



ARTICLE

Entropy Based Feature Fusion Using Deep Learning for Waste Object Detection and Classification Model

Ehab Bahaudien Ashary¹, Sahar Jambi², Rehab B. Ashari² and Mahmoud Ragab^{3,4,*}

¹Electrical and Computer Engineering Department, Faculty of Engineering, King Abdulaziz University, Jeddah, 21589, Saudi Arabia

²Information Systems Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, 21589, Saudi Arabia

³Information Technology Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, 21589, Saudi Arabia

⁴Department of Mathematics, Faculty of Science, Al-Azhar University, Naser City, Cairo, 11884, Egypt

*Corresponding Author: Mahmoud Ragab. Email: mragab@kau.edu.sa

Received: 26 April 2023 Accepted: 13 June 2023 Published: 09 November 2023

ABSTRACT

Object Detection is the task of localization and classification of objects in a video or image. In recent times, because of its widespread applications, it has obtained more importance. In the modern world, waste pollution is one significant environmental problem. The prominence of recycling is known very well for both ecological and economic reasons, and the industry needs higher efficiency. Waste object detection utilizing deep learning (DL) involves training a machine-learning method to classify and detect various types of waste in videos or images. This technology is utilized for several purposes recycling and sorting waste, enhancing waste management and reducing environmental pollution. Recent studies of automatic waste detection are difficult to compare because of the need for benchmarks and broadly accepted standards concerning the employed data and metrics. Therefore, this study designs an Entropy-based Feature Fusion using Deep Learning for Waste Object Detection and Classification (EFFDL-WODC) algorithm. The presented EFFDL-WODC system inherits the concepts of feature fusion and DL techniques for the effectual recognition and classification of various kinds of waste objects. In the presented EFFDL-WODC system, two major procedures can be contained, such as waste object detection and waste object classification. For object detection, the EFFDL-WODC technique uses a YOLOv7 object detector with a fusion-based backbone network. In addition, entropy feature fusion-based models such as VGG-16, SqueezeNet, and NASNet models are used. Finally, the EFFDL-WODC technique uses a graph convolutional network (GCN) model performed for the classification of detected waste objects. The performance validation of the EFFDL-WODC approach was validated on the benchmark database. The comprehensive comparative results demonstrated the improved performance of the EFFDL-WODC technique over recent approaches.

KEYWORDS

Object detection; object classification; waste management; deep learning; feature fusion



1 Introduction

Object detection is an easy task for humans. Children can start identifying typical objects, but teaching them to use the computer has been a challenging task for the past few years [1]. It identifies and localizes each instance of an object (like street signs, cars, persons, etc.) within the field of view. Object detection aims at identifying the object in the images [2]. Object detection intends to identify each sample of the predefined classes and deliver its coarse localization in images by axis-aligned boxes [3]. It is seen as a supervised learning issue. Modern object detection methods have access to large labelled images for training and were assessed on different canonical benchmarks. Likewise, other tasks like motion prediction, classification, scene understanding, segmentation, etc. [4]. Were the basic issues in computer vision. Initial object detection techniques were constructed as an ensemble of handcrafted feature extractors like Histogram of Oriented Gradients (HOG), Viola-Jones detector, etc. Such techniques are inaccurate, slow, and poorly performed on unknown datasets [5].

Conversely, pollution relevant to solid waste mismanagement becomes a global concern [6]. In the past, the enormous production of disposable goods has led to a significant rise in waste; the European household waste report stated that 5.2 tonnes of waste are produced per inhabitant [7]. Also, by the year 2050, the World Bank predicts that it may exceed 3 billion tons per annum. These principles detail that various kinds of litter were dispersed freely in environments. Plastic waste is a main concern since it presents long-term environmental harm [8]. To prevent environmental pollution and, consequently, protect wild organisms and human life, immediate measures are necessary to enable wise segregation and collection of garbage. Machine learning (ML) is one way to support waste sorting [9]. Recently ML-related systems that can support sorting processes have been applied, hastening this procedure a result. After the success of implementing deep convolutional neural networks (DCNN) for image classification, object detection reached significant progress depending on deep learning (DL) methods [10]. The new DL-based techniques outperformed the conventional detection techniques by huge margins. Deep CNN is a biologically inspired structure for calculating hierarchical features.

This study designs an Entropy-based Feature Fusion using Deep Learning for Waste Object Detection and Classification (EFFDL-WODC) model. In the presented EFFDL-WODC system, two major procedures can be involved, such as waste object detection and waste object classification. For object detection, the EFFDL-WODC technique uses a YOLOv7 object detector with a fusion-based backbone network. In addition, entropy feature fusion-based models such as VGG-16, SqueezeNet, and NASNet models are used. Finally, the EFFDL-WODC technique uses a graph convolutional network (GCN) model executed for the classification of detected waste objects. The performance validation of the EFFDL-WODC algorithm was validated on the benchmark dataset.

2 Literature Review

Majchrowska et al. [11] introduced an innovative benchmark dataset to *classify waste and detect waste* were merged collections from the aforementioned open-source dataset with unified annotations that cover almost every waste category: *non-recyclable, bio, plastic and metal, other, glass, paper, and unknown*. Eventually, presented a 2-stage detector for classification and litter localization. To categorize the detected waste into 7 categories, EfficientDet-D2 was utilized for localizing EfficientNet-B2 and litter. In [12], a smart waste management system was developed using TensorFlow and LoRa communication protocol-related DL method. This object detection technique was trained with waste images to make frozen inference graphs utilized for detecting an object that is done using cameras. Rahman et al. [13] introduced a method that presents an astute way to sort indigestible and digestible waste utilizing a CNN, a common DL paradigm.

In [14], devised ABRM-MGWODL approach targets to efficiently categorize and identify the waste materials to allow effective biomass recycling. The projected system follows 2 major procedures it can be waste object classification and waste object detection. The YOLOv4 approach was utilized in this study for the waste object detection and recognition procedure. Then, to classify recognized waste materials, the GCN approach was utilized. Lastly, with the modified grey wolf optimization approach, hyperparameter tuning of the GCN method was executed effectually, thereby enhancing the method's classifier outcome. Melinte et al. [15] presented a study to enrich the performance of CNN object detectors used to identify municipal wastes. To gain a precise and fast CNN structure, many kinds of Regional Proposal Networks (RPN) and Single Shot Detectors (SSD) were fine-tuned on the TrashNet database.

Niu et al. [16] modelled an innovative DL technique for solid waste mapping in higher resolution images. By incorporating a Swin-Transformer and a multi-scale dilated CNN, both global and local features have been aggregated. Reference [17] introduced a custom-built test rig to simulate real waste-gathered trucks that might be utilized to build the datasets and test different sensors. For the classifier task, CNNs were trained to 100% achieve accuracy.

3 The Proposed Model

In this article, we have presented a novel EFFDL-WODC technique for accurate and automated detection and classification of waste objects. The projected EFFDL-WODC technique inherits the concepts of feature fusion and DL approaches for the effectual recognition and classification of various kinds of waste objects. In the projected EFFDL-WODC system, two major procedures can be contained, such as YOLOv7-based waste object detection and GCN-based waste object classification. Fig. 1 illustrates the workflow of the EFFDL-WODC system.

