International Journal of Innovative Research in Computer Science & Technology (IJIRCST)
ISSN: 2347-5552, Volume-10, Issue-4, July 2022
https://doi.org/10.55524/ijircst.2022.10.4.22
Article ID IRP1296, Pages 183-191
www.ijircst.org

Deep Learning-based Classification of Materials into Biodegradable and Non-Biodegradable

Aaqib Rashid Bhat[1], Monika Mehra[2], Ravinder Pal Singh[3]

[1]M. Tech Scholar, Department of Electronics and Communication Engineering, RIMT University, Mandi Gobindgarh, Punjab, India
[2]Head, Department of Electronics and Communication Engineering, RIMT University, Mandi Gobindgarh, Punjab, India
[3]Professor, Department of Electronics and Communication Engineering, RIMT University, Mandi Gobindgarh, Punjab, India

Correspondence should be addressed to Aaqib Rashid Bhat; aaqibbhat76@gmail.com

ABSTRACT- The collection of real-time data from people, their cars, public transit, buildings, and other urban infrastructures like the energy grid and waste management systems are at the heart of the smart city concept. The insights gathered from the data may be used by municipal authorities to manage resources and services efficiently. An important study topic at the same time is the sharp increase in environmental degradation and deterioration that causes ecological imbalance. Additionally, the development of advanced waste management systems that can categorize rubbish according to its level of biodegradability is required for the worldwide expansion of smart cities. Some of the more common ones are paper, paper boxes, food, glass, and other garbage. A cost-effective technique to separate the waste from the enormous mountain of trash and garbage and classify the waste items is to use computer vision-based technologies. Recent developments in deep learning (DL) and deep reinforcement learning (DRL) have made it possible to categorize waste objects and identify and detect trash. In this sense, the research creates an intelligence model for smart cities. The technique's goal is to recognize and classify rubbish objects using the DL and DRL approaches. The two steps of the SYSTEM technique are objected classification based on DRL and object detection based on Mask Regional Convolutional Neural Network (CNN)… The CNN model uses the DenseNet model as a baseline model, and a deep learning network (DLN) is employed as the classifier. A based hyperparameter optimizer is also created to boost the efficiency of the DenseNet model.

KEYWORDS- Biodegradable, Densenet, CNN, Classification

I. INTRODUCTION

Local governments are attempting to push the limits of waste management in smart cities to minimize municipal solid waste and boost community recycling rates. Cities spend a lot of money removing trash from public areas, therefore effective municipal waste management programs yield better outcomes. Due to the rapid acceleration of urbanization, global population expansion, and industrialization, environmental degradation has drawn more attention. Global population growth is accelerating at an alarming rate, severely degrading the ecology and creating ideal conditions. Data from 2019 [1] show that India produces more than 62 million tons (MT) of solid waste per year.

There is a growing awareness of the necessity for waste separation based on biodegradable and non-biodegradable behavior. In the context of India, waste typically consists of items like plastic, paper, metal, rubber, textiles, glass, sanitary products, organics, infectious materials (clinical and hospital), electrical and electronic waste, and hazardous substances (chemical, paint, and spray), all of which are collectively categorized as non-biodegradable (NBD) and biodegradable (BD) wastes with corresponding shares of 48 percent and 52 percent [2].

This method still relies on human factors for waste decomposition [4]. However, improvements in artificial intelligence and deep learning architecture might reduce the productivity of the system in the next years. In particular, the AI model made it possible to transmit the machine's brain control system successfully and quickly. The implementation of recycling systems based on DL frameworks for waste categorization may be unavoidable given these improvements. [5]. The old waste categorization method has several drawbacks, including its inefficiency and reliance on human judgment. The classification performance has to be enhanced even though the machine learning (ML)-based trash categorization now in use works effectively. It is evident from looking at the existing deep learning-based waste classification method that there are two key reasons why enhancing classification performance is difficult [6]. On several datasets, the various frameworks of DL techniques initially cause them to behave in different ways. The availability of large-scale databases and trash photo datasets for training algorithms, like ImageNet7, is also lacking.

A bad categorization would also have a detrimental effect on classification performance since it comprises a variety of garbage. Plastic bottles and bags have diverse characteristics and shapes despite being made of plastic, therefore their disposal strategies also vary. Deep Reinforcement Learning (DRL), a subfield of machine learning (ML), combines Deep Learning with Reinforcement Learning (RL) (DL). Numerous algorithms in the literature [7] deal with the concept of iterative learning, displaying and upgrading data to better forecast outcomes and apply it in decision-making. The scientific community teaches DL primarily to utilize Graphical

Processing Units (GPU) to speed up the research and implementation, bringing them to a level where they outperform the bulk of conventional ML approaches in video analytics. DRL techniques are extended to three dimensions to extract Spatio-temporal information from the video stream that could distinguish the objects from one another. This study creates a (SYSTEM) model for the identification and categorization of recycling waste items based on intelligence DRL. The two steps of the SYSTEM method are object detection using Convolutional Neural Networks (CNN) and object classification using DRL. The CNN model also uses a deep Q-learning network (DQLN) as a classifier in addition to the DenseNet model as a baseline model.

II. LITERATURE REVIEW

One of the problems facing the globe, garbage has an impact on every living thing. The survey states that 74% of the plastic that enters the Philippine seas comes from trash. The most frequently thrown items in the Philippines are plastic, paper, and kitchen garbage, which together account for around 35,000 tons of daily waste [1]. Furthermore, 81% of the plastic debris was discharged into the ocean from land [02]. The two main causes of trash leakage, according to this study, are illegal dumping by various businesses and disposal sites that are close to rivers [04].

