# *Waste Industry: Innovations and Business Opportunities*

**Krishan Singh**

(Date: 15-05-2022)

## 1. *Problem Statement*

Every day, more than 28k tons of single-use plastic gets dumped into the world's ocean. Over 86 dump trucks full of discarded and used clothing are either dropped in a landfill or burdened. And over 109k tons of used electronics, a weight equivalent to more than 68m laptops, are discarded.

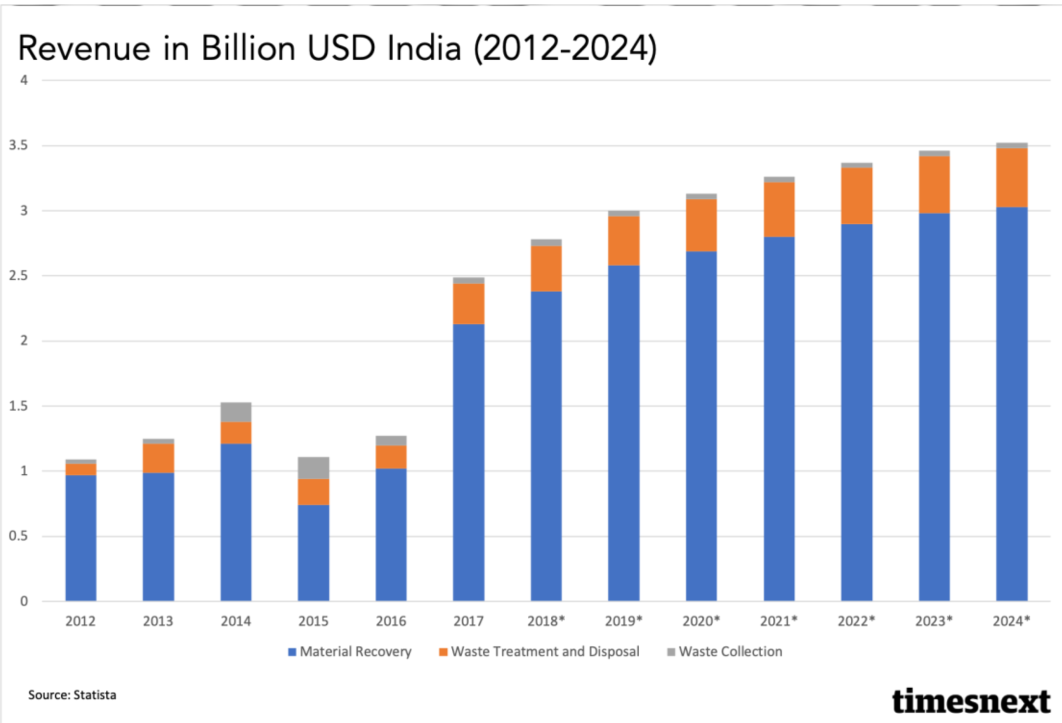
India accounts for about 18% of the world's population and 12% of global municipal waste generation. The country's population is continuously rising, and there are expectations that the waste generation in India would also see significant growth in its waste generation in the upcoming decades. Consequently, it creates a challenge in its management.

If these dumps are left unchecked, it will spell an environmental disaster. However, it can also spell big business opportunities.

## 2. *Market Need Assessment*

According to Statista, the 'Waste collection, treatment, and disposal activities, materials recovery' industry in India would amount to approximately 3.5 billion USD revenue by 2024.

