ISSN: 2037-4445

A Study on Solid Waste Classification Through Deep Learning Models - A Comparative Approach

Loganayagi S[1], Dr. D. Usha [2]

[1]PhD Research Scholar(part time), Computer Science , Mother Teresa Women's University, Kodaikanal, Tamil Nadu, India, [2] Research Guide, Assistant Professor, Department of Computer Science, Mother Teresa Women's University, Kodaikanal, Tamil Nadu, India

Abstract

The study aims to classify solid waste materials into three types using machine learning models and compares the performances of the machine learning models. Three standard models namely ResNet-50, VGGNet-16, and Inception-V3 model have been used in this research along with a newly developed custom-CNN (convolutional neural network) model for waste classification.

Methodology: Datasets of solid waste materials were acquired from 'Kaggle' as images for waste management. The images are processed and passed through all four models and the output is classified through image detection and classifier algorithms (XGBoost, Random Forest, and FC-DNN: Fully-Connected Deep-Neural-Network algorithm). The classified images as outputs are then stored under labels N: Non-recyclable, R: Recyclable, and O: Organic. Python' is used as the programming language software.

Findings and Performance evaluation: The obtained outputs are compared with 'accuracy rate' as a metric evaluation technique. Findings showed ResNet-50 model with the 'Random-Forest' classifier algorithm is more efficient and reliable than other models with a 72.44% accuracy rate.

Conclusion: The study concludes that ResNet when combined with the Random-Forest model is to perform efficiently in classifying solid waste materials with greater accuracy.

Keywords: Solid waste material, waste management, Classification of material, automation, Deep learning, VGGNet, InceptionNet, XGBoost.

1. Introduction

The Neural Networks (NNs) based models in identifying objects with classifier algorithms have been in trend currently, especially for face-detection and object-detection models. The core function of an object detection model, especially in real-time object detection, images are identified from video or image, and through a classifier algorithm and images are segregated and labeled. To identify and categorize the inputs/ images obtained from real-time object detection, a researcher or model developer requires necessary components: a hypothesizer verifier, feature detector, and a hypothesizer [1]. To determines and identify the necessary object within an image (localizing objects) and to classify the objects into different categories (object classification) is the real function of an identification model, whether it is a convolutional NN (CNN) or deep NN (DNN) model [2].

Object detection majorly could be utilized for autonomous driving [3], face recognition [4], image classification [5,6] and behavior analysis of humans [7]. Identifying objects in the image through an object identification algorithm focuses upon image resolution with classifier as the primary technique in NN models namely ResNet-50, InceptionNet, VGGNet-16, AlexNet, and other models along with face detection and the feature extraction algorithms [8]. The object detection algorithms such as KNN (k-nearest neighbor), MLP (multilayer perceptron), NB (Naïve Bayes), LR (linear regression), RF (random forest), SVM (support vector machine), decision tree, and others are most commonly utilized by

ISSN: 2037-4445

researchers for higher accuracy in ANN (artificial NN), CNN, DNN and other machine learning (ML) based models.

The current research adapts the ResNet-50 model, the VGG-16 model, and the InceptionNet model. The researcher aims at developing the custom CNN model. All four models will be combined with classifier algorithms such as the XGBoost algorithm, the Random Forest algorithm, and the Fully connected DNN algorithm.

2. Literature review

2.1 Object detection

Authors [9] focused exclusively on object recognition and identification/ detection model, in images. Authors argued, in image recognition and detection, the object detecting algorithm-based models purely rely upon pattern recognition, learning, and matching procedures through feature-based or appearance-based methods. Image intensity, brightness, number (single or multiple) of objects, color, size, pixels, contrast, and other features are extracted from input images and are later resized (pre-processed) according to the researcher's model adaptation.

Authors [10] studied multiple object identification in deep learning models that uses two-stage image recognition algorithms. The authors [11] found differences between the customary and contemporary deep learning models. According to both studies, the obtained findings by Deng et al., traditional detection models include a few stages such as feature extraction, feature classification, window sliding, pre-processing, feature selection, and post-processing. They classified one-stage and two-stage algorithms in object detection. Author [12] developed and analyzed the SPP-Net (spatial pyramid pooling) model, [13] focused on R-CNN (regions with CNN), [14] focused on Faster-RCNN and [15] studied Fast-RCNN as one-stage algorithms. Contrarily, the author [16] focused on the object detection models with Yolo-v1, whereas author [17] on Yolo-v2 and [18] on Yolo-v3. The latter Yolo models - based studies concluded that the two-stage algorithm models are popularly used due to its accuracy and performance in object segmentation and detection; however they lack in accuracy when it comes to identify the smaller objects and also with accurate detection of objects in the larger groups. The SSD (single shot detector) was studied by author [19] and Yolo-v4 by [20] which are also a classification of two-stage algorithms. From the intensive reviews, it is observed that the first stage (single-stage) relies on region proposal whereas the second stage (two-stage) relies on regression. Thus it's concluded that, rather than adopting single-stage algorithms, two-stage is efficient. However, it lacks in multi-category detection of objects, and thus when developing multi-objects detection models the single-stage is efficient and rapid for dependence datasets.

2.2 Applications and challenges in object detection models

Authors [21] examined digital images-based implementation of detection models. They affirmed that feature extraction (color, optical flow, edges, and gradient histograms) plays a vital and eminent role in object detection. Though recognition models have gained popularity, the application of models has a certain level of challenges like object rotation, lighting/ clarity, positioning, scaling method, mirroring, and occlusion condition hinders accuracy and precision of models in recognizing and identifying objects.

