Fall 12-7-2020

Computer Vision For Recycling

Pratima Kandel
University of Mary Washington

Computer Vision for Recycling

Submitted in partial fulfillment of the requirement for the degree of

Bachelors of Science in Computer Science

Submitted by

Pratima Kandel

Under the Supervision of

Dr. Ron Zacharski

Professor, Department of Computer Science

Department of Computer Science, College of Arts and Sciences

University of Mary Washington

November 25, 2020

Contents

Acknowledgements

I am indebted to my supervisor, Dr. Ron Zacharski who has treated me with utmost patience and empathy throughout the completion of this research project and thank him for his mentorship and co-operation; I thank my committee members Dr. Stephen Davies and Professor Gusty Cooper from the department of Computer Science for their guidance and support.

Pratima Kandel

Department of Computer Science

University of Mary Washington

Abstract

Americans recycle 32.1 percent of all the waste they create, as confirmed by the latest report from the Environmental Protection Agency [1]. However, the underlying issue that remains is that most Americans are not equipped with the knowledge of the correct methods of recycling – and the deficiency of that knowledge is the greatest quandary to ensuring that the country is "green" and environment friendly. A survey of two thousand American citizens revealed that 62 percent of them worry that this inadequate knowledge is causing them to recycle incorrectly [2].

The aim of this research is to develop an Android App, where with the use of one's smartphone camera, a person can capture an image of an item that they wish to recycle and the output displayed on the app will be either plastic, glass, metal or garbage(unknown). The first component of the research investigates whether a usable model can be built that can be used to robustly identify recyclable objects. A deep convolutional neural network (CNN) is built in Python and trained on a labeled data set of a thousand product images from various perspectives, to determine whether the object that is to be recycled is composed of plastic, metal or glass. In order to provide the most efficient approach, I experimented on well-known deep convolutional neural network architectures. By implementing transfer learning and fine tuning to the pre-trained models with a common data augmentation strategy ResNet101V2 model provided the best result with 82% test accuracy. A larger data set is required to reduce overfitting and increase the accuracy.

The chief purpose of this project is to develop an application based on a deep learning model that aids users to correctly identify the nature of objects that they deem recyclable and to widen the scope of healthy recycling - of household and domestic goods, which is a tiny but indispensable and effective individual step that needs to be taken in order to combat pollution and in the long run, prevent climate change while there is still time to do so.

Introduction

In the year 2018, the Revision of World Urbanization Prospects produced by the Population Division of the UN Department of Economic and Social Affairs (UN DESA) estimated that 55% of the world's population currently live in major cities and that figure is expected to expand up to 68% by 2050 [3]. The most pressing issue of urbanization and industrialization is the unrestrained rise in the generation of municipal solid waste (MSW) across the world. Annually, the world generates 2.01 billion tonnes of municipal solid waste; with at least 33 percent of it that does not undergo management in an environmentally safe process. Looking ahead, by 2050, this hazardous global waste is expected to grow to a staggering 3.40 billion tonnes, more than double the population growth forecast over the same period. Although comprising only 16 percent of the world's population, high-income countries are accountable for generating roughly 34 percent, or 683 million tonnes, of global waste [4].

The United States Environmental Protection Agency assessed that the total generation of municipal solid waste in the United States in 2018 alone was 292.4 million tons, approximately 23.7 million tons more than the amount generated in 2017. In the same fashion, the recycling rate in 2017 was 35.2 percent [3]. The EPA projects that about 75 percent of waste emanating from the U.S. can be recycled [5]. By contrast, recycling rates are even lower in under-developed countries. We may cite the South Asian country of Nepal as a case study where about 5% of MSW are recycled [6] and according to the publication of Asian Development Bank only six municipalities out of 58, use sanitary landfill sites for final disposal, whereas the 45 municipalities practice open dumping - including riverside and roadside [7].

For a foreseeable sustainable future, it becomes a vital necessity to develop an effective recycling system. Recycling plays an undeniably fundamental role in the environmental as well as economic sustenance of our planet. According to Towhid Babazadeh et al., the involvement of citizens in the recycling process constitutes the key role in targeting a higher percentage of recycling [8]. In addition, it is important to mention that products and packaging are becoming more multifaceted, complex and diverse. There are wide varieties of plastic alone being used to package the commonplace items we purchase on a daily basis. This complexity has in many ways altered consumers' understanding of what they deem recyclable [9]. At present, the average recycling contamination rate among communities and businesses sits at around 25% [10]. This is an indicator that roughly 1 in 4 items placed in a recycling container is actually not recyclable through curbside programs which in turn lead to colossal problems for the recycling economy [9]. Contamination significantly increases the cost to process recyclables and exercises a direct impact in the quality of recyclables entering the commodity markets. For instance, when a commodity made of paper or cardboard gets contaminated by food or liquids in the recycle container, apart from altering its quality, it also loses its ability to be recycled, thus being reduced to trash.

In the year 2018, the state of California was compelled to shut down more than 1000 recycling centers and processing plants due to reasons similar to the aforementioned issue. The collapse of recycling is primarily due to high contamination levels in the recycling stream which implies that the general populace disposes massive amounts of untreated and unidentified "garbage" in recycling bins. [11]

Contamination stems directly from a lack of awareness and recycling education among the citizens. We therefore can no longer neglect the need for a more well-organized and effectual system to help citizens distinguish the recyclable materials from non-recyclable ones; and the foundation of my research work lies on this very threshold. From this study, we can gather that by utilizing computer vision and deep learning algorithms, we can effectively help citizens correctly identify recyclable materials. In this research, I propose and formulate a computer vision android application using CNN algorithms that can recognize three types of recyclable materials composed of plastic, metal, and glass and an additional garbage class that includes non-recyclable items or unknown materials. It is hoped that this proposed technique can help to expedite the recycling initiative.

The remainder of the paper is structured as follows:

Section I offers an introduction to deep learning and CNN and an overview of related work. Section II expounds the architecture of the application and details of the core components. Section III presents more intricate details about the dataset used. Section IV elucidates the selection of the best deep convolutional neural network architectures for the task of classification. The experimental results and analyses of the image recognition process are exhibited in Section V. The process and the result of implementing the final model to a mobile application is explicated in Section VI. Section VII consists of the challenges and encounters of the model and the application. The final Section VIII delves into the conclusions and future work.

I. Related Work

I-A. An Introduction to Deep Learning and Convolutional Neural Network

Deep Learning*:* Deep learning is a machine learning technique also referred to as deep structured learning or hierarchical learning, where the algorithms built are structured in such a way that they mimic the human brain [12]. Perhaps, a straightforward way to comprehend deep learning is through the means of a succinct analogy such as this one.

Imagine a toddler whose first word is “dog." The toddler learns what a dog is and is not by pointing to objects and saying the word dog. The parent says, "Yes, that is a dog," or, "No, that is not a dog." As the toddler continues to point to objects, he becomes more aware of the common features that all dogs possess. What the toddler does, without knowing it, is clarify a complex abstraction, the concept of dog, by building a hierarchy in which each degree of abstraction is created with knowledge that was gained from the preceding layer of the hierarchy [13].

“Deep learning algorithms are stacked in a hierarchy of increasing complexity and abstraction.[13]” These hierarchies consist of large numbers of hidden layers, and in each of these layers there are non-linear processes to extracts features from the data and transform the data to different levels of abstraction. The first few layers extract low level features. For instance, the input data is a matrix of pixels, whereby the first layer abstracts the pixels and recognizes the edges of features in the image. Likewise, the upcoming layers combine features to create a complete representation of the image. The data is passed from layer to layer and each layer transforms the data turning the output of one layer as the input for the other. [12]

Convolutional Neural Network: A Convolutional Neural Network is one of the common neural networks that belong to the family of methods which categorically fall under deep learning. It is a deep neural network that was originally designed for the purpose of image analysis. A CNN contains two basic operations - convolution and pooling as shown in figure 1. The convolution operation through the usage of relevant filters, extracts features (feature map) from the data and preserves the spatial information of the data. Figure 1 shows a convolutional layer applying to 3x3 patches of the image and figure 2 shows the result of that convolution with the depth of the kernel being 64. The pooling operation provides an approach to down sampling feature maps by summarizing the presence of features in patches of the feature map [14]. The different CNN architectures include VGGNet, GoogLeNet, ResNet, DenseNet and MobileNet. While researching in this area, I experimented on several different deep convolutional neural network architectures to find the best CNN architecture that provides the highest accuracy when applied to the purpose of classifying recyclable vs non-recyclable objects.

