Waste Classification System using Convolutional Neural Networks

Submitted in partial fulfillment of
the requirements for the award of the degree of

Bachelor of Technology
in
Computer Science and Engineering

Submitted by
Abhinav Dixit (20151012)
Ishaan Rajput (20154086)
Abhishek Sharma (20154077)
Harshita Rastogi (20154041)
John Prasad (20154010)

Under the guidance of
Dr. Divya Kumar

Department of Computer Science and Engineering
Motilal Nehru National Institute Of Technology, Allahabad
Allahabad, UP, India

April,2018

UNDERTAKING

Motilal Nehru National Institute of Technology Allahabad

We declare that the work presented in this report titled "Waste Classification System using Convolutional Neural Networks", submitted to the Computer Science and Engineering Department, Motilal Nehru National Institute of Technology, Allahabad, for the award of the Bachelor of Technology degree in Computer Science & Engineering, is our original work. We have not plagiarized or submitted the same work for the award of any other degree. In case this undertaking is found incorrect, We accept that our degree may be unconditionally withdrawn.

April,2018

Abhinav Dixit (20151012)
Ishaan Rajput (20154086)
Abhishek Sharma (20154077)
Harshita Rastogi (20154041)
John Prasad (20154010)

Preface

This project presents a work in progress for a proposed method for Waste Classification. It proposes an automated system that segregates waste into 5 categories based on norms accepted globally which are container, organic waste, metal, paper and plastic(including wrappers). This automatic waste segregator uses a modern classification method known as Convolutional Neural Networks to classify the waste into various categories. This system paves way to better recycling and reuse processes that helps in efficient waste management.

CERTIFICATE

Certified that the work contained in the report titled *"Waste Classification System using Convolutional Neural Networks"*, by Abhinav Dixit, Ishaan Rajput, Abhishek Sharma, Harshita Rastogi and John Prasad, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

(Dr. Divya Kumar)
CSED Dept.
M.N.N.I.T, Allahabad

ACKNOWLEDGEMENT

We are profoundly grateful to our mentor Dr. Divya Kumar for his constant help, motivation, support and technical guidance. We express our sincere appreciation for his encouragement and advice that enabled us to pursue this project. Due to his assistance and guidance, we as a team were able to make this project a success.

Contents

Chapter 1

Introduction

A fully automated waste segregation system that can efficiently categorize waste is the need of the hour. This system will help avoid human intervention in the sorting process and hence control the adverse effect of waste on environment and human health. The waste that has been segregated can be directly sent to recycling plants. Currently there exists no system of segregation of wastes into categories like recyclable, organic and non-recyclable at a household level.

This project explores the advantages and accuracy associated with automatic separation of waste. The purpose of this project is to segregate the collected waste. Using the concepts of Artificial Neural Networks and Image Processing, the project is aimed at designing and developing a system that can be effectively utilized to segregate waste. By applying the concepts of recognition and classification in Artificial Neural Networks, the proposed system can be designed to rightly categorize the different types of waste. One such algorithm that is used for the category classification is the Convolution Neural Network (CNN), which is a part of the Machine Learning tools. It is a powerful and a deep layered network that helps in the classification process. The project is also aimed to further develop a system to a fully functional mechanical system that facilitates the physical segregation of the waste material, once classified.

1.1 Motivation

Image classification holds great relevance in this era considering its varied applications and the challenges involved. Great scope of development exists in this area considering its use which even ranges to extensive fields of remote sensing and computer vision. There are now several tasks in Computer Vision where the performance of our models is close to human, or even superhuman. But still several challenges persist even today. These challenges were a major motivation for us to chose image classification as the topic for our mini project. Choosing an apt machine learning algorithm based on the data sets and accuracy requirements is a tough decision altogether. This challenge motivated us to further explore different algorithms which would enable us to make better choices.

1.2 Problem Statement

To carry out the image classification of waste materials using Convolutional Neural Networks(CNN). Given an image of a random waste item, we seek to categorize the waste item into one of the 5 mentioned categories.
Categories for classification:

1. Container
2. Organic waste
3. Metal
4. Paper
5. Plastic

1.3 Challenges in Computer Vision

Computer vision is an interdisciplinary field that deals with how computers can be made for gaining high-level understanding from digital images or videos. Computer vision tasks include methods for acquiring, processing, analyzing and understanding digital images, and extraction of high-dimensional data from the real world in order to produce numerical or symbolic information.

Tradition Machine Learning Algorithms do not work effectively in Image Processing. The fact that the output has complex relation with the input presents difficulties in finding relations that can be exploited by a tradition classification or regression models. Other than this, the huge size of an image also present challenges for machine learning. Deep learning has distinct advantages over machine learning when processing unstructured data without polynomial relation like text, sound and images. When provided with sufficient training dataset, a deep neural net is able to find these non-linear patterns. Conventional neural networks can be deployed for analyzing images in theory, but in practice, it will be highly expensive from a computational perspective. The second challenge is achieved through CNN.