3.1 Object Detector: YOLO-v7 Model

In the presented EFFDL-WODC technique, the YOLO-v7 object detector is used in this study. YOLOv7 recommends a re-parameterized convolution that contained concatenation or residual connections [18]; its RepConv could not take the same connection. RepConvN, which does not have any identity connections, serves as a proper replacement for it under this condition. Within the single convolution layer, RepConv exploits a fusion of identity connections, 3×3 , and 1×1 convolutions. To design the structure of re-parameterized convolutional, the author applied RepConv without identity connections (RepConvN) after carrying out the study on the fusion of RepConv with several structures and the resultant efficiency of individual combinations. Based on the outcomes of the research work, it must not be some identity connection if the convolution layer, which comprises residual or concatenation, can be exchanged by re-parameterized convolutional. According to the structural diagram, the YOLOv7 network can be broken down into 3 various elements such as input, backbone, and head networks. The length and width of the mapping feature can be continually cut in half by Conv, BatchNorm and a SiLU (CBS) composite element, the efficient layer aggregation network (ELAN), and MP elements. Simultaneously, the count of output channels can be increased, equivalent to twice the count of input channels. The CBS combined element applied the convolutional + BN + activation function on the input mapping feature. It is proposed that utilize the ELAN element. At last, the output in the ELAN element takes twice as several channels as input. Afterwards, in the 1st convolutional, the breadth and length of the mapping feature can be cut in half by the lower branch, but the kernel size and stride can be improved by 1 and 2, correspondingly. The 2 levels of trees can be combined as one.

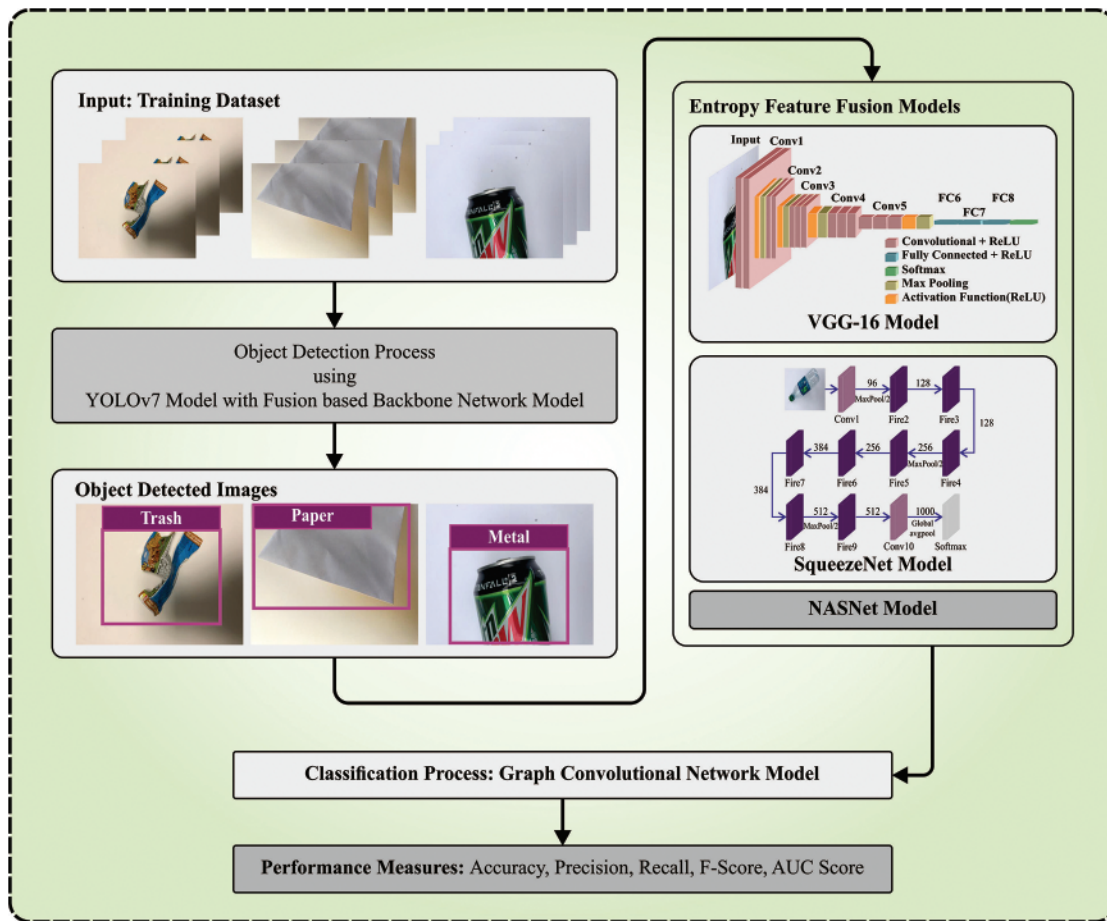


Figure 1: Workflow of EFFDL-WODC approach

Most explanation programs' output it is determining in the YOLO design that generates a single text document with annotation for every image. All the text files have annotations containing a bounding box called "BBox" for every graphical element which are demonstrated in the image. The scale of annotations is changed to it can be proportional to images, and its values differ from 0 in every system up to 1. Eqs. (1)–(6) is assisted as the basis for the adjustment method utilized in the computation utilizing the YOLO format.

$$dw = \frac{1}{W} \tag{1}$$

$$x = \frac{(x_1 + x_2)}{2} \times dw \tag{2}$$

$$dh = \frac{1}{H} \tag{3}$$

$$y = \frac{(y_1 + y_2)}{2} \times dh \tag{4}$$

$$w = (x_2 - x_1) \times dw \tag{5}$$

$$h = (y_2 - y_1) \times dh \quad (6)$$

H signifies the height of images, dh denotes the absolute height of images, W implies the width of images, and dw stands for the absolute width of pictures.

Algorithm 1: Pseudocode of YOLO

```

model = YOLOv7() // Initialize the YOLOv7 model
model.load_weights("YOLOv7.weights") // Load the pre-trained weights
image = load_image("test.jpg")
detections = model.detect(image)
for detection in detections:
    class_label = detection.class_label
    confidence = detection.confidence
    xmin, ymin, xmax, ymax = detection.bbox
    draw_bbox(image, xmin, ymin, xmax, ymax, class_label, confidence)
display_image(image)
  
```

3.2 Backbone Network: Feature Fusion-Based DL Model

In the backbone network of the YOLO-v7 model, the entropy-based feature fusion process is employed. Entropy-based feature fusion is a method used in ML to fuse features from various sources or models. The objective of feature fusion is to greater the robustness and accuracy of the method through additional data from various sources. In the entropy-based feature fusion method, the extracted feature from the models is estimated based on their entropy, a measure of the sum of randomness or uncertainty in the feature set. The feature with the lowest entropy contains more predictable and structured information, whereas the feature with high entropy contains more unpredictable or random data. The entropy of all the feature sets is evaluated, and the value is normalized to guarantee that they are on a similar scale. Then, the normalized entropy value is weighted dependent on the importance and summed to construct the weighted entropy value. The feature set with the lower weighted entropy value is selected as the combined feature set and utilized as input to the last model for the classification model. The sum of weights is equal to 1, (i.e., $\text{weight1} + \text{weight2} + \text{weight3} = 1$).