Similar issues faced Manila City, the capital of the Philippines, which generates more than 8,600 tons of trash daily from its 1.5 million citizens [06]. Plastic bags accounted for 679,957 of the waste in the water, followed by paper bags with 253,013 and food wrappers with 103,226. 38,394 articles of clothes and shoes, 55,814 tobacco-related goods, including 34,154 cigarettes, lighters, and wrappers, and 11,077 diapers, were also found. You could find this trash in various bodies of water, such as Manila Bay, and it might create poison and toxins in the ocean [08].

Furthermore, the Philippines still has a waste issue despite its strong environmental activity [09]. According to Rappler's reports [10], out of the 178 local government units (LGUs) in the Manila area, there are still 39.89 percent that do not adhere to the 10-year solid waste management plan, 27.53 percent that does not adhere to the rules for source segregation, 23.03 percent that does not adhere to the rules for the segregated collection, 44.38 percent that does not have functional materials recovery facilities, and 10.11 percent that do not adhere to the rules for application [12].

III. METHODOLOGY

A. Implementation and Specifications

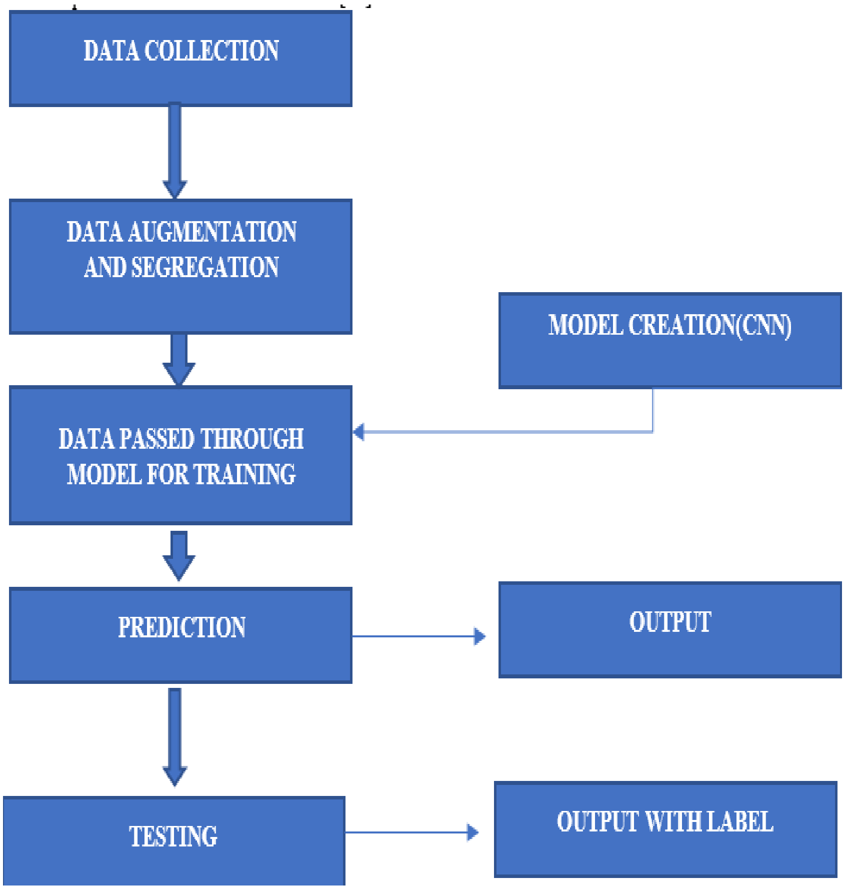

Figure 1: Flow diagram of the model

The segregation issue is solved using the supervised learning approach. First, images of various materials, including soda cans, were collected, including cardboard, glass, paper, plastic, and metal. Each of these categories has around 500 images. Each image was downsized to 64x64 and converted to grayscale from color. This is essential for changing the neural network to speed up processing and simplify the calculation. By converting

RGB images to grayscale, the network may use only black and white images. This is done because garbage color is not a crucial consideration. When the data is prepared, we provide output labels for every dataset class. To forecast the output category is done.

B. Training

The images from each category are initially converted into an array to get the data ready for training. Therefore, cardboard photographs will be stored in a NumPy array. Since the Numpy library functions well with matrices and arrays, we use it in this situation. An array of arrays will be created that contains all of the categories and their associated labels, as well as all of the photos. This array of arrays is sent into the neural network, and training is finished.

```
m,n,q = im1.shape # get the size of the images
imnbr = len(imlist) # get the number of images
print(imnbr)

immatrix = array([array(Image.open(pathtosave + im2)).flatten()
              for im2 in imlist],'f')
```

Snippet 1

Figure 2: Array of arrays

In fig. 2 is transformed into an array and saved in NumPy format.
In fig. 3, the matrix is an array of arrays of photos that is flattened into a 1-dimensional vector using the flatten function.

```
label=np.ones((num_samples,),dtype=int)
print(label)

label[0:403]=0 #cardboard
label[404:904]=1 #glass
label[904:1313]=2 #metal
label[1314:1907]=3 #paper
label[1908:2389]=4 #plastic
label[2390:]=5 #trash

data,Label=shuffle(immatrix,label,random_state=1)
train_data = [data,Label]
```

Snippet 2

Figure 3: Flatten Function

Output labels are added to the train data tuple.
The model is trained on the training set and verified on the test set, and the train and test data sets are also segregated.

