Automated Grocery List Generation by Refrigerator Scanning

Course: Internet of Things (IoT)

Semester: VI

Authors:
Siddhi Agrawal – 19UCC015
Ketan Jakhar – 19UCC020
Karan Aditte Singh – 19UCC025

Supervisor: Dr. Abhishek Sharma

Authors' email address: 19ucc015@lnmiit.ac.in , 19ucc020@lnmiit.ac.in, 19ucc025@lnmiit.ac.in

Project Dataset Link: https://github.com/marcusklasson/GroceryStoreDataset
Project GitHub Link: https://github.com/ketan-jakhar/IoT-project

Abstract

In a house, the kitchen and food play an important role in keeping the family together, and amongst all the appliances found there, the refrigerator is of great importance as it keeps the food items intact and fresh. With the innovations in technologies like Artificial Intelligence, the concept of the Internet of Things (IoT) came, that connected diverse appliances together through the internet. Internet Refrigerator is one such IoT device that was innovated with the expectations to make grocery management more convenient. It saves unnecessary cost, food wastage, plans an organized menu and shopping list.

This project uses three types of deep convolutional neural networks (CNN) and compares them for image classification by training and testing the ML models on a grocery datasetchosen from GitHub. Once the image is classified it can be detected also. This techniquegives accurate results and makes the whole RFID scanning process automated and smartand eventually send notifications to an app or dashboard about the list of grocery item tobe replenished. Here we have created a dashboard on NodeRed for the same and have also tried using the MIT Inventor app for app making.

Keywords: Computer Vision, Object Detection, Tensor Flow, RFID Scanning, CNN

1.1 Introduction

Computer Vision is a field of Artificial Intelligence that enables machines to see and process digital images, videos and analyze them deeply. In this field, object detection based on deep learning is a growing field that is a method to locate and identify objects. We have mainly three methods that we can use: Faster R-CNNs, You Only Look Once (YOLO), and Single Shot Detectors (SSDs). In this project, we go for the CNN technique to classify grocery objects. YOLO4 is one of the new techniques that can be used to detect objects and can be implemented as near future scope for this project which we tried to implement and integrate but couldn’t.

The Internet Refrigerators that are already proposed can provide consumers with product information, their nutritional value, consumption history, download recipes, and even suggest the right temperature. In 2002, Whirlpool's refrigerator transformed into a multimedia communications center such that the owner can surf the Internet, receive emails, and connect with phone and in 2003, LG’s Digital Multimedia Side-By-Side Fridge Freezer with LCD Display came having some added features like full internet access, checking on the latest news, weather. The latest attempt is the LG Home Chat showcased in Las Vegas in 2014 by Korea LG. The proposition is that users will be able to text the fridge and ask it about its content via a chat app called Line. It will also feature an in-built camera that takes a picture of its shelves' contents each time its doors are operated which keeps track of refrigerator contents and their freshness once we feed theirdetails. This project also tries to implement such a concept of Internet Refrigerator (RFID Scanning not included) and integrate IoT more in daily lives but with an

automated smart object detection algorithm to support it.

1.2 Problem Formulation

Image classification and Object Detection of Grocery items are performed using three different types of deep convolutional neural networks (CNN) to learn and experiment with deep neural networks and finally form a grocery detection technique that sends notifications related to missing grocery items to the Node-red dashboard or personal email as of a primitive implementation. For this project, the dataset was re-classified into 3 classes of fruits, vegetables, and packages, for more information readme file can be viewed in the dataset link.

1.2.1 Dataset Description

The GitHub repository contains the dataset of natural images of grocery items that were taken with a smartphone camera in different grocery stores as shown below. It has approx. 5000 natural images from 81 different classes of fruits, vegetables, and carton items (e.g.,juice, milk, yogurt). Those are divided into 42 coarse-grained classes, where e.g., the fine-grained classes 'Royal Gala' and 'Granny Smith' belong to the same coarse-grainedclass 'Apple'. For each fine-grained class, there is an image and a product description ofthe item and therefore there is structured labeling of the dataset. The 81 fine-grained classes and their coarse-grained classes can be found in classes.csv in the dataset folder.

