

# Rubbish-Classification

## Model Documentation

### 1 Dataset Labels

Images within the dataset were labelled using the following classifications:

0. Mixed Plastic
1. No Plastic
2. Plastic Bags
3. Plastic Bottles

Image names and their classification labels were stored within a .csv file named 'Images\_metatdata.csv' which uses the following structure:

| image_id     | image_id_root | label          |
|--------------|---------------|----------------|
| IMG_0409.JPG | IMG_0409      | No Plastic     |
| IMG_6949.JPG | IMG_6949      | Plastic Bottle |
| IMG_6950.JPG | IMG_6950      | No Plastic     |
| IMG_6951.JPG | IMG_6951      | Plastic Bags   |
| IMG_6952.JPG | IMG_6952      | Mixed Plastic  |

### 2 Folder structure for dataset

The folder structure used for loading this dataset is as follows:

---

```
dataset_root/  
  - Images/  
    - image001.JPG  
    - image002.JPG  
    - image003.JPG  
  - Images_metadata.csv
```

---

The following sections document the process for loading the data.

### 3 Required libraries

The following python code is used to load the required libraries:

```
%matplotlib inline
# python libraries
import os, cv2, itertools
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from tqdm import tqdm
from glob import glob
from PIL import Image
# pytorch libraries
import torch
from torch import optim, nn
from torch.autograd import Variable
from torch.utils.data import DataLoader, Dataset
from torchvision import models, transforms
# sklearn libraries
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
# ensure results are reproducible
np.random.seed(10)
torch.manual_seed(10)
torch.cuda.manual_seed(10)
```

### 4 Loading the dataset

The following python code is used to load the dataset:

```
data_dir = '../dataset_root'
all_image_path = glob(os.path.join(data_dir, '*', '*.JPG'))
imageid_path_dict = {os.path.splitext(os.path.basename(x))[0]: x for x in all_image_path}
label_type_dict = {
    'No Plastic': 'No Plastic',
    'Mixed Plastic': 'Mixed Plastic',
    'Plastic Bottle': 'Plastic Bottle',
    'Plastic Bags': 'Plastic Bags'
}

df_original = pd.read_csv(os.path.join(data_dir, 'Images_metadata.csv'))
df_original['path'] = df_original['image_id_root'].map(imageid_path_dict.get)
df_original['label'] = df_original['label'].map(label_type_dict.get)
df_original['label_num'] = pd.Categorical(df_original['label']).codes
df_original.head()
```

The dataset is now loaded and the resulting output should be similar to the following:

|   | image_id     | image_id_root | label      | path                                | label_num |
|---|--------------|---------------|------------|-------------------------------------|-----------|
| 0 | IMG_0409.JPG | IMG_0409      | No Plastic | ../dataset_root/Images/IMG_0409.JPG | 1         |
| 1 | IMG_0422.JPG | IMG_0422      | No Plastic | ../dataset_root/Images/IMG_0422.JPG | 1         |
| 2 | IMG_0440.JPG | IMG_0440      | No Plastic | ../dataset_root/Images/IMG_0440.JPG | 1         |
| 3 | IMG_0449.JPG | IMG_0449      | No Plastic | ../dataset_root/Images/IMG_0449.JPG | 1         |
| 4 | IMG_0451.JPG | IMG_0451      | No Plastic | ../dataset_root/Images/IMG_0451.JPG | 1         |

## 5 Loading the model

The inference script (Jupyter notebook) that is used to load the data and load the model is provided and is titled:

[Model\\_Data\\_Loader.ipynb](#)

This notebook can be used to load the trained model which is titled:

[Model\\_Trained.pt](#)

## 6 Reporting predictions on test dataset

Once the dataset and model have been loaded (using [Model\\_Data\\_Loader.ipynb](#)) and the trained model is loaded (using [Model\\_Trained.pt](#)) the predictions may be reported using the following methods:

1. Confusion matrix
2. Classification report
3. Classification label error plot