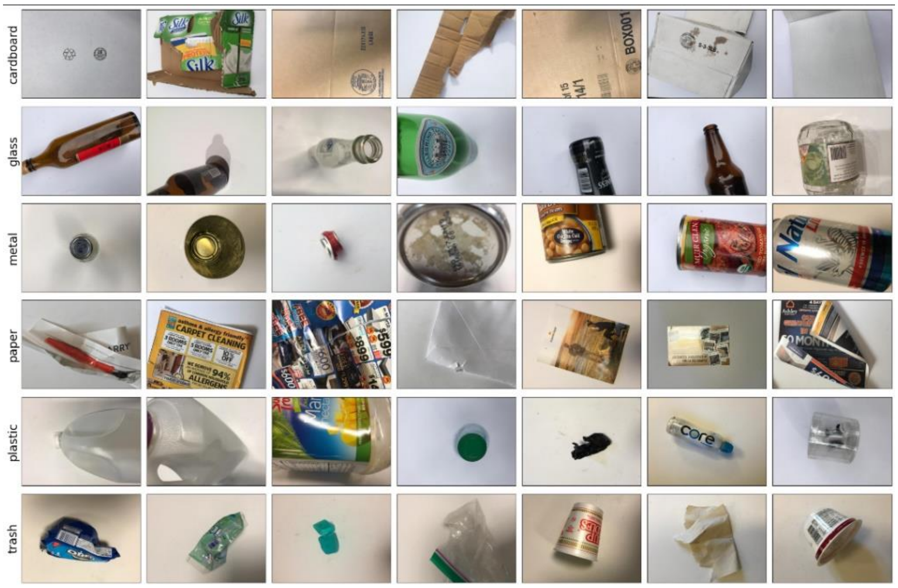

Figure 1 Sample images of the 1 set

3.2 Optimization and Regulation

Dropout

Dropout is a regularization technique employed to mitigate the issue of overfitting in deep learning models. Overfitting occurs when a model excessively tailors itself to the training data and performs poorly on new, unseen data. Dropout helps alleviate this problem by randomly deactivating a portion of the neural units within the network, thereby reducing overfitting. The dropout rate typically ranges between 0 and 1, where, for instance, setting it to 0.5 means that each neural unit has a 50% chance of being deactivated during each training iteration. The specific values for this technique should be fine-tuned based on the network's architecture and the characteristics of the dataset.

Early Stop

The Early Stop technique, similar to the Dropout technique, is employed to address the issue of overfitting. This technique keeps a watchful eye on the model's performance on the training dataset and assesses how this performance evolves throughout the training process using a predefined benchmark metric. Typically, the criterion metric is chosen from the model's performance metrics such as accuracy, error rate, or loss function. Training is halted if the benchmark metric fails to improve or decreases over a certain number of consecutive training iterations. Early stopping helps prevent unnecessary consumption of computational resources and optimizes training time. Proper application of this technique requires careful selection of the criterion metric.

Adam optimization

Adam (Adaptive Moment Estimation) optimization is an algorithm that incorporates the benefits of both momentum-based optimization and RMSProp optimization. Adam achieves adaptive learning rates by calculating exponential moving averages of previous gradients and squared gradients. This approach enables Adam to converge more rapidly and consistently, offering a swift initial approximation and an adaptive learning rate that depends on the gradient magnitude. Additionally, Adam handles sparse gradients effectively and typically converges faster than other optimization algorithms. Thanks to its high performance and minimal hyperparameter tuning demands, Adam is a frequently favored optimization technique in deep learning.

RMSProp optimization

Root Mean Square Propagation, or RMSProp for short, is a commonly utilized optimization algorithm in the realm of deep learning applications, contingent on the dataset. This algorithm adapts the learning rate for each parameter based on the average magnitude of the final gradients. It utilizes the second moment, which represents the mean of the squares, for updating parameters. Consequently, it delivers a more stable learning process by mitigating fluctuations in parameter updates.

SGD optimization

This optimization algorithm, known as Stochastic Gradient Descent (SGD), is a variant of gradient descent. Rather than utilizing the entire dataset for each parameter update, it computes gradients on a single training sample or a small dataset. This approach results in a faster and more efficient training process compared to using the entire training dataset. SGD updates the parameters using gradients computed from randomly selected samples, thereby accelerating the learning process.

