Trash Image Classification System using Machine Learning and Neural Network Algorithms

MSc Research Project
Data Analytics (MSCDA-B)

Himanshu Gupta
Student ID: x18203302

School of Computing
National College of Ireland

Supervisor: Dr. Muhammad Iqbal

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Himanshu Gupta
Student ID:	x18203302
Programme:	Data Analytics (MSCDA-B)
Year:	2020
Module:	MSc Research Project
Supervisor:	Dr. Muhammad Iqbal
Submission Due Date:	17/08/2020
Project Title:	Trash Image Classification System using Machine Learning and Neural Network Algorithms
Word Count:	XXX
Page Count:	15

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	13th August 2020

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Trash Image Classification System using Machine Learning and Neural Network Algorithms

Himanshu Gupta
x18203302

Abstract

Abstract goes here. You should provide a high-level (approx. 150 – 250 words) overview of your paper, its motivation, and the core findings. This is the teaser of your work – it'll probably be best to write it last.

1 Introduction

1.1 Background and Motivation

Municipal garbage exists in the world since the time of civilizations and with the increasing number of humans, the proportional garbage has also increased. The management of trash became essential around 500 BC due to a lack of a proper trash handling system to avoid throwing rubbish on streets which cause various unhygienic health concerns. In the early years waste was burned and disposed of but it was not very effective for a long time and created a nuisance for the city population. With time different countries came up with several laws to curb the city pollution within its limits and found that some garbage can be recycled and some can be used as a source of energy which makes segregation of garbage as one of the main aspects of trash management (Williams; 2005).

In today's time, garbage and trash are available in various forms from packaged food to shopping bags. Fast-paced aspects of urban lifestyle contributed more to garbage pollution which has created the necessity of trash classification. Though there is an existing mechanism to deal with this concern with the increasing locations of finding rubbish, the system needs to improve. The process of waste collections and segregation by humans involves mishandling and they can mix up the materials that resulted in wastage of time and resources.

1.2 Research Question and Objective

1.2.1 Research Question

"How effectively Machine Learning and deep learning approaches can be used for trash image classification and prediction ?"

1.2.2 Research Objective

The primary research objectives of this study are as follows

- Data downloading and pre-processing of trash images.
- Selection of relevant data features.
- Generating more data images and creating new data directory.
- Machine learning and transfer learning models implementation using VGG-19, ResNet-50, Keras Sequential, and XGBoost models.
- Models evaluation, cross-validation, and optimization.
- Comparison of Results achieved through various models.

1.3 Limitation and Challenges

Considerable researches have been conducted on garbage and trash classification using machine learning methods. The biggest challenge in the implementation of these systems is collecting the image data and its associated labels, bounding boxes, etc. The image data has its parameters like image size, image background, number of images, dimension reduction, augmentation which needs to be considered while building the machine learning models.

2 Related Work

It is found that the increase in the garbage quantity globally makes the environment very polluted for all the species including humans due to the environment polluted things like plastics trash, wrappers, carton, etc., thrown out outside making the separation of garbage and its respectability tough. Some researches focused on the combination of monitoring technology for the environment to classify garbage which helps in the recycling and solid waste management system with some real-time monitoring (Wang et al.; 2018). But in this research study, the primary focus would be developing models based on deep learning and transfer learning models to accurately classify trash images.

2.1 Image Classification

Modern problems of machine learning can be handle with the help of image classification because of lots of images available online. Several breakthroughs have been achieved so far related to object detection, image labeling, and their classification by various researchers. The various machine learning models are implemented on images and their performance are majorly depends on how well the feature extraction process implements in the system from the images (Sharma et al.; 2018). Image classification is not just used in simple applications but various remote sensed based applications as well and one such study explored to find the land cover mappings based on the objects using satellite images and aerial remote sensing data as studied by (Ma et al.; 2017). Image classification has useful in several applications like smart cities where visual surveillance takes the image of trash at various spots in the city resulted in better trash monitoring (Kaljahi et al.; 2019).