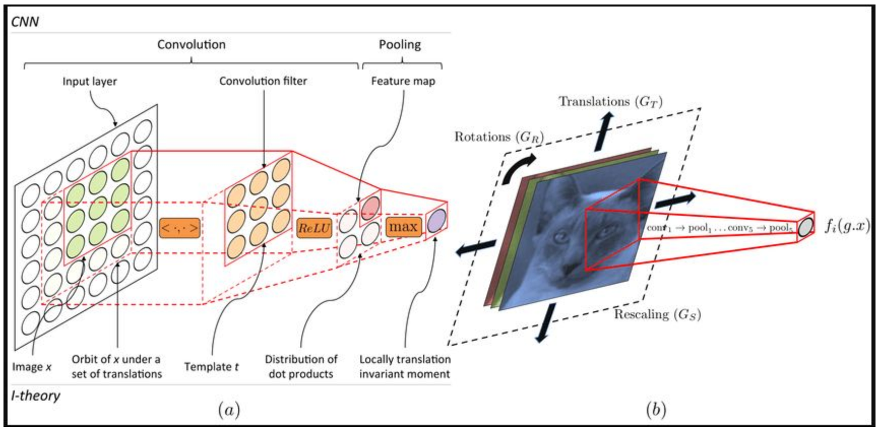

Figure 1. In modern CNNs, there are multiple convolutional and pooling layers [15]

Figure 2. shows the relevant filters and feature maps of the first layer (VGG16 model)

I-B. Deep Learning image-based waste classification

In recent years, image-based waste classification has garnered great attention from the research community. Several interesting solutions employing a diversified set of techniques and technologies have been proposed over the last few years. In this section we pin down the most relevant research on the topic of image-based waste classification.

Jay Donovan [16] proposed an image recognition system. He suggested the use of a Raspberry Pi-powered system equipped with a camera, to automatically sort garbage items based on the nature of the materials. The system relies on a custom software model based on Convolutional Neural Networks (CNNs) built on top of Google's TensorFlow 2 AI engine. The system in current implementation can segregate compostable and recyclable items.

Umut Özkaya and Levent Seyfi performed a comparative analysis for the classification of images in a TrashNet data set where the proposed method used a fine-tuned model. The best classification result was achieved using Google Net and an SVM Classifier with an impressive 97.86% accuracy [17].

Chu Yinghao, et al. propose a multilayer hybrid deep-learning system (MHS) to automatically sort waste disposed of by individuals in the urban public area. This system deploys a high-resolution camera to capture waste images and sensors to detect other useful information. The MHS uses a CNN based algorithm to extract image features and a multilayer perceptions (MLP) method to consolidate image attributes and other characteristics information to catalogue waste as recyclable or not. The MHS is trained and validated against manually labeled items, achieving overall classification accuracy higher than 90%. In this study, the researchers have evaluated a total of two classes of 50 different waste items, among which 40 are categorized as recyclable and 10 are labelled as 'others.' The recyclable materials are grouped into 4 main categories: paper, plastic, metal, and glass; the "others" class consists of fruit/vegetable/plant, kitchen waste, and others. [18]

Khasif Ahmad et al. proposed an intelligent fusion of Deep Features for improved waste classification. Utilizing deep architectures, namely AlexNet, GoogleNet, VggNet and ResNet pre-trained on ImageNet, they organized the waste under six classes. The best accuracy achieved was 95.58%, gained by using the Double Fusion (PSO) method. [19]

Victoria Ruiz et. al proposed several CNN architectures such as VGG-16, VGG-19, ResNet, Inception and Inception-ResNet for the automatic classification of waste. The experiments were completed on the TrashNet dataset, where the best classification results were achieved using a ResNet architecture with an average accuracy of 88.66%. [20]

The existing system closest to my specific domain of research is SpotGarbage – the developers of which created a mobile application designed to coarsely segment a pile of garbage in an image. The goal of the application is to enable citizens to track and report garbage in their neighborhoods. The dataset used was obtained through Bing Image Search and the authors extracted patches from the images to train their network. They also operated a pre-trained AlexNet model and obtained a mean accuracy of 87.69%. [21]

The majority of the proposed solutions rely heavily on deep architecture that are mostly fine-tuned or retrained from scratch. There are no feasible solutions or systems optimized for the purpose of educating citizens and laypeople and encouraging them to recycle appropriately. When one recycling garbage can is contaminated, the correct recycling items present within it needs to be thrown away as well in a landfill. This paper therefore, proposes the use of a mobile application to serve as an end-to-end solution that can differentiate between recyclable and non-recyclable household items in real-time with the use of one's android device. Many fine-tuned models were used in this study to select the best deep convolutional neural network architectures for classification.

II. System Architecture

To tackle the problematic issue of recycling contamination and to help improve waste classification efficiency, I present "RecycleRight", an Android application that can efficiently aid individuals in identifying recyclable and non-recyclable household items. Figure 3 depicts the details of the system architecture.

The core components are:

The Mobile Application: This is used to capture images and determine the recyclable objects. The application processes images on-device.
TensorFlow Lite: This is used to run TensorFlow models on mobile. It enables on-device machine learning inference with low latency and a small binary size [22].
Firebase Machine Learning: Firebase is a Google-backed application development software that enables developers to develop high quality apps rapidly [23]. On the other hand, Firebase Machine

learning encompasses all machine learning SDKs for mobile that require cloud based APIs in order to formulate predictions. One such SDK used in RecycleRight is ML Kit.

ML Kit: ML Kit is a mobile SDK that brings Google's machine learning expertise to smartphone apps in a powerful yet easy-to-use package [24]. ML Kit provides many ready-to-use APIs such as Image Labeling, Text Recognition, Barcode scanner etc. It also provides convenient APIs that help you use your custom TensorFlow Lite models in mobile apps [25].

The software architecture of my application consists of 4 stages. In the first stage, a Keras-Tensorflow model is built by implementing transfer learning and fine tuning a pre-trained ResNet101V2 Keras model. In the second stage, the built model is converted to TensorFlow Lite using TensorFlow's TFLiteConverter library. In the third stage, the converted TensorFlow Lite model is hosted with Firebase and additionally the ML Kit SDK is included in the app to keep users up to date with the latest version of my model, thus allowing us to configure ML Kit to automatically download model updates when the user's device is idle or charging, or has a Wi-Fi connection[25]. The TensorFlow Lite model is bundled with the application, allowing the model to be hosted in the cloud and on-device. And finally, ML Kit's custom model APIs are executed in the Android app to perform inference with the Firebase-hosted or app-bundled model [26].

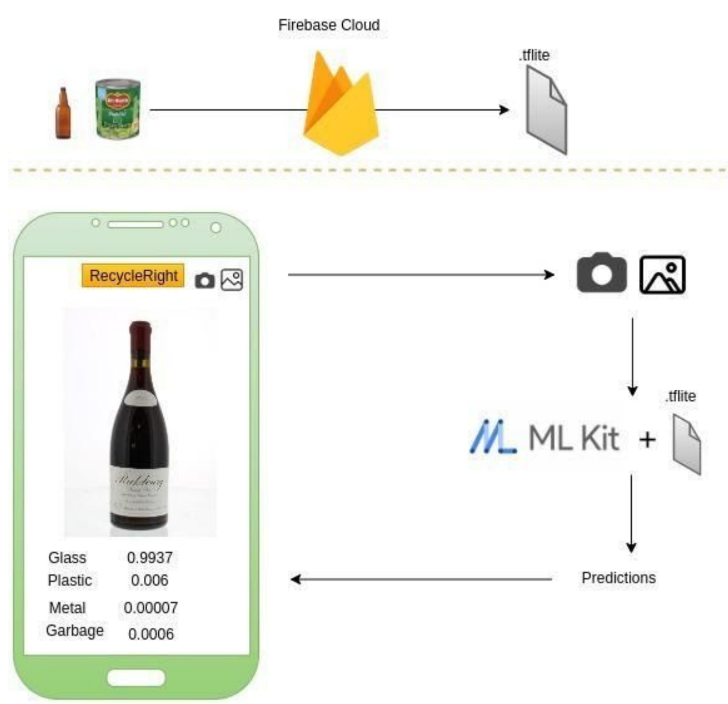

Figure 3. An overview of the system architecture

III. Dataset and Data Collection

The focal point of this study stresses on the need to classify recyclable materials and attempts to prove why waste management and recycling systems are among the essential segments of a sustainable economy. Educating individuals about which household item goes into the recyclable garbage container is an essential first step to eliminate recycling contamination and encourage correct and orderly recycling. Subsequently, our foremost purpose lies in recognizing some of the most common recyclable materials such as glass, plastic and metal. The dataset used to train the model is from the TrashNet [27] dataset. The images on this dataset consist of photographs of garbage taken on a white background [27]. More datasets of household items were collected from Google Open Images Dataset V6 and google images. Each image was resized down to 150 x 150 pixels. Figure 4 shows sample images of the TrashNet dataset. Figure 5 shows sample images collected from other sources. Table 1 entails the number of images for the four classes.

Figure 4. *S*ample images of the TrashNet dataset

Figure 5. *S*ample images of the Google Open Images Dataset and Google Search

Table I. Dataset Information

Recyclable materials types	Number of each material	Training (90%)	Validation (10%)
Plastic	720	647	73
Glass	801	720	81
Metal	486	431	55
Non-Recyclable materials/ Unknown type	580	516	64

IV. Models and Methods

IV-A. Methods

There exist several powerful techniques that have been used in this experiment to build large-scale deep learning models with relatively far smaller amounts of data.

Data Augmentation: It has been observed that training deep neural network models with larger data sets makes the model exponentially more skillful, thus increasing its accuracy and stability while reducing 'overfitting', a condition that occurs when the model learns and absorbs excessive details and noise in the training data to the point that it negatively impacts the performance of the model on the testing data [28].

Additionally, modern deep neural networks may in fact consist of billions of parameters, rendering these networks heavily reliant on big data. Assembling enormous datasets can be an extremely daunting task due to the manual effort required in collecting and labeling data.