A study by [22] focused on and examined the applications of deep learning-based object recognition in CNN models. They structured the features and interface according to deep learning models. Python was found to be a more efficient interface and used in major models, where Caffe, Microsoft-Cognitive Toolkit (CNTK), Theano, Keras, Tensorflow, MXnet, Chainer, Neon, Apache Singa, and Deeplearning4j are commonly used frameworks. However, the authors found that the performance of each deep learning model varies in object detection and thus affects the outcome. Hence it was evidently concluded that utilizing the appropriate number of datasets could enhance the accuracy.

Authors [23] also had specified that feature selection is a crucial phase in object detection and application of the model. However, they also reviewed and explored other techniques and processes involved in the detection model. They found temporal differentiating, optical flow, and subtraction of background (selected frame subtracted from the background frame, pixel functioning in Gaussian probability, multimodal distribution, and test frame and median frame subtraction) as significant phases that are perceived as challenges. Contrarily the authors insisted that, prior developing a model, the choice of tracking method is also significant. Tracking techniques vary for a model and the accuracy rate is affected when the computational cost is lesser. Finally, the study concluded that the higher the accuracy rate, the higher the computational complexity becomes. Statistical techniques, temporal differencing, background subtraction, and optical flow are the major contributors in object recognition models, where the developer should consider the challenges of darker shadows, illumination changes, and object occlusion.

3. Proposed system

a) Approach

The research mainly focuses on segregating and categorizing the wastes according to labels: recyclable, organic, and non-recyclable. The study adopts the CNN approach where labels are customized as per the requirement, based on relevant existing waste management models. Initially, the Custom CNN model developed is tested to evaluate accuracy against existing image augmentation and classification models. The produced outcome is compared with the ResNet model, VGGNet model, and InceptionNet model for identifying better image identification through better performance. Random forest as the classifier algorithm is used in the custom model and results are obtained. Later, 'Fully connected DNN (deep-neural network) and XGBoost' as the classifier algorithms are also used, in the respective customized models. Thus all four will be coded with the three classifiers algorithms to compare the better model.

Based on the applied classifier algorithms, the waste management model segregates the items under (a) ***non-recyclable wastes*** label, like plastic wraps, kitchenware and ceramics, food-tainted products (lunch boxes, plates, tumblers, paper napkins), plastic wraps and bubble wraps, and more; (b) ***organic wastes*** label, where foods (fruits, vegetables, milk, meats, pulses, grains), and natural pesticides, fertilizers and herbicides towards produce and farming; and finally (c) ***recyclable wastes*** namely: egg cartons, newspapers, carpets, aluminum foils, and cans, books, paper boxes and cardboards, glass containers, laundry detergent and cleanser bottles (made from degradable components).

b) Flow-diagram

The proposed approach of research is explained through a flow diagram (refer to figure 1).

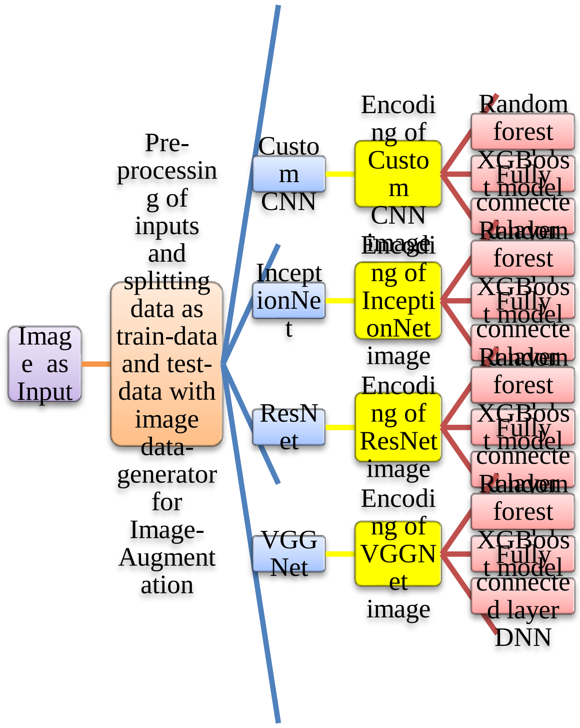

Figure 1: Flowchart of the proposed research

The research proposed encompasses four models towards identifying objects (wastes) and classifying the objects (wastes) according to the algorithm. The models in the study initially use images as input and they are preprocessed and stored in a folder. Once the preprocessed images are stored, they are classified as train-data and test-data (for augmentation of images). In the third stage the images are passed through network layers of four models, the custom CNN model, the ResNet model, the VGGNet model, and the InceptionNet model. Later in the fourth stage, the images are encoded and passed through the classification algorithm (random forest, xgboost, and fully connected DNN) to separate the images into folders (recyclable, organic, and non-recyclable). Finally, in the fifth stage, the research compares the performance of each model and evaluates the ResNet model, InceptionNet model, VGGNet model against the Custom CNN model with "accuracy" as performance metric where the generated metrics of labels (organic ('o'), non-recyclable ('N') and recyclable ('R')) are evaluated.

c) **Architecture*:***

i.Custom CNN model:

The researcher developed a custom CNN model with categorical cross-entropy for loss estimation and Adam-optimizer (refer to figure 2).