3.3 Classification Methods

CNN

Convolutional Neural Networks (CNNs) are widely employed in deep learning, particularly in image processing tasks. These models excel at feature extraction by taking into account the spatial structure of input data. In CNNs, convolution layers detect local patterns by applying filters to the input data. Pooling layers condense feature maps, reducing their size while preserving crucial information. Fully connected layers process feature maps and connect them to layers used for classification or regression. CNNs deliver excellent performance in visual tasks such as image classification, object recognition, and image segmentation.

The CNN model utilized in this study comprises four parts. Each part includes the ReLU activation function and 2-D convolution (CONV2D) layer, ensemble normalization (BatchNormalization), max-pooling (MaxPooling2D), and Dropout layers. The final part incorporates the Flatten layer, which transforms feature maps into vector format, as well as the Dense, BatchNormalization, and Dropout layers. In each part, the number of filters is halved.

```
tf.keras.layers.Conv2D(128, (3, 3), activation='relu'),
tf.keras.layers.BatchNormalisation(),
tf.keras.layers.Conv2D(128, (3, 3), activation='relu'),
tf.keras.layers.BatchNormalisation(),
tf.keras.layers.MaxPooling2D(2, 2),
tf.keras.layers.Dropout(0.2),
```

It's worth noting that the 2D convolution layers in Chapters 2 and 3 involve kernel and bias regularizations. Through experimentation, the optimal dropout layer value was determined to be "0.2." The optimization method that yielded the best results was Adam optimization.

MLP

A basic artificial neural network model, commonly referred to as a Multilayer Perceptron (MLP), consists of three layers: input, hidden, and output. MLPs assign weights and activation functions to each neural unit. Information flows from the input layer through the hidden layers and eventually to the output layer. The neurons in the hidden layers process and learn from the input data, enabling the network to comprehend complex relationships and patterns. MLPs can address a broad range of problems and represent a specific function with an adequate number of hidden layers and neural units. During training, optimization algorithms like gradient descent adjust the weights. MLPs find application in various machine learning tasks, including classification, regression, and time series prediction.

The MLP model developed in this study comprises a Flatten input layer, followed by five Dense layers, each accompanied by Dropout layers. In the second Dense layer, kernel and bias regularizations are incorporated. The final layer employs the "Softmax" activation function, which normalizes the output. The optimization method that yielded the best results was Adam. During model experimentation, it was observed that the introduction of unnecessary additional layers or altering the Batch Size value by halving or doubling it led to overfitting issues..

4 Results

CNN Model

The studies included the use of Adam, RMSprop, and SGD optimizations, and their impact on accuracy and loss values was examined. These investigations were conducted with BatchSize set to 32 and 100 epochs. Based on the results obtained from the experiments, it was observed that Adam optimization yielded the best results for the CNN model in the Garbage Classification dataset. Interestingly, the addition of kernel and bias regularizations to the CNN model led to a significant increase in loss values, despite offering only marginal improvements in accuracy rates. The figures below illustrate the results of the optimization model that performed best with the CNN model.

