International Journal of Computer Trends and Technology
ISSN: 2231 – 2803 / https://doi.org/10.14445/22312803/IJCTT-V69I9P104

Volume 69 Issue 9, 17-26, September 2021

Original Article

Garbage Classification and Detection for Urban Management

Kishan PS[1], Snehal Shah[2], Jitendra Jaiswal[3]

[1,2] *Computer Science with specialization in Data Science Engineering(Hons)*
[3] *Faculty of Engineering and Technology - Jain University, Kanakapura Taluk, Ramanagara District, Karnataka, India.*

Received Date: 21 August 2021
Revised Date: 22 September 2021
Accepted Date: 03 October 2021

Abstract - *There is an increase in population throughout the country. As the population increases, even the waste production increases. In India, there is garbage accumulation everywhere, including garbage pits and other places which are not meant for garbage accumulation. Due to this management is facing a lot of problems cleaning and there are many diseases spreading.*

This paper is focused on detecting and classification garbage using Deep Learning and Neural Network algorithms. CNN algorithm is used, applied, and analyzed. Confusion matrix and ROC curve are also used.

Here in this paper, two different datasets are used, one dataset was available online, and another dataset was built on our own. This paper comparison both datasets by using different algorithms like CNN, SVM, faster-RCNN, and values are recorded for future use.

Keywords *- Deep learning, Machine learning, Neural networks, CNN, SVM, Faster R-CNN.*

I. INTRODUCTION

The motivation of this project is to keep the city free from waste. In this era, management has become one of the biggest problems that need new technology to overcome. For that, garbage cleaning and collecting are very important. Nowadays, the old technique is outdated as it consumes a lot of time in cleaning and collecting garbage. In this new generation world, many people are eating outside food every day. Very few people throw the waste in the bins, and others throw it on roads or wherever there is the presence of litter. "Wastes are the major factor which is growing with the growth of the country".

Waste management is the major issue in this country which is causing many deadly diseases. We can improve waste management of waste in a smart and better way in this digital world. Technology is a boon that can be used in a proper way to achieve. We have built our own dataset by capturing the images from our locality. We have altogether collected around 700+ images which include garbage and non-garbage images. We have used this dataset and added extra features for analysis and calculation, which increased the accuracy of the existing work. The main objective of this project is to make some development for "Smart city" and "Digital India" and even maintaining cleanliness in the places such as beaches, rivers, institutions, public places, etc.

II. LITERATURE SURVEY

This is to make you gain an understanding of waste management planning concepts, components, strategies, and frameworks that are currently emerging in the field.[1] Significant challenges for waste management are waste collection and path collection. Waste collection is a daily task in cities during which social factors and environmental factors must be considered.[2] At the beginning of the invention of drones, they were called gizm. This vehicle or the aircraft is one that can fly in the sky with no human or pilot on it actually and is operated by a fixed ground station or a moving vehicle autonomously.[3] The TensorFlow-based algorithm, on the other hand, is relatively new as it was developed by Google recently. Being specialized for numerical computation and large-scale machine learning, TensorFlow provides a convenient front-end API for building applications.[4] The system can also be accessed through the Web. To obtain information on the filling of trucks and thus to plan their collection routes in an optimized manner, and facilitating the insertion of data to be made available to users, such as the collection schedule as well as out-of-service notifications for maintenance purposes.[5] The data has been received, analyzed, and processed in the cloud, which displays the status of the Garbage in the dustbin on the GUI on the web browser.[6] Waste must be sorted into categories. Machine learning concepts can be used here to make use of Robots, which will lead to zero human involvement in the future.[7] As the usage of IoT applications is very high, there are lakhs of devices connected which produce a large amount of collected data. Therefore ML plays an important role here in finding patterns, decision making, and analyzing that data.[8] A unique system for the classification of waste materials into groups using SVM, CNN, and ResNet-50 was developed, which gave 87% accuracy on the dataset.[9] A robot was introduced that could move on the ground for floor cleaning. They claimed that from the proposed architecture, the garbage was detected accurately using deep learning mechanisms.[10] A sensor is placed on the front of the waste in, these bins detect the objects which are approaching within 3 feet from it and lid of the bin will automatically open, and if the bin is full it will not work until and unless a reset buton on the bin is pressed after it is emptied.[11] Classification of waste for detection of polyethylene was done using CNN and image processing. Experiments to detect terephthalate, polyethylene, high-density polyethylene, polypropylene, and polystyrene were also done.[12] An Integrated system has been introduced, which is combined with Radio Frequency Identification, Global Position System, General Pocket Radio Service, Geographic data system, and web camera for solving the matter of sol waste. It gave better results in terms of speed of data transmission, real-time, precision, and reliability.[13] To avoid traffic jams, the truck routes are being planned by developing an optimal path planning algorithm that calculates the number of trucks to be used for waste collection and also minimizes the distance, and shows the route to be taken.[14] Cloud technology and mobile application are the classifications used for monitoring. The waste management process is automated by an Integrated sensing system. The smart bin uses ultrasonic and various gas sensors, which helps in automatically sensing the varied odorous gases and the maximum limit of waste. The information is then transmitted to the responsible authority.[15] Detecting garbage as well as marking approximate regions of garbage in an image was the main objective of this research, and it was achieved by training the neural network on the training dataset from GINI and testing it on test images. It was found that CNN had the highest performance among all, with an accuracy of 89.81%. [16] A smartphone-based app– “Spot Garbage” detects garbage and identifies the location of the garbage. CNN architecture is used for identifying wastes in images in this application. Garbage In Images (GINI) Bing search API was used to crawl the web for images of garbage and locations where patches of garbage were used for classification from the taken picture. [17] An automatic sorting trash can – “AutoTrash“, separates trash on the basis of compost and recycling properties. This is done using the Raspberry pi camera attached to the model.[18] The google net-based app is developed to localize and classify wastes which claim to have an accuracy rate of 63% to 77% for different waste types.[19] Faster R-CNN technique was used to get regin proposals and object classification, reaching an mAP of 68.3% in which waste was categorized into three different classes like paper, landfill, and recycling.[20] In water bodies, a deep transfer learning approach is applied to detect waste. With the help of the dataset - “AquaTrash" on a single stag detector Reti-naNet, the AquaVision model detects various waste items in which Resnet50 is used as the backbone and FPN.[21] In this paper, a model is proposed for improving waste collection, in which users take pictures of the waste item and then upload it to the waste collection company, which goes to the server, where the image will be identified and classified automatically using an image recognition system.[22] In this, a very hygienic and cheap system is proposed for detection of the garbage using AI algorithms in which robots are going o be used as a substitute for humans for finding and collecting the garbage[23] For increasing current rates of recycling and to reach the rates of demand, for the process of sorting waste, robotic technology a be implemented for selected waste materials.[24] The survey shows that in the waste management sector, many challenges like automating tasks that are hazardous for humans can b addressed using digital tools. Currently, in the field of waste management, few important digital technologies being used are cloud computing, robotics, data analytics, the internet of things, and artificial intelligence.[25] This system of smart garbage management uses a microcontroller, Wi-Fi module, and IR sensor. In this dustbins are cleaned as soon as it reaches its maximum level. If the cleaning is not done at a specific time, then it sends a message to the higher authority for taking appropriate actions against the concerned contractor.[26] Smart system for monitoring Dustin, which segregates the wastes into wet and dry, which therefore reduces manpower. [27] The YOLO v2 model is improving for detecting and recognizing garbage by the method of classifying network pre-train, target box dimension clustering, other methods. This has lower costs and better performance compared to the traditional system.[28] DL approach is proposed for real-time detection of garbage and cleanliness. To collect the street images, a camera is installed. the information which is processed is transmitted to the data server for analysis in the network. Faster RCNN is used for detecting the garbage on the street and is classified into categories for calculating the quantity of the garbage that is detected on frames.[29] Using tech in robot control, computer vision, and many different fields to take advantage of its maturity, built a system that visually classifies and separates the waste into different types effectively.[30]

