Automated Shopping Cart

1st Kuntal Gorai
Department of Computer Science
PES University
Bengaluru, India
kuntal0901@gmail.com

2nd S V S C Santosh
Department of Computer Science
PES University
Bengaluru, India
santoshsvsc2002@gmail.com

3rd Skanda S
Department of Computer Science
PES University
Bengaluru, India
skanda2594@gmail.com

4th Vijay Murugan A S
Department of Computer Science
PES University
Bengaluru, India
vijaymurugan1457@gmail.com

***Abstract*—The automated shopping cart aims to reduce the manual labor required for weighing and billing the items a user wishes to buy. In this paper, we build a shopping cart with weight sensors and an android application that identifies the item by scanning it with the help of a smartphone's camera and detecting the weight, making the shopping experience easier and simpler.LaTeX.**

***Index Terms*—Deep Neural Networks, Database, Automation, Weight Sensors, Shopping**

I. Introduction

Our project focuses on building an application - 'Automated shopping Cart' which is a shopping cart with the integration of an Android app and weight sensor for hassle-free shopping. The motivation for this project was to avoid the long queues that a customer has to wait for payment of the items they have added to the shopping cart. The primary features of our application include
a) An efficient deep learning model which can classify the items with very minimal error and faster classification.We aim to use transfer learning with MobileNetV2, which makes it lightweight, enabling it to run on resource-deficient devices and enabling the model to be trained again without the pre-trained saved model. MobileNetV2 has a deficient number of parameters that can be trained, which is also an advantage in our case.
b) An android app that can perform functionalities like classifying the items placed in front of the camera, adding items to a cart, removing items from the cart, and paying the bill for items added to the cart.
c) A load cell acts as a weight sensor and helps customers estimate the weight of the item they are trying to add to their cart.
d) An Arduino board to send the weights detected by the load cell with the android app.
e) Finally a cloud back-end - for which we plan to employ AWS, which will serve as a database to store the details about the items like cost, quantity, and description available in the supermarket, the order history consisting of orders that the customer had bought at the store. While the current market offers shopping carts that make use of items with RF-ID tags we plan to avoid such tedious tasks by making use of image classification with help of deep learning models like MobileNetv2 to classify the item and we plan to integrate the payment section in our android app which helped customers from waiting in a long queue enabling a hassle-free shopping.

Identify applicable funding agency here. If none, delete this.

II. Literature Survey

Up to five research papers were reviewed in context of different models and their architecture, viable hardware and IoT and are summarized in the following paragraphs.

Paper [1] by Jose Luis Rojas-Aranda(B) and others talks about developing a simple lightweight CNN-based model for classifying the 3 types of fruits taken into consideration in this paper, which are apple, banana, and orange. Their main aim is to classify these fruits based on their color and texture, which was implemented by making use of transfer learning with a pre-trained model, i.e. Mobile Net V2. The color was obtained from 3 techniques which were Single RGB Color where the color was sent as a vector of RGB values; RGB histogram where a histogram of all colors is sent and the peak from it is chosen as the color and RGB centroid using K means where the centroid of all colors was chosen. Taking both into consideration, the accuracy was also improved and the model worked very well for predicting the desired class both when the object was placed in plastic bags and when not in plastic bags, thereby proving sufficient for achieving the same. K means was used to get the color of the object. the model showed an accuracy of 93%. The model took into account plastic bags and was computationally inexpensive. The shortcomings include not considering varieties of groceries and needing to have a constant background.

Paper [2] intends to assist people with vision impairments in grocery shopping by giving assistance. This project made use of both generative and deterministic deep neural networks along with a simple linear classification model like the SVM. The main task in this paper was to make use of pre-trained CNNs like Densenet, AlexNet, and VGG16 to extract features and then convert the extracted features into a vector which enables them to be placed onto some classifier which enables classification based on the extracted features. This paper also made use of VAE (Variational AutoEncoder) to train based on natural images. The best accuracy for achieving the task was achieved by making use of DenseNet with an SVM classifier which can be understood due to the increase in the number of trainable layers, thereby leading to an increase in the number of parameters leading to better accuracy. It used a small data set of real life images to train. Because of the vast number of parameters involved, high GPU and CPU capacity, as well as good hardware, are required to execute rapid and accurate classification.