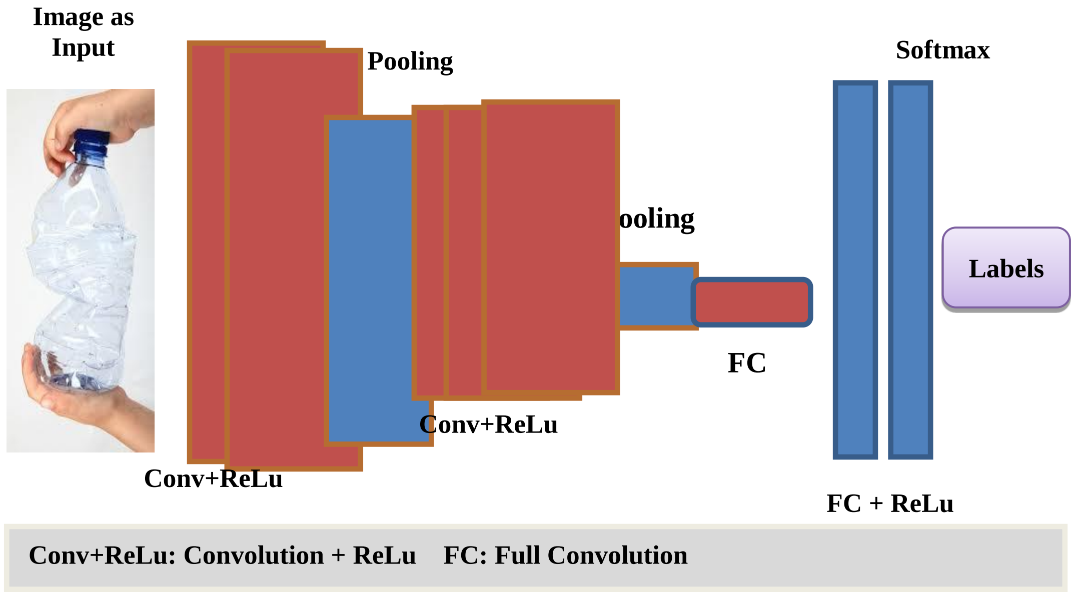

Figure 2: Custom CNN Architecture developed

In the developed custom CNN, the image (wastes) is used as input. Then the image is pre-processed and passed through the initial convolutional layer with 896 as the parameter. Later the image is passed through 2-convolutional 2d layers with parameter as 9248. Next, the image is passed through the pooling layer and dropout layer before passing through the next 3-convolutional layers with parameters 18496, 36928, and 36928 respectively. Again, the image is passed through the dropout-1 layer before the flattening layer, dense layer, and the process is repeated twice with dropout-2 layer, dense-1 layer and dropout-3 layer, dense-2 layer. Finally, the image is passed through the Full Convolution layer and the output as the processed and classified image is acquired and labeled.

The CNN model is developed with three methods: Random Forest (RF), XGBoost (XGB), and Fully Connected DNN (FCDNN). The metrics will be evaluated for all three methods of the CNN model developed for "accuracy".

ii.Resnet model:

The first process is to load the image as input from the folder with the ResNet model. The ResNet-50 comprises 'zero padding 2D' as the first layer followed by a single convolution layer (refer to figure 3).

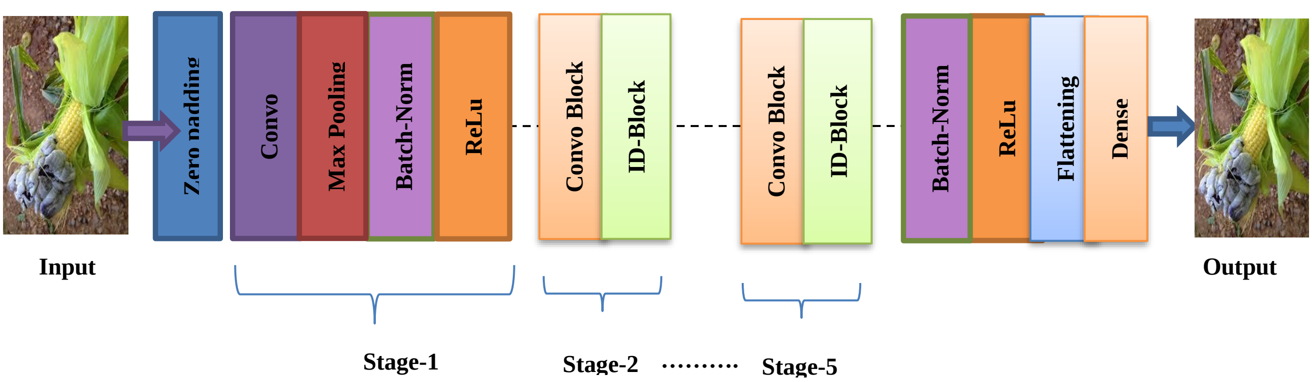

Figure 3: ResNet50 - architecture used

The ResNet-50 model encompasses input (image) as the initial stage which is passed through the model. 'Zero padding' is the first layer of the model, followed by convolution layer-1, max-pooling-1, and ID block-1. The ID Block-1 in the first stage consists of block1-1: Batch-Normalization, ReLu, Conv2D, and block1-2: Batch-Normalization, ReLu, Zero-padding, Conv2D, Batch-Normalization, ReLu. Also, block 1-0, and block 1-3 each include a convolution layer, respectively. In stage 2, the same process without max-pooling layer is developed as ID block-2 with convolution layer 2. Max pooling is included with the same set of ID blocks and convolution layer-2 in ID block-3. In stage three, convolution layer 3 is set with ID block-1, block-2, block-3, and block-4 (with max-pooling). Next, in stage four, convolution layer 4 is set with ID block-1 to block-23 (with max-pooling). In stage five, the convolution layer 5 includes 3 ID blocks.