C. Validation

The validation of the model is important because it tells us how well our model fits the random data in the test data set. We aim to improve validation precision. With one input convolution layer, 64x64 input pixels, and a 3x3 kernel size, Snippet 3 uses a sequential approach. The size of the matrix that traverses the image matrix, compares each pixel in the 3x3 pixel grid, and reduces it to a single pixel is fundamentally determined by the kernel size. A max-pooling layer is added to prevent convolution from causing each picture to become smaller than 64x64 by padding it with zeroes. In this instance, three convolution layers with convolution sizes of 64, 64, and 128 each are produced. . A dropout layer with a 0.5 dropout is also included. This layer aids in further increasing computation by rejecting particular neurons from the training process. A hidden layer with 128 neurons and a rectified linear activation function is now included in the model (relu). Because there are 6 classes, the output layer utilizes the softmax function and has 6 neurons. The Softmax function is used for multiclass classification.

```
model1 = Sequential()

model1.add(Convolution2D(64, (3,3), activation = 'relu' ,
                         input_shape=(img_rows, img_cols,1)))
#model1.add(Activation('relu'))
model1.add(MaxPooling2D(pool_size=(2, 2)))

model1.add(Conv2D(64,kernel_size=(2,2),activation="relu"))
model1.add(MaxPooling2D(pool_size=(2, 2)))

model1.add(Conv2D(128,kernel_size=(2,2),activation="relu"))
model1.add(MaxPooling2D(pool_size=(2, 2)))

model1.add(Dropout(0.5))

model1.add(Flatten())

model1.add(Dense(128))
model1.add(Activation('relu'))
model1.add(Dropout(0.5))
model1.add(Dense(64,activation="relu"))
model1.add(Dropout(0.25))

model1.add(Dense(6))
model1.add(Activation('softmax'))

print("model created")

model1.compile(loss='categorical_crossentropy', optimizer='adam',metrics=['accuracy'])

model1.fit(X_train, Y_train,batch_size=10,
                 epochs=25, verbose=1, validation_data=(X_test, Y_test))

model1.save("multi.h5")
```

Snippet 3

Figure 4: Sequential approach

Using the minimum dataset offered and the fundamental convolution model, we achieved an accuracy of roughly 92% on the test case and 92% on the validation case for 10 rounds. If the algorithm were run over additional epochs, the accuracy would increase considerably more. Another way to increase accuracy is to use known models like the VGG16 or the ImageNet.

IV. SYSTEM ARCHITECTURE

This study provides a method for classifying and identifying recyclable trash objects. The SYSTEM method creates a CNN using the DenseNet model to identify and conceal waste objects in the image. The DRL-based DQLN technique is also utilized to assign distinct class labels to the observed objects. The whole description of how these methods work is given in the next section.

Given an image dataset of a trash object, the formula X = Xin i=1 represents the number of training samples, with n being the total number of training samples. The training sample's labeled set is represented as c = ci, ci 1,..., C, where C is the number of garbage class images. They decide. (XLi, you) = TL NL i=1 Reject is the set of labeled training samples, where nL is the number of labeled training samples in JL, XLi is the ith labeled training sample, you are its label, and 1, 2,... C denotes its label. Similarly, indicates the unlabeled training set as Ju = XUi nU

XUi indicates the ith unlabelled training sample, whereas nU is the number of unlabelled training samples in JU, where i=1. They use f (Xi), or the predictive score of X I to calculate the output of the last layer of a DRL technique for each input picture Xi [20]. They identify the penultimate layer of a DRL method's output as xi for each input picture X I and take xi into account as the item spotted by the CNN model.

A. CNN Based Object Detection Model

The CNN model is initially used to identify any trash-related items in the picture. CNN is a simple, adaptable, and common framework for object detection, exposure, and sample segmentation that can successfully identify objects in an image while consistently producing a high-quality segmentation mask. The RPN, the second component of the Mask R-CNN, and recognition networks share all of the image convolution characteristics, hence supporting the suggestion for a mostly free region. Instead of using elective search, RPN is put into MaskR-CNN, and as a result, RPN and recognition networks both share the convolutional characteristic of a full map. Forecast combined boundaries are placed together with object scores everywhere, and a fully convolutional network may also be used (FCN).

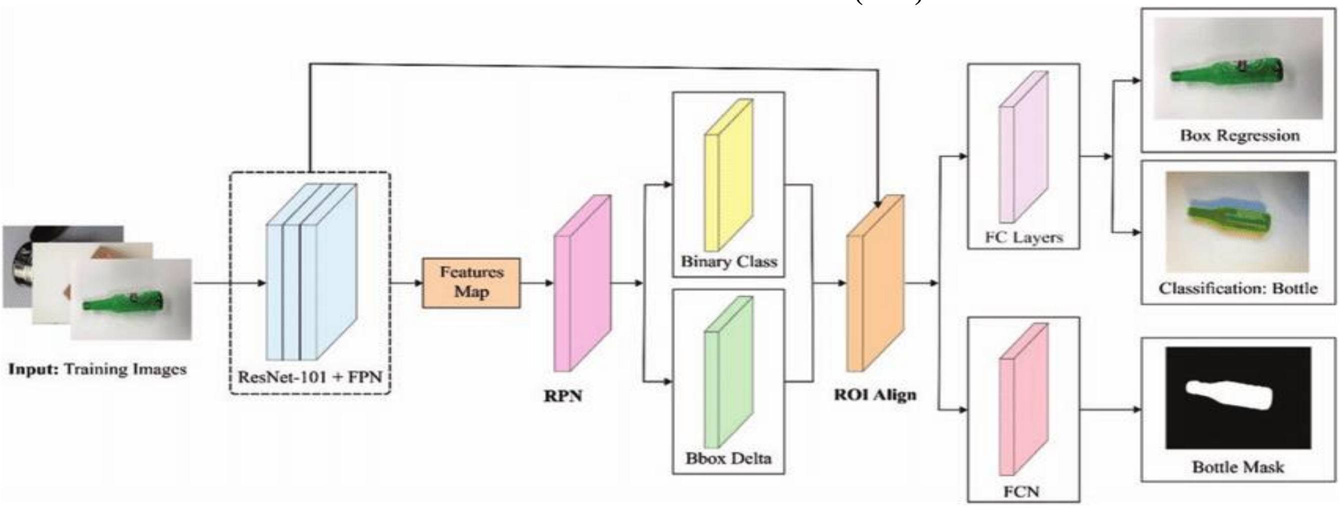

Figure 5: Structure of CNN for waste object detection[13]

As demonstrated by the given functions, the multi-task loss function is implemented during trained CNN with three components, including classifier loss of bounding box, place regression loss of bounding box and loss of mask.

