Waste Image Classification using Convolutional Neural Networks

FINAL CAPSTONE REPORT

Navdeep Raj | Data Science | 14 Aug 2021

ABSTRACT

In today's fast-growing economy, rapid advancement and industrialization have led to an enormous generation of waste which has become a matter of great concern resulting in environmental pollution and hazardous health problems. It is important to have an advanced waste filtration management system which can separate waste into different categories for better recycling and environmental cleaning.

Objective of this project is to establish a smart waste classification system which can separate waste from images using CNNs. This project uses TrashNet dataset for generating CNN models with six classes to classify the waste into. In this project various architectures like VGG, DenseNet, MobileNet, Inception, etc. have been used to train the data and achieve accuracy. All models used Softmax as the classifier as the last layer and highest accuracy achieved was 79%

BACKGROUND & BUSINESS OBJECTIVE

Garbage accumulation has been a long part of our civilization which has increased manifolds as the time progressed. Especially in today's current scenario waste collection is humongous in almost every corner of the world thus its high time to implement some atomization when it comes to managing waste so we can curb the pollutions of all kind and create a better and safer place for future generations. In earlier times, this was done manually where waste was segregated and burned up which led to toxic surroundings for the people but with the modern technology, we have been able to develop advanced recycling processes depending on the waste such as plastic, glass, metal, rubber, etc.

Key issue here is that if the waste is not properly separated it leads to ineffective recycling resulting in waste of time and resources.

The primary objectives of this project were

- Data pre-processing of trash images and create augmented data to obtain high classification accuracy.
- To train data on various architectures like VGG-16, ResNet-50, DenseNet, Inception, Custom layering, MobileNet, etc.
- Evaluate models majorly on accuracy metrics and class prediction probability.

DATASET

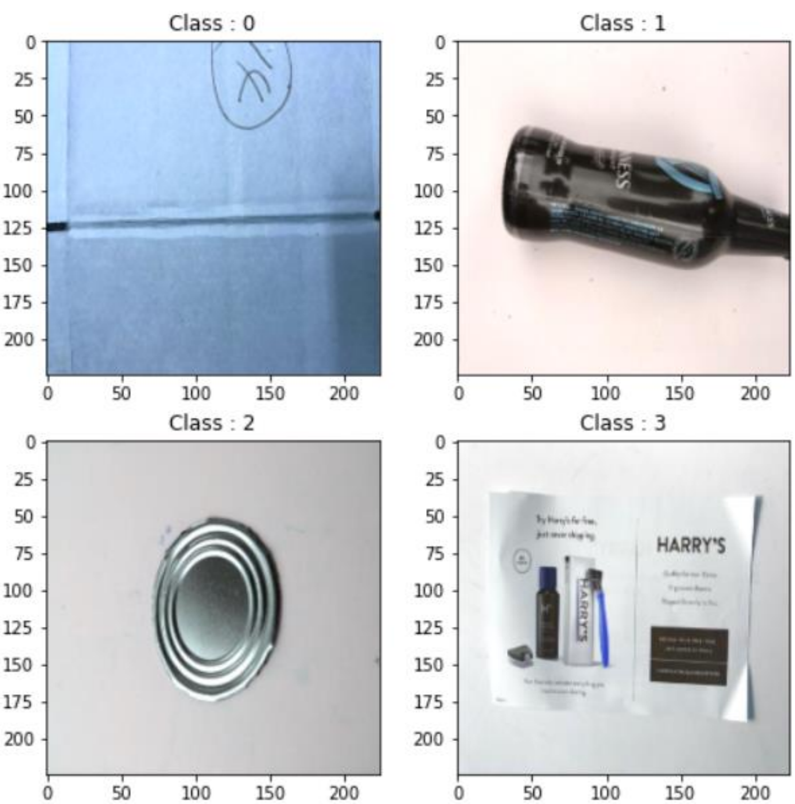
I have used the dataset that Thung and Young created. This dataset spans six categories and consists of 2527 images in total. Detailed number of images for each class can be found in the table below.

The pictures were all taken by Apple Iphone. We used approximately 75% from each category as training data, and 25% as development data. Therefore, our training set had 1894 images in total, and our development set had 633 images in total.