However, Data Augmentation is an efficient strategy that allows you to significantly increase the diversity of data to be available for training models, without having to actually collect new data [29]. Today, data augmentation is a sizable tool that is employed in every state-of-the-art model for image classification. Figure 6 shows an example of 6 computer generated augmented images used in my research that includes width/height shifting, zooming, horizontal flipping, size re-scaling and shearing techniques for each data instance to enhance the universality of the training model.

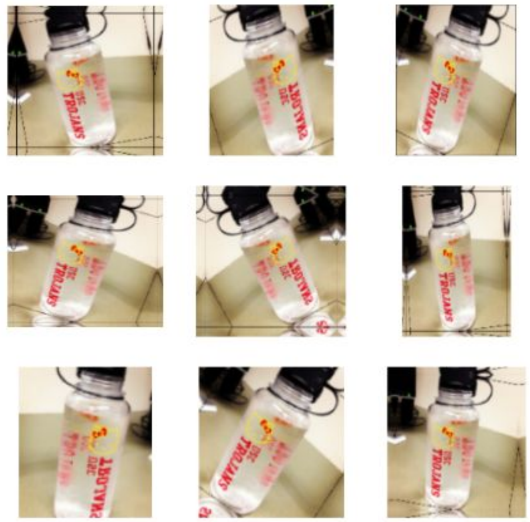

Figure 6. An example of 6 computer generated augmented images

Transfer Learning: Transfer learning is another handy machine learning method whereby a model developed for a certain task is reused as the starting point for a model on a second task [30]. Figure 7 depicts the goal of transfer learning which is to improve learning in Task 2 by leveraging knowledge from Task 1.

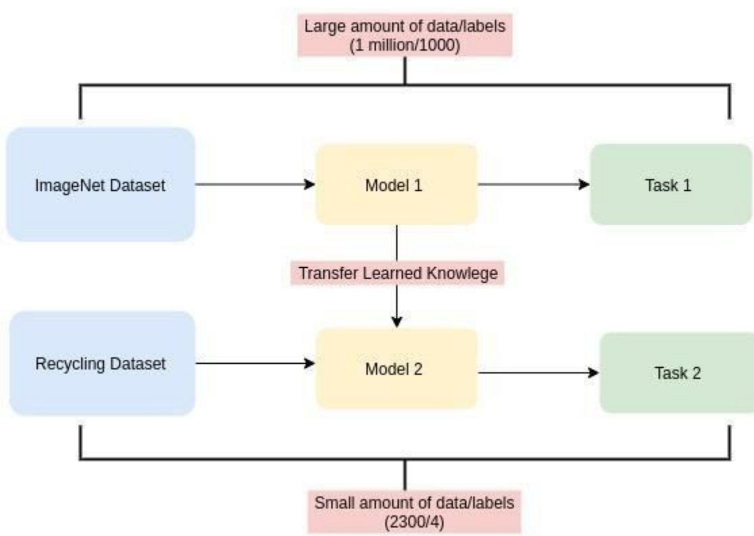

Figure 7. Transfer Learning

When surveying the context of deep learning, it is found to be a well-established fact that models which are designed to solve complex problems require equally exorbitant amounts of data. However, getting vast amounts of labeled data for supervised models can be incredibly arduous, considering the time and effort it takes to label data points. Transfer learning enables us to utilize knowledge from previously learned tasks and apply them to newer, related ones. If we have significantly more data for task 1, we may utilize its learning, and generalize this knowledge (features, weights) for task 2 (which has significantly lesser data) [31]. In the case of problems in the computer vision domain, certain low-level features, such as edges, shapes, corners and intensity, can be shared across tasks, and thus enable knowledge transfer among tasks [31]. Similarly, in this research the pre-trained models that have been used were trained on ImageNet dataset. The ImageNet dataset has millions of images pertaining to almost 1000 categories intended for computer vision research [32]. In this research, feature extraction and fine tuning are the two transfer learning techniques that have been implemented on these pre-trained models.

***Feature extraction**:* Figure 8 presents an example of the various hidden layers of the VGG-16 CNN architecture, one of the several models I have experimented with. In this architecture, the convolution and max pooling layers are the stages that work on feature extraction and the other stages are the classifiers.

Feature extraction can be implemented by instantiating the pre-trained model (VGG-16) network with the weights from the ImageNet, freezing the convolution and max pooling layers, removing the last fully connected classifier layer that takes a probability for each of the 1000 classes in the ImageNet and replacing it with my layer that instead takes probabilities for the four classes. The convolutional base extracts all the features associated with each image and is trained on the classifier that determines the image class given that set of extracted features [33].

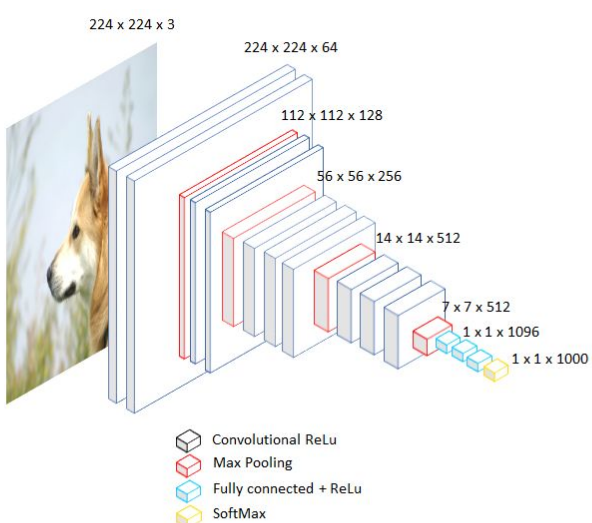

Figure 8. VGG-16 Keras model architecture [34]

Fine Tuning: In the preceding feature extraction experiment, the convolution and pooling layers were frozen and the classification stage layers were replaced, and in doing so, the knowledge that the pre-trained network formerly obtained from its previous task (ImageNet dataset) was preserved. That knowledge is then used to extract features from the current sample (Recycling dataset). Fine tuning, however, allows us to retrain not only the classifier stage (i.e. the fully connected layers) but also the feature extraction stage, i.e. the convolutional and pooling layers [31].

The first few layers in a neural network detect simpler and more general patterns. The deeper we advance into the architecture, the more specific to the data set and more complicated the patterns they detect [34]. Consequently, we could allow the last blocks of convolution and pooling layers to be retrained by unfreezing those blocks and jointly train the base model with both the newly added classifier and the unfrozen last layers. This allows us to 'fine-tune' the higher-order feature representations in the base model in order to align them with more precision for our specific task. Figure 9 depicts an overview of a fine-tuned model [33].

The fine-tuning steps used for the pre-trained models in this research are discussed hereunder.

Step 1: Replacing the last layer (softmax layer) of the pre-trained network with the new softmax layer that is relevant to my problem at hand. For example, a pre-trained network on ImageNet comes with a softmax layer with 1000 categories while on the other hand, my task is a classification of four categories, viz. plastic, metal, glass and unknown.

Step 2: Freezing the weights of the top layers of the pre-trained network keeps the weights that the model learned from its previous task, fixed. We freeze the top layers because they capture universal features like curves and edges which are also relevant to my new problem while unfreezing the last layers to get the network focus on learning data set to the current specific task.

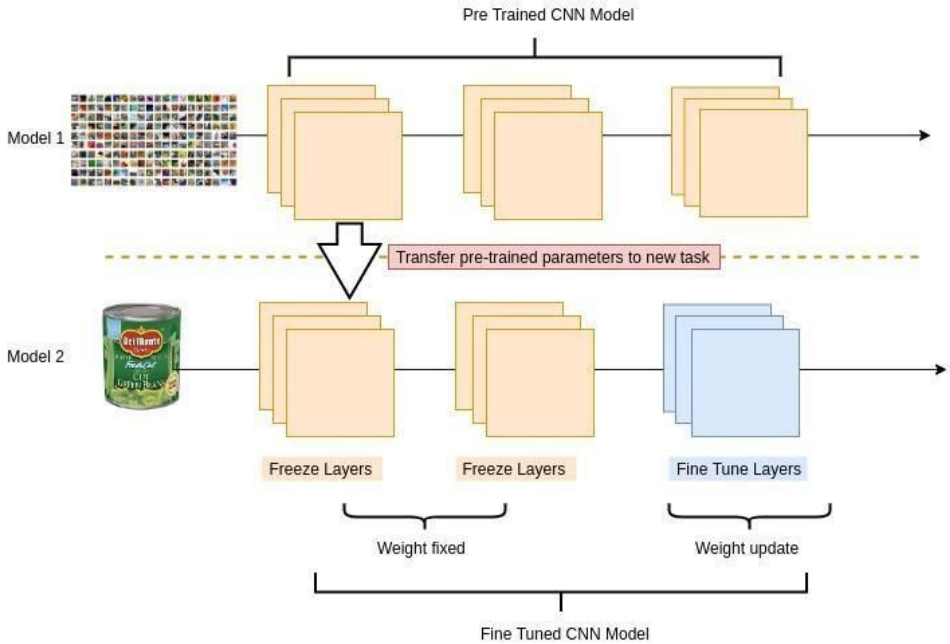

Figure 9. Fine Tuned Model

SoftMax: Softmax function in applied machine learning is used as an activation function in a neural network model. In this research, the softmax function is used for the last layer of the neural network where the classification task is completed. The neural network in my research is configured to output four values, one for each class in the classification task, and the softmax function is used to normalize the outputs, converting them from weighted sum values into probabilities that sum to one. Each value in the output of the softmax function is interpreted as the probability of membership for each class. [35]

For instance, ResNet101V2 convolutional neural network recognizes if an image is composed of plastic, metal, glass and unknown. Thus, the image must either be plastic, metal, glass or unknown but cannot simultaneously be more than one; therefore, these classes are mutually exclusive. The final fully connected layer of this network typically produces values like [-7.98, 2.39, 4.39, -7890] which are not normalized and cannot be interpreted as probabilities [36]. After adding the softmax layer to the network, the numbers are translated into a probability distribution as shown in Figure 10.