L = L cls + L box + L mask
L cls =− log[pi∗pi+(1−p∗i
)(1−pi)] (2)Lbox=r(ti −t∗i) (3)
Lmask=Sigmoid (Click) (4)

where pi is the identified ROI probabilities in the classifier loss. One of the ROIs was assumed to be the foreground or zero in any other situation, using Lcls and p I as the ground truth. The vector of precise controls for identifying bound boxes (ti) and the robust loss function for calculating the regression error (r) are used to describe the ground truth in situ regression loss in equation (3). To determine the outcomes of Km2 dimensions, each ROI employs a mask branch, an encoded K binary mask, and a resolution of mm. The DenseNet method is used in this work as the baseline model for CNN. By physically linking each layer, the DenseNet [22] DL design ensures efficient data flow across layers. DenseNets are used to represent the layers of the network, and each layer is connected to the layer above it. Take a look at an input image x0 that the convolution networks demonstrated processed. The network consists of N layers, and each layer carries out nonlinear transformations Fn (.).

Assume that each preceding convolution layer's feature map makes up layer n. Layers 0 through n 1 of the input feature map are concatenated and represented as x0,..., xn1. As a result, over an N-layer network, this method has N(N + 1)/2 connections. x is the current nth layer, [x0,..., xn1] is a concatenation of feature maps acquired from 0 to

n 1 layers, and Fn(.) is the composite functions of BN and ReLU. These expressions represent the output of the nth layers. The DenseNet framework is shown in Fig. 5.
The baseline model for the CNN in this study is the DenseNet technique. DenseNet [22] is a DL architecture that achieves efficient data flow across layers by directly connecting each layer. The network's layers are shown as Dense Nets and each layer is linked to the layer above it. Consider an input picture x0 that was processed by the convolution networks shown. There are N layers in the network, and each layer performs nonlinear transformations Fn (.).

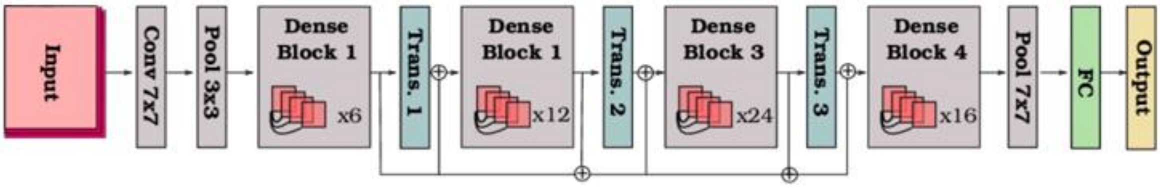

Figure 6: Structure of DenseNet[13]

All matched convolutional layers of the BN-ReLU-Conv sequences. Convolution is applied to the image, and then ReLU is used to produce the output feature map. This function displays nonlinearity in CNN models. The mathematical definition of the ReLU function is f (x0) = max (0, x0) (6)
Average pooling calculates the average value across all areas by dividing the input. The vector produced by averaging all of the feature maps with GAP is then applied to the SoftMax layer.
The DenseNet-169 model, which is based on the basic DenseNet architecture and has L (L + 1)/2 direct connections, is employed in this case.
The DenseNet model underwent a hyperparameter tuning using the dragonfly method to enhance the object identification outcomes (DFA). The DFA is founded by Mirjalili in 2016 [23]. Dragonflies naturally exhibit both static and dynamic behavior, emulating a metaheuristic method. Two essential phases of optimization are exploration and exploitation. The dynamic/static searching for food or evading the adversary, as well as these two stages, are based on dragonflies. SI is observed in the feeding and migratory studies of dragonflies. While feeding has been demonstrated to be a stationary swarm in optimal models, migration is represented as a dynamic swarm.
The swarms serve as the foundation for three distinct performances: separation, alignment, and cohesiveness. To avoid a static collision with a neighbor at this point, the separation model advises that people should separate (Eq. (7)).
The pace at which the agents interacted with people in their immediate vicinity is referred to as alignment (Eq. (8)). The cohesiveness model (Eq. (9)) also demonstrates the inclination of individuals near the herd's center. Going near food and avoiding foes are two extra actions that are added to these three crucial DA acts. The fact that survival is the main driving force for all swarms justifies the inclusion of these performances. So, if everyone travels in the direction of a meal
In the aforementioned scenarios, X stands for the immediate location of people, whereas Xj stands for the immediate location of people starting with the jth person. N represents the number of close neighbors, whereas Vj displays the speed of the jth neighbor [24]. The locations of food and hostile sources are denoted by the symbols X+ and X, respectively. After calculating the step vector (Eq. 13), the place vector is enhanced.

$$\nabla Xt+1 = (sSi + aAi + cCi + f\ Fj + eEi) + w\nabla Xt \quad (12)$$
$$Xt+1 = Xt + \nabla Xt+1 \quad (13)$$

The values of s, a, and c in Eq. (12), respectively, stand in for the separation, alignment, and cohesion coefficients, while the values of f, e, w, and t stand in for the food factor, enemy factor, inertia coefficients, and iteration number. The performance of both exploratory and exploitative optimization is enabled by the coefficients and mentioned variables. In this active swarm, the dragonflies tend to coordinate their flight. Despite the inadequate alignment during the static motion, the fittings for attacking the opponent were highly robust. As a result, the alignment coefficient was the biggest and the cohesion coefficient was the smallest during the exploration process; however, during the exploitation model, the alignment coefficient was minimum and the cohesion coefficient was the largest.