3.2.1 VGG-16 Model

VGG16 network is the first model applied for extracting feature sequences from the image. The structure of the convolution layer is similar to the VGG16 model [19], but a further block with 2 convolution and max pooling layers is used for downsampling the height of feature maps to 1 and extracting additional abstract features. To downsample the feature map, all the convolutional layers are followed by the Maxpooling layer. The typical VGG16 model exploits 2×2 max pooling through the entire network and 3×3 kernels in the convolution layer. VGG16 is a successive feature extraction that exploits a stack of max pooling and convolution layers for extracting the feature from the topmost layer to the bottommost layer. As mentioned before, the maximum depth of the VGG16 model degrades the network performance while it is training and causes the problem of disappearing gradients. This is because carrying out repetitive multiplication on the gradient makes their values very smaller while back propagated to the initial layer. To overcome these problems, a novel VGG16 model is developed, which integrates shortcut connections. The difference between the structure of convolution layers of

the VGG16 and the new VGG16 model with shortcut connection. The output feature vector of typical VGG16 architecture was determined as follows:

$$o = f(x, W) \quad (7)$$

From the expression, W indicates the weight parameter of the learned feature, and x denotes the input vector of the earlier layer. f refers to the mapping function that learns the better value of W and maps x into o . Furthermore, x is added afterwards to the f function. Therefore, in the presented VGG16 model, the output feature vector o can be formulated as follows:

$$o = F(x, \{W_i\}) + x_i \quad (8)$$

In the above equation, x_i indicates the resultant feature vector of the initial convolution layer at i -th blocks, and W_i denotes the weight parameter of i -th convolutional layers.

3.2.2 SqueezeNet Model

The SqueezeNet model function as a feature extracted and allows the input image to progress forward; it still attains a layer which is previously set (extraction feature layer) [20]. The procedure ends now, with the final layer output applied as a feature. The SqueezeNet of pretrained CNN-DL can be utilized. The SqueezeNet aims to construct a small NN with any parameters that are appropriate to computer memory and is very simply transferred through a computer network. The key component of SqueezeNet is named the fire module. A fire module includes the convolutional layer with “squeeze” and “expand” layers. Firstly, the input images are passed through an individual convolution layer termed “conv1”. A squeeze convolution layer has a single filter and is provided as an expanded layer that contains a fusion of 1×1 and 3×3 convolutions that capture spatial data (feature extraction) at different scales. This layer and eight “fire modules” are totalled “fire2” via “fire9”. Afterwards, in layers fire4, conv1, conv10, and fire8, max-pooling is implemented with the stride of 2. A dropout layer is added later in the Fire9 model to decrease over-fitting.

Algorithm 2: Pseudocode of entropy-based feature fusion process

Input: Initialization of DL

model1 = VGG-16()

model2 = SqueezeNet()

model3 = NASNet()

input_data = load_data(“input_data.csv”)

features1 = VGG-16.extract_features(input_data)

features2 = SqueezeNet.extract_features(input_data)

features3 = NASNet.extract_features(input_data)

compute entropy1 = calculate_entropy(features1)

compute entropy2 = calculate_entropy(features2)

compute entropy3 = calculate_entropy(features3)

Normalize the entropy values

entropy1_norm = normalize(entropy1)

entropy2_norm = normalize(entropy2)

entropy3_norm = normalize(entropy3)

weighted_sum = (entropy1_norm * weight1) + (entropy2_norm * weight2) + (entropy3_norm * weight3)

if weighted_sum == min(weighted_sum):

(Continued)

Algorithm 2 (Continued)

```

    fused_features = features1 + features2 + features3
else
    fused_features = features1 - features2 - features3
output_data = process_data(fused_features)
Output: Fused features

```

3.2.3 NASNet Model

NASNet (Neural Architecture Search Network) is a family of DNNs that was proposed by a reinforcement learning method to find an optimum NN structure [21]. The objective of NASNet is to automatically design NN architecture that performs better than human-designed architecture. NASNet has two building blocks: a search algorithm and a search space. The search space determines the set of potential NN architecture, and the search algorithm examines this space to search for a better structure. The search space in NASNet can be determined by the key components, a small NN module merged to form large architecture. These key components are intended to be both efficient and flexible, allowing them to be fused in many dissimilar ways to construct a broad range of structures. The search algorithm in NASNet exploits an RL method, where a NN agent is trained to construct new architecture and estimate their performance. The agent learns to explore the search space by producing new architecture that is the same as before. The agent is trained by a reward signal that is based on the performance of all the architectures.

3.3 Object Classification: GCN Model

For the automated classification of detected objects, the GCN model is exploited in this work. A typical CNN utilizes convolutional functions on a (multi-dimensional) array(s) containing a spatial meaning [22]. The CNN can be generally utilized to classifier drives like image recognition, while the images are realized as matrices from the Euclidean space. CNNs are demonstrated great efficiency in signal processing and visual analytics because of their innate ability to manage these types of structures, extracting meaningful features shared with data and utilized in various studies. Fig. 2 represents the framework of GCN. Afterwards, classification of appropriate functions and/or data transformation. During this case, the purpose of GCN is to offer a representation of nodes utilizing similar node features, among them the neighbour's node features. The outcome of the GCN technique is usually calculated as:

$$Y = \tilde{A}XW \quad (9)$$

Whereas X denotes the input data (that is, the water demand), the term Y signifies the output, but W defines the matrix \vec{o} with model parameters. Additionally, \tilde{A} represents the normalization adjacent matrix that is expressed as:

$$\tilde{A} = (\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}) \quad (10)$$

With $\hat{A} = A + I$, and whereas A represents the adjacent matrix relying on the graph but the connects signifies the correlation among the various time series, I denotes the identity matrix, and \hat{D} represents the diagonal degree matrix of \hat{A} .

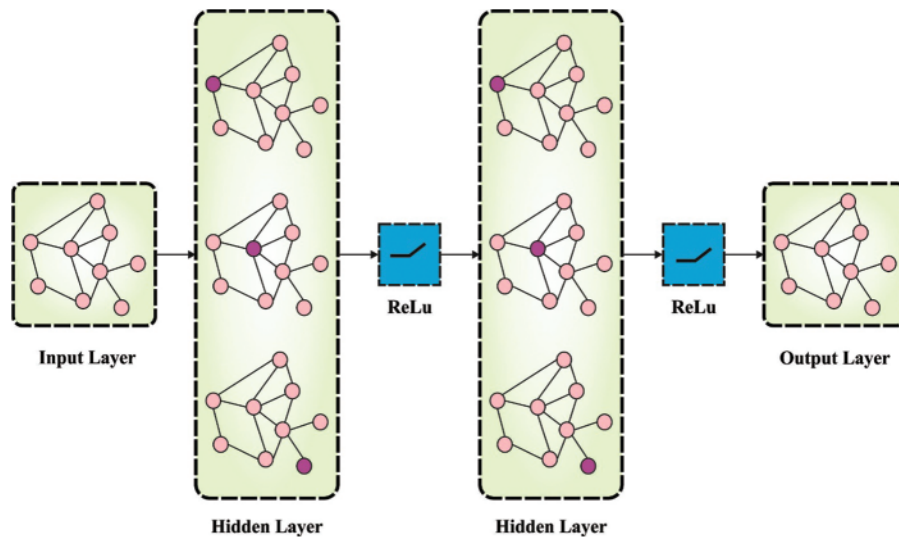


Figure 2: Architecture of GCN

4 Performance Validation

The proposed model is simulated using Python 3.6.5 tool on PC i5-8600k, GeForce 1050Ti 4 GB, 16 GB RAM, 250 GB SSD and 1 TB HDD. The parameter settings are given as follows: learning rate: 0.01, dropout: 0.5, batch size: 5, epoch count: 50, and activation: ReLU.