1.2.2 Method and Techniques Used

The training and test sets contain 2640 and 2485 images respectively. For a fair evaluation, we use a linear classifier with the learned representation from the different methods. The three different methods used are:

1) **Deep Neural Networks:** CNNs have been the state-of-the-art models in image classification ever since AlexNet achieved the best classification accuracy. When adapting pre-trained CNNs to new datasets, some previous results have revealed that if we use it directly as a feature extractor, or use the off-the-shelf features, or fine-tune it, it has a better result. Fine-tuning a CNN involves adjusting the pre-trained model parameters which can be done using SVM (Support Vector Machine), such that the network can classify images from a different dataset depending on the size of the new dataset and how similar it is to the previous one.

2) **Variational Autoencoders with only natural images:** Deep generative models, like the variational autoencoder (VAE), have become widely used in the ML community. For efficiency, we use low-level pre-trained features from a CNN as inputs to the VAE which is a probabilistic framework. The latent representations from VAEs are encodings of the underlying factors for how the data are generated.VAEs belongs to the family of latent variable models, which commonly has the form $p\theta(x, z) = p(z)p\theta(x|z)$, where $p(z)$ is a prior distribution(Gaussian) over the latent variables z and $p\theta(x|z)$ is the likelihood(or decoder)over the data x given z.

3) **Utilizing iconic images with multi-view VAEs:** Natural language is the most used modality to aid visual representation learning but its consistency has no guarantee. In our dataset, the product description of a Royal Gala apple explains the appearance of a red apple. But if it is represented with word embeddings, e.g. word2vec, the word 'royal' will be more similar to the words 'king'. Therefore, additional visual information about objects is more beneficial for learning meaningful representations, and in this project, a multi-view VAE is used for this purpose. This model is referred to as variational autoencoder canonical correlationanalysis (VAE-CCA).

The models are compared for their accuracy and then would generate a list of missing grocery items explaining their quantity and description using sensors and RFID (Radio Frequency Identification) scanning (hardware not used in this project but can be used as a future reference for expanding the project scope) and send it to an app for grocery or a dashboard for it to be notified to online grocery website or likewise.

1.2.3 Implementation

As discussed above we have used deep neural networks, such as Alexnet, VGG, DenseNet, as well as deep generative models, such as VAE. Furthermore, a multi-view VAE model is adopted

to make use of the images for each class and show that it improvesthe classification accuracy given the same model.

Libraries Used:

- **Tensorflow:** This framework or open-source library is used as a backend in this project. TensorFlow compiles many different algorithms and models together, enabling the user to implement deep neural networks in image recognition and classification. It functions by implementing a series of processing nodes, each node representing a mathematical operation, with the entire series of nodes beingcalled a "graph". It provides both high and low-level API.

- **Matplotlib:** This library provides an object-oriented API for embedding plots intothe application. We have used this library to generate graphs for the purpose of analysis.

- **Numpy:** An image is essentially an array of pixel values where each pixel is represented by 1 (greyscale) or 3 (RGB) values. NumPy contains and stores n-dimensional array objects and can easily perform tasks such as image cropping, masking, or manipulation of pixel values. It is also used for scientific computing.

- **Keras:** It provides only a high-level API that can use TensorFlow's functions. It isan open-source software library that provides an interface for neural networks andfocuses on being user-friendly, modular, and extensible. It has many implementations of neural building blocks such as layers, optimizer, activation functions, etc., and supports CNN.

- **PIL:** It’s an open-source library for image processing and performs tasks like rescaling, reading, saving in different image formats on an image. It can also beused for Image archives and Image display.

Code Flow:

1) Firstly, all the libraries and dependencies are imported. Then to work with some generators ahead for purposes like preprocessing, and comparing results, we runthe code for the confusion matrix function and create some annotations and set data dimensions. Following this, we mount the google drive and set the train, validation, and test data path and perform the initial preprocessing of data.

2) Now, there are usually four main operations in a CNN, which are Convolution, Activation Functions, Pooling or Sub Sampling, and Classification (Fully Connected Layer).

