TRASH AND RECYCLABLE MATERIAL IDENTIFICATION USING CONVOLUTIONAL NEURAL NETWORKS (CNN)

A thesis proposal to the faculty of the Graduate School of
Western Carolina University in partial fulfillment of the requirements for the degree of Master of Science in Technology

By
Rumana Sultana

Director: Dr. Robert Adams
Associate Professor
School of Engineering and Technology

Committee Members:
Dr. Martin Tanaka, School of Engineering and Technology
Dr. Paul Yanik, School of Engineering and Technology
Dr. Yanjun Yan, School of Engineering and Technology

April 2020

ACKNOWLEDGMENTS

I would like to express my gratitude to my thesis director, Dr. Robert Adams for his valuable instructions, suggestions, and engagement through the learning process of this master thesis. Furthermore, I would like to thank my committee members, Dr. Yanjun Yan, Dr. Paul Yanik, and Dr. Martin Tanaka for their assistance and encouragement.

Lastly, I offer my warmest regards and thanks to my all family members for their continued support.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

ABSTRACT

TRASH AND RECYCLABLE MATERIAL IDENTIFICATION USING CONVOLUTIONAL NEURAL NETWORKS (CNN)

Rumana Sultana

Western Carolina University (April 2020)

Director: Dr. Robert Adams

The aim of this research is to improve municipal trash collection using image processing algorithms and deep learning technologies for detecting trash in public spaces. This research will help to improve trash management systems and create a smart city. Two Convolutional Neural Networks (CNN), both based on the AlexNet network architecture, were developed to search for trash objects in an image and separate recyclable items from the landfill trash objects, respectively. The two-stage CNN system was first trained and tested on the benchmark TrashNet indoor image dataset and achieved great performance to prove the concept. Then the system was trained and tested on outdoor images taken by the authors in the intended usage environment. Using the outdoor image dataset, the first CNN achieved a preliminary 93.6% accuracy to identify trash and non-trash items on an image database of assorted trash items. A second CNN was then trained to distinguish trash that will go to a landfill from the recyclable items with an accuracy ranging from 89.7% to 93.4% and overall, 92%. A future goal is to integrate this image processing-based trash identification system in a smart trashcan robot with a camera to take real-time photos that can detect and collect the trash all around it.

CHAPTER ONE: INTRODUCTION

A city is best loved by people who live in it when it is healthy and hygienic. But in the era with a growing population, the people are attracted to the city area, and hence it is very difficult to maintain the cleanliness of a city. If we look towards the south Asian countries, we can easily understand how challenging it is, as the increasing number of residents of a city increases the production of trash. Though first world countries have a well-established trash management system, most of the developing countries manage trash very poorly. That is why trash management has been a crucial issue to consider. Overflowing of trash bins is a common scenario in most of the developing countries. Also, there is a tendency among people of these countries to dump the trash not inside the trashcan, but outside the can. This is a very unhygienic and awkward condition. There is no doubt that the surrounding area of the trashcan become a breeding place for germs. Passing beside a roadside trash bin in that situation is obviously not a good experience for people, especially for the newcomers, kids, senior citizens, etc. So, it is clear that uncollected trash in the developing countries and the litter along highways or other areas in the developed countries pose a serious problem for residents in terms of hygiene, neighborhood appeal, and environmental protection. The World Health Organization [1] has reported that "A significant amount of disease could be prevented through access to safe water supply, adequate sanitation services and better hygiene practices. Diarrheal disease alone amounts to an estimated 3.6 % of the total daily global burden of disease and is responsible for the deaths of 1.5 million people every year (WHO 2012). It is estimated that 58% of that burden, or 842,000 deaths per year, is attributable to unsafe water supply, sanitation and hygiene and includes 361,000 deaths of children under age five, mostly in low-income countries (WHO 2014)". Hence, considering the health issue associated with trash management is a very significant topic nowadays. There is also a need for reducing the high cost of trash collection. For example, CBSNewYork [2] published that New York city pays $300 million per year for collecting trash. It is a large amount for most of the developing countries. So, how could the problem be solved? Is it possible to change people's mind in a short time? No. But, implementing some new ideas to modify the condition could be possible.

One possibility to address these issues is to implement automatic systems for the collection and disposal of trash in public spaces. These systems help 1) reduce the cost of trash collection, 2) attract more tourism, and 3) improve the hygiene of citizens, not only for 3rd world countries but also for developed countries.

We are passionate about eventually building a smart trashcan that will detect and pick up the trash outside of the can using a retractable vacuum arm, monitor the trash level within the can with a built-in compressor plate to pack the trash, and alert the trash collection personnel when the trash level is high. We really hope to implement such a smart trashcan system to improve our living environment. However, we understand that this entire project would require a suite of expertise, ranging from image processing, vision analysis, motor control, manufacturing and possibly 3D printing, wireless communication, battery charging mechanism, and database management. These topics are way beyond the scope of this thesis.

In this research effort we focused on developing the ability of the smart trashcan robot to distinguish between items in an outdoor public area that should be collected and those that should not. Once this determination has been made, we then focused on the ability to distinguish between trash items and recyclable items. In both of these research efforts, we trained and tested convolutional neural networks using deep learning methods to test recognition accuracy.

The goal of this research is to improve the trash management system through trash detection beside trashcan by an attached digital camera and convolutional neural network architecture for object detection and recognition from image and implement an intelligent trash collecting system that can help to make a smart city. Finally, it can improve the environmental and ecological state of the society.

CHAPTER TWO: LITERATURE REVIEW

2.1 Research Background

Recent progress in deep learning research has contributed greatly to unparalleled improvements in computer vision. Convolutional neural networks (CNN) are one of the most powerful deep-learning algorithms, which has many applications in image classification, segmentation, and detection [3-6]. Therefore, in this paper CNN is proposed to perform trash detection and recognition.

Chu et al. [7] proposed a multilayer hybrid deep-learning system (MHS) that can sort trash disposed of by individuals in an urban public area. The system can automatically sort trash items as recyclable or otherwise. They used the AlexNet CNN [3] to extract key image features and optical sensors to detect other numerical feature information. This system used multilayer perceptrons (MLP) to classify the trash object by consolidating information collected from diverse channels. The proposed MHS achieved a mean accuracy higher than 90%, but the system can classify only 22 fixed items of trash in public areas. Other trash items on the road or in a park would not be counted in their system.

Bai et al. [8] presented a garbage pickup robot which can detect trash accurately and autonomously on the grass. They used a deep neural network, ResNet [9], for trash recognition and a navigation strategy to guide the robot to move around. With the trash recognition and automatic navigation functions, the robot can clean trash on the ground in parks or schools automatically. Their trash recognition accuracy reached above 95%. But the robot can detect trash only on grass. So trash on the road or in parking areas could not be identified by the robot.

The above two research efforts achieved very high accuracy in using CNN architectures. Based on these works, we are proposing a system that can identify trash items from any public space such as along a road, in a parking lot, in a recreation area or park, a community space, etc. Our ultimate goal is to build a trash collecting robot that self-navigates in a park or public space, looking for objects on the ground. This thesis is the first step towards that goal, to identify the objects from images.

2.2 Existing Technologies

Based on other researchers' previous work, we are going to use a convolutional neural network (CNN) architecture for detecting trash items from non-trash items as well as trash from recyclable items. The CNN is a concept from deep learning technology that is widely used nowadays for image and video processing, pattern recognition, object detection and object recognition from digital images. A brief summary of current research efforts in digital image and video processing, deep learning and CNNs is provided below.

2.2.1 Digital Image and Video Processing

Digital image processing involves the use of computer algorithms for the processing of digital images. It is a practical solution for image classification, feature extraction, multi-scale signal analysis, pattern recognition and projection. Digital image processing is a widely used technology for medical diagnosis and treatment, machine/robot vision, image transmission and encoding, remote sensing, and many other relevant areas.

An image refers to a 2D light intensity function $f(x, y)$, where (x, y) denotes spatial coordinates [10]. The value of $f(x, y)$ is proportional to the brightness or gray levels of the image at the point (x, y). A digital image can be represented by a light intensity function $f(x, y)$, that has been discretized both in spatial coordinates and in brightness into a 2D array of intensities. The elements of such a digital array are called picture elements or pixels. To be suitable for computer processing, an image $f(x, y)$ must be digitalized both spatially and in amplitude. Digitization of the spatial coordinates (x, y) is called image sampling. Amplitude digitization is called gray-level quantization. Figure 2.1 shows examples of digital images that have been discretized at different spatial resolutions. Figure 2.2 shows examples of digital images that have been discretized with 256 down to 2 levels of amplitude quantization.

Digital image transformations [11] are used to blur and sharpen digital images by spatial low pass, spatial high pass, Fourier representation, Fourier low pass and Fourier high pass filtering. Affine image transformation is a general term for transformations that preserve spatial linearity and include scaling, rotation, translation, mirroring, and shearing.

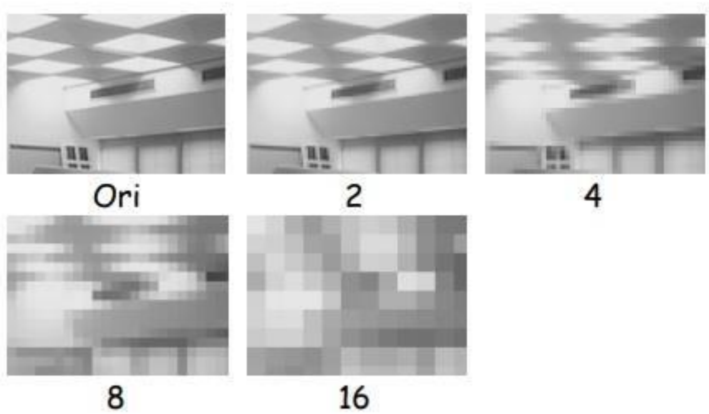


Figure 2.1: Images of different spatial resolution
[Source: http://www.eie.polyu.edu.hk/~enyhchan/DIP-F.pdf]

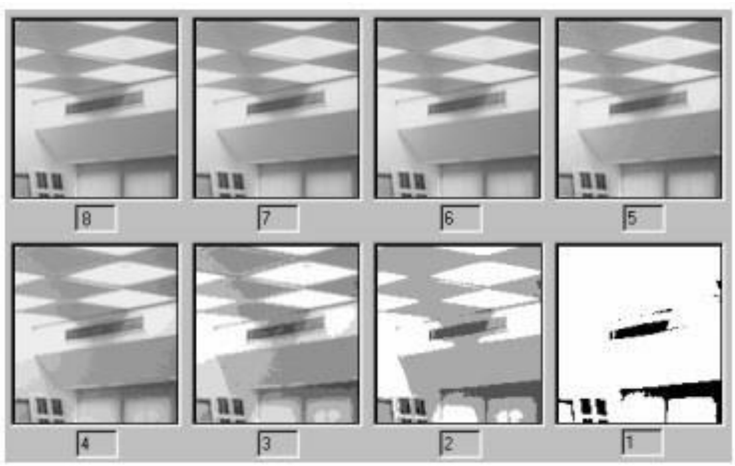


Figure 2.2: Images of different amplitude resolution
[Source: http://www.eie.polyu.edu.hk/~enyhchan/DIP-F.pdf]

Fundamental steps in the use of image processing techniques for object recognition include image acquisition, image preprocessing, image segmentation, feature extraction, and object recognition. Image acquisition refers to the method use to capture a digital image. Image preprocessing refers to the method used to convert the digital image into a format that is usable for computer algorithm processing. For example, many algorithms require a fixed image size and image type for the processing of images. Image segmentation involves segmenting a digital image into usable segments. Feature extraction involves identifying the presence of certain features within an image. For example, if one were searching for a wall clock in a room, one may search for round objects that resemble a clock. Object recognition involves finding the best match for a predetermined object within an image.

2.2.2 Deep Learning

Deep learning [12] is a type of machine learning that is able to classify images, data or sound into predetermined categories. Deep learning is usually implemented using a neural network architecture. The term "deep" refers to the number of layers in the network—the more layers, the deeper the network. Deep learning networks typically have anywhere from ten to several hundred layers. Applications of deep learning include facial recognition, optical character recognition (OCR), speaker identification, automotive assistive devices, and object recognition.

In a word, accuracy is the most important factor in assessing deep learning. Figure 2.3 shows the success story of image classification techniques over time after introducing the AlexNet architecture on the ImageNet challenge from 2012. From this bar graph one can see that deep learning techniques have surpassed human ability to classify images by ResNet-152 in 2015 with 3.57% error rate whereas human error was 5% for this task.

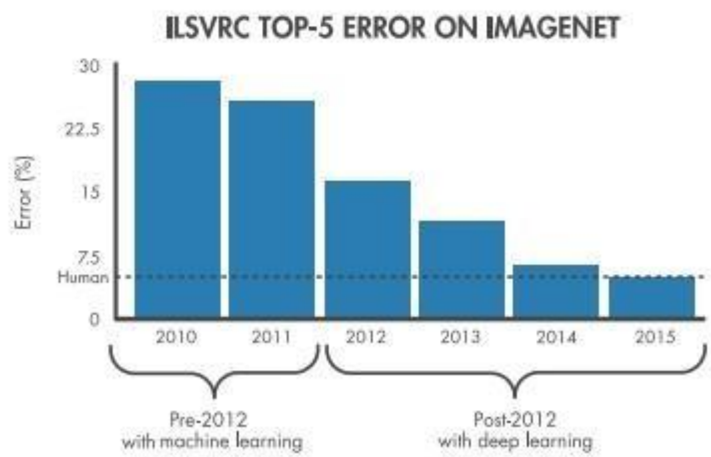


Figure 2.3 Error reduction comparison on ImageNet
[Source:
https://www.mathworks.com/content/dam/mathworks/ebook/gated/80879v00_Deep_Learning_ebook.pdf]

Another important factor of deep learning is to be able to handle the big amount of image dataset. The graph shown in Figure 2.4 clearly indicates that the performance of traditional machine learning approaches [13] is able to show a better performance than the deep learning approaches for smaller amounts of input data only. When the amount of data rises beyond a certain amount, the performance of traditional machine learning approaches becomes slower than the deep learning approaches or plateaued. On the other hand, the performance of deep learning approaches kept on improving when the amount of data increased.

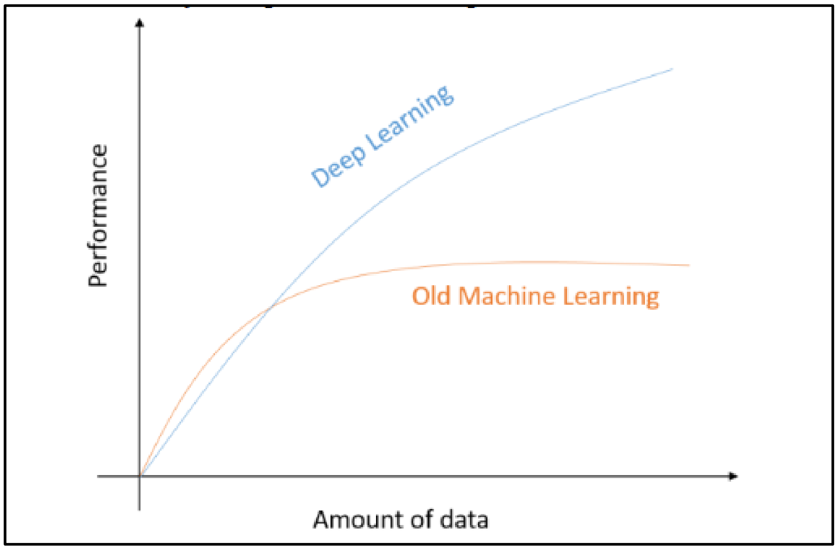


Figure 2.4: The comparison between the performance of deep learning and machine learning to handle large amount of data
[Source: https://arxiv.org/ftp/arxiv/papers/1803/1803.01164.pdf]

Deep learning neural networks utilize multiple layers with various processes between each layer. As shown in Figure 2.5, all deep learning neural networks have an input layer, several hidden layers, and an output layer. The layers are interconnected via nodes, or neurons, with each hidden layer using the output of the previous layer as its input.