Detailed number of images for training set and development set can also be found in the table below. Class Type Number of Images in Training Set Number of Images in Development Set

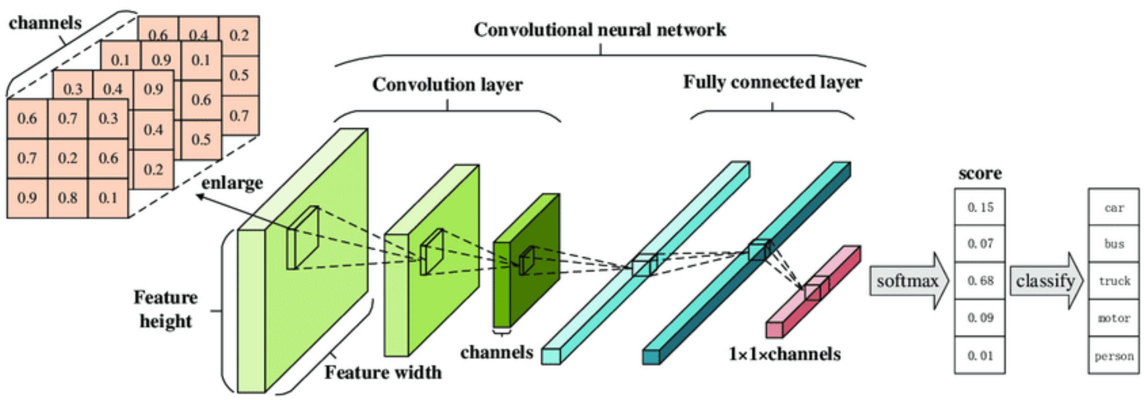
- Cardboard – Training – 302, Test – 101, Total – 403
- Glass - Training – 376, Test – 125, Total – 501
- Metal - Training – 307, Test – 103, Total – 410
- Paper - Training – 445, Test – 149, Total – 594
- Plastic - Training – 361, Test – 121, Total – 482
- Trash - Training – 103, Test – 34, Total – 137
- Total - Training – 1894, Test – 633, Total – 2527

All images were resized to 227x227 to best fit with our models. We incorporated data augmentation techniques including vertical flip, horizontal flip, width shift, height shift, zooming and rotation.



APPROACH - IMAGE CLASSIFICATION USING CNN & TRANSFER LEARNING

An Introduction to CNN - A Convolutional neural network (CNN) is a neural network that has one or more convolutional layers and are used mainly for image processing, classification, segmentation and also for other auto correlated data. Architecture of a Convolutional Neural Network (CNN). The traditional CNN structure is mainly composed of convolution layers, pooling layers, fully connected layers, and some activation functions. Each convolution kernel is connected to the part of feature maps. The input is connected to all of the output elements in the fully connected layer.



Source: https://www.researchgate.net/figure/Architecture-of-a-Convolutional-Neural-Network-CNN-The-traditional-CNN-structure-is_fig1_330106889

TRANSFER LEARNING

Deep learning has made considerable progress in recent years. This has enabled us to tackle complex problems and yield amazing results. However, the training time and the amount of data required for such deep learning systems are much more than that of traditional ML systems. There are various deep learning networks with state-of-the-art performance (sometimes as good or even better than human performance) that have been developed and tested across domains such as computer vision and natural language processing (NLP). These pre-trained networks/models form the basis of transfer learning.

Deep learning systems and models are layered architectures that learn different features at different layers (hierarchical representations of layered features). These layers are then finally connected to a last layer (usually a fully connected layer, in the case of supervised learning) to get the final output. This layered architecture allows us to utilize a pre-trained network (such as Inception V3 or VGG) without its final layer as a fixed feature extractor for other tasks. The key idea here is to just leverage the pre-trained model's weighted layers to extract features but not to update the weights of the model's layers during training with new data for the new task.

Transfer Models used in this project with pretrained weights from imagenet dataset:-

- MobileNetV2
- InceptionV3
- DenseNet121
- VGG16
- Xception
- ResNet-50

METHODOLOGY

This project employs the use of deep learning and transfer learning models for the classification and prediction of trash images. Firstly, Data preprocessing was done to assert the required specification for training and test data sets. Some Models implemented sequential Keras convolutional neural network developed from the baseline and on other hand pre-trained models were also used with frozen weights to train the data.