Once the image is passed through stage 5 (Conv-5), it is then passed through the Batch-Normalization layer, ReLu, Flattening layer, and finally through the Dense layer. Output is acquired and stored under the particular labeled folder.

iii.VGG model:

The VGG-16 model used in the research has 5convolution layers and max pooling-2D layers respectively (refer to figure 4). The model also comprises one flattening layer and a dense layer. The image is initially passed through 2convolution layers and one max-pool layer. Later, the pre-processed image is passed through one more block of 2sets of convolution layers and one max-pool layer. In the next layer, the image is passed onto 4sets of convolution layers with one max pool layer and the same process is repeated twice. Finally, the image is flattened and passed through the dense layer. The output is acquired and segregated with a classification algorithm under a labeled folder.

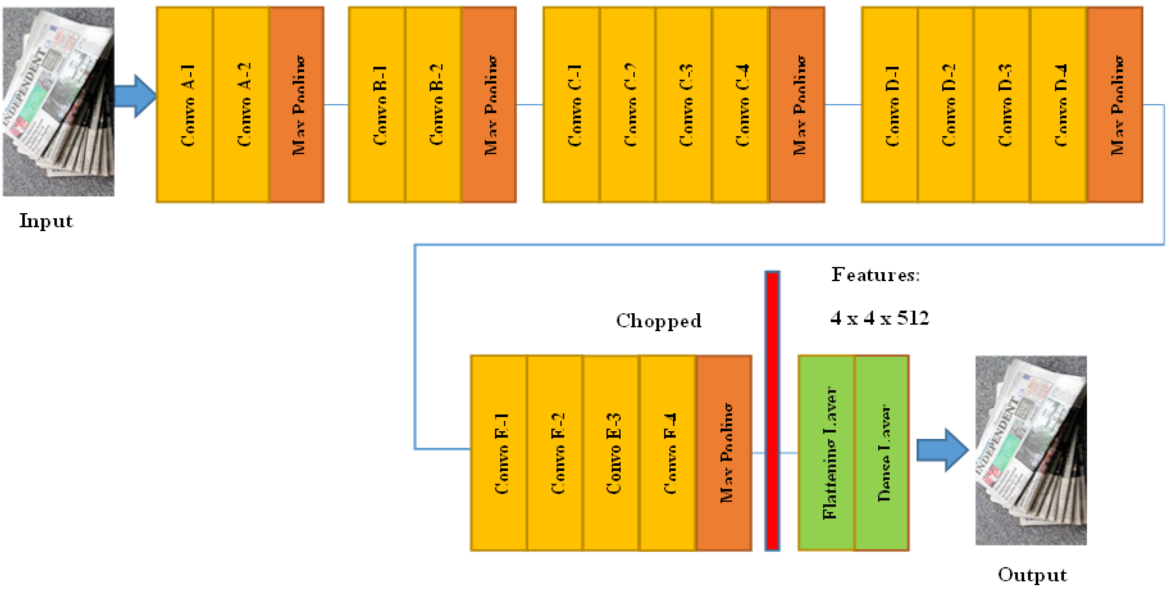

Figure 4: VGG16 - architecture utilized

iv.Inception model:

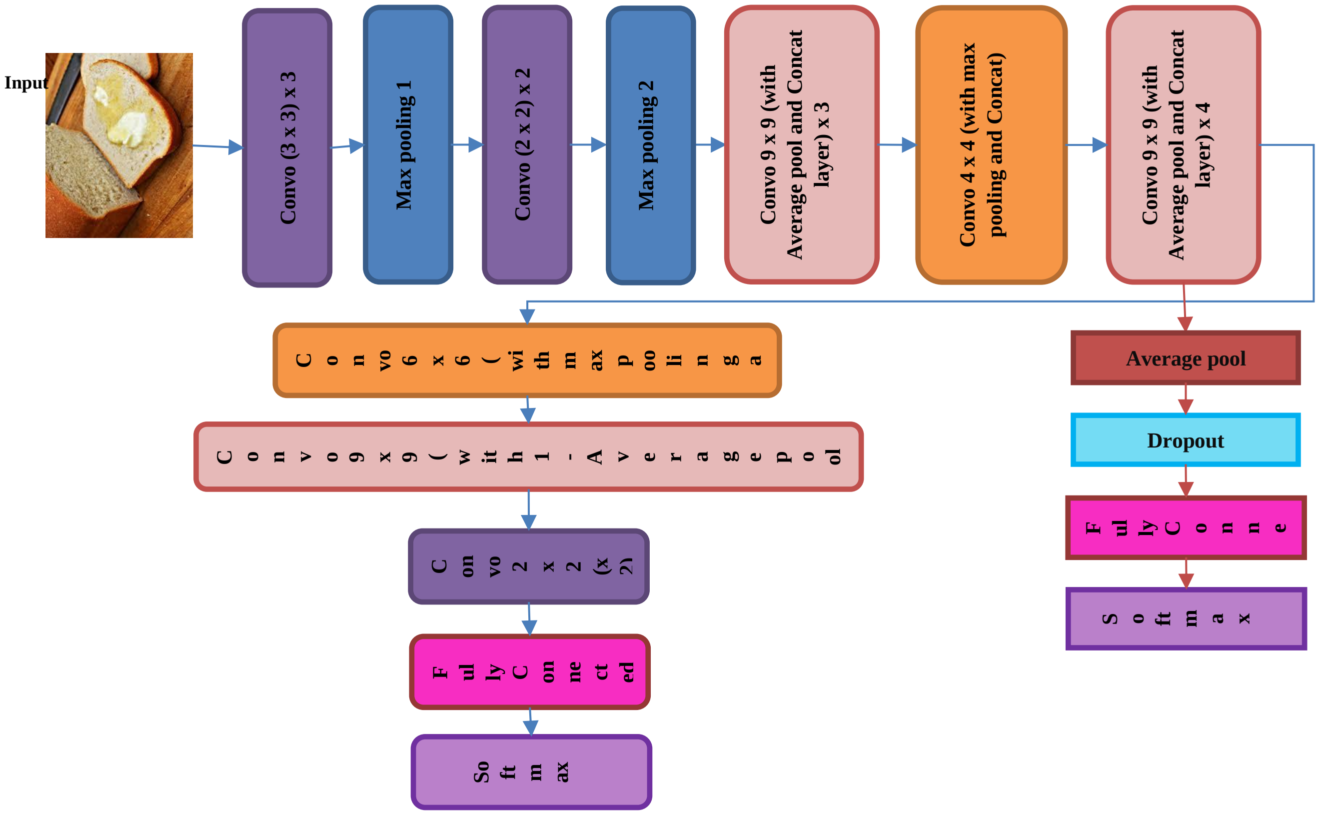

Figure 5: InceptionNet V3 - architecture utilized

The initial process in the inception model passes the image (input) to the convolution layer with Batch-Normalization and Activation layers (3x3), followed by the max-pooling layer (refer to figure 4). Later, 2sets of 2x2convolution layers with one max-pooling layer process the image and pass it to the inception layer. From stage-1 to stage-3, the inception layers include 7innermost convolution layers + 1outermost convolution layer and 1outermost average pool layer with 1convolution layer. In stage-4 the inception layer includes 3innermost convolution layers and 1outermost convolution layer with 1-max pooling followed by 4sets of inception layers (similar to stage-1), in stages 5 to 8. In stage-9, 4innermost convolution layers and 2outermost convolution layers with 1-max pooling layer could be seen. In stages 10 and 11, the outermost layers include 1convoluion layer and 1average pool layer with 1convolution layer, each respectively; whereas the innermost layer includes: 2sets with a) 3convolution layers and 1concat layer and b) 4convolution layers with 1concat layer. Lastly, 1average pool layer, followed by dropout layer, fully connected layer, and Softmax layer is seen.

Each inception layer is connected by the 'concat layer' at each end, respectively, totaling 10concat layers at the end of each stage, with exception of end-of-stage 10 (which is connected by convolution layer with stage-11). The 8^{th} concat layer is connected with another division where 1average pool, 2convolution layers, 1fully connected and 1softmax layers are seen. The final stage is the "output" where the result is obtained.

d) **Algorithm used:**

The machine learning (ML) algorithms utilized in the developed research towards classifying and segregating images as three categories are XGBoost, Random Forest, and Fully connected DNN.

i. **Algorithm for XGBoost (XGB) classifier:**
XGB is based on the decision tree process in the ML algorithm but identified as the gradient boosting algorithm. The pseudocode for the XGB algorithm is:
Step 1: Import images as input from 'kaggle';
Step 2: Load XGB-classifier from XGBoost in python;
Step 3: Once the model is loaded, preprocess the trained images and re-shape accordingly from sklearn in python;
Step 4: Import the metric "accuracy" from sklearn in python for performance score comparison;
Step 5: load the CSV file with NumPy function and then segregate dataset columns as inputs A and B through NumPy-array format;
Step 6: Lastly, the images are classified as test-datasets (B) and train-datasets (A) which are later classified under labels non-recyclable, recyclable, and organic.

ii. **Algorithm for Random Forest (RF):**
RF is based on the bagging-based method with differences in the feature subsets. The pseudo-code for the RF algorithm is:
Step 1: Initially, load the input images in a single folder;
Step 2: Next, through Random Forest consider 'n' random records from 'k' records;
Step 3: Output (classified images) from Python is generated as individual decision-trees for every single-sample processed;
Step 4: Lastly, the output is classified based on higher-voting/ majority-voting of classes, and the image is stored in the respective labeled folder (non-recyclable, recyclable, and organic) and the final class is obtained.

iii. **Algorithm for Fully-Connected DNN (FC-DNN):**
FC-DNN belongs to a neural network where it's classified as the ML-class-based algorithm. The pseudocode is:
Step 1: Initially, the FC-DNN model is defined and loaded (for instance: input features);
Step 2: Next, parameters along with hyper-parameters are initialized in the second stage, where iterations, layers (L) in neural-networks (NN), hidden layers' size, and the rate-of-learning (α) are included;
Step 3: Later, Num-iterations for loop such as forward propagation (current-loss estimation), backward propagation (current-gradient estimation), cost function, and update parameters (through back-prop grads and parameters) are calculated;
Step 4: Finally, with trained-parameters prediction and classification are made and output (image) is obtained and stored in labeled folders.

4. Methodology

4.1 Dataset

The study uses secondary data (images from the internet) as the source. The datasets (images) for waste classification are acquired from 'Kaggle' created by [24]. Total images available in the dataset were 25,474 of which for train-dataset was 22,564 and test-dataset was 2910. The images are pre-processed that includes the formats like, .jpeg, .png, .jpg and other extensions.

4.2 Training datasets:

S. No	Non-Recyclable	Recyclable	Organic
1.			
2.			

Table 1: Training the data

The total of trained datasets was 22,564 where 'N' images (Non-recyclable) was identified as 2847; 'R' images (Recyclable) was identified as 7,152 and 'O' images (Organic) was identified as 12,565.