III. METHODOLOGY

A. Data set

At the beginning of the work, the dataset was taken, which was available online. The dataset was taken from kaggle and implemented. Two different datasets from Kaggle are taken and merged, and then implemented in the CNN code. Here in this paper, different folders are created having different images of glass, plastic, metal, cardboard, and trained and tested the data set. Around 2500 images are used for this process to find the accuracy, misclassification, and perception. In this paper, our own dataset by capturing images of trash or garbage is prepared. Around 800 images of garbage to train and test the model are collected. Both the top view and side view of garbage are collected, also collected images with some errors and disturbance. Images are collected during dark, light, and sunny so that it will be easy for machines to learn and understand better. Images of non-garbage are collected to make machines understand that the images of non-garbage should not be captured or recognized. We have used around 300 images to recognize which are non-garbage, trained and tested them using Faster RCNN, and noted the result.

Fig. 1 Garbage(daylight) **Fig. 2 Garbage(night)**

B. Architecture

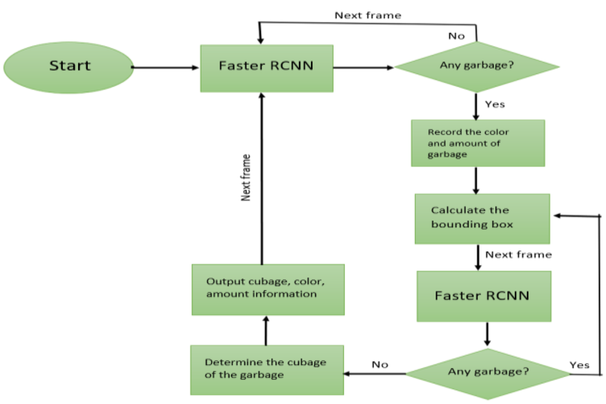


Fig. 3 CNN architecture

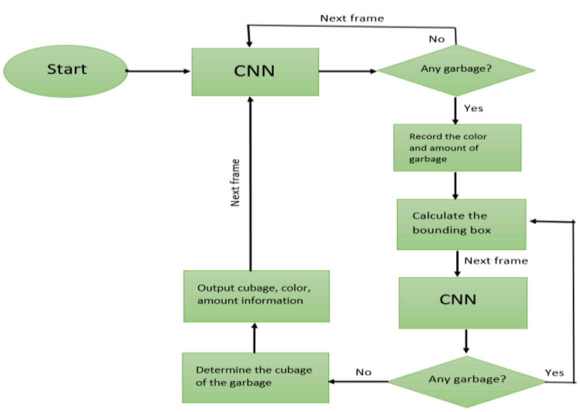


Fig. 4 Faster RCNN architecture

The architecture figure-3 shows the working of CNN and explains how our model works.

The architecture figure-4 shows the working of Faster RCNN on our model.