3) In the first step, the features are extracted from the input image. Small squares of input data are learned in the **Convolution** operation with the help of filters ensuring that the spatial relationship between pixels of a picture sustains. The depth, the stride, and the zero-padding control the feature map.

4) Next, the **Rectified Linear Unit (relu)** operation, is an element-wise non-linear operation that sets zero values for all negative pixel values.

5) Following this, **spatial pooling or subsampling** is used to make the feature dimensions and the number of parameters smaller to make it more manageable andavoid overfitting.

6) Lastly, in the **fully connected layer**, the input image is classified into various classes based on the training dataset. This layer allows the combination of features from the convolutional and pooling layers.

Code Test Runs, Results, and Graphs:

1) **TEST RUN 1:** There is a total of approx. 1500 large images and a sequential CNNthat takes an input image of size 224, 224, with 3 channels is created and in the first iteration, no regularization and augmentation are there to set the base. The model is compiled and fit using an optimizer with a learning rate of 0.0001. The model did not perform properly, so a good technique to begin adding more diverse data to the dataset through data augmentation is adopted.

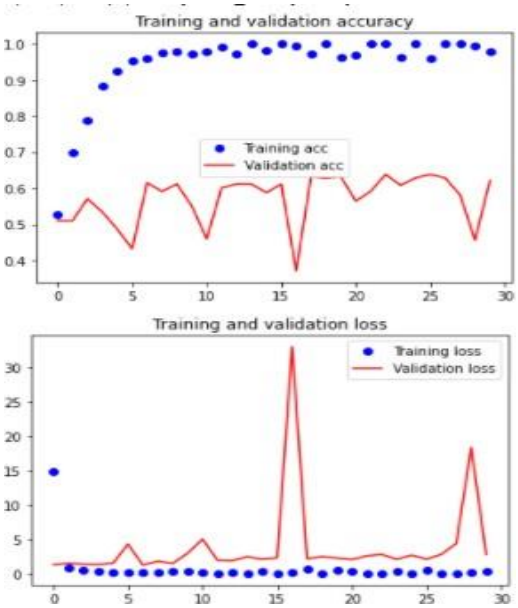
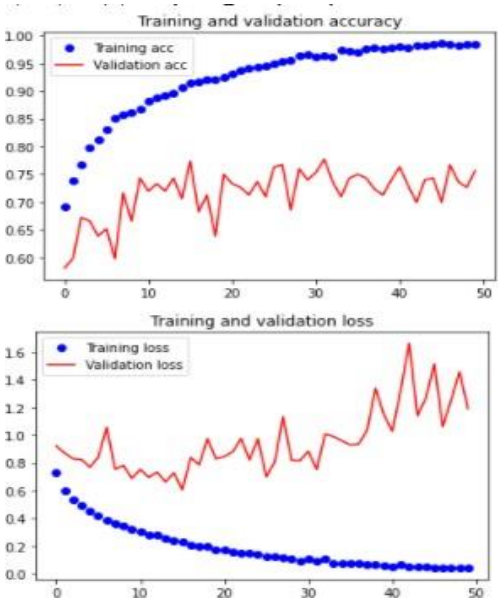
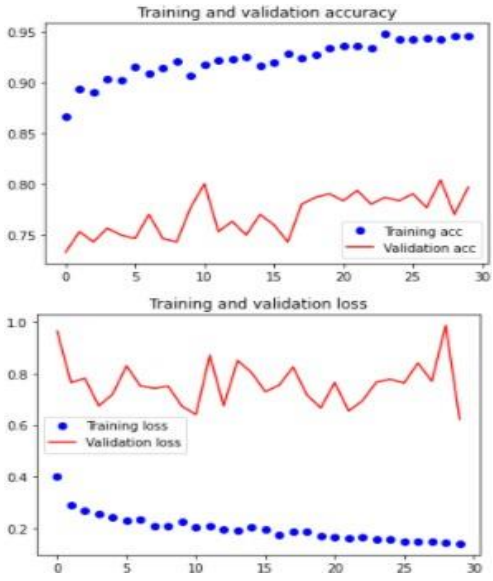


Fig. 1: This is the executed graph of validation and training data accuracy andloss for the base case test run.