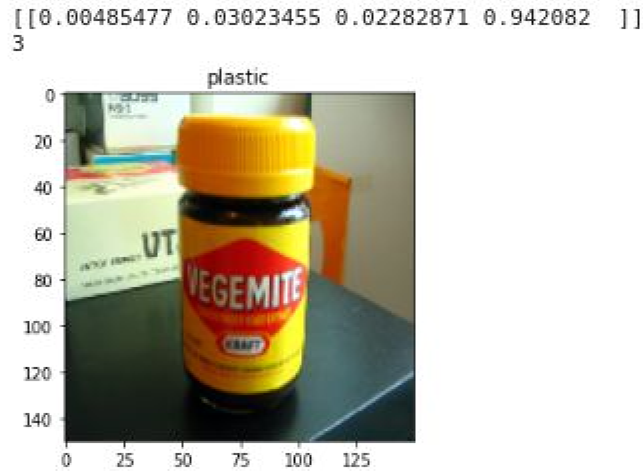

Figure 10. Probability distribution translated by the SoftMax function in the ResNet101V2 neural network

Optimization algorithm: In building a functional deep neural network, the goal is to reduce the difference between the predicted output and the actual output (also known as a Cost function (C) or Loss function). Hence, machine learning optimizers come into play when it comes to minimizing cost functions. Optimizers are algorithms or methods used to change the attributes of your neural network such as weights and learning rate in order to reduce the losses [37]. There are numerous optimization algorithms or strategies that are available for reducing the losses and to provide the most accurate results possible such as gradient descent, stochastic gradient descent, momentum, adgrad, etcetera. During the course of this research, I experimented with the Adam optimization algorithm among others.

Adam: It is an extension to Stochastic gradient descent and can be used in place of classical stochastic gradient descent to update network weights more efficiently [38]. Epistemologically, the name Adam is derived from adaptive moment estimation. Adam is an adaptive learning rate method, meaning it computes individual learning rates for different parameters [39]. The method of optimization is computationally powerful, straightforward to implement, has insufficient memory necessity, and is appropriate for problems that are wide in terms of data and/or parameters. Figure 9. compares training cost between Adam and other optimizers trained on the MNIST dataset with Adam securing huge performance gains in terms of speed of training than other optimization algorithms.

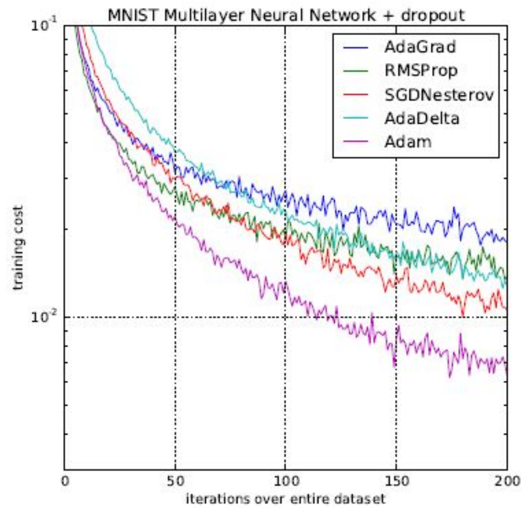

Figure 11. Adam and other optimizers trained on the MNIST dataset with Adam securing huge performance gains (speed of training) [40]

IV-B. Selection of CNN Architectures for Classification Task

VGG-19, VGG-16: VGGNet was created by the Visual Geometry Group at the University of Oxford [41]. ImageNet is an image database consisting of 14,197,122 images [42]. It is an initiative to help researchers, students and others in the field of image and vision research. ImageNet regularly hosts contests, and one such contest is the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) which challenges researchers around the world to innovate solutions that yield the lowest top-1 and top-5 error rates [43]. The competition gives out a 1000 class training set of 1.2 million images, a validation set of 50 thousand images and a test set of 150 thousand images. The VGG Architecture, in the 2014 contest, out-shined other state of the art models.

Aakash Kaushik lays out the exposition of the *VGG-19* by describing it as “a variant of the VGG model that is composed of 16 convolution layers, 3 fully connected layers, 5 max pool layers and 1 SoftMax layer.” It is 19 layers deep when calculating only the layers that have trainable weights, i.e., 16 convolution layers and 3 fully connected layers; whereas *VGG-16* comprises 16 deep layers. [44] Additionally, VGG-16 achieved 92.7% top-5 test accuracy in ImageNet. It was one of the most successful and popular models submitted to ILSVRC-2014 [45]. It makes amelioration over AlexNet by replacing large kernel-sized filters (11 and 5 in the first and second convolutional layer respectively) with multiple 3×3 kernel-sized filters placed one after another [46]. VGG16 was trained for weeks and availed NVIDIA Titan Black GPUs [45]. The default input image size of VGG Models is 224 x 224 [45].

Inception V3: Inception V3 by Google is the third version developed by Google in a series of Deep Learning Convolutional Architectures [47]. It is an advanced iteration of the Inception V2 model and both the models are hybrid structures of inception modules and residual connections. According to C. Szegedy, the "Inception module functions works by performing a convolution on an input with not one, but three different sizes of filters (1x1, 3x3, 5x5) where max pooling is performed and the resulting outputs are concatenated and sent to the next layer." [48]. By structuring the model to perform its convolutions on the same level, the network gets progressively wider; not deeper as shown in figure 12. On the other hand, residual connections make deeper and wider inception networks more efficient with lesser hyper-parameters [47]. Earlier inception module implementation required more training resources and time, with residual connection improvement, requirements reduced and models that turned out to be more efficient to train.

Inception V3 was trained using a dataset of 1000 classes from the original ImageNet dataset [47]. Additionally, Inception V3 was a first-runner up model for the ImageNet Large Visual Recognition Challenge in 2014. Apart from the previously existing features in Inception V2, Inception V3 contains RMSProp Optimizer, Batch Normalizations in the auxiliary Classifiers and label smoothing, a type of regularizing component added to the loss formula to prevent overfitting by subverting the network from becoming overconfident about a class. The default input image size of Inception-v3 is 299×299 [49]. Figure 13. depicts the layers of Inception V3.

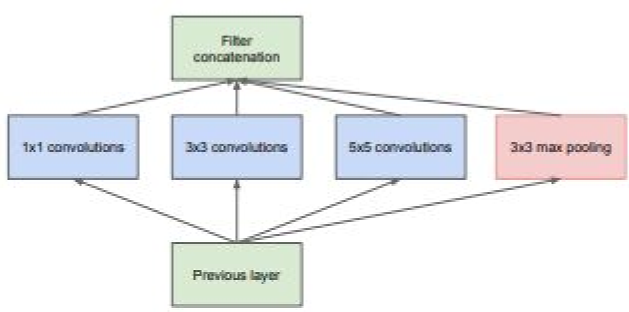

Figure 12. Inception module, naive bayes [48]

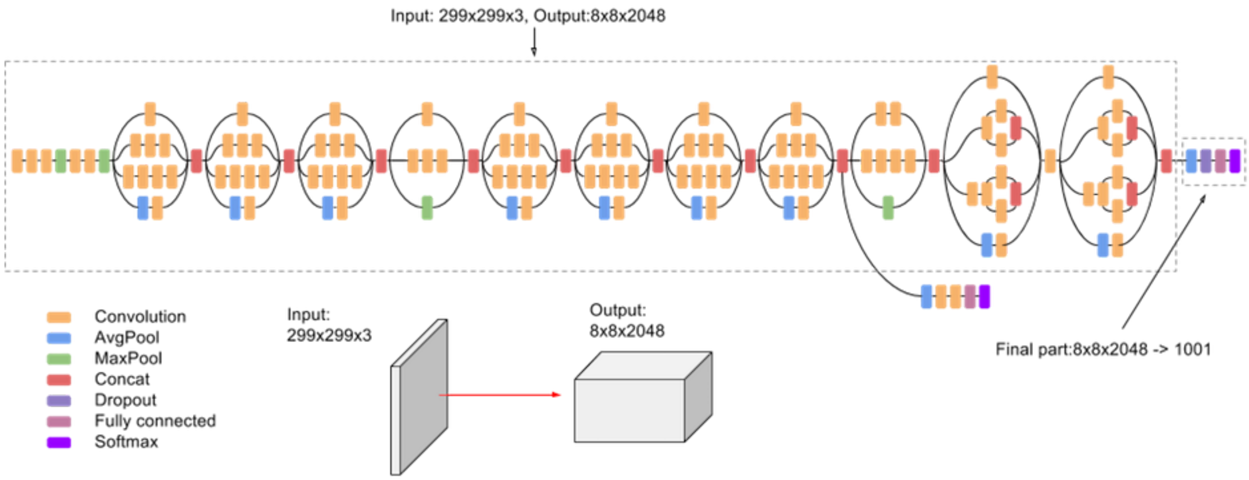

Figure 13. Inception V3 keras model architecture [50]

Xception*:* The Xception model is another deep convolutional neural network architecture which was inspired by Inception. This model replaces Inception's modules with depth wise separable convolutions. It can be described as a stack of linear depth-wise separable convolution layers with residual connections. When compared to Inception V3, Xception slightly outperforms Inception V3 on the ImageNet dataset and drastically "outperforms the same on a larger image classification dataset comprising 350 million images and 17,000 classes". Xception input size is 229x229. [51]

MobileNets: MobileNets uses depth-wise separable convolutions to build light weight deep neural networks [52]. In a single step, a standard convolution both filters and combines inputs into a new set of outputs. This is further split into two layers by the depthwise separable convolution - a separate layer for filtering and a separate layer for combining. The depthwise convolution (filtering layer) applies a single filter to each input channel. The pointwise convolution (combining layer) then applies a 1×1 convolution to combine the outputs of the depthwise convolution [53]. Figure 14. compares standard convolution with depth-wise separable convolution.