Steps:-

- ➔ Gathering and Pre-processing of data
- ➔ Splitting into train and validation sets and creating Np array files
- ➔ Data Augmentation
- ➔ Developing Sequential Keras model
- ➔ Developing transfer Models
- ➔ Models Evaluations

SEQUENTIAL MODEL

Sequential Keras architecture from the baseline model as follows:

- ➢ Convolution Layer: Conv2D is the first layer that generates convolutional kernel with (3*3) kernel size and layers learned from the input number of filters.
- ➢ Dropout Layer: To prevent the model get overfit on the trainable features Keras dropout layers were added at the frequency rate of (0.4)
- ➢ Flattening Layer: This is a simple layer that basically converts 3d features to one dimensional feature to make the features available for fully connected layers inside the network.
- ➢ Dense Layer: This layer added as an output layer that takes the output of the previous layer and passed to all its neurons and each neuron further generated outputs for the next layer. Two dense layers were added in the output with number of classes 6 and 'softmax' as an activation for multi-label classification

All models used Softmax as activation function for the last fully connected layer, and a categorical cross entropy loss function, whose formula is as follows.

$$L(y,\hat{y}) = -\sum_{j=0}^{M}\sum_{i=0}^{N}(y_{ij} * log(\hat{y}_{ij}))$$

For all models, we used Adam optimizer and a mini-batch size of 32. Besides the plain model, we tried adding dropout rate to fully connected layers.

RESULTS

Training and test accuracy of all models tested can be found in the table below. In general, our models had test accuracy ranging from about 70% to 80%.

Note: All transfer Models used pretrained weights i.e layers were frozen with weights from Imagenet data

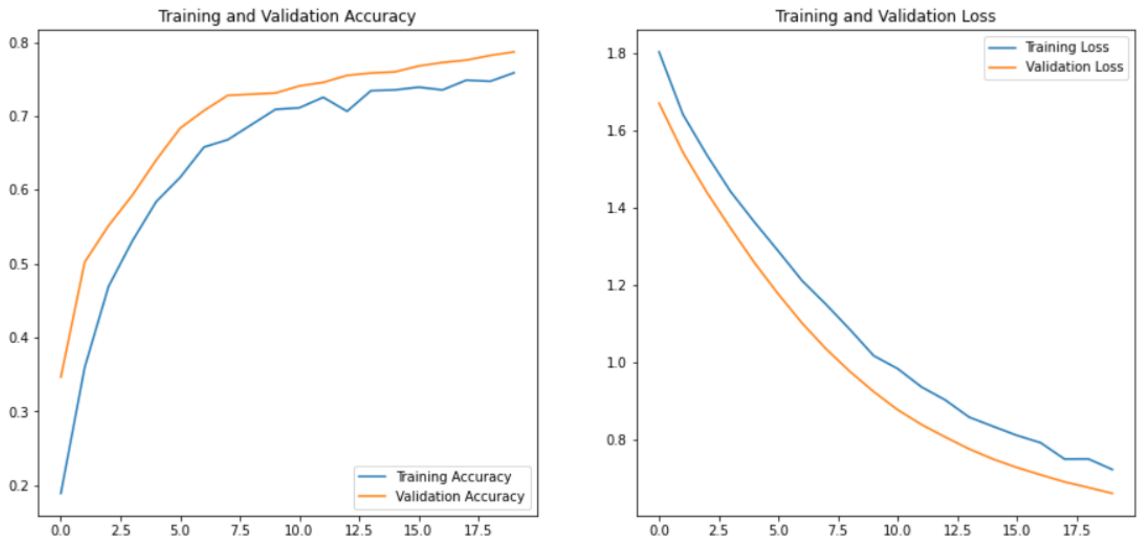
For all models

a) Output Activation – Softmax
b) Layer Acvtivation – Relu
c) Learning rate = .00005
d) Optimizer – Adam
e) Loss – CategoricalCrossentropy
f) Epochs – 20-25

MODEL	Training Accuracy	Validation Accuracy
Custom Model 1 using 3 Conv layers and 2 FC layers	0.475	0.482
Custom Model 2 using 4 Conv and 2 FC layers	0.454	0.494
InceptionV3	0.722	0.752
MobileNetV2	0.756	0.742
DenseNet121	0.700	0.731
VGG16	0.562	0.652
Resnet50	0.263	0.273
Xception	0.759	0.787
HyperModel using Random Search – Best Model	0.564	0.622

As seen from the above table, Inception & Xception models worked the best on our data