2.2 Deep Learning Techniques

Deep learning models are extensively used in computer vision problems. It is difficult to classify images and implement ML models in some domains like Medical images where the system cannot just rely on the traditional machine learning approaches such as texture, color of the medical images and need deep learning techniques (Lai and Deng; 2018) . In the areas of garbage and trash classification, popular research was presented by (Yang and Thung; 2016) where they have created the dataset of 400-500 of garbage images and applied SVM and convolutional neural network and it was found that SVM performed better than others with the accuracy of 63%, but the results are not significant to establish why the neural network was not worked better and the smaller data size limits the research exploration. Also, the possibility of having more than one object in the image has not considered while training the model. In very recent similar study (Satvilkar; 2018) very rich collection of trash images data used after Bing search and ML models like SVM, XGBoost, and CNN implemented for classification. A dropout layer of 25% has been added in CNN model layers to regularize the model and among all these models CNN has performed the best with the accuracy of 89 percent similarly Trash Net dataset has been used which were earlier collected by (Yang and Thung; 2016) having six trash categories like glass, paper, cardboard, etc which further used for classification based on deep learning approaches (Bircanoğlu et al.; 2018).

For the separation of waste and their sorting most picked out machine learning algorithms are Support vector machines and deep learning with CNN which have used different classifiers for the image classification. Here, SVM achieved 94.8% accuracy and CNN achieved 83% also SVM successfully classify more images that got contaminated by other waste material and were not so clear and the image size consideration is 256*256 (Sakr et al.; 2016). To make the waste segregation process quick deep learning can be used and one such deep learning framework is called Caffee framework which is open source and very reliable and generally use to classify images of the large dataset by using the power of GPU. It can also deploy a deep learning model on the cloud infrastructure unlike the regular CNN model (Sudha et al.; 2016). Region proposal generation based approaches of deep learning also explored in a study done by (Zhihong et al.; 2017) for object recognition and then forward those input to a classifier. State of the art method Fast R-CNN used which is the combination of RPN and VGG-16, here 'bottle' is the target category and the input data is of 1999 images into the model which integrates with the robotic vision.

In many IoT devices two types of components involved in the architecture, one is for identifying and classifying images of the bottle and another is using sensors proximity for identification of aluminum cans. Images get scanned by the machine and give a Boolean output whether the image is of a plastic bottle or not based on the identification algorithm of the bottle and if the proximity sensor value is high then it identifies as an aluminum can. This sort of electronic Bottle recycle machine (BRM) has been installed at one of the railway stations in India to detect bottles. Convolutional neural network classifier is used as supervised learning to extract the features from the input images and then compare those features with the known ones. The only limitation was the smaller data collection of around 400 images of cans and plastic bottles. CNN performs better as compared to the barcode mechanism with an accuracy of 80 to 100 with or without labels (Dhulekar et al.; 2018). One study researched to segregate plastic material from non-plastic material with the help of CNN architecture(Tarun et al.; 2019) in which the output is identified by one associated layer and it has two hidden layers. Output probabilities were calculated and 0.5 value considered as the threshold value for being classified into plastic and non-plastic, though it is binary classification CNN layered architecture can be observed which gives the accuracy of almost 98% in every scenario.

In one of the recent studies, image classification has been done with the help of CNN where images belong to categories like plastic, metal, cardboard, and paper. The sequential model of CNN implemented with layered architecture and using softmax function which gives values between 0 and 1 for all the 4 classes. A total of 1889 images trained and 188 images used for testing purposes. Accuracy for training data has found as 99.12 percent and for testing, it is found as 76.19% but no cross-validation has been performed for cross-validation to check the overfitting of the model (Sidharth et al.; 2020).