Chapter 2

Related Work

Image-based information is a key component of human progress in a number of distinct subject domains and digital image retrieval is a fast-growing research area with regard to both still and moving images. Convolutional neural networks were one of the first successful deep neural networks. Some of the recent advances in this field are as follows:

2.1 Neocognitron

The Neocognitron, developed by Fukushima in 1980s, provided a neural network model for translation-invariant object recognition, inspired by biology. Le Cun et al combined this method with a learning algorithm,i.e. back-propagation. These early solutions were mostly used for handwritten character recognition.

2.2 AlexNet

In the realm of image classification, one wellknown and highly capable CNN architecture is AlexNet, which won the 2012 ImageNet LargeScale Visual Recognition Challenge (ILSVRC). The architecture is relatively simple and not extremely deep, and is, of course, known to perform well. AlexNet was influential because it started a trend of CNN approaches being very popular in the ImageNet challenge and becoming the state of the art in image classification.

2.3 ImageNet

For the 2014 ImageNet challenge, Simonyan and Zisserman explored the effect of increasing the depth of a convolutional network on localization and classification accuracy. The team achieved results that improved the then state-of-the-art by using convolutional networks 16 and 19 layers deep. The 16-layer architecture includes 13 convolutional layers (with 3x3 filters), 5 pooling layers (2x2 neighbourhood max-pooling) and 3 fully-connected layers. All hidden layers use rectified (ReLu) activations. The fully-connected layers reduce 4096 channels down to 1000 softmax outputs and are regularized using dropout.

2.4 CRAFT

The 2016 winner of the object detection category in the ImageNet challenge is also CNN-based. The method uses a combination of CRAFT region proposal generation [55], gated bi-directional CNN, clustering, landmark generation and ensembling.

Chapter 3

Proposed Work

The project involves a comparative analysis of image classification using CNN. Our basic aim is to obtain high accuracy in classification of waste with images of varied qualities.

The waste segregation system mainly involves two stages:

1. Acquisition of the images of the waste material

2. Classification by the Convolutional Neural Networks

The fact that CNNs are trained end-to-end, from raw pixels to final classes, makes them much more advantageous for many tasks than manually designing a suitable feature extractor.We propose a multi-scale deep learning approach that can be used for classification,localization and detection.

The system is capable of classifying the object into best matched class and is thus segregated into the class. This can be extended so that the whole system is fully automated and the waste treatment/disposal can be performed accordingly.

Chapter 4

Background

4.1 Artificial Neural Networks

4.1.1 Artificial Neurons

Artificial neural network consists of artificial neurons. Artificial neuron is a mathematical function modeling a biological neuron. The neuron receives one or more weighted inputs and fires an output.The inputs represent dendrites and output represents an axon within neuroscience perspective. Weights corresponding to each input unit are denoted by vector $\vec{w} \epsilon R^{\mathrm{n}}$, where w_i is the weight of the input unit x_i. The potential p ϵ R is computed as the dot product of the input vector $\vec{x} \epsilon R^{\mathrm{n}}$ and the vector of weights $\vec{w}$. $| \vec{x} |$ denotes the length of the input vector.

$$p = \sum_{i=1}^{|\vec{x}|} x_i w_i \tag{4.1.1}$$

The activation y ϵ R is obtained after application of activation function ϕ on the potential p.

$$y = \phi(p) \tag{4.1.2}$$

The activation functions may be of various types and shapes, among others:

- *Logistic sigmoid function* – mathematical function in S shape rising values in interval (0, 1), t being the shape parameter:

$$y(p) = \frac{1}{1 + e^{-p}} \tag{4.1.3}$$

- *Step function* – foundational activation function. The output is of a binary form, 1 if the input meets specific threshold θ.

$$y(p) = \begin{cases} 1, & \\ 0, & \text{if } p < 0 \end{cases} \tag{4.1.4}$$

- *Ramp function* – as of 2015 the most popular activation function for deep neural networks. A unit using ramp function is also called a rectified linear unit (ReLU).

$$y(p) = max(p, 0) \tag{4.1.5}$$

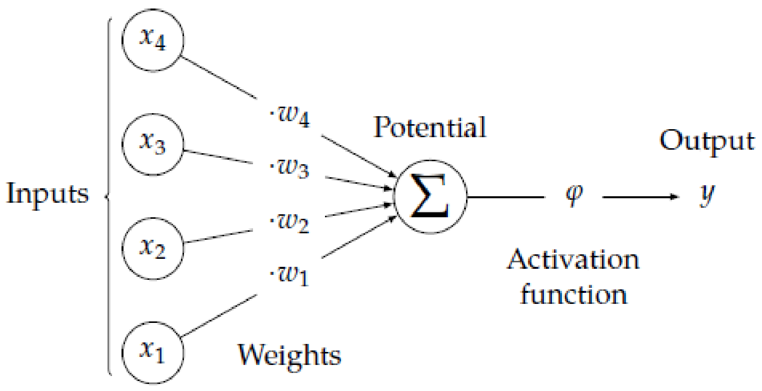

Figure 4.1: Model of an artificial neuron without bias

4.1.2 Feed-forward Pass

The neurons can be interconnected to form a graph. The output of a neuron is used as an input for other neurons. The acyclic such a network is called Feed-forward neural network-information flows only in one direction.