2) **TEST RUN 2: Data Augmentation** increases the size of the data set and reduces overfitting. The chosen features are as follows: rescaling, shear_range, zoom_range, and horizontal_flip. The training accuracy results increase but we further improve it by adding more augmentation features.





Data Augmentation (DA) results in Left Fig. 2 and Improved DA results in RightFig. 3. For Test Case 2.

3) Now for **Regularization**, we add a dropout layer to the model to reduce overfitting. The batch size is reduced to 20 and will be trained to 50 epochs. Sincethe results are good as both the training accuracy and loss and validation accuracyand loss are close, we finally make the predictions.

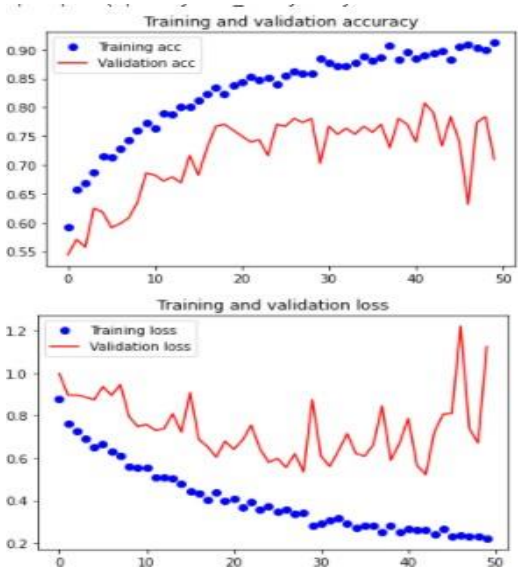


Fig. 4: Accuracy Graph after Regularization

4) **PREDICTION ANALYSIS:** It could be seen from the confusion matrix that the packages and fruits were still predicted with decent accuracy but still there were some incorrect predictions. Therefore, this model is improved further by using hyperparameter operations and testing.

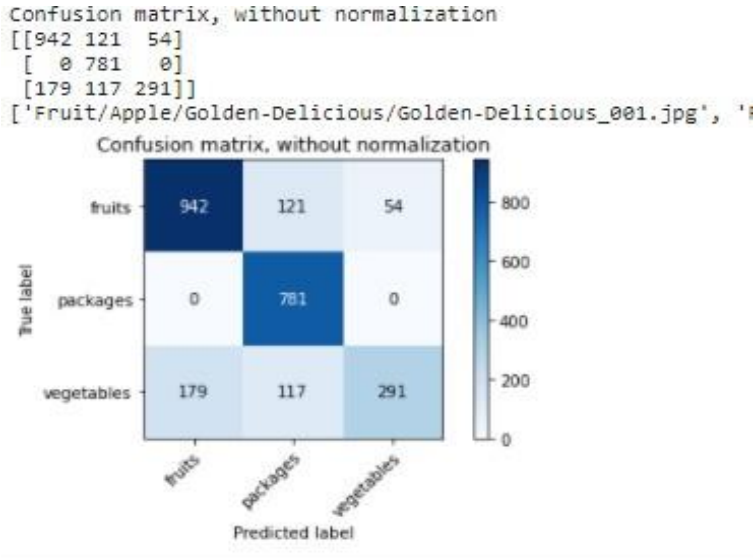
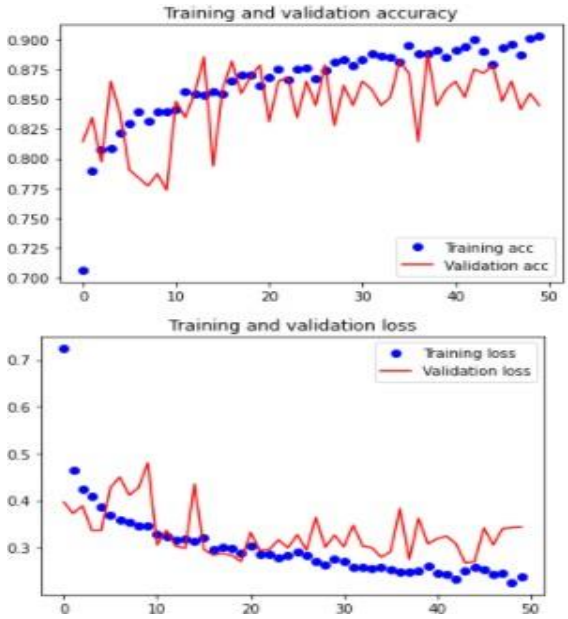


