PROJECT REQUIREMENTS SPECIFICATION

Automated Billing cart

UE19CS390A – Project Phase – 1

Submitted by:

Name	SRN
Kuntal Gorai	PES2U1G9CS198
S V S C Santosh	PES2UG19CS346
Skanda S	PES2UG19CS391
Vijay Murugan A S	PES2UG19CS454

Under the guidance of

Prof. Prajwala T.R

PES University

January - May 2022

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

FACULTY OF ENGINEERING

PES UNIVERSITY

(Established under Karnataka Act No. 16 of 2013)

Electronic City, Hosur Road, Bengaluru – 560 100, Karnataka, India

PES UNIVERSITY

PROJECT REQUIREMENTS SPECIFICATION

TABLE OF CONTENTS

1. Introduction 3
    1.1 Project Scope and Motivation 3
2. Literature Survey or Existing System 3-5
3. Product Perspective 5-6
    3.1 Product Features 5
    3.2 User Classes and Characteristics 5
    3.3 Operating Environment 5
    3.4 General Constraints, Assumptions and Dependencies 6
    3.5 Risks 6
4. Functional Requirements 6
5. External Interface Requirements 6-7
    5.1 User Interfaces 6
    5.2 Hardware Requirements 7
    5.3 Software Requirements 7
    5.4 Communication Interfaces 7
6. Non-Functional Requirements 7
    6.1 Performance Requirements 7
    6.2 Safety Requirements 7
    6.3 Security Requirements 7
7. Feasibility Study 8
8. Other Requirements 8

Appendix A: Definitions, Acronyms and Abbreviations 8

Appendix B: References 8-9

1. Introduction

- We are all forced to go through the tedious task of getting our items billed at shopping marts which have become an integral part of city life which is one choke point that every customer has to pass through.
- We aim to make the whole experience of billing items purchased easier and faster, trying to save a lot of the customers' time since each customer takes a variable amount of time in getting their items billed the conventional way.

1.1. Project Scope

Develop a fully working application which generates a final bill of all the items purchased by the customer without the hassle of standing in huge lines to get them billed the conventional way thereby, saving a lot of time and making it feasible for anyone with a smartphone to use the app.

2. Literature Survey or Existing System

A. Fruit Classification for Retail Stores Using Deep Learning

- **Summary:** This Paper proposed a MobileNetV2 based model to develop an efficient model to detect the Fruits on the basis of their shape and texture.To add in the context of color an additional feature i.e was taken for identifying it much better in cases of ambiguity. K-means was used to allow the model to internally produce the K colors, and then, these are concatenated at the end of the process. At the end softmax was used in order to give the final classification on the basis of probabilities.
- **Journal:** Springer, June 2020
- **Model:** MobileNetV2
- **Shortcomings**: Does Not Take into consideration the multiple varieties and the model detects objects only when there is a constant background.

B. A Hierarchical Grocery Store Image Dataset with Visual and Semantic Labels

- **Summary:** This Paper talks about the different models which can be used for classifying vegetables and fruits. This made use of a technique called transfer learning where a model is built on a larger dataset of images and these weights are now used as weights of our model and we fine tune it to classify our data by running a few epochs. Use of heavy computational models like VGG16, Densenet was used in order to convert images sent to the model thereby giving a feature vector. This feature vector has further been placed on to linear space using some classifiers like SVM the classification was achieved.
- **Journal:** Arvix/Research Gate, January 2020
- **Model:** DenseNet, VGG16, SVM

- **Shortcomings:** The Models used in this paper make use of a high amount of computational resources and need a lot of time for training.

C. Deployment of Deep Learning Models on Resource-Deficient Devices for Object Detection

- **Summary:** This paper talks about deployment of deep learning models on mobile devices which have less resources compared to highly equipped devices. The first step is to make a predefined training model on a highly equipped device.Here the trained model is tiny yolov2. Once done it needs to be sent to the android studio this can be done with the help of protobuf file generation. Protobuf file generation can be done with the help of Darkflow and OpenCV. Darknet is also used to classify the labels to the images.Next step is to use the tenserFlow module to help classify images in our app. The tensorflow uses three main functions that are classify,stylize,and detect. Adding .so and .jar files will help tensorflow detect ability in mobile apps. The .so and .jar file is built using bazel. Once all the files are available and compiled it is ready to be used for real time detection.
- **IEEE Paper:** May 2020
- **Results:** The proposed model on mobile was able to achieve 66.3 mAP with a speed of 21 frames per second.
- **Modules used:** Tensorflow,Darkflow,Darknet,Bazel
- **Shortcomings:** Generation of protobuf file was quite challenging, Incompatibilities between python and openCV ,Doesn't take care of object detection when mobile devices move at higher speed,object detection under low illumination.

D. Android application for Grocery Ordering system.

- **Summary:** This paper talks about the technique followed to develop a user-friendly application for people to shop groceries online. The design of the application followed by the views and implementation has been discussed in the paper concluded by the results obtained after testing the application on the device.
- **Journal:** IRJET, April 2020
- **Shortcomings:** The application was developed only for android devices and unavailable on iOS. The unique ID of all items must be scanned which is a tedious and time consuming task.

E. Automatic load design detector

- **Summary:** This paper talks about using load cells to measure static and walking load on a bridge and using these readings to measure the strength

of the bridge.. The system was able to measure the weights dynamically in real-time and display it on an LCD screen.