In this section, the waste object detection and classification performance of the EFFDL-WODC technique is examined on a database in the Kaggle repository [23], comprising 2467 instances with 6 class labels as represented in Table 1. Fig. 3 represents the sample images. Fig. 4 demonstrates the sample visualization outcomes.

The confusion matrices of the EFFDL-WODC algorithm are represented in Fig. 5. The outcomes imply that the EFFDL-WODC technique identifies six types of objects proficiently.

Table 1: Details of the database

Class	No. of instances
Cardboard	393
Glass	491
Metal	400
Paper	584
Plastic	472
Trash	127
Total no. of instances	2467

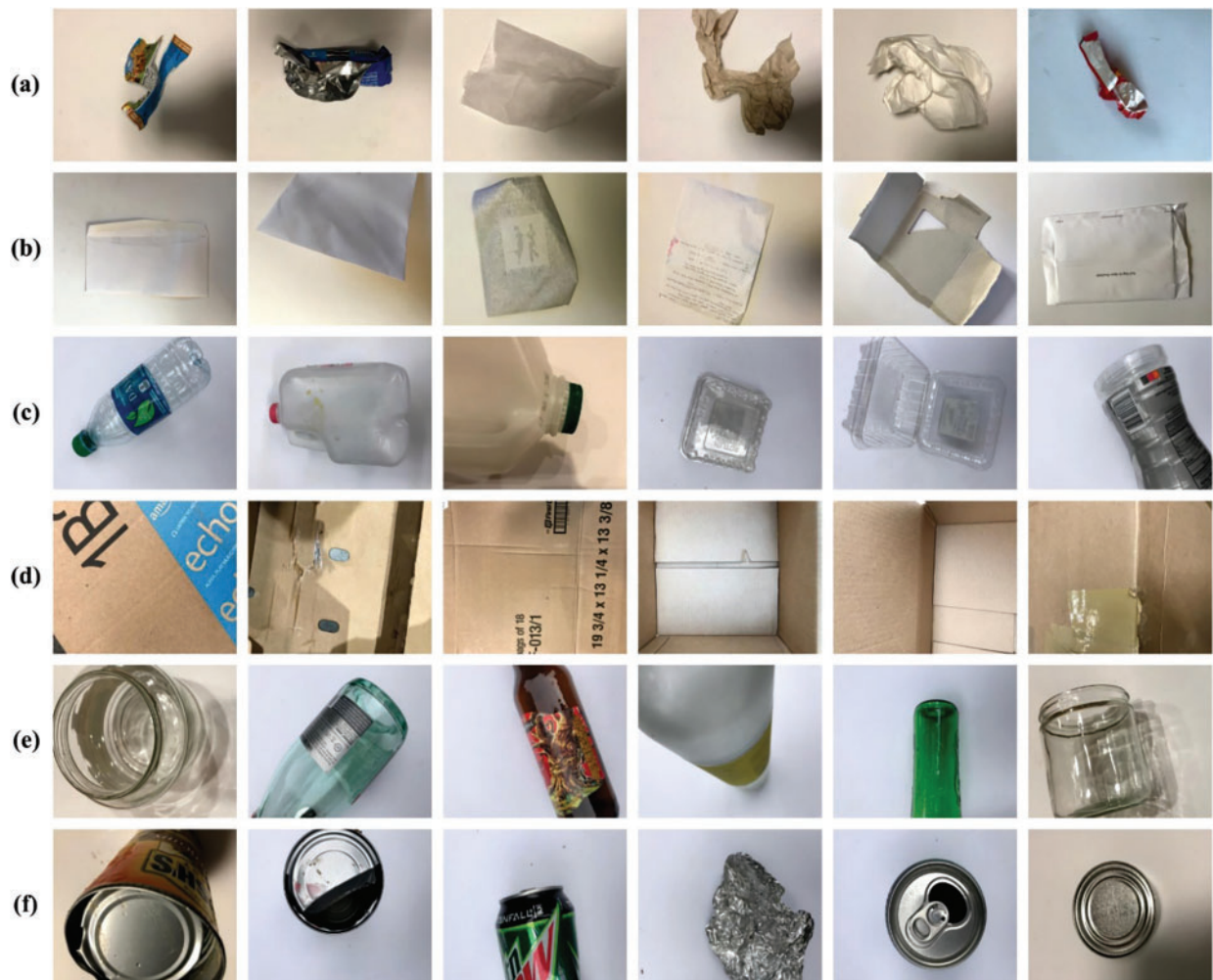


Figure 3: Sample images (cardboard, glass, metal, paper, plastic, trash)

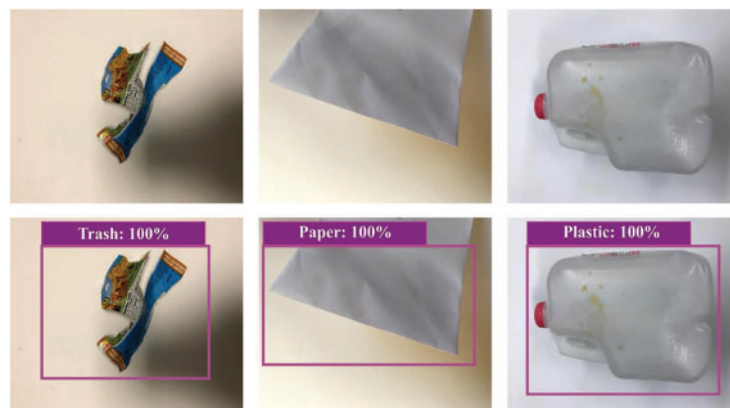


Figure 4: (Continued)