**Revenue of 'Waste collection, treatment and disposal activities' Industry in India from 2012 to 2024.**



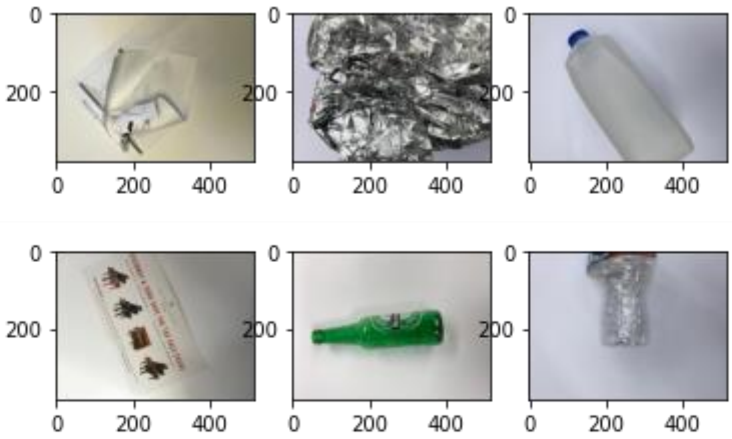

There is a big potential for companies in e-waste recycling and other waste management areas. The industry is generating 6.5 billion USD yearly. However, only 32% of recyclables are claimed and processed, leaving plenty of untapped opportunities. The business's key lies in finding out how to unlock previously untapped supply, processing it, and reselling it in some form.

## 3. ***Target Specifications and Characterization***

Everyone on this planet has to deal with trash every day and they end up in various undesirable places as mentioned in the problem statement to harm our environment, eventually harming us. Here the report aims to present a business model to formulate the solution to the problem that nobody can't get rid of, and also make it profitable as a business, in the process.

## 4. ***External References***

The dataset is taken from [Kaggle.com](Kaggle.com) contains 6 classifications: cardboard (393), glass (491), metal (400), paper(584), plastic (472) and trash(127). Here is the sample dataset:



## 5. ***Benchmarking***

There is no perfect model to segregate the garbage in the specified categories as technology evolves and norms change the materials used in products also change, so one must actively modify the model timely.
In this report, I have used transfer learning using the VGG-16 model on the specified dataset.

## 6. ***Applicable Patents***

There are many patents filed by companies around the world on various techniques to recycle/reuse/treat the waste generated by various industries some include:

- Landfill garbage management process **Patent number**: 6106197
- Method for treating waste printed circuit boards with molten mixture of inorganic salts **Patent number**: 6089479
- Shredder dust treatment process **Patent number**: 6086000

## 7. *Applicable Regulations*

Enacted in 1986, the Environment Protection Act aims to establish a good protection system for the environment. It gives the power to the central government to regulate all forms of waste and to tackle specific problems that may present themselves in different regions of India.

**Hazardous Wastes Management Regulations**

- Hazardous Wastes (Management, Handling and Transboundary) Rules, 2008, brought out a guide for the manufacture, storage, and import of hazardous chemicals and management of hazardous wastes.
- Biomedical Waste (Management and Handling) Rules, 1998, were formulated along parallel lines, for proper disposal, segregation, transport, etc, of infectious wastes.
- Municipal Solid Wastes (Management and Handling) Rules, 2000, aim at enabling municipalities to dispose of municipal solid waste scientifically.
- E-Waste (Management and Handling) Rules, 2011 have been notified on May 1, 2011, and came into effect on May 1, 2012, with the primary objective to reduce the use of hazardous substances in electrical and electronic equipment by specifying the threshold for use of hazardous material and to channelize the e-waste generated in the country for environmentally sound recycling. The Rules apply to every producer, consumer or bulk consumer, collection center, dismantler, and recycler of e-waste involved in the manufacture, sale, purchase, and processing of electrical and electronic equipment or components as detailed in the Rules.
- Batteries (Management & Handling) Rules, 2001 deal with the proper and effective management and handling of lead-acid battery waste. The Act requires all manufacturers, assemblers, re-conditioners, importers, dealers, auctioneers, bulk consumers, and consumers, involved in the manufacture, processing, sale, purchase, and use of batteries or components thereof, to comply with the provisions of Batteries (Management & Handling) Rules, 2001.