Fig. 5: Confusion Matrixfor First Prediction

5) **Fine-tune pre-defined network:** VGG16 and ResNet50 pre-trained model for transfer learning is used to compare and identify their performance. VGG is a sequential model consisting of only 3x3 convolutional layers that increase in depth while they are stacked on top of each other, followed by two fully connectedlayers, and last by a softmax classifier.



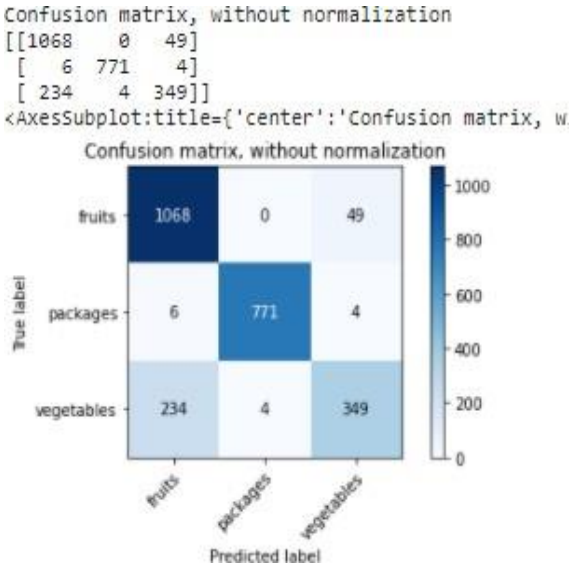
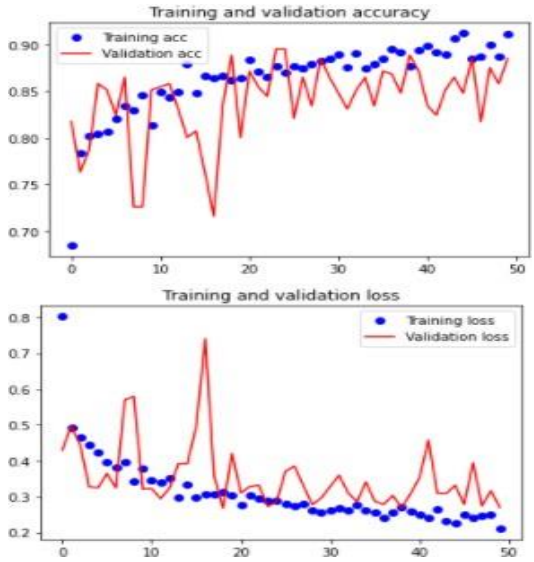
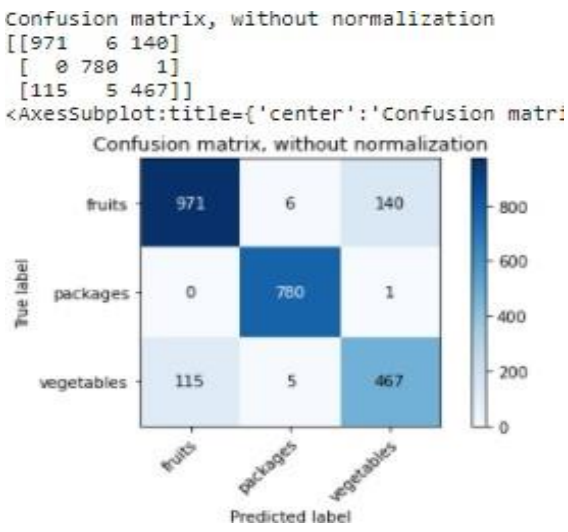


