

# Trash Detector with Machine Learning and Deep Learning

Dr. Preeti Sharma<sup>1</sup>, Abhishek Singh<sup>2</sup>, Diksha Pandey<sup>2</sup>, Jaypratap Singh Chahuan<sup>2</sup>

<sup>1</sup>Faculty, Information Technology, Inderprastha Engineering College Ghaziabad, Uttar Pradesh, India

<sup>2</sup>Student, Information Technology, Inderprastha Engineering College Ghaziabad, Uttar Pradesh, India

\*\*\*

**Abstract** - With the development of smart cities in major cities at home and abroad, especially the management of smart cities, how to improve the intelligence level of urban environment **monitoring and evaluation** has become an important research topic. It is of great value to rapidly and accurately **detect trash** from urban images in the application of intelligent urban management. This paper aims to adopt a deep learning strategy for **automatic trash detection**.

By using an open **source framework and model**, we look over trash detection results on trash images. In addition, to improve the accuracy of the method, a **data fusion and augmentation strategy** is proposed. As a result, experiments show that the method has favorable generalization ability and high-precision detection function

## 1. INTRODUCTION

Monitoring and cleanliness assessment of trash areas in urban scenes mainly rely on manual inspection and photographic record, which makes it a difficult and time consuming task. During the inspection process, human intervention and cumbersome problems often happen. The quality of sanitation work has been affected. Unlike pedestrians, vehicles and other objects, trash has no relatively clear definition. Due to the judgment of trash always has certain subjectivity, in different situations, it will produce different judgment results. Since the diversity of scenes where trash appears, accuracy of test results will be affected. With the development of smart cities, we expect to provide an automatic detection method of urban trash to help alleviate urban trash problems.

## 2. Literature Survey

Before the development of deep neural networks, features were manually designed, then followed by a classifier. Some research focused on the classification and recycling of trash a few years ago. For example, Sudha S et al. proposed a model for classifying objects as biodegradable and non-biodegradable. Although the traditional object detection already has some mature techniques, due to the morphological diversity, illumination diversity, background diversity and other factors of the target object, the detection precision for the unfixed form objects such as urban trash is still a tough problem to solve.

The past decade has witnessed a rapid development of massive data and high-performance computing systems such as graphics processing units (GPUs). Now region

based CNN detection methods have dominated many tasks of computer vision. It is such an exciting area that can extract the high-level features and the hierarchical feature representations of the objects. Girshick et al. introduced a region-based CNN (RCNN) for object detection, from 2014 to now, R-CNN, Fast R-CNN, Faster R-CNN, ION, HyperNet, SDP-CRC, YOLO, G-CNN, SSD and other increasingly fast and accurate object detection methods have emerged. There is very little research on trash detection at home and abroad now.

We invent Software which detects the trash on the road or street. We trained the model with labeled images. All the images are classified into 6 categories according to its size. This has been done for increasing the accuracy. These images are placed in six different folders and given names Atrash, Btrash to Ftrash. This helps us to detect the target objects such as plastic bottles from a height of several meters.

The major contributions of this paper are presented as follows.

1) We created a model to detect the trash on the street in real time. For this we trained our model with a large image dataset. We labeled the image before use. This consumes much time but for increasing accuracy it has to be done manually. No other Choice.

2) To optimize the performance of the model, we collect urban scene images containing trash and urban scene images without trash. By using a fine tuning strategy, we used the pre-trained model ssd mobilenet v1 for transfer learning which helps a lot in training model with good accuracy. All these images are categorized in different folders according to the size.

3) After all of these, the main work is to be done. We use tensorflow lite Converter to convert model for weak performance systems such as mobile. Tensorflow Lite, the next evolution of TensorFlow Mobile promises better performance to leverage hardware acceleration on supported devices. It also has few dependencies, resulting in smaller binaries than its predecessor. TensorFlow Lite is TensorFlow's lightweight solution for mobile and embedded devices. It enables on-device machine learning inference with low latency and a small binary size. TensorFlow Lite supports hardware acceleration with the Android Neural Networks API.

## 4. METHODOLOGY AND TECHNIQUE

### a. Installing TensorFlow-GPU 1.5

Type the following command in anaconda prompt to install Tensorflow GPU. C:\Users\Tensorflow> pip install tensorflow-gpu.

### b. Setting up TensorFlow Directory

The TensorFlow Object Detection API requires a specific directory structure as in its GitHub repository. It also requires several additional Python packages specific to the environmental variables (PATH and PYTHONPATH variables) and few commands to run or train an object detection model.

### c. Setting up the Anaconda Virtual Environment

### d. Gathering and Labeling Images

TensorFlow requires hundreds of images of an object to train a good detection classifier. To train a robust classifier, the training images must have random objects in the image along with the desired objects, variety of backgrounds, and lighting conditions. The images may include the partially obscured images, overlapped images, or only halfway in the picture.

### e. Generating Training Data

### f. Creating Label Map and Configuring Training

### g. Running the Training

### h. Exporting the Inference Graph

### i. Bazel Installation (LINUX)

Bazel is an open-source build and test tool similar to Make, Maven, and Gradle. It uses a human readable, high-level build language. Bazel supports projects in multiple languages and builds outputs for multiple platforms. Bazel supports large codebases across multiple repositories, and large numbers of users.