## 8. *Applicable Constraints*

1. Deep Learning Expert: The role of a deep learning expert is to process the data and perform data augmentation, labeling unlabeled images, image preprocessing, etc., and also formulate and train models

2. Software Developer: To get the model into production there are mobile and web platforms. A software developer handles those bug issues etc.

3. Law Advisor: Government keeps introducing new regulations or modifying them, therefore an expert law advisor is needed to ensure that the company does not violate any laws or regulations.

## 9. *Business Opportunity*

As previously mentioned, according to Statista, the ‘Waste collection, treatment, and disposal activities, materials recovery’ industry in India would amount to approximately 3.5 billion USD revenue by 2024.

Giants such as Waste Management, 15 billion USD per year, or Republic Services, 10 billion USD per year, vertically integrated these processes. However, half of the United States’ waste industry is fragmented, with more than 20,000 small businesses at the local level. China dominated the scene starting in 1990, importing over 70% of the world’s trash, almost 7 million tons every year. The business was lucrative, and Zhang Yin, China’s first female billionaire, made her fortune from recycling paper scraps from the United States. In 2018, China and other countries in Asia banned almost all incoming waste over environmental and health concerns stemming from poor disposal and hard-to-recycled materials.

Every day, more than 28k single-use plastic tons end up in oceans, not just in the form of water bottles but also toothpaste tubes, packaging and shipping materials, grocery bags, and more.
New entrepreneurs are emerging to clean up the world's oceans and landfills. They are using unique business models to unlock new supply chains and turn trash into treasure in the process.

## *10. <u>Concept Generation</u>*

The business of waste management makes money in three primary ways: Waste collection, recycling, and end of life. About 55% of waste management revenue comes from fees for waste collection from residential, small containers, and glass container services for homes and businesses. Facilities sell recyclable materials such as paper, glass, and metals to manufacturers. And finally, materials that don't get recycled or reused get treated and disposed of in a landfill or waste-to-energy plants. Landfills collect tipping payments from businesses, governments, and individuals, while energy plants sell energy directly to consumers or other plants.

## *11. <u>Concept Development</u>*

So, to build a classifier that correctly sorts out different types of waste, we will need a model architecture denominated as Convolutional Neural Network since our dataset will be composed mainly of labeled images, meaning that each image has a corresponding label that indicates the correct prediction (type of waste material) the model will have to provide as output. At the same time, the model will also need a fully-connected network after the convolutional module to transform an arbitrary response given by it into a set of values with a particular structure that will allow us to determine the class predicted by the model.

The aim of using a Convolutional Network to process an image resides in its ability to extract certain patterns or features from images with an invariance in position, rotation, and scale. To understand the power of these properties when detecting features in images, let's consider an example.

In a nutshell, the Convolution operation takes as input the pixels of an image and outputs a feature map, which is no more than a matrix of values that can perfectly represent an image.

To build the Convolutional Neural Network programmatically, we will be using Python code running on the web platform Google Colab, which provides a fast, comfortable, and robust cloud environment for training such large models. Furthermore, the Python programming language has a wide variety of libraries oriented to constructing machine learning algorithms that make training and evaluation much more accessible. Such libraries are Tensorflow, Keras, Numpy, Matplotlib, etc.

## *12. <u>Final Product Prototype</u>*

Back End

i. In this area mainly the Data Engineer, and Deep Learning Engineer work.
ii. Collect and preprocess images.
iii. The previous model should be kept updated with the new upcoming dataset.

Front End

i. In this area mainly the Software developer and UI/UX teamwork.
ii. The front end should be updated along with the backend.
iii. A software developer is responsible for updating the website/app which runs the model.

## 13. Product details

***How does it work?***
The first step, data input, differs between Machines and humans. In the case of humans, they struggle to handle and understand large datasets and require a lot of time for this to happen, whereas machines understand pretty quickly and require very less amount of time to learn. The more data an ML model receives, the better it can learn its fraud detection skills.