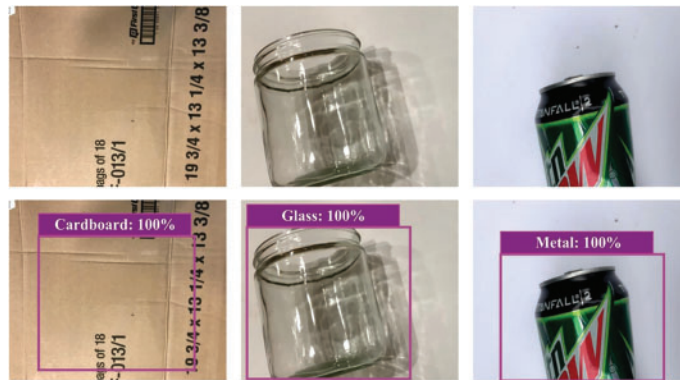


Figure 4: Sample visualization results

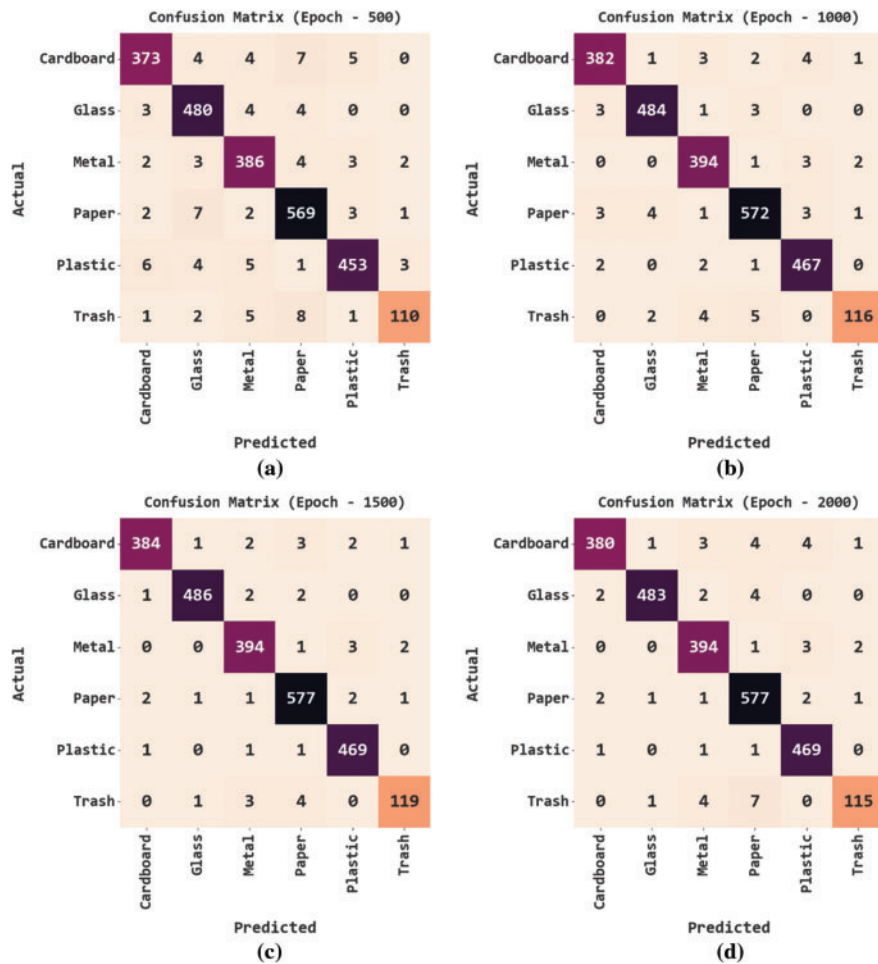


Figure 5: Confusion matrices of EFDL-WODC system (a) Epoch500, (b) Epoch1000, (c) Epoch1500, and (d) Epoch2000

In Table 2, an overall classifier outcome of the EFFDL-WODC technique with different epochs is given. The results imply the effectual results of the EFFDL-WODC technique under each epoch. For instance, with 500 epochs, the EFFDL-WODC technique gains $accu_y$ of 98.70%, $prec_n$ of 95.94%, $reca_l$ of 94.87%, F_{score} of 95.37%, and AUC_{score} of 97.03%. Meanwhile, with 1000 epochs, the EFFDL-WODC system gains $accu_y$ of 99.30%, $prec_n$ of 97.72%, $reca_l$ of 97.08%, F_{score} of 97.39%, and AUC_{score} of 98.33%. Concurrently, with 1500 epochs, the EFFDL-WODC approach reaches $accu_y$ of 99.49%, $prec_n$ of 98.25%, $reca_l$ of 97.84%, F_{score} of 98.04%, and AUC_{score} of 98.76%. At last, with 2000 epochs, the EFFDL-WODC methodology attains $accu_y$ of 99.34%, $prec_n$ of 97.88%, $reca_l$ of 97.05%, F_{score} of 97.44%, and AUC_{score} of 98.32%.

Table 2: Classification outcome of EFFDL-WODC approach with varying epochs

Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	AUC_{Score}
Epoch—500					
Cardboard	98.62	96.38	94.91	95.64	97.12
Glass	98.74	96.00	97.76	96.87	98.37
Metal	98.62	95.07	96.50	95.78	97.77
Paper	98.42	95.95	97.43	96.69	98.08
Plastic	98.74	97.42	95.97	96.69	97.69
Trash	99.07	94.83	86.61	90.53	93.18
Average	98.70	95.94	94.87	95.37	97.03
Epoch—1000					
Cardboard	99.23	97.95	97.20	97.57	98.41
Glass	99.43	98.57	98.57	98.57	99.11
Metal	99.31	97.28	98.50	97.89	98.98
Paper	99.03	97.95	97.95	97.95	98.65
Plastic	99.39	97.90	98.94	98.42	99.22
Trash	99.39	96.67	91.34	93.93	95.58
Average	99.30	97.72	97.08	97.39	98.33
Epoch—1500					
Cardboard	99.47	98.97	97.71	98.34	98.76
Glass	99.68	99.39	98.98	99.18	99.41
Metal	99.39	97.77	98.50	98.13	99.03
Paper	99.27	98.13	98.80	98.46	99.11
Plastic	99.59	98.53	99.36	98.95	99.51
Trash	99.51	96.75	93.70	95.20	96.76
Average	99.49	98.25	97.84	98.04	98.76
Epoch—2000					
Cardboard	99.27	98.70	96.69	97.69	98.23
Glass	99.55	99.38	98.37	98.87	99.11
Metal	99.31	97.28	98.50	97.89	98.98
Paper	99.03	97.14	98.80	97.96	98.95
Plastic	99.51	98.12	99.36	98.74	99.46

(Continued)

Table 2 (continued)

Class	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	AUC_{Score}
Trash	99.35	96.64	90.55	93.50	95.19
Average	99.34	97.88	97.05	97.44	98.32

Fig. 6 scrutinizes the accuracy of the EFFDL-WODC approach during the training and validation process on varying epochs. The figure notifies that the EFFDL-WODC technique gains maximal accuracy values over enhancing epochs. In addition, the higher validation accuracy over training accuracy depicts that the EFFDL-WODC system learns efficiently on varying epochs.