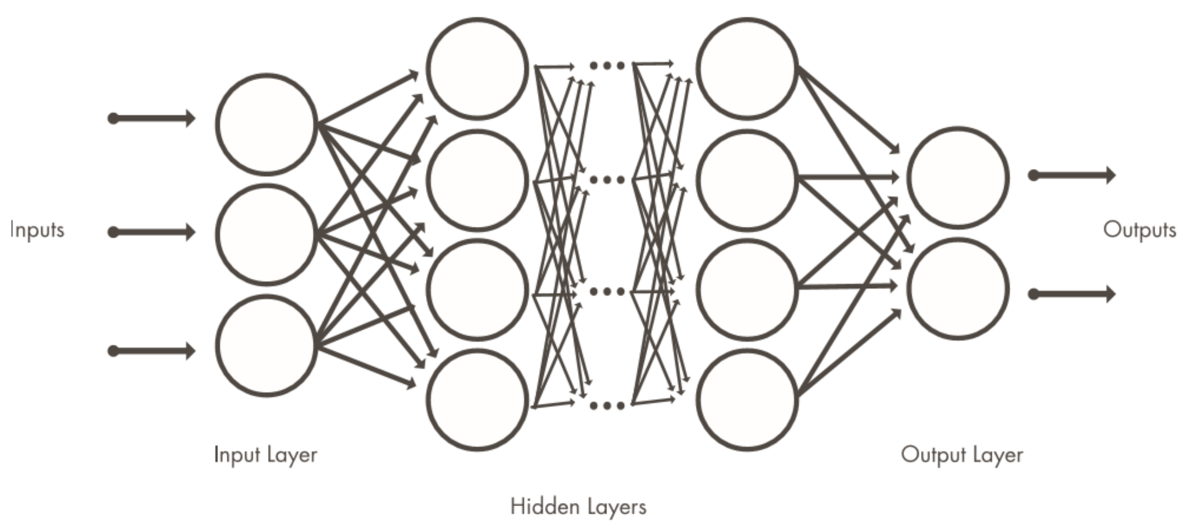


Figure 2.5: Deep neural network
[Source: MathWorks (https://goo.gl/zondfq)]

To automatically recognize which object is in each image from a set of images where each image contains one of several different categories of objects by using the deep learning network, it is necessary to label the images to have a training data set for the network. Using this training data, the network can then start to learn the object's specific features and associate them with the corresponding category. Each layer in the network takes in data from the previous layer, transforms it, and passes it on. The network increases the complexity and detail of what it is learning from layer to layer. The network learns directly from the data— without any influence from humans on what features are being learned. Figure 2.6 shows an example of the image classification process of a deep neural network. In this example the neural network has been trained to classify into four object categories of flower, cup, car, and tree. A flower is the input image. The output is a 1×4 array of likelihoods that the input image matches the given category. If the neural network is correct in this example, then the output likelihood of the flower category will exceed that of all other categories.

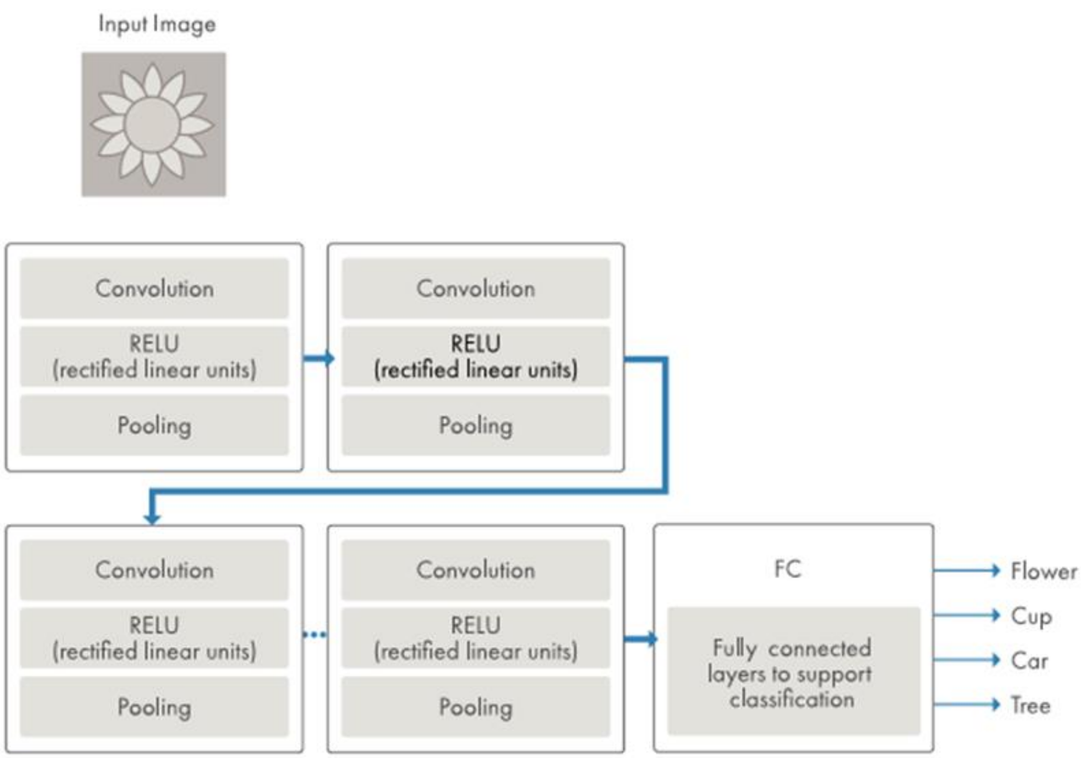


Figure 2.6: Image Classification by deep neural network
[Source: https://www.mathworks.com/content/dam/mathworks/ebook/gated/80879v00_Deep_Learning_ebook.pdf]

2.2.3 Convolutional neural network

A convolutional neural network (CNN) is a deep learning neural network that takes an image as the input, then assigns weights and biases to various features in the image. Through a learning process the weights and biases are refined and sections of the image are further processed. The CNN eventually differentiates features within the image from one another. Convolutional neural networks have been used by other researchers to analyze digital images for object recognition or classification [3-6]. Instead of primitive methods such as filtering by a hand-engineered process, after proper training, CNNs can filter/categorize images into different classes based on the input-output pairs. The architecture of a CNN is similar to the internal connecting system of neurons in the human brain. It is a combination of convolutional layers, pooling layers, fully connected layers, and normalization layers [3-6]. In the convolutional layers, each kernel generates a feature map by convolving the input image with moving kernels with certain window size and stride size. The ReLU (rectifier linear unit) is applied on the output to avoid gradient vanishing, and then pooling is applied to reduce noise and feature dimensions. After multiple convolutional layers and pooling, the features are then flattened to be fed into the fully connected layers, where each layer consists of sets of nodes (artificial neurons) in columns, and the output of every node (activation neuron) of a layer is mapped to the input of all nodes in the next layer.

CNNs use different multilayer perceptrons designed to minimize required preprocessing steps [14]. They have shared-weights architecture and translation invariance characteristics for which they are also known as shift invariant or space invariant artificial neural networks (SIANN). The applications of CNNs are image and video recognition, recommender systems, image classification, medical image analysis, and natural language processing.

A convolutional neural network [12] consists of an input and an output layer, as well as multiple hidden layers, as shown in Figures 2.7 and 2.8. The hidden layers of a CNN typically consist of convolutional layers, RELU layer i.e. activation function, pooling layers, fully connected layers, and normalization layers.

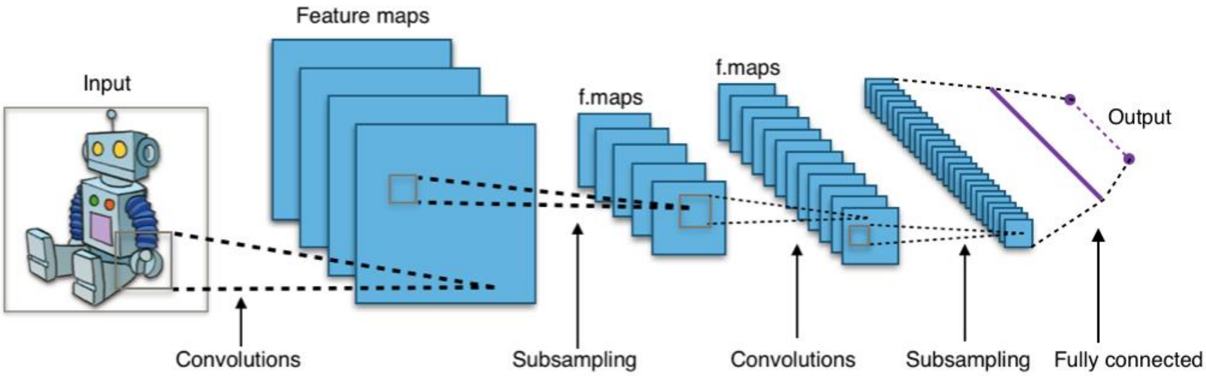


Figure 2.7: Design of a Convolutional Neural Network
[Source: Wikipedia (https://en.wikipedia.org/wiki/Convolutional_neural_network)

Feature Detection Layers perform one of three types of operations on the data: convolution, pooling, or rectified linear unit (ReLU). **Convolution** puts the input images through a set of convolutional filters, each of which activates certain features from the images. **Pooling** simplifies the output by performing nonlinear down sampling, reducing the number of parameters that the network needs to learn about. **Rectified Linear Unit (ReLU)** allows for faster and more effective training by mapping negative values to zero and maintaining positive values. These three operations (convolution, pooling, and ReLU) are repeated over tens or hundreds of layers, with each layer learning to detect different features.

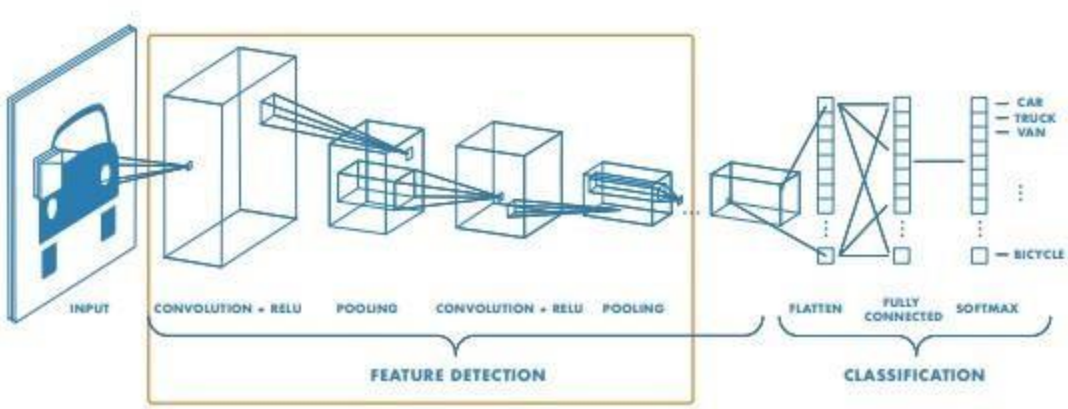


Figure 2.8: Feature Detection and Classification Layers in CNN
[Source: MathWorks (https://goo.gl/zondfq)]

Classification Layers shifts the architecture of a CNN to classification after feature detection. The **Fully Connected Layers (FC)** are essentially the traditional neural network is trained by backpropagation, and the last layer outputs a vector of K dimensions where K is the number of classes that the network will be able to predict. This vector contains the probabilities for each class of any image being classified. The final layer of the CNN architecture uses a **softmax** function to provide the classification output.

In mathematical form, the fully connected function can be expressed by the following equations of a forward pass and backward pass propagation rules [15]. The components of input vector **x** are the outputs of layer 1 that can be expressed as $a_j(1) = x_j$, where *j*=1, 2, 3….,n_1 and $n_1 = n$ is the dimensionality of **x**. The computation performed by neuron *i* in layer *l* is given by

$$z_i(l) = \sum_{j=1}^{n_{l-1}} W_{ij}(l) a_j\,(l-1) + b_j(l) \tag{2.1}$$

where *i*=1,2, 3,…., n_l with n_l being the number of neurons at layer *l* and *l*=2,……, *L* with *L* being the total number of layers, $W_{ij}(l)$ represents the weight between the j^{th} neuron in layer *l-1* and the i^{th} neuron in layer *l*, and $z_i(l)$ represent the net input to neuron *i* in the layer *l*, which is formed using all outputs from layer *l*-1. $b_j(l)$ is the bias value associated with the j^{th} neuron in the l^{th} layer. The activation value of neuron *i* in layer *l* is given by $a_i(l) = h(z_i(l))$ for *i* =1,2,3,….., n_l where *h* is an activation function. The value of network output node *i* is $a_i(L) = h(z_i(L))$ for *i*=1,2,3,…..,n_L. These are all the operations required to map the input of a fully connected feedforward network to its output. The relationship between the net input and the output of any neuron in any layer (except the first layer) is the same that could be denoted by $\delta_j(l)$ for any node *j* in any hidden layer *l*. $\delta_j(l)$ can be expressed as

$$\delta_j(l) \; = \frac{\partial E_j}{\partial z_j(l)} \tag{2.2}$$

where E_j is the error of the j^{th} neuron. As we will be proceeding backward in the network, we need the relationship between $\delta_j(l)$ and $\delta_j(l+1)$ so that we can start with $\delta_j(L)$ and find $\delta_j(L-1)$, $\delta_j(L-2)$, and finally reach at layer 2. The equation can be expressed by

$$\delta_j(l) = h'\left(z_j(l)\right) \textstyle\sum_i W_{ij}(l+1)\, \delta_i(l+1) \tag{2.3}$$

where $h(z)$ is the activation function given by

$$h(z) = \frac{1}{1 - e^{-z}} \tag{2.4}$$

and $h'(z)$ is the derivative given by

$$h'(z) = \frac{\partial h}{\partial z} = h(z)[1 - h(z)] \tag{2.5}$$

Through some algebra it can be shown that the rate of change of error with respect to network weights is:

$$\frac{\partial E_j}{\partial w_{ij}(l)} = a_j(l-1)\delta_i(l) \tag{2.6}$$

Similarly, the rate of change of error with respect to biases is:

$$\frac{\partial E_j}{\partial b_i(l)} = \delta_i(l) \tag{2.7}$$

These rates of change are used in the backpropagation model to update the weights and biases:

$$w_{ij}(l) = w_{ij}(l) - \alpha a_j(l-1)\delta_i(l) \tag{2.8}$$

$$b_i(l) = b_i(l) - \alpha\delta_i(l) \tag{2.9}$$

where the parameter α is the user-selectable learning rate constant. It is typically set to a small value during training process. The steepest descent method uses α in the process of seeking a path toward minimizing the error. The eventual local minimum obtained by the steepest descent method depends on the initial values of the weights. Figure 2.9 illustrates a situation in which the steepest descent method identified one of two local minima, depending on the initial values of two weights.

The accuracy of CNN can be enhanced by fine tuning dimensional parameters and local architecture structure [8]. Various CNN architectures of different variations have emerged in recent years [3]. AlexNet [3] is used in this work due to its in-field processing capabilities and low computational cost. AlexNet [3] was introduced in the 2012 ImageNet Challenge (ILSVRC). It reduced the image classification top-5 error from 26% to 15.3% significantly. It is well established for its highly capable architecture.

2.3 The scope of this thesis

Finally, the primary goal of the research work is to establish a methodology to design an intelligent trashcan that can take surrounding images and detect the trash from the images. In our proposed design, we will train a convolutional neural network for selecting trash items. With this trained system we will find out the trash item from an input image by processing the image through three algorithms. The first algorithm is object detection, the second one is a trash recognition algorithm, and a third is a

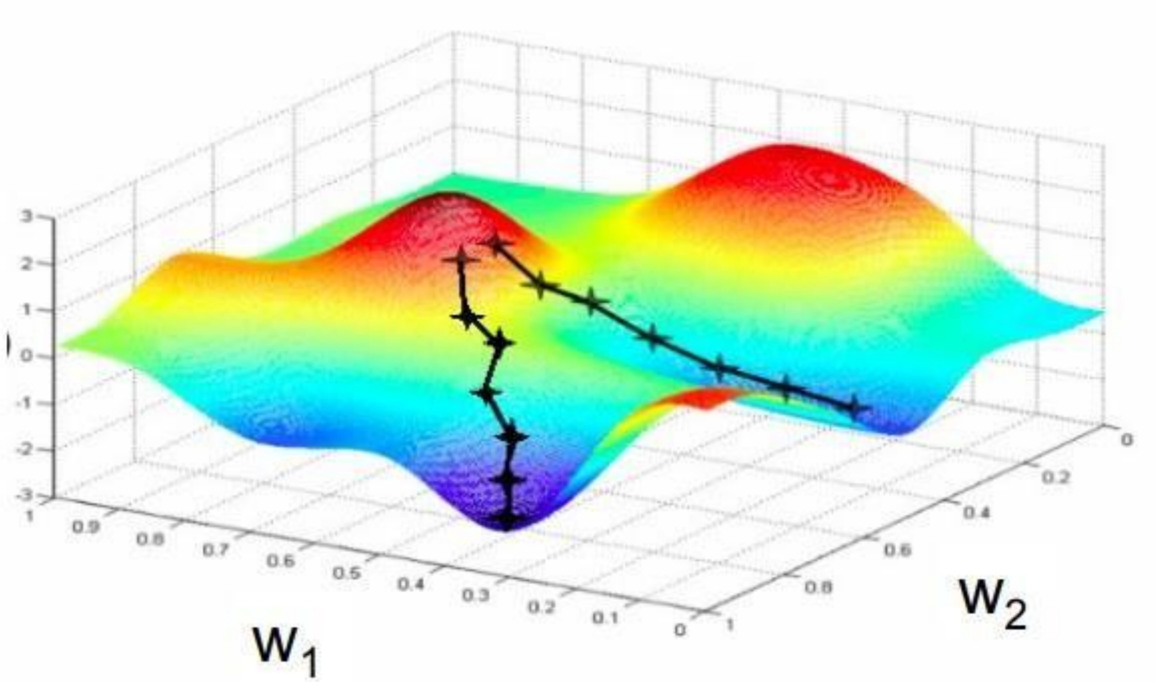


Figure 2.9: Example of gradient descent of weights
[Source: http://dsdeepdive.blogspot.com/2015/07/gradient-descent-with-python.html/]

detection of trash or recyclable item. This thesis focused on the second and third algorithms. Prototyping is beyond the scope of this thesis. Future work will involve design and building a robot to pick up the trash and sort it according to trash or recyclable items.