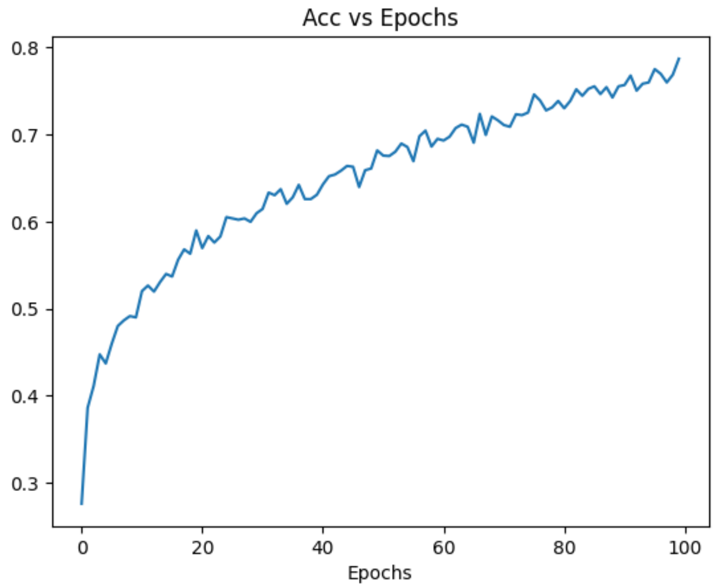

Figure 2 CNN model Accuracy - Epochs plot with Adam optimization

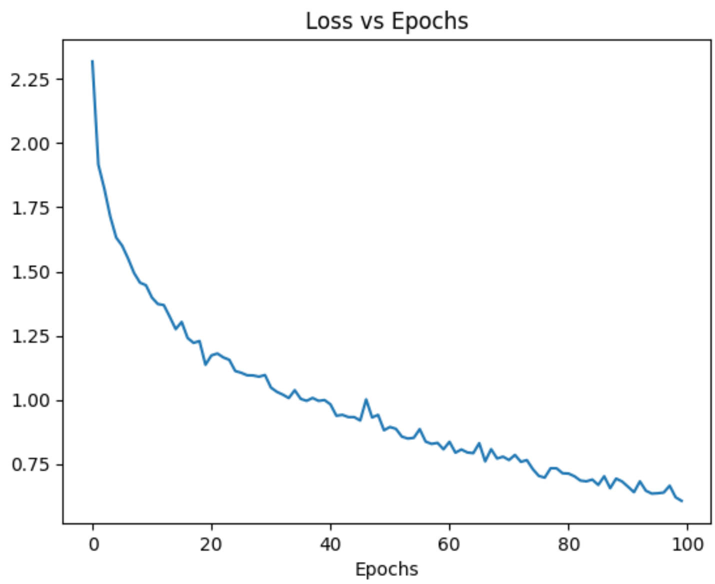

Figure 3 model Loss - Epochs graph with Adam optimization

```
test_loss,test_acc = model.evaluate(validation_generator,verbose=2)

16/16 - 6s - loss: 1.1703 - accuracy: 0.6044 - 6s/epoch - 363ms/step
```

Figure 4 CNN model Adam optimization Test results

The combination that yielded the poorest results with the CNN model was the use of SGD optimization with applied regularization methods. This aligns with existing research indicating that SGD optimization is not ideally suited for image processing tasks, a fact reaffirmed by the outcomes of this study. The figures below illustrate the results obtained.

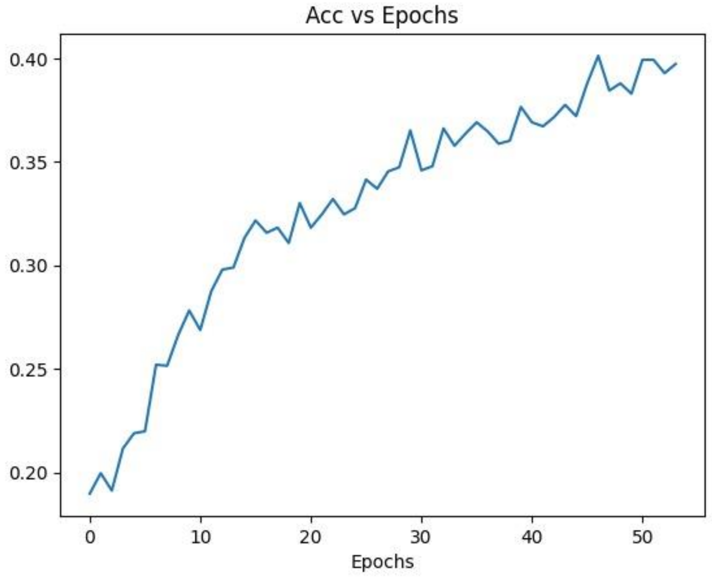

Figure 5 CNN model regression methods applied to SGD optimization Accuracy - Epochs graph