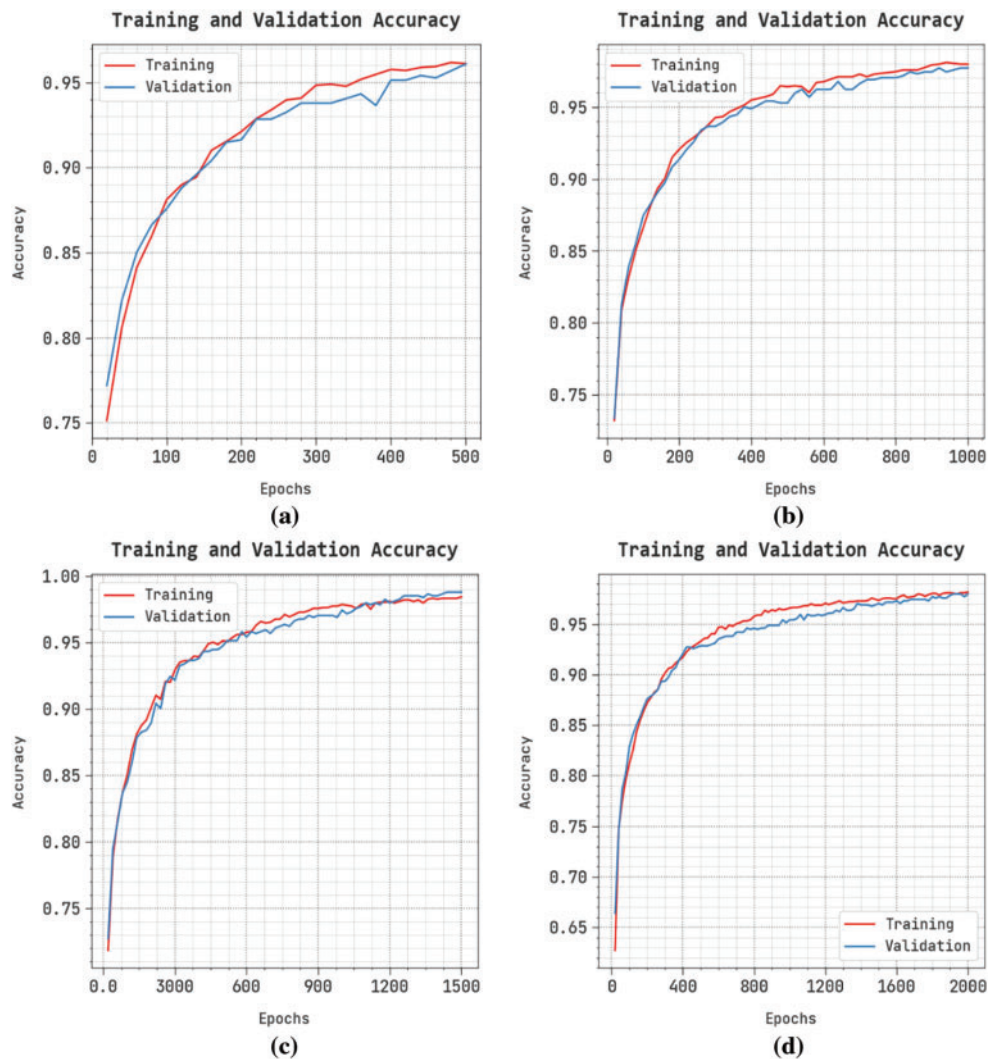


Figure 6: Accuracy curve of EFFDL-WODC approach (a) Epoch500, (b) Epoch1000, (c) Epoch1500, and (d) Epoch2000

The loss investigation of the EFFDL-WODC algorithm at the time of training and validation is exhibited on varying epochs in Fig. 7. The outcomes stated that the EFFDL-WODC algorithm gains closer values of training and validation loss. It can be inferred that the EFFDL-WODC method learns efficiently on varying epochs.

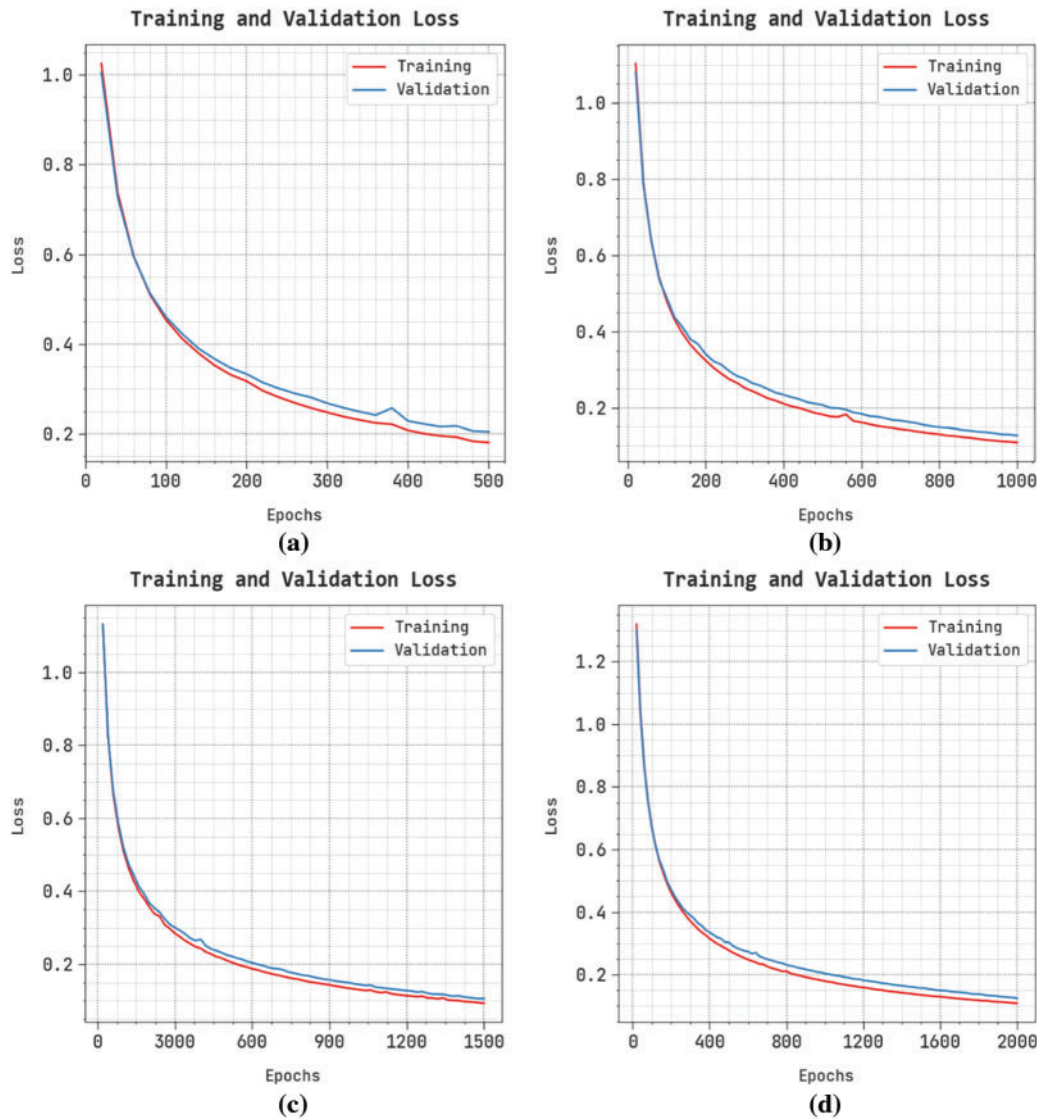


Figure 7: Loss curve of EFFDL-WODC approach (a) Epoch500, (b) Epoch1000, (c) Epoch1500, and (d) Epoch2000

A brief precision-recall (PR) curve of the EFFDL-WODC system is established on varying epochs in Fig. 8. The results state that the EFFDL-WODC approach results in maximal values of PR. Also, the EFFDL-WODC methodology can obtain superior PR values in all classes.