The neurons are divided into the disjoint sets, called layers l_1,..., l_k. Layers l_1,..., l_{k1} are hidden layers, l_k is an output layer. Formally layer l_0 is also considered, denoting the input data also called an input layer. The nodes in layer l_i receive as an input only the outputs of one or more connected neurons in layer l_{i1}. Neurons in a fully connected layer have connections to all activations of neurons in the previous layer. The outputs of an input layer l_0 are the input data.

The activations of neurons in output layer l_k represent the output of the network. It may represent probabilities of belonging to classes. The whole computation on a feed–forward neural network is designed with Forward propagation algorithm.

Forward propagation, the input vector is copied to the input layer. For every other layer (in topological order), firstly, the potentials are computed as multiplication of the activations of previous layer with the vector of weights of each neuron's connections. Then, the activation functions are applied to the potentials.

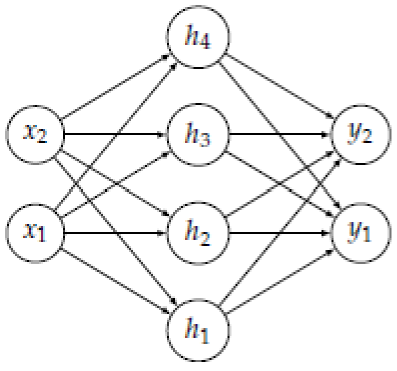

Figure 4.2: Model of a feed-forward neural network with one fully connected hidden layer and fully connected output layer

4.1.3 Loss Function and Optimization

In most learning networks, error is calculated as the difference between the actual output and the predicted output.

$$J(w) = p - \widehat{p} \tag{4.1.6}$$

The function that is used to compute this error is known as Loss Function J(.). Different loss functions will give different errors for the same prediction, and thus have a considerable effect on the performance of the model. One of the most widely used loss function is mean square error, which calculates the square of difference between actual value and predicted value. Different loss functions are used to deal with different type of tasks, i.e. regression and classification.

Error J(w) is a function of internal parameters of model i.e weights and bias. For accurate predictions, one needs to minimize the calculated error. In a neural network, this is done using back propagation. The current error is typically propagated backwards to a previous layer, where it is used to modify the weights and bias in such a way that the error is minimized. The weights are modified using a function called Optimization Function. Optimisation functions usually calculate the gradient i.e. the partial derivative of loss function with respect to weights, and the weights are modified in the opposite direction of the calculated gradient. This cycle is repeated until we reach the minima of loss function.

$$W^{(k+1)} = W^{(k)} - \frac{\partial J(W)}{\partial W^{(k)}} \tag{4.1.7}$$

We have used 2 types of optimization functions in building our model which are as follows:

1. *RMS Prop* – Rmsprop is a gradient-based optimization technique. Gradients of very complex functions like neural networks have a tendency to either vanish or explode as the energy is propagated through the function. And the effect has a cumulative nature – the more complex the function is, the worse the problem becomes. Rmsprop is a very clever way to deal with the problem. It uses a moving average of squared gradients to normalize the gradient itself. That has an effect of balancing the step size – decrease the step for large gradient to avoid exploding, and increase the step for small gradient to avoid vanishing.

2. *ADAM* – Adam stands for Adaptive Moment Estimation. It also calculates different learning rate. Adam works well in practice, is faster, and outperforms other techniques.

4.1.4 Backpropagation

A neural network is trained by selecting the weights of all neurons so that the network learns to approximate target outputs from known inputs. It is difficult to solve the neuron weights of a multi-layer network analytically. The *back-propagation algorithm* provides a simple and effective solution to solving the weights iteratively. The classical version uses gradient descent as optimization method. Gradient descent can be quite time-consuming and is not guaranteed to find the global minimum of error, but with proper configuration (known in machine learning as hyperparameters) works well enough in practice.

In the first phase of the algorithm, an input vector is propagated forward through the neural network. Before this, the weights of the network neurons have been initialized to some values, for example small random values. The received output of the network is compared to the desired output (which should be known for the training examples) using a loss function. The gradient of the loss function is then computed. This gradient is also called the error value. When using mean squared error as the loss function, the output layer error value is simply the difference between the current and desired output.

The error values are then propagated back through the network to calculate the error values of the hidden layer neurons. The hidden neuron loss function gradients can be solved using the chain rule of derivatives. Finally, the neuron weights are updated by calculating the gradient of the weights and subtracting a proportion of the gradient from the weights. This ratio is called the *learning rate*. The learning rate can be fixed or dynamic. After the weights have been updated, the algorithm continues by executing the phases again with different input until the weights converge.