***Data collection and processing***
Before building, training, and evaluating our model, we must gather a dataset of labeled waste images. Afterward, we have to split our data into two datasets (train and test). Each of them is only used in the corresponding phase of the project (training and evaluation of the model). This operation is critical to making the model capable of generalizing from the provided data to any input that the user will use in production.

***Model building and training***
As we are dealing with a relatively large dataset, it's convenient to use a common technique when training a model on such an amount of data called *Transfer Learning*. That refers to replacing the convolutional part of your model with an already trained one. So before training, your model will be able to extract useful features from the input images thanks to the trained convolutional part. Moreover, it will only have to train the last few dense layers, reducing the computing power and time involved. There are a lot of trained convolutional models available, but the most common ones are included in the Keras API that we are currently using for the project.

***Model evaluation***
The training time it takes depends on the hardware, model complexity, and dataset size. But in this case, with a Google Colab environment equipped with a Tesla P100 GPU, a dataset size of 2500 images approximately and a model with around 15,246,150 parameters, only around half a million of them are trainable.

***Applications***
Having seen the facilities that modern libraries like Tensorflow provide for building such complex machine learning algorithms, it's convenient to analyze the real benefits of this model when dealing with environmental threats. Additionally, its results in the evaluation phase and the generalization capability make it robust enough to be scalable to a larger dataset.

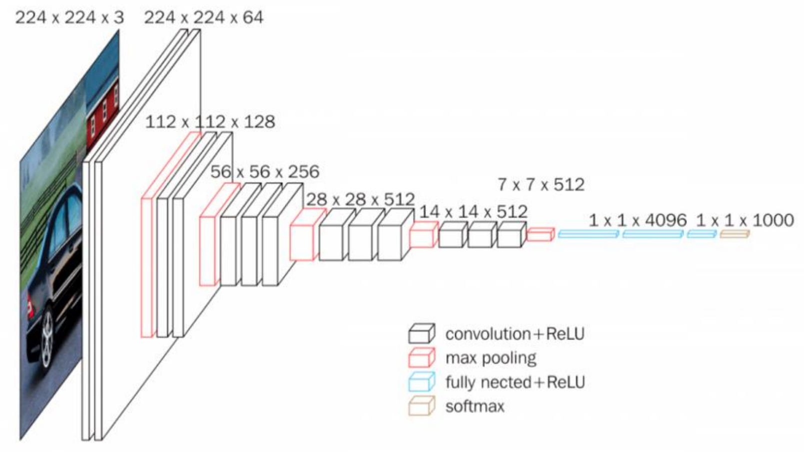
The model can be extended to build on an extensive dataset and scale it over time. Also, the collaboration of many persons is even more necessary to gather and interpret the data humanly.

***Data Source***
The dataset is taken from Kaggle.com as mentioned before, the dataset contains 6 classifications: cardboard (393), glass (491), metal (400), paper(584), plastic (472), and trash(127).

***Architecture***

<u>VGGNet</u>

VGGNet was born out of the need to reduce the number of parameters in the CONV layers and improve training time. There are multiple variants of VGGNet (VGG16, VGG19, etc.) which differ only in the total number of layers in the network. The structural details of a VGG16 network that has been used in the code have been shown below.