2.3 Transfer Learning Techniques

Sometimes to develop a better neural network we avoid to train the model layers from the scratch and use some pre-trained model and their knowledge on our custom dataset by making a small change in our architecture which is called transfer learning and the expense of training a new model is get saved while using the transfer learning (Gupta et al.; 2017). Pre-trained model ResNet 50 which is 50 layer CNN used as a feature extractor and for the classification of waste images into different categories multi-class SVM classifier used on the input image dataset developed by (Yang and Thung; 2016) After data splitting into 80:20 ratio of the train and test the model achieves an accuracy of 87% overall. The loss for training keeping constant after 12 epoch therefore for every input used backpropagation in the network. The average training accuracy was 94.5% (Adedeji and Wang; 2019) and to handle the complexity of big data especially with fewer annotations available the lightweight neural network used in combination with transfer learning along with SVM classifier to classify waste images which perform very optimize with 98.4% classification accuracy (Xu et al.; 2020). A hybrid of residual networks and inception networks called an inception-ResNet-v2 deep learning module successfully achieved an accuracy of 89% (Aral et al.; 2018). The performance of ResNet-50 and VGG is explored by (Srinilta and Kanharattanachai; 2019) in the classification of waste types and outperformed all other architectures with the accuracy of 91.30%. ReLU and three convolutional layers are used in the neural network. There are few limitations of deep neural network which affects model performance because of gradient dispersion (Simonyan and Zisserman; 2014) and this performance issue got resolved by using the ResNet50 architecture based on residual technique in the network as proposed by (He et al.; 2016).

3 Methodology

The proposed study look into the use of various deep learning and transfer learning models for the classification of trash images. Firstly, the use and importance of creating suitable data for the modelling has been asses thoroughly. Some data input requirements are depend on the particular method implementations such as sequential keras convolutional neural network has build from the scratch on other hand pre-trained models such as ResNet 50 and VGG-19 were required specific image dimensions. Cross validation techniques has been used to support experiment results. Moreover, not many studies conducted research on this newly available dataset TACO.

Methodology implemented includes the following procedures :

Figure 1: Drink Can

Figure 2: Cigarette

Figure 3: Clear Plastic Bottle

Figure 4: Plastic Film

Figure 5: Plastic Straw

- Gathering and Pre-processing of data
- Data Analyses and Creating suitable dataframe
- Developing and optimizing Sequential Keras model
- Developing and optimizing ResNet-50 architecture model
- Developing and optimizing VGG-19 architecture model
- Developing and optimizing XGBoost model
- Models Evaluations and Results discussions.

3.1 Data Description and Acquisition

The data used for this research are the collection of trash images collected and generated by (Proença and Simões; 2020). All the images are labelled manually and hosted on Flickr servers which make these images available to train object detection and deep learning algorithms. TACO has provided the python script for downloading the images. It is a publicly available image dataset named 'TACO' which have total 1500 images belongs to 60 categories[1] such as Cigarette ,Paper cup, Unlabeled litter, Clear plastic bottle, Drink can etc. All the images are having realistic background of diverse places.Majorly five class categories chosen from the dataset for this research as mentioned in the table 1 and have total 805 images considered initially.

3.2 Data Pre-processing

In Data pre-processing the 'annotation.json' file has been loaded and then categories, annotations and images values were extracted from the dataset and converted into the pandas data frames. All the unwanted columns and features dropped from the data frame. There was no missing value in the data.

Initially total number of images was less therefore more data has been generated using the following data augmentation techniques:

- Image Cropping
- Image Rotation

[1] http://tacodataset.org/

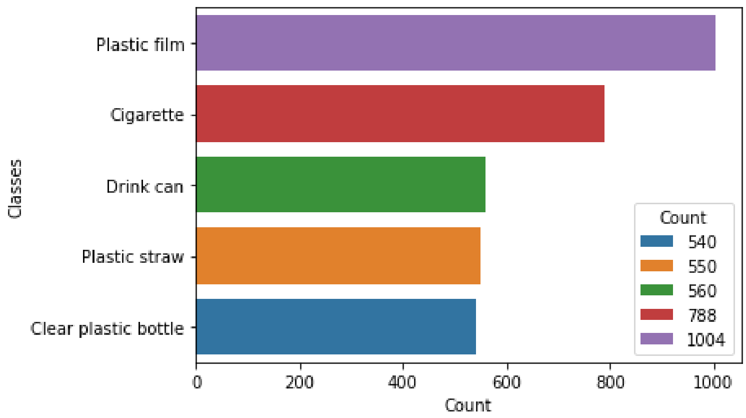

Figure 6: Data After Augmentation

	filename	category_Cigarette	category_Clear plastic bottle	category_Drink can	category_Plastic film	category_Plastic straw
0	batch_1/000010cropped.jpg	0	1	0	0	0
1	batch_1/000010rotated.jpg	0	1	0	0	0
2	batch_1/000010blur.jpg	0	1	0	0	0
3	batch_1/000010hflip.jpg	0	1	0	0	0
4	batch_1/000001cropped.jpg	0	1	0	0	0
...	...	...	...	...	...	...
3437	batch_9/000096cropped.jpg	0	0	0	0	1
3438	batch_9/000096rotated.jpg	0	0	0	0	1
3439	batch_9/000096blur.jpg	0	0	0	0	1
3440	batch_9/000096hflip.jpg	0	0	0	0	1
3441	batch_9/000096vflip.jpg	0	0	0	0	1

3442 rows × 6 columns

Figure 7: Dummy Categories

- Horizontal flip
- Gaussian Blurring
- Vertical flip

Bounding boxes for each file was used in calculating the minimum and maximum values for horizontal as well vertical axes of image which was further used for image cropping. To avoid the over edging some padding has been added while cropping. Cropped images were further augmented into three more kind of images. Rotate the images at 88 degrees randomly, horizontally flipped the images from left to right which gives mirror reflection. Finally blurred the cropped image using Gaussian blur at radius 0.2 for making the dataset more realistic. Images for 'Drink Can' and 'Plastic straw' were very less making the data much imbalanced, so few vertical flip images for these two categories were added in the dataset. All these images saved into a new data directory with renamed file names but same labels of original images. Total number of successfully generated new images is 3442 as shown in figure category wise 6.

4 Implementation

The implementation involves following steps: preparation of data for each model, defining the model architecture and parameters, training and testing the implemented models, evaluating the model results and cross validation. Validating the model for over fitting and under fitting and optimize the model parameters for reliable results. All the implemented models are based on the neural networks but with different architecture style. Firstly,

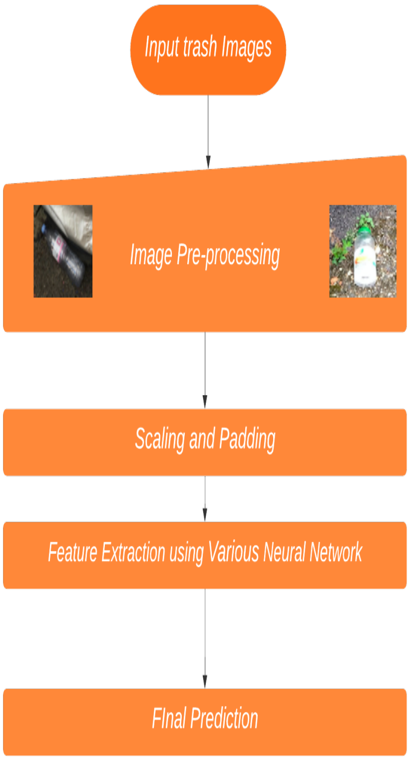

Figure 8: Process Flow

input array data split into the training and testing dataset in 80 to 20 ratio using sklearn machine learning library along with setting the stratify parameter true which split the data in such a way that subsets had same proportions of class labels.

During the model training 20 percent of training subset passed as validation data to monitor the validation accuracy which gets calculated at the end of each epoch. Keras and pre-trained models run for (50,100) number of epochs. For high number of epochs sometimes the model get over-fit therefore to stop the training early stopping was used with different patience values. Patience value is equivalent to number of epoch and after specific epoch being trained model monitor the validation loss and if the validation loss is not improving then training has stopped. After the model training, K-folds cross validation applied on different number of folds(10,20) which split the training into multiple folds and then train the model to measure the performance of the model and validate our results.