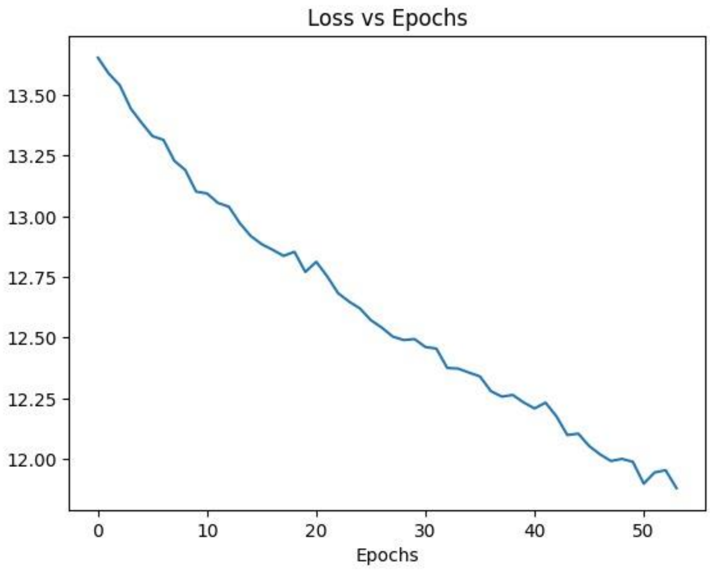

Figure 6 CNN model SGD optimization with regression methods applied Loss - Epochs plot

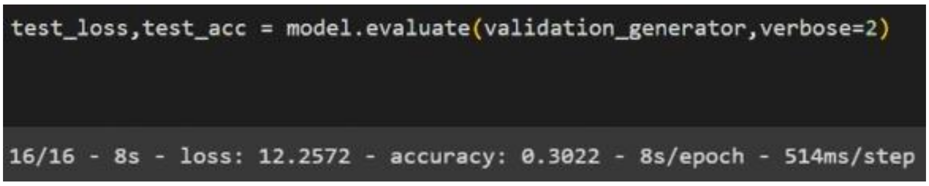

Figure 7 CNN model SGD optimization test results with regression methods applied

The experiment involving the CNN model and SGD optimization reveals that regression methods don't consistently provide a positive impact in the realm of Deep Learning Applications. This phenomenon can be attributed to both the specific dataset used and the compatibility between the Deep Learning model and the regression methods employed.

MLP Model

Based on the test results obtained from the experiments, it was observed that the MLP model achieved the highest accuracy rate when trained with the "RMSprop" optimization along with the inclusion of kernel and bias regularizations. Although the model was trained for 100 epochs, it was halted early at the 10th epoch due to overfitting. Nevertheless, it still outperformed both Adam and SGD optimizations. The results for the MLP model using regression methods and "RMSprop" optimization are displayed in the figures below.

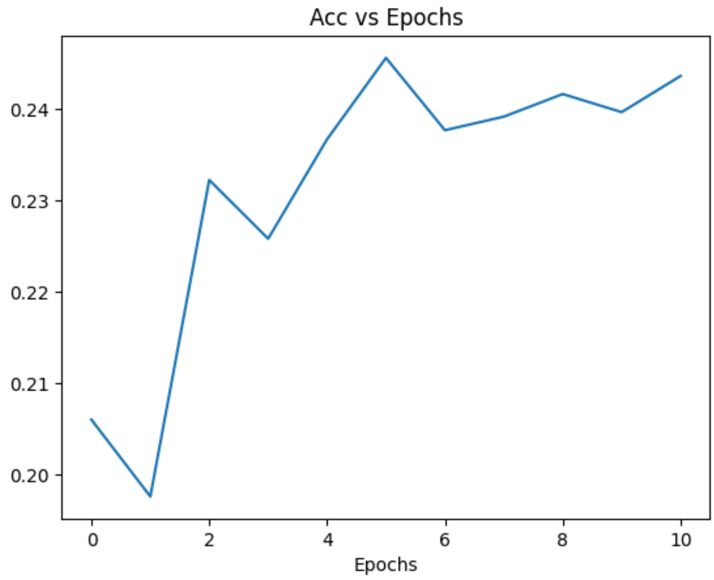

Figure 8 model RMSprop optimization with regression methods applied Accuracy - Epochs plot