Paper [3] researches faster deployment of deep learning model in resource deficit devices like mobile phones. The model that was used for training was YOLO (You Only Look Once). The model was built on a VOC data set. This data consisted of 20 labeled classes. The network had 19 convolution layers,2 fully connected layers, and a max pool layer for the reduction of dimensions. They used darknet with CUDA, dark flow, and OpenCV after the model was pre-trained to make the detection happen in real-time. The pre-trained model's initial weights are saved in a.weights file.. Next, they made use of Darkflow to convert this weight file to a protobuf file which is mobile device compatible. Once converted it is deployed into the mobile application. Finally, to be able to detect and classify images successfully, they made use of the TensorFlow module present in Android. Tensorflow module makes use of three functionalities Classify, Stylize and Detect. TensorFlow Made use of two more files i.e. .so (shared object) file and .jar file. These two files are built using Bazel. Once all the files are available in the Android Studio package, they can be deployed in real-time.The object detection model was able to detect and classify object in mobile device within fraction of seconds. But generation of protobuf file is quite challenging. Incompatibilities with versions of python and OpenCV. Moving a huge neural network to mobile devices and making device understand the deep layers of network is very difficult.

Paper [4] deals with how an Android application can be built from scratch with the help of Android Studio, which is Android's official IDE. It helps one build the highest quality applications for every android device providing extensive tools to help test the app, making it bug-free. The app developed here was an app for people to order groceries online. With a user-friendly GUI, the aim was to make sure anybody could easily order grocery items. The rest of the paper dealt with the step-by-step development of the described app. The application consisted of activities helping with navigation and product scanning, enabling users to browse through products and choose the ones they need. The categorization of grocery products is aimed at making the search for items easier than manually searching for them. The UUID was used to connect the smart card with the app and the UID of the items that were scanned. Upon testing the workings of the app, a success rate of 100 percent was obtained. Based on the commands received, the shopping cart was able to move in the preferred directions. At no point during the process did the application crash. There is 100% success rate in each direction of the motion control. However This application is not developed for iOS. The UID of all the items must be scanned.

Paper [5] Automatic Load Detector Design to Determine the Strength of Pedestrian Bridges Using Load Cell Sensor Based on Arduino developed a system that can test both static and dynamic loads on a surface using load cells. The load cell hx711 was used along with an arduino uno. This system The system was measured to give outputs with up to 8.5% error, which is not negligible.

III. MOTIVATION/SCOPE

Shopping malls are visited by a large number of people every day, and this causes huge queues for billing. To avoid this systems such as bar codes and RF ID tags have been used to reduce billing times. This system has shown a reduction in billing times, but the problem prevails. We use an image classification model trained on a custom data set to classify and bill grocery items like fruits and vegetables as and when they are added to the cart such that the final bill can be generated at the click of a button and hence eliminates billing time. Customers' purchases are processed by cashiers or self-service checkout equipment at retail establishments. The checkout process has already been sped up because most items have scannable barcodes. Fruits and vegetables, however, are frequently treated in diverse ways. The consumer or the cashier must manually determine the category of the item being purchased and search for it in the system. In order to assess its viability for such an application, we provide a first strategy for fruit and vegetable categorization with that use in mind.

IV. PROPOSED METHODOLOGY

A complete collection of images of fruits and vegetables was not readily available and was constructed. Multiple sources were referred and only the best were chosen to be included. Selenium was used to script and to scrape relevant images from websites. Various data augmentation techniques such as rotation of images by some degree, zoom in were used to expand the data set for better results. The downloaded images had different formats some of which might not be acceptable to be taken in as input to the models or did not have RGB pixels for all images so necessary conversions to .jpeg, .jpg, .png had to be done in addition to filter junk images coming out of google image search. A data set of 50 classes consisting of a total of 153k images was constructed.

After extensive research on what models to use, the ones giving the best accuracy were selected and trained with augmented and non-augmented data. EfficientNet, NASNet, and MobileNet-v2 were the models giving better accuracies and were selected. Ensemble learning was applied to the mentioned models.

Reading the input weights of the food placed in the cart is necessary. A Load Cell was used along with an Arduino-UNO to read the weight of the items in the cart. Each load cell was connected to one corner of the cart and calibrated using the micro-controller to be able to read the weights. The weights read by the load cell is sent to the Android app. Users can connect to the cart using Bluetooth and start shopping with the feature "Connect to Cart" provided in the main menu of our Android application.