Fig. 6: (Left)VGG16 Accuracy comparison

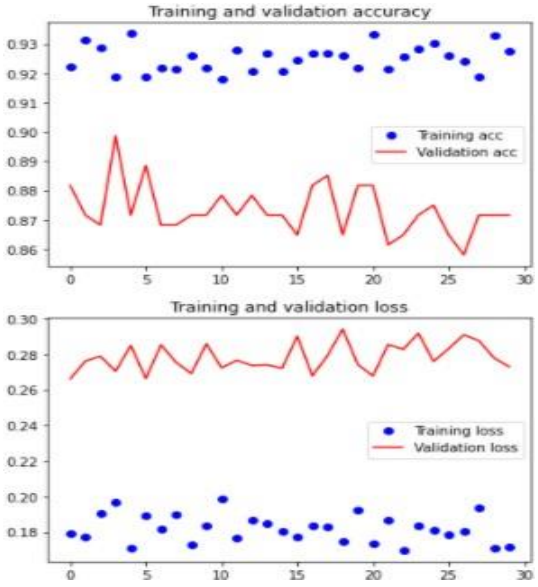
Fig. 7: VGG16 Confusion Matrix

6) Further, the model can be improved by removing the additional dropout layer as the train and validation in the accuracy and loss are quite close to each other. Thevalidation accuracy improves but the prediction results are similar.

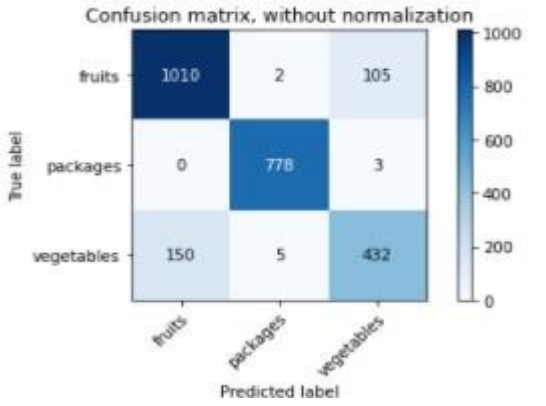




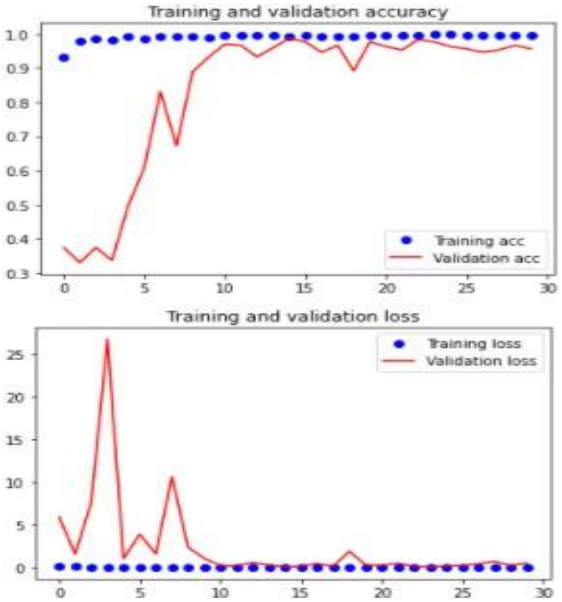
7) **Hyperparameter Tuning:** After changing the learning rate to 1e-5, the results come out to be consistent compared to VGG16 with learning rate 1e-4. However,the prediction did not change much. Thereby leading to the usage of ResNet50.