CHAPTER THREE: KEY METHODOLOGY

3.1 Structure of the trashcan

A smart trashcan, depicted in Figure 3.1, is proposed to detect and collect trash and recyclable items in public spaces. The proposed smart trashcan will be able to recognize trash and other obstacles in images that are captured by the camera. CCD cameras have wide-dynamic-range sensitivity for the change of ambient light. The trashcan robot will employ a CNN algorithm to improve the image comprehension and object recognition accuracy.

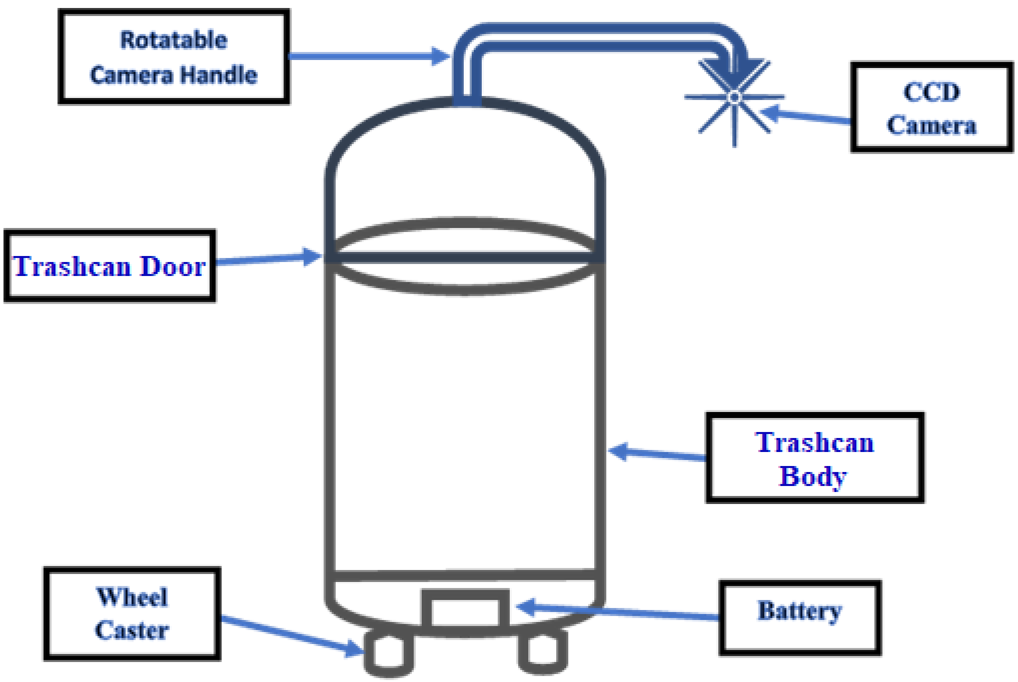


Figure 3.1: Structure of proposed trashcan

3.2 Hardware

3.2.1 Proposed hardware

Figure 3.1 shows a diagram of the proposed system with:

- A high-resolution CCD (Charge-coupled device) camera,
- Motorized rotating camera handle,
- Wheel caster,
- Battery, and
- Central processing unit (CPU).

The camera will capture images of floor objects, and the images will be transferred to processing unit to process by the CNN algorithm through an established computer interfaced between camera and CPU. The camera is placed at the top of the trashcan with a rotatable camera mount to maximize the marginal angle of view. Camera handle can be rotated for the camera to capture views from different angles on different sides. Figure 3.2 shows an example of how a CNN algorithm can use a bounding box method for identifying objects.

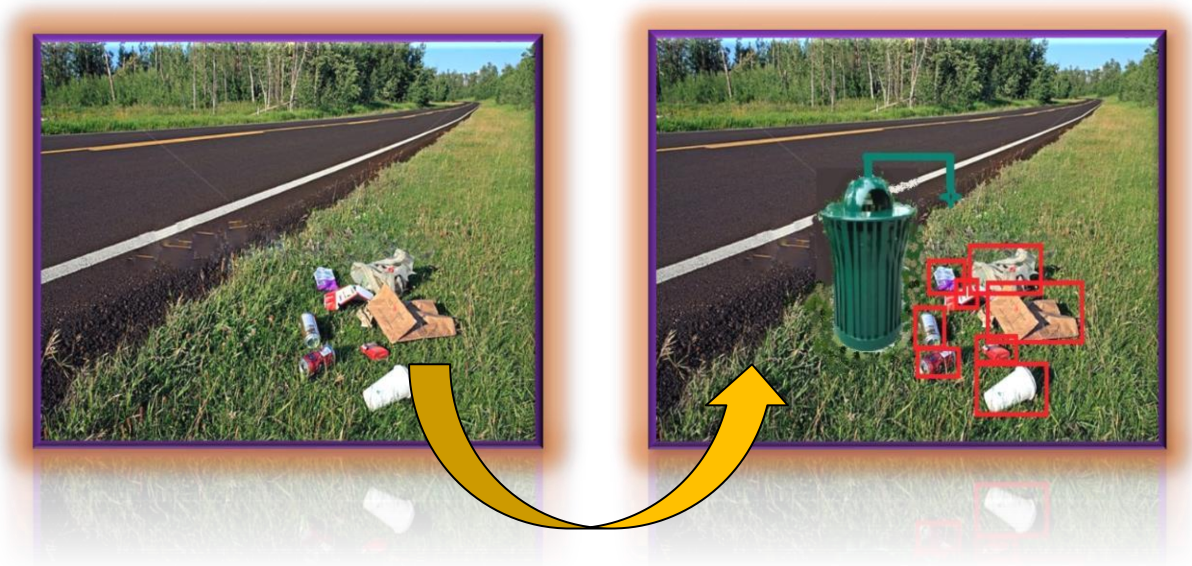


Figure 3.2: Example of Trash Detection
[Source: https://www.alamy.com/stock-photo-roadside-garbage-25647833.html]

3.2.2 Important hardware required for the thesis

In this thesis, we used a high-definition (HD) 1080p Logitech camera (C922 Pro Stream Webcam) to take image in real time testing in indoor environment shown in Figure 3.3. The Logitech C922 Pro Stream Webcam [18-20] is a full HD webcam that offers 1080p at 30 fps, 720p at 60 fps and 30 fps streaming capability. The features included full HD glass lens, dual microphone, activity light, flexible clip/base, and tripod attachment. Technical specifications are shown in Table 3.1.

We replaced our existing **graphics card** with NVIDIA® QUADRO K420 [21] shown in Figure 3.4 to meet the minimum requirement of running both graphics rendering and CUDA computations process on our office desktop. This GPU has a compute capability of 3.0 or higher that is the mandatory requirement to run a deep learning algorithm. The GPU has some advanced features including

DisplayPort 1.2 Connector, DisplayPort with Audio, DVI-I Dual-Link Connector, VGA Support, NVIDIA nView™ Desktop Management Software Compatibility, HDCP Support and NVIDIA Mosaic. To make the execution time of the training process faster, we extended the random-access memory (RAM) of our desktop to 16 GB.

Figure 3.3: Used camera for testing
[Source: https://images-na.ssl-images-amazon.com/images/I/B1nMdnP1ZUS.pdf]

Table 3.1: Hardware specification of Logitech C922 Pro Stream Webcam

Maximum Resolution:	Maximum frame rate is 1080p/30 fps - 720p/60 fps
Focus type	autofocus
Lens and Sensor Technology	Full HD Glass Lens
Dimensions	1.73 in (44 mm) x 3.74 in (95 mm) x 2.80 in (71 mm) including clip
Microphone Type	Built-in Dual Stereo
Recording	1080p 30 fps, 720p 60 fps, 720p 30 fps
Diagonal Field of View	Horizontally 78°
USB Protocol	2.0 serial
Image Capture (4:3 SD)	Yes
Image Capture (16:9 W)	Yes
Video Capture (4:3 SD)	Yes
Video Capture (16:3 9)	Yes

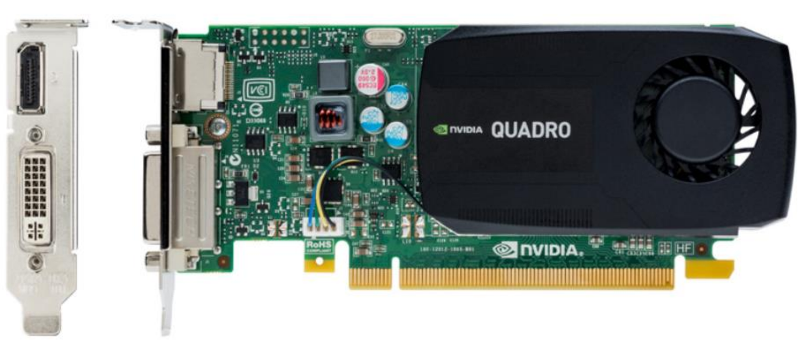


Figure 3.4: GPU with compute capability 3.0 or higher
[Source: https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/quadro-product-literature/13720_DS_NV_Quadro_K420_Aug25_US_NV_HR.pdf]

3.3 Decision Making Process

Figure 3.5 displays a flow chart of the decision-making process. People passing-by may throw objects into the trash robot, in which case the object will be classified as trash or recyclable and stored into separate inner bins. In addition to having a trash receptacle, the robot will be equipped with a camera to capture images and decide whether to take an object or not. It will pick up any object that it perceives to be either trash or recyclable. After grabbing an object, the trash collecting robot would then bring the object inside itself to examine it more closely. With a clearer image of the object, the robot would then classify the object into one of two categories: trash or recyclable. The robot will classify recyclable items as metal, plastic, glass, or fiber. Fiber includes any paper or cardboard item. In this research effort, we trained the AlexNet CNN [3] to classify images firstly as either "take" or "non-take" and secondly as either trash or recyclable. Testing of these CNN's with real outdoor images in public spaces produced quite accurate results.

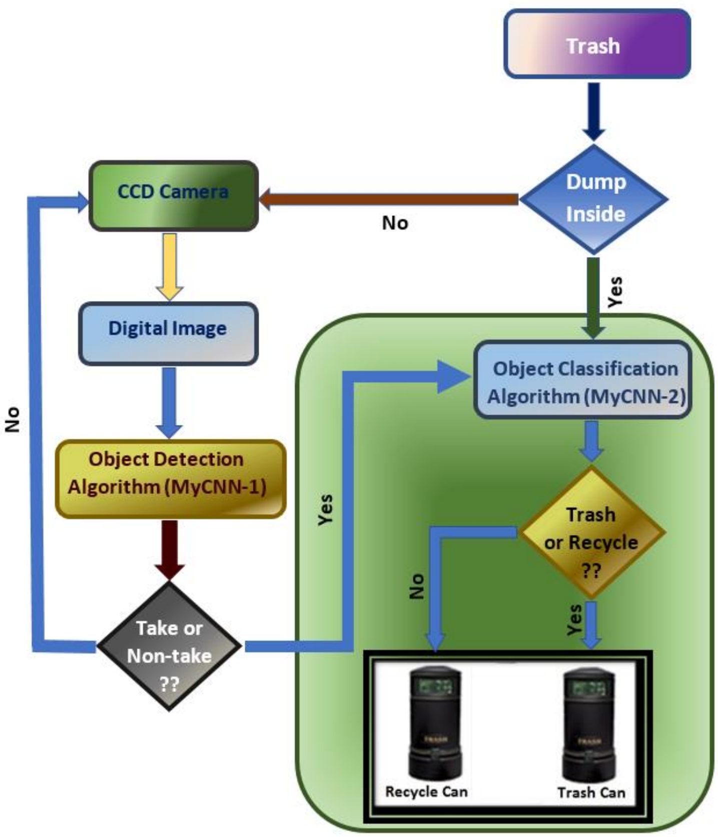


Figure 3.5: Flow Chart of the Decision-Making Process

3.4. AlexNet CNN Architecture

In this study, we used the MATLAB® version of AlexNet that consists of 25 layers including 5 Convolutional Layers and 3 Fully Connected Layers as shown in Table 3.2. Multiple Convolutional Kernels are used to extract required features in an image. Many kernels of the same size are used in a single convolutional layer. The output of the last Fully Connected Layer is fed into the 1000-way softmax function corresponding to 1000 class labels. So, the classification output is 1×1000 array of numerical weights of the probability of a match. Cross-Channel Normalization Layer is associated with the fourth and eighth layers. Max-pooling Layers are placed after the Cross-Channel Normalization Layers and the sixteenth layer. The ReLU nonlinearity is used after each convolutional layer. The neurons in the Fully Connected Layers are connected to all neurons in the previous layer, with 4096 neurons each [3]. The

Table 3.2: MATLAB AlexNet Architecture

Layer	Type
1	Data (227x227x3 Size Images)
2	96 kernels of size 11x11x3 Convolutions
3	ReLU
4	Cross Channel Normalization
5	3x3 Max Pooling
6	256 kernels of size 5x5x48 Convolutions
7	ReLU
8	Cross Channel Normalization
9	3x3 Max Pooling
10	384 kernels of size 3x3x256 Convolutions
11	ReLU
12	384 kernels of size 3x3x192 Convolutions
13	ReLU
14	256 kernels of size 3x3x192 Convolutions
15	ReLU
16	3x3 Max Pooling
17	4096 Fully Connected Layer
18	ReLU
19	50% Dropout
20	4096 Fully Connected Layer
21	ReLU
22	50% Dropout
23	1000 Fully Connected Layer
24	Softmax
25	Classification Output

number of the neurons in the last Fully Connected Layer (layer 23) is set to the number of categories (5 or 2) for training the AlexNet CNNs discussed in Chapter 4.

AlexNet CNN gathers features of known objects in the training process. Multiple images of the same object, as shown in Figure 3.6, were used in the training process.

3.5 Detectable Trash Categories

In this work, trash detection is not limited by image background. The algorithm can detect any object on road, grass, parking areas or any public spaces. For the primary indoor test (test 1) to check the Alexnet performance, we use a publicly available trash database, Trashnet [22]. It is a database of trash objects consist of 2527 images of six classes: glass, paper, cardboard, plastic, metal, and trash. Designated items were placed on a white poster board under sunlight and/or room lighting to take the pictures. The pictures have been resized to 227 x 227 as needed for our AlexNet CNN. We used images of 5 categories from this dataset summarized in Table 3.3. After successfully detection of an object by the indoor CNN algorithm, we used our own outdoor image dataset of take or non-take categories and then identify the take category. The examples of the take categories include bottle, can, plastic, paper, food item, etc. and the non-take categories include bird, cat, dog, flower bed, etc. Then, the take category passed through another CNN algorithm which would then identify it as the trash or recycle category. In this research effort we collected 1052 outdoor images of trash and non-trash items. Table 3.4 provides a summary of these images. The outdoor data set included images of the trash objects that was oriented (rotated) at different angles as shown in Figure 3.6.

