Project Name: Real Time Solid-Waste Identification

Table of Content

Demo

Refer to Video_File Folder.

Overview

This is a Real – Time Solid Waste Identification for detecting type of waste material.
This repository contains the code for Real Time Solid Waste Identification using python's various libraries.
It used Opencv, Tensorflow, Numpy, Pandas, Matplotlib, pickle, sklearn, xgboost, decision tree and random forest libraries.
These libraries help to perform individually one particular transformation.
Using Opencv, is a library of Python bindings designed to solve computer vision problems.
Numpy is used for working with arrays. It stands for Numerical Python.
Pandas objects rely heavily on Numpy objects.
Matplotlib is a plotting library.
The purpose of creating this repository is understand waste management system through machine learning rather than only theory.
These python libraries raised knowledge in discovering these libraries with practical use of it.
It leads to grow in my AI repository.
The video will help you to understand flow of output.

Motivation

The reason behind building Real Time Solid Waste Identification is that, in my country people are less aware about importance of separating trash which causes many diseases. I would like to take a step-in spreading awareness as well as most important is preventing illness with basic skills that every one can do. Also, if each one of us do our bit than those trash collector will save many hours of them which they spent in just separating it and as a whole nation we can concentrate on recycling waste and its benefits to reduce pollution as well as if recycling of trash done in correct manner then it will help many clothing or road construction companies to make use of it in making world a better place to live. It can lead to its purpose to provide hygienic and efficient place. It followed properly, can dramatically save fuel, labor and fleet maintenance cost. It also conserves energy. By building such kind of project, I understood how to work with images and different algorithms. Even I have started following it in my life, by changing few habits as carrying paper bags for shopping instead of plastic, keeping kitchen waste and general waste separate etc. These is not just a project to show to the companies but also it has made me learn many new things in Python Machine Learning Concepts and also to do a bit for society in whatever way possible for me.

Technical Aspect

Opencv makes use of Numpy. All the Opencv array structures are converted to and from Numpy arrays. It is used for all sorts of image and video analysis. In Opencv latest release is cv2 module. It is a cross-platform library using which we can develop real-time applications. cv2 means card verification value. Here, it is used to encode and decode secret messages inside an image file. Image files are basically a serialization of an image's pixels and RGB values. The file tells the computer which pixels to light up and with which color. When we use the cv2, function imread and pass it an image file, that image file is translated into a numpy array containing the RGB value for each pixel in the image.

Tensorflow is for fast numerical computing. It makes ML, DL faster and easier. It is free and opensource library. It has a particular focus on training and inference of deep neural networks. It allows developers to create large-scale neural networks with many layers and build models. It has faster compilation time than other DL libraries such as Keras and Touch.

Sklearn is known as scikit learn. It provides many ML libraries and algorithms for it. It provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python.

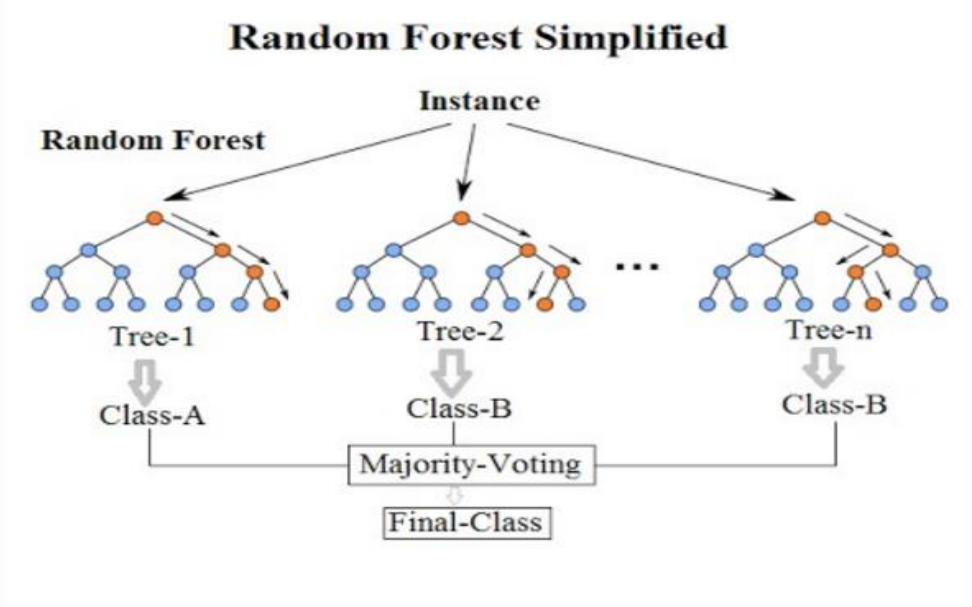
Numpy contains a multi-dimensional array and matrix data structures. It works with the numerical data. Numpy is faster because is densely packed in memory due to its homogeneous type. It also frees the memory faster.