It is a rather competent model proposed by Google's research team for mobile and embedded vision applications. In addition to depthwise convolution, MobileNets are thinner with lesser parameters because it prevents the model from producing additional hyper-parameters by using the reduced representations of the input and depending on model shrinkage parameters. Due to its compact nature, it is able to train faster with lower resource requirements. MobileNet input size is 224x224. [52]

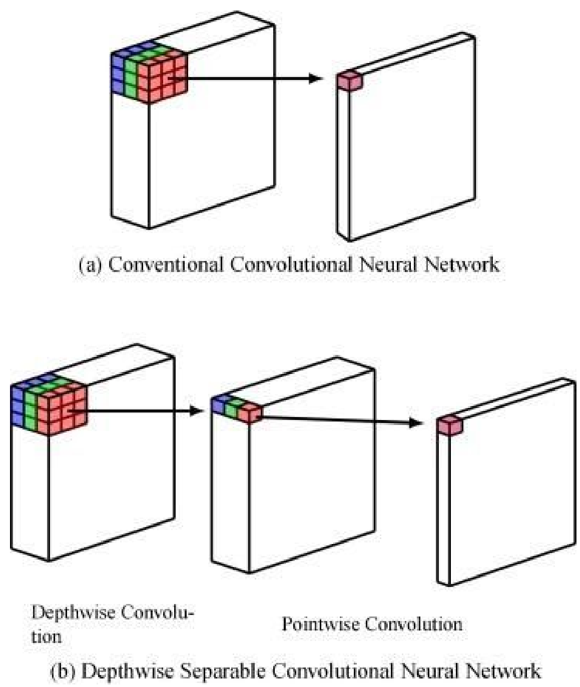

Figure 14. Comparison between conventional CNN and Depthwise Separable CNN [54]

ResNets: ResNet, an abbreviation for Deep Residual Networks model was the winner of the ImageNet challenge in 2015. ResNet is composed of 150+ layers that allow us to successfully train extremely deep neural networks with ease. Due to vanishing gradients, training very deep neural networks was a strenuous task prior to the inception of ResNet.

ResNet introduced a residual function between the layers called residual mapping. Residual mapping acts as a bridge between the input and the output of the blocks, proposing an elegant formation of residual blocks with three convolutional layers accompanied with batch normalization and rectified linear unit (ReLU) activation function. ResNet-101 is 101 layers deep. ResNet input size is 224x224. [55]

Densely Connected Convolutional Networks: Densely connected Convolutional Networks or DenseNets are one of the most efficient deep convolutional neural network structures because they contain shorter connections between layers close to the input and output [56]. DenseNets require “fewer parameters than an equivalent traditional CNN, as there is no requirement of learning redundant feature maps” [57]. Another cumbersome hassle with very deep networks was the problem to train, because of the flow of information and gradients. “DenseNets solve this issue since each layer has direct access to the gradients from the loss function and the original input image” as shown in figure 15 [57]. DenseNets are superior to previous research with their abilities to reduce parameters, strengthen the feature propagation and solve vanishing gradient problems as the network grows [56].

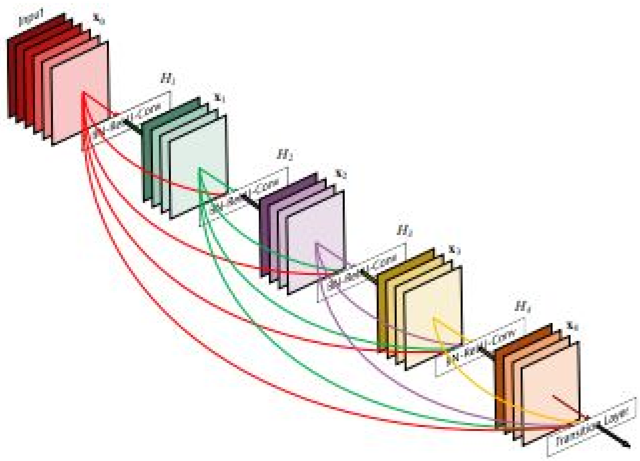

Figure 15. Exhibits a 5-layer dense block (DenseNet).
Each layer takes all preceding feature-maps as input. [56]

V. Experiments and Results

The experiments of this research were executed using Keras library with TensorFlow back end (version 2.3.0) on Google Colaboratory [58]. Around 35 testing images were collected from Google Search and the remaining pictures were clicked using a smartphone. In this analysis, there is no data augmentation method used on the testing and the validation data.

For the transfer learning experiments, a feature extraction technique was carried out by instantiating the models with the pre-trained weights from the ImageNet dataset. Table 2 shows the result of the classification accuracy of each pre-trained model.

Table II. Result of training from feature extraction experiment

Pre-trained Model	Accuracy of models % without fine tuning Softmax	Test Accuracy
VGG-16	80%	77%
VGG-19	74%	67%
MobileNetV2	78%	72%
InceptionV3	80%	73%
DenseNet121	81%	73%
ResNet101V2	81%	79%
Xception	82%	73%

In order to take advantage of model capacity, fine-tuning experiments were performed on the selection of the best performing models on the feature extraction experiment. The learning rate of Adam optimizers was kept as 0.00001, without weight decay. For all training experiments, the batch size was selected as 35. Table 3 provides classification accuracy of the fine-tuned models.

Table III. Result of training from fine tuning experiment

Pre-trained Model	Accuracy of fine tuned models % Softmax	Test Accuracy
VGG-16	79%	Low result when fine-tuned
MobileNetV2	80%	76%
InceptionV3	84%	76%
ResNet101V2	84%	82%
Xception	83%	76%
DenseNet	85%	Heavy Overfitting - Model not used

In Table 2, most methods have the lowest accuracy success of a mere 73%. The highest accuracy rate is secured by ResNet101V2 with 79%. The best classification accuracy with fine-tuned models in Table 3 is 82%. Likewise, ResNet101V2 has the highest accuracy on both the experiments. Figure 16 shows accuracy and loss curves for the pre-trained ResNet101V2 model on the feature extraction experiment. Similarly the feature experimented model is further trained on the fine tuned ResNet101V2 model, figure 17 shows the accuracy and loss curves for the fine-tuned model. Figure 18 presents examples of the predictions made by the ResNet101V2 fine-tuned model.

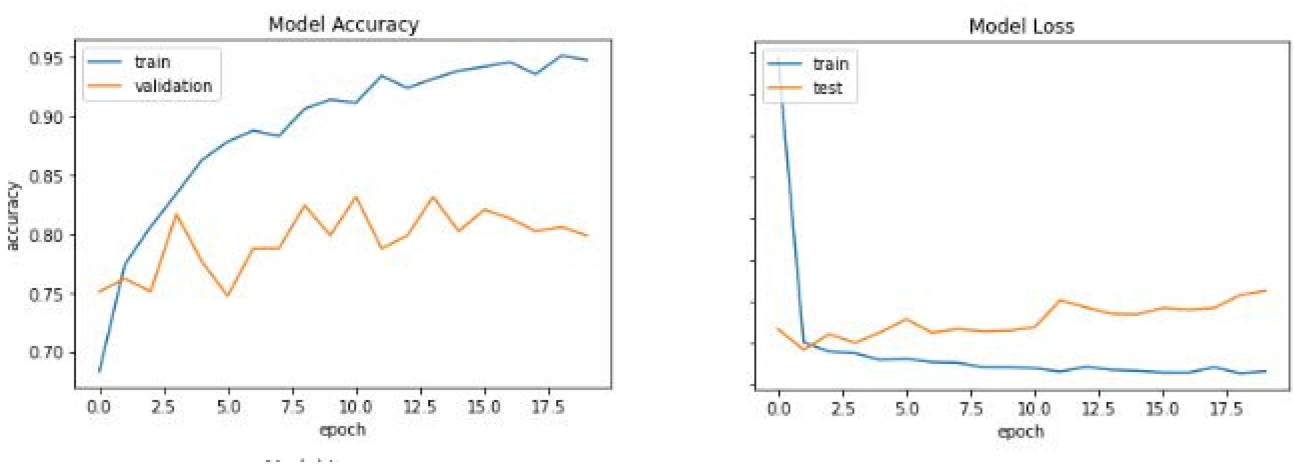

Figure 16. Feature Extraction Experiment - (ResNet101V2) Accuracy and Loss Curves