Table 1 shows, each image is appropriately trained and classified under the custom CNN model developed.

4.3 Testing datasets:

S. No	Images	Predicted Waste-Class
1		**Non-Recyclable**
2		**Recyclable**
3		**Organic**

ISSN: 2037-4445

S. No	Images	Predicted Waste-Class
4		Non-Recyclable
5		Recyclable
6		Organic
7		Non-Recyclable
8	 WORLD NEWS New website addletters.com generates cool images for users. Get it while it's hot!	Recyclable
9		Organic

ISSN: 2037-4445

S. No	Images	Predicted Waste-Class
10		Organic

Table 2: Testing the custom CNN model for classification of wastes

Table 2 shows, each image is classified and identified by the model accurately. However, the findings of the model showed that from 2910 images for testing, 397 were identified as Non-recyclable, 1401 as Organic, and 1112 as Recyclable with 3failed attempts.

4.4 Software and Statistical methods adapted

The research utilizes Python as software for the classification of images in waste management using existing neural network models along with the custom CNN model. Activation-functions used here are both ReLu and Softmax. Similarly, the categorical-cross-entropy, Gini-index, and information-gain are also utilized. A 2x2confusion-matrix is used to estimate the accuracy as a metric. The statistical techniques with used software are explained below:

4.4.1 Softmax activation-function:

The Softmax/ Softargmax (a normalized exponential-function) is generally used in the last stage of NN models in the 'output layer'. It's used to "squash" the multiple outputs into a range between 0-1, which in turn represents the direct probability of an outcome.

$$\alpha\,(\vec{x})_k = \frac{n^{x_k}}{\sum_{m=1}^{L} n^{x_m}} \quad \text{.. (1)}$$

where: α represents the Softmax, $\vec{x}$ represents the input-vector, n^{x_k} represents the input vector's standard exponential-function, n^{x_m}represents the output vector's standard exponential-function and *L* represents the multi-class classifier's class numbers.

4.4.2 ReLU activation-function:

The ReLU (rectified linear-unit) is commonly used by researchers in NN, especially the CNN models. It's a piecewise linear function, that provides the 'input' as direct output when the value obtained is positive and if not, it returns '0' as output. The formula is: $y = maximum\{0, Z\}$ *.. (2)*

4.4.3 Gini Index (GI):

The GI is specifically used by investigators in identifying objects like 'face detection' in the last 22years with variance in colors and advanced image resolutions towards identifying, classifying, and segregating images based on input features. Since GI provides better performance in image classification, researchers till-date attempted to develop more advanced algorithms that could provide rapid outcomes and with higher accuracy in classification.

GI (decision-tree method) basically segregates and 'labels' each node into a separate class. Contrarily, nodes that have root-nodes and non-terminal internal nodes together are categorized and classified into five variations based upon each node's attribute and their test conditions. Later, datasets are split and classified with a child node's degree-of-impurity [25]. The formula is:

$$GI = 1 - \sum_{n=1}^{a}(M)^2 \qquad \text{.. (3)}$$

4.4.4 Categorical Cross-Entropy:

The cross-entropy belonging to categorical in neural-network classification, particularly in identifying objects has been substantiated as an effective statistical measure. It is found by [26] that categorical cross-entropy is efficient with marginal loss-error values and maximum possibility in predictable values. The formula utilized for calculating the loss-epoch values here is:

$$-\sum_{y=1}^{A} x_i\,, y \log b_i\,, y \qquad \text{.. (4)}$$

4.4.5 Information Gain (IG):

Information-Gain is a process to acquire information through variables, attributes, and features of inputs. Calculation of IG is estimated through impurity criterion, samples, child nodes, and total-number-of input. The formula for estimation of IG is:

$$IG(X_n, a) = E(X_n) - \frac{T_{left}}{T} E(X_{left}) - \frac{T_{right}}{T} C(X_{right}) \qquad \text{..........................(5)}$$

Where: ***a*** is the split-on of features; X_n is the datasets in parent node dataset; X_{right} is the datasets of right child-node; A_{left} is the datasets of left child-node; ***E*** is the GI/entropy of impurity-criterion; S_{left} is the total-samples in datasets of left child-node; S_{right} is the total-samples in datasets of right child-node and ***T*** is the total samples.

4.4.6 Confusion Matrix (IG) and Accuracy prediction:

A 2x2confusion matrix is used in this research as a metric evaluation to predict the accuracy of models in classifying images. The confusion matrix has 2negative (False-Negative and True-Negative) and 2positive values (True-Positive and False-Positive). Total sample datasets (i.e. ***n***) processed in research are then classified into "predicted values" and compared with "original values" obtained through image classification models used. The accuracy metric is calculated with standardized formula, which is:

Accuracy (A) = (TN+TP)/(FN+FP+TN+TP) .... (6)

4.4.7 Software utilized:

The software for analyzing the datasets is '*Python'* as the programming language. *API* in python is used as the object detection method. The platform used is *Anaconda*. The Library adapted is the *Tensorflow with Keras.* Jupyter-Notebook as a development environment is adapted with Machine Learning (ML) and Deep Learning (DL) based NNs. Python is well-known among researchers as modular, user-friendly, extensible, and compatible, especially with Toolkit/ Theano and Tensorflow [27].

Python is adapted for simplicity, efficiency in probability, availability and reliability, adaptability. Keras in Tensorflow is known for its real-time documentation of objects and classification of multiple classes [28].