- **Journal:** IRJEEEMI, February 2020
- **Shortcomings:** system was measured to give outputs with upto 8.5 ~ 9 % error, averaging to 4~5% overall.

3. Product Perspective

In the day-to-day lives of people who visit shopping marts to purchase their needs, they spend an enormous amount of time waiting in long queues to get the billing done the conventional way. Hence, with the help of this app we aim to cut down the time spent by the customers in queues and make the whole shopping experience better.

3.1. Product Features

1. Faster detection of items by our deep learning models implemented.
2. Update of the weight of item added to cart through load cell to our app via arduino/raspberry pi
3. User friendly app to perform hassle free shopping..

3.2. User Classes and Characteristics

1. User

   Show items to our app to detect it.

   Add items to our cart containing load cell

   Verify the quantity and press add button on arduino

   Verify the items with items on the app

   Pay the final item through various payment gateways

2. Admin

   Make updates to the item based on stock available in mart

   Add users to database based on the user details sent to app

3.3. Operating Environment

- Hardware Platform - Mobile device, Arduino/Raspberry Pi
- Operating system - Android
- Software Components - Inputs include user real-time image capturing and weight from load cell. Outputs include updation of cart items and bill generation.

3.4. General Constraints, Assumptions and Dependencies

- Hardware constraints:
  - Ability of device to run the model,
  - Inaccuracy of load cell,
  - Training requires powerful GPUs.
- Software constraints:
  - Time optimization of algorithms used to classify images
  - Time to update the bill in the app.
  - A customer shows a cheaper product and places the expensive product into the cart i.e., fooling the system.
- Assumptions:
  - Objects are placed in a transparent bag
  - Final Payment is handled by some 3rd party interface.
  - Customer has an android phone with minimum computational resources to use the app.
- Dependencies: Keras, TensorFlow, OpenCV, DarkFlow

3.5. Risks

- Model Using up too much computational resources causing the app to crash
- Failure of Hardware

4. Functional Requirements

- Ability of system to classify items shown by user accurately
- Fetch item details and add item to bill
- The app is robust and is always connected to the Arduino when in use.

5. External Interface Requirements

5.1. User Interfaces

- The UI is an android application.
- Functionality to scan and classify objects.
- Functionality to the weight of the objects placed in the cart.
- Screen to view all the items in the cart.

- Screen to view the final computed bill and payment feature.

5.2. Hardware Requirements

- Android platform for the app
- Precise load-cell
- Reliable microcontroller(Arduino/Raspberry Pi)

5.3. Software Requirements

- Android Studio
- TensorFlow,Keras
- Firebase

5.4. Communication Interfaces

- Bluetooth/USB connection to arduino.
- Wired communication between arduino and the load cell.

6. Non-Functional Requirements

6.1. Performance Requirement

- Ability of the model to classify items within 3-4 seconds.
- Ability to deploy the model on resource deficient systems.
- Seamless performance of app without any lag.

6.2. Safety Requirements

- Take consent from user when connecting microcontroller to his device

6.3. Security Requirements

- A login authorization system using only a mobile number/email is done.
- Ensure the database can be modified by the authorized user/admin.

7. Feasibility Study

Upon referring to a few research papers in the field, YOLO(You Only Look Once), VGG19, RESNET, and ALEXNET, DenseNET are some available options, however they require a lot of computing power and optimizing them will be time-consuming because of the various parameters they come with.

Reflecting on the changes made to the cart on the app would be a tedious task.

Our model doesn't take into consideration those items already packed having a barcode on them.

8. Other Requirements

- Need a battery to power up the arduino and weight cell.

Appendix A: Definitions, Acronyms and Abbreviations

Transfer Learning: Transfer learning (TL) is a research problem in machine learning (ML) that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem.

VGG: Visual Geometry Group.

Appendix B: References

- Fruit Classification for Retail Stores Using Deep Learning by Jose Luis Rojas-Aranda(B), Jose Ignacio Nunez-Varela, J. C. Cuevas-Tello, and Gabriela Rangel-Ramirez
  link: https://link.springer.com/chapter/10.1007/978-3-030-49076-8_1
- A Hierarchical Grocery Store Image Dataset with Visual and Semantic Labels by Marcus Klasson, Cheng Zhang, Hedvig Kjellstrom
  Link: https://arxiv.org/abs/1901.00711
- Deployment of Deep Learning Models on Resource-Deficient Devices for Object Detection by Ameena Zaina , Dabeeruddin Syed
  Link: https://ieeexplore.ieee.org/document/9089651
- Android Application for Grocery Ordering System by Shubham B Dubey1, Gaurav M Kadam2, Omkar Angane3
  Link: https://www.irjet.net/archives/V8/i4/IRJET-V8I4678.pdf
- Automatic Load Detector Design to Determine the Strength of Pedestrian Bridges Using Load Cell Sensor Based on Arduino by Kurnia Paranita Kartika Riyanti, Ismail Kakaravada, and Abdussalam Ali Ahmed
  Link: https://doi.org/10.35882/ijeeemi.v4i1.3
- https://ieeexplore.ieee.org/abstract/document/9089651
- https://www.irjet.net/archives/V8/i4/IRJET-V8I4678.pdf
- https://arxiv.org/pdf/1901.00711.pdf
- https://link.springer.com/chapter/10.1007/978-3-030-49076-8_1
- https://medium.com/@BdourEbrahim/how-to-integrate-machine-learnng-models-into-your-app-b77717e2702

- ihttps://www.geeksforgeeks.org/how-to-download-google-images-using-python/