4.2 Convolutional Neural Networks

4.2.1 Justification

The problem with solving computer vision problems using traditional neural networks is that even a modestly sized image contains an enormous amount of information.

A monochrome 620x480 image contains 297,600 pixels. If each pixel intensity of this image is input separately to a fully-connected network, each neuron requires 297 600 weights. A 1920x1080 full HD image would require 2,073,600 weights. If the images are polychrome, the amount of weights is multiplied by the amount of colour channels (typically three). Thus, we can see that the overall number of free parameters in the network quickly becomes extremely large as the image size increases. Too large models cause overfitting and slow performance.

Furthermore, many pattern detection tasks require that the solution is translation invariant. It is inecient to train neurons to separately recognize the same pattern in the left-top corner and in the right-bottom corner of an image. A fully-connected neural network fails to take this kind of structure into account.

4.2.2 Basic Structure

The basic idea of the CNN was inspired by a concept in biology called the receptive field . Receptive fields are a feature of the animal visual cortex. They act as detectors that are sensitive to certain types of stimulus, for example, edges. They are found across the visual field and overlap each other.

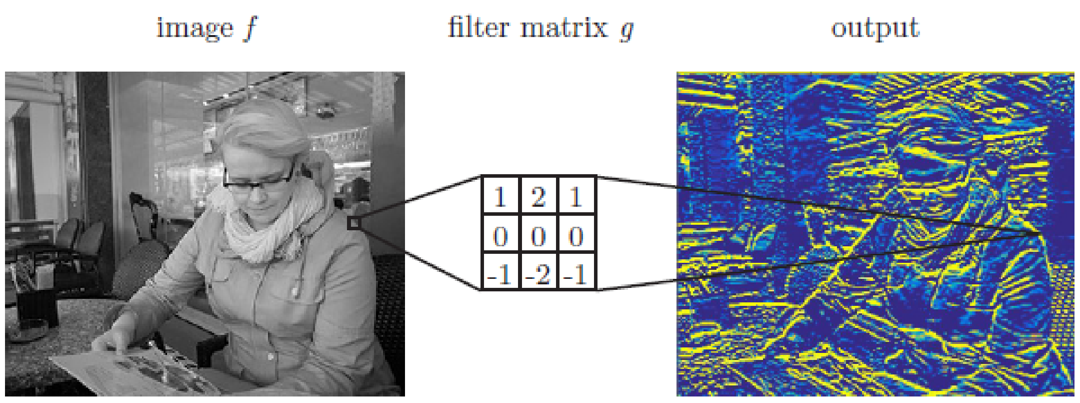

Figure 4.3: Detecting horizontal edges from an image using convolution filtering.

This biological function can be approximated in computers using the convolution operation. In image processing, images can be filtered using convolution to produce different visible effects. Figure above shows how a hand-selected convolutional filter detects horizontal edges from an image, functioning similarly to a receptive field.

A set of convolutional filters can be combined to form a convolutional layer of a neural network. The matrix values of the filters are treated as neuron parameters and trained using machine learning. The convolution operation replaces the multiplication operation of a regular neural network layer. Output of the layer is usually described as a volume. The height and width of the volume depend on the dimensions of the activation map. The depth of the volume depends on the number of filters.

Since the same filters are used for all parts of the image, the number of free parameters is reduced drastically compared to a fully-connected neural layer. The neurons of the convolutional layer mostly share the same parameters and are only connected to a local region of the input. Parameter sharing resulting from convolution ensures translation invariance. An alternative way of describing the convolutional layer is to imagine a fully-connected layer with an infinitely strong prior placed on its weights. This prior forces the neurons to share weights at different spatial locations and to have zero weight outside the receptive field.

Successive convolutional layers form a convolutional neural network (CNN). The backpropagation training algorithm, is also applicable to convolutional networks. In theory, the layers closer to the input should learn to recognize low-level features of the image, such as edges and corners, and the layers closer to the output should learn to combine these features to recognize more meaningful shapes.

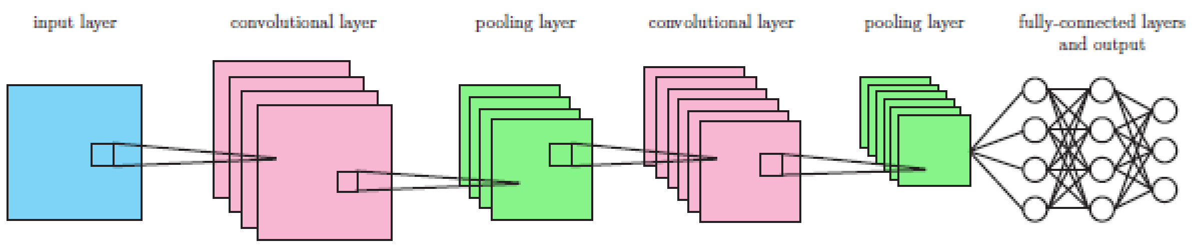

Figure 4.4: An example of a convolutional network.