4.1 The Sequential model - Keras

Keras models are layer of network topology use in implementing deep learning problems and constructed on top of machine learning platform. In this research the sequential model of Keras API was used to develop neural network [2]. Each network layer takes one tensor input and give one tensor output. For Keras Convolutional neural network and pre-trained model like ResNet-50 the X input features of images were converted into the numpy array before feeding into the input layer. Since, my labels columns containing the categorical values it was needed to convert them into the categorical data type. This has achieved by creating dummy variables for each category which has represented as '1' against each image as shown in figure 7. After considering grid search model parameters are used in the model creation.Rectified linear unit used as an activation function in the input layers which use zero as threshold value and gives 0 for all the negative values f(x)=max(0,x). ReLU mostly used in computer vision neural which improves the convergence value networks (Krizhevsky et al.; 2012). As this is a multi class image classification softmax activation function used in the output layer which gives probabilities between 0 and 1. Implemented Sequential Keras architecture from the baseline model as follows:

- Convolution Layer: Conv2D is the first layer which generate convolutional kernel with (3*3) kernel size and layers learned from input number of filters.
- Dropout Layer: To prevent the model get over fit on the trainable features keras dropout layers were added at the frequency rate of (0.5) which sets the hidden output neurons to 0 randomly at each training update phase.
- Batch Normalization Layer: This layer was used to increase the overall efficiency of the network by normalizing the input features. It performed re-scaling as well as re-centering of input layers.
- Flattening Layer: This is a simple layer which basically convert 3d features to one dimensional features to make the features available for fully connected layers inside the network.
- Dense Layer: This layer added as an output layer which takes the output of previous layer and passed to all its neurons and each neuron further generated outputs for

[2] https://keras.io/guides/sequential$_{m}odel/$

```python
def create_new_model():
  model = Sequential()
  model.add(ResNet50(include_top = False, pooling = 'avg', weights = 'imagenet'))
  # Second layer added for dropout
  model.add(keras.layers.Dropout(0.5))
  # Third layer as Dense for output 5-class classification
  model.add(Dense(5, activation = 'softmax'))
  # No need to train first layer (ResNet) model as it is already trained
  model.layers[0].trainable = False
  model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
  return model
```

Figure 9: ResNet-50 model function call

the next layer. Since, only one dense layer has been added in the output which takes number of classes (5 in this case) and 'softmax' as an activation for multi label classification

Model got trained with the batch size of 50 and along with 20% validation split and early stopping with patience 20.

4.2 ResNet-50 Pretrained Model

ResNet-50 architecture implemented which is a 50 layer deep neural network. Model is pre-trained on weights of very big image dataset called 'imagenet' which contained more than million images. Loaded the ResNet network layer as the first layer with the average pooling for the weights. This layer has already pre-trained so no need to train this layer again. Train the model with the same parameters given in the Keras without adding any extra layer because of less data availability. Training and testing accuracy as well as loss were calculated after model evaluation on train and test features. In ResNet 50 architecture there are three types of layers 1 max pool layer, 48 convolutional layers and 1 average pool layer. This model can be used for computer vision as well as non computer vision classification problems for better achieving better accuracy. For reducing the training error rate, ResNet use the concept of deep residual network.

4.3 VGG-19 Pretrained Model

Training a CNN model from the base has its own advantages and disadvantages. Lots of grid parameters are needed to run for finding the optimal setting in neural network. To resolve this issue adaptation of already optimized and learned model is beneficial. Therefore similar to ResNet-50 transfer learning model, VGG-19 (Visual Geometry Group) is trained on imagenet and proven to work very efficiently with smaller dataset (Jaworek-Korjakowska et al.; 2019). It consist of 19 deep layers such as dropout, max pooling, convolutional etc. VGG-19 used primarily for the classification of training layers consist of dropout as well as dense layer. In the implementation Keras Sequential model has been used which pre load and adapt the behaviour of VGG-19 to trained on the TACO custom dataset. Firstly, all the input features were pre processed using VGG pre processor and features were extracted in a batch size of 50. The final output of VGG model used to train the input of Keras layers. As the objective is to classify multi label class the loss function and optimizer setting were set similar to above models. K-fold cross validation methods with (10,20,50) folds applied on the developed model.