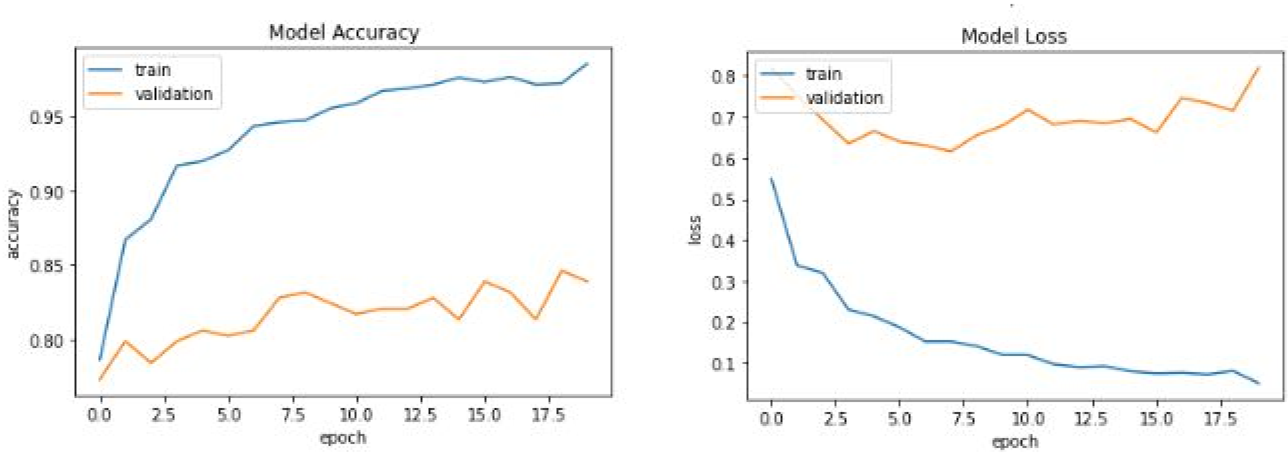

Figure 17. Fine Tuned ResNet101V2 Model Accuracy and Loss Curves

Figure 18. Predictions made by the ResNet101V2 fine-tuned model.

VI. Implementing the Final Model to ML Kit Android Application

The fine-tuned ResNet101V2 base model is then converted to a TensorFlow Lite model. After creating a project on Firebase console, the TensorFlow Lite model is bundled with the app where the custom model is run to perform inference by using ML Kit SDK. In the RecycleRight Application, there are two methods to provide an input image to the host model either through the use of a smartphone's camera or by choosing an existing image from the smartphone's gallery as shown in Figure 19. Similarly, when provided with an input image, the model performs inference and predicts if the image is composed of plastic, metal, glass or if it is garbage(unknown). Figure 20 demonstrates predictions made by using a smartphone camera.

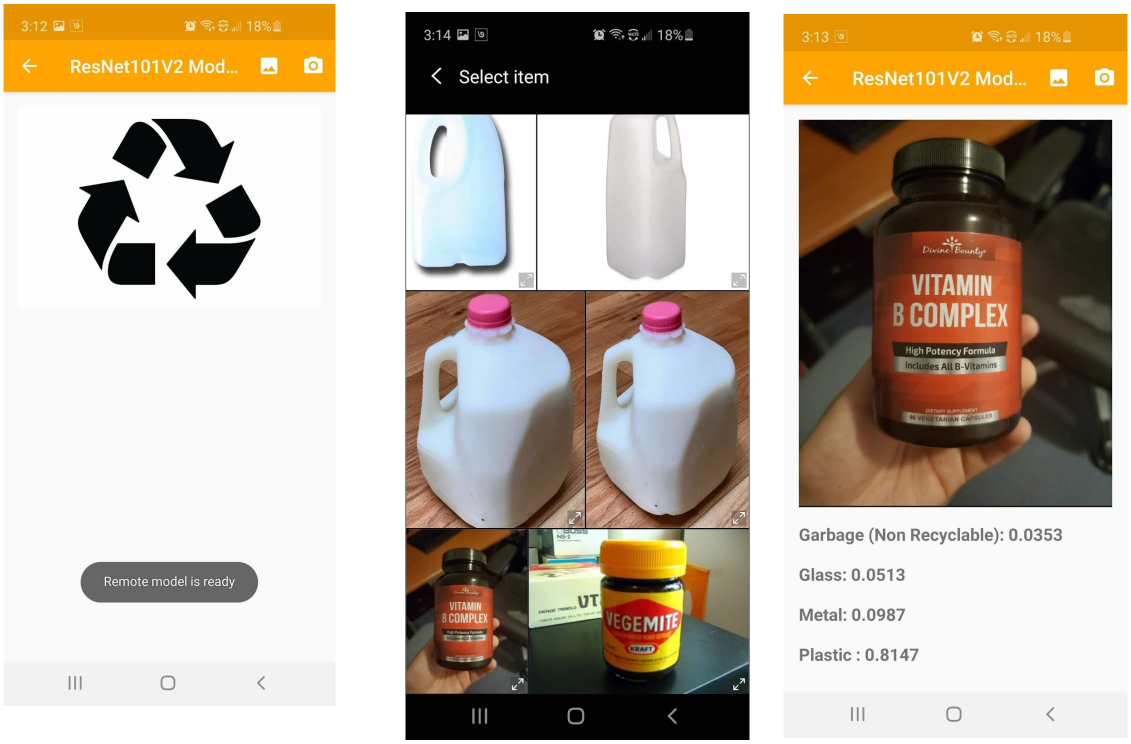

Figure 19. Demonstrates choosing an existing image from the smartphone's gallery

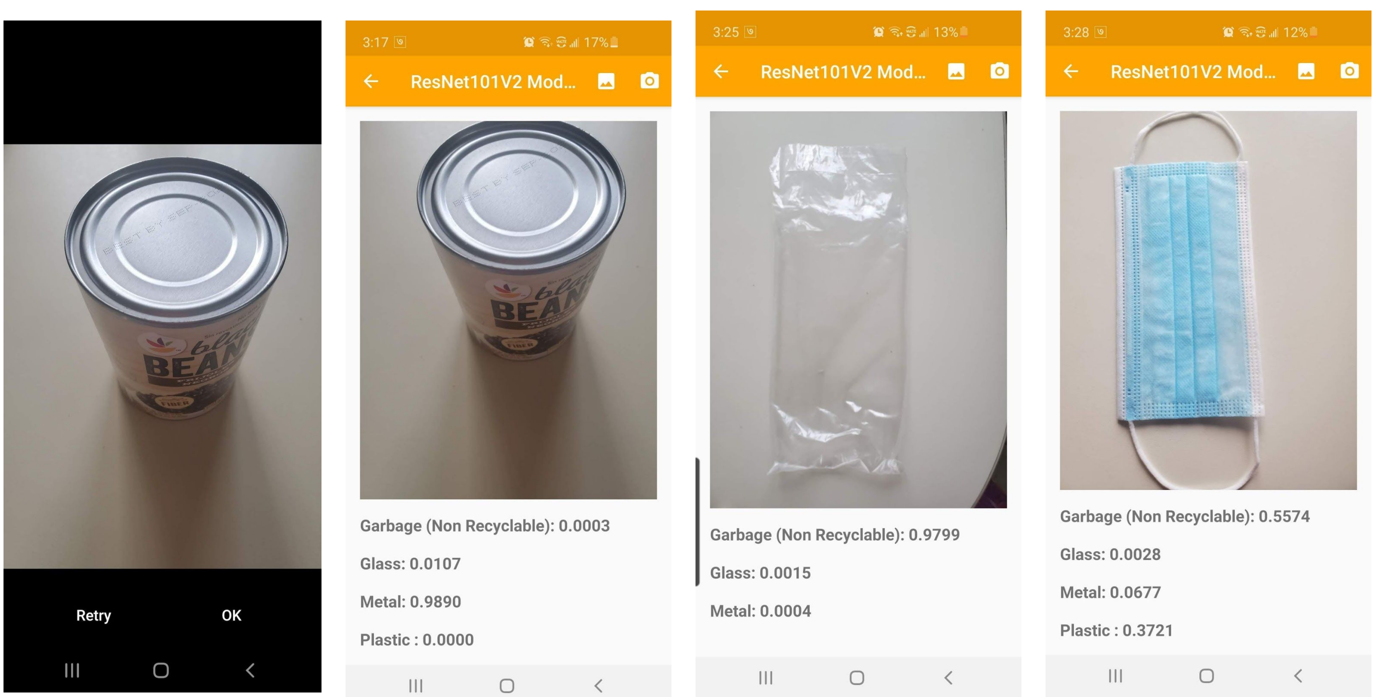

Figure 20. Demonstrates clicking an image with the use of the device's camera

VII. Problems

There are innumerable numbers of household items that are capable of being recycled and each of these items come in different shapes, sizes, forms, colors and packaging labels. The hurdle with deep learning is that just a few samples of each material proves to be insufficient data for the model to determine the nature of the item in question, whether the item is composed of glass, plastic or metal. One of the most persistent quandaries at the moment is the wide varieties of possible data. As we advance, there is an increase in products being introduced with each product having its own idiosyncrasies and unique nature, thereby rendering it more problematic for the model to predict. Therefore, to create a more accurate system, there needs to be a large and continuously growing data source.