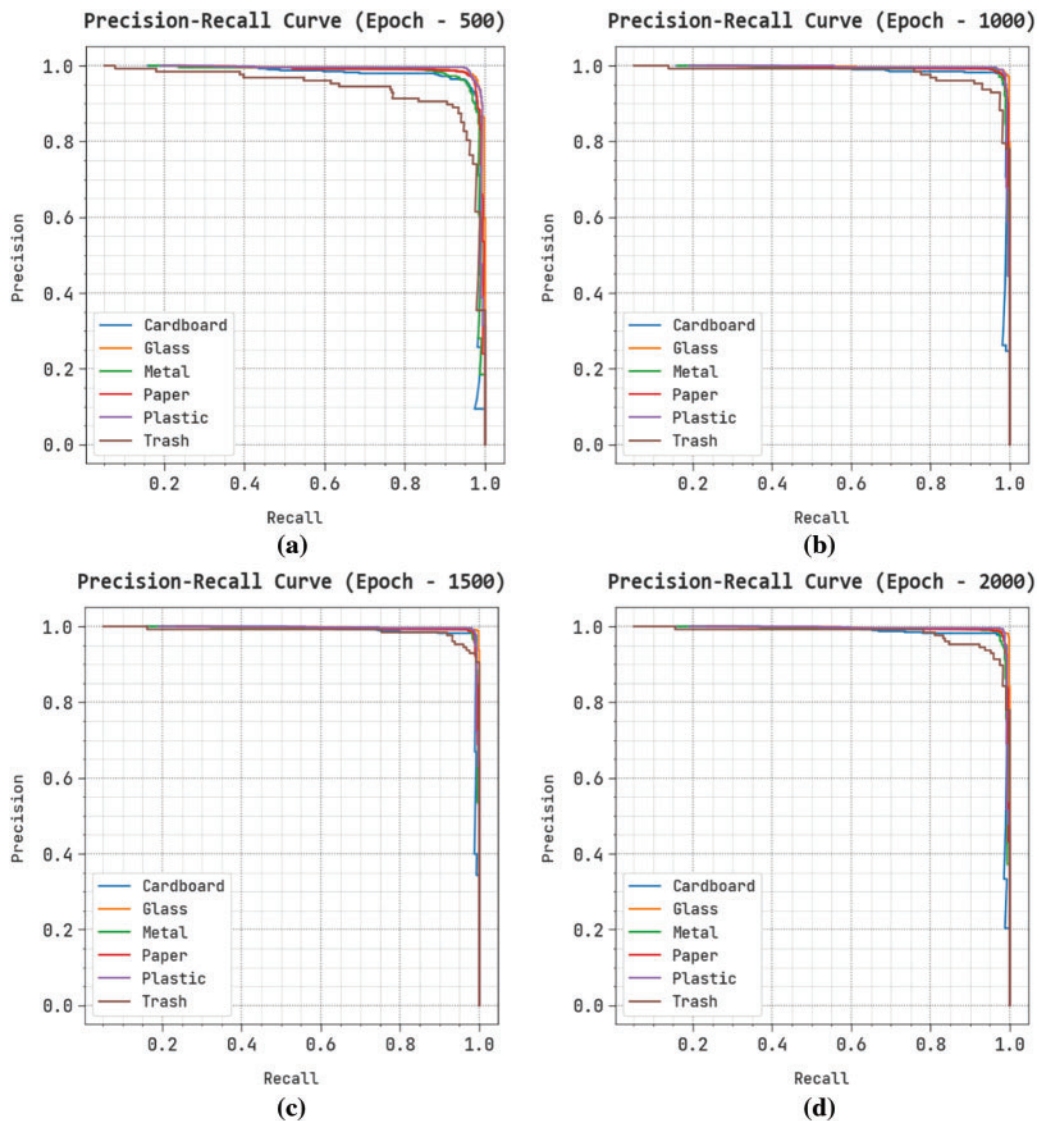


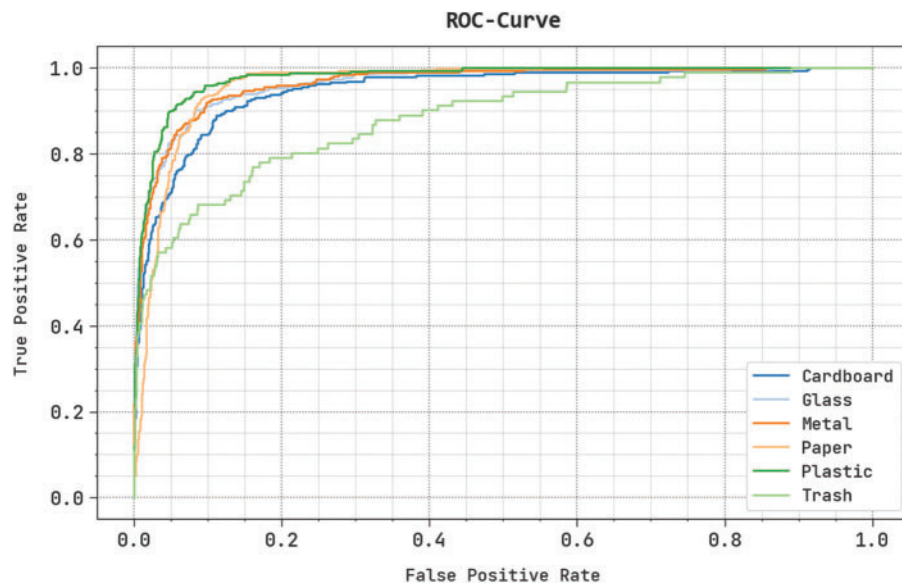
Figure 8: PR curve of EFFDL-WODC approach (a) Epoch500, (b) Epoch1000, (c) Epoch1500, and (d) Epoch2000

A comprehensive comparative analysis of the EFFDL-WODC system with other DL techniques is given in Table 3 [5]. The results depicted the ineffectual outcomes of the AlexNet model, whereas the ResNet50 and VGG16 approaches have gained somewhat improvised results. Then, the MLH-CNN, ARBM-MGWODL, and DLSODC-GWM techniques have portrayed considerable performance. However, the EFFDL-WODC and technique shows maximum outcomes with $accu_y$ of 99.49%, $prec_n$ of 98.25%, $reca_l$ of 97.84%, and F_{score} of 98.04%. These results pointed out the enhanced classification outcomes of the EFFDL-WODC technique.

Table 3: Comparative outcome of EFFDL-WODC methodology with other DL systems [5,14]

Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}
EFFDL-WODC	99.49	98.25	97.84	98.04
DLSODC-GWM	98.32	94.67	94.66	94.61
ABRM-MGWODL	99.01	96.85	96.81	96.82
MLH-CNN	92.37	90.81	91.36	90.78
AlexNet	52.59	41.91	49.86	43.39
RestNet50	75.12	71.82	71.92	71.155
VGG16	73.13	68.76	68.39	68.06

In Fig. 9, a ROC study of the EFFDL-WODC approach is revealed on the test database. The outcome stated that the EFFDL-WODC method led to improved ROC values. Besides, it can be clear that the EFFDL-WODC algorithm can extend improved ROC values on all classes.