5. Experiment results

The experimented outcomes are analyzed and accuracy of each model is calculated and a prediction of the better model was made. The outcomes are:

5.1 Custom CNN:

The accuracy calculation through Confusion-matrix evaluation for the ResNet, VGGNet, InceptionNet, and Custom CNN models are:

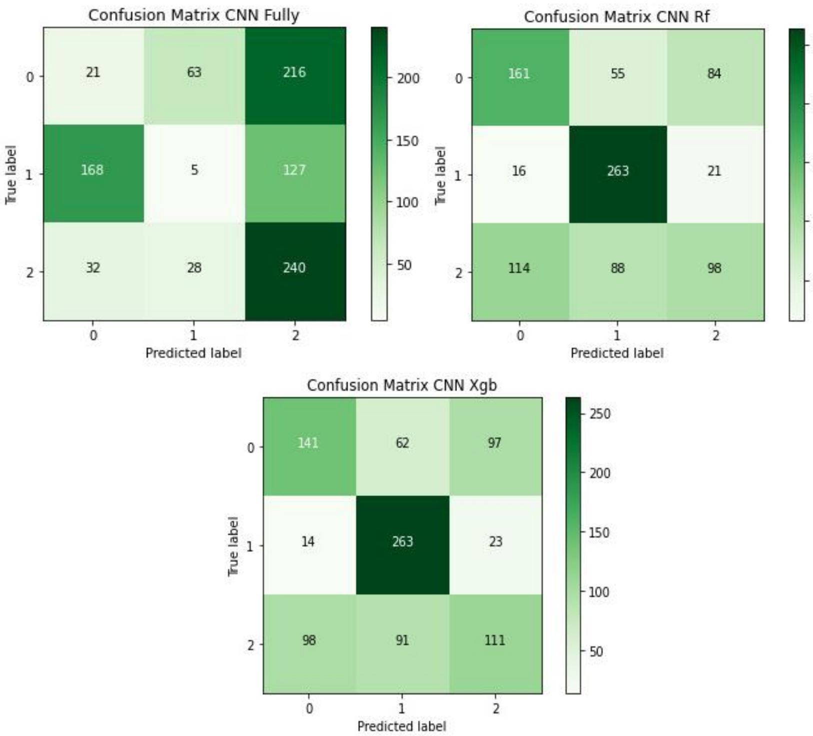

Figure 6: Custom CNN with FC-DNN, Custom CNN with RF, Custom with XGB

5.1 Findings:

The outcome/findings from data analysis show, the custom model developed procured 58% accuracy (in Random Forest model) as a higher accuracy rate and 29.56% accuracy (in FC-DNN model) as the lowest accuracy rate in custom model analysis. Contrarily, the ResNet model with Random Forest procures the highest accuracy of all models stating, in classifying and labeling wastes-management the ResNet with Random-Forest model is found efficient and reliable, followed by the VGGNet model and then Inception model.

6. Performance metrics

The research attempted to analyze ResNet, VGGNet, Inception against the custom CNN model developed. For evaluation of metrics, the research uses the “Accuracy” metric where the outcome of each model is compared. The accuracy scores are:

Model	Accuracy(Percentage)
Custom CNN + Fully Connected DNN Model	29.56%
Custom CNN + Xgb Model	57.22%
Custom CNN + Rf Model	58.00%
Resnet + Fully Connected DNN Model	34.55%
Resnet + Xgb Model	72.11%
Resnet + Rf Model	72.44%
Vgg + Fully Connected DNN Model	54.11%
Vgg + Xgb Model	69.00%
Vgg + Rf Model	67.11%
Inception + Fully Connected DNN Model	61.33%
Inception + Xgb Model	38.66%
Inception+ Rf Model	40.80%

Figure 18: Performance evaluation through accuracy metric

The custom CNN model developed procured lesser accuracy, however, it could be observed that the rate-of-accuracy in the Inception model (XGB and RF) is lesser than the custom CNN model (refer to figure 18). Henceforth, there is scope and area-of-development in the developed model in near future.

7. Conclusion and Suggestion

The study attempts and aims to analyze the NN models in identifying and classifying images towards segregating wastes-management under "N - Non-recyclable", "O – Organic" and "R – Recyclable" labels. The study focused on three existing models: ResNet-50, VGGNet-16, and InceptionNet-V3 models towards comparing the output with the custom CNN model. Datasets for research are acquired from 'Kaggle' created by [24] where the total sample as datasets is accounted as 25,474 images in which for training 22564 images and for testing 2910 images were used. The research used Python as ML software. The classifier algorithms (XGB, RF, and FC-DNN) in python for segregating images under the created labels (N, R, and O) in models were used with all four models. Accuracy as a metric was evaluated and compared with predicted against original values. Images are encoded with encoder-model in python and the outcomes are stored under classified-labeled-folders.

The findings showed, custom model lacks accuracy (FC-DNN=29.56%, XGB = 57.22% and RF = 58%) and ResNet with Random Forest acquired highest accuracy with 72.44% (also FC-DNN = 54.11% and XGB = 72.11%). This suggests that custom CNN could be modified and altered for higher accuracy which is possible since the Inception model showed a lower accuracy rate (in RF and XGB) than custom CNN.