VGG-19 architecture is deep CNN network which overcomes the shortcomings of Alexnet neural network (Shaha and Pawar; 2018). VGG-19 consist of following layers:

```
Model: "vgg19"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 128, 128, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 128, 128, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 128, 128, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 64, 64, 64)        0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 64, 64, 128)       73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 64, 64, 128)       147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 32, 32, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 32, 32, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 32, 32, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 32, 32, 256)       590080
_________________________________________________________________
block3_conv4 (Conv2D)        (None, 32, 32, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 16, 16, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 16, 16, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 16, 16, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 16, 16, 512)       2359808
_________________________________________________________________
block4_conv4 (Conv2D)        (None, 16, 16, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 8, 8, 512)         0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 8, 8, 512)         2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 8, 8, 512)         2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 8, 8, 512)         2359808
_________________________________________________________________
block5_conv4 (Conv2D)        (None, 8, 8, 512)         2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 4, 4, 512)         0
=================================================================
Total params: 20,024,384
Trainable params: 20,024,384
Non-trainable params: 0
_________________________________________________________________
```

Figure 10: VGG-19 Layer Structure

16 Convolutional layers which implemented in groups of (2,2,4,4,4) with different number of filters(64,128,512) , 3 fully connected layers, 5 max pooling layers and 1 softmax layer as shown in figure 10

4.4 eXtreme Gradient Boosting

XGBoost is available as open source package for Applied machine learning problems. The motivation behind implementation of this method is finding the good classification score because of smaller dataset. XGBoost model made by (Chen and Guestrin; 2016). XGBoost works based on gradient boosting and internally operates on tree learning techniques.

Multi Softmax activation function was used with 'mlogloss' function which used for evaluation metrics. For the data preparation data is converted to numpy array first and then the label column which contains the five classes was converted to categorical data type using pandas with category codes as [0,1,2,3,4]. All train and test features

```
xgb_params = {'num_class':5, # Number of Output classes
              'nthread':8, # Number of Parallel threads
              'gamma':0.1, # For minimizing loss
              'eval_metric':'mlogloss', # Multi classification evaluation
              'min_child_weight':3, # It performs regularization
              'subsample':0.7, # Percentage of rows consider while building subtree
              'max_depth': 16, # Maximum number of nodes between root and farthest leaf
              'objective': 'multi:softmax', # Multi label classification
              'seed': 1337, # For reproducibility
              'silent': True}
```

Figure 11: Parameters selected for XGBoost

reshaped according to input array shape which is (128,128,3) and saved into a 'CSV' file as numerical values. Numeric data was fetched and then optimized data structure was created using 'DMatrix' package which increase the training speed as well as the memory efficiency. Several input parameters passed for tuning the model as shown in . For cross validation xgboost model trained using its own cross validation function into 2 and 5 folds with '50' boosting round and '10' early stopping rounds while evaluation metrics keeps same as 'mlogloss'.

5 Evaluation

In this study implemented deep neural and machine learning models were evaluated and performance was measured based on several parameters like training accuracy, testing accuracy, validation loss, Average precision, recall, etc. Firstly, the sub-sampling of the 80% to 20% ratio divided between training and testing data. Test data keep aside at the beginning as an unseen data used for model evaluation. Models were getting trained mostly on training data but in addition to that 20% of training, data passed as validation during training. Results for model training using cross-validation were also recorded and discussed in the below sections.

Models	Accuracy	Accuracy after Data Augmentation
Sequential model - Keras	27%	61%
ResNet-50	71%	
VGG-19	82%	
XGBoost	29%	70%

Table 1: Evaluation of Classification Techniques

5.1 Experiment with the Sequential model

After plotting the training and validation accuracy plots it seems that the model stopped training at around 24 epoch because validation accuracy was not increasing as shown in fig 12. Accuracy on training augmented data was 89% and for testing around 60%. Cross-validation comparison for both model types shown in fig 13.

...