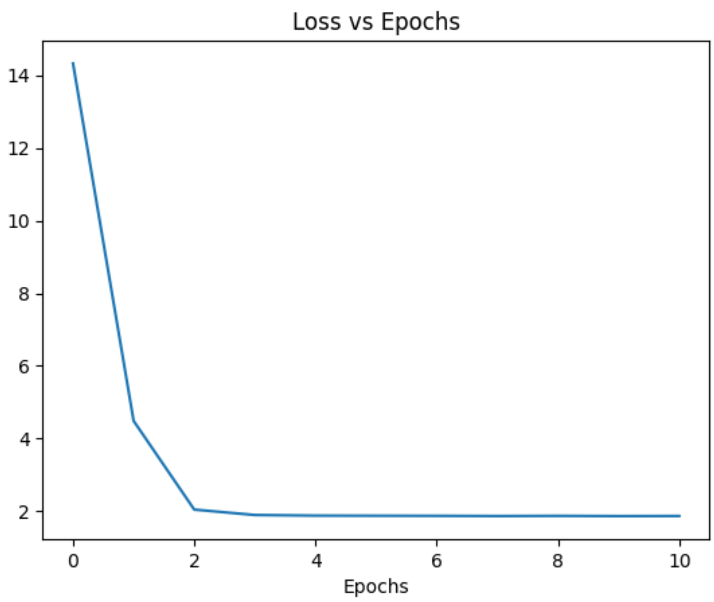

Figure 9 model RMSprop optimization with regression methods applied Loss - Epochs plot

```
test_loss,test_acc = model.evaluate(validation_generator,verbose=2)#RMS

16/16 - 2s - loss: 1.8778 - accuracy: 0.2366 - 2s/epoch - 130ms/step
```

Figure 10 MLP model RMSprop optimization test results with regression methods

The least favorable outcomes observed in the investigations involving the MLP model are the outcomes generated by the "SGD" optimization without the use of regression methods. These results are visible in the figures below.

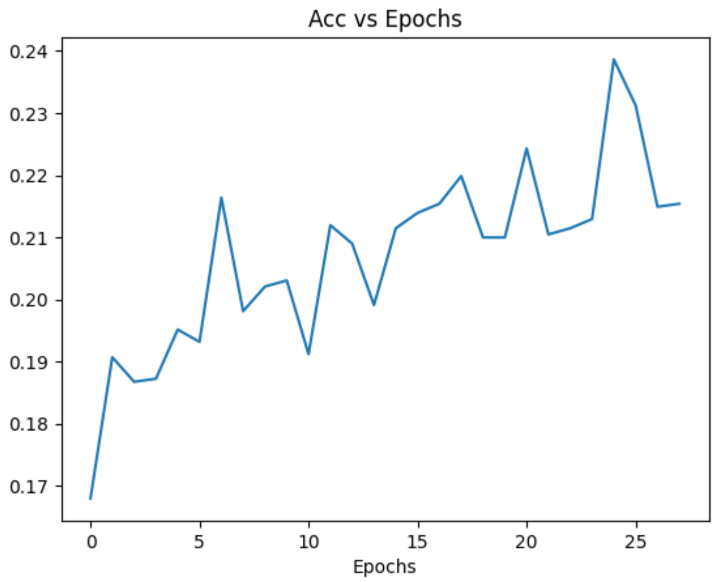

Figure 11 MLP model SGD optimization without regression methods Accuracy - Epochs plot

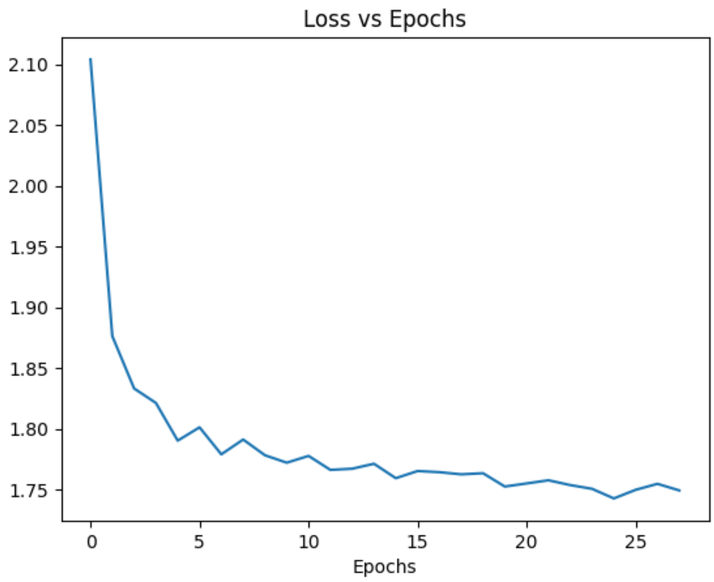

Figure 12 model SGD optimization without regression methods Loss - Epochs plot

```
test_loss,test_acc = model.evaluate(validation_generator,verbose=2)#SGD

16/16 - 2s - loss: 1.7522 - accuracy: 0.2187 - 2s/epoch - 133ms/step
```