C. Sequence Diagram

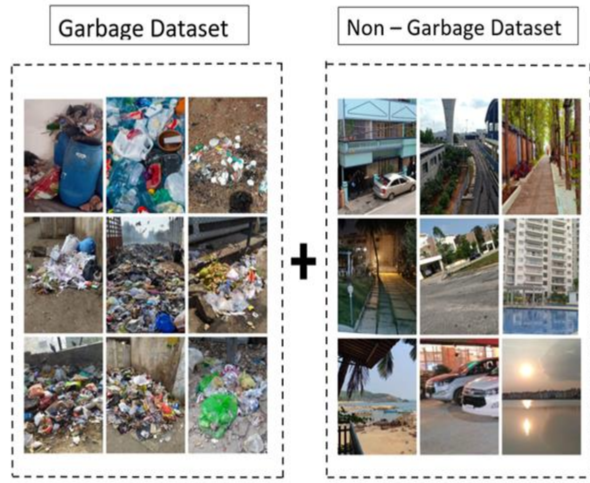


Fig. 5 Collection of the dataset

IV. TOOL DESCRIPTION

❖ Hardware Requirements

A minimum of 8gb RAM for our code is required to run smoothly and without any loss of data. A minimum of 250gb hard disk is required to run smoothly. A Graphics card is an addon.

❖ Software Requirements

Explanation about the software tools used in the paper are as follows:

- Deep Learning
- TensorFlow
- Keras
- Matplotlib
- CNN
- Confusion matrix
- AUC-ROC curve
- Faster RCNN
- SVM Support Vector Machine

A. Deep Learning

Deep structured learning is a part of machine learning methods that are based on artificial neural networks. Supervised, Semi-supervised and unsupervised learning can be used here. Computer vision, machine vision, speech, audio and object recognition, NLP natural language processing, image processing, and many more fields of deep learning can be applied.

As there are multiple layers in the network, the name “Deep” is used.

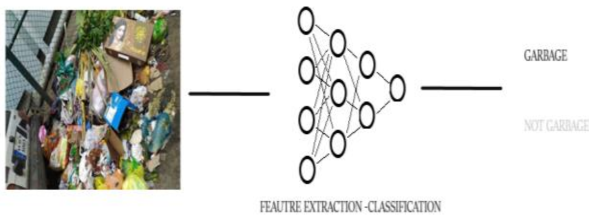


Fig. 6 Deep learning

B. TensorFlow

For deep learning, applications google primarily developed an open-source library known as TensorFlow. Traditional machine learning is also supported by TensorFlow. Originally it was developed for computing large numerical values. It is a math library that is based on data flow and also differentiable programming.

C. Keras

It is a neural network library. It is used for developing, evaluating deep learning models. It is an API mainly designed for human beings. It is used for reducing cognitive load, minimizes the number of user actions, and provides clear, actionable feedback upon user errors.

D. Matplotlib

Pyplot is an Application programming interface for matplotlib which acts as an open-source alternative to MATLAB. Matplotlib is used for the visualization of data in the form of graphs, charts, and plots.

E. CNN(Convolutional neural networks)

In deep learning, a class of neural networks is called Convolutional neural networks (CNN or ConvNet), which is commonly used to analyze visual imagery. Based on their shared-weights architecture and translation invariance characteristics, CNN is also called shift invariant or space invariant artificial neural networks. There are two layers, namely the input layer and output layer in CNN. With these layers, it also has many multiple hidden layers. The core building block of a CNN is a convolutional layer whose parameters consist of a set of learnable kernels, which is a small matrix. Because of its very good accuracy, CNNs are used for image classification and image recognition.

CNN allows extracting higher representation for images. In CNN, we define the image features and takes the raw pixel data of images, then train the model, extract the features automatically for classification. A convolution sweeps the window of images and calculates the input to filter dot product pixel values which allow convolution to highlight the relevant features.

In CNN, the input data is given where the next process is convolution and ReLu, which is the most commonly deployed activation function for the outputs of CNN neurons. Pooling is the next step in the network to reduce the area size and to reduce the number of parameters. This process continues multiple times. Under feature learning, the cycle of convolution, Relu, and pooling comes. Under the classification, there are three layers: flatten layer, fully connected layer, and a softmax layer. The final layer of the neural network is the softmax layer. A convolutional neural network can take in an input image, assign importance which is weights and biases, to various objects in the image and have the capability to differentiate one from the other.

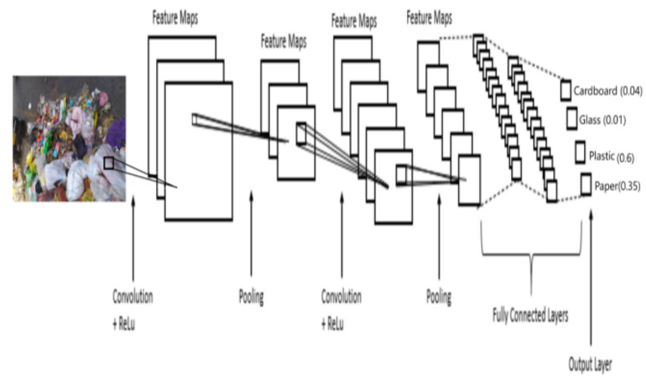


Fig. 7 CNN workflow

F. Confusion Matrix

For summarising the performance of a classification algorithm, there is a technique known as a confusion matrix. If we have an unequal number of observations in each class, classification accuracy alone can be misleading, and also if we have more than two classes in our dataset.

A confusion matrix is used for describing the performance of a classified model. The classification model consists of true positives and negatives, false positives, and negatives. When N is the number of target classes, an N x N matrix is used for evaluating the performance of a model, which can also be called a confusion matrix. The main purpose of using confusion matrices is that they can be used to visualize predictive analytics
like recall, accuracy, precision, and specificity. How confused the model is between the classes is shown by the confusion matrix.