**Figure 9:** ROC curve of the EFFDL-WODC approach

5 Conclusion

In this study, we have presented a novel EFFDL-WODC system for accurate and automated detection and classification of waste objects. The presented EFFDL-WODC technique inherits the concepts of feature fusion and DL approaches for the effectual recognition and classification of various kinds of waste objects. In the projected EFFDL-WODC system, two major procedures are contained, such as YOLOv7-based waste object detection and GCN-based waste object classification. For object detection, the EFFDL-WODC technique uses a YOLOv7 object detector with a fusion-based backbone network. In addition, entropy feature fusion-based models such as VGG-16, SqueezeNet, and NASNet models are used. Finally, the EFFDL-WODC technique uses a GCN model is carried out for the classification of detected waste objects. The performance validation of the EFFDL-WODC

system was validated on the benchmark dataset. The comprehensive comparison outcomes highlighted the improved performance of the EFFDL-WODC technique over recent approaches.

Acknowledgement: The authors gratefully acknowledge the technical and financial support provided by the Ministry of Education and Deanship of Scientific Research (DSR), King Abdulaziz University (KAU), Jeddah, Saudi Arabia.

Funding Statement: This research work was funded by Institutional Fund Projects under Grant No. (IFPIP: 557-135-1443).

Author Contributions: Conceptualization, E.B.A. and M.R.; Methodology, E.B.A., S.J. and R.B.A.; Software, E.B.A. and S.J.; Validation, E.B.A. and S.J.; Formal analysis, E.B.A. and M.R.; Investigation, E.B.A., M.R. and R.B.A.; Data curation, E.B.A. and S.J.; Writing—original draft, E.B.A. and M.R.; Writing—review & editing, E.B.A. and S.J.; Supervision, E.B.A.; Project administration, M.R.; Funding acquisition, E.B.A. All authors have read and agreed to the published version of the manuscript.

Availability of Data and Materials: The data presented in this study are available in this article.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] J. Bobulski and M. Kubanek, “Deep learning for plastic waste classification system,” *Applied Computational Intelligence and Soft Computing*, vol. 2021, pp. 1–7, 2021.
- [2] R. Azadnia, S. Fouladi and A. Jahanbakhshi, “Intelligent detection and waste control of hawthorn fruit based on ripening level using machine vision system and deep learning techniques,” *Results in Engineering*, vol. 17, no. 1, pp. 100891, 2023.
- [3] H. Panwar, P. K. Gupta, M. K. Siddiqui, R. M. Menendez, P. Bhardwaj *et al.*, “AquaVision: Automating the detection of waste in water bodies using deep transfer learning,” *Case Studies in Chemical and Environmental Engineering*, vol. 2, no. 57, pp. 100026, 2020.
- [4] S. Kumar, D. Yadav, H. Gupta, O. P. Verma, I. A. Ansari *et al.*, “A novel YOLOv3 algorithm-based deep learning approach for waste segregation: Towards smart waste management,” *Electronics*, vol. 10, no. 1, pp. 14, 2020.
- [5] F. S. Alsubaei, F. N. Al-Wesabi and A. M. Hilal, “Deep learning-based small object detection and classification model for garbage waste management in smart cities and iot environment,” *Applied Sciences*, vol. 12, no. 5, pp. 2281, 2022.
- [6] C. Wang, J. Qin, C. Qu, X. Ran, C. Liu *et al.*, “A smart municipal waste management system based on deep-learning and Internet of Things,” *Waste Management*, vol. 135, no. 3, pp. 20–29, 2021.
- [7] A. Alshammari and R. C. Chabaan, “Sppn-Rn101: Spatial pyramid pooling network with resnet101-based foreign object debris detection in airports,” *Mathematics*, vol. 11, no. 4, pp. 841, 2023.
- [8] Z. Kang, J. Yang, G. Li and Z. Zhang, “An automatic garbage classification system based on deep learning,” *IEEE Access*, vol. 8, pp. 140019–140029, 2020.
- [9] X. Li, T. Li, S. Li, B. Tian, J. Ju *et al.*, “Learning fusion feature representation for garbage image classification model in human-robot interaction,” *Infrared Physics & Technology*, vol. 128, no. 1, pp. 104457, 2023.
- [10] A. P. Saputra and Kusriani, “Waste object detection and classification using deep learning algorithm: YOLOv4 and YOLOv4-tiny,” *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 14, pp. 1666–1677, 2021.

- [11] S. Majchrowska, A. Mikołajczyk, M. Ferlin, Z. Klawikowska, M. A. Plantykov *et al.*, “Deep learning-based waste detection in natural and urban environments,” *Waste Management*, vol. 138, no. 6, pp. 274–284, 2022.
- [12] T. J. Sheng, M. S. Islam, N. Misran, M. H. Baharuddin, H. Arshad *et al.*, “An internet of things based smart waste management system using lora and tensorflow deep learning model,” *IEEE Access*, vol. 8, pp. 148793–148811, 2020.
- [13] M. W. Rahman, R. Islam, A. Hasan, N. I. Bithi, M. M. Hasan *et al.*, “Intelligent waste management system using deep learning with IoT,” *Journal of King Saud University—Computer and Information Sciences*, vol. 34, no. 5, pp. 2072–2087, 2022.
- [14] S. A. Althubiti, S. K. Sen, M. A. Ahmed, E. L. Lydia, M. Alharbi *et al.*, “Automated biomass recycling management system using modified grey wolf optimization with deep learning model,” *Sustainable Energy Technologies and Assessments*, vol. 55, no. 6, pp. 102936, 2023.
- [15] D. O. Melinte, A. M. Travediu and D. N. Dumitriu, “Deep convolutional neural networks object detector for real-time waste identification,” *Applied Sciences*, vol. 10, no. 20, pp. 7301, 2020.
- [16] B. Niu, Q. Feng, J. Yang, B. Chen, B. Gao *et al.*, “Solid waste mapping based on very high resolution remote sensing imagery and a novel deep learning approach,” *Geocarto International*, vol. 38, no. 1, pp. 2164361, 2023.
- [17] O. I. Funch, R. Marhaug, S. Kohtala and M. Steinert, “Detecting glass and metal in consumer trash bags during waste collection using convolutional neural networks,” *Waste Management*, vol. 119, pp. 30–38, 2021.
- [18] C. Dewi, A. P. S. Chen and H. J. Christanto, “Deep learning for highly accurate hand recognition based on YOLOv7 model,” *Big Data and Cognitive Computing*, vol. 7, no. 1, pp. 53, 2023.
- [19] H. Yang, J. Ni, J. Gao, Z. Han and T. Luan, “A novel method for peanut variety identification and classification by improved VGG16,” *Scientific Reports*, vol. 11, no. 1, pp. 15756, 2021.
- [20] H. Alhichri, Y. Bazi, N. Alajlan and B. Bin Jdira, “Helping the visually impaired see via image multi-labeling based on squeezeNet CNN,” *Applied Sciences*, vol. 9, no. 21, pp. 4656, 2019.
- [21] E. Cano, J. Mendoza-Avilés, M. Areiza, N. Guerra, J. L. Mendoza-Valdés *et al.*, “Multi skin lesions classification using fine-tuning and data-augmentation applying NASNet,” *PeerJ Computer Science*, vol. 7, no. 1, pp. e371, 2021.
- [22] A. Zanfei, B. M. Brentan, A. Menapace, M. Righetti and M. Herrera, “Graph convolutional recurrent neural networks for water demand forecasting,” *Water Resources Research*, vol. 58, no. 7, pp. 1–14, 2022.
- [23] <https://www.kaggle.com/datasets/asdasdasdasdas/garbage-classification>