Table 3.3: Trash categories in the Trashnet database (indoor training dataset)

Class	**Group**	**Example of Items**	**Quantity**
Metal	Metal	Soda can, Foil etc.	410
Plastic	Plastic	Water bottle, plastic cup, etc.	482
Paper	Paper	White paper	594
Cardboard	Cardboard	Cardboard.	403
Glass	Glass	Mug, Glass, etc.	501
		Total	2390

Table 3.4: Trash categories of images from the Outdoor training dataset

Class	Group	Example of Items	Quantity
Non-take	Classification 1	grass, birds, trees, sidewalk, etc.	352
Take	Classification 1	Water bottle, paper, food items, can, etc.	700
Trash	Classification 2	food items,	427
Recycle	Classification 2	Water bottle, paper, can, etc.	273
		Total	1052

3.6 Trash Detecting Algorithm

The trash detecting process shown in Figure 3.5 is divided into three steps. Object detection, object retrieval, and object classification. This research effort focused on the last two algorithms that we will discuss in detail. We developed an algorithm using a CNN to detect if an object is an item to collect (such as trash or recyclable) or an item to leave alone such as animal or another item. We then developed an algorithm to detect if a collected item is either trash or a recyclable item. We developed and tested CNN's in both indoor and outdoor settings. When an item is brought inside the robot, the object will either be placed in a trash bin or a recycling bin depending on the determination of the classification algorithm.

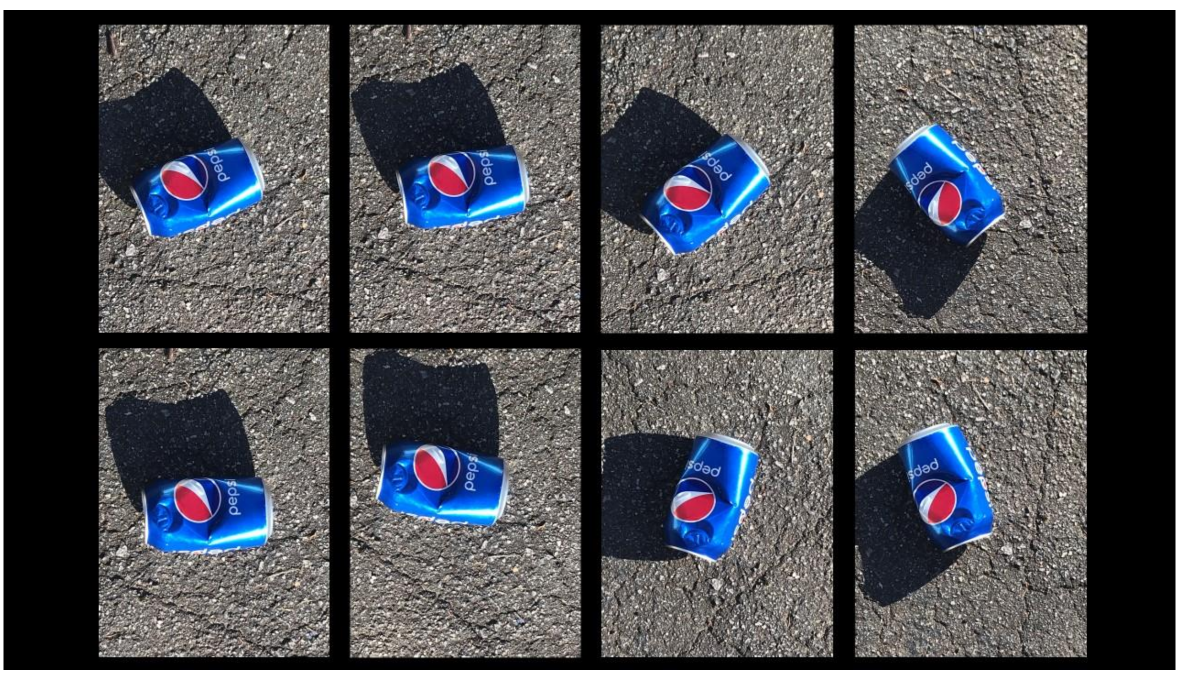


Figure 3.6: Example of images used by the training algorithm

CHAPTER FOUR: TEST PROCEDURE AND RESULTS

In this chapter we will present the procedure and results of testing the trashcan robot algorithms for selecting and sorting trash and recyclable items.

4.1 Test Design

We developed a set of procedures for training a Convolutional Neural Network (CNN) to classify objects as either trash, recyclable, or other. We used the AlexNet CNN architecture and trained it with take or non-take images in public spaces for a smart robot trashcan to decide to grab an object or not. More specifically, "take" means the identified item is a trash item to be grabbed, and "non take" means the identified item is not a trash item that should not be grabbed.

We trained AlexNet to perform a set of four tests as follows:

1. Test-1: Trained AlexNet with TrashNet [22] images and tested the resulting CNN with a subset of TrashNet images in 5 categories (metal, plastic, glass, paper, cardboard).
2. Test-2: Tested the same CNN using an indoor camera in real time focusing on trash objects.
3. Test-3: Trained AlexNet with outdoor images to classify as either "take" or "non take". Tested the resulting CNN with a subset of outdoor images.
4. Test-4: Trained AlexNet with outdoor images to classify as either landfill trash or recyclable. Tested the resulting CNN with a subset of outdoor images.

Tests 1 and 2 were preliminary tests to confirm accuracy of the AlexNet CNNs. We downloaded a publicly available CNN called Deep Learning Toolbox Model for AlexNet Network [23] for use in developing an algorithm in MATLAB. We modified the AlexNet architecture by changing the number of neurons in the last Fully Connected Layer to suit our application. We also downloaded a publicly available database, named TrashNet [22], a collection of 2390 trash images taken in an indoor environment and separated into 5 categories (metal, plastic, glass, paper, cardboard) to train the CNN. The preliminary training results on the TrashNet indoor images confirmed the applicability of CNN in this application with good accuracy.

Tests 3 and 4 are practical tests that could be implemented on the final trash robot design. The outdoor "take" and "non-take" images were all taken by us from the surroundings of human living areas on a college campus. Every image used to train the CNN is a real scenario of trash in our area. For the second task, we trained another AlexNet CNN using the "take" item images to further classify the items into landfill trash or recyclable. The detailed procedure concerning each of these tests is described below.

4.2 Test 1 – Five Categories in a Controlled Indoor Setting

As a primary test, deep learning methods for implementing Convolutional Neural Networks (CNN) in MATLAB were used to train AlexNet in the 5 categories (metal, plastic, glass, paper, cardboard) using 2390 images from a subset of the TrashNet image database. We tested the accuracy of our trained version of AlexNet using a subset of the TrashNet images that were not used for training. Results are shown in Table 4.1. Accuracy of detection exceeded 80% for all 5 categories. It should be noted that the images used in this test were taken in a controlled indoor environment with a consistent lighting background. That helps to explain why the results shown above are very accurate. Several examples of the images used in the test are shown in Figure 4.1.

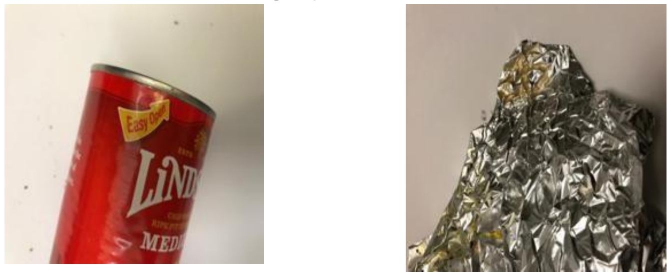
Table 4.1: Results of CNN classification using TrashNet indoor images

Category	Total count of images	Count of correctly detected images	Accuracy (%)
Metal	41	39	91.68
Plastic	48	38	81.25
Paper	59	53	89.83
Cardboard	40	37	92.5
Glass	50	46	92
Overall	238	213	89.50

4.3 Test 2 – Five Categories using a Real-Time Camera

Then, we proceeded to a camera focused on a white display board background in an indoor office shown in Figure 4.2, capturing images of 13 different objects. Each object was rotated to obtain images of each object at 10 different viewing angles, for a total of 260 captured images. A set of 10 images used to test one object is shown in Figure 4.3. The classification identified by the detection algorithm is indicated in the captured image title. For all ten images shown in Figure 4.3, the algorithm correctly identified the

Category: Metal



(a) Example of training images used as metal

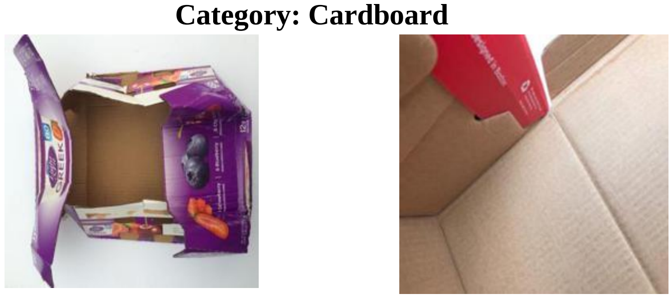
Category: Plastic

(b) Example of training images used as plastic

Category: Paper

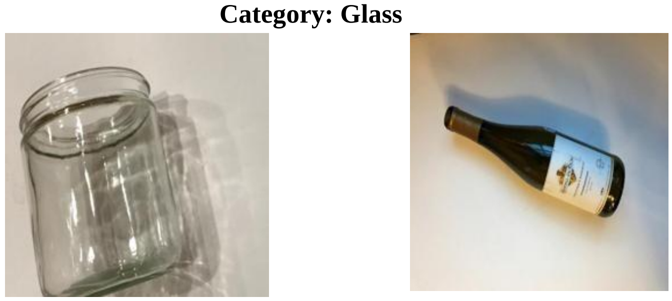
(c) Example of training images used as paper

Category: Cardboard



(d) Example of training images used as cardboard

Category: Glass



(e) Example of training images used as glass

Figure 4.1: Examples of training images used to detect five categories of recyclable materials.

bottle as plastic. Each image was tested using our trained version of AlexNet (from Test 1), which classified each image into one of 5 categories (metal, plastic, glass, paper, cardboard). The MATLAB program used to implement the real time detection presents a figure of the real time image with the detected object listed in the figure title. This is shown in Figures 4.4 and 4.5 in which the captured images of the 13 objects that were used to test the CNN. These figures show some sample results. For example, in Figure 4.4 “object 1” a metal can is correctly identified as metal from a top view, but incorrectly identified as plastic from a side view.

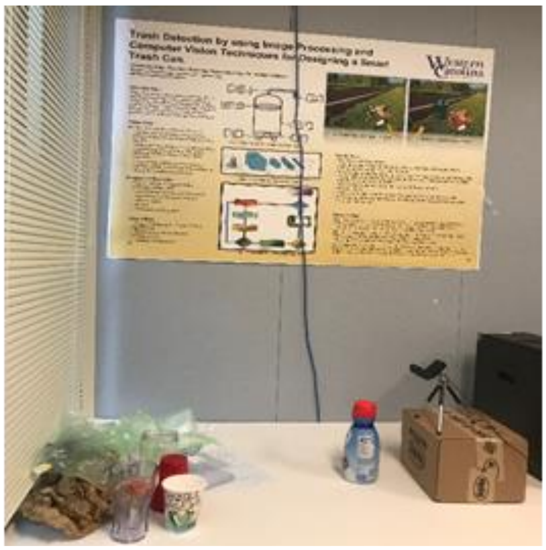


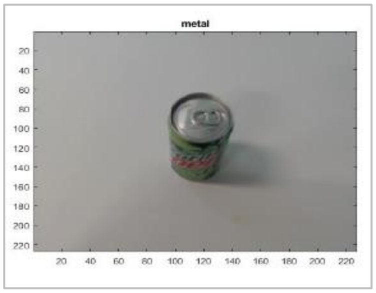
Figure 4.2: Indoor test environment (WCU Graduate Office)

A summary of the test results is shown in Table 4.2. The trained AlexNet was able to identify 7 out of 13 objects correctly with an accuracy of 90% or higher and was able to identify 3 out of 13 objects correctly with an accuracy between 70% and 80%. The green plastic bag, the plastic bottle, the plastic box, the white ceramic mug, the 3D printed green object, and the hard paper box were all successfully identified correctly with a 100% accuracy by our trained CNN. Other objects were difficult to identify perfectly. For example, for the red plastic cup, 80% of the images were correctly identified as plastic and 20% were incorrectly identified as metal. For the brown piece of paper, 70% of the images were identified as cardboard and 30% were incorrectly identified as plastic. For the white piece of paper with writing, 70% of the images were correctly identified as paper and 30% were identified as cardboard. For the clear glass, 60% of the images were correctly identified as glass and 40% were incorrectly identified as plastic. For the green soda can, 90% of the images were correctly identified as metal and 10% were incorrectly identified as plastic. For the orange plastic box, 20% of the images were correctly identified as plastic and 80% were incorrectly identified as paper or cardboard. We tested a ceramic mug that was categorized 100% as glass. We did not train our CNN for identifying ceramic material. Of the categories with which the CNN was trained, glass is the closest category to ceramic.

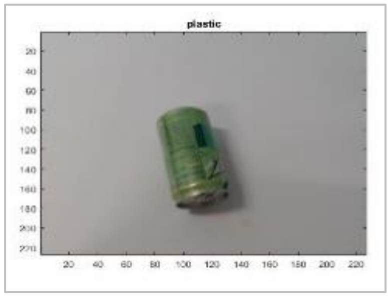
The CNN had some difficulty with three objects: brown paper, clear glass, and an orange plastic box. The brown paper was confused for cardboard and plastic. 70% of the brown paper images were classified as cardboard and 30% were classified as plastic. One of the possible reasons for this is that the CNN is unable to differentiate the color between cardboard and paper as the color is very close in both cases. In case of clear glass, 60% of the images were classified as glass and 40% were classified as plastic.

Figure 4.3: One full test procedure for an object

Object 1

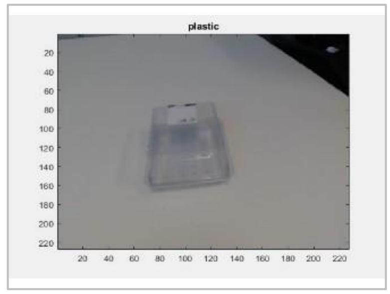


Correct detection of a green soda can



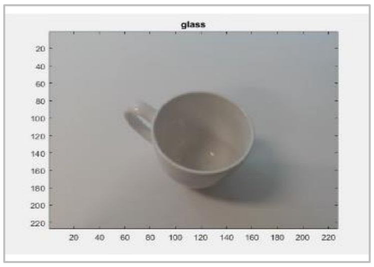
Incorrect detection of a green soda can as plastic

Object 2



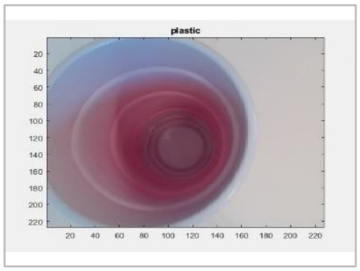
Correct detection of a plastic box

Object 3

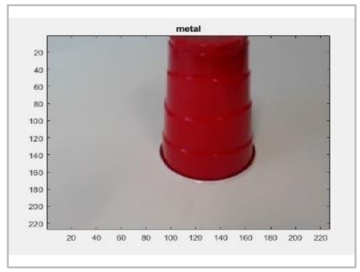


Correct detection of a white mug

Object 4

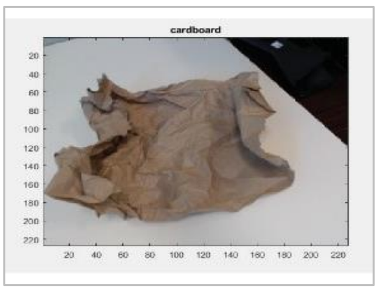


Correct detection of a plastic cup



Incorrect detection of a plastic cup as metal

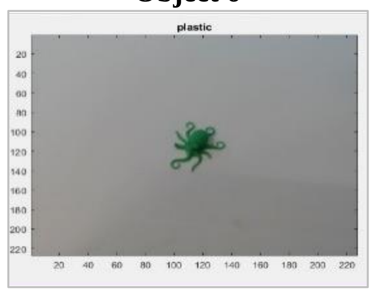
Object 5



Incorrect detection of brown paper as cardboard

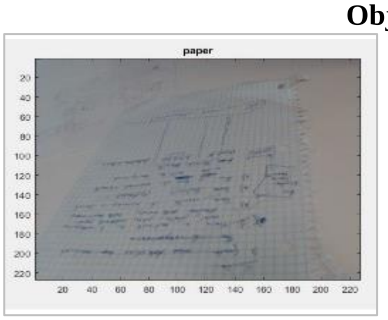
Incorrect detection of brown paper as plastic