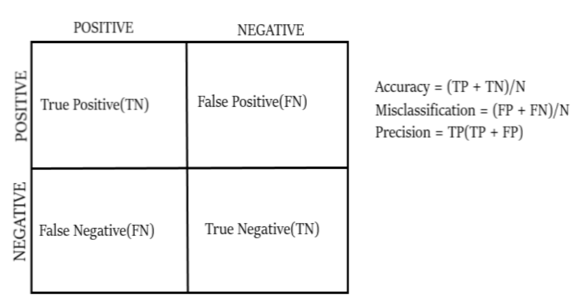


Fig. 8 Confusion Matrix

G. AUC-ROC curve

The area under the curve and Receiver operating characteristics curve are the full form of the AUC-ROC curve. The AUC-ROC curve is one of the most important evaluation metrics. Multi-class classification problem performance can be checked and visualized using this curve. The higher the area under the curve, the better the model is at predicting. When the AUC is higher, better, the model is differentiating (Yes and No).

Defining the terms which are used in the above graph:

TPR (True Positive Rate)/Recall/Sensitivity

TPR => TP / (TP+FN)

Specificity = >TN / (TN+FP)

FPR = 1 – Specificity => FP / (TN + FP)

Relationships between sensitivity and specificity are inversely proportional to each other, whereas TPR and FPR are directly proportional to each other.

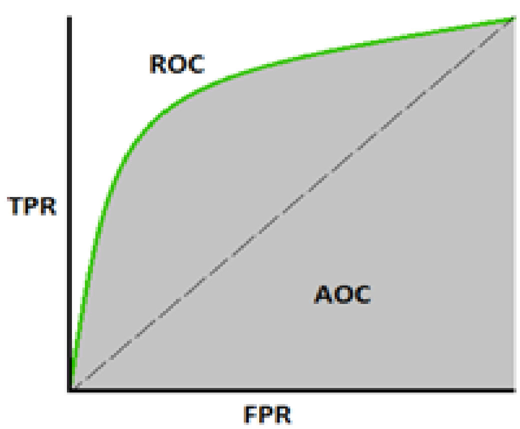


Fig. 9 AUC-ROC curve

H. Faster RCNN

Faster RCNN is a modified version of RCNN. The difference between RCNN and Faster RCNN is RCNN uses the selective search for generating Regions of interest, whereas Faster RCNN uses Region Proposal Network. Faster RCNN can run just once per image and get all regions instead of running a CNN 2000 times per image.

Here take the image as input. This image is passed to CNN, which generates Regions. The pooling layer is applied to all the regions which are generated to reshape them as per the input of CNN. Then these individual region is passed on to a fully connected network. A softmax layer is used on the fully connected layer. Along with the softmax layer, the linear regression layer is also used. The regression layer is used to output the bounding box coordinates. Instead of using different models, this model uses a single model which extracts features from the regions, divides them into classes, and returns the boundary boxes to identify classes simultaneously.

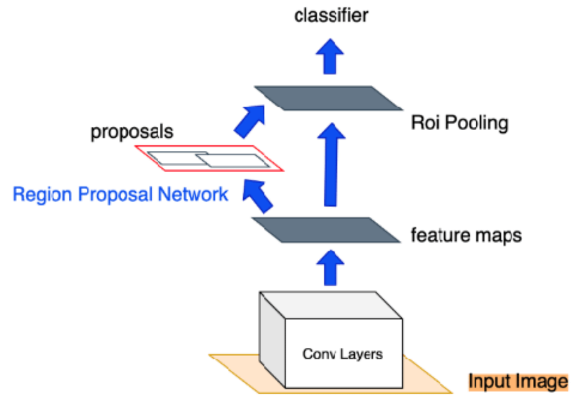


Fig. 10 Faster RCNN

I. SVM Support Vector Machine

In the segment of machine learning, support vector machines have a massive and positive impact in terms of observing and identifying the machine learning models. On the other hand, the main motive of the support vector machines is to analyze data and critical information for regression and classified analysis along with the help of learning algorithms. In the matter of detecting the false news from the research article and the news articles, the text summarization method is very crucial and important in the field. The text summarization is the combination of two huge broad engagements, such as abstraction and extraction. In the area of extraction method, this establishes a conclusion or summary paragraph which has been extracted from the suggested information or the text file. On the other hand, the abstraction methods introduced the summary by sowing the seed of an amount of fresh text from the suggested names or texts. This type of fresh summary text can transport the essence of the real informative text. In the area of text summarization process, there are several algorithmic methods under this section, such as Latent Semantic Analysis, Text Rank, and Lex rank.

V. IMPLEMENTATION

In the beginning, the CNN algorithm was implemented and calculated a few functionalities like finding the ROC curve, confusion matrix, and plotted graphs for training and testing of our model. CNN for object classification is used in our model. 7 different folders are created and classified the data to train and test the model. Folders are created in this way respectively: cardboard folder which contains images of cardboard, glass folder which contains different sizes and shapes of glasses, the metal folder contains metal items, paper folder, plastic folder, lump data which has a collection of data, trash folder. Around 2635 images are implemented for our analysis.

A. Application of CNN algorithm on an online dataset

While implementing the model few steps are followed, below are the steps:

a) Importing libraries

Installed TensorFlow python library.

Installed Keras, python library. Using these imported many functionalities for callbacks, layers, models, image processing.

Installed matplotlib python library for plotting images.

b) Generating Data

The batch size(single iteration in a number of the training set) is set to 32. The general size of the batch is 32, so I took it into consideration and used it.