4.2.3 Pooling and Stride

To make the network more manageable for classification, it is useful to decrease the activation map size in the deep end of the network. Generally the deep layers of the network require less information about exact spatial locations of features, but require more filter matrixes to recognize multiple high-level patterns . By reducing the height and width of the data volume, we can increase the depth of the data volume and keep the computation time at a reasonable level.

There are two ways of reducing the data volume size. One way is to include a pooling layer after a convolutional layer. The layer effectively down-samples the activation maps. Pooling has the added effect of making the resulting network more translation invariant by forcing the detectors to be less precise. However, pooling can destroy information about spatial relationships between sub parts of patterns. Typical pooling method is *max-pooling*. Max-pooling simply outputs the maximum value within a rectangular neighbourhood of the activation map.

Another way of reducing the data volume size is adjusting the *stride* parameter of the convolution operation. The stride parameter controls whether the convolution output is calculated for a neighbourhood centred on every pixel of the input image (stride 1) or for every nth pixel (stride n). Research has shown that pooling layers can often be discarded without loss in accuracy by using convolutional layers with larger stride value. The stride operation is equivalent to using a fixed grid for pooling.

4.2.4 Additional layers

The convolutional layer typically includes a non-linear activation function, such as a rectified linear activation function. Activations are sometimes described as a separate layer between the convolutional layer and the pooling layer.

Some systems, also implement a layer called local response normalization, which is used as a regularization technique. Local response normalization mimics a function of biological neurons called lateral inhibition, which causes excited neurons to decrease the activity of neighbouring neurons.

The final hidden layers of a CNN are typically fully-connected layers. A fully-connected layer can capture some interesting relationships parameter-sharing convolutional layers cannot. However, a fully connected layer requires a sufficiently small data volume size in order to be practical. Pooling and stride settings can be used to reduce the size of the data volume that reaches the fully-connected layers. A convolutional network that does not include any fully-connected layers, is called a *fully convolutional network* (FCN).

If the network is used for classification, it usually includes a softmax output layer. The activations of the topmost layers can also be used directly to generate a feature representation of an image. This means that the convolutional network is used as a large feature detector.

4.2.5 Regularization and data augmentation

Regularization refers to methods that are used to reduce overfitting by introducing additional constraints or information to the machine learning system. A classical way of using regularization in neural networks is adding a penalty term to the objective/loss function that penalizes certain types of weights. The parameter sharing feature of convolutional networks is another example of regularization.

There are several regularization techniques that are specific to deep neural networks. A popular technique called dropout attempts to reduce the co-adaptation of neurons. This is achieved by randomly dropping out neurons during training, meaning that a slightly different neural network is used for each training sample or minibatch. This causes the system not to depend too much on any single neuron or connection and provides an effective yet computationally inexpensive way of implementing regularization. In convolutional networks, dropout is typically used in the final fully-connected layers.

Overfitting can also be reduced by increasing the amount of training data. When it is not possible to acquire more actual samples, data augmentation is used to generate more samples from the existing data. For classification using convolutional networks, this can be achieved by computing transformations of the input images that do not alter the perceived object classes, yet provide additional challenge to the system. The images can be, for example, flipped, rotated or subsampled with different crops and scales. Also, noise can be added to the input images.

Chapter 5

Experimental Setup and Results Analysis

5.1 Dataset

Firstly, for doing this project, We needed to collect images for different classes that can be classified as waste.

Since there was no data set already available for the intended purpose, creation of the database from scratch was the most essential step.

We collected data from a software that scraped images from popular search engines like Google, Bing and Flickr. We collected images of 5 classes as our data set.This Data set had classes namely:

1. **Containers**
   This category contains around 531 images of plastic water,soda and soft drinks bottles, food containers, coloured bottles and medicine containers. 500 pictures are trained using the 31 test pictures.

2. **Organic Waste**
   This category contains 763 images of dried leaves, garden litter and yard waste. 702 images are trained using 61 test images.

3. **Metal**
   This category contains 609 images of spoons, knives, metal bucket, safety pins, nails, nuts and bolts, plates,keys, crushed metal cans,hammer and tins. 562 images are trained using 47 test images.

4. **Paper**
   This category contains 840 images of shredded paper, crumbled papers and newspapers.782 images are trained using 58 test images.

5. **Plastic**
   This category contains 559 images of polythene bags, wrappers and garbage bags. 529 images are trained using 30 test images.