Object 6

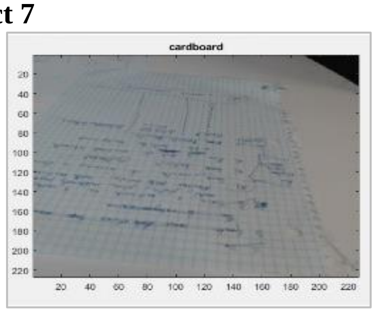


Correct detection of 3d printed object

Object 7



Correct detection of writing paper



Incorrect detection of writing paper as cardboard

Object 8

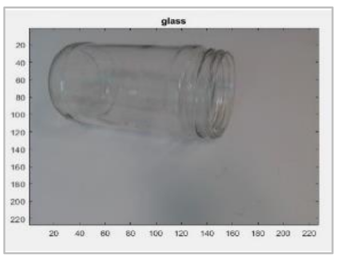
Correct detection of a box cardboard (corrugated paper)

Figure 4.4: Example of images of objects 1-8 used for testing

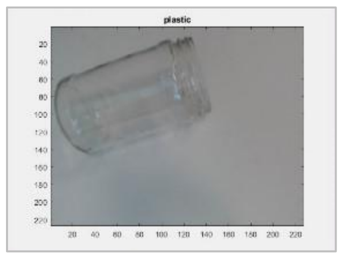
Table 4.2: Results of AlexNet CNN classification using indoor camera, trained on TrashNet images

Object number	Object	Actual Category	Detected Category
1	green soda can	metal	90% metal, 10% plastic
2	Plastic box	Plastic	100% Plastic
3	White glass mug	Glass	100% Glass
4	Red Plastic cup	Plastic	80% Plastic, 20% metal
5	Brown paper	Paper	70% cardboard, 30% plastic
6	3D printed green object	plastic	100% plastic
7	White paper with writing	Paper	70% Paper, 30% cardboard
8	Box	Cardboard	100% Cardboard
9	Clear glass	glass	60% glass, 40% plastic
10	Green Plastic bag	Plastic	100% Plastic
11	clear plastic cup	plastic	70% plastic, 30% glass
12	Plastic bottle	Plastic	100% Plastic
13	Orange plastic box	Plastic	20% plastic, 80% paper or cardboard

Object 9



Correct detection of a clear glass jar

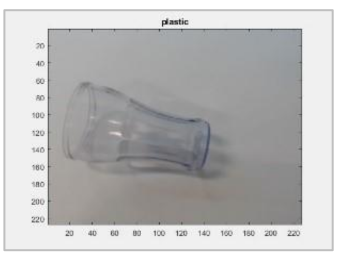


Incorrect detection of a clear glass jar as plastic

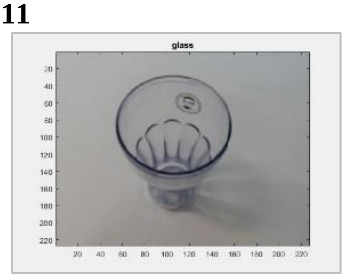
Object 10

Correct detection of a plastic bag

Object 11

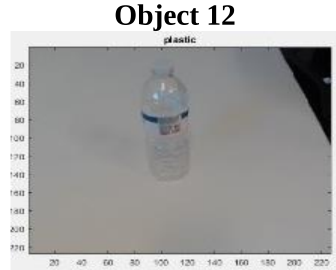


Correct detection of a clear plastic cup



Incorrect detection of a clear plastic cup as glass

Object 12



Correct detection of a plastic bottle

Object 13

Correct detection of an orange box

Incorrect detection of an orange box as cardboard

Incorrect detection of an orange box as paper

Figure 4.5: Example of images of objects 9-13 used for testing

The reason could be a similar reflection of light on glass and plastic. The same problem was showing for a clear plastic cup in which 70% of the images were classified as plastic and 30% were classified as glass.

4.3 Test 3 – Classifying Outdoor Images as either Take or Non-take

To develop training for outdoor images, we took 1052 digital pictures of outdoor scenes, and then put them into two categories of "take" or "not take". For the first outdoor test we captured the photos of items mainly on the grass, on a sidewalk, on the road, or in a flower bed. We then trained AlexNet on these 1052 images. Images in the "take" category included trash and recyclable items. Images in the "non-take" category included grass, birds, trees, sidewalk, etc. We tested the CNN on 316 similar types of images from these two categories (210 "take" images and 106 "non-take" images). Table 4.3 shows the results of this test. The overall classification accuracy was 93.6%. 97.6% of the "take" items were correctly identified and 85.8% of the "non-take" items were correctly identified. Figures 4.6 and 4.7 show a few examples of training images used to train the CNN and Figures 4.8 and 4.9 show the example of test images used to test the CNN, respectively, for classification of outdoor objects. When we test with a given image, the CNN algorithm makes a decision of "take" or "non-take" and displays the image with a title indicating the decision, as shown in Figure 4.8.

4.5 Test 4 – Classifying Outdoor Images as either Trash or Recyclable

Then we split the "take" image database into the two categories, "trash" and "recyclable," and trained another AlexNet CNN with 700 outdoor images from the "take" category. We then tested the CNN on 175 similar types of images from these two categories (107 "trash" images and 68 "recycle" images). The results (shown in Table 4.5) were as accurate as before. The overall classification accuracy was 92%. 89.7% of the "Recycle" items were correctly identified and 93.5% of the "Trash" items were correctly identified. Figure 4.11 and 4.12 show the example of images used to train and test the CNN, respectively. Figure 4.13 shows several MATLAB sample output images with decision of "trash" or "recycle" indicated in the image title.

“take” images

“non-take” images

(a) Take image on grass

(b) Non-take image on road

(c) Take image on grass

(d) Non-take image on road

Figure 4.6: Example of training images for outdoor object classification as either Take or Non-take

“take” images

“non-take” images

(e) Take image on grass

(f) Non-take image on road

(g) Take image on grass

(h) Non-take image on road

Figure 4.7: Example of more training images for outdoor object classification as either Take or Non-take

“take” images

(a) Take image on grass

“non-take” images

(b) Non-take image beside road

(c) Take image on road

(d) Non-take image on road

Figure 4.8: Example of test images for outdoor object classification as either Take or Non-take

“take” images

(e) Take image beside road

“non-take” images

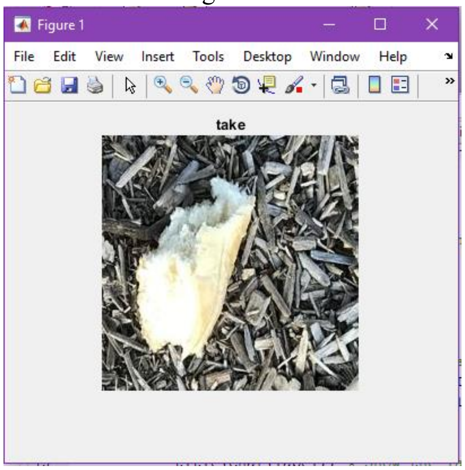
(f) Non-take image on park

Figure 4.9: Example of more test images for outdoor object classification as either Take or Non-take

Table 4.3: Results of CNN classification using outside images with 2 categories

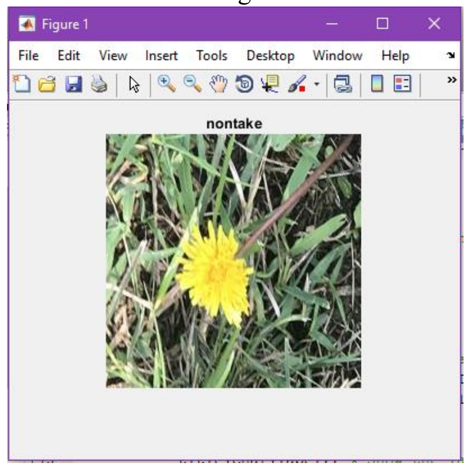
Category	Total count of images	Count of correctly detected images	Accuracy (%)
“take”	210	205	97.6
“non-take”	106	91	85.9
Overall	316	296	93.6

“take” images



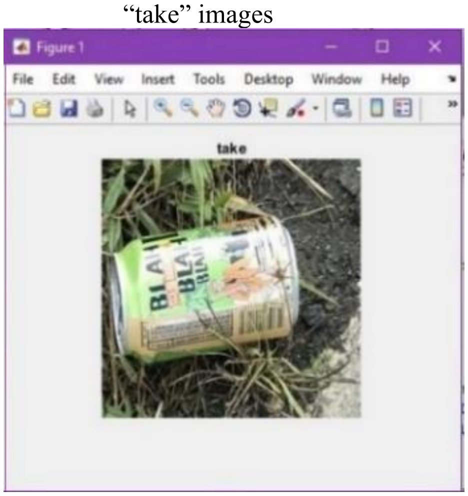
(a) Correct detection of take object in a park

“non-take” images



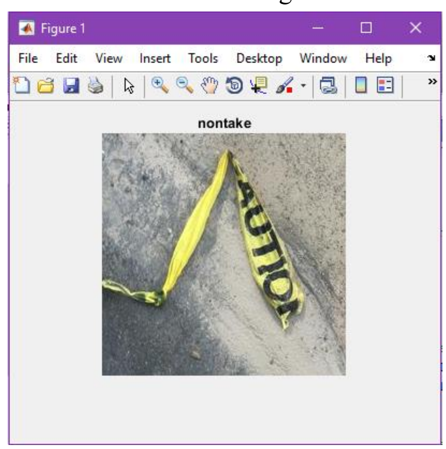
(b) Correct detection of non-take object on grass

“take” images



(c) Correct detection of take object beside road

“non-take” images



(d) Correct detection of non-take object on road

Figure 4.10: Sample output of test images for outdoor object detection as either Take or Non-take

Table 4.4: Results of CNN classification using outside images with 2 categories

Category	Total count of images	Count of correctly detected images	Accuracy (%)
“recycle”	68	61	89.7
“trash”	107	100	93.5
Overall	175	161	92

4.6 LabView Classification Interface for Testing

We have developed a LabVIEW classification interface for our thesis by using image processing and vision development module toolbox of National Instruments. We used LabVIEW version 2018 for this system. We have designed a front panel that can take images of an object with a camera and process the image by LabVIEW programming to rename, resize, reverse, subtract background, detect edge, "trash" images

(a) Trash object on grass

"recycle" images

(b) Recyclable object on park

"trash" images

(c) Food trash object on road

"recycle" images

(d) Recyclable object beside road

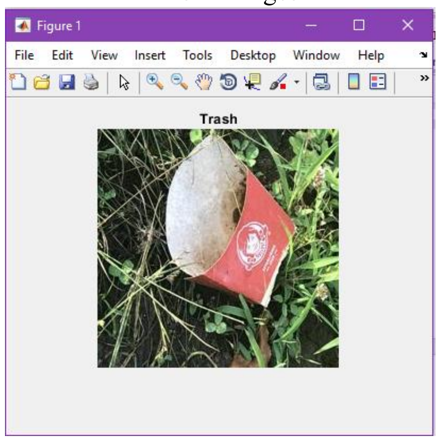
Figure 4.11: Example of training images for outdoor object classification as either Trash or Recyclable

(e) Trash object beside road

(f) recycle object beside road

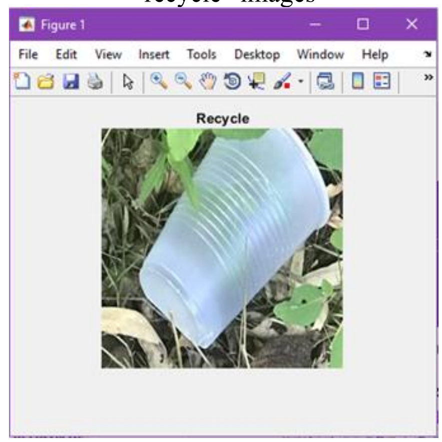
Figure 4.12: Example of test images for outdoor object classification as either Trash or Recyclable

convolve, and filter. As we need to pre-process our images before training and testing by a CNN, we developed this front panel-based interface to make the training process easy and time efficient. Finally, the system is linked with our MATLAB object detection and object classification algorithm to output the image with a label of the object "Take" or "Non-take" and "Trash" or "Recycle" on LabVIEW front panel after comparing the test image database with the training image database. These two outputs are “trash” images



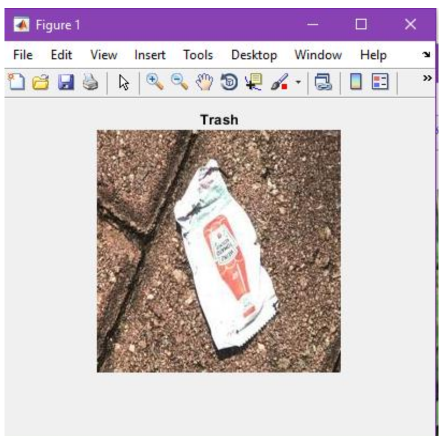
(a) Correct detection of trash object on grass

“recycle” images



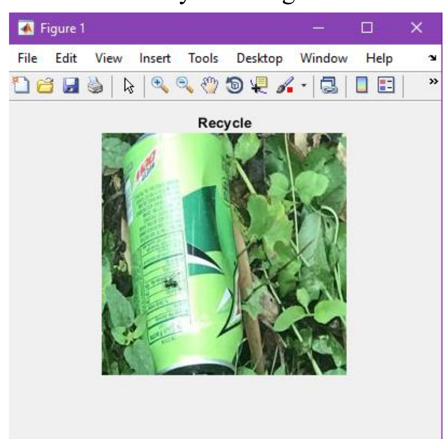
(b) Correct detection of recycle object on grass

“trash” images



(c) Correct detection of trash object on sidewalk

“recycle” images



(d) Correct detection of recycle object on grass

Figure 4.13: Sample output of test images for outdoor object classification as either Trash or Recyclable

categorical data produced from MATLAB Script Node that will be used for decision making. The robot decides the object should be grabbed or not and should be placed in trash bin or recycle bin. We have used the same camera discussed in chapter 3. We used Vision Development Module 2018 SP1 to develop the interface between camera and LABVIEW. It can be downloaded from the National Instruments website [24]. We used a new software Vision Acquisition Software version 18.0 downloaded from National Instruments website [25]. Figure 4.15 shows the indoor testing environment of the system.

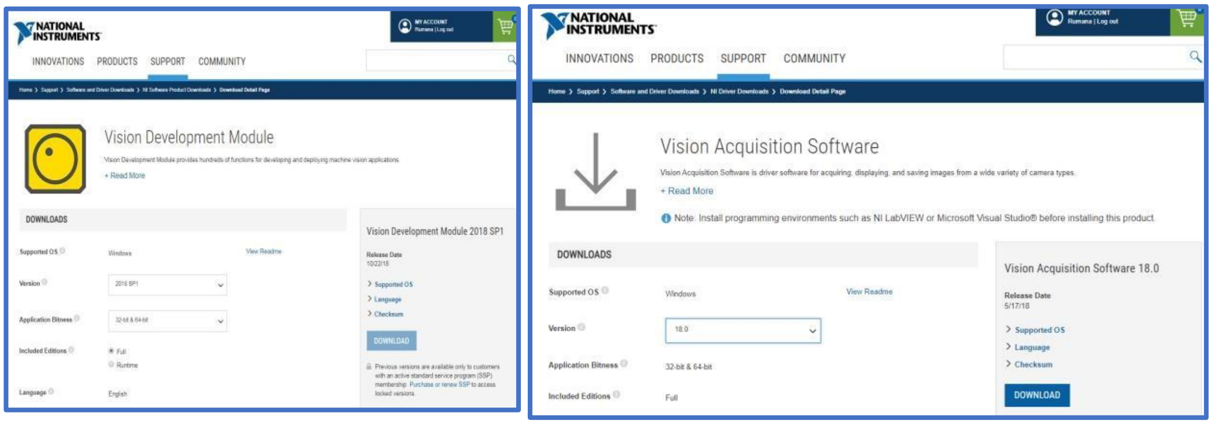


Figure 4.14: Vision Development Module & Vision Acquisition Software download pages
[Source: https://www.ni.com/en-us/support/downloads/software-products/download.vision-development-module.html#329460] and [https://www.ni.com/en-us/support/downloads/drivers/download.vision-acquisition-software.html#306475]

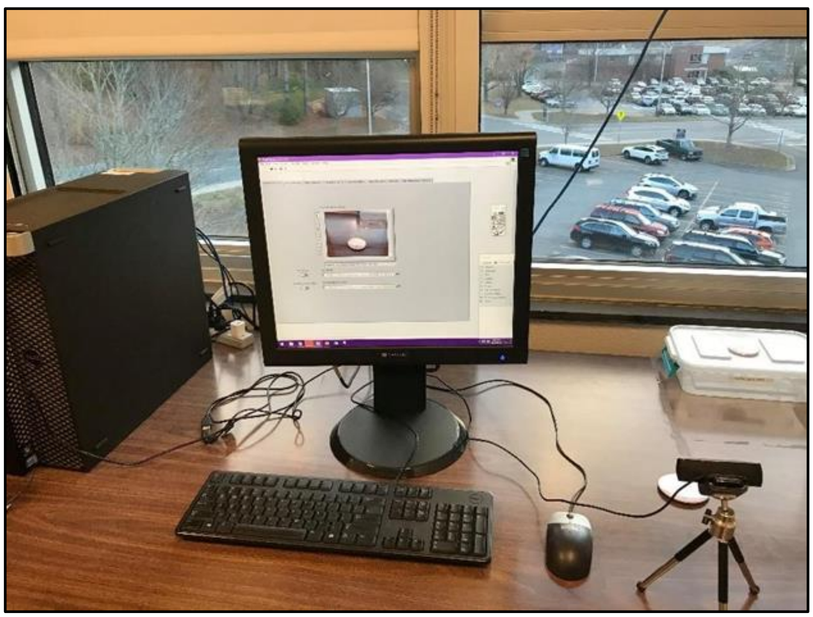


Figure 4.15: Office environment with the camera interface with LabVIEW

4.6.1 Functionality of the system

Function-1 (Input): At first the system took an image by a camera. Then, it established an interface between LabVIEW and a USB webcam by using the Vision Development module.