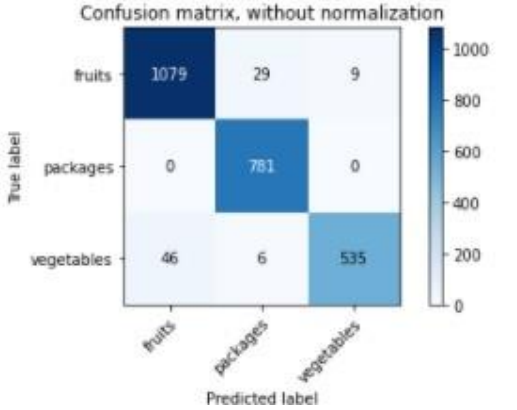

```
Confusion matrix, without normalization
[[1010    2  105]
 [   0  778    3]
 [ 150    5  432]]
<AxesSubplot:title={'center':'Confusion matr
```



8) ResNet is a functional model based on a network-in-network architecture. By using **RestNet50** we repeat the same process and then compile and run the model.In this, the validation results varied during the first 10 epochs but towards the end, the results were consistent. Results are shown below.




```
Confusion matrix, without normalization
[[1079   29    9]
 [   0  781    0]
 [  46    6  535]]
<AxesSubplot:title={'center':'Confusion matr
```



Node-RED linking for notifications:

Node-RED is a programming tool for wiring hardware devices, APIs, and other online services. It provides an online editor that makes it easy to bind flows together using thewide range of nodes in the palette that can be deployed to its runtime. It is used to show how the image is classified and helps us in detection in a handy tool with a dashboard like Node-RED. We need some dependencies or packages to be installed like node-red-contrib-browser-utils and node-red-contrib-image-output and then upload images from test data to check its working visually which can also be done with a webcam. After uploading each image one by one we can see the results of detection.

1.3 Results

When comparing the following accuracy of all the models:

S.No.	Model Name	Validation accuracy	Incorrect classified images
1	Base CNN with Dropout layer(DL)	81%	471/2485
2	VGG16 Transfer model with DL	88%	297/2485
3	VGG16 without DL	89%	267/2485
4	VGG16 with changed learning rate	92%	265/2485
5	ResNet50	96%	90/2485

The ResNet50 proved to be the best model in terms of both accuracy and loss. The accuracy validation results ranged around 96% while the confusion matrix described only one image (fruits) classified incorrectly. In conclusion, the problem of classifying different groceries at a high level can be solved using multiple convolutional neural networks such as the ResNet50. With a good, sufficient dataset, this network can definitely be used in real-life situations, the same cannot be said for the VGG models because the accuracy rates were not good enough.

1.4 Conclusion

We have seen different CNN ML models used for classification and identification of grocery products from the dataset and their prediction which concludes that ResNet50 is the best approach to do so. Also, we see that through connecting tools like Node-RED we can develop an easy way to connect our models and smart hardware to software for enhancing the usage of automation and smart accessibility in daily lives by developing simple dashboards or even apps using tools like MIT Inventor or simply Android Studio. We also tried using MIT Inventor App but couldn't succeed. Also, such technologies and integrated projects have a very good future scope in terms of accurate and new Object Detection(YOLO4) techniques being used, deployed in smart devices like Internet refrigerators with the help of hardware like Sensors and RFID Scanning technique and then finally making it robust, easy to use and accessible by integration git with software and apps.

Acknowledgments

We could successfully complete the project with guidance and assistance from our colleagues and friends. We respect and thank Dr. Abhishek Sharma for allowing us to do this project work and providing insight, expertise, and ideas that greatly assisted the research. We would also like to show our gratitude to the institution and various research papers for their valuable support. We are extremely fortunate for the completion of our project work with the cooperation of our group members.

1.5 Bibliography/References

[1] Marcus Klasson and Cheng Zhang and Hedvig Kjellström. *A Hierarchical Grocery Store Image Dataset with Visual and Semantic Labels*, IEEE Winter Conference on Applications of Computer Vision (WACV), 2019

[2] A. Razavian and Hossein Azizpour and J. Sullivan and S. Carlsson. *CNN Features Off-the-Shelf: An Astounding Baseline for Recognition*, journal 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014.

[3] Folasade Osisanwo, Shade Kuyoro, and Oludele Awodele. *Internet Refrigerator – A typical Internet of Things (IoT),* 3rd International Conference on Advances in Engineering Sciences & Applied Mathematics (ICAESAM'2015) March 23-24, 2015 London (UK)