Generated the data by using ImageDataGenerator for test and train data. Split the data into 0.1, which results in 2374 train data and 261 test data.

Using the label function, label the items and printed them for display so that how many labeled folders and data are there will be known.

c) Build Model

Build the model by using the Keras library functions by setting the values of Conv2D, MaxPooling2D, Dropout, Dense to make our model look in proper size and shape without any changes. When the images are of the same size and shape, it will be easy to analyze the tension. Adding layers to the model to make predictions.

Display the summary of the model to show the output shape and params of the layers, which gives a proper explanation.

d) Compiling the Model

Compiled our model using the compile attribute. Below are the parameters which should be used:

Optimizer: This is used to pass the optimizer which is to be used. Optimizers like SGD, Adam, and various other optimizers can be used. Here Adam optimizer is used.

Loss: Pass the loss function in the model.

Metrics: Pass the metric score function for the model to be scored. Here accuracy for our metric score is used.

e) Fitting the Model / Training the Model

Here in fitting, pass various parameters like batch size, epochs, validation data, and many more. Calculated Epochs on our model. Fit a model on the data and use that model. Used the data, i.e., the training data, for fitting the model.

First, the model path was given, and later epochs were set to 100 and trained the model. When epochs started to increase, the accuracy of the model also increased, and the loss of the model was decreased.

f) Model Evaluating

Once the model fitting is done, evaluate the model. By using this, it was possible to get the loss and accuracy of the model and print the score. Got 0.7695 accuracies, and the loss of the model was 0.6915.

g) Predicting the output

Here in this paper, output was predicted. Used the data, which was kept for testing purposes. By using it was possible to predict the output. Using the predicted keyword, test the model.

Plotted graphs for train and test using matplotlib.

Calculated the confusion matrix for the dataset. Calculated accuracy, precision, and misclassification.

Built a ROC curve.

B. Application of CNN algorithm on our own dataset

Here installed the required packages and implemented them. Set the image path to select the dataset. Two folders, one is a garbage dataset, and another is the non-garbage dataset, are created.

Plotted the images of garbage and non-garbage. Set the batch size to 32. Labeled the data here and generated it. Take the folders as classes. In this paper, there are 2 classes. Built the model for calculating the sequential values of the model. Compiled and trained the model to find the loss and accuracy.

Evaluated the model and noted the values. Took the train history and plotted the graph for loss, accuracy, val_loss, val_accuracy.

C. Application of SVM algorithm on our own dataset

Loaded the required packages like numpy, pandas, matplotlib, and OpenCV for SVM construction. Categorized the dataset into garbage and non-garbage. Trained the dataset and labeled the dataset. To increase the performance of the algorithm and for a better understanding of the algorithm, the image size was given. Created the array for the x-axis and shaped it into an array.

Took the y-axis and shaped it into an array for calculation purposes. To get the best fit for the dataset support vector classifier was used. So that later, it will be possible to predict the new images for classification.

D. Application of Faster RCNN on our own dataset

Here, to construct a model on Faster, RCNN used a pycharm platform and built the model. Imported the packages which are required to build our model. Used a parser function that takes input and builds the model for live detection. Allocated the memory for GPU and provided the image directory path.

Provided the model path and then labeled it. Given minimum threshold and then loaded the model.

Imported the model and packages for object detection using runtime. Saved the model for the machine to understand and built a detection function. Loaded the label map data for plotting.

Imported numpy, image from PIL, matplotlib, and warnings to suppress the matplotlib warnings. Give the path of the image for detection as it is the only backend that gives the image for detection of the image. Using OpenCV, read the image and then as the image needs to be tensor convert using TensorFlow converter command, and then model expects a batch of images have added an axis with the new axis function.

Here start detecting the images for the result to mark the boundaries and specify which part of the image has garbage, and if garbage is not found, then not to detect that part. Later, to display the output used in the image, show a command to display the images with detection using OpenCV.

VI. RESULT AND ANALYSIS

A. Result

a) Result analysis on an online dataset using CNN algorithm

The below is the result discussion of the CNN algorithm in detail.

Here, when trained and tested the data, the losses are getting less; in the starting, it is about 1.8 loss, and in the end, it comes around 0.6. In the graph below, you can see the graph "Loss v/s Epoch".

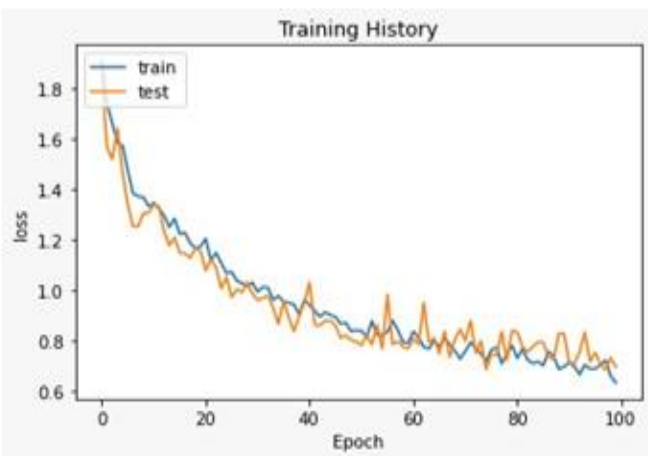


Fig. 11 Graph of Loss vs. Epoch

And, in the below fig-12 graph, we can see the accuracy of the train and test dataset; initially, it was low, and later-wards it increased slowly and steadily, and approximately 0.25 to around 0.8. In the fig graph, you can see the graph between "Accuracy and Epoch".