In this research the images used for training are from TrashNet dataset in which images have a white background. The data collected from other sources have colored backgrounds but none of these data have noises or other objects in their background. Likewise, when using the "RecycleRight" app, the user is provided with better prediction if the picture of the determinant item is clicked with less background noise. Figure 21 highlights the various predictions made by the same model of the same item clicked on different backgrounds.

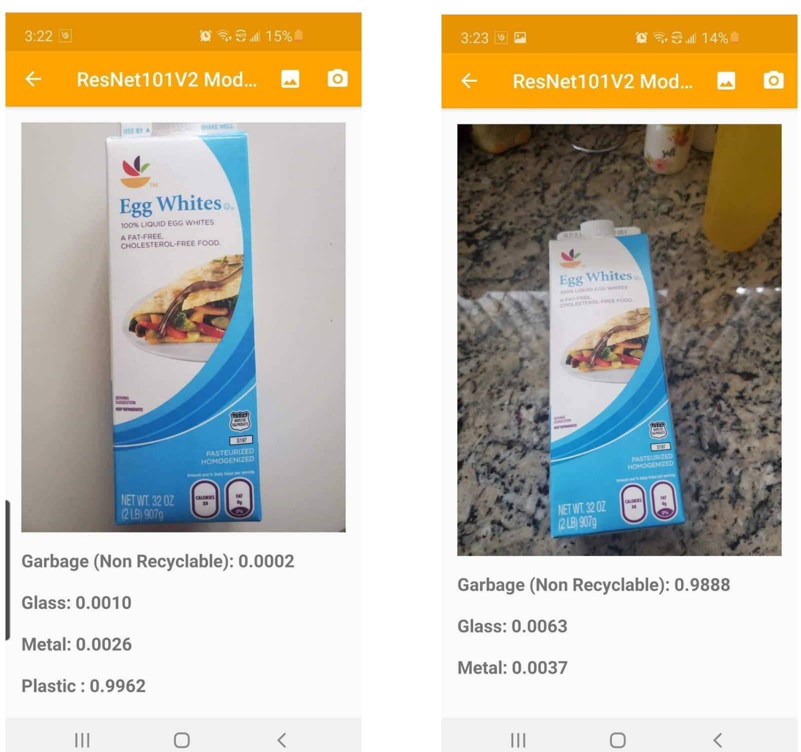

Figure 21. Predictions made by the model on the same item with different backgrounds

VIII. Conclusion and Future Work

Over the years, product packaging has been getting increasingly complex, making it harder for the layman to understand and sensibly differentiate between what can and cannot be recycled, which in turn promotes the generation of municipal solid waste (MSW) to keep increasing at a steady rate.

According to the World Bank's review report, it is expected that MSW will reach 2.2 billion tons per year by 2025 [4]. It has become abundantly evident that recycling is a viable solution to this enormous problem. The process of recycling, however, requires that the citizens reduce recycling contamination and recycle more often making it easier for the workers who separate the waste materials which is a time-consuming process which is currently (and most often) performed manually by hand.

As an attempt to solve this major stumbling block, I present in this research an innovative solution for educating citizens about what items can be recycled while simultaneously encouraging the art of recycling and of its application in daily life.

This work utilizes a computer vision android application along with a Convolutional Neural Network (CNN) in order to classify the recycled materials in real time with the use of one's smartphone camera. The CNN classifies and recognizes the waste materials under four categories: Plastic, glass, metal, and garbage. Classification of real-world examples for waste classification is far from being an easy computer vision task. Apart from the factor that there are wide possible data/waste, most of them being unique, waste can also be understood to be a shape-shifting material. When the state-of-the-art convolutional neural networks are meticulously trained with large amounts of data, they would be capable of producing greater results to solve problems of this nature. The end results of our experiments are promising and encouraging. Albeit being trained with just around a 2400 image dataset, all the CNN models were able to score more than 70% in test accuracy. By fine-tuning a 101 layered ResNet101V2 model that was pre-trained on ImageNet dataset and using the data augmentation strategy, I achieved 82% test accuracy. It may be concluded that learning the process of sorting waste materials is feasible with approaches based on modern deep learning techniques.

For future work, I will strategize refining the proposed novel architecture by expanding the dataset, widening the range of the types of waste included and employing the latest advances on convolutional neural network research. I will experiment with more sophisticated connection patterns and different optimization algorithms. The existing model of the app requires users to identify recyclable items by reading the percentage of the material it is composed of. I shall work to upgrade it by replacing the percentage reading, while improving readability and ensuring a smoother user experience, by displaying the materials the item is composed of. As an added feature, I will include facts and trivia about how users can contribute to small-scale domestic recycling. To elucidate further - if the item to be recycled is composed of plastic, the system will display “Recyclable. This item is composed of plastic.” This will be accompanied by the display of additional messages such as “Did you know that the smoke inhaled from a burnt plastic increases the risk of cancer? Thank you for contributing in saving the environment and societal health by recycling!” Apart from that, I will also implement computer vision strategy on the particular app by adding methods that could have the ability to minimize noise in the image background before the image itself is passed to the model to be determined. This solution could significantly elevate and ease the process of classification of recycling materials. Another important addition to the project will be redesigning the app by including pictures and descriptions of the most confusing and popular items that can be recycled, entailed with descriptions and instructions that further explain and elaborate on how to correctly recycle. This is an additional initiation to help citizens have a firmer grasp and better knowledge about the recycling materials.

References

1. United States Environmental Protection Agency. *Advancing Sustainable Materials Management*. Nov. 2020. www.epa.gov/sites/production/files/2020-11/documents/2018_ff_fact_sheet.pdf. Accessed 20 November 2020.

2. Waste Advantage. *Why Americans Aren't Recycling*. 22 Apr. 2019, wasteadvantagemag.com/why-americans-arent-recycling/. Accessed 20 November 2020.

3. Department of Economic and Social Affairs, United Nations. 2018 Revision of World Urbanization Prospects. United Nations, 16 May 2018, www.un.org/en/events/citiesday/assets/pdf/the_worlds_cities_in_2018_data_booklet.pdf. Accessed 20 November 2020.

4. Kaza Silpa, Yao Lisa C., Bhada-Tata Perinaz and Van Woerden Frank. "What a Waste 2.0: A Global Snapshot of Solid Waste Management to 2050. Urban Development." Washington, DC: World Bank. © World Bank. 2018. https://openknowledge.worldbank.org/handle/10986/30317 License: CC BY 3.0 IGO.

5. Rysavy, Tracy F. *Americans Are Bad at Recycling. Here's How the World Does It Better*. Green America, www.greenamerica.org/rethinking-recycling/americans-are-really-bad-recycling-only-because-were-not-trying-very-hard. Accessed 20 November 2020.

6. Troschinetz, Alexis M. and Mihelcic, James R. Sustainable recycling of municipal solid waste in developing countries. Waste Manag. 2009, 29, 915–923, doi:10.1016/j.wasman.2008.04.016.

7. Solid Waste Management in Nepal: Current Status and Policy Recommendations. 2013. https://www.adb.org/sites/default/files/publication/30366/solid-waste-management-nepal.pdf. Accessed on 20 November 2020.

8. Babazadeh Towhid, Nadrian Haider, Mosaferi Mohammad and Allahverdipour Hamid. Identifying Challenges and Barriers to Participating in the Source Separation of Waste Program in Tabriz, Northwest of Iran: A Qualitative Study from the Citizens' Perspective. Resources 2018, 7, 53, doi:10.3390/resources7030053.

9. "The Battle Against Recycling Contamination Is Everyone's Battle." *Waste Management*, 4 Apr. 2018, mediaroom.wm.com/the-battle-against-recycling-contamination-is-everyones-battle/. Accessed 18 November 2020

10. Melrose Recycling Committee. "Waste Management's Recycling Challenge." *Melrose Recycling Committee*, 12 April 2018, melrecyclingcommittee.wordpress.com/2018/04/12/waste-managements-recycling-challenge/. Accessed 18 November 2020

11. "THE RECYCLING CRISIS." *Recycleacrossamerica*, www.recycleacrossamerica.org/us-recycling-collapse. Accessed 12 October 2020

12. Geeky4Tech. "Deep Learning & How It's Work - Every Thing You Should Know." *A Better Way*, Blogger, 28 Sept. 2020, web.archive.org/web/20201101123322/www.geeky4tech.com/2020/09/deep-learning.html. Accessed 12 November 2020

13. Rouse, Margaret. "What Is Deep Learning and How Does It Work?" *SearchEnterpriseAI*, TechTarget, 16 Oct. 2019, searchenterpriseai.techtarget.com/definition/deep-learning-deep-neural-network. Accessed 12 November 2020

14. Kumar, Pratik. "Introduction to CNNs - Without Using MNIST!" *Medium*, Towards AI, 24 Oct. 2020, medium.com/towards-artificial-intelligence/introduction-to-cnns-without-using-mnist-ea62040341d0. Accessed 12 November 2020

15. Morère Olivier, Lin Jie, Veillard Antoine & Chandrasekhar Vijay. Nested Invariance Pooling and RBM Hashing for Image Instance Retrieval. 01 March 2016.