### j. Build in Android Studio with TensorFlow Lite AAR from JCenter

Use Android Studio to make changes in the project code and compile the demo app

### k. Deploying the model in Android

### l. Running the model Following are the steps to run a model:

- i. Build an Interpreter based on an existing FlatBufferModel
- ii. Optionally resize input tensors, if the predefined sizes are not desired.
- iii. Set input tensor values
- iv. Invoke inference
- v. Read output tensor values class Interpreter

### m. Object Detection API

TensorFlow Lite is designed for fast inference on small devices, which provides the APIs to avoid unnecessary copies at the expense of convenience.

### n. Compile Android app

### o. Install the app

### p. Sample Output

The following are the output images showing trash images detecting in real time with bounding boxes over trash with their labels.

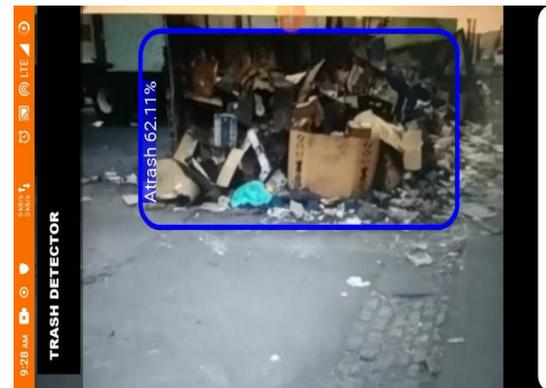


Figure 1

## 5. Architecture and General :-

TensorFlow Lite supports both Android and iOS platforms. The initial step involves conversion of a trained TensorFlow model to TensorFlow Lite file format (.tflite) using the TensorFlow Lite Converter. This converted model file is used in the application.

Following are the important components for deploying the model as shown in the architecture diagram:

1. Java API: A wrapper around the C++ API (for Android).
2. C++ API: The C++ API is responsible for loading the model file and invoking the interpreter for further processing and execution. The same library is used for Android and iOS.
3. Interpreter: Interpreter executes the model using the defined kernels. The interpreter supports selective kernel loading and developers can also define their own custom kernels that can be used to perform the execution.
4. On few Android devices, the Interpreter uses the Android Neural Networks API for hardware acceleration or default to using a CPU.

### 6. Architecture Image:-

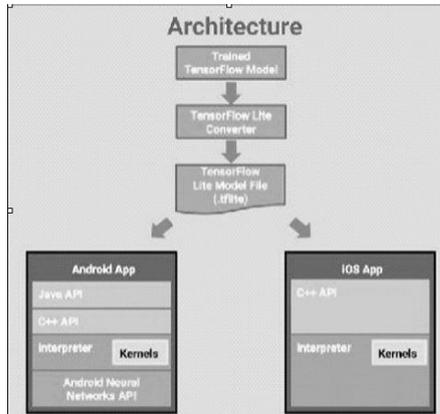


Figure 2

### 7. General Process Flow Image:-

a.

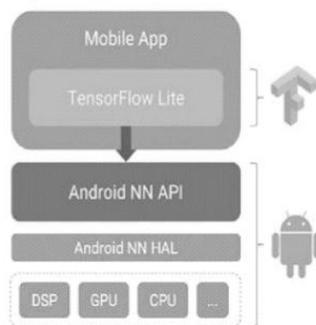


Figure 3

b.

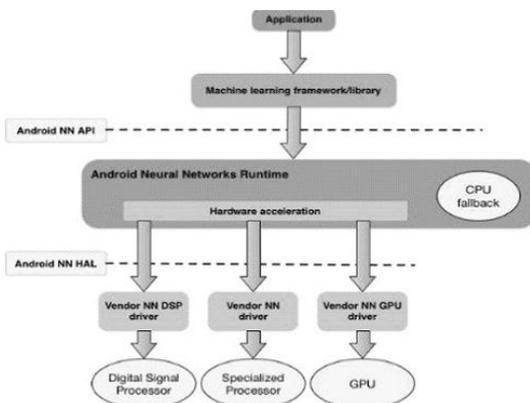


Figure 4

### 8. CONCLUSIONS

With the advent of deep network architectures that deals with Big Data, provides the best – in-class performance in terms of accuracy, scalability, adaptability without any

feature engineering. The project proposes the idea of detecting waste or trash using deep learning without involving manual work. Hence to implement it, the system works with a huge data set, pattern predictions for object detection and training algorithms. Further work includes, optimization of the results and prediction accuracies for various discrete inputs in real- time. Thus to ensure the good and accurate prediction the model which used to predict will be deployed in the server that helps in better portability and less cpu hungry application for the smartphone. And may be in future this project become an important asset to the society.

### REFERENCES

- [1] Hoornweg, Daniel, BhadaTata, et al. Environment: Waste production must peak this century, Nature, 502(7473), 615-7 (2013)
- [2] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, et al. Backpropagation applied to handwritten zip code recognition. Neural computation, vol.1,no.4, pp. 541-551 (1989).
- [3] Brinez L J C, Rengifo A, Escobar M. Automatic waste classification using computer vision as an application in colombian high schools, Networked and Electronic Media. IET, 10 (5)-10 (5) (2017)
- [4] Sudha S, Vidhyalakshmi M, Pavithra K, et al. An automatic classification method for environment: Friendly waste segregation using deep learning, Technological Innovations in ICT for Agriculture and Rural Development. IEEE, 65-70 (2016)
- [5] Erhan D, Szegedy C, Toshev A, et al. Scalable Object Detection Using Deep Neural Networks, 3(4), 21552162 (2013)
- [6] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation, In CVPR, pages 580-587, (2014)
- [7] Rad M S, Kaenel A V, Droux A, et al. A Computer Vision System to Localize and Classify Wastes on the Streets, International Conference on Computer Vision Systems. Springer, Cham, 195-204 (2017)