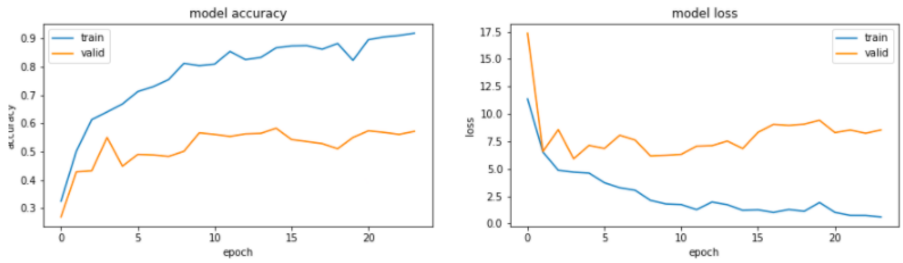

Figure 12: Sequential model Plots

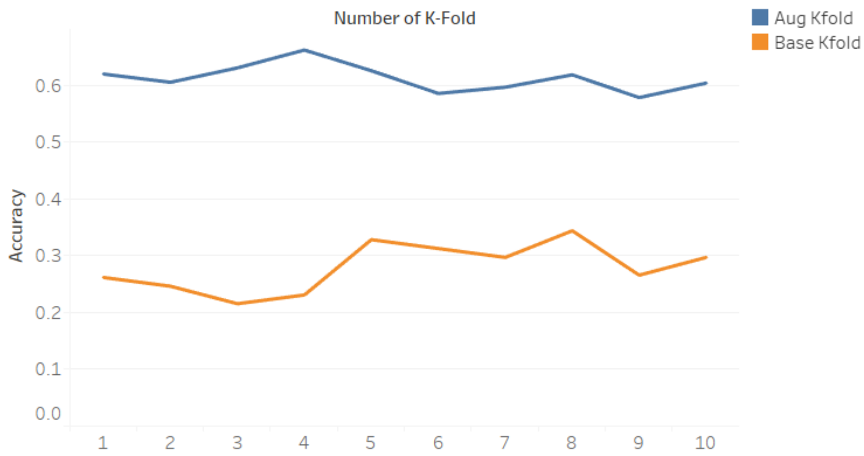

Figure 13: Sequential model cross validation plots

5.2 Experiment with ResNet-50

...

5.3 Experiment with VGG-19

...

5.4 Experiment with XGBoost

Classes	precision	recall	f1-score
Sequential model - Keras	27%	61%	
ResNet-50	71%		
VGG-19	82%		
XGBoost	29%	70%	

Table 2: Classification Report XGBoost

...

5.5 Discussion

The baseline Keras model developed from scratch on a smaller dataset as well as on the augmented dataset. There is a notable difference between the training and testing accuracy of both models 1 and 2. Also, the validation accuracy was not increasing after 10 epochs on the other hand training accuracy were keep on increasing which indicates the overfitting of the model. Though some CNN regularization techniques (Xu et al.; 2019) were applied to the model, no significant improvement was found.

6 Conclusion and Future Work

Restate your research question, your objectives and the work done. State how successful you have been in answering the research question and achieving the objectives. Restate the key findings. Discuss the implications of your research, talk about the efficacy of your research, and discuss its limitations.

Describe any proposals for future work or potential for commercialisation. Present MEANINGFUL future work. Sweeping more parameters in your simulation / model / platform is probably not meaningful. More discuss what could a follow up research project do, to better / differently approach / extend etc. your work.

References

Adedeji, O. and Wang, Z. (2019). Intelligent waste classification system using deep learning convolutional neural network, *Procedia Manufacturing* **35**: 607–612.

Aral, R. A., Keskin, Ş. R., Kaya, M. and Hacıömeroğlu, M. (2018). Classification of trashnet dataset based on deep learning models, *2018 IEEE International Conference on Big Data (Big Data)*, IEEE, pp. 2058–2062.

Bircanoğlu, C., Atay, M., Beşer, F., Genç, Ö. and Kızrak, M. A. (2018). Recyclenet: Intelligent waste sorting using deep neural networks, *2018 Innovations in Intelligent Systems and Applications (INISTA)*, IEEE, pp. 1–7.

Chen, T. and Guestrin, C. (2016). Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining-kdd'16.

Dhulekar, P., Gandhe, S. and Mahajan, U. P. (2018). Development of bottle recycling machine using machine learning algorithm, *2018 International Conference On Advances in Communication and Computing Technology (ICACCT)*, IEEE, pp. 515–519.