| VGG16 - Structural Details | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **#** | **Input Image** | | | **output** | | | **Layer** | **Stride** | **Kernel** | | **in** | **out** | **Param** |
| **1** | 224 | 224 | 3 | 224 | 224 | 64 | conv3-64 | 1 | 3 | 3 | 3 | 64 | 1792 |
| **2** | 224 | 224 | 64 | 224 | 224 | 64 | conv3064 | 1 | 3 | 3 | 64 | 64 | 36928 |
| | 224 | 224 | 64 | 112 | 112 | 64 | maxpool | 2 | 2 | 2 | 64 | 64 | 0 |
| **3** | 112 | 112 | 64 | 112 | 112 | 128 | conv3-128 | 1 | 3 | 3 | 64 | 128 | 73856 |
| **4** | 112 | 112 | 128 | 112 | 112 | 128 | conv3-128 | 1 | 3 | 3 | 128 | 128 | 147584 |
| | 112 | 112 | 128 | 56 | 56 | 128 | maxpool | 2 | 2 | 2 | 128 | 128 | 65664 |
| **5** | 56 | 56 | 128 | 56 | 56 | 256 | conv3-256 | 1 | 3 | 3 | 128 | 256 | 295168 |
| **6** | 56 | 56 | 256 | 56 | 56 | 256 | conv3-256 | 1 | 3 | 3 | 256 | 256 | 590080 |
| **7** | 56 | 56 | 256 | 56 | 56 | 256 | conv3-256 | 1 | 3 | 3 | 256 | 256 | 590080 |
| | 56 | 56 | 256 | 28 | 28 | 256 | maxpool | 2 | 2 | 2 | 256 | 256 | 0 |
| **8** | 28 | 28 | 256 | 28 | 28 | 512 | conv3-512 | 1 | 3 | 3 | 256 | 512 | 1180160 |
| **9** | 28 | 28 | 512 | 28 | 28 | 512 | conv3-512 | 1 | 3 | 3 | 512 | 512 | 2359808 |
| **10** | 28 | 28 | 512 | 28 | 28 | 512 | conv3-512 | 1 | 3 | 3 | 512 | 512 | 2359808 |
| | 28 | 28 | 512 | 14 | 14 | 512 | maxpool | 2 | 2 | 2 | 512 | 512 | 0 |
| **11** | 14 | 14 | 512 | 14 | 14 | 512 | conv3-512 | 1 | 3 | 3 | 512 | 512 | 2359808 |
| **12** | 14 | 14 | 512 | 14 | 14 | 512 | conv3-512 | 1 | 3 | 3 | 512 | 512 | 2359808 |
| **13** | 14 | 14 | 512 | 14 | 14 | 512 | conv3-512 | 1 | 3 | 3 | 512 | 512 | 2359808 |
| | 14 | 14 | 512 | 7 | 7 | 512 | maxpool | 2 | 2 | 2 | 512 | 512 | 0 |
| **14** | 1 | 1 | 25088 | 1 | 1 | 4096 | fc | | 1 | 1 | 25088 | 4096 | 102764544 |
| **15** | 1 | 1 | 4096 | 1 | 1 | 4096 | fc | | 1 | 1 | 4096 | 4096 | 16781312 |
| **16** | 1 | 1 | 4096 | 1 | 1 | 1000 | fc | | 1 | 1 | 4096 | 1000 | 4097000 |
| | | | | | | | **Total** | | | | | | **138,423,208** |

VGG16 has a total of 138 million parameters. The important point to note here is that all the conv kernels are of size 3x3 and ma pool kernels are of size 2x2 with a stride of two.

***Team required to develop***

- Deep Learning Engineer
- Software Developer
- Law Advisor

## *14. Code Implementation*

```
#Loading and unzipping dataset
!wget https://github.com/blaze-fire/feynn-labs-project-1/blob/main/archive.zip?raw=true
!unzip /content/archive.zip?raw=true
```

Importing the required Modules and libraries required

```
[ ] import numpy as np
    import pandas as pd
    import matplotlib.pyplot as plt
    import re
    import os
    import random
    from PIL import Image
    import tensorflow as tf
    from tensorflow.keras.preprocessing.image import ImageDataGenerator,load_img, img_to_array
    from sklearn.metrics import accuracy_score, confusion_matrix, ConfusionMatrixDisplay
```