B. DRL Based Waste Classification Model

. Following their identification in the image, the rubbish objects are next classified using the DQLN method. State, action, and reward are the three components of the DRL. To learn how to map the state space to the action space, the DRL agent is interested. Then a reward is granted to the DRL agent. The DRL agent's goal is to maximize overall rewards. A function that accepts an instance and returns the likelihood of each label is the classifier approach used by the DQLN model.

$$\pi(a|s) = P(at = a|st = s) \quad (14)$$

classification agent aims to appropriately identify the instances in the trained data more feasible. Since the classification agents obtain a positive reward if it correctly identifies instances, it attains their aim by maximizing the cumulative reward gt:

$$gt = \_\infty$$
$$k=0$$
$$\gamma\ krt + k \quad (15)$$

In reinforcement learning, there is a function that computes the quality of state-action combination named as Q function:

$$Q\pi\ (s, a) = E\pi\ [gt|st = s, at = a] \quad (16)$$

Based on the Bellman formula, the Q function is written as:

Qπ (s, a) = Eπ [rt + γQπ (st+1, at+1)|st = s, at = a] (17)

The classification agents are maximizing the cumulative reward by resolving the optimum Q∗ function [25], and the greedy approach in an optimum Q∗ function is the better classifier approach π∗ for DQLN. Π ∗ (a|s) =

_ 1, if a = argmaxaQ∗(s, a) 0, else (18)
Q∗
(s, a) = EA[rt + γ max Q∗
(st+1, at+1)|st = s, at = a] (19)

In the low-dimension finite state space, the Q function tabs. Although Q functions in the high dimensional continuous state space could not be handled, a deep Q-learning strategy is given that fits the Q function with DNN. When employing the deep Q-learning strategy, the communication data (s, a, r, and s_) acquired from (19) are retained in the experience replay memory M. The Deep Q network's gradient descent step is guided by the loss function as the agent randomly instances a mini-batch of transitions B from M.:

L(θk) = _ (s,a,r,s_)∈B
(y − Q(s, a; θk))2 (20)

where y implies the target evaluation of the Q function, the written of y is:

y = r + (1 − t)γ max
a_ Q(s_, a_; θk−1)

where s_ implies the next state of s, a_ represents the activities carried out by the agent in state s_, t = 1 if terminal = True; then t = 0. The derivative of the loss function (20) in terms of θ is:

∇*L(θk)∇θk*= −2 _ *(s, a, r, s_)∈B(y − Q(s, a; θk))∇Q(s, a; θk)∇θk* (22)

At this point, the optimum Q∗ function can be obtained by minimizing the loss function (20), and the greedy policy in the optimum Q∗ function gets the maximal cumulative reward. Therefore, the optimum classifier approach π∗ : S → A for DQLN has been obtained.

C. Performance Validation

The Python 3.6.5 program is used to evaluate the proposed method on a benchmark Garbage [26] classification dataset from the Kaggle repository. The dataset contains six different waste classifications. There are 410 images under Metal, 504 images under Paper, 403 images under Cardboard, 137 images under Thrash, 501 images under Glass, and 482 images under Plastic. Fig. 6 displays a few of the sample test images from the Kaggle datasets.

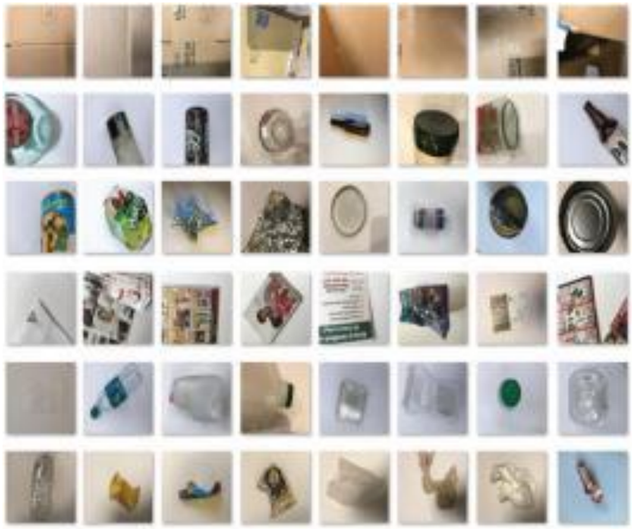

Figure 7: Sample images[13]

The CNN model's sample object detection results are shown in Fig. 7. The statistics showed that the CNN model successfully identified and concealed the items.

The confusion matrices for this approach with various amounts of training and testing data are shown in Fig. 8. The system approach has recognized 391 photos into cardboard, 489 images into glass, 405 images into metal, 496 images into paper, 474 images into plastic, and 130 images into trash using the applicable training/testing (90:10) dataset. Additionally, the SYSTEM method has recognized 363 photos into cardboard, 470 images into glass, 392 images into metal, 477 images into paper, 446 images into plastic, and 110 images into the trash on the applicable training/testing (80:20) dataset. The approach has also recognized 354 photos into cardboard, 461 images into glass, 382 images into metal, 467 images into paper, 440 images into plastic, and 104 images into the trash on the applied training/testing (70:30) dataset. The System algorithm has also recognized 349 pictures into cardboard, 461 photos into glass, 375 images into metal, 461 images into paper, 438 images into plastic, and 92 images into the trash on the applicable training/testing (60:40) dataset.