Gupta, D., Jain, S., Shaikh, F. and Singh, G. (2017). Transfer learning & the art of using pre-trained models in deep learning, *Analytics Vidhya* .

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Jaworek-Korjakowska, J., Kleczek, P. and Gorgon, M. (2019). Melanoma thickness prediction based on convolutional neural network with vgg-19 model transfer learning, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0.

Kaljahi, M. A., Palaiahnakote, S., Anisi, M. H., Idris, M. Y. I., Blumenstein, M. and Khan, M. K. (2019). A scene image classification technique for a ubiquitous visual surveillance system, *Multimedia Tools and Applications* **78**(5): 5791–5818.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks, *Advances in neural information processing systems*, pp. 1097–1105.

Lai, Z. and Deng, H. (2018). Medical image classification based on deep features extracted by deep model and statistic feature fusion with multilayer perceptron, *Computational intelligence and neuroscience* **2018**.

Ma, L., Li, M., Ma, X., Cheng, L., Du, P. and Liu, Y. (2017). A review of supervised object-based land-cover image classification, *ISPRS Journal of Photogrammetry and Remote Sensing* **130**: 277–293.

Proença, P. F. and Simões, P. (2020). Taco: Trash annotations in context for litter detection, *arXiv preprint arXiv:2003.06975* .

Sakr, G. E., Mokbel, M., Darwich, A., Khneisser, M. N. and Hadi, A. (2016). Comparing deep learning and support vector machines for autonomous waste sorting, *2016 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*, IEEE, pp. 207–212.

Satvilkar, M. (2018). *Image Based Trash Classification using Machine Learning Algorithms for Recyclability Status*, PhD thesis, Dublin, National College of Ireland.

Shaha, M. and Pawar, M. (2018). Transfer learning for image classification, *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, IEEE, pp. 656–660.

Sharma, N., Jain, V. and Mishra, A. (2018). An analysis of convolutional neural networks for image classification, *Procedia computer science* **132**: 377–384.

Sidharth, R., Rohit, P., Vishagan, S., Karthika, R. and Ganesan, M. (2020). Deep learning based smart garbage classifier for effective waste management, *2020 5th International Conference on Communication and Electronics Systems (ICCES)*, IEEE, pp. 1086–1089.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556* .

Srinilta, C. and Kanharattanachai, S. (2019). Municipal solid waste segregation with cnn, *2019 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST)*, IEEE, pp. 1–4.

Sudha, S., Vidhyalakshmi, M., Pavithra, K., Sangeetha, K. and Swaathi, V. (2016). An automatic classification method for environment: Friendly waste segregation using deep learning, *2016 IEEE Technological Innovations in ICT for Agriculture and Rural Development (TIAR)*, IEEE, pp. 65–70.

Tarun, K., Sreelakshmi, K. and Peeyush, K. (2019). Segregation of plastic and non-plastic waste using convolutional neural network, *IOP Conference Series: Materials Science and Engineering*, Vol. 561, IOP Publishing, p. 012113.

Wang, B., Zhou, W. and Shen, S. (2018). Garbage classification and environmental monitoring based on internet of things, *2018 IEEE 4th Information Technology and Mechatronics Engineering Conference (ITOEC)*, IEEE, pp. 1762–1766.

Williams, P. T. (2005). *Waste treatment and disposal*, John Wiley & Sons.

Xu, Q., Zhang, M., Gu, Z. and Pan, G. (2019). Overfitting remedy by sparsifying regularization on fully-connected layers of cnns, *Neurocomputing* **328**: 69–74.

Xu, X., Qi, X. and Diao, X. (2020). Reach on waste classification and identification by transfer learning and lightweight neural network.

Yang, M. and Thung, G. (2016). Classification of trash for recyclability status, *CS229 Project Report* **2016**.

Zhihong, C., Hebin, Z., Yanbo, W., Binyan, L. and Yu, L. (2017). A vision-based robotic grasping system using deep learning for garbage sorting, *2017 36th Chinese Control Conference (CCC)*, IEEE, pp. 11223–11226.