Importing the dataset from the local machine using pandas and making a data frame.

```
# Add class name prefix to each path based on class name include in filename
def add_class_name_prefix(df, col_name):
    df[col_name] = df[col_name].apply(lambda x: x[:re.search("\d",x).start()] + '/' + x)
    return df


def class_id_to_label(id):
    label_map = {1: 'glass', 2: 'paper', 3: 'cardboard', 4: 'plastic', 5: 'metal', 6: 'trash'}
    return label_map[id]


IMAGES_DIR = '/content/Garbage classification/Garbage classification'

train_file = '/content/one-indexed-files-notrash_train.txt'
val_file   = '/content/one-indexed-files-notrash_val.txt'
test_file  = '/content/one-indexed-files-notrash_test.txt'

df_train = pd.read_csv(train_file, sep=' ', header=None, names=['path', 'label'])
df_valid = pd.read_csv(val_file,   sep=' ', header=None, names=['path', 'label'])
df_test  = pd.read_csv(val_file,   sep=' ', header=None, names=['path', 'label'])

df_train = add_class_name_prefix(df_train, 'path')
df_valid = add_class_name_prefix(df_valid, 'path')
df_test  = add_class_name_prefix(df_test,  'path')

df_train['label'] = df_train['label'].apply(class_id_to_label)
df_valid['label'] = df_valid['label'].apply(class_id_to_label)
df_test['label']  = df_test['label'].apply(class_id_to_label)

print(f'Found {len(df_train)} training, {len(df_valid)} validation and {len(df_test)} samples.')
```

View a sample image of each class

```
import glob, os, random
img_list = glob.glob(os.path.join(IMAGES_DIR, '*/*.jpg'))

for i, img_path in enumerate(random.sample(img_list, 6)):
    img = load_img(img_path)
    img = img_to_array(img, dtype=np.uint8)

    plt.subplot(2, 3, i+1)
    plt.imshow(img.squeeze())
```

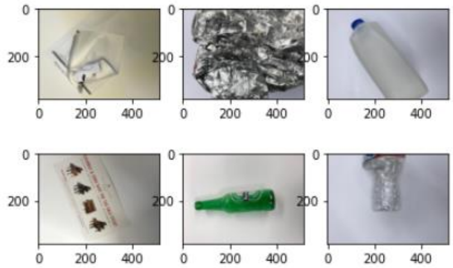


Image Preprocessing

```
# ImageDataGenerator: For data preprocessing
# flow_from_dataframe: Takes the Pandas DataFrame and the path to a directory and generates batches of augmented/normalized data

gen = ImageDataGenerator(rescale=1./255)    # rescaling the images between 0 and 1

gen_train = gen.flow_from_dataframe(
    dataframe=df_train,
    directory=IMAGES_DIR,
    x_col='path',
    y_col='label',
    color_mode="rgb",
    class_mode="categorical",
    batch_size=32,
    shuffle=True
)

gen_valid = gen.flow_from_dataframe(
    dataframe=df_valid,
    directory=IMAGES_DIR,
    x_col='path',
    y_col='label',
    color_mode="rgb",
    class_mode="categorical",
    batch_size=32,
    shuffle=True
)

test_gen = gen.flow_from_dataframe(
    dataframe=df_test,
    directory=IMAGES_DIR,
    x_col='path',
    y_col='label',
    color_mode="rgb",
    class_mode="categorical",
    batch_size=32,
    shuffle=False
)
```

```
Found 1768 validated image filenames belonging to 6 classes.
Found 328 validated image filenames belonging to 6 classes.
Found 328 validated image filenames belonging to 6 classes.
```