16. Donovan, Jay. "Auto-Trash Sorts Garbage Automatically at the TechCrunch Disrupt Hackathon." TechCrunch, 13 Sept. 2016, techcrunch.com/2016/09/13/auto-trash-sorts-garbage-automatically-at-the-techcrunch-disrupt-hackathon/. Accessed 20 November 2020

17. Özkaya, Umut and Seyfi, Levent. Fine-Tuning Models Comparisons on Garbage Classification for Recyclability. 7 Aug 2019, arXiv:1908.04393.

18. Chu Yinghao, Huang Chen, Xie Xiaodan, Tan,Bohai, Kamal Shyam and Xiong Xiao-Gang. Multilayer Hybrid Deep-Learning Method for Waste Classification and Recycling. *Computational Intelligence and Neuroscience.* 01 Nov. 2018, 1-9, doi:10.1155/2018/5060857.

19. Kashif Ahmad, Khalil Khan and Ala Al-Fuqaha. "Intelligent Fusion of Deep Features for Improved Waste Classification". *IEEE Access.* PP. 1-1. doi:10.1109/ACCESS.2020.2995681.

20. Ruiz Victoria, Sanchez Angel, Vélez Jose and Raducanu Bogdan. "Automatic Image-Based Waste Classification." 05 Oct. 2019. doi:10.1007/978-3-030-19651-6_41.

21. Gaurav Mittal, Kaushal B. Yagnik, Mohit Garg, and Narayanan C. Krishnan. "SpotGarbage: smartphone app to detect garbage using deep learning." *Association for Computing Machinery*, New York, NY, USA, 940–945. Sep. 2016. doi:10.1145/2971648.2971731

22. Farhoodfar, Avid. "Machine Learning for Mobile Developers: Tensorflow Lite Framework". 23 April 2019.

23. Rouse, Margaret. "What Is Google Firebase? - Definition from WhatIs.com." *SearchMobileComputing*, TechTarget, 25 Apr. 2019, web.archive.org/web/20201022161123/searchmobilecomputing.techtarget.com/definition/Google-Firebase. Accessed 20 November 2020.

24. Abeythilake, Udara. "Firebase Text Recognition with Android." *Medium*, Medium , 29 Oct. 2019, medium.com/@abeythilakeudara3/firebase-text-recognition-with-android-e46814c49dfe. Accessed 20 November 2020.

25. "ML Kit for Firebase." *Google*, Google, firebase.google.com/docs/ml-kit. Accessed 18 November 2020.

26. "Custom Models | Firebase." *Google*, Google, firebase.google.com/docs/ml-kit/use-custom-models. Accessed 18 November 2020.

27. Thung, Gary and Mingxiang, Yang. "Classification of Trash for Recyclability Status." 2016. http://cs229.stanford.edu/proj2016/report/ThungYang-ClassificationOfTrashForRecyclabilityStatus-report.pdf. Accessed 16 November 2020.

28. More, Vikas D. *Over Fitting in Context of Machine Learning*. 24 Sept. 2018, moredvikas.wordpress.com/2018/09/24/over-fitting-in-context-of-machine-learning/. Accessed 20 November 2020.

29. Ho Daniel, Richard Liaw and Eric Liang. "1000x Faster Data Augmentation." *The Berkeley Artificial Intelligence Research Blog*, 7 June 2019, bair.berkeley.edu/blog/2019/06/07/data_aug/. Accessed 18 November 2020.

30. Brownlee, Jason. "A Gentle Introduction to Transfer Learning for Deep Learning." *Machine Learning Mastery*, 16 Sept. 2019, machinelearningmastery.com/transfer-learning-for-deep-learning/. Accessed 18 November 2020.

31. Sarkar, Dipanjan. "A Comprehensive Hands-on Guide to Transfer Learning with Real-World Applications in Deep Learning." *Medium*, Towards Data Science, 17 Nov. 2018,

web.archive.org/web/20200501221004/towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a. Accessed 12 November 2020

32. Brownlee, Jason. “A Gentle Introduction to the ImageNet Challenge (ILSVRC).” *Machine Learning Mastery*, 5 July 2019, machinelearningmastery.com/introduction-to-the-imagenet-large-scale-visual-recognition-challenge-ilsvrc/. Accessed 12 November 2020.

33. “Face Mask Detection Using Deep Learning.” *LaptrinhX*, 6 Oct. 2020, laptrinhx.com/face-mask-detection-using-deep-learning-3383460095. Accessed 12 November 2020.

34. Roman, Victor. “CNN Transfer Learning & Fine Tuning.” *Medium*, Towards Data Science, 27 Mar. 2020, towardsdatascience.com/cnn-transfer-learning-fine-tuning-9f3e7c5806b2. Accessed 12 November 2020.

35. Brownlee, Jason. “Softmax Activation Function with Python.” *Machine Learning Mastery*, 23 June 2020, machinelearningmastery.com/softmax-activation-function-with-python/. Accessed 12 November 2020.

36. Wood, Thomas. “Softmax Function.” *DeepAI*, 17 May 2019, web.archive.org/web/20201115100835/deepai.org/machine-learning-glossary-and-terms/softmax-layer. Accessed 12 November 2020

37. Kumar, Satyam. “Overview of Various Optimizers in Neural Networks.” *Medium*, Towards Data Science, 9 June 2020, towardsdatascience.com/overview-of-various-optimizers-in-neural-networks-17c1be2df6d5. Accessed 13 November 2020

38. Bushaev, Vitaly. “Adam - Latest Trends in Deep Learning Optimization.” *Medium*, Towards Data Science, 24 Oct. 2018, towardsdatascience.com/adam-latest-trends-in-deep-learning-optimization-6be9a291375c. Accessed 16 November 2020

39. Pykes, Kurtis. “Adam Optimization Algorithm.” *Medium*, Towards Data Science, 6 June 2020, towardsdatascience.com/adam-optimization-algorithm-1cdc9b12724a. Accessed 16 November 2020

40. Diederik P. Kingma and Jimmy Lei Ba. Adam : A method for stochastic optimization. 2014. arXiv:1412.6980v9

41. “Visual Geometry Group.” *Visual Geometry Group - University of Oxford*, www.robots.ox.ac.uk/~vgg/. Accessed 12 October 2020.

42. *ImageNet*, www.image-net.org/. Accessed 12 October 2020.

43. “Image Classifier Using VGG-19 Deep Learning Model in Google Colab Notebook. Dishes Detection.” *LaptrinhX*, LaptrinhX, 23 Aug. 2020, laptrinhx.com/image-classifier-using-vgg-19-deep-learning-model-in-google-colab-notebook-dishes-detection-396031444. Accessed 13 November 2020.

44. Kaushik, Aakash. “Understanding the VGG19 Architecture.” *OpenGenus IQ: Learn Computer Science*, OpenGenus IQ: Learn Computer Science, 26 Feb. 2020, iq.opengenus.org/vgg19-architecture/. Accessed 13 November 2020.

45. Simonyan, Karen & Zisserman, Andrew. “Very Deep Convolutional Networks for Large-Scale Image Recognition.” 4 September 2018. arXiv 1409.1556.

46. “Face Recognition Using Transfer Learning.” *LaptrinhX*, LaptrinhX, 19 May 2020, laptrinhx.com/face-recognition-using-transfer-learning-3526715658. Accessed 13 November 2020.

47. Chaushevska Marija, Dzeroski Saso, Gjoreski Hristijan and Ivica Dimitrovski. *Hierarchical Classification of Diatom Images with Transfer Learning*, 24 Sept. 2020, http://hdl.handle.net/20.500.12188/9475.

48.. C. Szegedy et al., "Going deeper with convolutions," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 1-9, doi: 10.1109/CVPR.2015.7298594.

49. Guan, Qing et al. “Deep convolutional neural network Inception-v3 model for differential diagnosing of lymph node in cytological images: a pilot study.” Annals of translational medicine vol. 7,14 (2019): 307. doi:10.21037/atm.2019.06.29

50. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 2818-2826, doi: 10.1109/CVPR.2016.308.

51. Chollet, François. *Xception: Deep Learning with Depthwise Separable Convolutions*, 4 Apr. 2017, doi:arXiv:1610.02357.

52. A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” 17 April 2017. .arXiv preprint arXiv:1704.04861.

53. Howard, Andrew & Zhu, Menglong & Chen, Bo & Kalenichenko, Dmitry & Wang, Weijun & Weyand, Tobias & Andreetto, Marco & Adam, Hartwig. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. 2017.

54. KC, Kamal & Yin, Zhendong & Wu, Mingyang & Wu, Zhilu. Depthwise separable convolution architectures for plant disease classification. Computers and Electronics in Agriculture. 165. 2019. doi:10.1016/j.compag.2019.104948.

55. K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

56. G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, 2017, pp. 2261-2269, doi: 10.1109/CVPR.2017.243.

57. Ruiz, Pablo. “Understanding and Visualizing DenseNets.” *Medium*, Towards Data Science, 18 Oct. 2018, towardsdatascience.com/understanding-and-visualizing-densenets-7f688092391a. Accessed 12 November 2020.

58. Google, Google Colaboratory, colab.research.google.com/.