Function-2 (Image Processing): Then the image read by the image input control and processed by the graphics and sounds functionalities and signal processing functionalities. There will be different **Tab View** for every image processing tools like renaming, resizing, image subtracting, edge detecting reversing, rotating, and filtering.

- The resizing function in LabVIEW is a user defined Resize Button or VI which resized the image in a desired image size.

- The renaming function in LabVIEW is a user defined Rename Button or VI which renamed the image in a required image name such as “take” or “non take” and “trash” or “recycle”.

- The image subtracting function in LabVIEW is a user defined Subtraction button or VI which subtracted image 1 from image 2 and finally show the difference of the two images. Image 1 is a background image of clean green grass or clean road or clean white board and image 2 is an image of an object in the same background of image 1. The output for the image subtraction is the image of the object on the background.

- Image rotation function in LabVIEW can rotate the image. The transformation function can rotate the image by 90-degree, 270 degree and 0 degree, or generate the mirror of the image.

- The edge detection function in LabVIEW is a user defined Edge Detection Button or VI that detected the edge of an image using a convolution.

- The reverse function in LabVIEW is a user defined Reverse Button or VI that reversed an image in opposite direction.

Function-3 (Object detection): The Object Detection (Outdoor) Tab control can collect the data from a trained CNN in MATLAB and output the category of an object if it is take or non-take. It is processed by a MATLAB Script Node.

Function-4 (Object Classification): The Object Classification (Outdoor) Tab control can collect the data from another trained CNN in MATLAB and output the category of an object if it is trash or recycle. It is processed by a MATLAB Script Node, too.

Function-5 (Output): Then the output image is shown in the tab window for the current task by the image output indicator. For every function, there is separate output panel in a new tab.

4.6.2 Results of the system

Figure 4.16 shows a complete front panel of the system. It has seven tasks in seven tab window. The tasks are described below.

Task-1 (vision and image acquisition)

Figure 4.17 shows a continuous vision acquisition window. It can take picture anytime with the button “Save button” and “Save Background”. The images are saved in the given path.

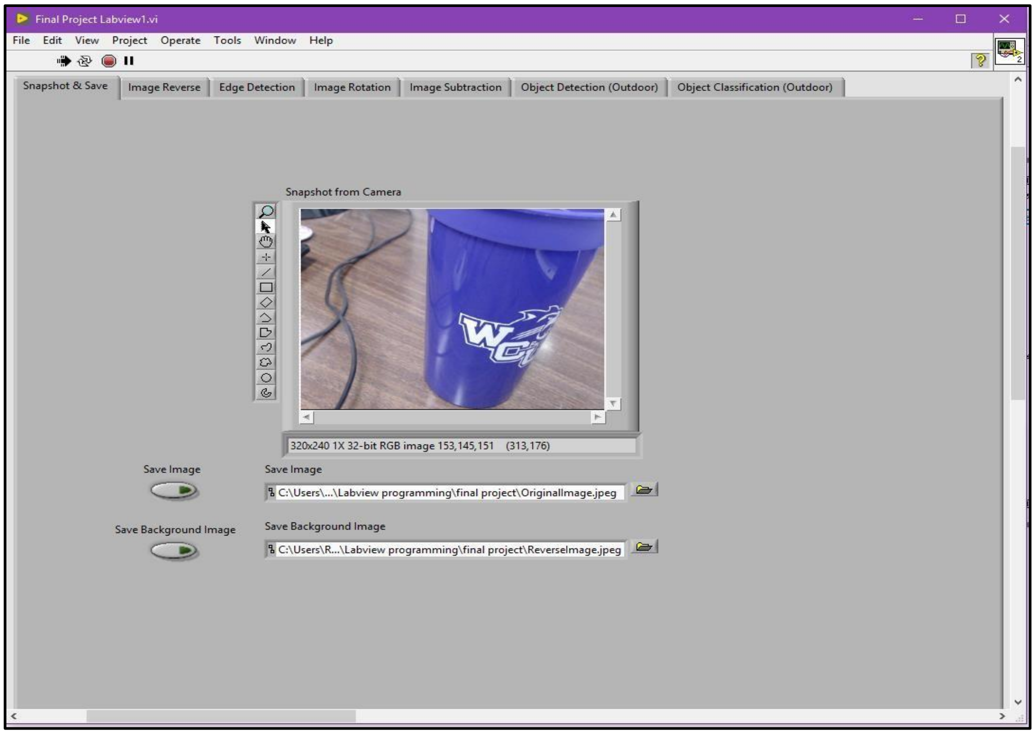


Figure 4.16: Complete front panel

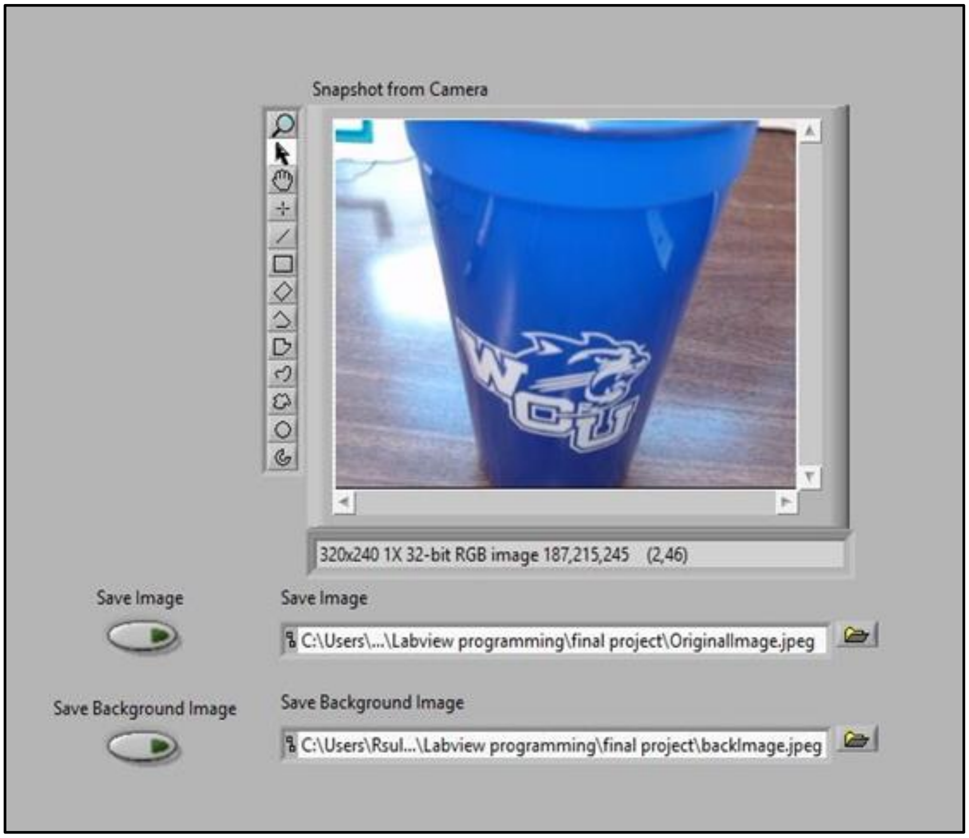


Figure 4.17: Output of Vision and Image acquisition Window

Task-2 (image reverse)

The image reverse had a glitch to change the color of the image, but otherwise worked as expected. We included the work to fix this problem in our future work list. Figure 4.18 shows the window with the output.

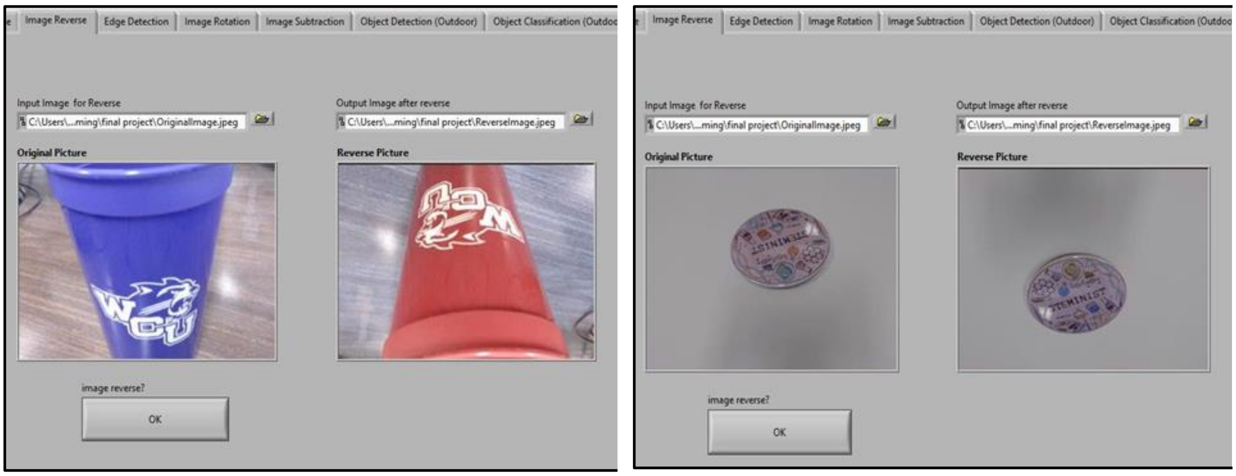


Figure 4.18: Output of Image Reverse Window

Task-3 (edge detection)

Figure 4.19 shows the gray color edge detection. Before applying edge detection, the image is converted into grayscale image.

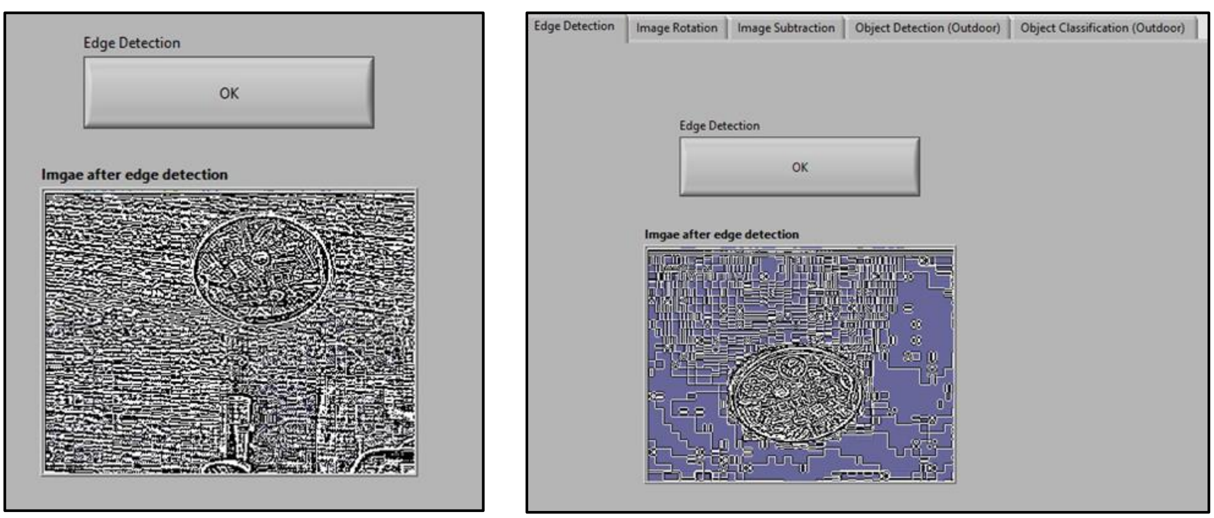


Figure 4.19: Output of Gray color edge detection of an image

Task-4 (image rotation)

The result showed the different angular rotation. Figure 4.20 and 4.21 shows the rotation of an image from original image angle 0^{o} to 90^{o}, mirror and 270^{o} controlled by the transformation button.

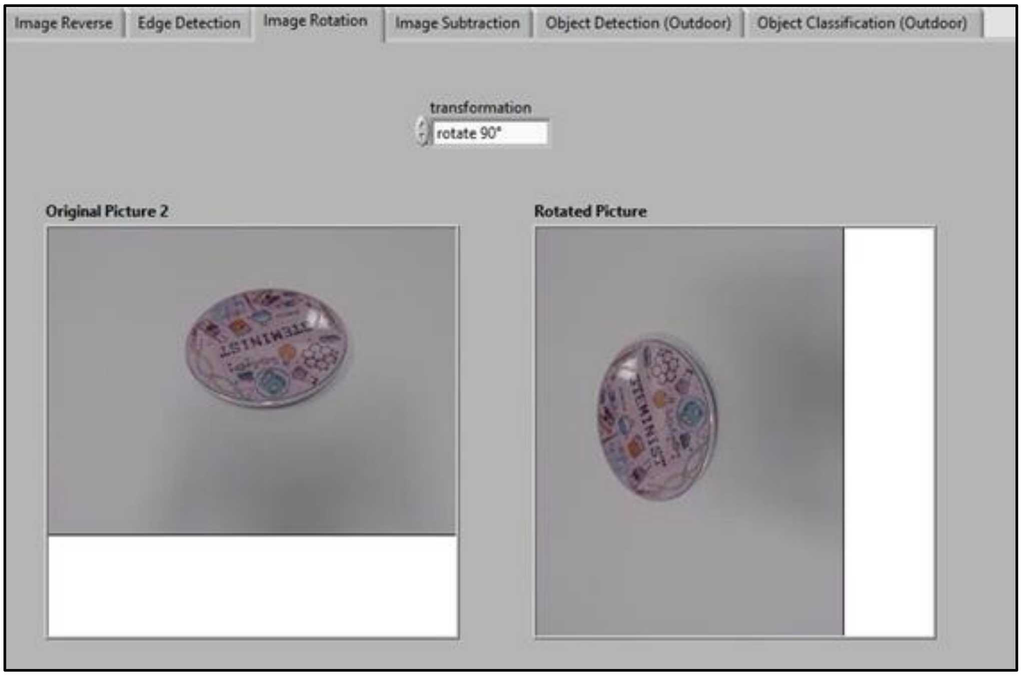


Figure 4.20: Output of Image Rotation Task of object 1 (90^{o}, Mirror and 270^{o})

Figure 4.21: Output of Image Rotation Task of object 2 (90^{o}, Mirror and 270^{o})

Task-5 (image subtraction)

The subtraction result is found by subtracting an original image from the background image. The result is better when the light reflection is lower. We observed good result in day light. Figure 4.22 shows the output of the task. In both cases the object boundary is shown with a white color in a black background.

The first object gets more accurate result with less noise whereas the second object has noisy output for the shadow effect.

