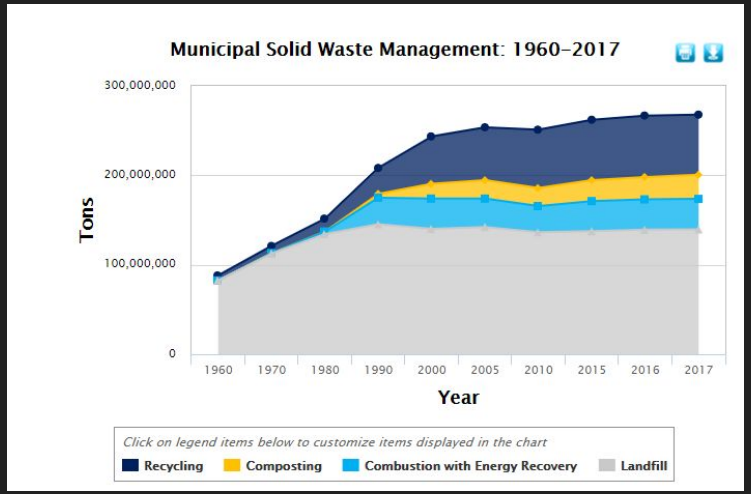
Waste Classifier Network

By: Carlos Prieto and Alexander Haynie

https://github.com/charlypri/MLfinalProject

The Problem

- Every year we dump 2,120,000,000 tons of waste on the planet.
- Types:
  - Construction waste
  - Industrial waste
  - Plastic waste
  - Household waste
  - Electronic waste
  - And many more

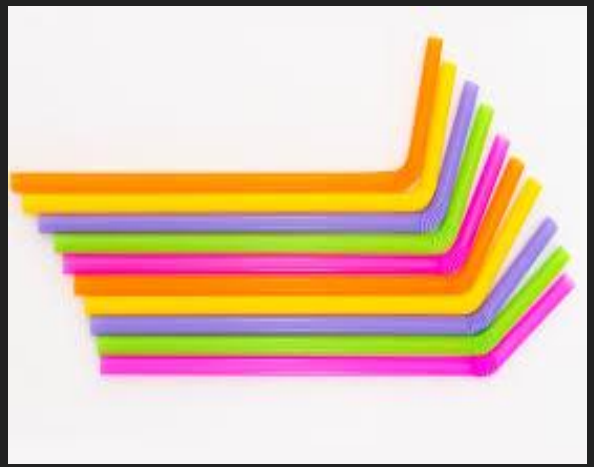


- Roughly 80% of items buried in landfills could be recycled
- 1 out of every 5 tons being thrown into landfills is organic material

Goal

The main goal of this project is to generate an image classifier that estimates whether a picture depicts a recyclable object or an organic one.

VS



Data

- Training Dataset
  - 12,567 “organic” images
  - 10,000 “recyclable” images
- Testing Dataset
  - 1401 “organic” images
  - 1112 “recyclable” images

Importing The Data

```
train_image = []
for i in tqdm(range(10000)):
    img = image.load_img('DATASET/TRAIN/O/O_'+str(i+1)+'.jpg', target_size=(50, 50, 3), grayscale=False)
    img = image.img_to_array(img)
    train_image.append(img)
for i in tqdm(range(10000)):
    img = image.load_img('DATASET/TRAIN/R/R_'+str(i+1)+'.jpg', target_size=(50, 50, 3), grayscale=False)
    img = image.img_to_array(img)
    train_image.append(img)
```

```
100%|██████████| 10000/10000 [00:15<00:00, 660.67it/s]
100%|██████████| 10000/10000 [00:13<00:00, 751.99it/s]
```

Neural Network

- We built a multilayer perceptron, Dense: Fully connected layer and the most common type of layer used on multi-layer perceptron models.
- Keras Conv2D is a 2D Convolution Layer, this layer creates a convolution kernel that is wind with layers input which helps produce a tensor of outputs.

Neural Network

- Max pooling is a sample-based discretization process. The objective is to down-sample an input representation (image, hidden-layer output matrix, etc.), reducing its dimensionality and allowing for assumptions to be made about features contained in the sub-regions binned.
- Dropout: Apply dropout to the model, setting a fraction of inputs to zero in an effort to reduce over-fitting.
- Flattening a tensor means to remove all of the dimensions except for one. This is exactly what the Flatten layer do.

Creating the Model

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),activation='relu',input_shape=(50, 50, 3)))#242, 208, 3 is original shape
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(2, activation='softmax'))
model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])
```

Results

- Used an image size of 50x50 pixels (significantly smaller than the actual image size)
- Fit the model with 5 epochs took over an hour
- Our code successfully identified the image being organic or recycling 87.4% of the time.
- If given a computer with better processing power we would be able to increase the accuracy.

```
ctr =0
for i in range(len(X_test)):
    if int(results['y_real'][i]) == (results['prediction'][i]) :
        ctr = ctr +1
print("The accuracy on the Test set is ", "{0:.2f}".format(ctr/len(X_test)*100) ,"%")
```

```
The accuracy on the Test set is  87.40 %
```

Conclusions

- Our method of image identification could be used to help aid the fight in waste management.
- Being able to quickly identify recyclable material with an 85% accuracy, we would be able to provide a cost effective aid to waste identification.