Figure 13 model SGD optimization test results without regression methods

Based on the accuracy rates achieved in the MLP model, the ranking of optimization types is as follows: RMS > Adam > SGD. It was observed that regression methods improved the accuracy rate and enhanced the model's consistency.

The test results obtained from all the optimizations and regression methods applied to the CNN and MLP models are summarized in the table below.

Table 1. Accuracy rates and Loss values in all trials

	Accuracy Rate	**Loss value**
With Adam optimization Regression Not Applied CNN model	0.6044	1.1703
With Adam optimization Regression Applied CNN model	0.6004	1.7584
With RMSprop optimization Regression Not Applied CNN model	0.5686	1.8843
With RMSprop optimization Regression Applied CNN model	0.5209	2.2246
With SGD optimization Regression Not Applied CNN model	0.4344	1.6300
With SGD optimization Regression Applied CNN model	0.3022	12.2572
With Adam optimization Regression Not Applied MLP model	0.2346	1.7227
With Adam optimization Regression Applied MLP model	0.2346	1.7618
With RMSprop optimization Regression Not Applied MLP model	0.2346	1.7228
With RMSprop optimization Regression Applied MLP model	0.2366	1.8778
With SGD optimization Regression Not Applied MLP model	0.2187	1.7522
With SGD optimization Regression Applied MLP model	0.2366	17.5239

5 Discussion

This study was carried out to reduce environmental pollution in the world we live in. Experiments were carried out for the detection and classification of recyclable wastes. These experiments were performed using the CNN model and MLP model. Some results were obtained by comparing the performance of these two models with different optimization and regression methods.

According to these results, it is seen that the CNN model is more successful than the MLP model in the field of image classification, while SGD optimization is behind Adam and RMSprop optimizations in this field.

In future studies, producing better results by making the dataset of better quality and thus creating a hardware system that can be used in real life and working with this system can significantly prevent environmental pollution.

References

1. cchangcs. (2018). *kaggle.com*. Retrieved from https://www.kaggle.com/datasets/asdasdasasdas/garbage-classification
2. Chen, Q., & Xiong, Q. (2020, 12). Garbage Classification Detection Based on Improved YOLOV4. *Journal of Computer and Communications*.
3. Chen, Y., Han, W., Jin, J., Wang, H., Xing, Q., & Zhang, Y. (2022, 01). Clean Our City: An Automatic Urban Garbage Classification Algorithm Using Computer Vision. *Journal of Physics: Conference Series*.
4. Huy, N. H., Tung, B. V., Linh, L. H., & Minh, N. D. (2023, 03). *Vinh University Journal of Sciences*.
5. Kang, Z., Yang, J., Li, G., & Zhang, Z. (2020, 08). An Automatic Garbage Classification System Based on Deep Learning. *IEEE Access: The Multidisciplinary Open Access Journal*.
6. Özkaya, U., & Seyfi, L. (2018, 11). Fine-Tuning Models Comparisons on Garbage Classification for. *The SETSCI Conference Indexing System*.
7. Signh, P., Rana, B., Tiwari, A., & Mittal, I. (2020, 07). Trash Classification: Classifying garbage using Deep Learning. *Journal of Engineering Sciences*.
8. Sürücü, S., & Ecemiş, İ. N. (2022, 05). Garbage Classification Using Pre-Trained Models. *European Journal of Science and Technology*.
9. Wu, Y., Shen, X., Liu, Q., Xiao, F., & Li, C. (2021, 11). A Garbage Detection and Classification Method Based on Visual Scene Understanding in the Home Environment. *Complexity*.
10. Yang, Z., Bao, Y., & Liu, Y. (2022, 12). Research on deep learning garbage classification system based on fusion of image classification and object detection classification. *Mathematical Biosciences and Engineering*.