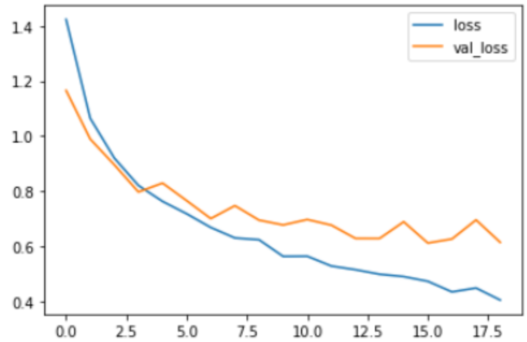
Model

```python
# Function defining model architecture

def build_model(num_classes):
    # Loading pre-trained ResNet model
    base_model = tf.keras.applications.VGG16(weights='imagenet', include_top=False)

    x = base_model.output
    x = tf.keras.layers.GlobalAveragePooling2D()(x)
    x = tf.keras.layers.Dense(1024, activation='relu')(x)
    predictions = tf.keras.layers.Dense(num_classes, activation='softmax')(x)

    model = tf.keras.Model(inputs=base_model.input, outputs=predictions)

    base_model.trainable = False

    return model

model = build_model(num_classes=6)

model.compile(optimizer='Adam',
              loss='categorical_crossentropy',
              metrics=[tf.keras.metrics.categorical_accuracy])

model.summary()
```

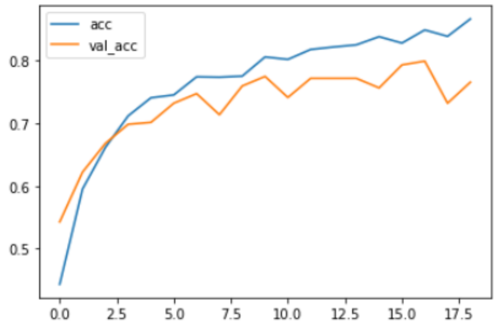
Model Training

```python
callback = tf.keras.callbacks.EarlyStopping(
    monitor="val_loss",
    patience=3,
    mode="auto",
    restore_best_weights=True,
)
# Model training

history = model.fit(
    gen_train,
    validation_data=gen_valid,
    callbacks=[callback],
    epochs=50
)
```

```python
import matplotlib.pyplot as plt

plt.plot(history.history['loss'], label='loss')
plt.plot(history.history['val_loss'], label='val_loss')
plt.legend()
plt.show()
```



```
[ ] plt.plot(history.history['categorical_accuracy'], label='acc')
    plt.plot(history.history['val_categorical_accuracy'], label='val_acc')
    plt.legend()
    plt.show()
```