About 500 images of each item were collected. These images include the waste items in different illumination, background colors and various angles. The code to train the CNN trims the number of images in each category to the minimum number images contained in a category. Thus, for each sub-category, equal number of training images is used in the database. Hence, the database created for the application of waste segregation consists of a total of 3,302 real time images along with images from various other sources. Care was taken to include images of objects as they would be when disposed off; e.g., crushed bottles, crumpled paper, etc. to make the training more robust to deal with real-time images.
The data set was divided into 2 folders in the ratio 1:10 for training and testing.It contained images like the following:

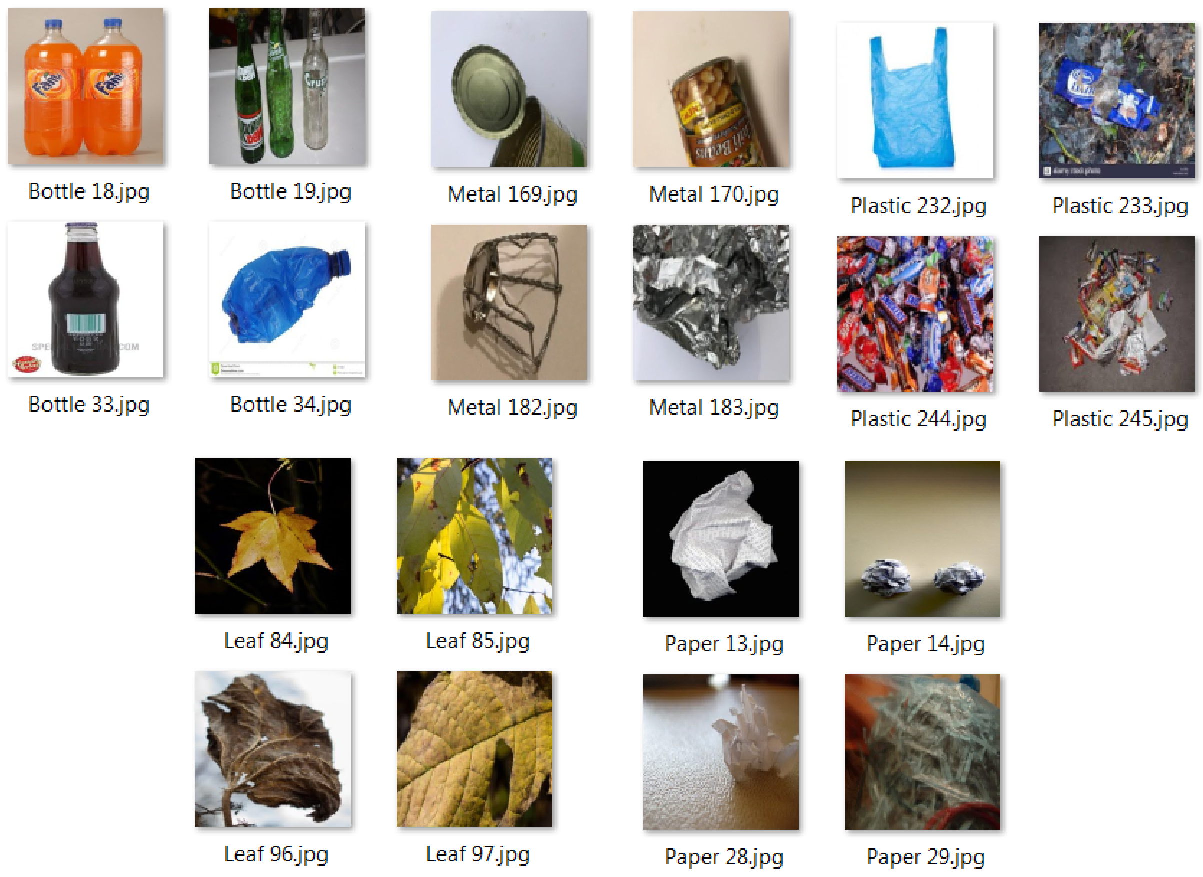

Figure 5.1: Example Data set

5.2 Softwares and Hardwares Used

- Software:

  Python 3

  Tensor-flow

  Keras

  Cuda (NVIDIA Computing Toolkit)

  Matplotlib

  SciPy

- Recommended Hardware:

  NVIDIA GT - 840m and higher

  Intel i5 - 6th gen(Dual Core) and higher

  GPU - VRAM : 4GB and higher

  SSDs

  RAM : At least the same RAM size as your GPU.

5.3 Result Analysis

5.3.1 First Trained Model

A Data set consisting of 2689 images for training and 350 images for testing was taken.

Image Size	No of Classes	Accuracy	Validation Accuracy	Loss	Validation Loss
128x128	4	.9924	.7373	.0379	2.3326

The first Data set contained images with similar background and very similar images for a class which hampered the result.
This model showed a plastic as cardboard as the color yellow was taken as a feature of cardboard and as plastics had white background it could not classify a bottle with yellow background as plastic and classified it as cardboard.
So the trained model was inefficient because of the biased data set.