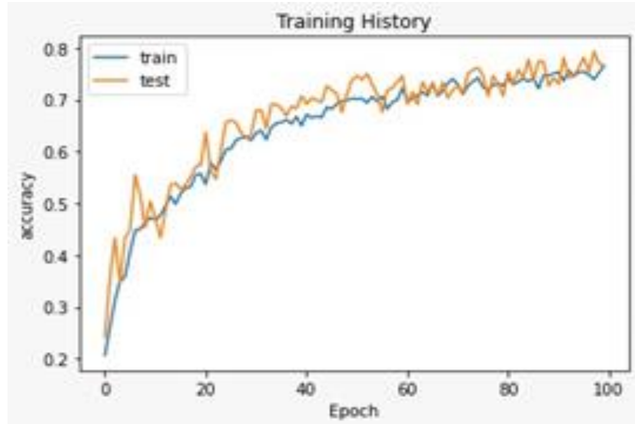


Fig. 12 Graph of Accuracy and Epoch

b) Result analysis on our own dataset using CNN algorithm

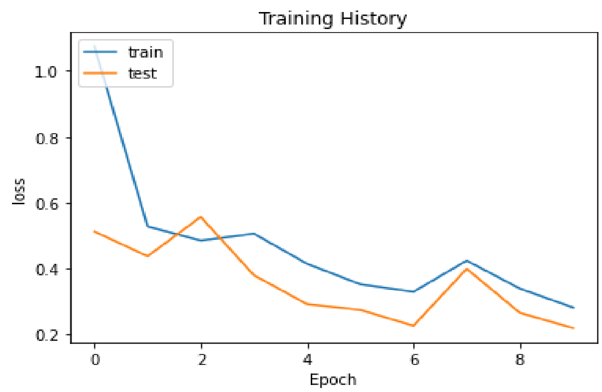


Fig. 13 Graph of Loss and Epoch

Here is plotted loss and epoch graph for the training model where the loss of the model has reduced as the epoch increases.

Here the loss is 0.282 for 10 epochs. The above fig-13 gives more accurate visuals for better understanding.

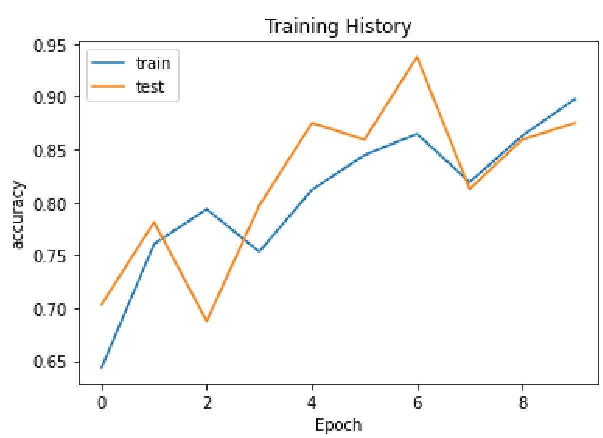
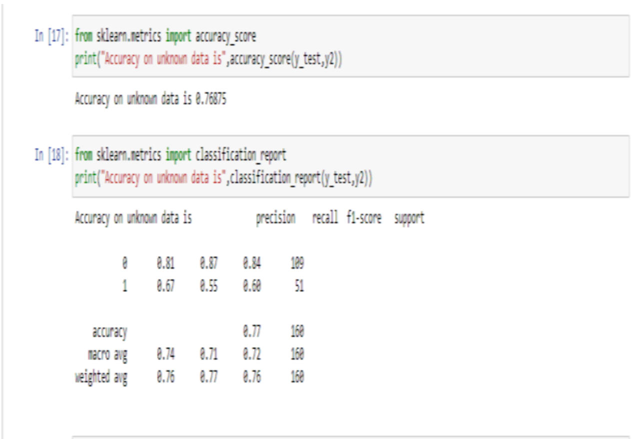


Fig. 14 Graph of Accuracy and Epoch

Here plotted accuracy and epoch graph. From this graph above, one can say that as the epoch increases, the accuracy of the model will also increase.

Here the accuracy is 0.897 when plotted with 10 Epoch. The above plot gives a better understanding.

c) Result Analysis on our own Dataset using SVM Algorithm



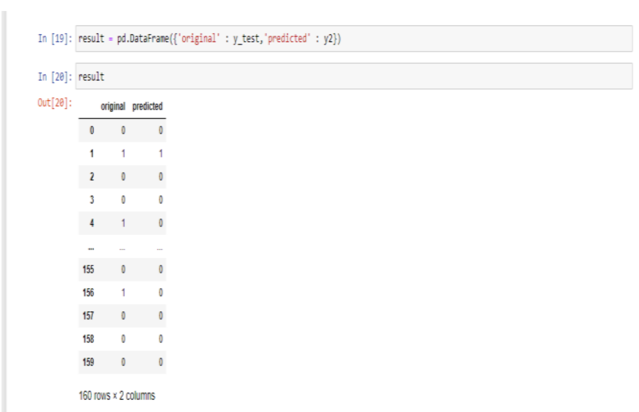
```
In [17]: from sklearn.metrics import accuracy_score
         print("Accuracy on unknown data is",accuracy_score(y_test,y2))

Accuracy on unknown data is 0.76875

In [18]: from sklearn.metrics import classification_report
         print("Accuracy on unknown data is",classification_report(y_test,y2))

Accuracy on unknown data is               precision    recall  f1-score   support

           0       0.81      0.87      0.84       109
           1       0.67      0.55      0.60        51

    accuracy                           0.77       160
   macro avg       0.74      0.71      0.72       160
weighted avg       0.76      0.77      0.76       160
```

Fig. 15 SVM Accuracy

Using Sklearn imported the accuracy score to note down the accuracy of our model.