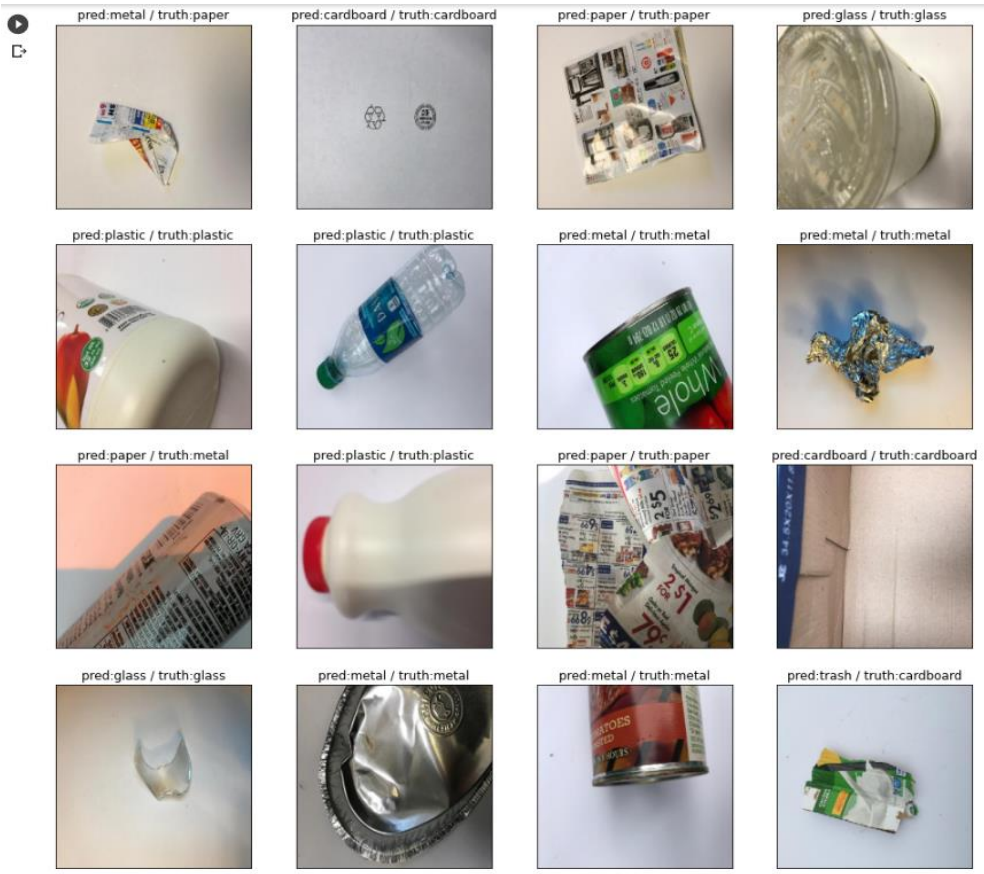



Evaluating the model

```
# Evaluating the model on test data

filenames = test_gen.filenames
nb_samples = len(filenames)

model.evaluate_generator(test_gen, nb_samples)
```

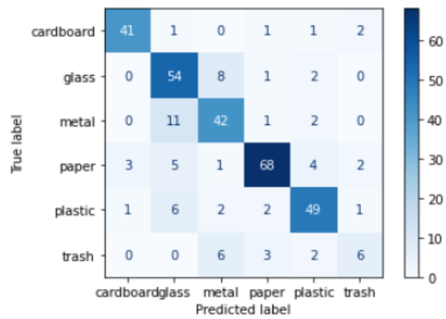
```
[ ] # Generating predictions on test data

    test_x, test_y = test_gen.__getitem__(1)
    preds = model.predict(test_x)
```

```
[ ] # Comparing predcitons with original labels

    labels = (gen_train.class_indices)
    labels = dict((v,k) for k,v in labels.items())
    plt.figure(figsize=(16, 16))
    for i in range(16):
        plt.subplot(4, 4, i+1)
        plt.xticks([])
        plt.yticks([])
        plt.grid(False)
        plt.title('pred:%s / truth:%s' % (labels[np.argmax(preds[i])], labels[np.argmax(test_y[i])]))
        plt.imshow(test_x[i])
```




```
# Confusion Matrix

y_pred = model.predict(test_gen)
y_pred = np.argmax(y_pred, axis=1)
print('Confusion Matrix')
cm = confusion_matrix(test_gen.classes, y_pred)
labels = ['cardboard', 'glass', 'metal', 'paper', 'plastic', 'trash']
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=labels)
disp.plot(cmap=plt.cm.Blues)
plt.show()
```

```
Confusion Matrix
```




```
# Accuracy

acc = accuracy_score(test_gen.classes, y_pred)
print("Accuracy is {} percent".format(round(acc*100,2)))
```

```
Accuracy is 79.27 percent
```

[GitHub - Code]

## 15. Conclusion

The Deep Learning model detects the type of garbage and segregates it accordingly. Accurate segregation of other types of garbage can be done by using the model and training it with the new images. As the economy of the country grows the waste generated per person is also going to increase which will worsen the waste management issues that the country faces now in various parts of India. I believe it's high time to act to save the environment, and adopting this business model deals with environmental issues simultaneously generating massive revenues for the company.

## References


- https://waste-management-world.com/recycling/recycling-in-india-a-market-in-transition/
- https://cpcb.nic.in/uploads/plasticwaste/Annual_Report_2019-20_PWM.pdf
- http://www.indiaenvironmentportal.org.in/files/file/e-waste-management-NGT-CPCB-report.pdf
- https://www.kaggle.com/datasets/asdasdasasdas/garbage-classification