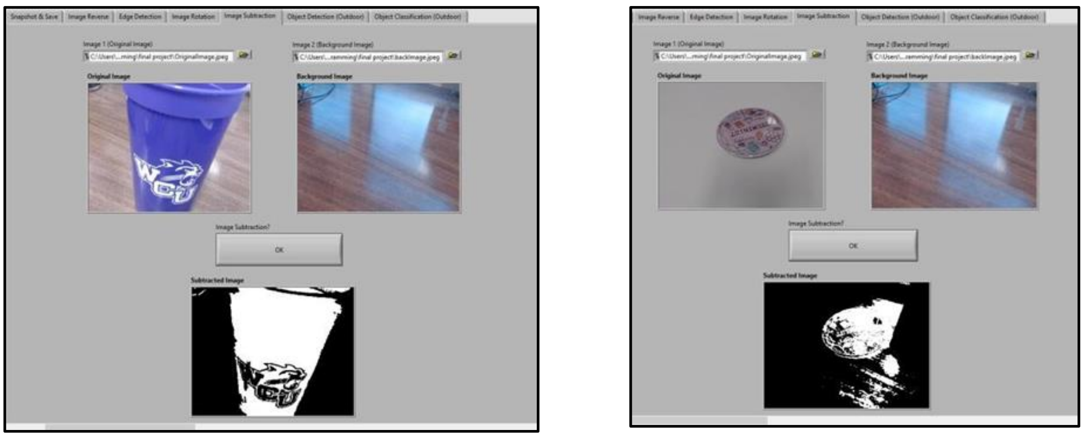


Figure 4.22: Output of Image Subtraction

Task-6 (object detection (outdoor))

The object detection window shows the testing result of a set of images in a folder ordered by their file names. It reports the result one by one in either of the two categories “Take” and “Non-take” with an object ID and its predicted class. Figure 4.23 shows the output. As the data is collected from the previously trained Alexnet CNN, the result is the same as in Table 4.3.

Task-7 (object classification (outdoor))

The object classification window shows the testing result of a set of images that are in the “take” category and stored in a folder. It reports the result one by one in either of the two categories “Trash” and “Recycle” with an object ID and its predicted class. Figure 4.24 shows the output. As the data is collected from the previously trained Alexnet CNN, the result is the same as in Table 4.4.

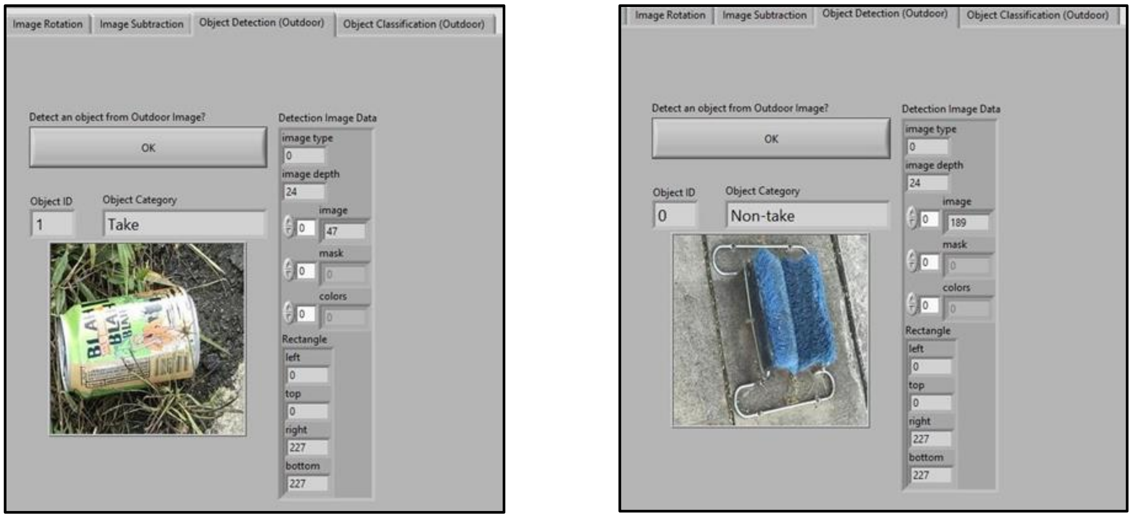


Figure 4.23: Output of object classification between Take and Non-take category

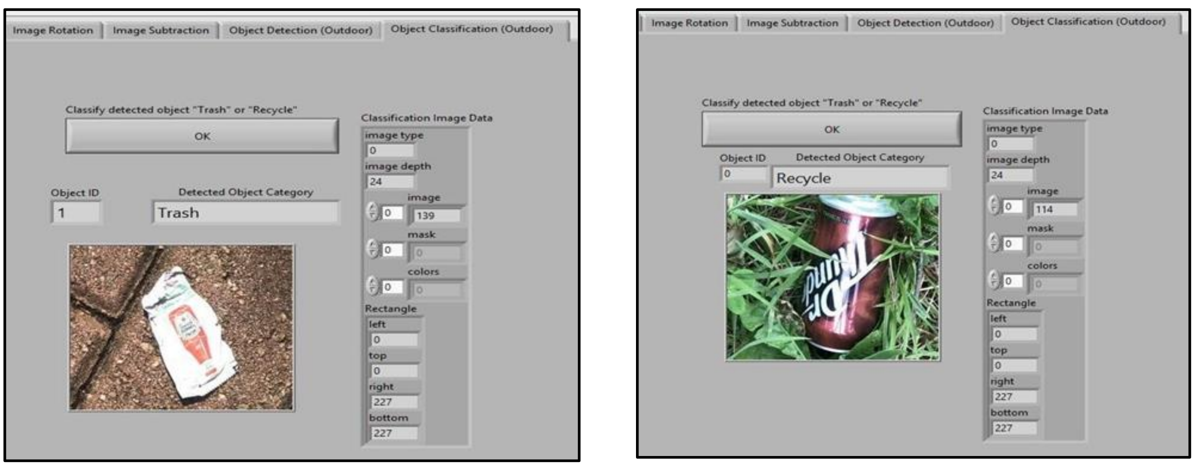


Figure 4.24: Output of object classification between Trash and Recycle category

CHAPTER FIVE: CONCLUSION AND FUTURE WORK

In this study we developed an CNN-based algorithm for detecting trash and non-trash, as well as further differentiating landfill and recyclable items in the trash category, for the purpose of developing an automatic trash collection system. We have conducted four tests, two for the initial performance testing in an indoor environment and two for the final trash classification task in an outdoor environment. The first initial test got 89.5% overall accuracy that assured us to proceed to the further testing procedure with AlexNet CNN. The result of the second initial test was also promising. Upon finding the successful results of the two initial tests we have conducted the final two tests with our original outdoor trash dataset. The accuracy of detection ranged from 89.7% to 93.5%. Integrating this image processing-based classification into smart trashcans will be more suitable for cleaning garbage on public spaces than the existing cleaning mechanisms used by road sweeper trucks or vacuum cleaning. Experimental results proved that the proposed algorithm can recognize garbage and recyclable material accurately. This algorithm can serve as a powerful tool for designing a trashcan robot for cleaning the garbage on a big lawn in a park or at school.

Future work will consist of using our two-stage trained CNN in an algorithm that can work with a microcontroller and a camera to move a trashcan robot around a public space and identify an object on the ground, then pick and sort the trash as landfill or recyclable. Future work will involve training the AlexNet architecture to detect objects surrounding the trashcan and to provide commands to the motors to move the trashcan robot in the direction of fixed objects. As the CCD cameras have wide-dynamic-range sensitivity for the change of ambient light, this camera should be mounted in the upper body of the trashcan. For initialization, the trash detection robot will first take images of its surrounding without any trash in it from multiple angles to allow image registration between multiple images to align them well. Then the camera at the top of the can will take the pictures of its surrounding at a specified time interval. The trash detection robot will apply a pattern recognition technique of neural network system to process the images to predict the trash position and its distance to the camera. The predicted trash will be surrounded by a bounding box to confirm the position of it. Our future work also consists of testing our trained outdoor CNN to identify trash or recyclable items with a real time camera similar to the test 2 with indoor setup. Currently, we were not able to conduct this test due to unavailability of high-performance laptop in hand as the deep neural network needs high performance GPU with compute capability 3.0 or higher, large memory and other high-end specifications.

Currently, we are facing some challenges to identify some objects for example, glass and plastic are often mixed-up with an accuracy less than 90%. These challenges happened perhaps due to the lighting effect, camera quality, environmental or weather factors, etc. We have a future goal to fix this problem. We will work on setting up a test to find out the best camera for our application based on its resolution, pixel quality, and other essential features.

REFERENCES

[1] World Health Organization Website https://www.who.int/water_sanitation_health/diseases-risks/diseases/en/

[2] CBSNewYork https://newyork.cbslocal.com/2018/01/29/nyc-garbage-pickup-charges/

[3] Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in Proceedings of Neural Information Processing System Conference, Lake Tahoe, CA, USA, December 2012.

[4] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. Lecun, “OverFeat: integrated recognition, localization and detection using convolutional networks,” in Proceedings of International Conference on Learning Representations, Banff, Canada, April 2014.

[5] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional neural networks,” in Proceedings of 13th European Conference on Computer Vision (ECCV), Zurich, Switzerland, September 2014.

[6] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proceedings of Conference on Computer Vision and Pattern Recognition, Las Vegas Valley, NV, USA, June 2016.

[7] Chu, Y., Huang, C., Xie, X., Tan, B., Kamal, S., & Xiong, X. (2018). “Multilayer Hybrid Deep-Learning Method for Waste Classification and Recycling,” Computational Intelligence and Neuroscience, 2018.

[8] J. Bai, S. Lian, Z. Liu, K. Wang and D. Liu, "Deep Learning Based Robot for Automatically Picking Up Garbage on the Grass," in IEEE Transactions on Consumer Electronics, vol. 64, no. 3, pp. 382-389, Aug. 2018.

[9] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, 2016, pp. 770–778.

[10] Fundamentals of Digital Image Processing, http://www.eie.polyu.edu.hk/~enyhchan/imagef.pdf

[11] Online Source: https://en.wikipedia.org/wiki/Digital_image_processing

[12] Introducing Deep Learning with MATLAB by MathWorks, https://www.mathworks.com/content/dam/mathworks/ebook/gated/80879v00_Deep_Learning_ebook.pdf

[13] Zahangir Alom, M., Taha, T. M., Yakopcic, C., Westberg, S., Sidike, P., Shamima Nasrin, M., ... & Asari, V. K. (2018). The history began from AlexNet: a comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*.

[14] Online Source: http://fourier.eng.hmc.edu/e176/lectures/ch10/node8.html

[15] Gonzales, R.C. and Woods, R.E., “Digital Image Processing,” 4th edition, Pearson, 2018.

[16] Online Source: http://dsdeepdive.blogspot.com/2015/07/gradient-descent-with-python.html/

[17] Online Source: https://www.alamy.com/stock-photo-roadside-garbage-25647833.html

[18] C922 PRO STREAM WEBCAM Setup Guide, Source link: https://images-na.ssl-images-amazon.com/images/I/B1nMdnP1ZUS.pdf

[19] Online Source: https://www.logitech.com/en-us/product/c922-pro-stream-webcam

[20] Online Source: https://the-gadgeteer.com/2018/05/25/logitech-c922-pro-stream-webcam-review/

[21] Online Source: https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/quadro-product-literature/13720_DS_NV_Quadro_K420_Aug25_US_NV_HR.pdf

[22] Trashnet, Dataset of images of trash, https://github.com/garythung/trashnet.

[23] AlexNet Tool Box, Mathworks, https://www.mathworks.com/matlabcentral/fileexchange/59133-deep-learning-toolbox-model-for-alexnet-network.

[24] National Instruments website, Source link: https://www.ni.com/en-us/support/downloads/software-products/download.vision-development-module.html#329460

[25] National Instruments website, Source link: https://www.ni.com/en-us/support/downloads/drivers/download.vision-acquisition-software.html#306475

APPENDIX A: MATLAB CODE FOR TRANSFER LEARNING (INDOOR)

The MATLAB Code shown below implements the algorithm for training the AlexNet CNN with five categories of metal, plastic, glass, cardboard, and paper.

```
%============================================================
% Filename:     TransferLearningVideo.m
% Author:       Rumana Sultana
% Course:       Thesis
% Thesis Title: TRASH AND RECYCLABLE MATERIAL IDENTIFICATION USING
%CONVOLUTIONAL NEURAL NETWORKS (CNN)
% Semester:     Fall 2019-Spring 2020

% Description: This is the script to run the training process for
% our thesis work with five categories. This training takes 2352 images from TrashNet database
% as image datastore of five categories (Metal, Plastic, Glass, Cardboard, Paper)
% and save the CNN with the name myAlexNet. We can use this CNN for testing
% or any other use later by loading in matlab workspace.
%
%% Load a pre-trained, deep, convolutional network
alex = alexnet; % loading alexnet CNN
layers = alex.Layers; % assigning the layers of Alexnet to a variable

%% Modify the network to use five categories
layers(23) = fullyConnectedLayer(5); % modifying the fully connected layer for five
                                     % categories(Metal, Plastic, Glass, Cardboard, Paper)
layers(25) = classificationLayer
clear alex; % free up some memory

%% Set up our training data
% myImages is a folder of 2352 images with five sub-folders,
% average images of approximately 500 for each category
allImages = imageDatastore('myImages', 'IncludeSubfolders', true, 'LabelSource', 'foldernames');

% Spliting the imagedatastore into 2 partition training imamges and testing
% images in a ration of 90:10
[trainingImages, testImages] = splitEachLabel(allImages, 0.9, 'randomize');
clear allImages;

%% Re-train the Network
opts = trainingOptions('sgdm', 'InitialLearnRate', 0.001, 'MaxEpochs', 20, 'MiniBatchSize', 64);
myAlexNet = trainNetwork(trainingImages, layers, opts);
save myAlexNet; % Saving the CNN

%% Measure network accuracy
predictedLabels = classify(myAlexNet, testImages); % comparing the test images with training images
accuracy = mean(predictedLabels == testImages.Labels);% accuracy calculation
save myAlexNetaccuracy
```

APPENDIX B: MATLAB CODE FOR TRANSFER LEARNING (OUTDOOR)

The MATLAB Code shown below implements the algorithm for training the AlexNet CNN with two categories of “Take” or “Non-Take.”

```
%=============================================================
% Filename:    TransferLearningVideoTNTosSEP17TN.m
% Author:      Rumana Sultana
% Course:      Thesis
% Thesis Title: Trash and recyclable material identification using convolutional neural networks (CNN)
% Semester:    Fall 2019-Spring 2020
% Description: This is the main script to run the training process for our thesis work. This training takes
%1052 images as image datastore of two categories "take" and "non take" and save the CNN with the
%name myAlexNetTNTosSep17TN. We can use this CNN for testing or any other use later by loading in
%matlab workspace.
%=============================================================
clear;
close all; tstart=tic;
%% Load a pre-trained, deep, convolutional network
alex = alexnet; % loading alexnet CNN
layers = alex.Layers % assigning the layers of Alexnet to a %variable

%% Modify the network to use five categories
layers(23) = fullyConnectedLayer(2); % modifying the fully connected layer for two class(take, non take)
layers(25) = classificationLayer
clear alex; % free up some memory

%% Set up our training data
% MyImageOSTSep17TN is a folder with 1052 images with two sub-folders take (700 images) and
%nontake (352 images)
allImages = imageDatastore('MyImageOSTSep17TN', 'IncludeSubfolders', true, 'LabelSource',
'foldernames');

% Spliting the imagedatastore into 2 partition training imamges and testing images in a ration of 75:25
[trainingTNTImagesTN, testTNTImagesTN] = splitEachLabel(allImages, 0.75, 'randomize');
clear allImages;
%% Re-train the Network
tic;
opts = trainingOptions('sgdm', 'InitialLearnRate', 0.001, 'MaxEpochs', 20, 'MiniBatchSize', 64);
myAlexNetTNTosSep17TN = trainNetwork(trainingTNTImagesTN, layers, opts);
save myAlexNetTNTosSep17TN % Saving the CNN

%% Measure network accuracy
predictedLabels = classify(myAlexNetTNTosSep17TN, testTNTImagesTN); % comparing the test
images with training images
accuracy = mean(predictedLabels == testTNTImagesTN.Labels)% accuracy calculation
save myAlexNetTNTosSep17TNaccuracy
time=toc(tstart);
```