Got 0.768 accuracies on our model when trained and tested the model.

Imported the classification report to get a detailed report of our model. The above figure shows the result of the model that has been reduced. This paper noted key terms that are precision, recall, F1-score, support, accuracy, macro average, and weighted average. This helps in a better understanding of our model.



```
In [19]: result = pd.DataFrame({'original' : y_test,'predicted' : y2})

In [20]: result

Out[20]:
```

	original	predicted
0	0	0
1	1	1
2	0	0
3	0	0
4	1	0
...	...	...
155	0	0
156	1	0
157	0	0
158	0	0
159	0	0

160 rows × 2 columns

Fig. 16 Original and Predicted result

Here is the result of the original and predicted score, which is in the form of a table that gives a better understanding of our work in the model.

d) Result analysis on our Dataset using Faster RCNN Algorithm

Fig. 17 Detection of garbage using Faster RCNN

As by seeing above, this is the result of the detection obtained when used Faster RCNN and got an accuracy of 65% on training the dataset.

B. Analysis

a) Analysis of CNN algorithm on an online dataset

The below fig-18 is the result got when implemented the data to the confusion matrix. 1500 true positive images and 83 false-positive images and 267 false negatives, and 524 true negative images were obtained.

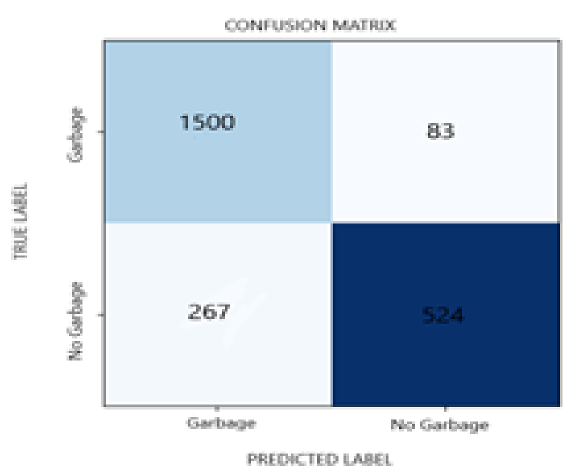


Fig. 18 Confusion Matrix

By the confusion matrix fig, we calculated the following,

Accuracy: 85.52%

Misclassification: 14.74%

Precision: 94.75%

The fig-19 is the ROC curve for the given data, which performs the classification threshold. From the fig-19, one can differentiate between the class, i.e., true positive and negative classes.

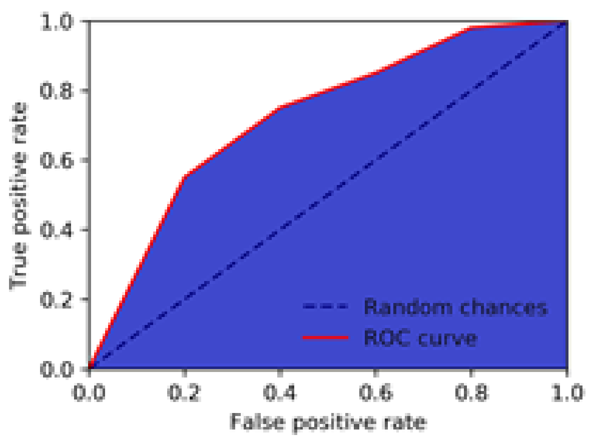


Fig. 19 AUC-ROC curve

Evaluated the model and got the result [0.6915320158004761, 0.76953125].

Comparison of object detection and object classification

Table 1. Comparision on online and own dataset

	CNN(object classification)	SVM(object classification)	Faster R-CNN(object detection)
Online dataset accuracy	85.5%	76.9%	80.3%
Our Own dataset accuracy	89.7%	76.8%	65.2%

VII. CONCLUSION AND FUTURE SCOPE

From this paper, it can be concluded that garbage classification is an important perspective for social welfare, and it becomes important because human intervention should be avoided to tackle many things such as sewage cleaning and intentionally filthy premises. At the age of making robotic intervention to clean garbage things, so if a robot could classify them, only it can clean a particular area. Here, in this paper implementation of CNN(Convolutional Neural Network) as a deep learning algorithm to classify garbage as either garbage or no garbage. This paper has a focus on individual objects since every individual object may not be garbage. Here, the accuracy of 85.52% is achieved and later verified the result with a confusion matrix and AUC-ROC curve. Implementation of Faster RCNN on our own dataset and calculated the result and got an accuracy of 65.25%. Implementation of SVM for our own dataset and got a result of 76.8% accuracy. Accuracy on our own dataset using the CNN algorithm is calculated and got an accuracy of 89.7%.

VIII. FUTURE SCOPE

The future scope of this model is to build a model with IOT for detection of Garbage and note the recordings and send it to the respective in charge.

It will be easy to detect the garbage using image processing and then implement it to IOT devices and help in improving the environment and reduce pollution that leads to a shift towards AI devices which will be a great help to society.

It will be the best model to clean the city, and our models help in locating the contaminated area more precisely and accurately.