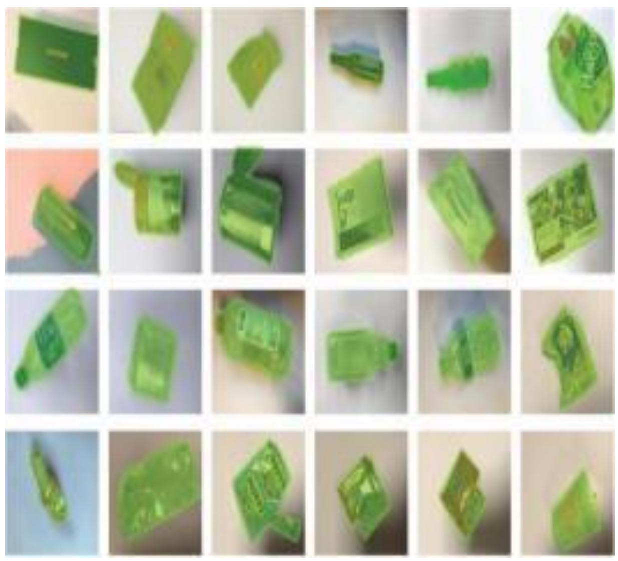

Figure 8: Sample object detection results[13]

With training and testing, the SYSTEM technique has an average recall of 0.975, precision of 0.978, F-score of 0.977, and accuracy of 0.993. (90:10). With training/testing (80:20), the SYSTEM technique has also advanced, attaining an average accuracy of 0.976, precision of 0.925, recall of 0.912, and F-score of 0.918.
Additionally, training and testing of the System technique yielded an average recall of 0.888, precision of 0.900, and F-score of 0.894, and an accuracy of 0.969. (70:30). With training and testing, the SYSTEM method additionally increased its average recall of 0.866, the precision of 0.884, the F-score of 0.873, and the accuracy of 0.964. (60:40).

Fig. 9 displays the accuracy evaluations of the System approach on the used dataset. The figure demonstrated that the training and validation dataset had the highest accuracy when using the SYSTEM approach. The results demonstrated that the training accuracy was inferior to the validation accuracy.
The loss analysis of the system technique on the used dataset is shown in Fig. 9. The graphic demonstrated how using this approach produced the lowest possible loss on the training and validation datasets. The outcomes exceeded the hypothesis that the validation loss would initially be smaller than the training loss.