8. Future enhancements

The study adopted the NN models and developed a custom CNN model to evaluate accuracy as a metric-rate in classifying and identifying wastes (images). However, the current model lacks in the accuracy rate, which in near future, will be focused on and developed with a 75% accuracy-rate based model. The aim is to surpass the existing accuracy rate of 72.44% and attain higher accuracy. Similarly, other metrics could also be used for evaluation (precision, F1-score, and recall) to obtain, a more reliable output with more comparative outcomes. In the future, an enhanced custom-CNN model could provide better and more efficient outcomes which could be used as references with other classifier algorithms by different researchers. Thus this study has positive scope in developing and enhancing the lacking features.

***References*:**

1. K. U. Sharma, and N. V. Thakur, A review and an approach for object detection in images, International Journal of Computational Vision and Robotics, 7(1/2), (2017), pp. 196-237.
2. Z-Q. Zhao, P. Zheng, S-T. Xu, and X. Wu, Object Detection with Deep Learning: A Review, IEEE Transactions on Neural Networks and Learning Systems - For Publication, Version 2, (2019), pp. 1-21.
3. X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, Multi-view 3d object detection network for autonomous driving, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - IEEE Xplore, 1161(07559), (2017), pp. 1907-1915.
4. Z. Yang, and R. Nevatia, A multi-scale cascade fully convolutional network face detector, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - IEEE Xplore, 1609(03536), (2016), pp. 1-6.
5. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, Caffe: Convolutional architecture for fast feature embedding, In 22nd MM Proceedings of the ACM International Conference on Multimedia, Orlando, Florida- USA, 1408(5093), (2014), pp. 675-678.
6. Krizhevsky, I. Sutskever, and G. E. Hinton, Imagenet classification with deep convolutional neural networks, Advances in Neural Information Processing Systems, 25, (2012), pp. 1-9.
7. Z. Cao, T. Simon, S-E. Wei, and Y. Sheikh, Realtime multi-person 2d pose estimation using part affinity fields, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - IEEE Xplore, 1161(08050), (2017), pp. 7291-7299.
8. H. I. Dino and M. B. Abdulrazzaq, A Comparison of Four Classification Algorithms for Facial Expression Recognition, Polytechnic Journal, 10(1), (2020), pp. 74-80.
9. S. Kumar, A. Balyan, and M. Chawla, Object detection and recognition in images, International Journal of Engineering Development and Research, 5(4), (2017), pp. 1029-1034.
10. J. Deng, X. Xuan, Wang, Z. Li, H. Yao and Z. Wang, A review of research on object detection based on deep learning, Journal of Physics: Conference Series, 1684(012028), (2020), pp. 1-7.
11. P. Jadhav, V. Koli, P. Shinde, and M. Pawar, Object Detection using Deep Learning, International research journal of engineering and technology (IRJET), 7(9), (2020), pp. 275-279.
12. K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, IEEE Transactions on Pattern Analysis & Machine Intelligence, 37, (2015), pp. 1904-1916.
13. R. Girshick, J. Donahue, T. Darrel, and J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - IEEE Xplore, Columbus, 81, (2014), pp. 580-587.
14. S. Q. Ren, K. M. He, R. Girshick and J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, In Advances in neural information processing systems, Montreal, 1506(01497), (2016), pp. 91-99.
15. R. Girshick, Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, 169, (2015), pp. 1440-1448.
16. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - IEEE Xplore, 1506(02640), (2016), pp. 779-788.
17. J. Redmon, and A. Farhadi, YOLO9000: better, faster, stronger, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - IEEE Xplore, 1612(08242), (2017), pp. 7263-7271.
18. J. Redmon, and A. Farhadi, Yolov3: An incremental improvement, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - IEEE Xplore, 1804(02767), (2018), pp. 161-168.

ISSN: 2037-4445

19. W. Liu, D. Anguelov, D. Erhan, et al., SSD: Single Shot MultiBox Detector, In European Conference on Computer Vision, 1512(02325), (2016), pp. 21-37.
20. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, YOLOv4: Optimal Speed and Accuracy of Object Detection, In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) - IEEE Xplore, 2004(10934), (2020), pp. 1-17.
21. Sunil and Gagandeep, Study of Object Detection Methods and Applications on Digital Images, International Journal of Scientific Development and Research (IJSDR), 4(5), (2019), pp. 491-497.
22. R. Pathak, M. Pandey, and S. Rautaray, Application of Deep Learning for Object Detection, In International Conference on Computational Intelligence and Data Science (ICCIDS-2018), Procedia Computer Science, 132, (2018), pp. 1706-1717.
23. M. Tiwari and R. Singhai, A Review of Detection and Tracking of Object from Image and Video Sequences, International Journal of Computational Intelligence Research, 13(5), (2017), pp. 745-765.
24. Sepal, Waste management dataset updated version 2, (2020). Information on https://www.kaggle.com/sapal6/waste-classification-data-v2
25. P. M. Tripathi, K. Verma, L. K. Verma, and N. Parveen, Facial Expression Recognition Using Data Mining Algorithm, Journal of Economics, Business and Management, 1(4), (2013), pp. 343-346.
26. Y. Zhou, M. Zhang, X. Wang, et al., MPCE: A Maximum Probability Based Cross-Entropy Loss Function for Neural Network Classification, IEEE Access, 7, (2019), pp.: 146331-146341.
27. V. Urkude, and P. Pandey, A Deep Machine Learning Neural Network for Real Time Object Classification using Keras & Tensorflow, International Journal of Technical Innovation in Modern Engineering & Science (IJTIMES), 5(7), (2019), pp. 266-271.
28. Z-Q. Zhao, P. Zheng, S-T. Xu, and X. Wu, Object Detection with Deep Learning: A Review, IEEE Transactions on Neural Networks and Learning Systems – In press, Version 1, (2017), pp. 1-21.