REFERENCES

[1] Brennan Gillis, Bob Kenney, Martin Gillis, Mike Wilkinson, Michelle Adams, Nicole Perry, Carla Hill, Rochelle Owen - Waste Management Practices: Literature Review pg:2

[2] Tran Anh Khoa, Cao Hoang Phuc, Pham Duc Lam, Le Mai Bao Nhu, Nguyen Minh Trong, Nguyen Thi Hoang Phuong, Nguyen Van Dung, Nguyen Tan-Y, Hoang Nam Nguyen, and Dang Ngoc Minh Duc - Waste Management System Using IoT-Based Machine Learning in University

[3] Saurabh Jotwani1, Avesh Sheikh2, Prof. Urvashi Agrawal3, Dr. Narendra Bawane4 - To develop a Garbage detection Drone. Journal of Interdisciplinary Cycle Research Research , 441

[4] Anjali Pradipbhai Anadkat, B V Monisha, Manasa Puthineedi, Ankit Kumar Patnaik, Dr. Shekhar R, Riyaz Syed - Drone-based Solid Waste Detection using Deep Learning & Image Processing. Alliance International Conference on Artificial Intelligence and Machine Learning (AICAAM), (2019) 362.

[5] Kellow Pardini, Joel J.P.C. Rodrigues, Ousmane Diallo, Ashok Kumar Das, Victor Hugo C. de Albuquerque, and Sergei A. Kozlov - A Smart Waste Management Solution Geared towards Citizens. Sensors (2020), 20, 2380. Pg: 12 of 15

[6] Parkash, Prabu V - IoT Based Waste Management for Smart City. DOI: 10.15680/IJIRCCE.2016. 04020 Pg: 1267

[7] Praveen Kumar Gupta, Vidhya Shree, Lingayya Hiremath and Sindhu Rajendran - The Use of Modern Technology in Smart Waste Management and Recycling: Artificial Intelligence and Machine Learning

[8] Fotios Zantalis, Grigorios Koulouras, Sotiris Karabetsos and Dionisis Kandris - A Review of Machine Learning and IoT in Smart Transportation. Future Internet 11(4) (2019) 94.

[9] Olugboja Adedeji, Zenghui Wang - Intelligent Waste Classification System Using Deep Learning Convolutional Neural Network

[10] M. A. Viraj J. MUTHUGALA, S. M. BHAGYA P. SAMARAKOON and MOHAN RAJESH ELARA - Tradeoff Between Area Coverage and Energy Usage of a Self-Reconfigurable Floor Cleaning Robot Based on User Preference

[11] Subham Chakraborty, Sayan Chowdhury, Soumyadip Das, Sayansri Ghosh, Rimona Dutta, Sayan Roy Chaudhuri - Segregable Smart Moving Trash Bin

[12] Bobulski, J., and Kubanek, M - Waste Classification System Using Image Processing and Convolutional Neural Networks. Springer International. (2019).

[13] Md. Shafiqul Islam, Maher Arebey, M.A. Hannan, Hasan Basri - Overview for solid waste bin monitoring and collection system. Innovation Management and Technology Research (ICIMTR), International Conference 21-22 (2012).

[14] Maria-Victoria Bueno-Delgado, Jose-Luis Romero-Gazquez, Pilar Jimenez and Pablo Pavon-Marino - Optimal path planning for selective waste collection in smart cities, Sensors, (2019).

[15] Debajyoti Misra, Gautam Das, Triankur Chakrabortty and Debaprasad Das - IoT-based waste management system monitored by cloud. Journal of material cycles and waste management 20 (2018).

[16] Mandar Satvilkar – Image-Based Trash Classification using Machine Learning Algorithms for Recyclability Status (2018)

[17] Gaurav Mittal, Kaushal B.Yagnik, Mohit Garg & Narayanan C. Krishnan – SpotGarbage: a smartphone app to detect garbage using deep learning. (2016).

[18] J. Donovan – Auto-Trash Sorts Garbage Automatically (2018).

[19] Rad, Kaenel, Droux - A computer vision system to localize and classify wastes on the streets (2017).

[20] Ren, He, Girshick and Sun - Faster RCNN: Towards Real-Time Object Detection with Region Proposal Networks (2017).

[21] Harsh. P, P.K. Gupta, M. Khubeb S, R.Morales-Menendez, Prakhar B, Sudhansh S, Iqbal H. S - AquaVision: Automating the detection of waste in water bodies using deep transfer learning (2020).

[22] Piotr. N and Teresa. P - Application of deep learning object classifier to improve e-waste collection planning (2020) 1-9.

[23] Siddhant. B, Seema. P, Ishita.S, Prof. Alpesh. P, Prof. Jagruti.M, Dr. Rajesh.T- Automatic Garbage Detection and Collection

[24] R. Sarc, A. Curtis, L. Kandlbauer, K. Khodier, K.E. Lorber, R. Pomberger - Digitalisation and intelligent robotics in the value chain of circular economy-oriented waste management – A review. (2019) 472-492.

[25] Holger Berg , János Sebestyén , Phillip Bendix , Kévin Le Blevennec (VITO), Karl Vrancken - Digital waste management (2020).

[26] Neha.S, Sayli. B, Shraddha.T, Varsha.D - A Survey on Garbage Collection and Monitoring System for Smart cities using IOT 2018 v5

[27] Anushree. K, Panchami.S, Pooja.S, Pratheeksha. N, Namratha. P - Smart Dustbin Monitoring System (2019).

[28] Ying. L, Zhishan. G, Guoyun. L and Shikai. W - Research on Automatic Garbage Detection System Based on Deep Learning and Narrowband Internet of Things (2018).

[29] Ashok.S, Arun Kumar.HN, Niranjan, Shekar.A, Arun Kumar.DR - A Deep Learning Approach for Real-Time Garbage's Detection and Cleanliness Assessment (2020).

[30] Alvaro.S, Javier.P.C and Ignacio.RN - Intelligent Garbage Classifier ISSN 1989-1660