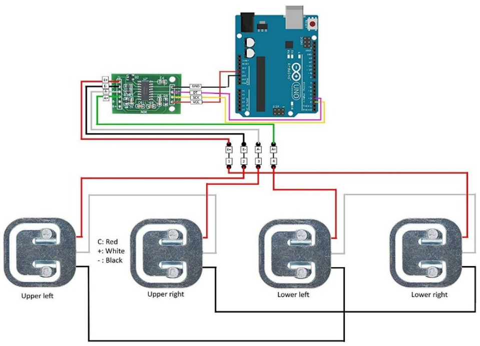

Fig. 1. Circuit Diagram

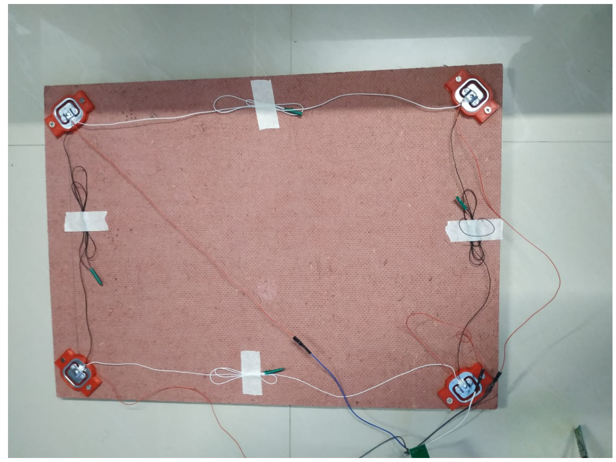

Fig. 2. View of the load cells

Fig. 3. Top View connected with the microcontroller

When a force, such as tension, compression, pressure, or torque, is transformed into a load cell, an electrical signal that can be measured and standardised is generated. Specifically, a force transducer. According to the force on the load cell, the electrical signal changes. It operates under the theory of a "Wheatstone Bridge." The sensor's body is often made of stainless steel or aluminium, which gives it two vital qualities: (1) the capacity to withstand enormous loads; and (2) the flexibility to gently bend and then return to its original shape when the force is removed.

By gently deforming when force (tension or compression) is applied, the metal body acts as a "spring." It returns to its original form unless it is overloaded. Due to the strain gage's changing shape and the flexure's deformation, a Wheatstone Bridge circuit produces a differential voltage fluctuation and electrical resistance. As a result, the voltage change and the physical force applied to the flexure are inversely related.

The database was setup on Firebase. Login authentication was setup on Firebase and database connections was setup and linked to android studio where the android app was developed. The prices of items are stored in the database which is fetched once a day in-case the prices vary. Once the item is detected, the price is calculated based on the weight fetched by the weight sensor. Once the user is starts shopping, they can scan and add items to cart by clicking on the scan icon provided. Upon completion, they can checkout and complete the payment.

Various pages are given on the app which include "User Profile Page" where users can edit their profiles. Instructions on how to use the cart and the app is given in the "Instructions Page". Users can view their past orders in their profile page which shows all the previous orders placed.

V. Experimental Results and observations

For measuring the performance and narrowing down on the ones that gave the best accuracy, graphs were plotted against training steps and accuracy and against loss and training steps. As the training steps increased accuracy of training and validation sets converged to the higher accuracy. The models which gave the best accuracy were EfficientNet, NASNet, and MobileNet-v2. The observations made while training are shared below

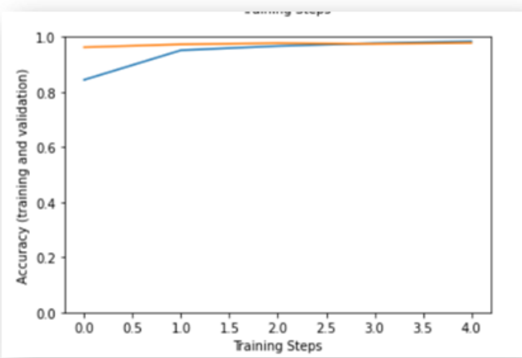

Fig. 4. EfficientNet V2 Training and Validation accuracy

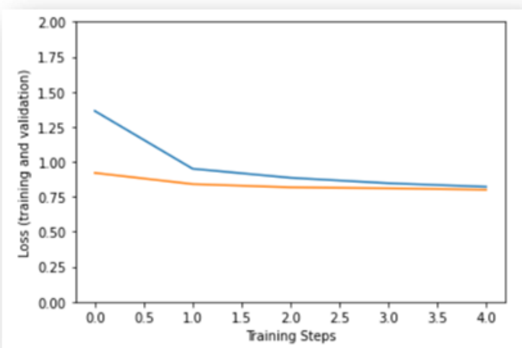

Fig. 5. EfficientNet V2 Training and Validation loss

For better accuracy, models trained over augmented data were also considered and the best three were selected to proceed with Ensemble learning. Each model's accuracy doubles as the weight of its prediction and the final prediction is the weighted average of outputs of all three models.