The MATLAB Code shown below implements the algorithm for training the AlexNet CNN with two categories of “Recycle” or “Trash.”

```
%===========================================================
% Filename:     TransferLearningVideoTNTosSEP17TrashRecycle.m
% Author:       Rumana Sultana
% Course:       Thesis
% Thesis Title: Trash and recyclable material identification using convolutional neural networks (CNN)
% Semester:     Fall 2019-Spring 2020

% Description: This is the main script to run the training process for our thesis work. This training takes
% 700 images as image datastore of two categories "recycle" and "trash" and save the CNN with the
%name myAlexNetTNTosSep17TrashRecycle. We can use this CNN for testing or any other use later by
%loading in matlab workspace.
%===========================================================

%% Load a pre-trained, deep, convolutional network
alex = alexnet;
layers = alex.Layers

%% Modify the network to use five categories
layers(23) = fullyConnectedLayer(2); % (recycle, trash)
layers(25) = classificationLayer
clear alex; % free up some memory

%% Set up our training data

allImages = imageDatastore('MyImageOSTSep17TrashRecycle', 'IncludeSubfolders', true, 'LabelSource',
'foldernames');

[trainingTNTImagesTrashRecycle, testTNTImagesTrashRecycle] = splitEachLabel(allImages, 0.75,
'randomize');
clear allImages;

%% Re-train the Network
opts = trainingOptions('sgdm', 'InitialLearnRate', 0.001, 'MaxEpochs', 20, 'MiniBatchSize', 64);
myAlexNetTNTosSep17TrashRecycle = trainNetwork(trainingTNTImagesTrashRecycle, layers, opts);
save myAlexNetTNTosSep17TrashRecycle

%% Measure network accuracy
predictedLabels = classify(myAlexNetTNTosSep17TrashRecycle, testTNTImagesTrashRecycle);
accuracy = mean(predictedLabels == testTNTImagesTrashRecycle.Labels)
save myAlexNetTNTosSep17TrashRecycleaccuracy
```

APPENDIX C: RESULTS OF OUTDOOR TESTING

Table C.1: Results of the outdoor detection task (Part 1 of 4)

Testing Using AlexNet Architecture							
Matlab Training program name: TransferLearningVideoTNTosSEP17TN.m							
Matlab training program name: C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\TransferLearning\							
SL per category	**Predicted label**	**Actual label**	**Predicted label = Actual Label**		**total count**	**predicted label number**	**accuracy**
1	nontake	nontake	TRUE	1			
2	nontake	nontake	TRUE	1			
3	nontake	nontake	TRUE	1			
4	nontake	nontake	TRUE	1			
5	nontake	nontake	TRUE	1			
6	nontake	nontake	TRUE	1			
7	nontake	nontake	TRUE	1			
8	nontake	nontake	TRUE	1			
9	nontake	nontake	TRUE	1			
10	nontake	nontake	TRUE	1			
11	nontake	nontake	TRUE	1			
12	nontake	nontake	TRUE	1			
13	nontake	nontake	TRUE	1			
14	nontake	nontake	TRUE	1			
15	nontake	nontake	TRUE	1			
16	nontake	nontake	TRUE	1			
17	take	nontake	FALSE	0			
18	nontake	nontake	TRUE	1			
19	take	nontake	FALSE	0			
20	nontake	nontake	TRUE	1			
21	nontake	nontake	TRUE	1			
22	nontake	nontake	TRUE	1			
23	nontake	nontake	TRUE	1			
24	nontake	nontake	TRUE	1			
25	take	nontake	FALSE	0			
26	nontake	nontake	TRUE	1			
27	nontake	nontake	TRUE	1			
28	take	nontake	FALSE	0			
29	take	nontake	FALSE	0			
30	nontake	nontake	TRUE	1			
31	nontake	nontake	TRUE	1			
32	nontake	nontake	TRUE	1			
33	nontake	nontake	TRUE	1			
34	nontake	nontake	TRUE	1			
35	nontake	nontake	TRUE	1			
36	nontake	nontake	TRUE	1			
37	nontake	nontake	TRUE	1			
38	nontake	nontake	TRUE	1			
39	nontake	nontake	TRUE	1			
40	take	nontake	FALSE	0			
41	nontake	nontake	TRUE	1			
42	nontake	nontake	TRUE	1			
43	nontake	nontake	TRUE	1			
44	nontake	nontake	TRUE	1			
45	nontake	nontake	TRUE	1			
46	nontake	nontake	TRUE	1			
47	nontake	nontake	TRUE	1			
48	nontake	nontake	TRUE	1			
49	nontake	nontake	TRUE	1			
50	nontake	nontake	TRUE	1			

Table C.2: Results of the outdoor detection task (Part 2 of 4)

51	take	nontake	FALSE	0			
52	nontake	nontake	TRUE	1			
53	nontake	nontake	TRUE	1			
54	nontake	nontake	TRUE	1			
55	nontake	nontake	TRUE	1			
56	nontake	nontake	TRUE	1			
57	nontake	nontake	TRUE	1			
58	nontake	nontake	TRUE	1			
59	nontake	nontake	TRUE	1			
60	nontake	nontake	TRUE	1			
61	nontake	nontake	TRUE	1			
62	nontake	nontake	TRUE	1			
63	nontake	nontake	TRUE	1			
64	nontake	nontake	TRUE	1			
65	nontake	nontake	TRUE	1			
66	nontake	nontake	TRUE	1			
67	nontake	nontake	TRUE	1			
68	nontake	nontake	TRUE	1			
69	nontake	nontake	TRUE	1			
70	nontake	nontake	TRUE	1			
71	nontake	nontake	TRUE	1			
72	nontake	nontake	TRUE	1			
73	nontake	nontake	TRUE	1			
74	nontake	nontake	TRUE	1			
75	nontake	nontake	TRUE	1			
76	nontake	nontake	TRUE	1			
77	nontake	nontake	TRUE	1			
78	nontake	nontake	TRUE	1			
79	nontake	nontake	TRUE	1			
80	nontake	nontake	TRUE	1			
81	take	nontake	FALSE	0			
82	take	nontake	FALSE	0			
83	take	nontake	FALSE	0			
84	take	nontake	FALSE	0			
85	nontake	nontake	TRUE	1			
86	nontake	nontake	TRUE	1			
87	take	nontake	FALSE	0			
88	nontake	nontake	TRUE	1			
89	nontake	nontake	TRUE	1			
90	nontake	nontake	TRUE	1			
91	nontake	nontake	TRUE	1			
92	nontake	nontake	TRUE	1			
93	take	nontake	FALSE	0			
94	nontake	nontake	TRUE	1			
95	nontake	nontake	TRUE	1			
96	nontake	nontake	TRUE	1			
97	take	nontake	FALSE	0			
98	nontake	nontake	TRUE	1			
99	take	nontake	FALSE	0			
100	nontake	nontake	TRUE	1			
101	nontake	nontake	TRUE	1			
102	nontake	nontake	TRUE	1			
103	nontake	nontake	TRUE	1			
104	nontake	nontake	TRUE	1			
105	nontake	nontake	TRUE	1			
106	nontake	nontake	TRUE	1	106	91	0.858490566

Table C.3: Results of the outdoor detection task (Part 3 of 4)

107	take	take	TRUE	1			
108	take	take	TRUE	1			
109	take	take	TRUE	1			
110	take	take	TRUE	1			
111	take	take	TRUE	1			
112	take	take	TRUE	1			
113	take	take	TRUE	1			
114	take	take	TRUE	1			
115	take	take	TRUE	1			
116	take	take	TRUE	1			
117	take	take	TRUE	1			
118	take	take	TRUE	1			
119	take	take	TRUE	1			
120	take	take	TRUE	1			
121	take	take	TRUE	1			
122	take	take	TRUE	1			
123	take	take	TRUE	1			
124	take	take	TRUE	1			
125	take	take	TRUE	1			
126	take	take	TRUE	1			
127	take	take	TRUE	1			
128	take	take	TRUE	1			
129	take	take	TRUE	1			
130	take	take	TRUE	1			
131	take	take	TRUE	1			
132	take	take	TRUE	1			
133	take	take	TRUE	1			
134	take	take	TRUE	1			
135	take	take	TRUE	1			
136	take	take	TRUE	1			
137	take	take	TRUE	1			
138	take	take	TRUE	1			
139	take	take	TRUE	1			
140	take	take	TRUE	1			
141	take	take	TRUE	1			
142	take	take	TRUE	1			
143	take	take	TRUE	1			
144	take	take	TRUE	1			
145	take	take	TRUE	1			
146	take	take	TRUE	1			
147	take	take	TRUE	1			
148	take	take	TRUE	1			
149	take	take	TRUE	1			
150	take	take	TRUE	1			
151	take	take	TRUE	1			
152	take	take	TRUE	1			
153	take	take	TRUE	1			
154	take	take	TRUE	1			
155	take	take	TRUE	1			
156	take	take	TRUE	1			

Table C.4: Results of the outdoor detection task (Part 4 of 4)

157	take	take	TRUE	1			
158	take	take	TRUE	1			
159	take	take	TRUE	1			
160	take	take	TRUE	1			
161	take	take	TRUE	1			
162	take	take	TRUE	1			
163	take	take	TRUE	1			
164	take	take	TRUE	1			
165	take	take	TRUE	1			
166	take	take	TRUE	1			
167	take	take	TRUE	1			
168	take	take	TRUE	1			
169	take	take	TRUE	1			
170	take	take	TRUE	1			
171	take	take	TRUE	1			
172	take	take	TRUE	1			
173	take	take	TRUE	1			
174	take	take	TRUE	1			
175	take	take	TRUE	1			
176	take	take	TRUE	1			
177	take	take	TRUE	1			
178	take	take	TRUE	1			
179	take	take	TRUE	1			
180	take	take	TRUE	1			
181	take	take	TRUE	1			
182	take	take	TRUE	1			
183	take	take	TRUE	1			
184	take	take	TRUE	1			
185	take	take	TRUE	1			
186	take	take	TRUE	1			
187	take	take	TRUE	1			
188	take	take	TRUE	1			
189	take	take	TRUE	1			
190	take	take	TRUE	1			
191	take	take	TRUE	1			
192	take	take	TRUE	1			
193	take	take	TRUE	1			
194	take	take	TRUE	1			
195	take	take	TRUE	1			
196	nontake	take	FALSE	0			
197	take	take	TRUE	1			
198	take	take	TRUE	1			
199	take	take	TRUE	1			
200	take	take	TRUE	1			
201	take	take	TRUE	1			
202	take	take	TRUE	1			
203	take	take	TRUE	1			
204	take	take	TRUE	1			
205	take	take	TRUE	1			
206	take	take	TRUE	1			
207	take	take	TRUE	1			
208	take	take	TRUE	1			
209	nontake	take	FALSE	0			
210	nontake	take	FALSE	0	210	205	0.976190476
	Total Predicted Labels		=	**296**			
	Total Count		=	**316**			
				0.936708861			

Table C.5: Results of the outdoor classification task (Part 1 of 2)

Testing Using AlexNet Architecture								
Matlab training program name: TransferLearningVideoTNTosSEP17TrashRecycle.m								
SL per category	Predicted labe	Actual label	Predicted label = Actual Label		total count	predicted label number	accuracy	File name
1	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle1.jpg' }
2	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle10.jpg' }
3	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle101.jpg'
4	Trash	Recycle	FALSE	0				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle106.jpg'
5	Trash	Recycle	FALSE	0				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle107.jpg'
6	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle109.jpg'
7	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle11.jpg' }
8	Trash	Recycle	FALSE	0				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle111.jpg'
9	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle114.jpg'
10	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle115.jpg'
11	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle116.jpg'
12	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle119.jpg'
13	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle12.jpg' }
14	Trash	Recycle	FALSE	0				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle121.jp
15	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle130.jp
16	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle133.jp
17	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle137.jp
18	Trash	Recycle	FALSE	0				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle138.jp
19	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle143.jp
20	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle15.jpg
21	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle157.jp
22	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle158.jp
23	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle162.jp
24	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle164.jp
25	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle166.jp
26	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle172.jp
27	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle175.jp
28	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle177.jp
29	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle183.jp
30	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle184.jp
31	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle190.jp
32	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle200.jp
33	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle202.jp
34	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle204.jp
35	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle208.jp
36	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle21.jpg
37	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle210.jp
38	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle211.jp
39	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle215.jp
40	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle217.jp
41	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle221.jp
42	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle225.jp
43	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle236.jp
44	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle237.jp
45	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle240.jp
46	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle243.jp
47	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle249.jp
48	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle25.jpg
49	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle254.jp
50	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle269.jp
51	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle270.jp
52	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle271.jp
53	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle31.jpg
54	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle32.jpg
55	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle35.jpg
56	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle38.jpg
57	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle39.jpg
58	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle46.jpg
59	Trash	Recycle	FALSE	0				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle52.jpg
60	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle55.jpg
61	Trash	Recycle	FALSE	0				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle56.jpg
62	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle71.jpg
63	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle73.jpg
64	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle79.jpg
65	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle87.jpg
66	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle89.jpg
67	Recycle	Recycle	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle9.jpg'
68	Recycle	Recycle	TRUE	1	68	61	0.897059	{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Recycle\Recycle90.jpg
1	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash1.jpg' }
2	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash100.jpg'
3	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash101.jpg'
4	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash106.jpg'
5	Recycle	Trash	FALSE	0				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash109.jpg'
6	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash111.jpg'
7	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash112.jpg'
8	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash116.jpg'
9	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash12.jpg' }
10	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash129.jpg'
11	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash13.jpg' }
12	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash134.jpg'
13	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash136.jpg'
14	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash139.jpg'
15	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash143.jpg'
16	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash144.jpg'
17	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash153.jpg'
18	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash16.jpg' }
19	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash160.jpg'
20	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash162.jpg'
21	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash166.jpg'
22	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash168.jpg'
23	Recycle	Trash	FALSE	0				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash176.jpg'
24	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash18.jpg' }
25	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash184.jpg'

Table C.6: Results of the outdoor classification task (Part 2 of 2)

26	Recycle	Trash	FALSE	0				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash186.jpg'
27	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash2.jpg' }
28	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash20.jpg' }
29	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash200.jpg'
30	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash204.jpg'
31	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash205.jpg'
32	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash206.jpg'
33	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash225.jpg'
34	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash227.jpg'
35	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash231.jpg'
36	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash24.jpg' }
37	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash241.jpg'
38	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash243.jpg'
39	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash252.jpg'
40	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash254.jpg'
41	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash259.jpg'
42	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash263.jpg'
43	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash265.jpg'
44	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash274.jpg'
45	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash276.jpg'
46	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash278.jpg'
47	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash289.jpg'
48	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash297.jpg'
49	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash30.jpg' }
50	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash300.jpg'
51	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash305.jpg'
52	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash309.jpg'
53	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash317.jpg'
54	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash324.jpg'
55	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash326.jpg'
56	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash327.jpg'
57	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash328.jpg'
58	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash334.jpg'
59	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash338.jpg'
60	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash344.jpg'
61	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash349.jpg'
62	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash358.jpg'
63	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash36.jpg' }
64	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash361.jpg'
65	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash363.jpg'
66	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash364.jpg'
67	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash367.jpg'
68	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash368.jpg'
69	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash371.jpg'
70	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash372.jpg'
71	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash374.jpg'
72	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash375.jpg'
73	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash377.jpg'
74	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash381.jpg'
75	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash382.jpg'
76	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash383.jpg'
77	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash394.jpg'
78	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash396.jpg'
79	Recycle	Trash	FALSE	0				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash399.jpg'
80	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash4.jpg' }
81	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash40.jpg' }
82	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash401.jpg'
83	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash403.jpg'
84	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash406.jpg'
85	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash409.jpg'
86	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash410.jpg'
87	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash411.jpg'
88	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash418.jpg'
89	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash419.jpg'
90	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash42.jpg' }
91	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash420.jpg'
92	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash422.jpg'
93	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash426.jpg'
94	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash43.jpg' }
95	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash65.jpg' }
96	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash66.jpg' }
97	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash72.jpg' }
98	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash8.jpg' }
99	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash80.jpg' }
100	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash81.jpg' }
101	Recycle	Trash	FALSE	0				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash83.jpg' }
102	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash84.jpg' }
103	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash85.jpg' }
104	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash86.jpg' }
105	Trash	Trash	TRUE	1				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash87.jpg' }
106	Recycle	Trash	FALSE	0				{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash92.jpg' }
107	Trash	Trash	FALSE	0	107	100	0.934579	{'C:\Users\Rsultana1\Downloads\TransferLearning_FoodClassification\FileExchangeEntry\MyImageOSTSep17TrashRecycle\Trash\Trash99.jpg' }
	Total Predicted Labels		=	**161**				
	Total Count		=	**175**				
				0.92				