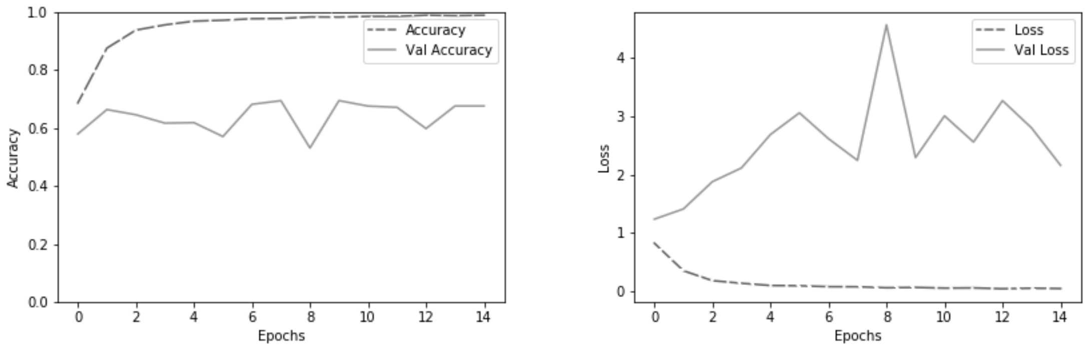

Figure 5.2: Accuracy and Loss of 128x128

A new Data set was collected with bigger image size and greater variety so that the model could predict any new image it was handed with accuracy.

5.3.2 Second Trained Model

The new Data set was used to train a newer model for reducing inaccuracy in the previous model. It contained 3506 images for training and 220 for testing.

Image Size	No of Classes	Accuracy	Validation Accuracy	Loss	Validation Loss
256x256	5	.3927	.3095	10.6877	11.1299

This Model returned absurdly wrong and inaccurate prediction.
We found that the strides that we were taking to make the feature matrix was set to default and was very low. This was making the feature matrix huge and thus the model could not catch features to accurately predict the images.

5.3.3 Final Trained Model

The same above data set was taken.

Image Size	No of Classes	Accuracy	Validation Accuracy	Loss	Validation Loss
256x256	5	.9848	.7456	0.0864	2.1231

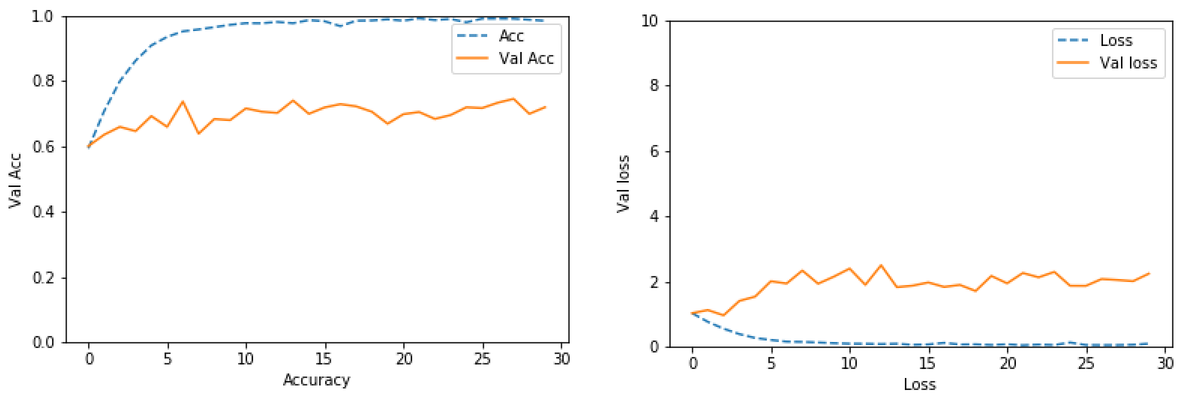

Figure 5.3: Accuracy and Loss of 256x256

After doing hyperparameter tuning, we got our the best result for the following parameters. We used 3303 images of size 256 X 256 splitting them into training and test datasets of ratio roughly 12:1. We used deeper CNN structure than before with following layers.

- Layer 1 - CNN (followed by Batch Normalization and Max Pooling)

  Filters - 96

  Kernel Size - 11 x 11

  Strides - 4

  Activation Function - ReLU

- Layer 2 - CNN (followed by Batch Normalization and Max Pooling)

  Filters - 256

  Kernel Size - 5 x 5

  Activation Function - ReLU

- Layer 3-4 - CNN

  Filters - 384

  Kernel Size - 3 x 3

  Activation Function - ReLU

- Layer 5 - CNN (followed by Max Pooling)

  Filters - 256

  Kernel Size - 3 x 3

  Activation Function - ReLU

- Layer 6 - Flattening
- Layer 7 - Fully Connected Hidden Layer

  Neurons - 2048

  Activation Function - ReLU
- Layer 8 - Fully Connected Hidden Layer

  Neurons - 2048

  Activation Function - ReLU
- Layer 9 - Output Layer

  Neurons - 5

  Activation Function - Softmax

We used Adam Optimizer for minimizing the Categorical Cross-entropy Loss. The model was trained for 30 epochs with batch size of 16.