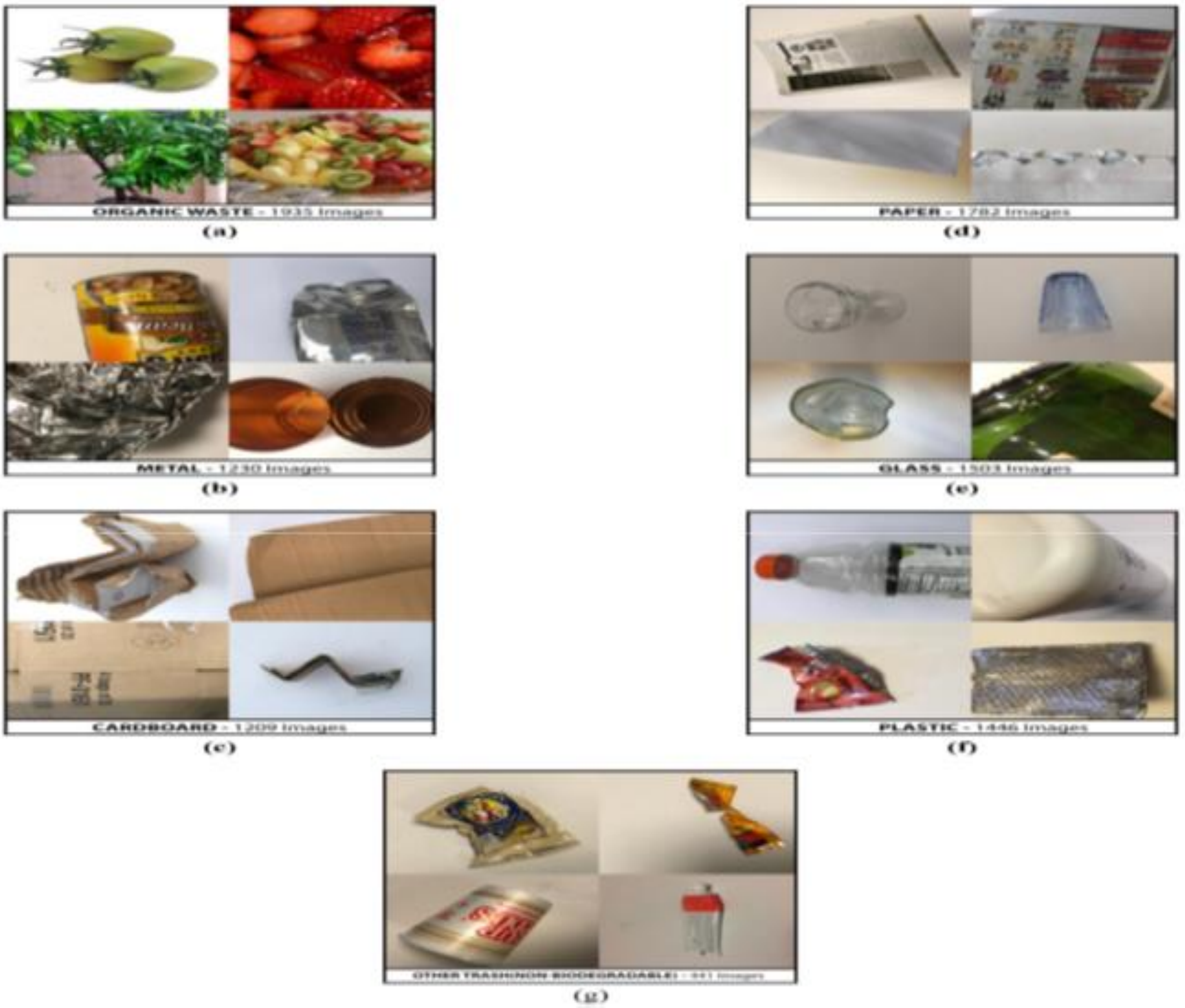

Figure 9: Classification of the Dataset

V. SIMULATION AND RESULTS

The DenseNet121 and CNN depicted that reached superior accuracy of 92%, and 82% respectively. Finally, the SYSTEM algorithm has resulted in a maximum performance with a maximal accuracy of 92 %
Epoch =2
No of Classes: 2
Classes : ['B', 'N']
Image Shape : (32, 256, 256, 3)
Model: "functional_1"
Total params: 18,510,146
Trainable params: 188,162
Non-trainable params: 18,321,984
Epoch 2/2
689/689 [==============================] - ETA: 0s - loss: 2.9639 - accuracy: 0.9373
Epoch 00002: val_loss did not improve from 4.05685
689/689 [==============================] - 11954s 17s/step - loss: 2.9639 - accuracy: 0.9373 - val_loss: 6.3158 - val_accuracy: 0.9208

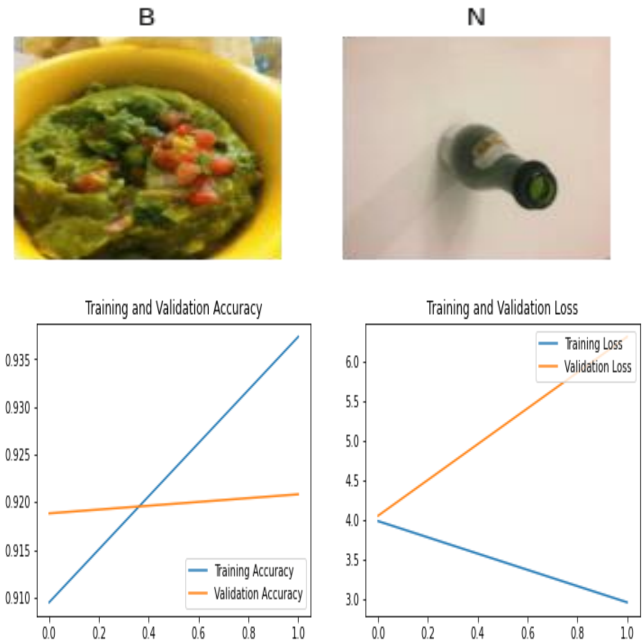

Figure 10: Training and validation

The figure 10 shows training and validation graph of images of the dataset.
Epoch =5
Epoch 1/5
[==============================] - ETA: 0s - loss: 1.6966 - accuracy: 0.9629
Epoch 00005: val_loss did not improve from 3.71658
689/689 [==============================] - 6191s 9s/step - loss: 1.6966 - accuracy: 0.9629 - val_loss: 7.5181 - val_accuracy: 0.9175

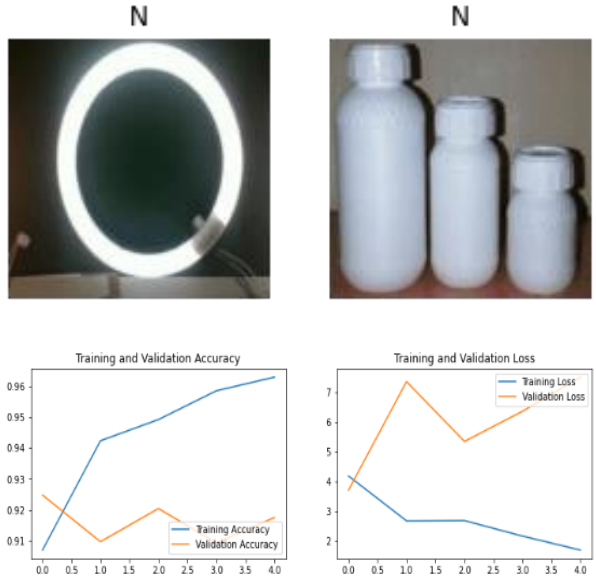

Figure 11: Training and Validation

The Figure 11 shows the training and Validation of the dataset.
173/173 [==============================] - 1230s 7s/step - loss: 7.5181 - accuracy: 0.9175
Accuracy on the Test Set = 91.75 %

Model Saved!
True : [0 0 0 ... 1 1 1]
Epoch =10
Epoch 00010: val_loss did not improve from 4.26417
689/689 [==============================] - 6272s 9s/step - loss: 1.0752 - accuracy: 0.9788 - val_loss: 7.9558 - val_accuracy: 0.9235

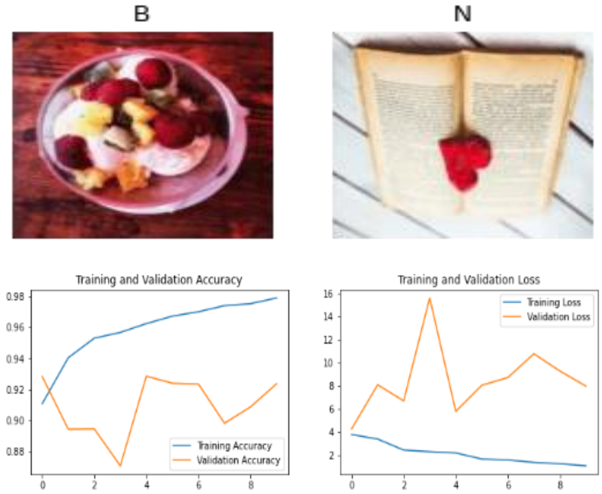

Figure 12: Training and Validation

The figure 11 and 12 shows the Training and Validation graph of the dataset.
173/173 [==============================] - 1242s 7s/step - loss: 7.9558 - accuracy: 0.9235
Accuracy on the Test Set = 92.35 %

Confusion Matrix

Figure 13 shows the Confusion Matrix. To summarize the performance and prediction result of our classification algorithm, we used a confusion matrix technique.