VI. Conclusion

This paper just put out one way of building a model from scratch using pretrained models for classifying groceries using image recognition. The final ensemble model uses EfficientNet

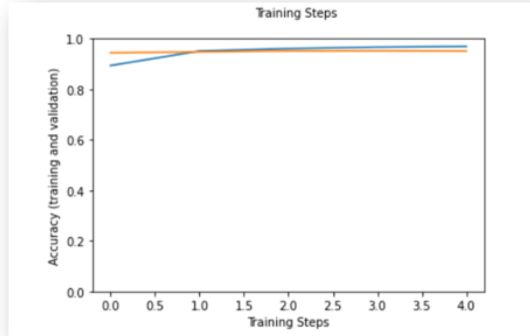

Fig. 6. MobileNet V2 Training and Validation accuracy

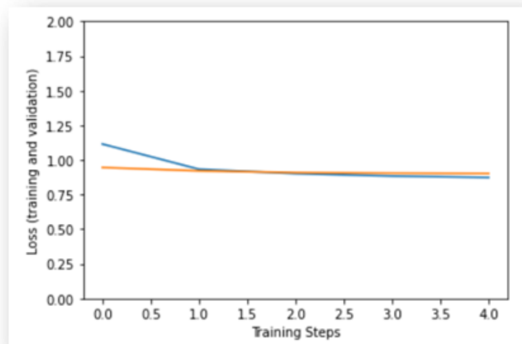

Fig. 7. MobileNet V2 Training and Validation loss

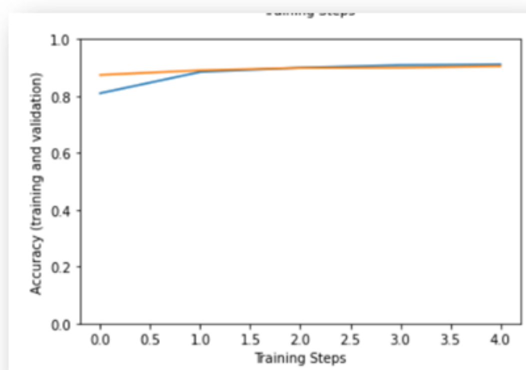

Fig. 8. NasNet Training and Validation accuracy

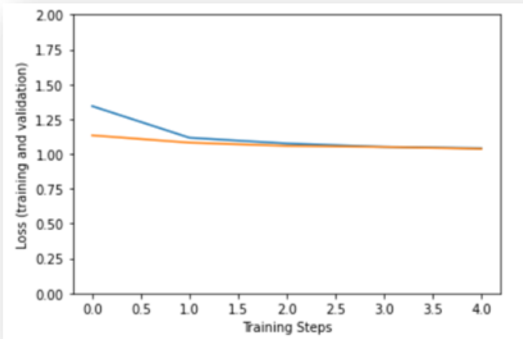

Fig. 9. NasNet Training and Validation loss

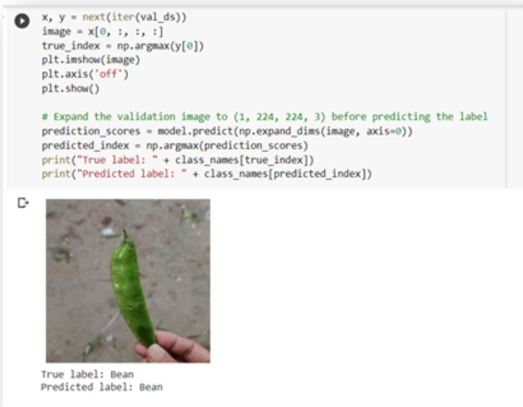

Fig. 10. An example prediction

with augmentation, MobileNet V2 with augmentation and NasNet. MobileNet V2 shows an accuracy of 96.5, EfficientNet shows an accuracy of 97.67 and NasNet shows an accuracy of 96.4. The overall model takes weighted average of out of all three models. With the help of the app built, the shopping experience of the users is seamless, fast and hassle free. The best way to check if the model is working as expected is to give the user option to challenge the output given by the model to ensure that they are correct.

REFERENCES

[1] JL. Rojas-Aranda, JI. Nunez-Varela," Fruit Classification for Retail Stores Using Deep Learning ", Springer, June 2020

[2] M. Klasson, C. Zhang, H. Kjellstrom, "A Hierarchical Grocery Store Image Dataset with Visual and Semantic Labels" Arxiv, January 2020.

[3] A. Zaina, D. Syed," Deployment of Deep Learning Models on Resource-Deficient Devices for Object Detection ",Department of Electrical Computer Engineering Texas AM University College Station, Texas, U.S.A., IEEE, May 2020.

[4] SB. Dubey, GM. Kadam, O. Angane," Android Application for Grocery Ordering System", Irjet, April 2021

[5] K. Riyanti, I. Kakaravada, A.Ahmed, " An Automatic Load Detector Design to Determine the Strength of Pedestrian Bridges Using Load Cell Sensor Based on Arduino ", IJEEEMI, February 2022