Chapter 6

Conclusion and Final Results

In this project, an automated Waste Segregator System using one of the Machine Learning tools,Convolution Neural Networks, was designed and executed. The Waste Segregator System is devised to achieve segregation of waste thereby reducing human intervention in the handling of waste items. The model gave maximum validation accuracy of 74% with 98.48% accuracy on the training set. The model is giving good results on images with aspect ratio close to 1:1. We used our model for prediction of images in Figure 6.1.

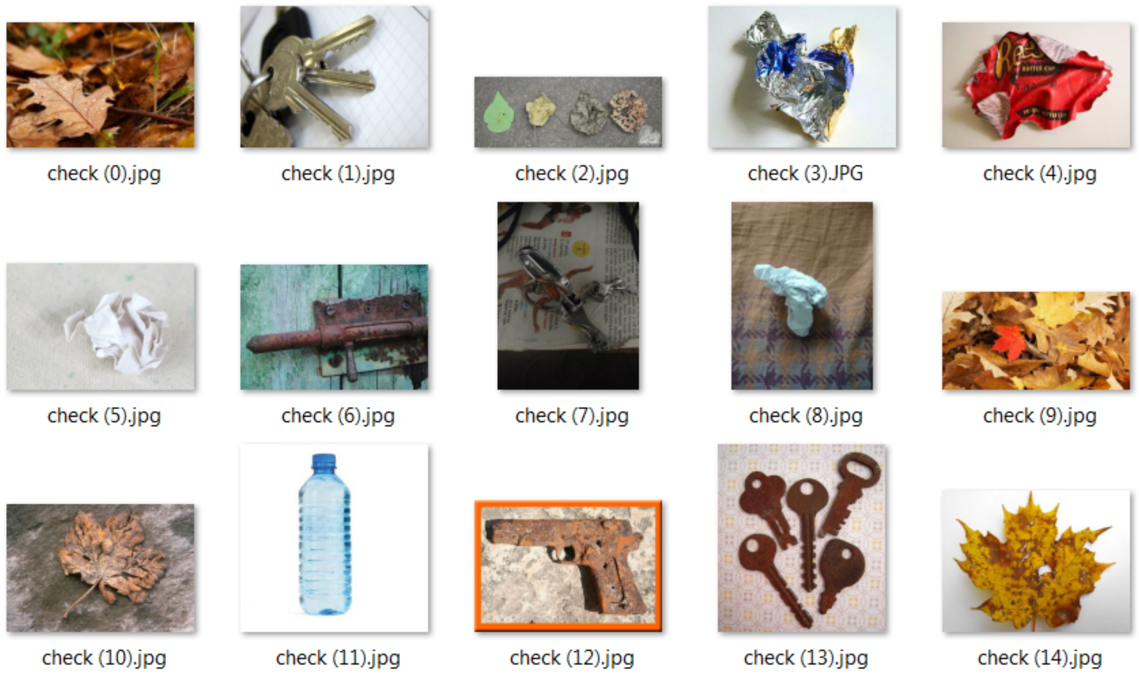

Figure 6.1: Data set where Prediction was performed

The model correctly classified 11/14 images. It could not classify images 2,6 and 12 correctly. The expansion in the waste categories' database will help in increasing the accuracy rate when training the network for the classification purpose. Moreover, the image dataset can be location specific to adapt better to a certain environment. The model will also get better when deployed as it can simultaneously learn new readings hence improving the model.

Chapter 7

Future Works

1. *Localization* — In addition to Classification of Waste, we can use techniques such as Bounding Box Localization, Instance Segmentation etc. to locate the classified object in the image.

2. *Generalization* — Rather than constricting our search among 5 classes we can use pre-trained models (like VGG-16) to generalize this approach over all kinds of waste. The model can be fine-tuned to fit our requirements and provide wider and better results.

3. *Automation* — Build a more sophisticated mechanical system to automatically separate and segregate the cluster of wastes.

4. Increasing the number of categories to which the waste items can be classified into with the expansion in the database.

5. Improving the accuracy rates of the system by adding more training and test samples to the existing dataset. (5) (4) (3) (1) (2)

Bibliography

[1] Artificial neural networks by udemy. `https://www.udemy.com/deeplearning/`.

[2] Convolutional neural networks by udemy. `https://www.udemy.com/deep-learning-convolutional-neural-networks-theano-tensorflow/`.

[3] Marko, J. Waste sorting using neural networks, Fall 2016.

[4] Stenroos, O. Object detection from images using convolutional neural networks. Master's thesis, Aalto University, July 2017.

[5] Yoshua Bengio, Ian Goodfellow, A. C. *Deep Learning.* MIT Press.