Pandas module mainly works with the tabular data. It contains Data Frame and Series. Pandas is 18 to 20 times slower than Numpy. Pandas is seriously a game changer when it comes to cleaning, transforming, manipulating and analyzing data.

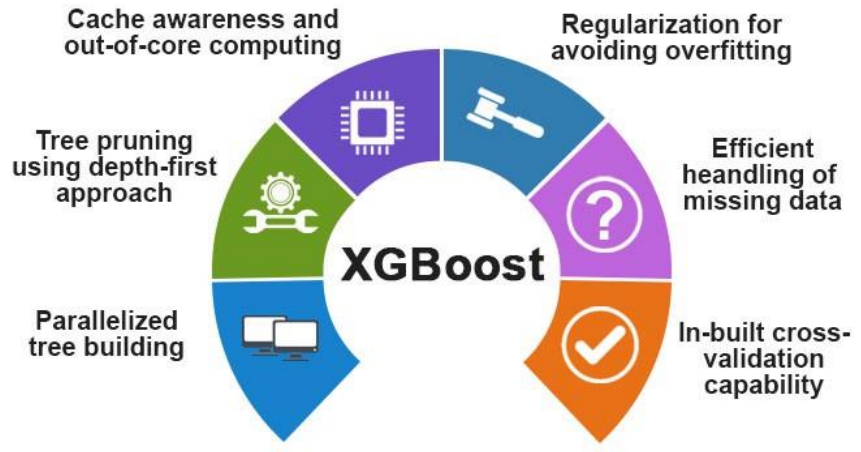
Xgboost is an implementation of gradient boosted decision trees designed for speed and performance. It is recently dominating applied ML industry. XGBoost uses more regularized model formalization to control over-fitting which gives it better performance than GBM.

Random forest is a supervised algorithm. It creates decision trees on randomly selected data samples, gets prediction from each tree and selects the best solution by means of voting.
Decision tree is a supervised algorithm, where data is continuously split according to certain parameter. Dtrees is a flow-chart like structure in which each internal node represents a test on a feature. It allows us to analyze fully possible consequences of a decision. Provide a framework to quantify the values of outcome and probabilities of achieving them.
Pickle module is used for serializing and de-serializing python object structures. The process to convert any kind of python objects (list, dict) into byte streams (0s and 1s) is called pickling or serialization or flattening or marshalling.
Matplotlib is used for EDA. Visualization of graphs helps to understand data in better way than numbers in table format. Matplotlib is mainly deployed for basic plotting. It consists of bars, pies, lines, scatter plots and so on. Inline command display visualization inline within frontends like in Jupyter Notebook, directly below the code cell that produced it.

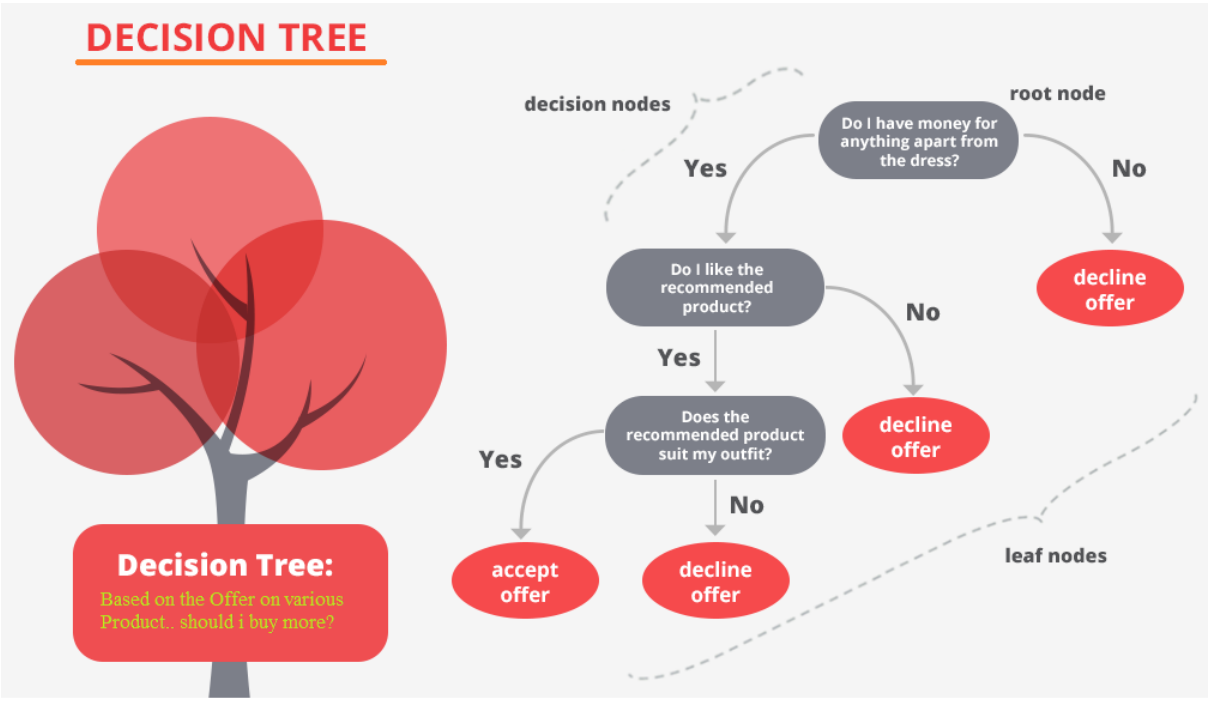
RF Architecture:



XGBoost Architecture:



Decision Tree Architecture:



Installation

Using intel core i5 9th generation with NVIDIA GFORECE GTX1650.
Windows 10 Environment Used.
Already Installed Anaconda Navigator for Python 3.x
The Code is written in Python 3.8.
If you don't have Python installed then please install Anaconda Navigator from its official site.
If you are using a lower version of Python you can upgrade using the pip package, ensuring you have the latest version of pip, *python* -m pip install --*upgrade pip and press Enter*.

Run/How to Use/Steps

Keep your internet connection on while running or accessing files and throughout too.
Follow this when you want to perform from scratch.

Open Anaconda Prompt, Perform the following steps:

Creating Virtual Environment named “SW”. You can give any name of your choice.

conda create -n SW python=3.6

y

conda activate SW
pip install tensorflow==2.2
pip install opencv-python
pip install xgboost
pip install sklearn

cd <PATH>

You can also create requirement.txt file as, pip freeze > requirements.txt

run files

conda deactivate

Creating Virtual Environment is necessary so that you do not have to install packages every-time you run the code. Once all required packages are installed in virtual environment then you only need to access/open the virtual environment and run the final file.

Follow this when you want to just perform on local machine.
Download ZIP File.
Right-Click on ZIP file in download section and select Extract file option, which will unzip file.
Move unzip folder to desired folder/location be it D drive or desktop etc.
Open Anaconda Prompt, write cd <PATH> and press Enter.
eg: cd C:\Users\Monica\Desktop\Projects\Python Projects 1\RealTime_Solid_Waste
Now, open virtual environment that you have created ie
conda activate SW
In Anconda Prompt, pip install -r requirements.txt to install all packages.
In Anaconda Prompt, write python <filename>.py and press Enter. That is,
In Anaconda Prompt, write python 1)Image_to_histogram.py and press Enter.
This creates hist.csv file inside Dataset folder.
Create model folder before running second file.
In Anaconda Prompt, write python 2)Classifier.py and press Enter.
It creates RFCmodel.sav and XGBmodel.sav file inside empty model folder.
In Anaconda Prompt, write python 3)RealTimePredictions.py and press Enter.
This will display camera screen and when you show waste material to it, it will display type of waste or name of it.
Create a folder name model inside your working folder to save results.
Please be careful with spellings or numbers while typing filename and easier is just copy filename and then run it to avoid any silly errors.
You can also run all codes from Command Prompt instead of Anaconda Prompt after setting Environmental Variable Path Settings.
Note: Please be careful with spellings or numbers while typing filename and easier is just copy filename and then run it to avoid any silly errors.
Note: cd <PATH>
[Go to Folder where file is. Select the path from top and right-click and select copy option and paste it next to cd one space <path> and press enter, then you can access all files of that folder] [cd means change directory]

Directory Tree/Structure of Project

Folder: RealTime_Solid_Waste
1)Image_to_histogram.py [5-6mins to run]
2)Classifier.py [30-45mins to run]
3)RealTimePredictions.py [5-10secs to run]

To Do/Future Scope

Can do for dry and wet waste.
Can use sensors in bins to measure fill levels and alert municipality or to check correct type of item dropped in.

Technologies Used/System Requirement/Tech Stack

Credits

keshavoct98