Below are the accuracy & loss curves for our best model - Xception



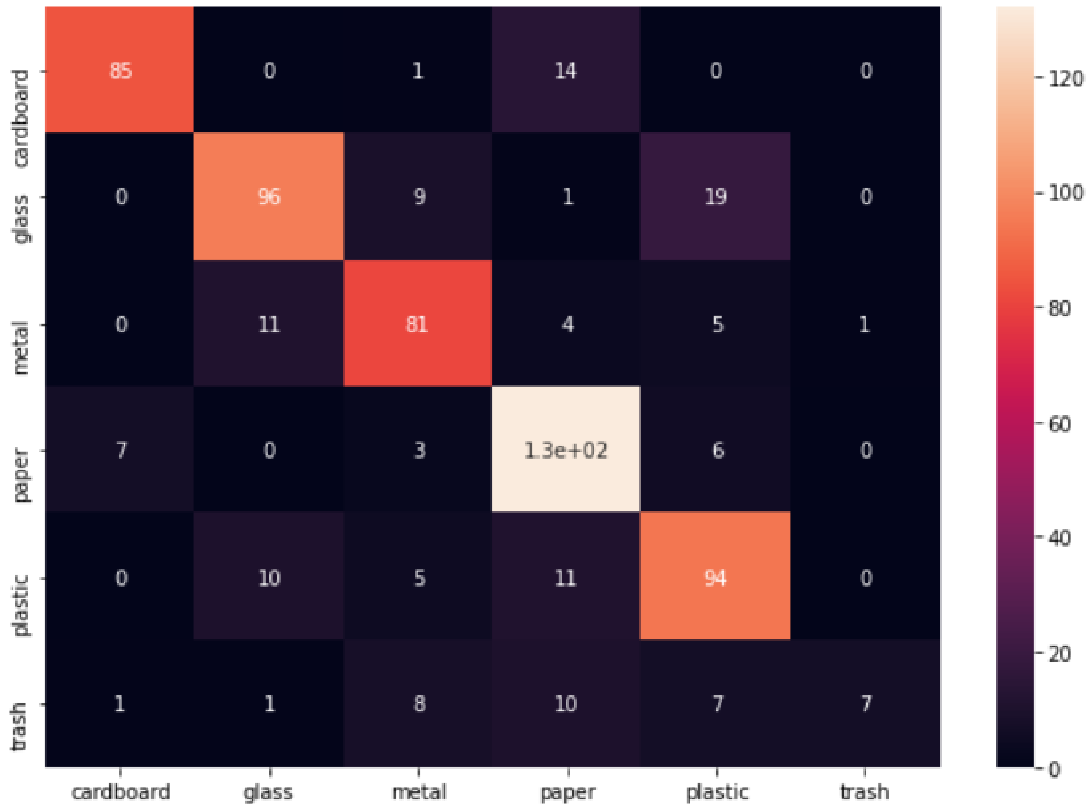
Classification report for Xception Model

Classification Report

	precision	recall	f1-score	support
cardboard	0.91	0.85	0.88	100
glass	0.81	0.77	0.79	125
metal	0.76	0.79	0.78	102
paper	0.77	0.89	0.82	148
plastic	0.72	0.78	0.75	120
trash	0.88	0.21	0.33	34
accuracy			0.79	629
macro avg	0.81	0.72	0.73	629
weighted avg	0.79	0.79	0.78	629

As seen our predictions are pretty good for our labels except Trash because we have very few test samples – 34

Confusion Matrix for Xception Model



CONCLUSION

- ➢ We tried a lot of models to see which models worked best on our data, **Xception** & Inception proved to be the best, but maybe with more tuning we can achieve an improvement in accuracy by 4-5%.
- ➢ We achieved the best accuracy of 79% for validation accuracy with **Xception** model
- ➢ Even with limited challenges, the overall performance of the models is satisfactory and the results obtained are accurate and sufficient enough to use in similar applications.
- ➢ Project was conducted with full genuineness and according to the findings it can be seen that the transfer learning models outperformed custom models.
- ➢ Scope can be widened by using techniques like object detection algorithm and segmentation for achieving higher accuracy with a balanced dataset and more data size for scalability
- ➢ Further data are needed to achieve higher accuracy rates. In the context of our proposed model, we have achieved high classification success.

REFERENCES

- G. Thung and M. Yang. "Dataset of images of trash”
- https://github.com/KhazanahAmericasInc
- https://github.com/frankplus/trash-cnn
- Himanshu Gupta, http://norma.ncirl.ie/
- https://sicara.ai/blog/author/julie-prost