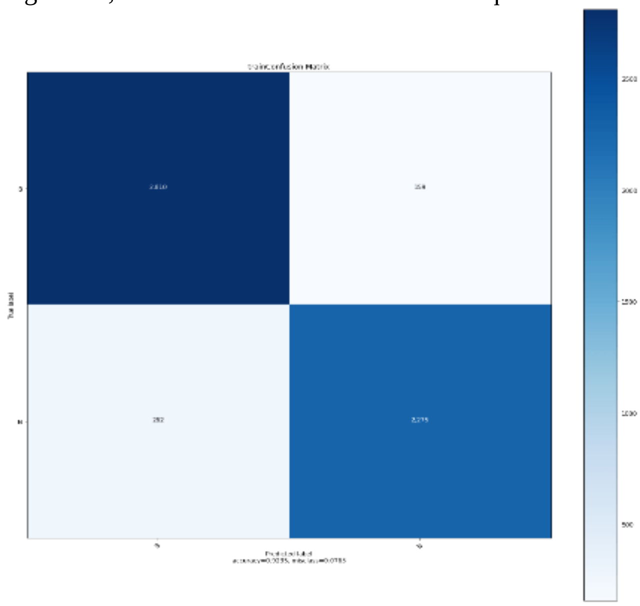

Figure 13: Confusion Matrix

VI. CONCLUSION

In this paper, a novel system approach for the detection and categorization of recycling trash objects for smart cities is described. For the purpose of detecting and masking waste items in the picture, the System approach generates a CNN with the Dense Net model. Using DFA, a hyperparameter adjustment was performed to improve the DenseNet model's results for object detection. To categorize the observed items into separate class labels, the DRL-based DQLN approach is also used. The System approach can identify items of different sizes and orientations. An exhaustive experimental study is conducted to determine the effectiveness in terms of several metrics to guarantee the enhanced waste classification results of the System approach. The experimental findings demonstrated that the system algorithm outperformed the prevailing methods. In the future, this method might be implemented as a smartphone app to assist in the real-time categorization of trash objects.

REFERENCES

[1] K. D. Sharma and S. Jain, “Overview of municipal solid waste generation, composition, and management in India,” Journal of Environmental Engineering, vol. 145, no. 3, pp. 04018143, 2019.

[2] Y. Wang and X. Zhang, “Autonomous garbage detection for intelligent urban management,” in Proc. Of the MATEC Web Conf., vol. 232, China, pp. 01056, 2018.

[3] R. S. S. Devi, V. R. Vijaykumar and M.Muthumeena, “ Waste segregation using deep learning algorithm,” International Journal of Innovative Technology and Exploring Engineering, vol. 8, pp. 401‐403, 2018.

[4] N. J. G. J. Bandara and J. P. A. Hettiaratchi, “Environmental impacts with waste disposal practices in a suburban municipality in Sri Lanka,” International Journal of Environment and waste management, vol. 6, no. 1/2, pp. 107, 2010.

[5] A. T.García,O. R. Aragón,O. L. Gandara, F. S.García and L. E.G. Jiménez, “Intelligent waste separator,” Computación y Sistemas, vol. 19, no. 3, pp. 487–500, 2015.

[6] J. Zheng, M. Xu, M. Cai, Z.Wang and. Yang, “Modeling group behavior to study innovation diffusion based on cognition and network: An analysis for garbage classification System in shanghai, China,” International Journal of Environmental Research and Public Health, vol. 16, no. 18, pp. 3349, 2019.

[7] Y. Chu, C. Huang, X. Xie, B. Tan, S. Kamal, et al., “Multilayer hybrid deep-learning method for waste classification and recycling,” Computational Intelligence and Neuroscience, vol. 2018, pp. 1–9, 2018.

[8] D. Ziouzios, D. Tsiktsiris, N. Baras and.Dasygenis, “A distributed architecture for smart recycling using machine learning,” Future Internet, vol. 12, no. 9, pp. 141, 2020.

[9] O. Adedeji and Z. Wang, “Intelligent waste classification System using deep learning convolutional neural network,” Procedia Manufacturing, vol. 35, pp. 607–612, 2019.

[10] Y. Chu, C. Huang, X. Xie, B. Tan, S. Kamal, et al., “Multilayer hybrid deep-learning method for waste classification and recycling,” Computational Intelligence and Neuroscience, vol. 2018, pp. 1–9, 2018.

[11] B. Gan and C. Zhang, “Research on the algorithm of urban waste classification and recycling based on deep learning technology,” in 2020 Int. Conf. on Computer Vision, Image and Deep Learning (CVIDL), Chongqing, China, pp. 232–236, 2020.

[12] Y. Narayan, “Deep Waste: Applying deep learning to waste classification for a sustainable planet,” arXiv preprint arXiv:2101.05960, 2021

[13] Al Duhayyim, Mesfer & Eisa, Taiseer & Al-Wesabi, Fahd & Abdelmaboud, Abdelzahir & Hamza, Manar & Zamani, Abu & Rizwanullah, Mohammed & Marzouk, Radwa. (2022). Deep Reinforcement Learning Enabled Smart City Recycling Waste Object Classification. Computers, Materials & Continua. 71. 5699-5715. 10.32604/cmc.2022.024431.